ABSTRACT

QOS PROVISIONING IN MULTIMEDIA STREAMING

by
Hong Zhao

Multimedia consists of voice, video, and data. Sample applications include video conferencing, video on demand, distance learning, distributed games, and movies on demand. Providing Quality of Service (QoS) for multimedia streaming has been a difficult and challenging problem. When multimedia traffic is transported over a network, video traffic, though usually compressed/encoded for bandwidth reduction, still consumes most of the bandwidth. In addition, compressed video streams typically exhibit highly variable bit rates as well as long range dependence properties, thus exacerbating the challenge in meeting the stringent QoS requirements of multimedia streaming with high network utilization. Dynamic bandwidth allocation in which video traffic prediction can play an important role is thus needed.

Prediction of the variation of the I frame size using Least Mean Square (LMS) is first proposed. Owing to a smoother sequence, better prediction has been achieved as compared to the composite MPEG video traffic prediction scheme. One problem with this LMS algorithm is its slow convergence. In Variable Bit Rate (VBR) videos characterized by frequent scene changes, the LMS algorithm may result in an extended period of intractability, and thus may experience excessive cell loss during scene changes. A fast convergent non-linear predictor called Variable Step-size Algorithm (VSA) is subsequently proposed to overcome this drawback. The VSA algorithm not only incurs small prediction errors but more importantly achieves fast convergence. It tracks scene changes better than LMS. Bandwidth is then assigned based on the predicted I frame size which is usually the largest in a Group of Picture (GOP). Hence, the Cell Loss Ratio (CLR) can be kept small. By reserving bandwidth at least equal to the predicted one, only prediction errors need to be buffered. Since

the prediction error was demonstrated to resemble white noise or exhibits at most short term memory, smaller buffers, less delay, and higher bandwidth utilization can be achieved. In order to further improve network bandwidth utilization, a QoS guaranteed on-line bandwidth allocation is proposed. This method allocates the bandwidth based on the predicted GOP and required QoS. Simulations and analytical results demonstrate that this scheme provides guaranteed delay and achieves higher bandwidth utilization.

Network traffic is generally accepted to be self similar. Aggregating self similar traffic can actually intensify rather than diminish burstiness. Thus, traffic prediction plays an important role in network management. Least Mean Kurtosis (LMK), which uses the negated kurtosis of the error signal as the cost function, is proposed to predict the self similar traffic. Simulation results show that the prediction performance is improved greatly as compared to the LMS algorithm. Thus, it can be used to effectively predict the real time network traffic.

The Differentiated Service (DiffServ) model is a less complex and more scalable solution for providing QoS to IP as compared to the Integrated Service (IntServ) model. We propose to transport MPEG frames through various service classes of DiffServ according to the MPEG video characteristics. Performance analysis and simulation results show that our proposed approach can not only guarantee QoS but can also achieve high bandwidth utilization. As the end video quality is determined not only by the network QoS but also by the encoded video quality, we consider video quality from these two aspects and further propose to transport spatial scalable encoded videos over DiffServ. Performance analysis and simulation results show that this can provision QoS guarantees. The dropping policy we propose at the egress router can reduce the traffic load as well as the risk of congestion in other domains.

QOS PROVISIONING IN MULTIMEDIA STREAMING

by
Hong Zhao

A Dissertation
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy in Electrical Engineering

Department of Electrical and Computer Engineering

May 2004

APPROVAL PAGE

QOS PROVISIONING IN MULTIMEDIA STREAMING

Hong Zhao

Dr. Nirwan Ansari, Dissertation Advisor — Date
Professor of Electrical and Computer Engineering, NJIT

Dr. Yun. Q. Shi, Co-Advisor — Date
Professor of Electrical and Computer Engineering, NJIT

Dr. Richard Haddad, Committee Member — Date
Professor of Electrical and Computer Engineering, NJIT

Dr. Sirin Tekinay, Committee Member — Date
Associate Professor of Electrical and Computer Engineering, NJIT

Dr. Anthony Vetro, Committee Member — Date
Senior Principal Technical Staff of Mitsubishi Electric Research Lab, Mitsubishi Research Lab

BIOGRAPHICAL SKETCH

Author: Hong Zhao

Degree: Doctor of Philosophy

Date: May 2004

Undergraduate and Graduate Education:

- Doctor of Philosophy in Electrical Engineering,
  New Jersey Institute of Technology, Newark, NJ, 2004
- Master of Science in Electrical Engineering,
  Xian Jiaotong University, Xian, P. R. China, 1991
- Bachelor of Science in Electrical Engineering,
  Taiyuan University of Technology, Taiyuan, P. R. China, 1986

Major: Electrical Engineering

Presentations and Publications:

H. Zhao, N. Ansari and Y. Q. Shi, "Delay guaranteed bandwidth allocation for real time video delivery," accepted by *IEE Proceedings on Communications.*

N. Ansari, H. Liu, Y. Q. Shi and H. Zhao, "Dynamic bandwidth allocation for VBR video transmission," *Journal of Computing and Information Technology*, vol. 4, pp. 309-317, Nov. 2003.

H. Zhao, N. Ansari and Y. Q. Shi, "Efficient predictive bandwidth allocations for real time videos," *IEICE Transactions on Communications*, vol. E-86-B, No. 1, pp. 443-450, Jan. 2003.

N. Ansari, H. Liu, Y. Q. Shi and H. Zhao, "On modeling MPEG video traffics, *IEEE Transactions on Broadcasting,* vol. 48, No. 4, pp. 337-352, Dec. 2002.

H. Zhao, N. Ansari and Y. Q. Shi, "Transmission of real time video over IP Differentiated services," *IEE Electronics Letters*, vol. 38, pp. 1151-1153, Sep. 2002.

H. Zhao, N. Ansari and Y. Q. Shi, "Self-similar traffic prediction using Least Mean Kurtosis," *Proc. of IEEE International Conference on Information Technology: Coding and Computing (ITCC03)*, Las Vegas, pp. 352-355, Apr. 2003.

H. Zhao, N. Ansari and Y. Q. Shi, "A fast non-linear adaptive algorithm for video traffic prediction," *Proc. of IEEE International Conference on Information Technology: Coding and Computing (ITCC02)*, Las Vegas, pp. 54-58, Apr. 2002.

H. Zhao, N. Ansari and Y. Q. Shi, "Bandwidth prediction for real time video delivery," *Proc. of the 36th Annual Conference on Information Sciences and Systems (CISS'02)*, Princeton, pp. 527-531, Mar. 2002, pp. 527-531.

H. Zhao, N. Ansari and Y. Q. Shi, "QoS guaranteed MPEG video transmission," submitted to *IEEE International Workshop on Multimedia Signal Processing (MMSP2004)*.

ACKNOWLEDGMENT

I am greatly indebted to my advisor, Dr. Nirwan Ansari, for his support, patience, and advice on doing research, writing a dissertation, and pursuing an academic career. I thank him for his insights and suggestions that have helped shape my research skills. His invaluable guidance encouraged me constantly throughout the preparation of this dissertation work.

A very special thanks goes to my co-advisor, Dr. Yun Q. Shi, who advised and helped me in various aspects of my research. Discussions with him always gave me wonderful hints to problems. I am also very grateful to him for his non-academic help that encouraged me in overcoming difficulties.

Many thanks also go to the other members of my committee: to Dr. Anthony Vetro, both for his detailed comments on my proposal and for writing so many reference letters on my behalf for my job-hunt; to Dr. Richard Haddad, for his help and encouragement first in his classes then in producing this dissertation; to Dr. Sirin Tekinay, especially for her reviewing this dissertation work in such a short time.

I am extremely thankful that as I am completing my tenure at NJIT I can say that I have enjoyed my time here. This is because of all the wonderful people I have been surrounded by. I am grateful to them for all kinds of help, academic and non-academic. It is they who made the daily life in the lab more colorful. Thanks are directed to all friends in the CCSPR lab and ANL lab.

I would like to acknowledge the support provided by the ECE Department as a teaching assistant and the support given by the New Jersey Commission on Higher Education via the NJITOWER project, and the New Jersey Commission via NJCWT as a research assistant.

Lastly, but most definitely not least, I would like to thank and acknowledge my family for their support: to my parents for their love and support; to my husband for his support during the time that I was engaged in this study.

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

LIST OF TABLES

Table **Page**

LIST OF FIGURES

LIST OF FIGURES
(Continued)

Figure **Page**

LIST OF ACRONYMS

ACF:	Autocorrelation Function
ABR:	Available Bit Rate
AF:	Assured Forwarding
AIC:	Akaike Information Criterion
AR:	Auto-Regressive
ARIMA:	Auto-Regressive Integrated Moving Average
ATM:	Asynchronous Transfer Mode
CAC:	Call Admission Control
CBR:	Constant Bit Rate
CBQ:	Class Based Queuing
CIF:	Common Intermediate Format
CLP:	Cell Loss Probability
CLR:	Cell Loss Rate
D-BIND:	Deterministic Bounding Interval Length Dependent
DBA:	Dynamic Bandwidth Allocation
DCT:	Discrete Cosine Transform
DiffServ:	Differentiated Services
DSA:	Dynamic Search Algorithm
DSCP:	Differentiated Service Code Point
EF:	Expedited Forwarding
FARIMA:	Fractional ARIMA
FGN:	Fractional Gaussian Noise
FIFO:	First In First Out
FSA:	Fixed Step Size Algorithm
GOP:	Group of Picture

LIST OF ACRONYMS
(Continued)

IETF:	Internet Engineering Task Force
IntServ:	Integrated Service
IP:	Internet Protocol
ISDN:	Integrated Services Digital Network
ISP:	Internet Service Provider
ISO:	International Organization for Standardization
ITU:	International Telecommunication Union
LMK:	Least Mean Kurtosis
LMS:	Least Mean Squares
LRD:	Long Range Dependence
MPEG:	Moving Picture Experts Group
MSE:	Mean Square Error
MMPP:	Markov Modulated Possion Process
NTSC:	National Television System Committee
PAL:	Phase Alternating Line
PDF:	Probability Density Function
PHB:	Per Hop Behavior
PQ:	Priority Queuing
QCIF:	Quarter CIF
QoS:	Quality of Service
QDBA:	QoS Guaranteed Dynamic Bandwidth Allocation
RCBR:	Renegotiated Constant Bit Rate
RSVP:	Resource Reservation Protocol
RTP:	Real Time Protocol
RTSP:	Real Time Streaming Protocol

LIST OF ACRONYMS
(Continued)

SIP:	Session Initial Protocol
SLA:	Service Level Agreement
SNR:	Signal to Noise Ratio
SBA:	Static Bandwidth Allocation
TCP:	Transmission Control Protocol
UDP:	User Datagram Protocol
UPs:	User Parameters
VBR:	Variable Bit Rate
VSA:	Variable Step-size Algorithm
VC:	Virtual Connection
VP:	Virtual Path
VoD:	Video on Demand
WFQ:	Weight Fair Queuing

CHAPTER 1

INTRODUCTION

Multimedia traffic has become one of the major traffic to be transported through networks. Multimedia applications such as video conferencing, video on demand, distance learning, distributed games, and movies on demand, are cluttering different distribution networks with a tremendous amount of traffic. These applications require high bandwidth and stringent Quality of Service (QoS). Provisioning QoS for multimedia streaming has been a difficult and challenging problem. When video traffic is transported over a network, the end video quality is determined not only by the network QoS but also by the encoded video quality. We thus need to consider QoS from these two aspects: video processing and network QoS provisioning to support transporting multimedia traffic over networks.

1.1 Quality of Service

Intuitively, QoS represents qualities like how fast data can be transfered, how much the receiver has to wait, how accurate the received data are, and how much data are likely to be lost. The International Organization for Standardization (ISO) defines QoS as a concept for specifying how good the offered networking services are. Thus, typical metrics used for QoS include:

- The available bandwidth
- The end-to-end transfer delay
- The data delay variation
- The data loss ratio

Measuring video quality is a complex undertaking, as video quality depends on:

- Human spatial temporal contrast sensitivity
- Variability in the frame rate
- Variability in the distortion factor
- Peak Signal to Noise Ratio

Quality assessment is far more complicated when it comes to transmission of video, since the end quality of video is determined not only by network QoS parameters but also by the encoded video quality. The encoded video quality is completely determined by the video encoder and its parameters such as quantizer, frame resolution, and frame rate, while the decoded video quality depends on the same encoder parameter selection and the networking QoS parameters.

QoS is often perceived/measured differently in the video and networking communities. In the networking community, QoS is measured in terms of delay, jitter, packet losses, and bounds on delay. As a result, the design goal from the network's perspective is to meet negotiated QoS guarantees as well as to maximize the network utilization. In contrast, the designer of a video system is considered with maximizing decoded video quality, which is clearly affected by the negotiated network transmission parameters. There is a trade off between the two.

1.2 Video Compression

When multimedia traffic is transported over a network, video traffic consumes most of the bandwidth. Compression is usually employed to reduce the bandwidth requirements and achieve transmission efficiency. Organizations such as ISO/IEC and ITU (International Telecommunication Union) have proposed several standards for video compression.

1.2.1 MPEG

MPEG, which stands for the Moving Picture Experts Group, is an ISO/IEC working group, established in 1988 to develop standards for digital video and audio formats. There are several MPEG standards being used or in development. Each compression standard was designed with a specific application and bit rate in mind, although MPEG compression scales well with increased bit rates.

- MPEG-1

  MPEG-1 was published in 1993 as ISO/IEC 11172 [1]. It is a three-part standard defining audio and video compression coding methods and a multiplexing system for interleaving audio and video data so that they can be played back together. MPEG-1 principally supports video coding up to 1.5Mbps. It is used in video-CD systems for storing video and audio in CD-ROM.

- MPEG-2

  MPEG-2 is an extension of MPEG-1, and is directed toward broadcasting at higher bit rates; it provides extra algorithms for efficiently coding interlaced video, supports a wide range of bit rates, and facilitates multichannel surround coding. The biggest commercial application for MPEG-2 is DVD. DTV (Digital Television) has also adopted MPEG-2 as well.

- MPEG-4

  MPEG-4 has been developed to provide users a new level of interaction with visual contents. It provides technologies to view, access and manipulate objects rather than pixels, with great robustness at a wide range of bit rates. The MPEG-4 natural video standard consists of tools that support applications such as digital television, video streaming, and mobile multimedia.

- MPEG-7

  MPEG-7 is also called the Multimedia Content Description Interface. It

provides a framework for multimedia content processing including content manipulation, filtering and personalization, as well as the integrity and security of the content. While MPEG-4 natural video focused on representation and manipulation of video objects, MPEG-7 focuses on the description of multimedia content. Prior to MPEG-4, the efforts were mainly on coding video information.

- MPEG-21

  Work on this standard, also called the Multimedia Framework, has just begun. MPEG-21 will attempt to describe the elements needed to build an infrastructure for the delivery and consumption of multimedia content, and how they will relate to each other.

1.2.2 H.26x

ITU has completed the following standards:

- H.261

  H.261 is an ITU standard designed for two-way communication over ISDN lines (video conferencing) and supports data rates in multiples of 64Kbps.

- H.262

  Generic coding of moving pictures and associated audio information, same as MPEG-2.

- H.263

  It was designed for low bit rate communication; early drafts specified data rates less than 64Kbps, but this limitation has now been removed. It is expected that the standard will be used for a wide range of bit rates, not just low bit rate applications. It is expected that H.263 will replace H.261 in many applications.

1.2.3 Scalable Video Coding

Scalable compression is useful in today's heterogeneous networking environment in which different users require different rates, resolution, display, and computational capabilities. If the video can be encoded into different layers, receivers can obtain the appropriate video quality based on their available bandwidth. Thus, scalable video encoding has been proposed. There are three types of scalable encoding.

- SNR scalability

  SNR-scalable compression refers to encoding a sequence in such a way that different quality video can be reconstructed by decoding a subset of the encoded bit stream. Pictures are first encoded with coarse quantization in the base layer, and the differences between the reconstructed and original are then encoded in the enhancement layer.

- Temporal scalability

  Temporal scalability in a video stream refers to the property that allows the user to extract from the single bit stream a video sequence at different frame rates (although at the same spatial resolution). The base layer is coded at a lower frame rate. The enhancement layer provides the missing frames to form a video with a higher frame rate. For MPEG video, I and P frames compose the base layer, and B frames form the enhancement layer.

- Spatial scalability

  Spatial scalability in a video stream refers to the property that allows the user to extract from the single bit stream a video sequence at different spatial resolutions (although at the same frame rate). The base layer is coded at a lower spatial resolution. The reconstructed base layer is up sampled to form the prediction for the high resolution in the enhancement layer. The differences

between up sampled decoded base layer and their original picture are encoded in the enhancement layer.

1.3 Network Support

There are several networks coexisted in the Internet. For different networks, different technologies are needed to support transporting video traffic.

1.3.1 ATM Networks

ATM provides several options for transporting video traffic:

- Constant Bit Rate (CBR) service
- Real time VBR (rt-VBR) service
- Renegotiated CBR (RCBR) service
- Available Bit Rate (ABR) service

CBR and rt-VBR services can be easily used. The CBR service provides a constant bandwidth pipe that can be used to support stringent QoS guarantees on the Cell Loss Ratio (CLR), maximum Cell Transfer Delay (maxCTD), and Cell Delay Variation (CDV). The rt-VBR service provides guarantees on the same QoS parameters, but since it optionally uses statistical multiplexing, the resulting guarantees are probabilistic in nature. For adaptive video applications, some researchers suggested using the ABR service, which guarantees a Minimum Cell Rate (MCR) in addition to a CLR. It is assumed in this case that video sources can adapt their rates according to network feedback that is conveyed using an explicit rate based congestion mechanism. Rate adaptation is performed by adjusting the quantization values in the encoding algorithm at the expense of variable video quality. Since VBR-coded video is known to exhibit multiple time scale variations and the bit rate is strongly modulated by scene changes, renegotiated CBR has been introduced to accommodate this behavior. The

bandwidth for a connection can be renegotiated several times during the life time of the connection. This approach promises to provide CBR-like service at a much lower cost. Naturally, RCBR will require a more complicated traffic management than CBR.

1.3.2 IP Networks

The IP-based Internet was not designed to support QoS guarantees. There is a growing interest within the telecommunications industry in providing voice and video services over IP networks. The Internet Engineering Task Force (IETF) has been exploring various possibilities for supporting video over IP. The followings are some proposals made by IETF:

- Real-Time Transport Protocol (RTP)
- Integrated Services (IntServ)
- Differentiated Services (DiffServ)

RTP is the first step toward supporting multimedia over the Internet. It is a session layer protocol that runs on top of the User Datagram Protocol (UDP). This means that RTP is transparent to network routers, and cannot be used to support network-level QoS guarantees. However, it provides several functions that are useful for real time communications. RTP relies on the Real Time Control Protocol (RTCP) to convey various types of information.

IntServ is one of the notable efforts to support QoS over the Internet. IntServ used the Resource Reservation Protocol (RSVP) as a working protocol for signaling in the IntServ architecture. This protocol assumes that resources are reserved for every flow requiring QoS at every router hop in the path between a receiver and a transmitter, using end-to-end signaling, and must maintain a per-flow soft-state at every router in the network. "Soft state" regards the reservation state as cached

information that is installed and periodically refreshed by the end hosts. Unused state is timed out by the routers. If the route changes, the refresh messages automatically install the necessary state along the new route [2]. RSVP uses the soft state method for reservation protocols.

DiffServ is an architecture proposed by IETF that allows different QoS levels to different classes of aggregated traffic flows, as opposed to individual flows. Traffic entering a network is classified and possibly conditioned at the boundaries of the network, and assigned to different aggregate behaviors. Each aggregate behavior is identified by a single Differentiated Service Code Point (DSCP). Within the core of the network, packets are forwarded according to the Per-Hop Behaviors (PHBs) associated with DSCPs.

1.3.3 Protocols for Multimedia Streaming

The Internet has been used primarily for the reliable transmission with minimal or no delay constraints. The TCP/IP protocols were designed for this type of traffic and work very well in this context. However, multimedia traffic, which comprises a significant portion of potential multicast traffic, possesses different characteristics, and hence requires the use of different protocols to provide the necessary services. In addition, the "slow start" TCP congestion control mechanism can interfere with the audio and video "natural" playout rate. Since there is no fixed path for datagrams to flow across the Internet, there is no mechanism for ensuring that bandwidth needed for multimedia is available between the sender and the receiver, and thus quality of service cannot be guaranteed. In addition, TCP does not provide timing information, a critical requirement for multimedia support.

Fig. 1.1 shows a protocol architecture for multimedia streaming. Because of their unpredictable delay and availability, TCP/UDP are not suitable for applications with real time characteristics. The RTP is a thin protocol providing support for

applications with real-time properties, including timing reconstruction, loss detection, security and content identification. RTCP is the control protocol that works in conjunction with RTP, but RTP can be used without RTCP if desired. RTSP, the application level Real Time Streaming Protocol (RTSP), aims to provide an extensible frame work to enable controlled delivery of real time data. It is designed to work with established protocols such as RTP, HTTP, and others to provide a complete solution for streaming media over the Internet. It supports multicast as well as unicast. It also supports interoperability between clients and servers from different vendors. Resource ReServation Protocol (RSVP) is a network control protocol that allows Internet applications to reserve resources along the data path to obtain special QoS for their data flows.

Session Initial Protocol (SIP) developed by IETF, and H.323 developed by ITU are two application level signaling protocols. Both SIP and H.323 provide

- Call control, call set up, and tear down.
- Basic call features such as call waiting, call hold, call transfer, call forwarding, call return, and call identification.
- Capabilities exchange.

SIP is less restrictive and easier to implement as compared to H.323. Thus, more applications are moving to SIP.

Overall, application level signaling protocol SIP and H.323 are designed to initiate and direct delivery of media data from media servers. RTP is the transport protocol for the delivery of real time data while RTCP is a protocol for monitoring delivery of RTP packets. UDP and TCP are lower layer protocols for RTP/RTCP/RTSP/SIP/H.323 packets, and IP provides a common platform for delivering UDP/TCP packets over the Internet. RSVP is the protocol that reserves

the resources for real time applications at the router on the path. The combination of these protocols provides a complete streaming service over the Internet.

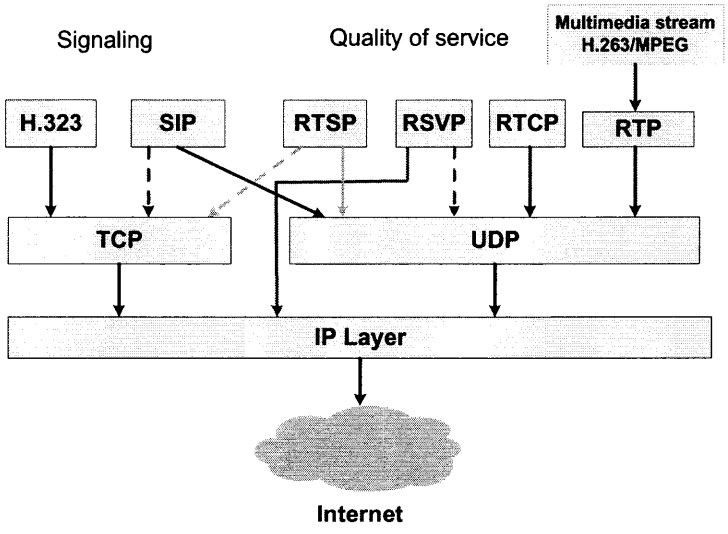

Figure 1.1 Protocol stacks for multimedia streaming.

1.4 Existing Bandwidth Allocations for MPEG Videos

Variable Bit Rate (VBR) is one of the major services to be supported by broadband packet switched networks. Video is inherently dynamic, and MPEG [3] video coding results in VBR. If the bandwidth is allocated according to the peak rate of the video traffic, no packet loss occurs, but a substantial amount of bandwidth is wasted during most of the transmission. On the other hand, if the bandwidth is not allocated close to the peak rate, large delay and excessive packet loss may be experienced. So in transporting the VBR video traffic, QoS guarantees provisioning is not trivial due to the bursty characteristics of the VBR traffic.

Issues related to transport MPEG video over network can be divided into two categories:

- Transport of pre-stored MPEG video

- Transport of real-time MPEG video.

1.4.1 Video on Demand

Video on Demand is one of the major applications provided by B-ISDN [4] [5] [6] [7]. It is a technology which enables a viewer to choose a video from a large selection, specify the video's start time, and have the video sent over a telecommunication channel to his home.

- Store and Forward (SAF)

  SAF is a well known approach [6]. The components of this architecture are Information Warehouses (IWHs), Central Office (CO) service circuits, and the Customer Premise Equipment (CPE). The IWHs are connected to the network via high capacity STS-3 (155Mbps) or STS-12 (622Mbps) lines, and therefore the information trunking between the IWH and the customer's CO can be done faster than real time. At the user's local CO, the information is buffered, the data rate is then converted to the video coding rate, and then possibly decoded to the original video signal form. The video signal is then transported to the user in the form which corresponds to the local access switching and transmission parameters and the user's CPE capabilities. Thus, the MPEG video data are first transported from the IWH to CO in bursts, and then the video data are buffered at the CO before delivering to the playback destination in real time.

- Constant Rate Transmission and Transport (CRTT)

  The basic concept of CRTT [7] is transporting a large amount (but not all) of video data to a playback buffer on the user side before the playback is commenced. The key issue of CRTT is to determine the required transmission bandwidth as well as the capacity of the playback buffer to guarantee neither underflow nor overflow during the playback. Considering a MPEG video sequence of N frames is played back at a fixed rate of F frames per second, the

CRTT scheme first sets up a fixed start-up delay d; then the minimum required transmission bandwidth, $b_{min}(d)$, to guarantee no buffer underflow during the playback time can be determined by

$$b_{min}(d) = \max_{d \le n \le N-1} \frac{F}{n}(\sum_{i=1}^{n+1} x_i - \sum_{i=1}^{d} x_i), \tag{1.1}$$

where $x_i (i = 1, 2, \ldots, N)$ are the video data in bytes of the ith frame. On the contrary, for a given playback buffer capacity, B, the maximum transmission bandwidth $b_{max}(d, B)$ to guarantee no buffer overflow can be determined by

$$b_{max}(d, B) = \min_{0 \le n \le N(B)-1} \frac{F}{n+1}(\sum_{i=1}^{n+1} x_i - \sum_{i=1}^{d} x_i + B), \tag{1.2}$$

where $N(B) = max\{n : \sum_{i=1}^{n} x_i + B < C\}$. Hence, the minimum required capacity of playback buffer to guarantee no overflow is given by

$$B_{min} = min\{B : D(B)\} \tag{1.3}$$

where

$$D(B) = \{d : d \le \bar{d}(B) \ , b_{min}(d) \le b_{max}(d, B)\}, \tag{1.4}$$

$$\bar{d(B)} = max\{d : \sum_{i=1}^{d} x_i \le B\}. \tag{1.5}$$

Transporting MPEG video data in CRTT is via CBR, and thus it has a lower transport cost than using VBR and greater playback flexibility than using best effort service; however, CRTT requires a playback buffer with a very large capacity and also involves long delay before the starting of playback.

1.4.2 Real Time Video

Emergence of multimedia communications has inspired a number of dynamic bandwidth allocation algorithms [8], [9], [10], [11], [12], [13], [14], [15].

- Dynamic Search Algorithm (DSA)

  DSA, proposed by Fulp *et al.* [11], dynamically adjusts the resource allocation based on the measured quality of service. The Cell Loss Probability (CLP) is calculated up to the current period, and the service rate for the next period is adjusted based on the current CLP. Let t_n denote the *nth* renegotiation instant, and the interval between renegotiation points t_n and t_{n+1} as the *nth* update interval U_n. The service rate during U_n is constant and is denoted as μ_n.

  During the *nth* interval, let the number of arrivals be represented by A_n, and the number of losses as L_n. The CLP of the *nth* interval is then calculated as $P_n = L_n/A_n$. The cumulative CLP of all the intervals up to and including the *nth* is

$$P_{0...n} = \frac{\sum_{i=0}^{n} L_i}{\sum_{i=0}^{n} A_i}.$$

  The CLP desired by the user is denoted as Q_l. The goal of DSA is to adjust the server rate in order to provide the desired Q_l as efficiently as possible with few renegotiations.

  At each renegotiation point, DSA adjusts the server rate according to the following formula:

$$\mu_{n+1} \leftarrow \mu_n + \frac{K}{\alpha} \times ln\frac{P_n}{Q_l},$$

$$\alpha = \begin{cases} 1 & \text{if } P_n > Q_l \\ 2 & \text{if } P_n \leq Q_l. \end{cases}$$

The constant K amplifies the response of the error function and this product ultimately determines how much the server rate can be increased or decreased. Parameter α allows the rate to be increased twice as fast as it can be. Owing to the difficulty of assessing the CLP on line and indirect relationship of the current CLP and User Parameters (UPs) with future bandwidth requirements, this approach is not effective enough to enhance the QoS and improve the bandwidth utilization.

- D-BIND

  Knightly and Zhang [12] proposed a renegotiated deterministic service model to support the VBR video. They used a traffic model called Deterministic Bounding Interval Length Dependent (D-BIND) model to support the VBR video. With the D-BIND model, sources characterize their traffic with P rate interval pairs $\{(R_k, I_k)|k = 1, 2\cdots, P\}$ where R_k is the bounding rate over the interval I_k, which are specified to the network at the connection-setup time. The network then performs the CAC tests based on both the D-BIND traffic parameters and the requested QoS parameters. The D-BIND constraint function is defined as:

$$\begin{aligned} b(t) &= \frac{R_k I_k - R_{k-1} I_{k-1}}{I_k - I_{k-1}}(t - I_k) + R_k I_k \\ & I_{k-1} \leq t \leq I_k, \end{aligned} \tag{1.6}$$

  with $b(0) = 0$ and $b(\cdot)$ repeating for $t > I_P$, such that

$$b(t) = b(t - [t/I_P]t) \text{ for } t > I_P. \tag{1.7}$$

  Renegotiating for more resources requires the admission control strategy to test if the maximum delay of all the connections can still be met. This requires the construction of the constraint function of the aggregate traffic of the connections

that are admitted in the network. This clearly results in a much more expensive renegotiation as compared to RCBR. For live video, Knightly and Zhang [16] proposed that the D-BIND model parameters of the last M frames of the video be estimated and compared to the already reserved parameters. A heuristic algorithm is then used to decide when to renegotiate and what parameters to reserve. The estimation of R_k is computationally expensive, especially for on line use.

- Frequency Domain Prediction

  Chong *et al.* [13] presented a novel approach to dynamically allocate the transmission bandwidth for transporting real-time video in ATM networks. The video traffic is described in the frequency domain: the low frequency signal captures the slow time variation of consecutive scene changes; the high frequency signal exhibits the feature of the strong frame autocorrelation. Their study indicated that the video transmission bandwidth in a finite buffer system is essentially characterized by the low frequency signal. Thus, they proposed a method to dynamically allocate the bandwidth based on predicting the low frequency part of the video rate input sequence.

- Wavelet Domain Prediction

  Wavelet transformation is an emerging technique that has a significant advantage in analyzing time-domain signals. The wavelet transform approximately diagonalizes the correlation matrix of the input data that permits independent adaptation of each of the filter coefficients. In addition, it reduces the eigenvalue spread of the correlation matrix of the input data. It is well known that the autocorrelation matrix of the input data governs many properties of LMS algorithm. Particularly, the eigenvalue spread of the above matrix determines the convergence speed of LMS; the larger the spread,

the slower the convergence speed is. When wavelet transform is used, the autocorrelation matrix is transformed to

$$R_v = E[v_n \cdot v_n^T] = E[Rx_n \cdot (Rx_n)^T] = R \cdot R_x \cdot R^T \tag{1.8}$$

with a possible eigenvalue spread reduction, where R is the wavelet transform matrix, and x_n and v_n are the input data vector and wavelet-transformed input data vector, respectively. Thus, the wavelet predictor converges faster, and hence tracks scene changes better. The problem is that the computational complexity is rather high and not suitable for on line traffic prediction. We propose a fast non-linear algorithm in Chapter 2, that can not only track scene change better, but also has a low computation complexity.

- Adaptive Linear Prediction

  An adaptive linear prediction scheme was proposed by Adas [14]. This scheme does not require any prior knowledge of the video statistics nor does it assume stationary, and is thus very suitable for on-line real time prediction. The simplicity and relatively good performance make it of particular interest. One problem associated with the LMS algorithm is its slow convergence. Thus, when there are scene changes, the bit rate variation is so high that the prediction error can be large. Xu and Qureshi [17] proposed a composite MPEG traffic prediction scheme which smoothes the predicted data based on predicting relative changes of frame size between adjacent GOPs by using LMS. Since I, P and B possess different statistical characteristics, this method is not effective in guaranteeing the CLR and needs re-negotiations for every frame, a big burden to network management.

CHAPTER 2

EFFICIENT BANDWIDTH PREDICTIONS FOR MPEG VIDEOS

Variable Bit Rate is one of the major applications to be supported by broadband packet switched networks. Video is inherently dynamic, and MPEG video coding results in VBR. If the bandwidth is allocated according to the peak rate of video traffic, no packet losses occur, but a substantial amount of the bandwidth is wasted during most of the transmission. On the other hand, if the bandwidth is not allocated close to the peak rate, large delays and excessive packet loss may be experienced.

Bandwidth should be allocated on demand, and dynamic bandwidth allocation and negotiations during the connection's life time should be considered. As bandwidth prediction plays an important role in bandwidth allocation, we propose our bandwidth prediction schemes in Section 2.2 and Section 2.3.

2.1 Characteristics of MPEG Video

For compression, an MPEG video is obtained by using intraframe technique, which exploits the spatial redundancy within a picture, as well as interframe technique, which exploits the temporal redundancy present in a video sequence. An MPEG stream contains three major frame types:

- I frame - An intra-coded picture, is an encoded picture based entirely on the information in that frame.
- P frame - A predictive-coded picture, is based on motion compensated prediction between that frame and the previous reference frame.
- B frame - A bidirectionally predictive-coded frame, is based on motion compensated prediction between that frame and the previous or next reference frame.

An MPEG video stream is divided into units called Group of Pictures (GOPs). A GOP consists of an I frame and an arrangement of B and P frames as shown in Fig. 2.1[1]. I frame must appear regularly in the stream, since they are needed to decode subsequent inter-coded frames such as P frames and B frames. The decoding process cannot begin until an I frame is received.

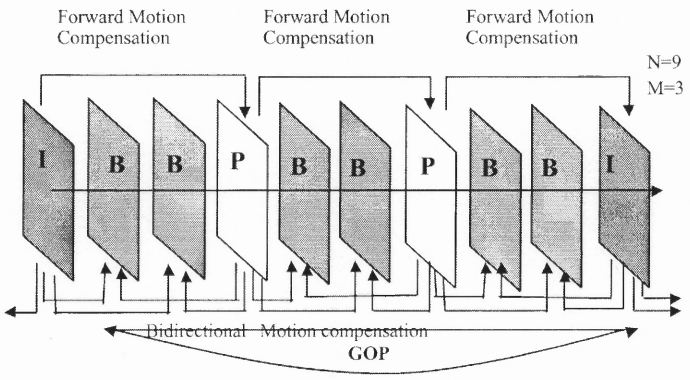

Figure 2.1 GOP structure of an MPEG video.

An MPEG video is inherently VBR. The variation of the bit rate generated in the codification is produced by both extrinsic and intrinsic reasons when a fixed quality is set. The extrinsic ones are produced by the changes of the complexity and activity of the sequence to be coded. The intrinsic reasons are related to the codification modes applied on the frames. The changes in the output rate of an MPEG encoder are attributed to the following three aspects:

- The encoding of one block to the next within a picture.
- From one picture to the next within the video sequence being encoded.
- From one scene to the next within the video sequence.

[1]This figure was the courtesy from Y. Q. Shi and H. Sun, *Image and Video Compression for Multimedia Engineering*, CRC Press, 2000, p. 334.

Thus, within each frame type, the size of the I frame varies depending on the picture content; P and B frames vary depending on the motion present in the scene as well as the picture content.

Through the analysis of the MPEG video trace we find that I frames often have large frame sizes, and B frames have small frame sizes. Most of the time, when the I frame size changes significantly, the P and B frame size also change significantly, implying that the increase or decrease of the I frame size often indicates the increase or decrease of the P and B frame sizes, and therefore we only need to predict the I frame size.

2.2 Predicting the Relative Size Change of the I Frames

Let I_k be the size of the I frame of the kth GOP and I_{k-1} be the size of the $(k-1)th$ GOP, then the relative size change of I frame s_k is defined by

$$s_k = \frac{I_k - I_{k-1}}{I_{k-1}}. \tag{2.1}$$

The sequence s_k is much smoother than the sequence I_k. So the linear adaptive prediction will perform better if we predict the sequence s_k instead of the sequence I_k; the I frame size can then be retrieved by

$$I_k = s_k I_{k-1} + I_{k-1}. \tag{2.2}$$

A one-step linear predictor can be used to predict the s_k sequence, i.e., prediction of s_{k+1} using a linear combination of the current and previous values of s_k. The number of the current and previous values of s_k used to predict s_{k+1} is called the order of the linear predictor. The pth-order linear predictor has the following form:

$$\hat{s}_{k+1} = \sum_{l=0}^{p-1} w_l s_{k-l} = \mathbf{W}^T \mathbf{S}_k, \tag{2.3}$$

where p is the order of the linear predictor, and w_l, for $l = 0, 1, \ldots, p-1$, are the prediction filter coefficients. The prediction error is

$$e_k = s_k - \hat{s}_k. \tag{2.4}$$

The LMS predictor minimizes the mean squares error by adaptively adjusting the coefficient vector $\mathbf{W}$. In the normalized LMS algorithm [14], if we use the one-step linear predictor, $\mathbf{W}$ is updated by

$$\mathbf{W_{k+1}} = \mathbf{W_k} + \frac{\mu \mathbf{e_k S_k}}{\|\mathbf{S_k}\|^2}. \tag{2.5}$$

The Akaike Information Criterion (AIC) [18] is used to choose the best order not greater than 12. The AIC criterion associates a cost function with the order of the filter. It was found by numerous simulations that the autocorrelation of the prediction error e_k is close to that of the white noise. Thus, we use one-step, 12-order adaptive linear predictor for both our algorithm and the composite prediction scheme [17].

The performance of our algorithm for the video trace CD122 [2], justified by the inverse Signal to Noise Ratio (SNR^{-1}) is

$$SNR^{-1} = \frac{\sum e^2(n)}{\sum s^2(n)} = 0.0040.$$

Fig. 2.2 shows that forecasted values appear close to the actual values except at sharp transitions, which are most likely due to scene changes.

Fig. 2.3 shows the actual and forecasted frame size variation proposed by Xu and Quresh [17], which predicts the relative size change of every frame between two adjacent GOPs. The sequence is defined as follows:

$$f_k = \frac{F_k - F_{k-d}}{F_{k-d}}, \tag{2.6}$$

[2]The video trace CD122 was the courtesy of Wu-chi Feng of the Ohio State University.

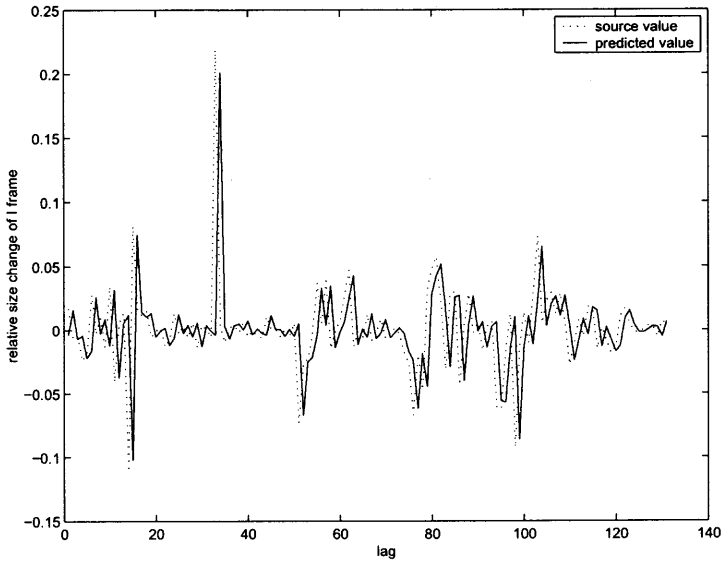

Figure 2.2 Actual and forecasted I frame size variation for CD122.

where F_k is the size of the kth frame in the MPEG encoded video sequence, d is the length of the GOP, and f_k is the relative change of frame sizes between adjacent GOPs. Since I, B, P frames are coded with different compression levels, they have different statistical properties. f_k is more fluctuated than s_k, and so the prediction error of f_k shown in Fig. 2.3 is larger than that of s_k shown in Fig. 2.2.

The performance of the composite prediction scheme for CD122 is

$$SNR^{-1} = \frac{\sum e^2(n)}{\sum s^2(n)} = 0.8471.$$

From Figs. 2.2 and 2.3 and their SNR^{-1}, our proposed scheme is more accurate than the composite prediction scheme [17]. On the other hand, it is not possible to allocate bandwidth based on every frame size, because the negotiation frequency is very high, thus imposing a big burden to the network. Hence, we should not predict every frame size.

Since the sequence $\{s(k)\}$ is smoother than the sequence $\{I(k)\}$, the prediction error is smaller than predicting directly the original sequence.

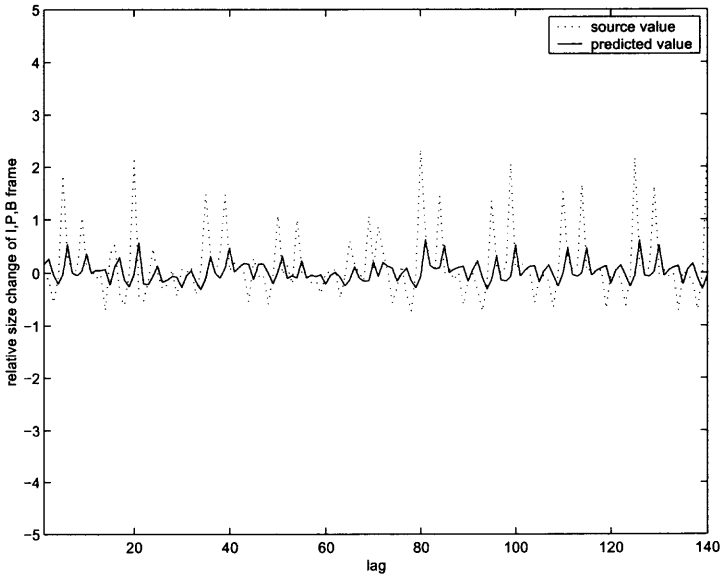

Figure 2.3 Actual and forecasted frame size variation (I, P, B) for CD122.

2.3 The Fast Algorithm for Video Traffic Prediction

The algorithm proposed in the previous section not only smoothes the predicted data, reduces the renegotiation frequency, but also achieves much smaller prediction error than that of the composite MPEG traffic prediction scheme. The only drawback is its slow convergence. In VBR video traffic characterized by frequent scene changes as shown in Fig. 2.4 [3], the LMS algorithm may result in an extended period of intractability, and thus may experience excessive cell loss during scene changes; hence, we propose a fast convergent nonlinear adaptive algorithm to predict the relative size changes of I frames. This new algorithm converges fast, and hence, tracks scene changes better.

[3]The author would like to thank Mark Garrett of Telcordia and Wu-chi Feng of the Ohio State University for the video traces provided by them.

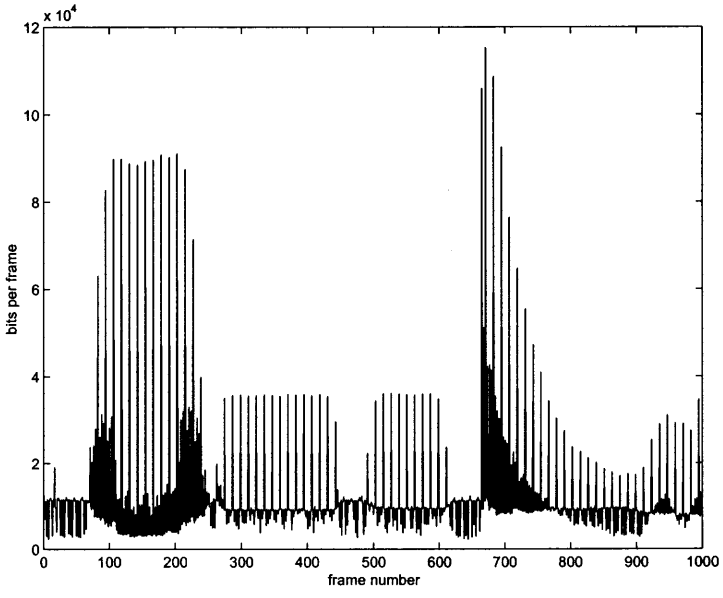

Figure 2.4 Video traffic *Star Wars*.

2.3.1 The Fast Algorithm

In the standard LMS algorithm as shown below, μ is a constant; we refer to this algorithm as the Fixed Step-size Algorithm (FSA). Since the video traffic is bursty, if we increase the step size μ, we can achieve fast convergence at the cost of a large prediction error. On the other hand, the prediction error can be made small by decreasing the step size μ at the cost of the convergence rate. The choice of the step size reflects the trade off between the misadjustment and the speed adaptation.

$$\hat{s}_{k+1} = \sum_{l=0}^{p-1} w_l s_{k-l} = \mathbf{W}^T \mathbf{S}_k, \tag{2.7}$$

$$e_k = s_k - \hat{s}_k. \tag{2.8}$$

$$\mathbf{W_{k+1}} = \mathbf{W_k} + \frac{\mu \mathbf{e_k} \mathbf{S_k}}{\|\mathbf{S_k}\|^2}. \tag{2.9}$$

Kwong and Johnston [19] proposed a variable step size algorithm for adjusting the step size μ_k:

$$\mu'_{k+1} = \alpha\mu_k + \gamma e_k^2, \tag{2.10}$$

with $0 < \alpha < 1$, $\gamma > 0$, and

$$\mu_{k+1} = \begin{cases} \mu_{max} & \text{if } \mu'_{k+1} > \mu_{max} \\ \mu_{min} & \text{if } \mu'_{k+1} < \mu_{min} \\ \mu'_{k+1} & \text{otherwise.} \end{cases} \tag{2.11}$$

The initial step size μ_0 is usually taken to be a little large, although the algorithm is not sensitive to the choice. As can be seen from Eq. (2.10), the step size is always positive and is controlled by the size of the prediction error, and the parameters α and γ. Intuitively, a large prediction error increases the step size to provide faster tracking. If the prediction error decreases, the step size will be decreased to reduce the misadjustment. The constant μ_{max} is chosen to ensure that the Mean Square Error (MSE) of the algorithm remains bounded. Usually, μ_{min} is chosen to be close to the value that has been chosen for the fixed step size algorithm. We propose to modify Eq. (2.10) to the following:

$$\mu'_{k+1} = \alpha\mu_k + \gamma(q_1 e_k^2 + q_2 e_{k-1}^2), \tag{2.12}$$

where $q_1 + q_2 = 1$ to accommodate the video traffic characteristics. Since a video frame sequence consists of many scenes, the bit rate varies greatly among different scenes, while during a scene, the bit rate in the frame sequence has a strong auto-correlation, and thus better prediction performance is expected. Equation (2.12) includes an additional error term e_{k-1}, to smooth out the drastic change of μ_k (i.e., a spike) during the transition from one scene to another as shown in Fig. 2.5, thus easing the buffer management. We refer to this algorithm as the fast convergent Variable Step-size Algorithm (VSA), and Kong and Johnston's method as KVSA in this thesis. Here,

e_k and e_{k-1} are the current and previous prediction errors, respectively, and q_1 and q_2 are their respective weights. We empirically found that $\alpha = 0.98$, $\gamma = 0.015$, $q_1 = 0.7$ and $q_2 = 0.3$ provide the best performance in all our real video trace simulations.

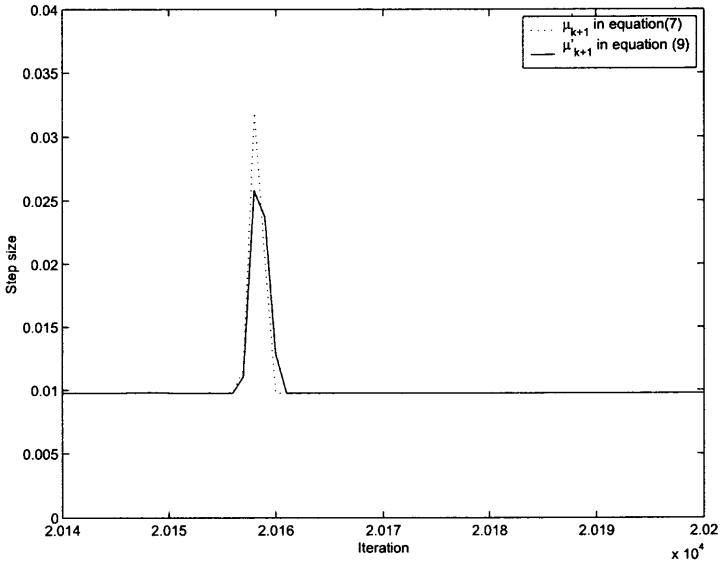

Figure 2.5 Comparison of step-size.

2.3.2 Simulation Results

Simulations on four 1.5-hour long empirical VBR traffic data sets were conducted. These data sets correspond to the relative size change of I frames. For performance comparison among VSA, KVSA and FSA, we use $SNR^{-1} = \frac{\sum e^2(n)}{\sum s^2(n)}$ as a metric, which is the ratio of the sum of squares of the prediction errors and sum of squares of the input data. For a fair comparison, VSA, KVSA and FSA use the same 12-order and one-step ahead prediction, and parameters α and γ are the same in both VSA and KVSA. The results are shown in Table 2.1.

From Table 2.1, VSA and KVSA incur smaller prediction error than FSA in all the four tested sequences. VSA further reduces the prediction errors as shown in Table 2.1. The performance has been improved greatly if we use VSA instead of

Table 2.1 Performance Comparison of FSA, KVSA, and VSA Predictors on Relative Size Change of I Frames

($\frac{\sum e^2(n)}{\sum s^2(n)}$ is used as a metric)

Sequence	FSA	KVSA	VSA	Improvement(%)
CD122	0.0040	0.0035	0.0032	20
Talk2	0.0078	0.0071	0.0069	12
News	0.0247	0.0213	0.0210	15
SoccerWM	0.0512	0.0438	0.0404	21

FSA as shown in Table 2.1. The percentage improvement is with respect to VSA over FSA. Figure 2.6 shows the convergence properties of FSA ($\mu = 0.009$) and VSA. Note that VSA converges much faster than FSA. If the step size is increased to $\mu = 0.3$ for FSA, the convergence is faster as shown in Fig. 2.7 (note that the MSE is expressed in dB), but the prediction error is increased greatly; here, the iteration represents the iteration index, in this case, $\frac{\sum e^2(n)}{\sum s^2(n)} = 0.0191$ for FSA, $\frac{\sum e^2(n)}{\sum s^2(n)} = 0.0032$ for VSA.

2.4 Summary

Prediction of the relative size change of I frames is proposed in this chapter. As the sequence $\{s(k)\}$ is smoother than the sequence $\{f(k)\}$, better prediction performance can be achieved. Compared to the scheme proposed by Adas [14], the number of predictors is reduced from three to one, thus reducing the burden to the network.

A variable step size predictor is also proposed in this chapter for VBR videos, where the step size adjustment is controlled by the squares of the prediction errors to reduce the trade off between misadjustment and tracking ability of the fixed step size LMS algorithm. Simulation results show that VSA not only incurs small prediction errors but more importantly also achieves fast convergence. The additional overhead over the VSA for computation is essentially one more weight update at each time

step, so that the increase in complexity is minimal. The computational complexity is much smaller as compared to Wavelet prediction. The prediction, when combined with dynamic allocation, could provide a solution that achieves network efficiency and meet QoS guarantees. Since VSA can significantly reduce the CLR, we introduce the bandwidth allocation based on predicted I frames and GOP using VSA in the next chapter.

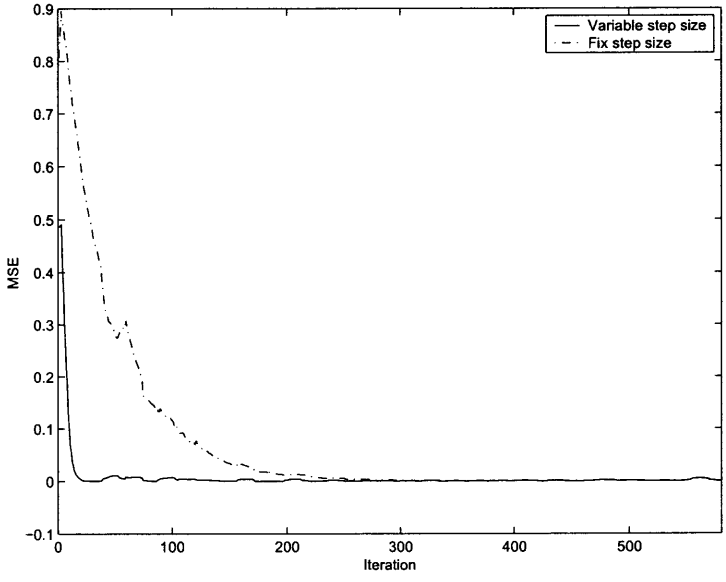

Figure 2.6 Comparison of convergence properties of FSA ($\mu = 0.009$) and VSA on CD122 trace.

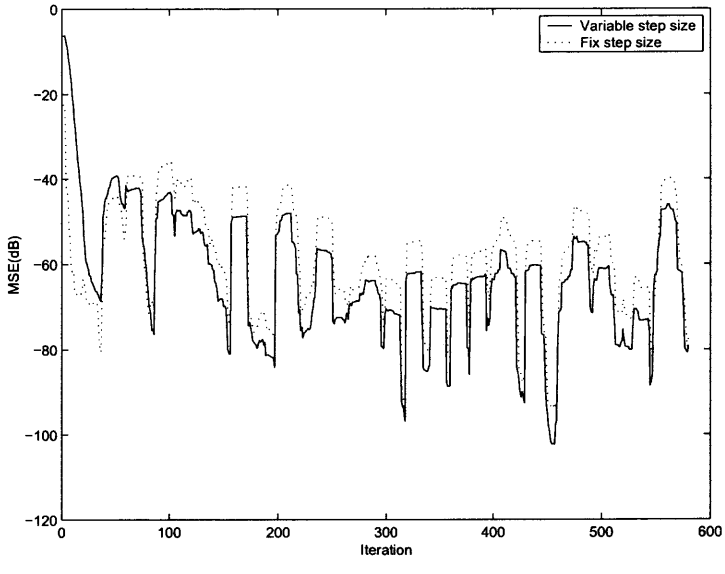

Figure 2.7 Comparison of MSE (dB) of VSA and FSA ($\mu = 0.3$) on CD122 trace.

CHAPTER 3

DYNAMIC BANDWIDTH ALLOCATIONS FOR MPEG VIDEOS

An efficient resource allocation comprises determining optimal buffer sizes, assigning bandwidth, and other resources in order to meet the desired QoS expressed in terms of parameters such as queuing delay, transmission time, packet loss probability, and bit error rate. There are two kinds of bandwidth allocation method:

- Static Bandwidth Allocation (SBA)
- Dynamic Bandwidth Allocation (DBA)

In SBA, the available resources are assigned to the source traffic at service initialization and are kept the same throughout the lifetime of the connection. The bandwidth can be allocated according to the peak rate or the mean cell rate of the VBR traffic. For highly correlated traffic like video, if the bandwidth is allocated according to the peak rate of the traffic, no packet loss occurs, but a substantial amount of bandwidth is wasted during most of the transmission. On the other hand, if the bandwidth is allocated based on the source mean cell rate, the video service will suffer from unacceptable losses and delays, especially those with hard real time constraints as shown in [20]. Thus, dynamic bandwidth allocation is needed for VBR traffic. To implement the dynamic bandwidth allocation, there are two different operations: synchronous and asynchronous operations [13]. In synchronous operation, the bandwidth is periodically adapted at a fixed time interval. In asynchronous operation, the bandwidth will be updated whenever a need is detected [13], [21], [22], [23]. The asynchronous operation can significantly reduce the adaptation frequency.

Prediction can be used in many ways for allocating bandwidth for the VBR video traffic. The bandwidth allocation mechanism can be activated by each prediction, or

only those predictions whose results are larger than a certain threshold. The variation reflects the trade-off between the network utilization and the overhead for bandwidth negotiation. The choice for a network application depends on many factors, such as the network service model, latency, and implementation complexity.

3.1 Dynamic Bandwidth Allocation and its Queuing Performance

Since the increase or decrease of I frames often indicates the increase or decrease of P and B frames, we propose to allocate bandwidth based on the predicted I frame size using VSA to improve the QoS and network utilization. Queuing performance is also analyzed for our proposed scheme in Section 3.1.2.

3.1.1 Dynamic Bandwidth Allocation Based on Predicted I Frames Using VSA

A single server First In First Out (FIFO) queue is simulated. The assumed network service model is RCBR [24]. Let I_k be the size of the predicted I frame of the kth GOP, R be the transmission rate for the previous GOP and δ be a threshold, then the dynamic bandwidth allocation algorithm can be stated as follows:

- if $|I_k - R/N| < \delta$, then the transmission rate remains unchanged.
- if $|I_k - R/N| \geq \delta$, then $R = I_k \times N$,

where N is the number of frames needed to be transmitted per second.

The negotiation frequency can be reduced significantly because only I frames need to be checked. Since the I frame size in a GOP is the largest most of the time, the bandwidth allocated is very close to the largest one needed for transmission of frames in the GOP, and therefore CLR can be kept small. The CLRs for different values of δ for the sequence CD122 are shown in Fig. 3.1.

Fig. 3.2 shows the performance of the video trace "Talk2" show using our prediction algorithm, where $\delta = 1000$, and the buffer size stands for the size of

the buffer needed at the switch. The prediction performance is

$$SNR^{-1} = 0.0069.$$

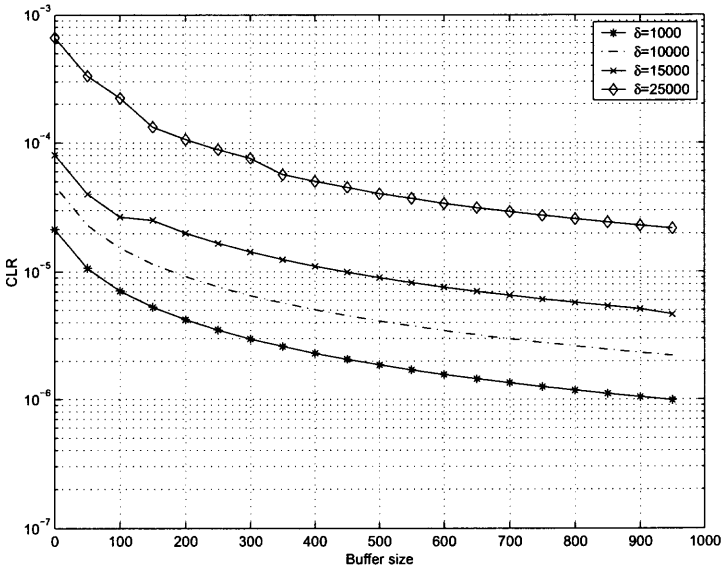

Figure 3.1 CLR versus buffer size for different values of δ.

3.1.2 Impact of Autocorrelation on Queue Size

The frame size trace from the output of the MPEG video contains all statistical information about the encoded video. The frame by frame correlation depends on the pattern of the GOP, and in principle always looks like Fig. 3.3 if the same GOP pattern is used for the whole sequence. For this example, the GOP pattern is IBBPBBPBBPBBI....

Fig. 3.3 shows the Autocorrelation Function (ACF) of the MPEG coded *Star Wars* (lag is expressed in terms of the frame number), in which the largest positive peaks stem from I frames, the larger positive ones from P frames, and the smallest ones from B frames. A large I frame is followed by two small B frames, then a middle

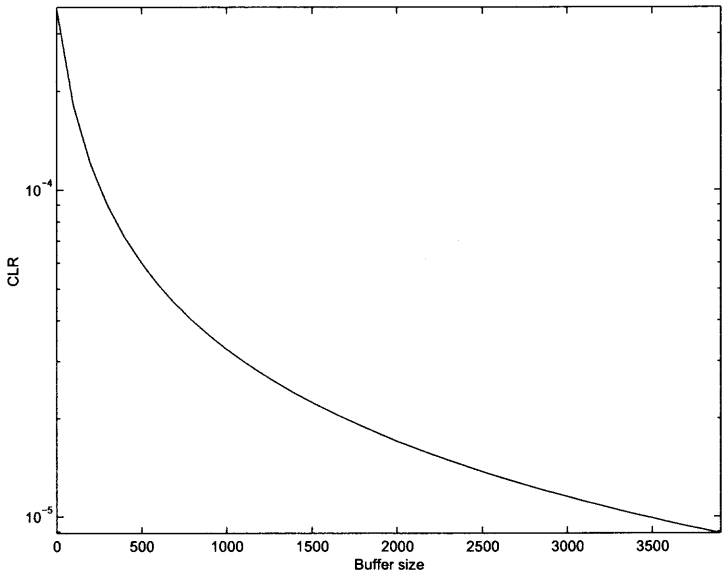

Figure 3.2 CLR versus buffer size for the Talk2 show $\delta = 1000$.

size P frame is followed by two small B frames again. The pattern between two I frame peaks is repeated with slowly decaying amplitude of the peaks.

This figure shows that the MPEG video is highly correlated. If it is not served at a rate close to the peak rate, large queues, large delays and excessive cell loss will result, but if the bandwidth is reserved at least equal to the predicted value, only the error caused by the prediction needs to be buffered. If the error resembles white noise or at most short memory, only small buffers will suffice, and high utilization and small delays can be achieved.

By reserving the bandwidth at least equal to the predicted values, only prediction errors need to be buffered. The autocorrelation of the prediction errors of the relative size change of I frames for video sequences CD122 and Talk2 are shown in Figs. 3.4 and 3.5, respectively. Note that they resemble white noise, a rather uncorrelated process. Since traffic autocorrelation has great impact on the queuing performance, understanding how the queue responds to different autocorrelations is very important.

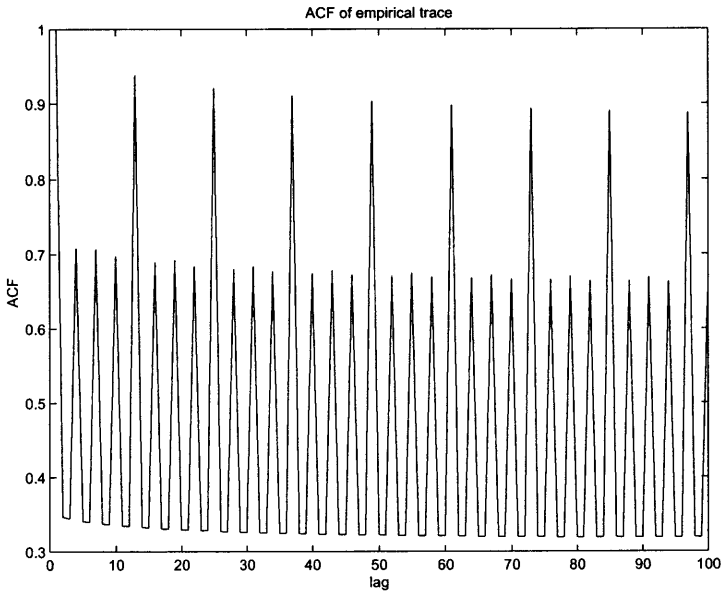

Figure 3.3 ACF of an MPEG video.

Li and Hwang [25] analyzed this impact in the frequency domain. Consider a Markov modulated Poisson process, where the underlying Markov Chain (MC) is used to reflect the time correlation of the input process at the macro-level; each state of MC is associated with a constant input rate. The local dynamics of individual packet arrivals at the micro-level, while in each state of MC, is characterized by the Poisson process. This process is described by an N state discrete time MC transition matrix P which is diagonalizable, and its associated input rate vector $\vec{\gamma}$. The input rate correlation is expressed as follows:

$$R(n) = \sum_{\lambda_l \in \Omega_r} \psi_l \lambda_l^{|n|} + \sum_{\lambda_l \in \Omega_C Im(\lambda_l) \geq 0} 2|\psi_l||\lambda_l|^{|n|} \cos(|n|\omega_l T + \theta_l T). \tag{3.1}$$

where λ_l is the l^{th} eigenvalue of P, ψ_l is associated with both the l^{th} eigenvector of P and the input rate $\vec{\gamma}$, $\omega_l = arg(\lambda_l)$, T is the time unit, Ω_r is the subset of the real eigenvalues of P, Ω_c is the subset of complex eigenvalues of P, and $Im(x)$ is the imaginary part of x.

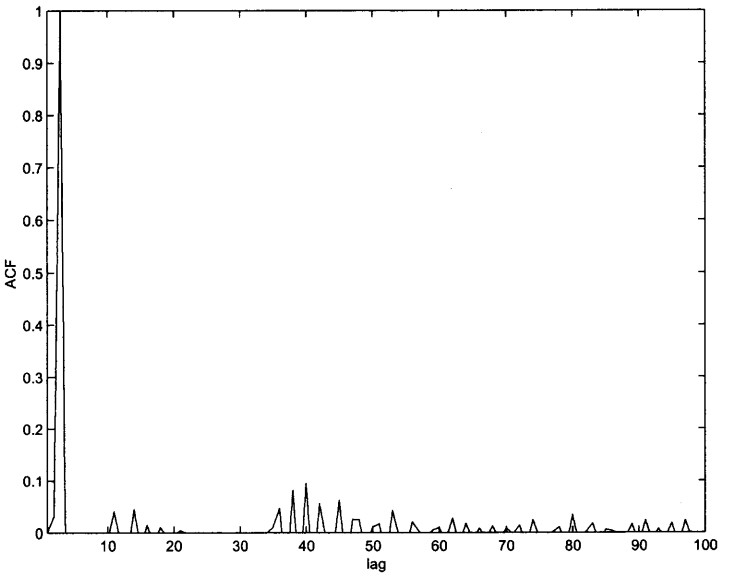

Figure 3.4 Autocorrelation of the prediction error for CD122.

The power spectral function of the input rate process is defined by the discrete-time Fourier transform of $R(n)$: $P(\omega) = \sum_{n=-\infty}^{\infty} R(n)e^{-\sqrt{-1}n\omega T}$, which is equal to

$$P(\omega) = \sum_{l} \frac{\psi_l(1-\lambda_l^2)}{1-2\lambda_l\cos(\omega T)+\lambda_l^2} \,. \tag{3.2}$$

From Eq. (3.1) and Eq. (3.2), essential characteristics of the input correlation (or its power spectrum) are captured by the input MC eigenvalues, and at the same time one may identify the contribution of each eigenvalue to both $R(n)$ and $P(\omega)$. For the positive real λ_l, the larger the λ_l, the stronger the correlation for $R(n)$, the more the input power is in the low frequency band of $P(\omega)$ [26]. A negative real eigenvalue contributes most power in the high frequency band. Each conjugate pair of eigenvalues contribute two symmetric bell shaped curves centered at the resonant frequencies $\omega = \pm\omega_l$ in $P(\omega)$, that corresponds to a damped sinusoidal term in $R(n)$; the bell position is described by the resonant frequency $\omega_l = arg(\lambda_l)$ while the bell shape is characterized by the damping coefficient λ_l.

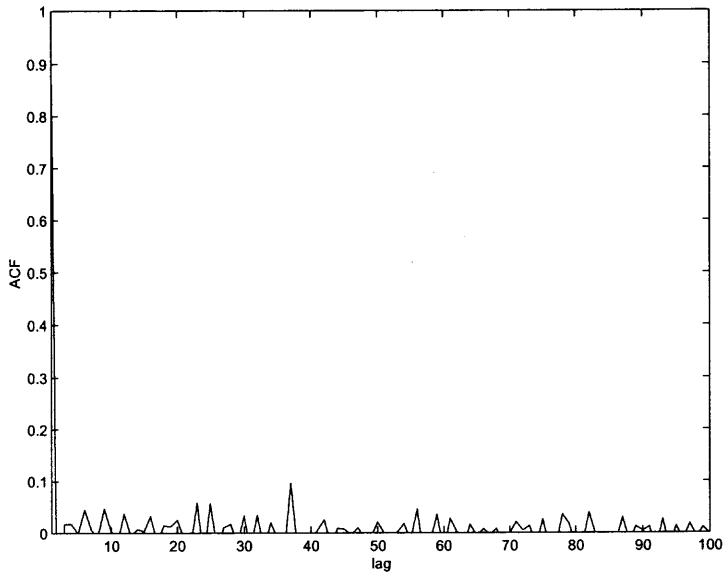

Figure 3.5 Autocorrelation of the prediction error for Talk2.

Li and Hwang [25] analyzed the queue response to the input correlation or the power spectrum by an N-state periodic chain to match the input power spectrum. While in state i, the packet arrivals are characterized by a Poisson process with input rate γ_i. The model discussed here is a typical two-dimensional MC in the discrete-time domain. The mean queue size is expressed as follows:

$$E[q] = \frac{\bar{\gamma}^2}{2(1-\bar{\gamma})} + \frac{1}{1-\bar{\gamma}} \sum_{j=0}^{N-1} \sum_{l=0}^{N-1} c_l \gamma(j,l), \tag{3.3}$$

with

$$\gamma(j,l) = \sum_{k=0}^{j} (\gamma_k - \bar{\gamma}) - \sum_{k=0}^{l} (\gamma_k - \bar{\gamma}), \tag{3.4}$$

$$\bar{\gamma} = \frac{1}{N} \sum_{l} \gamma_l. \tag{3.5}$$

c_l is the l^{th} element of the boundary vector $\vec{c}$ [27]. The first item in Eq. (3.3) is also equal to the mean queue size of the $M/D/1$ model, which in our case is equivalent to the queue response to the white noise input. The second moment of the queue is

given by

$$E[q^2] = \frac{\bar{\gamma}^4 - \bar{\gamma}^3 + 3\bar{\gamma}^2}{6(1-\bar{\gamma})^2} + \frac{1-\bar{\gamma}+\bar{\gamma}^2}{(1-\bar{\gamma})^2} \sum_{j=0}^{N-1} \sum_{l=0}^{N-1} c_l \gamma(j,l) + \frac{1}{1-\bar{\gamma}} \sum_{j=0}^{N-1} \sum_{l=0}^{N-1} c_l \gamma^2(j,l). \tag{3.6}$$

The boundary vector $\vec{c}$ is defined by $\vec{c} = [c_0, c_1, \ldots, c_{N-1}]$. The most complexity involved is to solve the N linear equations for the boundary vector $\vec{c}$ described by the following equations

$$\sum_{j=0}^{N-1} c_j W^{ij} e^{(1-z_i)\sum_{k=0}^{j}(\gamma_k - \bar{\gamma})} = 0, \quad \text{for } i \neq 0, \tag{3.7}$$

$$N \sum_{j=0}^{N-1} c_j = 1 - \bar{\gamma}, \quad \text{for } z_0 = 1. \tag{3.8}$$

z_i is always recursively obtained from the following equation

$$z = W^i e^{-(1-z)\bar{\gamma}}, \quad \text{for } i = 0, 1, \ldots, N-1. \tag{3.9}$$

By examining the queue response to various input correlation properties on the basis of the input power spectrum in the discrete frequency domain, we conclude that the queuing behavior is dominated by input power in the low frequency band, and many high-frequency components existing in the input process can be replaced by a constant input rate, with little impact on the queue response [25]. Thus, we can neglect the high frequency power in the power spectrum. This is true especially for multimedia traffic queuing analysis, where the input process often contains the dominant low frequency power as shown in Fig. 3.6. The larger the autocorrelation, the more input power is in the low frequency band and the longer the mean queue size is.

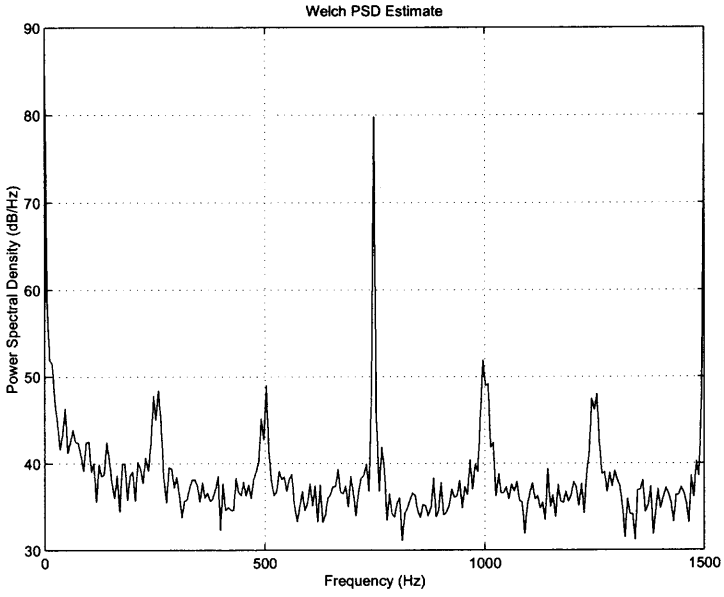

Figure 3.6 The power spectrum of the Talk2 show.

From the above analysis, we observe that the queuing performance is mainly dominated by input streams with high positive correlation. The higher the positive correlation in the time domain, the more the input power is in the low frequency band, and so the mean queue size will be larger. From the network perspective, the key point is to assign the link capacity to a given multimedia traffic in order to provide guaranteed quality of service. Video traffic is highly correlated, as shown in Fig. 3.3. The power spectrum, as shown in Fig. 3.6, has spikes appeared at harmonic frequencies due to the strong autocorrelation of periodic I frames. More input power is located in the low frequency band, thus resulting in large queue size, and hence we cannot guarantee the quality of service. A large queue will introduce a long delay that cannot be tolerated for video delivery especially for real time videos. Here, we propose a bandwidth allocation scheme based on the predicted I frame size (retrieved from s_k) in which only prediction errors need to be buffered. From Figs. 3.4 and 3.5, we note that the prediction error resembles white noise, or at most short memory, a rather "uncorrelated" process resembling white noise; this has little impact on the

queuing dynamics, and thus smaller buffers, less delay, and higher utilization can be achieved. Similar results are also derived for other sequences such as SoccerWM, News and Simpsons [28].

3.2 QoS Guaranteed Bandwidth Allocation

The bandwidth allocation scheme proposed in Section 3.1 can keep the CLR small and reduce negotiation frequency. As bandwidth is allocated based on the predicted I frames in this method, the network bandwidth utilization is not high, hereby the bandwidth utilization is defined as the ratio of the actual required bandwidth over the allocated bandwidth. If the bandwidth is allocated based on the predicted GOP, the network utilization can be increased, but the QoS is not guaranteed. In this section, we propose a QoS guaranteed on-line bandwidth allocation based on the predicted GOP using VSA and required QoS. Here, we consider only delay as the QoS parameter, and the bandwidth is renegotiated based on the delay bound and the predicted GOP at a given resource utilization. This scheme allocates the bandwidth based on the predicted GOP and delay requirement, thus, not only the QoS can be guaranteed but also the high bandwidth utilization can be achieved.

3.2.1 QoS Requirements

QoS is a generic notion that can be defined at various levels in the protocol stack, particularly the application and network levels. Application-level QoS is essentially visual and hard to measure in an objective manner [29]. It depends on the viewer's sensitivity to glitches in the video (which can be caused by lost packets), variability in the frame rate, variability in the distortion factor, and so forth. We should correlate these application-level QoS requirements to network-level QoS requirements in terms of throughput, delay, and packet loss.

Throughput requirements: Video is produced by displaying frames at a fixed rate

known as the *playback rate*. This rate varies from one video format to another. Several standardized video formats are available, including NTSC (30 frames/s) and PAL (25 frames/s). To ensure continuous streaming of video, the rate at which frames are transported over the network must be, on average, no smaller than the playback rate.

Delay requirements: Delay requirements clearly vary depending on the applications. For interactive video communication applications, a maximum end to end delay of some 150-400ms [30] is appropriate, while a much longer delay would be tolerable for a user simply watching a recorded clip or movie in a video playback application.

Loss requirements: In early MPEG reference models, cell loss rates lower than 10^{-9} were proposed, but rates of 10^{-4} are currently being considered as acceptable. The acceptable loss rate depends on several factors, including the compression scheme, duration of the loss interval, relative importance of the lost data and error concealment mechanism. In general, the loss requirement at the network level is expected to be in the range of 10^{-2} to 10^{-6} [30]. The network may provide a video stream with multiple loss rates, depending on the importance of the transmitted data. The effect of cell loss depends not only on the average cell loss rate but also on the distribution of cell losses over time. Periods of high cell loss rate due to network congestion can have a serious detrimental impact on the image quality.

3.2.2 Delay Guaranteed Bandwidth Allocation

In this section, we consider only delay as the QoS parameter. The bandwidth is allocated based on the predicted future traffic and the QoS under the given resource utilization. There are two methods in changing a Virtual Circuit (VC) bandwidth. One is by closing the existing VC and opening a new one with the new allocation, and the other is by changing the allocation without closing it. Both methods incur some overheads, and thus these adjustments should not be made too often. There is a trade-

off between transmission efficiency and processing efficiency in the design of dynamic bandwidth allocation. On the one hand, a high transmission efficiency can always be achieved by more frequent adaptation of the bandwidth to its required bandwidth. On the other hand, since the decision of the dynamic bandwidth allocations is often made at the network layer, the frequency of the bandwidth adaptation is limited by the network processing time. Analysis shows that the video transmission bandwidth needs to be adapted only at the interval of several hundred milliseconds, which is feasible in the practical network design. Owing to the above reasons, predicting GOP is suitable for practical applications. Fig. 3.7 shows the autocorrelation of the GOP for *Star Wars*. Note that the autocorrelation is high, and thus better prediction performance by our proposed fast non-linear adaptive algorithm [28] is expected. We thus propose to use this algorithm to predict the GOP size.

Our proposed QoS guaranteed dynamic bandwidth allocation (QDBA) scheme is based on the predicted GOP size and QoS requirements. We shall first define the following notations.

- $M-$ the number of frames in the GOP
- $N-$ the number of frames to be transmitted per second (i.e., 30 frames/sec for NTSC)
- $\rho-$ required bandwidth utilization
- $\bar{\rho}-$ average bandwidth utilization during the service time
- μ_0- initial bandwidth
- $\mu(i)-$ allocated bandwidth at time slot i
- $Q(i)-$ queue size at the end of time slot i
- $\bar{q}-$ average queue size

- $G_o(i)-$ actual size of the GOP at time slot i
- $G_p(i)-$ predicted size of the GOP at time slot i
- $R_o(i)-$ original bandwidth required at time slot i
- $R_p(i)-$ predicted bandwidth at time slot i
- $R_t(i)-$ required bandwidth to meet the delay constraint at time slot i
- $T-$ maximum allowed delay
- $R-$ current transmission rate

The QDBA algorithm at time slot i can be expressed as follows:

- $Q(i-1) = max\{G_o(i-1) + Q(i-2) - \mu(i-1) * M/N, 0\}$

$$R_p(i) = G_p(i) * \frac{N}{M}$$

$$R_t(i) = Q(i-1)/T$$

- if $R - R_p(i) \geq (1-\rho) \times R$, then

$$R = max\{R_p(i),\ R_t(i)\},\ \mu(i) = R$$

- else if $R < R_t(i)$

$$\mu(i) = R_t(i),\quad R = \mu(i)$$

- otherwise, $\mu(i) = R$, which keeps the bandwidth unchanged.

The QDBA algorithm at time slot i computes the queue size at the last time slot, and computes the required bandwidth to meet the delay. Three cases are to be considered in allocating the bandwidth. In the first case, if the current

transmission rate exceeds the bandwidth utilization, it means that the predicted bandwidth is smaller than the current transmission rate, and thus the bandwidth utilization is smaller than required. In this case, a renegotiation is triggered, and the QDBA algorithm will choose a larger one between the predicted bandwidth $R_p(i)$ and delay constrained bandwidth $R_t(i)$ in order to ensure the guaranteed delay. In the second case, if the current transmission rate is lower than the delay constrained rate, the transmission rate is changed to meet the delay requirement. Otherwise, the transmission rate is kept unchanged. Note that this algorithm provides the guaranteed delay. If the delay is very stringent, the delay is met at the cost of the bandwidth utilization. Simulations have been conducted on several video traces. For example, for the *Star Wars* video trace, the prediction performance using the non-linear fast convergence algorithm [28] has a 29% improvement over that of [14]. For prediction itself, our fast algorithm can achieve better bandwidth utilization and QoS guarantees than using the LMS prediction. This QDBA algorithm provides a delay guaranteed bandwidth allocation at the given resource utilization.

Figure 3.8 shows the actual delay for the video trace *Star Wars*, in which case the maximum allowed delay is $T = 0.01s$ with bandwidth utilization $\rho = 0.8$. Note that the maximum allowed delay is guaranteed. Figure 3.9 shows the actual allocated bandwidth when the required resource utilization is $\rho = 0.9$ and the maximum allowed delay is $T = 0.1$. The renegotiation frequency is $f = 3629$, and the average utilization is $\hat{\rho}$=0.9266 for the whole one and a half hour video trace *Star Wars*. To reduce the renegotiation frequency, we may decrease the bandwidth utilization ρ or increase the maximum delay. Figure 3.10 shows the actual allocated bandwidth when $\rho = 0.5$, and $T = 0.2$. In this figure, the renegotiation frequency has been decreased greatly, in which case $f = 531$. Figures 3.11 and 3.12 show the actual allocated bandwidth under the same condition ($\rho = 0.8$ and $T = 0.1s$) for video traces *Talk Show* and *SoccerWM*, respectively. Since *SoccerWM* is more "dynamic" than *Talk*

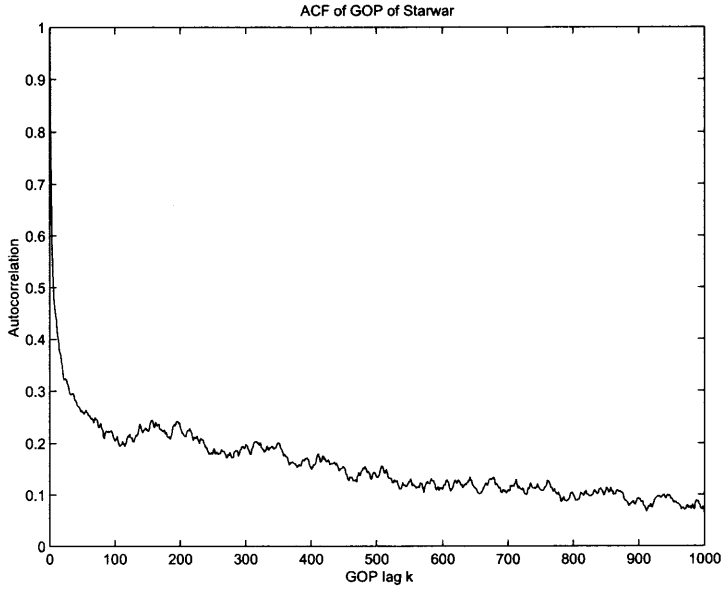

Figure 3.7 Autocorrelation of the GOP for *Star Wars.*

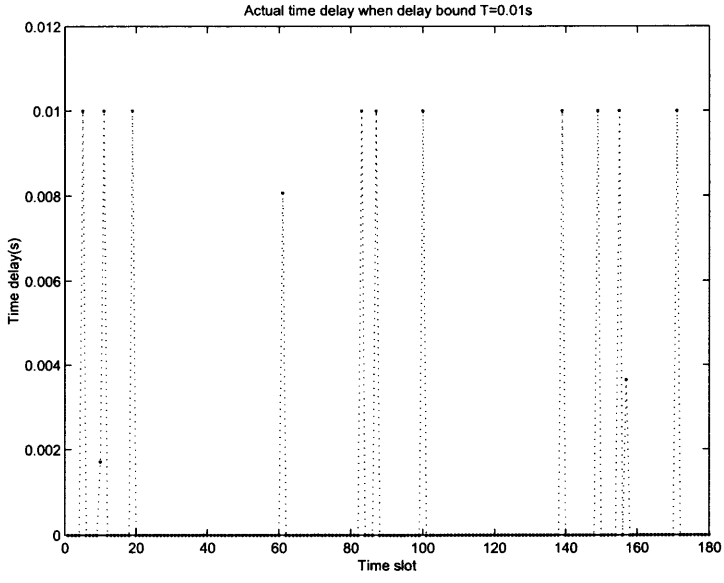

Figure 3.8 Actual delay when $\rho = 0.8$ and $T = 0.01s$ for *Star Wars.*

Show, the renegotiation frequency in *SoccerWM* is expected to be higher than that in *Talk Show* under the same bandwidth utilization and delay requirements. The renegotiation frequency for *Talk Show* and *SoccerWM* is 376 and 755, respectively, and the achieved bandwidth utilization is $\hat{\rho} = 0.9450$ and $\hat{\rho} = 0.9099$, respectively. Table 3.1 presents the simulation results of the impact of the maximum delay and the required bandwidth utilization on the renegotiation frequency. The average queue size is decreased with decreasing required bandwidth utilization and decreasing delay as shown in Fig. 3.13.

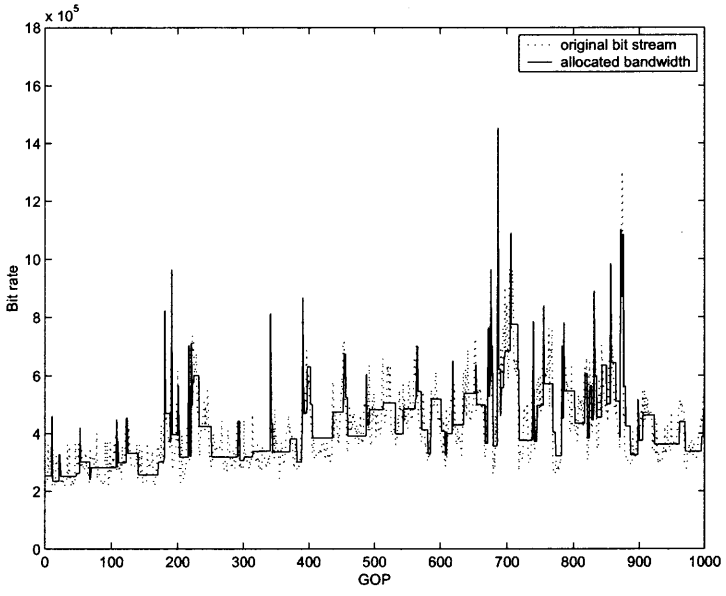

Figure 3.9 Actually allocated bandwidth when $\rho = 0.9$ and $T = 0.1s$ for *Star Wars*.

3.2.3 Performance Analysis

The QoS guaranteed bandwidth allocation scheme allocates the bandwidth based on the delay bound and required resource utilization. Note from Table 3.1 that the achieved average bandwidth utilization is higher than the required most of the time. Only in the case that the delay is very stringent and high utilization is required, is

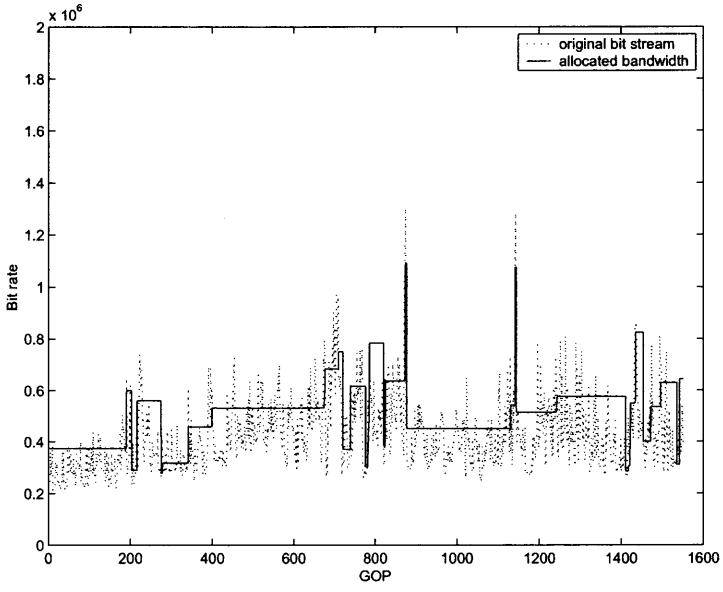

Figure 3.10 actually allocated bandwidth when $\rho = 0.5$ and $T = 0.2s$ for *Star Wars*.

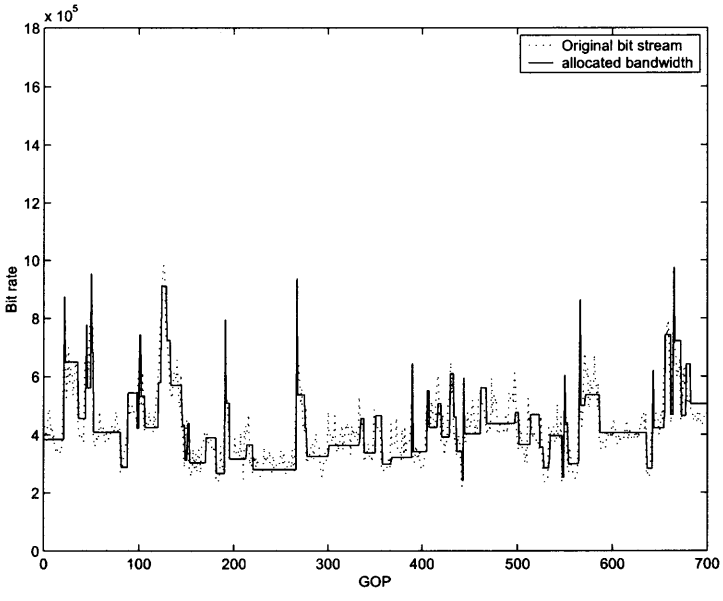

Figure 3.11 Actually allocated bandwidth when $\rho = 0.8$ and $T = 0.1s$ for *Talk Show*.

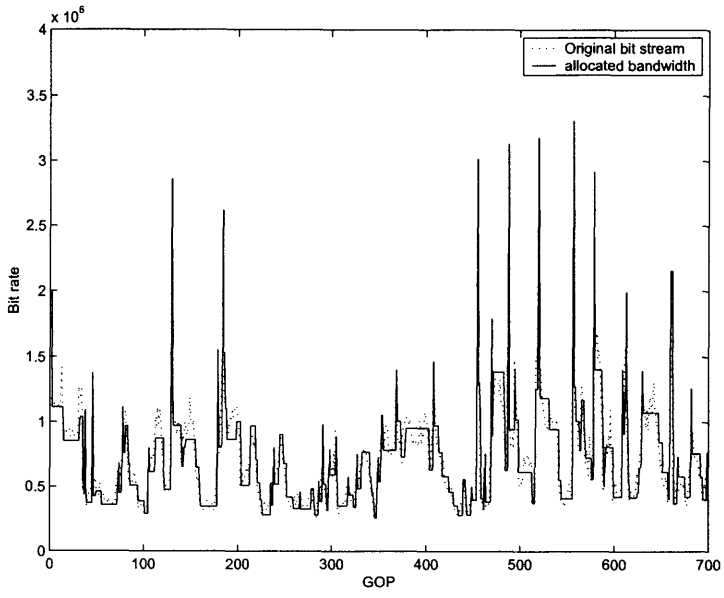

Figure 3.12 Actually allocated bandwidth when $\rho = 0.8$ and $T = 0.1s$ for *SoccerWM*.

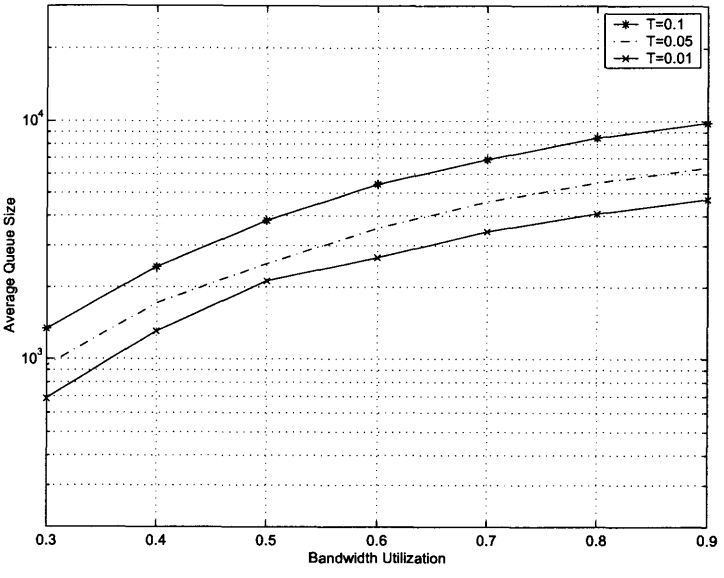

Figure 3.13 Average queue size vs different average utilization for *Star Wars*.

Table 3.1 The Number of Renegotiation When $T = 0.05$ and $T = 0.01$

ρ	0.9	0.8	0.7	0.6	0.5	0.4	0.3
FREQ.($T = 0.01$)	5318	3983	3008	2147	1536	914	430
FREQ.($T = 0.05$)	4392	3085	2164	1432	909	533	271
Achieved Util.($T = 0.01$)	.8516	.8439	.8292	.8014	.7659	.6922	.6059
Achieved Util.($T = 0.05$)	.9040	.8893	.8644	.8233	.7677	.6989	.5911

the average utilization lower than the required. This is attributed to the additional flexibility provided by another constraint, delay. If the required delay is not satisfied, it can trigger a negotiation to adapt the required bandwidth. The delay is met at the cost of the bandwidth utilization.

The probability distribution of the prediction error and the corresponding Gaussian-fit distributions are shown in Fig. 3.14. Note that the prediction error is Gaussian-like. The PDF for the Gaussian random variable X is given by

$$f_X(x) = \frac{1}{\sqrt{2\pi}\,\sigma} e^{\frac{-(x-m)^2}{2\sigma^2}} \quad -\infty < x < \infty. \tag{3.10}$$

The predictor either underestimates the bandwidth ($\epsilon_n < 0$), overestimates the bandwidth ($\epsilon_n > 0$), or accurately estimates the bandwidth ($\epsilon_n = 0$), where ϵ_n is the prediction error. Thus, $R_o = R_p + \epsilon_n$. Then

$$R - R_o = R - (R_p + \epsilon_n) < (1 - \rho)R \tag{3.11}$$

$$R - R_p < (1 - \rho)R - \epsilon_n. \tag{3.12}$$

The difference between $R - R_o$ and $R - R_p$ is the prediction error which is Gaussian-like with Gaussian distribution. For the target bandwidth utilization ρ, we

use $R - R_p$ instead of $R - R_o$ to decide whether to change the current transmission rate in our algorithm; since R_o is not known at that time, we use the predicted value to replace it. The achieved bandwidth utilization in our algorithm is changed to

$$\rho' = \frac{R - (1-\rho)R - \epsilon_n}{R} = \rho - \frac{\epsilon_n}{R}. \tag{3.13}$$

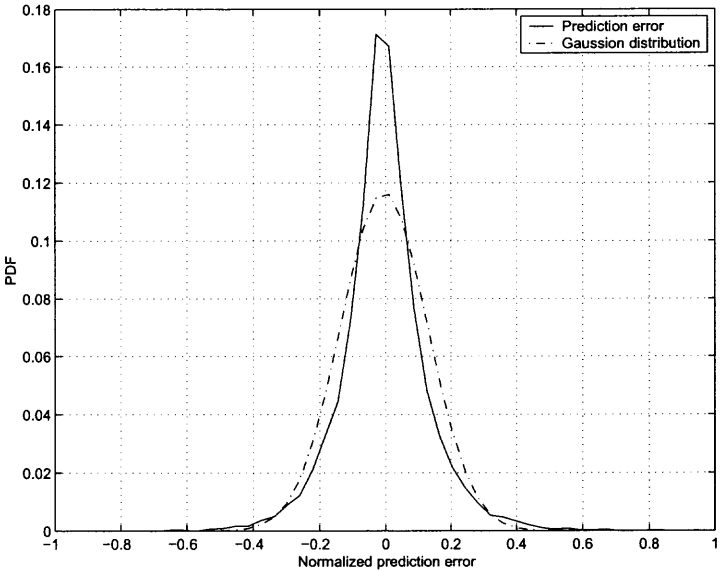

Figure 3.14 PDF of the prediction error for *Star Wars* and the corresponding Gaussian-fit distribution.

For the underestimated case $\epsilon_n > 0$, the achieved bandwidth utilization is lower than the required; for the overestimated case $\epsilon_n < 0$, the achieved bandwidth utilization is higher than the required. The variation of the average bandwidth utilization is $\frac{\epsilon_n}{R}$. The probability distribution of the random variable ϵ_n is symmetric as shown in Fig. 3.14. The probability of the random variable ϵ_n exceeding a threshold δ can be expressed as follows:

$$P[|\epsilon_n| \geq \delta] = \frac{1}{\sqrt{2\pi}\sigma}[\int_{-\infty}^{-\delta} e^{-\frac{(\epsilon_n - m)^2}{2\sigma^2}} d\epsilon_n +$$

$$\int_{\delta}^{\infty} e^{-\frac{(\epsilon_n - m)^2}{2\sigma^2}} d\epsilon_n] = 2Q(\frac{\delta - m}{\sigma}), \tag{3.14}$$

where m and $\sigma > 0$ are the mean and standard deviation of random variable ϵ_n and the Q-function is defined by

$$\begin{aligned} Q(x) &= \frac{1}{\sqrt{2\pi}} \int_{x}^{\infty} e^{-\frac{t^2}{2}} dt \\ &\simeq \frac{1}{(1-a)x + a\sqrt{x^2 + b}} \frac{1}{\sqrt{2\,\pi}} e^{\frac{-x^2}{2}} \end{aligned} \tag{3.15}$$

where $a = 1/\pi$ and $b = 2\pi$ [31].

The larger the threshold δ, the smaller the probability $P[\epsilon_n > \delta]$. The change of the bandwidth utilization $\frac{\delta}{R}$ is much smaller compared with δ, and at the same time the probability distribution of ϵ_n is symmetric, and thus this effect on utilization may be neglected. This can be proved from the statistical perspective. The expected value of $\rho^{'}$ is expressed as follows:

$$E[\rho^{'}] = E[\rho - \frac{\epsilon_n}{R}] = \rho - \frac{1}{R} \times E[\epsilon_n]$$

$$E[\epsilon_n] = 0$$

$$E[\rho^{'}] = \rho \tag{3.16}$$

Thus, the required bandwidth utilization can be achieved from the statistical perspective. Actually, the average utilization is higher than the required if the delay requirement is not very stringent. This is due to the algorithm which sets a threshold of the utilization, which, if exceeded, triggers a renegotiation. If the delay constraint is very stringent, the delay can be guaranteed at the cost of decreasing the bandwidth utilization as shown in Table 3.1. Note from Table 3.1 that the achieved average bandwidth utilization is higher than the required and the renegotiation frequency

can be reduced by decreasing the required bandwidth utilization or increasing the delay.

3.3 Summary

Dynamic bandwidth allocation based on predicted I frames is first proposed in this chapter. As I frame is the largest in a GOP most of time, the CLR can be kept small. From the buffering standpoint, instead of buffering highly correlated input traffic directly, our proposed scheme buffers the residuals (errors) of the prediction, a rather "uncorrelated" process resembling white noise, thus requiring much smaller buffer space.

In order to further improve bandwidth utilization and guarantee quality of service, a delay guaranteed bandwidth allocation is also proposed in this chapter. In this scheme, the bandwidth is allocated based on the predicted GOP, delay bound, and bandwidth utilization. Simulation and analytical results demonstrate that this scheme does provide the guaranteed delay with high bandwidth utilization, and thus can be used for on line real time video delivery.

CHAPTER 4

SELF-SIMILAR TRAFFIC PREDICTION USING LEAST MEAN KURTOSIS ALGORITHM

Recent studies of high quality, high resolution traffic measurements have revealed that network traffic appears to be statistically self similar. Contrary to the common belief, aggregating self-similar traffic streams can actually intensify rather than diminish burstiness. Thus, traffic prediction plays an important role in network management. In this chapter, Least Mean Kurtosis (LMK), which uses the negated kurtosis of the error signal as the cost function, is proposed to predict the self similar traffic. Simulation results show that the prediction performance is improved greatly over the Least Mean Square (LMS) algorithm.

4.1 Introduction

Recent studies have shown that aggregate Internet traffic does not comply to the Poisson model. It exhibits long term correlations which cannot be modeled by a Markov model. Leland *et al.* [32] analyzed the Ethernet traffic data and showed that the generally accepted argument for the "Possion-like" nature of aggregate traffic that aggregate traffic becomes smoother as the number of traffic sources increases has very little to do with reality. "Self-similar" or "fractal like" processes can describe more effectively the actual network traffic. A self similar phenomenon exhibits structural similarity across all (or at least a wide range) of time scales. For self similar traffic, there is no natural length for a burst; traffic bursts appear on a wide range of time scales. From a mathematical point of view, self similar traffic differs from other traffic models in the following way [33]. Let s be a time unit representing a time scale, such as $s = 10^m$ seconds ($m = 0, \pm 1, \pm 2 \ldots$). For every time scale, let $X^{(s)} = X_n^{(s)}$ denote the time series computed as the number of units (packets, bytes, cells, etc)

per time unit s in the traffic stream. Traditional traffic models possess the property that as s increases, the "aggregated" process, $X^{(s)}$, tends to be a sequence of *i.i.d.* random variables (covariance stationary white noise). However, for a self-similar process, they either appear visually indistinguishable from one another ("exactly self-similar") but distinctively different from pure noise, or they converge to a time series with a non-degenerate autocorrelation structure ("asymptotically self-similar"). A mathematical definition of a self-similarity process will be given in Section 4.2.1.

Two formal mathematical models that yield elegant representations of the self-similarity phenomenon are the Fractional Gaussian Noise (FGN) and the Fractional Autoregressive Integrated Moving Average (FARIMA) processes. Since the FARIMA process is much more flexible with regard to the simultaneous modeling of the short term and long term behavior of a time series than the FGN process, the FARIMA process is adopted here to simulate the network traffic.

The prediction of network traffic plays an important role in resource allocation and network management. Many types of traffic have the property of Long Range Dependence (LRD), and aggregated Internet traffic also shows long term correlations. The higher correlation in the time domain, the longer the mean queue size, and thus the delay will be long. From the network perspective, the key point is to assign the link capacity to a given traffic in order to provide guaranteed quality of service. By prediction, we can not only achieve this but also keep the bandwidth utilization high. In this chapter, we propose to predict the self similar traffic by the least mean kurtosis algorithm. This prediction is based on a higher order statistics rather than the second order statistics used in the LMS algorithm. Simulation results show that this LMK algorithm achieves better performance than LMS in predicting self similar traffic generated by the FARIMA model. In Section 4.2, self-similar traffic and its widely used models are introduced; the LMK algorithm is also proposed to predict

the self similar traffic in this section. Section 4.3 presents the performance analysis of our proposed scheme.

4.2 Traffic Prediction

The ability to predict traffic within a network is one of the fundamental requirements of network design and management. The prediction quality depends on the amount of uncertainty that accompanies the prediction and the nature of traffic itself. Prediction must be as accurate as possible so that bandwidth and buffer resources are not wasted and at the same time QoS can be guaranteed. The LMK algorithm is proposed to predict the self similar traffic, and is shown to outperform the LMS algorithm.

4.2.1 The Self Similar Traffic Model

A self-similarity time series has the property that when aggregated, the new short time series has the same autocorrelation function as the original. Each point in the short time series is the sum of multiple original points. The self-similar process is defined as follows [32]: Let $X = (X_t : t = 0, 1, 2, \cdots)$ be a covariance stationary stochastic process with mean μ, variance σ^2, and autocorrelation function $r(k), k > 0$. In particular, we assume that X has an autocorrelation function of the form

$$r(k) \sim k^{-\beta} L(t) \text{ as } k \to \infty, \tag{4.1}$$

where $0 < \beta < 1$ and L is slowly varying at infinity. For our discussion below, we assume for simplicity that L is asymptotically constant. For each $m = 1, 2, 3 \cdots$, let $X^{(m)} = (X_k^{(m)} : k = 1, 2, 3 \cdots)$ denote the new covariance stationary time series obtained by averaging the original series X over non-overlapping blocks of size m. That is, for each $m = 1, 2, 3, \cdots$, $X^{(m)}$ is given by

$$X_k^{(m)} = \frac{1}{m}(X_{km-m+1} + \cdots + X_{km}), k \geq 1.$$

Its corresponding autocorrelation function is $r^{(m)}$. The process is called (exactly) second-order self-similar with self-similarity parameter $H = 1 - \beta/2$, if for all $m = 1, 2, \cdots$, $var(X^{(m)}) = \sigma^2 m^{-\beta}$ and

$$r^{(m)}(k) = r(k) \text{ as } k \geq 0. \tag{4.2}$$

X is called (asymptotically) second-order self-similar with self-similarity parameter $H = 1 - \beta/2$ if for all k large enough,

$$r^{(m)}(k) \to r(k), \text{ as } m \to \infty, \tag{4.3}$$

with $r(k)$ given by Eq. (4.1). Intuitively, the most striking feature of (exactly or asymptotically) second-order self similar process is that their aggregated process $X^{(m)}$ possesses a non-degenerate correlation structure, as $m \to \infty$.

In practice, traffic model is often used to simulate the network traffic. An important requirement of practical traffic modeling is to generate synthetic data sequences that exhibit similar features as the measured traffic. For self-similar traffic, there are two formal mathematical models: FGN and FARIMA processes, that generate elegant representations of the self similarity phenomenon. Since the FARIMA process is more accurate in simulating network traffic than FGN, we adopt FARIMA to generate traffic used in this chapter.

The FARIMA(p, d, q) process, an extension to ARIMA(p, d, q), is defined as [34]:

$$\phi(B)\nabla^d X_t = \theta(B)\epsilon_t, \tag{4.4}$$

where d is the indicator for the strength of long range dependence and assumes the value between 0 and 1/2. ϵ_t is a Gaussian white noise, and

$$\phi(B) = 1 - \phi_1 B - \phi_2 B^2 - \cdots \phi_p B^p$$

$$\theta(B) = 1 - \theta_1 B - \theta_2 B^2 - \cdots \theta_q B^q$$

are polynomials of degree p and q, respectively, in the backward shift operator B. The operator $\nabla^d = (1-B)^d$ can be expressed by the binomial expansion

$$(1-B)^d = \sum_{k=0}^{\infty} \binom{d}{k} (-1)^k B^k, \tag{4.5}$$

$$\binom{d}{k} = \frac{d!}{k!(d-k)!} = \frac{\Gamma(d+1)}{\Gamma(k+1)\Gamma(d-k+1)}, \tag{4.6}$$

where $\Gamma(x)$ denotes the gamma function; note that for all positive integers, only the first $d+1$ terms are non-zero in Eq. (4.6). The FARIMA$(0,d,0)$ process with $0 < d < 1/2$, is stationary and long range dependence with an autocorrelation function

$$\rho_k = \frac{\Gamma(1-d)\Gamma(k+d)}{\Gamma(d)\Gamma(k+1-d)} \sim \frac{\Gamma(1-d)}{\Gamma(d)} k^{2d-1} \text{ as } k \to \infty. \tag{4.7}$$

To generate FARIMA traffic data, the FARIMA process can be approximated by the linear process in the form of [35]

$$X_k = \sum_{i=0}^{I} c_{k-i}\epsilon_i, \tag{4.8}$$

where ϵ_i is an $i.i.d$ random variable. ϵ_i may be Gaussian or non-Gaussian. For Gaussian FARIMA(0,d,0),

$$c_k = \frac{\Gamma(k+d)}{\Gamma(d)\Gamma(k+1)}. \tag{4.9}$$

c_k can be iteratively obtained as follows

$$c_0 = 1, \tag{4.10}$$

$$c_{k+1} = \frac{k+d}{k+1} \cdot c_k. \tag{4.11}$$

$H = d + 0.5$, and thus we can generate FARIMA traffic according to the parameter H. In our simulations, the FARIMA traffic with $H = 0.8$ is used to evaluate the performance of the LMK predictor.

4.2.2 The LMK Algorithm

Since the FARIMA model can yield elegant representations of the self similarity phenomenon, and it is more flexible than FGN with regard to the simultaneous short-term and long-term behavior of a time series, FARIMA is used to simulate the network traffic.

Let $X(n)$ be the time series traffic generated by the FARIMA model. A *pth*-order linear predictor has the form

$$\hat{x}(n+1) = \sum_{l=0}^{p-1} h(l)x(n-l), \tag{4.12}$$

$$e(n) = \hat{x}(n) - x(n) = \mathbf{h}^T\mathbf{X} - x(n), \tag{4.13}$$

where $\mathbf{h}$, $\mathbf{X}$ are vectors of adaptive filter coefficients and input signal, respectively. Let

$$\mathbf{h} = [h(0), h(1), \cdots h(p-1)]^T,$$

$$\mathbf{X} = [x(n-1), x(n-2), \cdots x(n-p)]^T.$$

In the LMK algorithm, the cost function is defined to be the negated kurtosis [36]:

$$\begin{aligned} J_{LMK}(\mathbf{h}) &= 3E^2[e(n)^2] - E[e^4(n)] \\ &= 3[E(\mathbf{h}^T\mathbf{X} - x(n))^2]^2 - E[\mathbf{h}^T\mathbf{X} - x(n)]^4. \end{aligned} \tag{4.14}$$

Taking the gradient with respect to the vector $\mathbf{H}$,

$$\nabla J_{LMK}(\mathbf{h}) = 12E[\mathbf{h}^T\mathbf{X} - x(n)]^2 E[\mathbf{h}^T\mathbf{X} - x(n)]\mathbf{X}$$

$$- 4E[\mathbf{h}^T\mathbf{X} - x(n)]^3\mathbf{X}$$

$$= 4\{3E[e^2(n)]E[e(n)] - 4E[e^3(n)]\}\mathbf{X}. \quad (4.15)$$

The mean value $E[\mathbf{h}^T\mathbf{X} - x(n)]^2$ will be estimated by the following recursive equation:

$$G(n) = \beta G(n-1) + (1-\beta)e^2(n). \quad (4.16)$$

Using this estimate and the ensemble estimate of $E[\mathbf{h}^T\mathbf{X} - x(n)]$, $\nabla J_{LMK}(\mathbf{h})$ can be expressed as the following equation:

$$\tilde{\nabla} J_{LMK}(\mathbf{h}(n)) = 4[3G(n) - e^2(n)]e(n)\mathbf{X}. \quad (4.17)$$

According to the method of the steepest descent adaptive weight-update algorithm [37], LMK can be characterized by

$$\mathbf{h}(n+1) = \mathbf{h}(n) - \frac{1}{4}\mu\{\tilde{\nabla} J_{LMK}(\mathbf{h}(n))\}, \quad (4.18)$$

where μ is the step size, $\tilde{\nabla} J_{LMK}(\mathbf{h}(n))$ is an approximation of the gradient vector $\nabla J_{LMK}(\mathbf{h})$, $G(n)$ is an iterative approximation of $E[\mathbf{h}^T\mathbf{X} - x(n)]$, and β is the forgetting factor that controls the memory of the error power estimator. $\beta = 0.7$ is empirically found to work well in all our simulations. Eq. (4.18) can further be normalized as follows:

$$\mathbf{h}(n+1) = \mathbf{h}(n) + \frac{\mu[3G(n) - e^2(n)]e(n)\mathbf{X}}{(\mathbf{X}^T\mathbf{X})^2}. \quad (4.19)$$

Tanrikulu *et al.* [36] have compared the computational complexities of LMS and LMK, and found that LMS requires $O[2N+1M, N+1A]$ and LMK requires $O[2N+5M, N+3A]$ where N is the number of adaptive coefficients, M denotes multiplication, and A denotes addition. Therefore, only four extra multiplications and two extra additions which are independent of N are necessary for the LMK algorithm.

Table 4.1 Comparison of the SNR^{-1} Performance of LMK and LMS Predictors on Self Similar Traffic

	SNR^{-1}										
	1	2	3	4	5	6	7	8	9	10	Ave.
LMK	.0156	.0169	.0152	.0161	.0156	.0170	.0154	.0202	.0170	.0159	.0165
LMS	.0398	.0385	.0410	.0370	.0424	.0376	.0421	.0415	.0349	.0422	.0397

4.3 Performance Analysis and Comparison

A simulation is conducted on the self-similar traffic generated by the FARIMA model. Since the self similar process has the the property of long range dependence, and the history of long range dependence process has significant impact on the present value of the process, it is natural to assume that the longer the dependence, the better the prediction. Östring and Sirisena [38] considered the prediction of long range dependent process and demonstrated that long-range dependence has only marginal value in improving prediction. It is the *short term correlation* within the structure of a self similar process rather than the long term correlation that dominates the performance of the predictors. So the Akaike information criterion is used to choose the best order not greater than 12. The AIC associates a cost function with the order of the filter. It was found by numerous simulations that the autocorrelation of the prediction error $e(n)$ is close to that of the white noise. Thus, one-step, 12-order adaptive filter is used for both the LMK and LMS algorithm. The performance of the algorithm is quantified by the inverse Signal to Noise Ratio ($SNR^{-1} = \frac{\sum e^2(n)}{\sum x^2(n)}$). Table 4.1 shows the performance comparison of LMK based and LMS based predictors.

The Hurst parameter of the generated self-similar traffic is $H = 0.8$; this simulation is repeated 50 times, and the resulting ensemble averaged SNR^{-1} is plotted

for LMK and LMS in Fig. 4.1. Note that the step sizes are chosen as $\mu_{LMS} = 0.6$, $\mu_{LMK} = 0.7$ that provide the least mismatch for each algorithm, in other words, the least SNR^{-1}. The forgetting factor in LMK is chosen to be $\beta = 0.7$ which provides less mismatch and faster convergence. In Fig. 4.1, it is clear that LMK not only converges as quickly as LMS but also produces significantly less prediction error; thus, LMK outperforms LMS in predicting the self similar traffic. Table 4.1 shows the performance comparison of SNR^{-1} for LMK and LMS. Note that LMK incurs smaller prediction error in all experiments than LMS; ten results and their average values are listed here. Thus, the performance has improved greatly if the LMK algorithm is used instead of the LMS algorithm to predict the self similar traffic.

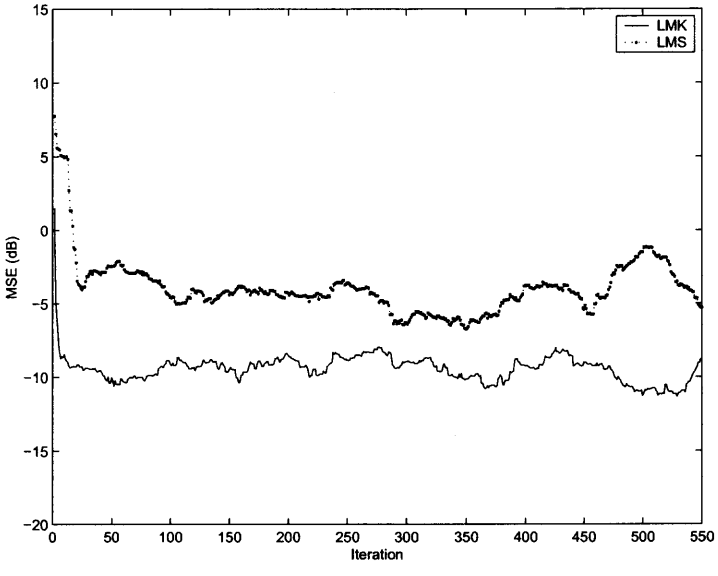

Figure 4.1 Averaged output SNR^{-1} versus number of iteration for self similar traffic

4.4 Summary

Aggregate Internet traffic does not comply to the Poisson model, but can be more effectively described by the self-similar process. In this chapter, the LMK adaptive algorithm which uses the negated kurtosis of the error signal as the cost function to

predict the self-similar network traffic generated by the FARIMA model is proposed. Simulation results show that LMK incurs much smaller prediction error as compared with LMS. Since the prediction performance can be improved greatly with only a small extra computation, LMK can be used to effectively predict the real time network traffic.

CHAPTER 5

REAL TIME VIDEO TRANSMISSION OVER DIFFSERV

Multimedia applications require high bandwidth and guaranteed quality of service. The current Internet, which provides "best effort" services, cannot meet the stringent QoS requirements for delivering MPEG videos. In this chapter, we first propose to transport single layer MPEG videos through various service models of DiffServ; then by considering QoS from both encoding side and network side, we further propose to transport spatial scalable encoded MPEG videos over DiffServ. Performance analysis and simulation results show that our proposed approaches can not only guarantee QoS but can also achieve high bandwidth utilization.

5.1 The Current Internet

Internet offers "best effort" service, in which the network allocates bandwidth among all of the instantaneous users as best as it can and attempts to service all of them without making any explicit commitment as to the rate or any other service quality [39]. Real time applications especially for MPEG video with bursty characteristics often do not work well across the best effort service because of variable queuing delays and congestion losses. The trend of recent and current developments in IP networks is towards supporting end to end QoS. Both telecommunications industry and research community have spent a great deal of effort on investigating and developing network technologies that could provide QoS over IP based networks. The first attempt focuses on providing an automatic optimization of IP traffic over switched based networks such as ATM (e.g., MPOA, IP switching). However, the main disadvantage of these approaches is that the application software does not have an interface which can control the specific capabilities of the underlying network. Another approach, proposed by the Internet Engineering Task Force, has focused on

provisioning QoS support to the Internet service model. Two service models have been proposed: the Integrated Service and Differentiated Service models.

5.1.1 Integrated Service

The Integrated Service model proposes two service classes in addition to Best Effort Service: guaranteed service for applications requiring fixed delay bound, and controlled load services for applications requiring reliable and enhanced best effort service. It integrates these services with controlled link-sharing, and it is designed to work well with multicast as well as unicast. The philosophy of this model is that "there is an inescapable requirement for routers to be able to reserve resources in order to provide QoS for specific user packet streams, or flows. This in turn requires flow-specific state in routers" [2]. IntServ uses the Resource Reservation Protocol as a working protocol for signaling in the IntServ architecture as shown in Fig. 5.1. In order to make a reservation at a node, the RSVP daemon calls two local decision procedures, admission control and policy control. Admission control determines whether the node has sufficient available resources to supply the requested QoS. Policy control determines whether the user has the administrative permission to make the reservation. If either check fails, the RSVP program returns an error notification to the application process that originated the request. If both checks succeed, the RSVP daemon sets parameters in a packet classifier and packet scheduler to obtain the desired QoS. The packet classifier determines the QoS class for each packet and the scheduler orders packet transmission to achieve the promised QoS for each packet.

This protocol assumes that resources are reserved for every flow requiring QoS at every router hop in the path between the receiver and the transmitter, using end to end signaling and must maintain a per-flow "soft state" at every router in the network. "Soft state" regards the reservation state as cached information that is installed and

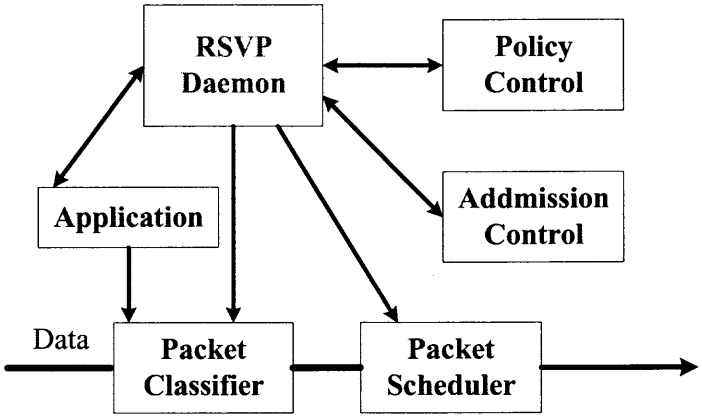

Figure 5.1 Reservation process at a node.

periodically refreshed by the end hosts. Unused state is timed out by the routers. If the route changes, the refresh messages automatically install the necessary state along the new route [2]. Thus, every router has to maintain soft state for every flow that passes through it. This makes IntServ considerably more complex and difficult for a large user population. Because of this, Differentiated Service is introduced.

5.1.2 Differentiated Service

DiffServ is an IETF recommendation, which allows different QoS levels to different classes of aggregated traffic flows, as opposed to individual flows. Traffic entering a network is classified and possibly conditioned at the boundaries of the network, and assigned to different behavior aggregates. Each aggregate behavior is identified by a single Differentiated Service Code Point (DSCP). Within the core of the network, packets are forwarded according to the Per-Hop Behaviors (PHBs) associated with DSCP. The per hop behavior determines the priority, the maximum delay in the transmission queues, the link sharing bandwidth, and the probability of a packet to be dropped. There are two types of PHB: Assured Forwarding (AF) and Expedited Forwarding (EF). Traffic that is characterized as EF will receive the lowest latency, jitter, and assured bandwidth services. The AF PHB group provides delivery of IP

packets in four independently forwarded AF classes, where each AF class is allocated a certain amount of forwarding resources in each Differentiated Service (DS) node [40].

The DiffServ model ensures high scalability by separating the operations performed at the borders of the network from those accomplished in the core network. The DiffServ architecture is shown in Fig. 5.2. Figure 5.3 shows the edge router functions. Edge routers perform more complex tasks such as traffic classifying, marking and shaping. The classifier is used to select packets based on the content of packet headers according to the defined rules. Marking is the process of setting the DS code point in a packet based on the defined rules. Shaping is the process of delaying packets within a traffic stream to make it conform to some defined traffic profile. The temporal properties of a traffic stream selected by a classifier are measured in the meter. The instantaneous state of this process may be used to affect the operation of a marker, shaper, or dropper. Prioritized routing will be performed within core routers. Bandwidth brokers perform admission control, manage network resources, and configure leaf and edge routers. The function in core routers is to decide which packet is to be sent to the next hop, in which order and at which rate. Queuing disciplines are deployed in core routers.

In order for a customer to receive differentiated services from its Internet Service Provider (ISP), it must have a Service Level Agreement (SLA) with its ISP. A SLA basically specifies the service classes supported and the amount of traffic allowed in each class. A SLA can be static and dynamic. Static SLAs are negotiated on a regular basis, e.g. monthly and yearly. Customers with dynamic SLAs must use a signaling protocol, e.g., RSVP, to request for services on demand.

The key difference between IntServ and DiffServ is that IntServ provides end to end QoS service on a per-flow basis, while DiffServ is intended to provide service differentiation among the traffic aggregates to different users over a long time scale,

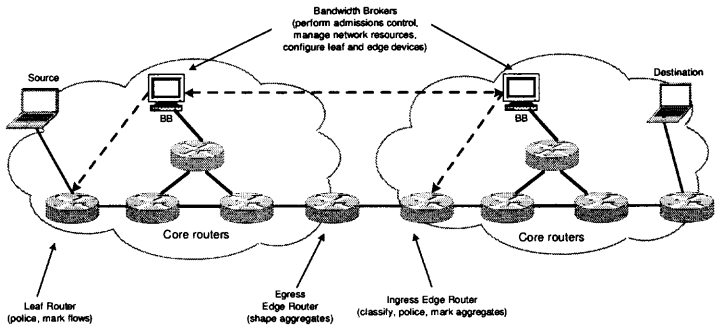

Figure 5.2 DiffServ Architecture

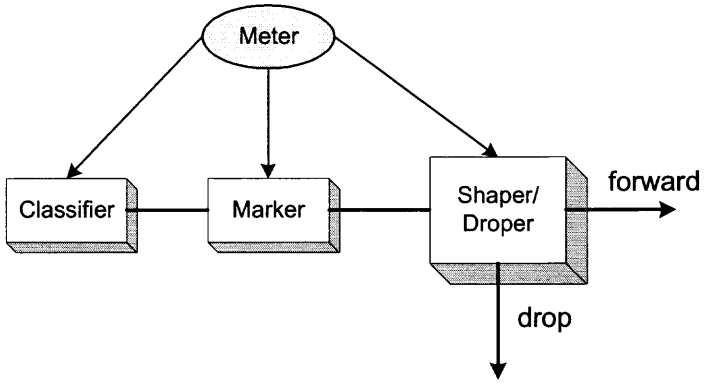

Figure 5.3 Edge router function

and is thus more scalable and less complex than IntServ. Here, we propose to transmit real time videos over DiffServ using EF and AF class, i.e., I frame packets by EF PHB, and P and B frames by AF PHB.

5.2 Transporting Single Layer MPEG Videos over DiffServ

MPEG employs both intraframe and interframe coding techniques for compression. To decode a B frame, both the previous and future I and P frames are needed; to decode a P frame, the previous P or I frame is needed. Thus, different frames should be treated differently.

5.2.1 Transporting Architecture

DiffServ provides two types of PHB: EF and AF. The EF PHB, often referred to as the Premium service [40] [41], is defined as a forwarding treatment for a particular DiffServ aggregate where the departure rate of the aggregate's packets from any DiffServ node must equal or exceed a configurable rate. The EF traffic should receive this rate independent of the intensity of any other traffic attempting to transit the node [41]. The AF PHB group provides delivery of IP packets in four independently forwarded AF class (AF1, AF2, AF3, AF4). Within each AF class, an IP packet can be assigned one of three different levels of drop precedence. A DS node must allocate a configurable, minimum amount of forwarding resources to each implemented AF class. Thus in a DS node, the level of forwarding assurance of an IP packet depends on

- How much forwarding resource has been allocated to the AF class that the packet belongs to.
- What the current load of the AF class is.
- In case of congestion within the class, what the drop precedence is.

In the following, we assume that one frame is packetized into one packet, and there are three AF classes and one EF class in the DiffServ domain. The network topology is composed of two edge routers and one core router conforming with the IP DiffServ model. The edge router is responsible for classifying arriving packets to the appropriate class and connect the DS domain to other DS or non-DS-capable domains. Within the core router, packets are forwarded according to the per-hop behavior associated with the DS code point. In our scheme, $10Mbps$ link Ethernet is considered. AF1, AF2, AF3 are each allocated with a bandwidth of $2Mbps$ and the EF class is allocated $4Mbps$. Class Based Queuing (CBQ) [42] is used to manage EF and AF service classes so that each user can receive the appropriate resources based

on packet marking. The respective class of data packets is identified as a DSCP in the Internet Protocol header. Since I frames are the most important and largest of the three types of frames, our IP DiffServ MPEG video packet marking algorithm can be summarized as follows [43]:

- if the arriving packet belongs to I frame
  DSCP=EF
- if the arriving packet belongs to P frame
  DSCP=AF low drop precedence
- if the arriving packet belongs to B frame
  DSCP=AF medium drop precedence

Within the DiffServ domain, each DSCP is mapped to a PHB. The PHB specifies the service by which the data packets are transported to the next hop.

5.2.2 Simulation Results and Performance Evaluation

In MPEG, frames do not have the same importance as some frames depend on the others, and so we propose to assign different frames to different PHBs. In our scheme, I frames are transmitted by EF class, and P and B frames are transmitted by $AF1$ class with different drop precedence. Here, different drop precedence is treated as allocating different bandwidth to P and B frames, as P frames are more important than B frames and have large size. The excess bandwidth left in the $AF1$ class is allocated to P and B frames proportionately. Not only can QoS be guaranteed but a high bandwidth utilization can also be achieved by using this approach, as demonstrated by simulations on the 1.5 hour *Star. Wars* video.

Figure 5.4 shows the average delay of I frames at different EF class bandwidth. In our scheme, $4Mbps$ is allocated to the EF class with the average delay close to 0, so the QoS of I frames is guaranteed using the EF class, and at the same time the

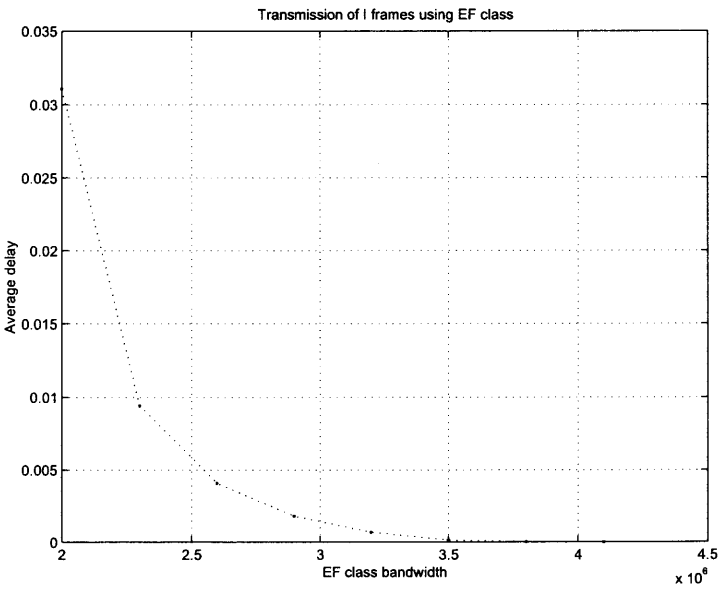

Figure 5.4 Average delay of I frames at different EF bandwidth

high bandwidth utilization can be achieved. Since an I frame is usually inserted into a stream approximately every half second, if we define one frame duration as a time slot ($33ms$ for NTSC, $40ms$ for PAL) and the number of frames in each GOP is 12, an I frame occupies only one out of 12 time slots; i.e., the EF traffic load attributed to I frames is rather small as shown in Fig. 5.5; other traffic flows in the EF can benefit from this. If we transmit all frames over EF, although QoS can be guaranteed, the bandwidth utilization becomes very low, because P and B frames are much smaller than I frames, and thus the bandwidth is wasted most of time. Figure 5.6 shows the transmission time of I frames using EF with a bandwidth $3.8Mbps$ for video trace *Star Wars*. Note that the delay of I frames can be guaranteed and the average delay is close to 0 as shown in Fig. 5.4.

P and B frames are transmitted by the AF1 class. P frames are allocated $0.8Mbps$ and B frames are allocated $0.4Mbps$. Since an AF class may also be configured to receive more forwarding resources than the minimum when excess resources are available either from other AF classes or other PHB groups [44]. Here,

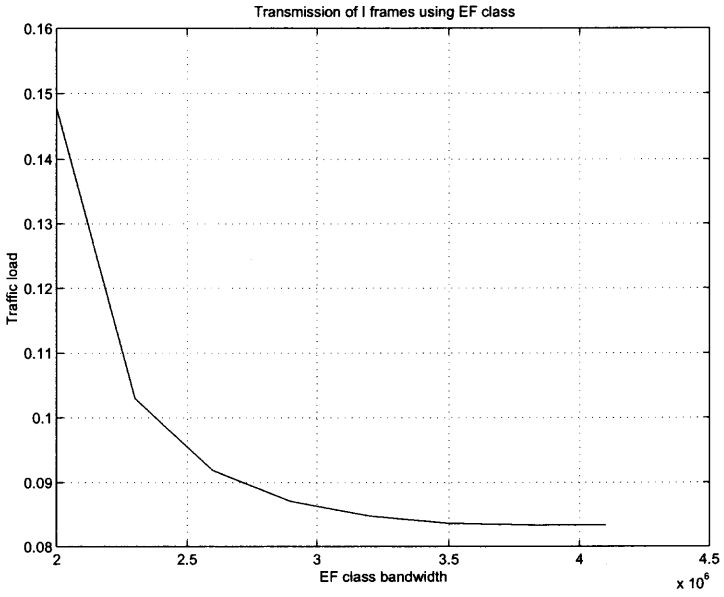

Figure 5.5 Traffic load of EF class when transmitting I frames

the algorithm to allocate the "excess" bandwidth left in the AF1 class to P and B frames according to their DSCP is proposed.

$$R_{eP} = R_P + q1 * R_{Left} * (1 - T_{load}) \tag{5.1}$$

$$R_{eB} = R_B + q2 * R_{Left} * (1 - T_{load}) \tag{5.2}$$

where R_{eP} and R_{eB} are the actual bandwidth allocated to P and B frames respectively; R_P and R_B are the minimum bandwidth allocated to P and B frames, respectively. R_{Left} is the leftover bandwidth of $AF1$ class, T_{load} is the current traffic load, and $q1, q2$ are weights for P and B, respectively. Figures. 5.7, and 5.8 show the average delay of I, P and B frames versus different traffic load. Note that the average delay for I frames, regardless of the traffic load, is close to 0, i.e., QoS is guaranteed, because the EF class provides the departure rate of the aggregate packets equal or exceed a configurable rate. The EF traffic should receive this rate independent of the

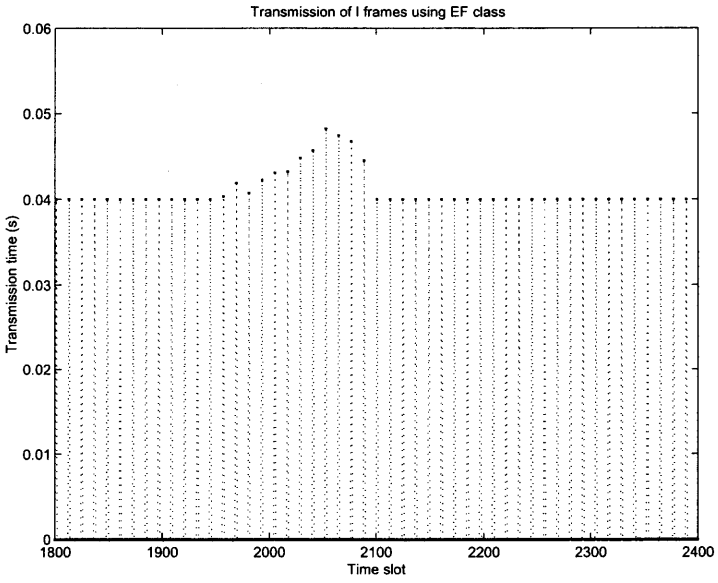

Figure 5.6 Transmission time of I frames for *Star Wars*

intensity of any other traffic attempting to transit the node. The average delay of P and B frames varies with the current traffic load and depends on how the excess resource left in the AF1 class is allocated to P and B frames. Figure 5.7 shows the average delay of P and B frames when $q1 = 0.8$, $q2 = 0.2$. If we want to decrease the average delay of P frames, we can adjust the values to $q1 = 0.9$, $q2 = 0.1$; the average delay is shown in Fig. 5.8. Fig. 5.9 shows the average delay of P and B frames with different weights.

Figure 5.10 shows the average delay of I, P, and B frames in the "best-effort" scenario, i.e., the current Internet. $4Mbps$ is allocated to the real time traffic. The average delay of I frame becomes larger with the current traffic load increasing, but it has less effect on the P and B frames because they have a smaller frame size. I frame is the most important of all the three frame types, as the other two types of frames depend on it; if the delay exceeds the playback time, the I frame will be deleted, and at the same time, the other frames which depend on I frames will be deleted. As the I frame has not been given the high priority in the "Best Effort" scenario, QoS cannot

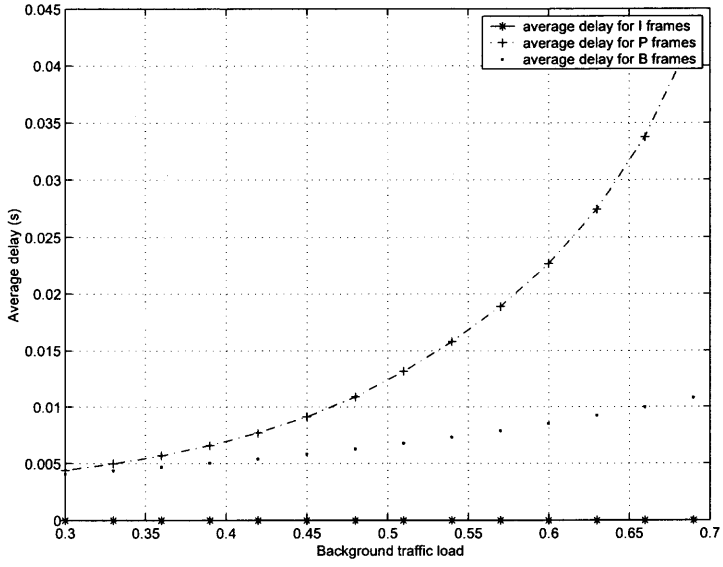

Figure 5.7 Average delay of I, P and B frames $q1 = 0.8, q2 = 0.2$

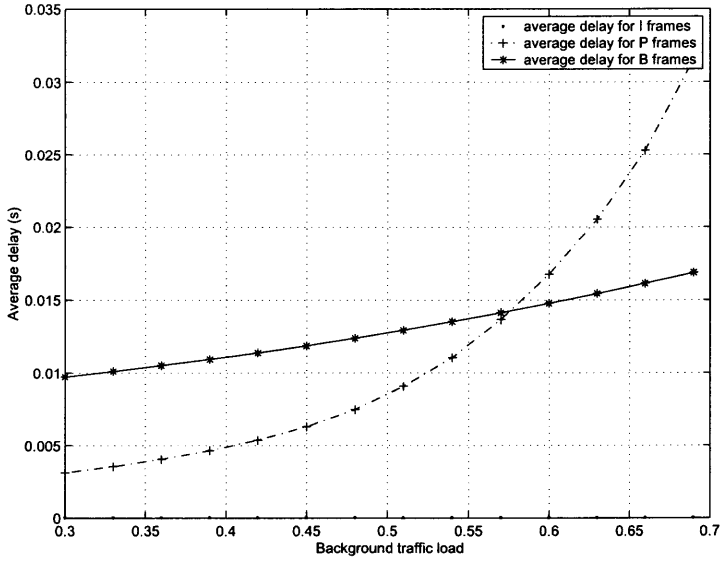

Figure 5.8 Average delay of I, P and B frames $q1 = 0.9, q2 = 0.1$

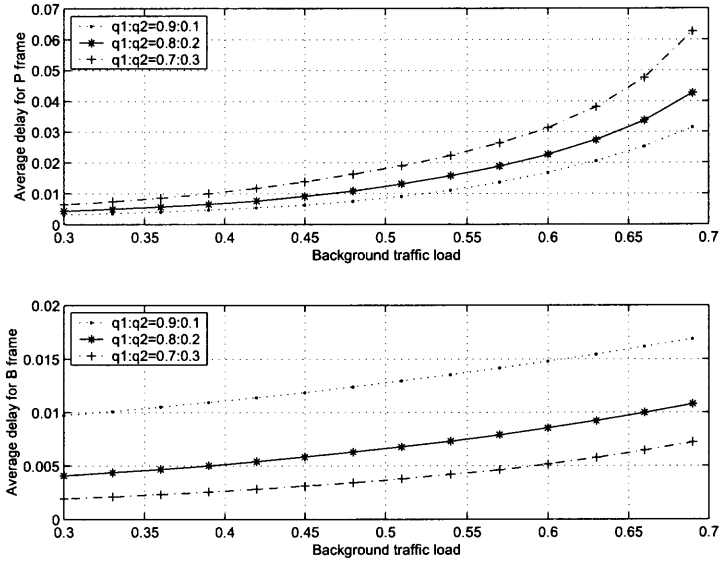

Figure 5.9 Average delay of P and B frames

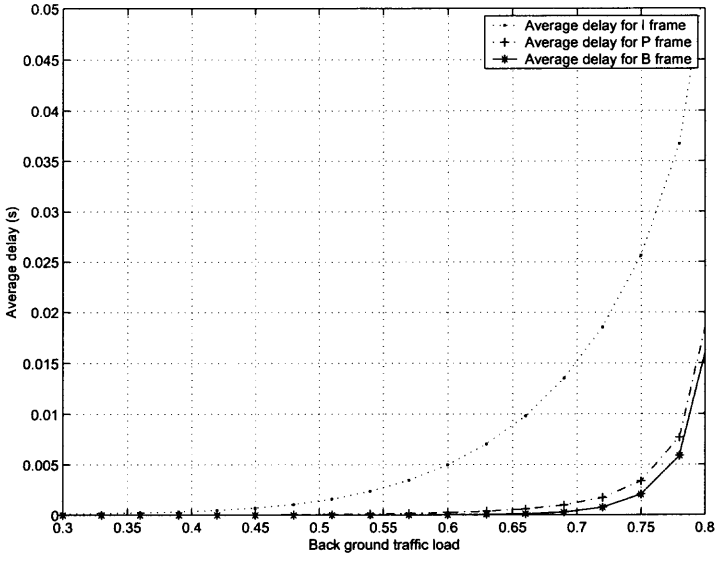

Figure 5.10 Average delay of I, P and B frames in the Best Effort scenario

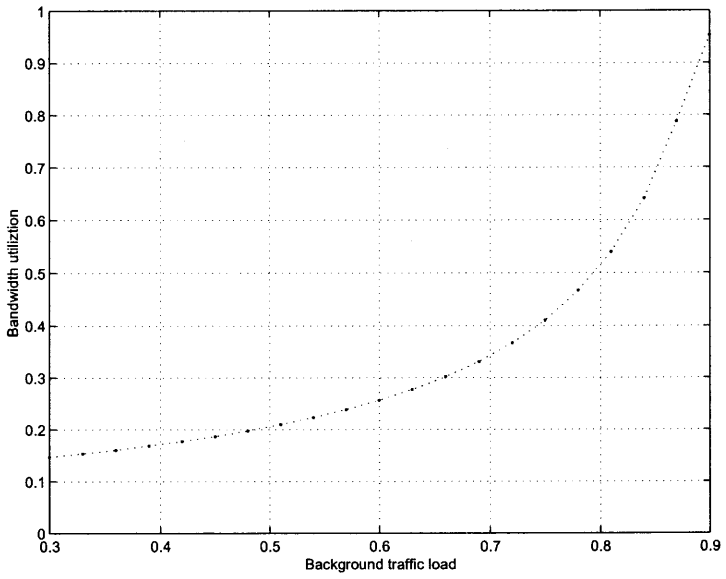

Figure 5.11 Bandwidth utilization in the Best Effort scenario

be guaranteed. If the traffic load is light in the "best-effort" network, we can achieve good QoS, but the bandwidth utilization is very low as shown in Fig. 5.11.

5.3 QoS Guaranteed Multiple Layer MPEG Video Transmission over DiffServ

The end quality of video depends not only on the network QoS parameters, but also on the encoded video quality. In order to provision QoS, we need to consider QoS from two aspects: network and encoding sides.

At network side, DiffServ differentiates QoS levels into different classes of aggregated traffic flows as discussed in Section 5.1.2. It provides scalable service without the need for per-flow state and signaling at every hop. Thus, DiffServ is used to transport layered videos.

At the encoding side, it is necessary to make an appropriate choice of a video compression algorithm that well matches the particular channel characteristics [45]. If the exact rate at which the network can carry information without any impairment

is known, then the best video quality will be achieved by compressing videos to fit exactly within that rate. Secondly, videos carried in the future networks will be to a large extent scalable encoded videos. Scalable encoded videos will dominate because they facilitate heterogeneous multimedia services over heterogeneous wire line and wireless networks. Thirdly, sometimes lowering the quality at the encoder but suffering few channel losses usually results in better decoded quality than allowing losses to occur during transmission, even if the encoded quality was superior in this high loss transmission. In other words, it is better to lower the quality of the videos at the encoder side, thus decreasing the bit rate of the videos during network congestion or severe network conditions, than to maintain high quality at the encoding side but to suffer high loss in which the change of the video quality cannot be controlled, but the change of the video quality at the encoder can be controlled. Thus, scalable video coding is introduced. There are three main types of scalability: spatial scalability, temporal scalability and SNR scalability. Spatial scalability is defined to support different picture resolutions in a single video stream. SNR scalability facilitates at least two different video qualities. The video information provided by the base layer can be improved by one or more enhancement layers carrying additional information. However, the base and enhancement layers have the same spatial video resolution. Temporal scalability is usually achieved by skipping certain frames of a video sequence.

In this scheme, the MPEG layered video traces provided by Martin Reisslein *et al.* [46] are used. Figure 5.12 shows the overview of encoding modes. Single layer (not scalable) encoding, temporal scalable encoding, and spatial scalable encoding are the three main categories of encoding modes. The spatial scalable encoded video traces adopted here are encoded into two layers: a base layer and an enhancement layer. The base layer provides video frames that are one fourth of the original size. Adding the enhancement layer to the base layer restores the original video frames. In order

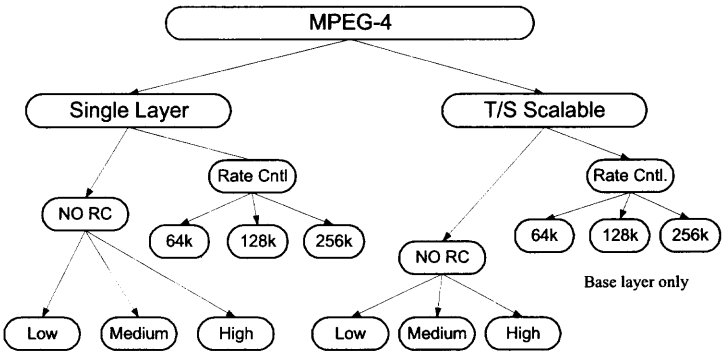

Figure 5.12 Overview of MPEG4 encoding modes

to reduce the video bit rate during periods of network congestion, the enhancement layers are either not transmitted or dropped from the severs or routers. Single layer encoded videos are also used to compare performance with layered videos through different networks.

5.3.1 Transport Architecture

The DiffServ architecture [47] is shown in Fig. 5.13 where the source and destination are connected through domains A and B. Each domain consists of a Bandwidth Broker (BB), core, edge, and leaf routers. The BB will exchange control messages with edge and leaf routers for the purpose of resource management. From this point on, leaf routers are interchangeably referred to as edge routers. The edge router is responsible for classifying arriving packets to the appropriate class and connect the DS domain to other DS or non-DS capable domains. Within the core routers, packets are forwarded according to the PHB associated with each DS code. There are several packet scheduling algorithms such as Priority Queuing (PQ), Weight Fair Queuing (WFQ), and Class Based Queuing (CBQ). PQ is a basic scheme that simply allows designed high priority traffic first access to available bandwidth; it provides no means of controlling the allocation of bandwidth. Also, PQ often results in all but the highest priority applications being completely starved of bandwidth. WFQ

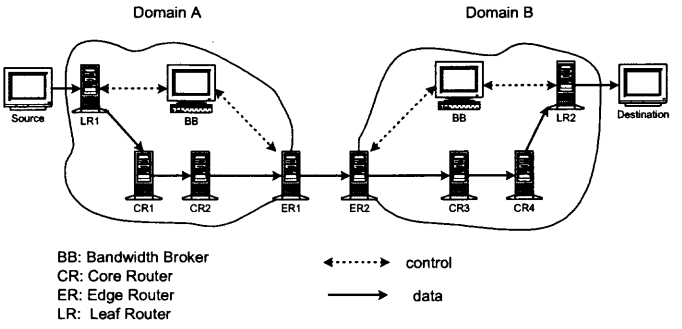

Figure 5.13 The DiffServ architecture.

overcomes this limitation by providing for each of a small number of traffic classes a fixed proportion of bandwidth. Its drawbacks are that classification is complex and limited, and explicit rate control for traffic classes is not ensured. CBQ was developed as a progression of earlier efforts such as WFQ to give IP a more flexible set of service parameters. So, in this thesis, CBQ is used to manage AF1 class with different drop precedence. We propose to use AF1 class to transport the layered video traffic. A Differentiated Service node must allocate a configurable, minimum amount of forwarding resources to each independent AF class.

The spatial scalable encoded video traces are used here [1]. Every encoded frame has a base layer and an enhancement layer. If the base layer is lost, the respective enhancement is of no use. Our IP DiffServ MPEG spatial layered video packet marking algorithm can be summarized as follows:

- If the arriving packet belongs to the base layer, DSCP=AF1 low drop precedence.
- If the arriving packet belongs to the enhancement layer, DSCP=AF1 medium drop precedence.

[1]The spatial scalable encoded video traces used in this chapter were the courtesy of Martin Reisslein *et. al* of Arizona State University.

- If the arriving packet belongs to the FTP traffic, DSCP=AF1 high drop precedence.

MPEG employs both intraframe and interframe coding techniques for compression. To decode a B frame both the previous and future I and P frames are needed; to decode a P frame, the previous P or I frames is needed [48]. For the GOP pattern of $IB_1B_2P_1B_3B_4P_2B_5B_6P_3B_7B_8$, the discard of different frames of the base layer has the following impact:

- The discard of an I frame results in the discard of 14 frames and related enhancement frames.
- The discard of P_1 frame results in the discard of 11 frames and related enhancement frames.
- The discard of P_2 frame results in the discard of 8 frames and related enhancement layers.
- The discard of P_3 frame results in the discard of 5 frames and related enhancement layers.
- B frame discard results in no additional frame discard.

This dropping policy can be deployed at edge router ER1 to reduce the traffic load of the domain B network. In our scheme, if the base layer packets dropped at core routers belong to I frames, the related 14 base layer frames, and respective enhancement layer frames are dropped at ER1, because these related frames are of no use without the reference frame I. For the loss of different P frames, the related frames are dropped at edge router ER1 based on the above dropping policy. If the packets dropped at core routers belong to the enhancement layer, no additional frames are dropped at ER1. Each frame of the enhancement layer is used only to improve the quality of the same frame in the base layer; it has no impact on other frames.

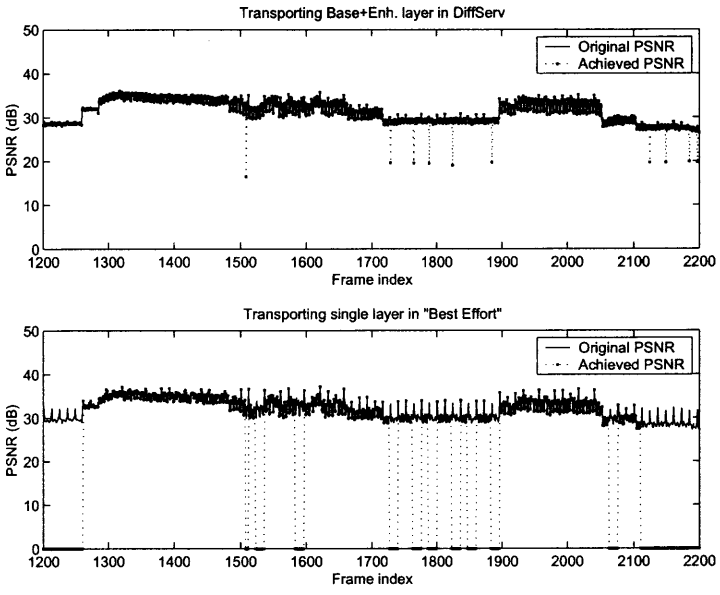

Figure 5.14 Comparison of PSNR through different networks for "Silence of the Lambs" with medium quality (traffic load 0.7).

5.3.2 Simulation Results and Performance Evaluation

In our scheme, the base layer is transmitted by AF1 class with low drop precedence, and the enhancement layer is transmitted by AF1 class with medium drop precedence. Here, we treat different drop precedence as allocating different bandwidth. We define one frame duration as a time slot ($33ms$ for NTSC, $40ms$ for PAL). At every time slot, the base layer is transmitted first; if there is leftover bandwidth, transmit the enhancement layer and the FTP traffic. The buffer size for the base and enhancement layers is 4000 bits, respectively. The AF class may also be configured to receive more forwarding resources than the minimum allocated when excess resources are available either from other AF classes or from other PHB groups [44]. The minimum bandwidth allocated to AF1 is $2Mbps$; the excess bandwidth derived from other classes depends on the traffic load with the maximum of $2Mbps$.

The end video quality is evaluated by PSNR defined as follows:

$$PSNR = 10 \cdot log_{10} \frac{p^2}{MSE}, \tag{5.3}$$

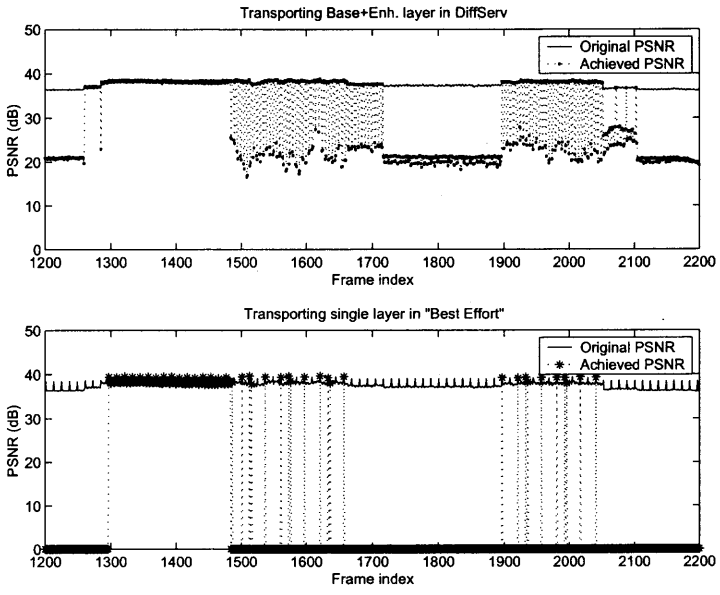

Figure 5.15 Comparison of PSNR through different networks for "Silence of the Lambs" with high quality (traffic load 0.7).

where p denotes the maximum luminance value of a pixel (255 in $8-bit$ pictures). Human visual system is more sensitive to luminance information than chrominance; here, we only consider PSNR of luminance. The Mean Square Error (MSE) for an individual video frame n is defined as:

$$M_n = \frac{1}{D_x \cdot D_y} \sum_{x=1}^{D_x} \sum_{x=1}^{D_y} [I(n,x,y) - \bar{I}(n,x,y)]^2. \tag{5.4}$$

$I(n,x,y), n = 0,1,\ldots,N-1, x = 1,\ldots,D_x, y = 1,\ldots,D_y$, denotes the luminance value of the pixel at (x,y) of the nth frame in the video sequence with N frames. The mean MSE for a video sequence of N video frames is

$$\bar{M} = \frac{1}{N} \sum_{n=0}^{N-1} M_n. \tag{5.5}$$

In our scheme, we consider spatial scalable encoded videos in CIF (Common Intermediate Format 352 × 288 pixels for each frame). Decoding only the base layer produces the video in QCIF (Quarter CIF 176 × 144 pixels for each frame), while

decoding both layers restores the video in the CIF format [46]. Note that the base layer QCIF format may be up-sampled and displayed in the CIF format: this up-sampling results in coarse-grained, low quality CIF format video. Fig. 5.14 shows the comparison of the original and achieved PSNR through different networks at traffic load 0.7 for the video trace "Silence of the Lambs" with medium quality. Here, the medium quality means that the quantization parameters set for I, P, and B frames are 10, 14, and 16, respectively; for high quality, 4, 4, and 4, respectively [46]. We assume domains A and B are both DS capable $10Mbps$ Ethernet link. In DiffServ, the spatial layered video is transmitted. In the "best effort"scenario, $2Mbps$ is allocated to the real time traffic. Medium quality single layer video is transmitted in the "best effort" scenario. From Fig. 5.14, we note that the original PSNR of each frame can be achieved most of the time in DiffServ; even at congestion, we still can keep the QoS of the base layer, and the dropping of enhancement layer frames has no impact on other frames. Fig. 5.15 shows the PSNR comparison in transmission of the high quality video trace "Silence of the Lambs" with the same network condition as for the medium quality video trace. The average original PSNR for this trace is 37.8 dB; by transmitting through different networks, the achieved average PSNR is 35.8 dB for our proposed scheme, while the average achieved PSNR is only 22.6 dB in the "best effort" service. Fig. 5.16 shows the achieved average PSNR of the video trace "Silence of the Lambs" with high quality level transported through DiffServ and the "best effort" service at different traffic loads. Note that the traffic load has less impact on the achieved average PSNR in DiffServ than in the "best effort" service because we may drop the bandwidth carrying the enhancement layer and keep the base layer in DiffServ during congestion. The QoS cannot be guaranteed in the network that provides the "best effort" service, because the QoS depends on the current traffic load.

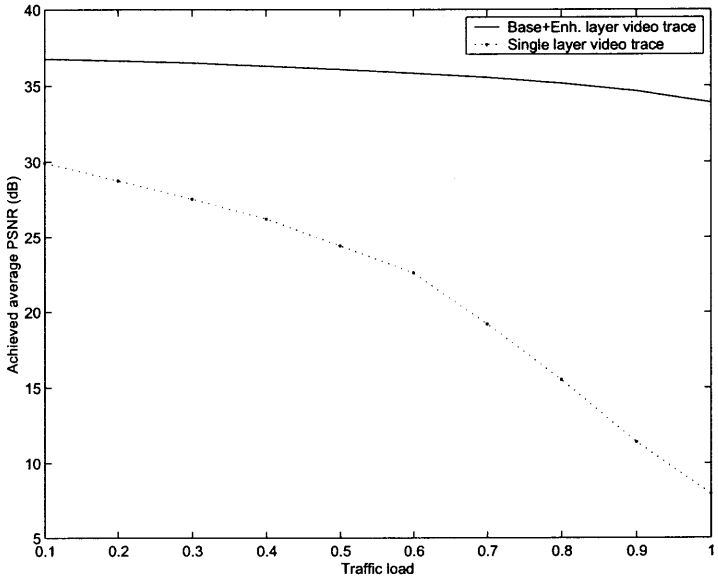

Figure 5.16 Achieved average PSNR at different traffic loads for the video trace "Silence of the Lambs".

5.4 Summary

DiffServ provisions QoS in the IP network and provides a less complex and more scalable method as compared with IntServ. If we transmit all MPEG frames over EF, although QoS can be guaranteed, the bandwidth utilization becomes very low. Our proposed scheme by separating I frames from P and B frames and transporting them over different PHBs not only guarantees QoS but also achieves a high bandwidth utilization.

By taking into consideration the end video quality from the network and application levels, we have further proposed transmitting spatial scalable encoded videos through DiffServ using the AF class. Simulation results show that our proposed method can achieve the original PSNR most of the time. Even during congestion, we can still maintain the basic QoS of the video trace by keeping the base layer. Traffic load has less impact on our end quality because we give high priority to the base layer, thus maintaining the basic quality. Since there may be several domains between the

source and destination, our proposed dropping policy at the egress edge router can reduce the traffic load which would otherwise traverse through other DS or non DS capable domains along the path, thus increasing the bandwidth utilization in other domains.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

VBR traffic is one of the major services to be supported by broadband packet switched networks. Transporting VBR video streams while guaranteeing a required level of QoS is a challenging problem due to the bursty nature of the MPEG video traffic. In this dissertation, we first propose to predict the relative size change of I frames using LMS algorithm, i.e., s_k instead of the original sequence I_k. Owing to a smoother sequence s_k compared with the original one and the composite sequence, better prediction has been achieved. One problem associated with LMS algorithm is its slow convergence. In VBR video traffic characterized by the frequent scene changes, the LMS algorithm may result in an extended period of intractability, and thus experience excessive cell loss during scene changes. A variable step size is thus proposed to overcome this drawback, first using KVSA, and later VSA. Our simulations show that VSA not only incurs small prediction errors but more importantly also achieves fast convergence. This new algorithm converges faster, and hence tracks scene changes better.

The prediction, when combined with dynamic bandwidth allocation, can provision both network efficiency and QoS guarantees. There is a trade-off between the network utilization and the overhead for bandwidth negotiation. In this dissertation, a dynamic bandwidth allocation based on the predicted I frame has been proposed. Since only I frame need to be checked, this method has greatly reduced the renegotiation frequency with a small CLR. The spectral density of the prediction errors has been justified analytically to be rather uncorrelated, and thus using small buffer space. To further improve the bandwidth utilization, QDBA bandwidth allocation scheme has been proposed to allocate bandwidth for video traffic. This scheme is based on buffer monitoring and the predicted GOP using VSA;

the goal of this scheme is to meet the stringent QoS requirements with an acceptable utilization. Simulations and analytical results demonstrate that this scheme does provide the guaranteed delay, and thus can be used for on-line real time video delivery.

Network traffic appears to be self similar, and self similar traffic has great impact on network performance. Thus, network traffic prediction plays an important role in network management. LMK, which uses the negated kurtosis of the error signal as the cost function, is proposed to predict the self similar traffic. Simulation results show that LMK incurs much smaller prediction error as compared with LMS. Since the prediction performance can be improved greatly with only a small extra computation, LMK can be used to effectively predict the real time network traffic.

DiffServ provisions QoS in the IP network and provides a less complex and scalable method as compared to IntServ. If we transmit all MPEG frames over EF, although QoS can be guaranteed, the bandwidth utilization becomes very low. Our proposed scheme in transporting different frames through different service classes of DiffServ not only guarantees QoS but also achieves a high bandwidth utilization. By taking into consideration the end video quality from the network and application levels, we have further proposed transmitting spatial scalable encoded videos through DiffServ using the AF1 class. Simulation results show that our proposed method can achieve the original PSNR most of the time. Even during congestion, we can still maintain the basic QoS of the video trace by keeping the base layer. Traffic load has less impact on our end video quality because we give high priority to the base layer, thus maintaining the basic quality. Our proposed dropping policy at the egress router can reduce the traffic load which would otherwise traverse through other DS or non DS capable domains along the path, thus increasing the bandwidth utilization in other domains.

As the end video quality depends on both application level QoS control and network support, a promising future research direction is to combine these two aspects

and design an efficient control system to meet the stringent QoS requirements of multimedia streaming with acceptable network resource utilization. The original design of IP network fails to effectively support large scale content delivery like streaming media multicast. While "IP multicast" is an extension to provide multi-point packet delivery, there are still many issues in deploying IP multicast such as scalability, network management, and deployment. Application level multicast, aiming at building a multicast service on top of the Internet, is a promising research direction.

REFERENCES

[1] ISO/IEC, "Coding of moving pictures and associated audio for digital storage media at up to 1.5 mbit/s," *ISO/IEC 11172*, 1993.

[2] R. Braden, D. Clark, and S. Shenker, "Integrated service in the Internet architecture: an overview," *RFC 1633*, 1994.

[3] ISO/IEC, "Generic coding of moving pictures and associated audio: Video," *ISO/IEC Standard 13818-2*, 1995.

[4] J. Y. B. Lee, "On a unified architecture for video-on-demand services," *IEEE Trans. Multimedia*, vol. 4, pp. 38–47, Mar. 2002.

[5] C. C. Y. Choi and M. Hamdi, "A scalable video-on-demand system using multi-batch buffering techniques techniques," *IEEE Trans. Broadcast.*, vol. 49, pp. 178–191, Mar. 2003.

[6] A. D. Gelman, H. Kobrinski, L. S. Smoot, and S. B. Weinstein, "A store-and-forward architecture for video-on-demand service," *ICC'91*, pp. 842–846, 1991.

[7] J. M. McManus and K. W. Ross, "Video-on-demand over ATM: constant rate transmission and transport," *IEEE J. Select. Areas Commun.*, pp. 1087–1098, 1996.

[8] N. Ansari, H. Liu, Y. Q. Shi, and H. Zhao, "Dynamic bandwidth allocation for VBR video transmission," *Journal of Computing and Information Technology*, vol. 4, pp. 309–317, Nov. 2003.

[9] Z. Fu, X. Meng, and S. Lu, "A transport protocol for supporting multimedia streaming in mobile Ad Hoc networks," *IEEE J. Select. Areas*, vol. 21, pp. 1615–1626, Dec. 2003.

[10] Q. Zhang, W. Zhu, and Y. Q. Zhang, "Resource allocation for multimedia streaming over the Internet," *IEEE Trans. Multimedia*, vol. 3, pp. 339–355, Sep. 2001.

[11] E. W. Fulp and D. S. Reeves, "On-line dynamic bandwidth allocation," *Proc. Intl. Conf. on Networking Protocols*, pp. 134–141, 1997.

[12] E. W. Knightly and H. Zhang, "D-BIND: An accurate traffic model for providing QoS guarantees to VBR traffic," *IEEE/ACM Trans. Networking*, vol. 5, pp. 219–231, Apr. 1997.

[13] S. Chong, S. Q. Li, and J. Ghosh, "Predictive dynamic bandwidth allocation for efficient transport of real-time VBR video over ATM," *IEEE J. Select. Areas Commun.*, vol. 13, pp. 12–23, Jan. 1995.

[14] A. M. Adas, "Using adaptive linear prediction to support real-time video under RCBR network service model," *IEEE/ACM Trans. Networking*, vol. 6, pp. 635–644, Oct. 1998.

[15] X. Wang, S. Jung, and J. S. Meditch, "Dynamic bandwidth allocation for VBR traffic using adaptive wavelet prediction," *ICC'98*, vol. 1, pp. 549–553, 1998.

[16] H. Zhang and E. W. Knightly, "RED: A new approach to support delay sensitive VBR video in packet switched networks," *Proc. 5th Workshop on Networking and Operating System Support for Digital Video*, pp. 275–286, Apr. 1995.

[17] W. Xu and A. Qureshi, "Adaptive linear prediction of MPEG video traffic," *Fifth Intl. Symp. on Signal Processing and its Application*, pp. 67–70, Aug 1999.

[18] M. Hayes, *Statistical Digital Signal Processing and Modeling.* New York: Wiley, 1991.

[19] R. H. Kwong and E. W. Johnston, "A variable step size LMS algorithm," *IEEE Trans. Signal Processing*, vol. 40, pp. 1633–1642, Mar. 1993.

[20] N. Ansari, H. Liu, Y. Q. Shi, and H. Zhao, "On MPEG modeling," *IEEE/ACM Trans. Broadcast.*, vol. 48, pp. 337–347, Dec. 2002.

[21] Y. Afek, M. Cohen, E. Haalman, and Y. Mansour, "Dynamic bandwidth allocation policies," *Proc. IEEE INFOCOM'97*, pp. 1096–1104, 1997.

[22] S. K. Biswas and R. Izmailov, "Design of a fair bandwidth allocation policy for VBR traffic in ATM networks," *IEEE/ACM Trans. Networking*, vol. 8, pp. 212–223, Apr. 2000.

[23] S. Rampal, D. Reeves, Y. Viniotis, and D. Argrawal, "Dynamic resource allocation based on measured QoS," *Proc. of the Fifth ICCCN-Intl. Conf. on Computer Commn.*, pp. 24–27, 1996.

[24] M. Grossglauser, S. Keshav, and D. Tse, "RCBR: A simple and efficient service for multiple time scale traffic," *ACM SIGCOMM*, pp. 219–230, 1995.

[25] S. Q. Li and C. L. Hwang, "Queuing response to input correlation functions: discrete spectral analysis," *IEEE/ACM Trans. Networking*, vol. 1, pp. 522–533, Oct. 1993.

[26] S. Q. Li, "A general solution technique for discrete queuing analysis of multimedia traffic on ATM," *IEEE/ACM Trans. Commun.*, vol. 39, pp. 1115–1132, Jul. 1991.

[27] S. Q. Li and H. D. Sheng, "Discrete queuing analysis of multimedia traffic with diversity of correlation and burstiness properties," *INFOCOM'91.*, vol. 1, pp. 368–381, 1991.

[28] H. Zhao, N. Ansari, and Y. Q. Shi, "A fast non-linear adaptive algorithm for video traffic prediction," *Proc. IEEE Intl. Conf. on Information Technology, Coding and Computing*, pp. 54–58, Apr. 2002.

[29] G. Karlsson, "Asynchronous transfer of video," *IEEE Commun. Mag.*, vol. 24, pp. 118–126, Aug. 1996.

[30] M. Krunz, "Bandwidth allocation strategies for transporting variable bit rate video traffic," *IEEE Commun. Mag.*, vol. 37, pp. 40–46, Jan. 1999.

[31] A. Leon-Garcia, *Probability and Random Processes for Electrical Engineering.* Addison Wesley, 1994.

[32] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, "On the self similar nature of Ethernet traffic (extended version)," *IEEE/ACM Trans. Networking*, vol. 2, pp. 1–15, Feb. 1994.

[33] V. S. Frost and B. Melamed, "Traffic modeling for telecommunications networks," *IEEE Commun. Mag.*, vol. 32, pp. 70–81, Mar. 1994.

[34] J. R. M. Hosking, "Modeling persistence in hydrological time series using fractional differencing," *Water Resources Research*, vol. 20, pp. 1898–1908, 1984.

[35] P. Kokoszka and M. Taqqu, "Parameter estimation for infinite variance fractional ARIMA," *The Annals of Statistics*, vol. 24, pp. 1880–1993, 1996.

[36] O. Tanrikulu and A. G. Constantinides, "Least-mean kurtosis: A novel high-order statistics based adaptive filtering algorithm," *IEE Electronics Letters*, vol. 30, pp. 189–190, Feb. 1994.

[37] S. Haykin, *Adaptive Filter Theory.* Prentice Hall, 1991.

[38] S. A. M. Östring and H. Sirisena, "The influence of long range dependence on traffic prediction," *Proc. ICC2001*, vol. 4, pp. 1000–1005, 2001.

[39] D. D. Clark and W. Fang, "Explicit allocation of best effort packet delivery service," *IEEE/ACM Trans. Networking*, vol. 6, pp. 362–373, Aug. 1998.

[40] S. Blake, D. Black, E. Davis, Z. Wang, and W. Weiss, "RFC 2475: An architecture for different services," 1998.

[41] V. Jacobson, K. Nichols, and K. Poduri, "RFC 2598: An expedited forwarding PHB," June 1999.

[42] S. Floyd and V. Jacobson, "Link-sharing and resource management models for packet networks," *IEEE/ACM Trans. on Networking*, vol. 3, pp. 365–386, Aug. 1995.

[43] H. Zhao, N. Ansari, and Y. Q. Shi, "Transmission of real-time video over IP differentiated services," *IEE Electronics Letters*, vol. 38, pp. 1151–1153, Sep. 2002.

[44] J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski, "RFC 2597: Assured forwarding PHB group," June 1999.

[45] T. V. Lakshman, A. Ortega, and A. R. Reibman, "VBR video: trade-offs and potentials," *Proceedings of the IEEE*, vol. 86, pp. 952–978, 1998.

[46] M. Reisslein, J. Lassetter, O. L. S. Ratnam, F. H. Fitzek, and S. Panchanathan, "Traffic and quality characterization of scalable encoded video: a large-scale trace-based study part 1: overview and definitions," *http://www.eas.asu.edu/trace*, 2002.

[47] S. Bakiras and V. O. K. Li, "Efficient resource management for end-to-end QoS guarantees in Diffserv networks," *ICC2002*, vol. 2, pp. 1220–1224, 2002.

[48] M. Furini and D. Towsley, "Real time traffic transmission over the Internet," *IEEE Trans. Multimedia*, vol. 3, pp. 33–40, Mar. 2001.