Available online at www.sciencedirect.com
SCIENCE DIRECT
Computer Communications 28 (2005) 1852–1861
computer communications
www.elsevier.com/locate/comcom

An optimal cache algorithm for streaming VBR video over a heterogeneous network

Shin-Hung Chang[a,b,*], Ray-I Chang[c], Jan-Ming Ho[a], Yen-Jen Oyang[b]

[a]Institute of Information Science, Academia Sinica, Nankang, Taipei 115, Taiwan, ROC
[b]Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan, ROC
[c]Department of Engineering Science and Ocean Engineering, National Taiwan University, Taipei, Taiwan, ROC

Received 5 July 2004; revised 15 December 2004; accepted 10 January 2005
Available online 2 February 2005

Abstract

High quality video content for on-demand services is usually stored and streamed in a compressed format with a VBR (variable bit rate) property; however, the streaming traffic is extremely bursty. If there is no client buffer to regulate the video's delivery, the backbone WAN (wide area network) bandwidth needs to allocate the video's peak bit rate to guarantee playback quality. To reduce the bandwidth requirement in the backbone WAN, previous researchers have proposed a Video Staging Mechanism to cache portions of the video in a video proxy close to clients. In this paper, we propose a very effective OC (optimal cache) algorithm to handle the Video Staging Mechanism and prove theoretically that the proxy cache computed by our OC algorithm for each video is minimal when all other resources remain constant. On the basis of experiment results, we cache the least amount of video data in the video proxy by using the OC algorithm, and reduce the WAN bandwidth requirement by an amount equal to that of conventional algorithms. In contrast, given the equal size of the storage in a video proxy, the OC algorithm reduces the bandwidth requirement in the backbone WAN much more than conventional algorithms.

Keywords: Heterogeneous network; Video cache; Video staging; Video streaming; VBR video

1. Introduction

Due to insufficient WAN bandwidth and unstable transmission, it is difficult for many service providers to stream high quality audio/visual content. Therefore, most people still rent or purchase video tapes, VCDs, or DVDs. With advances in network technology, streaming continuous media content (video or audio) across networks has become practical, but users still cannot enjoy high quality audio/visual content on-line. Currently, various commercial products, including Real Media, Microsoft Media and Apple QuickTime, can deliver this on-line content. The majority of these streaming products usually provide an on-demand service of low-bit rate video content. However, with the rapid growth of streaming services, customers are becoming increasingly sensitive to video playback quality. Viewing poor-quality video content with a low bit rate on small computer screens leaves users dissatisfied.

A high-quality video for on-demand services is generally stored and streamed in a compressed format that naturally contains a VBR (variable bit rate) property; however, streaming traffic is extremely bursty [16,18–20,22–24]. Because of a high-quality video's huge size and critical bandwidth constraint, the problem of streaming the content across a variety of networks, especially over the Internet, is most challenging [27,28].

The Internet's architecture is heterogeneous and consists of many ISPs (Internet Service Providers) that interconnect with each other via a backbone WAN owned by a third party. The WAN covers a wide area by interconnecting many access networks. An access network is installed to connect end users to the backbone WAN. Meanwhile, an

* Corresponding author. Address: Institute of Information Science, Academia Sinica, Nankang, Taipei 115, Taiwan, ROC. Tel.: +886 227883799x1656; fax: +886 227824814.
E-mail address: viola@iis.sinica.edu.tw (S.-H. Chang).

0140-3664/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2005.01.010

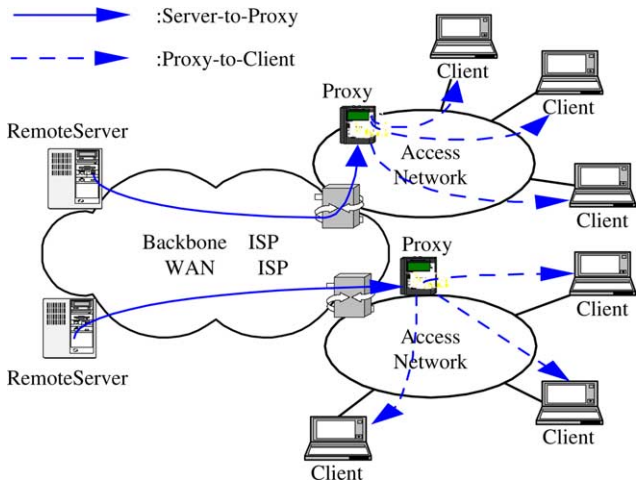

Fig. 1. Heterogeneous network architecture on the Internet. An illustration of the video streaming system with video proxies installed.

end user can access the Internet for data delivery services through an ISP via access networks, as shown in Fig. 1. Typical examples of access networks are PSTN, HFC, XDSL, or LAN (local area network). Generally, traffic load within an access network is easy to manage and manipulate; however, as the backbone WAN is usually shared by a large number of users, it is more difficult for network service providers to guarantee quality of service (QoS). Hence, it is more costly to deliver contents across a backbone WAN than across an access network. If the WAN bandwidth can be significantly reduced while streaming videos, video streaming services will become cheaper and more popular. Currently, there are two major technologies that can reduce the bandwidth requirement in the backbone WAN while streaming bursty videos. The techniques are as follows:

(1) *Video smoothing*. This technique flattens the bit rate fluctuation of the inter-frame by utilizing a client buffer (called a smoothing buffer in this paper). By averaging the transmission rate of consecutive video frames, the end-to-end peak bandwidth (from server to client) can be reduced. Of course, the bandwidth requirement in the backbone WAN is also reduced. This issue has been well researched and many works have been published on the subject [4,5,10,12–15,17,21,25,26]. By using the optimal traffic smoothing algorithm in [17], a minimal smoothing rate for streaming each video across a network can be obtained. However, if the backbone WAN bandwidth is less than this minimal smoothing rate, the quality of the video transmission cannot be guaranteed.
(2) *Video proxy*. Proxy technology has been widely used for improving service quality and distributing content, as shown in Fig. 1. Web content (e.g. hypertext and image data) is cached in a web proxy close to a client and end-users can retrieve it from the web proxy via the ample local access link. By reducing content retrieval from the remote web server, the WAN bandwidth requirement is reduced and the remote web server traffic is off-loaded [11]. However, compared with the small size of web content, video content is usually huge, so caching an entire video to eliminate the WAN bandwidth requirement is unrealistic. Hence, it is impractical to apply the web proxy to directly handle video caching services.

The best way to stream high-quality video is to replicate a mirror video server close to clients in the access network using access network bandwidth only (i.e. without using WAN bandwidth). However, in a video streaming system, a large number of videos are stored in the video server for on-demand services and the total amount of video content usually tends to a high Terabytes level. In terms of cost-benefits, replicating video servers in all access networks is uneconomical and impractical. Therefore, we propose the installation of a video proxy in the access network to cache some video content. The objective is to reduce the bandwidth requirement in the backbone WAN.

Many proxies for handling video contents have been designed for different purposes (such as eliminating startup latency, on-line smoothing, and reducing WAN bandwidth) [1–3,6–9]. Among these proxies, the Video Staging Mechanism, first proposed by Zhang et al., caches only a pre-selected portion of a video's data into a video proxy close to clients. Also, an algorithm for handling the Video Staging Mechanism was presented in [6]. We refer to it as the CC (cut-off cache) algorithm. It is a one-pass algorithm that sequentially compares each frame in a video with a given cut-off rate (the allocated WAN bandwidth). 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 excess portion of the video frame and stores it in the video proxy prior to delivery. An illustration of the CC algorithm is presented in Fig. 2(a). The algorithm is a good design for system implementation. However, a compressed video usually has a large size variation between frames. One frame size may be very small, so the allocated WAN bandwidth may not be fully utilized, as shown in Fig. 2(b). If this unutilized WAN bandwidth could be used to pre-fetch subsequent video data, the storage requirement in the video proxy would be reduced further, as shown in Fig. 2(c).

Zhang et al. also proposed an enhanced CAS (cut-off after smoothing) algorithm to handle the Video Staging Mechanism. This is the most effective algorithm in [6]. It is a two-pass algorithm that combines the CC algorithm and video smoothing technologies to further reduce the requirement for proxy storage and WAN bandwidth. In the two-pass process, the smoothing algorithm is run first,

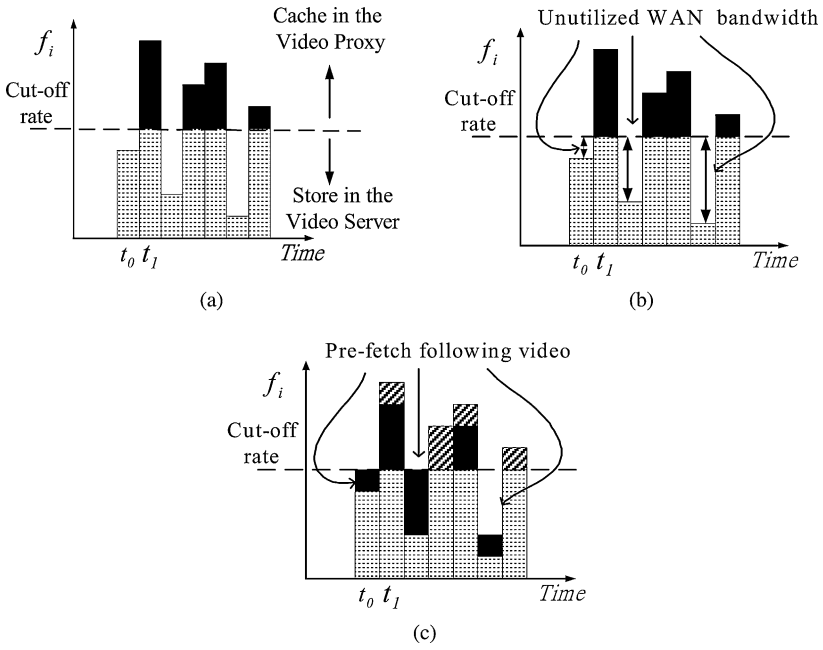

Fig. 2. (a) Illustration of the CC algorithm. (b) Unutilized WAN bandwidth. (c) An illustration of using unutilized WAN bandwidth to pre-fetch video data.

followed by the CC algorithm. According to the experiment results in [6], the CAS algorithm is more effective than the CC algorithm. However, it is more complicated to handle the server streaming schedule and the client buffer control computed with the CAS algorithm than with the CC algorithm.

To solve the above mentioned problems, we propose an optimal approach for handling the Video Staging Mechanism. Given a video's content and specific resources, including the allocated WAN bandwidth, client buffer, and startup latency, we propose a one-pass algorithm, called the Optimal Cache (OC) algorithm, to compute the subset of a video cached in a video proxy with linear complexity (O(n), where n is the number of frames). This is the same as the CC algorithm. In addition, we theoretically prove that the proxy cache computed by our OC algorithm is minimal. This indicates that the OC algorithm caches the least video content and reduces WAN bandwidth requirement by the same amount as conventional algorithms. In contrast, if the size of allocated cache is given in the video proxy, the OC algorithm requires less WAN bandwidth than the CC or the CAS algorithms to provide QoS-guaranteed video streaming services. By experimenting with several benchmark videos [30], we show that our OC algorithm is the most effective in terms of proxy storage, WAN bandwidth requirement, and utilization of allocated WAN bandwidth.

The rest of this paper is organized as follows. The optimal storage problem is formulated in Section 2. Our proposed algorithm is presented in Section 3. The analysis and experimental results are presented in Section 4. Finally, in Section 5, we present our conclusions.

2. Problem formulation

The goal of this paper is to compute the smallest amount of a video's content that must be cached in a proxy, subject to the allocated WAN bandwidth, while providing QoS-guaranteed video streaming services. We refer to this as the cache minimization (CM) problem. For a clear formulation of the CM problem and to clearly explain our proposed algorithm, we state the following definitions. A video's content, V, is represented by a sequence of video frames $\{f_i > 0 | -1 \le i < n, f_{-1} = 0\}$, where f_i is the size of the ith video frame and n is the total number of video frames. The video size is denoted by $|V| = \sum_{i=-1}^{i=n-1} f_i$. The time period from receiving to playing the video by the client is called the startup latency, denoted by L and the client buffer size is denoted by B. We formulate the problem on the basis of the discrete time model. Let T_i represent the time period between the playback of two consecutive frames (f_{i-1} and f_i), where $0 \le i < n$. Without loss of generality, T_i is $1/$*frame rate* and the initialized value $T_0 = L$. The time instance of the ith frame playback at the client is defined by $t_i = t_{i-1} + T_i$, where $-1 \le i < n$ and $t_{-1} = 0$.

Let $S = \{r_i | 0 \le i < n\}$ 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 instance t_{i-1} and t_i. r_{WAN} represents the allocated WAN bandwidth (the maximum bandwidth that can be used to deliver video content across the WAN), where $r_i \le r_{\mathrm{WAN}}$. To simplify network resource management, we assume that network services with minimal delay and no loss are used for streaming videos across networks. In addition, the available network bandwidth under the access network is assumed to be ample.

The video proxy is designed to cache parts of a video's content, V. Let $C=\{c_i \geq 0 | -1 \leq i < n\}$ represent a sequence of cached data sets in the video proxy, where c_i indicates the cached size of the ith video frame and the total cumulative cached size is denoted by $|C|=\sum_{i=-1}^{i=n-1} c_i$. Finally, the CM problem is formulated as follows:

Problem. Given a video, the CM problem is to determine a subset C of this video pre-cached into the video proxy such that the cumulative cached size $|C|$ is minimal, while the startup latency L, the client buffer size B, and the allocated WAN bandwidth r_{WAN} remain constant.

3. Optimal cache (OC) algorithm

To guarantee video playback quality, frame f_i should be available at the client for display at time instance t_i. Except for consuming f_i for playback, the client also receives $r_i T_i$ (bits) of video data simultaneously from the remote video server. Let $\{b_i | -1 \leq i < n\}$ represent a sequence of the buffer occupancy at the client and the initial value $b_{-1}=0$. The buffer occupancy represents the data aggregation that consists of the pre-fetched video content in the client buffer and the newly arrived video content from the remote video server. Therefore, b_i can be computed by $\min\{B, b_{i-1}+(r_i T_i)-f_{i-1}\}$, as shown in Fig. 3(a).

The buffer occupancy b_i must not be smaller than f_i (the frame required for playback at time instance t_i). Unfortunately, the allocated WAN bandwidth, r_{WAN}, might not be large enough and might cause buffer underflow (i.e. if b_i is smaller than f_i). Therefore, the OC algorithm schedules the client to retrieve the excess part of the ith frame, denoted by $f_i(c_i)$, from the closed video proxy at time instance t_i, where $c_i=f_i-b_i$, as shown in Fig. 3(b). If the allocated WAN bandwidth results in buffer overflow (i.e. b_i is equal to B), the streaming rate of the video server needs to be modified to $(B-b_{i-1}+f_{i-1})/T_i$ at the time instance t_i, as shown in Fig. 3(c). The detailed OC algorithm is as follows:

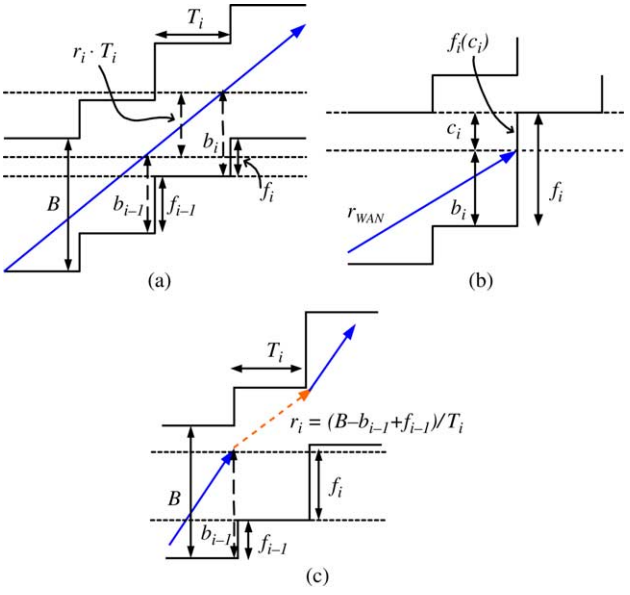

Fig. 3. (a) Without considering the video proxy installed, the client buffer occupancy b_i at time instance t_i is computed by $\min\{B, b_{i-1}+(r_i T_i)-f_{i-1}\}$. (b) Given the allocated WAN bandwidth r_{WAN}, the cached portion of the video frame f_i is denoted by $f_i(c_i)$, where c_i is the cached size. (c) If buffer overflow occurs, the server streaming rate must be modified.

Algorithm: Optimal Cache (OC) Algorithm

//Given a video $V=\{f_i>0 | -1 \leq i < n, f_{-1}=0\}$, *where n is the number of frames;*

//b_i *is the client buffer occupancy at the time instance* t_i.

//B indicates the size of the client buffer;

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

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

(2) ***repeat***

(3) { $\quad i=i+1;$

(4) $\quad r_i=r_{WAN};$

(5) $\quad b_i=min\{B, b_{i-1}+(r_i \times T_i)-f_{i-1}\};$

(6) ***if*** $(f_i \leq b_i)$

(7) { $\quad c_i=0;$

(8) $\quad$ ***if*** $(b_i == B)$ { $r_i=(B-b_{i-1}+f_{i-1})/T_i;$}} /*buffer is overflow*/

(9) ***else*** /* buffer is underflow*/

(10) { $c_i=f_i-b_i;\ b_i=b_i+c_i=f_i;$

(11) $\quad$ *Cache* $f_i(c_i)$ *into the video proxy;*}

(12) ***until*** $(i > (n-1))$};

In Fig. 4, assuming that the bandwidth in the access network is infinite, we show two possible schedules for video streaming services, subject to QoS-guaranteed video playback (without buffer underflow or overflow). This assumption is reasonable because the bandwidth of an access network is usually 10–100 times larger than that of the backbone WAN. Using these possible schedules, we prove that the streaming schedule computed by the OC algorithm caches the minimal amount of video data in the video proxy to provide QoS-guaranteed video playback.

Theorem. *The proxy cache computed by the OC algorithm is smaller than, or equal to, that computed by other algorithms, while the startup latency, client buffer size, and allocated WAN bandwidth remain constant.*

Proof. Let $\{t_{u_1}, t_{u_2}, \ldots, t_{u_m}\}$ represent a sequence of time instances of buffer underflow during the computation of the OC algorithm. $\{c_i | -1 \leq i < n,\ c_i^* \geq 0\}$ indicates

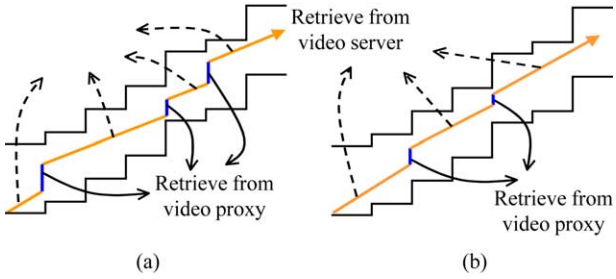

Fig. 4. Two possible schedules for video streaming services, subject to QoS-guaranteed playback.

a sequence of cached data computed by the OC algorithm. We use Mathematical Induction [29] to prove that at any time instance $\{t_{u_k} | 1 \le k \le m\}$, the size of the cumulative cached video content computed by the OC algorithm, $\sum_{i=-1}^{i=u_k} c_i^*$, is minimal. For a clear formulation of the proof, we define the cumulative transmission Eq. (1) as the amount of data sent by the remote video server and retrieved from the video proxy before time instance t_i.

$$G(i) = G(i-1) + r_i T_i + c_i \tag{1}$$

(1) Given that $k=1$, we prove that $\sum_{i=-1}^{i=u_1} c_i^*$ is minimal by contradiction.
   (a) We assume there is a new algorithm caching less content than the OC algorithm between time instances t_{-1} and t_{u_1}. Let $\{c'_i | -1 \le i < n,\ c'_i \ge 0\}$ indicate a sequence of cached data computed by this new algorithm. The cumulative cached video data, from time instances t_{-1} to t_{u_1}, is denoted by $\sum_{i=-1}^{i=u_1} c'_i$. Hence Eq. (2) is formulated as

$$\sum_{i=-1}^{i=u_1} c'_i < \sum_{i=-1}^{i=u_1} c_i^* \tag{2}$$

   (b) To avoid buffer underflow at time instance t_{u_1}, $G'(u_1) \ge G^*(u_1)$.
   (c) Let t_x represent a time instance between time instances t_{-1} and t_{u_1}. We can now rewrite $G'(u_1) \ge G^*(u_1)$ and by the transposition of Eq. (1), we formulate Eq. (3) as

$$\sum_{i=x+1}^{i=u_1} c'_i \ge [G^*(x) - G'(x)] + \sum_{i=x+1}^{i=u_1} [(r_i^* - r'_i)T_i] + c_{u_1}^* \tag{3}$$

   - Assume that time instance t_x is the last buffer overflow that occurs between time instances t_{-1} and t_{u_1}; hence, $G^*(x) - G'(x) \ge 0$. Also, from the computation of the OC algorithm, it appears that $\sum_{i=x+1}^{i=u_1} [(r_i^* - r'_i)T_i] \ge 0$ and $\sum_{i=-1}^{i=u_1} c_i^* = c_{u_1}^*$. Consequently, Eq. (4) can be derived as follows:

$$\sum_{i=-1}^{i=u_1} c'_{u_1} \ge \sum_{i=-1}^{i=u_1} c_{u_i}^* \tag{4}$$

   - If the buffer overflow does not occur between time instances t_{-1} and t_{u_1}, we set $t_x = t_{-1}$ and Eq. (4) holds.

   (d) Because Eq. (4) violates Eq. (2), we conclude that this new algorithm does not exist and $\sum_{i=-1}^{i=u_1} c_i^*$ is minimal.

(2) *Hypothesis*. Given that $k=m-1$, the cumulative cached video data $\sum_{i=-1}^{i=u_{m-1}} c_i^*$ is minimal.

(3) Given that $k=m$, we prove that $\sum_{i=-1}^{i=u_m} c_i^*$ is minimal by contradiction.
   (a) We assume that there is a new algorithm caching less content than the OC algorithm from time instances t_{-1} to t_{u_m}. Hence Eq. (5) is formulated as

$$\sum_{i=-1}^{i=u_m} c'_i < \sum_{i=-1}^{i=u_m} c_i^* \tag{5}$$

   (b) To avoid the buffer underflow at time instance t_{u_m}, $G'(u_m) \ge G^*(u_m)$.
   (c) Let t_x represent a time instance between time instances $t_{u_{m-1}}$ and t_{u_m}. We can now rewrite $G'(u_m) \ge G^*(u_m)$ and by transposition of Eq. (1), formulate Eq. (6).

$$\sum_{i=x+1}^{i=u_m} c'_i \ge [G^*(x) - G'(x)] + \sum_{i=x+1}^{i=u_m} [(r_i^* - r'_i)T_i] + c_{u_m}^* \tag{6}$$

   - Assume that t_x is the last buffer overflow that occurs between time instances $t_{u_{m-1}}$ and t_{u_m}; hence, $G^*(x) - G'(x) \ge 0$. Also, from the computation in the OC algorithm, it appears that $\sum_{i=x+1}^{i=u_m} [(r_i^* - r'_i)T_i] \ge 0$ and $\sum_{i=x+1}^{i=u_m} c'_i \ge c_{u_m}^*$ is derived. From Hypothesis 2, Eq. (7) can be derived as follows:

$$\sum_{i=-1}^{i=u_m} c'_i \ge \sum_{i=-1}^{i=u_m} c_i^* \tag{7}$$

   - If the buffer overflow does not occur between time instances $t_{u_{m-1}}$ and t_{u_m}, we set $t_x = t_{u_{m-1}}$. We can now transpose Eq. (6) and rewrite this equation as follows: $\sum_{i=-1}^{i=u_m} c'_i \ge \sum_{i=0}^{i=u_m} [(r_i^* - r'_i)T_i] + \sum_{i=-1}^{i=u_m} c_i^*$. According to the computation in the OC algorithm, it appears that $\sum_{i=0}^{i=u_m} [(r_i^* - r'_i)T_i] \ge 0$, so Eq. (7) holds.

   (d) Because Eq. (7) violates Eq. (5), we conclude that this new algorithm does not exist and $\sum_{i=-1}^{i=u_m} c_i^*$ is minimal.

(4) Finally, we conclude that the proxy cache computed by the OC algorithm is smaller than, or equal to, that computed by other algorithms. □

4. Experiment results

We now present the results of simulations conducted to test the effectiveness of the proposed approach and compare its performance with that of other methods. Using several benchmark videos, we test the OC algorithm, the conventional CC algorithm, and CAS algorithm. The encoding parameters of the videos and the input parameters used in our experiments are described in Table 1. Meanwhile, the statistics of four video streams, namely,

Table 1
Parameters used in our experiments

Parameters	Values	Parameters	Values
Encoder inputs	384×288	Frame rate	24
Quantizer	$I=10$, $P=14$, $B=18$	Startup latency	1 s
Encoding pattern	IBBPBBPBBP-BB	Client buffer	200 kB

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
James Bond	115.91179	596.73	29.86	2.97	3.14
News	73.23109	359.945	23.18	1.87	2.38

video size, average video bit rate, maximal frame size, minimal frame size, and frame size variance, used in our experiments are presented in Table 2. The experimental results are evaluated according to the following three performance indices:

(1) The proxy cache requirement $=(|C|/|V|)\times 100\%$.
(2) The WAN bandwidth utilization $=\left(\sum_{i=-1}^{i=n-1}[(f_i-c_i)/T_i]/\left[r_{\mathrm{WAN}}\sum_{i=-1}^{i=n-1}T_i\right]\right)\times 100\%$.
(3) The requirement of WAN bandwidth $=r_{\mathrm{WAN}}/(|C|/|V|\times 100\%)$.

4.1. The proxy cache requirement

Usually, a video server for on-demand services stores a huge number of videos. If the cache of each video is small, the total cache required to build a video proxy will be dramatically reduced. With an easy-to-install, low-cache video proxy, service scalability will be significantly improved by constructing video proxies. Given the same resources, we show by experiment which algorithm caches the smallest portion of a video in the video proxy, while guaranteeing video playback quality. We present the relation between the cached percentage of each benchmark video and the variation of the allocated WAN bandwidth. Based on the average bit rate of each benchmark video, the range of bandwidth variation is ± 200 kbps. The different storage requirements of the benchmark videos computed by the CC, CAS, and OC algorithms are presented in Fig. 5.

When the allocated WAN bandwidth increases, the storage requirement computed by the CC, CAS and OC algorithms decreases. In experiments on the four benchmark videos, we show that, on average, the OC algorithm reduces the storage requirement of the CC algorithm by more than 30%, and that of the CAS algorithm by more than and 10%, if we stream each benchmark video with its average bit rate. Additionally, the decreasing slope of experiment curve computed by the OC algorithm is sharper than those computed by the CC and CAS algorithms. The larger the frame variation (as shown in Fig. 5(c) and (d)), the later the resulting curves will meet. Hence, the OC algorithm reduces the storage even further when the allocated WAN bandwidth is more than sufficient, particularly for a video with large frame size variations.

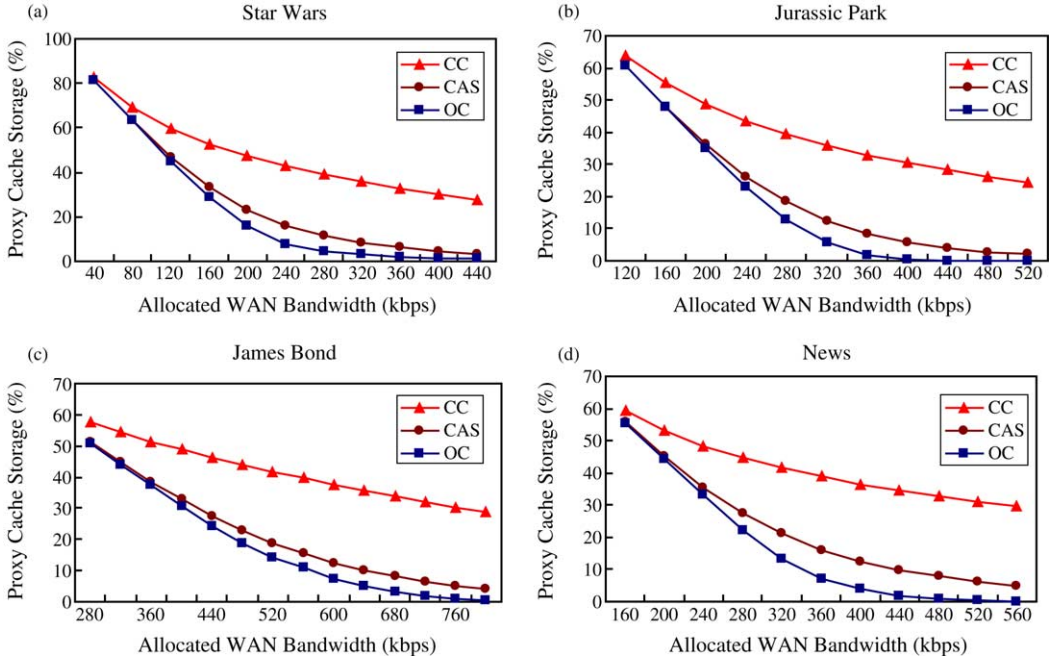

Fig. 5. Experimental results of the proxy storage requirements. (a) Star Wars, (b) Jurassic Park, (c) James Bond, (d) News.

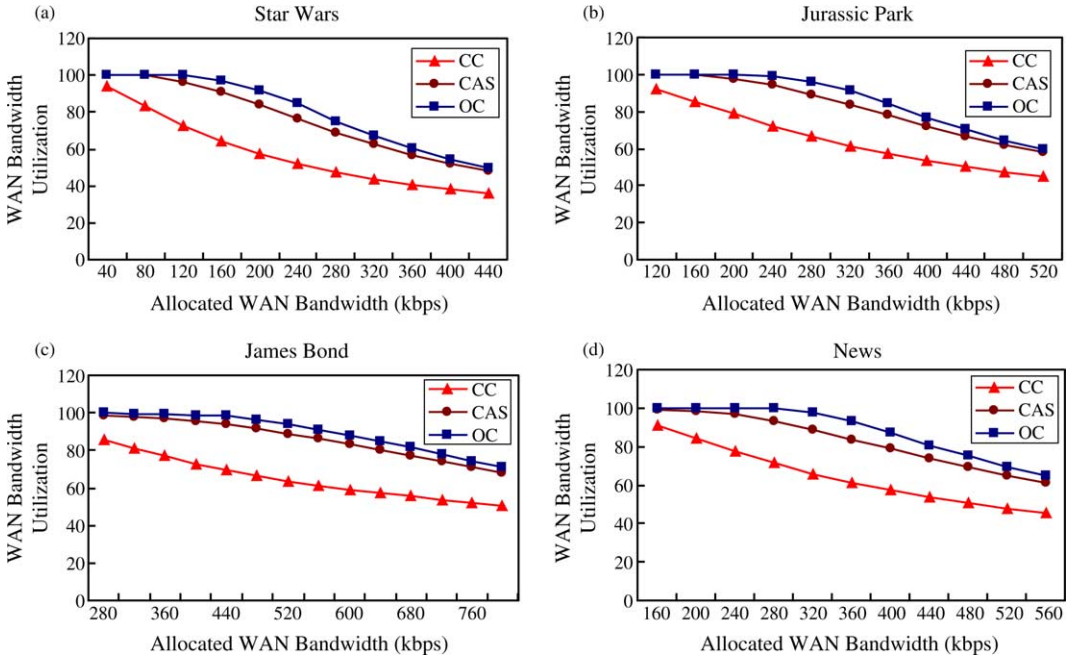

Fig. 6. Experimental results of allocated WAN bandwidth utilization. (a) Star Wars, (b) Jurassic Park, (c) James Bond, (d) News.

4.2. The utilization of allocated WAN bandwidth

Currently, the cost of using WAN bandwidth is high compared to that of using bandwidth in LAN. A well-designed, on-demand system must fully utilize the allocated WAN bandwidth at all times. In a distributed video-streaming system, high bandwidth utilization implies that many more video requests can be served simultaneously. In our experiment, we present the relation between the percentage of bandwidth utilization and the allocated WAN bandwidth and use the CC, CAS, and OC algorithms to compute the cached video data in the video proxy. By simulation, we stream the four benchmark videos with a different streaming schedule computed by each algorithm.

The experimental results presented in Fig. 6 show the utilization of the allocated WAN bandwidth for streaming the different benchmark videos. Based on the average bit rate of each video, the range of WAN bandwidth variation is $\pm$ 200 kbps. According to this result, the WAN bandwidth

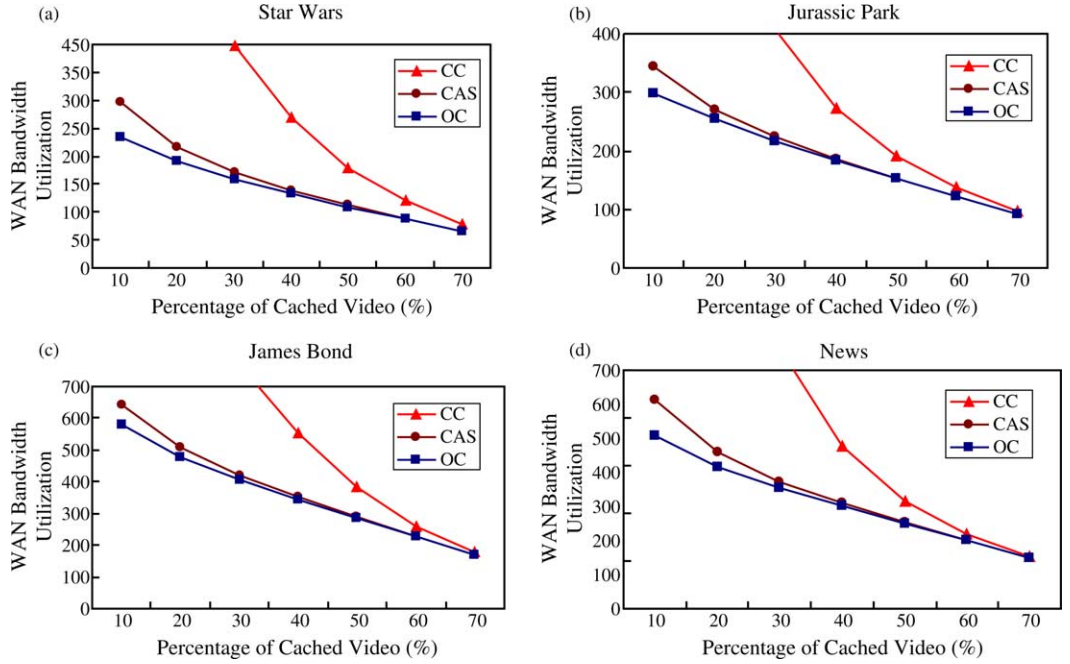

Fig. 7. Experimental results of the allocated WAN bandwidth requirement. (a) Star Wars, (b) Jurassic Park, (c) James Bond, (d) News.

utilization percentage decreases as the allocated WAN bandwidth increases. This indicates that the higher the allocated WAN bandwidth, the more waste of bandwidth there is. Therefore, in our simulation, we stream each video with its average bit rate.

Because of frame size variation, the bandwidth utilization of the CC algorithm without considering client buffer control decreases rapidly. Also, because the buffer overflow occurs in streaming a video, bandwidth utilization cannot reach 100% all the time by using the CAS and OC algorithms. However, the OC algorithm achieves, on average, 35% more utilization of the allocated WAN bandwidth than the CC algorithm and 11% more than the CAS algorithm, if each benchmark video is streamed with its average bit rate. Therefore, the OC algorithm is better at utilizing the allocated WAN bandwidth than the CC and CAS algorithms

4.3. The requirement of WAN bandwidth

In this experiment, we present the relation between the WAN bandwidth requirement and the percentage of video cached. It has been observed that the WAN bandwidth requirement decreases as the video cache percentage increases. In Fig. 7, the curve is sharper when the percentage of cached video is low. This indicates that the smaller the cache, the more effective the bandwidth reduction. Therefore, we cache the smallest amount of video content in the video proxy, subject to guaranteed video playback quality.

In Fig. 7, we present the WAN bandwidth requirement computed by the CC, CAS, and OC algorithms when the proxy storage increases. Compared with the CC algorithm, the OC algorithm can, on average, reduce the allocated WAN bandwidth requirement by more than 50% when the cached data of a video is less than 30%. Additionally, compared with the CAS algorithm, the OC algorithm can, on average, reduce the allocated WAN bandwidth by more than 15% when the cached data of a video is less than 15%.

5. Conclusion

In this paper, we propose a one-pass Optimal Cache (OC) algorithm for Video Staging and prove theoretically that it is an optimal algorithm. Based on the experiment results on several benchmark videos, we believe that the OC algorithm is the best algorithm for handling the Video Staging Mechanism. In terms of the video cache size, the bandwidth utilization, and the WAN bandwidth requirement, the OC algorithm is more effective than the conventional CC and CAS algorithms.

According to the experiment results, the improvement between the CAS and OC algorithm is less than that between the CC and OC algorithm. However, the CC and OC algorithms are one-pass algorithms, whereas the CAS algorithm is a two-pass algorithm. The video streaming schedule and client buffer control computed by the CAS algorithm are difficult to handle. Therefore, the OC algorithm is a more efficient approach for handling and distributing a video proxy in the delivery of high quality video streaming services.

References

[1] S.-H. Chang, R.-I. Chang, J.-M. Ho, Y.-J. Oyang, PSC: a priority selected cache algorithm for streaming video over Internet, Proceedings of the IEEE International Conference on Networks (ICON) (2003).

[2] S.-H. Chang, R.-I. Chang, J.-M. Ho, Y.-J. Oyang, An effective approach to video staging in streaming applications, Proceedings of the IEEE Globe Communication Conference (GLOBECOM) (2002).

[3] S.-H. Chang, R.-I. Chang, J.-M. Ho, Y.-J. Oyang, OC: an optimal cache algorithm for video staging, Proceedings of the IEEE International Conference on Networking (ICN) (2002).

[4] R.-I. Chang, M.C. Chen, M.T. Ko, J.M. Ho, Schedulable region for VBR media transmission with optimal resource allocation and utilization, Information Sciences (2002).

[5] K. Zhu, Y. Zhuang, Y. Viniotis, Achieving end-to-end delay bounds by EDF scheduling without traffic shaping, Proceedings of IEEE Conference on Computer Communications (INFOCOM) (2001).

[6] Z.-L. Zhang, Y. Wang, D.H.C. Du, D. Su, Video staging: a proxy server based approach to end-to-end video delivery over wide-area-networks, IEEE/ACM Transaction on Networking (2000).

[7] W.-H. Ma, D.H.C. Du, Reducing bandwidth requirement for delivering video over wide area network with proxy server, Proceedings of the IEEE International Conference on Multimedia and Expo (ICME) (2000).

[8] Z. Miao, A. Ortega, Proxy caching for efficient services over the Internet, Proceedings of the Ninth International Packet Video Workshop, (1999).

[9] S. Sen, J. Rexford, D. Towsley, Proxy prefix caching for multimedia streams, Proceedings of IEEE Conference on Computer Communications (INFOCOM) (1999).

[10] R.-I. Chang, M.C. Chen, J.M. Ho, M.T. Ko, Characterizing the minimal required resources for admission control of pre-recorded VBR video transmission by an O($n \log n$) algorithm, Proceedings of International Conference on Computer Communications and Networks (ICCCN) (1998).

[11] I. Kim, H.Y. Yeom, J. Lee, Analysis of buffer replacement policies for WWW proxy, Proceedings of the 20th International Conference on Information Networking, (1998).

[12] W.-C. Feng, J. Rexford, A comparison of bandwidth smoothing techniques for the transmission of prerecorded compressed video, Proceedings of IEEE Conference on Computer Communications (INFOCOM) (1997).

[13] R.-I. Chang, M. Chen, M.T. Ko, J.-M. Ho, Designing the on-off CBR transmission schedule for jitter-free VBR media playback in real-time networks, Proceedings of the IEEE International Conference on Real-Time and Embedded Computing Systems and Applications (RTCSA) (1997).

[14] J. Rexford, S. Sen, D. Towsley, Online smoothing for live, variable-bit-rate video, Proceedings of the International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV) (1997).

[15] M. Grossglauser, S. Keshav, On CBR service, Proceedings of IEEE Conference on Computer Communications (INFOCOM) (1996).

[16] J.M. McManus, K.W. Ross, Video on demand over ATM: constant-rate transmission and transport, Proceedings of IEEE Conference on Computer Communications (INFOCOM) (1996).

[17] J. Salehi, Z.-L. Zhang, J. Kurose, D. Towsley, Supporting stored video: reducing rate variability and end-to-end resource requirements through optimal smoothing, Proceedings of the ACM SIGMETRICS (1996).

[18] A.R. Reibman, A.W. Berger, Traffic descriptors for VBR video teleconferencing over ATM networks, IEEE/ACM Transactions on Networking June (1995).
[19] M. Grossglauser, S. Keshav, D. Tse, RCBR: a simple and efficient service for multiple time-scale traffic, Proceedings of ACM SIGCOMM August (1995).
[20] H. Zhang, E.W. Knightly, A new method to support delay-sensitive VBR video in packet-switched networks, Proceedings of International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV) (1995).
[21] W. Feng, S. Sechrest, Smoothing and buffering for delivery of prerecorded compressed video, Proceedings of IS&T/SPIE MMCN (1995).
[22] E.W. Knightly, D.E. Wrege, J. Liebeherr, H. Zhang, Fundamental limits and tradeoffs of providing deterministic guarantees to VBR video traffic, Proceedings of ACM SIGMETRICS (1995).
[23] M. Garrett, W. Willinger, Analysis, modeling and generation of self-similar VBR video traffic, Proceedings of ACM SIGCOMM August (1994) 269–280.
[24] E. Chang, A. Zakhor, Scalable video data placement on parallel disk arrays, Proceedings of IS&T/SPIE Symposium on Electronic Imaging Science and Technology (1994).
[25] S.S. Lam, S. Chow, D.K.Y. Yau, An algorithm for lossless smoothing of MPEG video, Proceedings of ACM SIGCOMM (1994).
[26] T. Ott, T.V. Lakshman, A. Tabatabai, A scheme for smoothing delay-sensitive traffic offered to ATM networks, Proceedings of IEEE Conference on Computer Communications (INFOCOM) (1992).
[27] M.C. Chuah, R.L. Cruz, Approximate analysis of average performance of (sigma, rho) regulators, Proceedings of IEEE Conference on Computer Communications (INFOCOM) (1990).
[28] D. Ferrari, Client requirements for real-time communication services, IEEE Network Magazine November (1990).
[29] F.P. Preparata, M.I. Shamos, Computational geometry: an introduction,, third ed. T&M Computer Science, (1985).
[30] http://www-info3.informatik.uni-wuerzburg.de/MPEG/traces/

Further Reading

[1] R.-I. Chang, M.C. Chen, J.-M. Ho, M.T. Ko, An effective and efficient traffic-smoothing scheme for delivery of online VBR media streams, Proceedings of IEEE Conference on Computer Communications (INFOCOM) (1999).

Shin-Hung Chang received the B.S. degree in Computer Science and Information Engineering Department from Fu Jen Catholic University, Taiwan, in 1996 and the M.S. degree in Computer Science and Information Engineering Department from National Taiwan University, Taiwan, in 1998. He received Ph.D. degree from Computer Science and Information Engineering Department, National Taiwan University, Taipei Taiwan in 2005. His research interests include audio/video codec, media streaming, media relay proxy, network planning, and peer-to-peer (P2P) applications.

Ray-I Chang received his Ph.D. degree in Electrical Engineering and Computer Science from National Chiao Tung University in 1996. Then, he joined Institute of Information Science, Academia Sinica, as a Postdoctoral Fellow in Computer Systems and Communications Laboratory (CSCL). In 2002, he joined Department of Information Management, National Central University, as a leader of Multimedia Networking Laboratory. Now he is an Assistant Professor of the Department of Engineering Science and Ocean Engineering, National Taiwan University. He is a member of IEEE.

Jan-Ming Ho received the B.S. degree in Electrical Engineering from National Cheng Kung University, Taiwan, in 1978 and the M.S. degree from Institute of Electronics at National Chiao Tung University, Taiwan, in 1980. He received the Ph.D. degree in electrical engineering and computer science from Northwestern University, USA, in 1989. He joined the Institute of Information Science, Academia Sinica, Taipei Taiwan, as a associate research fellow in 1989 and was promoted to research fellow in 1994. He visited IBM T.J. Watson Research Center in the summers of 1987 and 1988, Leonardo Fibonacci Institute for the Foundations of Computer Science, Italy, in summer the of 1992, and Dagstuhl-Seminar on "Combinatorial Methods for Integrated Circuit Design," IBFI-Geschaftsstelle, Schlo£ Dagstuhl, Fachbereich Informatik, Bau 36, Universitat des Saarlandes, Germany, in October 1993. He is a member of the IEEE and ACM. His research interests target at the integration of theoretical and application-oriented research, including mobile computing, environment for management and presentation of digital archive, management, retrieval, and classification of Web documents, continuous video streaming and distribution, video conferencing, real-time operating systems with applications to continuous media systems, computational geometry, combinatorial optimization, VLSI design algorithms, and implementation and testing of VLSI algorithms on real designs. He is associate editor of IEEE Transactions on Multimedia. He was program chair of the Symposium on Real-Time Media Systems, Taipei, 1994–1998, general co-chair of the International Symposium on Multi-Technology Information Processing, 1997, and general co-chair of IEEE RTAS 2001. He was also a steering committee member of the VLSI Design/CAD Symposium, and program committee member of several previous conferences including ICDCS 1999, and IEEE Workshop on Dependable and Real-Time E-Commerce Systems (DARE'98), etc. In domestic activities, he is program chair of the Digital Archive Task Force Conference, the First Workshop on Digital Archive Technology, a steering committee member of the 14th VLSI Design/CAD Symposium and the International Conference on Open Source 2002, and is also a program committee member of the 13th Workshop on Object-Oriented Technology and Applications, the Eighth Workshop on Mobile Computing, the 2001 Summer Institute on Bio-Informatics, Workshop on Information Society and Digital Divide, the 2002 International Conference on Digital Archive Technologies (ICDAT2002), the APEC Workshop on e-Learning and Digital Archives (APEC2002), and the 2003 Workshop on e-Commerce, e-Business, and e-Service (EEE'03).

Yen-Jen Oyang received the B.S. degree in Computer Science and Information Engineering from National Taiwan University, Taipei Taiwan, in 1982, the M.S. degree in Computer Science from the California Institute of Technology, USA, in 1984, and the Ph.D. degree in Electrical Engineering from Stanford University, USA, in 1988. He is currently a Professor in the Department of Computer Science and Information Engineering, National Taiwan University, Taipei Taiwan. His research interests include data mining/machine learning, video on demand, and disk storage scheduling.