Brief Papers

On Modeling MPEG Video Traffics

Nirwan Ansari, Hai Liu, Yun Q. Shi, *Senior Member, IEEE*, and Hong Zhao

***Abstract*—This paper traces the development/evolution of three of our recently proposed MPEG coded video traffic models, that can capture the statistical properties of MPEG video data. The basic ideas behind these models are to decompose an MPEG compressed video sequence into several parts according to motion/scene complexity or data structure. Each part is described by a self-similar process. These different self-similar processes are then combined to form the respective models. In addition, Beta distribution is used to characterize the marginal cumulative distribution (CDF) of the self-similar processes. Comparison among the three models shows that the latest model (called the simple IPB composite model) is the most practical one in terms of accuracy and complexity. Simulations based on many real MPEG compressed movie sequences, including *StarWars*, have demonstrated that the simple model can capture the autocorrelation function (ACF) and the marginal CDF very closely.**

***Index Terms*—ARIMA, autocorrelation, motion pictures, MPEG, self-similar process, traffic modeling.**

I. Introduction

THE TREND to transmit videos over a network is emerging. Traffic models are important for network design, performance evaluation, bandwidth allocation, and bit-rate control. It was observed, however, that traditional models fall short in describing the video traffic because video traffic is strongly autocorrelated and bursty [1]. To accurately model video traffic, autocorrelations among data should be taken into consideration. A considerable amount of effort on video modeling has been reported that include: Markov Modulated Rate Process (MMRP) [2], Discrete Auto-Regressive Process [DAR (1)] [3], Fluid Models [4], Markov-Renewal-Modulated transform expand sample (TES) Models [5], Long Range Dependency (LRD) models or Self-Similar models [6], $M/G/\infty$ input process models [7], GBAR Model [8], and Markov chain model [9].

Markov-Renewal-Modulated TES models are used to model motion JPEG encoded picture sequence. One of the drawbacks of the TES approach is that the ACF of a TES process for lags beyond one cannot be derived analytically. The ACF can only be obtained by searching in the parameter space, and thus good results can hardly be guaranteed. The ACF of MPEG encoded video sequence is quite different from that of JPEG sequence as shown in Figs. 1 and 2. Thus this method fails to capture the second-order statistics of MPEG sequences.

The $M/G/\infty$ input process model is a compromise between the LRD and short range dependence (SRD) models [7]. Simulation results were found to be better than those of a self-similar process when the switch buffer is relatively small. Better results than those of DAR (1) model were found when the buffer size is large. The results were obtained from JPEG and MPEG-2 I-frame sequences.

The MPEG video model presented in [9] is a Markov chain model based on the Group of Pictures' (GOP) level process rather than the frame level process. This has the advantage of eliminating the cyclical variation in the MPEG video pattern, but at the expense of decreasing the resolution of the time scale. Typically a GOP has duration of a half second, which is considered long for high speed networks. Of particular interests in video traffic modeling are the frame-size distribution and the traffic correlation. The frame size distribution has been studied in many existing works.

Krunz [10] proposed a model for MPEG video, in which, a scene related component is introduced in the modeling of I frames, but ignoring scene effects in P and B frames. The scene length is i.i.d. with common geometric distribution. I frames are characterized by a modulated process in which the local variations are modulated by an Auto-Regressive (AR) process that varies around a scene related random process with log-normal distribution over different scenes; i.e., two random processes were needed to characterize I frames. The sizes of P and B frames were modulated by two i.i.d. random processes with log-normal marginals. This model uses several random process and need to detect scene changes, thus complicating the modeling process.

In this paper, we analyze and compare three of our recently developed MPEG video traffic models that can capture the LRD characteristics of video ACF [11]–[13]. The basic ideas behind these models are to decompose an MPEG compressed video sequence into several parts according to motion/scene complexity or data structure; each part described by a self-similar process. These different self-similar processes are then combined in respective fashion to form the models. In addition, Beta distribution is used to characterize the marginal cumulative distribution (CDF) of the self-similar processes. Comparison among three models leads to the observation that the simple IPB composite model is preferred in terms of accuracy and simplicity. Simulations on many real MPEG compressed movie sequences in-

Manuscript received August 16, 2002; revised October 25, 2002. This work was supported in part by the New Jersey Commission on High Education via the NJI-TOWER project, and NJ Commission on Science and Technology via the NJ Center for Wireless Telecommunications.

The authors are with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, University Heights, Newark, NJ 07102 USA.

Digital Object Identifier 10.1109/TBC.2002.806794

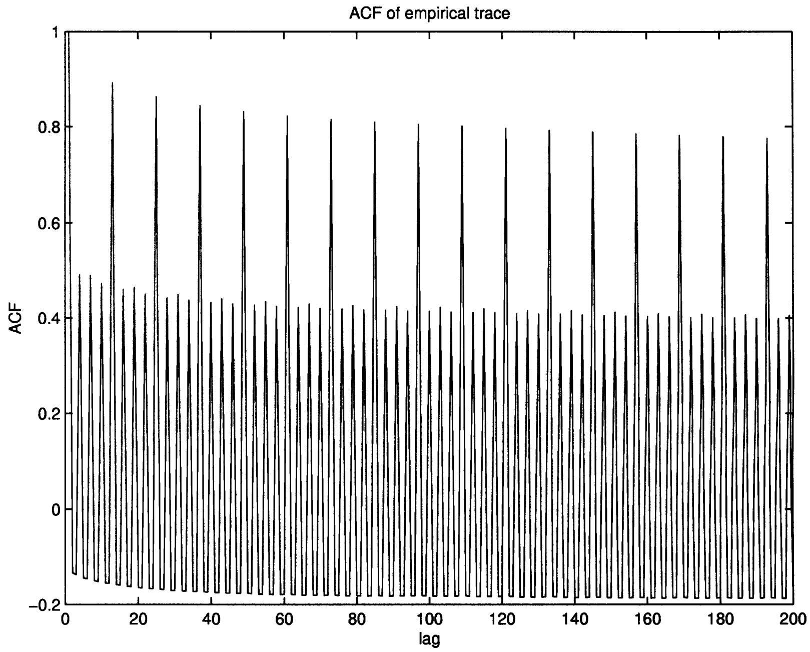

Fig. 1. ACF of MPEG compressed video *StarWars*.

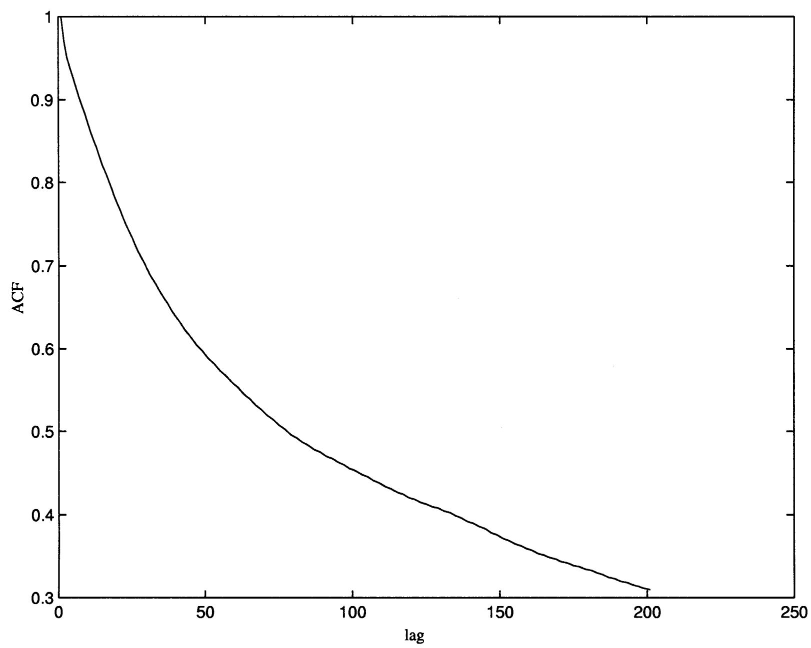

Fig. 2. ACF of JPEG compressed video *StarWars*.

cluding *StarWars* have demonstrated that our new simple model can capture the LRD of ACF and the marginal CDF very well. The rest of the paper is organized as follows. In Section II, empirical data and their ACFs are described. Section III discusses the decomposition of data according to motion/scene complexity or data structure. Modeling of each part and combination of the decomposed parts are discussed in Section IV. Using Beta distribution to model CDF of the video traffic is presented in Section VI. Experiments are discussed in Section VII. Network performance in terms of the average queue size is presented in Section VIII. Concluding remarks are given in Section IX.

II. Empirical MPEG Encoded Data and Their ACFs

The MPEG coded data of *StarWars*[1] were used as the representative of the empirical data among many MPEG coded video sequences. The *StarWars* video sequence contains scenes ranging from low complexity/motion 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 two hours at the frame rate of 25 frames per second. The frames were organized as follows:

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

IBBPBBBPBBPBB IBBPBB…, i.e., 12 frames in a Group of Pictures (GOP). I frames are those which were coded by using intraframe coding method (exploit the spatial redundancy within a picture), P frames are those which were coded by using interframe coding technique (exploit the temporal redundancy within a video sequence), and B frames were also coded by interframe coding technique except that they are predicted via both forward and backward methods.

For a stationary process $X = \{X_n\colon n = 1, 2, \ldots\}$ with mean μ and variance σ^2, the autocorrelation function[2] of X is defined by:

$$r(k) = \frac{E[(X_n - \mu)(X_{n+k} - \mu)]}{\sigma^2}. \quad (1)$$

The ACF of the frame size of the MPEG coded *StarWars* is shown in Fig. 1, and it is quite different from that of JPEG coded *StarWars* (see Fig. 2). The ACF of MPEG coded data depends on the GOP pattern, and in principle always looks like Fig. 1 if the same GOP pattern is used for the whole sequence. From Fig. 1 we note that the ACF fluctuates around an envelope, reflecting the fact that, after the use of motion estimation and motion compensation techniques, the dependency between frames is reduced. This characteristics should be taken into consideration in modeling MPEG coded video sequences.

III. Decomposition of MPEG Data

The fluctuation of the ACF as shown in Fig. 1 has convinced us that such fluctuation can hardly be captured by a single random process, and has further led us to the intuitive belief that the data should be decomposed into several parts, each captured by a random process. This section describes the intuition that has resulted in the three proposed decompositions.

A. Decomposition According to Motion/Scene Complexity

With the conjecture that the fluctuation of the ACF of MPEG coded video data was caused by motion/scene complexity and the GOP structure, we proposed to divide the traffic data into three different parts—inactive part, active part, and the most active part, inspired by [2]. Suppose $f(i)$ is the number of bits in the ith frame. The video traffic can be classified as follows:

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

2) Similarly, the noninactive 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.

B. Decomposition According to MPEG Data Structure

Although the model based on the decomposition introduced above can model each part of the video traffic very well, it cannot capture the ACF of the whole sequence very well. This slight discrepancy highly depends on how one defines motion/scene complexity and has thus inspired us to decompose the MPEG data according to the MPEG data structure.

[2]In reality, (1) is known as the autocovariance function rather than the autocorrelation function. The difference between the two is a constant offset of $(\mu/\sigma)^2$. However, (1) has been interchangeably referred to either in the literature for video traffic modeling.

Specifically, in the second proposed model, we decompose the MPEG traffic into 10 sub-sequences X_I, X_P, X_{B_1}, X_{B_2} …, 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.

Although the above-described model is more accurate than the first one, it uses eight random processes to model eight different B frame subsequences. Since B frames are usually much smaller than I and P frames, and the B frames play less important role than I and P frames. The modeling of B frames may not be as critical as that of I and P frames in estimating network performance [9]. So we can further simplify this model. Through careful analysis, we found the B frames have similar properties in terms of the coding mechanisms, and so we combine all the B frames into one subsequence, resulting in three sub-sequences X_I, X_P and X_B. As before, X_I consists of all I frames, X_P consists of all P frames, but, now, X_B consists of all B frames.

IV. Modeling Each Part and Combining Parts to Obtain MPEG Coded Traffic Models

To obtain a model that can capture the ACFs of MPEG data, we model each part by a self-similar process and then combine these processes in an appropriate fashion. In this section, the utilization of the self-similar processes is justified. Three different ways in combining, leading to three different models, are described.

A. Markov Modulated Self-Similar Processes Model

The ACF of each self-similar process is very different from that of the original sequence. For the sake of brevity, only the ACF associated with the active part is shown in Fig. 3. The fluctuation is no longer that drastic. We have used $k^{-\beta}$, $e^{-\beta k}$, and $e^{-\beta\sqrt{k}}$ to approximate the ACFs of a self-similar process, a Markov process and an $M/G/\infty$ input process, respectively. From Fig. 3, it is quite clear that $k^{-\beta}$ is a better approximation of the ACF of the active part of the MPEG coded *StarWars* empirical data than the other two. We therefore use self-similar processes s_1, s_2, and s_3 to model these processes. Using the least squares fit, 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 self-similar processes are $H = 0.8339$, 0.8465, and 0.7802, respectively.

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 adopted 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, where states S_1, S_2, and S_3 representing the inactive part, the active part and the most active part, respectively, 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:

$$p_{ij} = \frac{N_{ij}}{N_i}, \quad (2)$$

where N_i is the total number of times that the system goes through states S_i, N_{ij} is the number of times that the system

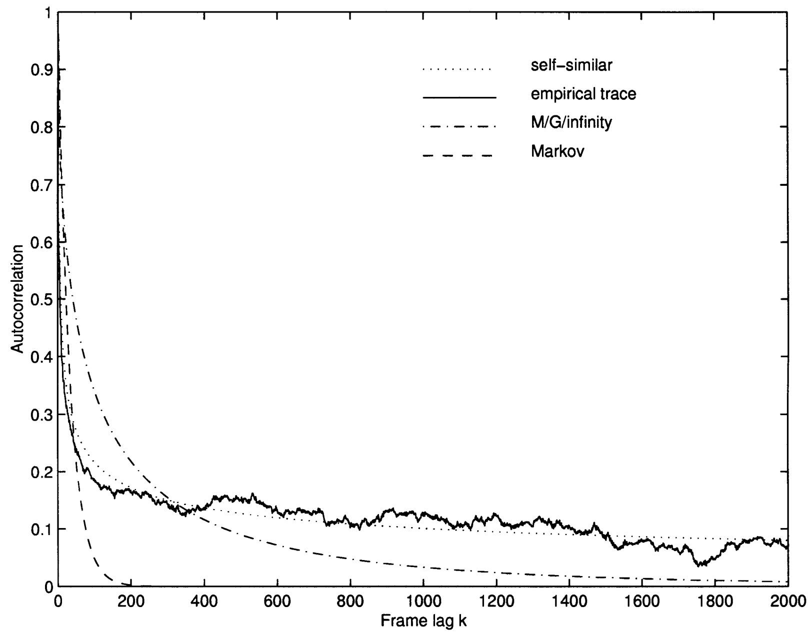

Fig. 3. ACF of the active part of *StarWars*.

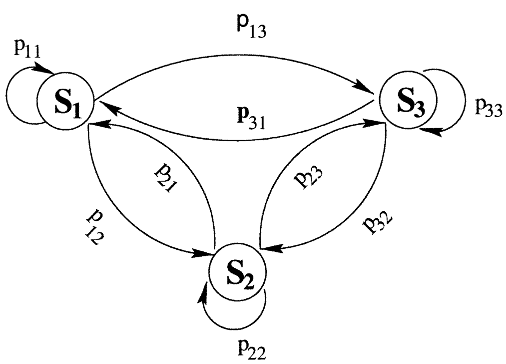

Fig. 4. A Markov modulated self similar process for modeling MPEG videos.

make a transition to state S_j from state S_i. For the *StarWars* 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 the video traffic.

B. Structurally Modulated Self-Similar Processes Model

Since the technique used to classify inactive, active and very active parts of MPEG video traffic adopted in the first model (refer to Section III-A) is rather crude, it is conceivable that the performance of the first model (decomposing video into three parts based on motion/scene complexity) can be improved significantly with an advanced technique which can accurately identify motion/scene complexity. This task is by no means easy, and is a hot research subject in the field of computer vision and video processing. On the other hand, the GOP structure of MPEG data is universal regardless of motion/scene complexity. Therefore, the decomposition according to the MPEG data structure is practical and straightforward. In the second model, we decompose the MPEG traffic into several subsequences based on the GOP structure, X_I, X_P, X_{B_1}, X_{B_2}, ... 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 also 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. For the sake of brevity, only the approximation for X_P is shown in Fig. 5. We have obtained similar results for X_I, X_{B_1}, X_{B_2}, ... It is quite obvious that self-similar processes are the better choice, justifying our usage of self-similar processes for modeling these data.

Using the least squares fit, $\beta = 0.4662$, 0.3404, 0.4468, 0.4779, 0.4294, 0.4656, 0.4380, 0.4682, 0.4465, and 0.4606 are derived for X_I, X_P, X_{B_1}, X_{B_2}, ..., and X_{B_8}, respectively, for *StarWars*. In this model, we combine X_I, X_P, X_{B_1}, X_{B_2}, ..., and X_{B_8} in a manner similar to the GOP pattern to obtain the model for the MPEG coded traffic. This model can be used to more accurately generate traffic data than the first one.

Although the model described above is more accurate than the first one, it uses eight self similar random processes, one for each of the different B frame subsequences. So, we can further simplify this model by modeling each of the three sub-sequences X_I, X_P and X_B as discussed in Section III-B by a self-similar process, hence referred to as the simple IPB composite model. To show that the self-similar process is a better choice for the simple model, we have conducted the following experiment. That is, we use $k^{-\beta}$, $e^{-\beta k}$, 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 model I, P, and B frame sub-sequences of 18 commonly used MPEG coded video sequences. Designing and conducting such a set of experiments in justifying the utilization of self-similar random processes to

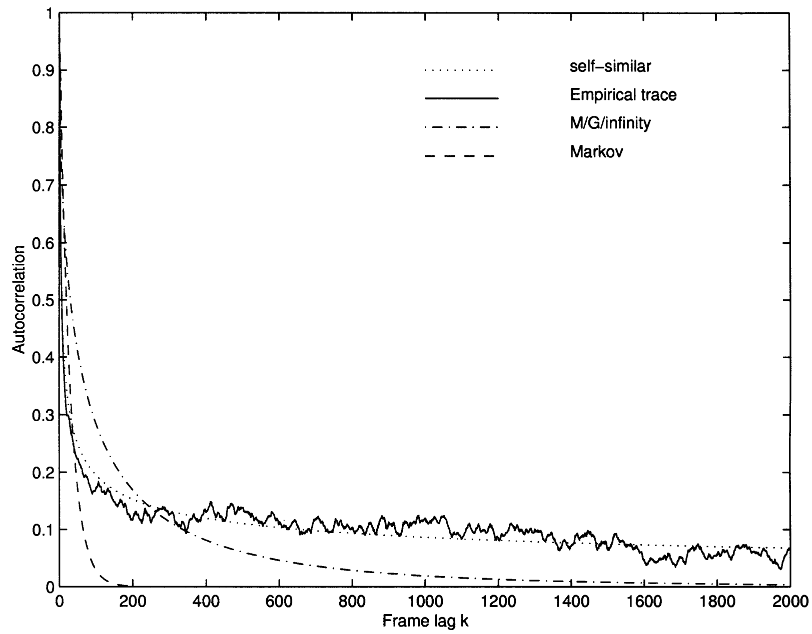

Fig. 5. Approximation for ACF of P frames by: Self similar, $M/G/\infty$, and Markov processes.

TABLE I
LEAST SQUARE ERRORS OBTAINED BY SELF-SIMILAR (SS) PROCESS, MARKOV, AND $M/G/\infty$

Trace	I frame			P frame			B frame		
	SS	$M/G/\infty$	Markov	SS	$M/G/\infty$	Markov	SS	$M/G/\infty$	Markov
StarWars	1.5820	5.0527	7.2517	0.6630	12.8669	25.4433	0.5987	13.7523	32.0705
Asterix	0.6339	2.6476	5.2218	0.3783	0.7556	2.9004	0.4667	2.0326	11.2931
MrBean	0.9406	0.6141	3.5079	3.0020	1.7339	9.7293	4.1578	1.5768	9.5433
Atp	1.3261	1.1945	1.0703	1.2523	1.0084	0.6863	4.5886	1.8858	1.1787
Bond	4.4744	1.2134	6.4118	0.4621	2.3412	5.0441	1.2248	1.5295	12.9623
Dino	0.4879	1.1936	5.7375	0.5437	2.2678	7.7840	3.6411	11.4657	31.2375
Fuss	0.5588	0.5812	0.4402	1.2778	0.7110	0.3453	1.5033	0.4278	0.3447
Movie2	0.1746	0.1902	0.7299	0.0233	0.2179	1.0820	0.2631	1.7634	4.8799
Mtv	0.2763	0.2871	1.8391	0.4315	1.3929	7.1952	1.2207	5.4241	15.1273
News	0.7754	0.3993	0.2381	0.9677	0.3199	0.2755	5.7957	0.9597	1.3095
Race	0.1841	0.2840	0.2831	0.9239	0.5328	0.0693	2.7580	0.8312	0.2331
Sbowl	0.4207	0.4543	0.1725	1.3894	1.6837	3.5870	3.8844	2.3000	4.2929
Simpson	0.4425	0.1208	1.3436	0.0798	0.4945	2.2366	0.0218	0.4408	3.1401
SocerWM	0.3094	0.4095	0.0651	1.2413	0.7335	0.0243	2.9909	0.8010	0.2886
Star2	2.5977	2.0756	5.0999	1.8096	5.5754	10.8903	8.5032	19.8118	45.0521
Talk2	0.5827	0.5781	0.7382	0.9721	0.2476	1.2716	9.5476	1.9688	3.7166
Talk	0.4451	0.2816	0.8290	0.9303	0.2405	1.0145	7.4527	1.4122	1.3407
Term	0.0715	0.1667	0.3802	0.0330	0.3697	1.2219	0.2067	0.7083	2.5137
Average	0.9047	0.9858	2.2978	0.9101	1.8607	4.4889	3.2681	3.8426	9.4020

characterize I, P, and B frame subsequences is part of the contribution of this article. In fact, P and B frame subsequences were not modeled by the $M/G/\infty$ process [7] as mentioned in Section I. Table I shows the least squares errors between the ACF of the empirical data and that of the approximated ACF using these three random processes for each of the 18 video sequences. The average mean square errors are shown at the bottom of the table. Note that the self-similar process is the best among the three random processes. For most of the video sequences, it appears that the self-similar processes are the better choices for I, P, and B frames. Modeling these three parts in a way similar to the GOP structure leads to our simple IPB composite model. The synthesized data generated by our proposed simple model and its ACF are very close to the empirical data and its ACF. The performance of this simple model is further discussed in Sections VII and VIII.

TABLE II
ESTIMATED PARAMETERS FOR THE SIMPLE IPB COMPOSITE MODEL USING LEAST SQUARES FIT

Trace	I frame			P frame			B frame		
	γ	η	β	γ	η	β	γ	η	β
StarWars	4.0605	10.4233	0.4662	1.6605	12.0277	0.3404	1.6431	14.0724	0.3040
MrBean	4.9569	6.1068	0.5684	1.0084	6.1457	0.4385	1.9798	43.88	0.2631
Asterix	4.43	6.0825	0.6078	1.7008	5.7095	0.5271	1.6014	8.2953	0.3562
Atp	4.7191	8.9661	0.8415	2.3581	12.7828	0.6541	3.4765	23.4751	0.5302
Bond	3.6932	9.7717	0.4578	2.4559	11.3658	0.3635	3.3685	17.5669	0.2764
Dino	6.1765	10.6556	0.5776	1.3718	6.5978	0.3731	1.7872	17.2641	0.2912
Fuss	2.9395	5.6374	0.8184	2.1856	6.7743	0.6731	2.3760	19.2525	0.6129
Movie2	2.6122	5.7122	0.7699	1.3527	8.8534	0.6419	1.1375	18.8076	0.5041
Mtv	2.9316	7.2989	0.6757	2.3029	11.6746	0.5133	1.666	12.7058	0.3710
News	2.6119	5.3078	0.7167	1.3900	7.7165	0.5330	2.7786	20.0555	0.3653
Race	2.6884	6.7643	0.7777	2.4372	12.4445	0.6813	2.6811	21.7449	0.5529
Sbowl	4.0798	5.5735	0.7900	2.7713	7.7488	0.5410	3.0845	18.3636	0.4624
Simpson	4.7542	6.0256	0.6626	1.3053	7.3020	0.6056	1.9526	44.0132	0.4242
SoccerWM	2.3526	5.7727	0.7667	1.5712	6.9937	0.6312	1.0401	8.8374	0.5180
Star2	3.0648	8.1755	0.5609	0.7524	4.4219	0.4300	0.8869	9.5508	0.2865
Talk2	4.8427	7.4619	0.6583	2.0912	11.3488	0.5470	3.9607	50.2894	0.3288
Talk	4.2759	6.2969	0.7001	1.2827	6.0661	0.6013	2.3166	10.6146	0.46
Term	3.8739	7.8080	0.7606	2.3778	8.6690	0.6701	2.1026	158.3204	0.5103

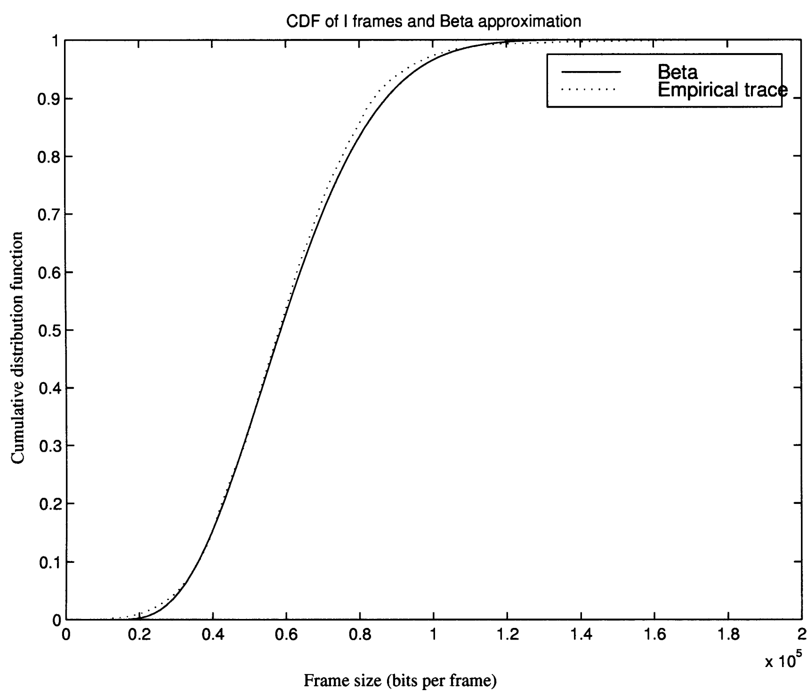

Fig. 6. CDF of I frames and its approximation by Beta distribution.

V. GENERATING SYNTHETIC MPEG STREAMS BASED ON THE SIMPLE MODEL

To synthesize video traffic using our simple IPB composite model requires a self similar traffic generator. There are several options to generate the self similar traffic. Two of the most commonly used methods are the exactly self similar fractional Gaussian noise (FGN) [14] and asymptotically self similar fractional autoregressive integrated moving-average (F-ARIMA) process [14]. F-ARIMA [6], [15], [16] can be used to match any kind of ACF, and is thus adopted here to generate the self similar random process. The algorithm to generate F-ARIMA process is given in the Appendix.

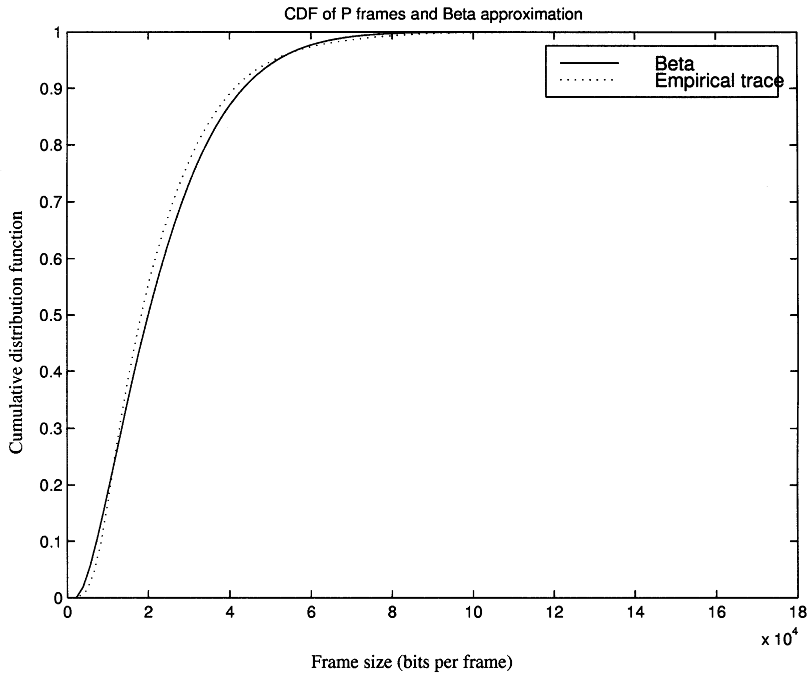

Fig. 7. CDF of P frames and its approximation by Beta distribution.

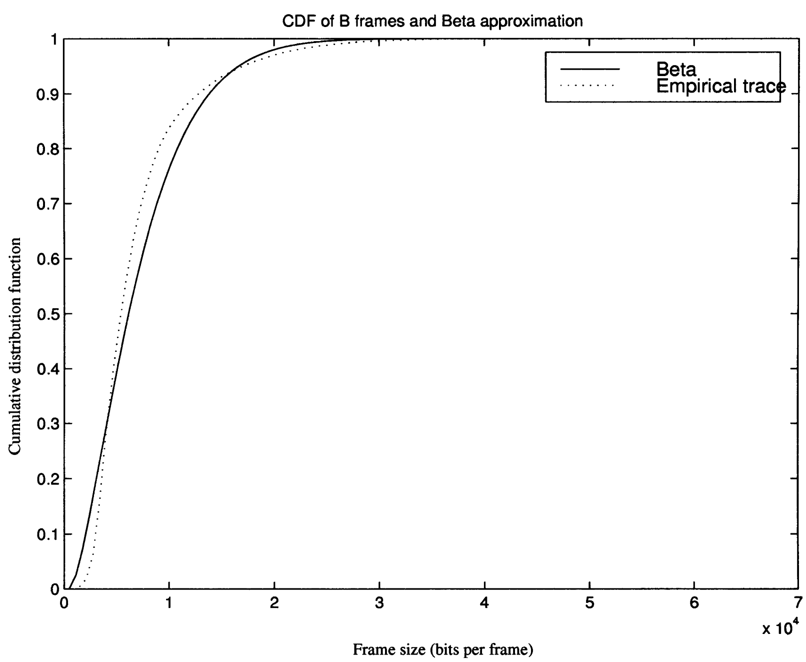

Fig. 8. CDF of B frames and its approximation by Beta distribution.

Video traffic can be synthesized by combining of the three obtained self similar processes in a way similar to the GOP structure. Table II shows the estimated parameter β for different video traces using the least squares fit.

VI. Modeling CDF of I, P, and B Frames Using the Beta Distribution

As mentioned at the beginning, the CDF is another important statistics for video traffic modeling. We use Beta distribution [17] to model the marginal distributions of these processes. The marginal distribution of a Beta distribution process has the following form

$$f(x;\gamma,\,\eta,\,\mu_0,\,\mu_1) = \begin{cases} \dfrac{1}{\mu_1-\mu_0}\dfrac{\Gamma(\gamma+\eta)}{\Gamma(\gamma)\Gamma(\eta)}\left(\dfrac{x-\mu_0}{\mu_1-\mu_0}\right)^{\gamma-1}\left(1-\dfrac{x-\mu_0}{\mu_1-\mu_0}\right)^{\eta-1} \\ \qquad\qquad \mu_0\le x\le\mu_1,\ 0<\gamma,\ 0<\eta \\ 0 \qquad\qquad \text{otherwise.} \end{cases} \tag{3}$$

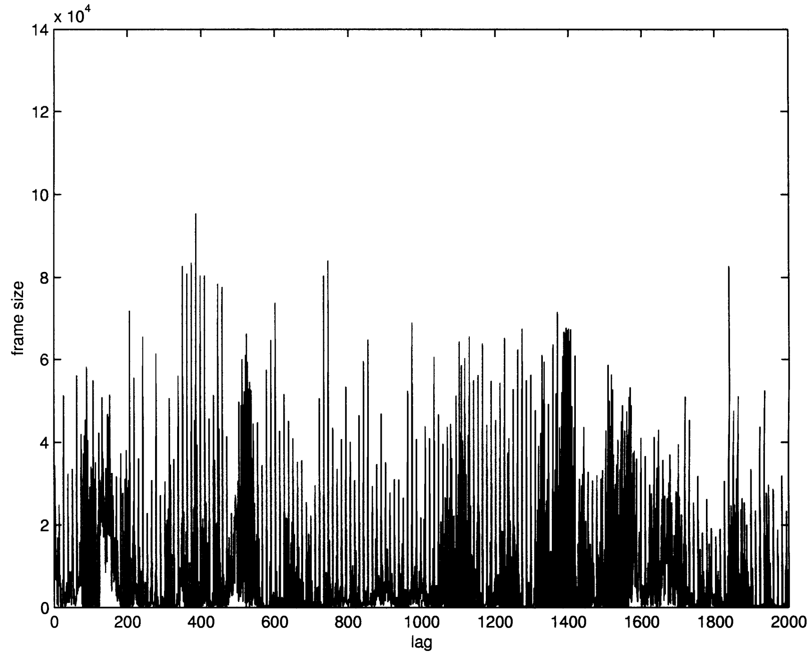

Fig. 9. Traffic data generated by our model for *StarWars*.

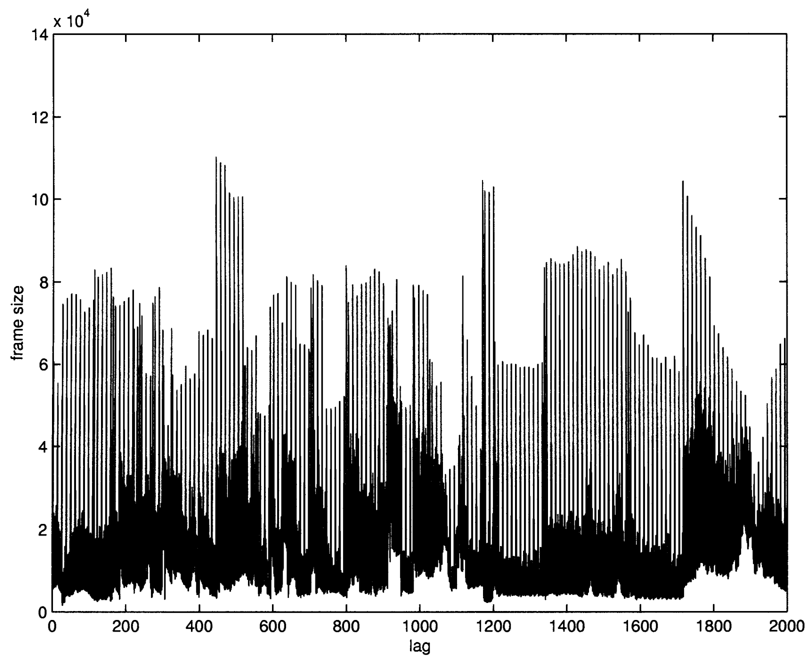

Fig. 10. The empirical traffic trace for *StarWars*.

where γ and η are the 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 used to estimate the parameters of Beta distribution, η and γ:

$$\eta = \frac{1-\overline{x}}{s^2}[\overline{x}(1-\overline{x}) - s^2] \tag{4}$$

$$\gamma = \frac{\overline{x}\hat{\eta}}{1-\overline{x}} \tag{5}$$

where

$$\overline{x} = \frac{1}{N}\sum_{i=1}^{N} x_i, \tag{6}$$

$$s^2 = \frac{N\sum_{i=1}^{N} x_i^2 - \left(\sum_{i=1}^{N} x_i\right)^2}{N(N-1)}, \tag{7}$$

and N is the number of data in the data set.

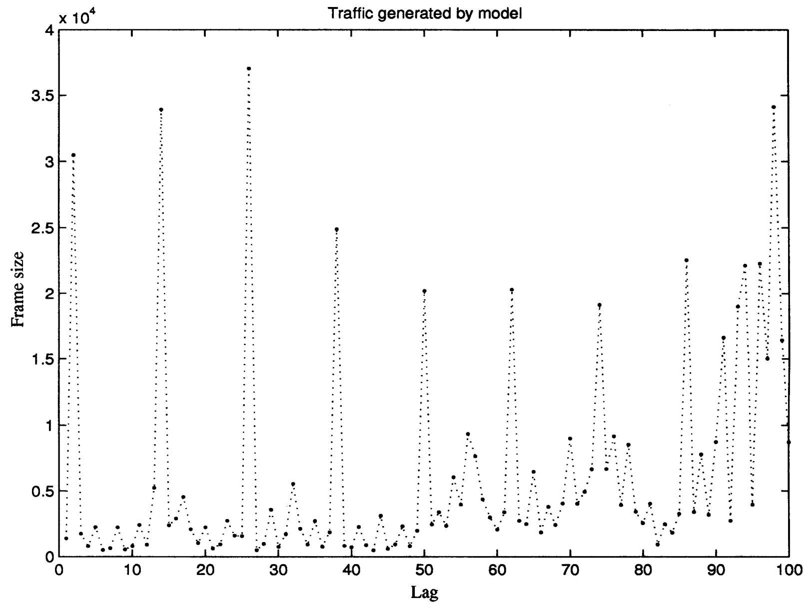

Fig. 11. The first hundred samples of the traffic data generated by our model.

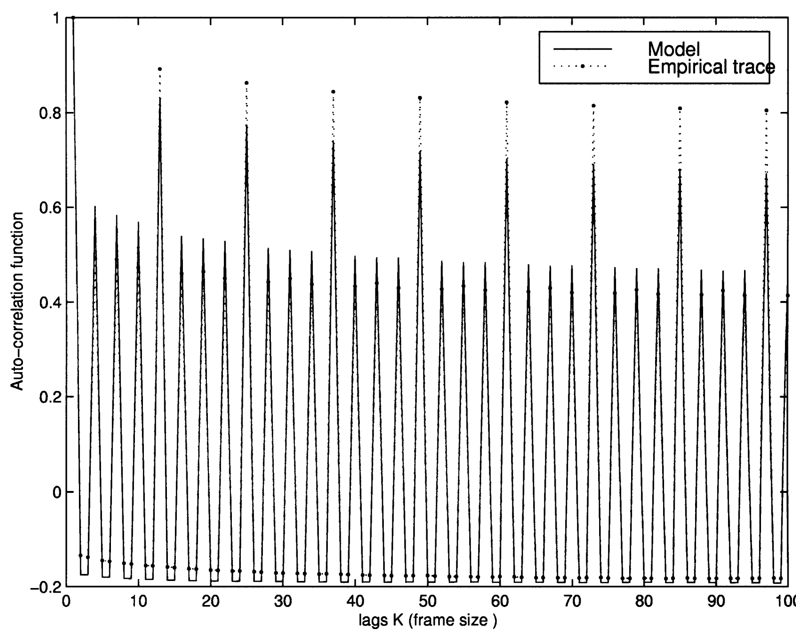

Fig. 12. ACFs of the empirical trace and our traffic model for *StarWars*.

The parameters for the simple IPB composite model, γ and η, are listed in Table II for different video traces. The simulations demonstrate that the Beta-distribution follows the CDF very closely in our simple model. The CDF of I, B, and P frames and their approximations by Beta distribution of the simple model are shown in Figs. 6–8.

VII. Experiments and Discussions

The performance of our simple IPB composite model on several traces is evaluated. Traces generated by the simple IPB composite model are shown in Figs. 9 and 11 with large and small lags. For comparison, the corresponding empirical trace is shown in Fig. 10. From Figs. 9 and 10, we note that the simple IPB composite model can generate traffic, which are similar to the empirical data trace. 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. 11 that the traffic generated by our model can capture this kind of characteristics. Figs. 12 and 13 show the ACF of the empirical trace and synthesized data using the simple IPB composite model; we can see our simple model follows the ACF very well. For simulations on different video traces such as News, Race, SoccerWM and Talk2, similar results are achieved. So the simple model, though simpler than the second model, is almost as accurate as the second model in tracking the ACF of MPEG data. ACF is a very important factor from the networking perspective, because traffic autocorrelation has an important impact on queuing performance. Li and Hwang [18] examined the queue response to various input correlation

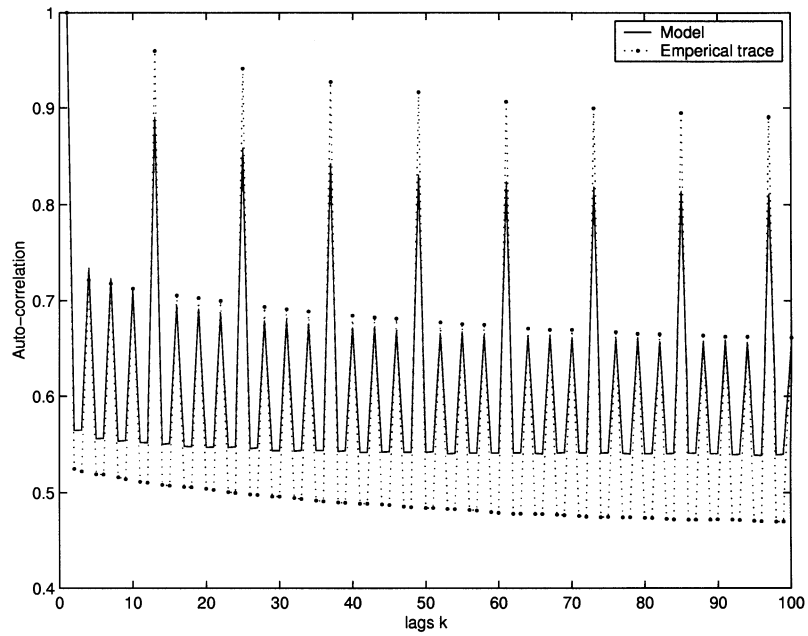

Fig. 13. ACFs of the empirical trace and our traffic model for video trace *Atp*.

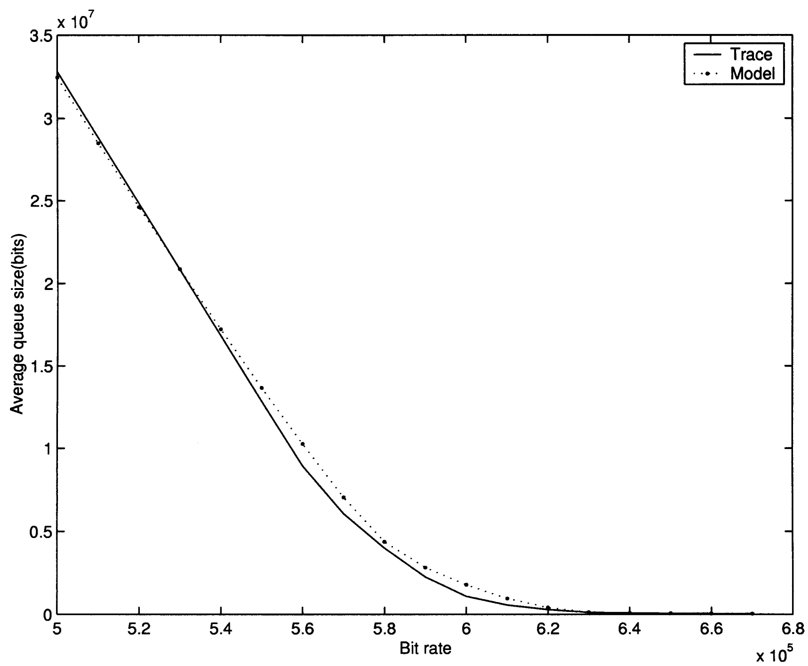

Fig. 14. Average queue size at different service rates for synthesized data and empirical trace.

properties on the basis of input power spectrum in discrete-frequency domain and concluded that the mean queue size is dominated by the low frequency power in the power spectrum. High positive correlation will introduce more input power in the low frequency band, thus resulting in a larger mean queue size. The larger the autocorrelation, the larger the mean queue size. The larger the mean queue size will introduce longer delay and larger cell loss, thus deteriorating the Quality of Service (QoS). The validity of network simulations depends on the accuracy of the traffic model, which in turn depends on how close the video model has captured the statistics of the real video traffic especially the autocorrelation. Our simple IPB composite model follows the ACF very well, and thus it can play an important role for designing and testing future communication networks that will carry multiplexed video traffics.

VIII. Queuing Performance

While a good traffic model is expected to capture statistical properties of the underlying empirical data trace, the ultimate goal is to predict network performance accurately for the purpose of allocating network resources. The queuing performance should be deemed as a crucial factor that determines the appropriateness of a traffic model [7]. Therefore, traffic models are commonly used to predict the queuing performance at a switch; the appropriateness of a model is also determined by its ability to accurately predict the actual queuing behavior. To further verify the appropriateness of our proposed simple model, we have studied its queuing performance and compared it with that of the empirical trace.

The system used here is a single server first-in first-out (FIFO) queue with infinite buffer size. Our synthetic traffic was used as a source traffic to the single server queue. The performance is compared with the same system using the empirical trace as the source traffic. A single arrival process is assumed in our simulation, and its service rate is assumed to be constant. We conducted the simulation on video trace Atp. Fig. 14 shows the average queue size using the synthesized data generated by our proposed simple model and the empirical trace. From Fig. 14, there is a little difference in the average queue size for both synthesized data and empirical trace with different service rates. Thus, our simple model is accurate enough for the purpose of evaluating network performance.

IX. Conclusions

In this paper, we have proposed a simple traffic model for MPEG coded video streams. The simple model, which consists of three self-similar processes, is successfully fitted to different empirical video sequences. Simulation results showed that not only the ACF of video traffic can be captured accurately, but 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 CDF of video traffic, but the traffic generated by this model is also more similar to the empirical trace. The queuing performance in a single FIFO system with different service rates using our proposed model is compatible with that using empirical data. Thus our proposed simple IPB composite model, which is rather accurate, can be adopted in network simulations.

Appendix

The following is the algorithm for generating the F-ARIMA process [6], [16]:

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

2) For $k = 1, 2, \ldots, N-1$, calculate ϕ_{kj}, $j = 1, 2, \ldots, k$ iteratively using the following formulae

$$N_k = r(k) - \sum_{j=1}^{k-1} \phi_{k-1,j} r(k-j) \tag{8}$$

$$D_k = D_{k-1} - N_{k-1}^2 / D_{k-1} \tag{9}$$

$$\phi_{kk} = N_k / D_k \tag{10}$$

$$\phi_{kj} = \phi_{k-1,j} - \phi_{kk}\phi_{k-1,k-j}, \quad j = 1, \ldots, k-1 \tag{11}$$

$$m_k = \sum_{j=1}^{k} \phi_{kj} X_{kj} \tag{12}$$

$$\nu_k = (1 - \phi_{kk}^2)\nu_{k-1}. \tag{13}$$

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, the autocorrelation function can be calculated in a recursive way as

$$r(0) = 1, \qquad r(k+1) = \frac{k+d}{k+1} r(k) \tag{14}$$

where $d = H - 0.5$.

ACFs of F-ARIMA and FGN generated traffic are less than $k^{-\beta}$ for small k. To compensate for the under-estimation of ACFs of a self-similar process, (14) used to generate the 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 data to be Beta distributed, the following mapping can be used

$$Y_k = F_\beta^{-1}(F_N(X_k)), \quad k > 0 \tag{15}$$

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.

References

[1] V. Paxson and S. Floyd, "Wide-area traffic: The failure of Poisson modeling," in *Proc. ACM SIGCOMM'94*, London, U.K., Aug. 1994, pp. 257–268.

[2] P. Skelly, M. Schwartz, and S. Dixit, "A histogram-based model for video traffic behavior in an ATM multiplexer," *IEEE/ACM Trans. Networking*, vol. 1, no. 4, pp. 446–459, Aug. 1993.

[3] D. P. Heyman and T. V. Lakshman, "Source models for VBR broadcast-video traffic," *IEEE/ACM Trans. Networking*, vol. 4, no. 1, pp. 40–48, Feb. 1996.

[4] B. Maglaris *et al.*, "Performance models of statistical multiplexing in packet video communications," *IEEE J. Select. Areas Commun.*, vol. 36, no. 7, pp. 834–844, Jul. 1988.

[5] B. Melamed and D. E. Pendarakis, "Modeling full-length VBR video using Markov renewal modulated TES models," *IEEE J. Select. Areas Commun.*, vol. 16, pp. 600–611, Jun. 1997.

[6] M. W. Garrett and W. Willinger, "Analysis, modeling and generation of self-similar VBR video traffic," in *Proc. ACM SIGCOMM'94*, London, U.K., Aug. 1994, pp. 269–280.

[7] M. M. Krunz and A. M. Makowski, "Modeling video traffic using $M/G/\infty$ input processes: A compromise between Markovian and LRD models," *IEEE J. Select. Areas Commun.*, vol. 16, no. 5, pp. 733–749, Jun. 1998.

[8] D. P. Heyman, "The GBAR source model for VBR video-conferences," *IEEE/ACM Trans. Networking*, vol. 5, no. 4, pp. 554–560, Aug. 1997.

[9] O. Rose, "Simple and efficient models for variable bit rate MPEG video traffic," *Performance Evaluation*, vol. 30, pp. 69–85, 1997.

[10] M. Krunz and S. K. Tripathi, "On the characterization of VBR MPEG streams," in *Proc. SIGMETRICS'97*, Cambridge, MA, Jun. 1997, pp. 192–202.

[11] H. Liu, N. Ansari, and Y. Q. Shi, "Models of MPEG coded video traffic: Sequentially modulated self-similar process," in *Proc. IFIP Broadband Communications'99*, Hong Kong, Nov. 10–12, 1999, pp. 63–72.

[12] ——, "Markov-modulated self-similar processes: MPEG coded video traffic modeler and synthesizer," in *Proc. Globcom'99*, Rio de Janeiro, Brazil, Dec. 1999, pp. 1184–1188.

[13] ——, "A simple model for MPEG video traffic," in *Proc. of IEEE Intel. Conf. Multimedia and Expo*, vol. 1, New York City, Jul. 31–Aug. 2, 2000, pp. 553–556.

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

[15] H. E. Hurst, "Long-term storage capacity of reservoirs," *Trans. Am. Soc. Civil Eng.*, vol. 116, pp. 770–799, 1951.

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

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

[18] S. Q. Li and H. D. Sheng, "Queue response to input correlation functions: Discrete spectral analysis," *IEEE/ACM Trans. Networking*, vol. 1, pp. 522–533, Oct. 1993.