Modeling Full-Length Video Using Markov-Modulated Gamma-Based Framework

Uttam K. Sarkar, Subramanian Ramakrishnan, and Dilip Sarkar, *Senior Member, IEEE*

***Abstract*—All traffic models for MPEG-like encoded variable bit rate (VBR) video can be broadly categorized into 1) data-rate models (DRMs) and 2) frame-size models (FSMs). Almost all proposed VBR traffic models are DRMs. DRMs generate *only* data arrival rate, and are good for estimating average packet-loss and ATM buffer overflowing probabilities, but fail to identify such details as percentage of frames affected. FSMs generate sizes of individual MPEG frames, and are good for studying frame loss rate in addition to data loss rate. Among three previously proposed FSMs: 1) one generates frame sizes for full-length movies without preserving group-of-pictures (GOP) periodicity; 2) one generates VBR video traffic for news videos from scene content description provided to it; and 3) one generates frame sizes for full-length movies without preserving size-based video-segment transitions. In this paper, we propose two FSMs that generate frame sizes for full-length VBR videos preserving both GOP periodicity and size-based video-segment transitions.**

First, two-pass algorithms for analysis of full-length VBR videos are presented. After two-pass analysis, these algorithms identify size-based classes of video shots into which the GOPs are partitioned. Frames in each class produce three data sets, one each for I-, B-, and P-type frames. Each of these data sets is modeled with an axis-shifted Gamma distribution. Markov renewal processes model (size-based) video segment transitions. We have used QQ plots to show visual similarity of model-generated VBR video data sets with original data set. Leaky-bucket simulation study has been used to show similarity of data and frame loss rates between model-generated VBR videos and original video. Our study of frame-based VBR video revealed that even a low data-loss rate could affect a large fraction of I frames, causing a significant degradation of the quality of transmitted video.

***Index Terms*—Frame-size traffic model, Gamma distribution, leaky-bucket simulation, MPEG, QQ plot, variable bit rate (VBR) video.**

I. Introduction

VIDEO traffic is expected to be the major source for broadband integrated services digital networks (B-ISDN) [11], [1]. Because of large bandwidth requirement for communication of high-quality uncompressed video over B-ISDN, it is expected that most, if not all, video will be encoded with MPEG-like data compression techniques [3]. These compression algorithms can provide very high compression ratio while maintaining good quality of decompressed video. However, MPEG-coding provides different amount of compression for different frames (see [3, Fig. 1 and Table 1] for examples) and results in variable bit rate (VBR) data, known as VBR video.

The MPEG standard permits *intra* (I), *predicted* (P), and *bidirectional* (B) encoding of frames. A group-of-pictures (GOP) structure of (N, M) cyclic format, composed of N frames, starts from an I frame and ends before the next I frame; every Mth frame is a P frame, and $(M-1)$ frames between every I–P, P–P, or P–I frames pair are B frames. For instance, if $N = 6$ and $M = 3$, a (6, 3) GOP structure with IBBPBB sequencing of frames results. Encoding of an I frame being independent of other frames results in a low compression ratio but provides a point of access. Encoding of a P frame using motion-compensated prediction of (most recent) previous I or P frame provides usually a higher compression ratio than that of I frames. Encoding of a B frame using bidirectional prediction based on nearest pair of past and future I–P or P–P or P–I frames provides usually the highest compression ratio compared with I and P frames. While the MPEG standard permits use of many GOP structures for encoding a video, *typically only one GOP structure is used for encoding all frames of a full-length video*. Thus, a cycle or period of N frames appears in an MPEG-coded VBR video.

Accurate traffic models of VBR video are necessary for prediction of performance of any proposed (and/or designed) B-ISDN during its operation. Several traffic models have been proposed in the literature. They include first-order autoregressive (AR) [13], discrete AR (DAR) [6], [10], [16], Markov renewal process (MRP) [2], MRP transform-expand-sample (TES) [12], finite-state Markov chain (MC) [1], [6], [14], [16], Gamma-beta auto-regression (GBAR) [4], and GOP GBAR [3] models.

A. Two Categories of Traffic Models

These traffic models can broadly be classified into two categories: 1) data-rate models (DRMs) and 2) frame-size models (FSMs). In a data-rate model, only the rate at which data are arriving at a link is generated for performance prediction purpose. Almost all models, including AR, DAR, MRP, MRP TES, MC, GBAR, and models in [5] and [7], fall under this category. These models are good for predicting average packet-loss probability and ATM buffer overflowing probability. However, they fail to identify such details as percentage of frames affected. Even a small rate of data loss involving I frames may affect perceptual quality of received video significantly, but the same amount of data loss in B frames would have far less impact. (For an example, see Section I-B.)

Manuscript received April 30, 2001; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor T. V. Lakshman.

U. K. Sarkar was with the Department of Computer Science, University of Miami, Coral Gables, FL 33124 USA. He is now with the Indian Institute of Management Calcutta, Joka, Calcutta 700104, India (e-mail: uttam@iimcal.ac.in).

S. Ramakrishnan is with the Department of Mathematics, University of Miami, Coral Gables, FL 33124-4250 USA (e-mail: ramakris@math.miami.edu).

D. Sarkar is with the Department of Computer Science, University of Miami, Coral Gables, FL 33124-4245 USA (e-mail: sarkar@cs.miami.edu).

Digital Object Identifier 10.1109/TNET.2003.815292

In a frame-size model, sizes of individual MPEG frames are generated, and hence, data-rate information can be obtained from the frame-size information. Moreover, during loss performance modeling, location of lost data can be precisely identified for getting a better understanding of the quality of video at the receiving end. The inherent frame-by-frame burst nature of MPEG videos are preserved in these category of models. Models reported in [2], [3], and [9] fall under this category. Compared with the number of DRMs, only few FSMs have been proposed. It is believed that the main obstacle in the development of an FSM is to find some standard statistical distribution fitting different frame types.

Data rates of short-length videos with small variation between frames can be modeled by AR models. The GBAR model, being an autoregressive model with Gamma-distributed marginals and geometric autocorrelation, captures data-rate dynamics of VBR video conferences well. However, it is not suitable for general MPEG video sources [3]. Data rates of videos of longer lengths with scene variations have been modeled with DAR, MRP, and MC. All these models were motivated to generate traffic load for the study of cell-loss rate when multiple videos are multiplexed over a broadband link in a B-ISDN. Empirical studies simulating leaky buckets have shown these models capture cell-loss properties of ATM networks quite closely. However, they do not capture periodicity or cyclic property present within each GOP of a typical MPEG video.

B. Frame-Size VBR Models

During transmission one needs to consider the fact that MPEG videos contain very little redundancy. Moreover, loss of a part of an I frame affects all frames in its GOP, but loss of a whole B frame affects only that frame. Thus, even an apparently small rate data loss may affect perceptual quality of received video significantly (or a large rate of loss of data may affect it very little). For instance, a 10^{-4} bit-loss ratio (BLR) may appear small enough to be acceptable, but consider an MPEG encoded video with 15 frames per GOP, and 10^{-4} BLR affecting only 1% frames of all video frames. Now, if all of those affected frames are I frames, it will affect 15% of the video, which is a perceptually significant degradation of video quality. Thus, for study of MPEG video transmission a frame-level model is essential to get the necessary details of the effect of loss during transmission.

Relatively fewer models have been proposed for modeling of full-length movies that preserve distributions of I, P, and B frames. The GOP GBAR model attempts to capture overall statistical properties of I, P, and B frames of MPEG movies. It does so by using three GBAR models for the generation of three random variables that have Gamma-distributed marginals and geometric autocorrelations. The B frames are obtained from one of these random variables. The P frames are the sum of two random variables and I frames are the sum of all three. However, this model does not attempt to capture shot-level regularity of sizes of I, P, and B frames, which is a typical characteristic of the contents of any long video or any full-length movie [2]. The model in [9] assumes that the change of a scene changes the average size of I frames, but not the sizes of P and B frames. Reference [2, Table IV] shows that the average sizes of P and B frames can vary 20% and 30%, respectively, which are statistically significant; our analysis of frames from a full-length video shows much wider variations (see Table I). Also, the sizes of the frames are drawn from log-normal distributions following [8] which are not quite good fits.[1] The model in [2] requires that the video to be modeled has been segmented into shots based on texture and motion. Moreover, it uses an AR model which has been reported to be unsatisfactory for full-length movies; for instance, see [3].

C. Outline of This Paper

In the next section, we present our full-length video analysis algorithms. Following content-based MPEG video traffic modeling reported in [2] and [15], our first objective is to partition video into clips. It is assumed that most, if not all, video clips are at least one GOP long. With this assumption, the first two steps in our analysis are as follows. 1) The sizes of all frames in a GOP are added to obtain a sequence of GOP sizes for a movie, and adjacent GOPs of similar size are combined, by a moving average method, to identify video clips. *We use only the size of GOPs and not the content type to group GOPs in clips.* A full-length movie generates thousands of video clips. Next, these video clips are grouped into a smaller number (seven, in our case studies) of classes. We use geometrically separated class-size boundaries. All frames of each type (I, B, or P) belonging to a class are separated to obtain a data set. This approach groups similar-sized I, B, and P frames together (see Section II-C). Each frame type in a class is modeled with an axis-shifted Gamma distribution. For our case study, a total of $3 \times 7 = 21$ Gamma distributions are used.

For analysis of transition between classes, adjacent clips in a class are merged together to form video segments. The lengths of video segments are modeled with Gamma and geometric distributions. The inter-class transitions of video segments are analyzed to construct two state-transition probability matrices.

In Section III, we propose two models, one for each inter-class transition matrix. For validation of these models, full-length *synthetic* movies are generated using the proposed models. In Section IV, two standard measures—QQ plot and leaky-bucket simulation—are used to show that the geometrically separated classification technique nicely captures the behavior of I, B, and P frames of all classes into Gamma distributions. This is significant, especially for I and B frames which had hitherto remained less amenable to mimicking any regular statistical distribution. Both QQ plots and the leaky-bucket simulation results show that our models are closer to the real movies as compared with other known models in the literature. Only the results for two full-length movies—*Crocodile Dundee* and *ET*—are reported in this paper. We discuss our observations, possible uses of proposed VBR video models, and future extensions of our work in Section V.

II. Analysis of VBR Video

This section outlines our two-pass analysis algorithms for extracting the parameters of a full-length VBR video. Let $F = F_1, F_2, F_3, \ldots F_{n_f}$ be the sequence of n_f frames obtained from MPEG encoding of a full-length video for

[1]It must be acknowledged that one distribution is yet to be found that fits all frames of either I or B types of a movie well.

modeling. Since estimation of DLR requires only the size of a frame and not the actual data, a frame for the modeling purpose is represented by its serial number, type—I, B, or P—and size in bytes after the MPEG encoding. While not required by the MPEG standard, the underlying GOP usually follows an (N, M) cyclic format in which the first frame of a GOP is an I frame, every Mth frame is a P frame, and the rest are B frames. For this study, it is assumed that a movie has been encoded with one GOP structure. The size of a GOP is the sum of the sizes of all N frames in the GOP. We denote successive GOPs by $G_1, G_2, \ldots, G_{n_g}$ for our reference in this paper. Without any loss of generality, we assume $n_f = N * n_g$, as the last few frames of an incomplete GOP which may remain at the end of a full-length movie with 175 000–200 000 frames can be ignored for the purpose of modeling without any noticeable impact on the model.

A. Formation of Clips

A *clip* of length k is any consecutive sequence of k GOPs, $k \geq 1$, that is, $G_{i+1}, G_{i+2}, \ldots, G_{i+k}$ for some i, $0 \leq i \leq (n_f - k + 1)$. We denote successive clips by $C_1, C_2, \ldots$ and a set of clips by C. We use the notation $G_i \in C_j$ to indicate G_i belongs to C_j, and *length* (C_j) to indicate the number of GOPs in C_j.

We group similar-size GOPs to obtain a set of video clips. During clip construction, let the average size of a GOP in the partially formed clip of length k starting with G_{i+1} be $clip\_avg = (\sum_{l=1}^{k} size(G_{i+l}))/k$. The next GOP, G_{i+k+1}, is included in the current partial clip if the size of G_{i+k+1} does not differ from $clip\_avg$ by more than a user-provided *threshold* value. The smaller the value of *threshold*, the smaller the *length* of each clip and, consequently, the higher the total number of clips formed. The choice of a value for *threshold* for the modeling is not very critical and we have observed that any value close to the average size of a B frame can be used with good results. For the results reported in this paper, the value of *threshold* was 4500 for *Crocodile Dundee*, whose average B-frame size is 4445.82 bytes, and 2000 for *ET* with average B-frame size 2003.14 bytes. The pseudocode below explains the method of clip formation.

```
Algorithm Clips_From_GOPs
Input
    G: {G_1, G_2, ...}, the set of GOP;
    threshold: user provided value;
Output
    C: {C_1, C_2, ...}, the set of clips;
begin
  j = 1; /*index of the current clip*/
  C_1 = {G_1};
    /*C_1, the current clip starts with G_1*/
  clip_avg = size(G_1);
    /*average size of GOP in current clip*/
  for (i = 2; i ≤ n_g; i++){
    if abs(clip_avg − size(G_i)) ≤ threshold {
      /*continue expanding the current clip*/
      C_j = C_j ∪ {G_i};
        /*expand C_j by inserting G_i into it*/
      clip_avg = (∑_{G_k∈C_j} size(G_k))/length(C_j);
        /*recompute clip_avg for C_j*/
    } else {
      /*begin a new clip with G_i*/
      j = j + 1;
      C_j = {G_i};
        /*the new clip C_j starts with G_i*/
      clip_avg = size(G_i);
        /*initialize clip_avg for C_j*/
    }
  } /*for */
end;
```

The next step in our VBR video data analysis, outlined below, is partitioning of the clips into *shot classes*.

B. Formation of Shot Classes From Clips

A *shot class* of length k, $k \geq 1$, is a union of k *distinct*, but not necessarily consecutive, clips. We represent shot classes by $S_1, S_2, \ldots$ and $C_j \in S_i$ denotes all GOPs belonging to clip C_j belong to shot class S_i. Every clip belongs to one and only one shot class.

We construct the shot classes by partitioning the entire range of GOP sizes into the desired number n ($n = 7$ for the studies reported) subintervals, one for each shot class. We experimented with several partitioning methods and found that a geometric partitioning (explained below) results in a statistically significant number of GOPs in each partition such that each partition is amenable to a statistical modeling.

We have observed that the presence of a few *too small* and *too large* GOPs introduces undesirable biases in any model unless those extreme sizes are treated more like exceptions. In our algorithm, the smallest and the largest 1-percentile GOPs are initially set aside as being too extreme. The GOP sizes corresponding to these 1 and 99 percentile points are referred to by variables a and b, respectively, in the pseudocode below. The remaining interval of GOP sizes, namely, $[a, b]$, is partitioned into n subintervals. The successive partitioning boundaries of these intervals increase in a geometric progression with a as the first term and $b = ar^n$ as the $(n+1)$th term. This results in n subintervals $[a, ar]$, $[ar, ar^2]$, …, $[ar^{n-1}, ar^n]$ where $r = e^{((\ln b - \ln a)/n)}$ is the common ratio of the progression. The first subinterval is now extended to the left to include the range of 1% *small* GOP sizes initially set aside. Similarly, the last subinterval is extended to the right. A clip is now made to belong to a shot class if the average size of a GOP in this clip falls in the interval for that shot class. It may be noted that a partitioning with subintervals of equal length produced rather unsatisfactory results. Moreover, very few GOPs would then belong to shot classes which correspond to large average GOP sizes, making any statistical observation less meaningful.

The algorithm Get_Shot_Classes shows how shot classes are formed.

```
Algorithm Get_Shot_Classes
Input
    G: {G_1, G_2, ...}, the set of GOPs;
    C: {C_1, C_2, ...}, the set of clips;
    n: number of shot classes;
```

Output
S: $\{S_1, S_2, \ldots, S_n\}$, *the set of* n *shot classes*
begin
$a = 1$ *percentile value of* $\{size(G_i), 1 \le i \le n_g\}$;
/*GOPs smaller than a are too small*/
$b = 99$ *percentile value of* $\{size(G_i), 1 \le i \le n_g\}$;
/*GOPs bigger than b are too big*/
compute r *such that* $ar^n = b$;/*$r = e^{((\ln b - \ln a)/n)}$;
common ratio of shot class boundaries.*/
$interval_1 = [\min_{i,1\le i\le n_g}\{size(G_i)\}, ar]$;
/*subinterval for the first shot class*/
for $(i = 2; i < n; i++)$
/*subintervals for intermediate shot classes*/
$interval_i = [ar^{i-1}, ar^i]$;
$interval_n = [ar^{n-1}, \max_{i,1\le i\le n_g}\{size(G_i)\}]$;
/*subinterval for the last shot class*/
for $(j = 1; j \le |C|; j++)$
/*for each clip compute its average GOP size*/
$clip\_avg[j] = (\sum_{G_k \in C_j} size(G_k))/length(C_j)$;
for $(j = 1; j \le |C|; j++)\{$
/*for each clip*/
find k, $1 \le k \le n$, *such that* $clip\_avg[j] \in interval_k$;
/*shot class for clip C_j*/
insert C_j *into* S_k;
/*all GOPs of C_j would belong to S_k*/
$\}$
end;

The I, B, and P frames in the GOPs of each shot class S_i, $1 \le i \le n$, are now separated to obtain three sets of frames S_{iI}, S_{iB}, and S_{iP} denoting the I, B, and P frames in S_i, respectively. Thus, the shot-class-finding algorithm partitions the clips into n shot classes, eventually separating the frames into $3n$ data sets. As discussed next, the strength of this partitioning stems from the fact that these data sets can be modeled quite accurately.

C. Statistical Characterization of I, P, and B Frames

In this section, we present the observations and analysis of data sets obtained from our shot classification algorithm. Table I shows that geometrical separation of class boundaries kept enough data points in each class (the smallest has 1579) for doing meaningful statistical analysis and modeling. Before we discuss the statistical models, we present distributions of I, P, and B frames of all seven classes.

Seven plots in Fig. 1 are for the I frames of the movie *Crocodile Dundee* (CD). These plots show three facts.

1) The ordering created by our size-based classification of GOPs is preserved in the distribution of sizes of I frames. In other words, cumulative distribution of S_{kI} is quite regular and is to the left of that of $S_{(k+1)I}$, for all k, $1 \le k \le 6$.
2) Frequency distributions of two sets of I frames coming from two adjacent classes of GOPs have some overlap.
3) Frequency distributions of two sets of I frames coming from two nonadjacent classes of GOPs have very little (less than 10%) or no overlap.

These three characteristics of I-frame data sets are also present in P- and B-frame data sets (see Figs. 2 and 3). Plots in Fig. 4 show mean sizes of I, P, and B frames of all seven classes of CD. We can see that for all frame types, the mean sizes increase

TABLE I
I-, B-, AND P-FRAME CHARACTERISTICS IN ALL SHOT CLASSES (CD)

shot class	frame type	Parameters			
		offset(u)	μ	σ	#frame
1	I	10824	19794.18	2111.30	2014
	B	38	555.11	470.91	8056
	P	103	2989.74	988.49	2014
2	I	18157	23583.11	1933.77	2586
	B	72	1107.36	795.84	10344
	P	2813	4969.10	1059.29	2586
3	I	19717	26562.41	2205.68	5026
	B	203	2157.11	1151.20	20104
	P	4704	7211.88	1116.18	5026
4	I	22355	28954.57	2116.28	9045
	B	500	3753.84	1566.81	36180
	P	6662	9399.43	1329.58	9045
5	I	25005	31574.16	2765.58	5940
	B	868	5905.07	2050.99	23760
	P	8769	12292.38	1791.43	5940
6	I	26222	35222.84	3271.03	2899
	B	1563	8720.97	2610.28	11596
	P	11675	15921.50	2166.74	2899
7	I	29891	40294.68	4353.22	1579
	B	4128	13427.61	4148.17	6316
	P	15274	21236.11	3958.19	1579

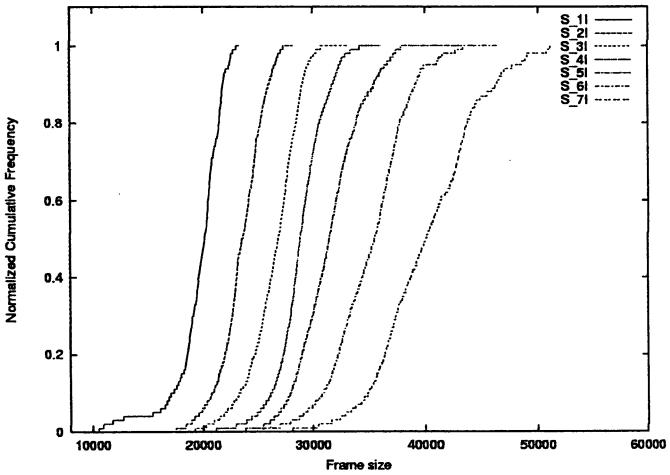


Fig. 1. Separation of I frames (CD).

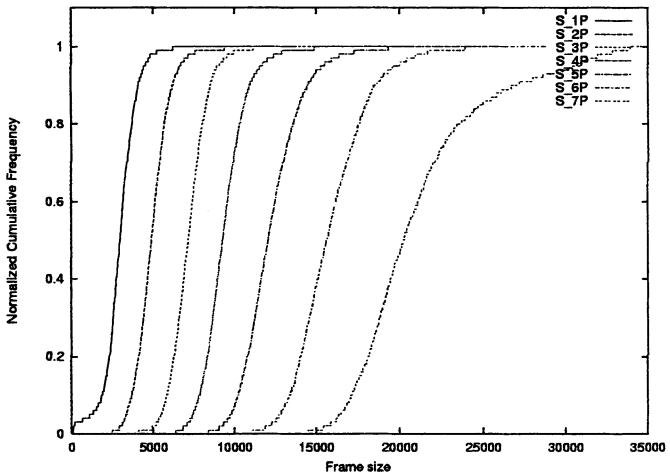


Fig. 2. Separation of P frames (CD).

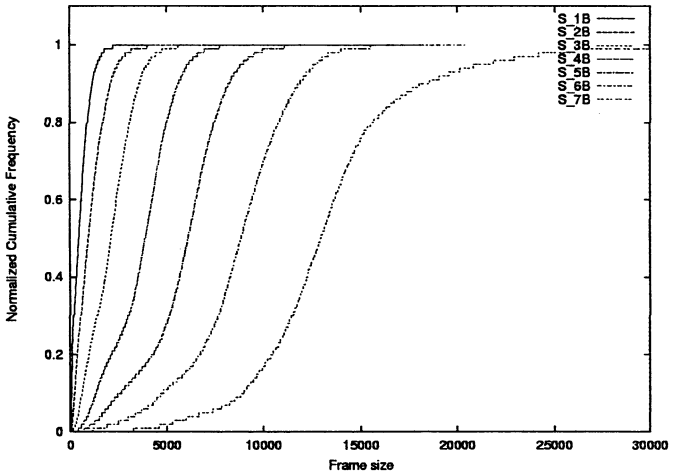


Fig. 3. Separation of B frames (CD).

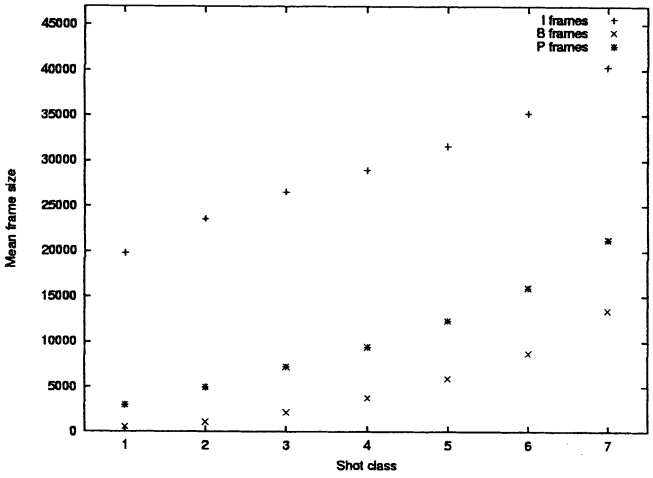


Fig. 4. Mean size of I, B, and P frames in shot classes (CD).

monotonically with shot-class index. Thus, geometrically separated GOP-size boundaries classified I, P, and B frames well. A very useful observation is that in a class of small-size GOPs, I, P, and B frames are also small. It must be noted that such a regular statistical behavior may not hold for a GOP-by-GOP comparison. Figs. 5–7 show similar separation of I, P, and B frames of the movie ET. The monotonical growth of the mean sizes with shot-class index are depicted in Fig. 8.

The next task is to model distributions of I, P, and B frames in each class. In other words, we need to find some standard statistical distributions to fit $(3 \times 7 =) 21$ empirical distributions. The observed patterns and quite successful frame-size models in [3] motivated us to model each frame-size distribution with a Gamma density function

$$\text{Gamma}(x;\, \alpha,\, \beta) = \frac{1}{\beta^{\alpha}\Gamma(\alpha)}\, x^{\alpha-1} e^{-x/\beta}. \qquad (1)$$

Let $\overline{x}$ be the estimated mean and s^2 be the estimated variance of a data set. If the data set has been drawn from a Gamma-distributed population, its parameters α and β can be estimated as $\hat{\beta} = s^2/\overline{x}$ and $\hat{\alpha} = \overline{x}/\hat{\beta}$.

Fig. 9 shows that this conventional estimation of parameters shifts the distribution function to the left and height of the peak

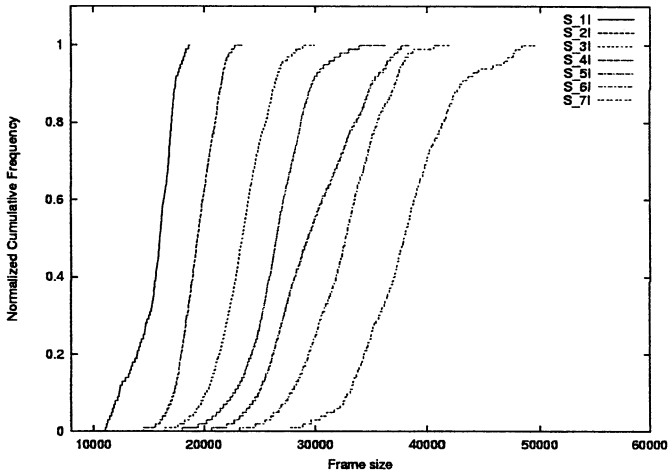


Fig. 5. Separation of I frames (ET).

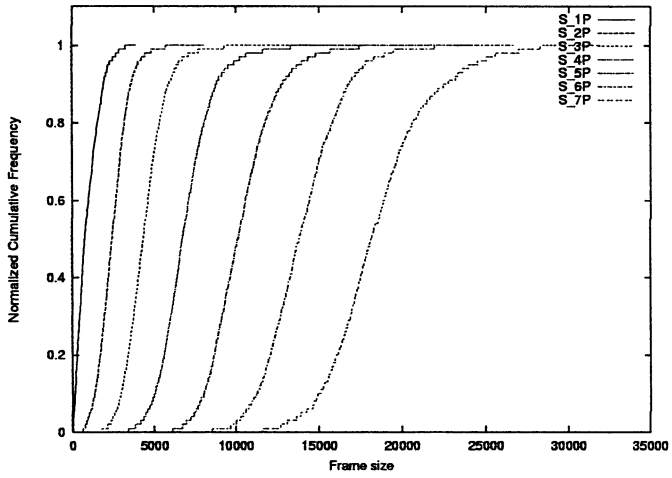


Fig. 6. Separation of P frames (ET).

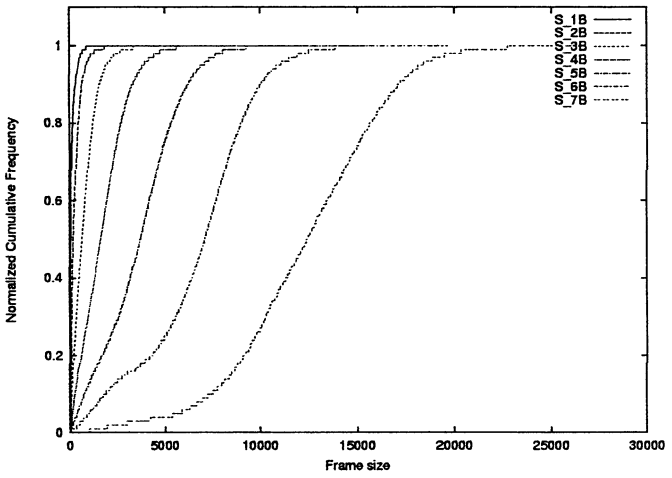


Fig. 7. Separation of B frames (ET).

is smaller than that of the data set. A close look and some educated guessing led us to shift each point of the data set by a constant number of bits u. (We discuss later how we chose the value of shift u.) Estimates for two Gamma parameters, after shifting each point by u units, are given by $\hat{\beta} = s^2/(\overline{x} - u)$ and $\hat{\alpha} = (\overline{x} - u)/\hat{\beta}$. The selection of u is complicated by the fact

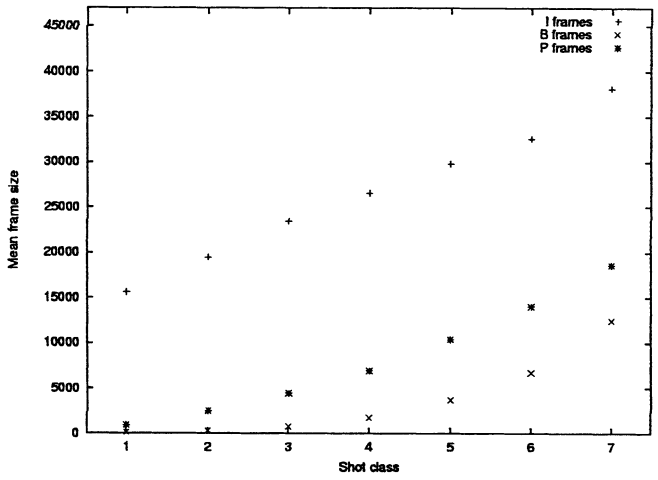


Fig. 8. Mean size of I, B, and P frames in shot classes (ET).

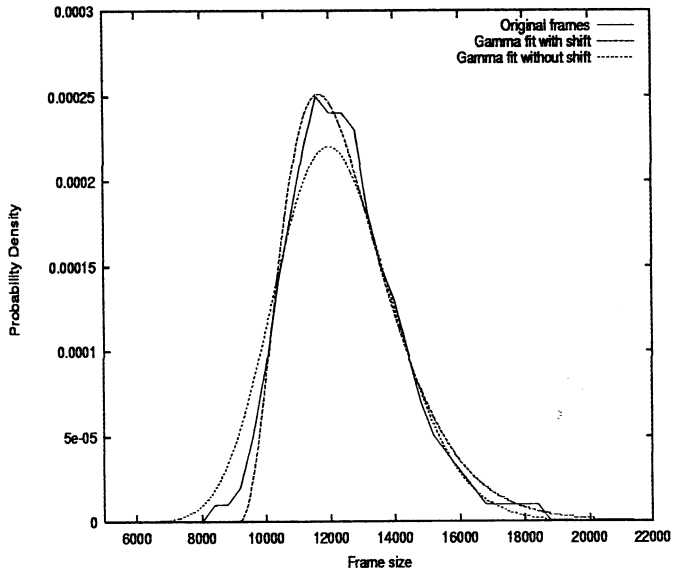


Fig. 9. Gamma fit for P frames in shot class 5 (CD).

that in most data sets there are a few points that are too far to the left or right. These data points should be excluded during analysis and modeling. There are many sophisticated techniques to identify these points. However, we resort to a simple heuristic: Ignore 1% of the data points and set the value of the shift u at one percentile value. Table I shows shift values, mean, and standard deviation for all data sets of CD. These distributions are used for generation of synthetic VBR videos. Results reported in Section IV demonstrate that they nicely capture statistical properties of the data that they model.

The analysis technique used for modeling the duration of video segments is described next.

D. Formation of Video Segments

A video *segment* in a shot class is a *maximal consecutive* sequence of clips belonging to that shot class. It is maximal in the sense that no proper subset of a segment qualifies as a segment, and hence, it is also a maximal consecutive sequence of GOPs in the shot class. The *length* of a segment is the number of GOPs it contains. We examined the clips in each shot class to form

TABLE II
PROBABILITY TRANSITION MATRIX P_A FOR MODEL A (CD)

state	state						
	1	2	3	4	5	6	7
1	0.95	0.03	0.01	0.01	0	0	0
2	0.02	0.89	0.08	0.01	0	0	0
3	0.01	0.03	0.79	0.15	0.02	0	0
4	0	0.01	0.08	0.77	0.13	0.01	0
5	0	0	0.01	0.21	0.65	0.12	0.01
6	0	0	0	0.02	0.24	0.61	0.13
7	0	0	0	0	0.02	0.25	0.73

TABLE III
PROBABILITY TRANSITION MATRIX P_B FOR MODEL B (CD)

state	state						
	1	2	3	4	5	6	7
1	0	0.65	0.20	0.10	0.02	0.01	0.02
2	0.23	0	0.60	0.13	0.03	0.01	0
3	0.03	0.15	0	0.71	0.09	0.02	0
4	0	0.02	0.37	0	0.59	0.02	0
5	0	0	0.03	0.61	0	0.34	0.02
6	0	0	0.01	0.06	0.60	0	0.33
7	0	0	0	0.02	0.05	0.93	0

the video segments and recorded their lengths. The segments were constructed because the length of a segment captures the burst length of similar-size GOPs in a video more accurately than does the length of a clip. Following [2], a Gamma distribution with parameters $\hat{\alpha_{L_i}}$ and $\hat{\beta_{L_i}}$ estimated from observed lengths of segments in class S_i, is used later in this paper for estimating the lengths of video segments in one of our models. In the next section, we present two methods for estimation of inter-class transitions of video shots.

E. Inter-Shot Class Transitions

Let P be a $|S| \times |S|$ transition probability matrix where p_{ij} gives the probability of transition from S_i to S_j. The matrix P has the stochastic property that $\sum_{j=1}^{|S|} p_{ij} = 1$ for $i = 1, 2, \ldots, |S|$. We compute the transition matrix P by two different methods and call the resulting matrices P_A and P_B, respectively. The matrix P_A supports self-transitions but P_B excludes self-transitions. That is, the principal diagonal elements in P_B are zeros. The implications of these two different kinds of transition probability matrices are discussed in Section III.

The algorithms for computing P_A and P_B are quite similar. They both compute the transition probabilities from normalized relative frequency of transitions among shot classes as one sequentially traverses all GOPs $G_1, G_2, \ldots, G_{n_g}$ in the original video. For P_A, we set $p_{ij} = (f_{ij}/f_i)$, where f_{ij} is the total number of transitions from S_i to S_j, and f_i is the total number of transitions out of S_i. The transition matrix P_B is computed in a similar manner except that all self-transitions are ignored. The matrices P_A and P_B for *Crocodile Dundee* are shown in Tables II and III; those for *ET* are quite similar and have not been shown.

III. MODELING OF FULL-LENGTH VIDEO

Two models, called models A and B, for generation of video frame-size sequences are described next. Both the models use

Markov renewal processes. Model A uses only the matrix P_A for inter- and intra-state transitions. Model B uses Gamma-distributed random variables for lengths of video segments and P_B for inter-state transitions. In both models, we used a function $nextstate(s, P)$ to find the next state for a transition from the current state s using the probability transition matrix P. The function $nextstate(s, P)$ is implemented by generating a random number r in the real interval [0, 1] and then finding the smallest integer j as the next state such that $r \leq \sum_{i=1}^{j} p_{si}$. In both models, each shot class corresponds to a state of the underlying Markov chain.

A. Generation of Synthetic Video: Model A

In this model, the next state is determined by using the state transition matrix P_A after generating all frames of a GOP in the current state s and the process is repeated until the desired number of frames is generated. Since the diagonal elements of P_A are nonzero (see Table II) $nextstate(s, P_A)$ might find any state including s as the next state. It may be noted this state transition scheme generates video segments whose lengths would be geometrically distributed. The size of an I-, B-, or P-type of frame in a state is estimated in two stages: first, the parameters of the shifted Gamma distribution for that frame type and state (as obtained in Section II-C) are used to draw a value from this distribution, and then this value is added to the offset u to incorporate the axis translation (see Section II-C).

The MPEG traces used in this paper have (6,3) cyclic MPEG format, so N is 6 in the algorithm that follows.

Algorithm Generate_Full_Video_Using_P_A
Input
  n_f: *number of frames to be generated*;
  $\hat{\alpha_{sR}}$, $\hat{\beta_{sR}}$, u_{sR}; $1 \leq s \leq |S|$, $R \in \{I, B, P\}$: *Gamma parameters and offset for I, B, and P frame types in each class*;
  P_A: *probability transition matrix*;
Output
  F': *a sequence of* n_f *frame sizes*;
begin
  $s =$ *initial random state*, $1 \leq s \leq |S|$;
  $count = 0$;/*number of frames generated*/
  $F' = \phi$;
  while ($count < n_f$) { /*generate **one** GOP in state s, the current state*/
    $for(k = 1; k \leq N; k++)$ {
      /* generate N frames in each GOP*/
      case (type of kth frame in GOP):
        I:*draw* $f \sim Gamma(\hat{\alpha_{sI}}, \hat{\beta_{sI}})$;
          /*draw f from Gamma distribution*/
          $f = f + u_{sI}$;
          /*add offset to estimate frame size*/
        B:*draw* $f \sim Gamma(\hat{\alpha_{sB}}, \hat{\beta_{sB}})$;
          $f = f + u_{sB}$;
        P:*draw* $f \sim Gamma(\hat{\alpha_{sP}}, \hat{\beta_{sP}})$;
          $f = f + u_{sP}$;
      insert $< count, frame - type, f >$ *into* F';
      count = *count* $+ 1$;
    }
  $s = nextstate(s, P_A)$;
    /*change current state using P_A*/
  }
end;

B. Generation of Synthetic Video: Model B

In this model, inter-state transitions are controlled by the state transition matrix P_B. The number of GOPs generated while in a state is modeled by a Gamma distribution of segment length for this state. The parameters for these distributions for all states were estimated from the segment lengths of the original video (see Section II-C). After the frames corresponding to a segment in a shot class are generated, the transition matrix P_B is used to determine the next shot class or state. Since the diagonal elements of P_B are zeros (see Table III), unlike Model A, the next state would always be different from the current state in this model. The process is repeated until the desired number of video frames is generated. Procedure Generate_Video_Segment, detailed below, generates the frames corresponding to a segment of a given state. This is followed by algorithm Generate_Full_Video_Using_ P_B which uses this procedure in any state and then makes a transition to the next state in order to generate frame sizes for the whole video.

1) Generation of Individual Segments: Procedure Generate_Video_Segment determines the number of GOPs to be generated at the current state by using the Gamma parameters of segment length distribution for this state. As in model A, the individual I, B, and P frames are generated by their respective Gamma models.

Procedure Generate_Video_Segment(s,Q)
Input:
  s: *the current state*;
  $\hat{\alpha_{sR}}$, $\hat{\beta_{sR}}$, u_{sR}; $R \in \{I, B, P\}$: *Gamma parameters and offset for I, B, and P frame types in state s;*
  $\hat{\alpha_{L_s}}$, $\hat{\beta_{L_s}}$: *Gamma parameters of segment length for state s;*
Output:
  Q: *sequence of frame sizes for a segment generated in state s*;
begin
    $Q = \phi$;
    draw $n_L \sim Gamma(\hat{\alpha_{L_s}}, \hat{\beta_{L_s}})$;
     /*estimated segment length for current state*/
    for $(j = 1; j \leq n_L; j++)$ {
     /*generate n_L GOPs in state s*/
     for $(k = 1; k \leq N; k++)$ {
      /* generate N frames in each GOP*/
      case (type of kth frame in GOP)
        I:*draw* $f \sim Gamma(\hat{\alpha_{sI}}, \hat{\beta_{sI}})$;
          /*draw f from the Gamma distribution*/
          $f = f + u_{sI}$;
          /*add *offset* to estimate frame size*/
        B:*draw* $f \sim Gamma(\hat{\alpha_{sB}}, \hat{\beta_{sB}})$;
          $f = f + u_{sB}$;
        P:*draw* $f \sim Gamma(\hat{\alpha_{sP}}, \hat{\beta_{sP}})$;
          $f = f + u_{sP}$;
      insert $< count, frame - type, f >$

```
            into Q;
            count = count + 1;
              /*global counter*/
          }
        }
end;
```

2) Generation of Full Video: The full video in this model is generated by starting from an initial random state and then making the state transitions using P_B. In any state s, the sequence of frames Q corresponding to a segment in s is generated by calling procedure Generate_Video_Segment (s, Q). The process is repeated until the desired number of frames is generated.

```
Algorithm Generate_Full_Video_Using_P_B
Input:
    n_f:number of frames to be generated;
    P_B:transition probability matrix;
Output:
  F'': a sequence of n_f frame sizes;
begin
    s = initial random state, 1 ≤ s ≤ |S|;
    count = 0;/*number of frames generated*/
    F'' = φ;
    while (count < n_f) {
      call Generate_Video_Segment(s,Q);
    /*generate Q, a segment of frames for state s*/
    insert all frames of Q into F'';
    count = count + |Q|;
    s = nextstate(s, P_B);
      /*change current state using P_B*/
  }
end;
```

IV. Model Validation

We have presented multilevel characterization techniques for full-length VBR video data sets. Also, two models for generation of synthetic full-length VBR videos have been proposed. To validate these models, model-generated VBR videos have been compared with original VBR videos. Following standard techniques in the literature, we show quartile-quartile (QQ) plots and data loss observed from simulation of leaky bucket.

A. QQ Plots

The QQ plot of two data sets is a visual inspection method for verification of their similarity. In this method, for a given percentile rank (say, 10%), a pair of values of data (say, $\langle 1293,1243 \rangle$) from two data sets are obtained. Usually, several pairs of values are collected for desired range of percentile values and are plotted. If two data sets are identical, a straight line, described by $y = x$, is obtained. Thus, the closer the plot to the line $y = x$, the better the similarity between the data sets.

First, the traces of the movie CD are discussed. The plot in Fig. 10 depicts the similarity of the original VBR video data set with that synthetically generated using our Model A (which assumes geometrically distributed video shot lengths). As can be seen, the frame sizes of the two videos are almost identical; the only exceptions are a few large-size frames in the synthetic movie. These exceptions may practically be ignored, since the number of such frames is only a small fraction of a percent. The Model-B generated video produced a similar QQ plot (see Fig. 11). Although Model A overestimated the frame sizes and Model B underestimated them, these deviations were very small and, practically, the frame-length distributions of synthetic and original videos are indistinguishable for both models. The similarity of the synthetic and original data sets of the movie ET are illustrated in Figs. 12 and 13 for Models A and B, respectively.

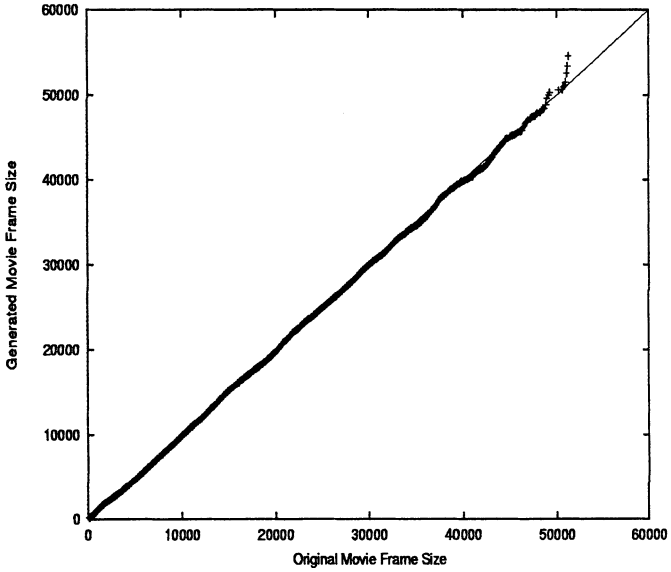


Fig. 10. QQ plot for the whole movie generated using Model A (CD).

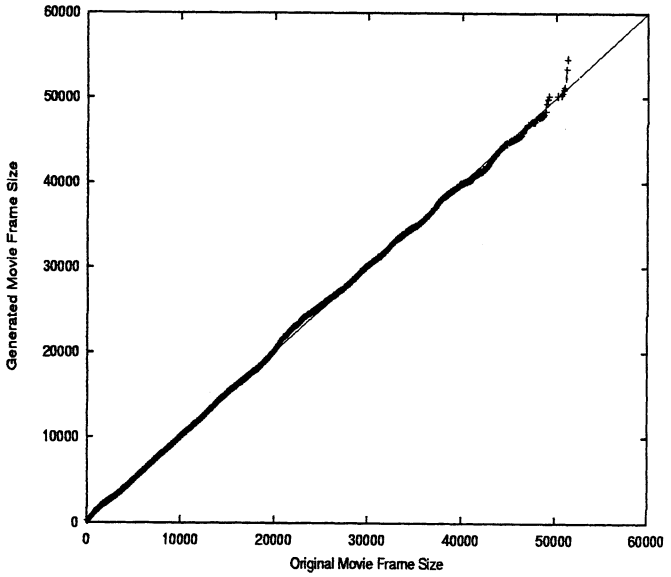


Fig. 11. QQ plot for the whole movie generated using Model B (CD).

B. Buffer Overflow Loss

A QQ plot depicts *global* similarity of two data sets. However, if the elements of these two data sets are ordered by frame index, as in the case of actual video frames, a QQ plot does not reveal any information about local distributions of the frames.

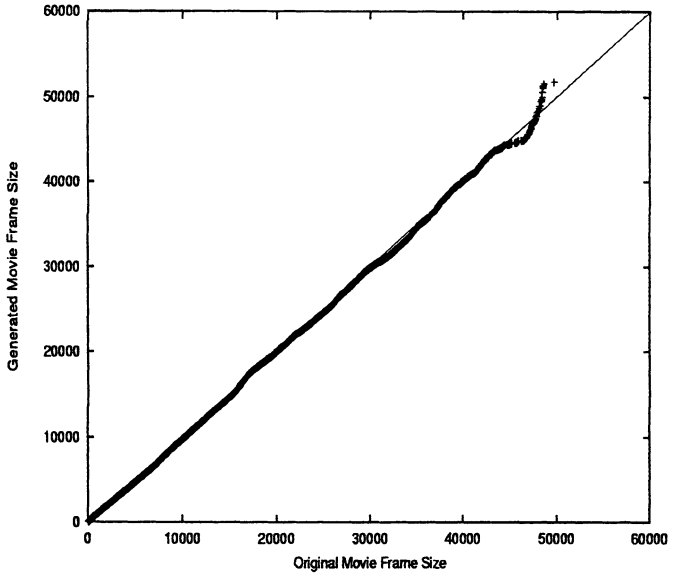


Fig. 12. QQ plot for the whole movie generated using Model A (ET).

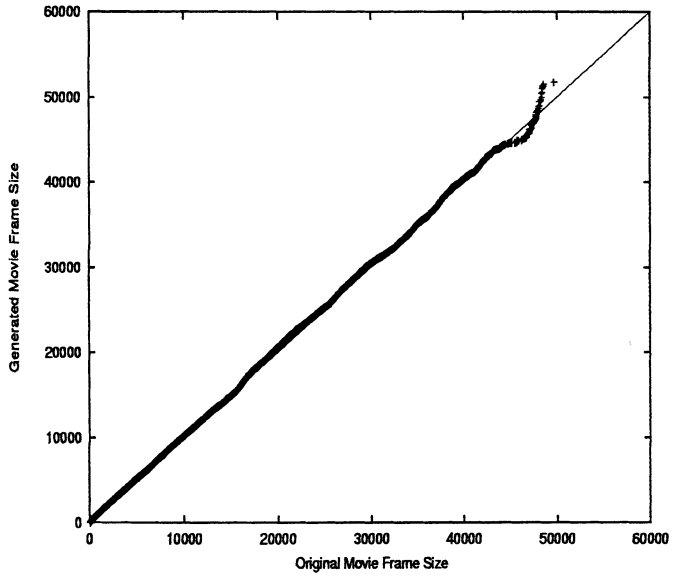


Fig. 13. QQ plot for the whole movie generated using Model B (ET).

For instance, one dataset may have all the large data values together and another dataset may have these large and small data values nicely interleaved, and yet both may show identical QQ plots. For communication of VBR videos over B-ISDN, temporal ordering of the frames plays a critical role in DLR; for a given data transmission rate, the occurrence of long runs of large-size frames (known as burstiness) has higher DLR than the absence of them. Hence, temporal burstiness of original VBR video must be preserved in the data generated by a good model. The most commonly used test for measuring this behavior is passing the data through a generic buffer with capacity c and drain rate factor d. For our study, buffer capacity is expressed in terms of mean frame size of the VBR source and is independent of d. For a 25-frames/s source, $c = 20$ ms corresponds to one half of a mean frame size of the VBR video. The drain rate factor d is the ratio of the number of bytes actually drained per second to the average data rate of the VBR video.

TABLE IV
PERCENTAGE OF BIT LOSS IN ORIGINAL AND GENERATED MOVIES (CD)

buffer capacity(c)	movie type	drain rate factor (d) 2.000	3.000	4.000	5.000
0 ms	original	18.303	4.116	0.358	0.011
	Model A	18.144	4.064	0.365	0.014
	Model B	17.059	3.302	0.267	0.007
10 ms	original	14.222	2.360	0.183	0.002
	Model A	14.022	2.325	0.179	0.005
	Model B	12.918	1.861	0.125	0.002
20 ms	original	10.480	1.326	0.079	0.000
	Model A	10.211	1.301	0.082	0.001
	Model B	9.083	1.024	0.054	0.000
30 ms	original	7.126	0.721	0.033	0.000
	Model A	6.836	0.707	0.036	0.000
	Model B	5.769	0.539	0.022	0.000
40 ms	original	4.404	0.375	0.011	0.000
	Model A	4.172	0.365	0.014	0.000
	Model B	3.383	0.267	0.007	0.000

TABLE V
PERCENTAGE OF FRAMES AFFECTED IN ORIGINAL MOVIE (CD)

buffer capacity(c)	frame type	drain rate factor (d) 2.000	3.000	4.000	5.000
0 ms	All	17.383	9.168	0.921	0.054
	I	97.737	54.433	5.528	0.323
	B	0.483	0.037	0.000	0.000
	P	4.627	0.430	0.000	0.000
10 ms	All	16.130	5.407	0.524	0.018
	I	92.770	32.053	3.146	0.107
	B	0.442	0.034	0.000	0.000
	P	2.241	0.251	0.000	0.000
20 ms	All	14.757	3.040	0.310	0.000
	I	85.637	18.031	1.860	0.000
	B	0.425	0.031	0.000	0.000
	P	1.210	0.083	0.000	0.000
30 ms	All	12.888	1.949	0.120	0.000
	I	74.818	11.503	0.722	0.000
	B	0.410	0.030	0.000	0.000
	P	0.870	0.069	0.000	0.000
40 ms	All	9.478	0.952	0.054	0.000
	I	54.433	5.528	0.323	0.000
	B	0.402	0.029	0.000	0.000
	P	0.828	0.065	0.000	0.000

Tables IV–VII show data loss rates for various buffer capacities and drain rates for original and synthetic traces of CD. Each cell of Table IV shows percentage data loss for original and full-length synthetic movies from the two models. For instance, when $c = 20$ ms and $d = 3$, the original movie suffers 1.326% data loss. With an identical buffer and transmission setting, full-length movies generated by Model A and Model B suffer 1.301% and 1.024% data loss, respectively. Although the differences are very small, one can see that 1) Model A shows a higher data loss than the original movie, and 2) Model B shows a lower data loss than the original movie. This observation also holds for other communication settings.

During our simulation study, percentage of total frames lost and percentage of each type of frames lost were recorded. Some of the results are reported in Tables V–VII. The additional insight obtained from the frame-specific data-loss pattern is quite revealing. For the original movie, when $c = 20$ ms and $d = 3$, only 3.04% of all frames are affected; a closer look reveals that most of them are I frames—as many as 18.031% I frames have been affected. The impact of an affected I frame propagates

TABLE VI
PERCENTAGE OF FRAMES AFFECTED IN MOVIE GENERATED WITH MODEL A (CD)

buffer capacity(c)	frame type	drain rate factor (d) 2.000	3.000	4.000	5.000
0 ms	All	17.355	8.758	1.002	0.054
	I	97.156	52.270	6.009	0.323
	B	0.558	0.000	0.000	0.000
	P	4.744	0.275	0.000	0.000
10 ms	All	16.181	5.281	0.518	0.025
	I	93.017	31.558	3.108	0.148
	B	0.382	0.000	0.000	0.000
	P	2.540	0.131	0.000	0.000
20 ms	All	14.690	3.086	0.270	0.009
	I	85.626	18.460	1.619	0.055
	B	0.290	0.000	0.000	0.000
	P	1.351	0.055	0.000	0.000
30 ms	All	12.570	1.778	0.125	0.001
	I	73.698	10.660	0.753	0.003
	B	0.227	0.000	0.000	0.000
	P	0.815	0.007	0.000	0.000
40 ms	All	8.933	1.002	0.054	0.000
	I	52.274	6.009	0.323	0.000
	B	0.199	0.000	0.000	0.000
	P	0.529	0.000	0.000	0.000

TABLE VII
PERCENTAGE OF FRAMES AFFECTED IN MOVIE GENERATED WITH MODEL B (CD)

buffer capacity(c)	frame type	drain rate factor (d) 2.000	3.000	4.000	5.000
0 ms	All	17.326	7.489	0.774	0.037
	I	97.270	44.673	4.644	0.223
	B	0.517	0.000	0.000	0.000
	P	4.617	0.261	0.000	0.000
10 ms	All	16.241	4.322	0.407	0.013
	I	93.705	25.838	2.441	0.079
	B	0.331	0.000	0.000	0.000
	P	2.417	0.096	0.000	0.000
20 ms	All	14.702	2.516	0.187	0.001
	I	86.008	15.071	1.124	0.007
	B	0.241	0.000	0.000	0.000
	P	1.241	0.024	0.000	0.000
30 ms	All	11.879	1.458	0.084	0.000
	I	69.827	8.749	0.502	0.000
	B	0.186	0.000	0.000	0.000
	P	0.705	0.000	0.000	0.000
40 ms	All	7.636	0.774	0.037	0.000
	I	44.673	4.644	0.223	0.000
	B	0.174	0.000	0.000	0.000
	P	0.447	0.000	0.000	0.000

across the whole GOP containing the I frame. Thus, one can conclude that as high as 18.031% of the movie would be affected. To reduce the percentage of affected I frames, it is necessary to increase the buffer size (creating the effect of video smoothing) or the drain rate (increasing bandwidth allocation), or both. The plots in Fig. 14 show that the data loss affects I frames several times more than it affects P or B frames. The least affected frames are B frames.

Simulation studies with traces of the movie ET show similar results. To conserve space, we do not show them here as tables. Instead, we illustrate some of these observations as plots in Figs. 15–20. It may be noted that for a reasonably high drain rate factor hardly any P or B frames are affected but still the loss of I frames could be significant. For example, Figs. 19 and 20 show that for a drain rate factor of 4.5, a large percentage (5%–20%)

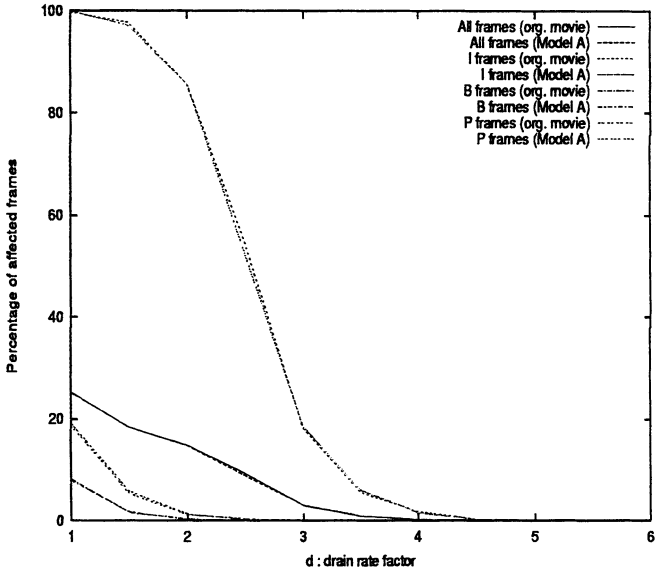


Fig. 14. Affected frames versus drain rate in Model A. Buffer capacity = 20 ms (CD).

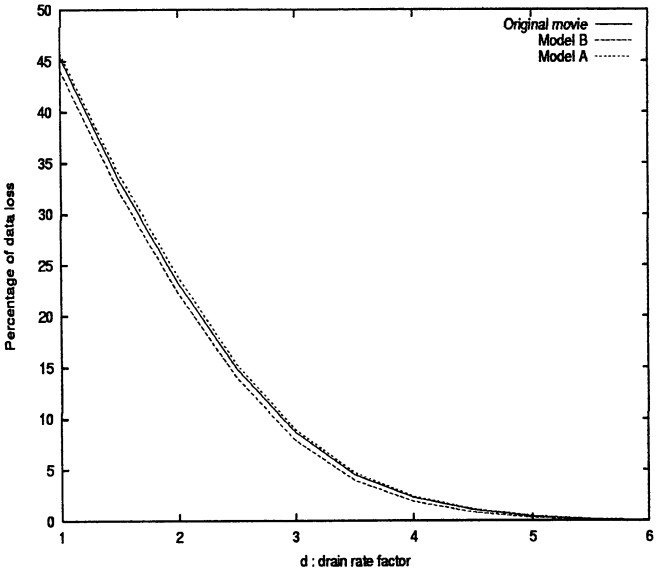


Fig. 15. Data loss versus drain rate. Buffer capacity = 20 ms (ET).

of I frames were affected, but hardly any P or B frames were affected, consequent to which the plots for P and B frames practically coincided with the horizontal axis. From these figures, we can see that model-generated traces very closely follow the data-loss trends of the original movie. Although differences are very insignificant, Model-A generated traces show higher and Model-B generated traces show lower data (and frame) loss rates than that of the original movie.

V. DISCUSSION AND CONCLUSIONS

Frame-size-based models of VBR videos, especially full-length movies, are essential for understanding the effect of data loss during transmission of MPEG-like compressed VBR videos over B-ISDN. However, no satisfactory frame-size-based model has been reported in the past. The limited success of past efforts in frame-by-frame modeling of full-length VBR

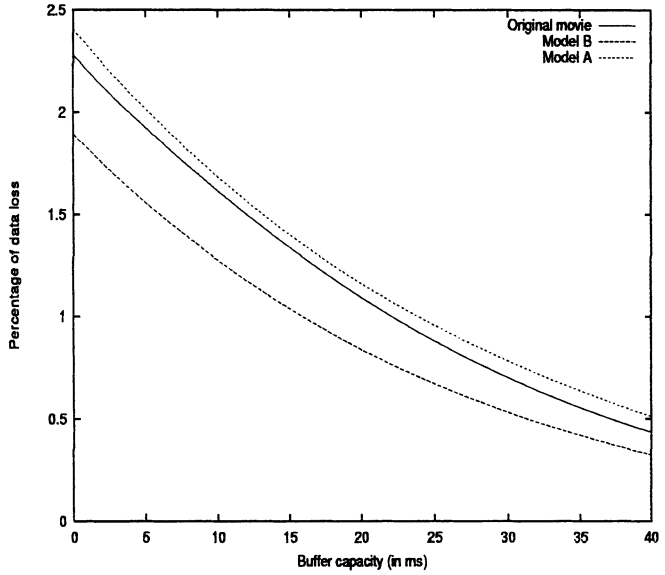


Fig. 16. Data loss versus buffer capacity. Drain rate = 4.5 (ET).

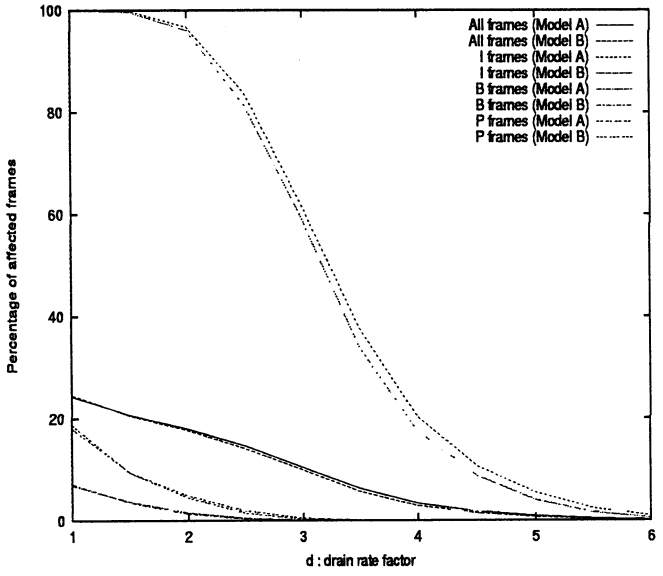


Fig. 18. Affected frames versus drain rate in Models A and B. Buffer capacity = 20 ms (ET).

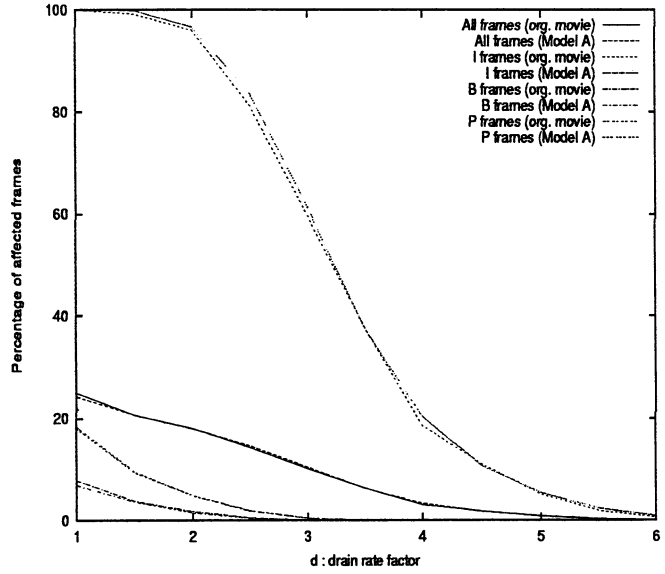


Fig. 17. Affected frames versus drain rate in Model A. Buffer capacity = 20 ms (ET).

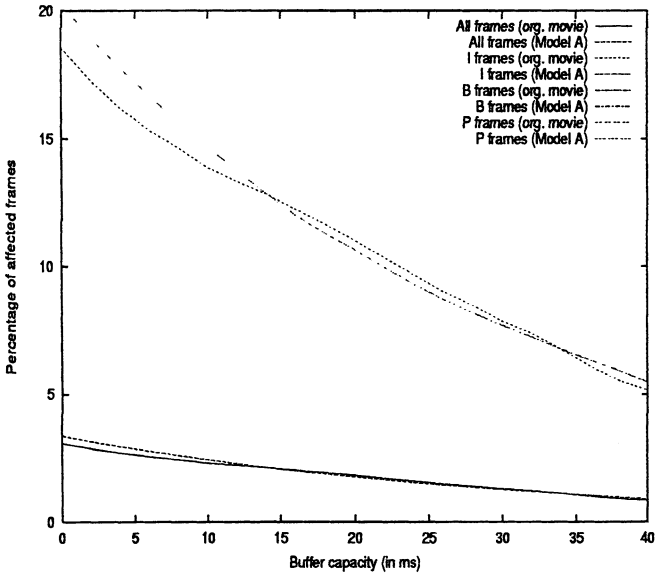


Fig. 19. Affected frames versus buffer capacity in Model A. Drain rate = 4.5 (ET).

video is not due to lack of effort, but because of the complex nature of frame-size data sets. Three different compression techniques applied to I, P, and B frames produce different amounts of compression. Moreover, different segments of a full-length movie produce frames of different sizes, because of composition or content of picture and temporal similarity of adjacent pictures. A universal VBR video model must have enough parameters to capture all classes of video segments, and all three frame types (I, B, and P) in each class of video segments.

In this paper, we have presented algorithms for analysis of full-length VBR videos. After two-pass analysis of a VBR video, our algorithms identify and partition (size-based) classes of video shots. Frames in each class produce three data sets, one each for I-, B-, and P-type frames. Each of these data sets fits an axis-shifted Gamma distribution, whose parameters are estimated form the data set it models.

Using these Gamma distributions and Markov renewal processes, we have proposed two models for generation of synthetic VBR videos. These, being frame-size models, generate sizes of I, B, and P frames. Thus, one can study types of frames being affected during communication.

We have used QQ plots to show visual similarity of model-generated VBR video data sets with original data set. Similarity of local burstiness of model-generated VBR videos and original video have been validated using leaky-bucket simulation technique. Full-length videos generated by both models preserved local burstiness of original video. Our study of frame-based VBR video revealed that even a low data-loss rate could affect

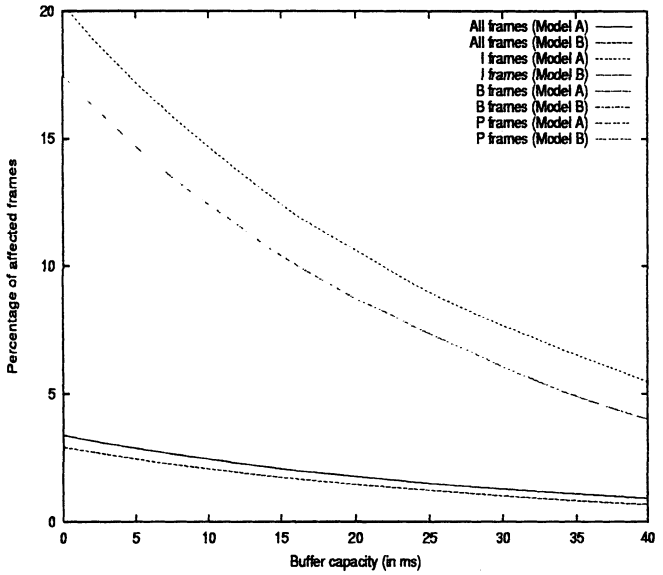


Fig. 20. Affected frames versus buffer capacity in Models A and B. Drain rate = 4.5 (ET).

a large fraction of I frames causing a significant degradation of the quality of transmitted video.

In summary, we provide not only two good models for generation of synthetic VBR video for study of B-ISDN, but also a tool for understanding of the quality of transmitted video when communication is subject to data loss. We are now analyzing more full-length videos for modeling them. Once a good number of videos have been modeled, we plan to study bandwidth gain obtainable from multiplexing them.

ACKNOWLEDGMENT

The authors would like to thank W. Feng for the MPEG traces.

REFERENCES

[1] K. Chandra and A. R. Reibman, "Modeling one- and two-layer variable bit rate video," *IEEE/ACM Trans. Networking*, vol. 7, pp. 398–413, June 1999.

[2] A. M. Dawood and M. Ghanbari, "Content-based MPEG video traffic modeling," *IEEE Trans. Multimedia*, vol. 1, pp. 77–87, Mar. 1999.

[3] M. Frey and S. Nguyen-Quang, "A Gamma-based framework for modeling variable-rate MPEG video sources: The GOP GBAR model," *IEEE/ACM Trans. Networking*, vol. 8, pp. 710–719, Dec. 2000.

[4] D. P. Heyman, "The GBAR source model for VBR videoconferences," *IEEE/ACM Trans. Networking*, vol. 5, pp. 554–560, Aug. 1997.

[5] D. P. Heyman and T. V. Lakshman, "Source models for VBR broadcast-video traffic," *IEEE/ACM Trans. Networking*, vol. 4, pp. 40–48, Feb. 1996.

[6] D. P. Heyman, A. Tabatabai, and T. V. Lakshman, "Statistical analysis and simulation study of video teleconferencing traffic in ATM," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 2, pp. 49–59, Mar. 1992.

[7] M. M. Krunz and A. M. Makowski, "Modeling video traffic using M/G/∞ input processes: A compromise between markovian and LRD models," *IEEE J. Select. Areas Commun.*, vol. 16, pp. 733–748, June 1998.

[8] M. Krunz, R. Sass, and H. Hughes, "Statistical characteristics and multiplexing of MPEG streams," in *Proc. IEEE INFOCOM*, 1995, pp. 455–462.

[9] M. Krunz and S. K. Tripathi, "Scene-based characterization of VBR MPEG-compressed video traffic," in *Proc. ACM SIGMETRICS*, 1997, pp. 192–202.

[10] D. M. Lucantoni, M. F. Neuts, and A. R. Reibman, "Methods for performance evaluation of VBR video traffic models," *IEEE/ACM Trans. Networking*, vol. 2, pp. 176–180, Apr. 1994.

[11] P. Manzoni, P. Cremonesi, and G. Serazzi, "Workload models of VBR video traffic and their use in resource allocation policies," *IEEE/ACM Trans. Networking*, vol. 7, pp. 387–397, June 1999.

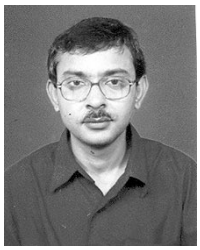
[12] B. Melamed and D. E. Pendarakis, "Modeling full-length VBR video using markov-renewal-modulated TES models," *IEEE J. Select. Areas Commun.*, vol. 16, pp. 600–611, June 1998.

[13] M. Nomura, T. Fuji, and N. Ohta, "Basic characteristics of variable rate video coding in ATM environment," *IEEE J. Select. Areas Commun.*, vol. 7, pp. 752–760, June 1989.

[14] Q. Ren and H. Kobayashi, "Diffusion approximation modeling for Markov modulated bursty traffic and its applications to bandwidth allocation in ATM networks," *IEEE J. Select. Areas Commun.*, vol. 16, pp. 679–691, June 1998.

[15] U. K. Sarkar, S. Ramakrishnan, and D. Sarkar, "Modeling full-length video using Markov-modulated Gamma-based framework," in *Proc. IEEE Globecom*, 2001, pp. 1–5.

[16] J.-L. Wu, Y. -W Chen, and K.-C. Jiang, "Two models for variable bit rate MPEG sources," *IEICE Trans. Commun.*, vol. E78-B, pp. 737–745, 1995.



Uttam K. Sarkar received the B.Tech., M.Tech., and Ph.D. degrees from the Department of Computer Science and Engineering, Indian Institute of Technology, Kharagpur.

He is currently an Associate Professor at the Indian Institute of Management, Calcutta. He was on sabbatical with the Department of Computer Science, University of Miami, Coral Gables, FL, from 2000 to 2002. His research interests include video traffic modeling, multimedia systems, and combinatorial algorithms.

Dr. Sarkar was a recipient of the B. P. Poddar Merit Scholarship from the Indian Institute of Technology, Kharagpur.

Subramanian Ramakrishnan received the B.Stat. (Honors), M.Stat., and Ph.D. degrees from the Indian Statistical Institute, Calcutta, in 1974, 1975, and 1982, respectively.

Since 1982, he has been a faculty member with the University of Miami, Coral Gables, FL, where he is currently an Associate Professor in the Department of Mathematics. His research interests include the foundations of probability theory and stochastic processes, theory of gambling, and recently, discrete-time queueing networks and theoretical computer science.

Dilip Sarkar (SM'96) received the B.Tech degree from the Indian Institute of Technology, Kharagpur, the M.S. degree from the Indian Institute of Science, Bangalore, and the Ph.D. degree from the University of Central Florida, Orlando.

He is currently an Associate Professor in the Department of Computer Science, University of Miami, Coral Gables, FL. His research interests include VBR video traffic modeling, multimedia communication over broadband and wireless networks, middleware and web computing, design and analysis of algorithms, parallel and distributed processing.

Dr. Sarkar is a member of the IEEE Computer Society and the Association for Computing Machinery (ACM).