Characterizing Variable Bit Rate Video Sources

Computer Systems Laboratory
Stanford University

August 31, 1996

Abstract

In this paper we survey the various statistics that have been used to characterize variable bit rate video sources. The insights provided by many of them are explored. We come up with novel ways of inferring information from some of them.

We suggest a new measure for the burstiness that is based on the observation, of how short term probability distribution functions for the bit rate per scene add upto give a probability distribution function for the whole sequence. Thus it is able to isolate the short term burstiness of a source.

User parameter control is often effected by using mechanisms like the leaky bucket. We look at two popular traffic descriptors from the view point of characterizing the traffic by their parameters. We argue that the sliding window and the leaky bucket are essentially equivalent measures. A simple conjecture of how the leaky bucket might behave in the case of self similar source traffic is also presented.

Finally we tie everything together by presenting the statistics obtained for the different type of codecs and some interesting analysis.

1 Introduction

Several coding algorithms have been proposed for variable bit rate (VBR) video. The variety in coding techniques has made the task of characterizing video sources difficult. We present a survey of statistical characteristics of some video codecs.

In networks like ATM which support the virtual circuit abstraction, with resource reservation, we need to characterize the traffic generated by the source so that we can reserve bandwidth efficiently. Thus a user informs the network using a traffic descriptor what the traffic would look like. The network would do call admission based on the traffic descriptor and the QoS required. And finally the network has to monitor the traffic to ensure that the user obeys the traffic descriptor it provided.

The choice of which statistics to use as traffic descriptor is still an open question. We here contribute to this selection, by providing insight into the kind of information that can be inferred from various statistics that are collected.

The rest of the paper is organized as follows. Section 2 gives an overview of the various kind of compression algorithm used by the popular codecs. It also describes the structure of the codecs. Section 3 describes the various statistics that have been proposed in literature and the inferences that can be made using them. In section 4 we propose two new statistics that help us to infer the burstiness of the source and justify their choice. Section 5 presents some observations made in the literature. Section 6 presents conclusions and future work.

2 Codecs

There are a wide variety of algorithms used for encoding of video. While standards like MPEG [6] are becoming widespread other non standard algorithms based on conditional replinishment are still popular. Most algorithms however are derived from one of the following base classes

2.1 Codecs with Conditional Replinishment

In a conditional replinishment system, the image is segmented into small regions. The easiest and the most prevalent method is to break up the image into fixed size blocks (like 8x8). When a new frame is captured, for each block a number known as the *distance* is calculated. If we represent the frame as a two dimensional array of blocks where F_n denotes the *nth* frame ,$F_n^{x,y}$ denotes the (x,y) block in the *nth* frame and $F_n^{x,y}(i,j)$ denotes the (i,j) the pixel in the corresponding block, then the *distance* for block (x,y) $d^{x,y}$ is calculated as

$$d^{x,y} \;=\; D(F_{n-1}^{x,y},\; F_n^{x,y}) \tag{1}$$

The choices of the function $D(.,.)$ are mean squared difference or taking the difference for one line in the block or just the absolute sum of differences of some random pixels. For example, for 8x8 blocks one possible distance function could be

$$D(F_{n-1}^{x,y},\; F_n^{x,y}) \;=\; \sum_{i=1}^{8}\sum_{j=1}^{8} |F_n^{x,y}(i,j) - F_{n-1}^{x,y}(i,j)| \tag{2}$$

Based on $d^{x,y}$ a decision is whether to transmit the block or not. If we decide that the *(x,y)th* block needs to be transmitted then either we can code $F_n^{x,y}$ which corresponds to *intraframe* coding or we can code $F_n^{x,y} - F_{n-1}^{x,y}$ which is *interframe* coding. Thus there are two thresholds involved CRT : conditional replinishment threshold and IIT : intra/interframe threshold [22], which are used as

$$\begin{array}{lllll} d^{x,y} & < & CRT & & \textit{do nothing} \\ CRT & \leq & d^{x,y} & < \; IIT & \textit{do interframe coding} \\ d^{x,y} & \geq & IIT & & \textit{do intraframe coding} \end{array}$$

Both systems that do inter-frame coding [14] and those that only use intraframe coding [5] [15] have been used. For systems that do only intraframe IIT is equal to CRT. The coding process generally consists of a transform stage, where the blocks go through a linear transform

Figure 1: Block Diagram for a Conditional Replinishment Coder

(and quantization) , followed by variable length coding like huffman or arithmetic coding. The bit stream is then packetized and sent over the network. At the receiving end the coding is inverted and the blocks are displayed on the screen. The block diagram is shown in figure 1. The degree of compression of this system depends mainly on the number of changed elements, and also on the method used to code them.

In practice instead of using $F_{n-1}^{x,y}$ in eqn. 2 we use $\overline{F}_{n-1}^{x,y}$, which is what a receiver would see after the block was encoded and then decoded. Typically there is a loss in quality so one would prefer to use this. Since the coded picture is highly compressed, motion detection in the pel domain (as in eqn 1) due to quantization noise might cause false motion detection. Motion detection can also be done after we have done the linear transform (i.e. in the transform domain) and been experimented with, in [14].

Interframe coders generally perform better than intraframe because they use temporal correlation between frames as well as spatial correlation within the frame, but the effect of transmission errors degrades the superiority since in the intraframe coding system errors are confined to a single frame, whereas in the interframe coders they propagate in further frames.

In block selection based algorithms there would be some blocks which don't change (for whom $d^{x,y}$ is always small). They have to be transmitted periodically in order to ensure that errors don't propagate and for example in a video conferencing system people who join the conference later do get the whole image. These blocks constitute the background. One approach to this could be that the blocks are sent periodically. Or we could optimize on this and send only those blocks that have not been sent recently because they were classified as motion blocks.

2.2 MPEG like Codecs

The MPEG [6] video compression algorithm (figure 2) was developed by the Moving Pictures Expert Group. It achieves compression by exploiting both the spatial and the temporal correlation in a sequence. The key idea in MPEG is block based *motion compensation*. The image is broken down into 8x8 blocks. Block based motion compensation is used for the reduction of temporal redundancy and transform domain based compression for the reduction of spatial redundancy. Motion compensation is also the basis of other compression algorithms for visual telephony like H.261 [12]. IT assumes that locally the current picture can be modeled as a translation of the picture at some previous time. Locally means that the amplitude and the direction of the displacement need not be the same everywhere in the picture. In the temporal dimension, motion-compensated interpolation is a multi-resolution technique: a subsignal with low temporal resolution (typically 1/2 or 1/3) of the frame rate is code and the full resolution is obtained by interpolation of the low-resolution signal and addition of a correction term. The signal and addition of a correction term.

In this approach the *distance* from all nieghbouring blocks (or some subset of them) is calculated and then block with the minimum *distance* is used to code the current block (in interframe mode). So here we have to send the index of the block used as a reference also. This is popularly known as the *motion vector* and the method is known as *Motion Compensation*. Motion compensation plays a significant role in interframe image coding. If one can estimate the motion displacement in successive frames, the difference between the current block and the motion-compensated block, either in the pel or transform domain becomes very small and thus leads to efficient coding.

The frame used to calculate the difference is not necessarily the previous frame. So for coding F_n we are not restricted to use only F_{n-1}. If we use only frames F_i such that $i < n$ then it is called *causal* or *pure predictive coding*. The frames for which this is used are called P frames. If we place no restriction on i i.e. it can be greater than n then it is called *non-causal* or *interpolative coding*. The frames for which this is used are known as called B frames. There are also I frames which are coded independent of any other frame. This serve as reference points for random access and for coding other frames.

To find which block has the minimum *distance* from a given block is a computationally expensive process. Thus for purposes of finding the reference block, 4 blocks are clumped together. This is also known as the *macroblock* and is the basic coding unit in MPEG. Thus the motion vectors are for 16x16 blocks. Also the search is limited to blocks that are a distance of $-16..+16$ pixels away. To minimize the overhead associated with transmitting the motion vectors, the motion information is compressed using variable length codes.

The organization of frames in MPEG is quite flexible and is not specified by the standard. One is free to choose how to organize the sequence of I ,P and B frames. This choice will depend on the application specific parameters such as random accessibility and coding delay.

Figure 2: Block Diagram for MPEG Coder

2.3 JPEG based Codecs

JPEG was developed by Joint Photographic Expert Group. It is a still picture compression standard. In a JPEG based codec each frame is coded as a separate picture. So this is actually a special category of MPEG based codecs where only *I* frames are sent [24]. That is we don't exploit the temporal redundancy present in a bit stream.

3 Existing Characterizations of Video Traffic

There is a wide variety of statistics that can be collected for the traffic generated by a video codec. In this section we present statistics that have been used in literature, and try to motivate what kind of inferences can be made from the collected information. Unless otherwise stated that the traffic is represented by a sequence of observations, $X_1, X_2, X_3 ... X_n$, where X_i is the bits in the *ith* observation period. The lengths of the observation period depend on the context. $\overline{X}$ denotes the mean of all the observations.

Peak to Mean Ration or *PMR* is often used as an important traffic descriptor for the burstiness. *PMR* is defined as

$$PMR = \frac{max(X_1, X_2, X_3 ... X_n)}{\overline{X}} \tag{3}$$

When circuits are statistically multiplexed the network reserves bandwidth so that at most times it can meet the demands of the different connections. Thus it can admit more calls as compared to a a situation where deterministic guarantees are given. *SMG* refers to the ratio of calls it can admit in the former scenario to the number it can admit when deterministic guarantees are given.

PMR bounds the statistical multiplexing gain (*SMG*) because , while a single source must be allocated approximately peak bandwidth, the allocation approaches the mean bandwidth as the number of combined sources increases.

A relevant and important question is the question of what intervals are peaks considered over. The peak to mean ratio can be varied dramatically by changing the interval over which the peaks are considered. Also the PMG should be taken with a grain of salt as the gains reduce dramatically when we impose constraints on the cell loss rates (*CLR*), which means constraints like no more than x% of the data should be lost.

An interesting way to see the effect of VBR traffic and how SMG occurs is shown in figure 3. We want bursty sources to add up to the bandwidth of the link (thus a cbr like distribution). The cartoons correspond to distributions in [10]. Here the graphs are showing the probability distribution for the values of X_i's. Thus a *VBR* source would have a distribution that is spread out while a *CBR* source would have a distribution that is a peak. Figure 3. shows how we expect *VBR* sources to add up to give us a *CBR* source.

The peak to mean ratio depends on the input video sequence used. For broadcast channels [10] the maximum bit rate was about 1.6 times the mean bit rate and about 4 times the minimum bit rate. Thus the SMG was limited by a factor of 2. [10] also observed that the pdf's of many broadcast sources which are individually bursty add upto give a pdf which has a sharp peak (like a CBR codec) .

[3] brings up an intersting point about using PMR as a measure for VBR gain. If scene changes are ignored a PMR of 1-3 is obtained, but if scene changes are included then a PMR of about 5 is observed. Since in normal operation scene changes are considered this would represent the calculated VBR gain. But in practice a CBR codec would increase quantization for getting better compression when there is a lot of motion as the error is less perceptible and would thus reduce the peaks without any noticeable degradation in quality thus reducing the peaks.

Inter Arrival Distributions here we plot the probability distribution of the the interarrival time between *events*. Let us consider a subset of observations E such that $X_i \in E$, if X_i satisfies some condition (like $X_i > \overline{X}$). Then we are interested in the distribution of $|i-j|$ such that $X_i,\ X_j \in E$ and $i < k < j,\ X_k \in E$ is true for no k.

For example the interarrival packet distribution is given when we define $X_i \in E,\ if\ X_i > 0$.(Here it is assumed that observations X_i correspond to individual packets). Often it is more instructive to plot inter arrival distributions of bursts. The inter-arrival time distribution can be used for figuring out a interval size for which the PMR ratio is appropriate or point out cases in which it doesn't mean anything and can be varied to almost anything.

Figure 3: Statistical Multiplexing Gain: The figures correspond to probability distribution function of bits per frame. It shows how several VBR sources add up to give an almost CBR source.

Figure 4: Cartoon of two probable distributions for the Interarrival times

To illustrate the point consider two Inter-arrival distributions for bursts figure 4 (a) would suggest that an interval that includes the first peak would be a good descriptor for the PMR ratio while the figure 4 (b) indicates that the PMR ratio doesn't mean anything and can be varied arbitrarily.

Bits per Frame or *BPF* Here we plot the X_i, where the observation period corresponds to the inter-frame time [1]. This shows what the output data looks like. An interesting comment made by [7] where the data is for the movie Star Wars, is that the bpf corresponds to the story line of the film, the action being intense in the introduction, then quite placid in the second quarter as the protagonist is developed and it picks up as the conflict progresses, pauses slightly and then builds to a climactic (and bandwidth rich) finale.

Peak Vs Averaging Interval Here we try to capture how the peaks vary with the averaging interval. Let $X_m^{(n)} = \sum_{i=(n-1)m+1}^{nm} X_i$. Then we plot $max(X_1^{(n)}, X_2^{(n)}, ...)$ versus n . One of the methods used to shape the traffic to reduce its burstiness is to use a buffer. This statistic shows for how large interval we need to buffer if one wants to limit the peak. On the flip side it gives you the delay that you should be willing to tolerate if one requires a particular peak bitrate.

One of the methods proposed [21] to do a fair comparison between *VBR* and *CBR* sources is to equate peak rate for *VBR* source to the mean rate for the *CBR* source. But a *CBR* codec like those which use buffer occupancy level as feedback generally has a delay associated with them, while *VBR* codecs are unconstrained and thus don't have delays. Thus to equate the delays we consider peaks that are calculated over the interval corresponding to the delay of the *CBR* code. So it also shows the delay that should be used if one wants to achieve a given statistical multiplexing gain. (this is especially true for on-off sources, thus for video it is a pessimistic view of the situation).

Minimum Vs Averaging Interval Here we try to capture how the minimum varies with the averaging interval. Let $X_m^{(n)} = \sum_{i=(n-1)m+1}^{nm} X_i$. Then we plot $min(X_1^{(n)}, X_2^{(n)}, ...)$ versus n. This can give us an idea of the CBR component of the traffic. One of the question of interest is how the time is divided. Do we use a window that can start at

[1] the observation period are assumed to be aligned with the frame generation epoch

any epoch or one that can start at only integer multiples of the averaging interval. One would expect this curve to be periodic with the period conforming to the period when the autocorrelation between the frames dies out.

Autocorrelation between Frames Here we plot the auto-correlation of the sequence of bits generated per frame with increasing interframe distance. It is a good measure of *persistence* [9] of a scene, *i.e.*how long a scene continues. If there are more than one type of frames like in MPEG, then oscillations are observed as there is correspondence between frames of a particular type. For example, in MPEG the plot peaks for every I frame sent.

[22] calculate the distribution of the duration of peak length versus the peak height. This along with the auto-correlation function can be used to see how much would smoothing of the video source helps. That is how much buffering would help us in reducing the peak rate. Thus if very high peaks come for periods that are large it would be difficult to smooth the traffic. It is pointed out by [22] and [18] that it is very difficult to absorb bit rate fluctuations of current by means of using an output buffer as is done in practice.

The autocorrelation function can also be used to estimate a good spacing for I frames in a MPEG sequence. [19] study the effect of varying the interframe spacing (N) between I frames in an MPEG coder and also varying the quantization factor (q) on various statistics. As N increases the average bits per frame decreases initially and then increases. The initial decrease occurs because prediction error can be coded very efficiently, but slowly the error accumulates and the efficiency of inter-frame coding goes down.

Probability Distribution Function for BPF Here we plot the probability distribution for X_i, where the observation period corresponds to the inter-frame time. This is a potentially exciting piece of information. For many codec this distribution tends to be bimodal and thus using only the mean value of BPF does not give us an accurate picture of what the traffic looks like. For example for an MPEG based codec we have three types of frames I , P and B , and the distribution especially for small number of frames is trimodal [2].

Also it gives us some idea of the burstiness of the traffic. Although it is not clear what should be considered more bursty among figure 5 (a) or figure 5 (b). The figures correspond to the distribution of actual MPEG coders with a varied quantization step and frame sequencing [9].

It has been observed that the pdf for the *BPF* for MPEG codecs [19] with (N=1, q=4) [3] that is without interframe coding is quite different from the one with (N=16,q=4) interframe coding. The average bit rate is higher and the the spread of the probability distribution is more. This has also been observed by [23] for video conferencing situations with a decreased resolution between frames *i.e.*is when we don't send frames at 30fps but send them at lower frame rates.

Leaky Bucket and Sliding Window have been used as traffic descriptors for B-ISDN networks. Traffic descriptors can be statistical or operational [1]. The statistical approach is the more conventional one in traffic theory, and focuses on statistical parameters, such as

[2]small meaning not involving a scene change

[3]N is the spacing between I frames and q is the quantization factor

Figure 5: Cartoon of two probable distributions for Bits per Frame

the long-term average and burst parameters mentioned earlier. Such an approach may be motivated by the apparent availability of methodologies for predicting performance of a network stressed by traffic with a given statistical parameters. However, associated with a statistical framework for traffic is the difficult task of verifying a set of statistical traffic parameters. By the very nature of their definitions, they require a lengthy observation intervals.

With the operational approach, focus is placed on traffic that is either just compliant or not compliant. This is achieved by equating a operational traffic descriptor with a parameterized compliance testing mechanism. The parameters for the leaky bucket and the sliding window can be used as characterizations for the traffic. We show that the information content of the characterization of traffic using the leaky bucket or the sliding window is the same.

Leaky Bucket A leaky bucket is a counter that increments by one for each cell emitted, upto a maximum value, and decrements at a given rate to as low as zero - a cell can be emitted if the counter is less than the maximum value.

The leaky bucket can be considered as an imaginary FIFO buffer of size N_{max} bits with a constant drain rate $\overline{R}$ bits per frame period. Let N_i be the bucket fullness (in bits) at the end of a frame period i. Since R_i bits arrive in the frame period i, $N_i = max\{0, N_{i-1} + R_i - \overline{R}\}$, if we ignore the finite capacity of the bucket. Thus the traffic would be in compliance if we chose the bucket capacity N_{max} such that $N_i \ < \ N_{max} \ \forall i$.

Sliding Window A sliding window specifies that no more than a given number of of bits (or cells) can be emitted in a time interval of a specified length, where the time interval can begin at any epoch. The sliding window can be described by two parameters, the time duration S_win, and the maximum number of bits that can be transmitted in that time window, W_{max}. An alternate description could use W_{max} and the negotiated average bit-rate, $\overline{R} = W_{MAX}/S_{WIN}$. Mathematically the constraint on the channel rate that is imposed by the sliding window is $\sum_{j=k+1}^{k+S_{win}} R_J \leq W_{max}, \ \forall k$.

Another way of looking at the leaky bucket is that the maximum bucket capacity N_{max} determines our averaging interval and over this period the average is the drain rate $\overline{R}$. Now the analogy between leaky bucket and sliding window definition becomes clearer. Leaky Bucket is a sliding window of size N_{max} with a history. $Bucket\ Size\ (N_{max})/Rate(\overline{R_{sw}}) = 2 * Averaging\ Time$ Where $\overline{R_{sw}}$ is the negotiated average rate for the leaky bucket.

And in this average time we can send twice the allocated rate because of the history. That is if we had not sent anything in the previous interval then we can send twice the size of the bucket in one interval. These are exactly the results obtained empirically by [21]. In the analysis of a sequence using sliding window and leaky bucket to capture it's burstiness if we start with a leaky bucket with a full instead of an empty bucket both contain same amount of information.

Also if the traffic is covariance stationary [4] [11], which some LAN traffic and video traffic has been shown to be, then since the mean of the observations remains fixed as we change the time scale that is the averaging interval, increasing the bucket size would give the same average rate to get complaint traffic. This has been observed by some other researchers for video traffic [20].

For **hierarchical or layered coding schemes** we would like to do all the above individually for each layer and also for the codec as a whole. The following are especially useful

The Bandwidth Per Layer We plot the fraction of the bandwidth generated by each layer of the codec versus various encoding parameters. One of the desirable qualities of a layered encoder is that the various layers in the hierarchy carry nearly equal amounts of data. Thus the distribution of data carried in a particular layer of the hierarchy as a fraction of the total is useful.

Although the norm used for the amount of data should not be the bit rate but an image enhancement function *i.e.*how much visual information is carried by the data stream. But this is tough to do although some progress in obtaining good distortion measures [2].

Bits per Frame for each layer of the codec should helps us to gather useful insights into the distribution of the data.For example it is useful to have the bits per frame that were generated by the interframe and the intraframe coding in a MPEG codec. As it has been observed that the bits per frame for a MPEG sequence tend to be similar to a JPEG like sequence, when a high quantization factor is used. One of the reasons that this might be true is the shift in the number of blocks that are intraframe coded because of the poor quality resulting form the high quantization used. The bits per frame statistics for the inter and intra frame bit rates would show this trend if it exists.

In designing two layer codecs one has the choice to choose the base layer to be constant bit rate (CBR) or variable bit rate (VBR) [16]. In most systems the CBR layer is implemented

[4] Let $X = (X_t : t - 0, 1, 2....)$ be a *covariance stationary* process - that is a process with constant mean $\mu = E[X_t]$, finite variance $\sigma^2 = E[(X_t - \mu)^2]$ and an autocorrelation function $r(k) = E[(X_t - \mu)(X_{t+k} - \mu)]/E[(X_t - \mu)^2]$ that depends only on k.

using the buffer occupancy level as the feedback mechanism [3]. The hierarchy in most systems have been based on spectral selection using DCT based algorithms [19] [14] [10].

However the pdf's for the bits per frame for each layer in many of these systems shows that the base layer without the use of feedback turns out to be CBR. This is different from the above as there is no delay involved as is generally associated with a CBR system with buffer occupancy level as a feedback. Thus as earlier stated we would not smooth out the peaks for the VBR sources to compare the multiplexing gain.

4 Proposed Characterizations

In this section we present two new statistics, and motivate our reasons for collecting them.

Probability Distribution for the Scaled Bits Per frame The distribution in figure 5 (a) shows what we believe arises form mixing the slowly varying changes in the video sequence and the rapidly changing things. If one were to consider the pdf over a small portion of an MPEG sequence one would observe that we have two - three peaks corresponding to the different frames. But if we are to consider larger periods where scene changes occur (or for example periods where the auto-correlation functions drops below 70% of the maximum) these peaks are smoothed out.

Figure 6 provides some insight into how this happens. 6 (a) is the bpf for the input sequence. We consider three portions of it and 6 (b) (c) (d) refer to the pdf for the individual pieces and (e) is the pdf for the entire sequence. Thus having a scene with higher information content would cause the curve to move to the right and viceversa. The shifts could be brought about also by a change in encoding parameters like the quantization factor.

Thus the pdf for the whole sequence tends to mix the short range and long range behavior. Which suggests that it might be more instructive to observe the pdf of the bits per frame which has been scaled by a slowly moving average, for example in MPEG the value of the bpf for the corresponding I frame. And we could use a window size based on how the auto-correlation function is shaped to calculate the moving average.

The R/S variation or the rescaled adjusted range scale statistics. This was original introduced by H.E.Hurst who applied it to data on the Nile river. The heuristic behind the R/S statistic is to capture the fluctuations in a given time series in order to size, for example, a dam so that it neither underflows or overflows given a (finite) empirical record. The problem is exactly analogous to sizing a buffer in a packet network with given input traffic. The R/S statistic looks at worst case behavior over a set to time and is formally calculated as follows. From a sequence of observations, $X_1, X_2, X_3 ... X_n$, we define a sequence of adjusted partial sums, $W_j = (X_1 + X_2 ... + X_j) - j\mu(n), j = 1, 2, 3...n$ where $\mu(n)$ is the mean of the first n observations. The adjusted range is the widest range of values observed upto time n.

$$R(N) = max(0, W_1, W_2, W_3....W_n) - min(0, W_1, W_2, W_3...W_n) \tag{4}$$

Figure 6: The Shifting and addition process

Normalizing the adjusted range by the sample standard deviation S(n) of $X_1, X_2, X_3 ... X_n$, we obtain the rescaled adjusted range statistic, $R(n)/S(n)$. Hurst showed empirically that for many naturally occurring time series the expected value of R/S asymptotically follows a power law

$$E[R(n)/S(n)] \approx n^H, \ as\ n \ \rightarrow \ \infty \tag{5}$$

This has been used recently [11] [7] to demonstrate the self similar nature of some traffic sequences. The exponent H in the equation 5 is close to 0.72 if the traffic has long range dependence and tends to be closer to 0.5 for traffic with no long range dependence. We propose the use of the R/S statistic by itself as a measure to characterize the burstiness of the traffic.

To build intuition for what the R/S statistics captures. Let us rewrite the W_j's as $W_j = \sum_{k=1}^{j}(X_k - \overline{X}(n))$. Let MAX denote the index such $max(0, W_1, W_2, W_3 .... W_n) = W_{MAX}$ and W_{MIN} be defined similarly. Now let us consider two cases

$MAX < MIN$ then the period between MAX and MIN denotes a period that we got a consistent period of low X_k's s.t. $X_k - \overline{X}(n)$ is negative. Thus if we are considering a leaky bucket (figure 7) with R as an approximation to $\overline{X}(n)$ and the incoming data rate to be X_k, then the period denoted by $[MAX, MIN]$ is when we accumulate tokens in the bucket.

Figure 7: Leaky Bucket

$MIN < MAX$ then the period between MAX and MIN denotes a period that we got a consistent period of high X_k's s.t. $X_k - \overline{X}(n)$ is positive. (Since we ignore the case when the min or max are zero) it denotes a period when we were draining the tokens from the bucket and the worst such drain.

Thus we are trying to capture the periods where we stress the network the most. The above values are scaled by the variance. To see why this should help, consider two sequences, one in which every alternate value of X_i is zero thus the sequence has a high variance. Consider another sequence where the which has the same number of zero X_i's and identical non-zero X_i's and thus the same variance, but the non-zero X_i's occur clumped together. The probability distributions for both would be identical. But the second sequence is more bursty and this would correspond to a higher R/S value for that time scale as compared to the first sequence.

5 Putting Statistics to Use

Some of the above statistics have been collected and analyzed in literature. In this section we present how some interest inferences could be derived from the statistics calculated.

For MPEG like codecs there are two components to the inter-frame coded data, the motion vectors and the coding of the prediction error. The bits per frame plotted for the two streams together is rather uninteresting. But as suggested *if the bpf for each layer is plotted separately* we can gain the following insight into the two streams. It has been shown [17] that the characteristics of the data stream for the motion vectors does not change with time. Infact it is a CBR stream with the mean being very accurately predicted by the average over a few frames. The prediction error for large-motion scenes to small-motion scenes can vary by a factor of 10. But in case of the small-motion the motion vector information is larger (than the prediction error), thus overall ratio is limited to like 3-4. This agrees with the ratios observed by [9] and [21]. Thus the statistical gain would be more if we separated the two components and identified the CBR component. This would also be true for the hierarchical codecs where the base layer by the choice of the parameters becomes CBR.

Another intresting fact that was obtained by studying the *probability distribution function for the motion vectors separately*, was that for video-conferencing situations [14] the pdf for motion vectors peaked at the origin which tends to suggest that in Video conferencing situation, as motion estimation is costly it can be eliminated and the blocks at the same position can be used for prediction. Studying the *pdf for the stream of indices for the blocks transmitted* by a conditional replinishment scheme [17] shows that motion is concentrated in the lower half of the frame. This is more true in television like sequences where the camera is being constantly focused on the subject. In situations like video-conferencing where the camera is fixed the motion tends to be quite well distributed.

For many hierarchical codecs there is a minimum amount of bandwidth that has to be assigned to the base layer for the heirarchical coding to be efficient. A plot of the *distribution of bandwidth per layer* with the encoding parameters brings out where the breakeven point is. It also helps us to reason about observed behavior, as in [14], where the tradeoff between bandwidth allocated to the base layer in a two layer codec versus the picture quality in the presence of packet loss is studied. More bandwidth allocated to the lower layer in case of packet loss increases the picture quality but reduces the SMG. The peak to mean ratio was found to be about 4-7. It was also noted that reducing the bit rate of each layer would reduce the total bit rate, but in cases where the first layer became highly distorted it inversely increased the bit rate. This study was for video conferencing type situations and the codec used was based on conditional replinishment with motion estimation and interframe coding.

Thus it is imperative to study the various streams in a layered codec separately to do a good job at characterizing the aggregate stream. Even if the codecs are not layered explicitly, but we use different coding techniques on different parts of a frame or in the temporal dimension (for example using different quantization matrices or quantization factors in MPEG and JPEG). Thus there are *logical streams* present in the overall steam. In order to characterize the overall stream we should study each stream separately. The study by [10] showed that for motion pictures with medium amount of motion the pdf for bits per frame tends to be similar to the Erlang distribution [5] [4] [8]. But for other higher motion scenes the distribution looks different as is expected because it is the sum of distributions generated by using different encoding parameters and entropy coding techniques on different parts of the sequence.

Although we have stressed the importance of studying various *logical streams* separately and trying to model the overally distribution form the individual distributions. An unstated assumption in the process is that the streams are not correlated. The *autocorrelation* for MPEG for example shows us that this need not be true. [19] have also considered the correlation the stream at the slice level [6] rather than at the frame level for the MPEG stream. They show that the oscillation observed in the auto-correlation for a sequence of bpf is considerably reduced thus it might be possible that the slices be treated independently. The *autocorrelation* between

[5] $p(x) = \frac{1}{(k-1)!}\frac{k}{\mu}\frac{(kx)^{k-1}}{\mu^{k-1}} \quad e^{-\frac{kx}{\mu}}$

[6] A slice is a horizontal strip within a frame, is the basic processing unit in the MPEG coding scheme. Coding Operations on blocks and macroblocks can only be performed when all pixels for a slice are available. A slice is an autonomous unit since coding a slice is done independently from its nieghbours.

bits per frame has led both [22] and [19] to suggest that smoothing to reduce the burstiness of the traffic doesn't help as scenes last for a long time. We feel this is partly correct as the measurements were made on a sequence which used the same encoding parameters throughout. It might not remain true if we increased the quantization factor to force a shift of the bpf curve as mentioned earlier.

The above list is not exhaustive and can't be. We have tried to show how some of the statistics help in gaining useful insights into the traffic generated by codecs.

6 Conclusions and Future Work

We have presented some novel ways of interpreting traditionally collected statistics. The effects on scene change and the *CLR* on using the *PMR* as an index of the statistical gain were brought out. We showed how we could use the autocorrelation function to find the averaging intervals and spacing between I frame in an MPEG based coder. Various uses of the pdf for the bits per frame were explained .

We have also proposed a new way to measure burstiness which isolates the short range statistics from the long range statistics. We also showed that the leaky-bucket and the sliding window when used as traffic descriptors are closely related. Finally we argued as to how a leaky bucket would model covariance stationary traffic.

Future work consists of validating many of the observations by collecting the required statistics for various codecs and quantifying some of the qualitative ideas presented in the paper.

References

[1] A. W. Berger and A. E. Eckberg. A b-isdn/atm traffic descriptor, and its use in traffic and congestion controls. *Proceedings of IEEE Infocom*, pages 266–270, Dec 91.

[2] Navin Chaddha and Teresa H. Y. Meng. Psycho-visual based distortion measure for monochrome image compression. *SPIE: Visual Communications and Image Processing*, 1993.

[3] Hin Soon Chin, John W. Goodge, Roy Griffiths, and David J. Parish. Statistics of video signals for viewphone-type pictures. *IEEE Journal on Selected Areas in Communication*, 7(5):826–832, June 1989.

[4] H.A. David and H.T.David. *Statistics: An Appraisal*. The Iowa State University Press, 1983.

[5] Ron Frederic. An internet posting: Some info on the nv 2.x packet format, December 1992.

[6] D. Le Gall. Mpeg: A video compression standard for multimedia applications. *Communications of the ACM*, 34(4):305–313, April 1991.

[7] M. Garett. Analysis, modeling and generation of self similar vbr video traffic. *Proceedings of ACM Sigcomm*, 94.

[8] Ulf Grenander and Murray Rosenblatt. *Statistical Analysis of Stationary Time Series.* John Wiley and Sons, New York, 1957.

[9] Hemant Kanakia, Partho P. Mishra, and Amy Reibman. An adaptive congestion control scheme for real time packet video transport. *Proceesings of ACM SigComm*, 1993.

[10] Fumio Kishino, Katsutoshi Manabe, Yasuhito Hayashi, and Hiroshi Yasuda. Variable bit rate coding of video signals for atm networks. *IEEE Journal on Selected Areas in Communication*, 7(5):801–806, June 1989.

[11] W.E. Leland, M.S. Taqqu, W. Willinger, and D.V. Wilson. On the self-similar nature of ethernet traffic (extended version). *IEEE/ACM TRacnsactions on Networking*, 2(1), Feb 1994.

[12] Ming Lou. Overview of the px64 kbit/s video coding standard. *Communications of the ACM*, 34(4):60–64, April 1991.

[13] Nasser M. Marafih, Ya-Qin Zhang, and R.L.Pickholtz. Modeling and queueing analysis of vbr coded video sources in atm networks. *Proceedings of the IEEE Infocom*, pages 325–329, 1992.

[14] M.Ghanbari. Two layer coding of video signals for vbr networks. *Journal of Selected Areas in Communication*, 7(5):771–781, 1989.

[15] Henry Moreton and Jeannine Smith. Hdcc - a software based compression algorithm for video conferencing, 1994.

[16] Samuel P. Morgan and Amy R. Reibman. Statistical multiplexing comparison of one- and two-layer video codecs for teleconferencing. *Proceedings of the 5th Internation Workshop on Packet Video*, March 1993.

[17] Ryozo Kishimoto nad Yoshifumi Ogata and Fumio Inumaru. Generation interval distribution characteristics of packetized variable rate video coding data streans in an atm network. *IEEE Journal on Selected Areas in Communication*, 7(5):833–841, June 1989.

[18] P. Pancha and M. El Zarki. A look at the mpeg video coding standard for variable bit rate video transmission. *Proceddings of IEEE Infocom*, May 1992.

[19] P. Pancha and M. El Zarki. A study of mpeg coding for vbr video transmission. *IEEE Communication Magazine*, 1994.

[20] Craig Partridge. Private communication.

[21] Amy R. Reibman and Arthur W. Berger. On vbr video teleconferencing over atm networks. *Proceedings of IEEE Infocom*, 1992.

[22] Willem Verbiest and Luc Pinnoo. A variable bit rate video codec for asynchronous transfer mode networks. *IEEE Journal on Selected Areas in Communication*, pages 761–770, June 1989.

[23] Willem Verbiest, Luc Pinnoo, and B. Voetan. The impact of atm concept on video coding. *IEEE Journal on Selected Areas in Communication*, 6(9):1623–1632, Dec 1988.

[24] Gregory K. Wallace. The jpeg still picture compression standard. *Communications of the ACM*, 34(4):31–44, April 1991.