An Effective Approach to Video Staging in Streaming Applications

Shin-Hung Chang[+‡], Ray-I Chang[*], Jan-Ming Ho[+], and Yen-Jen Oyang[‡]

[+]Institute of Information Science, Academia Sinica, Taipei, Taiwan.
[‡] Dept. of Computer Science & Information Engineering, National Taiwan University, Taipei, Taiwan.
*Dept. of Information Management, National Central University, Chungli, Taiwan.

Abstract-- **Due to advances in network technologies, providing streaming services over the Internet has gained in popularity. Because a video stream is in compressed format, it is naturally with the variable bit rate (VBR) property and its traffic is highly burst. With the installation of a video proxy between access networks (e.g. local area networks, LAN) and backbone networks (e.g. wide area networks, WAN), video staging is proposed to cache part of the requested video into the video proxy close to clients. In this mechanism, the video can be streamed using a constant bit rate (CBR) network service across the backbone WAN and the WAN bandwidth requirement can be significantly reduced. In this paper, we propose a new approach, called "caching selected after smoothing" (CSAS), to handling video staging. The aim of the CSAS algorithm is to integrate our previous video caching and smoothing technique so as to reduce the required WAN bandwidth even more. Experiments on benchmark videos using different evaluation indices, including the proxy storage requirement, the WAN bandwidth requirement, and the WAN bandwidth utilization, show that our approach is more effective than the conventional "cut after smoothing" (CAS) algorithm.**

I. Introduction

With advances in broadband technologies, providing streaming services over the Internet has gained in popularity. Multimedia applications (e.g., digital library, video on demand and distance learning) require video streaming services to provide a more attractive and effective presentation. Because of its high bandwidth requirement, a video is usually stored and streamed in a compressed format. Therefore, it is naturally with the variable-bit-rate (VBR) property and its peak bit rate is generally much larger than its average bit rate. Due to the burst nature of a video, it is a challenging problem for service providers to provide QoS (quality-of-service) guaranteed video streaming services. Moreover, this problem is more complicated when the video is delivered across the Internet.

The Internet architecture is heterogeneous and it consists of many Internet service providers (ISPs) as shown in Fig. 1. These ISPs interconnect to each other by the backbone WAN network that is owned by the third party. Each client accesses the Internet through an ISP via the so-called access network. Typical examples of access networks include HFC, XDSL, ISDN, or LAN. Because the backbone WAN network is shared by a large number of clients, the delivery quality is difficult to guarantee. Hence it is generally more costly to deliver contents under the backbone WAN network than under the access network.

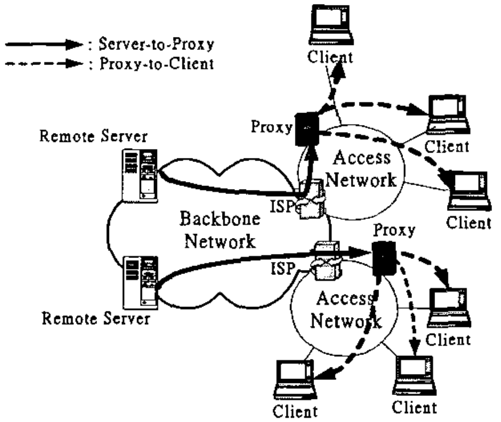


Fig. 1. Illustration of heterogeneous Internet with proxies installed.

To reduce the required WAN bandwidth in video streaming applications, two major technologies have been proposed by previous researchers. (1) Video smoothing: This technique can flatten the bit rate fluctuation of the inter-frame by utilizing the client buffer (smoothing buffer). By averaging the transmission rate of consecutive video frames, the end-to-end peak bandwidth (from server to client) can be reduced significantly. Of course the backbone WAN bandwidth requirement is also reduced. This issue is well studied and much research has been published in this area [6][7][8][9][10]. (2) Video proxy: The proxy technology is a widely used technique for a variety of services. Take the web service for example. The web proxy is the most popular technique to reduce the WAN bandwidth by caching remote web contents with small size (hypertext or image contents) [11]. However, relative to the small size of web contents, it is unrealistic to use the web proxy to provide for video streaming services because the web proxy is uneconomic and inefficient for managing large video contents. Therefore, many proxies for handling video contents were designed by several groups of researchers for different purposes [3][4][5].

Video staging, first proposed by Z. L. Zhang et al.,

0-7803-7632-3/02/$17.00 ©2002 IEEE

caches only a pre-selected portion of the remote video content by utilizing the disk storage in the video proxy [2]. They also proposed an approach to handling video staging. In this paper, we refer to this approach as the "cut-off caching (CC)" algorithm. In order to further reduce the WAN bandwidth requirement, Z. L. Zhang et al. integrated the CC algorithm with the video smoothing technique and proposed an enhanced algorithm, called the "cut after smoothing (CAS)" algorithm (first performing the video smoothing, and then running the CC algorithm). Their experimental results show that the CAS algorithm can reduce the required WAN bandwidth more than the CC algorithm.

In our previous works [1], we have proposed an "optimal cache (OC)" algorithm with linear time complexity ($O(n)$, where n is the total frame number) to handle video staging. With given resources (the startup latency, the client buffer size and the allocated WAN bandwidth), the OC algorithm can minimize the required caching storage in the video proxy. If the same size of cache storage in the video proxy is allocated for providing QoS-guaranteed video streaming, the streaming schedule computed by the OC algorithm uses less WAN bandwidth than that computed by the best-known CC algorithm.

In this paper, we focus on proposing an effective approach to handling video staging. The proposed algorithm, called "caching selected after smoothing (CSAS)" algorithm, is an integration of the OC algorithm [1] and the video smoothing technique [6]. Experiments on benchmark videos using different evaluation indices, including the proxy storage requirement, the WAN bandwidth requirement, and the WAN bandwidth utilization, show that our approach is more effective than the conventional CAS algorithm (extended from the CC algorithm).

The rest of this paper is organized as follows. Related works and problem formulations are described in section II. Our proposed algorithm is presented in section III. The analysis and experimental results are presented in section IV. Finally, we state the conclusions of this paper in section V.

II. Related Works And Problem Formulations

For a clear formulation of the problem and to clearly explain the proposed algorithm, we state the following definitions: A video content can be represented by a sequence of video frames $V = \{f_i > 0 \mid 0 \le i < n, f_{-1} = 0\}$, where f_i is the size of the $i-th$ video frame and n is the total number of video frames. When the video V is requested, each video frame f_i is sequentially streamed to the client for playback. The time period from receiving to playing the video at the client is called the startup latency, denoted by L. In this paper, we formulate the problem on the basis of a discrete time model. Let T_i represent the time period between two consecutive frames' (f_{i-1} and f_i) playback, where $0 \le i < n-1$. Without loss of generality, T_i is $1/frame\,rate$ and the initialized value $T_0 = L$. The time index of the $i-th$ frame playback at the client is defined by $t_i = t_{i-1} + T_i$, where $0 \le i < n$ and $t_{-1} = 0$.

Let $\bar{S} = \{\bar{r}_i \mid 0 \le i < n\}$ represent a video streaming schedule of the remote video server, where $\bar{r}_i$ indicates the rate applied to stream the video out from the video server between the time index t_{i-1} and t_i. For simplified resource management, we assume that network services with minimal delay and no loss is used for streaming videos across the network. Let r_{WAN} indicate the allocated bandwidth used for the end-to-end delivery across the backbone WAN. Additionally, the available network bandwidth across the access network is assumed to be ample.

When the video proxy is constructed to reduce the WAN bandwidth requirement, part of a video will pre-select to be cached in the video proxy close to clients. Let $C = \{c_i \ge 0 \mid -1 \le i < n, c_{-1} = 0\}$ represent a sequence of the pre-selected cached data, where c_i indicates the size of video retrieved from the video proxy at the time index t_i. Additionally, the accumulation of pre-selected video cache size is denoted by $|C| = \sum_{i=-1}^{i=n-1} c_i$.

Consequently, there is a trade-off between the proxy cache storage requirement, $|C|$, and the allocated WAN bandwidth, r_{WAN}. By using the storage space in the video proxy, the partial video content can be retrieved from the video proxy during video streaming instead, so the required WAN bandwidth can be reduced. Given the proxy cache storage, the goal is to propose an effective approach to reducing more WAN bandwidth requirement. Relatively, with the same allocated WAN bandwidth, we can also use less proxy cache storage to provide QoS-guaranteed streaming services.

A. Cut-off Cache (CC) Algorithm

The CC algorithm sequentially compares each video frame with the given cut-off rate (the allocated WAN bandwidth, r_{WAN}). If an entire frame cannot be transmitted by this cut-off rate in a frame period (the duration of each frame playback), the CC algorithm cuts the excessive portion of this frame and stores it into the video proxy as shown in Fig. 2. In the CC algorithm, c_i is designed as $f_i - r_{WAN} \times T_i$. If $c_i \le 0$, then c_i will be set to zero (none of the $i\text{-}th$ video frame is pre-selected to cache in the video proxy). The peak WAN bandwidth requirement can be reduced from $max\{f_i/T_i\}$ to r_{WAN}, because part of a video is accessed from the nearby video proxy.

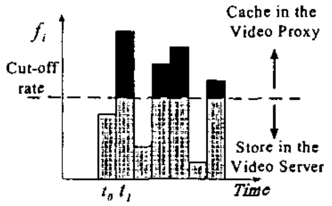


Fig. 2. A cut-off example to illustrate the CC algorithm.

B. *Cut After Smoothing (CAS) Algorithm*

To further reduce the WAN bandwidth requirement, the CAS algorithm is proposed by integrating the CC algorithm with the video smoothing technique: given a video content $V = \{f_0\ f_1 \cdots f_{n-1}\}$, first, the video smoothing technique is applied and a smooth streaming schedule, $\tilde{S} = \{\tilde{r}_i \mid 0 \le i < n\}$, of the remote video server is computed. Second, the CC algorithm is applied and c_i is designed as $(\tilde{r}_i - r_{WAN}) \times T_i$. If $c_i \le 0$, then c_i will be set to zero (none of the video data is retrieved from the video proxy at time index t_i). Additionally, the finial streaming schedule of the remote server is $\{\bar{r}_i = r_{WAN} \mid 0 \le i < n\}$.

By integrating the video smoothing technique, the CAS algorithm is proposed to further reduce the WAN bandwidth requirement. In this paper, we integrate our proposed OC algorithm with the video smoothing technique and propose a new caching algorithm for handling video staging.

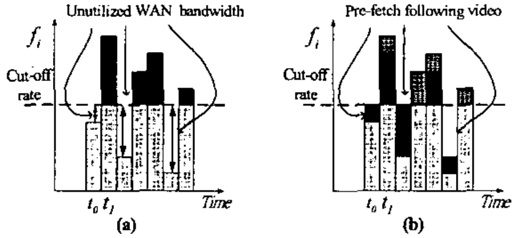


Fig. 3. An example of using the unutilized WAN bandwidth to pre-fetch the following video data, illustrating the OC algorithm.

III. Proposed Algorithm

A. *Optimal Cache (OC) Algorithm*

The main idea behind the OC algorithm is to use the unused WAN bandwidth in the CC algorithm to pre-fetch following video data as shown in Fig. 3(a) and 3(b). The OC algorithm is stated in detail in [1].

B. *Caching Selected After Smoothing (CSAS) Algorithm*

The main idea behind the CSAS algorithm is designed by integrating two processes, the video smoothing process and the OC process. Fig. 4 illustrates the operation of the CSAS algorithm.

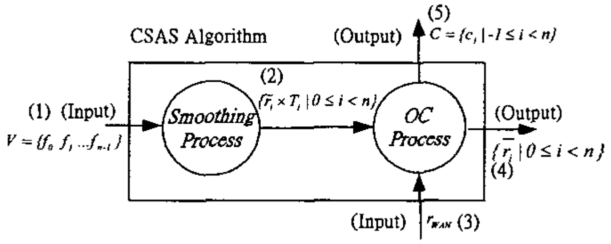

CSAS Algorithm
(Output)
(5)
$C = \{c_i \mid -1 \le i < n\}$
(1) (Input)
$V = \{f_0\ f_1 \cdots f_{n-1}\}$
Smoothing Process
(2)
$\{\tilde{r}_i \times T_i \mid 0 \le i < n\}$
OC Process
(Output)
$\{\bar{r}_i \mid 0 \le i < n\}$
(4)
(Input)
r_{WAN} (3)

Fig. 4. The CSAS algorithm.

(1) First, a video $V = \{f_0\ f_1 \cdots f_{n-1}\}$ is fed into the smoothing process of the CSAS algorithm. (2) Through the smoothing process, an end-to-end basis (from server to client) streaming schedule $\tilde{S} = \{\tilde{r}_i \mid 0 \le i < n, \tilde{r}_{-1} = 0\}$ (smoothed) is computed. The peak bandwidth of this streaming schedule is indicated by $\tilde{S}^*$. After the smoothing process, the regulation video data $\{\tilde{r}_i \times T_i \mid 0 \le i < n\}$ at the client buffer is formulated. (3) Given the allocated WAN bandwidth, r_{WAN}, this regulation video data is fed into the OC process, where $r_{WAN} \le \tilde{S}^*$. (4) The final streaming schedule $\{\bar{r}_i \mid 0 \le i < n\}$ of the remote server is computed. (5) The cached data set $C = \{c_i \mid -1 \le i < n\}$ is also computed during the OC process.

In this algorithm, the client buffer consists of two parts, the smoothing buffer and the staging buffer. The smoothing buffer is designed to use in the video smoothing process and the staging buffer is designed to use in the OC process. Let b_i represent the data aggregation that consists of the pre-fetched video at the staging buffer and the new arrival video from the remote video server. Therefore, it is computed by $b_i = min\{B, b_{i-1} + (\bar{r}_i \times T_i) - (\tilde{r}_{i-1} \times T_{i-1})\}$, where B is the size of the staging buffer. According to the streaming schedule (after the smoothing process), b_i must not be smaller than $\widetilde{r}_i \times T_i$. Let $K(x,y)$ indicate a partial data (with size *x*) of the data *y*. Because the allocated WAN bandwidth, r_{WAN}, might not be large enough and cause $b_i < \widetilde{r}_i \times T_i$ (buffer underflow), the streaming schedule computed by the CSAS algorithm is designed to retrieve the video data $K(c_i, \widetilde{r}_i \times T_i)$ from the video proxy at time index t_i, where $c_i = \widetilde{r}_i \times T_i - b_i$. Additionally, the streaming rate, $\overline{r_i}$, of remote server needs to be adjusted if $b_i == B$ (buffer overflow). A detailed description of the CSAS algorithm follows:

```
*Algorithm:* **Caching Selected After Smoothing (CSAS)**
//$b_i$ is the staging buffer occupancy at time index $t_i$;
//$B$ indicates the size of the staging buffer;
(1) By applying the video smoothing technique, the
    streaming schedule $\{\widetilde{r}_i \mid 0 \le i < n\}$ is computed;
(2) Given the allocated WAN bandwidth $r_{WAN}$;
(3) $i = -1$; $b_i = 0$; $\widetilde{r}_i \times T_i = 0$;
(4)  repeat {
(5)    $i = i + 1$;
(6)    $\overline{r_i} = r_{WAN}$; $b_i = min\{B, b_{i-1} + (\overline{r_i} \times T_i) - (\widetilde{r}_{i-1} \times T_{i-1})\}$
(7)    if ($b_i \ge \widetilde{r}_i \times T_i$) then
(8)    {   $c_i = 0$;
(9)        if ($b_i == B$) {$\overline{r_i} = (B - b_{i-1} + \widetilde{r}_{i-1} \times T_{i-1})/T_i$ }}
(10)   else {$_i c_i = \widetilde{r}_i \times T_i - b$; $b_i = \widetilde{r}_i \times T_i$;
(11)           cache $K(c_i, \widetilde{r}_i \times T_i)$ into the video proxy;}
(12) } until ($i > (n-1)$); Output {$C$ and $\overline{S}$};
```

IV. Simulation Results

In this section, we test the CSAS algorithm and the conventional CAS algorithm by several benchmark videos [12]. Encoding parameters of benchmark videos and parameters used in our experiments are described in TABLE 1. The statistics of video streams used in our experiments is also presented in TABLE 2.

TABLE 1. Parameters used in our experiments.

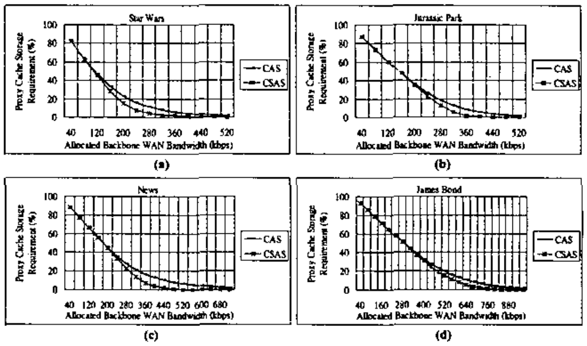
Parameters	Values	Parameters	Values
Encoder Inputs	384x288	Frame Rate	24
Quantizer	I=10, P=14, B=18	Startup Latency	1sec
Encoding Patten	IBBPBBPBBPBB	Client Buffer	200kB

TABLE 2. Statistics of video streams used in our experiments.

Video Stream	Video Size (MB)	AVG Bit Rate (kbps)	Frame Size (kB)		
			MAX	AVG	STD
Star Wars	44.4088	218.278	15.24	1.14	1.58
Jurassic Park	62.36151	306.519	14.6	1.59	1.8
News	73.23109	359.945	23.18	1.87	2.38
James Bond	115.91179	596.73	29.86	2.97	3.14

A. *Proxy Cache Storage Requirements*

To improve the system scalability in the installation of video proxies, the cache storage allocated for serving each video must be precisely controlled. A good algorithm for handling video staging should cache as little portion of the video as possible in the video proxy subject to the allocated WAN bandwidth. For different benchmark videos, the cache storage requirement computed by the CSAS and CAS algorithms is presented in Fig. 5.



(a) (b) (c) (d)

Fig. 5. Proxy cache storage requirement.
(a) Star Wars (b) Jurassic Park (c) News (d) James Bond.

When the allocated WAN bandwidth increases, the cache storage requirement computed by all the above algorithms decreases. However, experiments on these benchmark videos show that our CSAS algorithm can averagely reduce the cache storage requirement in the video proxy by more than 10% less than when computed by the CAS algorithm, if these benchmark videos are streamed with its average bit rate. Additionally, we observed that the decreasing slope computed by the CSAS algorithm is sharper than when computed by the CAS algorithm. Therefore, when the allocated WAN bandwidth increases, our algorithm reduces proxy cache storage even more. This improvement is significant.

B. *Allocated WAN Bandwidth Utilization*

Because the WAN network bandwidth is a costly resource, we must utilize it sufficiently at all times. In a distributed video streaming system, high bandwidth utilization implies that more requests can be served simultaneously. We use all of the above algorithms to compute the pre-cached data in the video proxy. By simulation, we stream these benchmark videos and observe the allocated WAN bandwidth utilization. The experimental results are presented in Fig. 6.

By using the CSAS algorithm, experimental results show that the allocated WAN bandwidth utilization can averagely increase with more than 10%, if these benchmark videos are streamed with its average bit rate. The improvement is significant.

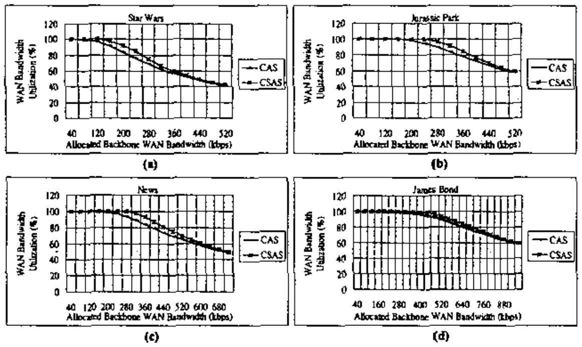


Fig. 6. Allocated WAN bandwidth utilization.
(a) Star Wars (b) Jurassic Park (c) News (d) James Bond.

C. *WAN Bandwidth Requirements*

There is a trade-off between the proxy cache storage and the allocated WAN bandwidth. In Fig. 7, we present the WAN bandwidth requirement computed by the CSAS algorithm and CAS algorithm when the proxy cache storage increases. Experimental results show that the CSAS algorithm can averagely reduce the allocated WAN bandwidth requirement with more than 12%, when the cached data of a video is less than 20%. The improvement is significant.

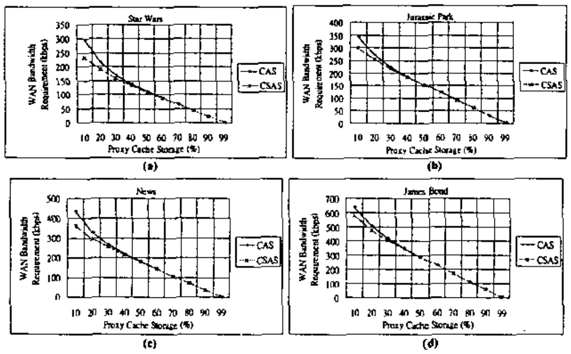


Fig. 7. WAN bandwidth requirement.
(a) Star Wars (b) Jurassic Park (c) News (d) James Bond.

V. CONCLUSIONS

In our previous works, we have proposed the OC algorithm with linear time complexity to handle the video staging. In this paper, we integrate the OC algorithm with the video smoothing technique and proposed a more effective algorithm, the CSAS algorithm, to handle video staging. From the basis of experimental results on several benchmark videos, we conclude that our proposed algorithms are more effective than the conventional CAS algorithm by several evaluation indices, including the small proxy cache storage requirement, small WAN bandwidth requirement, and high WAN bandwidth utilization. The cache storage requirement in the video proxy can be reduced with more than 10% while streaming these videos in the same WAN bandwidth. Relatively, with the same proxy cache storage size, the allocated WAN bandwidth requirement can be reduced with more than 12%. Finally, experimental results show that the WAN bandwidth utilization can averagely increase with more than 10%. The improvement is significant.

REFERENCES

[1] Shin-Hung Chang, Ray-I Chang, Jam-Ming Ho and Yen-Jen Oyang, *"OC: An Optimal Cache Algorithm for Video Staging,"* in proc. of IEEE International Conference on Networking 2002, (ICN 2002).

[2] Zhi-Li Zhang, Yuewei Wang, David H. C. Du, and Dongli Su, *"Video Staging: A Proxy-Server-Based Approach to End-to-End Video Delivery over Wide-Area-Networks,"* in proc. of IEEE/ ACM Transaction on Multimedia 2000.

[3] Wei-Hsiu Ma, and David H. C. Du, *"Reducing Bandwidth Requirement for Delivering Video Over Wide Area Network With Proxy Server,"* in proc. of IEEE International Conference on Multimedia and Expo 2000, (ICME 2000).

[4] Zhourong Miao and Antonio Ortega, *"Proxy Caching for Efficient Services over the Internet,"* in proc. of *9th* International Packet Video Workshop 1999, (PVW 1999).

[5] Subhabrata Sen, Jennifer Rexford, and Don Towsley, *"Proxy Prefix Caching for Multimedia Streams,"* in proc. of IEEE INFOCOM 1999.

[6] Ray-I Chang, M. Chen, Jan-Ming Ho, and M.T. Ko, *"An Effective and Efficient Traffic-Smoothing Scheme for Delivery of Online VBR Media Streams,"* in proc. of IEEE INFOCOM 1999.

[7] Ray-I Chang, M. Chen, M. T. Ko, and Jan-Ming Ho, *"Designing the On-off CBR Transmission Schedule for Jitter-free VBR Media Playback in Real-time Networks," in proc. of IEEE RTCSA 1997.*

[8] Jennifer Rexford, Subhabrata Sen, and Don Towsley, *"Online Smoothing for Live, Variable-bit-rate Video,"* in proc. of NOSSDAV 1997.

[9] Wu-Chi Feng and Jennifer Rexford, *"A Comparison of Bandwidth Smoothing Techniques for the Transmission of Prerecorded Compressed Video,"* in proc. of IEEE INFOCOM 1997.

[10] J. Salehi, Zhi-Li Zhang, J. Kurose, and Don. Towsley, *"Supporting Stored Video: Reducing Rate Variability and End-to-End Resource Requirements through Optimal Smoothing,"* in proc. of ACM SIGMETRICS 1996.

[11] Ilhwan Kim, Yeom, H.Y., and Joonwon Lee, *"Analysis of Buffer Replacement Policies for WWW Proxy,"* in proc. of *12th* International Conference on Information Networking 1998 (ICOIN 1998).

[12] http://nero.informatik.uniwuerzburg.de/MPEG/traces/