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

Abstract

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). 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. We also design several video staging methods and evaluate their effectiveness in reducing the backbone WAN bandwidth requirement. 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.*

1 Introduction

Real-time distribution of stored video over high-speed networks is a crucial component of many emerging multimedia applications. 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. 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.

Since the backbone WAN is normally shared by a large number of institutions and users, it is generally more expensive to deploy additional resources in the backbone WAN than in local area networks. Thus the WAN bandwidth is a much more critical and costly resource than that of campus-wide LANs. On the other hand, it is generally much easier to add more bandwidth to a LAN. Given the emerging gigabit LAN networking technologies such as Gigabit Ethernet, the cost of installing and running a local-area gigabit network becomes increasingly cheaper. Hence, 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.

In this paper, we present a novel *proxy-server-based, network-conscious* approach to the end-to-end 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*, 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*. 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 later. The video data are staged at the proxy server on a fairly long period of time instead of caching in and purged out dynamically.

Our proxy-server-based, network-conscious approach to

*The authors are with Department of Computer Science & Engineering, University of Minnesota, 200 Union Street S.E, Minneapolis, MN 55455. Email: {ywang,zhzhang,du,dsu}@cs.umn.edu. The second author is supported in part by a Grant-in-Aid grant from Univerity of Minnesota.

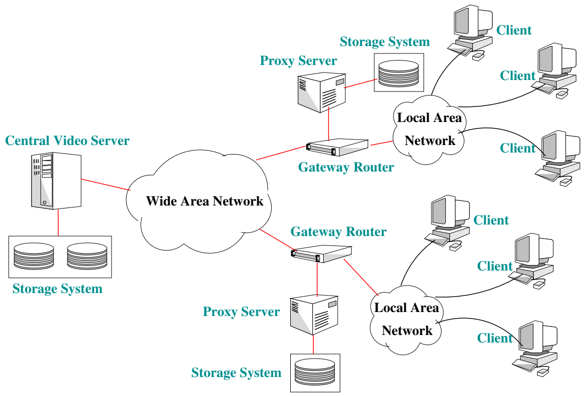


Figure 1: Video delivery over a simple heterogeneous networking environment

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 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.

The problem of end-to-end real-time transport of stored video over a network has been studied in many contexts. 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 (see [2] and reference therein) 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 the underlying networking environment. Our proxy-server-based, network-conscious approach, on the other hand, explicitly takes the underlying heterogeneous networking environment into consideration, and as a result, our solution can be optimized accordingly.

The remainder of our paper is organized as follows. In Section 2, we present video staging techniques for a single video stream and in Section 3, we develop two heuristic video staging schemes for multiple video streams. The paper is concluded in Section 4.

2 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 [2] into the design of video staging methods to achieve further backbone WAN bandwidth reduction when clients have extra buffering capabilities available for smoothing.

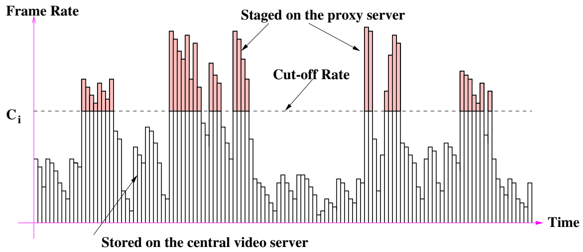


Figure 2: A simple video staging method using a cut-off rate

2.1 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 will be 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 some amount of disk bandwidth as well as storage capacity available at the proxy server for storing and delivering the staged video data. Since the resources (especially the disk bandwidth) at the proxy server are limited, it is thus important to consider the effectiveness of video staging in reducing the backbone WAN bandwidth for a given video.

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 2. 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 (in fact, the whole video is always stored at the central video server). From Figure 2, 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 $\sum_{j=1}^{N_i} s_i^{j,u}$. The bandwidth reduction ratio, R_i, which is defined as the ratio of the backbone WAN bandwidth reduction to the disk bandwidth consumed at the proxy server, is thus given by $R_i = \frac{P_i - T_i}{D_i}$. When there are a large number of concurrent accesses from users on the LAN, the effective disk bandwidth consumed by each video stream may be much less 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 $R_i = \frac{P_i - T_i}{\tilde{D}_i}$.

2.2 Video Staging with Smoothing

If clients have extra buffering capabilities, the video smoothing (e.g. [2]) 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 he optimal video smoothing algorithm developed in [2] is used for video smoothing.

2.2.1 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 [2] 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 2.1, 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 the lower part, $\tilde{s}_i^{j,l} = \tilde{s}_i^j - (\tilde{s}_i^j - C_i)^+$, $j = 1, \ldots, N_i$, and the upper part, $\tilde{s}_i^{j,u} = (\tilde{s}_i^j - C_i)^+$, $j = 1, \ldots, N_i$. 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 $\sum_{j=1}^{N_i} \tilde{s}_i^{j,u}$.

2.2.2 Cut-off Before Smoothing

As in Section 2.1, 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 the lower part, $s_i^{j,l} = s_i^j - (s_i^j - C_i)^+$, $j = 1, \ldots, N_i$, and the upper part, $s_i^{j,u} = (s_i^j - C_i)^+$, $j = 1, \ldots, N_i$. 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

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

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 $T_i = \tilde{P}_i = (\max_{1 \leq j \leq N_i} \tilde{s}_i^{j,l})/F$ 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 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, $B_u = B * \frac{C_i}{P_i * F}$ amount of the client buffer is used to smooth the upper part of the video, and $B_l = B * (1 - \frac{C_i}{P_i * F})$ amount of the client buffer is used to smooth the lower 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.

2.3 Empirical Evaluation

In this section, we empirically evaluate the video staging methods presented in Section 2.2. The evaluation is carried out based on simulation using a MPEG-1 trace of video *Wizard of Oz*. This video trace has a total of 41762 frames and, is approximately 23 minutes long with a play back rate of 24 frames per second.

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 2.2.2. 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 onto the LAN. In the next round, the buffers are swapped. The effective disk bandwidth depends on the block size used in disk retrieval. Based on the experimental study in [4], we choose a block size of 200KB which yields an effective bandwidth of approximately 5MB/s. From Table 2.2.2, we see that the storage capacity of an Elite-9 disk is about 9GB.

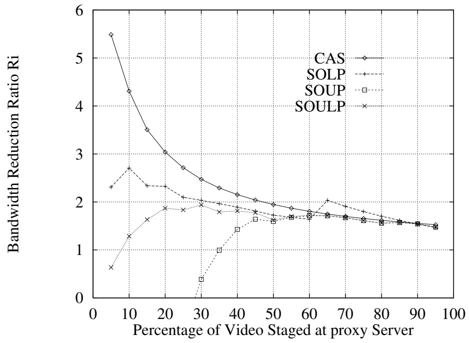


Figure 3: Ratio of backbone WAN bandwidth reduction to proxy server disk bandwidth: single stream

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 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 2.1) 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 will thus be 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 (SOLP), smoothing on the lower part (SOUP), and smoothing on the upper and lower parts (SOULP). In the simulation study below, we will assume that *all clients have a total smoothing buffer of size 1MB*.

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 bandwidth reduction ratio R_i plotted as a function of the percentage of video data staged at the proxy server is shown in Figure 3. 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.

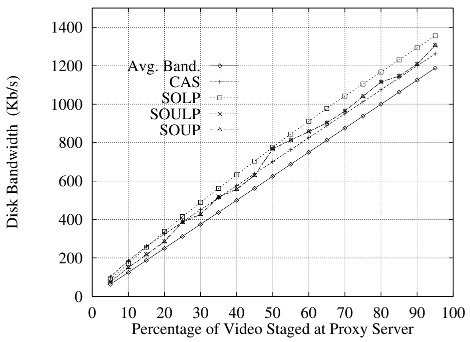
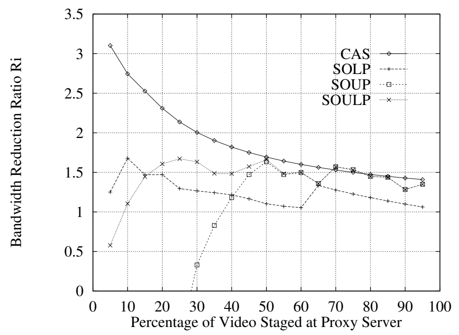


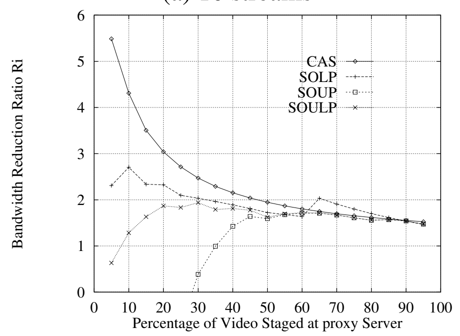
Figure 4: Effective per-stream disk bandwidth requirement

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 4 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 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 5 (a) and (b) show, respectively, the bandwidth reduction ratio R_i under the four video staging methods when 10 and 100 streams of *Wizard of Oz* are statistically multiplexed. It can be observed 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 5: Bandwidth reduction ratio: *Wizard of Oz*

3 Multiple Video Staging Schemes

We study the problem of how to design multiple video staging schemes at a proxy server for a LAN with a video access profile. We first formulate the problem formally and then develop two heuristic algorithms.

3.1 Problem Formulation

Given a set of videos, the expected number of accesses to these videos may vary drastically depending on their popularity. Use 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 [1, 3]. In formulating the problem of design multiple video staging scheme, 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 $1 \leq i \leq M$, 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 the video access profile.

Suppose that the disk system 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, $1 \leq i \leq M$, and fix a video staging method, say, the CAS method. For any given cut-off rate C_i, let T_i 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, $1 \leq i \leq M$, 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 in the design of a multiple video staging scheme is to choose C_i, $1 \leq i \leq M$, 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. Due to the page limitation, we do not present the formal formulation of this problem here. The readers may refer to [5] for the details of the problem formulation.

The optimal solution to this problem is prohibitively complicated, if not impossible. 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 3.4.

3.2 Staging Hot Video Only (SHVO)

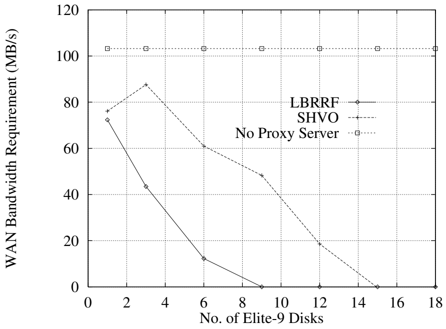
The first heuristic algorithm, referred to as the *staging hot video only* (SHVO) algorithm, regards the user access pattern as the most important factor in determining which video to stage at the proxy server. Intuitively, if a video is "hot, i.e., it has a large number of concurrent accesses, then 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 (as in Figure 4) required to transfer video i from the proxy server to a user, provided 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:

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

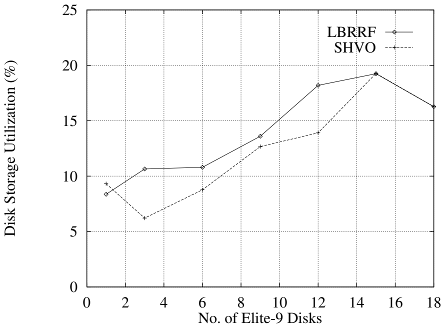
Under this heuristic algorithm, either a video is entirely staged at the proxy server or not at all. However, the Video k, it may be possible that only a portion of a_i users access the video from the proxy server, while the rest of the users access video k from the central video server.

3.3 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 2. 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

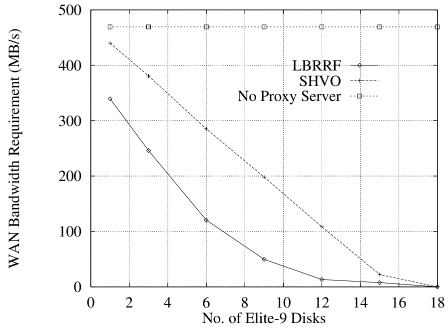


(a) Backbone WAN bandwidth reduction

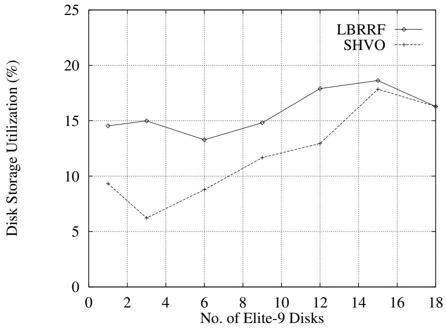


(b) Proxy server disk storage utilization

Figure 6: 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 7: Impact of proxy server disk resources: clients have no smoothing buffer, $\alpha = 0.3$.

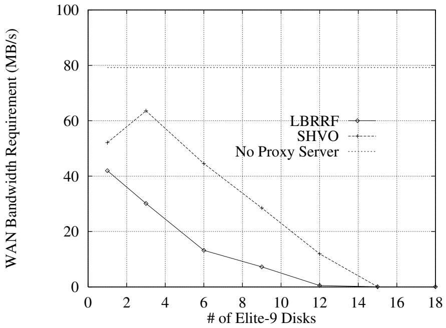
portion of the video at the proxy server will consume some fraction of the total disk bandwidth at the proxy server. In some sense, this fraction of the total disk bandwidth is allocated to video i. Clearly, the video with largest R_i should be favored when allocating the disk bandwidth at the proxy server, given that everything else is equal. More 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, provided 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 should be chosen, and CP_i percentage of its video data as determined by the CAS method will be staged at the proxy server. The readers may refer to [5] for the details of LBRRF algorithm.

3.4 Empirical Evaluation

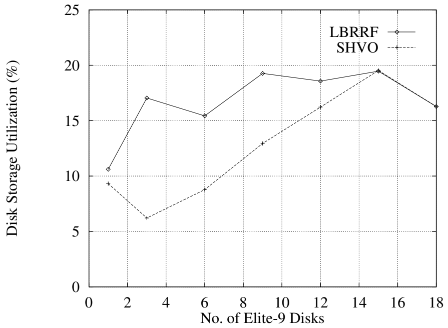
In this section, we evaluate the performance of the two heuristic approaches, SHVO and LBRRF based on simulation. The disk system model is again based on the Seagate Elite-9 disk, the parameters of which is listed in Table 2.2.2. We assume that there are a total of 50 videos (i.e., $M = 50$), and there 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 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. All sets of experiments use a skewness parameter $\alpha = 0.3$. In all the experiments, when clients have extra smoothing buffer, then 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 7 (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 backbone 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 obvious even when there are only a couple of disks available at the proxy server.

Figure 7 (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. This is partially because the LBRRF may stage portion of a video at the proxy server while the SHVO algorithm always stage the entire video at the proxy.



(a) Backbone WAN bandwidth reduction



(b) Proxy server disk storage utilization

Figure 8: Impact of proxy server disk resources: clients have 1MB smoothing buffer, $\alpha = 0.3$.

In the second set of experiments, we assume that clients have a smoothing buffer of size 64KB. In Figure 6 (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 very 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 6 (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.

One more set of experiment is shown in Figure 8 when the size of the client buffer increases to 1MB. The same observation applies here although the difference in the performance of the two heuristic algorithms appears to get smaller as the size of client smoothing buffer increases. In this case 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.

4 Conclusion

In this paper, we 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). By taking the heterogeneous underlying networking environment into consideration, we developed an effective video delivery technique called *video staging* via intelligent utilization of the disk bandwidth and storage capacity available at proxy servers attached to the LANs. 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.

References

[1] A. Dan, D. Sitaram, and P. Shahabuddin. Scheduling Policies for an On-Demand Video Server with Batching. *Proceedings of ACM Multimedia'94*, pages 15–21, October 1994.

[2] J. Salehi, Z.-L. Zhang, J. Kurose, and D. Towsley. 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.

[3] W. Tetzlaff and R. Flynn. Disk Striping and Block Replication Algorithms for Video File Servers. *Porceedings of ICMCS'96*, pages 590–597, June 1996.

[4] Y. Wang and D. Du. Weighted Striping in Multimedia Servers. *ICMCS'97*, June 1997.

[5] Y. Wang, Z.-L. Zhang, D. H. Du, and D. Su. A Network-Conscious Approach to End-to-End Video Delivery over Wide Area Networks Using Proxy Servers. Technical Report TR98-0xx, Department of Computer Science, University of Minnesota, February 1998.