A Network-Conscious Approach to End-to-End Video Delivery over Wide Area Networks Using Proxy Servers

Yuewei Wang, Zhi-Li Zhang,* David H.C. Du, and Dongli Su
Department of Computer Science and Engineering
University of Minnesota
200 Union Street S.E., Minneapolis, MN 55455
Email: {ywang,zhzhang,du,dsu}@cs.umn.edu

Abstract

Real-time distribution of stored video over wide-area networks is a crucial component of many emerging distributed multimedia applications. The heterogeneity in the underlying network environments is an important factor that must be taken into consideration when designing an end-to-end video delivery system.

In this paper, we present a novel *network-conscious* approach to the problem of end-to-end video delivery over wide-area networks using proxy servers situated between local-area networks (LANs) and a backbone wide-area network (WAN). A major objective of our approach is to reduce the backbone WAN bandwidth requirement. Towards this end, we develop a novel and effective video delivery technique called *video staging* via intelligent utilization of the disk bandwidth and storage space available at proxy servers. Using this video staging technique, only part of a video stream is retrieved directly from the central video server across the backbone WAN whereas the rest of the video stream is delivered to users locally from proxy servers attached to the LANs. In this manner, the WAN bandwidth requirement can be significantly reduced, particularly when a large number of users from the same LAN access the video data. We design several video staging methods and evaluate their effectiveness in trading the disk bandwidth of a proxy server for the backbone WAN bandwidth. We also develop two heuristic algorithms to solve the problem of designing a multiple video staging scheme for a proxy server with a given video access profile of a LAN. Our results demonstrate that the proposed proxy-server-based, network-conscious approach provides an effective and scalable solution to the problem of the end-to-end video delivery over wide-area networks.

Keywords: End-to-End Video Delivery, Heterogeneous Networking Environment, LAN, MPEG, Proxy Server, Statistical Multiplexing Gains, Video Smoothing, Video Staging, WAN

---

*Zhi-Li Zhang was supported in part by a University of Minnesota Graduate School Grant-in-Aid grant and NSF CAREER Award grant NCR-9734428. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the National Science Foundation.

1 Introduction

Real-time distribution of stored video over high-speed networks is a crucial component of many emerging multimedia applications including distance learning, digital library, Internet TV broadcasting and video-on-demand systems. Because of its high bandwidth requirement, video is typically stored and transmitted in compressed format. As a result, video traffic can be highly bursty, possibly exhibiting rate variability spanning multiple time scales. This is particularly the case when constant-quality variable-bit-rate (VBR) compression algorithms are used [9]. Due to the bursty nature of compressed video, support for quality-of-service (QoS) guarantees for real-time transport of stored video across a network is therefore a challenging problem. This problem is further compounded when video is delivered over a wide-area network (WAN) where several heterogeneous networks are interconnected.

The heterogeneity in the underlying network environments is an important factor that must be taken into consideration in the design of many distributed multimedia applications. For example, consider a distance learning application in a large university which has several geographically separate campuses. Each campus has its own campus-wide high-speed local area network (LAN). These campus networks are typically interconnected to each other through a backbone wide-area network owned by a third party. Suppose that the distance learning center is situated in the main campus with a central video server supplying video-based multimedia course materials to all campuses over the wide-area network. The backbone WAN is typically shared by a large number of institutions or users, and it is generally more expensive to deploy additional resources in the backbone WAN than in local area networks. Given the emerging gigabit networking technologies such as Gigabit Ethernet and Fibre Channel, the cost of installing and running a local-area gigabit network becomes increasingly cheaper. On the other hand, the WAN bandwidth is a much more critical and costly resource[1] than that of campus-wide LANs. Therefore, reducing the total bandwidth requirement of the backbone WAN should be an important objective in the design of a real-time video delivery system in such a scenario. The heterogeneous networking environment of the aforementioned example is also fairly common in other settings, e.g., in a large corporation where its intranet consists of several geographically dispersed LANs interconnected by a wide-area network leased from a network service provider, or in a residential setting where several residential access networks (operated by one network service provider) are connected to a large backbone wide-area network operated by another service provider.

In this paper, we present a novel *proxy-server-based, network-conscious* approach to the end-to-end

[1] As an indication of the potential cost of backbone WAN bandwidth, the University of Minnesota, as one of the participants in the Internet-II project, recently leased an OC-3 link (with a bandwidth of 155Mb/s) from Minneapolis to Chicago which costs 1.7 million dollars biennially.

video delivery over wide-area networks. For simplicity of discussion, the wide-area network in question is assumed to comprise several local area networks interconnected by a backbone wide-area network (see Figure 1 for a simple example), although our approach can be applied to networks with more general topology and configuration. Video streams are delivered from a central video server through the backbone WAN to a large number of users in the local area networks. As part of the network system architecture, we also assume that a *special server with a disk storage system*, which we shall refer to as a *proxy (video) server*[2], is installed in each LAN and is directly attached to the gateway router connecting the LAN to the backbone WAN. This assumption is quite reasonable, given the relatively low cost of PC servers today. *The major objective of our proxy-server-based, network conscious approach is to reduce the bandwidth requirement in the backbone wide-area network*, whereas the bandwidth of LANs is assumed to be bountiful and thus not a major concern. We develop an effective video delivery technique called *video staging* via intelligent utilization of the disk bandwidth and storage capacity available at proxy serves attached to the LANs. The basic idea behind the video staging technique is to *prefetch a predetermined amount of video data and store them a priori at proxy servers* — this operation is referred to as *staging*. Using the video staging technique, only part of video data is retrieved directly from the central video server across the backbone WAN whereas the rest of the video data is delivered to users from proxy servers attached to the LANs. In this manner, the WAN bandwidth requirement can be significantly reduced, particularly when a large number of users from the same LAN access the video data.

Our proxy-server-based, network-conscious approach to the problem of end-to-end video delivery across wide-area networks has several salient features and advantages. Because of the large storage space at a proxy server, for a given video, a sizeable portion of its data can be staged at a proxy server. The video staging technique is designed in such a manner that the video data can be delivered across the backbone WAN using a constant-bit-rate (CBR) network service. Hence only fixed amount of bandwidth needs to be reserved from the central video server across the backbone WAN to a LAN, allowing simple admission control and scheduling mechanisms to be employed to ensure QoS guarantees for video delivery across the backbone WAN. This bandwidth reservation can also be done on an aggregate basis when multiple video streams are delivered from the central video server across the backbone WAN to the same LAN, thereby further simplifying the resource management and control in the backbone WAN. Furthermore, since the disk bandwidth and storage capacity available at a proxy server are shared by all users attached to the same LAN, statistical multiplexing gains can be effectively exploited to improve

[2]Although we use "proxy server" as the name for this special server, however, as will be clear later, the usage of proxy server in our context of real-time video delivery is quite different from the typical usage of a proxy server as a caching device in the context of web-based data applications. Despite this difference, we decide to borrow the terminology *proxy server* for lack of better nomenclature.

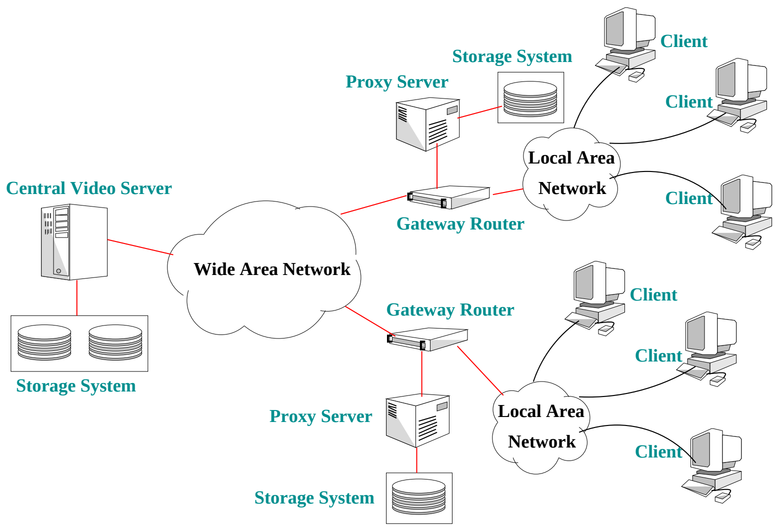


Figure 1: Video delivery over a simple heterogeneous networking environment

resource (e.g. disk bandwidth) utilization at the proxy server when multiple staged video streams are retrieved from the disk storage system of the proxy server across the LAN to various users on the LAN.

We design several video staging methods and study their effectiveness in trading the disk bandwidth of a proxy server for the backbone WAN bandwidth. Given this trade-off in the disk bandwidth requirement of proxy server and the backbone WAN bandwidth requirement for each video stream, we proceed to investigate the problem of how to determine the amount of video data from a collection videos to be staged at a proxy server with fixed disk bandwidth and disk storage space. We develop two heuristic algorithms to solve this problem. We evaluate our approach using simulations based on MPEG-1 video traces. Our results demonstrate that the proposed proxy-server-based, network-conscious approach provides an effective and scalable solution to the problem of the end-to-end video delivery over wide-area networks.

The remainder of our paper is organized as follows. In Section 2, we describe our problem setting and present our proxy-server-based, network-conscious approach. In Section 3, we present various video staging techniques in the context of a single video stream. In Section 4, we develop two heuristic algorithms to solve the problem of designing multiple video staging scheme for a proxy server with a given video access profile of a LAN. In Section 5, comparison with related work is made. The paper is concluded in Section 6

2 Problem Setting

In this paper, we study the problem of end-to-end video delivery over heterogeneous networking environments. A simple example is shown in Figure 1, where several local area networks are interconnected by a backbone wide-area network. As an important part of the network system architecture, we also assume that a proxy video server is installed in each LAN and is directly attached to the gateway router connecting the LAN to the backbone WAN. A central video server system with a large disk farm is connected to the backbone WAN through a high-speed LAN backbone (from the perspective of clients in other LANs across the backbone WAN, the central video server system can be viewed as if it is attached directly to the backbone WAN). An exemplar video delivery architecture over a rather *simplistic* heterogeneous networking environment is shown in Figure 2, where the LANs are connected to the backbone WAN through OC3 links (with 155 Mb/s bandwidth) and the backbone WAN has two backbone switches connected by an OC48 links (with 2.48 Gb/s bandwidth). Upon request, video streams are delivered from a central video server across the backbone WAN to a large number of clients attached to the LANs.

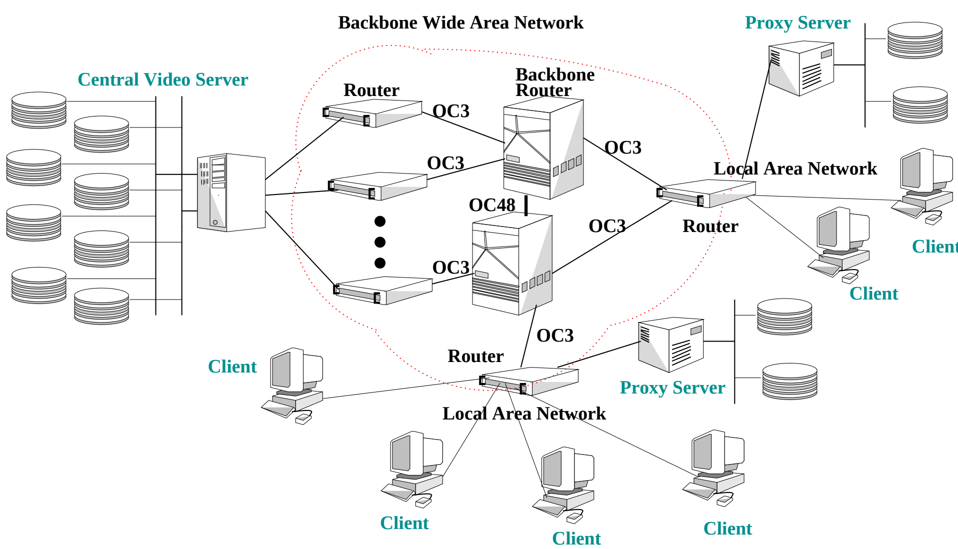


Figure 2: An exemplar proxy-server-based video delivery architecture

In a typical heterogeneous networking environment we consider in this paper, we assume that the backbone WAN and the LANs belong to different administrative domains, in other words, owned by different entities. There are frequently a large number of users concentrated at a LAN concurrently accessing the central video server across the backbone WAN. Under these circumstances, reducing the backbone WAN bandwidth required to delivery video from the central video server to users on the LANs is therefore a major objective in the design of the end-to-end video delivery system. Instead of replicating the central video server at each LAN, which is generally too expensive to be practical, installing inexpensive proxy (video) server with appropriate amount of resources such as disk bandwidth and storage space is likely to be a most feasible and cost-effective approach to achieve this objective.

The fundamental contribution of our proxy-server-based, network-conscious approach to the problem of end-to-end video delivery over heterogeneous networks is the notion of *video staging*. The basic idea behind the video staging technique is to prefetch a *predetermined* amount of video data and store them *a priori* at proxy servers — this operation is referred to as *staging*. Unlike the *caching* technique commonly used in a proxy web server, where data files are cached in and purged out of the proxy web server based on on-line prediction of the random user access pattern, the video staging technique we develop in this paper determines the video data to be staged at a proxy video server based on several important factors which we will explain below. The video data are staged at the proxy server on a fairly long period of time instead of caching in and purged out dynamically. For example, in the case of a distance learning application, staged video data can be determined on a daily basis based on the course materials offered each day and the expected user access pattern to these materials.

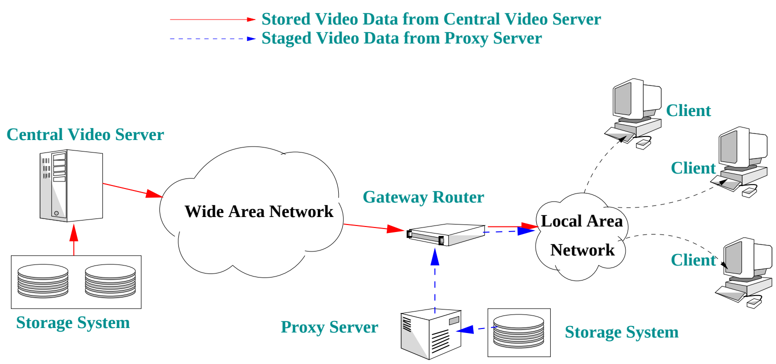


Figure 3: Proxy-server-assisted video delivery

The objective of the video staging technique is to reduce the total *expected* backbone WAN bandwidth required for delivering video to a large number of users on a LAN. As illustrated in Figure 3, for a given video, if a portion of its video data is staged at a proxy server attached to a LAN, then when a user on the LAN accesses the video, only part of the video data is retrieved directly from the central video server across the backbone WAN while the rest of the video data is delivered to the user from the proxy server. Since only a portion of the video data is transmitted across the backbone WAN, the bandwidth required is thus reduced. Moreover, if the video is accessed by a large number of users at the LAN, then this reduction in the backbone WAN bandwidth requirement can be significant. In the extreme case where the whole video is staged at the proxy server, then no backbone WAN bandwidth is required, and the video is delivered locally from the proxy server. Clearly, this reduction in the backbone WAN bandwidth requirement is achieved by consuming certain amount of resources such as the disk bandwidth and storage capacity at the proxy server. In light of the limited resources at a proxy server, it is therefore important to stage video data at a proxy server in such a manner that the expected total backbone WAN bandwidth required to deliver video to users on the associated LAN is maximally reduced while efficiently utilizing the resources at the proxy server.

For a given video, the decision of whether to stage the entire video, or a portion of it, or none of it at a proxy server hinges on many factors. One important factor is the effectiveness of video staging in reducing the backbone WAN bandwidth requirement for the given video. This effectiveness will depend on both the characteristics of the video and the method used to decide which portion of the video to be staged at the proxy server. Such a method is referred to as a *video staging method*. Another important factor is the access pattern of a LAN, namely, the expected concurrent accesses to the video during a certain period of time. If the video is expected to be accessed numerous times by a large number of users on a LAN, it will likely be a good idea to stage the entire or a large portion of the video at the proxy server to reduce the backbone WAN bandwidth required to transmit the video. On the other hand, if the video is seldomly accessed, it may be better only to stage a small portion of it or none at all so that the disk bandwidth and storage space can be used for staging other videos. Given the video collection at the central video server, the expected number of accesses to each video can be derived from user access pattern of a LAN. This information is referred to the *video access profile* of the LAN. For a proxy server with limited amount of resources, particularly limited amount of disk bandwidth and storage capacity, it is crucial to take both the video access profile and video characteristics into consideration when deciding the amount of video data to be staged at the proxy server. For a given collection of videos and a video access profile of a LAN, the problem of determining the amount of video data to be staged at the proxy server so as to minimize the total backbone bandwidth requirement is referred to as the *multiple video staging design* problem. The focus of the paper is thus on the developing video staging methods and, based on these methods, solving the multiple video staging design problem.

Before we delve into the details of our approach, we would like to point out that there are several implementation issues that must be resolved when applying the video staging technique in practice. For instance, if a portion of a video is staged at a proxy server while the rest of it is stored at the central video server, then these two portions of the video data must be synchronized during the playback of the video. This synchronization can be done either at the proxy server side or at the user side. In the former case, the video data stored at the central video server will be transmitted to the proxy server first, merged with the video data staged at the proxy server, and then delivered to a user. The disadvantage of this approach is that the processing capability of the proxy server can be a potential performance bottleneck. In the latter case, the two portion of the video data are delivered to a user separately, and then synchronized.

This requires extra buffering capability and incurs more overhead at the user side. Another related issue is the signaling of the video delivery system, i.e., the issue of sending control signals to both the central video server and the proxy server to initiate the playback of a video stream. For instance, these issues may be investigated in the context of RTP (Real-Time Transport Protocol) [22] and RTSP (Real-Time Streaming Protocol) [23] protocols. Investigation of these issues is outside the scope of this paper, and will be the topics of future research.

3 Video Staging: a Single Video Case

The effectiveness of staging video data at a proxy server in reducing the backbone bandwidth requirement can be measured by the ratio of the amount of the backbone WAN bandwidth reduction to that of the disk bandwidth required at the proxy. This ratio will be referred to as *bandwidth reduction ratio*. In this section, we present several video staging methods for a single video, and based on these methods, we study the effectiveness of video staging in reducing the backbone WAN bandwidth. We also integrate the optimal video smoothing technique [21] into the design of video staging methods to achieve further backbone WAN bandwidth reduction when clients have extra buffering capabilities available for smoothing.

3.1 Video Staging without Smoothing

We first consider the case where clients have no extra buffering capabilities available for smoothing and describe a simple video staging method for this case. This simple method will form the basis for our study. In order to simplify resource management at the backbone WAN, we will assume that a CBR network service with minimal delay and no loss is used for video transport across the backbone WAN. Without the presence of a proxy server, when a video is delivered from the central video server to a user at a LAN, the bandwidth reserved across the backbone WAN must then be equal to the peak rate of the video. With the presence of a proxy server, however, we can stage a portion of the video at the proxy server so that a portion of the video data is delivered directly from the proxy server to a user on the LAN. In this way, we can use the resources (disk bandwidth and storage space among others) available at the proxy to reduce the backbone WAN bandwidth required for video delivery across the backbone WAN. However, this reduction is achieved by devoting certain amount of disk bandwidth and storage capacity available at the proxy server to store and deliver the staged video data. Since the resources (especially the disk bandwidth) at the proxy server are limited, it is important to consider the effectiveness of video staging in reducing the backbone WAN bandwidth for a given video.

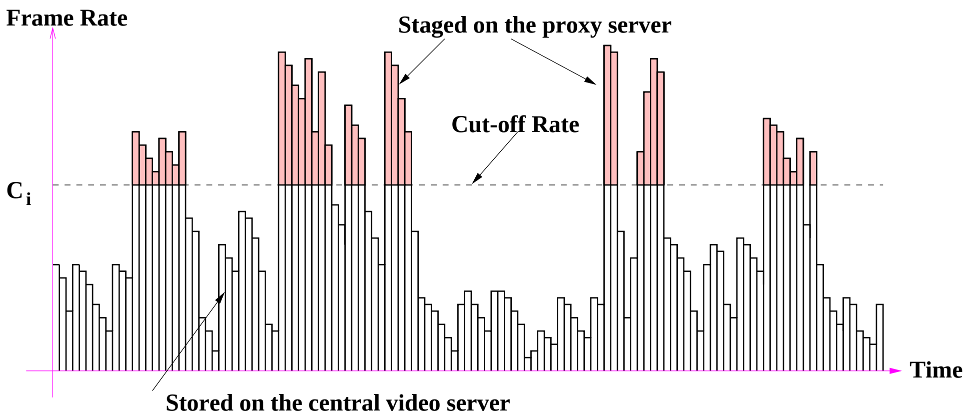


Figure 4: A simple video staging method using a cut-off rate

Consider a video indexed by i. Let F be its frame period (measured in seconds), i.e., the time interval during two consecutive frames are displayed, and let N_i be its total number of frames. For $j = 1, \ldots, N_i$, the size of the jth frame is s_i^j bits. Then the peak rate of this video, P_i, measured in bits per second, is given by $P_i = (\max_{1 \le j \le N_i} s_i^j)/F$. As a simple video staging method, we choose a *cut-off rate* C_i, where $0 \le C_i \le P_i * F = \max_{1 \le j \le N_i} s_i^j$, and divide video i into two parts as illustrated schematically in Figure 4. The lower part consists of a sequence of partial frames with size $s_i^{j,l} = s_i^j - (s_i^j - C_i)^+$, $j = 1, \ldots, N_i$, where $x^+ = \max\{x, 0\}$. The upper part consists of a sequence of partial frames with size $s_i^{j,u} = (s_i^j - C_i)^+$, $j = 1, \ldots, N_i$. The upper part will be duplicated and staged at the proxy server whereas the lower part will remain stored at the central server[3] (in fact, the whole video is always stored at the central video server). From Figure 4, we note that the smaller C_i is, the more video data is staged at a proxy server. Moreover, as C_i decreases, the lower part of the video becomes less burstier, and eventually approaches to an essentially constant-bit-rate stream.

During the playback of video i, only the lower part of the video is transferred from the central video server across the backbone WAN, thus reducing the backbone WAN bandwidth requirement from P_i to $T_i = C_i/F$. The upper part of the video is delivered directly from the proxy server, consuming $D_i = (\max_{1 \le j \le N_i} s_i^{j,u})/F$ amount of disk bandwidth in the worst case. It also consumes an amount of disk storage space equal to $\sum_{j=1}^{N_i} s_i^{j,u}$. Define the *bandwidth reduction ratio*, denoted by R_i, as the ratio of the backbone WAN bandwidth reduction to the disk bandwidth consumed at the proxy server. Then $R_i = \frac{P_i - T_i}{D_i}$. When there are a large number of concurrent accesses from users on the LAN, the effective

[3]Since the upper part contains the "bursty portion" of the video data while the lower part the "less bursty" portion, it is clearly beneficial to store the lower part instead of the upper part at the central server so that the reserved backbone WAN bandwidth can be more efficiently utilized.

disk bandwidth consumed by each video stream may be much less than D_i due to statistical multiplexing gains. The potential statistical multiplexing gain can be significant because the staged video (the upper part of video i) at the proxy is generally bursty. Let $\tilde{D}_i$ denote the effective disk bandwidth consumed in this case. Then the bandwidth reduction ratio becomes $R_i = \frac{P_i - T_i}{\tilde{D}_i}$.

3.2 Video Staging with Smoothing

If clients have extra buffering capabilities, the video smoothing [21, 5, 19, 20] can be incorporated into the design of video staging methods to further reduce the backbone WAN bandwidth requirement. For simplicity, we assume that all clients on the same LAN have a buffer of size B for smoothing (when the client smoothing buffer sizes differ, B can be taken to be the smallest one). Given this client buffer, there are two basic approaches: we can either perform video smoothing first, and then select a cut-off rate (this approach is referred to as *cut-off after smoothing* (CAS)); or select a cut-off rate first, and then perform smoothing on either part of the video or both parts (this approach is referred to as *cut-off before smoothing* (CBS)). We describe these two approaches below, and assume that the optimal video smoothing algorithm developed in [21] is used for video smoothing.

3.2.1 Cut-off After Smoothing

Consider video i with N_i frames and a sequence of frames with size s_i^j, $j = 1, \ldots, N_i$. For a buffer size B, the optimal smoothing algorithm [21] generates the "smoothest" transmission schedule consisting of a sequence of transmission sizes $\tilde{s}_i^j$ (referred to as smoothed frames), $j = 1, \ldots, N_i$. Let $\tilde{P}_i = (\max_{1 \le j \le N_i} \tilde{s}_i^j)/F$ be the peak rate of this smoothed transmission schedule. As in Section 3.1, we choose a cut-off rate C_i, where $0 \le C_i \le \tilde{P}_i * F$, and divide the smoothed transmits schedule into two parts: the lower part consists of a sequence of partial smoothed frames with size $\tilde{s}_i^{j,l} = \tilde{s}_i^j - (\tilde{s}_i^j - C_i)^+$, $j = 1, \ldots, N_i$; and the upper part consists of a sequence of partial smoothed frames with size $\tilde{s}_i^{j,u} = (\tilde{s}_i^j - C_i)^+$, $j = 1, \ldots, N_i$. The upper part will be duplicated and staged at the proxy server whereas the lower part will remain stored at the central server. Hence, during the playback of video i, only $T_i = C_i/F$ amount of backbone WAN bandwidth is reserved for the transmission of the lower part of video i across the backbone WAN, while $\tilde{D}_i = (\max_{1 \le j \le N_i} \tilde{s}_i^{j,u})/F$ amount of disk bandwidth is required in the worst case to transfer the upper part from the proxy server to a user on the LAN. The total disk storage space consumed in the proxy server is $\sum_{j=1}^{N_i} \tilde{s}_i^{j,u}$.

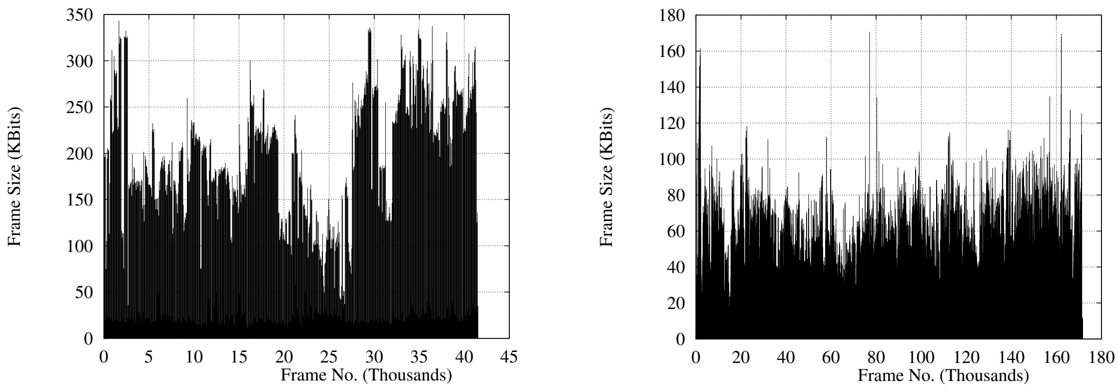
3.2.2 Cut-off Before Smoothing

As in Section 3.1, let $P_i = \max_{1 \le j \le N_i}, s_i^j$ is the peak rate of video i, which has N_i frames and a sequence of frames with size s_i^j, $j = 1, \ldots, N_i$. Under the cut-off before smoothing approach, we first choose a cut-off rate C_i, where $0 \le C_i \le P_i * F$, and divide the video into two parts: the lower part consists of a sequence of partial frames with size $s_i^{j,l} = s_i^j - (s_i^j - C_i)^+$, $j = 1, \ldots, N_i$, and the upper part consists of a sequence of partial frames with size $s_i^{j,u} = (s_i^j - C_i)^+$, $j = 1, \ldots, N_i$. As before, the lower part will remain stored at the central video server while the upper part will be duplicated and staged at the proxy server. There are three possible ways to apply the optimal smoothing algorithm after the cut-off: we can use the client buffer to smooth either the lower part or the upper part or both parts to reduce the rate variability in transmitting these parts.

If considerable rate variability exists in the lower part of video i, using the client buffer to smooth the lower part will generate a smoother transmission schedule, thus reducing the backbone WAN bandwidth requirement that must be reserved across the backbone WAN. Formally, denote this smoother transmission schedule by $\tilde{s}_i^{j,l}$, $j = 1, \ldots, N_i$. Then the reserved backbone WAN bandwidth is $T_i = \tilde{P}_i = (\max_{1 \le j \le N_i} \tilde{s}_i^{j,l})/F$ which is likely to be smaller than C_i/F, the backbone WAN bandwidth that must be reserved if the lower part is not smoothed. We will refer to this video staging method as *smoothing on the lower part* (SOLP).

In contrast, using the client buffer to smooth the upper part of video i will reduce the burstiness of the video data staged at the proxy server, thereby reducing the disk bandwidth required to transfer the video data from the proxy to clients. We shall refer to this video staging method as *smoothing on the upper part* (SOUP). This method may be beneficial when the upper part of the video is very bursty whereas the lower part is almost constant-bit-rate (e.g., when the cut-off rate C_i is fairly small).

We can also smooth both the upper part and lower part of video i by appropriately partitioning the client buffer into two separate buffers. This method shall be referred to as *smoothing on the upper and lower parts* (SOULP). There are many possible ways to partition the buffer. As an heuristic approach, we partition the buffer according to the ratio of the cut-off rate C_i to the peak rate P_i, namely, $B_l = B * \frac{C_i}{P_i * F}$ amount of the client buffer is used to smooth the lower part of the video, and $B_u = B * (1 - \frac{C_i}{P_i * F})$ amount of the client buffer is used to smooth the upper part of the video. Using this method, both the reserved backbone WAN bandwidth and the disk bandwidth required at the proxy server may be reduced. However, the amount of reduction will depend on both the cut-off rate C_i and the video characteristics.



(a) *Wizard of Oz* (b) *Star Wars*

Figure 5: Video traces of *Wizard of Oz* and *Star Wars*

3.3 Empirical Evaluation

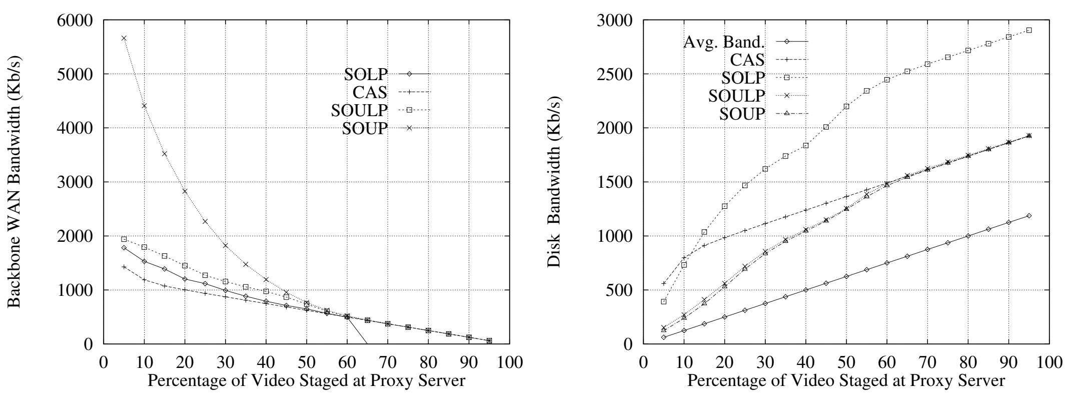
In this section, we empirically evaluate the video staging methods presented in Section 3.2. The evaluation is carried out based on simulation using MPEG-1 traces. Two MPEG-1 video traces, *Star Wars* and *Wizard of Oz*, used in this simulation study are shown in Figure 5. The video *Wizard of Oz* has a total of 41762 frames and is approximately 23 minutes long, while the video *Star Wars* has a total of 174055 frames and is approximately 96 minutes long. The frame rate is 24 frames per section for both videos.

In our simulation, the disk system at the proxy server is modeled based on the Seagate Elite-9 disk, the relevant parameters of which is listed in Table 1. We assume that the proxy server employs a simple two-buffer scheme for delivering staged video data of a video stream to a user on the LAN: during each round, one buffer is used to retrieve a block of video data from the disk system while the other is used to hold the previously retrieved video data block that is currently being transferred to the LAN. In the next round, the role of the buffers are swapped. The effective disk bandwidth depends on the block size used in disk retrieval. Based on the experimental study in [29], we choose a block size of 200KB which yields an effective bandwidth of approximately 5MB/s. From Table 1, we observe that the storage capacity of an Elite-9 disk is about 9GB.

Capacity (GB)	9.09
Rotational Speed (RPM)	5400
Average Rotational Latency (ms)	5.56
Average Seek Time (ms)	11.5
Instant Transfer Rate (MB/s)	5.5-8.125

Table 1: **Elite-9 Disk parameters**

The purpose of our empirical study is to evaluate the effectiveness of using the disk bandwidth at the proxy server to reduce the backbone WAN bandwidth requirement. We first consider the case where only a single video stream is accessed, and thus there is no statistical multiplexing gain in disk retrieval. Assume that clients have no extra buffering capability for smoothing the video transmission. It is clear that for each unit of disk bandwidth consumed at the proxy server, we can reduce one unit of the backbone WAN bandwidth required by staging the appropriate amount of video data at the proxy server. Thus the bandwidth reduction ratio R_i (as defined in Section 3.1) is one. When clients have extra buffers for smoothing, this observation still holds if the *cut-off after smoothing* (CAS) staging method is used. However, this will not be the case when the *cut-off before smoothing* approach is used. The focus of our empirical evaluation is thus on the comparison of the four video staging methods with smoothing: the cut-off after smoothing (CAS) method and the three methods based on the cut-off before smoothing approach — *smoothing on the upper part* (SOUP), *smoothing on the lower part* (SOLP), and *smoothing on the upper and lower parts* (SOULP). In the simulation study below, we assume that *all clients have a total smoothing buffer of size 1MB*.

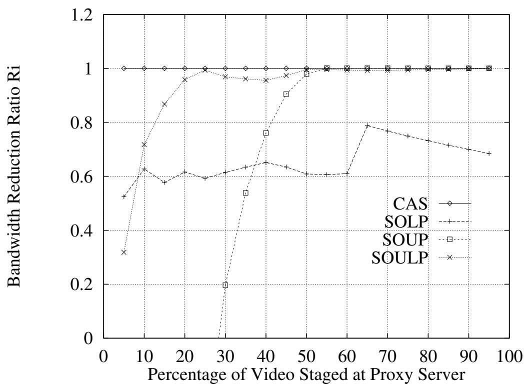


(a) Backbone WAN bandwidth requirement (b) Proxy server disk bandwidth requirement

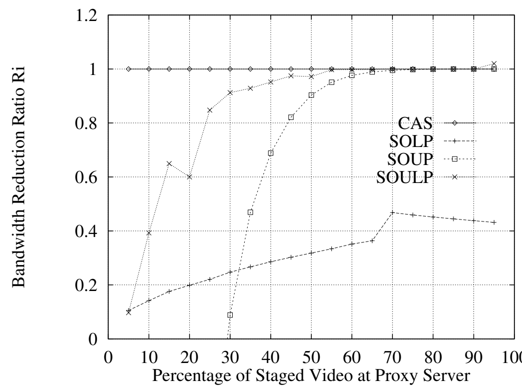
Figure 6: Resource Requirements for a single stream of *Wizard of Oz*

In order to make fair comparison, we choose the cut-off rate for the four methods in such a manner that the percentage of video data staged at the proxy server is kept the same. For a *single* stream of *Wizard of Oz*, the backbone WAN bandwidth requirement under the four methods is plotted as a function of the percentage of video data staged at the proxy server in Figure 6 (a). The corresponding disk bandwidth requirement at the proxy server is plotted in Figure 6 (b). The bandwidth reduction ratio R_i plotted as a function of the percentage of video data staged at the proxy server is shown in Figure 7 (a). We observe that all three cut-off-before-smoothing methods have a bandwidth reduction ration R_i not exceeding 1. The SOULP method outperforms both SOLP and SOUP methods because the latter two have either very high disk bandwidth requirement or backbone WAN bandwidth requirement.

As more video data is stored at the proxy server, SOUP becomes more effective and its performance approaches to that of SOULP and CAS. This is due to the fact that the lower part of the video data becomes an essentially constant-bit-rate stream as the the cut-off rate becomes smaller (thus more video data is staged at the proxy). For the same reason, as more video data is staged at the proxy server, smoothing on the lower part of the video data has less effect. Consequently, the performance of SOLP does not have any significant improvement. As shown in Figure 7 (b), the same observation applies to the result obtained using a single stream of *Star Wars* trace.



(a) *Wizard of Oz*



(b) *Star Wars*

Figure 7: Ratio of backbone WAN bandwidth reduction to proxy server disk bandwidth: single stream

When a large number of users on a LAN play back a video *randomly* (in other words, multiple streams of a single video are played back randomly), the effect of statistical multiplexing gains in reducing the disk bandwidth requirement $\tilde{D}_i$ at the proxy may be significant. As a result, the effective per-stream disk bandwidth requirement $\tilde{D}_i$ may be much smaller than that in the worst-case, which is given by $D_i = (\max_{1 \le j \le N_i} \tilde{s}_i^{j,u})/F$. Here $\tilde{s}_i^{j,u}$ represents the video data retrieved from the disk system of the proxy at the jth frame period, $j = 1, \ldots, N_i$. In theory, as the number of random accesses increases, $\tilde{D}_i$ may approach the average disk bandwidth requirement given by $\frac{\sum_{j=1}^{N_i} \tilde{s}_i^{j,u}}{N_i * F}$. Figure 8 shows the effective per-stream disk bandwidth requirement at the proxy server when 100 streams of *Wizard of Oz* are played back randomly. Note that since we assume that a CBR network service is used for video transmission across the backbone WAN, the per-stream backbone WAN bandwidth requirement will be the same (as shown in Figure 6 (a)) regardless of the number of random accesses to the video.

Therefore, when a video is accessed multiple times independent, we compute the bandwidth reduction ratio R_i using $\tilde{D}_i$ instead of the worst-case D_i. Figure 9 (a) and (b) show, respectively, the bandwidth reduction ratio R_i under the four video staging methods when 10 and 100 streams of *Wizard*

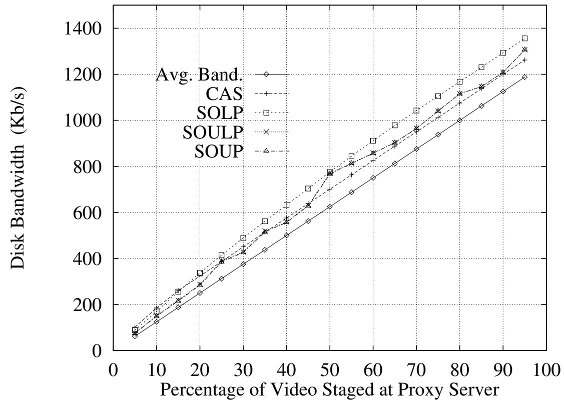
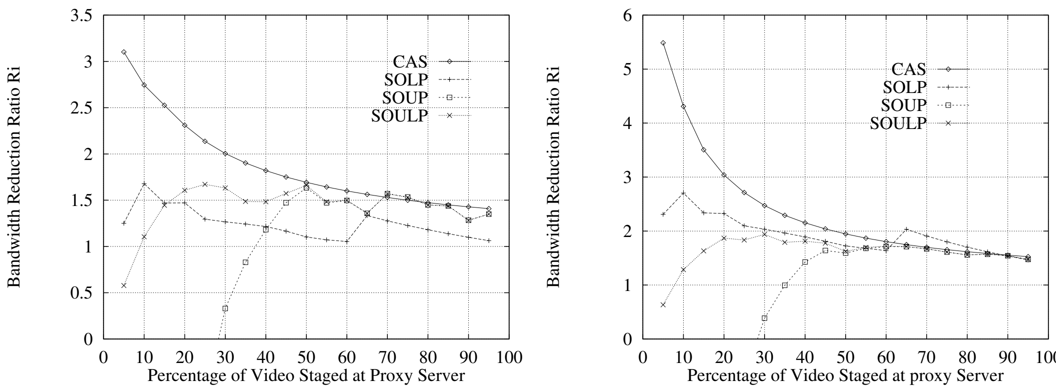


Figure 8: Effective per-stream disk bandwidth requirement when 100 streams of *Wizard of Oz* are statistically multiplexed

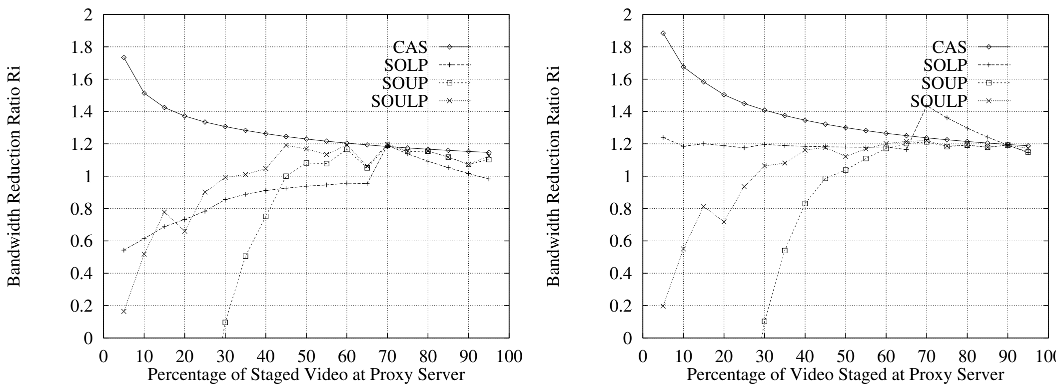
of Oz are statistically multiplexed. The results from the same simulation conducted using the *Star Wars* trace are shown in Figure 10. These figures show that the CAS method outperforms the three cut-off before smoothing methods most of the time and their performance tends to converge when a large amount of video data is staged at the proxy server.



(a) 10 streams (b) 100 streams

Figure 9: Bandwidth reduction ratio for multiple streams of *Wizard of Oz*

Before we leave this section, however, we would like to point out one short-coming of the CAS method. For simplicity, we have assumed that all clients have a smoothing buffer of the same size. When client smoothing buffer sizes differ, the CAS method has to use the smallest buffer size to smooth a video stream before selecting a cut-off rate and then stage the corresponding upper part at a proxy server. On the other hand, the cut-off before smoothing approach can better accommodate this heterogeneity in client buffer capabilities by performing smoothing at the time of video retrieval and transfer. Due to space limitation, the results from our study investigating this issue will not be included here.



(a) 10 streams (b) 100 streams

Figure 10: Bandwidth reduction ratio for multiple streams of *Star Wars*

4 Video Staging: Multiple Video Case

In the previous section, we have studied the effectiveness of video staging in reducing the backbone WAN bandwidth for a given video. One important problem remaining to be addressed is how to choose the cut-off rate so as to determine the amount of video data to be staged at a proxy server. In this section, we will study this problem in the context of multiple video staging with a given video access profile for a LAN. We first formulate the problem formally and then develop two heuristic algorithms. Lastly the two algorithms are empirically evaluated. Comparison with the approach where no proxy server is used is also made.

4.1 Problem Formulation

Given a set of videos, the expected number of accesses to these videos may vary dramaticallly depending on their popularity. User access pattern is thus an important factor that must be taken into account when determining the amount of video data to be staged at a proxy server. Zipf-like distributions have been commonly used to characterize user access pattern in a video-on-demand environment [3, 26]. In Zipf distribution, the skewness in the popularity of a set of M videos is represented by a skewness parameter α, $0 < \alpha \leq 1$: the probability that video i is accessed is given by $\frac{f_i}{\sum_{j=1}^{M} f_j}$, where $f_i = \frac{1}{i^{1-\alpha}}$, for $1 \leq i \leq M$. When $\alpha = 1$, each video is equally accessed with probability $\frac{1}{M}$. In general, the smaller the skewness factor is , the skewer the distribution is[4].

[4]For example, Zipf distribution with a skewness parameter of 0.271 has been used in the literature [3, 26]. The value is chosen because it closely matches a published video store rental distribution [14] of 92 videos.

In formulating the multiple video staging design problem, we assume that the user access pattern at a LAN is characterized by a known Zipf distribution with skewness parameter α. This information can be gathered, for example, from the past user access patterns either on a daily, weekly or monthly basis or on an even longer time period. Suppose that the number of videos stored in the central video server is M, and the total number of expected concurrent accesses to these videos is A. The M videos are numbered in the descending order of their popularity. (Thus video 1 is most popular one, i.e., it has the largest expected number of accesses. Videos having the same expected number of accesses are ordered arbitrarily.) Therefore, for $1 \le i \le M$, the expected number of accesses to video i is $a_i = A * \frac{f_i}{\sum_{j=1}^{M} f_j}$. By abuse of notation, we define $A = (a_1, a_2, \ldots, a_M)$, and refer to this vector as the *video access profile*.

Suppose that the disk subsystem of the proxy server has a total amount of disk bandwidth, B_{dsk}^{proxy}, and a total amount of storage space S_{disk}^{proxy}. Consider video i, $1 \le i \le M$, and fix a video staging method, say, the CAS method. For any given cut-off rate C_i, let T_i ($= \frac{C_i}{F}$) be the backbone WAN bandwidth required to transmit the portion of the video stored at the central video server. Similarly, let D_i be the effective disk bandwidth required to transfer the portion of the video staged at the proxy server to a client, assuming that there are a_i number of random accesses to the video. The corresponding disk storage space used is S_i. Thus for video i, $1 \le i \le M$, choosing the cut-off rate C_i will reduce the backbone WAN bandwidth requirement by $P_i - T_i$, where P_i is the backbone WAN bandwidth requirement if no video data is staged at the proxy server. However, this reduction is achieved by consuming D_i amount of disk bandwidth and S_i amount of disk storage space at the proxy server. Clearly, the objective here is to choose C_i, $1 \le i \le M$, in such a manner that the *total* expected backbone WAN bandwidth requirement is minimized (or equivalently, the total reduction in the backbone WAN bandwidth requirement is maximized) subject to the constraint that the total disk bandwidth and storage capacity requirements can be sustained by the proxy server. Formally, we have

Multiple Video Staging Design Problem

Given a video access profile $A = (a_1, a_2, ..., a_M)$ and a disk system at the proxy server with a total amount of disk bandwidth, B_{disk}^{proxy}, and a total amount of disk storage capacity, S_{disk}^{proxy}, determine the cut-off rate C_i for each video i, $1 \le i \le M$, such that the following total reduction in the backbone WAN bandwidth is maximized

$$\sum_{i=1}^{M} a_i * (P_i - T_i)$$

Subject to the disk bandwidth constraint

$$\sum_{i=1}^{M} a_i * D_i \leq B_{disk}^{proxy}$$

and the disk storage constraint

$$\sum_{i=1}^{M} S_i \leq B_{storage}^{proxy}.$$

Due to the difficulty in computing the parameters involved, unfortunately, this optimization problem does not have a simple solution. Although an exhaustive search is possible, the computational complexity of this search is prohibitively high. As an alternative, we present two heuristic algorithms. We remark that given the disk system today (for example, Seagate Elite-9), the performance of the proxy server is bounded more by the disk bandwidth constraint than the storage capacity constraint. Therefore, in our heuristic algorithms, we will focus only on the disk bandwidth constraint. This is justified by the empirical evaluation conducted in Section 4.4.

4.2 Staging Hot Video Only (SHVO)

The first heuristic algorithm, referred to as the *staging hot video only* (SHVO) algorithm, regards the user access pattern as the most important factor in determining which video to stage at the proxy server. Intuitively, if a video is “hot”, i.e., it has a large number of concurrent accesses, then staging this video entirely at the proxy server is likely to yield significant reduction in the backbone WAN bandwidth. This is because no backbone WAN bandwidth is required for delivering the video. Formally, let $A = (a_1, a_2, ..., a_M)$ be the video access profile where we have $a_1 \geq a_2 \geq ... \geq a_M$. For each i, let D_i be the effective per-stream disk bandwidth required to transfer video i from the proxy server to a user, given that there are a_i concurrent random accesses to the video. Then the number of hot videos, k, can be staged at the proxy server is determined by the following constraint:

$$\sum_{i=1}^{k} a_i * D_i \leq B_{disk}^{proxy} < \sum_{i=1}^{k+1} a_i * D_i.$$

Under this heuristic algorithm, either a video is entirely staged at the proxy server or not at all.

Input: M videos and its associated video access profile $A = (a_1, a_2, ..., a_M)$,
the total disk bandwidth at the proxy server, B_{disk}^{proxy},
and the incremental cut-off percentage δP

Output: the cut-off percentage CP_i for each video.

Other Parameters: the disk bandwidth requirement $D_i(CP_i)$ for a given CP_i
and the bandwidth reduction ratio $R_i(CP_i)$ for a given CP_i

$LBRRF(A, B_{disk}^{proxy}, \delta P)$
{
  Initialize CP_i to 0, $i = 1, \ldots, M$;
  Initialize B_{disk}^{avail} to B_{disk}^{proxy};
  `while` $(B_{disk}^{avail} > 0)$ {
    Find video i with the maximum $R_i(CP_i + \delta P)$;
    $B_{disk}^{avail} := B_{disk}^{avail} - (D_i(CP_i + \delta P) - D_i(CP_i))$;
    $CP_i := CP_i + \delta P$;
    }
  `return` the cut-off percentage CP_i, $i = 1..M$;
}

Figure 11: The LBRRF Algorithm

4.3 Largest Bandwidth Reduction Ratio First (LBRRF)

The second heuristic algorithm, referred to as the *largest bandwidth reduction ratio first* (LBRRF) algorithm, considers the effectiveness of video staging in reducing the backbone WAN bandwidth as the most important factor in determining which video and what percentage of it to be staged at the proxy server. This effectiveness is measured in the bandwidth reduction ratio R_i, as defined in Section 3. We use the cut-off after smoothing (CAS) method as the video staging method for each video. The basic idea behind the LBRRF approach is explained as follows.

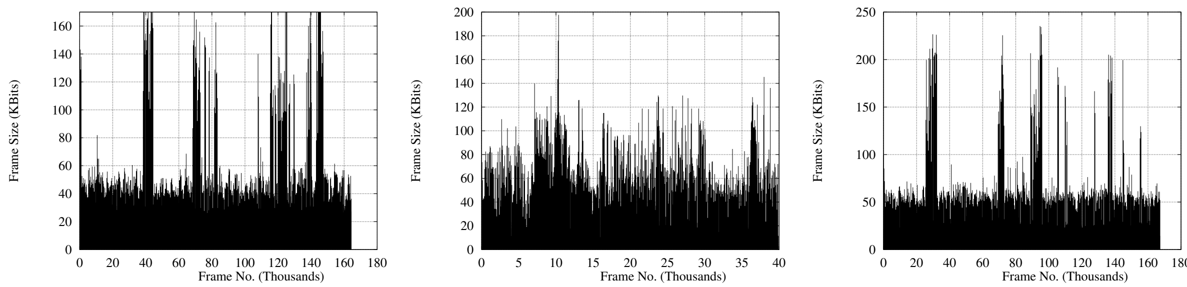
For a video i, staging a portion of the video at the proxy server consumes certain fraction of the total disk bandwidth at the proxy server. In some sense, this fraction of the total disk bandwidth is allocated to video i. Clearly, the video with largest R_i should be favored when allocating the disk bandwidth at the proxy server, provided that everything else is equal. Formally, for each i, $1 \leq i \leq M$, suppose CP_i percentage of video i is staged at the proxy server. Let $D_i(CP_i)$ denote the corresponding effective per-stream disk bandwidth requirement, given that there are a_i concurrent random accesses to the video. Let $R_i(CP_i)$ be the resulting bandwidth reduction ratio. For the given CP_i, the video with largest R_i is chosen, and CP_i percentage of its video data as determined by the CAS method will be staged at the proxy server.

The LBRRF algorithm is presented in Figure 4.3. The algorithm starts with $CP_i = 0$ for all videos and sets the currently available disk bandwidth B_{disk}^{avail} to B_{disk}^{proxy}, the total disk bandwidth at the proxy server (Lines 3-4). It then iterates by incrementing CP_i by δP amount at each step (Lines 5-9). The δP is used to control the complexity of the algorithm. At each step, the effective disk bandwidth requirement $D_i(CP_i + \delta P)$ and the corresponding bandwidth reduction ratio $R_i(CP_i + \delta P)$ are computed for each video i. The video with largest R_i is selected (Line 6) (with the appropriate amount of disk bandwidth allocated to it), and the available disk bandwidth B_{disk}^{avail} is adjusted accordingly (Line 7). The algorithm continues until there is no disk bandwidth available at the proxy server. The complexity of the algorithm is $O(M * \frac{B_{disk}^{avail}}{F(\delta P)})$, where $F(\delta P)$ is the smallest amount of disk bandwidth allocated as determined by δP at a single step.

4.4 Empirical Evaluation

In this section, we evaluate the performance of the two heuristic approaches, SHVO and LBRRF based on simulation. The disk system model is again based on the Seagate Elite-9 disk, the parameters of which is listed in Table 3.3. We assume that there are a total of 50 videos (i.e., $M = 50$), and there are a total of 500 concurrent accesses (i.e., $A = 500$). Since we do not have 50 different video traces, we generate a Zipf-distribution-based video access profile with 50 videos and 500 concurrent accesses by randomly choosing 50 videos from the following 5 MPEG-1 video traces: *Wizard of Oz*, *Star Wars*, *CNN*, *MTV*, and *Princess Bride*. The traces of *Wizard of Oz* and *Star Wars* have been shown in Figure 5. The traces of the other three videos are shown in Figure 12. The peak rates of these 5 videos range from 5.5Mb/s to 10.3Mb/s, and the average rates range from 0.47Mb/s to 1.2Mb/s. Except for the last set of experiments reported here (Figure 17), all other sets of experiments use a skewness parameter $\alpha = 0.3$. In all the experiments, whenever clients have extra smoothing buffer, the CAS video staging method is used.

In the first set of experiments, we assume that clients have no extra buffer available for smoothing. In Figure 13 (a), the backbone WAN bandwidth requirement under both the SHVO algorithm and the LBRRF algorithm is plotted as a function of the number of Elite-9 disks available at the proxy server. As the total disk bandwidth at the proxy server increases, the backbone WAN bandwidth requirement is significantly reduced. We also observe that the LBRRF algorithm performs better than the SHVO algorithm. This is expected because the LBRRF algorithm tends to utilize the disk bandwidth more efficiently. When there are about 18 disks, all videos can be staged at the proxy server, thus no back-

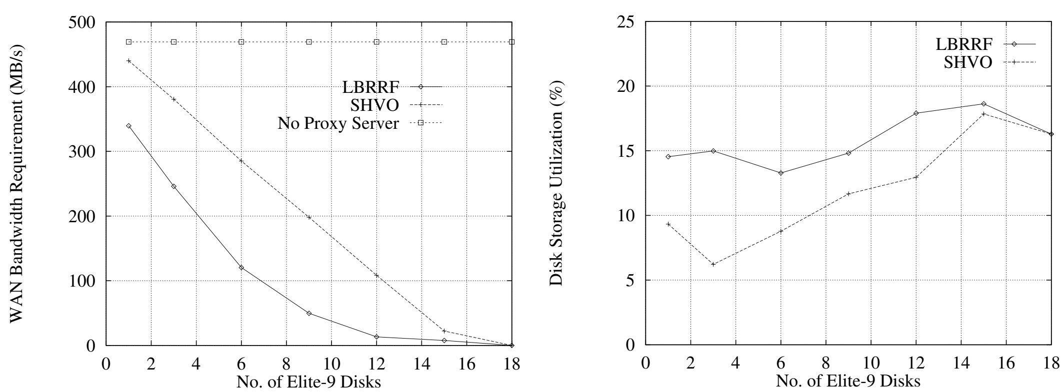


(a) *CNN* (b) *MTV* (c) *Princess Bride*

Figure 12: Three additional video traces used in simulation.

bone WAN bandwidth is required. In order to demonstrate the effectiveness of our proxy-server-based approach, we also plot the backbone WAN bandwidth requirement when no proxy server is used. The superiority of the proxy-server-based approach in reducing the backbone WAN bandwidth is evident even when there are only a couple of disks available at the proxy server.

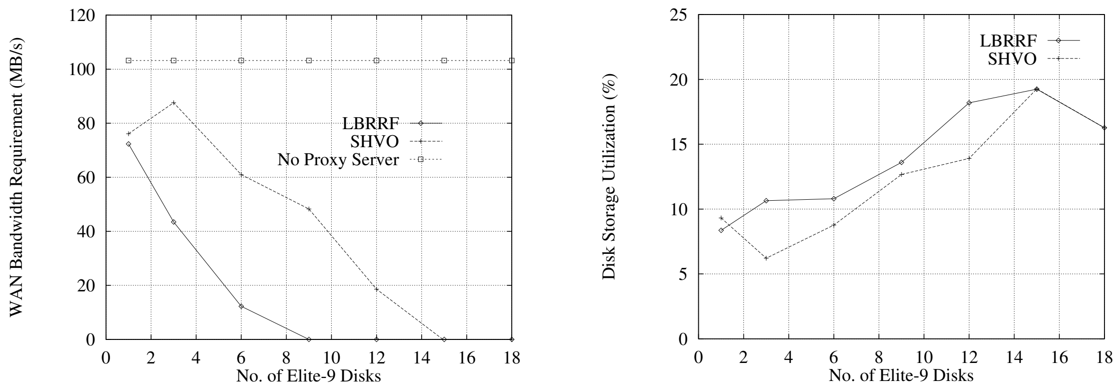
Figure 13 (b) shows the storage utilization of the two algorithms. Less than 20% of the total disk storage space available at the proxy server is used. This confirms our observation that, given the current disk system, the disk bandwidth at the proxy server is more likely to be the bottleneck than the disk storage capacity. We observe that the LBRRF algorithm consumes more disk storage space than the SHVO algorithm. The resaon is as follows. Although LBRRF algorithm only stage a portion of a video, it stage more videos than SHVO algorithm. Further more, by only storing the lower portion of the videos, LBRRF algorithm can more effectively utilize the disk bandwidth of the proxy server and thus can afford to store more video data than SHVO algorithm.



(a) Backbone WAN bandwidth reduction (b) Proxy server disk storage utilization

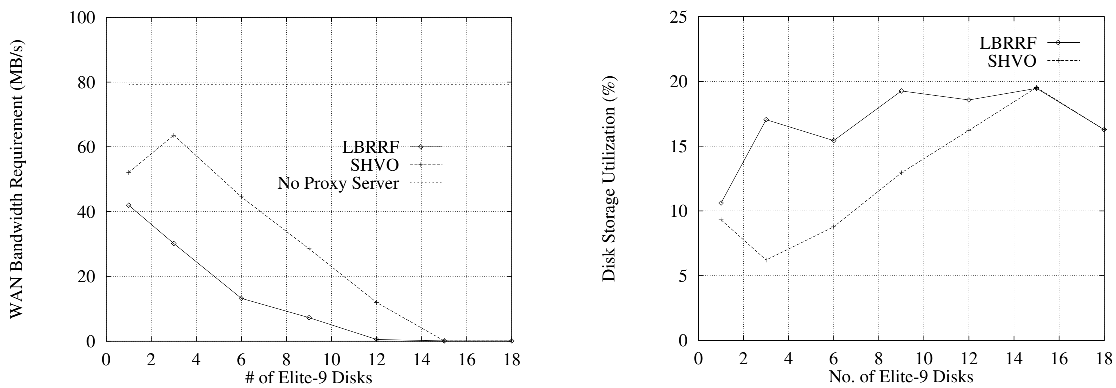
Figure 13: Impact of proxy server disk resources: clients have no smoothing buffer, $\alpha = 0.3$.

In the second set of experiments, we assume that clients have a smoothing buffer of size 64KB.



(a) Backbone WAN bandwidth reduction (b) Proxy server disk storage utilization

Figure 14: Impact of proxy server disk resources: clients have 64KB smoothing buffer, $\alpha = 0.3$.

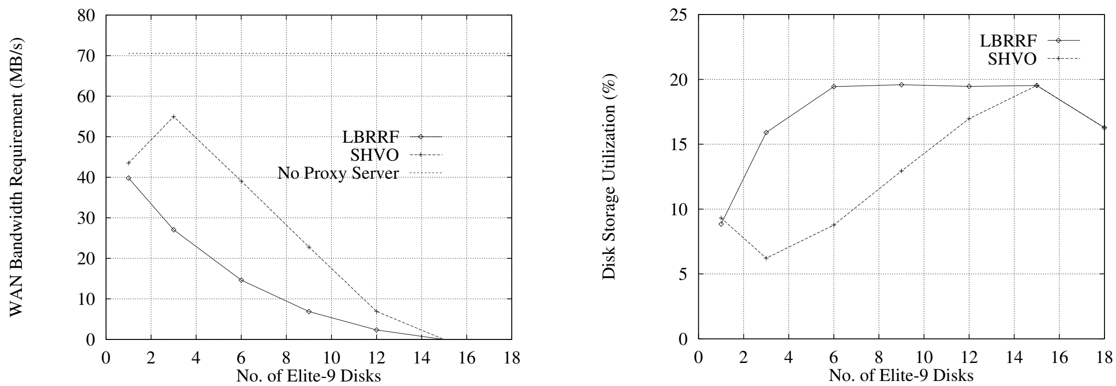


(a) Backbone WAN bandwidth reduction (b) Proxy server disk storage utilization

Figure 15: Impact of proxy server disk resources: clients have 1MB smoothing buffer, $\alpha = 0.3$.

In Figure 14 (a), The backbone WAN bandwidth requirement under both the SHVO algorithm and the LBRRF algorithm is plotted as a function of the number of Elite-9 disks available at the proxy server. In this case, the LBRRF algorithm also outperforms the SHVO algorithm. In particular, when there are more than 9 disks, the total backbone WAN bandwidth requirement under the LBRRF algorithm is close to zero whereas the SHVO algorithm requires 15 disks to reduce the backbone WAN bandwidth requirement to zero. In order to demonstrate the superiority of the proxy-server-based approach in reducing the backbone WAN bandwidth, we also plot the backbone WAN bandwidth requirement when no proxy server is used but video smoothing is performed with the given client buffer. As shown in Figure 14 (b), the storage utilization of the two algorithms in this case is still less than 20% of the total disk storage space available at the proxy server.

The results from two more sets of experiments are shown in Figure 15 and Figure 16 where clients



(a) Backbone WAN bandwidth reduction (b) Proxy server disk storage utilization

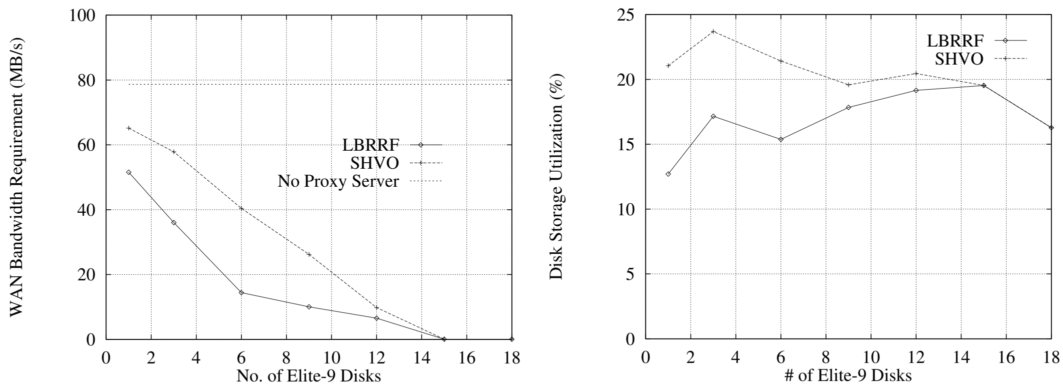
Figure 16: Impact of proxy server disk resources: clients have 8MB smoothing buffer, $\alpha = 0.3$.

are assumed to have a smoothing buffer of size 1MB and 8MB respectively. The same observation applies to these two cases, although the difference in the performance of the two heuristic algorithms appears to get smaller as the size of client smoothing buffer increases. In these cases, the proxy-server-based approach can still achieve significant reduction in the backbone WAN bandwidth even when the number of disks at the proxy server is small.

In the last set of experiments, we investigate the impact of the skewness parameters α on the performance of the two heuristic algorithms. We change α from 0.3 to 0.9, thus the Zipf distribution becomes less skewer. In other words, the accesses to the videos are more evenly distributed. Since the disk storage space is not the bottleneck, the "flatter" video access profile would enable more videos to be staged in the proxy server. As a result, we would expect that the performance of the SHVO algorithm should somewhat be improved. This is confirmed by our experiments, as the results in Figure 17 (a) show. It is also interesting to notice that Figure 17 (b) shows that the SHVO algorithm has a higher disk storage utilization than the LBRRF algorithm in this case.

5 Related Work

The problem of end-to-end real-time transport of stored video over a network has been studied in many contexts [1, 2, 30, 32, 10, 28, 27, 29, 25, 5, 11, 12, 13, 16, 19, 20, 24, 33]. In particular, a number of researchers have considered using video smoothing techniques to reduce the variability in transmitting stored video from a server to a client across a high-speed network [6, 15, 18, 19, 21, 33] by take advantage of the client buffer capabilities. These video smoothing techniques (implicitly) assume a homogeneous underlying network environment, and thus do not address the issue of heterogeneity in



(a) Backbone WAN bandwidth reduction (b) Proxy server disk storage utilization

Figure 17: Impact of proxy server disk resources: clients have 1MB smoothing buffer, $\alpha = 0.9$.

the underlying networking environment. Our proxy-server-based, network-conscious approach, on the other hand, explicitly takes the underlying heterogeneous networking environments into consideration. In such a problem setting, the most critical aspect of the end-to-end video delivery across heterogeneous networking environments. We present an effective technique — video staging using proxy servers — to address the problem how to reduce the backbone WAN bandwidth requirement. Note that the backbone WAN bandwidth requirement can also be reduced by either directly applying the video smoothing techniques with large client buffers or prefetching a video stream into a client's disk storage system. However, to achieve the same effect as our approach. considerable cost and overhead must be added to the client system. For example, a 23 MB client buffer is needed to transmit the VBR compressed MPEG-1 *star war* video in a constant-bit-rate, real-time fashion. Prefetching part or a whole video stream into a client disk would require considerable disk space and incur large start-up latency overhead. Moreover, the prefetched data is only available to a single user. In contrast, our proxy-server-based, network-conscious approach provides a much more cost-effective and scalable way to deliver video across a wide-area network to a large number of users situated in a single local-area network. This is because the considerably larger disk bandwidth and storage space available at the proxy server are shared by all the users. In addition, the video smoothing techniques are also incorporated into our approach to exploit the client buffering and storage capabilities.

Proxy (web) servers [7] together with web caching techniques (see, e.g., [17] and references therein) have been widely used in web-based data applications to reduce data transfer latency across the Internet by caching frequently-accessed data locally. Similar notions such as Internet object cache [4, 31] and host proximity service [8] have also been proposed with a similar objective. Due to the real-time playback nature of video delivery and the large amount of disk bandwidth and storage space required at the proxy server, we believe that applying these caching techniques to the problem of end-to-end delivery is unlikely to yield considerable improvement in the system performance. On the contrary, it may even degrade the system performance because of the potential large overhead involved in the real-time transfer of video data. In contrast, the video staging technique developed in this paper determines what video data to be placed at a proxy server by taking both video characteristics and a video access profile of a LAN into consideration. The objective is to reduce the total expected backbone WAN bandwidth requirement when delivering video to a large number of users at a given LAN.

6 Conclusion

In this paper, we have presented a novel *network-conscious* approach to the problem of end-to-end video delivery over wide-area networks using proxy servers situated between local-area networks (LANs) and a backbone wide-area network (WAN). A major objective of our approach is to reduce the backbone WAN bandwidth requirement. Towards this end, we have developed a novel and effective video delivery technique called *video staging* through intelligently utilizing the disk bandwidth and storage space available at the proxy servers. We have designed various video staging methods and evaluated their effectiveness in trading the disk bandwidth of a proxy server for the backbone WAN bandwidth. We also developed heuristic algorithms for designing a multiple video staging scheme in a proxy server with a given video access profile for the associated LAN. Our results demonstrate that the proposed proxy-server-based, network-conscious approach provides a cost-effective and scalable solution to the problem of the end-to-end video delivery over wide-area networks.

Our study is only an initial study of the proposed proxy-server-based, network conscious approach. There are still many issues, both theoretical and practical, that must be addressed. The synchronization and signaling issues mentioned in Section 2 are two examples. Investigation of these issues will be the topic of our future research.

References

[1] BERSON, S., GHANDEHARIZADEH, S., MUNTZ, R., AND JU, X. Staggered Striping in Multimedia Information Systems. *Proceedings of the ACM SIGMOD International Conference on Management of Data* (1994).

[2] DAN, A., KIENZLE, M. G., AND SITARAM, D. Dynamic Segment Replication Policy for Load-Balancing in Video-on-Demand Servers. *ACM Multimedia Systems, Vol. 3, No. 3,* (1995), 93–103.

[3] DAN, A., SITARAM, D., AND SHAHABUDDIN, P. Scheduling Policies for an On-Demand Video Server with Batching. *Proceedings of ACM Multimedia'94* (October 1994), 15–21.

[4] DANZIG, P. B., HALL, R. S., AND SCHWARTZ, M. F. A case for caching file objects inside internetworks. In *Proc. ACM SIGCOMM* (Aug. 1993).

[5] FENG, W., AND SECHREST, S. Critical bandwidth allocation for delivery of compressed video. *Computer Communications* (Special issue on Systems Support for Multimedia) (Oct. 1995).

[6] FENG, W., AND SECHREST, S. Smoothing and buffering for delivery of prerecorded compressed video. In *IS&T/SPIE Multimedia Computing and Networking* (San Jose, CA, Feb. 1995), pp. 234–232.

[7] FIELDING, R., GETTYS, J., MOGUL, J. C., FRYSTYK, H., AND BERNERS-LEE, T. Hypertext transfer protocol – HTTP/1.1. Internet draft `draft-ietf-http-v11-spec-01.txt`, IETF, June 1996.

[8] FRANCIS, P. A call for an internet-wide host proximity service (hops). Manuscript, NTT Software Labs, 1997.

[9] GARRETT, M., AND WILLINGER, W. Analysis, modeling and generation of self-similar VBR video traffic. In *Proc. ACM SIGCOMM* (London, England UK, Aug. 1994), pp. 269–280.

[10] GHANDEHARIZADEH, S., AND KIM, D. On-line Reorganization of Data in Scalable Continuous Media Servers. Tech. Rep. USC-CS-TR96-634, Department of Computer Science, University of Southern California, 1996.

[11] GROSSGLAUSER, M., KESHAV, S., AND TSE, D. RCBR: A simple and efficient service for multiple time-scale traffic. In *Proc. ACM SIGCOMM* (Cambridge, MA, Aug. 1995), pp. 219–230.

[12] KNIGHTLY, E. W., WREGE, D. E., LIEBEHERR, J., AND ZHANG, H. Fundamental limits and tradeoffs of providing deterministic guarantees to VBR video traffic. In *Proc. ACM SIGMETRICS* (Ottawa, Canada, May 1995), pp. 98–107.

[13] LAM, S. S., CHOW, S., AND YAU, D. K. Y. An algorithm for lossless smoothing of MPEG video. In *Proc. ACM SIGCOMM* (London, England, Aug. 1994).

[14] MAGAZINE, V. Video Store Magazine. Tech. rep., December 12 1992.

[15] MCMANUS, J. M., AND ROSS, K. W. Video on demand over ATM: Constant-rate transmission and transport. In *Proc. IEEE INFOCOM* (San Francisco, CA, Mar. 1996).

[16] OTT, T., LAKSHMAN, T. V., AND TABATABAI, A. A scheme for smoothing delay-sensitive traffic offered to ATM networks. In *Proc. IEEE INFOCOM* (Florence, Italy, May 1992), pp. 776–785.

[17] PADMANABHAN, V. N., AND MOGUL, J. C. Using predictive prefetching to improve world wide web latency. *Computer Communication Review 26*, 3 (July 1996).

[18] REIBMAN, A. R., AND BERGER, A. W. On VBR video teleconferencing over ATM networks. In *Proc. IEEE GLOBECOM* (1992), pp. 314–319.

[19] REIBMAN, A. R., AND BERGER, A. W. Traffic descriptors for VBR video teleconferencing over ATM networks. *IEEE/ACM Transactions on Networking 3*, 3 (June 1995), 329–339.

[20] REIBMAN, A. R., AND HASKELL, B. G. Constraints on variable bit-rate video for ATM networks. *IEEE Transactions on Circuits and Systems for Video Technology 2*, 4 (Dec. 1992), 361–372.

[21] SALEHI, J., ZHANG, Z.-L., KUROSE, J., AND TOWSLEY, D. Supporting stored video: Reducing rate variability and end-to-end resource requirements through optimal smoothing. In *ACM International Conference on Measurement and Modeling of Computer Systems (ACM SIGMETRICS)* (Philadelphia, PA, May 1996).

[22] SCHULZRINNE, H., CASNER, S., FREDERICK, R., AND MCCANE, S. RTP: A transport protocol for real-time applications. Request for Comments RFC 1889, Network Working Group, 1994.

[23] SCHULZRINNE, H., RAO, A., AND LANPHIER, R. Real-time streaming protocol (RTSP). Internet draft, IETF, 1997.

[24] SHROFF, N., AND SCHWARTZ, M. Video modeling within networks using deterministic smoothing at the source. In *Proc. IEEE INFOCOM* (1994), pp. 342–349.

[25] TAN, S., CAMPBELL, R., CHEN, Z., LIAO, W., RAILA, D. K., KON, F., AND VALDEZ, M. Adaptation and synchronization in low-bandwidth internet video. In *World Wide Web Consortium Workshop on Real Time Multimedia in the WWW (RTMW '96)* (INRIA Sophia Antipolis, France, Oct. 1996).

[26] TETZLAFF, W., AND FLYNN, R. Disk Striping and Block Replication Algorithms for Video File Servers. *Porceedings of ICMCS'96* (June 1996), 590–597.

[27] TOBAGI, F., PANG, J., BAIRD, R., AND GANG, M. Streaming RAID - a Disk Array Management System for Video Files. *Proceedings of the ACM Multimedia'93* (August 1993), 393–400.

[28] VIN, H., AND RANGAN, V. Admission Control Algorithm for Multimedia on-demand servers. *IEEE Journal on Selected Areas in Communications* (November 1993), 56–69.

[29] WANG, Y., AND DU, D. Weighted Striping in Multimedia Servers. *ICMCS'97* (June 1997).

[30] WANG, Y., LIU, J., DU, D., AND HSIEH, J. Video Allocation over Disk Arrays for Video-on-Demand. *Proceedings of ICMCS'96, Hiroshima, Japan* (June 1996).

[31] WESSELS, D. Internet cache protocol (icp). Internet draft, IETF, Mar. 1997.

[32] WOLF, J., YU, P., AND SHACHNAI, H. DASD Dancing: A Disk Load Balancing Optimization Scheme for Video- on-Demand Computer Systems. Tech. Rep. IBM Research Report RC 19987, IBM T.J. Watson Research Division, 1995.

[33] ZHANG, Z.-L., KUROSE, J., SALEHI, J. D., AND TOWSLEY, D. Smoothing, statistical multiplexing and call admission control for stored video. *IEEE Journal of Selected Areas in Communication, Special Issue on Real-Time Video Services in Multimedia Networks* (Aug. 1997).