In B. Butscher, E. Moeller, and H. Pusch, editors, *Interactive Distributed Multimedia Systems and Services, Eurepean Workshop IDMS'96, Berlin, Germany*, volume 1045 of *LNCS*, pages 245–264. Springer Verlag, Heidelberg, Germany, March 1996

Deterministic Admission Control Strategies in Video Servers with Variable Bit Rate Streams

Johannes Dengler[1], Christoph Bernhardt, Ernst Biersack
Institut Eurécom[2]

Abstract: Video servers are a key component in multimedia systems. Due to the real-time requirements and high resource demand of digital media, a video server must restrict the number of simultaneously serviced media streams. We consider the admission control problem in video servers for the retrieval of media data from secondary storage. Admission control decides whether or not a new request can be accepted without affecting the service given to the already admitted streams. Traditional retrieval methods, such as cyclic retrieval of variable size data segments or retrieval at the stream's mean bit rate, either cannot profit from smoothing media traffic over larger intervals or suffer from excessive buffer demand and latency. We introduce, for the first time, retrieval techniques for variable bit rate data that are non-buffer-conserving in nature and cover all traditional methods as special cases. For all the schemes, we carry out a comparative performance analysis and show how they allow to trade-off buffer requirement, disk I/O efficiency, and latency. All the schemes considered support the full set of VCR operations such as fast forward, pause, or fast reverse.

Keywords: Video Server, Admission Control, VBR

1 Introduction

Emerging high-speed networks allow the introduction of interactive distributed multimedia services, such as *video-on-demand* (VOD), *news-on-demand*, *tele-shopping*, and *distance learning*. A typical scenario consists of a **video server** connected to *clients* via a *communication network*. The video server stores digitized, compressed continuous media information on high-capacity secondary or tertiary storage [Rowe93]. The secondary storage devices are random accessible and provide short seek times compared to tertiary storage. An on-demand copy of the requested material is provided via the network to the client upon request.

While high bandwidth may become ubiquitous at modest cost, video servers are regarded as the critical components of future interactive multimedia systems. Their design differs significantly from that of traditional data storage and retrieval servers because real-time storage and retrieval techniques are required [Rang92],[Stei91]. Additionally, video servers must provide efficient mechanisms for storing, retrieving, indexing, and manipulating data in large quantities at high speeds [Rowe93]. A video

[1] Now with: McKinsey & Company, Inc., Taunusanlage 21, 60325 Frankfurt, Germany, email: JoDengler@aol.com.
[2] 2229 Route des Crêtes, 06904 Sophia-Antipolis — France, Phone: +33 93002611, FAX: +33 93002627, email: {bernhard,erbi}@eurecom.fr

server can only deliver a limited number of video streams simultaneously. Before admitting a new client, a video server must consequently use an admission control algorithm that is not needed in traditional servers for file storage.

We identify and formalize traditional schemes for the retrieval of *variable bit rate* video data from magnetic disks. Traditional methods, such as the cyclic retrieval of variable size data segments or the retrieval at the stream's mean bit rate, either cannot profit from smoothing media traffic over larger intervals, or they suffer from excessive buffer demand and latency. We derive novel techniques that are generally non-buffer-conserving in nature and cover traditional methods as special cases. When offering a deterministic service, the novel schemes can drastically decrease the buffer requirement and server latency in an interactive multimedia service by sacrificing some disk I/O efficacy, and vice versa. We state the admission control criteria for the investigated schemes and derive the buffer, latency, and efficiency trade-off. The trade-offs are first derived in theory and illustrated through by simulations using traces of variable bit rate videos.

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. Video information can be encoded to produce: (i) a **constant bit rate stream (CBR)** of variable quality, or (ii) a **variable bit rate stream (VBR)** of constant quality. CBR video has the advantage of being easy to deal with from both the network and the server perspective. VBR video corresponds more closely to the actual data format of compressed video, and is thus preferable from the application point of view. However, VBR requires more sophisticated resource reservation mechanisms in server and network to guarantee a good utilization of existing resources and a constant quality playback. We focus here on a VBR based video server. Our models are in fact generic enough to be able to accommodate CBR as a special case.

In the simplest case, continuous playback can be ensured by buffering the entire stream prior to initiating the playback [Gemm95]. Such a scheme, however, requires very large buffer space and may also yield a very large 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 information read from secondary storage, and the cumulative consumption function $c(t)$ denoting the video information sent to clients.[3] The difference is often referred to as *backlog* function [Knig94]:

[3] 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.

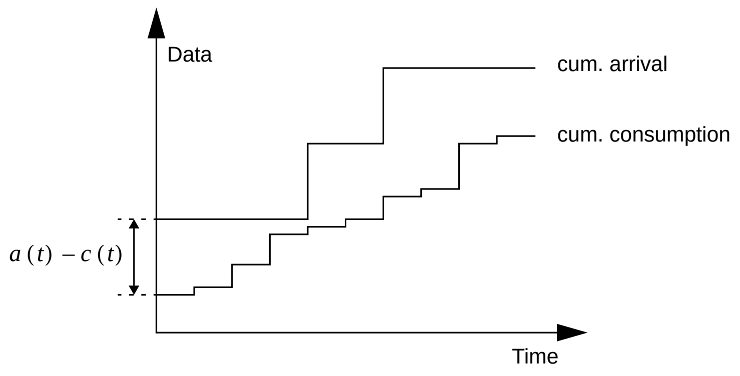


Fig. 1. Backlog function: cumulative arrival – cumulative consumption

We define a situation of **buffer starvation**[4] at time t as $a(t) - c(t) < 0$. If b_{total} denotes the total amount of available buffer in a video server, then $a(t) - c(t) > b_{total}$ will lead to **buffer overflow**.

In order to avoid buffer starvation or buffer overflow, almost all approaches to multi-stream continuous media retrieval that address these constraints have the following characteristics [Gemm95]:

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

A video server that operates in rounds does generally avoid starvation by *reading ahead* an amount of data that lasts in terms of playback duration through the next round. If on a round-by-round basis the arrival of data never falls behind the consumption, the scheduling algorithm is referred to as **buffer-conserving.** Algorithms which proceed in rounds but are **non-buffer-conserving** are also conceivable but more complex [Gemm95]. Such an algorithm would allow the arrival to fall behind the consumption in one round, and then make it up later.

Offering VCR functions, such as fast forward and fast reverse, can have great impact on the bandwidth and buffer requirements. The VCR functions fast forward and fast reverse can either be implemented by playing back media at a rate higher than normal, or by continuing playback at the normal rate while skipping some data. Whereas the former approach yields significant increase in the data rate requirement [Dey94], data skipping may be complicated by the presence of inter-data dependencies introduced by compression schemes that reduce the temporal redundancies in a video stream, or may result in output of poor quality due to higher compression. The admission control schemes considered in this paper allow VCR functions under the condition that the data rate required to support these functions is *not higher* than the data rate for normal playback.

[4] The terms *buffer starvation* and *buffer underflow* are used interchangeably throughout this paper.

Choosing VBR as data model for a video server requires one to choose one of two models for storing data on the disks of a video server. There are two ways of mapping video data onto **data blocks (segments)** stored on the disk. Chang and Zakhor [Cha94a] have identified two techniques, referred to as **constant data length (CDL)** and **constant time length (CTL)**:

- CTL data placement is characterized as having *variable length data blocks* with *constant real-time playback duration* τ for stream s_i. During any one service round of duration τ, $\tau \cdot r_i$ frames[5] are retrieved from secondary storage, where r_i denotes the constant frame rate of stream s_i.

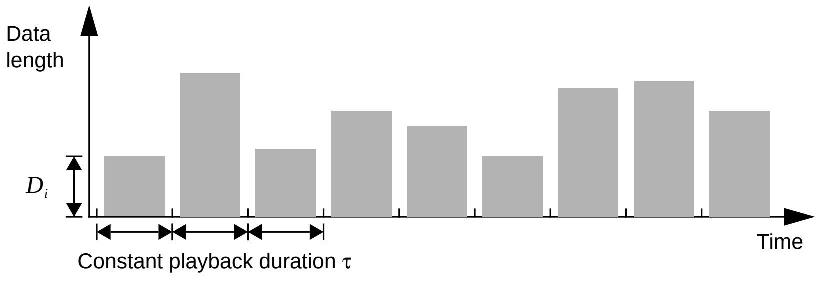


Fig. 2. Constant time length data placement

- For CDL, the size of all the data blocks is the same and the playback time of one data block can vary from data block to data block. Notice however, that the size of the data blocks is constant for all retrievals of data for stream s_i (all service rounds) but need not be the same for all streams. They may vary from stream to stream depending on the characteristics of the video.

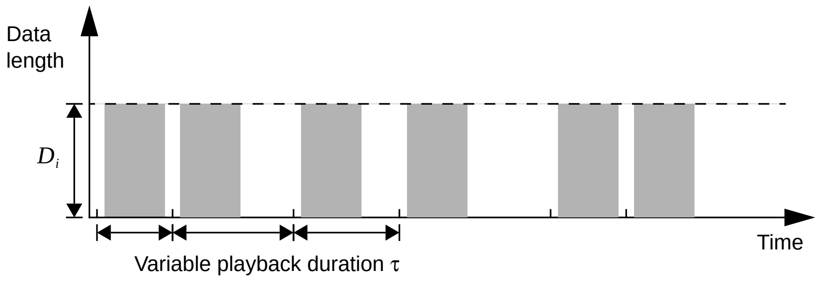


Fig. 3. Constant data length data placement

[5] We assume $\tau \cdot r_i$ to take integer values such that during each service round exactly $\tau \cdot r_i$ frames are read. If the assumption is dropped, the number of integral frames is given by $\lfloor \tau \cdot r_i \rfloor$.

Each of the two data placement strategies, CTL and CDL, has several advantages and disadvantages. Given the real-time requirements of continuous media and the periodic nature of video playback, CTL data placement appears to be the more natural strategy. It can easily be implemented because media quanta are always handled in terms of frames for which the time-scale is essential. A sequence of frames that must be sent to a client can therefore easily be mapped to disk I/O requests. Moreover, CTL data placement allows for disk scheduling algorithms that proceed in rounds of constant length in time, which can significantly reduce the seek overhead.

When using CDL data placement, the amount of data retrieved at a time does not vary. It can correspond in size to a disk block, allowing for efficient disk layout. Since CDL experiences the variation of variable bit rate video in the delays between consecutive retrievals it at first glance appears incompatible with round-based disk scheduling. An admission control criterion for CDL data placement must regard the playback time contained within the retrieved media data with respect to the block's disk service time.

These two data placement strategies should not be regarded as being exclusive. Chang and Zakhor [Chan94] as well as del Rosario and Fox [dR95] propose the combination of *both* CDL and CTL data placement strategies *at once* such that for every stream, constant size blocks are retrieved within equidistant time intervals. This method is referred to as **pseudo-constant bit rate (PCBR)** because it makes a VBR video stream appear as constant bit stream. While PCBR offers smooth disk and network traffic, it entails some major disadvantages: first, PCBR retrieval of course requires substantial buffer space because all burstiness of variable bit rate video is smoothed by *prefetching*. A large buffer may be needed to conceal the burstiness of the video. Second and related to the prefetching of media data, a large **start-up latency** can be introduced. Furthermore, if the client jumps to a different portion of the video, the prefetched data will become obsolete and the video server will be required to prefetch again.

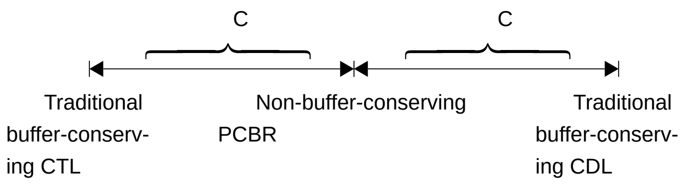

C
C
Traditional buffer-conserv-ing CTL
Non-buffer-conserving PCBR
Traditional buffer-conserv-ing CDL

Fig. 4. Classification of storage and retrieval schemes

A comparative analysis of these various data placement strategies, which has been missing so far, will be carried out for CTL and PCBR in this paper. Throughout Section 4 we derive general non-buffer-conserving data retrieval schemes that cover CTL and PCBR as special cases, as indicated by figure 4. Specifically, we are able to show that between CTL and PCBR, respectively, there exists a continuum of non-buffer-conserving retrieval strategies that allows for a trade-off between disk I/O efficacy, latency, and buffer demand. A detailed analysis of CDL and its comparison with CTL and PCBR is omitted in this paper due space limitations.

3 Constraint Functions for Variable Bit Rate Video

3.1 Maximum Media Traffic

To provide deterministic QOS, the admission control criterion must employ *worst-case assumptions*, thus setting strict bounds to the number of streams to be admitted. The traffic model employed is necessarily deterministic in nature. We use frame size traces of MPEG-1 encoded video sequences[6] obtained at the University of Würzburg [Rose95] for the admission control schemes. A video trace consists of exact frame sizes expressed in bits or bytes, and the frames' arrival times [dR95]. If frames are assumed to arrive at a monotone rate, the video playback rate can be used interchangeably to express the arrival times. The so-called **empirical envelope** presented in [Knig95] provides the most accurate traffic constraint function for a given video trace. Consider the empirical envelope $\varepsilon(\tau)$ as a worst-case traffic model as follows. If $A_i[t, t+\tau]$ denotes an upper bound to the amount of data consumed by a stream s_i in the time interval $[t, t+\tau]$, an empirical envelope function is defined as:

$$\varepsilon_i(\tau) = \max_t A_i[t, t+\tau] \tag{1}$$

The function $\varepsilon_i(\tau)$ is defined for all τ, $0 \le \tau \le T_{total}$ where T_{total} defines the total playback duration of the video. Clearly, if $\tau = k \cdot r_i^{-1}$ for any integer k, τ takes the worst-case sum of k consecutive frames[7] within stream s_i. Therefore, if the value of τ corresponds to the playback time r_i^{-1} of a single frame of stream s_i, $\varepsilon_i(\tau)/\tau$ will correspond to the stream's peak bit rate,[8] and if τ corresponds to the playback time of the whole video, then $\varepsilon_i(\tau)/\tau$ will take the mean bit rate of stream s_i.

For any value of τ between the playback duration of one video frame and the video's duration, $\varepsilon_i(\tau)/\tau$ consequently takes values between the peak bit rate and the mean bit rate. We regard frames to be *consumed instantaneously* when they are scheduled to be sent off to the client, that is, between the deadlines of two consecutive frames no media data are consumed while at the point of the deadline, all data of the due frame are consumed at once. The empirical envelope $\varepsilon_i(\tau)$ will consequently take the form of a staircase function if we define:[9]

[6] The sequences are encoded with GOP-size 12 and pattern 'IBBPBBPBBPBB'. Each sequence contains 40,000 frames of size 384×288 picture elements with 24 bit color information. We simulate replay of the sequences at 30 frames per second.

[7] Without loss of generality we refer to the video information segments that are sent to the client as frames. However, several segments may make up one video image or may contain several images, depending on the chosen granularity, and may contain audio information as well.

[8] The peak bit rate is defined as the size of the largest frame divided by the playback duration of one frame.

$$\varepsilon_i(\tau) = \begin{cases} 0 & \text{for } \tau = 0 \\ \max\limits_t A[t, t + kr_i^{-1}] & \text{for } \tau \in ((k-1)r_i^{-1}, kr_i^{-1}], k = 1, 2, \ldots \end{cases} \tag{2}$$

For any scheduling discipline, using admission control tests with the empirical envelope results in the *highest resource utilization* achievable in a deterministic service with full VCR capabilities. By definition, the empirical envelope is an *optimal* traffic constraint function because no other traffic constraint function can for a given interval state a worst-case video traffic that is less than the value of the empirical envelope [Knig95].[10]

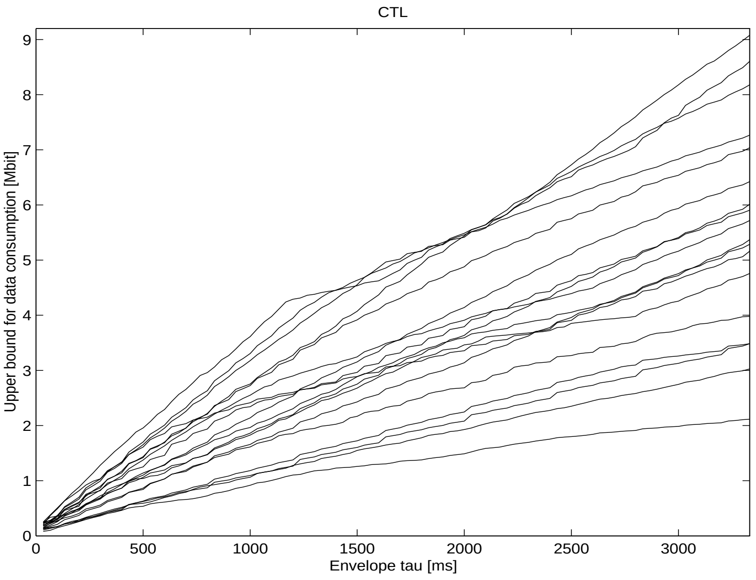


Fig. 5. Empirical envelopes as optimal traffic constraint functions.

We use empirical envelope functions $\varepsilon_i(\tau)$ that were obtained from the eighteen available video traces. Function values for $0 \le \tau \le 3333$ ms are shown in figure 5. It is reasonable to assume that the required video trace information is available in a video server since it can be very easily extracted from the frame size information contained within JPEG or MPEG frame headers. The computational complexity of traffic constraint functions, such as the empirical envelope, is also not very critical since we assume most on-demand multimedia services be rather read-only [Gemm95]. A traffic constraint function like the empirical envelope can therefore always be computed once *off-line*.

[9] Note that in [Knig95], values of $\varepsilon_i(\tau)$ for the intervals $((k-1)r_i^{-1}, kr_i^{-1}], k = 1, 2, \ldots$ are obtained by interpolation between $\varepsilon_i((k-1)r_i^{-1})$ and $\varepsilon_i(kr_i^{-1})$.

[10] To see this, consider another traffic constraint function $\tilde{\varepsilon}$, with $\tilde{\varepsilon}(\tau) < \varepsilon(\tau)$ for $\tau \in [0, T_{total}]$. By definition (1), $\tilde{\varepsilon}(\tau)$ must then be the empirical envelope.

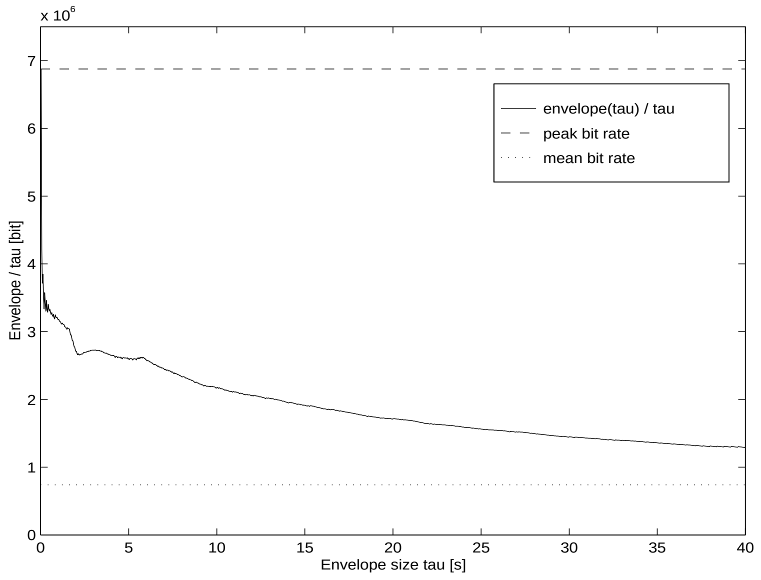


Fig. 6. Worst-case traffic $\varepsilon(\tau)$ in τ, divided by τ

For any value of τ between the playback duration of one video frame and the video's total playback time, $\varepsilon_i(\tau)/\tau$ consequently takes values between the peak bit rate (dashed line) and the mean bit rate (dotted line). Specifically, if τ equals the playback duration of one frame, $\varepsilon_i(\tau)/\tau$ will take the peak bit rate of the video, and if τ corresponds to the duration of the whole video, then $\varepsilon_i(\tau)/\tau$ will take the mean bit rate of the video stream. All values given for the MTV video trace.

Figure 6 underscores the significance of the interval τ with respect to the worst-case observed traffic bit rate $\varepsilon_i(\tau)/\tau$ in τ. When τ increases, the ratio, $\varepsilon_i(\tau)/\tau$ quickly decreases from the peak bit rate,[11] illustrated by the dashed line, and converges toward the mean bit rate that is indicated by the dotted line.

We now proof some useful properties of $\varepsilon_i(\tau)$:

Theorem 1. $\varepsilon_i(\tau_a) + \varepsilon_i(\tau_b) \geq \varepsilon_i(\tau_a + \tau_b) \quad \forall \tau_a, \tau_b > 0,\ i \in \{1, \ldots, n\}$ (3)

Proof. Let $i \in \{1, \ldots, n\}$ and $\tau_a, \tau_b > 0$ be arbitrary but fixed.
From definition (3) holds $\varepsilon_i(\tau_a) = \max_t A_i[t, t + \tau_a]$ and
$\varepsilon_i(\tau_b) = \max_t A_i[t, t + \tau_b]$.
$\Rightarrow \exists\ \dot{t}_a, \dot{t}_b$ such that $\varepsilon_i(\tau_a) = A_i[\dot{t}_a, \dot{t}_a + \tau_a]$ and $\varepsilon_i(\tau_b) = A_i[\dot{t}_b, \dot{t}_b + \tau_b]$.
$\Rightarrow \varepsilon_i(\tau_a) \geq A_i[t_a, t_a + \tau_a] \quad \forall t \neq \dot{t}_a$ and $\varepsilon_i(\tau_b) \geq A_i[t, t + \tau_b] \quad \forall t \neq t_b$
Therefore, we conclude that $\exists\ t_c$ such that:

[11] Note that this most likely resembles the size of the largest I-frame in the MPEG-1 video trace, divided by its playback duration.

$$
\begin{aligned}
\varepsilon_i(\tau_a + \tau_b) &= \max_t A_i[t, t + \tau_a + \tau_b] \\
&= A_i[\dot{t}_c, \dot{t}_c + \tau_a + \tau_b] \\
&= A_i[\dot{t}_c, \dot{t}_c + \tau_a] + A_i[\dot{t}_c + \tau_a, \dot{t}_c + \tau_a + \tau_b] \\
&\le \varepsilon_i(\tau_a) + \varepsilon_i(\tau_b)
\end{aligned}
$$

Corollary 1. $\varepsilon_i(m_i\tau) \le m_i\varepsilon_i(\tau) \quad \forall\tau > 0\,,\ m_i \in \{1, 2, \ldots\}\,,\ i \in \{1, \ldots, n\}$ (4)

Therefore, with growing τ_i the worst-case disk I/O requirement decreases. Theorem 2 states that the empirical envelope is *monotonously increasing* over its domain. Because of (9) and theorem 2, the total buffer requirement for stream s_i will increase if a larger value for τ_i is chosen:

Theorem 2. $\varepsilon_i(\tau)$ is monotonously increasing over its domain $[0, T_{total}]$, that is,
$\varepsilon_i(\tau) \ge \varepsilon_i(\tau') \quad \forall\tau, \tau' \in [0, T_{total}]\,,\ \tau \ge \tau'\,,\ i \in \{1, \ldots, n\}$ (5)

Proof. Let $\tau, \tau' > 0$ with $\tau \ge \tau'$ be arbitrary but fixed.
From definition (3) holds $\varepsilon_i(\tau) = \max_t A_i[t, t + \tau]$ and $\varepsilon_i(\tau') = \max_t A_i[t, t + \tau']$.
$\Rightarrow \exists\ t_1, t_2$ such that $\varepsilon_i(\tau) = A_i[t_1, t_1 + \tau]$ and $\varepsilon_i(\tau') = A_i[t_2, t_2 + \tau']$

$$
\begin{aligned}
\Rightarrow \varepsilon_i(\tau) &= A_i[t_1, t_1 + \tau] \\
&\ge A_i[t_2, t_2 + \tau] \\
&= A_i[t_2, t_2 + \tau'] + A_i[t_2 + \tau', t_2 + \tau] \\
&\ge A_i[t_2, t_2 + \tau'] + 0 \\
&= \varepsilon_i(\tau')
\end{aligned}
$$

Theorem 3. Consider the modified domain $M = \{0, r_i^{-1}, 2r_i^{-1}, \ldots\}$. Then $\varepsilon_i(\tau)$ is even strictly monotonously increasing over *M*, that is,
$\varepsilon_i(\tau) > \varepsilon_i(\tau')\,,\ \forall\tau, \tau' \in M\,,\ \tau > \tau'\,,\ i \in \{1, \ldots, n\}$ (6)

Proof. The proof is analogous to the proof of theorem 2.■

4 Constant Time Length Retrieval

4.1 Disk Model

Disk throughput is maximized when the seek times are minimized. We consider round-based retrieval techniques where each stream is served at-most once during each round. We use the **SCAN** algorithm where the head sweeps back and forth between the edge of the disk and its center serving the requests of one round. By ordering the requests according to the position of the data on the disk, SCAN minimizes the seek overhead.

Before we can formulate the exact admission control criterion for a single disk using the SCAN algorithm, we list all factors that need to be taken into account when computing the total time it takes to serve *n* streams during one round.

Let t_{track}, t_{seek} and t_{rot} express the **track-to-track seek time**, the **maximum seek time** and the **maximum rotational latency**, respectively, and let r_{disk} denote the **disk transfer rate**, and c_{cyl} the **capacity of a disk cylinder**. For the SCAN disk scheduling algorithm, in the worst case the maximum rotational latency of t_{rot} is introduced for each of the *n* streams. Furthermore, in the worst case a **full seek operation** across all cylinders is carried out during each round, which takes t_{seek}.

4.2 Constant Time Length Retrieval

After considering buffer-conserving CTL data placement we generalize CTL data placement and derive a novel data retrieval scheme for non-buffer-conserving CTL. The scheme contains both buffer-conserving CTL and PCBR as special cases. We finally summarize the fundamental trade-offs that exist between disk utilization, buffer demand, and start-up latencies.

Deterministic Admission Control

Consider the deterministic upper bound to the media traffic of a video in an interval τ given by the empirical envelope function $\varepsilon_i(\tau)$. Let D_i be the amount of data consumed by s_i during a disk round and let $\tilde{D}$ denote the sum of data that is consumed by all streams during a disk service round. Clearly, if τ corresponds to the duration of a disk service round, then $\varepsilon_i(\tau)$ will give an upper bound to the random variable D_i. An upper bound to the random variable $\tilde{D}$ can therefore be given by:

$$\tilde{D} \le \sum_{i=1}^{n} \varepsilon_i(\tau) \qquad (7)$$

Consequently, we obtain the **admission control criterion** that restricts the total disk I/O demand to satisfy the following condition:

$$\sum_{i=1}^{n} \frac{\varepsilon_i(\tau)}{r_{disk}} + \sum_{i=1}^{n} \left\lfloor \frac{\varepsilon_i(\tau)}{c_{cyl}} \right\rfloor \cdot t_{track} + n \cdot (t_{track} + t_{rot}) + t_{seek} \le \tau \qquad (8)$$

A lower bound for the total buffer requirement b_{total} is given by:

$$2\sum_{i=1}^{n} \varepsilon_i(\tau) \le b_{total} \qquad (9)$$

However, this lower bound can be improved (reduced) because of the well-known worst-case characteristics of *subsequent data requests*. Consider two consecutive media data requests for stream s_i. Each one can in the worst case produce a read request for $\varepsilon_i(\tau)$ media data. Now assume that the requests refer to adjacent video quanta, that is, the video data of the second request is meant to be played back right after the data produced by the first request.[12] Two subsequent requests can then only total $\varepsilon_i(2\tau)$. According to corollary 1, this is always smaller than or equal to $2\varepsilon_i(\tau)$. The deterministic total buffer requirement hence decreases to:

$$\sum_{i=1}^{n} \varepsilon_i(2\tau) \le b_{total} \tag{10}$$

4.3 Generalization of Constant Time Length Retrieval

Up to now, all papers on periodic retrieval schemes with deterministic service guarantees have assumed that

- the length of the **disk service round**, during which data for each stream are read exactly once from disk, and
- the length of the **CTL round,** for which we consider the worst case data consumption given by $\varepsilon_i(\tau)$ have the *same duration*. We are now going to distinguish the two and demonstrate the advantages of doing so.

The disk scheduling is still assumed to proceed in rounds of length τ. Additionally, we introduce a set $T = \{\tau_1, \ldots, \tau_n\}$ of **CTL rounds** depending on the video characteristics. The media data for stream s_i are retrieved from the disk such that during each τ_i, $\tau_i \ge \tau$, enough data to last for τ_i are retrieved. The CTL round duration τ_i must be a multiple m_i of the disk service round duration τ. The admission control criterion can now be extended to the use of $T = \{\tau_1, \ldots, \tau_n\}$ by the assumption that a request for media data of stream s_i is evenly distributed over $m_i = \tau_i/\tau$ disk service rounds. While over τ_i, the scheme is still buffer-conserving, i.e. the number of consumed frames equals the number of retrieved frames, in any disk service round within the CTL round, fewer frames may be read from the disk than be consumed by the client. The worst case data request during each disk service round, that is, an upper bound to $\tilde{D}$, is given by:

$$\sum_{i=1}^{n} \left\lceil \frac{\varepsilon_i(\tau_i)}{m_i} \right\rceil \tag{11}$$

Because of corollary 1, the worst-case data request is smaller than or equal to the corresponding sum of data requests for the case of $\tau_1, \ldots, \tau_n = \tau$ that was covered in the previous section:

[12] This can always be assumed if only the VCR functions play, pause and stop are considered.

Admission Control

Analogous to (8), we can now state the admission control criterion for restricted disk I/O:

$$\sum_{i=1}^{n}\left\lceil\frac{\varepsilon_i(\tau_i)}{m_i}\right\rceil r_{disk}^{-1}+\sum_{i=1}^{n}\left\lfloor\left\lceil\frac{\varepsilon_i(\tau_i)}{m_i}\right\rceil c_{cyl}^{-1}\right\rfloor\cdot t_{track}+n\cdot(t_{track}+t_{rot})+t_{seek}\le\tau \qquad (12)$$

The effect of the choice τ_i and τ on the number of clients that can be admitted to deterministic service is demonstrated in figure 7. The disk service round duration is varied between 33 ms and 3 s while the CTL parameter m_i takes integer values between 1 and 12. A CTL can therefore be up to 36 s long. Obviously, the smallest number of streams can be admitted if both parameters are chosen small.

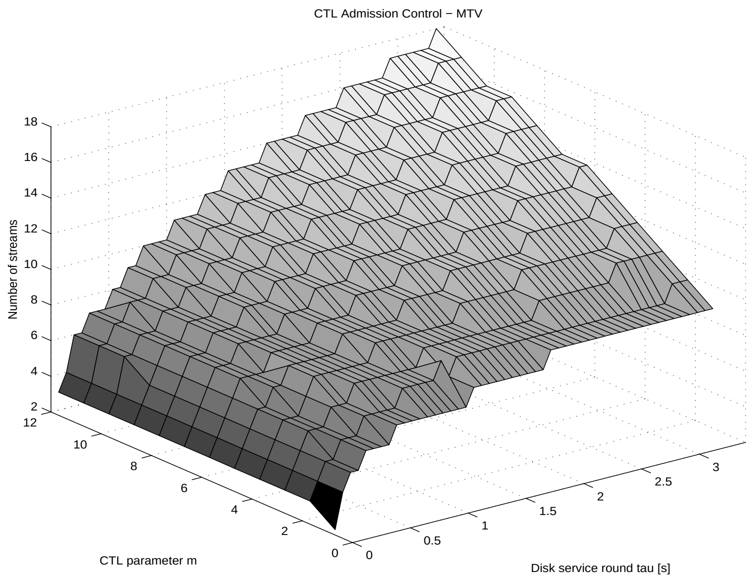


Fig. 7. Effect of service round and CTL parameter on the number of streams

In particular, there are two noteworthy effects of how the parameters τ and τ_i are chosen:

- The right edge of the graph indicates the curve for traditional, buffer-conserving CTL, i.e. $\tau = \tau_i$. The maximum number of admitted streams rises with a growing parameter τ for two reasons: first, as τ grows toward the duration of a video T_{total}, the amount of media data $\varepsilon_i(\tau)$ that must be retrieved in the worst case tends to get characterized by the mean bit rate instead of the peak bit rate.[13] The second reason for better disk I/O efficacy is given by a decrease in seek overhead when

[13] Recall that if τ takes the playback duration of one frame, then $\varepsilon_i(\tau)$ will equal the peak bit rate. For more detailed explanation, see section 3.3.1.

the media data are retrieved in larger segments.

- All other values depict admission control results for non-buffer-conserving CTL. Obviously, much larger numbers of simultaneous streams can be admitted. Notice that the service is still deterministic in nature, i.e. the server can guarantee the timely play-out of every frame of the video. Naturally, the greatest number of streams will be admitted if both τ_i and τ take large values.

Throughout the following section we assume τ to remain constant and evaluate the effect of τ_i with respect to the optimal start-up latency and the buffer requirements under deterministic conditions.

Start-up Latency and Buffer Requirements

The **start-up latency** is defined as the delay between user interaction and feedback by the server. Note that the delay introduced by the network and by buffering at the client site, for instance in order to synchronize several streams, is not being considered. Thus, the start-up latency of a video server is given by how long it takes the server from the reception of a playback request until the first frame is submitted to the network.

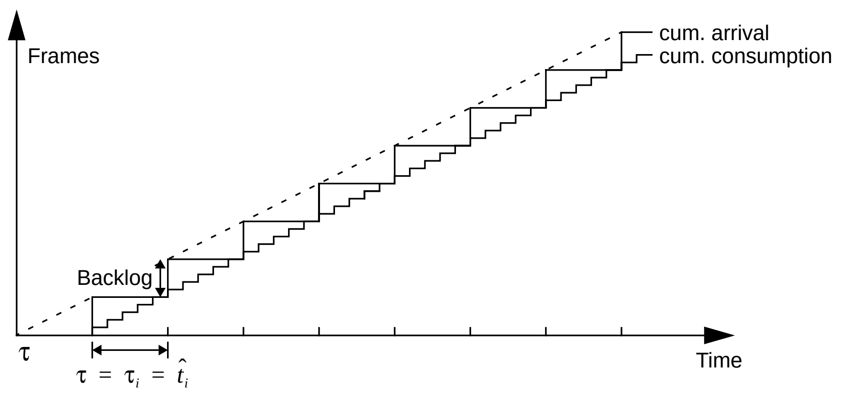
Generally, a delay of τ is introduced because a client may have to wait for a whole disk service round duration before its request can be processed. The video server must then delay the play-out for another period $\hat{t}_i$ after the video request is processed so to guarantee that buffer starvation is avoided during the playback of the video. Consequently, the start-up latency is generally given by $\tau + \hat{t}_i$.

When using the concept of alternating buffers[14] with a CTL round of duration τ_i that equals the disk service round duration, $r_i \cdot \tau$ frames are retrieved from the disk into the buffer till the end of the first round. The frames are then sent to the client during the second round of length τ. Therefore, $\hat{t}_i$ must equal τ, resulting in an overall start-up latency of 2τ. This situation is depicted in figures 8 (a-b).

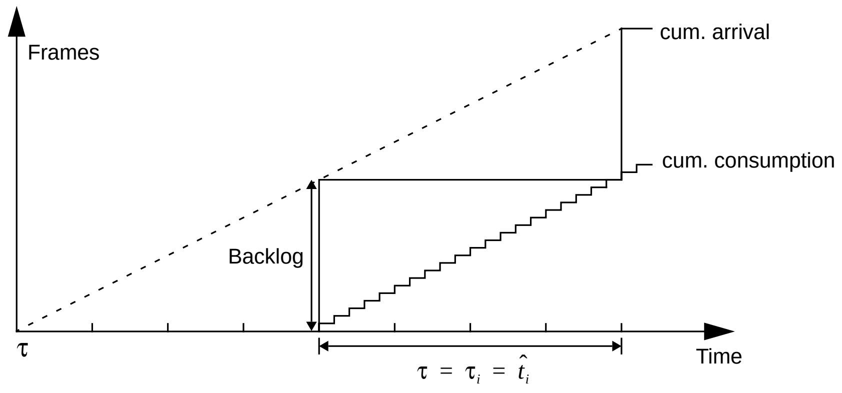
With a disk service round of τ, $\tau < \tau_i$, both the buffer requirements and the start-up latency can under deterministic conditions be reduced compared to a disk service round of τ_i. Every m_i subsequent disk service round of length τ, the video server reads sufficient media data to gain τ_i video playback for stream s_i. Yet the server does not have to delay playback for as long as τ_i because the data received from a number of disk service rounds before the expiration of τ_i may permit the server to play out frames under deterministic conditions. The server can so guarantee that buffer starvation is avoided even if playback is delayed for a shorter time than τ_i. This is demonstrated in figure 8 (c) where $\hat{t}_i$ is considerably smaller than τ_i.

Let random variable Z_i denote the number of frames contained within the media data D_i of stream s_i that are retrieved during a disk service round, and let observations of Z_i be denoted as $z_{i,l}$.

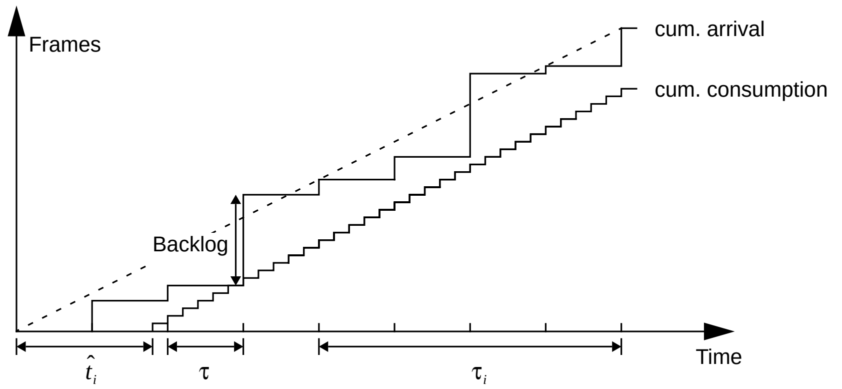
[14] The concept implies that one buffer gets refilled from storage while the other is emptied to the network.



(a) Small disk service round τ and parameter $\tau_i = \tau$



(b) Large disk service round τ and parameter $\tau_i = \tau$



(c) CTL parameter τ_i is a multiple of τ, i.e. $\tau_i = m_i\tau$

Fig. 8. Traditional and generalized CTL data placement

The observation $\hat{z}_{i,t}$ of the total number of frames $\hat{Z}_t$ read for stream s_i at time t (**cumulative arrival**) can be stated as (13) while the **cumulative consumption** of frames $\hat{c}_{i,t}$ at time t is given by (14):

$$\hat{z}_{i,t} = \begin{cases} 0 & \text{for } t \le \tau \\ \sum_{l=1}^{\lfloor t/\tau \rfloor} z_{i,l} & \text{for t} > \tau \end{cases} \tag{13}$$

$$\hat{c}_{i,t} = \begin{cases} 0 & \text{for } t \le \hat{t}_i \\ \lfloor r_i (t - \hat{t}_i) \rfloor & \text{for } t > \hat{t}_i \end{cases} \tag{14}$$

Obviously, buffer starvation is avoided if the **backlog function**, that is the number of frames $\hat{z}_{i,t} - \hat{c}_{i,t}$ contained in the buffer, is greater than or equal to zero at all times. The problem of minimizing the start-up latency thus leads to the following rule for the **play-out delay** $\hat{t}_i$ (the total duration of the video is denoted by T_{total}):

$$\hat{t}_i = \min\left(t_i \in [\tau, T_{total}] \mid \sum_{l=1}^{\lfloor t/\tau \rfloor} z_{i,l} \ge \lfloor r_i \cdot (t - t_i) \rfloor, \forall t \in (t_i, T_{total}] \right) \tag{15}$$

The deterministic buffer requirement b^i_{total} for stream s_i is closely related to the worst case backlog. We first define the **maximum backlog** bl^i_{max} for stream s_i in terms of frames as

$$bl^i_{max} = \max_t \left(\sum_{l=1}^{\lceil t/\tau \rceil} z_{i,l} - \lfloor r_i (t - \hat{t}_i) \rfloor \right) \tag{16}$$

The **deterministic buffer requirement** b^i_{total} for stream s_i is then derived using the empirical envelope function as

$$\varepsilon_i\left(\frac{bl^i_{max}}{r_i} \right) \le b^i_{total} \tag{17}$$

Notice that an arbitrary round-based disk scheduling algorithm may return video data for a stream at any time during a disk service round because concurrent requests are reordered to minimize the seek overhead. The worst case with regard to the start-up latency is thus given by assuming the *latest* possible arrival of media data while the worst case concerning the minimum buffer requires all media data to arrive as *early* as possible.

If VCR functionality, such as pause, stop and play, has to be provided to the client, the starting position within a video can also be very significant because playback starting at different frames produces different values for $\hat{z}_{i,t}$.

Theorem 4 states that throughout the playback of a stream, after each period of τ_i, there always remain the same number of frames in the buffer. This number is determined by the play-out delay $\hat{t}_i$. When using $\tau_i \equiv \tau$, for instance, the buffer is always half empty in terms of frames at the end of a round because the play-out delay equals the disk service round duration as shown in figures 8 (a–b).

It is therefore sufficient to calculate the play-out delay requirements for the first $\lfloor r_i \tau_i \rfloor$ frames as playback starting positions because all subsequent starting frames of the video can be viewed as phase-shifts from one of the first $\lfloor r_i \tau_i \rfloor$ frames, that is, they will periodically produce the same numbers for the deterministic start-up latency. To find the minimum play-out delay for a starting frame, one has to assume $\hat{t}_i = 0$ so to cause one or several conditions of buffer underflow $\hat{z}_{i,t} - \hat{c}_{i,t} < 0$. To prevent buffer starvation, the play-out must be delayed by at least the number of frames of the worst encountered buffer underflow, i.e. by $\min_t \{\hat{z}_{i,t} - \hat{c}_{i,t}\} \cdot r_i^{-1}$.[15]

The overall minimum play-out delay $\hat{t}_i$ is then determined by the smallest sufficient play-out delay of all observed starting frames. Again, both the minimum start-up latency $\tau + \hat{t}_i$ and the corresponding minimum buffer requirement can be computed once when the video is stored on the video server.

Theorem 4. Let $\tau_i > 0$ be arbitrary but fixed, $k, l \in \{1, 2, \ldots\}$. Then follows

$$\hat{z}_{i,t} - \hat{c}_{i,t} \equiv \hat{z}_{i,t'} - \hat{c}_{i,t'} \qquad \forall t, t' : t = k\tau_i,\ t' = l\tau_i . \tag{18}$$

Proof. Let $k, l \in \{1, 2, \ldots\}$, $t = k\tau_i$ and $t' = l\tau_i$ with $\tau_i > 0$ arbitrary but fixed. Because during τ_i the server is guaranteed to retrieve $r_i\tau_i$ frames, $\Rightarrow \hat{z}_{i,t} = r_i t$ and $\hat{z}_{i,t'} = r_i t'$.

From (14) then follows:

$$\begin{aligned}
\hat{z}_{i,t} - \hat{c}_{i,t} &= r_i t - \lfloor r_i (t - \hat{t}_i) \rfloor \\
&= k r_i \tau_i - \lfloor k r_i \tau_i - r_i \hat{t}_i \rfloor \\
&= k r_i \tau_i - k r_i \tau_i + \lfloor r_i \hat{t}_i \rfloor \\
&= \lfloor r_i \hat{t}_i \rfloor \\
\hat{z}_{i,t'} - \hat{c}_{i,t'} &= r_i t' - \lfloor r_i (t' - \hat{t}_i) \rfloor \\
&= l r_i \tau_i - \lfloor l r_i \tau_i - r_i \hat{t}_i \rfloor \\
&= l r_i \tau_i - l r_i \tau_i + \lfloor r_i \hat{t}_i \rfloor \\
&= \lfloor r_i \hat{t}_i \rfloor \\
&= \hat{z}_{i,t} - \hat{c}_{i,t}
\end{aligned}$$

[15] Similar methods for determining the start-up latency are stated in detail in [Chan94],[dR95].

In order to demonstrate the effect of the choice of τ_i, we have calculated minimum values of $\hat{t}_i$ and the associate buffer requirement for the MTV video trace independent of the starting frame.[16]

The values for $\hat{t}_i$ presented in table 1 allow for full VCR functionality because we consider the start-up latency of the worst case. The figures indicate that when using non-buffer-conserving CTL, one can reduce the start-up latency compared to buffer-conserving CTL by sacrificing disk I/O efficiency. For example, buffer-conserving CTL with a service round of 4 s entails a worst-case latency of 8 s whereas non-buffer-conserving CTL, for a disk service round of 1 s and a CTL round of 4 s, only requires a latency of 3.2 s.

τ [s]	τ_i [s]	Start-up latency		Buffer requirement for one stream		Number of admitted streams	
		Seconds	%	Bytes	%	Total	%
1	1	2.0	100%	676,724	100%	7	100%
1	2	2.6	132%	1,044,698	154%	8	114%
1	3	2.7	135%	1,182,035	175%	8	114%
1	4	3.2	160%	1,324,987	196%	9	129%
1	1,333	66.8	3,340%	12,362,938	1,826%	23	329%
1	1	2.0	100%	676,724	100%	7	100%
2	2	4.0	200%	1,324,987	195%	9	129%
3	3	6.0	300%	1,937,257	286%	9	129%
4	4	8.0	400%	2,339,260	346%	10	143%

Table 1. Start-up latency, buffer requirements, and admitted streams for CTL

For comparison, corresponding values in the lower half of the table are for the case that a *disk service round* equals the *CTL round*.

Values for "Start-up latency", correspond to $\hat{t}_i + \tau$. Column "Buffer requirement for one stream," states the corresponding minimum buffer requirement. The value for $\tau_i = \tau = 1\text{s}$, for instance, corresponds to $\varepsilon_i(2\tau)$. Row 5 contains the values for PCBR retrieval ($\tau_i = 1{,}333\text{ s}$) which is a special case of CTL data placement. All values are given for the MTV video trace.

If τ_i equals the duration of the whole video,[17] the proposed scheme effectively becomes the PCBR retrieval presented in [Chan94],[dR95]. Therefore, we provide a generalization for PCBR and CTL retrieval that covers the continuum between the two. We have also computed the deterministic buffer requirement.[18] The figures in

[16] For $\tau = 1\text{ s}$ and $m_i = \{1, 2, 3, 4, 1.333\times10^3\}$.

[17] $\tau_i = 1.333\times10^3$ results in PCBR retrieval at the mean bit rate for a video of roughly twenty minutes (30 fps).

table 1 demonstrate that the buffer requirement grows s*ignificantly slower* than m_i. For example, with a disk service round of 1 s and a CTL round of 1 s, the deterministic buffer requirement is 676,724 bytes. If we keep a disk service round of 1 s and increase the CTL round by a factor of 4 to 4 s, the buffer requirement will less than double from 676,724 bytes to 1,324,987 bytes. However, when choosing a disk service round equal to the CTL round, both of 4 s, the buffer requirement almost quadruples to 2,339,260. Hence, almost 1 Mbyte can be saved *per stream* if non-buffer-conserving CTL is employed.

This leads us to the following observations:

- When increasing the CTL round duration, one can admit more streams, i.e. improve the I/O efficiency at the expense of more buffer space and a higher start-up latency.
- When employing *non*-buffer-conserving CTL, the buffer demand and start-up latency are lower than for buffer-conserving CTL.
- The PCBR retrieval technique proposed in [Chan94],[dR95] is a *special case* of non-buffer-conserving CTL.

PCBR requires a start-up latency for the MTV video trace of well over *one minute*, and corresponding buffer allocation of close to *12 Mbyte per admitted stream*. Note that the play-out of the stream must be delayed for this long not only at the beginning of the playback but also after each pause. Similar results showing that PCBR is extremely demanding in terms of buffer space have been obtained by de Rosario and Fox [dR95] who have evaluated the buffer demand and start-up latency only for PCBR.

5 Conclusion

There are various papers that consider buffer-conserving CTL retrieval [Chan94]. However, we are the first to propose *non*-buffer-conserving CTL schemes where the CTL round length is *decoupled* from the disk round length. This permits to individually adapt the duration of the CTL round length *for each stream* to the buffer and start-up latency constraints at the client, while maintaining a common disk round length.

The separation of the disk service round from the CTL round, as done for the non-buffer-conserving retrieval, pays off. For non-buffer-conserving retrieval as compared to buffer-conserving retrieval we observe that

- a slightly smaller number of streams can be admitted
- the buffer requirements and start-up latencies decline drastically.

[18] The minimum amount of buffer space that guarantees that buffer overflow is avoided, given the play-out delay $\hat{t}_i$.

Furthermore, the PCBR scheme which performs the retrieval of media data at the stream's mean bit rate, is shown to be a special case of CTL retrieval. While PCBR achieves the most efficient disk I/O, it suffers the largest start-up latency and buffer demand.

Acknowledgment

The work described in this paper was supported by the Siemens Nixdorf AG, Munich.

6 References

[Chan94] E. Chang and A. Zakhor. "Admissions Control and Data Placement for VBR Video Servers." In *Submitted to Proceedings of the 1st International Conference on Image Processing*, Austin, Texas, November 1994.

[Dey94] J. K. Dey, J. D. Salehi, J. F. Kurose, and D. Towsley. "Providing VCR Capabilities in Large-Scale Video Servers." In *Proceedings of the 2nd ACM International Conference on Multimedia*, pages 25–32, October 1994.

[dR95] J. M. de Rosario and G. Fox. "Constant Bit Rate Network Transmission of Variable Bit Rate Continuous Media in Video-On-Demand Servers." Technical Report SCCS-677, Northeast Parallel Architectures Center, 111 College Place, Syracuse University, Syracuse, NY, July 31 1995.

[Gemm95] 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.

[Knig94] 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, October 1994.

[Knig95] E. W. Knightly, D. E. Wrege, J. Liebeherr, and H. Zhang. "Fundamental Limits and Tradeoffs of Providing Deterministic Guarantees to VBR Video Traffic." In *Proceedings Joint International Conference on Measurement & Modeling of Computer Systems (Sigmetrics '95 / Performance '95)*, volume 23 of *Performance Evalution Review*, pages 98–107, Ottawa, Canada, May 15-19 1995. Also available as technical report TR-94-067, International Computer Science Institute, Berkeley, CA.

[Rang92] P. V. Rangan, H. M. Vin, and S. Ramanathan. "Designing an On-Demand Multimedia Service." *IEEE Communications Magazine*, 30(7):56–65, July 1992.

[Rose95] O. Rose. "Statistical Properties of MPEG Video Traffic and Their Impact on Traffic Modelling in ATM Systems." Technical Report 101, Institute of Computer Science, University of Wuerzburg, University of Wuerzburg, February 1995.

[Rowe93] L. A. Rowe and R. R. Larson. "A Video-on-demand System." The University of California at Berkeley and International Computer Science Institute, 1993.

[Stei91] R. Steinmetz and R. G. Herrtwich. "Integrierte verteilte Multimedia-Systeme." *Informatik-Spektrum*, 14:249–260, 1991.

Appendix

A Disk Model Characteristics

Throughout the simulations a disk model with the characteristics of the “Micropolis 4110 AV” was used:

Micropolis 4110AV	
Maximum seek time t_{seek}	20.0ms
Track-to-track seek time t_{track}	1.5ms
Maximum rot.l latency t_{rot}	11.11ms
Disk cylinder capacity c_{cyl}	4×10^6 bits
Disk transfer rate r_{disk}	24×10^6 bps

Table 2. Characteristics of the Micropolis AV 4110 hard disk drive