Modeling Video Sources for Real-Time Scheduling

Aurel A. Lazar, Giovanni Pacifici, Dimitrios E. Pendarakis

Department of Electrical Engineering
and
Center for Telecommunications Research
Columbia University, 500 West 120th Street, New York, NY 10027-6699

Abstract

What is the impact of autocorrelation of Variable Bit Rate (VBR) video sources on real-time scheduling algorithms? Our results show that the impact of long term, or interframe, autocorrelation is negligible, while the impact of short term, or intraframe, autocorrelation can be significant. Such results are essentially independent of the video coding scheme employed. To derive these results, we introduce a model that is based on statistical analysis performed on actual video data. Our model accurately captures the distribution and the autocorrelation function of the source bit stream on both the frame and the slice level. We show that the original video data sequence can be modeled as a collection of stationary subsequences called scenes. Within a scene, a model is derived for both the sequence of frames and of slices. In previous work at the slice level, the pseudo-periodicity of the autocorrelation function made it difficult, to develop a simple yet accurate model. One of the new elements introduced in this work is that we present a generalization of previous methods, that can easily capture this pseudo-periodicity and is suited for modeling a greater variety of autocorrelation functions. The generality of our model lies in that, by simply tuning a few parameters, it is able to reproduce the statistic behavior of sources with different types and levels of correlation.

1 Introduction

Video traffic is expected to be one of the major sources of loading of broadband networks, along with voice and data traffic. Variable Bit Rate (VBR) video traffic will impose very stringent real-time constraints on the network, especially in the case of interactive services. Since these traffic streams have a complex structure, their effect on the network may be much more complex than that found by simple, analytically tractable traffic models.

A first model of VBR video traffic appears in [1] where a video source is described as a first-order autoregressive process AR(1) with marginal pdf Gaussian and an exponential autocorrelation function. An ARMA process followed by a memoryless nonlinear filter was used in [2]. The method is able to match the mean, variance and autocorrelation function but not the marginal pdf of the sequence. More work on the distribution of frame sizes appears in [3] and [4].

In [5] the authors propose a new methodology, called TES, for modeling video sources. The main characteristic of this method is that it can generate an arbitrary distribution for the number of bits in a frame as well as model the frame correlation structure.

Our work was motivated by the need to evaluate the performance of a multiplexer with various schedulers, when loaded with VBR video traffic. Several questions related to this problem were asked: What timing aspects of the video are impacting the performance of the multiplexer? How can this be evaluated?

In order to answer these questions, accurate video models were needed, in particular to predict queueing behavior. Development of models that are accurate, yet as simple as possible, was necessary since availability of actual video data is still limited. Our motivation was to construct parametric models that offer increased flexibility in reproducing the statistical behavior of different types of video sources, thus allowing a greater range of experimental results.

The novel element of our modeling approach is the introduction of a new model for VBR video traffic that captures the behavior on all three time scales: *scene*, *frame* and *slice*. This model is based on statistical analysis that we performed on actual video data that belongs to the entertainment movie StarWars. We show that the original video data sequence can be modeled as a collection of stationary subsequences that we call *scenes*. We further show that the scene changes can be modeled as a Bernoulli process. Within a scene, a model is derived for both the sequence of frames and of slices.

We use this VBR video model to explore the performance of a link scheduler associated with an ATM node. Our results show that for real-time scheduling the impact of long term or interframe correlation is negligible, while the impact of short term or intraframe correlation is significant. The basic reason this result is true, is that the duration of a frame is large enough compared to the QOS delay requirements in the network, to render correlation at the frame level insignificant.

This paper is organized as follows: In section 2 we describe the experimental video data used and present the relevant statistics calculations. In section 3 we introduce our model of the video data. In 4 we discuss multiplexing of the the video sources on a link scheduler, while section 5 presents the results describing the performance of the link scheduler. Finally, in section 6 we present our conclusions and directions for future study.

2 Experimental Real Time Video Data

The actual data that we have used was provided by Dr. Mark W. Garrett of Bellcore. This data set represents the bandwidth output of a simple variable bit rate (VBR) coder. The source material is based on the movie "Star Wars".

The data set includes the information content in bytes per frame for approximately 2 hours of video. The frame rate is 24 per second and each frame is further broken down to the 30 slices, each slice consisting of 16 video lines. To allow computation of a complete movie, the coding algorithm employed is a simplified DCT, similar but not identical to JPEG. For a more detailed description of this data set, see [6].

The autocorrelation of the sequence of the frames sizes is displayed in figure 1. It decreases slowly as the lag increases (thereby showing a shortcoming of the models proposed in [1, 2] that do not reproduce this strong correlation between successive frames with large lags).

Figure 2 shows the autocorrelation function for the sequence of slice sizes.As expected, a pseudo-periodicity of 30 slices can be observed. The existence of this pseudo-periodicity is one of the main difficulties in modeling the autocorrelation function on the slice level, as will be explained in section 3.3.

3 Modeling VBR Traffic Streams

This section presents our models for VBR video sources on both the frame and slice time scale. In section 3.1 we show that the sequence of both frames and slices can be modeled as a collection of stationary subsequences called scenes. We state the conditions that identify a scene change and derive the distribution of scene durations. In sections 3.2 and 3.3, we present the frame and slice level models, respectively. The segmentation of sequences into scenes is embedded in both levels. The TES method, mentioned in section 1, is used in both models. As already mentioned however, for previous work at the slice level, the pseudo-periodicity of the autocorrelation function made it difficult to derive a simple, yet accurate model. We therefore introduce a new method, which is a generalization of the TES method that enables us to easily capture the pseudo-periodicity of the autocorrelation function and is suited for modeling a greater variety of autocorrelation functions.

3.1 Modeling VBR Video Sequences as a Collection of Scenes

By viewing the sample path of the original video sequence one observes that the sequence of frame sizes (F_n) consists of segments of variable length, that expose high correlation. Within each of these segments, sizes are very close in value. The same is true for the sequence of slice sizes, (S_m).

As can be easily verified, at the boundaries of these segments large changes in the magnitude of the frame or slice sizes occur. These segment boundaries coincide in the time sequence of both frames and slices. Intuitively one expects these segments to correspond to different *scenes* of the movie, i.e., portions of the movie without shifts of view supplied by editing or abrupt camera panning or zooming. By observing the actual movie we assessed that such scene changes do indeed coincide with large changes in the magnitude of frame or slice sizes.

Our analysis shows that scene lengths follow a geometric distribution, with parameter

$$p = \frac{1}{1 + \mathbb{E}(L_n)} \ , \qquad (1)$$

where L_n denotes the scene duration series [7].

3.2 Modeling the Bit Stream at the Frame Level

The classic TES method [8] is used to statistically reproduce the autocorrelation functions of empirical data, while guaranteeing an exact match of the marginal distribution. A TES method defines a random walk on the circular unit interval. This walk can be equivalently expressed as a sequence of modulo-1 arithmetic operations, i.e.,

$$U_n = U_{n-1} \oplus V_n \ , \qquad (2)$$

where (V_n) denotes a sequence of i.i.d. random variables with common density f_V [8].

In our case, we can express the innovation process (V_n) as

$$V_n = (1 - X_n)(L + (R - L)Z_n) + X_n \left(\frac{-\alpha_c}{2} + \alpha_c Z_n \right) \ , \qquad (3)$$

where $R = -L = \frac{\alpha_s}{2}$, (Z_n) is a sequence of i.i.d. random variables with marginals in $[0, 1)$, X_n is sequence of i.i.d. Bernoulli variables with $Pr\{X_n = 1\} = p$, p is the probability of having a scene change at step n and α_c, α_s are two parameters that affect the correlation of the U_n's. In particular, smaller values of α_s give rise to higher autocorrelation within a scene, while smaller values of α_c give rise to higher autocorrelation during scene changes.

We found that the parameters $\alpha_c = 0.28$, $\alpha_s = 0.02$ and $\mathbb{E}(L_n) = 100$ best matched our original VBR video data. Figure 1 plots the autocorrelation function of the generated sequence and the original video data up to lag 500.

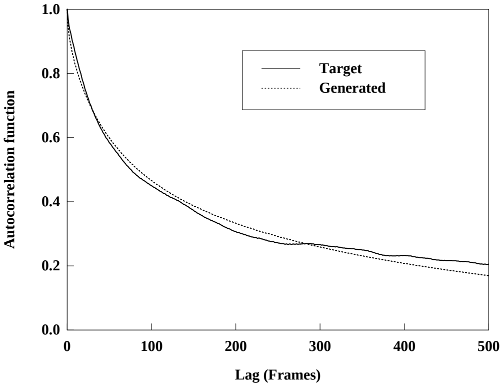

Figure 1: Autocorrelation function of the frame sizes for the original sequence and the match generated with parameters $\alpha_c = 0.28$, $\alpha_s = 0.02$, $\mathbb{E}(L_n) = 100$.

3.3 Modeling the Bit stream at the Slice Level

As already mentioned in section 2 the autocorrelation at the slice level exposes a pseudo-periodicity of 30 slices. This is expected, since in the sequence of slice sizes, slices that differ by 30, belong to the same area of different frames. This implies that the time sequence of slice sizes exposes two forms of autocorrelation: temporal, which is mainly interframe, and spatial or intraframe.

Our approach models *directly* the slice bit stream by introducing a generalization of the TES method, namely allowing the innovation function to be modulated by a function of time:

$$U_n = U_{n-1} \oplus a(n) V_n \ , \qquad (4)$$

where, once again, V_n denotes a sequence of i.i.d random variables with common density f_v. It still can be shown that equation 4 gives rise to a sequence (U_n) of correlated identically distributed random variables with uniform marginals in $[0, 1)$ [7]. To give some intuition for this modification, we describe the function $a(n)$ used in modeling the slice sizes:

$$a(n) = \begin{cases} 1 & \text{if } 0 \le n_{mod\ s} \le \frac{s}{2} - 1, \\ -1 & \text{if } \frac{s}{2} \le n_{mod\ s} \le s - 1, \end{cases} \qquad (5)$$

where, in our case, $s = 30$.

Intuitively, the random walk on the unit circle proceeds in one direction for a duration equal to half of the frame size and to the opposite direction for the other half. This results in a high correlation every 30 slices. The modulating function $a(n)$ could also take more complex forms, making it possible to model a wide range of autocorrelation functions, but for our purposes we have found the function $a(n)$ given in equation 5 to be sufficient.

The innovation function V_n of equation 4 is again given by equation 3. The probability of a scene change is given by equation 1 with the difference that the expected duration of a scene in number of *frames*, $\mathbb{E}(L_n)$, should be multiplied by

$s = 30$ to give the expected scene duration in number of *slices*, $\mathbb{E}(L_n^{'})$.

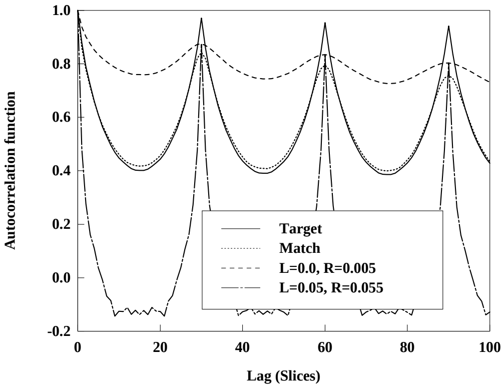

Figure 2: Autocorrelation function of the sequence of slice sizes for the original sequence, the match generated with parameters $L = 0.003$, $R = 0.008$, $a_c = 0.28$, $\mathbb{E}(L_n^{'}) = 3000$ and, autocorrelation functions with the same temporal, but different spatial correlation than the original sequence.

.

Small values of $R - L$ result in high temporal correlation, while small values of $R + L$ lead to high spatial and temporal correlation. By varying these two parameters we were able to match the autocorrelation function of the slice size sequence. The best matching of the autocorrelation function was found for $L = 0.003$ and $R = 0.008$. The parameters a_c and $\mathbb{E}(L_n)$ where kept at the values of the optimal match at the frame level. The autocorrelation function for these parameters of the modified TES process is compared in figure 2 with the autocorrelation of the original process.

In the same figure we plot the autocorrelation for two additional set of parameters. Both sets have the same value of step size $R - L$ and, hence, the same temporal autocorrelation. However, one corresponds to higher spatial autocorrelation than the original sequence and the other to lower.

4 Scheduling of Multiplexed Video Sources

In this and the next section, we will present the results of simulation experiments which illustrate the impact of the parameters of our VBR video models, on the performance of a class of integrated networks with the capability of *efficiently* providing quality of service (QOS). These networks are called Asynchronous Time Sharing (ATS) based because of the way the main network resources (switching and communication bandwidth, buffer space and processing capacity) are allocated [9].

In order to give a quantitative framework for evaluating the performance of networks that provide QOS guarantees the concept of *schedulable region* is used [10]. Informally, the schedulable region is the region in the space of possible loads for which a scheduling algorithm guarantees quality of service.

4.1 The Architecture and Framework for ATS

We consider a network with quality of service guarantees, based on the Asynchronous Time Sharing principle. Four classes of traffic are supported. Three of the traffic classes, Class I, II and III, transport user traffic and are defined by a set of performance constraints. The fourth class, Class C, transports traffic of the network management system, and is not subject to specific QOS constraints.

Class I traffic is characterized by 0 % *contention cell loss* and a maximum end-to-end time delay between the source and destination, denoted by S^I. Class II traffic is characterized by ϵ % contention cell loss. It is also characterized by an end-to-end time delay distribution with a larger support than Class I. The maximum end-to-end time delay is S^{II}. In general, $S^I \leq S^{II}$. For Class I and II traffic, there is no retransmission policy for lost cells. Class III traffic is characterized by 0 % *end-to-end cell loss* that is achieved with an end-to-end retransmission policy for error correction. If requested, it is also characterized by a *minimum* average user throughput Γ and a *maximum* average user time delay T.

4.2 Real-Time Traffic Source Models

For the purposes of this section we consider each of the traffic classes, defined via Quality of Service constraints in section 4.1, to carry information of a very specific type. Class I is assumed to consist of K^I video calls, and Class II of K^{II} voice calls. Class III consists of K^{III} data sources.

For video sources, two cases are considered: transmission is done either on a frame by frame or on a slice by slice basis. Multiplexing of K^I video calls is assumed to be accomplished by uniformly distributing the frame (slice) start at intervals of T/K^I ms, where $T = T_f = 1/24s$ is the fixed frame duration or $T = T_s = 1/720$ is the fixed slice duration, respectively. For more details see [7].

A single voice call is modeled as an on-off source with constant arrivals. It is characterized by an exponentially distributed active period, in which cells are generated with constant rate $c^{II}/(L_h + L_p)$ with $c^{II} = 64$ Kbit/s, and an exponentially distributed silence period, in which no cells are generated [7].

Finally, the data traffic is modeled as a Poisson source.

4.3 The Schedulable Region

The fundamental concept, on the basis of which all comparisons of performance will be based, is that of the *schedulable region* [10]. Intuitively, the *schedulable region* of a queueing system is the set of points in the space of possible loads for which the quality of service is guaranteed. The size of the schedulable region depends on the scheduling algorithm used, the values of the QOS parameters and the statistics of the traffic load.

For the sake of simplicity we will assume that Class C traffic is negligible and, therefore limit our present study of the schedulable region to a queueing system with three user classes (Class I, II and III). Finally, we will assume that each of the buffers has infinite capacity.

5 Experimental Evaluation of the Scheduling Algorithms

In this section, we present the results of simulation experiments which illustrate the impact of different VBR video traffic and some properties of the scheduling algorithms and the associated schedulable regions.

The scheduling algorithms that will be used are static priority scheduling (SPS) and MARS, the MAGNET II real-time scheduling algorithm [10]. When static priority scheduling (SPS) is employed, Class I cells are always transmitted ahead of Class II, and Class II cells are always transmitted ahead of those of Class III. This scheduling scheme is simple to implement, and is thus often considered for scheduling of real-time traffic. When MARS is employed, transmission resources are time-shared between traffic classes according to a cycle scheme [9].

MARS adaptively sets the parameters which govern this cycle scheme, based on observations of cell arrivals and departures. The scheduling algorithm is based on the intuition that in order to achieve high throughput, each cycle should serve only the cells whose transmission cannot be further delayed to satisfy the QOS requirements. For details on this scheduling algorithm see [10].

All of the simulations reported in this section are based on the assumptions of a fixed cell size of $(L_p+L_h) = 1024$ bits/cell, a fixed cell header of $L_h = 98$ bits/cell and a link capacity of $C = 100$ Mbit/s. In addition, for every simulation run a transient period of 2 seconds was allowed before any measurements were taken, and the simulations were run for 180 seconds. The 95 % confidence bounds for the measured performance criteria were well within 5 % of the observed values.

To obtain each point in the plots, the values of K^I and K^{III} (the number of calls of Classes I and III) were fixed, and simulations were run to determine the maximum number K^{II} of voice calls for which the QOS could be satisfied over the whole period of 180 seconds. For simplicity and in order to facilitate the interpretation of our results, only the two dimensional projections of the schedulable regions onto the plane $K^{III} = 0$ are plotted. K^I was varied from 0 to 20 video calls in steps of 1.

5.1 Performance Results for Video Sources Multiplexed on the Frame Level

When we compared the schedulable region obtained by multiplexing parts of the original sequence to that obtained by multiplexing Class I calls with the statistics of the matching model that was derived in section 3.2, we obtained a very satisfactory match. We then plotted the schedulable regions obtained for Class I sources corresponding to two different models, one with high and one with low autocorrelation. In both models the average scene length $\mathbb{E}(L_n) = 100$ and $\alpha_c = 0.30$. The QOS vector used was $[S^I, S^{II}, \varepsilon, T] = [2ms, 2ms, 0.005, 1ms]$.

What we observed is that the difference in the size of the schedulable regions is negligible [7]. Only when most of the load is provided by Class I traffic does a very small difference appear, but it is still too small to be considered significant. We repeated these experiments for a wide range of model parameters and QOS parameter values, without being able to record significant differences in the schedulable regions. This result can be attributed to the fact that *the duration of a frame is large enough compared to the delay requirements in the network.* Specifically, the duration of a frame is 1/24sec=41.66msec, which is very much larger than the QOS parameters S^I and S^{II} that we have used. If one attempts to increase these parameters to the point where they are of the same order of magnitude as the frame duration, utilization of the system approaches 100%, rendering any difference due to autocorrelation, again, negligible.

5.2 Performance Results for Video Sources Multiplexed on the Slice Level

In this set of experiments the QOS vector used is $[S^I, S^{II}, \varepsilon, T] = [2ms, 2ms, 0.005, 1ms]$.

Figure 3 compares the schedulable region obtained by multiplexing parts of the original sequence to that obtained by multiplexing Class I calls with the statistics of the matching model that was derived in section 3.3. The schedulable regions are found using MARS scheduling. As one can observe, the matching is very satisfactory. The fact that it cannot be considered perfect can be attributed to the fact that the original sequence cannot be considered as stationary. On the contrary, our data shows that the sequence consists of parts with higher activity and parts with lower activity. In any case, the very good match of the schedulable regions is the best verification of our model in terms of performance criteria.

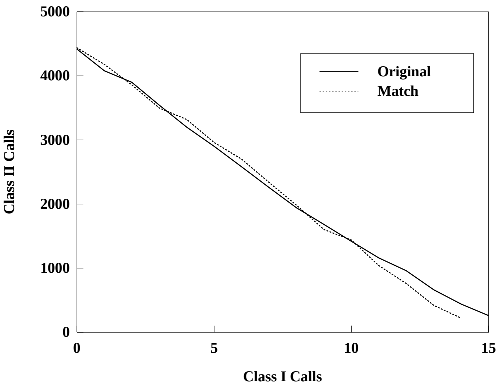

Figure 3: Comparison of the schedulable regions as obtained from the original sequence and from the matching model respectively, for MARS scheduling and multiplexing at the slice level.

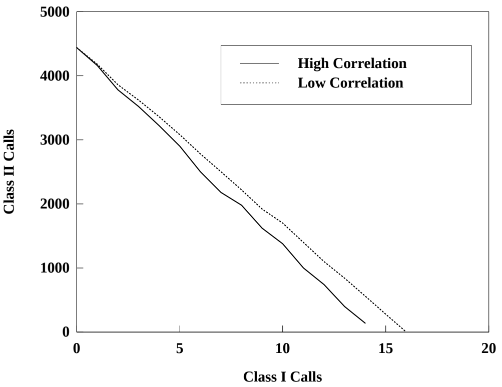

Figure 4: Comparison of the schedulable regions as obtained for video sources with high and low autocorrelation respectively, for MARS scheduling, and multiplexing at the slice level.

In our next experiment we investigated the impact of the autocorrelation function on the size of the schedulable region. The results are plotted in figure 4, where the schedulable region is plotted for two different sets of video source parameters. The two sets are the same as the ones shown in figure 2 and correspond to low and high autocorrelation, respectively. It is evident that the increase in the autocorrelation function of Class I sources has a significant impact on the size of the schedulable region; the reduction being of the order of 10% in the number of Class II sources that can be scheduled. One should observe the intuitive result that the difference increases with the number of Class I sources present in the traffic mix.

The last two experiments are repeated for SPS scheduling. The results are shown in figure 5. In addition to the well established fact that the schedulable region for MARS is larger than the respective one for SPS, one can make some interesting observations. First, the results are qualitatively similar. In particular the match in the schedulable regions between the model and the original sequence is again very satisfactory, providing further evidence of the general validity of our model. On the other hand, SPS proves to be less sensitive to autocorrelation than MARS. The intuitive explanation of this result is that MARS is designed to increase the schedulable region by serving Class I cells only when their transmission cannot be further delayed. As such, this mechanism is more effective when traffic of Class I is more “random”, allowing the scheduler to delay cells of Class I and accommodate more Class II calls. When

Class I traffic is highly correlated, there will be periods when the best a scheduler can do is yield priority to Class I, thus bringing the performance of MARS and SPS closer.

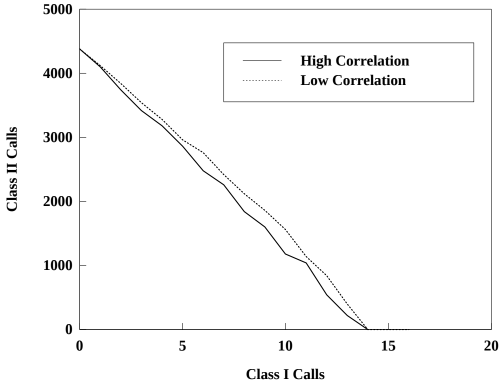

Figure 5: Comparison of the schedulable regions as obtained for video sources with high and low autocorrelation respectively, for SPS scheduling, and multiplexing at the slice level.

6 Conclusions

In this paper we have addressed both the modeling problem for real time video sources and the performance of scheduling systems loaded with multiplexed video sources.

We have presented models that can accurately match the behavior of the video bit stream both at the frame and the slice level. At the frame level the focus is on modeling the sample path of the video source stream as an aggregation of scenes, representing different levels of activity. At the slice level the emphasis was on modeling the pseudo-periodicity that the slice bit rate sequence exposes.

Our results clearly indicate that the magnitude of this impact depends on the time scale on which video transmission will be done in ATM networks.

On the frame level, we have shown that the time distance between successive frames is too large for the autocorrelation function to have a significant impact. It should be pointed out though, that these results hold in the case of *real-time* video transmission. If this is not the case, as for example in remote image retrieval systems, or in the case of off-line video transmission between two distant sites, autocorrelation might have a different impact.

The results for transmission at the slice level are quite different. Since the time scale of video transmission in this case is comparable to that of QOS delay requirements, spatial or intraframe autocorrelation has a significant impact, which increases with the percentage of video traffic in the total traffic mixture. On the other hand, temporal, or interframe autocorrelation does not have a significant effect.

Our results suggest possible ways of transmitting video over ATM networks, so that correlation is decreased and the number of sources simultaneously accommodated increased: One could for example permute the order of slices transmitted either randomly or deterministically in such a way that neighboring slices of a frame are sufficiently separated in time from each other. Such a transmission scheme is indeed currently under investigation.

The raw data that we used for our modeling purposes was obtained with a coding algorithm similar but not identical to JPEG. The proposed MPEG standards will probably expose different traffic characteristics. However, we strongly believe that our results can be easily applied to the case of video traffic produced from MPEG encoders. This is so because we showed that the impact of long term correlations on real-time scheduling systems is negligible, while short term correlations can be of significance. It is natural to expect that this result will apply, regardless of the coding scheme used. On the slice level in particular, all existing work points out to similar autocorrelation functions as the one that we modeled [11]. In addition the models that we introduced here can be useful in the study of any future coding scheme, since they allow for reproduction of sequences with a wide range of distributions and autocorrelation functions by just tuning a few parameters.

7 Acknowledgement

The work reported here was supported by the National Science Foundation under Grant # ECD-88-11111.

References

[1] B. Maglaris, D. Anastasiou, P. Sen, G. Karlsson, and J. Robbins, "Performance models of statistical multiplexing in packet video communications," *IEEE Transactions on Communications*, vol. Com-36, pp. 834–844, July 1988.

[2] R. Gruenefelder, J. P. Cosmas, S. Manthrope, and A. Odinma-Okafor, "Characterization of video codecs as autoregressive moving average processes and related queueing system performance," *IEEE Journal on Selected Areas in Communications*, vol. 9, pp. 284–293, April 1991.

[3] P. Skelly, S. Dixit, and M. Schwartz, "A histogram-based model for video behavior in an ATM network," in *Proceedings of the IEEE INFOCOM*, (Florence, Italy), pp. 95–105, May 1992.

[4] D. P. Heyman, A. Tabatabai, and T. Lakshman, "Statistical analysis and simulation study of video teleconference traffic in ATM networks," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 2, pp. 49–59, March 1992.

[5] D.-S. Lee, B. Melamed, A. Reibman, and B. Sengupta, "Analysis of a video multiplexer using tes as a modelling methodology," in *Proceedings of GLOBECOM*, (Phoenix, Arizona), pp. 1.3.1–5, December 1991.

[6] M. W. Garrett and M. Vetterli, "Congestion control strategies for packet video," in *Proceedings of Fourth International Workshop on Packet Video*, (Kyoto, Japan), August 1991.

[7] A. A. Lazar, G. Pacifici, and D. E. Pendarakis, "Modeling of video sources for real time scheduling," CTR Technical Report 324-93-03, Center for Telecommunications Research, Columbia University, New York, February 1993.

[8] B. Melamed, "TES: A class of methods for generating autocorrelated uniform variates," *Operations Research Society of America Journal on Computing*, 1991.

[9] A. A. Lazar, A. T. Temple, and R. Gidron, "An architecture for integrated networks that guarantees quality of service," *International Journal of Digital and Analog Communication Systems*, vol. 3, pp. 229–238, April–June 1990.

[10] J. M. Hyman, A. A. Lazar, and G. Pacifici, "Real-time scheduling with quality of service constraints," *IEEE Journal on Selected Areas in Communications*, vol. 9, pp. 1052–1063, September 1991.

[11] P. Pancha and M. E. Zarki, "A look at the MPEG video coding standard for variable bit rate transmission," in *Proceedings of the IEEE INFOCOM*, (Florence, Italy), pp. 85–94, May 1992.