Performance Analysis of a VBR Video Server with Gamma Distributed MPEG Data

Raúl V. Ramírez-Velarde, [1] and Ramón M. Rodríguez-Dagnino, [2]

Monterrey Institute of Technology (ITESM)
Sucursal de correos “J” C.P. 64849, Monterrey, N.L., México

ABSTRACT

In this paper, we propose analytical models to capture the statistical behavior of real traces of MPEG-4 encoded variable bit rate (VBR) video data in a video server. We study the scattered disk storage of video frames and periodic scheduling policies and we calculate the user disk service rate, buffer size, and the maximum number of simultaneous subscribers by using the Chernoff bound asymptotic technique. We have included a self-similar Gamma model which seems to be very close to the actual data behavior.

Keywords: VBR video server, Chernoff bound, self-similar video.

1. INTRODUCTION

The merging of computer technology and digital telecommunications has made it possible to develop many multimedia applications running on very high speed networks and reaching many homes in a cost-effective manner by using Internet such as interactive television, digital radio and television, remote shopping, and home and office automation.

The development of a Digital Video Server System (DVSS) results important not only for entertainment as a video rental center, but also for distance learning applications. With these, an individual can access any video from the catalog as long as the video is available in storage. In the DVSS the video is stored on disk in digital format, delivered through a high-speed telecommunications network on the frame-by-frame basis and we will assume in our analysis that the requested video will always be available for the subscriber.

In this paper we focus on investigating the buffer requirements and the following service rate parameters: a) The amount of time the disk subsystem allocates to each user as a function of the characteristics of the selected video file, b) the storage on disk of those files and c) the load of the whole system. We will develop and analyze several stochastic models that will allow us to establish the previous service parameters. These models are:

1. Gamma Service Cycle Mean
2. Strand Gamma Frame Size Distribution
3. Gamma Fractal Noise

The model named *Service Cycle* captures the variation of data size between delivery cycles (considering each video cycle a stream of constant-size video frames), which is similar to analyzing video traffic at a GOP level. The model named *Strand* uses information about the frame size probability distribution for the entire video segment. These models are compared with a *Fractal* model, which seems to be more appropriate for modeling these data, since it takes advantage of the self-similar nature of video information to estimate operation parameters of the DVSS. We will also develop easy to compute asymptotics based on the Chernoff bound that none-the-less provide very close approximations to the actual tail probability values. We compare the performance of these stochastic models with simulations of real traces generated by the MPEG-4 video encoder, which may be suitable for video services through wireless networks.

The rest of the paper is organized as follows: In section 2, we analyze the video storage technique of the video server. In section 3, we describe the video server scheme and the video encoder. In section 4 we discuss the main operational parameters, which are essential for determining the maximum number of users that can be serviced simultaneously. In section 5, we establish the mean service rate parameters typically used on the analysis of video servers. In Section 6, we briefly describe the Chernoff bound used in the paper, and the

[1] Computer Science Department. rramirez@itesm.mx
[2] Centro de Electrónica y Telecomunicaciones. rmrodrig@itesm.mx

calculations of these bounds for the Gamma distribution. Sections 7 and 8 constitute the core of this paper. In Section 7 we analyze the Gamma Service Cycle and Gamma Strand Frame Size stochastic models, whereas in section 8, we analyze the Gamma Fractal Noise model. Finally, in section 9 we make our conclusions.

2. STORAGE OF MULTIPLE VIDEOS

The video server stores the video files in a disk or in an array of disks. When a user requests a video file, the different frames that make up the movie are retrieved one-by-one in a sequential order. Once a video frame has been retrieved from the disk, it will be delivered through a network link to the user according to the playback rate stated in frames per second.

Our storage layout is the same as the one studied in [8]. Each block of $n_{v,i}$ display units (typically frames) is separated from the next and the previous block by exactly $I_{v,i}$ seconds for video strand i. The amount of display units $n_{v,i}$ is called *granularity* and the separation $I_{v,i}$ is called *interleaving*. The separation between blocks leaves a gap that does not necessarily lay empty as it is used to store other videos. The process of storing video files in every gap is called *merging*. Other important parameters are: a) $\overline{S}_{vf,i}$, which is the mean size of the video frame in bytes for video strand i, b) R_{dr}, the hard disk data transfer rate in bits/sec and c) $R_{pl,i}$, the replay rate required by video i in frames per second.

The scattering and the interleaving parameters define a storage pattern in bits of length $L_i = n_{v,i}\overline{S}_{vf,i} + I_{v,i}R_{dr}$ and it is characterized by the pair (M_i, G_i) where $M_i = n_{v,i}\overline{S}_{vf,i}$, and $G_i = I_{v,i}R_{dr}$, (see Fig. 1).

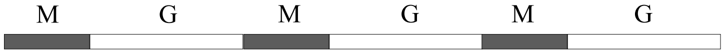


Figure 1. Storage pattern in bits.

3. DIGITAL VIDEO SERVER SCHEME

3.1. SAS scheme

The video server dedicates a separate buffer to each user i, which is attended during a service time l_i and this service time depends on the buffer depletion rate $\mu_{d,i}$. The mean depletion rate is given by $\overline{\mu}_{d,i} = \overline{R}_{pl,i}\overline{S}_{vf,i}$, where $\overline{R}_{pl,i}$ is the mean replay rate for user i. In the Switched Arrival System (SAS) scheme, service time l_i is enough for fetching and storaging in buffer i a number of frames to be sent to the user display subsystem. It will allow compliance with the play-back rate until the buffer completes a service cycle [2], see also Fig. 2.

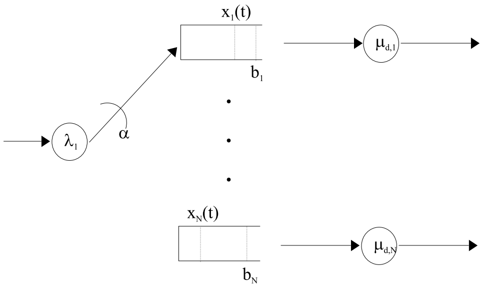


Figure 2. Switch Arrival Service (SAS) scheme of buffers.

In the SAS scheme with N users, each buffer i is associated with a service request of that particular user. Let us assume that each buffer i has a capacity b_i and $x_i(t)$ is the state of the buffer at time t. Let us also consider that $L_{seek,i}$ and $L_{rot,i}$ are the seek delay and plate rotation delay for request i. So, $\alpha_i = L_{seek,i} + L_{rot,i}$ is the switching time for user i. This is the time needed for the disk head to be positioned in the correct cylinder and allocation block. Similarly, $\mu_{d,i}(t)$ is the depletion rate as a function of time, whereas $\lambda_i(t) = R_i - \mu_{d,i}(t)$ is the fill rate for buffer i in bits per second, where $R_i = R_{dr}\, f_i$ is the disk throughput for user i, and f_i is a factor which depends on the switching delay between requests, interleaving and number of frames delivered. On the other hand, $l_i(r)$ is the service duration in a cycle r for user i. When the Service Scheduling Policy is periodic, we always have that $l_i(r) = l_i(r+1)$. Finally, we will call a period in which all pending service requests are serviced exactly one at the time a *Service Cycle*.

The disk subsystem must attend each request by filling the designated buffer with video frames either up to a certain level or for a fixed period of time. The order and manner in which these requests are attended is called a service scheduling policy. In general, a service scheduling policy $\Phi(r, \mathbf{x})$ is a function which returns a permutation of the order in which the buffers will be serviced depending on cycle r and the state $\mathbf{x} = \mathbf{x}(t)$ of all the buffers. The policy is called *open loop* if the permutation yielded by it depends on time or service cycle k, and it is called *closed loop* when the permutation depends on the state $\mathbf{x}$ of the buffers. If $\phi(t)$ is a buffer which will be serviced at time t, then the service scheduling policy is called *periodic* when $\phi(t) = \phi(t+T)$, where T is a constant service cycle time.

In this paper, we will consider the *Periodic Open Loop* (POL) [12] scheduling policy which serves each user i during a constant predefined time l_i, as opposed to the *Periodic Closed Loop* (PCL) scheduling policy which serves each client until $x_i(t) = b_i$.

In this SAS scheme the main focus is the empty state of the buffer. This means we are mainly concerned with characterizing the state in which the disk subsystem did not provide sufficient video frames for all buffers, that is, that the probability that one or some of the buffers will reach the empty state.

3.2. Video Encoder

The MPEG video encoder takes as its input stream a sequence of digitized frames, each of them containing a two-dimensional array of pixels (pels). In order to specify the data rate of the video encoder, it is important to know the number of frames per second, the number of lines per frame, and the number of pels per line. For each pel, both luminance and chrominance information is stored. The video strands used in this paper are encoded by using MPEG-4 [6], and they are stored in a disk by using a merging pattern. Similarly to MPEG-2, MPEG-4 is a layered encoding scheme. This means that the video data stream consists of a basic layer stream that contains the most important video data and one or more enhancement layers, which can be used to improve the quality of the video sequence. Unlike MPEG-1 and MPEG-2, which are frame based, MPEG-4 is object based. Each scene is composed of Video Objects (VOs) which are individually encoded. The scalability layers of each VO are called Video Object Layers (VOLs). Each VOL consists of an ordered sequence of frames called Video Object Planes (VOPs). For each VOP, the encoder makes the processing of shape, motion, and texture characteristics. Shape information is encoded by bounding the VO with a rectangular box and then dividing the bounding box into Macro Blocks (MBs). Each MB is classified as lying inside the object, on the objects border, or outside the object but inside the bounding box.

The titles of the video files used in this paper are: Jurassic Park, Silence of the Lambs, Star Wars IV, Mr. Bean, Star Trek: First Contact, From Dusk Till Dawn, The Firm, Die Hard III, Starship Troopers, Formula 1 car race, Alpine Ski and Soccer European Championship 1996. Each video file was recorded at a frame rate of 25 frames/sec, with a luminance resolution of 176×144 pels and 4:1:1 chrominance subsampling at a color depth of 8 bits. The encoding was done without rate control, so they are Variable Bit Rate (VBR) traces. The entire video stream is one VO and there is only one layer. The Group of Pictures (GOP) pattern was set to *IBBPBBPBBPBB*. The quantization parameters were fixed at 4 for I, B and P frames. There are roughly 90,000 frames per trace (see [4] for a more detailed description of the video files).

4. THE GLOBAL TIME BALANCE RELATIONS

The main parameters for the performance of the video server are: The service-rate constant k_i, the maximum number of simultaneous subscribers N_{max} and the user buffer size b_i.

Since all buffers are depleted in parallel, then the reading time of data blocks needed for the disk storage subsystem to replenish all buffers must be less than or equal to the minimum buffer emptying time. One buffer will empty faster than the others because either its size is smaller or its playback rate is faster. Assume that buffer i is filled with exactly $k_i n_{v,i}$ frames of size $S_{vf,i,j}$, where $j = 1, 2, ..., k_i n_{v,i}$. We may also assume that buffer i is filled with k_i blocks of video and that each block contains $n_{v,i}$ frames. To comply with the playback-rate demands, relation (1) must hold. We shall call the following inequality (1) the *Global Time Balance Relation*

$$\alpha(N) + \upsilon(N, \mathbf{k}) + \tau(N, \mathbf{k}) \leq \min\left(k_i \frac{n_{v,i}}{R_{pl,i}}\right) \tag{1}$$

where N is the number of concurrent users in a given time, $\mathbf{k} = [k_1, k_2, \ldots, k_N]$, $\alpha_T = \alpha(N) = \sum_{i=1}^{N} \alpha_i$ is the service cycle switching delay, $\upsilon(N, \mathbf{k}) = \sum_{i=1}^{N} \upsilon_i(k_i)$ is the service cycle interleaving time (the time the disk head spends over interleaving gaps) and $\tau(N, \mathbf{k}) = \sum_{i=1}^{N} \tau(k_i)$ is the time spent by the disk fetching frames for all subscribers in a service cycle. In our system, this translates to (see [11])

$$N(L_{seek} + L_{rot}) + \sum_{i=1}^{N}\left(k_i I_{v,i} + \frac{\sum_{j=1}^{k_i n_{v,i}} S_{vf,i,j}}{R_{dr}}\right) \leq \min\left(k_i \frac{n_{v,i}}{R_{pl,i}}\right). \tag{2}$$

The total time is the addition of video frames fetching time, plus the time spent by the disk head over the interleaving space, plus the switching time. The disk throughput for user i can be written as

$$R_i = R_{dr} f_i = R_{dr}\left[\frac{\tau_i(k_i)}{\alpha_i + \upsilon_i(k_i) + \tau_i(k_i)}\right] \tag{3}$$

as we mention above, factor f_i is the proportion of time that the disk spends fetching video frames for user i divided by the total time it spends servicing request i.

We also define the service cycle disk throughput as [3]

$$R_{sc} = R_{dr} f_{sc} = R_{dr}\left[\frac{\tau(N, \mathbf{k})}{\alpha(N) + \upsilon(N, \mathbf{k}) + \tau(N, \mathbf{k})}\right] \tag{4}$$

where f_{sc} is the proportion of time within a complete cycle that the disk takes in transferring video frames.

The ratio between the accumulated video data demand and the disk throughput (R) can be defined as the *system load* ρ. Namely,

$$\rho = \frac{1}{R}\sum_{i=1}^{N} \mu_{d,i}. \tag{5}$$

Another important parameter is N_{max}, which is the maximum number of subscribers that a scheduling policy Φ can service simultaneously.

Finally, $b_i = l_i \lambda_i$ is video data buffer for user i. Let us call φ_i the number of video frames fetched from disk for user i. Then, $l_i = \varphi_i / R_i$. Since $\lambda_i = R_i - \mu_{d,i}$, by having $R_i \gg \mu_{d,i}$, we get $b_i = \varphi_i$.

These operational parameters have the following properties:

Proposition I. There exist values $k_i = 1, 2, \ldots$ such that inequality (1) holds if and only if $\rho < 1$.

Proposition II. N_{max} is found by computing N when the service-rate constants, k_i become infinite.

Proposition III. There exist buffer capacity values called $b_{min,i} = l_i \lambda_i$, which are the stable long-term values of the buffer level $\mathbf{x}(t_m) = [x_1(t_m), x_2(t_m), \ldots, x_N(t_m)]$.

[3] On the rest of this paper, R_{sc} will be referred just as R.

5. MEAN SERVICE RATE PARAMETERS

In the *Quality Proportional (QP)* model the amount of blocks transferred to a customer is proportional to the playback rate $k_i = kR_{pl,i}$ [11]. We call k the proportional constant or more specifically the *Proportional Service-Rate Constant*. This constant specifies the disk throughput for each video strand, meaning that on the average $k\overline{R}_{pl}\overline{n}_v\overline{S}_{vf}$ bytes will be transferred to the user buffer in each service cycle.

We may now find for the mean disk throughput

$$\overline{R} = R_{dr}\left[\frac{\frac{k\overline{R}_{pl}\overline{n}_v\overline{S}_{vf}}{R_{dr}}}{k\overline{R}_{pl}\left(\overline{I}_v+\frac{\overline{n}_v\overline{S}_{vf}}{R_{dr}}\right)+\overline{\alpha}}\right]. \tag{6}$$

By balancing out the average disk throughput $\overline{R}$ with the average depletion rate of all buffers $\overline{\mu}_d = N\overline{R}_{pl}\overline{S}_{vf}$ and taking into consideration the stability condition $\overline{\rho} = \overline{\mu}_d/\overline{R} < 1$ (see Proposition I), where $\overline{\rho}$ is the mean system load, we can find k, say $\overline{k}$, as follows,

$$\overline{k} \geq \left\lceil\frac{N\overline{\alpha}}{\overline{n}_v - N\overline{R}_{pl}\left(\overline{I}_v+\frac{\overline{n}_v\overline{S}_{vf}}{R_{dr}}\right)}\right\rceil. \tag{7}$$

Using Proposition II the maximum number of concurrent subscribers is given by

$$\overline{N}_{\max} = \left\lfloor\frac{R_{dr}\overline{n}_v}{\overline{R}_{pl}\left(\overline{I}_v R_{dr}+\overline{n}_v\overline{S}_{vf}\right)}\right\rfloor. \tag{8}$$

And using property III, the buffer size for video strand i is given by

$$\overline{b}_i \geq \overline{k}\,\overline{R}_{pl,i}\overline{n}_{v,i}\overline{S}_{vf,i}. \tag{9}$$

6. CHERNOFF BOUND

The Chernoff bound can be used to find an approximate tail probability value. This computation is expressed in terms of an exponential probability decay function called the rate function in the theory of large deviations. We need to know

$$P(X_1 + \ldots + X_n \geq \varphi = na) \leq e^{-n\ell(a)} \tag{10}$$

for $a > \mathbf{E}[X_1]$, where

$$\ell(a) = -\ln\left(\inf_\theta e^{-\theta a}M(\theta)\right) = \sup_\theta\left(\theta a - \ln M(\theta)\right), \tag{11}$$

and $M(\theta) = \mathbf{E}\left[e^{\theta X_1}\right] < \infty$, for θ in some neighborhood of 0.

Suppose that we define $\phi = P(X_1 + ... + X_n \geq \varphi = na)$, ϕ could be for example the probability that the time required to deliver n frames $(X_1, \ldots, X_n)$ from disk will be greater that the allocated time φ. The problem can be stated as: "Find a suitable disk fetch time φ such that the disk subsystem will saturate, and thus fail to deliver the required amount of disk information at most with probability ϕ (very small)".

Now suppose that the video files are stored in such a way that the units of information (video frames) to retrieve from the disk have a Gaussian *pdf* (in accordance with the Central Limit Theorem) with parameters μ and σ. The rate function for such a random variable is

$$\ell(a) = \frac{1}{2}\left(\frac{a-\mu}{\sigma}\right)^2. \tag{12}$$

Hence,

$$a = \mu + \sigma\sqrt{\frac{-2\ln(\phi)}{n}}. \tag{13}$$

On the other hand, if we assume a two parameter Gamma *pdf* for the random variable of units of information (as suggested in [3, 5, 9 and 10]), with scale parameter λ and shape parameter α, where $\mu = \alpha/\lambda$ and $\sigma^2 = \alpha/\lambda^2$.

$$\ell(a) = a\lambda - \alpha - \alpha\ln\left\{\frac{a}{\mu}\right\}. \tag{14}$$

After some algebra we find that

$$a = \mu\left[1 + \ln\left\{\frac{a}{\mu}\right\}\right] - \frac{\ln(\phi)}{n\lambda} \tag{15}$$

which does yield an implicit solution for a. An approximation can be obtained if we redefine Eq. (15) as

$$a = \mu\left[1 + \ln\left\{\frac{a'}{\mu}\right\}\right] - \frac{\ln(\phi)}{n\lambda} \tag{16}$$

where a' can be obtained with the Gaussian approximation of Eq. (13). Thus, an approximated explicit solution for a is given by

$$a \approx \mu\left[1 + \ln\left\{1 + \frac{\sigma}{\mu}\sqrt{\frac{-2\ln(\phi)}{n}}\right\}\right] - \frac{\ln(\phi)\sigma^2}{n\mu}. \tag{17}$$

7. STOCHASTIC MODELS FOR SERVICE RATE PARAMETERS

7.1. Framework

In the following sections we assume that all stored video files have the same playback rate, granularity and mean frame size however, not necessarily the same variance and that the DVSS follows a *QP* service policy. We will also consider that the switching delay α is constant for all users and service rounds and deterministic interleaving.

In order to be able to define stochastic models for service rate parameters (mainly k), we begin by reformulating the global time balance relation (1), meaning

$$\alpha(N) + \upsilon(N,k) + \tau(N,k) \leq kn_v^{(\min)}; \quad n_v^{(\min)} = \min n_{v,i}\ \forall i \tag{18}$$

where k is the unique service-rate proportionality constant as established in *QP* model of Section 5.

The disk throughput is now

$$R = R_{dr}\left[\frac{\tau(N,k)}{\alpha(N) + \upsilon(N,k) + \tau(N,k)}\right]. \tag{19}$$

We also define, in accordance with Proposition II, the maximum number of subscribers as

$$N_{\max} = \lim_{k\to\infty}\frac{R_{dr}}{\mu_d}\left[\frac{\tau(N,k)}{\alpha(N) + \upsilon(N,k) + \tau(N,k)}\right]. \tag{20}$$

The *Success Probability* (P_{succ}) is the probability that all required video frames for all users in a service round will be read and placed on their respective buffers on time. The *Saturation Probability* (P_{sat}) is the probability that the disk throughput will be exceeded by the accumulated demand bit rate. The Saturation Probability is defined as

$$P_{sat} = \phi = P[\mu_d > R] = 1 - F_{\mu_d}(R) = 1 - P_{succ}. \tag{21}$$

According to Proposition I, the Saturation Probability can also be defined as

$$P_{sat} = P\,[\alpha(N) + \upsilon(N,k) + \tau(N,k) > kn_v^{(\min)}\,]. \tag{22}$$

We characterize $\tau(N,k)$ in the following manner. Let $Y_i = \{Y_i(t), t > 0, i = 1, \dots, n\}$ be the continuous-time process where $Y_i(t)$ is the amount of video information inserted in user buffer i at time t by the disk subsystem. Let $X_i(t) = Y_i(t) - Y_i(t-1)$ be its strictly positive increment process. We will assume that $Y_i(t)$ has independent increments and that $X_i(t)$ is stationary. Also $\mathbf{E}[X_i(t)] = \overline{S}_{vf,i}$ and $\mathbf{Var}[X_i(t)] = \sigma_i^2$. The process of filling user buffers conforms to the discrete accumulated time series $X_i^{(m)} = \sum_{i=1}^{m} X_i(t) = Y_i(m) - Y_i(0)$. Thus $\tau(N,k) = \sum_{i=1}^{n} X_i^{(m)}/R_{dr}$.

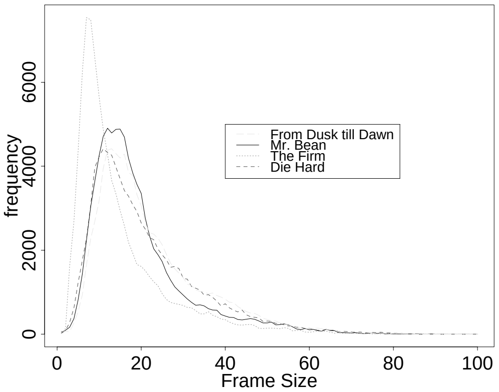


Figure 3. Histogram for the video frame size of four video traces.

7.2. Gamma Service Cycle Mean Model (GCM)

In this model we shall consider only the characteristics of individual strands within a service cycle, then we will consider the video stream as a flow of constant-size video frames. The typical number of frames delivered to each user in a service cycle is close to the service rate (i.e. $kR_{pl}n_{vs}$ frames per second). These considerations are close to modeling the video stream at the GOP level (again see [3, 9 and 10]). Video frame disk-fetching time is now

$$\tau(N,k) = \frac{\sum_{i=1}^{N} \sum_{j=1}^{kR_{pl,j}n_{v,j}} S_{vf,i,j}}{R_{dr}} = \frac{kR_{pl}n_v \sum_{i=1}^{N} \overline{S}_{vf,i}}{R_{dr}} = \frac{kR_{pl}n_v\varphi}{R_{dr}} = \frac{kR_{pl}n_v Na}{R_{dr}} \tag{23}$$

where φ is the minimum number of video frames that must be fetched from disk to avoid buffer starvation and that complies with the global time balance relation, whereas a is the contribution of each user video stream according to the Chernoff bound. We can see from the histograms in Fig. 3 that most video traces are not symmetric and they can be fitted by a Gamma *pdf*. Then, by using the Chernoff bound presented in subsection 6, Eq. 17, we can now obtain explicit expressions for k, N_{max} and b_i. For instance, by using Proposition I, the value of the service-rate proportionality constant is given by

$$k \geq \left\lceil \frac{N\alpha}{n_v^{(\min)} - NR_{pl}\overline{I}_v - \frac{NR_{pl}\overline{n}_v\overline{S}_{vf}}{R_{dr}}\left[1 + \ln\left\{1 + \frac{\sigma_{\max}}{\overline{S}_{vf}}\sqrt{\frac{-2\ln(\phi)}{N}}\right\} - \frac{\ln(\phi)\sigma_{\max}^2}{N\overline{S}_{vf}}\right]} \right\rceil. \tag{24}$$

Similarly, by using Proposition II, we can express the maximum number of subscribers as

$$N_{\max} = \left\lfloor \frac{\overline{n}_v \left(1 + \frac{\ln(\phi)\sigma_{\max}^2 R_{pl}}{R_{dr}\overline{S}_{vf}}\right)}{R_{pl}\overline{I}_v + \frac{R_{pl}\overline{n}_v\overline{S}_{vf}}{N' R_{dr}} \left(1 + \ln\left\{1 + \frac{\sigma_{\max}}{\overline{S}_{vf}}\sqrt{\frac{-2\ln(\phi)}{\overline{N}_{max}}}\right\}\right)} \right\rfloor \tag{25}$$

where $\overline{N}_{max}$ is defined as in Eq. (8).

According to Proposition III, the buffer size for each video strand is just $b_i = kR_{pl}\overline{n}_v\varphi/N$. That is

$$b_i = kR_{pl}\overline{n}_v \left[\overline{S}_{vf}\left(1 + \ln\left\{1 + \frac{\sigma_i}{\overline{S}_{vf}}\sqrt{\frac{-2\ln(\phi)}{N}}\right\}\right) - \frac{\ln(\phi)\sigma_i^2}{N\overline{S}_{vf}}\right]. \tag{26}$$

It is well known that we obtain a Gamma random variable after adding an arbitrary number of Gamma random variables with the same scale parameter. We assume that video streams for different users have the same frame size mean and different variance, so the correct distribution of the sum is not necessarily a Gamma random variable. However, we can obtain an approximate Gamma random variable after considering the same variance for the video streams of different users. The common variance that we consider is the largest variance of all the video streams.

7.3. Strand Gamma Frame Size Distribution Model (GFS)

In this model, we assume that for a single strand, the frames that read on a single service cycle will have different size. This means that for each user, the size of all the video frames read is the sum of $kR_{pl}n_v$ random variables. The global time balance inequality can now simply be written as

$$N\alpha + NkR_{pl}\overline{I}_v + \frac{\varphi}{R_{dr}} \leq kn_v^{(\min)}, \tag{27}$$

and from the Chernoff bound and Proposition I, the service rate proportionality constant can be expressed as

$$k \geq \left\lceil \frac{N\alpha - \frac{\ln(\phi)\sigma_{\max}^2}{R_{dr}\overline{S}_{vf}}}{n_v^{(\min)} - NR_{pl}\overline{I}_v - \frac{R_{pl}\overline{n}_v\overline{S}_{vf}}{R_{dr}}\left[1 + \ln\left\{1 + \frac{\sigma_{\max}}{\overline{S}_{vf}}\sqrt{\frac{-2\ln(\phi)}{NR_{pl}\overline{n}_v\overline{k}}}\right\}\right]} \right\rceil \tag{28}$$

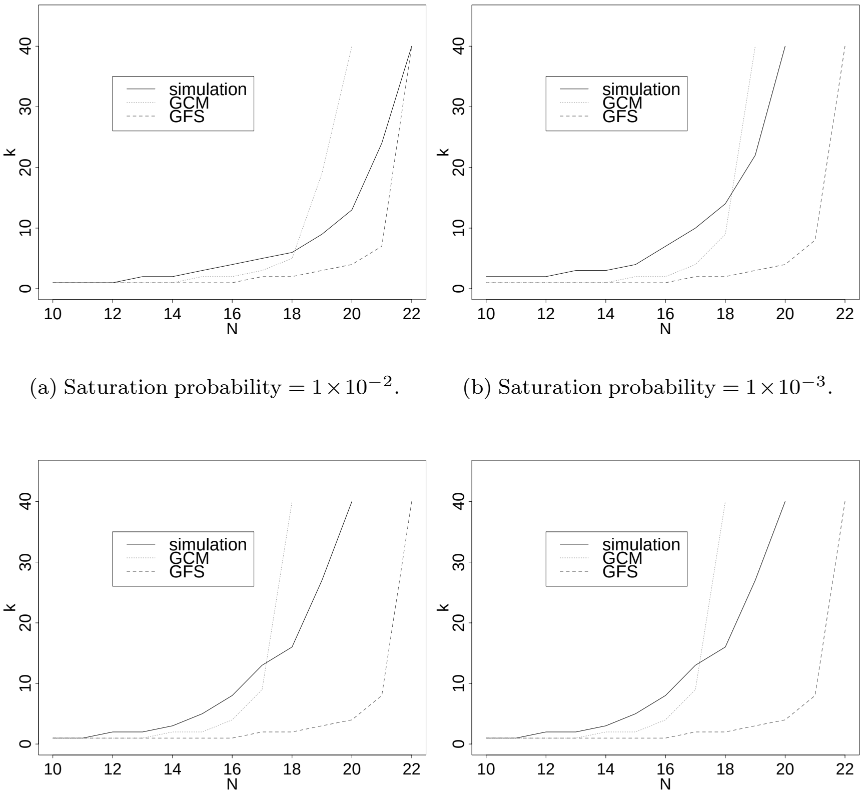
where $\overline{k}$ may be obtained by Eq. (7).

Similarly, by using Proposition II, the maximum number of subscribers is given by

$$N_{\max} = \left\lfloor \frac{\overline{n}_v}{R_{pl}\overline{I}_v + \frac{R_{pl}\overline{n}_v\overline{S}_{vf}}{R_{dr}}} \right\rfloor. \tag{29}$$

And by using Proposition III, the buffer size for each video strand is just φ/N, or equivalently

$$b_i = kR_{pl}\overline{n}_v S_{vf}\left(1 + \ln\left\{1 + \frac{\sigma_i}{S_{vf}}\sqrt{\frac{-2\ln(\phi)}{NkR_{pl}\overline{n}_v}}\right\}\right) - \frac{\ln(\phi)\sigma_i^2}{NS_{vf}}. \tag{30}$$



(a) Saturation probability $= 1 \times 10^{-2}$.

(b) Saturation probability $= 1 \times 10^{-3}$.

(c) Saturation probability $= 1 \times 10^{-5}$.

(d) Saturation probability $= 1 \times 10^{-6}$.

Figure 4. Proportionality constant k as a function of the number of active subscribers.

7.4. Model Discussion

In Fig. 4 we show the computed values of k for each model and data as a function of the number of simultaneous users at different saturation probability requirements. In Fig. 4a we use a fairly high saturation probability of 1×10^{-2}, in Fig. 4b we show results for a slightly smaller saturation probability equal to 1×10^{-3} and in Figures 4c and d we show results for a saturation probabilities of 1×10^{-5} and 1×10^{-6} respectively. The last saturation probability might be typical in practice.

As expected, the service-rate proportionality constant k increases hyperbolically as the number of simultaneous subscribers increases. We can also see that simulation data results tend to have a less steep hyperbolic bend around 12 to 18 users compared to all models. It seems that this set of models is not able to provide accurate estimates for the proportionality constant. However, somehow they do bound the maximum number of simultaneous subscribers. For instance, at $P_{sat} = 1 \times 10^{-2}$ the GCM model indicates $N_{max} = 19$, simulation data indicates 22, whereas the GFS model indicates 22. At $P_{sat} = 1 \times 10^{-3}$ the GCM model indicates $N_{max} = 19$, simulation data indicates 19, and the GFS model indicates again 22 users. At $P_{sat} = 1 \times 10^{-5}$ the GCM model indicates $N_{max} = 18$, simulation data requires $N_{max} = 20$, and GFS model indicates 22. Finally, at $P_{sat} = 1 \times 10^{-6}$, the GCM model shows $N_{max} = 17$, simulation data shows $N_{max} = 20$, and GFS model shows again $N_{max} = 22$.

From these results we can elaborate several conclusions. First, stochastic models can give higher and lower bounds for the service-rate proportionality constant k only at low and high ranges of simultaneous subscribers, failing to do so at middle user ranges. Furthermore, these values are better than those obtained by mean value models. Secondly, compared to simulation results, the Service Cycle model is better than the Frame Size model. Last, the models are able to give upper and lower bounds on maximum number of simultaneous users that can be serviced.

We can also observe that the Service Cycle model is usually too restrictive, while the Frame Size model is too loose. We are now faced with the "Goldie Locks" problem: The Service Cycle Mean model is too hot, while the Strand Frame Size model is too cold. Is there any way to find a model which is just right?

Upon close examination, not detailed here, we find that the main difference between GCM and GFS models is the way in which the models characterize the variation of frame size under user and time aggregation (user aggregation is N, time aggregation is $n = kR_{pl}n_v$). We find that the Service Cycle model says that the aggregated variance depends directly upon the number of variables being read $\sigma^{(n)} = \kappa_1 kR_{pl}n_v\sigma$, while the Strand Frame Size model says, more conventionally, that the variance depends on the aggregation of frames by a square root law $\sigma^{(n)} = \kappa_2(kR_{pl}n_v)^{0.5}\sigma$.

It appears clear that the model we seek for would characterize variability under accumulation in the Fractal or Self-Similarity manner, by considering the following increment variance aggregation structure: $\sigma^{(n)} = n^H\sigma, \quad 0.5 < H < 1$.

8. SELF-SIMILAR MODEL

8.1. Mathematical Description of Self-Similarity

There are several, not equivalent, definitions of self-similarity. The standard one states that a continuous-time process $Y = \{Y(t), t \geq 0\}$ is self-similar (with self-similarity or Hurst parameter H) if it satisfies condition [1]

$$Y(t) \triangleq a^{-H}Y(at),\ t \geq 0,\ a > 0,\ 0.5 < H < 1 \tag{31}$$

where the equality is in the sense of finite-dimensional distributions. It is important to note that taking $t = 1$ and $a = t$, Eq. (31) also means that

$$f_{Y(1)}(x) = |t^H| f_{Y(t)}(|t^H|x) \tag{32}$$

where $f_{Y(t)}(x)$ is the *pdf* of $Y(t)$. Also, it is assumed that

$$\mathbf{E}[Y^2(t)] = \sigma^2 t^{2H} \tag{33}$$

where $\sigma^2 = E[Y^2(1)]$. Eq. (33) gives us a possible alternative to the variance addition problem stated in Section 7.4. For our discrete case this translated to

$$\sigma^{(n)} = n^H\sigma,\ 0.5 < H < 1. \tag{34}$$

8.2. Gamma Fractal Noise

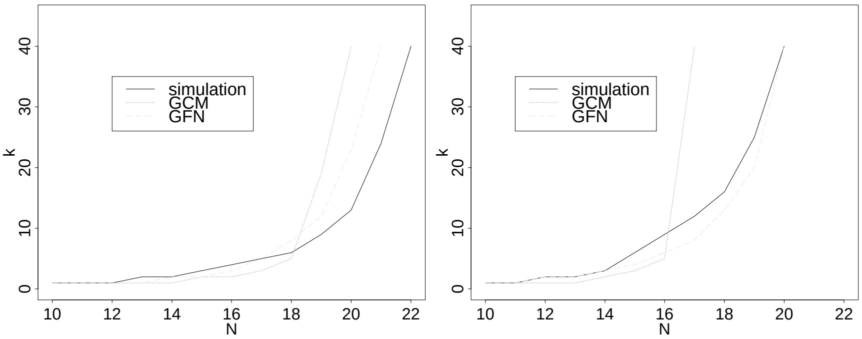
In this model, we define a continuous-time process and later tie it to the discrete-time increment process. Let $Y(t) = \mu t + Z(t)$. The stochastic process $Z(t)$ is self-similar with $\mathbf{E}[Z(t)] = 0$ and $\mathbf{E}[Z^2(t)] = \sigma^2 t^{2H}$. Let $X^{(n)}$ be its accumulation process [7]

$$X^{(n)} = \Sigma_{t=1}^{n} X(t) = Y(n) - Y(n-1) + Y(n-1) \ldots - Y(0) = n\mu + Z(n) \triangleq n\mu + n^H Z(1). \tag{35}$$

From where it follows that the video frame accumulation probability is

$$P(X_1 + \ldots + X_n > \varphi) = P(n\mu + n^H Z(1) > \varphi) = P\left(Z(1) > \frac{\varphi - n\mu}{n^H}\right). \tag{36}$$

The random variable $Z(1)$ is characterized by a three parameter Gamma *pdf*, with parameters: shape α, scale λt^{-H} and location $-\alpha/\lambda t^H$. Its rate function is



(a) Saturation probability $= 1 \times 10^{-2}$ (b) Saturation probability $= 1 \times 10^{-6}$

Figure 5. Proportionality constant k as a function of the number of active subscribers. This figure compares the GFF model and the GCM model versus simulation data results.

$$\ell(a) = (a+\mu)\lambda - \alpha - \alpha \ln\left(1 + \frac{a}{\mu}\right). \tag{37}$$

We can now find an approximation for φ,

$$\varphi \approx n\mu\left[1 + n^{H-1}\left(\ln\left\{1 + \frac{\sigma\sqrt{-2\ln\phi}}{\mu}\right\}\right)\right] - \frac{\ln(\phi)n^H\sigma^2}{\mu}. \tag{38}$$

Again, using the global time balance relation (27) and the Gamma asymptotic we find the proportionality constant of the Gamma Fractal Noise model as follows

$$k \geq \left[\frac{N\alpha - \frac{\ln(\phi)\sigma_{\max}^2\left(NR_{pl}\overline{n}_v\overline{k}\right)^H}{R_{dr}\overline{S}_{vf}}}{n_v^{(min)} - NR_{pl}\overline{I}_v - \frac{NR_{pl}\overline{n}_v\overline{S}_{vf}}{R_{dr}}\left[1 + \left(NR_{pl}\overline{n}_v\overline{k}\right)^{H-1}\ln\left\{1 + \frac{\sigma_{\max}}{\overline{S}_{vf}}\sqrt{-2\ln(\phi)}\right\}\right]}\right]. \tag{39}$$

The buffer size for each video strand is given by

$$b_i = kR_{pl}\overline{n}_v\overline{S}_{vf}\left(1 + \left(NkR_{pl}\overline{n}_v\right)^{H-1}\ln\left\{1 + \frac{\sigma_i}{\overline{S}_{vf}}\sqrt{-2\ln(\phi)}\right\}\right) - \frac{\ln(\phi)\sigma_i^2 N^{H-1}\left(kR_{pl}\overline{n}_v\right)^H}{\overline{S}_{vf}}. \tag{40}$$

This model can be called fractal since the zero-mean three-parameter Gamma *pdf* does agree with the fractal scaling condition stated in Eq. (32).

In Fig. 5 we show several plots of the k parameter versus N for the models GFN and GCM in comparison with the simulation data results for $P_{sat} = 1 \times 10^{-2}$ and $P_{sat} = 1 \times 10^{-6}$. In both figures we clearly see that the GFN model renders an improvement on the prediction for the value of k around the middle ranges of simultaneous subscribers. But more importantly, in Fig. 5b we are able to see that the GFN model follows the simulation data curve in most of the cases, although it still underestimates k within the middle range.

9. CONCLUSIONS

Since the models we have been using rely on the Chernoff bound, which tries to approximate the tail of a probability distribution, it is not surprising that we should obtain better results at very low saturation probability requirements and higher user loads. Also, we can see that incorporating the self-similarity characteristics attributed to video data along with the usage of a skewed probability distribution such as Gamma renders a good improvement on the prediction of digital video server behavior.

The data generated by a digital video server is in fact self-similar, with Hurst parameter maybe close to, but certainly less than 1. This fact implies a long-range dependance characteristic that has important implications on the development of video data models, such as absence of the Markov memoryless property and the inability to expect smooth behavior under periodic averaging.

The considerations above, when taken into account for the development of models of many kind of network data delivery, not only video, will allow a much better design of the data services that must be understood and produced in the near future.

REFERENCES

1. J. Beran, R. Sherman, M. S. Taqqu and W. Willinger. “Long-range Dependence in Variable-Bit-Rate Video Traffic”. IEEE Transactions on Communications, vol. 43, No. 2-4, pp. 1566-1579. April 1995.
2. C. Chase and P. J. Ramadge. “Periodicity and Chaos from Switched Flow Systems: Contrasting Examples of Discretely Controlled Continous Systems”, in IEEE Transactions on Automatic Control, Vol. 38, No. 1, pp. 70-83. January 1993.
3. A. Chodorek and R. D. Chodorek. “Characterization of MPEG-2 Video Traffic Generated by DVD Applications”. 1st European Conference on Universal Multi-Service Networks, pp. 62-70. ECOMN 2000.
4. H. P. Fitzek and M. Reisslein. “MPEG-4 and H.263 Video Traces for Network Performance Evaluation”, Technical University Berlin, Technical Report TKN-00-06, October 2000.
5. M. Garret and W. Willinger. “Analysis, Modeling and Generation of Self-Similar VBR Video Traffic”. ACM SigComm, London, pp. 269-280. September 1994.
6. R. Koenen (Editor). “Overview of the MPEG-4 Standard”. ISO/IEC JTC1/SC29/WG11 N4496, March 2002.
7. K. Park and W. Willinger. “Self-Similar Traffic and Performance Evaluation”. John Wiley and Sons, Inc. New York, USA, 2000.
8. P. V. Rangan and H. M. Vin. “Designing File Systems for Digital Video and Audio”. 13th Symposium on Operating Systems Principles, vol. 25, No. 5, pp. 69-79. October 1991.
9. O. Rose. “Statistical Properties of MPEG Video Traffic and their Impact on Traffic Modeling in ATM Systems”. IEEE Conference on Local Computer Networks, pp. 397-406. October 1995.
10. U. K. Sarkar, S. Ramakrishnan and D. Sarkar. “Segmenting Full-Length VBR Video into Shots for Modeling with markov-Modulated Gamma-Based Framework”. Internet Multimedia Management Systems II, SPIE vol. 4519, pp. 191-202. 2001.
11. H. M. Vin and P. V. Rangan. “Designing a Multiuser HDTV Storage Server”. IEEE Journal on Selected Areas in Communications, Vol. 11, No. 1, pp. 153-164. January 1993.
12. J. C. Yee and P. Varaiya. “An Analytical Model for real-Time Multimedia Disk Scheduling”. 3rd International Workshop on Network and Operating Systems Support for Digital Audio and Video, Springer-Verlang, pp. 276-288. November 1992.