Video Staging: A Proxy-Server-Based Approach to End-to-End Video Delivery over Wide-Area Networks

Zhi-Li Zhang, *Member, IEEE*, Yuewei Wang, *Member, IEEE*, David H. C. Du, *Fellow, IEEE*, and Dongli Su

***Abstract*—Real-time distribution of stored video over wide-area networks (WANs) 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 approach to the problem of end-to-end video delivery over WANs using proxy servers situated between local-area networks (LANs) and a backbone WAN. A major objective of our approach is to reduce the backbone WAN bandwidth requirement. Toward this end, we develop an 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 approach provides an effective and scalable solution to the problem of the end-to-end video delivery over WANs.

***Index Terms*—End-to-end video delivery, heterogeneous networking environment, MPEG, proxy server, video smoothing, video staging, video streaming.**

I. 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 [3]. 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 WAN 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 WAN. 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 Fiber 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 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 WAN 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 WAN operated by another service provider.

In this paper, we present a novel proxy-server-based approach to the end-to-end video delivery over WANs. For simplicity of discussion, the WAN in question is assumed to comprise several local area networks interconnected by a backbone WAN (see Fig. 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

Manuscript received August 21, 1998; revised December 19, 1998 and July, 1999; approved by IEEE/ACM Transactions on Networking Editor S. McCanne. This work was supported in part by University of Minnesota Graduate School Grant-in-Aid, National Science Foundation CAREER Award Grant NCR-9734428, and National Science Foundation Grant ANI-9903228. This paper was presented in part at the IEEE INFOCOM'98 conference.

Z.-L. Zhang and D. H. C. Du are with the Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN 55455 USA (e-mail: zhzhang@cs.umn.edu; du@cs.umn.edu).

Y. Wang and D. Su are currently with 3CX Inc., San Jose, CA 95125 USA (e-mail: ywang@3cx.com; dsu@3cx.com).

Publisher Item Identifier S 1063-6692(00)06797-2.

DOCKET ALARM Find authenticated court documents without watermarks at docketalarm.com.

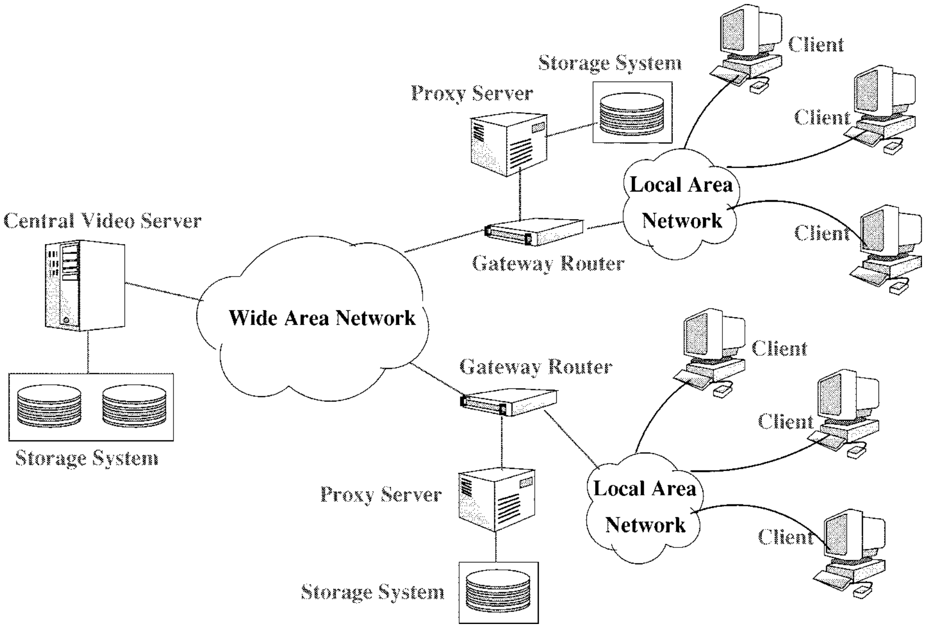

Fig. 1. Video delivery over a simple heterogeneous networking environment.

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*,[1] 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 video delivery approach is to reduce the bandwidth requirement in the backbone WAN*, 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 approach to the problem of end-to-end video delivery across WANs has several salient features. Because of the large storage space at a proxy server, for a given video, a sizable 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 (peak) 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.

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 approach provides an effective and scalable solution to the problem of the end-to-end video delivery over WANs.

The remainder of our paper is organized as follows. In Section II, we describe our problem setting and present our proxy-server-based approach. In Section III, we present various video staging techniques in the context of a single video stream. In Section IV, we develop two heuristic algorithms to solve the problem of designing multiple video staging scheme for a proxy

[1] 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

DOCKET ALARM
Find authenticated court documents without watermarks at docketalarm.com.

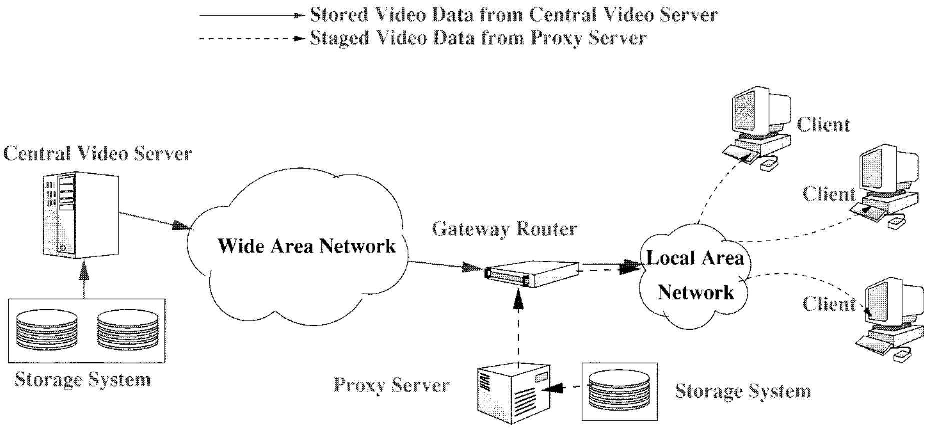

Fig. 2. Proxy-server-assisted video delivery.

II. 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 Fig. 1, where several local area networks are interconnected by a backbone WAN. 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).

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 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 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.

The objective of the video staging technique is to reduce the total backbone WAN (peak) bandwidth required for delivering video to a large number of users on a LAN. As illustrated in Fig. 2, 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 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

DOCKET ALARM Find authenticated court documents without watermarks at docketalarm.com.

reduce the backbone WAN bandwidth required to transmit the video. On the other hand, if the video is infrequently 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 real-time transport protocol (RTP) [7] and real-time streaming protocol (RTSP) [8] protocols. Investigation of these issues is outside the scope of this paper, and will be the topics of future research.

III. 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

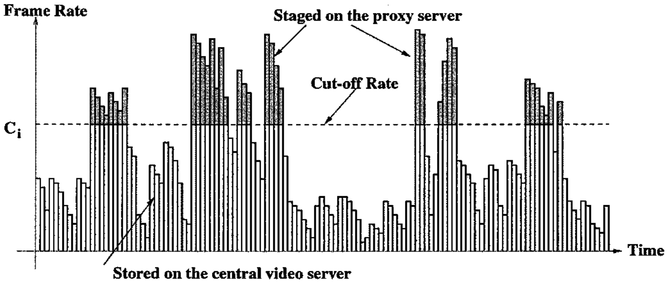

Fig. 3. Simple video staging method using a cut-off rate.

WAN bandwidth reduction when clients have extra buffering capabilities available for smoothing.

A. 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.

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 Fig. 3. 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[2] (in fact, the whole video is always stored at the central video

[2] 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

Find authenticated court documents without watermarks at docketalarm.com.

server). From Fig. 3, 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 bursty, and eventually approaches to an essentially CBR 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 $R_i = (P_i - T_i/D_i)$. *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 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 $R_i = (P_i - T_i/\tilde{D}_i)$.

B. Video Staging with Smoothing

If clients have extra buffering capabilities, the video smoothing [6], [2], [4], [5] 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 [6] is used for video smoothing.

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 [6] 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 III-A, 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 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) Cut-Off Before Smoothing: As in Section III-A, 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 CBS 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 CBR (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, $B_l = B * (C_i/P_i * F)$ amount of the client buffer is used to smooth the lower part of the video, and $B_u = B * (1 - (C_i/P_i * F))$ 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.

C. Empirical Evaluation

In this section, we empirically evaluate the video staging methods presented in Section III-B. The evaluation is carried out based on simulation using MPEG-1 traces. Two MPEG-1

DOCKET ALARM Find authenticated court documents without watermarks at docketalarm.com.

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.

WHAT WILL YOU BUILD? | sales@docketalarm.com | 1-866-77-FASTCASE