Hierarchical Modeling of Variable Bit Rate Video Sources

Deepak S. Turaga and Tsuhan Chen
Electrical and Computer Engineering
Carnegie Mellon University, Pittsburgh, PA 15213
{dturaga, tsuhan}@cmu.edu

Abstract

We propose models for variable bit rate (VBR) video traffic that allow for different frame types present in the video, different activity levels of different frames and a variable group of pictures (GOP) structure. We use doubly Markov processes to capture these properties. The performance of these models is evaluated in terms of the stochastic properties of the generated trace as well as using network simulation with five such statistically multiplexed traces. We then illustrate the need to model VBR traces not just at the frame level, but also at lower levels, e.g., at the group of blocks (GOB) level and propose a scheme to partition the frame data generated by our models into these finer hierarchical levels using statistics from training data.

1. Introduction

Variable bit rate (VBR) video coding allows for great flexibility in terms of selection of video coding parameters, efficient compression ratios and can maintain desired video quality. Bitstreams from various VBR video sources can also be efficiently multiplexed over the network using statistical multiplexing techniques. All the above factors have led to VBR video encoders being the preferred mode of coding video streams and the focus of this paper is on modeling such video sources. Modeling video sources is important as it allows for network designers to estimate the parameters of networks like packet loss probabilities and end-to-end delays so that they can guarantee a desired quality of service (QoS).

Modeling VBR video traffic poses difficulties as the bit rate for a given video sequence is determined by a large number of factors. Different compression schemes can lead to different bit rates for the same video sequence. Models for VBR traffic are dependent on the choice of the compression scheme with H.263 [1] and MPEG-4 [2] being some of the latest video coding standards. These standards allow for three different kinds of coding schemes for a video frame in order to improve coding efficiency. A frame may be Intra (I), Predictive (P) or Bidirectionally-predictive (B). An I frame is coded in isolation from other frames using transform coding, quantization and entropy coding. A P frame is predictively coded, which means that a prediction is formed using a previously coded frame and only the difference between the prediction and the actual frame is coded. A B frame is predicted bidirectionally, which means that the prediction is formed using both its previous frame as well as the successive frame. An I frame is often used to efficiently code frames corresponding to scene changes, i.e. frames that are different from preceding frames and therefore cannot be easily predicted. Frames within a scene are similar to preceding frames and hence may be coded predictively as P or B for increased efficiency. Frames between two successive I frames, including the leading I frame, are collectively called a group of pictures (GOP). We illustrate a GOP in Figure 1.

Contact Author: Prof. Tsuhan Chen, Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15213. (email: tsuhan@cmu.edu)

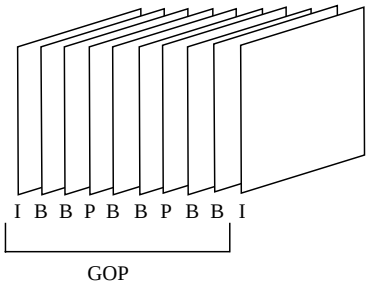

Figure 1. Group of Pictures

In Figure 1 the group of pictures illustrated has one I frame, two P frames and six B frames. Typically, multiple B frames are inserted between two consecutive P or between I and P frames and we show this in the figure. The work in this paper focuses on modeling explicitly video traffic consisting of I and P frames, and can be easily extended to B frames.

Several models for VBR video traffic have been proposed in literature. Maglaris et al [3] have proposed a model for the coding bit rate of a single video source using interframe predictive coding. Sen et al [4] propose models for different activity levels using correlated Markov models and use queuing analysis to estimate the packet loss and delay. Yegenoglu et al [5] propose a model for VBR video using a time dependent autoregressive (AR) model to represent data from different activity levels. Izquierdo and Reeves [6] have performed a survey of different statistical models proposed to model VBR video traffic.

Most of the work done in literature does not explicitly take into account differences between I and P frames. Some work done by Doulamis et al [7] models I, P and B frames explicitly with an additional layer corresponding to the activity level of the video scene. This is a good model for video traffic. However, they impose a constraint of a fixed GOP structure. They assume that every GOP consists of an I frame followed by a fixed number of P and B frames in a fixed pattern and this pattern repeats itself throughout the video sequence and hence we call this model the Fixed GOP Model. Each GOP is viewed as belonging to one of three activity levels, high-activity, medium-activity and low-activity, where activity corresponds to the average bit rate during the GOP. Chandra and Reibman [8] model I and P frames explicitly and allow for a variable GOP structure. However, their model requires a large number of parameters and they do not allow for any temporal correlation or different activity levels for I frames.

In this paper we introduce several models that are flexible enough to allow for the variable GOP structure and also model the characteristics of the video traffic well. We describe models for I, P and B frames that are doubly Markov in nature, to account for trace having different activity levels as well as different types of frames. These models are extensions of models for I and P frames that we introduced in [9] and generate the trace in terms of bits per frame. We examine both the stochastic properties of the trace, e.g., the autocorrelation function, as well as the delay and loss probability encountered by the trace using network simulations. We show that the trace generated by our model can predict the delay and packet loss probability encountered by real data accurately. We generate trace using a frame as a unit, however this may not be the same unit used while packetizing the trace. For instance the bits for one row of blocks in a frame, also called a group of blocks (GOB), may be viewed as one unit. We propose to partition the frame data into GOB data using statistics of GOBs in frames and compare the performance of these traces with actual GOB traces.

This paper is organized as follows. Section 2 describes the models and Section 3 includes a discussion of results in terms of stochastic parameters as well as in terms of network simulations. Section 4 has a brief discussion of the modeling of data at different hierarchical levels. We then conclude with the summary of the models and their performance and identify future research directions.

2. Activity Adaptive Models

Video sequences have large variations in action levels between scenes. This leads to large variations in the bits per frame within I, P or B frames, corresponding to different activity levels. An accurate model needs to capture the effect of having different frame types, as well as this variation in activity level within frames of one type. We thus propose a number of doubly stochastic processes to model both the activity level changes and I, P and B frames corresponding to a certain activity level. As before, the temporal correlation between I, P or B frames corresponding to an activity level is captured using AR(1) processes with Gaussian distributions. Our models are flexible and allow for a variable GOP structure.

2.1. Trace Characteristics

Typical traces with I, P and B frames and variable GOP structure are created as follows. The video encoder first identifies which frames of the sequence need to be coded as I frames. These are frames that lie across scene changes and so cannot be coded efficiently using prediction. This may be determined by creating a prediction for every frame and counting the number of blocks in the frame that need to be intra coded, i.e., cannot be predicted well. Frames that have a large number of intra coded blocks are classified as I frames. After identifying the I frames the sequence is encoded with a repeating the pattern of two B frames followed by a P frame, until the next I frame is reached. Clearly, when the interval between two I frames is not a multiple of three this pattern cannot always be inserted. In that case the last pattern is terminated when the I frame is reached. An example sequence of coded frames is as shown in Figure 2.

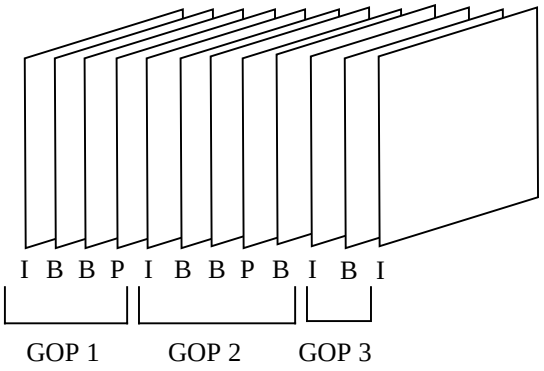

Figure 2. Variable Length GOPs with B frames

From the sequence we see that the interval between the first two I frames is a multiple of three and so the BBP pattern can be repeated once, but after this none of the successive I frames have an interval that is a multiple of three, so in all the other cases the BBP pattern is terminated when the next I frame is reached. Traces generated in such a way have a variable length GOP structure as well as all three kinds of frames. We propose a number of doubly stochastic processes to model both the activity level changes and I, P and B frames corresponding to a certain activity level. The temporal correlation between I, P or B frames corresponding to an activity level is captured using AR(1) processes with Gaussian distributions.

2.2. Type I Models

We divide the video sequence frames into three different activity levels, high-activity, medium-activity and low-activity, based on the number of bits needed to code the frame. The Type I models, as for the I and P models, choose between generating an I frame, a P frame or a B frame before deciding the activity level of the video frame. The traces that we wish to model have some specific characteristics. For instance there are a lot of repeated pairs of B frames, but no

instance of three or more consecutive B frames. Similarly, P frames never occur consecutively due to the way in which we create the traces that we wish to model. So we may modify the structure of the Markov chain corresponding to I, P and B frames in order to account for these specifics in our traces. Hence, instead of having only three states corresponding to I, P and B frames, we introduce an artificial fourth state that we call the BB state. For each transition into this state two B frames are produced. As against this transitions into any of the other three states, I, P or B, result in only one frame being generated. This BB state simulates the repeated B frame structure in our real traces. The other constraints on having no more than two repeated B frames and having no repeated P frames are automatically satisfied when we estimate the transition probabilities of the model from our training data.

2.2.1. Type I Doubly Markov Model

This model has two Markov chains, the outer one having I, P, B and BB states, as described earlier, and the inner one having states corresponding to the activity levels of the type of frame generated. Each of the frames generated, may belong to one of the three activity levels. Our proposed model looks as shown in Figure 3.

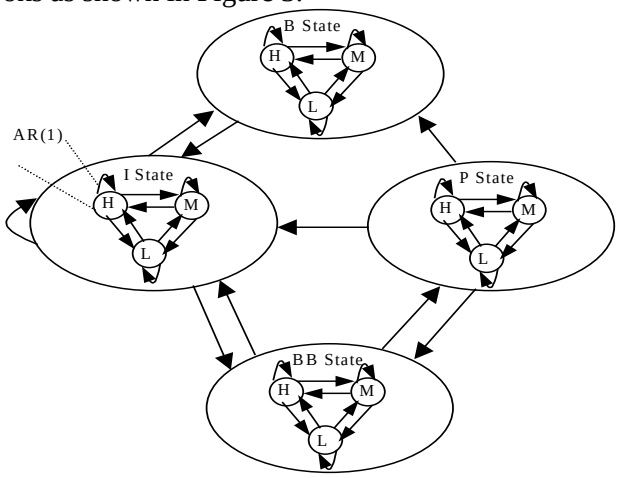

Figure 3. Type I Double Markov Model for trace with B frames

As can be seen from the figure, the outer Markov chain has some constraints on its structure due to the particular characteristics of the data we are trying to model. For instance, there are no transitions in either direction between the BB state and the B state, or self transitions for either of these states, to prevent the occurrence of three or more consecutive B frames. Similarly there are no self-transitions in the P state, as we cannot have consecutive P frames. Also, we realize that we generate a single B frame only when the BBP pattern needs to be prematurely terminated, so the B state may transition only to the I state. All these conditions need not be imposed on the model, training it using the real trace will ensure this transition, structure, however we may use this apriori knowledge to reduce the number of parameters to be estimated during training.

Within each of the four outer states, we have another Markov chain that determines the activity level of the frame to be generated. These activity Markov chains are never restarted. The process of generating data using this model is as follows. We first decide to generate an I frame, a P frame, a B frame or two consecutive B frames. After this, we determine the activity level of the frames using the inner Markov chain. The frames are generated from a Gaussian probability density function (pdf), using an AR(1) processes to capture the temporal correlation between them. The two frames in the BB state are generated from the same activity level.

The training procedure for this model is as follows. Using the sequence of I, P, B and repeated B frames from the real trace we may estimate the initial and transition probabilities of

the outer Markov chain. We then need to estimate the inner Markov chain initial and transition probabilities and the means, variances and AR process parameter ρ for each of the AR(1) processes. We separately collect all I frames, all P frames, all single B frames and all repeated B frames from the real trace. We then use two thresholds for each of these and divide them into high-activity, medium-activity and low-activity. From these sets of data we can estimate the means, variances and AR process parameter ρ of all the AR(1) processes. By looking at the sequence of transitions between these activity levels for each of the four frame types, I, P, B or BB, we can estimate the initial and transition probabilities for the inner Markov chain.

2.2.2. Type I Simplified Model

As before, we try to reduce the parameters for the model by removing Markov chains when they are not necessary. For the I and B states we found experimentally that $P(S(n) = S_i \mid S(n-1) = S_j) \approx P(S(n) = S_i)$ for the inner Markov chain. This means that the transition probabilities between the inner Markov chain states are the same as the unconditional probabilities of being in any of them, so we can replace the Markov chain with a set of unconditional probabilities with which we generate a frame belonging to a certain activity level. This may be explained by the fact that the I and B states occur infrequently and always in different GOPs, typically with a large interval between them. So the dependence of the current activity level on the previous activity level is small. This is however not true for the BB or the P states as they occur frequently and many times in the same GOP. Hence these Markov chains cannot be replaced. The simplified model is as shown in Figure 4.

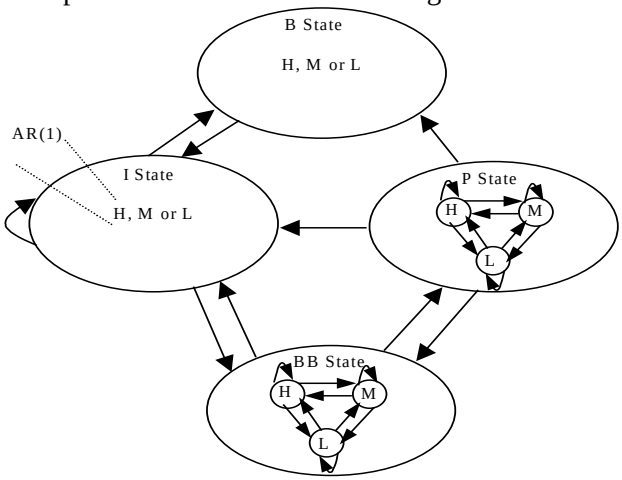

Figure 4. Type I Simplified Model

As for the Doubly Markov model, the structure of the model is chosen keeping in mind the particular characteristics of the data we are trying to model. For this model, we first decide whether we want to generate an I frame, a P frame, an individual B frame or a pair of B frames and following this we decide which activity level this frame/frames should belong to. For the I and B states we decide with a fixed probability the activity level of the generated frame, while for the BB and P states we use the Markov chain to determine the activity level of the frames. As before, none of the inner activity Markov chains are restarted.

The training procedure for this model is very similar to that for the Doubly Markov model. The only difference is that for the I and the B states the unconditional probability of generating a frame belonging to a certain activity level is just the number of frames at that activity level divided by the total number of frames of that type.

2.3. Type II Models

We also implement the Type II models where we first choose which activity state the current frame belongs to before deciding to generate an I, P or a B frame. Here again, using knowledge of the training data characteristics, we choose to use the four state I, P, B and BB model to generate frames. This Markov chain, is now, however the inner Markov chain for this model.

2.3.1. Type II Doubly Markov Model

We also implement the Type II model, where the outer Markov chain corresponds to the activity level and within each activity state there is another Markov chain corresponding to the I, P, B and BB states. This model may be shown as in the following figure.

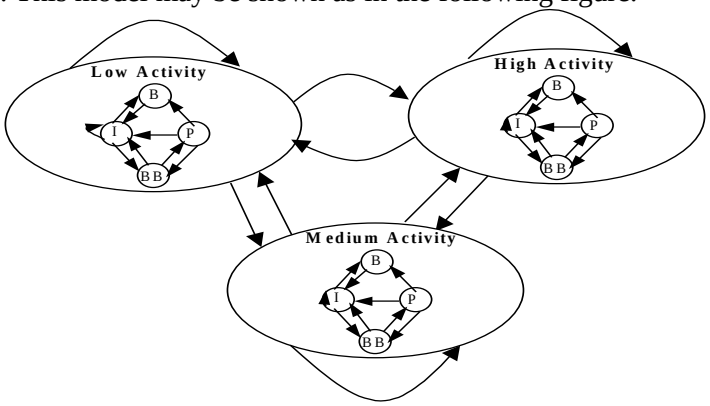

Figure 5. Type II Doubly Markov Model

This model makes the assumption that the entire GOP belongs to one activity level, however, this assumption is not very valid and the performance of the model suffers due to this. In order to generate data using this model, we first use the outer Markov chain to decide the activity level of the current GOP, following which we generate the GOP using the inner Markov chain. When the inner Markov chain transitions back to the I state, we have completed generating one GOP, and so we use the outer Markov chain to determine the activity level of the next GOP. We reinitialize all inner Markov chains after generating one GOP.

In order to train this model, we first obtain the mean bit rate for each GOP and then use two empirical thresholds to partition this sequence of GOPs into activity levels. This sequence of activity levels may be used to train the outer Markov chain. The means, variances and AR process parameter ρ for each of the AR(1) processes may be estimated using the knowledge of the activity level of each frame. We then collect sets of sequences of I, P, B and repeated B frames for each activity level and each of these sets is used to train the initial and transition probabilities for the inner Markov chains.

Our training data shows a great dependency in terms of deciding the current frame based on the previous frame and so we cannot replace any of the inner Markov chains with unconditional probabilities. So, this model cannot be simplified further.

3. Results and Discussion

In order to evaluate the performance of our models we look at both the stochastic properties of the generated data as well as use network simulations to look at the loss probabilities and delays encountered by the traces. All the models were trained on the same data and characteristics of the generated bit rate were compared with those of the real data. The training data was from two different sequences. The first was a high motion video sequence made up of

advertisements. We call this sequence Ads. This sequence had frequent scene changes, camera zooms and pans and a lot of motion. The second sequence was a news clip and we call it News. This sequence contained news reports from different locations and hence it contained a moderate amount of motion and some scene changes. Sample frames from both the sequences are shown in Figure 6.

Figure 6. Sample frames from Ads (left) and News

Both sequences consisted of five minutes of data sampled at 15 Hz, making a total of 4500 frames. Each sequence was converted to bits using a H.263 standard compliant video codec. A random GOP was achieved as described in Section 2.1, and a frame is coded as an I frame if more than 70% of its blocks need to be intra coded. In order to illustrate the need for the flexible GOP structure, we compare our results with the Fixed GOP model.

3.1. Stochastic Properties of Modeled Traces

We compare the mean squared error in modeling the real autocorrelation function by our models with the error using the trace generated by the Fixed GOP model. The mean squared error in autocorrelation function using our proposed models, is smaller by an order of magnitude for both the Ads as well as the News sequence. These results are included in the following table with the entry in each column corresponding to the mean squared error in modeling real autocorrelation function normalized by the error in modeling the real autocorrelation function using the Fixed GOP model.

Table 1. Error in modeling real autocorrelation function normalised by Fixed GOP error

Sequence	Fixed GOP Model Error	Type I Doubly Markov Error	Type I Simplified Error	Type II Doubly Markov Error
Ads	1	0.083	0.081	0.101
News	1	0.074	0.077	0.092

From the table we can see that our models produce traces that are statistically similar to the real data as the error is around 10~13 times smaller than that for a fixed GOP model.

3.2. Network Simulations

In order to actually translate this statistical similarity into the ability to actually predict the packet loss probabilities and delay, we perform the following simulation. We first look at a set of real traces and packetize them with one frame being viewed as one packet. We view each trace as being generated by a different video source and packets from these different sources are statistically multiplexed into a common buffer. Since each packet corresponds to a frame, our packets arrive at regular intervals, thereby leading to a certain periodicity in the system. In order to reduce this periodicity we uniformly distribute the starting times of the different sources within one frame interval of each other. So packets from the same source arrive at regular intervals of one another, but the packets from different sources start arriving at different times, which are uniformly distributed within one frame interval. We then drain the fixed size buffer at a fixed drain rate and evaluate the loss probability and delay encountered by the packets in this set up and

repeat this experiment for different buffer sizes and drain rates. Using the same setup we then replace each video source by our model and evaluate the loss probabilities and delay for our models. For comparison, we also replace each source with the fixed GOP model and evaluate the loss probability and delay for the packets generated by this model. The setup for this simulation is as shown in Figure 7.

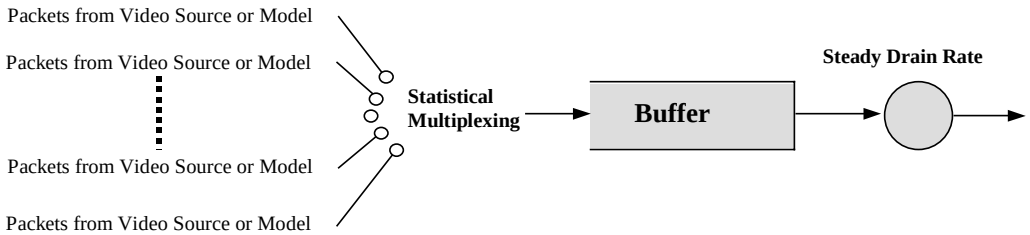

Figure 7. Simulation setup to evaluate delay and loss probability for statistically multiplexed trace

In our simulation we use five source or model traces. The results of these simulations are shown in the following figures. We present these simulation results for the trace generated by one of our models, the Type I Simplified Model and the trace generated using the Fixed GOP model.

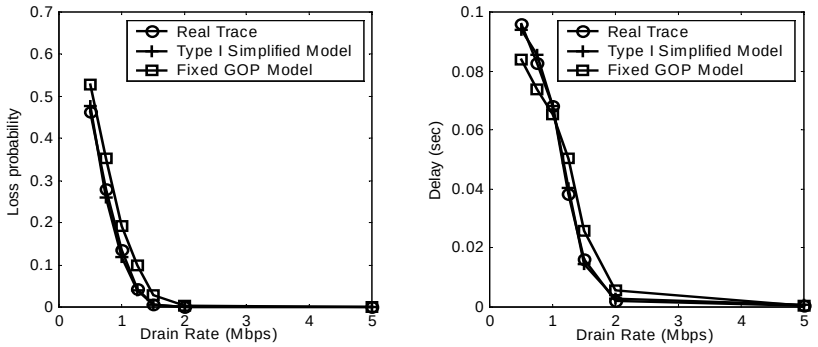

Figure 8. Loss probability and delay for real and modeled traces (Ads) using multiplexed streams

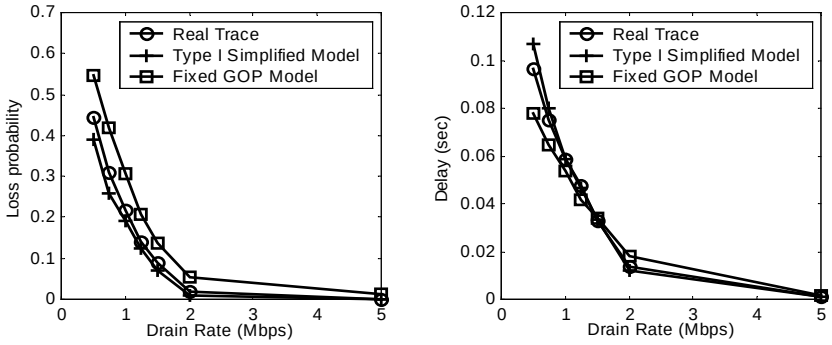

Figure 9. Loss probability and delay for real and modeled traces (News) using multiplexed streams

Figure 8 shows loss probability and delay results for statistically multiplexed streams for the Ads sequence, while Figure 9 shows results for the News sequence. We can see from the figures that the performance of our model is better than the performance of the fixed GOP model. This is indicated by the fact that our predictions for the loss probability and delay are close to the real data. In terms of squared error, our prediction of the loss probability for the real data is 18 times smaller for Ads and 6 times smaller for News than the error for the fixed GOP model. Similarly the squared error in our prediction of the delay is around 20 times smaller for Ads and

3.5 times smaller for News than the error for the fixed GOP model. The gains for the Ads sequence are larger as it is a high motion sequence with frequent scene changes and changes in activity levels, so a fixed GOP model cannot accurately capture all these variations.

4. Modeling at different hierarchical levels

The models we have proposed create data in terms of bits per frame that is accurately representative of real video trace data. When video is transmitted over a network the data is packetized and the packetization is not necessarily one packet corresponding to one frame. Instead smaller units like bits for a group of blocks (GOB) are grouped together in one packet and these packets are transmitted. In order to capture all the properties of a trace composed of GOB packets we need to retrain models using these traces. This leads to a large number of models for many different kinds of packetization schemes. Instead, we propose to capture the properties of the frame trace using our models and then partition this data into a GOB trace or a trace with any other unit, given the statistical properties of the desired unit in a frame. We illustrate this idea by generating GOB traces from the frame traces and compare the performance of these traces in predicting the delay and loss probability encountered by a real trace composed of GOB sized packets. In order to partition our frame trace into a GOB trace we collect the pdfs of GOBs in different types of frames at different activity levels. Knowing the activity level and the type of the current frame, these pdfs are used to create the data appropriately. One constraint that we have to satisfy is that the sum of bits for all GOBs in a frame should total to the bits needed for that frame. We ensure this in the following way. If there are N GOBs in a frame (for instance there are 9 GOBs in a QCIF frame) we generate N–1 GOB sizes using the appropriate pdf and the last GOB is given the difference between the number of bits for the frame and total of these N–1 GOBs. In order to verify the need for this partitioning using pdfs we also partition the frame data into GOB data by assigning $1/N$ of the frame bits to each GOB. We call this scheme the Mean partition scheme as we assign the mean value to each of the GOBs. We illustrate the performance using the same network simulation as described before, only now we use the GOB traces. The results in terms of loss probability and delay for the Ads sequence are shown in Figure 10.

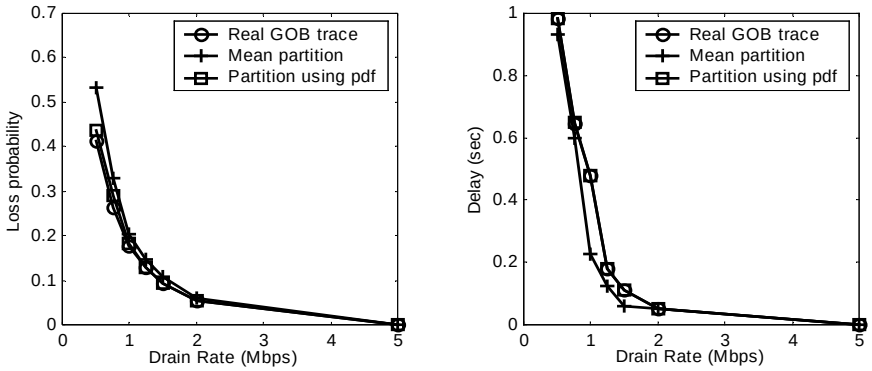

Figure 10. Loss probability and delay for GOB traces for Ads sequence

We can see from the figure that using the pdf of the GOB data provides accurate predictions of the delay and loss probability of real GOB data as opposed to using the mean partition scheme. In terms of squared error, using the pdf to partition data provides nearly 20 times smaller error in predicting loss probability than the mean partition scheme. In fact, the prediction for the delay is identical to the real delay, while the prediction error using the mean partition scheme is large. Hence, using the proposed method we can use our models for the frame trace to create data and then partition it appropriately using statistics from training, so that we can capture the effect of different packetization schemes.

5. Conclusion

We propose several models for VBR video sources that allow for a flexible GOP structure, thereby modeling typical traces better. We propose two different kinds of models for data with I, P and B frames. These are the Type I models that choose the type of the frame first before deciding the activity level the frame belongs to and the Type II models that decide the activity level of the frame before choosing the type of the frame. We show that the generated traces are statistically similar to the real data using the error in the autocorrelation function as a measure, which is smaller by a factor of 10~13 over using a fixed GOP model. We also evaluate the performance of the models in terms of predicting the loss probability and delay when we run these traces through a network simulation and show that the delay and loss probabilities predicted by our models are accurate. Our predictions for loss probability are 6~18 times smaller in squared error than predictions using a fixed GOP model and our predictions for delay are 3~20 times smaller than the fixed GOP model predictions. We also include a discussion for the need to model the data at different hierarchical levels to account for different packetization schemes and show that using the statistical properties of actual traces, we can partition our frame data into GOB data. We show that using the actual pdf to partition frame data can result in around 20 times smaller error in predicting loss probability and very small error in predicting delay than just using the mean partition scheme. Future work involves validating these models across a larger variety of sequences, capturing the properties of rate controlled video traffic and partitioning frame into other lower levels. We are also examining the need for the use of the actual pdf while generating the frame data

References

[1] *Video Coding for Low Bit Rate Communication*, ITU-T Recommendation H.263 Version 2, Jan. 1998.

[2] Motion Pictures Experts Group, "Overview of the MPEG-4 standard", ISO/IEC JTC1/SC29/WG11 N2459, 1998.

[3] B. Maglaris, D. Anastassiou, P. Sen, G. Karlsson and J.D. Robbins, "Performance models of Statistical Multiplexing in Packet Video Communication," *IEEE Trans. Comm.*, vol. 36, pp. 834-43, 1988.

[4] P. Sen, B. Maglaris, N, Rikli and D. Anastassiou, "Models for Packet Switching of Variable-Bit-Rate Video Sources," *IEEE J. on Select. Areas in Comm.*, vol. 7, no. 5, June 1989.

[5] F. Yegenoglu, B. Jabbari and Ya-Qin Zhang, "Motion-classified Autoregressive Modeling of Variable Bit Rate Video," *IEEE Trans. CSVT.* vol. 3, no. 1, Feb 1993.

[6] M. Izquierdo and D. Reeves, "A survey of statistical source models for variable-bit-rate compressed video," Multimedia Systems, vol.7, no.3, pp. 199-213.

[7] N. Doulamis, A. Doulamis and S. Kollias, "Modeling and Adaptive Prediction of VBR MPEG Video Sources," pp. 27-32, 1999 IEEE Third Workshop on Multimedia Signal Processing, September 1999, Copenhagen, Denmark.

[8] K. Chandra and A. Reibman, "Modeling One- And Two-Layer Variable Bit Rate Video," *IEEE/ACM Trans. Networking*, vol. 7, no. 3, pp. 398-413.

[9] D. Turaga and T. Chen, "Activity-Adaptive Modeling of Dynamic Multimedia Traffic," International Conference on Multimedia and Exposition, August 2000.