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.

E-mail address: viola@iis.sinica.edu.tw

Abstract-- Due to advances of network technologies, it has become practical to stream the stored video over Internet. Because the video stream is in a compressed format, it is naturally with the variable bit rate (VBR) property and the stream traffic is highly burst. By installing the video proxy between the access network (e.g. local area network, LAN) and the backbone network (e.g. wide area network, WAN), Video Staging is proposed to cache part of a video into the video proxy closed to clients [1]. In this manner, 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 proposed a new approach, called the caching selected after smoothing (CSAS), to handling Video Staging. The aim of this approach is to integrate the OC algorithm [1] and the video smoothing technique [7] so as to enhance the effectiveness of reducing the WAN bandwidth. The algorithm we proposed is superior to the conventional algorithms in two ways. Our method use less proxy storage to provide video streaming services if the WAN allocated bandwidth is same. With the same proxy storage, our method can also use less WAN bandwidth to provide video streaming services, especially in videos with high variety of bit rate. From the basis of experimental results on several benchmark videos, we conclude that our proposed algorithm is more effective than conventional one by several evaluation indices, including the small proxy storage requirement, the small WAN bandwidth requirement, and the high WAN bandwidth utilization.

Index terms-- video streaming, constant-bit-rate (CBR), variable-bit-rate (VBR), video staging, video

smoothing and video proxy.

I. INTRODUCTION

With advances in broadband technologies, it has gained in popularity to provide streaming services over the network. Many multimedia applications (e.g. digital library, video on demand and distant learning) currently require video streaming services to produce more attractive presentations. Because of its high bandwidth requirement, a video content is usually stored and streamed in a compressed format. With compression technologies applied, the video content is naturally with variable-bit-rate (VBR) property and the peak bit rate of a video content is generally much larger than its average bit rate. Hence the stream traffic across the network is highly burst. Due to the burst nature of compressed videos, it is a challenging problem for service providers to provide the quality-of-service (QoS) guaranteed video streaming services. Moreover, this problem is currently 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 Figure 1. These ISPs interconnect to each other by the backbone network that is owned by the third party. The backbone network is generally referred to as the wide area network (WAN). Each client accesses the Internet through an ISP via the so-called access network. Typical examples of access networks include XDSL, ISDN, and LAN. Because the backbone WAN 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 than under the access network.

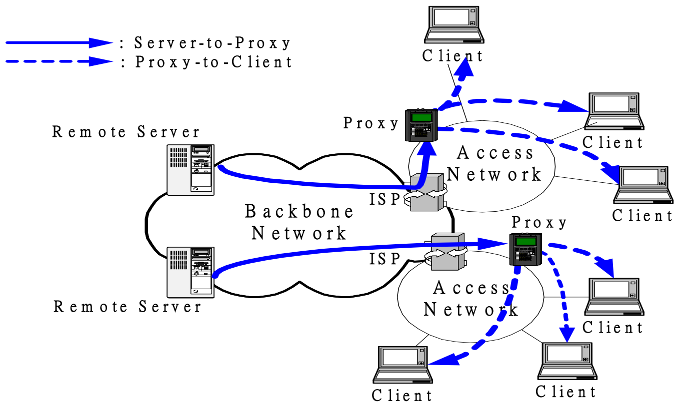

Figure 1. Illustration of heterogeneous Internet with proxies installed.

To reduce the WAN bandwidth required, 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. 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 (form server to proxy) is also reduced. This issue is well studied and much research has been published in this area [7]. (2) Video staging: 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. It is mainly designed to handle the hypertext or the image contents and thus the granularity of web caching is generally web contents with a small size [6]. However, relative to the small size of web contents, it is unrealistic to switch the web proxy to provide video services directly because web proxy is expensive for managing large video contents. For different purposes, many proxies for handling the video content are designed by several groups of researchers[3][4][5]. In this paper, we focus on proposing an effective approach to handling Video Staging.

Video Staging, first proposed by Z. L. Zhang et al., caches only a pre-selected portion of the remote video by utilizing the disk storage in the video proxy[2]. They presented an algorithm to handle Video Staging. In this paper, we refer to this algorithm as “cut-off caching (CC)” algorithm. In order to further reduce the WAN bandwidth requirement, Z. L. Zhang et al. integrated the CC algorithm with video smoothing techniques and proposed enhanced algorithm, the cut-off after smoothing (CAS, first performing the video smoothing, and then doing the CC algorithm). From their experimental results, the CAS algorithm can reduce the WAN bandwidth required even more.

In our previous works, we have proposed an optimal caching (OC) algorithm with linear time complexity to handle Video Staging. With given resources (the startup latency, the client buffer size and the WAN available bandwidth), the OC algorithm can minimize the caching storage in the video proxy. In addition, if the cache storage in the video proxy is allocated same size, the OC algorithm can provide QoS-guaranteed video streaming services with less WAN bandwidth required than when computed by the CC algorithm.

In this paper, we propose a new effective algorithm to handle Video Staging; this algorithm, called caching selected after smoothing (CSAS) algorithm, is based on the OC algorithm integrated with video smoothing technologies. Our experimental results show that the proposed algorithm is more effective than conventional ones by several evaluation indices, including the small proxy cache storage, the small WAN bandwidth, and the high WAN bandwidth utilization.

The rest of this paper is organized as follows: Related works and the problem formulations are described in section II. Our proposed algorithm is presented in section III. The analysis and experimental results are presented in section VI. 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 $V = \{f_i > 0 \mid -1 \le i < n, f_{-1} = 0\}$, 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 the playback. The time period from receiving to playing the video at the client is called the startup latency L. In this paper, we formulate the problem on the basis of the discrete time model. Let T_i represent the time period between two consecutive frames' (f_i and f_{i+1}) playback, where $-1 \le i < n-1$. Without loss of generosity, T_i is $1/frame\ rate$ and the initialized value $T_{-1} = L$. The time index of the $i-th$ frame playback at client is defined by $t_i = t_{i-1} + T_{i-1}$, where $0 \le i < n$ and $t_{-1} = 0$.

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

When the video proxy is constructed to reduce the WAN bandwidth requirement, part of a video will be pre-selected to cache in the video proxy closed to clients. Let $C = \{c_i \ge 0 \mid -1 \le i < n\}$ 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 cache size is denoted by $|C| = \sum_{i=-1}^{i=n-1} c_i$.

By using the storage space in the video proxy, the partial video content is retrieved from the video proxy instead so the WAN bandwidth can be reduced. Consequently, there is a trade-off between the proxy cache storage requirement and the available WAN bandwidth. In this paper, given the available proxy cache storage $|C|$, the goal of a good pre-selected caching algorithm is to minimize the allocated peak WAN bandwidth requirement, R_{WAN}, with the minimum computation overhead.

A. Previous Algorithm

The CC algorithm sequentially compares each video frame with the given cut-off rate (the available WAN bandwidth, R_{WAN}). If an entire frame cannot be transmitted by the 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 Figure 2.

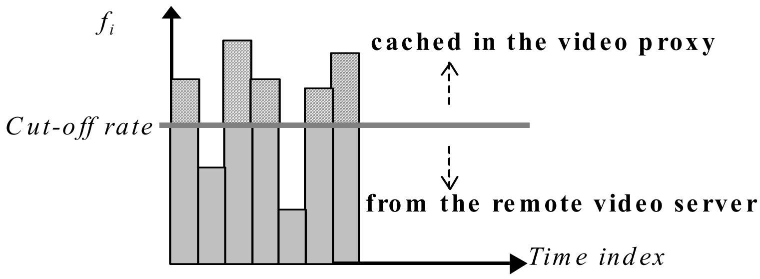

Figure 2: A cut-off example to illustrate the CC algorithm.

In the CC algorithm, c_i is designed as $f_i - R_{WAN} \times T_i$. If $c_i \leq 0$, then c_i will be set to zero (none of the i-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. To further reduce the WAN bandwidth requirement, the CAS algorithm is proposed by integrating the CC algorithm with video smoothing techniques. In the CAS algorithm: Given a video content *V*, first, the video smoothing technique is applied. A streaming schedule, $\tilde{S} = \{\tilde{r}_i \mid -1 \leq i < n\}$, 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 $c_i \leq 0$, then c_i will be set to zero (none of the video data is retrieved from the video proxy at time index t_i).

By integrating CC algorithm and video smoothing techniques, the CAS algorithm is proposed to further reduce the WAN bandwidth requirement. In this paper, with the similar idea, we integrate the OC algorithm with the video smoothing techniques to propose new caching algorithm for handling Video Staging.

III. PROPOSED ALGORITHM

A. Optimal Caching (OC) Algorithm

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

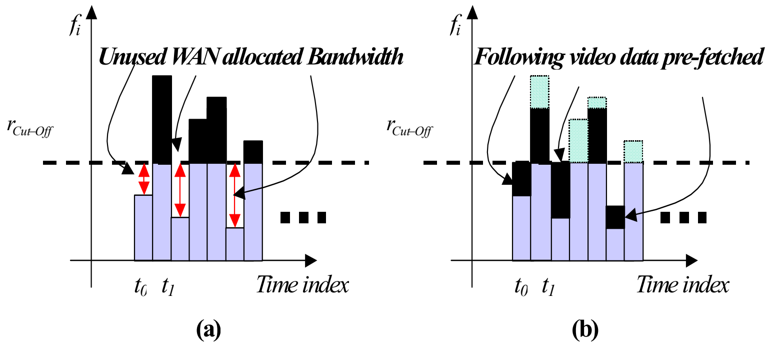

Figure 3. An example of using the unused WAN allocated bandwidth to pre-fetch the following video data, illustrating the OC algorithm.

B. Caching Selected After Smoothing (CSAS) Algorithm

The main idea behind the CSAS algorithm is designed by combining two processes, the video smoothing process and the OC process. A diagram in Figure 5 is presented to illustrate the operation of the CSAS algorithm.

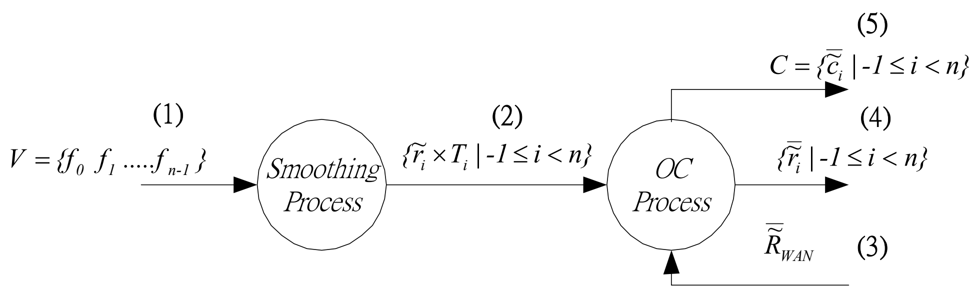

Figure 5: The CSAS algorithm.

(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 smoothing process, an end-to-end basis (from server to client) streaming schedule $\tilde{S} = \{\tilde{r}_i \mid -1 \le i < n\}$ is computed. The peak bandwidth of this streaming schedule is indicated by $\tilde{S}^*$. After the smoothing process, the regulation video data $\{\tilde{r}_i \times T_i \mid -1 \le i < n\}$ required by the client is formulated. (3) Given the WAN allocated bandwidth, R_{WAN}, the regulation video data is fed into the OC process, where $R_{WAN} < \tilde{S}^*$. (4) The final streaming schedule $\tilde{\tilde{S}} = \{\tilde{\tilde{r}}_i = R_{WAN} \mid -1 \le i < n\}$ of the remote server is designed. (5) The cached data set $C = \{c_i \mid -1 \le i < n\}$ is also computed during the OC process. The detail description of CSAS algorithm is presented in the follows.

Algorithm: **Caching Selected After Smoothing (CSAS) Algorithm**

// b_i is the staging buffer occupancy at time index t_i.

//Given the WAN allocated bandwidth R_{WAN};

(1) $i = -1$; $b_i = 0$;

(2) By applying the video smoothing technique, the streaming schedule $\{\tilde{r}_i \mid -1 \le i < n\}$ is computed;

(3) repeat {

(4) $i = i + 1$;

(5) $b_i = min\{B, b_{i-1} + (R_{WAN} - \tilde{r}_{i-1}) \times T_i\}$;

(6) if ($\tilde{r}_i \times T_i \le b_i$) then $c_i = 0$;

(7) else {$c_i = \tilde{r}_i \times T_i - b_i$; $b_i = \tilde{r}_i \times T_i$;

(8) Cache $K(\tilde{r}_i \times T_i, c_i)$ into the video proxy;}

(9) }until ($i > (n-1)$);

In this paper, the staging buffer is defined to use for the OC process of the CSAS algorithm. Let b_i

represent the data aggregation that is consists of the pre-fetched video at the staging buffer and new arrival video from the remote video server. Hence it is computed by $min\{B, b_{i-1} + (R_{WAN} - \tilde{r}_{i-1}) \times T_i\}$, where B is the size of the staging buffer. According to the streaming schedule, b_i must not be smaller than $\tilde{r}_i \times T_i$ (the regulated video data at time index t_i). Let $K(A,B)$ indicate the partial data with size B of the data A. Because the WAN allocated bandwidth, R_{WAN}, might not be large enough and cause $b < \tilde{r}_i \times T_i$, the main idea behind the proposed CSAS algorithm is scheduled to retrieve $K(\tilde{r}_i \times T_i, c_i)$ from the video proxy at time index t_i, where $c_i = \tilde{r}_i \times T_i - b_i$.

IV. EXPERIMENTAL RESULTS

In this section, we test the CSAS algorithm and the conventional algorithm by several benchmark videos. Encoding parameters of benchmark videos and parameters used in our experiments are presented in Table 1. Additionally, the statistics of video streams used in our experiments is 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 constructing video proxies, the cache storage allocated for serving each video must be precisely controlled. Given same resources, a good algorithm, for handling Video Staging, should cache as little portion of the video as possible in the video proxy. For different benchmark videos, the cache storage requirement computed by our algorithm and conventional one is presented in Figure 6.

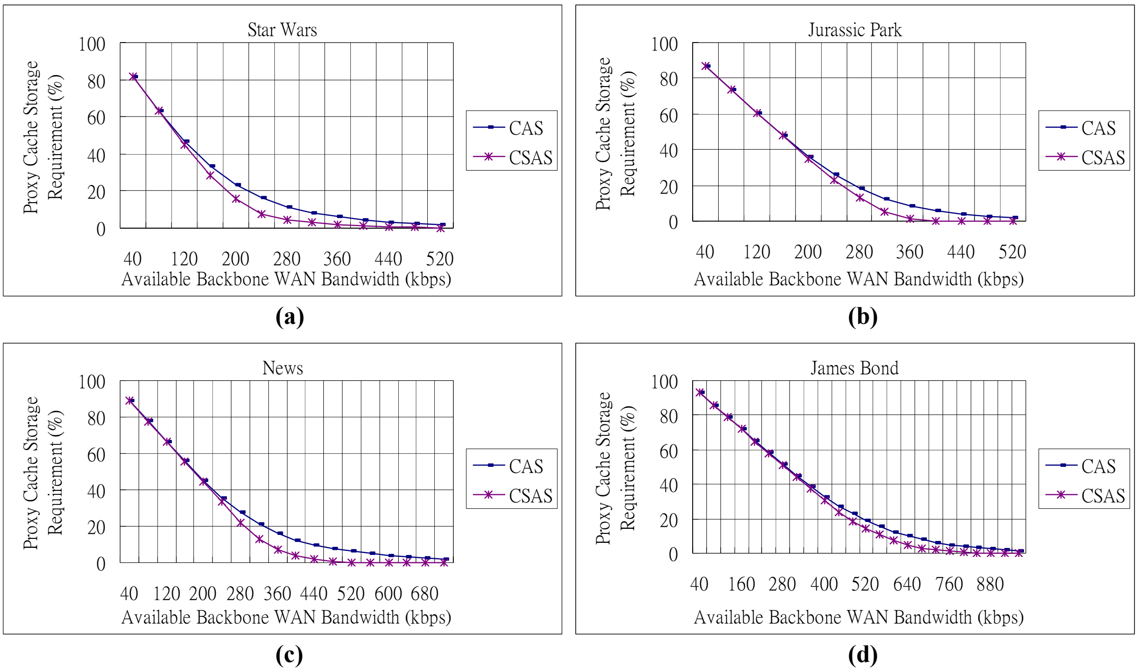

Figure 6. Proxy cache storage requirement.

(a) Star Wars (b) Jurassic Park (c) News (d) James Bond.

When the available WAN bandwidth increases, the cache storage requirement computed by these algorithms decreases. However, experiments on several benchmark videos show that our proposed algorithm can averagely reduce the cache storage requirement in the video proxy by more than 10% less than when computed by conventional 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 WAN allocated bandwidth increases, our algorithm reduces proxy cache storage even more. This improvement is significant.

B. WAN Available 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 four benchmark videos and observe the WAN allocated bandwidth utilization. The experimental results are presented in Figure 7.

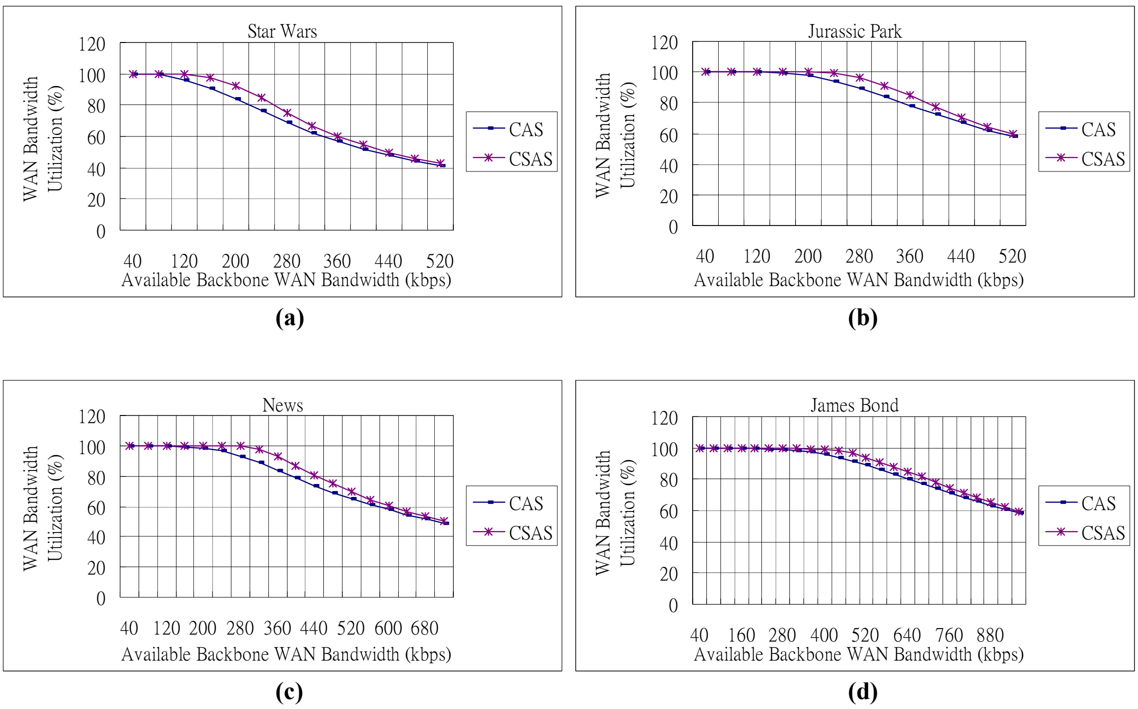

Figure 7: WAN allocated bandwidth utilization.

(a) Star Wars (b) Jurassic Park (c) News (d) James Bond.

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

C. WAN Bandwidth Requirements

There is a trade-off between the proxy cache storage and the WAN available bandwidth. In Figure 8, we present the WAN bandwidth requirement computed by the CSAS algorithm and CAS algorithm. By the CSAS algorithm, experimental results show that the WAN allocated bandwidth utilization can averagely increase with more than 10%, if these benchmark videos are streamed with its average bit rate. The improvement is significant.

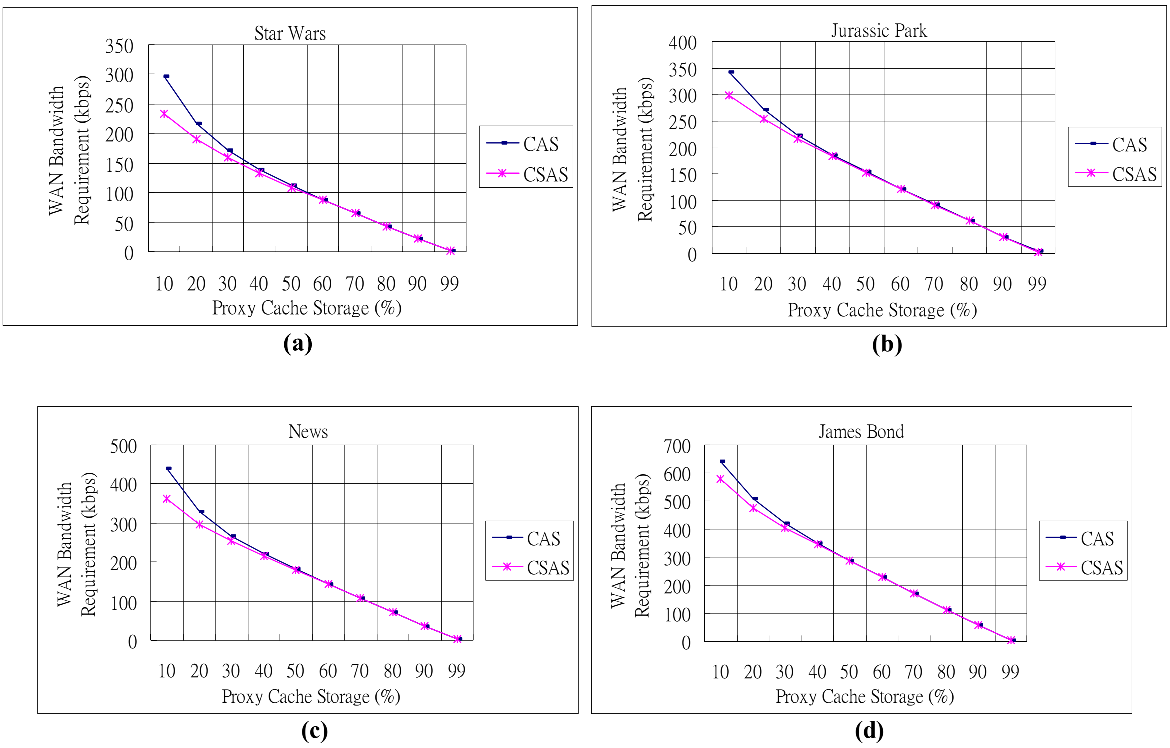

Figure 8: Proxy Disk Peak Bandwidth Requirement.

(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 techniques 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 conventional one by several evaluation indices, including the small proxy cache storage requirement, small WAN bandwidth requirement, and high WAN bandwidth utilization. The CSAS algorithm can averagely reduce the cache storage requirement in the video proxy by more than 10% less than when computed by conventional algorithm. The WAN bandwidth requirement is reduced more than 10% while using the CSAS algorithm with same proxy storage size. Additionally, by the CSAS algorithm, experimental results show that the WAN allocated bandwidth utilization can averagely increase with more than 10%.

VI. REFERENCES

[1]Ray-I Chang, Shin-Hung Chang, and Jam-Ming Ho, "*OSC: Optimal Selective Caching for video transmission with proxy servers",* TR-IIS-00-015, Technical Report, 2000.

[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*, August 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'99).

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

[6]Ilhwan Kim; Yeom, H.Y.; Joonwon Lee , "*Analysis of buffer replacement policies for WWW proxy*", *in proc. of Twelfth International Conference on Information Networking* 1998 (ICOIN'98).

[7]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.

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