ABSTRACT

VIDEO TRAFFIC MODELING AND DELIVERY

by
Hai Liu

Video is becoming a major component of the network traffic, and thus there has been a great interest to model video traffic. It is known that video traffic possesses short range dependence (SRD) and long range dependence (LRD) properties, which can drastically affect network performance. By decomposing a video sequence into three parts, according to its motion activity, Markov-modulated self-similar process model is first proposed to capture autocorrelation function (ACF) characteristics of MPEG video traffic. Furthermore, generalized Beta distribution is proposed to model the probability density functions (PDFs) of MPEG video traffic.

It is observed that the ACF of MPEG video traffic fluctuates around three envelopes, reflecting the fact that different coding methods reduce the data dependency by different amount. This observation has led to a more accurate model, structurally modulated self-similar process model, which captures the ACF of the traffic, both SRD and LRD, by exploiting the MPEG structure. This model is subsequently simplified by simply modulating three self-similar processes, resulting in a much simpler model having the same accuracy as the structurally modulated self-similar process model.

To justify the validity of the proposed models for video transmission, the cell loss ratios (CLRs) of a server with a limited buffer size driven by the empirical trace are compared to those driven by the proposed models. The differences are within one order, which are hardly achievable by other models, even for the case of JPEG video traffic.

In the second part of this dissertation, two dynamic bandwidth allocation algorithms are proposed for pre-recorded and real-time video delivery, respectively. One is based on scene change identification, and the other is based on frame differences. The proposed algorithms can increase the bandwidth utilization by a factor of two to five, as compared to the constant bit rate (CBR) service using peak rate assignment.

VIDEO TRAFFIC MODELING AND DELIVERY

by
Hai Liu

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 2000

APPROVAL PAGE

Video Traffic modeling and Delivery

Hai Liu

Dr. Y. Q. Shi, Dissertation Advisor — Date
Associate Professor, Department of Electrical and Computer Engineering, NJIT

Dr. N. Ansari, Dissertation Co-Advisor — Date
Professor, Department of Electrical and Computer Engineering, NJIT

Dr. R. Malik, Committee Member — Date
Associate Professor, Department of Electrical and Computer Engineering, NJIT

Dr. H. F. Sun, Committee Member — Date
Deputy Director, Mitsubishi Electronics Information Technology Center

Dr.N. Uzun, Committee Member — Date
Assistant Professor, Department of Electrical and Computer Engineer, NJIT

BIOGRAPHICAL SKETCH

Author: Hai Liu

Degree: Doctor of Philosophy in Electrical Engineering

Date: May 2000

Undergraduate and Graduate Education:

- Doctor of Philosophy in Electrical Engineering, May 2000
  New Jersey Institute of Technology, Newark, NJ, U.S.A.
- Master of Science in Electrical Engineering, January 1988
  Northwest Telecommunication Engineering Institute , Xi'an, China.
- Bachelor of Science in Electrical Engineering, June 1985
  Taiyuan University of Technology, Taiyuan, China.

Major: Electrical Engineering

Publications and Presentations:

H Liu, N. Ansari, and Y. Q. Shi, "An Accurate Analytical Model for MPEG Coded Video Traffic," IEE Electronic Letter, Aug. 1999, vol. 35, No. 17, pp. 1449-1450.

H. Liu, N. Ansari, and Y. Q. Shi, "Modeling MPEG Coded Traffic Using Sequentially Modulated Self-Similar Process," Broadband Communications, Ed. by Danny H. K. Tsang and P. J. Kuhn, Kluwer Academic Publishers, pp. 63–72, 1999.

H. Liu, N. Ansari, and Y. Q. Shi, "Modeling MPEG Coded Traffic Using Markov-Modulated Self-Similar Process," Proceedings of IEEE MMSP'99, Copenhagen, Denmark, Sep. 1999, pp.363–368.

H. Liu, N. Ansari, and Y. Q. Shi, "Markov-Modulated Self-Similar Process: Video Traffic Modeler and Synthesizer," Proceedings of IEEE GLOBECOM'99: High Speed Networks, Rio de Janeiro, Brazil, December 5-9, 1999, pp.1184-1188.

H. Liu, N. Ansari, and Y. Q. Shi, "Dynamic Bandwidth Allocation Based on Scene Change Identification," Proceedings of IEEE International Conference on Information Technology: Coding and Computing, March 27-29, 2000, Las Vegas, Nevada, pp.284-288.

H. Liu, N. Ansari, and Y. Q. Shi, "A Simple Model for MPEG Video Traffic," to be presented at ICME'2000.

H. Liu, N. Ansari, and Y. Q. Shi, "Dynamic Bandwidth Allocation for Real-Time video Transmission," submitted to GLOBECOM'2000.

H. Liu, N. Ansari, and Y. Q. Shi,"Modeling MPEG Coded Traffic by Markov-Modulated Self-Similar Processes," to appear in a special issue in VLSI Signal Processing.

H. Liu, N. Ansari, and Y. Q. Processing, "Scene Change Driven Dynamic Bandwidth Allocation for VBR Video Delivery, " in preparation for journal submission.

ACKNOWLEDGMENT

I would like to take this opportunity to thank my advisor, Dr. Yun Qing Shi. His sharp insight into problems enlightens my research direction. He has fostered within me a love of research and the courage of overcoming difficulties. Many detailed discussions helped me find the right ways to solve problems. I am a much better researcher because of his excellent guidance.

My thanks also go to my co-advisor, Dr. Nirwan Ansari, for his help on my research. Discussions with him always gave me wonderful hints to problems. His excellent lectures gave me the fundamental knowledge on ATM, and provided me the knowledge to do my research work. I am also very grateful to him for his non-academic help that made the research possible.

I am also thankful to my committee members, Dr. Huifang Sun, Dr. Raashid Malik and Dr. Necdet Uzun for their meticulous reading of my dissertation and probing questions, especially to Dr. Malik for his discussion over alternative ways to generate data with different auto-correlations.

I have had the pleasure to be together with some talented people in the Center for Communication and Signal Processing Research at NJIT. It is really a great pleasure to study and get along with them. I am grateful to them for all kind of help, academic and non-academic. It is them who made the daily life in the lab more colorful. Thanks are directed to all friends in the Lab.

My life would not be so happy and colorful without my wife's love and support. Her understanding makes the life so easy. Without her support, it is impossible to finish the research. I truly appreciate Dongqing, my wife, from the bottom of my heart.

TABLE OF CONTENTS

Chapter **Page**

Chapter **Page**

LIST OF TABLES

Table **Page**

LIST OF FIGURES

Figure **Page**

Figure **Page**

Figure **Page**

CHAPTER 1

INTRODUCTION

1.1 New Services

Have we ever thought of holding meetings without leaving home? Have we ever imagined to go shopping at home? We are coming to an age of visual communications. New kinds of information services are emerging. Information service providers promise to provide us the following new services:

- Movies on demand (MOD)
- News on demand
- Near video on demand (NVOD)
- Video on World Wide Web
- Personal video conferencing
- Desktop video-telephony

They even promise to provide more advanced video services listed below:

- Broadband video-telephony
- Broadband video conferencing
- Video-surveillance
- Video mail
- Video retrieval
- Existing quality TV distribution
- High definition distribution

- Pay-TV

These services require transmission of a tremendous amount of data. For example, the delivery of NTSC video requires about 270Mbps. For HDTV, the required bandwidth is even larger. Such requirement of large bandwidth is very costly. Customers are willing to pay for these services only if the quality of service (QoS) can be guaranteed at a reasonable price. The key to successful deployment of these services is the provision of QoS at a reasonable price

1.2 Video Compression

As pointed out early, video traffic contains a huge amount of data. Large bandwidths are needed to deliver these data. To be able to deliver video traffic at a reasonable price, video data must be compressed without significant quality degradation. Since there is a lot of redundant information among video data, video compression techniques can be readily used to reduce bandwidth requirements. Many well known video compression techniques have been proposed:

- DCT coding
- Sub-band coding
- Fractal
- Run length coding
- Wavelet
- Motion estimation

These techniques can substantially reduce the volume of video data.

1.3 Standards and MPEG

In order for video information to be accessed across a wide range of services and equipments, international standards for video coding, decoding, and frame structure have to be specified. MPEG-2 and H.261 are among the most famous standards, while H.262 is aimed at providing different quality of service for different users on ATM networks.

Various MPEG versions were developed by Motion Picture Expert Group, formerly known as ISO/IEC JTC1/SC29/WG11. The first version (MPEG-1) is for video storage. MPEG-2 has been finalized for higher resolution images and correspondingly higher data rate. Currently the object-based MPEG-4 is being finalized for better image quality and more functionalities.

The MPEG standards outline compression technologies, define the bit stream syntax that makes the video stream transmission possible, describe the multiplexing of video and audio, and stipulate the position of synchronization information. It basically consists of the following parts:

- Systems
- Video
- Audio
- Compliance
- Software simulation
- Digital storage media-command and control (DSM-CC)
- Real-time interface for system decoders
- DSM Reference Script Format

The video part defines the basic objects, video stream structure, decoding process, scalability extensions, profiles and levels. The DCT and motion estimation techniques are used. Scalabity, profiles and levels are used to provide different QoS to different services. The image quality is adjusted by receiving equipments according to their decoding ability.

1.4 Network Technologies

In order to provide these services, the network must be flexible and fast enough to meet all requirements imposed by these new services. B-ISDN networks promise to provide such flexibility that allows transmission of voice, data, and video service over a single network via a single high speed user connection.

To deliver these data to end users, high speed transmission media from local switches to end users' home are required. There are several options for these connections. For example, the existing cable infrastructure can be used to deliver these services because cable TV is deployed almost everywhere. In the area where cable TV is not available, xDSL (such as ADSL-Asymmetric Digital Subscriber Line) can exploit the existing telephone network to deliver high bit rate traffic.

As the cost of fiber optics decreases, the fiber may be installed up to some central points, from which the existing coax cable networks can be connected. Examples of such kind of connections are FTTC (Fiber to the Curb), FTTB (Fiber to the Building), and HFC (Hybrid Fiber-Coax). This kind of connection provides reasonable performance to cost ratio. The highest performance connections in terms of transmission speed are FTTH (Fiber to the Home).

ATM is the backbone of B-ISDN. It uses cell switching technology to transmit traffic. The cell has 48 byte payload and 5 byte header. Compared with packet switching technology, control information contained in the cells is very small. This is possible because fiber optics can provide low error rate transmission, the end

terminals have powerful processing ability, and the connection oriented technology is used. Thus, these switches can operate at high speed. Since the cell size is small, the switching speed and transmission rate can be sufficiently high that jitter is small enough to transmit time sensitive traffic such as voice and video.

ATM networks consist of physical, ATM, ATM adaption (AAL), and higher layers. The physical layer is composed of two sub-layers, the transmission convergence (TC) and physical media dependent part (PMD), which serves to adopt different transmission media. ATM layers are used to control traffic flow, to generate and extract cell header, to translate cell's VPI/VCI , and to multiplex and demultiplex traffic originated from different sources. AAL is also composed of two sublayers, convergence sublayer and the segmentation and reassemble sublayer. AALs are used to adapt different services for ATM transmission. The layers provide flexibility to ATM network, and facilitate the transmission of many kinds of services over a single network. Four types of services are defined, and correspondingly, the following four types of AALs are defined:

- AAL-1, is used to emulate circuit, using constant bit rate transmission to serve class A services
- AAL-2, using variable bit rate transmission, used to serve class B services
- AAL-3/4, using variable bit rate transmission, connection oriented, is used to serve class C services
- AAL-5, connectionless data transfer, is used to serve class D services

AAL-1, AAL-2 and AAL-5 can be used to transmit video streams.

Constant bit rate (CBR) transmission is easier to implement from the network management viewpoint, but at the cost of variable quality of service, and/or at the cost of network efficiency. Compressed video is variable bit rate (VBR) in nature,

and hence, it is efficient to transmit compressed video stream by VBR service with guaranteed quality of service.

1.5 Traffic Parameters and Quality of Service Parameters

In order to use VBR service, customers must provide parameters of their traffic to the service providers. There has not been a good way so far yet to specify the video traffic. One way to specify the traffic is to use leaky bucket parameters, such as burst length, sustained bit rate, and peak bit rate. Another way is to use traffic models to describe traffic. Traffic parameters are important to the service providers to allocate resources and calculate charges. Customers must make sure that the traffic parameters are satisfied so that QoS can be guaranteed.

To guarantee quality of service, QoS parameters must also be specified before connections are established. For video traffic, the following parameters are usually specified:

- Cell delay variation tolerance
- Maximum cell delay allowed
- Cell loss ratio (CLR)
- Cell error ratio (CER)
- Severely errored cell block ratio (SECBR)

These parameters are essential and manageable for service providers, but it is difficult to map the actual quality requirements of customers to these parameters. There has not been a good way to mapping the quality requirements to those parameters yet because the relationship between QoS and these parameters has not been understood thoroughly. For example, the same CLR may result in quite different QoS for the same video stream. It is still an open research issue.

CHAPTER 2
TRAFFIC MODELS FOR VIDEO TRAFFIC

Traffic models are important to network engineers. They can be used to evaluate network performance, design admission control algorithms, dynamically allocate bandwidth, and specify customer traffic. Traffic models are employed in two fundamental ways: as part of an analytical model or to drive a network simulation program.

Traditionally, telephony network traffic has been modeled by Poisson processes. The arrival of telephony customers are essentially independent, and therefore, this kind of models work well. Moreover, Poisson processes are analytically tractable, and many results about network performance, such as blocking and delay probability, can be obtained.

Many years of study on local and wide area network traffic shows that Internet and ATM traffic are bursty and strongly dependent [1, 2, 3, 4, 5, 6, 7]. These characteristics have profound impacts on network performance because they affect the queuing behaviors of networks (see [8] and references in it). In general, the dependence will degrade the network performance [9] because it often results in buffer overflow. Empirical studies and statistical analysis on network traffic, gathered from high speed networks, provide evidence of prevalence of self-similar patterns [1, 6, 5]. Obviously, it is inappropriate to use traditional traffic models to describe ATM and Internet traffic [7] because tradtional models focus on a very limited range of time scale while the traffic exhibits correlations over a wide range of time scale.

Recent development of coding standards for digital video, such as H.261, H.262, H.263, MPEG-2, MPEG-4, has made it feasible to transmit video data over networks. Transmission of video data over network will be commonplace. Video traffic will be the dominating part of network traffic, and therefore, it is necessary to model video traffic. Due to the fact that the contents of a video stream is strongly dependent in nature, video traffic itself possesses burstiness and long range dependence [5], and

thus new methods to model network traffic, especially methods to model video traffic, are necessary.

There are two ways to transmit video over network. One way is to control the quantization steps of encoder so that the output of the encoder is CBR. The drawback of this method is that the quality of video is variable, especially in cases with many scene changes and high complex motions. The video quality can vary to a large extent.

An alternative to CBR is variable bit rate (VBR) transmission. VBR does not attempt to control the output bit rate of encoder, but produces a variable bit rate so that video quality can be constant. This makes the bandwidth allocation very difficult or makes the network utilization very low. To increase network utilization or statistical multiplexing gain (SMG), good models for VBR video are needed .

To model VBR video traffic accurately, autocorrelations among data should be taken into consideration. A considerable amount of effort on video modeling has been reported, which can be classified into statistical models and deterministic models. The statistical models include:

- Renewal process models
- Markov models
- Transform expand sample (TES) model
- Markov modulated process model
- Histogram -based model
- Auto-regressive models
- Fluid model
- $M/G/\infty$ input process model

- GBAR model
- Long range dependent model or self-similar model

which can be categorized into two classes:

- Short range dependent models (SRD),
- Long range dependence models (LRD).

These models are used to capture two statistical factors: marginal distribution (first-order statistics) and autocorrelation function (second-order statistics) of traffic data. LRD models can capture long range dependence, while SRD models can capture short range dependence. The impact of traffic dependence on queuing performance measures, such as queue length, waiting time, cell loss rate, can be very dramatic. It is the common belief that traffic dependence will degrade queuing performance.

To describe network traffic, the models should not only capture the first-order statistics, but also the second-order statistics. Almost all modern traffic models take into consideration the traffic dependence to some extent. The difference between SRD and LRD models are the extent to which the dependence is considered.

Because of the statistical nature of network traffic, deterministic models are not so popular as statistical models. A few deterministic models have been reported. Two models found in the literatures are D-BIND and (σ, ρ) models.

A traffic model is a key element in network analysis and simulation. Clear understanding of traffic characteristics is the most important step toward the success of network evaluation. For the sake of good understanding of traffic modeling, some of the existing traffic models will be presented in details in the following sections.

2.1 Renewal Process Models

Tradtional traffic can be described by a random point process. A random point process $\psi = \{t_n : n \geq 1\}$ is a sequence of random points t_n at which an event

occurred, where

$$0 < t_1 < t_2 < \cdots. \tag{2.1}$$

with $t_n \to \infty$ as $n \to \infty$.

The point process has two equivalent processes: counting process and inter-arrival process. The counting process $\{N(t) : t \geq 0\}$ for ψ denotes the number of points that fall in the interval $(0, t]$. Let $T_n = t_n - t_{n-1}$, then the process $\varphi = \{T_n : n > 0\}$ is the inter-arrival process. Furthermore, if φ is i.i.d, then this random point process is called a renewal process.

Since renewal processes are analytically tractable, they have been traditionally used as traffic models. Poisson Process is a special case of renewal process. Due to the fact that Poisson processes have some elegant properties, they have been used widely in telephone industry. A Poisson process is an independent increment process. Its inter-arrival process is exponentially distributed, thus making it memoryless, and greatly simplifying the analysis of a queuing system.

It is believed that traffic burstiness can be explained to a large extent by two factors: the shape of marginal distribution and autocorrelation [9]. Strong positive autocorrelations are strong resources of burstiness. The autocorrelation of renewal processes, however, vanishes for all non zero lags, and therefore they fail to model modern network traffic.

Phase-type renewal process is an important special renewal process [9]. Its associated arrival process can be modeled as the time spent for a continuous Markov process $\{C(t) : 0 < t < \infty\}$, whose state space is $\{0, 1, \cdots, m\}$ and has an absorbing state, to go to absorption. To get a sample of the process A_n, start the Markov process C with the initial distribution π, the elapsed time is the value of the sample A_n. All the samples are obtained from the same initial distribution π. This kind of model is analytically tractable.

2.2 Transform Expanded Sample Models

Transform expanded sample techniques are a kind of non-linear regression, which can be used to capture the marginal distribution and the ACF of an empirical traffic data simultaneously [10, 11, 12, 13, 14]. The TES model was used to model various types of ATM traffic, including H.261 coded video conference traffic, JPEG coded video, and MPEG coded video. A TES random process is generated from the so called background process by two transformations. Let $\{U_n : n = 0, 1, \ldots\}$ be the background random process with F_B being the distribution, then the model is generated by the following formula:

$$X_n = F^{-1}(F_B(U_n)) \tag{2.2}$$

where F is the desired distribution function. F^{-1} is the inverse of F. U_n defines a random walk on the unit circle using modulo-1 (fractional part) arithmetic, defined as $x = 1 - \lceil x \rceil$, $\lceil x \rceil$ is the largest integer less or equal to x. The desired distribution is often expressed in the form of a histogram. If the distribution of U_n is uniform, the formula to generate the model data becomes:

$$X_n = F^{-1}(U_n) \tag{2.3}$$

There are two different TES models: TES$^+$ and TES$^-$, differed by the background process adopted. The background process used in generating TES$^+$ models is given by

$$U_n^+ = \begin{cases} U_0, & \text{if } n = 0 \\ < U_{n-1}^+ + V_n >, & \text{if } n > 0 \end{cases} \tag{2.4}$$

where $< x >$ is the fractional part of x. U_0 is uniformly distributed on $[0, 1)$ and $V_n : n = 1, 2, \ldots$, is $i.i.d$ with marginal distribution F_V and is called the innovation function. The background process used in generating TES$^-$ model is given by:

$$U_n^- = \begin{cases} U_n^+, & \text{if } n \text{ is even} \\ 1 - U_n^+, & \text{if } n \text{ is odd} \end{cases} \tag{2.5}$$

These background processes are Markovian with uniform distribution. V_n used here should be independent of U_0.

In general, $\{V_n\}$ is obtained from distribution F_v, which is typically restricted to step functions in order to simplify the parameter search. The most simple method to get $\{V_n\}$ is given by:

$$V_n = L + (R - L)Z_n \qquad -0.5 \leq L < R < 0.5 \tag{2.6}$$

where Z_n are $i.i.d$ and uniformly distributed on $[0, 1)$. L and R can be replaced by α and ϕ as follows:

$$\alpha = R - L \tag{2.7}$$

and

$$\phi = \frac{R + L}{R - L}, \tag{2.8}$$

which are more convenient for use because α controls the magnitude of the autocorrelation function and ϕ controls the oscillations.

A "smoothing" operation or stitching transformation may be applied to U_n before the inverse transformation F^{-1} is applied, so that the traffic model seems more "Homogeneous." The stitching function has the following form:

$$S_\xi(U_n) = \begin{cases} U_n/\xi, & 0 \leq U_n \leq \xi \\ (1 - U_n)/(1 - \xi), & \xi \leq U_n < 1. \end{cases} \tag{2.9}$$

and thus, the formula for TES model is given by:

$$X_n = F^{-1}(S_\xi(U_n)). \tag{2.10}$$

All the TES background processes are Markovian and uniformly distributed on $[0, 1)$ regardless the innovation methods used, and that the inversion method can ensure that we can always transform the background process to the desired distribution, that is, the TES models always have the distribution F regardless of F_v and ξ used. This decouples the fitting of marginal distribution and autocorrelation functions.

Through appropriate selection of ξ and F_v, some autocorrelation functions can be fitted very well.

2.2.1 Video Conference Model Using TES Model

Melamed et al. developed a model for the number of bits per group-of-blocks (GOB) using TES models [12, 13]. They found that the bit rate data contained a significant periodic component at the GOB level. They modeled the periodic component by:

$$P_n = \sum_{i=1}^{K} (A_i \cos \omega_i n + B_i \sin \omega_i n) \tag{2.11}$$

and removed it from the empirical data, where K, ω_i, A_i and B_i were estimated from the empirical trace. The residual process then becomes:

$$R_n = X_n - P_n \tag{2.12}$$

where X_n is the number of bits per GOB. The residual process was modeled by TES process with parameters $\alpha = 0.5, \phi = 0.3$ and $\xi = 0.5$ for stitching.

2.2.2 Generalized TES Model

In the previous model, the innovation process is $i.i.d$. Instead of using $i.i.d$ innovation process, Lazar et al. used a TES process with an innovation process which was not $i.i.d$, but depended on scene changes [15]. The scene change process was incorporated into the innovation process as:

$$V_n = (1 - W_n)(L + (R - L)Z_n) + W_n \left(-\frac{\alpha_c}{2} + \alpha_c Z_n \right) \tag{2.13}$$

where $\{Z_n\}$ is $i.i.d$ with uniform distribution on $[0, 1)$. W_n is $i.i.d$ with Bernoulli distribution, and was used to model scene changes. Scene length has a geometric distribution with parameter:

$$p = \frac{1}{1 + E[L_n]} \tag{2.14}$$

where L_n is the duration of a scene, $E[L_n]$ is the expected value of L_n.

For MPEG coded video, the video frame can be classified into I, P, and B frames according to the GOP pattern. Each frame was modeled by a separate TES model. The model uses the background process U^+ for I, B frames, and U^- for P frames. It requires nine parameters, and is not analytical. The model captures SRD [16].

2.3 Markovian Models

Like Poisson processes, Markov processes have been studied extensively by mathematicians. Let $\{S_t : t \in \Re\}$ be a continuous time random process with a sample space Ω, then the process is said to be a Markov process if:

$$Pr\{S_{t_n} \leq x_n / S_{t_{n-1}} \leq x_{n-1}, \cdots, S_{t_1} \leq x_1\} = Pr\{S_{t_n} \leq x_n / S_{t_{n-1}} \leq x_{n-1}\} \quad (2.15)$$

for any $t_1 < t_2 < \cdots < t_n$ and any $x_i \in \Re$.

Unlike a renewal process, a Markov process does introduce certain dependence, and is thus an important class of modern traffic models. Markov chain models are more suitable to model traffic which can be divided into discrete states. The slotted system traffic can be modeled as a Markov chain [9], whose states correspond to the number of successive idle slots separating successive arrivals. The number of arrivals between the idle slots itself can also be modeled by a Markov process.

2.3.1 Markov Chain Model

The most representive one using Markov chain to model video traffic was developed by D. P. Heyman [17]. The traffic data used was a 30 minute video conference sequence with no scene change. There were three persons talking in the video. The model was created as follows. Let X_n be the number of cells per frame, Y_n be the integer part of $X_n/10$. They proposed to model $\{Y_n : n = 1, 2, \cdots, N\}$ as a Markov

chain with transition matrix P, whose element p_{ij} is estimated by:

$$\hat{p}_{ij} = \frac{\text{number of transitions from } i \text{ to } j}{\text{number of transitions out of } i} \tag{2.16}$$

The transition matrix for the random process Y_n can be approximated by:

$$P = \rho I + (1 - \rho)Q \tag{2.17}$$

where ρ is the autocorrelation at lag 1. I is the identity matrix. Each row of Q consists of probability density function of the empirical data. Each row of Q has the probabilities $(f_0, f_1, \ldots, f_K, F_K^c)$, defined by:

$$f_i = \binom{i + r + 1}{i} p^r (1 - p)^i \tag{2.18}$$

$$F_K^c = \sum_{i>K} f_i \tag{2.19}$$

where K is the peak rate in cells per frame, p and q were obtained from empirical data by approximating f_i from the histogram of the traffic data. In this case, the model is called discrete autoregressive (DAR(1)) model.

DAR(1) was also used by D. P. Heyman to model VBR broadcast video traffic [18]. Scene changes were taken into consideration, and identified by the second difference:

$$\Delta_i = \frac{(X_{i+1} - X_i) - (X_i - X_{i-1})}{\frac{1}{m}\sum_{k=1}^{m} X_{i-k}} \tag{2.20}$$

where X_i is the number of cells in the ith frame. When Δ_i is negative and large enough, a scene change is said to occur.

Frame sizes in every scene were modeled as a Markov chain or DAR, while the number of cells for the frame immediately following a scene change frame was estimated by:

$$Y_n = a + bX_n + \epsilon_n \tag{2.21}$$

where $\{\epsilon_n\}$ are independent and identically distributed normal random variables with zero mean. X_n is the number of cells in the nth scene change frame, and Y_n is the number of cells in the frame next to the nth scene change frame.

Scene length and scene changes were modeled as independent processes with some kinds of distributions, such as Gamma, Weibull and generalized Pareto distributions. No one distribution is applicable to all kind of traffic.

Instead of using the states of Markov chain to represent different bit rates, the states of a Markov chain can also be used to represent different deviations from mean value. This kind of model was proposed by Pancha and EI Zarki [19].

2.3.2 Markov Renewal Process (MRP)

Markov renewal process is more general than Markov process, and it is more suitable to model video traffic than Markov chain mentioned before. A Markov renewal process consists of two parts, a Markov chain $\{M_n\}$ and its associated arrival time process $\{\tau_n\}$, subject to the condition that $\{M_{n+1}, \tau_{n+1}\}$ depends only on $\{M_n\}$. Since the arrival time may have any distribution, the Markov renewal process is more versatile. One method to use Markov renewal process to model video traffic is to use the Markov chain to model the scene changes of the video while using the associated arrival process to model the scene length.

Lucantoni, et al. proposed a Markov renewal process [20], in which the rates of video streams were divided into 40 equidistant levels and assigned a state in the Markov chain to each level. Geometric distribution were fitted to sojourn time at each level. The difference between this kind of models and MMPP models is that, for MRP, the bit rate is fixed at each level instead of being probabilistic. Compared with DAR models, sample paths generated by this kind of model are more similar to empirical trace.

2.3.3 Markov Arrival Process (MAP)

Markov arrival process [9] is one kind of renewal process with tractability and versatility. The inter arrival process for MAP is a phase type renewal process. Unlike the phase-type process introduced before, MAP uses different initial distributions

for different samples. The initial distributions for restarting the Markov process in a MAP process depend on the previous transient state from which the Markov process enters into the absorption states. MAP process obeys the superposition rule, that is, the superposition of two independent MAPs results in a MAP. This property is quite useful in evaluating the performance of multiplexers because the multiplexed traffic can still be modeled as a single MAP model.

2.3.4 Markov Modulated Poisson Process

Markov modulated processes [21] are used widely in video traffic modeling. A Markov modulated process is a random process whose parameters (such as arrival rate and mean) are controlled (modulated) by a Markov process. They can be used to capture the randomness at different scale, and therefore, are versatile to capture traffic characteristics.

Let $M(t)$ be a continuous Markov process with state space $\{1, 2, \ldots, m\}$. If the probability law of a process is determined by the state of the current state of $M(t)$ completely, then the process is called a Markov modulated process. The Markov modulated process can be classified by the processes modulated. The most well known one is Markov modulated Poisson process.

Markov modulated Poisson processes are used widely in traffic modeling [22, 23, 24]. A Markov modulated Poisson process is a doubly stochastic process where the arrival rate of a Poisson process is defined by the state of a Markov chain, and therefore, called Markov modulated Poisson process. A two state MMPP is shown in Fig. 2.1.

This process introduces some correlations between successive inter-arrival time, and therefore can capture the bursty characteristics of video traffic to some extent. It is a special case of a MAP process, and enjoys the feature of tractable queuing analysis. Important properties of queuing systems can be derived. Like a

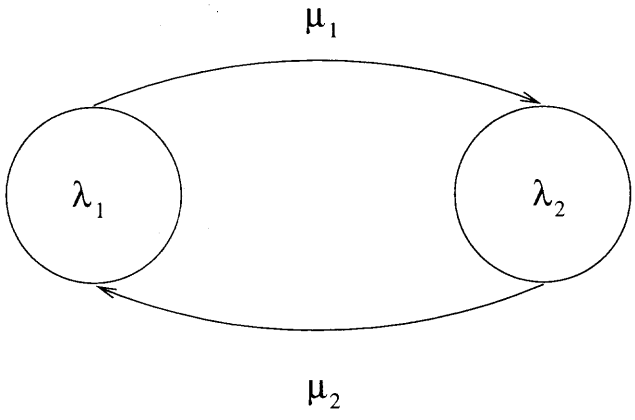

Figure 2.1 Two state MMPP

MAP process, the MMPP obeys superposition law, that is, the superposition of two MMPPs is still a MMPP. The property is very useful in the queuing performance analysis of multiplexed traffic.

With different number of states of the modulating Markov chain, the MMPP can be used to model different traffic [25, 24, 22]. The simplest case is to use MMPP to model On-OFF traffic. In this case, the arrival rate during the OFF period is zero, while the arrival process during the ON period is the arrival rate of traffic. This model is sometime denoted interrupted Poisson Process (IPP). Another simple example is to use two state MMPP to model video traffic with two different arrival rates. This is the so called Switched Poisson Process (SPP). An n-state MMPP was used to model the aggregated voice traffics, each with an ON-OFF behavior.

2.3.5 Markov Modulated TES Model

The bulk of previous work in video source modeling has been confined to short sequences of empirical data or video conference. There are few scene changes in these videos. To take into consideration of scene changes, Markov modulated TES model was proposed by B. Melamed [8]. The modeling of full length movies consists of the following five steps:

- Segment the full movies into several scenes.
- Cluster the scenes into a number of classes (smaller than the number of scenes).
- Assign a class tag to each scene.
- Model the bit rate of each class by a TES model.
- Model scene length as a renewal process

The transition from one class to another class is dominated by a Markov process, and therefore, the model for the whole movie is a Markov renewal modulated TES model. The movie was classified into four classes, and thus the model is a four state Markov renewal modulated TES model.

2.3.6 Markov Modulated Autoregressive Models

The Markov modulated autoregressive (AR) model was proposed by Yegenoglu et. al [26]. Bit rates of the video stream were quantized into N levels. Each level was assigned to a different state in a Markov chain. Corresponding to each state, there is an AR(1) process whose parameters are determined by the states, and therefore, for each state i, there are a unique set of coefficients for the AR processes.

Suppose Y_n is the number of bits of the nth frame. The state of the Markov chain at time n is X_n ($X_n \in \{0, 1, \cdots N-1\}$). The number of bits at time $n+1$ is given by:

$$Y_{n+1} = \begin{cases} a(i)Y_n + G(\mu(i), \sigma(i)^2), & \text{if } X_{n+1} = X_n = i \\ G(\eta(i), \nu(i)), & \text{if } X_{n+1} \neq x_n; X_{n+1} = i \end{cases} \tag{2.22}$$

where $a(i)$ is the autocorrelation at lag 1 and state i, and G is a Gaussian random variable. μ and σ are mean and variance of the Gaussian process, respectively. These parameter can be obtained as follows:

$$a(i) = 1 - \frac{D^2(i)}{2\nu(i)}, \tag{2.23}$$

$$\mu(i) = \frac{\eta(i)D^2(i)}{2\nu(i)}, \tag{2.24}$$

$$\sigma(i)^2 = D^2(i)\left(1 - \frac{D^2(i)}{4\nu(i)}\right), \tag{2.25}$$

$$D^2(i) = E[(Y_{n+1} - Y_n)^2 | X_{n+1} = X_n = i], \tag{2.26}$$

where, D, ν, and η are estimated from empirical data using the formulae.

From the formulae we can see that the traffic rates at the same level are modeled by AR processes, while the number of bits for the first frame after a level change is the sample of a Gaussian process.

2.4 Histogram Based Model

Histogram based models were used to model the VBR video traffic at the cell level [27]. They are applicable to the case when the traffic is smoothed using uniform smoothing mode or deterministic smoothing mode. Results were obtained based on the assumption that inter-arrival time within a frame is Poisson distributed. Under this assumption, the system behaves approximately like an M/D/1/K queue on the frame-by-frame basis. For different frames, the cell arrival rate may be different.

Ignoring the transient queuing behavior, the queuing performance for every frame can be obtained as a function of λ from the M/D/1/K system. The performance of the system can be obtained by conditioning the performance over the range of λ.

Suppose the buffer occupancy is $p(n|\lambda = \lambda_I)$ under the condition that the arrival rate is λ_I, then the buffer occupancy distribution would be

$$p(n) = \sum_{I=1}^{N} p(n|\lambda = \lambda_I)p(\lambda = \lambda_I), \tag{2.27}$$

where $p(\lambda = \lambda_I)$ is the histogram of arrival rate λ, and N is the number of intervals (bins) in the bit rate histogram.

2.5 Auto-Regressive Models

Auto-regressive processes have been used extensively in modeling VBR video traffic. Auto-regressive models introduce the dependence while keeping the randomness at some extent. Intuitively, the size of one frame depends on the size of the preceding frames. Perhaps this is the reason that the auto-regressive models were used extensively. An AR process of order p has the following form:

$$X_n = a_0 + \sum_{r=1}^{p} a_r X_{n-r} + \epsilon_n, \qquad n \in [1, 2, \cdots]. \tag{2.28}$$

It can be easily verified that the autocorrelation function (ACF) of an $AR(p)$ process can be expressed as:

$$\rho_n = \sum_{r=1}^{p} a_r \rho_{n-r}, \tag{2.29}$$

where ρ is the ACF at lag n.

ARMA processes have also been used to model video traffic. An ARMA process is given by:

$$X_n = a_0 + \sum_{r=1}^{p} a_r X(n-r) + \sum_{k=0}^{q} b_k e(n-k). \tag{2.30}$$

In general, for video conference, the traffic can be modeled as AR(1) models [17, 28]. If scene changes are included in the video, every scene can be modeled by an AR model, and the scene length can be modeled by another process.

2.5.1 Autoregressive Models at Cell Level

The AR(1) models, proposed by Maglaris et al, [29] and Nomura at al [30], have the following form:

$$X_n = a + bX_{n-1} + \epsilon_n, \tag{2.31}$$

where X_n is the number of cells in the nth frame, and ϵ_n is an $i.i.d$ white Gaussian noise process.

2.5.2 Autoregressive Moving Average (ARMA) Models

Grunenfelder, et al [31] developed a model, from a four second video conference sequence, for the number of cells generated by an encoder. The random process, X_i, representing the number of cells, is given by:

$$X_i = g(\alpha X_{i-m} + Y_i + v_i), \qquad |\alpha| < 1 \tag{2.32}$$

$$Y_i = \sum_{k=-m/2}^{m/2} h_k \varepsilon_{i-k} \tag{2.33}$$

where v_i is white noise. The parameters for the model were estimated from the long-term mean, variance and auto-covariance of the empirical sequence. The coefficients, h_k, of the MA part were obtained from Fourier analysis. $g(.)$ is a zero-mean nonlinearity function, which has the form of $aV_i + b$. This model requires 10003 parameters.

2.5.3 Composite AR and Markov Chain Model

Ramamurthy and Sengupta proposed a composite model consisting of three processes, two of which were AR processes and the other was Markov chain [32]. One of the AR processes was used to match short range dependence, while the other was used to match long range dependence. Markov chain was used to capture scene changes. The model has the following form:

$$T_i = X_i + Y_i + Z_i, \tag{2.34}$$

where

$$X_i = a_1 X_{i-1} + A_i, \tag{2.35}$$

$$Y_i = a_2 Y_{i-1} + B_i, \tag{2.36}$$

and

$$Z_i = K_i C_i. \tag{2.37}$$

A_i and B_i are normally distributed. Z_i is used to generate extra bits to simulate a scene change. C_i is normally distributed, whose mean and variance values depend on K_i. K_i is the state of a Markov chain with $K_i \in \{0, 1, 2\}$. The model for scene change is shown in Fig. 2.2. The Markov process is used to generate the two frames following a scene change, that were observed to have larger frame size than other frames. The scene was assumed to have binomial distribution.

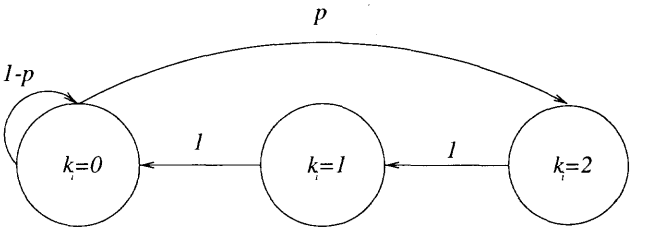

Figure 2.2 Markov Chain of scene change process.

2.6 "Scenic" Model

Frater et al. [33] proposed a model based on scene change. This model can be seen as a generation of DAR(1). Scene changes were also identified, but with a slightly different formula. A combination of median and averaging filters were used to detect scene changes. It models VBR video in the following way:

1. The sequence of scene lengths is $i.i.d$ and has the distribution $p(x) = \frac{a}{x^n + b^2}$, where a and b are constants, and n is estimated from the empirical trace.

2. The sequence of mean rate for every scene is $i.i.d$ and has negative binomial distribution;

3. When there is sufficient jump in the bit rate, there is definitely a scene change;

4. Within a scene, the sequence of number per frame is $i.i.d$;

5. The variation of the bit rate within a scene can be ignored.

6. The process controlling the scene length and the process generating bit rates were independent.

The frame size in every scene was modeled as a DAR process, and the time spent at each level has negative-binomial distributions. No Markovian chain was used, and significantly better results were obtained.

2.7 Fluid Models

Unlike the models introduced so far, fluid models treat traffic as a fluid stream [34], that is, cells arrive continuously, not at discrete time. The fluid approach can capture the burstiness of ATM traffic, and is thus appropriate for ATM traffic. In most cases, fluid models have two states: ON and OFF. The durations of ON and OFF are random and *i.i.d.* The traffic arrives at constant rate during the On period while there is no traffic during the OFF period. The distribution of ON and OFF period are different. They may be exponentially distributed or heavy-tail distributed. Obviously, it is suitable to model traffic on large scale.

Although the fluid model is conceptually simple and analytically tractable, it is also beneficial for simulations because enormous savings in computing can be obtained, due to the fact that infeasible cell arrival simulation can be replaced by feasible simulations of fluid models with comparable accuracy. In the queuing context, the waiting time is the time needed to clear the buffer, while the loss probability can be calculated in terms of overflow volumes.

A Markov modulated fluid model can also be used to model traffic with different arrival rates [35]. Its queuing performance can be analyzed as Markov modulated constant rate traffic. This kind of model is more suitable for VBR video traffic. To model VBR video traffic, the arrival rate is often quantized into several bins, with each corresponding to one constant rate. Histogram for each constant rate and the probability from one rate to another rate can be obtained from the empirical trace.

2.8 $M/G/\infty$ Input Process Model

In an $M/G/\infty$ queuing system, the number of busy servers is a random process. An $M/G/\infty$ input process can be defined by an $M/G/\infty$ queuing system, and its ACF form can be controlled by choosing the distribution of service time [36]. The $M/G/\infty$ input process was proposed as a model of VBR intra-coded video streams.

An $M/G/\infty$ input process can be defined as follows: For the discrete-time $M/G/\infty$ queue system, the arrival is an $i.i.d$ Poisson process with mean rate λ. Let the number of customers arrived during the slot $[n, n+1)$ be ξ_{n+1}. All the customers arrived during this period will be served at the beginning of the next slot, that is, the customers arrived in the slot $[n, n+1)$ will be served at the beginning of slot $[n+1, n+2)$. Suppose the service time is $i.i.d$ with a common distribution G, and the service time for customer i arrived in the slot $[n, n+1)$ is $\sigma_{n+1,i}$, $i = 1, 2, \ldots, \xi_{n+1}$. Initially, there are b_{0-} customers in the system, and the time needed to serve these customers are $\sigma_{0,1}, \sigma_{0,2}, \ldots, \sigma_{0,b_{0-}}$. If $\sigma_{0,1}, \sigma_{0,2}, \ldots, \sigma_{0,b_{0-}}$ are mutually independent, the process b_n, which is the number of customers in the system at time n, is then called an $M/G/\infty$ input process.

The ACF of b_n can have many kinds of forms, depending on the distribution of G used. When G is a Pareto distribution, the process possesses long range dependence[36]. In general, b_n is not stationary, but there exists a stationary and ergodic process b_n^* such that :

$$\{b_{n+k}, n = 0, 1, \cdots\} \Rightarrow \{b_n^*, n = 0, 1, \cdots\} \qquad \text{as } k \to \infty \tag{2.38}$$

and the stationary and ergodic version b_n^* was used to model the video traffic. If the distribution of service time G is Weibull-like, the ACF has the form $\rho_k = e^{-\beta\sqrt{k}}$, which provides a good fit to the empirical ACF. The ACF is summable, and therefore, it is a SRD model.

2.9 GBAR Model

GBAR model was proposed by D. P. Heyman [37]. The GBAR model is formed by two random variables, one with Gamma distribution and the other with Beta distribution. The gamma random variable, denoted as $Ga(\beta, \lambda)$, has the following distribution:

$$f_G(t) = \frac{\lambda(\lambda t)^\beta}{\Gamma(\beta+1)} e^{-\lambda t}. \tag{2.39}$$

The Beta random variable, denoted as $Be(p, q)$, has the following distribution:

$$f_B(t) = \frac{\Gamma(p+q)}{\Gamma(p+1)\Gamma(q+1)} t^{p-1}(1-t)^{q-1}, \qquad 0 < t < 1 \tag{2.40}$$

It is well known that the sum of two independent gamma random variables with distribution $Ga(\alpha, \lambda)$ and $Ga(\beta, \lambda)$ is still a gamma variable with distribution $Ga(\alpha+\beta, \lambda)$, and the product of two independent Beta variables with distribution $Be(\alpha, \beta-\alpha)$ and $Ga(\beta, \lambda)$ is a gamma random variable with distribution $Ga(\alpha, \lambda)$. From the results, if X_{n-1} has $Ga(\alpha, \lambda)$ distribution, A_n has $Be(\alpha, \beta-\alpha)$ distribution, and B_n has $Ga(\beta-\alpha, \lambda)$ distribution, and if the three variables are mutually independent, then the variable:

$$X_n = A_n X_{n-1} + B_n \tag{2.41}$$

is still a gamma random variable with distribution $Ga(\beta, \lambda)$. The process $\{X_n\}$ is stationary with autocorrelation function:

$$r(k) = \left(\frac{\alpha}{\beta}\right)^k, \qquad k = 0, 1, 2, \cdots. \tag{2.42}$$

The process $\{X_n\}$ is a AR(1) process, and hence, is called GBAR(1) process.

The mean and variance of process $\{X_n\}$ are β/λ and β/λ^2. They can be estimated from empirical data. Suppose that the mean and variance of the empirical data are m and ν, then λ and β can be estimated by:

$$\hat{\lambda} = \frac{m}{\nu} \tag{2.43}$$

and

$$\hat{\beta} = \frac{m}{\nu^2}. \tag{2.44}$$

The parameter α can be estimated from the autocorrelations of the empirical data by equation (2.42). Assume that the data have the property:

$$r(k) \approx \rho^k, \ k = 1, 2, \cdots, K \tag{2.45}$$

for some sufficiently large K, then

$$\hat{\alpha} = \hat{\rho}\hat{\beta} \tag{2.46}$$

The GBAR model can accurately predict the cell loss rate and mean queue size for video conferences coded with H.261 based coder. It is more accurate than the DAR model.

2.10 Long Range Dependent Models

ACFs of the models introduced so far have exponential forms. Such ACFs are summable, that is, $\sum_{k=0}^{\infty} \rho_k < \infty$. The power spectrums are bounded at low frequency. Recently, a number of studies supported by extensive statistic analysis indicate the presence of persistent correlations over a large scale. Actually, it has been found that such phenomenon occurs quite often in nature [38, 39], such as rainfall, the annual growth of tree rings, hydraulics, and economics. This kind of phenomenon is also found in ATM, LAN, WAN, and VBR video traffic, and is well described by long range dependence processes [40, 6, 5, 41].

A long range dependence process has a ACF that is not summable, i.e., $\sum_{k=0}^{\infty} \rho_k = \infty$. Its power spectrum at low frequency is unbounded and approaches infinity as the frequency approaches zero [42]. It has been argued that the queuing performance of network in the presence of LRD traffic can not be evaluated accurately by Markov models. There are evidences that the presence of long term correlations

will degrade the queuing performance [43]. New models having LRD behavior should be used to model network traffic and capture the LRD characteristics.

2.10.1 Fractional ARIMA Model (FARIMA)

Fractional ARIMA process is a generalized ARIMA process [44, 45, 46]. ARIMA is an incremental process which is defined by a difference operator. Let Δ be the difference operator, then:

$$\Delta X_k = (X_k - X_{k-1}) \tag{2.47}$$

The operation can be iterated as follows:

$$\Delta^2 X_k = (X_k - X_{k-1}) - (X_{k-1} - X_{k-2}) \tag{2.48}$$

and

$$\Delta^n X_k = \sum_{i=0}^{n} \binom{n}{i} (-1)^i X_{k-i}, \; n = 1, 2, 3, \cdots \tag{2.49}$$

where n is an integer. X_k is the so-called ARIMA process. If the difference equation above is generalized to the non-integer case, then a fractional ARIMA process is obtained. The generation can be done by Gamma function in the following way:

$$\Delta^d X_k = \sum_{i=0}^{\infty} \binom{d}{i} (-1)^i X_{k-i}, -1/2 < d < 1/2 \tag{2.50}$$

where $\binom{d}{i}$ is generalized factorial function.

$$\binom{d}{i} (-1)^i = \frac{\Gamma(-d+i)}{\Gamma(-d)\Gamma(i+1)} \tag{2.51}$$

$\Gamma(x) \triangleq \int_0^\infty t^{x-1} e^{-t} dt$ is the gamma function.

It can be verified that the ACF of FARIMA process has the following form:

$$\rho_k = \frac{\Gamma(d+1)}{\Gamma(d) k^{2d-1}}. \tag{2.52}$$

If $0 < d < 0.5$, the process exhibits LRD.

Fractional ARIMA process can be approximated by linear process of the form:

$$X_k = \sum_{i=0}^{I} c_{k-i}\varepsilon_i \tag{2.53}$$

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

$$c_k = \frac{\Gamma(k+d)}{\Gamma(d)\Gamma(k+1)} \tag{2.54}$$

c_k can be iteratively obtained as follows:

$$c_0 = 1, \tag{2.55}$$

$$c_{k+1} = \frac{k+d}{k+1}c_k \tag{2.56}$$

ε_k is $N(0,\sigma)$. d can be estimated from empirical data, and it is the only parameter required by the model..

2.10.2 Fractional Gaussian Noise Model (FGN)

FGN is obtained from stationary increments of a fractional Brownian motion (FBM). Fraction Brownian motion $B_H(t)$ is a Gaussian process with the following properties [44]:

1. $E[B_H(t)] = 0$
2. $B_H(0) = 0$
3. $B_H(t+\delta) - B_H(t)$ has the distribution $N(0, \sigma|\delta|^H)$
4. $B_H(t)$ is an increment independent process
5. $E[B_H(t)B_H(s)] = \sigma^2/2(|t|^{2H} + |s|^{2H} - |t-s|^{2H})$,

where $0 < H < 1$ is the Hurst parameter. It is clear that $var[B_H(t)] = \sigma^2|t|^{2H}$.

The increment is known as FGN. Let $G_H(t)$ be the increment of $B_H(t)$, that is, $G_H(t) = \frac{1}{\delta}(B_H(t+\delta) - B_H(t))$, then $G_H(t)$ is a stationary process with the following properties:

1. $G_H(t)$ has distribution $N(0, \sigma\delta^{H-1})$

2. $E[G_H(t+\tau)G_H(t)] = \sigma^2 H(2H-1)|t|^{2H-2}$ for $\tau \gg t$.

For the discrete case, the ACF of the FGN is given by:

$$\rho_k = \frac{1}{2}\left(|k+1|^{2H} - 2|k|^{2H} + |k-1|^{2H}\right) \tag{2.57}$$

.

If $0.5 < H < 1$, then, as $k \to \infty$, $\rho_k \to H(2H-1)k^{2H-2}$, and the FGN exhibits LRD. H is the only parameter required for this model.

2.11 Deterministic Models

The models introduced so far are statistical models. In general, most statistic models are not powerful enough to capture the burstiness, or they are too complicated to analyze, and therefore are difficult to be used for connection admission control (CAC) [47]. Furthermore, if network resources are allocated based on statistical models, it is often impossible to provide clients guaranteed QoS.

Deterministic models have no such drawbacks. If traffic is modeled by deterministic models, then CAC can be easily implemented, and the end-to-end performance can be easily guaranteed. The drawbacks of deterministic traffic models are that the number of required parameters may be very large, and if the network is managed based on deterministic models, its efficiency may be very low.

A deterministic traffic model is one that parametrically describes the worst case behavior of a traffic stream. It provides deterministic upper bound on each source's arrivals, and thus, network resources are allocated based on the worst case requirements.

A number of deterministic models have been proposed and some of which have been adopted by ATM, such as leaky-bucket or (σ, ρ) models, and (peak-rate, bursty-length, average-rate) models. Those models are well known already. In this section,

we only describe the deterministic bounding interval-length dependent (D-BIND) model.

The task to define a deterministic model for video traffic is to find a traffic constraint function. Let $A_j(t_1, t_2)$ be the total number of bits arriving on connection j in the interval $[t_1, t_2]$, if $A_j(t, s) \leq b_j(t), \forall s, t > 0$, then $b_j(t)$ is a traffic constraint function.

The D-BIND model uses several rate-interval pair (R_k, I_k). R_k is a bounding rate or worst case rate over interval I_k. It is clear that the model can capture traffic burstiness at different time scale. Using the pairs of parameters, a bounding function for traffic can be obtained as:

$$b(t) = \frac{R_k I_k - R_{k-1} I_{k-1}}{I_k - I_{k-1}}(t - I_k) + R_k I_k, \tag{2.58}$$

where $I_{k-1} \leq t \leq I_k$. It is a piecewise linear upper-bound function.

CHAPTER 3

SELF-SIMILAR PROCESS

3.1 Introduction

Recent studies and statistical analysis of high-quality, high-resolution traffic, supported by extensive measurements, have revealed a new phenomenon, which is an important ramification to network modeling, design, and management [6, 36, 8]. It is found, in the analysis of hundreds of millions of observed packets on several Ethernet LAN's at the Bellcore Morristown Research and Engineering Center, and in the analysis of a few millions of observed frame data of VBR video, that the video traffic appears to be statistically *self-similar*, that is, the statistical characteristics of traffic seem to be similar at any time scale. Self-similar traffic is characterized by "burstiness" across an extremely wide range of time scale. Unlike any other random processes, the aggregation of self-similar processes still has the characteristics of self-similarity. This is contrary to the common belief that the burstiness will disappear after the aggregation of many traffic streams.

Self-similar process is one kind of LRD process and the self-similar phenomenon can also be found in many disciplines, such as hydraulics and economics. Hurst found the phenomenon when he studied the long-term storage in water reservoirs [38]. Statistical self-similarity manifests itself in a variety of different ways: a spectral density function that diverges at the origin, and a non-summable autocorrelation function.

The indication of self-similar phenomenon in network traffic spurred an on going debate on whether self-similar models should be used in network performance evaluation and resource management. Some of the results support the view that self-similarity has drastic impact on queuing performance [48, 49, 50, 51, 43], while other results support the view that self-similarity has little impact on queuing performance because of the fact that the buffer capacity is limited in practice [52].

More work is required to demonstrate the validity of self-similar models. For example, the following questions need to be answered [6]: 1) What is the physical explanation of self-similarity; 2) What is the impact of self-similarity on network and protocol design and performance analysis. Analytical techniques need to be developed to analyze the impact of self-similarity on network performance.

3.2 Self-Similarity

Self-similarity is not a new idea. It is used commonly in the theory of fractals and chaos. Self-similarity is a concept extended from the shape similarity in basic geometry. A figure is said to be self-similar if the figure is composed of small figures with the similar shape, and the small figure is composed of even smaller figures with the similar shape, and so on (see Fig.3.1 for a self-similar figure).

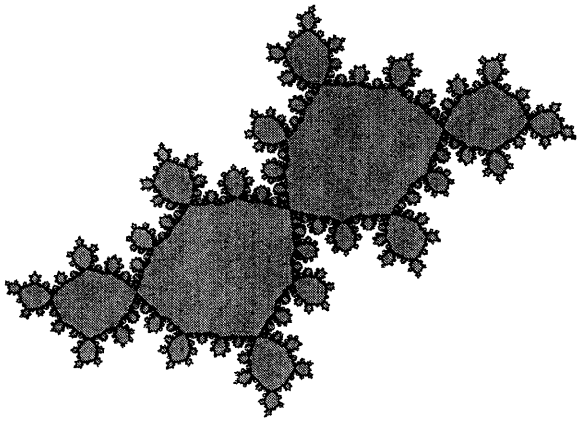

Figure 3.1 An example of self-similar picture

The concept of self-similarity is adopted to a random process to describe the phenomenon that the process has the same statistical characteristics at any time scale. Now let us consider a stationary process $X = \{X_n : n = 1, 2, \ldots\}$ with mean

μ and variance σ^2. The autocorrelation function and the variance of X are denoted as:

$$r(k) = \frac{E[(X_n - \mu)(X_{n+k} - \mu)]}{\sigma^2} \tag{3.1}$$

and

$$\sigma^2 = E[(X_n - \mu)^2]. \tag{3.2}$$

X is said to be SRD if $\sum_{k=0}^{K=\infty} r(k)$ is finite; otherwise, the process is said to be LRD [42].

Let X defined above have the following autocorrelation function:

$$r(k) \to k^{-\beta} L(k), \qquad k \to \infty \tag{3.3}$$

where $0 < \beta < 1$, and L is a slowly varying function as $k \to \infty$, i.e., $\lim_{t\to\infty} L(tx)/L(t) = 1$ for all $x > 0$. Consider the aggregated process

$$X^{(m)} = \{X_t^{(m)}\} = \{X_1^{(m)}, X_2^{(m)}, \ldots\},$$

where

$$X_t^{(m)} = \frac{1}{m}(X_{tm-m+1} + \cdots + X_{tm}), \qquad t, m \in \{1, 2, \cdots\}. \tag{3.4}$$

X is said to be exactly second-order self-similar if

$$var X^{(m)} = \sigma^2 m^{-\beta} \tag{3.5}$$

and

$$r^{(m)}(k) = r(k) \tag{3.6}$$

for all $m \in \{1, 2, 3, \cdots\}$ and $k \in \{0, 1, 2, \cdots\}$ [42]. Here, $r^{(m)}(k)$ is the autocorrelation function of $X^{(m)}$. In fact, Equation (3.5) is sufficient to define a self-similar process, since Equations (3.3) and (3.6) can be derived from Equation (3.5). The statement that Equation (3.5) is sufficient to define an exact self-similar process follows directly from the following statements [42].

Statement 1 *Process X satisfies condition (3.5) , if and only if its autocorrelation function is:*

$$r(k) = \frac{1}{2}\left[(k+1)^{2-\beta} - 2k^{2-\beta} + (k-1)^{2-\beta}\right] \triangleq g(k), \qquad 0 < \beta < 1, k \in \{0, 1, \cdots\} \tag{3.7}$$

Statement 2 *For the autocorrelation function $r(k) = g(k)$, the following limiting equation holds:*

$$\lim_{k\to\infty} \frac{r(k)}{k^{-\beta}} = \frac{1}{2}(2-\beta)(1-\beta) = H(2H-1) \tag{3.8}$$

where

$$H = 1 - \frac{\beta}{2}, \qquad 0 < \beta < 1$$

Statement 3 *Process X satisfies the condition;*

$$r^{(m)}(k) = r(k), \qquad k \in \{0, 1, \cdots\}, m \in \{2, 3, \cdots\} \tag{3.9}$$

if the condition (3.5) holds or if $r(k) = g(k)$ for $0 < \beta < 1$.

Statements 1 and 3 tell us that Equation (3.5) and Equation (3.6) are equivalent. Statement 2 implies that Equation (3.3) and Equation (3.5) are equivalent. Condition (3.5) gives explicit expression for $r(k)$. This expression only allows $L(k)$ to be some constants, but not to be any slowly varying function.

The following statement yields the spectral density function of a self-similar process.

Statement 4 *Process X satisfies the condition (3.5) if and only if its spectral density function is:*

$$f(\lambda) = c|e^{2\phi i\lambda} - 1|^2 \sum_{l=-\infty}^{\infty} \frac{1}{|\lambda + l|^{3-\beta}}, \qquad -\frac{1}{2} \le \lambda \le \frac{1}{2} \tag{3.10}$$

where c is a constant given by normalization, $\int_{-(1/2)}^{1/2} f(\lambda)d\lambda = \sigma^2$.

From the above discussion, the following definition of exactly second order self-similar process is obtained [42].

Definition 1 *Process X is called exactly second-order self-similar with parameter $H = 1 - \beta/2, 0 < \beta < 1$, if its autocorrelation function is (3.7), i.e., $r(k) = g(k)$.*

It is apparent that a self-similar process is a kind of LRD process. Since empirical video traffic exhibits self-similarity and long range dependence, it is intuitive to use self-similar processes to model video traffic. This is one of the most often used processes to capture LRD of video traffic.

Hurst parameter $H = 1 - \beta/2$, $(0 < \beta < 1)$ is used to measure the similarity of a process. It is the only parameter needed to describe a second-order self-similar process. For a process with self-similarity, $1/2 < H < 1$.

3.3 Asymptotically Second-order Self-Similar Process

Besides the exact second-order self-similar process, the so called asymptotically second-order self-similar process is also commonly used in video traffic model. The asymptotically second-order self-similar process is defined as [42]:

Definition 2 *Process X is called asymptotically second-order self-similar with parameter $H = 1 - \beta/2, 0 < \beta < 1$, if for all $k \in \{1, 2, \ldots\}$,*

$$\lim_{m\to\infty} r^m(k) = \frac{1}{2}[(k+1)^{2-\beta} - 2k^{2-\beta} + (k-1)^{2-\beta}] \triangleq g(k). \tag{3.11}$$

The asymptotically second-order self-similar process has the following properties:

Statement 5 *if*

$$\lim_{k\to\infty} \frac{r(k)}{k^{-\beta}} = c, \qquad 0 < \beta < 1 \tag{3.12}$$

where $0 < c < \infty$ is a constant, then

$$\lim_{k\to\infty} \frac{varX^m}{m^{-\beta}} = c \tag{3.13}$$

where c is also a constant.

Statement 6 *If (3.12) holds, then*

$$\lim_{m\to\infty} r^{(m)}(k) = g(k) \tag{3.14}$$

Processes generated by F-ARIMA algorithm are asymptotically self-similar.

3.4 Estimation of Hurst Parameter

Many methods for estimating the self-similarity parameter H or the intensity of long-range dependence in a time series are available, some of which are described in details by Beran in his book [44]. The methods encountered often are aggregated variance method, the R/S method, Periodogram Method, and Whittle estimator method. Two of them are described here.

3.4.1 Aggregated Variance Method

The aggregated method estimates the Hurst parameter based on the aggregated process described by Equation (3.2). For successive values of m, get the corresponding aggregated process $X^{(m)}$, estimate the variances of the process $X^{(m)}$ for each m using the following formula,

$$\widehat{Var}X^{(m)} = \frac{1}{N/m}\sum_{t=1}^{N/m}(X_t^{(m)})^2 - \left(\frac{1}{N/m}\sum_{t=1}^{N/m}X_t^{(m)}\right)^2, \tag{3.15}$$

and then plot the logarithm of the $\widehat{Var}X^{(m)}$ versus $\log m$, where N is the length of the time series. If the process is self-similar, the plot should form a straight line, with slope $\beta = 2H-2, -1 \le \beta < 0$. By measuring the slope, the Hurst parameter H can be obtained. This method can be used to estimate H of the F-ARIMA process and FGN process.

3.4.2 Absolute Values of the Aggregated Series

For this kind of method, the following values are calculated for every m,

$$\frac{1}{N/m}\sum_{t=1}^{N/m}\left|X_t^{(m)}\right|, \tag{3.16}$$

then the logarithm of this value is plotted versus $\log m$. If the process is self-similar, the plot should be a line with slope $H - 1$.

CHAPTER 4

MODELING VIDEO TRAFFIC USING MARKOV-MODULATED SELF-SIMILAR PROCESSES

4.1 Introduction

Some statistical traffic models introduced in Chapter 2 can be categorized into two classes:

- Short Range Dependence (SRD) models,
- Long Range Dependence (LRD) models.

They are used to capture two statistical factors: marginal distribution (first-order statistics) and autocorrelation function (second-order statistics) of traffic data. LRD models can capture long range dependence, while SRD models can capture short range dependence.

While the importance of long range dependence is arguable, the impact of short term autocorrelation in traffic processes on queuing performance of finite buffer can be very drastic (see [8] and references in it). Simulation results show that the queuing performance of network with strong and weak autocorrelation traffic may be quite different.

Thus, a model should capture not only the first-order statistics, but also the second-order statistics. SRD models (such as DAR(1), MMRP, Fluid Flow, and Regression models) can capture short-term autocorrelation, but fail to capture long-term dependence. LRD models, on the other hand, can capture long-term dependence, but underestimate the short term dependence.

The importance of long range dependence is among the most arguable issues in video modeling. Most of the results, however, support the view that LRD has drastic impact on queuing performance [48, 49, 50, 51]. Only a few results support the view that LRD has little impact on queuing performance [43]. It is at least not harmful to capture the LRD by a traffic model.

Most of the work in video source modeling had been largely confined to a short period of video sequence and video conference. The scene change or drastic motion frames are rare in these sequences. As a result, bit rates are relatively low and bit rate changes are rather small compared with that of full motion movies.

The $M/G/\infty$ input process model is a compromise between LRD and SRD models [36]. Simulation results were found to be better than those of self-similar process when the switch buffer is relatively small. Better results than those of DAR(1) model was found when the buffer size is large. The results were obtained from JPEG and MPEG-2 I sequences. As will be shown below, ACF of MPEG sequences is quite different from that of JPEG sequence or that of I sequences. In our opinion, it is almost impossible to accurately capture the ACF of MPEG compressed data by a simple function such as the exponential function, and thus this method fails to capture the second-order statistics of MPEG sequences.

Markov-Renewal-Modulated TES (transform expand sample) models was used to model JPEG encoded motion pictures. One of the drawbacks of TES approach is that the ACF of a TES process for lags beyond one can not be derived analytically. It can only be obtained by searching in the parameter space, and thus good results can hardly be guaranteed [36]. One of the important tasks of traffic modeling is to obtain an analytical model so that the network performance can be obtained analytically. TES model fails to provide such an analytical model.

The ACF of MPEG and JPEG coded video sequences are quite different, and the models used so far for video sequence can not characterize the MPEG coded video. We propose to model MPEG compressed video sequence by the Markov modulated self-similar processes. The basic idea behind our proposed model is to decompose the original video sequence into several sequences that can be modeled by self-similar processes. It has been found that video traffic possesses self-similarity, and thus it is natural to model video traffic by self-similar processes. Self-similar

processes have very simple ACF forms, and therefore, are easier to analyze than other kinds of processes. The model tries to capture the SRD of the MPEG coded video by a Markov process, and the LRD by a self-similar process.

4.2 Empirical Data and ACF

The empirical data used here was MPEG-I coded data of *Star Wars*[1]. The source contains materials ranging from low complexity/motion scenes to those with high and very high actions.

The data file consists of 174,136 integers, whose values are frame sizes (bits per frame). The movie length is approximately 2 hours at 24 frames per second. The original video was captured as 408 lines by 508 pels, and then interpolated to 240×352 (Luminance - Y), and 120×176 (Chrominance - U and V). Every frame was partitioned into blocks of 8× 8 pixels. These data blocks were transformed using discrete cosine transform (DCT). After the DCT transformation, coefficients were quantized and Huffman coded. Run length coding was further used to reduce bit rate. Motion estimation techniques were used to compress data volume. The frames were organized as follows: IBBPBBPBBPBB IBBPBB..., i.e., 12 frames in a Group of Pictures (GOP). I frames are those which use intra frame coding method (without motion estimation), P frames are those which use inter frame coding technique (with motion estimation), and B frames are predicted using both forward and backward prediction.

The ACF of frame size of MPEG coded *Star War* is shown in Fig 4.1, and it is quite different from the ACF of frame size of JPEG coded movies *Star Wars* (see Fig 4.2). The ACF of MPEG coded data fluctuates around an envelope, reflecting the fact that, after the use of motion estimation techniques, the dependence between

[1]The MPEG-I coded data were the courtesy of M.W.Garrett of Bellcore and M.Vetterli of UC Berkeley.

frames is reduced. This characteristic should be taken into consideration in modeling MPEG coded video sequences. We propose to use different self-similar processes with different ACF to reflect the fluctuation of ACFs. The basic idea behind this method is to divide the sequence into three different sequences, each modeled by a separate self-similar process. The transition among these processes is governed by a Markov chain, whose transition matrix can be obtained from empirical data.

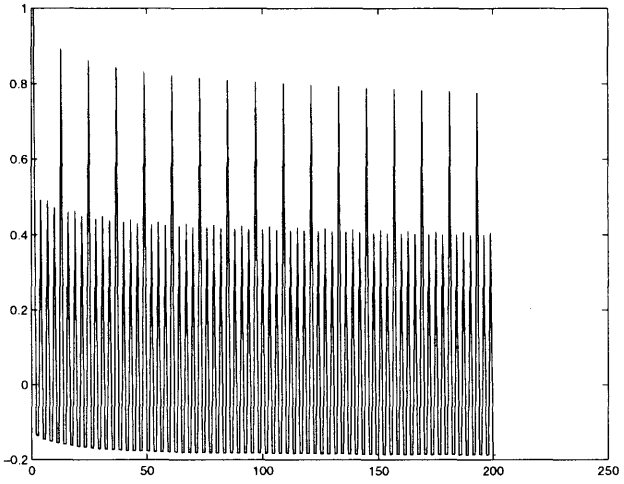

Figure 4.1 ACF of MPEG compressed video *Star Wars*

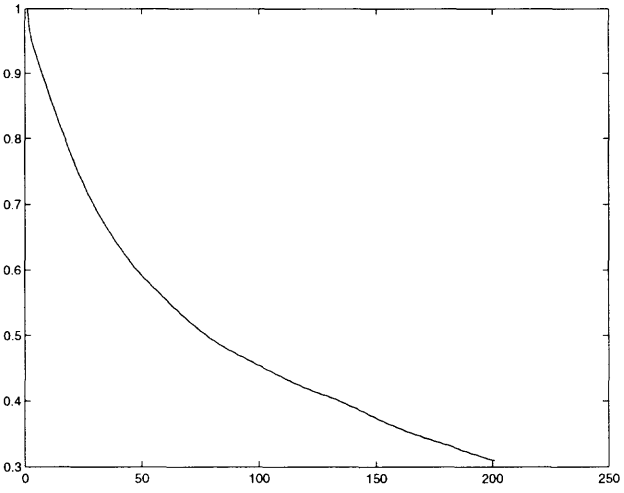

Figure 4.2 ACF of JPEG compressed video *Star Wars*

4.3 Classification of MPEG Data

It is apparent that the ACF of MPEG compressed video traffic can not be approximated by a single function $r(k) = k^{-\beta}$ because this kind of function decreases monotonously, while the ACF of a MPEG compressed video traffic fluctuates drastically. By comparing the JPEG compressed data and the MPEG compressed data, we may find that bit rate variation of the MPEG compressed video sequence is larger than that of the JPEG compressed video sequence. We therefore suggest to divide the traffic data into three different parts—inactive part, active part, and the most active part (authors in [27] also pointed out that a video bit rate process has three main components: a slowly changing component, a more quickly changing component, and an impulsive component). Suppose $f(i)$ is the number of bits in the *ith* frame. The video traffic can be classified as follows

1. If $f(i+1)/f(i) > T, i = 2, 3, \cdots$, then $f(i+1)$ belongs to the non-inactive part; otherwise, $f(i+1)$ belongs to the inactive part, where T is a threshold value.

2. Similarly, the non-inactive part can be classified into the active and most active part.

Taking these three data sets as three different random processes, we can calculate their ACFs.

4.4 Modeling of Classified Data

The ACF of each process is very different (as shown in Fig. 4.3, 4.4 and 4.5) from that of the original sequence. The fluctuation is no longer that big. We have used $k^{-\beta}$, $e^{-\beta k}$ and $e^{-\beta\sqrt{k}}$, corresponding to the ACFs of a self-similar process, a Markov process, and an $M/G/\infty$ input process, respectively, to approximate ACFs of these three processes. It is quite clear that $k^{-\beta}$ is a better approximation of ACFs of these

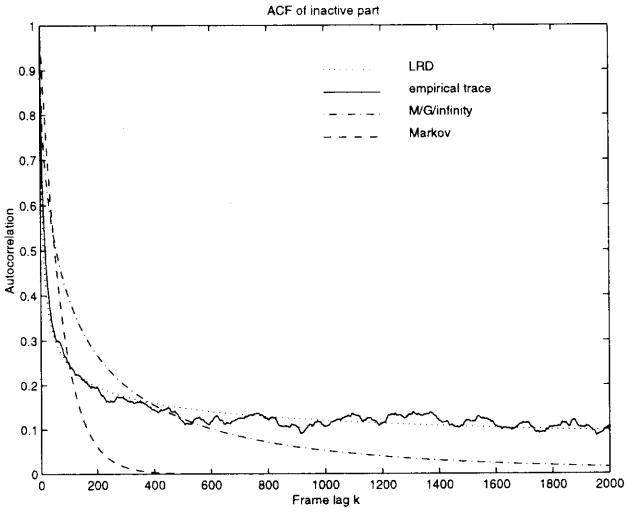

Figure 4.3 ACF of the inactive part of *Star Wars*

classified data, and we therefore use self-similar processes s_1, s_2, and s_3 to model these processes, respectively.

Using the least square method, we obtained $\beta = 0.3321$, 0.3069, and 0.4396 for the active, inactive, and most active part, respectively. The corresponding Hurst parameters for these processes are $H = 0.8339$, 0.8465, and 0.7802.

Beta distribution [53] is used to model the marginal distributions of these processes. The marginal distribution of a Beta distribution process has the following form

$$f(x;\gamma,\eta,\mu_0,\mu_1) = \begin{cases} \frac{1}{\mu_1-\mu_0}\frac{\Gamma(\gamma+\eta)}{\Gamma(\gamma)\Gamma(\eta)}(\frac{x-\mu_0}{\mu_1-\mu_0})^{\gamma-1}(1-\frac{x-\mu_0}{\mu_1-\mu_0})^{\eta-1} \\ \qquad\qquad \mu_0 \le x \le \mu_1, 0<\gamma, 0<\eta \\ 0 \qquad\qquad otherwise \end{cases} \tag{4.1}$$

where γ and η are shape parameters, and $[\mu_0, \mu_1]$ is the domain where the distribution is defined.

Beta distribution is quite versatile and can be used to model random processes with quite different shapes of marginal distributions. The following formulae are

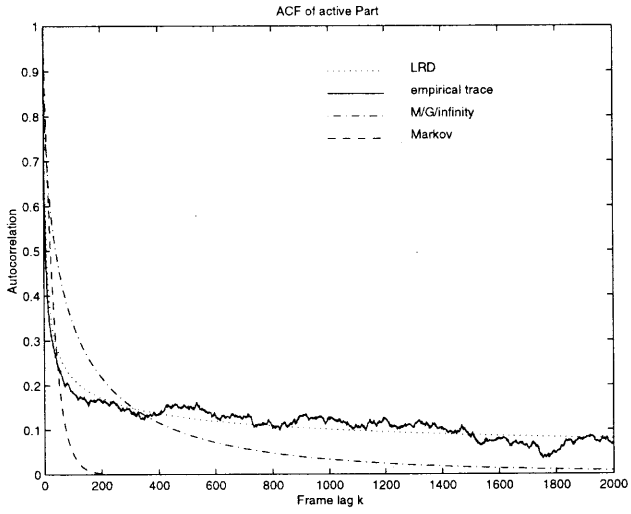

Figure 4.4 ACF of the active part of *Star Wars*

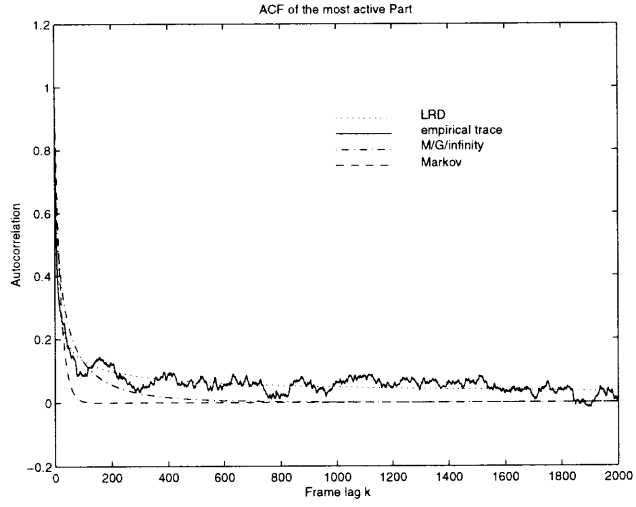

Figure 4.5 ACF of the most active part of *Star Wars*

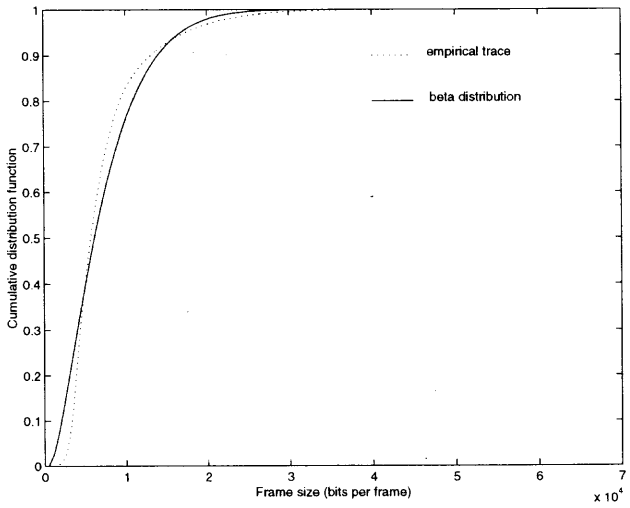

Figure 4.6 CDF of the inactive part and the corresponding Beta distribution

used to derive the parameters of Beta distribution [53]:

$$\hat{\eta} = \frac{1-\bar{x}}{s^2}[\bar{x}(1-\bar{x}) - s^2] \tag{4.2}$$

$$\hat{\gamma} = \frac{\bar{x}\hat{\eta}}{1-\bar{x}} \tag{4.3}$$

where

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i, \tag{4.4}$$

$$s^2 = \frac{N\sum_{i=1}^{N} x_i^2 - \left(\sum_{i=1}^{N} x_i\right)^2}{N(N-1)}, \tag{4.5}$$

and N is the number of data in the data set. Using the classified data sets, $\hat{\gamma} = 1.6179$, $\hat{\eta} = 13.7810$ for the inactive process, $\hat{\gamma} = 1.7977$, $\hat{\eta} = 12.1980$ for the active process, and $\hat{\gamma} = 5.3550$, $\hat{\eta} = 11.4134$ for the most active process. The marginal distributions of the empirical data and corresponding Beta distributions are shown in Fig. 4.6, 4.7, and 4.8.

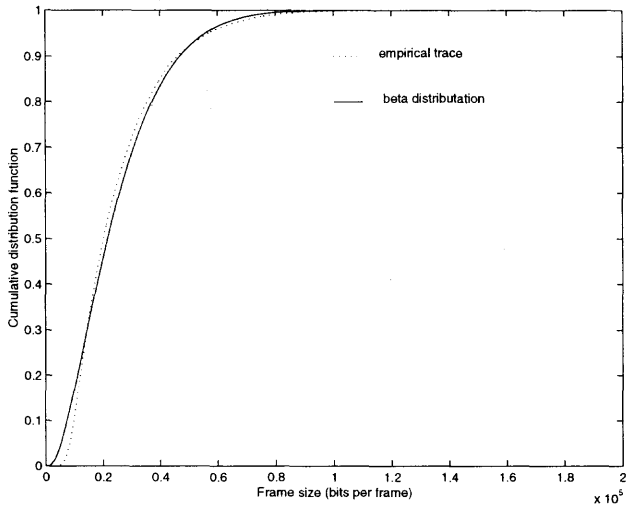

Figure 4.7 CDF of the active part and the corresponding Beta distribution

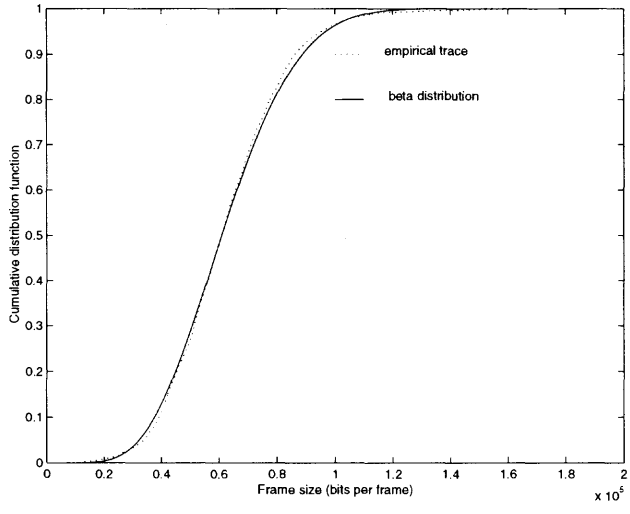

Figure 4.8 CDF of the most active part and corresponding Beta distribution

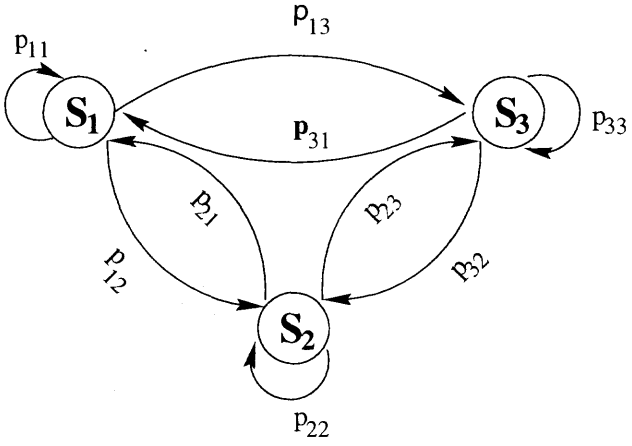

Figure 4.9 A Markov modulated self-similar process model for MPEG video

4.5 Modeling the MPEG Data

To model the whole data set, we need a process to govern the transition among the processes s_1, s_2, and s_3 obtained above. Markov chain is often used because of its simplicity.

Using Markov chain as the dominating process, our model for MPEG video traffic can be described by the state diagram shown in Fig 4.9, where state S_1, S_2, and S_3 correspond to the three respective self-similar processes. At state S_i, bit rates are generated by process s_i. The transition probability from S_i to S_j can be estimated from the empirical data as follows:

$$p_{ij} = \frac{N_{ij}}{N_i}, \tag{4.6}$$

where, N_i is the total number of times that the system goes through state S_i, N_{ij} is the number of times that the system make transition to state S_j from state S_i. For the *Star Wars* video, the following transition matrix

$$\hat{P} = \begin{bmatrix} 0.0002 & 0.9998 & 0 \\ 0.1174 & 0.5232 & 0.3594 \\ 0.0209 & 0.9791 & 0 \end{bmatrix}$$

is obtained. This matrix is useful for the synthesis of video traffic.

4.6 Video Traffic Synthesis

To synthesize video traffic using our model requires self-similar traffic generator. Some methods are available to generate approximate self-similar traffic. Two of the most commonly used methods are exactly self-similar fractional Gaussian noise (FGN) [44] and asymptotically self-similar fractional autoregressive integrated moving-average (F-ARIMA) process [44]. F-ARIMA can be used to match any kind of ACF. It takes a long time to generate the video traffic since F-ARIMA is an iterative process . The F-ARIMA process can be generated by the following algorithm [5, 38, 39]:

1. Generate X_0 from a Gaussian distribution $N(0, \nu_0)$. Set initial values $N_0 = 0, D_0 = 1$

2. For $k = 1, 2, \cdots, N-1$, calculate $\phi_{kj}, j = 1, 2, \cdots, k$ iteratively using the following formulae

$$N_k = r(k) - \sum_{j=1}^{k-1} \phi_{k-1,j} r(k-j) \tag{4.7}$$

$$D_k = D_{k-1} - N_{k-1}^2 / D_{k-1} \tag{4.8}$$

$$\phi_{kk} = N_k / D_k \tag{4.9}$$

$$\phi_{kj} = \phi_{k-1,j} - \phi_{kk}\phi_{k-1,k-j}, \quad j = 1, \cdots, k-1 \tag{4.10}$$

$$m_k = \sum_{j=1}^{k} \phi_{kj} X_{kj} \tag{4.11}$$

$$\nu_k = (1 - \phi_{kk}^2)\nu_{k-1} \tag{4.12}$$

Finally, each X_k is chosen from $N(m_k, \nu_k)$. In this way, we obtain a process X with ACF approximating to $r(k)$.

To generate a self-similar process approximately, autocorrelation function can be calculated in a recursive way as

$$r(0) = 1, r(k+1) = \frac{k+d}{k+1} r(k) \tag{4.13}$$

where $d = H - 0.5$.

ACFs of F-ARIMA and FGN generated traffic are less than $k^{-\beta}$ for small k. To compensate the under estimation of ACFs of a self-similar process, Equation (4.13) used to generate F-ARIMA traffic can be enlarged for small k. New self-similar traffic generators need to be devised so that more exact self-similar traffic can be generated.

Distribution of these data is Gaussian. For the data to be Beta distributed, the following mapping can be used

$$Y_k = F_\beta^{-1}\left(F_N(X_k)\right), k > 0 \tag{4.14}$$

where X_k is a self-similar Gaussian process, F_N is the cumulative probability of the normal distribution, and F_β^{-1} is the inverse cumulative probability function of the Beta model.

Video traffic can be synthesized by a combination of the three obtained self-similar processes via a Markov process, whose transition matrix was given in the last section (see Fig. 4.10 for a traffic example). In the empirical data trace, the size of I frame is often larger than the size of P frame and B frame, implying that a large frame is often followed by several small frames. It is shown in Fig. 4.10 that the traffic generated by our model can capture this kind of characteristic. A piece of the empirical traffic trace is shown in Fig. 4.11

4.7 Cell Loss Rate of Network

Cell Loss Rate (CLR) is an important queuing performance of an ATM network. To justify the queuing performance of our model, our synthetic traffic was used as the source traffic to an ATM switch with a limited buffer size. The performance is compared to the same system using empirical data as the source traffic. A single arrival process is assumed in our simulation, and its service rate is assumed to be

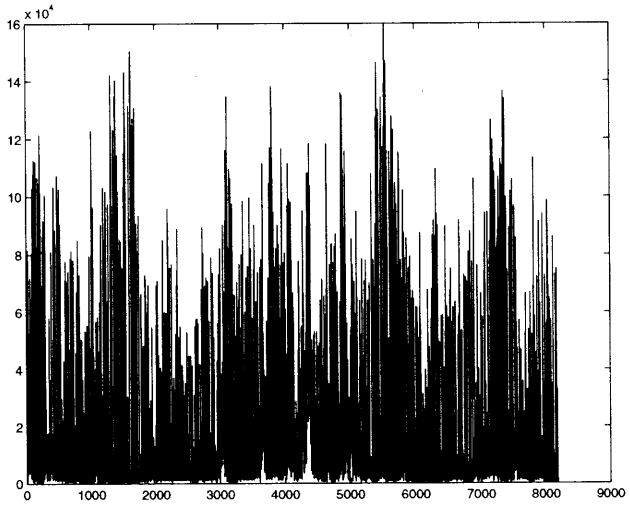

Figure 4.10 Traffic generated by model

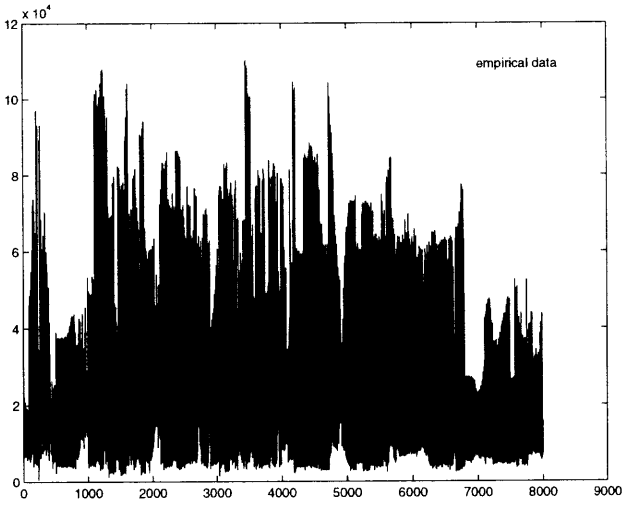

Figure 4.11 A piece of empirical traffic trace

Table 4.1 CLRs for different service rate and buffer size

Buffer size(cells)	4000cells/s		6000cells/s		9000cells/s	
	Trace	Model	Trace	Model	Trace	Model
20	2.24E-2	9.95E-2	2.09E-3	2.30E-2	1.50E-4	4.25E-4
40	1.24E-2	7.43E-2	1.36E-3	1.34E-2	8.07E-5	1.34E-4
60	6.81E-3	5.44E-2	9.72E-4	9.25E-3	7.19E-6	2.28E-5
100	2.30E-3	2.72E-2	4.57E-4	2.91E-3	0	0
200	3.55E-4	4.22E-3	1.00E-5	3.40E-5	0	0
400	6.14E-5	6.40E-4	0	0	0	0

constant. To simplify the simulation process, the time is sliced. Every slice is used to transmit one cell (48 bytes). We also assume that cells in a frame must arrive at the switch during the period of this frame. This corresponds to the case that no traffic shaping is applied. Cells are dropped when the switch buffer overflows.

Based on the switch model, performance at different service rates and buffer sizes is examined. Simulation results using empirical data and traffic model are shown in Table 4.1. The results show that the CLRs obtained using video trace and our proposed model are very close for both high and low service rates.

CHAPTER 5

MODELING VIDEO TRAFFIC USING STRUCTUALLY MODULATED SELF-SIMILAR PROCESSES

5.1 Introduction

Most models appeared in the literatures are confined to JPEG coded sequences, which are far from common use than MPEG coded sequences. Since characteristics of MPEG coded traffic are quite different from those of JPEG coded video traffic, models for JPEG traffic are not suitable for MPEG traffic. Therefore, new models need to be explored to model MPEG coded video sequences.

In [54], a unified approach was proposed to model both MPEG and JPEG coded video traffic. To model MPEG coded data, a background process (a process which will be transformed to three different processes) need to be searched. There is not a good way to search for the background processes, and therefore, the method can not be used to systematically model MPEG video traffic.

In [55], I, P, and B frames were modeled separately. I frames were described by three parts: scene length, average I frame size over a scene, and variations from the average frame size for the scene. P and B frames were modeled as $i.i.d$ processes. As we will show, I, P, and B frames are not $i.i.d$ but LRD processes. Since B and P frames occupy a very large portion of the whole sequence and B the size of frames is also rather large, we believe that the impact of B and P frames can not be ignored, and it is inappropriate to model P and B frames as $i.i.d$ processes.

Although the proposed Markov modulated self-similar processes model introduced in the last chapter tried to capture the LRD and SRD characteristics of video, the pattern of the ACF of MPEG coded video, such as the fluctuation of the ACF, however, can be matched better by exploiting the MPEG structure.

In this chapter, we propose a new model, structurally modulated, self-similar processes that are shown to be able to capture both SRD, LRD, and the fluctuation of

the ACF. Traffic data are decomposed into several parts according to the MPEG data structure, each modeled as a self-similar process rather accurately. These processes are then modulated structurally in a manner similar to how the frames are grouped into the GOP (Group of Pictures) pattern.

5.2 ACF of Empirical Data

As mentioned in last chapter, the ACF of frame size of MPEG coded *Star War* (shown again in Fig 5.1) fluctuates around three envelopes, reflecting the fact that, after the use of motion estimation and forward/backward prediction techniques, the dependence between frames is reduced. This characteristic should be taken into consideration in modeling MPEG coded video sequences.

It is clear that the ACF of MPEG coded video sequences varies with the pattern of GOP, that is, it has the same pattern as GOP, and therefore, we propose to decompose the sequence into I, P, B_1, $B_2, \cdots$, B_8 according to the GOP pattern, and model each part by a different self-similar process.

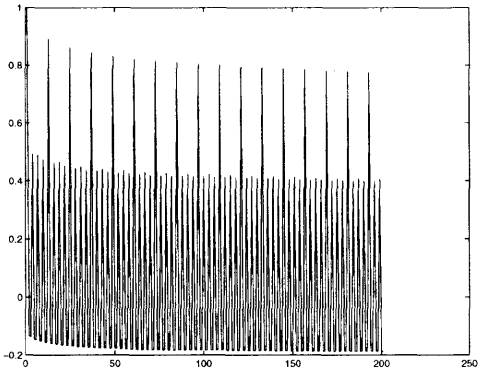

Figure 5.1 ACF of MPEG compressed video of *Star Wars*

5.3 Modeling MPEG Traffic

In order to model MPEG coded data, we decompose the MPEG traffic into 10 subsequences $X_I, X_P, X_{B_1}, X_{B_2}, \cdots$, and X_{B_8}. X_I consists of all I frames, X_P consists of

all P frames, the first B frames in all GOPs constitute X_{B_1} , the second B frames in all GOPs constitute X_{B_2}, and so on. We have used $k^{-\beta}$, $e^{-\beta k}$ and $e^{-\beta\sqrt{k}}$, corresponding to the ACFs of a self-similar process, a Markov process, and an $M/G/\infty$ input process, respectively, to approximate ACFs of these processes (see Fig. 5.2 to 5.11). The mean square errors are shown in Table 5.1. It is quite obvious that self-similar processes are better choices. We therefore use self-similar processes to model these data.

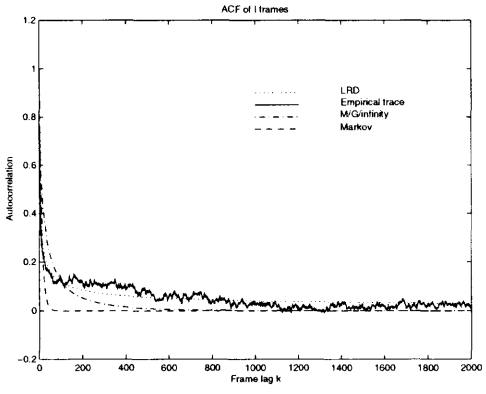

Figure 5.2 Approximation for ACF of I frames by : LRD, $M/G/\infty$, and Markov processes

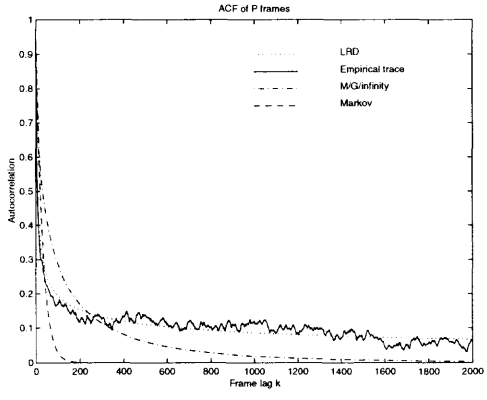

Figure 5.3 Approximation for ACF of P frames by : LRD, $M/G/\infty$, and Markov processes

Using least squares method, we obtain $\beta = 0.4663$, 0.3546, 0.4468, 0.4779, 0.4294, 0.4656, 0.4380, 0.4682, 0.4465, and 0.4606 for $X_I, X_P, X_{B_1}, X_{B_2}, \cdots$, and

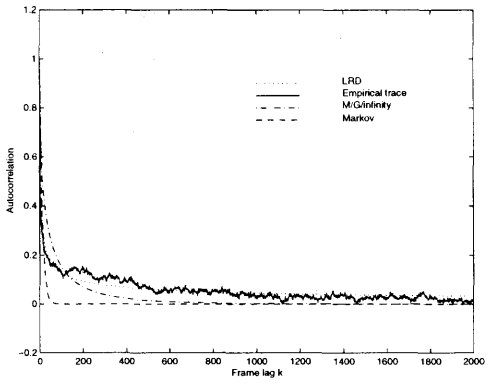

Figure 5.4 Approximation for ACF of B_1 frames by : LRD, $M/G/\infty$, and Markov processes

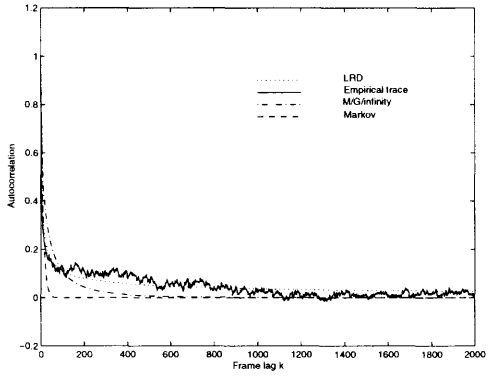

Figure 5.5 Approximation for ACF of B_2 frames by : LRD, $M/G/\infty$, and Markov processes

X_{B_8}, respectively. The corresponding Hurst parameters for these processes are $H =$ 0.7668, 0.8227, 0.7766, 0.7610, 0.7853, 0.7672, 0.7810, 0.7659, 0.7768, and 0.7697.

To model marginal distributions of these processes, the Beta distributions introduced in the last chapter are used. Using these formulae, $\hat{\eta} = 1.5237$, 1.5699, 1.4172, 1.3016, 1.6858, 1.6329, 1.7276, 1.4218, 4.0585, 1.5402, and $\hat{\gamma} = 12.7263$, 11.1939, 8.1089, 8.1604, 11.8499, 13.9278, 12.2180, 8.6536, 10.4233, 11.1768 are obtained for X_I, X_P, X_{B_1}, X_{B_2}, $\cdots$, and X_{B_8}, respectively.

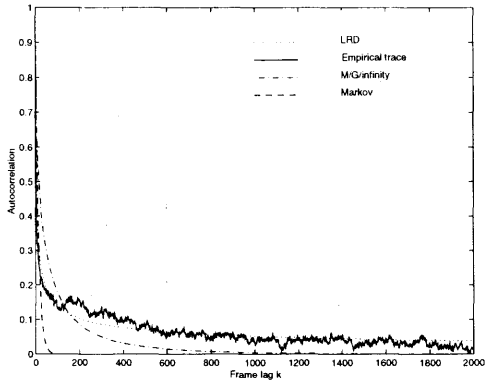

Figure 5.6 Approximation for ACF of B_3 frames by : LRD, $M/G/\infty$, and Markov processes

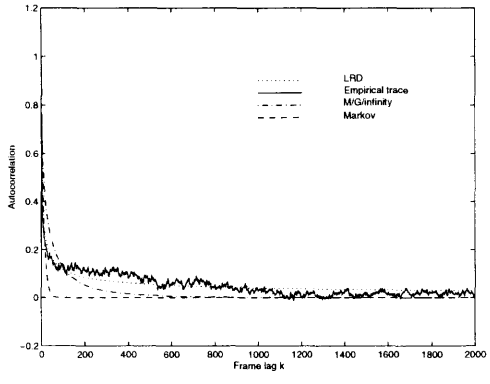

Figure 5.7 Approximation for ACF of B_4 frames by : LRD, $M/G/\infty$, and Markov processes

By combining X_I, X_P, X_{B_1}, X_{B_2},$\cdots$, and X_{B_8} in a manner similar to the GOP pattern, a model for MPEG coded traffic is obtained. This model can be used to generate traffic data.

5.4 Generating Traffic Data

To generate video traffic, we need to generate self-similar traffic first. F-MARIA is used again to produce self-similar Gaussian random processes. For data to be Beta

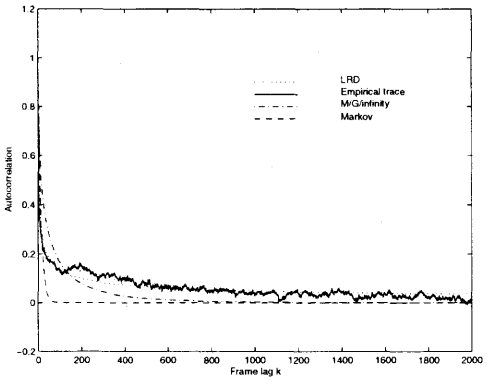

Figure 5.8 Approximation for ACF of B_5 frames by : LRD, $M/G/\infty$, and Markov processes

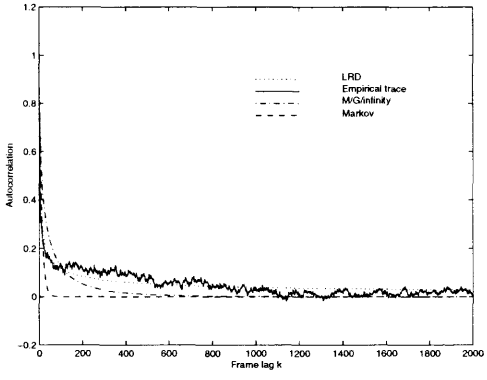

Figure 5.9 Approximation for ACF of B_6 frames by : LRD, $M/G/\infty$, and Markov processes

distributed, the following mapping is used once again:

$$Y_k = F_\beta^{-1}\left(F_N(X_k)\right), k > 0 \tag{5.1}$$

where X_k is a self-similar Gaussian process, F_N is the cumulative probability of the normal distribution, and F_β^{-1} is the inverse cumulative probability function of the Beta model.

Simply following the GOP pattern, we can combine these self-similar processes to generate traffic data. A piece of traffic data generated is shown in Fig. 5.12. The corresponding ACF is shown in Fig. 5.13. Fig. 5.14 shows the ACF of traffic data

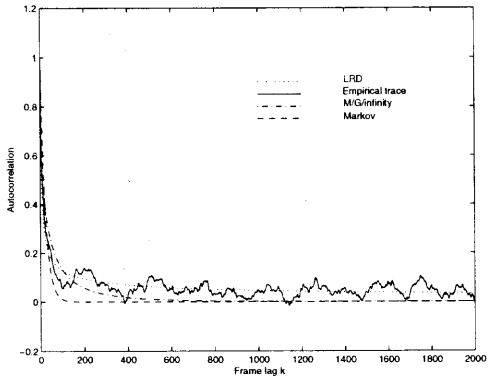

Figure 5.10 Approximation for ACF of B_7 frames by : LRD, $M/G/\infty$, and Markov processes

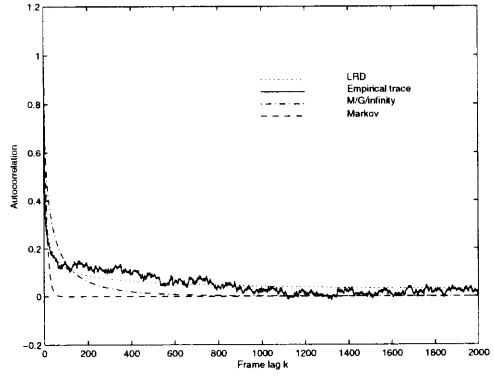

Figure 5.11 Approximation for ACF of B_8 frames by : LRD, $M/G/\infty$, and Markov processes

generated by our model (large scale version). It is clear that LRD , SRD and the fluctuation can be captured by this model.

5.5 Cell Loss Rate

Cell Loss Rate (CLR) is an important queuing performance of an ATM network. To justify the queuing performance of our model, our synthetic traffic was used as the source traffic to an ATM switch with a limited buffer size. The performance is compared to the same system using empirical data as the source traffic. A single

Table 5.1 Least square errors obtained by self-similar process, Markov and $M/G/\infty$ methods

	I	P	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8
LRD	0.46	0.35	0.45	0.48	0.43	0.47	0.44	0.47	0.45	0.46
$M/G/\infty$	5.13	12.3	5.40	4.80	6.23	5.19	5.83	5.05	5.03	5.34
Markov	7.98	21.0	9.45	7.30	11.1	8.12	10.2	7.95	7.22	8.75

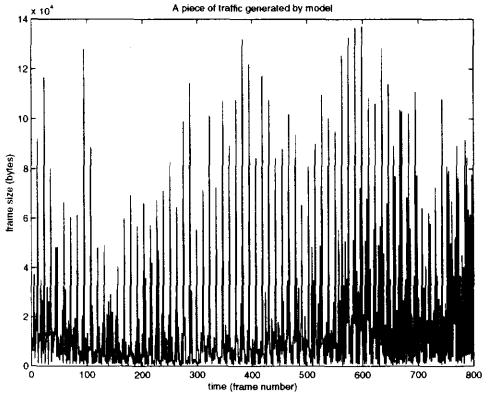

Figure 5.12 Traffic data generated by our model

arrival process is assumed in our simulation, and its service rate is assumed to be constant. To simplify the simulation process, the time is sliced. Every slice is used to transmit one cell (48 bytes). We also assume that cells in a frame must arrive at the switch during the period of this frame. This corresponds to the case that no traffic shaping is applied. Cells are dropped when the switch buffer overflows.

Based on the switch model, performance at different service rates and buffer sizes is examined. The simulation results using empirical data and traffic model are shown in Table 5.2. As shown, CLR obtained based on our proposed model is very close to that based on the empirical data, and the results are better than the results obtained by Markov modulated self-similar process models.

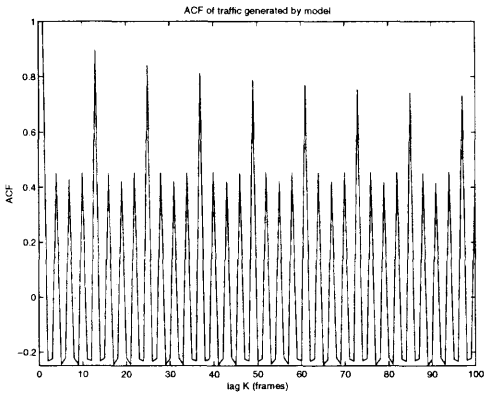

Figure 5.13 ACF of traffic data generated by our model

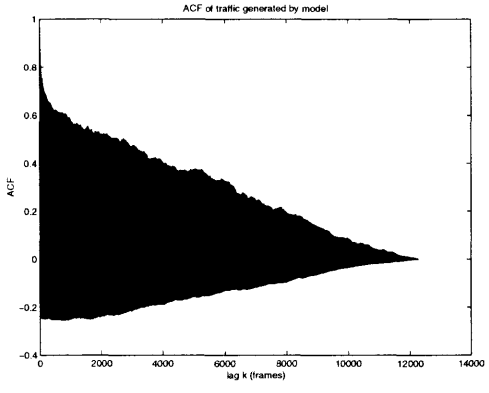

Figure 5.14 ACF of traffic data generated by our model (large scale version)

Table 5.2 CLRs for different service rate and buffer size

Buffer size(cells)	4000cells/s		6000cells/s		9000cells/s	
	Trace	Model	Trace	Model	Trace	Model
20	2.24E-2	6.57E-2	2.09E-3	1.15E-2	1.50E-4	1.63E-4
40	1.24E-2	4.61E-2	1.36E-3	6.75E-3	8.07E-5	4.94E-5
60	6.81E-3	3.13E-2	9.72E-4	3.69E-3	7.19E-6	1.02E-5
100	2.30E-3	1.28E-2	4.57E-4	1.01E-3	0	0
200	3.55E-4	5.61E-4	1.00E-5	3.60E-6	0	0
400	6.14E-5	0	0	0	0	0

CHAPTER 6

A SIMPLER VIDEO TRAFFIC MODEL FOR MPEG VIDEO

6.1 Introduction

In the previous chapter , we proposed a structurally modulated self-similar process, which depends on the decomposition of MPEG sequences into I, P, $B_1, B_2, \ldots$ according to the GOP structure. Through careful analysis we found that the first frame in the data file of *Star wars* provided by Dr. Garrett was not the I frame but the second B frame after the I frame. After the removal of the first several B and P frames, we obtain a file in which the first frame corresponds to the I frame. After editing the data file, we decompose the data into I, P, and B frames and calculate the ACF of I, P, and B frames. We find that the whole sequence of B frames can be modeled by a self-similar process. That means that the earlier model we proposed, although works well, can be further simplified.

In this chapter, we propose a simpler model for MPEG coded video, in which the original sequences are decomposed into three parts, each of which can be modeled by a self-similar process. Compared with the model introduced in the previous chapter, the model is rather simple.

6.2 Modeling MPEG Traffic

In order to model MPEG coded data, we decompose the MPEG traffic into three sub-processes, X_I, X_P, X_B. X_I consists of all I frames, X_P consists of all P frames, and X_B consists of all B frames. We have used $k^{-\beta}$, $e^{-\beta k}$, and $e^{-\beta\sqrt{k}}$, corresponding to the ACFs of a self-similar process, a Markov process, and an $M/G/\infty$ input process, respectively, to approximate ACFs of these processes. The ACF of each sub-process and its approximation by Markov process, $M/G/\infty$ process, and self-similar process are shown in Fig. 6.1 to 6.3. The sums of squares of errors obtained by the three kinds of methods are tabulated in Table 6.1. It is quite obvious that self-similar

Table 6.1 Least square errors obtained by self-similar process, Markov and $M/G/\infty$ methods

	I	P	B
LRD	1.5820	0.6630	0.5987
$M/G/\infty$	5.0527	12.8669	13.7523
Markov	7.2517	25.4433	32.0705

processes are better choices. We therefore use self-similar processes to model these data.

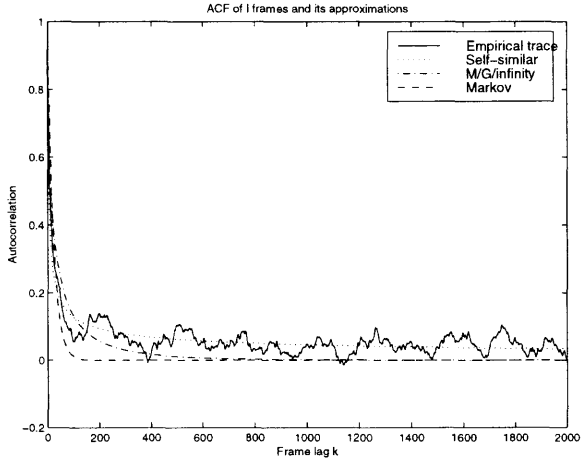

Figure 6.1 Approximation for ACF of I frames by : LRD, M/G/∞, and Markov processes.

Using the least squares method, $\beta = 0.4662, 0.3404, 0.3040$ are derived for X_I, X_P, X_B, respectively. The corresponding Hurst parameters for these processes are $H = 0.7669, 0.8296, 0.8480$, respectively.

6.3 Matching CDF of I, P, and B Frames

To model marginal distributions of these processes, we again use Beta distributions which have the following form of probability density function:

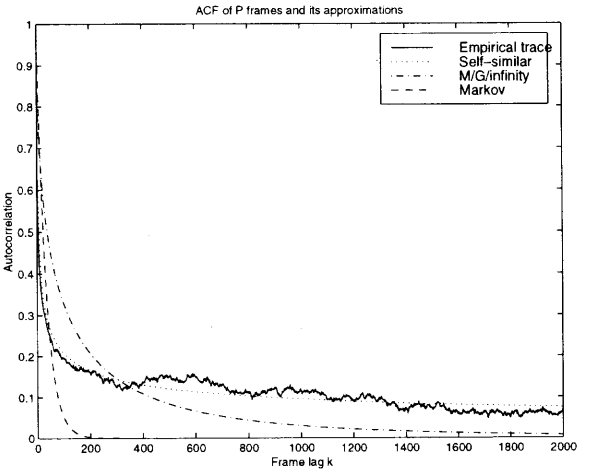

Figure 6.2 Approximation for ACF of P frames by : LRD, M/G/∞, and Markov processes.

$$f(x;\gamma,\eta,\mu_0,\mu_1) = \begin{cases} \frac{1}{\mu_1-\mu_0}\frac{\Gamma(\gamma+\eta)}{\Gamma(\gamma)\Gamma(\eta)}(\frac{x-\mu_0}{\mu_1-\mu_0})^{\gamma-1}(1-\frac{x-\mu_0}{\mu_1-\mu_0})^{\eta-1} \\ \qquad \mu_0 \le x \le \mu_1, 0 < \gamma, 0 < \eta \\ 0 \qquad \text{otherwise,} \end{cases} \tag{6.1}$$

where γ and η are shape parameters, and $[\mu_0, \mu_1]$ is the domain where the distribution is defined. They can be estimated by the following formulae [53]:

$$\hat{\eta} = \frac{1-\bar{x}}{s^2}[\bar{x}(1-\bar{x}) - s^2] \tag{6.2}$$

$$\hat{\gamma} = \frac{\bar{x}\hat{\eta}}{1-\bar{x}} \tag{6.3}$$

where

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i, \tag{6.4}$$

$$s^2 = \frac{N\sum_{i=1}^{N} x_i^2 - \left(\sum_{i=1}^{N} x_i\right)^2}{N(N-1)}, \tag{6.5}$$

and N is the number of data in the data set. Using these formulae, $\hat{\gamma}$ = 4.0605, 1.6605, 1.6431, and $\hat{\eta}$ = 10.4273, 12.0277, 14.0742 are derived for X_I, X_P, X_B, respectively. The CDF of the three parts and their corresponding approximation by Beta distribution are shown from Fig. 6.4 to 6.6 .

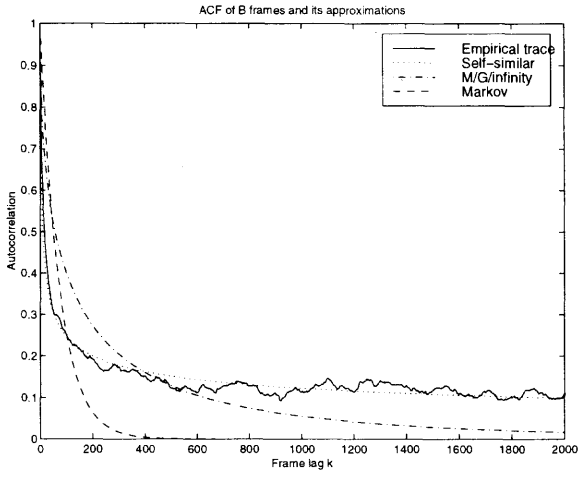

Figure 6.3 Approximation for ACF of B frames by : LRD, M/G/∞, and Markov processes.

6.4 Synthesis of Video Traffic and Simulation Results

By combining X_I, X_P, X_B in a manner similar to the GOP pattern, a model for MPEG coded traffic is obtained. This model can be used to generate traffic data.

Fig. 6.7 shows a trace of the empirical video traffic, and the trace generated by our model is shown in Fig. 6.8. It is clear that the traffic generated by our model is very similar to the empirical trace itself. For the real MPEG trace, the frame pattern is IBBPBBPBBPBBIBB$\cdots$. Generally speaking, in the same GOP, I frames are larger than P frames, while P frames are larger than B frames. This pattern is demonstrated by the traffic generated by our proposed model, and therefore, our model can capture this feature accurately. Since traffic is random, the appropriateness of a traffic model should be judged by its statistical properties rather than the mere similarity between these two figures. This can be demonstrated by the ACF of the generated traffic shown in Fig. 6.9.

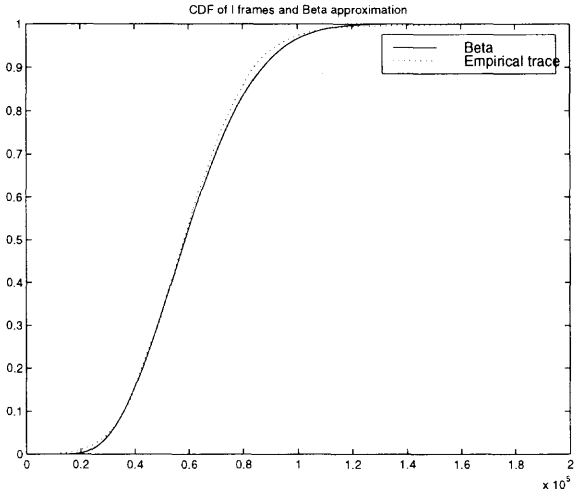

Figure 6.4 CDF of I frames and its approximation by Beta distribution.

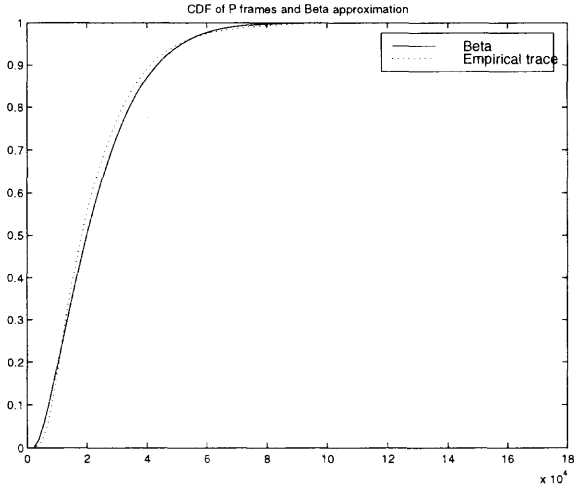

Figure 6.5 CDF of P frames and its approximation by Beta distribution.

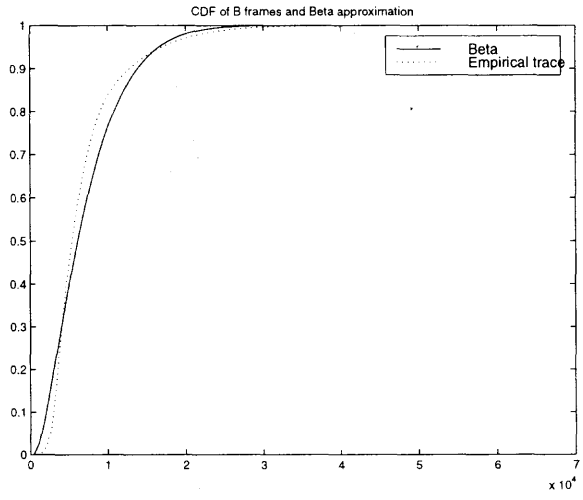

Figure 6.6 CDF of I frames and its approximation by Beta distribution.

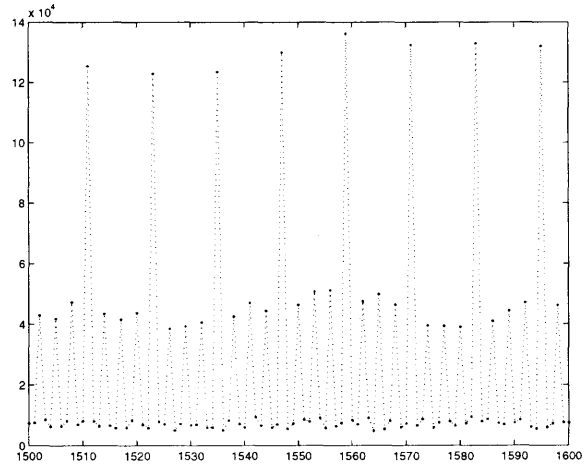

Figure 6.7 A trace of the empirical traffic data.

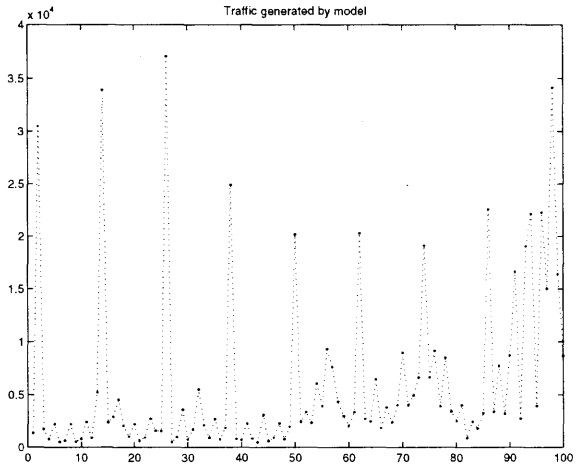

Figure 6.8 Traffic data generated by our model

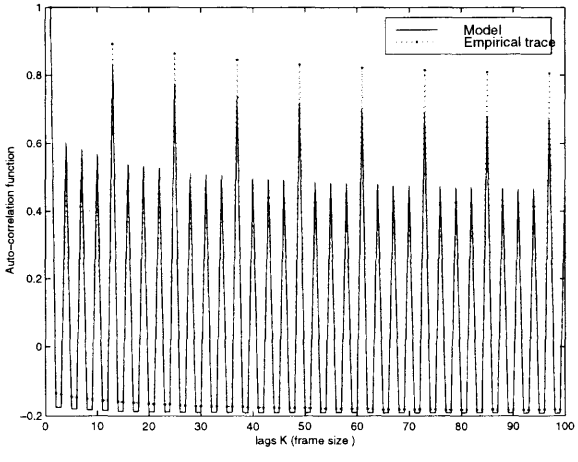

Figure 6.9 ACF of traffic data generated by our model

CHAPTER 7

DYNAMIC BANDWIDTH ALLOCATION FOR PRERECORDED VIDEO DELIVERY

7.1 Introduction

It is well known that ATM network can be used to transmit many kinds of traffic with guaranteed quality of service. Video traffic has been predicted to be the major component of future broadband network traffic, and therefore, an efficient video traffic transmission mechanism for ATM network is important to network operators. Bit stream of video traffic produced by a Codec, however, is naturally VBR (variable bit rate). For example, the ratio of the peak-to-average bit rate may be as large as 12 for *Star wars*. For such kind of traffic, it is very difficult to meet the QoS parameters (such as delay and cell loss ratio) while keeping high network utilization.

To improve network utilization, some bit rate control [56] and traffic shaping algorithms [57],[58] were developed. Owing to the stringent delay requirement, traffic shaping techniques have limited effect on network utilization. Bit rate control methods, on the other hand, can be used to improve network utilization, but at the cost of varying video quality.

Another choice to improve network utilization is to use dynamic bandwidth allocation algorithms. This kind of algorithms allows users to dynamically reserve or adjust network resources. When the reserved resource is not enough for the user to transmit its traffic, a renegotiation can be initiated to ask for more resource. If the reserved resource is more than enough, some bandwidth can be released. In such a way, network utilization can be improved significantly.

One major class of dynamic resource management algorithms is based on parameter measurements. Several measurement based dynamic bandwidth allocation algorithms, which initiate their renegotiation processes based on the

actual measurement of the cell loss ratio (CLR) or user parameters (UPs), have been proposed [59, 60]. To manage resources, RM cells may be required.

In [60], the CLR was calculated up to the current period, and the service rate for the next period was adjusted based on the current CLR. Owing to the difficulty for assessing the CLR on line and the indirect relationship of the current CLR and UP with the future bandwidth requirements, these approaches are not effective enough to enhance QoS and improve network utilization.

Another major class of algorithms is based on prediction techniques. It is mainly used for real-time video transmission. For these algorithms, the user parameters or bit rate were predicted based on the available information, and resources were allocated based on the predicted results. It is important to predict these parameter accurately so that network resources can be used efficiently.

In [61], VBR service was proposed to be used, and therefore the QoS can be guaranteed. The leaky bucket parameters (peak rate, sustain rate) were adjusted based on UPs calculated for every GOP. UPs can be inherently inaccurate because they were calculated from previous GOPs. To reduce the buffer size, the source quantization step was adjusted on line. The drawback of this algorithm is that the user parameters (peak rate, sustain rate, and bursty length) need to be renegotiated for each GOP, which is a big burden to network management.

Adas [62] proposed to use adaptive linear prediction to support dynamic bandwidth reallocation. It is claimed that by predicting the average bit rate of the next GOP and allocating bandwidth based on the predicted results, the network utilization can be improved by a factor of 1.9-3.0 [62]. Since the bit rate variation is very high, it is very difficult to predict the average bit rate in the next GOP accurately, and the prediction error can be very large. The required number of renegotiation is also very large.

A new algorithm is proposed here, which is applicable to pre-recorded video, in which a renegotiation process is initiated only when a scene change occurs. It is well know that bit rate changes dramatically only when scene change occurs, and thus, renegotiation is necessary only at that time. Intuitively, the renegotiation frequency should be lower compared with that of Reininger's and Adas' algorithms [62, 61]. The CLR can be guaranteed to be zero if the maximum bandwidth for every scene can be satisfied.

7.2 Scene Change Identification

A representative portion of an empirical video trace is plotted in Fig. 7.1. The vertical axis represents the number of bits per frame, and the horizontal axis represents the corresponding frames. It is clear that the data appears to be composed of stationary segments. Average bit rate of every segment changes abruptly from one segment to another. Visually, these abrupt transitions coincide with scene changes [8].

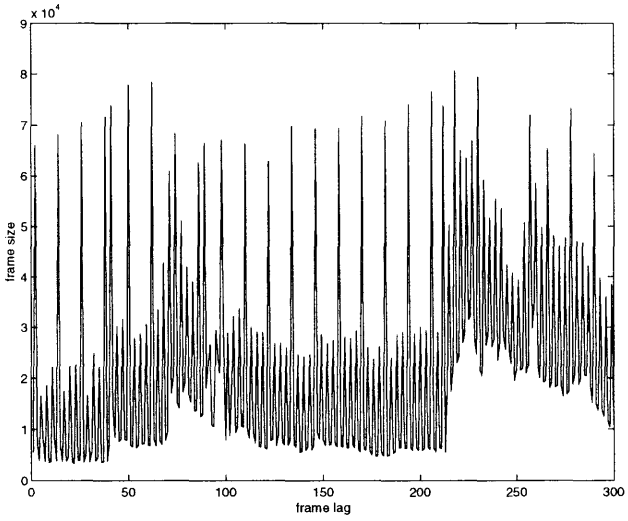

Figure 7.1 Video traffic of *Star Wars*

To detect scene changes in the empirical trace, the algorithm introduced in [8] is adopted. A scene change is declared if the change of the number of bits between

successive frames exceeds certain threshold in a sustained manner. Denotes X_n as the number of bits in the *nth* frame, and J_{min} the threshold. Define J_n as follows:

$$J_n = \begin{cases} 1 & |X_n - X_{n-1}| > J_{min}, \\ 0 & otherwise. \end{cases} \tag{7.1}$$

The indicator function for the *nth* frame is then given by

$$S_n = \begin{cases} 1 & if\ J_n = 1, J_{n-1} \neq 1, J_{n-2} \neq 1, \cdots, J_{n-L_{min}} \neq 1 \\ 0 & otherwise, \end{cases} \tag{7.2}$$

where L_{min} is the minimal scene length in frames. The frame whose indicator function is one corresponds to a scene change. It was experimently found that for video *star wars*, $J_{min} = 5000$, and $L_{min} = 2$.

7.3 Dynamic Bandwidth Allocation Based on Scene Changes

For MPEG coded video movies, the ratio of peak rate to average rate is very large. If CBR service is used to transmit MPEG video traffic, it is difficult to guarantee QoS while keeping bandwidth utilization high. Owing to the fact that network utilization must be low if the bandwidth is allocated to a value equal to the peak rate, and the delay or the CLR will be high if average bit rate is allocated.

It is intuitive that the bit rate change during a scene is not large, and hence, low CLR, small delay, and high network utilization can be expected. Based on this premise, the following dynamic bandwidth allocation algorithm is proposed:

1. Identify the scene change.
2. If scene change occurs, determine the maximum bandwidth needed (which can be found in advance for stored video).
3. Initiate a negotiation process for the maximum bandwidth for this scene.
4. Go to Step 1.

Note that the maximum bandwidth for every scene should be determined and stored beforehand. During the retrieval process, if a scene change is detected, the maximum bandwidth for the new scene can be read and the bandwidth can be allocated for this scene.

Let C be the maximum number of bits that can be transmitted by the channel, and M be the total number of bits transmitted. The bandwidth utilization factor is defined as:

$$\rho = \frac{M}{C}. \tag{7.3}$$

For the *star wars* video, if peak bit rate is used in CBR service, the utilization factor is 0.0842. Using the proposed method, the utilization factor is 0.1962 (an improvement by a factor of 2.3). Compared with Reininger's method [61], the renegotiation frequency can be reduced by a factor of 7 if our proposed method is used. Compared with Adas's method [62], the number of renegotiation is reduced by a factor of 7 while the CLR is zero.

7.4 Improving Bandwidth Utilization

The algorithm introduced above achieves zero CLR at the expense of a lower bandwidth utilization. The bandwidth utilization can be improved by using a buffer. To improve network utilization, the following procedure is proposed.

1. Identify the scene change.
2. If scene change occurs, determine the mean bandwidth X_i of this scene.
3. Initiate a negotiation process to acquire bandwidth βX_i for this scene, where $\beta > 1$ is a constant.
4. Go to step 1.

The average bandwidth for every scene should be determined off-time. For stored video traffic, it is very easy to find the average bit rate of each scene.

Suppose that the movie has N scenes. X_i is the mean frame size of the *ith* scene, and L_i is the number of frames in the *ith* scene. Then,

$$M = \sum_{i=1}^{N} L_i X_i \tag{7.4}$$

$$C = \sum_{i=1}^{N} \beta L_i X_i \tag{7.5}$$

and

$$\rho = \frac{M}{C} = \frac{\sum_{i=1}^{N} L_i X_i}{\sum_{i=1}^{N} \beta L_i B_i} = \frac{1}{\beta}, \tag{7.6}$$

where M is the total number of bits of the movie, and C is the maximum number of bits that the source can transmit. It is clear that the bandwidth utilization is the reciprocal of β.

Video trace *star wars* is used in our simulations. The data file for the trace consists of 174,136 integers, whose values are frame sizes (bits per frame). The largest frame has 185267 bits, and the smallest one has 476 bits. This is a good representative movie sequence for benchmark comparison. It contains materials ranging from low complexity/motion scenes to those with high and very high actions.

In Fig.7.2, the relationship between CLR and buffer size is illustrated for different values of β. The value of β varies from 2 to 9 with a step size of 0.5. When β is larger than 9, the CLR becomes zero for any buffer size. The curve at the top corresponds to $\beta = 2$. Compared with CBR service, CLR can be reduced by an order of 2-3 using our proposed techniques with fixed buffer size and bandwidth utilization. Likewise, with fixed CLR and bandwidth utilization, the buffer size can be reduced by a factor of 2-4.

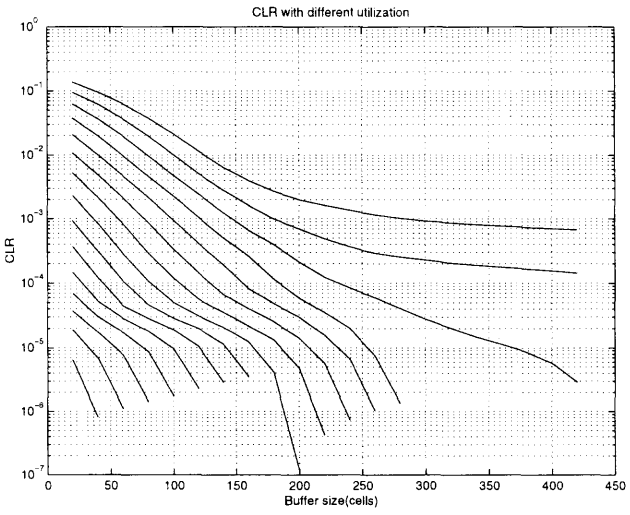

Figure 7.2 CLR versus buffer size under different network utilization factors (β changes from 2 to 9 with a step size of 0.5 from top to bottom)

7.5 Minimizing Renegotiations

In general, the network should renegotiate its bandwidth as frequently as possible to achieve maximum network utilization. Too many renegotiations are, however, a big burden to network management, and therefore, it is desirable to reduce the number of renegotiations as small as possible. Most of the scenes are short. The mean bit rates among the consecutive scenes vary only a little, and thus we can further reduce the number of renegotiations. Two methods are proposed. If a new renegotiation process is triggered only when the mean bit rate of scenes changes drastically, the renegotiation frequency can be reduced while bandwidth utilization and CLR are kept almost unchanged. The frequency can also be reduced by combining consecutive scenes whose lengths are smaller than L_T, thus resulting in longer scene length. In this case, only the scene identification algorithm defined in Section 7.2 needs to be modified as:

$$S_n = \begin{cases} 1 & if\ J_n = 1, J_{n-1} \neq 1, J_{n-2} \neq 1, \cdots, J_{n-L_{min}} \neq 1, L(n) > L_T \\ 0 & otherwise, \end{cases} \tag{7.7}$$

where $L(n)$ is the length of the nth scene, L_T is the threshold used to the combined scene, that is, the consecutive scenes which are shorter than L_T are combined together. To reduce the number of renegoations, we can also keep the definition of S_n unchanged, but let L_{min} to be L_T instead of 2, which was used in [8].

The second negoation reduction method is employed in our simulation. When the values of L_T used in our simulation are 12, 20, 50, 100, 200, 300, 400, 500, 1000, the number of renegotiations after scene combinations become 1730, 1520, 888, 620, 422, 273, 228, 133, respectively. CLR versus buffer size for different β are shown in Fig. 7.3 to Fig. 7.5, corresponding to $L_T = 100$, 300, and 1000, respectively. In these figures, different curve corresponds to different β, which varies from 2 to 10 from top to the bottom, with a step size of 0.5. The 3-D plots shown in Fig. 7.6- 7.8 correspond to CLR versus buffer size and L_T for $\beta = 4, 6, 8$. The performance is kept almost unchanged even for the case of $L_T = 300$, in which case, the number of renegotiations is 273, which is almost 11 times smaller than that needed using the algorithm introduced in Section 7.3. Only when L_T is larger than 300, the performance begins to degrade gracefully for the case of small β. For such a long movie, 273 renegotiations are not a heavy burden to network management.

Simulations also demonstrate that, if the other two performance parameters are fixed, then bandwidth utilization can be improved by a factor of 2 to 5, the buffer size can be reduced by a factor of 2 to 4, and the CLR can be reduced by an order of 2 to 3. Through scene combination techniques, the frequency of renegotiation can be reduced significantly, i.e, the renegotiation cost is kept small. The proposed algorithms can be efficiently used for video delivering.

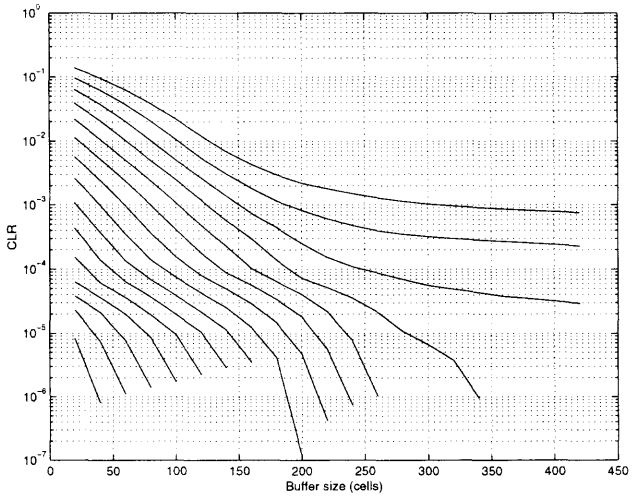

Figure 7.3 CLR versus buffer size when $L_T = 100$. β changes from 2 to 9 with a step size of 0.5 from top to bottom

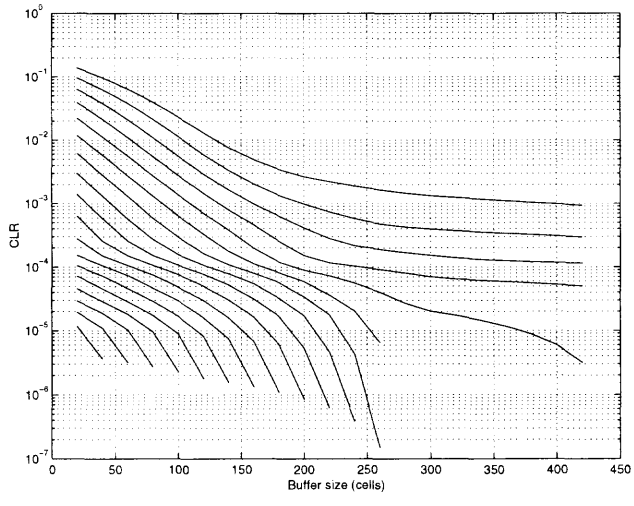

Figure 7.4 CLR versus buffer size when $L_T = 300$. β changes from 2 to 9 with a step size of 0.5 from top to bottom

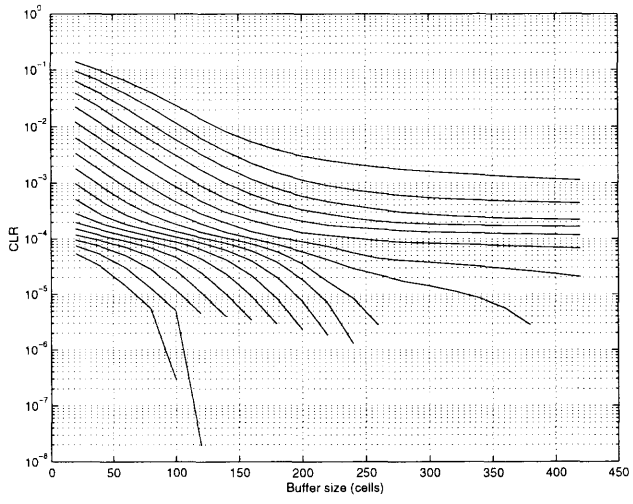

Figure 7.5 CLR versus buffer size when $L_T = 1000$. β changes from 2 to 9 with a step size of 0.5 from top to bottom

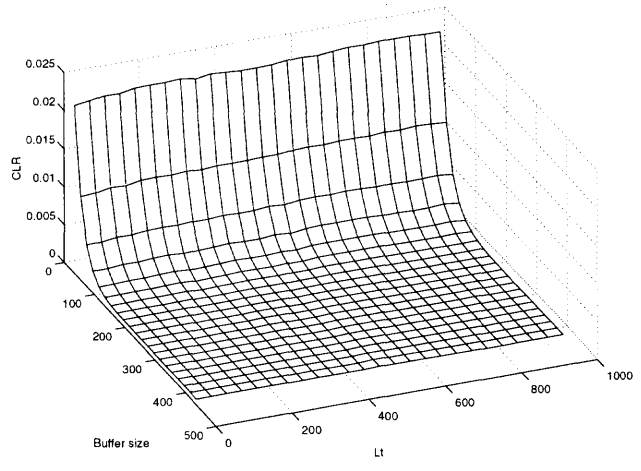

Figure 7.6 CLR versus buffer size and L_T, $\rho = 4$

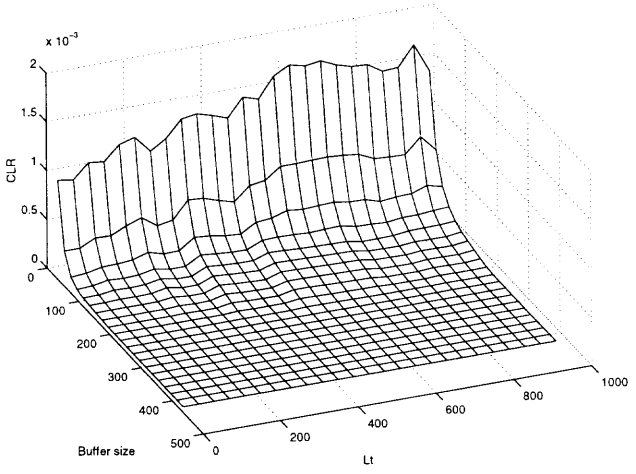

Figure 7.7 CLR versus buffer size and $L_T, \rho = 6$

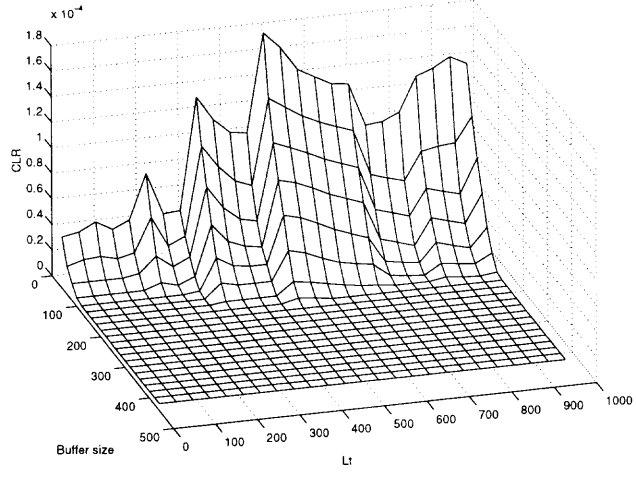

Figure 7.8 CLR versus buffer size and $L_T, \rho = 8$

CHAPTER 8

DYNAMIC BANDWIDTH ALLOCATION FOR REAL-TIME VIDEO DELIVERY

8.1 Introduction

The algorithms introduced in the last chapter, though can be used to deliver real-time video with some modification, the bandwidth utilization, however, is not expected to be high due to the fact that the frame size may change drastically in real-time video delivery even if there is no scene change.

In this chapter, algorithms which can be used to improve bandwidth utilization for non-interactive real-time video transmission are proposed. If the renegotiation is triggered by video sources, the prediction is not necessary because one or two frames can be delayed for non-interactive real-time video delivery. The algorithms introduced in this chapter require a delay of one to two frames.

8.2 Dynamic Bandwidth Allocation for Real Time MPEG Video Transmission

As pointed out earlier, if the renegotiation is initiated by the video source (or Codec), the bandwidth allocation may be implemented more easily and the allocation may be more efficient because the frame size is known before transmission.

Suppose the current transmission rate is R bits per second, the size of the frame that will be transmitted is S bits, and the frame refresh rate is N frame per second. The bandwidth reallocation can be carried out in the following way:

- if $|R - N \times S| < \delta$, transmission rate S is kept unchanged,
- $R = S \times N$, otherwise,

where δ is a threshold.

The value of δ affects the bandwidth utilization, the required buffer size, renegotiation frequency, and cell loss ratio. In order to increase bandwidth utilization and

Table 8.1 The number of renegotiations and bandwidth utilization

δ	1000	2000	3000	4000
FREQ.	71007	67229	63557	60197
Util.	0.9992	0.9988	0.9986	0.9982

decrease CLR, the value of δ should be small; on the other hand, to achieve a small number of renegoations, the value of δ should be large enough. To justify the effect of δ, an ATM switch with a limited buffer size is simulated for different values of δ. Simulation results are shown in Fig. 8.1.

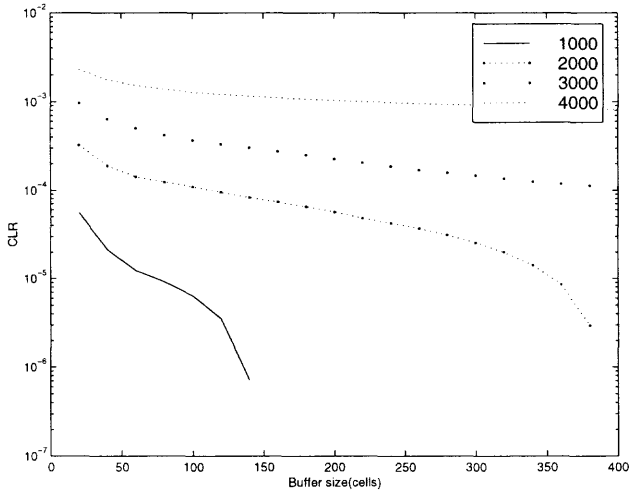

Figure 8.1 CLR versus buffer size for dynamic bandwidth allocation

From the simulation results we can see that to achieve satisfactory CLR, the value of δ should be very small, implying that a large number of renegotiations is needed. From the figure, when the buffer size is large enough, the CLR decreases rapidly, implying that buffering is effective in reducing CLR. This is difficult to achieve by CBR service. The actual bandwidth allocation for $\delta = 1000$ is shown in Fig. 8.2.

The renegotiation frequency and bandwidth utilization for different thresholds are shown in Table 8.1

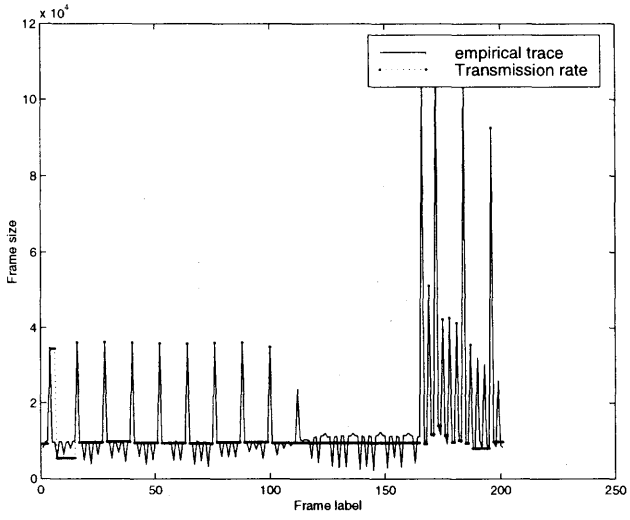

Figure 8.2 Actually allocated bandwidth

8.3 Reducing the Number of Renegotiations

Although the bandwidth utilization through the dynamic bandwidth allocation discussed above is high, the renegotiation is still a big burden to network management. To reduce the renegotiation frequency while keeping the bandwidth utilization reasonably high, an algorithm based on I frames is introduced in the this section.

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 I frame size changes significantly, P and B frame size also change significantly, implying that the increase or decrease of I frame size often indicates the increase or decrease of P and B frame sizes, and therefore, we can base on I frames to allocate bandwidth to improve QoS and network utilization.

To allocate bandwidth, the algorithm requires to hold on I frames to determine the size of I frames. When I frame size changes significantly, a renegotiation can be triggered to ask for reallocation of bandwidth.

Suppose the I frame of the *kth* GOP is just ready to be transmitted. Let I_k be the size of the 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 need to be transmitted per second.

Since only I frames need to be checked, the negotiation frequency can be reduced significantly. Since the size of I frames is the largest in a GOP most of the time, the bandwidth allocated is very close to the largest one needed for a period, and therefore the CLR can be kept small. The negotiation frequency and bandwidth utilization are tabulated in Table 8.2 while the CLR for different values of δ are shown in Fig. 8.3. The results demonstrate that the renegotiation frequency can be reduced significantly.

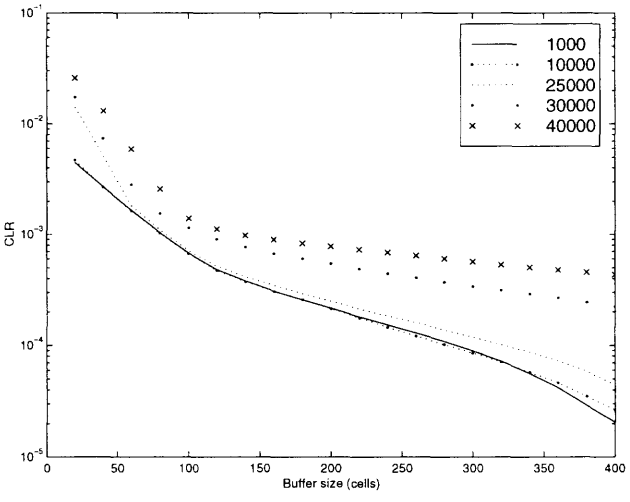

Figure 8.3 CLR versus buffer size for different value of δ

The actual bandwidth allocated when $\delta = 3000$ is shown in Fig. 8.4. For $\delta = 1000$, 10000, and 25000, the CLR almost remains unchanged even when δ changes significantly, implying that the number of renegoations can be reduced significantly without degrading the CLR performance.

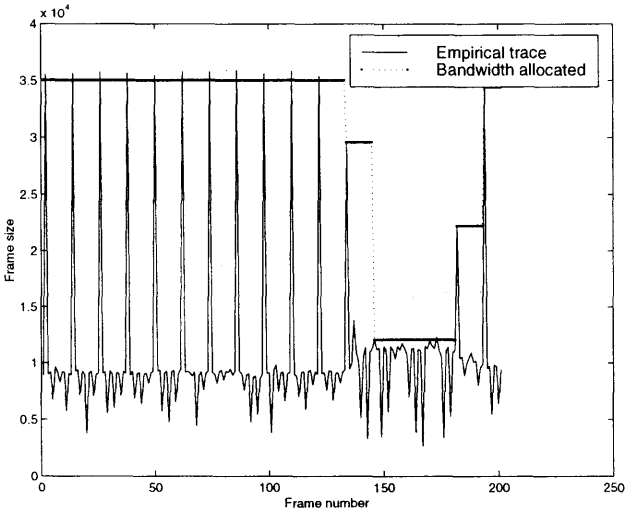

Figure 8.4 Actually allocated bandwidth when δ=5000

Table 8.2 The number of renegotiations and bandwidth utilization

δ	1000	10000	15000	20000
FREQ.	9208	2479	1694	1211
Util.	0.2584	0.2583	0.2588	0.2589

8.4 Increasing Bandwidth Utilization

As mentioned earlier, most of the time, I frames have the largest frame size in the respective GOPs, i.e., that most of the time the bandwidth is not used efficiently. From the analysis of the empirical trace, the difference between B frames for each GOP is not large. We can use this characteristic to increase bandwidth utilization.

Again suppose that I_k is the size of the I frame of the kth GOP, B_k is the size of the B frame immediately following the I frame in the kth GOP, and R is the transmission rate for the previous GOP, and δ is a threshold. Assuming these two frames are ready for transmission, then the dynamic bandwidth allocation algorithm can be altered as follows:

- if $|B_k + \alpha(I_k - B_k) - R/N| < \delta$, the transmission rate remains unchanged.

Table 8.3 The number of renegotiations and bandwidth utilization

δ	1000	7000	11000	19000
FREQ.	8113	2482	1611	763
Util.	0.3511	0.3508	0.3503	0.3515

- otherwise $R = [B_k + \alpha(I_k - B_k)] \times N$,

where δ is still the threshold, and α is a parameter which can be used to adjust the trade off between CLR and bandwidth utilization.

In general, the value of α should be less than one in order to have high bandwidth utilization. The value of α, however, can be larger than one if very good CLR performance is needed. The CLR performance for different α and δ are shown In Fig. 8.5 and Fig. 8.6, . The renegotiation frequency and bandwidth utilization for $\delta = 0.7$ are shown in Table 8.3. The actual bandwidth allocated for the case $\delta = 9000$ and $\alpha = 0.7$ is shown in Fig. 8.7

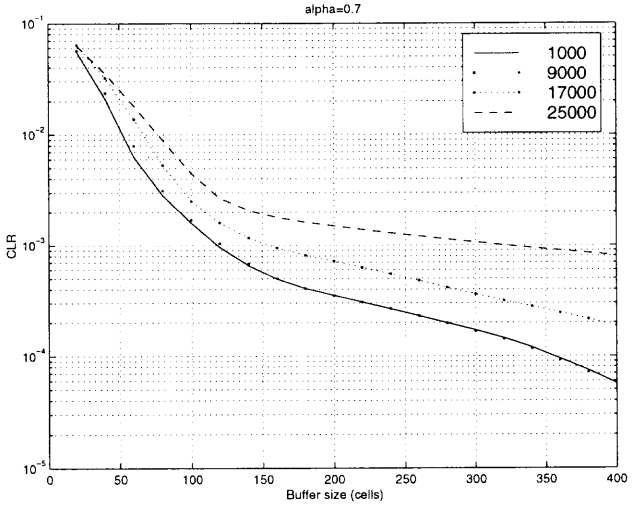

Figure 8.5 CLR performance versus buffer size with $\alpha = 0.7$, δ=1000, 9000, 17000, and 25000, respectively

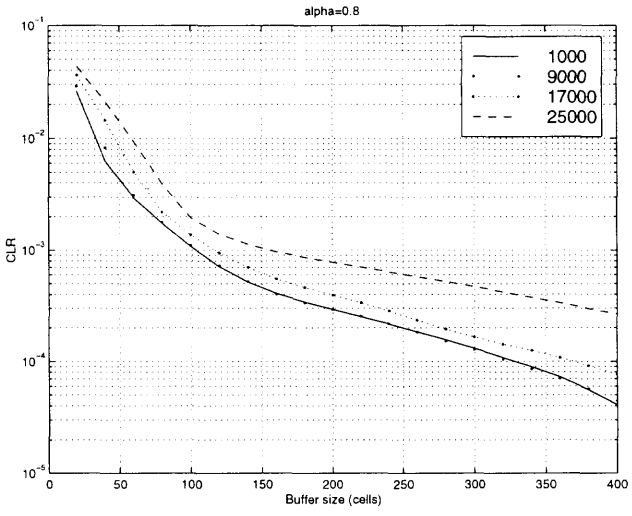

Figure 8.6 CLR performance versus buffer size with $\alpha = 0.8$, δ=1000, 9000, 17000, and 25000, respectively

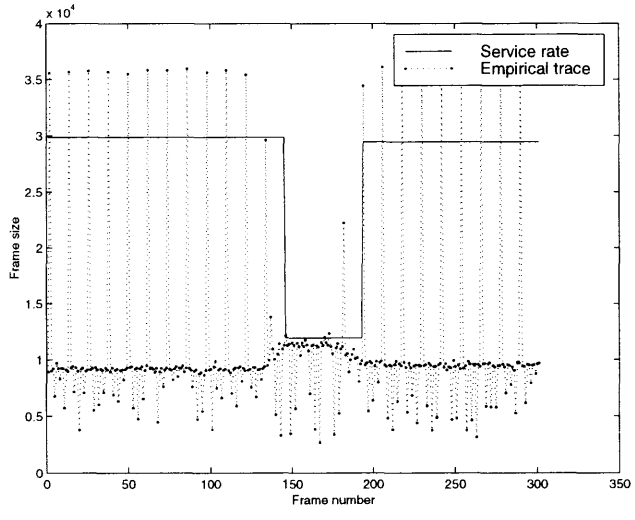

Figure 8.7 Actually allocated bandwidth when δ=9000 and α=0.8

CHAPTER 9

CONCLUSIONS

9.1 Summary

In this dissertation, several MPEG coded video traffic models have been developed. The performance in terms of MMSE and CLR are calculated and simulated.

It is well known that video traffic has short range dependence and long range dependence. Traffic dependence has drastic effect on cell loss ratio and other network performance. To capture LRD and SRD, a Markov-modulated self-similar process model has been proposed to capture ACF characteristics of video traffic.

The PDF of traffic has strong impact on network performance. To describe video traffic accurately, the PDF of traffic should be captured very well. To model MPEG coded traffic, we have proposed to use generalized Beta distribution to model the PDFs of MPEG video traffic. The results demonstrated that generalized Beta functions match well to the distribution of MPEG video traffic.

The ACF of MPEG video traffic fluctuates around three envelopes, implying the facts that different coding methods reduce the data dependence by different amount. To model the traffic more accurately, a structurally modulated self-similar process model has been proposed. This model can capture the ACF of MPEG video traffic very well, and therefore it is a very accurate traffic model for MPEG video traffic.

To justify the performance of the proposed models, the self-similar traffic is generated by F-ARIMA algorithm. Through the simulation of an ATM switch with a limited buffer size, the CLR obtained by empirical trace and traffic models are compared. The difference between CLRs obtained by traffic models and empirical trace are within one order, which is hardly achievable by other models, even for the case of JPEG coded videos.

Although accurate, the traffic model proposed in Chapter 5 is complicated. To simplify the traffic model, a simpler traffic model was proposed. The traffic model consists of three self-similar processes. Simulation results showed that not only the ACF of video traffic can be captured accurately, the GOP pattern can also be reproduced. This is a better traffic model in the sense that it can not only capture the ACF and PDF of video traffic, but the traffic generated by this model is more "similar" to the empirical trace.

A good traffic model is only the first step to design a good network. To make the network utilization high, a good network management policy is required. Video traffic is bursty. The peak-rate to average-rate ratio is very large. It is very difficult to achieve high bandwidth utilization with guaranteed quality of service. To increase bandwidth utilization and guarantee quality of service, an efficient bandwidth allocation algorithm has been proposed for delivery of pre-recorded video sequence. The algorithms have been proved to be efficient in increasing bandwidth utilization, reducing buffer size and reducing CLR.

For real-time video traffic, it is even more difficult to achieve high bandwidth utilization because of the time limitation and unavailability of information needed. In order to increase bandwidth utilization for real-time video with no interactivity, an algorithm based on I and B frames was proposed. For a reasonable renegotiation frequency, the bandwidth utilization can be improved by the use of dynamic bandwidth allocation algorithms.

9.2 Future Work

Generation of traffic data using self-similar processes is time consuming. It is favorable for industry to use simple and accurate traffic model. A simpler traffic model is more attractive. P and B frames have some relations with I frame. It is possible to develop models for B and P frames based on the traffic model of I frames.

Interactive video transmission over ATM is more attractive for the entertainment industry. An efficient video transmission algorithm with little delay needs to be developed. Nonlinear prediction algorithm, such as neuron networks, may be used to predict bandwidth needed for video transmission because the change of frame size itself is not linear.

APPENDIX A

LIST OF ACRONYMS

AAL ATM Adaptation Layer

ACF Auto-Correlation Function

ADSL Asymmetric Digital Subscriber Line

AR Auto-Regressive

ARIMA Auto-Regressive Integrated Moving Average

ARMA Auto-Regressive Moving Average

ATM Asynchronous Transfer Mode

B-ISDN Broadband Integrated Service Digital Network

CAC Call Admission Control

CBR Constant Bite Rate

CDF Cumulative Distribution Function

CDV Cell Delay Variation

CER Cell Error Ratio

CLR Cell Loss Ratio

DAR Discrete Auto-Regressive

D-BIND Deterministic Bounding Interval Length Dependent

DCT Discrete Cosine Transformation

DSM Digital Storage Media

DSM-CC Digital Storage Media Command and Control

FARIMA Fractional ARIMA

FGN Fractional Gaussion Noise

FTTB Fiber to the Building

FTTC Fiber to the Curb

FTTH Fiber to the Home

GOP Group of Pictures

HDTV High Definition Television

HFC Hybrid Fiber Coax

ISDN Integrated Service Digital Network

JPEG Joint Picture Expert Group

LAN Local Area Network

LRD Long Range Dependence

MPEG Motion Picture Expert Group

MAP Markov Arrival Process

MMPP Markov Modulated Poisson Process

MMSSP Markov Modulated Self-Similar Process

MOD Movies On Demand

MRP Markov Renewal Model

NVOD Near Video On Demand

PDF Probability Density Function

PMD Physical Media Dependent

QoS Quality of Service

SECBR Severely Error-ed Cell Block Ratio

SMG Statistical Multiplexing Gain

SPP Switched Poisson Process

SRD Short Range Dependence

TC Transmission Convergence

TES Transform Expand Samples

VBR Variable Bit Rate

VC Virtual Circuit

VCI Virtual Circuit Identifier

VOD Video On Demand

VP Virtual Path

VPI Virtual Path Identifier

REFERENCES

1. W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, "on the self-similar nature of ethernet traffic," *ACM/SIGCOMM Computer Communication Review*, pp. 183–193, 1986.

2. J. Bera, R. Sherman, M. S. Taqqu, and W. Willinger, "Long-range dependence in variable bit-rate video traffic," *IEEE Transaction on Communications*, pp. 1566–1579, 1995.

3. J. Beran, M. Taqqu, W. Willinger, and R. Sherman, "Long-range dependence in variable-bit-rate video traffic," *IEEE Transactions on Communications*, vol. 43, pp. 1566–1579, 1995.

4. H. J. Fowler and W. E. Leland, "Local area network traffic characteristics, with implications for broadband network congestion management," *IEEE Journal on Selected Areas in Communications*, pp. 1139–1149, 1991.

5. M. W. Garrett and W. Willinger, "Analysis, modeling and generation of self-similar VBR video traffic," in *Proceedings of ACM SIGCOMM'94*, vol. 1, (London, U.K.), pp. 269–280, August 1994.

6. W. E. Leland, W. W. S. Taqqu, and D.V.Wilson, "On the self-similar nature of ethernet traffic (extended version)," *IEEE/ACM Transactions on Networking*, vol. 2, pp. 1–14, Febuary 1994.

7. V. Paxson and S. Floyd, "Wide-area traffic: The failure of poisson modeling," in *Proceedings of SIGCOMM'94*, (London, U.K.), pp. 257–268, August 1994.

8. B. Melamed and D. E. Pendarakis, "Modeling full-length VBR video using markov-renewal-modulated tes models," *IEEE Journal on Selected Areas in Communications*, vol. 16, pp. 600–612, June 1998.

9. V. S. Frost and B. Melamed, "Traffic modeling for telecommunications networks," *IEEE Communications Magazine*, no. 3, pp. 70–81, 1994.

10. D. L. Jagerman and B. Melamed, "The transition and autocorrelation structure of TES process part i:general theory," *Stochastic Models*, no. 2, pp. 193–219, 1992.

11. D. L. Jagerman and B. Melamed, "The transition and autocorrelation structure of TES process part ii:special cases," *Stochastic Models*, no. 3, pp. 499–527, 1992.

12. B. Melamed and D. Pendarakis, "A TES-based model for compressed star wars video," in *Proceedings of IEEE GLOBCOM'94*, pp. 120–126, 1994.

13. B. M. D. Raychaudhuri, B. Sengupta, and J. Zdepski, "TES-based video source modeling for performance evaluation of integrated networks," *IEEE Transactions on Communications*, no. 10, pp. 2773–2777, 1994.

14. B. Melamed, "TES: A class of methods for generating autocorrelated uniform variates," *ORSA Journal on Computing*, no. 4, pp. 317–329, 1991.

15. A. Lazar, G. Pacifini, and D. E. Pendarakis, "Modeling video sources for real-time scheduling," in *Proceedings of IEEE GLOBCOM'93*, pp. 835–839, 1993.

16. D. Reininger, B. Melamed, and D. Raychaudhuri, "Variable bit-rate MPEG video: characteristics, modeling and multiplexing," in *Proceedings of the 14th International Teletraffic Congress–ITC 14*, (Antibes Juan-les-Pins, France), pp. 295–306, 1994.

17. D. P. Heyman, A. Tabatabai, and T. V. Lakshman, "Statistical analysis and simulation study of video teleconferencing traffic in ATM networks," *IEEE Transactions on Circuits and Systems for Video Technology*, no. 1, pp. 49–59, 1992.

18. D. P. Heyman and T. V. Lakshman, "Source models for VBR broadcast-video traffic," *IEEE/ACM Transaction on Networking*, vol. 4, pp. 40–48, Febuary 1996.

19. P. Pancha and M. E. Zarki, "Bandwidth-allocation schemes for variable-bit-rate MPEG sources in ATM networks," *IEEE Transactions on Circuits and Systems for Video Technology*, no. 3, pp. 190–198, 1993.

20. D. M. Lucantoni, M. F. Neuts, and A. Reibman, "Methods for performance evaluation of VBR video traffic models," *IEEE/ACM Transactions on Networking*, no. 2, pp. 176–180, 1994.

21. W. Fischer and K. Meier-Hellstern, "The markov-modulated poisson process (mmpp) cook-book," *Performance Evaluation*, no. 18, pp. 149–171, 1991.

22. A. Arvidsson and C. Lind, *Using Markovian Models to Replicate Real ATM Traffics, Performance Modelling and Evaluation of ATM Network.* London, England: Chapman & Hall, 1996.

23. M. H. Rossiter, "A switched poisson model for data traffic," *Australian Telecommunication Research*, no. 1, pp. 53–57, 1987.

24. H. Heffes and D. Lucantoni, "A markov modulated characterization of packetized voice and datatraffic and related statistical multiplexer performance," *IEEE Journal on Selected Areas in Communications*, no. 6, pp. 856–868, 1986.

25. R. Gusella, "Characterizing the variability of arrival process with indexes of dispersion," *IEEE Journal on Selected Areas in Communications*, no. 2, pp. 203–211, 1991.

26. F. Yegenoglu, B. Jabbari, and Y. Q. Zhang, "Motion-classified autoregressive modeling of variable bit-rate video," *IEEE Transactions on Circuits and Systems for Video Technologies*, no. 1, pp. 42–53, 1993.

27. S. P., M. Schwartz, and S.Dixit, "A histogram-based model for video traffic behavior in an ATM multiplexer," *IEEE/ACM Transactions on Networking*, vol. 1, pp. 446–459, August 1993.

28. D. Heyman, T. V. Lakshman, A. Tabatabai, and H.Heeke, "Modeling teleconference traffic from VBR video coders," in *Proceedings of IEEE ICC'94*, vol. 3, (New Orleans), pp. 1744–1748, 1994.

29. B. Maglaris, "Performance models of statistical multiplexing in packet video communications," *IEEE Journal on Selected Areas in Communications*, vol. 36, pp. 834–844, July 1988.

30. M. Nomura, T. Fujii, and N. Ohta, "Basic characteristics of variable bit rate video coding in ATM enviroment," *IEEE Journal on Selected Areas in Communications*, pp. 752–760, 1989.

31. R. Grunenfelder, J. P. Cosmas, S. Manthorpe, and A. Odinma-Okafor, "Characterization of video codecs as autoregressive moving average processes and related queueing system performance," *IEEE Journal on Selecter Areas in Communications*, no. 3, pp. 284–293, 1991.

32. G. Ramamurthy and B. Sengupta, "Modeling and analysis of a variable bit rate multiplexer," in *Proceedings of IEEE INFOCOM*, pp. 817–827, 1992.

33. M. R. Frater, J. F. Arnold, and P. Tan, "A new statistical model for traffic generated by VBR coders for television on the broadband isdn," *IEEE Transactions on Circuits and Systems for Video Technology*, no. 6, pp. 521–526, 1994.

34. A. I. Elwalid and D. Mitra, "Fluid models for the analysis and design of statistical multiplexing with loss priorities on multiple classes of bursty traffic," in *Proceedings of IEEE INFOCOM'92*, pp. 3c.4.1–3c.4.11, 1992.

35. J. Zhang, "Performance study of markov modulated fluid flow models with priority traffic," in *Proceedings of IEEE INFOCOM'93*, pp. 1a.2.1–1a.2.8, 1993.

36. M. Krunz and A. M. Makowski, "Modeling video traffic using $m/g/\infty$ input processes: A compromise between Markovian and LRD models," *IEEE Journal on Selected Areas in Communications*, vol. 16, pp. 733–749, June 1998.

37. D. P. Heyman, "The GBAR source model for VBR videoconferences," *IEEE/ACM Trans. on Networking*, vol. 5, pp. 554–560, August 1997.

38. H. E. Hurst, "Long-term storage capacity of reservoirs," *Transactions of the Am. Socity of Civil Engineering*, vol. 116, no. 770-799, 1951.

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

40. W. Leland and D. Wilson, "High time resolution measurements and analysis of lan traffic: Implications for lan interconnection," in *Proceedings of IEEE INFOCOM'91*, (Bal Harbour), 1991.

41. S. Klivansky, A. Mukherjee, and C. Song, "On long-range dependence in nsfnet traffic," tech. rep., College of Computing, Georgia Institute of Technology, 1994.

42. B. Tsybakov and N. Georganas, "On self-similar traffic in ATM queues: Definitions, overflow probability bound, and cell delay distribution," *IEEE/ACM Transactions on Networking*, vol. 5, pp. 397–408, 1997.

43. B. K. Ryu and A. Elwalid, "The importance of long-range dependence of VBR video traffic in ATM traffic engineering: Myths and realities," in *Proceedings of ACM Sigcomm'96*, (Stanford University, CA), pp. 3–14, August 1996.

44. J. Beran, *Statistics for Long-Memory Processes.* New York: Chapman & Hall, 1994.

45. P. Kokoszka and M. Taqqu, "Parameter estimation for infinite variance fractional ARIMA," *The Annals of Statistic*, pp. 1880–1993, 1996.

46. G. Samorodnitsky and M. S. taqqu, *Stable Non-Gaussian Process: Stochastic Models with Infinite Variance.* New York, London: Chapman & Hall, 1994.

47. E. W. Knightly and H. Zhang, "D-BIND: An accurate traffic model for providing QoS guarantees to VBR traffic," *IEEE/ACM Transactions on Networking*, no. 2, pp. 219–231, 1997.

48. R. A. M. Zukerman and T. Neame, "Performance of a single server queue with self-similar input," in *Proceedings of IEEE ICC'95*, vol. 3, (Seattle, WA), pp. 461–465, 1995.

49. N. G. Duffield and N. O'Connell, "Large deviations and overflow probabilities for the general single server queue, with applications," DIAS-STP-93-30, Dublin Institute for Advandced Studies, 1993.

50. C. Huang, M. Devetsikiotis, I. Lambadaris, and A. R. Kaye, "Fast simulation for self-similar traffic in ATM networks," in *Proceedings of IEEE ICC' 95*, (Seattle), pp. 11–22, June 1995.

51. I. Norros, "A storage model with self-similar input," *Queueing Systems*, vol. 14, pp. 387–396, 1994.

52. M. Grossglauser and J.-C. Bolot, "On the relevance of long-range dependence in network traffic," *IEEE/ACM Transactions on Networking*, 1998.

53. G. J. Hahn and S. S. Shapiro, *Statistical Models in Engineering.* New York: John Wiley & Sons Inc., 1967.

54. C. Huang, M. Devetsikiotis, I. Lambadaris, and A. R. Kaye, "Modeling and simulation of self-similar variable bit rate compressed video: a unified approach," in *Proceedings of SIGCOMM'95*, (Cambridge, MA), pp. 114–125, 1995.

55. M. Krunz and S. K. Tripathi, "On the characterization of VBR MPEG streams," in *Proceedings of SIGMETRICS'97*, (Cambridge, MA), pp. 192–202, June 1997.

56. M. Hamdi, J. W. Roberts, and P. Rolin, "Rate control for VBR video coders in broad-band networks," *IEEE Journal on Selected Areas in Communications*, vol. 15, pp. 1040–1051, 1997.

57. W. chi Feng and J. Rexford, "A comparison of bandwidth smoothing techniques for the transmission of prerecorded compression video," in *IEEE INFOCOM*, (Kobe, Japan), pp. 58–66, April 1997.

58. J. Rexford, S. Sen, J.Dey, W. Feng, J. Kurose, J. Stankovic, and D. Towsley, "Online smoothing of live, variable-bit-rate video," in *the International Workshop on Network and Operating System Support For Digital Audio and Video(NOSSDAV'97)*, (St. Louis, Missouri), pp. 249–258, May 1997.

59. K. Shiomoto, S. Chaki, and N. Yamanka, "A simple bandwidth management strategy based on measurements of instantaneous virtual path utilization in ATM networks," *IEEE/ACM Transactions on Networking*, vol. 6, pp. 625–634, 1998.

60. E. W. Fulp and D. S. Reeves, "An algorithm for dynamic bandwidth allocation of MPEG videos," in *on-line proceedings of IEEE RTSS workshop on resource Allocation in Multimedia System*, (Washington, DC), December 1996.

61. D. Reininger, M. Ott, G. Michelitsch, and G. Welling, "Dynamic bandwidth allocation for distributed multimedia with adaptive qos," in *on-line proceedings of IEEE RTSS workshop on resource Allocation in Multimedia System*, (Washington, DC), December 1996.

62. A. M. Adas, "Using adaptive linear prediction to support real-time VBR video under RCBR network service model," *IEEE/ACM Transactions on Networking*, vol. 6, pp. 635–644, 1998.