Euromicro 96, September 2-5, 1996, in Prague

Statistical Admission Control in Video Servers with Variable Bit Rate Streams and Constant Time Length Retrieval

Ernst Biersack, Frédéric Thiesse
Institut Eurécom
2229 Route des Crêtes, 06904 Sophia Antipolis, FRANCE
{erbi,thiesse}@eurecom.fr

Abstract

We consider the admission control problem in video servers for the retrieval of media data from disk storage. We assume that the I/O bandwidth of the server disk is limited. Given a certain I/O bandwidth, admission control decides whether or not a new client can be accepted without affecting the quality of service promised to the already admitted clients. Assuming variable bit rate (VBR) video streams, we consider an admission control policy with both, deterministic and statistical service guarantees for constant time length retrieval (CTL) and evaluate its performance in terms of the number of clients admitted.

1. Introduction

Video servers store digitized, compressed continuous media information on high-capacity secondary or tertiary storage [8]. The secondary storage devices allow for random access and provide short seek times compared to tertiary storage. Video server design differs significantly from that of traditional data storage servers due to the large size of the objects stored and the real-time requirements for their retrieval. The critical resources in a video server are disk bandwidth, storage volume, and main memory. Given a fixed amount of these resources, a video server can only deliver a limited number of video streams simultaneously. Before admitting a new client, a video server must use an admission control algorithm to check if there are enough resources for serving the additional client.

We recently introduced a generalization of the constant time length data retrieval called **GCTL**. For GCTL we introduce both, a worst-case deterministic and a statistical description of the amount of video data that need to be retrieved during each round that will then be used to define a deterministic and a statistical admission control criterion.

2. Deterministic Retrieval Schemes in Video Servers

A video server must meet the requirements that stem from the continuous nature of audio and video and must guarantee the delivery of continuous media data in a timely fashion. We assume that video information is encoded as a **variable bit rate stream** (**VBR**) of *constant* quality. VBR requires sophisticated resource reservation mechanisms for the server and network to achieve a good utilization of the resources while maintaining a constant quality playback.

2.1 Deterministic Constraint Function for VBR Video

To provide deterministic quality of service (QoS) for VBR video, the admission control must employ *worst-case assumptions* about the data rate of the VBR video when computing the number of streams to be admitted. To offer deterministic service, we use a traffic model that is deterministic. The so-called *empirical envelope* presented in [6] provides a deterministic traffic constraint function for a given video trace. If $A_i[t, t+\tau]$ denotes the amount of video data consumed by a stream s_i in the interval $[t, t+\tau]$, an upper bound on A_i can be given by the **empirical envelope function** $\varepsilon_i(\tau)$ that is defined as:

$$\varepsilon_i(\tau) = \max_t A_i[t, t+\tau], \forall t \in [0, T_{total} - \tau]$$

2.2 Round-Based Retrieval Schemes

In the simplest case, continuous playback can be ensured by buffering the entire stream prior to initiating the playback [5]. Such a scheme, however, requires very large buffer space and causes a very large start-up latency. Consequently, the problem of efficiently servicing a single stream becomes one of preventing buffer starvation while at the same time minimizing the buffer requirement and the start-up latency. In the most general sense, the buffer requirement in a video server at time t can be stated as the difference between the cumulative arrival function $a(t)$ of the video data read from secondary storage, and the cumulative consumption function $c(t)$ denoting the video data sent to clients.[1] The difference is referred to as **backlog function** [7]

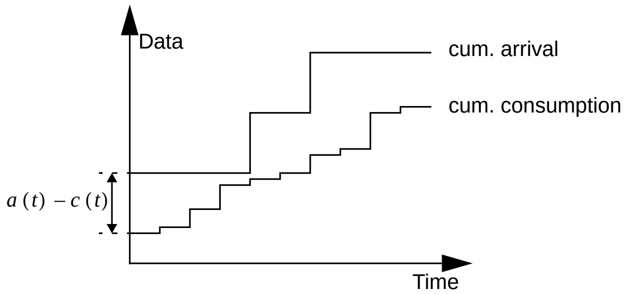

Figure 1. Backlog function: cumulative arrival – cumulative consumption

We say that **buffer starvation** at time t occurs if $a(t) - c(t) < 0$. If b_{total} denotes the total amount of available buffer in a video server, then $a(t) - c(t) > b_{total}$ will cause **buffer overflow**.

In order to avoid buffer starvation or buffer overflow, almost all approaches to multi-stream continuous media retrieval have the following characteristics [5]:

1. Processing stream requests in cyclic rounds.
2. Arrival keeps up with consumption.

A video server that operates in rounds generally avoids starvation by *reading ahead* an amount of data that lasts in terms of playback duration through the next round (see figure 2). Data retrieval techniques determine the way data is read from the disk during a service round.

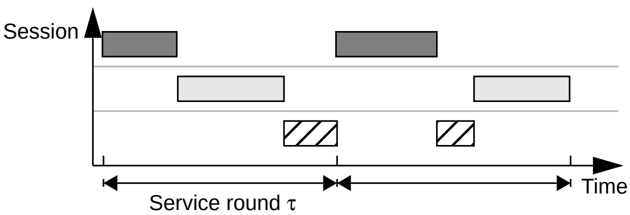

Figure 2. Sequence of service rounds

The admission control scheme considered in this paper allows VCR functions such as fast forward, reverse, or pause) under the condition that the data rate required to support these functions is *not higher* than the data rate for normal playback.

[1] The functions $a(t)$ and $c(t)$ can be alternatively stated in terms of frames or in terms of media data. If stated in terms of frames, the deterministic buffer requirement in terms of data is then determined by the relation between a number of frames and their respective maximum data size.

Using VBR as data model for a video, one can map video data onto **data blocks (segments)** stored on the disk in two ways: **constant time length** (**CTL**) and **constant data length** (**CDL**) [1]:

- CTL retrieval is characterized as having *variable length* data blocks with *constant playback* duration τ for stream s_i (see figure 3). During any service round of duration τ, $\tau \cdot r_i$ frames are retrieved from secondary storage, where r_i denotes the constant frame rate of stream s_i. Since successive frames of a VBR video differ in size, CTL retrieval results in a periodic but volume-variant retrieval.

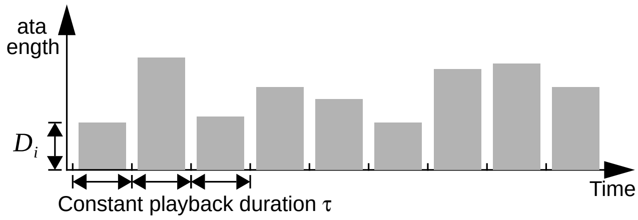

Figure 3. Constant time length retrieval

- If we simply inverse the two properties concerning periodicity and data block size, we get constant data length (CDL) retrieval that combines *non-periodic* retrieval with *constant* amounts of data from the disk. At first sight, CDL might seem incompatible with *round-based* disk retrieval, but if we introduce the restriction that the distances between retrieval operations must be multiples of τ, we get sequences of what we call **active** and **idle** rounds. During an active round, a *constant* size data block is read from the disk. Since the data must always be sufficient to supply the client during the following round even in the worst case, the (fixed) size of the data block retrieved is $\varepsilon_i(\tau)$. During an idle round, no data at all is retrieved.

Each of the two retrieval strategies, CTL and CDL, has advantages and disadvantages. Given the real-time requirements of continuous media and the periodic nature of video playback, CTL retrieval appears to be the more natural approach. It can easily be implemented because media quanta are always handled in terms of frames. A sequence of frames that must be sent to a client can therefore easily be mapped to disk I/O requests. In the following, we will limit our considerations to CTL.

3. Generalized CTL

3.1 Introduction

Up to now, all papers on periodic retrieval schemes have assumed that

- the **disk service round**, during which data for each stream are read exactly once from disk, and
- the **CTL round,** for which we compute the worst case data consumption given by $\varepsilon_i(\tau)$

have the *same length*.

We will distinguish the two and propose to make the CTL round a *multiple* of a disk service round (The same generalization can be applied to CDL, as shown in [4]):

- The disk scheduling and retrieval still proceeds in rounds of length τ.
- However, we use a set $T = \{\tau_1, \ldots, \tau_n\}$ of CTL rounds with $\tau_i \geq \tau$. To avoid starvation, we require that the amount of data retrieved for stream s_i from the disk during each interval τ_i must last for a period of τ_i. The CTL round duration τ_i is an integer multiple m_i of the disk service round duration τ (see figure 4).When $A_i[t, t+\tau_i]$ is the amount of data consumed by stream s_i during a period τ_i, we require that the amount of data retrieved during any of the $m_i = \tau_i/\tau$ disk service rounds is the same, namely $(A_i[t, t+\tau_i])/m_i$. Note that during any disk service round (of duration $\tau \leq \tau_i$) within the CTL round, *fewer* frames may be read from the disk than are consumed by the client.

In the following, we will refer to the CTL retrieval where CTL rounds and disk rounds have the same length ($\tau = \tau_i$) as **traditional** CTL. When CTL rounds and disk rounds have different length ($\tau \neq \tau_i$), the scheme is referred to as **Generalized CTL (GCTL)** retrieval. The traditional CTL can be regarded as a special case of GCTL with $m_i = 1$.

We have shown that the separation of disk service round and CTL round helps smooth the VBR traffic and allows to significantly reduce the buffer demand and the start-up latency while admitting the same number of clients.

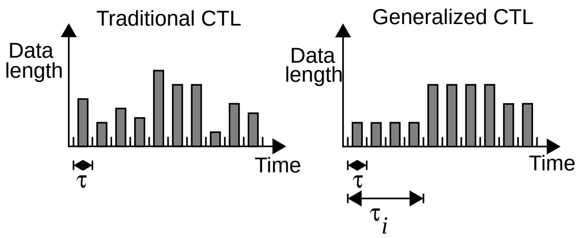

Figure 4. Traditional and generalized CTL

Table 1 gives a summary of the performance of CDL with various combinations of τ and τ_i for the 'MTV' trace and illustrates the advantages of generalized CTL over traditional CTL. If we take, for instance, the case where 9 clients can be admitted we see that GCTL reduces the buffer requirement by more than 700 KByte and the start-up latency by more than 90%.

	τ [sec]	τ_i [sec]	Start-up latency [sec]	Buffer requirement per stream [byte]	Number of admitted streams
traditional CTL	1	1	2.0	673,470	6
	2	2	4.0	1,315,291	8
	3	3	6.0	1,933,156	8
	4	4	8.0	2,335,040	8
	5	5	10.0	2,712,945	8
	6	6	12.0	3,085,291	9
generalized CTL	1	1	2.0	673,470	6
	1	2	2.7	901,445	7
	1	3	3.2	1,069,069	7
	1	4	3.4	1,142,250	7
	1	6	3.8	1,259,659	8
	2	2	4.0	1,315,291	8
	2	4	5.3	1,700,241	8
	2	6	5.8	1,880,080	8
	2	8	6.2	1,957,973	9

Table 1. Deterministic service using GCTL

3.2 Deterministic Admission Control

The number of streams admitted is limited by length of a disk service round, the available buffer space and the disk bandwidth. The admission control criterion for GCTL is given by

$$\sum_{i=1}^{n} \left\lceil \frac{\varepsilon_i(\tau_i)}{m_i} \right\rceil \cdot r_{disk} + \sum_{i=1}^{n} \left\lceil \left\lceil \frac{\varepsilon_i(\tau_i)}{m_i} \right\rceil \cdot c_{cyl} \right\rceil \cdot t_{track} +$$

$$+ n \cdot (t_{track} + t_{rot}) + t_{seek} \leq \tau$$

In this formula, r_{disk} denotes the disk bandwidth, c_{cyl} equals the capacity of a single cylinder and t_{track}, t_{rot} and t_{seek} denote the track-to-track seek time, the rotational latency and the maximum seek time for a complete scan over the entire disk.

3.3 Statistical Description of VBR Video

A deterministic service can be assured at any time during the playback by using the worst case traffic characterization, given by $\varepsilon_i(\tau_i)$, for the admission control criterion. Deterministic service results in an inefficient use of the server's resources, such as disk bandwidth and buffer space. As can be seen from the plots in figure 5 and 6, during the majority of all possible GCTL rounds, a client

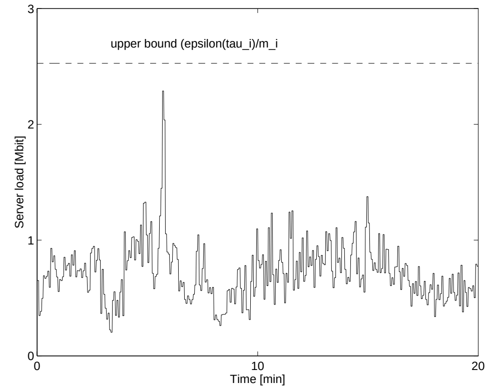

Figure 5. Actual data rate for GCTL ('MTV' trace)

consumes much less data than the envelope function $\varepsilon_i(\tau_i)$ suggests. During most playbacks, the largest data block does not even reach the upper bound because the upper bound is derived from all *possible* disk service rounds.

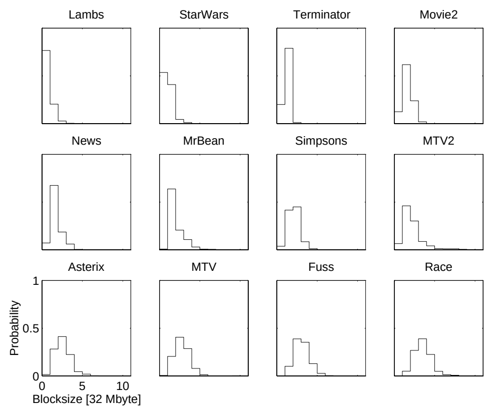

Figure 6. Data rate histograms for GCTL

To provide statistical service guarantees, a video server is required to obtain a precise *statistical traffic characterization*. The traffic characterization is used to compute the probability of a server overload proposed to a client as part of the set of QoS parameters. The probability of an overload is the probability of a situation where the video server cannot deliver all requested media data under the given real-time constraints.

Chang et al. [2] have proposed a technique for the computation of the probability of a server overload using a characteristic video bit rate histogram. Beyond these exact methods, they further propose two methods for estimating the distribution function of the requested data: first, by applying the **central limit theorem**, and second, based on a variation of **Cramer's theorem** known as Bahadur-Rao theorem. Vin et al. [9] have also proposed a method of estimating the distribution of disk bandwidth requirements by a normal distribution using the central limit theorem.

Since the normal distribution can only be regarded as a good approximation for a large number of clients, we use a precise estimate by histogram convolution for a medium size video server designed for several tens of clients. As Dengler [3] showed, such a histogram can be easily computed if traces of the videos are available, for instance when the video is stored on the server.

Given a **characteristic data rate histogram** h_i for each video the characteristic server load histogram after the admission of a new stream s_{n+1} can be computed by convolving the server load histogram H_n with the data rate histogram of the newly requested video h_{n+1}. Convolution can be thought of as multiplying two polynomials whose coefficients are the elements of H_n and h_{n+1} since this is the same algebraic operation. Suppose l_H and l_h denote the length, i.e. the number of bins, of the vectors H_n and h_{n+1}, respectively. Then the result H_{n+1} is a vector (histogram) of length $l_H + l_h + 1$ whose k-th element is given by

$$H_{n+1}(k) = \sum_{i=1}^{k-1} H_n(i) \cdot h_{n+1}(k-i)$$

3.4 Statistical Admission Control

If the random variable $\tilde{D}$ denotes the server load during a CTL round, the probability of a server overload $p(\tilde{D} > D_{n+1}^{max})$ is obtained by computing the tail of H_{n+1} beyond the maximum data rate D_{n+1}^{max}:

$$p(\tilde{D} > D_{n+1}^{max}) = \sum_{k = D_{n+1}^{max}+1}^{\infty} H_{n+1}(k)$$

D_{n+1}^{max} is the maximum amount of data that can be retrieved during a single disk service round τ to serve the $n+1$ streams. D_{n+1}^{max} can be defined as the product of the time available for data transfer and the disk's transfer rate[2]:

$$D_{n+1}^{max} = (\tau - t_{seek} - (n+1) \cdot (2 \cdot t_{track} + t_{rot})) \cdot r_{disk}$$

Figure 7 shows the overload probability as a function of the number of simultaneous streams for $\tau = 1s$ and $\tau_i = 4s$. In order to investigate the effect of a higher disk bandwidth, which changes the ratio of data transfer to seek operations, we also did the same simulations for a transfer rate that is four times higher than the one before.

When a streams terminates, the analogue computation

[2]We assume that there occurs only one track-to-track seek operation during the retrieval of one stream's data block and one more when the server switches from one stream to the next. In our simulations this simplification has no influence on the results since the capacity of a disk cylinder is always larger than the size of a data block for all considered τ and τ_i, i.e. $c_{cyl} > \varepsilon_i(\tau_i) / m_i$.

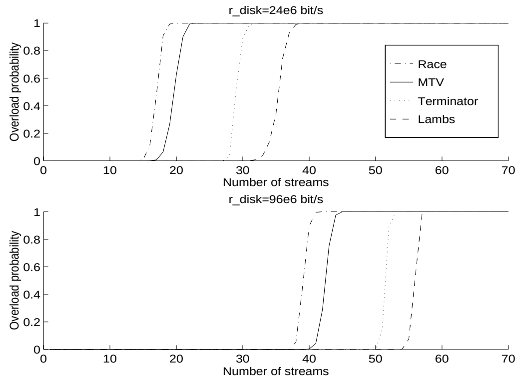

Figure 7. Overload probabilities for GCTL

called de-convolution must of course be performed in order to get the new load histogram.

We calculated overload probabilities of different videos for the homogeneous case, i.e. we considered clients which request all the same video. All simulations were then repeated a second time for a disk transfer rate that was four times as high as before. In order to reduce the size of the resulting histograms we keep the information about the data blocks that are retrieved during a disk service round at a granularity of 4 Kbyte, e.g. a data block from the interval [0; 4095] is rounded to 2048 Bytes, a block of size [4096; 8191] is rounded to 6144 Bytes, and so on. Our computations with finer granularities showed that this aggregation suffices in most cases to get sufficiently precise results.

Video	Determin. admission		Statistical admission		$\frac{\varepsilon_i(\tau_i)}{m_i}$ [Mbit]	mean bit rate [Mbit/s]
	N_{th}	N_{det}	$N_{10^{-4}}$	$G_{10^{-4}}$		
Lambs	107	19	35	0.84	0.85	0.21
StarWars	94	17	31	0.82	0.97	0.27
Terminator	53	25	32	0.28	0.57	0.31
Movie2	88	16	26	0.63	1.04	0.41
News	48	10	23	1.30	1.89	0.44
MrBean	52	10	21	1.1	1.75	0.50
Simpsons	69	13	22	0.69	1.33	0.53
MTV2	37	8	17	1.13	2.47	0.57
Asterix	56	11	18	0.64	1.61	0.64
MTV	36	7	16	1.29	2.53	0.70
Fuss	55	11	16	0.45	1.66	0.78
Race	46	9	15	0.67	1.98	0.88

Table 2. Statistical admission control for GCTL (Disk transfer rate: $r_{disk} = 24 \cdot 10^6$ **bit/s)**

Our results are summarized in table 2 ($r_{disk} = 24 \cdot 10^6$ bit/s) and 3 ($r_{disk} = 96 \cdot 10^6$ bit/s).

$N_{10^{-x}}$ denotes the number of streams that can be admitted without exceeding an overload probability of 10^{-x}. For example, a probability of 10^{-4} means that on the average every 10^4 th disk service round an overload occurs. For $\tau = 1$ s this is equivalent to an overload after 2.8 hours. The values are compared to the following results of deterministic admission control:

- $N_{th} = \lfloor \tau \cdot m_i \cdot r_{disk} / \varepsilon_i(\tau_i) \rfloor$ denotes the theoretical number of clients that could be admitted if no seek operations occured.
- N_{det} denotes the maximum number of admitted streams for deterministic GCTL retrieval as presented in the previous section.

Video	Deterministic admission		Statistical admission		$\frac{\varepsilon_i(\tau_i)}{m_i}$ [Mbit]	mean bit rate [Mbit/s]
	N_{th}	N_{det}	$N_{10^{-4}}$	$G_{10^{-4}}$		
Lambs	428	41	56	0.37	0.85	0.21
StarWars	376	39	54	0.38	0.97	0.27
Terminator	212	48	54	0.13	0.57	0.31
Movie2	352	38	49	0.29	1.04	0.41
News	192	28	48	0.71	1.89	0.44
MrBean	208	29	45	0.55	1.75	0.50
Simpsons	276	34	46	0.35	1.33	0.53
MTV2	148	23	42	0.83	2.47	0.57
Asterix	224	30	42	0.40	1.61	0.64
MTV	144	23	41	0.78	2.53	0.70
Fuss	220	30	40	0.33	1.66	0.78
Race	184	27	38	0.41	1.98	0.88

Table 3. Statistical admission control for GCTL (Disk transfer rate: $r_{disk} = 96 \cdot 10^6$ **bit/s)**

We see that the increase in the number of clients from deterministic to statistical service differs a lot for the considered videos. Therefore, we further investigated the relations between the characteristics of a video and the gain $G_{10^{-x}}$, which is defined as the improvement of the number of streams in the statistical case when compared to the deterministic case:

$$G_{10^{-x}} = \frac{N_{10^{-x}} - N_{det}}{N_{det}}$$

During our further analysis, we found that there seems to be a correlation between the gain and the ratio of the size of the largest retrieved data block $\varepsilon_i(\tau_i)/m_i$ and the mean bit rate mbr_i (see figure 8). This phenomenon can be explained as follows: An overload occurs, whenever there is not enough available I/O bandwidth to retrieve the amount of data that is needed to supply the client. If the peak bit rate is much higher than the mean bit rate, only a few parts of the video can take advantage of the reserved disk I/O bandwidth and thus an additional stream can cause overloads when such parts are retrieved by several streams at the same time, which is quite improbable. For the same reason, when the peak bit rate does not differ very much from the mean bit rate, an additional stream causes overloads during many service rounds, because there is not enough wasted bandwidth that the additional stream can use.

Second, the number of admitted streams in the deterministic case, which depends on the peak bit rate, should be low to reduce the total number of seek operations. Seek times reduce the free I/O bandwidth, i.e. videos that allow for a high number of streams in the deterministic case yield low gains in the statistical case because they require many seek operations.

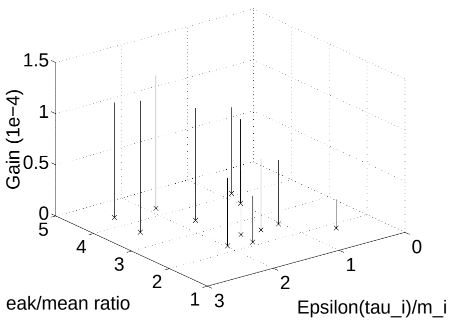

Figure 8. Gain as function of $\varepsilon_i(\tau_i)/m_i$ and the mean bit rate

An upper bound n_{max} to the number of admitted streams is given by the seek times that are independent from the movie and the disk transfer rate. For the values we used in our simulations, we get $n_{max} = 69$ for $\tau = 1$ s. Thus, the statistical multiplexing gain becomes smaller for increasing disk rates because a higher number of admissions leads to a higher seek overhead, i.e. the ratio of the total data transfer time to the total seek time becomes smaller.

Furthermore, it is important to keep in mind that we only calculated the probability of an overload event itself without taking into account how many data get lost. At first sight it seems as if probabilities which are weighted by the size of the lost data blocks might deliver better statements about a certain quality of service but since we consider compressed videos, this is not the case, e.g. for an MPEG-compressed movie the loss of one single I-frame can have much worse consequences on the subsequent playback than the loss of several B-frames.

4. Conclusion

We presented a generalization of constant time length data retrieval (GCTL) and defined a deterministic and a statistical admission control criterion for it.

We saw that statistical admission control can admit up to twice as many clients as deterministic admission control for very low overload probabilities. Furthermore, we investigated the correlations between the gain in the statistical case and the characteristics of the considered videos.

5. References

[1] E. Chang and A. Zakhor. Admission control and data placement for vbr video servers. In *Submitted to Proceedings of the 1st International Conference on Image Processing*, Austin, Texas, November 1994.

[2] E. Chang and A. Zakhor. Variable bit rate mpeg video storage on parallel disk arrays. In *Proceedings of the 1994 1st International Workshop on Community Networking*, pages 127–137, San Francisco, CA, July 1994.

[3] J. Dengler. Admission control strategies and design issues for video servers. Master's thesis, University of Mannheim, Mannheim, November 1995.

[4] J. Dengler, C. Bernhardt, and E. W. Biersack. Deterministic admission control strategies in video servers with variable bit rate streams. In B. Butscher, E. Moeller, and H. Pusch, editors, *Interactive Distributed Multimedia Systems and Services, European Workshop IDMS'96, Berlin, Germany*, volume 1045 of *LNCS*, pages 245–264. Springer Verlag, Heidelberg, Germany, Mar. 1996.

[5] D. J. Gemmell, H. M. Vin, D. D. Kandlur, P. V. Rangan, and L. A. Rowe. Multimedia storage servers: A tutorial and survey. *IEEE Computer*, 28(5):40–49, May 1995.

[6] E. Knightly, D. Wrege, and J. Liebeherr. Fundamental limits and tradeoffs of providing deterministic guarantees to vbr video traffic. Technical Report TR-94-067, EECS Department, U.C. Berkeley, UCB, 1994.

[7] E. W. Knightly, R. F. Mines, and H. Zhang. Deterministic characterization and network utilizations for several distributed real-time applications. In *Proc. of IEEE WORDS'94*, Dana Point, CA, Oct. 1994.

[8] L. A. Rowe and R. R. Larson. A video-on-demand system. The University of California at Berkeley and International Computer Science Institute, 1993.

[9] H. M. Vin, P. Goyal, A. Goyal, and A. Goyal. A statistical admission control algorithm for multimedia servers. In *Proceedings of the 2nd ACM International Conference on Multimedia*, San Francisco, CA, October 1994.

Appendix

The following table depicts the characteristics of the video traces that we used for our simulations:

Name	Compression ratio	Bit-rate		Peak / Mean ratio
		Mean bit-rate [Mbps]	Peak bit-rate [Mbps]	
Mr. Bean	150:1	0.50	6.55	13.0
Asterix	119:1	0.64	4.22	6.6
Fuss	98:1	0.78	5.36	6.9
Lambs	363:1	0.21	3.84	18.3
Movie 2	186:1	0.41	4.94	12.0
MTV	108:1	0.70	6.56	9.3
MTV 2	134:1	0.57	7.19	12.8
News	173:1	0.44	5.43	12.4
Race	86:1	0.88	5.79	6.6
Simpsons	143:1	0.53	6.88	12.9
Star Wars	285:1	0.27	3.57	13.4
Terminator	243:1	0.31	2.28	7.2

Table 4. Characteristics of the twelve video traces