Maximizing Network Utilization for Streaming Video

Spiridon Bakiras and Victor O.K. Li
University of Hong Kong
Department of Electrical & Electronic Engineering
Pokfulam Road
Hong Kong
email: {sbakiras,vli}@eee.hku.hk

Abstract— **Transmission of variable-bit-rate (VBR) video over packet-switched networks is a very challenging problem, and has received much attention in the research community. The burstiness of VBR video makes it very hard to design efficient transmission schemes that will achieve a high level of network utilization. Promising work has been done recently for the transmission of stored video, based on the idea of video prefetching. These protocols use a buffer located at the client's set-top box (STB) to store future frames that are sent when the transmission link is underutilized. Experimental results have shown that video prefetching can achieve a utilization of almost 100% without any need for buffering inside the network. However, no admission control algorithm has been proposed for these protocols to enable their deployment. In this paper, we use the theory of effective bandwidths to develop an admission control algorithm. Each user is allowed to interact with the system, and the admission decision will be based on the users' viewing behavior and the required Quality of Service (QoS).**

I. Introduction

The fast development of the Internet, and the introduction of integrated services and RSVP [1], have made possible the transmission of real-time traffic (e.g. audio and video) that have stringent Quality of Service (QoS) requirements. While internet telephony and video-conferencing are already deployed in the current Internet with limited success, the deployment of applications which require the transmission of high quality video is still lacking. An application like Video-on-Demand (VoD) will allow a customer to select any movie from a video server and view it on his screen while having the ability to perform any type of VCR-like operation [2].

This kind of application is quite attractive, but it can result in very poor network utilization if efficient transmission schemes are not employed. Network utilization is defined as the summation of the individual mean rates of all videos currently transmitted, divided by the service rate (or the link capacity). Video is typically compressed at the Motion Picture Experts Group (MPEG) format. The output of an MPEG encoder is very bursty and the corresponding peak to mean ratio is very high. This property of variable-bit-rate (VBR) video practically prohibits the provision of deterministic QoS guarantees (in this paper we select the packet loss rate at the local switch as the QoS metric). This is because we will have to allocate enough bandwidth to accommodate the peak rate, in order to assure that there will be no loss at the switch. The alternative is to provide statistical QoS guarantees, that is, we guarantee that the loss rate will not exceed a predefined small value (e.g. 10^{-6}). The challenge in providing statistical QoS is to design admission control algorithms that will accurately estimate the required bandwidth.

Most of the proposed schemes in the literature use a buffer at the customer's set-top box (STB) to smooth the video traffic and, therefore, reduce significantly the peak rate and the rate variability [3], [4], [5]. Video smoothing can provide both deterministic and statistical QoS guarantees, with the latter being more desirable as it offers higher network utilization. In [5], for example, a simulation study showed that the optimal smoothing algorithm can support, under deterministic service and for a buffer size of 256 KBytes, 185 smoothed *Star Wars* streams (with an average rate of 0.37 Mbps) on a 155 Mbps link, a utilization of 44%, while for statistical service, 304 streams can be supported (without any loss) for a utilization of 73%. The authors also provided an admission control algorithm, assuming a bufferless switch and based on large deviation techniques, which was shown to be quite accurate.

Recently, the idea of video prefetching has been proposed to provide statistical QoS guarantees for the transmission of VBR video over packet-switched networks. These protocols use the buffer located at the STB to prefetch future frames in periods of low link utilization. The original idea was proposed in [6] with the centralized Join-the-Shortest-Queue (JSQ) prefetching protocol. This scheme can be implemented when there is only one centralized video server, since the prefetching algorithm has to know the frame sizes from all ongoing connections, in advance. In [7] a decentralized version was introduced which allowed prefetching when multiple distributed servers fed a number of customers over a common link. This scheme employs window flow control, with each video server operating independently from the others. The decentralized protocol, however, did not perform well compared to the centralized JSQ prefetching protocol. In [8] we proposed a solution to this problem of distributed video prefetching, by smoothing the MPEG traffic over each group of pictures (GOP) at the video servers, and using a central controller to coordinate the prefetching of future frames between all the video servers. Experimental results have shown that these protocols perform very well compared to video smoothing schemes, and they can achieve a utilization of almost 100%.

It is clear that video prefetching is a very attractive scheme for the transmission of MPEG video traffic. However, there is no admission control algorithm proposed for these protocols that would enable their deployment. In addition, their performance is very sensitive to user interactions, such as temporal jumps, as all the prefetched frames of the user issuing an interaction re-

quest will have to be discarded. In this paper we introduce an admission control algorithm which will decide on the admission of new requests, based on the user's viewing behavior and the required QoS. We first use the traffic model proposed in [9] to model each video trace as a discrete-time Markov modulated deterministic process (D-MMDP). We then use the theory of effective bandwidths [10], [11], [12] to calculate the effective bandwidth for a number of connections, which will depend on the individual traffic parameters, the STB buffer size, the user activity model, and the required QoS. We will show that the effective bandwidth approach is very accurate, and it adapts very well to different system parameters, such as buffer size or level of interactivity. We used 10 MPEG-1 traces [13] to feed our analytical model, and our results indicate that video prefetching is very effective even in an environment where interaction requests are very frequent.

The rest of this paper is organized as follows. In Section II we give a brief overview of distributed video prefetching and we describe its basic principles. In Section III we describe the overall system model which includes the video traffic model that was used to model each video source, and our assumptions regarding the user activity model. In Section IV we develop the analytical model for the admission control algorithm while in Section V we present our numerical results. Section VI concludes our work.

II. Overview of Distributed Video Prefetching

Consider the network architecture shown in Fig. 1. Distributed video servers are connected to the network. These servers may belong to the same or different video service providers. The clients are connected to the network through a switch which may, for example, be an Internet Service Provider (ISP) router. Each client has an STB for video decoding which also includes the buffer used for prefetching. When a client makes a request for a particular video, an admission control module (which is located at the local switch) will decide whether that request can be accepted without violating the targeted QoS of the existing connections. If the request is accepted, a connection will be established between the server and the client through the core network. We assume that a reservation protocol (such as RSVP) is implemented inside the core network to reserve resources for that request. In this work we do not consider the problem of admission control inside the core network, but we only concentrate on the local switch where the users have access to the network. In other words, we consider a VoD-like application where the only type of traffic at the output of the local switch will be stored video. In this case, the admission decision (for the local switch) will be made by the VoD service provider. Inside the core network the different connections between the video servers and the clients will follow different routes and they will be multiplexed with other types of traffic. Therefore, the admission control algorithm will be a more general one which will depend on the core network architecture (i.e. the VoD service provider will not be involved). Multiple clients will have access to the video servers through several different switches. Video prefetching tries to maximize the number of clients served by one such switch, assuming that all the clients connected to that switch will be allowed to share a maximum amount of bandwidth C (e.g. 45 Mbps). It is clear that by doing so, the overall network utilization will be maximized. We assume that the switches in Fig. 1 are bufferless, that is, all packets that exceed the capacity C are dropped. Finally, to facilitate the simulation experiments in Section V, we assume that the video sequences are transmitted in fixed size packets of 1Kb.

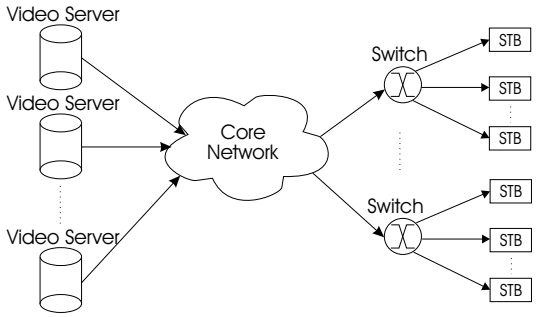


Fig. 1. The network architecture.

The objective of a prefetching protocol is to send additional frames to the different clients when the transmission link is under utilized. The additional frames are buffered at the STBs, and the prefetching protocol ensures that all customers have similar number of prefetched frames. This is easy to do when there is only one video server in the network, since the video sequences are already stored, and the exact frame sizes of each sequence are known a priori (this is the case considered in [6] with the JSQ prefetching protocol). However, when there are multiple distributed servers this scheme can not be implemented. In [8] we proposed a scheme for distributed video prefetching which is based on the main principles of JSQ prefetching. The traffic from each GOP (which is usually 12 or 15 frames) is first smoothed at the video servers before entering the core network. A central controller, located at the local switch, is then used to coordinate the prefetching of future frames from all the video servers. Since the traffic from each connection will be constant over a period of a few frames, the controller can use this information to coordinate the prefetching, in a manner similar to the JSQ prefetching protocol. The coordination is performed with control messages that are sent from the controller to the video servers at every frame period. Our admission control algorithm will assume that this distributed prefetching protocol is implemented in the network of Fig. 1. It should be noted that smoothing is not performed in order to reduce the peak rate of the video sequences, as in the typical video smoothing schemes. In our protocol we smooth each video sequence so as to keep the bit-rate of each connection constant for a small period of time. A detailed description of the protocol is beyond the scope of this paper, and the reader is referred to [8] for further details.

To demonstrate the underlying principle of video prefetching, let us consider the output bit-rate (Fig. 2) from a number of video connections when each connection is sending one frame per frame period (e.g. without prefetching). There will be some periods where the aggregate bit-rate will be less than the link capacity C, and some periods where it will exceed the link capacity. If we want to keep the loss rate small, we need to place a buffer at the local switch to hold the packets that can not be transmitted on time. This method is presented in detail in

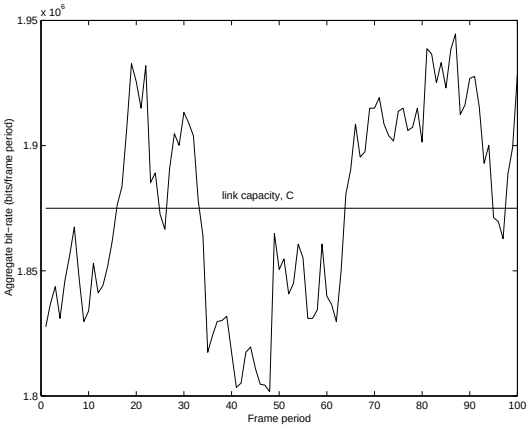


Fig. 2. Output bit-rate without prefetching.

[14] where the authors describe and simulate several proposed admission control algorithms. However, the maximum utilization that could be obtained from any of those algorithms was around 85% (which was very close to the maximum obtainable utilization). Video prefetching, on the other hand, can offer a utilization of almost 100% without any need of buffering at the switches. This is achieved by sending additional frames to the clients when the output bit-rate is less than the link capacity (i.e. prefetching). The additional frames will be used to avoid playback starvation when the bit-rate exceeds the link capacity and some frames can not be transmitted on time. In other words, the frames that would have to be buffered at the switch under a non-prefetching scheme, are sent in advance to the clients so that the aggregate bit-rate at the switch will never exceed the allocated bandwidth. We can, therefore, assume that there is a large virtual buffer of size B placed at the local switch, which is physically distributed among the several STBs (Fig. 3). Since the video traffic is smoothed over each GOP at the video servers, we have to preload the STB buffer with up to k frames prior to the beginning of playback (where k is the GOP size). This is also called the start-up latency. The average size of the virtual buffer will then be

$$B = \sum_{i=1}^{N} (B_i - k \cdot \mu_i) \tag{1}$$

where N is the number of active connections, B_i is the STB buffer size for connection i, and μ_i is the mean frame size for connection i. The buffer occupancy $Q(t)$ of the virtual buffer at time t will be

$$Q(t) = B - \sum_{i=1}^{N} l_i(t) \tag{2}$$

where $l_i(t)$ is the buffer level for connection i at time t. The prefetching algorithm tries to keep the STB buffers as full as possible or, in other words, keep the buffer occupancy $Q(t)$ as low as possible. This is the main difference between video prefetching and typical video smoothing schemes. In a video smoothing scheme the STB buffer is only used to smooth the video traffic (i.e. reduce the peak rate and the rate variability), and during some periods of time it can be almost empty.

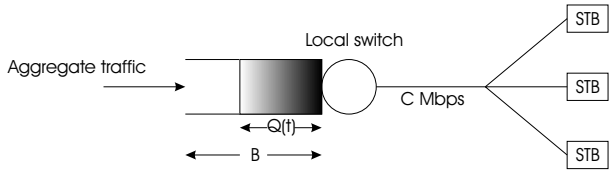


Fig. 3. The virtual buffer at the local switch.

III. System Model

A. The Video Traffic Model

Many different schemes have been reported in the literature for the modeling of VBR video traffic [9], [15], [16]. In [15] the authors used a first order autoregressive (AR) process that matches very well the statistics of the real video traces (i.e. the first and second moments, and the autocorrelation function). The queueing analysis, however, for the calculation of the loss rate at the mutliplexer is very difficult, so the authors proposed an alternative method. They modeled the traffic as a superposition of a number of independent and identical on-off Markov fluids. This model provides a good match to the statistics of the real traces, and in addition, it facilitates the calculation of the loss rate at the multiplexer. In [16] a histogram-based model is introduced that models a video source as a Markov modulated Poisson process (MMPP). An eight-bin histogram of the real data is first constructed, and the resulting values are used as Poisson rates in an eight-state Markov chain. This model matches very well the statistics of the real source, but there are too many state transition probabilities that have to be calculated and used in the queueing model.

For our analysis we selected the model proposed in [9], as it is simple, accurate, and provides a simple formula for the calculation of the effective bandwidth for a number of video sources. In addition, the authors used this model to represent the traffic generated by an MPEG-2 source which is first smoothed over each macro-frame, which is exactly what we consider in this paper. In the following paragraphs we will give a brief description of the traffic model, but the reader should refer to [9] for more details. The first step is to measure the peak rate, bottom rate, mean, variance, and autocorrelation function of the smoothed trace. These parameters are then matched, using some simple formulas, to a discrete-time Markov modulated deterministic process (D-MMDP). More specifically, each video source is modeled as a superposition of M independent and identically distributed two-state discrete-time Markov chains (Fig. 4). In state 1, packets are generated at a constant rate of r_1 packets/frame period, while in state 2 the rate is $r_2 > r_1$. The state transition probabilities are α and β. We can, therefore, model the traffic from each video source i, with only these five parameters $(M_i, r_{1,i}, r_{2,i}, \alpha_i, \beta_i)$.

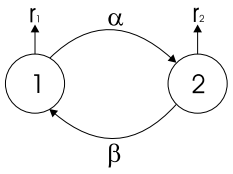


Fig. 4. Two-state discrete-time Markov chain.

In an interactive system, though, the user will be able to perform any kind of VCR-like functions. The only function that will affect the traffic generated from the video server for the particular connection, is the fast playback (i.e. fast forward, fast reverse). There are many ways in which we can display a video sequence at a faster rate. For example, we can send the same sequence, but increase the display rate (e.g. 90 fps, instead of 30 fps). The disadvantage of this method is that the display device might not be able to support such a high display rate. The alternative is to skip a number of frames during the display, and keep the same display rate. In this work, we selected this second method to implement the fast playback operation.

There are three types of frames generated by an MPEG encoder: intraframes (I), predictive frames (P), and bi-directional frames (B) [17]. The I-frames are coded independently of other frames, and for that reason they are used for random access. The P-frames are coded with respect to a previous I/P-frame, so in general they are smaller than I-frames. Finally, the B-frames are coded with respect to a previous and a future I/P-frame. B-frames are usually much smaller than I or P-frames. A number of frames, typically 12 or 15, are grouped together to form a group of pictures (GOP) which has a regular pattern. In our protocol, during a fast playback operation we will skip all the B-frames of the MPEG sequence, and send only the I and P-frames. Since the P-frames require ony the previous I/P-frame for decoding, all the transmitted frames will be decodable at the STB. The video traces that we used in our simulations were captured at 24 fps, and each GOP had the pattern $IBBPBBPBBPBB$. In this case, when a user initiates a fast forward request, the video server will send only 4 frames per GOP and they will be displayed at 24 fps. To the user, it will seem like the display rate is 3 times faster. The I and P-frames of each GOP will first be smoothed at the video server so as to reduce the variability of the resulting traffic. We can then use the same traffic model for the case of fast playback, by considering only the I and P-frames of the video trace. The resulting five parameters $(M_i^{IP}, r_{1,i}^{IP}, r_{2,i}^{IP}, \alpha_i^{IP}, \beta_i^{IP})$ will model the traffic generated by connection i when it is in the fast playback mode.

The above traffic model is quite accurate, and it matches well the first and second moments and the autocorrelation function of the real video sequence. In Table I we have summarized the statistics (the first moment μ, and the standard deviation σ) of a real *Soccer* trace [13] and the sequence generated by this model, for both normal and fast playback. In Fig. 5, the corresponding autocorrelation functions are depicted. The traffic model parameters for the case of normal playback are (9,1,10,0.003,0.012) and for fast playback are (8,2,17,0.0087,0.0313).

B. User Interactivity

In an interactive VoD system each user will be able to perform any type of VCR-like functions, at any time. As we will see in the following section, serving an interaction request requires additional system resources (i.e. more bandwidth) and, therefore, we should take this fact into account when designing the admission control algorithm. There are two ways to perform admission control in an interactive system: we can either perform admission control each time a new request or an interaction request arrives (with interaction requests having priority over the new), or reserve some amount of bandwidth for the interaction requests during the admission control of new requests. We believe that the latter is more suitable for a VoD-like application, since an increased blocking probability of new requests is more desirable than an increased blocking probability of interaction requests.

To properly account for the effect of user interactions, we need a user activity model. Without loss of generality, in this paper, we assume that each user follows the two-state activity model proposed in [18]. In this model, the user starts in normal playback state, and stays there for a period of time which is exponentially distributed with mean $1/\lambda$. He then moves to the interaction state where he will issue an interaction request. He will stay in the interaction state for a period of time which is again exponentially distributed with mean $1/\mu$, and move back again to the normal playback state. This will be repeated until the end of the video sequence. The parameters λ and μ are the interaction arrival and service rates, respectively. In order to perform admission control in such a system, we need to consider each type of user interaction separately. This will be the subject of the following section. Note that we will not consider any *pause/stop* operations in our analysis. If the admission control algorithm admits more connections based on the assumption that some of them will be paused or stopped at any time, the required QoS will be violated if this assumption turns out to be optimistic.

IV. Admission Control

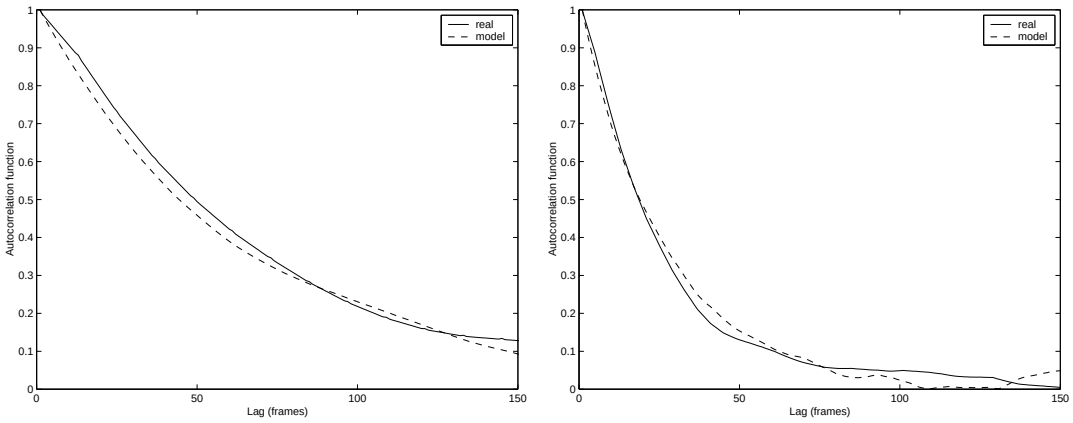
For the admission control algorithm, we chose to use the theory of effective bandwidths [10], [11], [12], and this decision was based on the following two facts.

1. *Simplicity* – The effective bandwidth is defined as the minimum service rate required to satisfy the desired loss rate for a number of connections feeding a common buffer. The most important property of effective bandwidth is that it is additive, which means that we can simply add the individual effective bandwidths in order to find the effective bandwidth for the aggregate traffic. This property leads to very simple admission control decisions which is very important in real-time applications such as streaming video.
2. *Accuracy* – This argument is certainly not true in some cases. The effective bandwidth approach is based on an asymptotic approximation of the buffer loss probability. In particular, the loss probability at a buffer of size B is approximated by
$$P(x > B) \approx e^{-\delta B} \tag{3}$$
where $\delta = -\log p/B$, and p is the targeted loss probability. This approximation is based on the assumption that the buffer size in infinite, which is not true for commercial switches. For normal buffer sizes and very bursty sources, such as video, the effective bandwidth approach is very conservative and thus not appropriate for admission control [19], [11], [14]. In video prefetching, however, the virtual buffer size is very large (e.g. for 100 clients with 1MB buffer each, the buffer size B is 100 MB), and this approximation is quite accurate. In the next section we

TABLE I

STATISTICS OF THE REAL AND MODEL GENERATED *Soccer* TRACE.

	Normal rate		Fast rate	
	μ (packets)	σ (packets)	μ (packets)	σ (packets)
Real	25	11	42	18
Model	25	11	41	17



(a) Normal rate. (b) Fast rate.

Fig. 5. Autocorrelation function of the real and model generated *Soccer* trace.

will show, through simulation experiments, that the effective bandwidth approach is indeed very accurate when utilized in a prefetching scheme.

Kesidis *et al.* [12] have provided the solution to the effective bandwidth problem, for different types of Markov sources. For the D-MMDP model described in the previous section, the effective bandwidth c_i (in packets/frame period) for a connection i, is

$$c_i = \frac{M_i}{\delta}\ln\Big[\frac{1}{2}\big(a(\delta) + \sqrt{a^2(\delta) + 4b(\delta)}\big)\Big] \qquad (4)$$

where

$$a(\delta) = (1-\alpha_i)e^{\delta r_{1,i}} + (1-\beta_i)e^{\delta r_{2,i}}$$

and

$$b(\delta) = e^{\delta(r_{1,i}+r_{2,i})}(\alpha_i + \beta_i - 1)$$

Let us call $\hat{C}_n$ the effective bandwidth (packets/frame period) of the aggregate traffic, and C (packets/frame period) the total available bandwidth. Assume that there are currently $N-1$ connections in progress, and there is a request for a new connection. The admission control algorithm will calculate $\hat{C}_n$ from the following equation

$$\hat{C}_n = \sum_{i=1}^{N} c_i \qquad (5)$$

If $\hat{C}_n \leq C$ the new connection will be admitted; otherwise, it will be rejected. Note that the individual effective bandwidths c_i will have to be calculated during each connection request, since the buffer size B will be different. The above admission control algorithm is applicable only in a system that does not support user interactions. In the following two subsections we will show how the different user interactions affect the number of admissible connections.

A. Fast Forward/Reverse

As we mentioned in Section III, a *fast rate* request issued by a client will increase significantly the amount of traffic sent for that particular client, since only the I and P-frames will be sent. Therefore, we should reserve some amount of bandwidth during admission control, so that subsequent *fast rate* requests will not violate the QoS requirements of any client. The extreme case would be to reserve enough bandwidth to accommodate the scenario where all the clients are on *fast rate* mode simultaneously. Obviously this is a very conservative approach, and the resulting network utilization would be very poor. The alternative is to reserve less bandwidth, and then reject some interaction requests according to some rule. More specifically, we will reserve an amount of bandwidth which will be able to accommodate m concurrent *fast rate* operations, while maintaining a low blocking probability for interaction requests. Let us call λ_f the arrival rate of *fast rate* requests, and μ_f the corresponding service rate. Since both parameters are exponentially distributed according to our user activity model, we can model this system as an $M/M/m/m/N$ queueing system, that is, an m-server loss system with finite customer population N (Fig. 6). We are interested in finding a number m, such that the stationary probability p_m is less than a small number, where m is the number of customers in the system (i.e. the number of users served in *fast rate* mode simultaneously). The desired value of p_m will be set by

the VoD service provider, according to their policy. In this work we will assume that $p_m \leq 0.01$. The formula for p_m is easy to derive and it is given by [20]

$$p_m = \frac{\binom{N}{m}\left(\frac{\lambda_f}{\mu_f}\right)^m}{\sum_{i=0}^{m}\binom{N}{i}\left(\frac{\lambda_f}{\mu_f}\right)^i} \quad (6)$$

Then, m will be the smallest integer that satisfies the inequality $p_m \leq 0.01$.

The admission control algorithm will be based on the assumption that there will always be m users in *fast rate* mode. The system will also keep track of the number of connections that are in *fast rate* mode at all times, and it will block an interaction request if this number is equal to m. When a customer initiates or terminates a *fast rate* request, all the prefetched frames in the STB buffer will have to be discarded, since they are no longer useful. As the average service time of such requests will normally be very small (around 10-20 seconds), it is not efficient to fill up the STB buffer again with future frames. We will, therefore, assume that only one frame per frame period is sent to a connection that operates in *fast rate* mode. As a result, the buffer size B initially given in (1) will now be equal to

$$B = \frac{(N-m)}{N}\sum_{i=1}^{N}(B_i - k\cdot\mu_i) \quad (7)$$

where N is the total number of ongoing connections, including the new request.

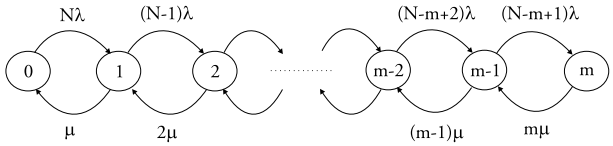


Fig. 6. The state-transition-rate diagram for the $M/M/m/m/N$ queueing system.

We will next calculate the effective bandwidth for the case where all connections are served in fast rate. Let us call c_i^{ip} the individual effective bandwidths, and $\hat{C}_f$ the effective bandwidth of the aggregate traffic. With the traffic parameters $(M_i^{IP}, r_{1,i}^{IP}, r_{2,i}^{IP}, \alpha_i^{IP}, \beta_i^{IP})$ we can calculate c_i^{IP} for every connection i, using (4). Then, $\hat{C}_f$ will be

$$\hat{C}_f = \sum_{i=1}^{N} c_i^{IP} \quad (8)$$

Therefore, the average effective bandwidth in *fast rate* mode per connection is C_f/N. Assuming that there will always be m connections in *fast rate* mode, the effective bandwidth $\hat{C}$ for a system that supports only *fast rate* requests will be

$$\hat{C} = \frac{(N-m)}{N}\hat{C}_n + \frac{m}{N}\hat{C}_f \quad (9)$$

where $\hat{C}_n$ is given in (5). It is clear that this type of user interaction will decrease the network utilization, as $\hat{C}_f > \hat{C}_n$.

The idea of reserving some amount of bandwidth for accommodating *fast rate* requests was first proposed in [21] where the authors considered two different approaches. In the first one, a *fast rate* request is delayed until there are available resources, and the admission control ensures that the probability that this delay exceeds a certain value is small. This approach can be implemented in our scheme as well, if we allow the interaction requests to be queued up instead of blocking them. In the second approach, there is no delay associated with an interaction request, but when there are not enough system resources to serve all the interaction requests, the picture quality of the users in *fast rate* mode is degraded. However, for a VoD system to be competitive with the existing video rental services, it should offer a better service to the user. In our scheme, the picture quality is never degraded, and the system response to user interactions is instantaneous. The blocking probability p_m can also be set to a very small value, practically eliminating blocked requests. In addition, the work in [21] considered peak rate bandwidth allocation for each connection, leading to low network utilization. In our scheme, we employ video prefetching and statistical multiplexing which can increase significantly the network utilization. This is basically the main contribution of our work. To our knowledge, there is no admission control scheme in the literature that can be directly applied in a real VoD system. In this work we propose a complete solution which considers all the aspects of a real system: transmission protocol (i.e. prefetching), user interactivity, and statistical multiplexing for VBR video.

B. Jump Forward/Backward

In typical video transmission schemes, *jump* operations do not affect the network utilization. In video prefetching, however, frequent *jump* requests will degrade the utilization of the system. Suppose a user initiates a *jump* request during normal playback. Since this operation will take the user to a point in the video sequence which will be quite far (where far means anything more than 10-15 seconds) from the current point, all the prefetched frames will have to be discarded, as they will not be displayed. It is clear that when those requests are frequent, the buffer size B will decrease and, thus, the effective bandwidth for the same connections will increase.

Modeling the effect of *jump* requests on the buffer size B is not easy. However, if we keep the buffer size constant, it is easy to model the event of buffer loss using a continuous time MMDP (C-MMDP) model, similar to the discrete version that was presented in Section III. The difference is that the transition probabilities α and β (Fig. 4) will now represent transition rates. We will model the event of buffer loss by assuming that a *jump* request will trigger the arrival of additional traffic at the switch, which is equal to the average buffer level of a connection. Let us call λ_j the arrival rate of *jump* requests. The corresponding service rate is infinite, since after issuing the interaction request the user will return instantaneously to normal playback. We will need to calculate the four parameters for the C-MMDP model, namely r_1^j, r_2^j, α^j, and β^j.

- Clearly, $r_1^j = 0$, since there will be no additional traffic when no *jump* request is issued.
- The rate at which *jump* requests arrive will be $\alpha^j = N\lambda_j$.
- Similarly, $\beta^j = 1 - N\lambda_j$.

- To calculate r_2^j we need to find the average buffer level for one connection. Based on (3), the average virtual buffer occupancy will be $1/\delta$. Therefore, the average buffer level for one connection will be $r_2^j = (B - \frac{1}{\delta})/N$.

Kesidis *et al.* [12] have also provided the solution for the effective bandwidth for the C-MMDP model, which is given by

$$C_j = \frac{1}{2\delta}\Big(-a(\delta) + \sqrt{a^2(\delta) - 4b(\delta)}\Big) \tag{10}$$

where

$$a(\delta) = \alpha^j + \beta^j - \delta(r_2^j - r_1^j)$$

and

$$b(\delta) = \delta^2 r_1^j r_2^j - \delta(\alpha^j r_2^j + \beta^j r_1^j)$$

Note that C_j is given here in packets/sec, so the corresponding value in packets/frame period would be $\hat{C}_j = C_j/f$, where f is the frame rate. Then, the effective bandwidth $\hat{C}$ for a system that supports only *jump* requests will be

$$\hat{C} = \hat{C}_n + \hat{C}_j \tag{11}$$

C. The Complete Admission Control Algorithm

After analyzing the effect of different types of user interactions on the effective bandwidth, we are ready to present the complete admission control algorithm. The inputs of the algorithm will be the traffic parameters of all ongoing connections (including the new request), the arrival and service rates for the different user interactions, the STB buffer size B_i, and the link capacity C. The admission control will be performed as follows.

1. Calculate m. This value will depend on the number of active users N, including the new request.
2. Calculate the buffer size B from (7).
3. Calculate $\hat{C}_n$, $\hat{C}_f$, and $\hat{C}_j$.
4. Calculate the effective bandwidth $\hat{C}$ from the following equation

$$\hat{C} = \frac{(N-m)}{N}\hat{C}_n + \frac{m}{N}\hat{C}_f + \hat{C}_j \tag{12}$$

5. If $\hat{C} \leq C$ admit the new request, else reject it.

V. Numerical Results

In order to investigate the accuracy of our admission control algorithm, we used 10 real MPEG-1 traces that are available in the public domain [13]. They covered a wide variety of contents, including movies, news, talk shows, sports, music, and cartoons. All the traces were captured at $f = 24$ fps and the GOP pattern was $IBBPBBPBBPBB$. Even though these traces have some problems (e.g. some frames are dropped), they are very bursty and they exhibit self-similarity, which makes them suitable for our simulations. The total number of frames for each trace was 40,000, and by using four copies of each trace we created 10 sequences, each of approximately 111 minutes. The link capacity C was assumed to be 45 Mbps. In Table II we have summarized some characteristics of the different GOP smoothed MPEG traces.

The experiments were performed as follows. We created a random sequence of requests for different movies, and using our admission control algorithm, a certain number of those requests were accepted. For each of the accepted requests, we chose a random starting point in the movie (the beginning of a GOP), and we started by transmitting one frame from each connection, with all buffers being initially empty. From the next time slot the prefetching algorithm in [8] was used to coordinate the transmissions until the end of the experiment. When a connection displayed the last MPEG frame of the movie, the same movie started again from the beginning, with an empty buffer. Each connection started in normal playback, and stayed there for a period of time which was exponentially distributed with mean $1/\lambda$. Then it moved to the interaction state where it issued an interaction request. The time spent in *fast rate* mode was exponentially distributed with mean $1/\mu_f = 30$ seconds. The requested offset during a *jump* request was uniformly distributed between 1 and 1000 seconds. We simulated 2×10^8 frame periods for different buffer sizes, and interaction arrival rates λ. Finally, we counted the packet loss rate after an initial period of 50,000 frames (to allow the buffers to fill up). The required loss rate was set to 10^{-6}.

TABLE III
ACTUAL AND PREDICTED NETWORK UTILIZATION WITHOUT USER INTERACTIONS.

	Utilization (%)	
	128 KB	1 MB
Admission control	89.7	98.9
Actual	93.2	98.9

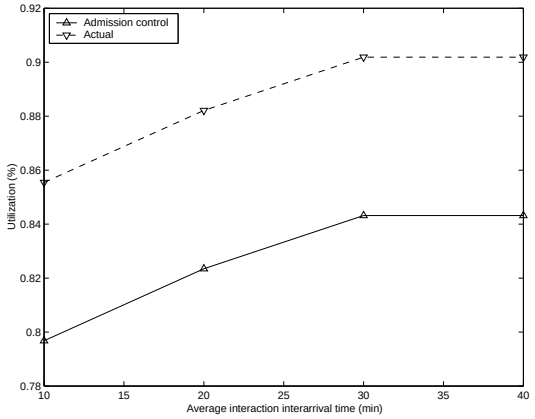
In the beginning, we simulated a system without user interactions. The results are given in Table III for two different buffer sizes, 128 KB and 1 MB. The actual utilization was obtained by admitting additional *Asterix* connections to the requests that were already accepted by the admission control algorithm, and running the experiment again until the QoS requirement was violated. We can see that for a moderate buffer size of 1 MB, the admission control algorithm is very accurate, and it predicted exactly the number of connections that could be admitted. In addition, the network was able to work at a utilization of almost 100%. For a small buffer size of 128 KB, there is an overestimation of around 3%, but the admission control resulted in a utilization of 90%. Comparing those numbers with the utilization achieved by typical video smoothing schemes (i.e. around 73% for the optimal smoothing algorithm [5]), the effectiveness of video prefetching is clearly illustrated.

In Fig. 7 we have plotted the network utilization as a function of the average interaction interarrival time (i.e. $1/\lambda$), for different STB buffer sizes, and different types of user interactions. For a buffer size of 128 KB and *fast rate* requests (Fig. 7(a)), there is an overestimation of about 6%, which is caused by: (1) the small buffer size, and (2) the assumption that there are always m connections in *fast rate* mode (as described in the previous section). For *jump* requests, though, the admission control is more accurate (Fig. 7(b)), and the required bandwidth is only overestimated by approximately 2%. Another interesting result which is depicted in Fig. 7(b), is that for small buffer size the system performance in the presence of only *jump* requests, is

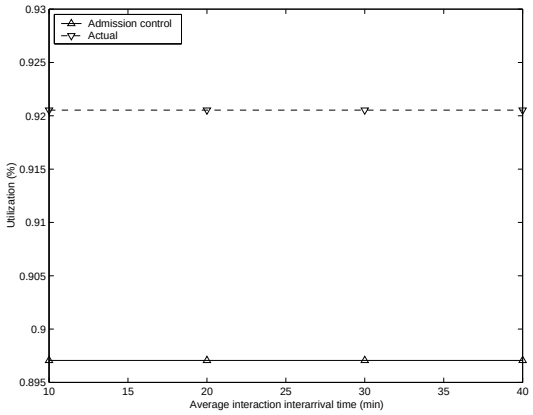
TABLE II

CHARACTERISTICS OF THE SMOOTHED MPEG-1 VIDEO SEQUENCES.

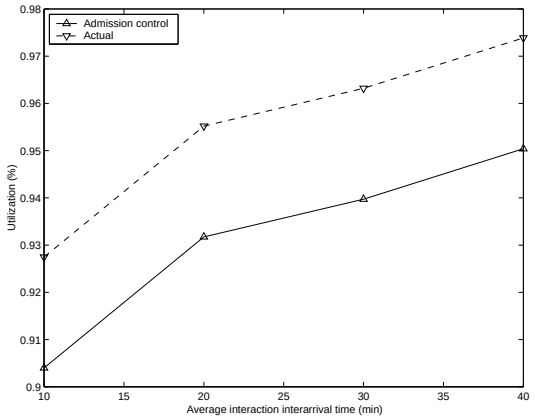
Sequence	Normal rate		Fast rate	
	μ (packets)	σ (packets)	μ (packets)	σ (packets)
Asterix	22	10	38	14
ATP Tennis	22	8	39	13
Mr Bean	18	9	32	13
James Bond	24	9	51	20
Jurassic Park	13	5	25	8
Mtv	20	14	34	19
News	15	7	29	12
Race	31	11	48	17
Soccer	25	12	42	18
Talk show	15	5	27	7



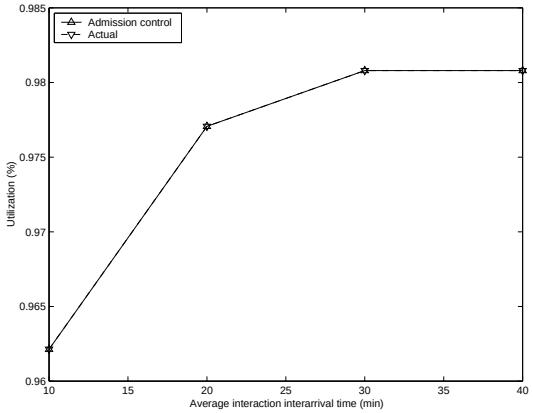
(a) *Fast rate* requests, $B_i = 128$ KB.



(b) *Jump* requests, $B_i = 128$ KB.



(c) *Fast rate* requests, $B_i = 1$ MB.



(d) *Jump* requests, $B_i = 1$ MB.

Fig. 7. Network utilization as a function of $1/\lambda$.

independent of the average interaction arrival rate. This can be explained by the fact that each buffer is filled up very fast after it is emptied during an interaction request (due to its small size). Even for very frequent interaction requests, the client's

buffer will be filled up completely before the next request arrives, making the effect of *jump* requests practically invisible to the prefetching protocol.

For a buffer size of 1 MB, the admission control algorithm is very accurate. In Fig. 7(c), we can see that for *fast rate* requests the algorithm overestimates the required bandwidth by 2%, which is again caused by the assumption of always having m users in *fast rate* mode simultaneously. We should note that the measured blocking probability for *fast rate* requests in all our experiments ranged between 0.003 and 0.008, which is well below the targeted blocking probability of 0.01. For *jump* requests, the algorithm predicts exactly the number of admissible connections. Moreover, for an average interaction interarrival time of more than 20 minutes, the network utilization is very close to the maximum obtainable utilization (which is shown in Table III). We can, therefore, argue that by placing a buffer which is sufficiently large at the STBs, we can practically eliminate the effects of user interactions, and keep the network utilization well above 90%.

VI. Conclusions

We have presented a call admission control algorithm for streaming video where each user is allowed to interact at any time during normal playback. This algorithm utilizes the recently proposed idea of video prefetching, for the transmission of the video sequences. The theory of effective bandwidths was used to design the admission control algorithm for a system that supports full user interactivity. We have shown that the proposed algorithm is very accurate and it adapts very well to different system parameters, such as the buffer size or level of interactivity. The numerical results indicate that video prefetching is very effective and, combined with our proposed admission control algorithm, it can achieve a network utilization of nearly 100%. In addition, it performs very well even in an environment where user interactions are very frequent.

Acknowledgments

This research is supported in part by the University Grant Committee, Hong Kong, Area of Excellence in Information Technology, Grant No. AOE 98/99.EG01, and by the State Scholarships Foundation of Greece.

References

[1] X. Xiao and L. M. Ni, "Internet QoS: a big picture," *IEEE Network*, pp. 8–18, March/April 1999.

[2] V. O. K. Li and W. J. Liao, "Distributed multimedia systems," *Proceedings of the IEEE*, vol. 85, no. 7, pp. 1063–1108, July 1997.

[3] J. M. McManus and K. W. Ross, "Video-on-demand over ATM: constant-rate transmission and transport," *IEEE Journal on Selected Areas in Communications*, vol. 14, no. 6, pp. 1087–1098, August 1996.

[4] J. D. Salehi, Z. L. Zhang, J. F. Kurose, and D. Towsley, "Supporting stored video: Reducing rate variability and end-to-end resource requirements through optimal smoothing," in *Proceedings ACM SIGMETRICS*, May 1996, pp. 222–231.

[5] Z. L. Zhang, J. F. Kurose, J. D. Salehi, and D. Towsley, "Smoothing, statistical multiplexing and call admission control for stored video," *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 6, pp. 1148–1166, August 1997.

[6] M. Reisslein and K. W. Ross, "Join-the-shortest-queue prefetching protocol for VBR video on demand," in *Proceedings IEEE International Conference on Network Protocols (ICNP)*, October 1997, pp. 63–72.

[7] M. Reisslein, K. W. Ross, and V. Verillotte, "A decentralized prefetching protocol for VBR video on demand," in *Proceedings 3rd European Conference on Multimedia Applications, Services and Techniques (ECMAST)*, May 1997, pp. 388–401.

[8] S. Bakiras and V. O. K. Li, "Smoothing and prefetching video from distributed servers," in *Proceedings IEEE International Conference on Network Protocols (ICNP)*, October 1999, pp. 311–318.

[9] J. Ni, T. Yang, and D. H. K. Tsang, "Source modelling, queueing analysis, and bandwidth allocation for VBR MPEG-2 video traffic in ATM networks," *IEE Proceedings on Communications*, vol. 143, no. 4, pp. 197–205, August 1996.

[10] A. I. Elwalid and D. Mitra, "Effective bandwidth of general Markovian traffic sources and admission control of high speed networks," *IEEE/ACM Transactions on Networking*, vol. 1, no. 3, pp. 329–343, June 1993.

[11] R. Guerin, H. Ahmadi, and M. Naghshineh, "Equivalent capacity and its application to bandwidth allocation in high-speed networks," *IEEE Journal on Selected Areas in Communications*, vol. 9, no. 7, pp. 968–981, September 1991.

[12] G. Kesidis, J. Walrand, and C. Chang, "Effective bandwidths for multiclass Markov fluids and other ATM sources," *IEEE/ACM Transactions on Networking*, vol. 1, no. 4, pp. 424–428, August 1993.

[13] O. Rose, "Statistical properties of MPEG video traffic and their impact on traffic modeling in ATM systems," in *Proceedings 20th Annual Conference on Local Computer Networks*, 1995, pp. 397–406.

[14] E. W. Knightly and N. B. Shroff, "Admission control for statistical QoS: theory and practice," *IEEE Network*, pp. 20–29, March/April 1999.

[15] B. Maglaris, P. Anastassiou, P. Sen, G. Karlsson, and J. D. Robbins, "Performance models of statistical multiplexing in packet video communications," *IEEE Transactions on Communications*, vol. 36, no. 7, pp. 834–843, July 1988.

[16] P. Skelly, M. Schwartz, and S. Dixit, "A histogram-based model for video traffic behavior in an ATM multiplexer," *IEEE/ACM Transactions on Networking*, vol. 1, no. 4, pp. 446–459, August 1993.

[17] B. G. Haskell, A. Puri, and A. N. Netravali, *Digital video: an introduction to MPEG-2*, Chapman & Hall, 1997.

[18] V. O. K. Li, W. Liao, X. Qiu, and E. W. M. Wong, "Performance models of interactive video-on-demand systems," *IEEE Journal on Selected Areas in Communications*, vol. 14, no. 6, pp. 1099–1109, August 1996.

[19] G. L. Choudhury, D. M. Lucantoni, and W. Whitt, "Squeezing the most out of ATM," *IEEE Transactions on Communications*, vol. 44, no. 2, pp. 203–217, February 1996.

[20] L. Kleinrock, *Queueing systems, Vol. I: theory*, John Wiley & Sons, 1975.

[21] J. K. Dey-Sircar, J. D. Salehi, J. F. Kurose, and D. Towsley, "Providing VCR capabilities in large-scale video servers," in *ACM Multimedia*, October 1994, pp. 25–32.