Statistical characterization of MPEG VBR video at the Slice layer

Michael R. Izquierdo* and Douglas R. Reeves

*IBM Corporation
Research Triangle Park, North Carolina 27709

Department of Electrical and Computer Engineering
North Carolina State University
Raleigh, North Carolina 27695

ABSTRACT

In this paper, we statistically characterized four VBR encoded video sequences, containing I/B/P frames, at the Slice layer with the goal of developing an accurate source model to better understand the bit-rate behavior of these sources. We presented the cells/slice distributions and showed that it is "heavy tailed" and fits the Pareto distribution better than Gamma. We showed that an 8-state Markov Chain fits the cells/slice distribution well, reaching steady state after 37 to 80 transitions (2 to 5 frames). We also showed that the autocorrelation function is quasi-periodic which is mostly due to the frame sequence pattern rather than spatial content. We discussed the impact of I/B/P sequences on multiplexing and dynamic bandwidth allocation and proposed a multiplexing method called Time Shifted Multiplexing (TSM); whereby, the multiplexer attempts to overlap I and P frames of one video stream with B frames of an other. This tends to reduce both Peak-to-Mean-Ratio and Coefficient-of-Variation of the multiplexed output stream. We showed that coefficient-of-variation reduced in half and bandwidth requirements reduced by 41% using TSM.[1]

Keywords: MPEG, VBR, variable bit rate, Slice layer, statistical characterization

1. INTRODUCTION

The Asynchronous Transfer Mode (ATM) Network has gained much attention as an effective means to transfer voice, video and data information over computer networks. The use of a fixed size, fifty-three byte cell to transfer data makes ATM well suited to support isochronous type services like voice and video[1]. The small cell size makes it possible to interleave cells from multiple sources over a common communications link; thereby, providing low end-to-end latency. Much work has been done in the area of transporting compressed video over ATM addressing such issues as bandwidth allocation, source modeling, multiplexing, encoding methods and quality of service (QoS).

Variable bit rate (VBR) encoding has several advantages over constant bit rate (CBR) encoding such as consistent and subjectively better video quality, simpler encoder design and increased multiplexing gain. One paper compared the luminance signal-to-noise ratio (SNR) of CBR and 1-layer VBR and showed significant reductions in SNR of up to 7 dB[2]. Another paper showed that the statistical multiplexing of multiple VBR sources provided a gain of a factor of two over CBR[3]. Recently a gain of slightly higher than four was found to be possible with cell loss probabilities of 10^{-6} [4].

1. This work was supported by the Air Force Office of Scientific Research under Grant F49620-92-J-0441, and by an IBM Corporation Resident Study Fellowship.

One of the main drawbacks of VBR is that its burstiness increases the probability of cell loss by making it difficult to determine bandwidth requirements. Burstiness is caused by the fact that the encoder is not controlling the quantization scale dynamically in order to maintain a constant bit rate. In this sense, VBR is referred to as an *open-loop* encoding method. The user must specify bandwidth requirements when establishing a connection in order for the network to determine if enough resources, such as buffers and communications links, exist. This is a preventative congestion control method in that flow control is done at the source in an attempt to avoid congestion[5]. The user can specify bandwidth at the peak rate, but this would waste a significant amount of bandwidth. If bandwidth is improperly specified, high cell loss could occur.

Dropping cells from the video stream indiscriminately can cause serious degradations in QoS. This can be remedied by an effective transport protocol; however, MPEG-1 was developed to operate over lossless connections and does not define a transport protocol[1] to deliver compressed video data over networks. As a result, several techniques have been developed to either reduce the amount of cell-loss or minimize its effects. One such method splits the video stream into two high and low priority streams with the high priority stream containing video data most affected by cell loss[6 7 8].

We investigated a method of transporting video using AAL-5[9] where Convergence Sublayer (CS) packets are built from MPEG-1 slices. The Slice layer is the lowest independent data unit in MPEG which makes it possible to provide error-detection, correction or concealment on a portion of a frame rather than the whole frame. This should increase the error resiliency of the video stream. We used MPEG-1 Video Packet Elementary Streams (VPES) which did not contain audio. MPEG streams containing multiplexed video and audio were not available at the time of this study. We used a model of a video server, shown in Figure 1, which sends CS packets of the VPES stream over ATM. The video server parses VPES, in accordance to the MPEG-1 syntax[10], and builds CS packets from slices. Non-slice data, such as Sequence layer headers, are placed in separate CS packets. Packets are segmented into 53 byte cells consisting of a 48 byte payload. Cells are loosely packed in that slice boundaries are not crossed. While this increases the overall number of cells, it does prevent multiple slices from being affected by a single cell loss. Cells are transported over the network to a video client which reassembles the packets and checks for any errors. If an error occurs, the video client could request that a slice be retransmitted if there is sufficient time. Another possibility would be to employ a forward error correction scheme using Reed-Solomon codes.

We focused our attention on the video server output cell stream and assumed that the video was located on a local file system. The objective was to characterize the cell generation process at the output of the Segmenter and determine an appropriate model. Once this is done, then the effective bandwidth (C) for a certain buffer size (B) is determined.

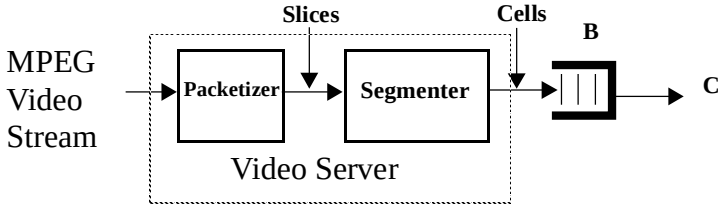


FIGURE 1. VBR video server model to analyze cell generation statistic

1. This has recently changed with the announcement of the MPEG-2 draft standard which defines a Transport Layer for lossy connections.

We studied four video sequences obtained from the Portable Video Research Group called *Bike*, *Flowg*, *Tennis* and *UnderSiege*. Each video sequence was encoded using a quantization scale (q) I/B/P triplet of (4,8,4). All sequences contained 15 slices per frame and 150 frames except for *UnderSiege* which contained 731 frames. Each sequence had SIF resolution (352x240 pixels) with N=6 and M=2 which equals the number of frames between I frames and the number of frames between anchor frames respectively. We calculated metrics such as the mean picture density (D) and compression ratio (CR) in order to gain an understanding of the encoding effectiveness with different scene content. This was possible because each sequence was encoded in the same manner. We calculated statistics such as peak-to-mean ratio (PMR) and coefficient-of-variation (CV) in order to determine the amount of variability in the data. We compared the probability mass function (pmf) of each of the video sequences to the Gamma and Pareto distributions by using QQ plots. We also compared the steady-state distribution of an 8-state Markov Chain to the pmf. We showed the autocorrelation function and discussed the impact of I/B/P sequences on multiplexing and dynamic bandwidth allocation.

2. PREVIOUS WORK IN VIDEO MODELING

Simulations require effective source models in order to understand and identify the impact of compressed video sources on computer networks. Most of the work has focused on video conference type sequences and consequently did not include frames encoded using interpolation (B frames). The reasoning was that B frames required too much time to encode for real time video conference type applications. The majority of the models have been frame based in that cells for a particular frame are presented to the network at the beginning of a frame interval and transmitted within the interval with either random or uniform cell spacing. This is less accurate than a slice source model, but less demanding on simulation resources. Frame based models do not capture the effects of spatial content within a frame on queueing performance. It would be interesting to see if a slice based model produced significant changes in queueing performance when compared to a frame based model. In any case, a model must capture both the distribution and autocorrelation functions of the source in order to be useful.

There has been some discrepancy as to what type of distribution is appropriate for compressed video data. Some have shown that for video conference data the Normal distribution[11 12] was appropriate while others suggested the Gamma distribution[3 4 13]. The Gamma distribution also fits the cells/slice distribution well for low quality sources, but did not fit well for higher quality sources[13]. Some have pointed out that a more accurate distribution might be a combination of distributions (Gamma/Exponential[4] or three Gaussian[14]). One paper showed that a combined Gamma/Pareto distribution fits the left and right tail distributions well[15]. Recent work has focused attention on the distribution per frame type. Here again, discrepancies arise where the Log-Normal distributions was used for I, P and B frames[16] in one case while Weibull was used for P and B frames in another[17]. It would appear that the discrepancy is attributable to both the content and length of the video sequence studied and that a single distribution will not fit all video sequences[18].

Distributions capture the spatial characteristics of a video sequence, but not its temporal characteristics. For this, one has to observe the autocorrelation function at various lags. The autocorrelation function of a source is an important characteristic and can greatly affect the performance of queueing networks[19]. Early work indicated that the autocorrelation function decayed exponentially for video conference type data; thereby, suggesting that AR or Markov Chains might be appropriate. Other work indicated that the autocorrelation function consists of two parts; a fast decaying part for short lags and a slowly decaying part for long lags[20]. Empirical models, such as TES[22], have been shown to match the autocorrelation

function quite well; however, a workstation is required to determine model parameters by trial and error[1]. A recent study showed that the autocorrelation function did not decay exponentially, but had a hyperbolic decay typical of self-similar sources[15].

Early work used AR[4 11 23] processes to model single VBR sources. AR(2) models have been found to be sufficiently accurate for traffic studies; however, a DAR process based on a discrete multi-state Markov Chain was found to be more accurate for video conference data[4]. Markov Chains have been used to model multiple VBR sources in a multiplexed environment[12 24 25 26 27]. More sophisticated models have been developed for VBR sequences with scene changes which are not adequately modeled with single AR process. One model used two AR processes and two complementary processes (used to determine the occurrence of a scene change) which are modulated by a three state Markov Chain[20]. Another used multiple AR processes, one for the number of block per field and a second for the number of bits per block[28]. In all of these models only I and P frames are included, not B frames. One paper suggested that ignoring B and P frames could severely under estimate cell loss rates[17]. Recent work has focused on modeling each frame type individually, cycling through each model based on the video frame sequence pattern (e.g. IBBPBBP...)[16 17].

3. EXPERIMENTAL INVESTIGATION OF MPEG SLICE STATISTICS

3.1. Statistical data

Table 1 shows the cells/slice statistics for the four video sequences. The peak number of cells/slice was determined by I frames. One can see that the peak-to-mean ratio (PMR) is positively correlated with both compression ratio (CR) and coefficient-of-variation[2] (CV) in that they increase with PMR. *UnderSiege* had the highest PMR while *Flowg* had the lowest. *Flowg* also had the lowest compression ratio while *UnderSiege* had the highest. Frame density (D = bytes/frame) was inversely correlated to PMR and CV. Note that for all videos, the minimum cells/slice is one standard deviation away from the mean while the peak is multiple standard deviations away. At the Slice layer, CV was significantly higher than typical values obtain at the frame layer which can be anywhere from 0.2 to 0.5. This was due to the fact that, at the Slice layer, spatial variations within a frame are included in the statistics. Since all of the videos were encoded using the same method, the statistical differences must be due to content.

Video	**Peak**	μ	σ	ν	CV	**PMR**	**D**	**CR**
Bike	40	6.44	7.04	49.56	1.09	6.21	4,284	29.58
Flowg	105	26.61	25.70	660.49	.97	3.95	18,797	6.74
Tennis	53	12.02	13.12	172.13	1.09	4.41	8,306	15.26
UnderSiege	40	4.46	4.98	24.80	1.12	8.96	2,848	44.50

TABLE 1. MPEG ATM Cells/Slice statistics

Figure 3 shows the first 300 slices (20 frames) of each video sequence. We can see the distinctive peaks and valleys caused by the significant difference in size between I,P and B frame slices. The peaks are caused by I and P frame slices and the valleys are caused by B frame slices. The shape of the peak is a result of spatial content within a frame.

1. Work is apparently in progress to automate this process by using neural network techniques.
2. Calculated using the formula σ/μ

3.2. Distribution comparisons

We first determined if these sequences would fit the Gamma distribution. To determine goodness-of-fit, we used the QQ plot[29]. For the Gamma distribution, we used the continuous version of the Gamma probability density function (pdf) given by (1). The equations to calculate α and λ are shown in (2) and are derived using the Method-of-Moments[30] where ν is the variance μ is the mean of the sample data. Parameter estimates for all of the video sequences is given in Table 2.

$$f(x) = \frac{\lambda (\lambda x)^{\alpha - 1}}{\Gamma(\alpha)} e^{-\lambda x} \tag{1}$$

$$\alpha = \frac{\mu^2}{\nu} \qquad \lambda = \frac{\mu}{\nu} \tag{2}$$

Video	α	λ
Bike	0.8345	0.1297
Flowg	1.0716	0.0403
Tennis	0.8389	0.0698
UnderSiege	0.8034	0.1800

TABLE 2. Gamma pdf parameters.

Next we tried to determine if the Pareto distribution will provide a better fit. Here we used the generalized form of the Pareto distribution shown in (3).

$$f_p(x) = \frac{\Gamma(\alpha + k)\, \lambda^{\alpha} x^{k-1}}{\Gamma(\alpha)\, \Gamma(k)\, (\lambda + x)^{k+\alpha}} \qquad x, \alpha, \lambda, k > 0 \tag{3}$$

We set both α and λ to 1 and adjust the third parameter, k, until a good fit was achieved. We can see from Figure 5 that the distributions for all sequences have significantly long tails. Also, all seem to have an exponential shape except for the *Flowg* sequence which has a more typical Gamma shape. None of the sequences appeared to fit the Gamma distribution well as indicated by the QQ plots in Figure 6; however, the Pareto function showed a better fit.

We then used an 8-state discrete-time, discrete-state Markov Chain to determine if its steady-state distribution gave a better fit. The number of states was selected based on the number of standard deviations from the minimum cells/slice to the maximum. For *UnderSiege*, this turned out to be about 8 standard deviations so we used 8 states for all sequences. Each state represented no greater than one standard deviations worth of cells/slice. For example, state 0 for *Bike* represented a range 1 to 5 cells/slice (40/8) rather than 1 to 7 cells/slice. The transition probability matrices show that there is a high probability of the sequence staying in the same state and there is a high probability of being in state 0. This is caused by the frame type sequence pattern. Steady-state was reached after 37, 46, 50 and 80 transitions for *UnderSiege*, *Bike, Flowg,* and *Tennis* respectively. Figure 2 shows the transition probability matrix for one of the sequences.

.

i \ j								
	0.9143	0.0564	0.0160	0.0056	0.0028	0.0021	0.0021	0.0007
	0.2428	**0.5405**	0.1358	0.0462	0.0116	0.0116	0.0087…	0.0029
	0.0510	0.3214	**0.4082**	0.1480	0.0561	0.0051	0.0102	0
	0.0915	0.0423	0.3028	**0.3944**	0.1197	0.0493	0	0
	0.1667	0.0500	0.0667	**0.3833**	0.2000	0.1167	0.1067	0
	0.1463	0.1220	0	0.1707	**0.2927**	0.1951	0.0488	0.0244
	0	0.0400	0	0.0400	0	0.4400	**0.4800**	0
	0	0	0	**0.5000**	0	0	**0.5000**	0

FIGURE 2. Markov Chain transition probability matrix for *the Bike v*ideo.

3.3. Autocorrelation function

The autocorrelation function, shown in Figure 4, was most interesting in that for all sequences it is quasi-periodic with a slow negative decay. Peaks occur every 45 slices which are caused by I and P frames. Correlations of between 0.4 to 0.5 are seen at lags of 5000 for *UnderSiege* and 1100 for the other videos which is statistically significant. Figure 9 gives a closer look at the shape of the autocorrelation function up to a lag of 350 slices. It would appear that the frame sequence pattern greatly influences the structure of the autocorrelation function rather than the specific spatial content of each frame.

None of the distributions proposed will produce the autocorrelation given by these sequences. Each distribution assumes that each sample is independent and memoryless which produces an autocorrelation which drops off quickly at short lags. It might be possible to distort the samples of the Pareto distribution in order to reshape the autocorrelation function. Also, the Markov Chain could be seeded periodically in order to change the autocorrelation function. We have seen that this tends to reshape the autocorrelation function to exhibit the peaks and valleys shown in the sample data. Since the autocorrelation function appears to be strongly influenced by the frame type sequence pattern it appears that a source model based on frame type would be more appropriate.

3.4. Dynamic bandwidth allocation issues

There is an abrupt increase in the number of cells/slice when the sequence goes from B frames to I or P frames. If a Traffic Predictor is monitoring the data link at frame intervals it might mistakenly request additional bandwidth. This would especially be true if the Traffic Predictor requests additional bandwidth when the bandwidth increased by more than one standard deviation[13]. Using a trigger point (point of significant bit rate change) of one standard deviation might be appropriate for sequences containing I and P frames which is typical of sequences used in video conference type studies, but it does not appear to be appropriate for I/B/P sequences. For I/B/P sequences, the Traffic Predictor can base decisions on differences of the aggregate size of multiple adjacent frames, possibly comparing the size differences between adjacent Group-of-Frames (GOP). This would reduce the responsiveness of the Traffic Predictor (depending on frame rate) to the range of about 0.1 to 0.5 seconds, making it less susceptible to false triggers.

4. MULTIPLEXING

Looking at the cells/slice data shown in Figure 3 one cannot help but notice that the distinct peaks and valleys generated by an I/B/P stream will cause problems when multiplexing these sequences. Since the sequences are quasi-periodic and not random in terms of arrivals, then multiplexing them improperly could result in significant inefficiencies in bandwidth utilization. If the sequences are not properly multiplexed then significant variation could occur in the resultant video stream. This is illustrated in Table 1 which shows that a staggered multiplexing of the *Bike*, *Flowg* and *Tennis* sequences resulted in a CV of 0.48, and 0.91 when no staggering was used. This is almost a two-fold increase! Also, if peak rate allocation is used, the non-staggered stream would require 41% more bandwidth than the staggered stream. Since the video sequences have a strong periodic component, this problem will not diminish over time and the traffic stream will not smooth out. However, by staggering the video streams one can achieve a certain degree of smoothing and the resultant stream assumes a more classical Gamma shape which can be seen in Figure 7. Note that the QQ plot shows a good fit. Figure 7 contains the histogram when no staggering is done and shows two distinctive gaussian distributions; the first with high mean and small standard deviation and the second with low mean and large standard deviation..

Video	**Peak**	μ	σ	ν	σ/μ	**PMR**
$TSM_{stagger}$	115	44.48	21.55	464.40	0.48	2.59
$TSM_{no\ stagger}$	162	45.06	40.87	1670.36	0.91	3.60

TABLE 3. MPEG ATM multiplexer cells/slice statistics

We staggered the video sequences by frame type in order to insure that I and P frames from one video did not overlap I and P frames of another video. In this case, *Flowg* was staggered by 15 slices and *Tennis* by 30. One could surmise that sequences which have a large number of B frames would provide increased multiplexing gain. We called this method *Time Shifted Multiplex (*TSM). A video server would inspect values of N and M in order to determine the proper stagger sequence. This method does require the multiplexer to be cognizant of frame type. This does not seem unreasonable, especially when one considers an environment where users are stopping playback, fast-forwarding, rewinding, etc. The video server could look at the current frame type pattern mix and decide on the best time to start playback. It does not appear that this method can be used dynamically throughout playback since this would require frames to be delayed by the video server in order to insure proper frame overlap. This can cause problems when frames are not delivered to the client in time for playback, thereby causing frame slips. It might be possible to delay frames within specified time constraints in order to avoid this problem.

5. CONCLUSION

We have shown that the video sequence compression ratio was greatly affected by its content. *Flowg* had the lowest compression ratio while *UnderSiege* had the highest. Peak-to-mean ratio was found to be correlated with coefficient-of-variation and compression ratio. Also, the coefficient-of-variation at the Slice layer was higher than at the Frame layer. The Pareto distribution was found to provide a better fit than Gamma and an 8-state Markov Chain produced a pdf which fits the video data well. While it appears that a single distribution does not fit data at the Frame layer the Pareto distribution provided a good fit for the four sequences at the Slice layer.

The autocorrelation function was quasi-periodic with negative decay and was not exponential or hyperbolic. Therefore, it would appear that models assuming memoryless properties or independence of arriv-

als would not be appropriate. It might be possible to distort samples from the Markov Chain or Pareto distributions in order to obtain an autocorrelation function with a similar shape. The autocorrelation function appears to be strongly influence by the frame type pattern and the packetization process. This would indicate that a slice model based on frame type might be more appropriate.

We showed that a Traffic Predictor, used in dynamic bandwidth allocation schemes, must take into account the significant frame size differences between I and B frames. This could cause the Traffic Predictor to incorrectly request more bandwidth at GOP boundaries (B to I frame). We proposed that bandwidth decisions be based on longer time increments, such as GOP times. Monitoring the size of I frames only would also provide a good indicator for bandwidth changes.

Finally, it was shown that care must be taken when multiplexing I/B/P sequences. If frame type is not taken into account, then increases in coefficient-of-variation and, therefore bandwidth requirements could result. We showed that coefficient-of-variation doubled and peak bandwidth increased by 41%. Once I and P frames from multiple video streams overlap they will continue to overlap throughout playback if they have the similar values of N and M. We proposed a multiplexing method called TSM which multiplexes by frame type and showed that frame type plays a significant role in video multiplexing.

Future work will involve studying a wider variety of VBR encoded video sequences in order to determine a slice based model which is consistent for all sequences and matches the autocorrelation function shown. We would like to determine if slice based models provide any significant difference to frame based models when estimating effective bandwidth and buffer requirements. We would also like to explore multiplexing of I/B/P sequences further which would use TSM.

6. ACKNOWLEDGEMENTS

We would like to thank the Portable Video Research Group for providing the four video sequences as well as encoding information. We would also like to thank Brian Smith, from the University of California at Berkeley, for providing the *mpeg_play* software decoder and information regarding MPEG decoding. We would also like to thank Rex Dwyer for his help and Mark Garrett for his comments and advice on video modeling.

7. REFERENCES

1. M. Kawarasaki and B. Jabbari, “B-ISDN Architecture and Protocol,” IEEE J. Select. Areas Commun., vol. 9, no. 9, pp. 1405-1415, December 1991.
2. D.G. Morrison, “Variable Bit Rate Video Coding for Asynchronous Transfer Mode Networks,” Br. Telecom Technol. J. (UK), Vol. 8, No. 3, July 1990, pp 70-80.
3. W. Verbiest, L. Pinno, and B. Voeten, “The Impact of the ATM Concept on Video Coding,” IEEE J. Select. Areas Commun., vol.6, no. 9, pp. 1623-1632, December, 1988.
4. D. Heyman, A. Tabatabai, and T. V. Lakshman, “Statistical Analysis and Simulation Study of Video Teleconference Traffic in ATM Networks,” IEEE J. Select. Areas Commun., vol. 2, no. 1, pp. 49-59, March 1992.
5. M. Sidi, et al.,”Congestion Control Through Input Rate Regulation,” in Proc. IEEE GLOBECOM ‘89, pp 49.3.1-49.3.5.
6. F. Kishino, K. Manabe, Y. Hayashi, and H. Yasuda, “VBR Coding of Video Signals for ATM Networks,” IEEE J. Select. Areas Commun., vol. 7, no. 5, pp. 801-806, June 1989.
7. M. Ghanbari, “Two-layer Coding of Video Signals for VBR Networks,” IEEE J-SAC, Vol. 7, No. 5, pp771-781, June 1989.
8. M. Ghanbari and V. Seferidis, “Cell-Loss Concelament in ATM Video Codecs,” IEEE Trans. Cir. Sys. Video Tech., Vol. 3, No. 3, June 1993, pp 238-247.

9. ITU(CCITT) Recommendations, B-ISDN (ATM), I.363 - Section 6, July 1993.

10. Motion Pictures Expert Group, “Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to about 1.5 Mbits/sec,” ISO 2-11172 rev, Part 2: Video, November 22, 1991.

11. M. Nomura, T. Fujii, and N. Ohta, “Basic Characteristics of Variable Rate Video Coding in ATM Environment,” IEEE J. Select. Areas Commun., vol. 7, no. 5, pp. 752-760, June 1989.

12. P. Sen, B. Maglaris, N. E. Rikli, and D. Anastassiou, “Models for Packet Switching of Variable-Bit-Rate Video Sources,” IEEE J. Select. Areas Commun., vol. 7, no. 5, pp. 865-869, June 1989.

13. P. Pancha and M. El Zarki, “A Look at the MPEG Video Coding Standard for Variable Bit Rate Video Transmission,” INFOCOM ‘92, pp 85-94.

14. R. Kishimoto, Y. Ogata, and F. Inumaru, “Generation Interval Distribution Characteristics of Packetized Variable Rate Video Coding Data Streams in an ATM Network,” IEEE J. Select. Areas Commun., vol. 7, no. 5, pp. 833-841, June 1989.

15. M.W. Garrett and W. Willinger, “Analysis, Modeling and Generation of Self-Similar VBR Video Traffic,” SIGCOMM ‘94, pp 269-280.

16. J. Enssle, “Modelling and Statistical Multiplexing of VBR MPEG Compressed Video in ATM Networks,” To Be Published ‘94.

17. D. P. Heyman, A. Tabatabai and T.V. Lakshman, “Statistical Analysis of MPEG2-Coded VBR Video Traffic,” 6th Intl. Workshop on Packet Video, pp B2.1 - B2.5.

18. D.P. Heyman and T.V. Lakshman, “Source Models for VBR Broadcast-video Traffic,” INFOCOM ‘94, pp664-671.

19. M. Livny, B. Melamed, and A. K. Tsiolis, “The Impact of AutoCorrelation on Queueing Systems,” Management Science, vol. 39, no. 3, pp. 322-339, March 1993.

20. F. Yegenoglu, B. Jabbari, Y. Q. Zhang, “Motion-Classified Autoregressive Modeling of Variable Bit Rate Video,” IEEE Trans. on Circuits and Systems for Video Tech., vol. 3, no. 1, pp 42-53, February 1993.

21. B. Melamed and B. Sengupta, “TES Modeling of Video Traffic,” IEICE Trans. Commun., vol. E75-B, no. 12, pp 1292-1300, December 1992.

22. R. Grunenfelder, J. Cosmas, S. Manthorpe, and A. Odinma-Okafor, “Characterization of Video Codecs as AutoRegressive Moving Average Processes and Related Queueing System Performance,” IEEE J. Select. Areas Commun., vol. 9, no. 3, pp. 284-293, April 1991.

23. B. Maglaris, D. Anastassiou, P. Sen, G. Karlsson, and J. Robbins, “Performance models of statistical multiplexing in packet video communications,” IEEE Trans. Commun., vol. 36, no. 7, pp. 834-844, July 1988.

24. S. Huang, “Modeling and Analysis for Packet Video,” GLOBECOM 1989, pp 881-885.

25. Y. Yasuda, H. Yasuda, N. Ohta, and F. Kishino, “Packet Video Transmission through ATM Networks,” GLOBECOM 1989, pp. 876-880.

26. P. Skelly, M. Schwartz and Sudhir Dixit, “A Histogram-Based Model for Video Traffic Behaviour in an ATM Multiplexer,” IEEE/ACM Transactions on Networking, Vol. 1, No. 4, pp. 446-459, August 1993.

27. B. Jabbari, F. Yegenoglu, Y. Kuo, S. Zafar, and Y. Q. Zhang, “Statistical Characterization and Block-Based Modeling of Motion-Adaptive Coded Video,” IEEE Trans. on Circuits and Systems for Video Tech., vol. 3, no. 3, pp 199-207,June 1993.

28. R. Gnanadesiran, *Methods for Statistical Data Analysis of Multivariate Observations*, John Wiley & Sons, 1977.

29. Mendenhall, Scheaffer and Wackerly, *Mathematical Statistics with Applications*, 2 nd edition, Duxbury Press, 1981.

30. Pancha and M. El Zarki, “Bandwidth-allocation Schemes for Variable-Bit-Rate MPEG Sources in ATM Networks,” IEEE Trans. on Circuits and Systems for Video Tech., vol. 3, no. 3, pp 190-198, June 1993.

31. B.C. Arnold, “Pareto distribution,” Encyclopedia of Statistical Sciences, Vol. 6, New York: Wiley, pp 569-574.

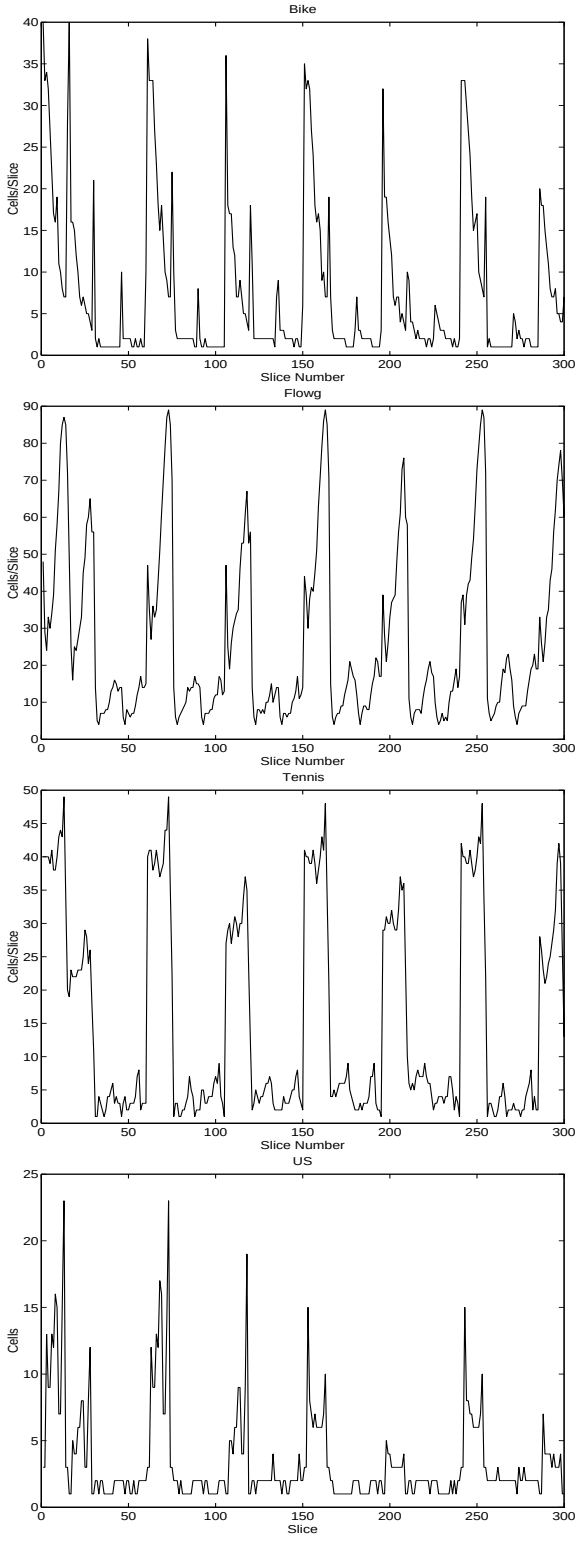


FIGURE 3. Cells/slice plot of first 300 slices from *Bike*, *Flowg*, *Tennis* and *UnderSiege*.

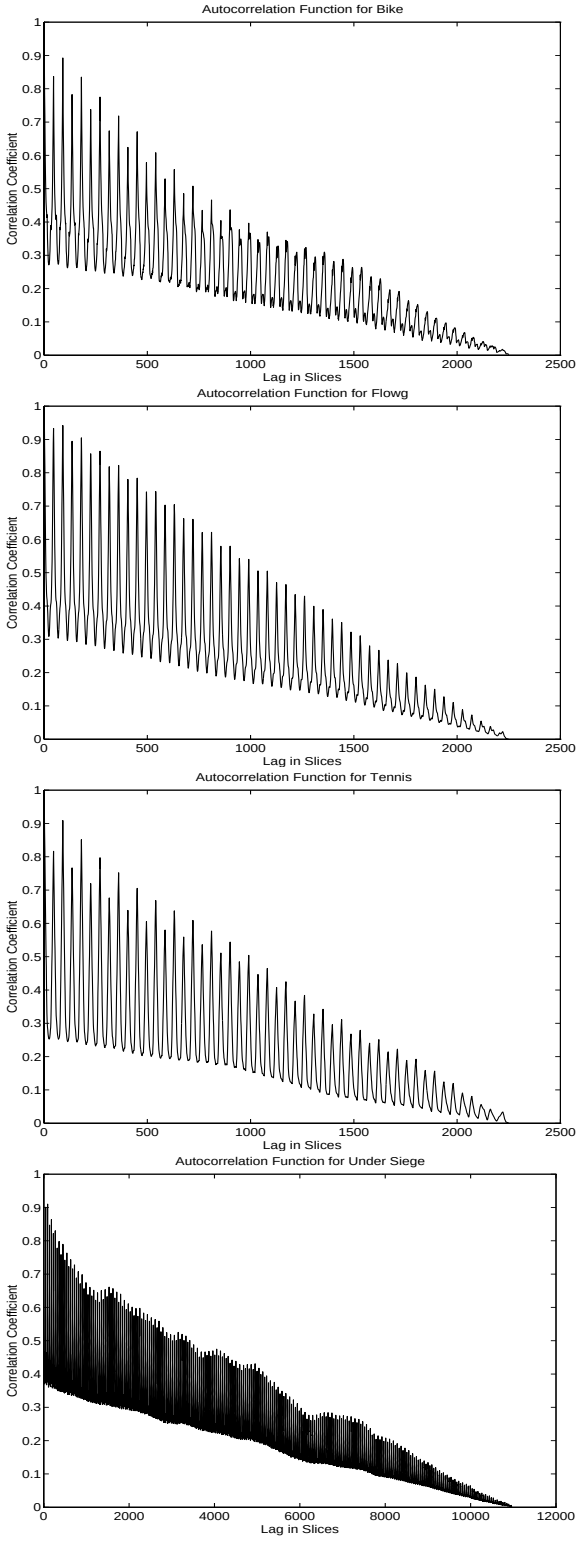
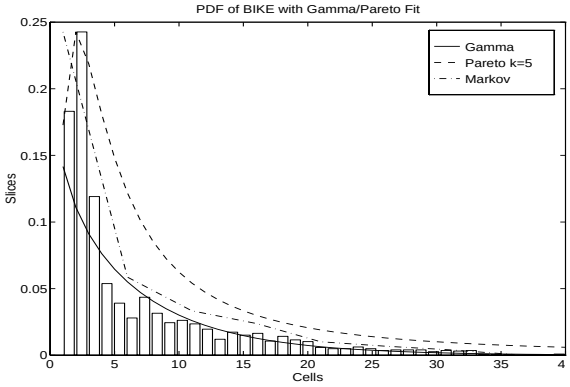
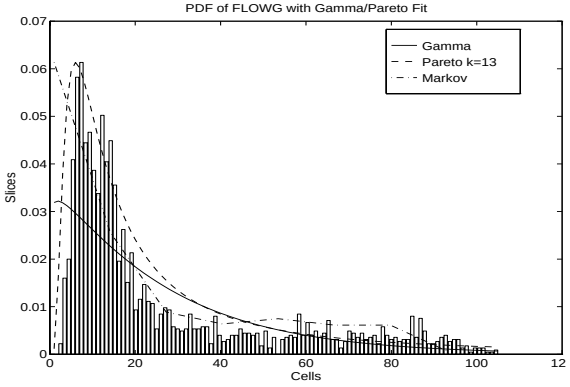
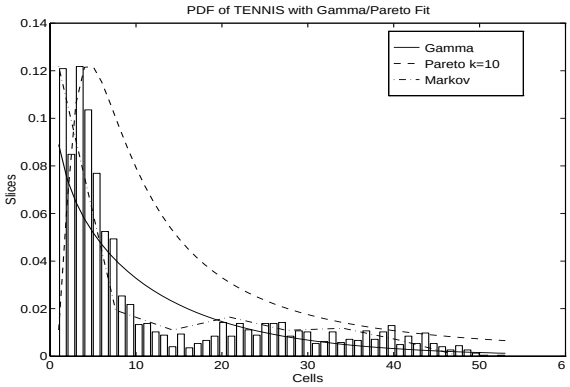


FIGURE 4. Autocorrelation function for *Bike*, *Flowg*, *Tennis* and *UnderSiege*.







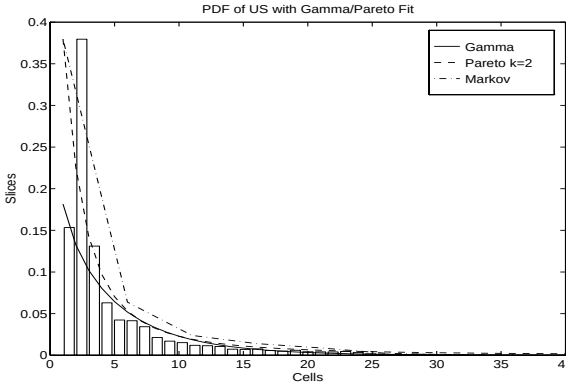
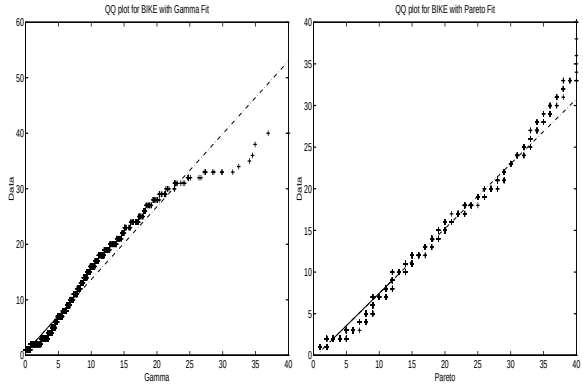
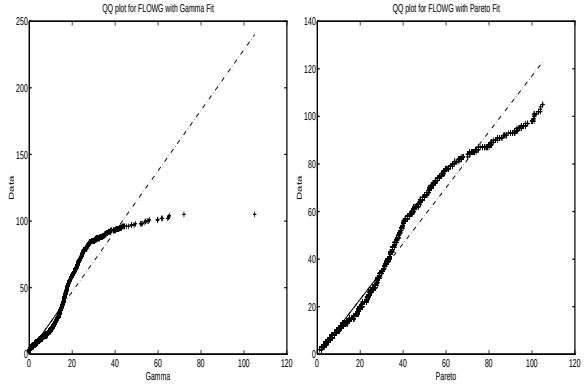
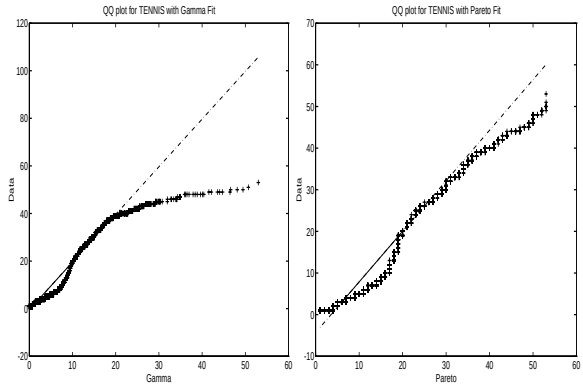


FIGURE 5. Cells/slice histograms of *Bike*, *Flowg*, *Tennis* and *UnderSiege*. with Gamma, Pareto and Markov pdf.







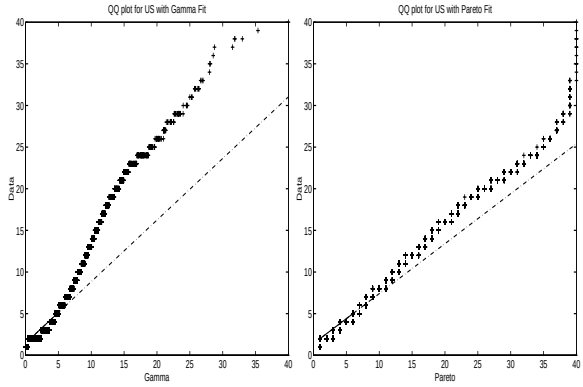
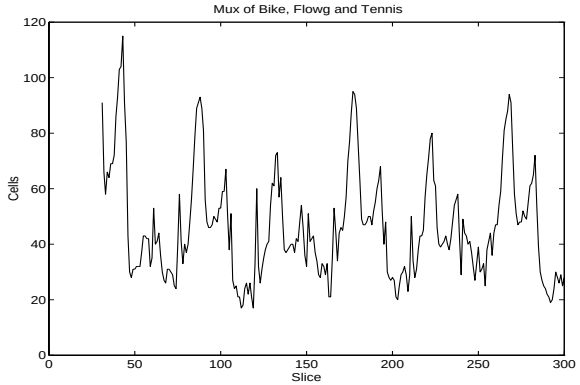
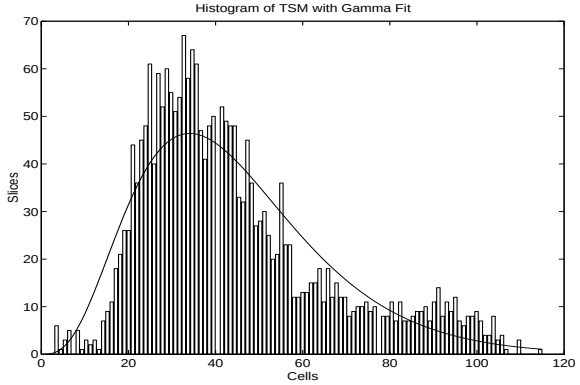


FIGURE 6. QQ plots of *Bike*, *Flowg*, *Tennis* and *UnderSiege* Gamma and Pareto distributions.





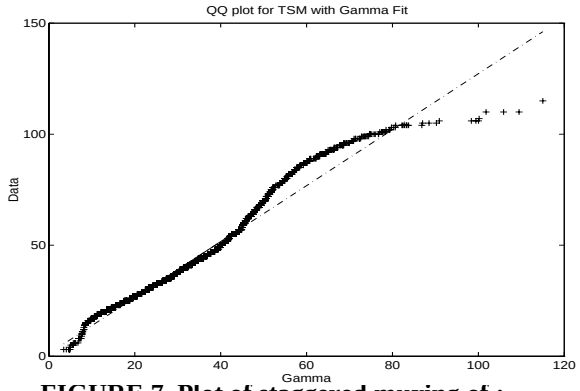


FIGURE 7. Plot of staggered muxing of ; Histogram with Gamma overlay and QQ plot.

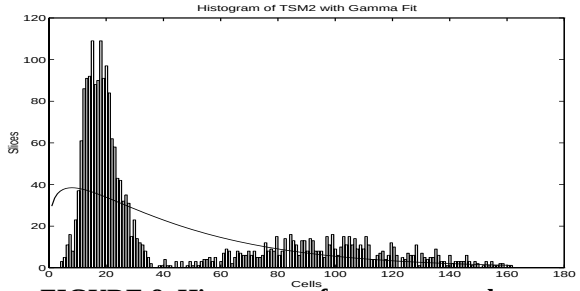
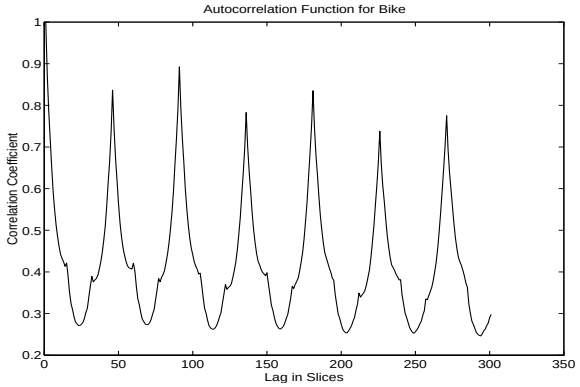
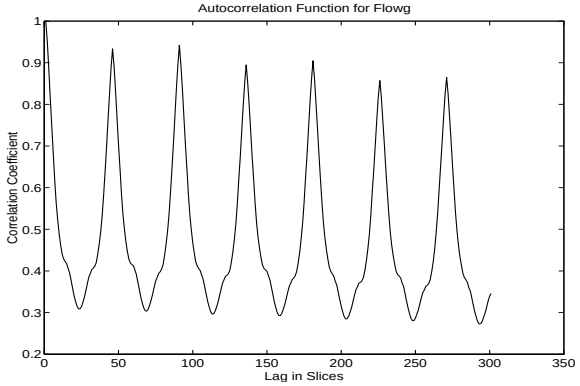
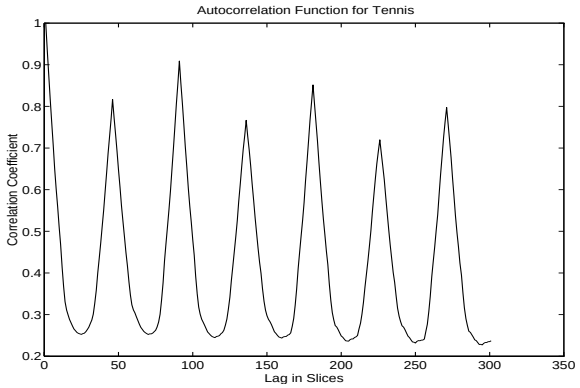


FIGURE 8. Histogram of non-staggered muxing of *Bike, Flowg, Tennis*.







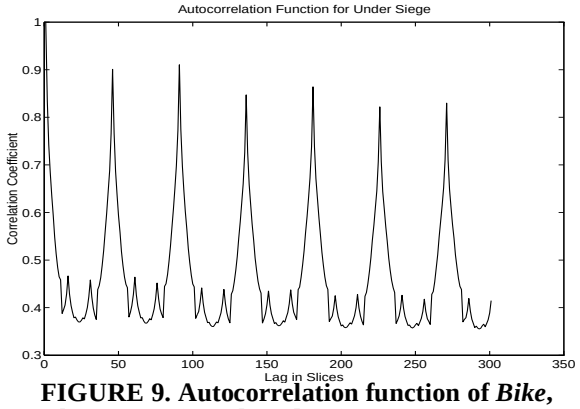


FIGURE 9. Autocorrelation function of *Bike, Flowg, Tennis* and *UnderSiege*.