Technical Report

OC: An Optimal Cache Algorithm for Video Staging

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.
E-mail: viola@iis.sinica.edu.tw. Phone: 886-2-27883799 ext. 1656. Fax: 886-2-27824814.

ABSTRACT

The video content for on-demand services is generally stored and streamed in a compressed format. The compressed video is naturally with the VBR (variable-bit-rate) property and the stream traffic is highly burst. Subject to the QoS-guaranteed playback, the WAN bandwidth needs to allocate the video's peak bit rate if there is no client buffer for regulating the video delivery [7][8][9][10][11]. To reduce the requirement of WAN bandwidth, *Video Staging* [2], first proposed by Zhang *et al.*, caches parts of a video content in the video proxy closed to clients. Therefore, the video can be streamed across the WAN with CBR (constant-bit-rate) services and its bandwidth requirement is significantly reduced.

In this paper, we propose an *Optimal Caching (OC)* algorithm to handle the Video Staging problem. We also prove that the cache storage computed by our OC algorithm is minimal. Relatively, if the same cache size is given, OC requires less WAN bandwidth than [2] does to provide streaming services. By doing experiments on several benchmark videos [12], we show that the OC algorithm can reduce the cache storage requirement by over 30% while comparing to [2]. With the same proxy cache storage of [2], we can reduce the WAN bandwidth requirement with more than 50%. Additionally, the WAN bandwidth utilization can also be increased by over 30%.

Index terms – Video Staging, VBR (variable-bit-rate), CBR (constant-bit-rate), Video Proxy, and Optimal Caching (OC).

1. INTRODUCTION

At present, most of customers who purchase digital products (such as CD, VCD, DVD, *etc.*) still need to receive them by mail (or similar delivery) services. It is not convenient. With advances in network technologies, streaming continuous media (such as video) across networks has become practical. Instead of receiving these digital products from delivery services, customers can enjoy the video content on-line. However, with the rapid growth of streaming services, customers are becoming more and more sensitive to the service quality. Poor-quality videos with low bit rate have not satisfied current customers. The video content for on-demand services is generally stored and streamed in a compressed format. The compressed video is naturally with the VBR (variable-bit-rate) property and the stream traffic is highly burst. Because of it's huge size and critical bandwidth constraint, it is the most challenging problem to stream high-quality videos across a variety of networks; in particular, across the Internet.

Nowadays Internet consists of lots ISPs (Internet Service Providers). These ISPs interconnect with each other by the backbone network. The backbone network is generally referred to as the wide area network (WAN). In addition, each end-user accesses the Internet through an ISP via the so-called access network. Typical examples of access networks are PSTN, XDSL, ISDN, and LAN. An illustration of the Internet heterogeneity is presented in Figure 1. The backbone WAN is usually shared by large number of clients and it is more difficult for service providers to guarantee the delivering quality. Therefore, it is generally more costly to deliver contents across the backbone WAN than across the access network.

Taking the Internet heterogeneity into account, proxy technologies have been widely used for improving the service quality and content distribution as shown in Figure 1. Take web services for example. Web contents (e.g. hypertext and image data) are cached in the web proxy closed to clients and end-users can retrieve them from the web proxy instead via the high-speed local access link. By reducing content retrievals from the remote web server, the WAN bandwidth requirement decreases and the remote web server traffic is off-loaded [6]. Comparing with the small size of web content, however, the size of video is usually huge. Caching an entire video to eliminate the WAN bandwidth requirement is unrealistic. It's impractical to directly switch the web proxy to handle video services. For different purposes, many proxies for handling the video are designed by several groups of researchers [1][2][3][4][5]. Video Staging, first proposed by

Zhang et al., caches only a pre-selected portion of a remote video into the video proxy [2]. In this paper, we focus on proposing an optimal approach to handling Video Staging.

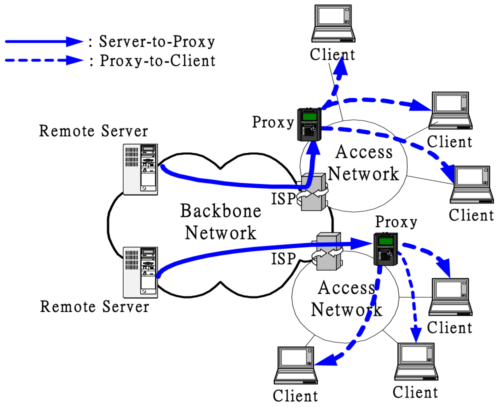


Figure 1. The heterogeneity of Internet with proxies.

In [2], they presented an algorithm to handle Video Staging. In this paper, we refer to their algorithm as the 〝Cut-off Caching (CC)〞 algorithm. The main idea under this algorithm is sequentially comparing each video frame with a given cut-off rate (the WAN available 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 excessive portion of this frame and pre-stores it in the video proxy. An illustration of the CC algorithm is presented in Figure 2.

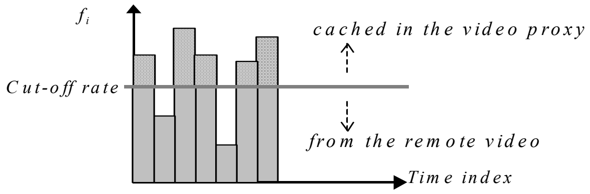
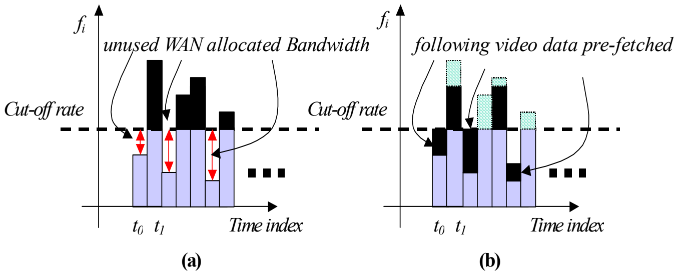


Figure 2. Illustration of the CC algorithm.

The CC algorithm is a good design in system implementation. However, the compressed video usually has large size variation between video frames. One frame size may be very small and it will lead not to fully utilize the WAN available bandwidth as shown in Figure 3 (a). If the unused WAN available bandwidth can be used to pre-fetch following video data, the cached storage requirement in the video proxy will be reduced even more as shown in Figure 3 (b).



(a) (b)

Figure 3. Illustration of using unused WAN bandwidth to pre-fetch video data.

Given a video content and specific resources, including the WAN available bandwidth, client buffer, and startup latency, we propose an Optimal Caching (OC) algorithm to handle Video Staging. The time complexity of the OC algorithm is also linear complexity ($O(n)$ where n is the number of frames), same as the CC algorithm. In this paper, we also prove that the cache storage computed by our OC algorithm is minimal. Relatively, if the same cache size is given, OC requires less WAN bandwidth than [2] does to provide streaming services. By doing experiments on several benchmark videos [12], we show that the OC algorithm can reduce the cache storage requirement by over 30% while comparing to [2]. With the same proxy cache storage of [2], we can reduce the WAN bandwidth requirement with more than 50%. Additionally, the WAN bandwidth utilization can also be increased by over 30%.

The rest of this paper is organized as follows. The optimal 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, we state the conclusions of this paper in section 5.

2. 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 clients for playback. The time period from receiving to playing the video at client is called the startup latency *L*. The client buffer size is denoted by B. 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 instance 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 instance t_i and t_{i+1}. r_{WAN} represents the WAN available bandwidth. For simplify resource managements, we assume that network services with the minimal delay and no loss is used for streaming videos across networks in this paper. Additionally, the network available bandwidth under the access network is assumed to be ample. When a video proxy is installed, let $C = \{c_i \mid -1 \le i < n, c_i \ge 0\}$ represent a sequence of cached data, where c_i indicates the cached size of the *i-th* video frame and the total cumulative cached size is denoted by $|C| = \sum_{i=-1}^{i=n-1} c_i$. The Cache Minimization (CM) problem is formulated as follows:

Problem: *Given a video, the CM problem is defined to determine a subset* C *of this video pre-caching into the video proxy such that the cumulative cached size* $|C|$ *is minimal while the same startup latency, client buffer size, and WAN available bandwidth.*

3. OPTIMAL CACHING (OC) ALGORITHM

For QoS-guaranteed streaming services, frame f_i should be available at the client for display at time instance t_i. Except consuming f_i for playback, the client also receives at most $r_{WAN} \times T_i$ (bits) video data from the remote video server simultaneously. Let $\{b_i \mid -1 \le i < n\}$ represent a sequence of the buffer occupancy and the initial value $b_{-1} = 0$. The buffer occupancy represents the data aggregation that consists of the pre-fetched video at the client buffer and new arrival video from the remote video server. Therefore, b_i can be computed by $min\{B, b_{i-1} + (r_i \times T_i) - f_{i-1}\}$ as shown in Figure 4.

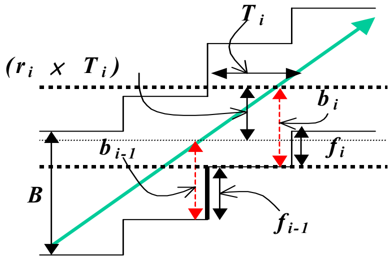


Figure 4. Without considering video proxy installed, the client buffer occupancy b_i at time instance t_i can be computed by $min\{B, b_{i-1} + (r_i \times T_i) - f_{i-1}\}$.

The buffer occupancy b_i must not be smaller than f_i (the frame required for playback at time instance t_i). Unfortunately, the WAN available bandwidth, r_{WAN}, might not be large enough and cause buffer underflow (b_i is smaller than f_i). Therefore, the main idea behind the OC algorithm is scheduling client to retrieve the excessive part of the *i-th* frame, $f_i(c_i)$, from the closed video proxy at time instance t_i, where $c_i = f_i - b_i$, as shown in Figure 5.

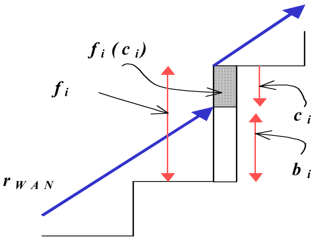


Figure 5. Given the WAN available bandwidth r_{WAN}, the cached portion of the video frame f_i is denoted by $f_i(c_i)$, where c_i is the cached size.

The OC algorithm is presented in detail as follows:

Algorithm: Optimal Caching (OC) Algorithm

```
//Given a video $V = \{f_i > 0 \mid -1 \le i < n, f_{-1} = 0\}$, where n is the number of frames;
//$b_i$ is the client buffer occupancy at time instance $t_i$.
//B indicates the size of client buffer;
//Given the WAN available bandwidth $r_{WAN}$;
i = -1;  $b_i$=0;
repeat
{$i = i + 1$;
    $r_i = r_{WAN}$;
    $b_i = min\{B, b_{i-1} + (r_i \times T_i) - f_{i-1}\}$;
    if ($f_i \le b_i$)
    {   $c_i$ = 0;
        if ($b_i$ == B) then $r_i = (B - b_{i-1} + f_{i-1})/T_i$ ;}   /*buffer is overflow*/
    else                                                    /* buffer is underflow*/
    {   $c_i = f_i - b_i$;  $b_i = b_i + c_i = f_i$;
        Cache $f_i(c_i)$ into the video proxy;}}
until (i > (n-1));
```

Theorem: *The proxy cache storage computed by the OC algorithm is smaller than or the same as that computed by other algorithms while the same startup latency, client buffer size, and WAN available bandwidth.*

Proof:

(1) Let $\{t_{u_1}, t_{u_2}, \ldots, t_{u_m}\}$ represent a sequence of time instance that buffer underflow occurs during the computation of the OC algorithm. $\{c_i^* \mid -1 \le i < n, c_i^* \ge 0\}$ indicates a sequence of cached data computed by the OC algorithm. We use Mathematical Induction to prove that at any time instance $\{t_{u_k} \mid 1 \le k \le m\}$, the size of the cumulative cached video computed by the OC algorithm, $\sum_{i=-1}^{i=u_k} c_i^*$, is minimal.

(2) For a clear formulation of the proof, we define the cumulative transmission function $G(i) = G(i-1) + r_{i-1} \times T_{i-1} + c_i$ as the amount of data sent by the remote video server and retrieved from the video proxy before the time instance t_i.

(3) Given that $k = 1$, we prove $\sum_{i=-1}^{i=u_1} c_i^*$ is minimal by contradiction.

(a) We assume that there is a new algorithm caching less than the OC algorithm from the time instance t_{-1} to t_{u_1}. Let $\{c_i' \mid -1 \le i < n, c_i' \ge 0\}$ indicate a sequence of cached data computed by this new algorithm, and the cumulative cached video data, from the time instance t_{-1} to t_{u_1}, is denoted by $\sum_{i=-1}^{i=u_1} c_i'$. Hence the equation (I) is formulated.

$$\sum_{i=-1}^{i=u_1} c_i' < \sum_{i=-1}^{i=u_1} c_i^* \ldots\text{(I)}$$

(b) To guarantee the playback quality and avoid the buffer underflow, it appears that $G'(u_1) \ge G^*(u_1)$ at the time instance t_{u_1} is true.

(c) Let t_x represent a time instance between the time instance t_{-1} and t_{u_1}. We now rewrite $G'(u_1) \ge G^*(u_1)$ as follows: $G'(x) + \sum_{i=x}^{i=u_1-1}(r_i' \times T_i) + \sum_{i=x+1}^{i=u_1} c_i' \ge G^*(x) + \sum_{i=x}^{i=u_1-1}(r_i^* \times T_i) + c_{u_1}^*$.

(d) By transposition, we formulate equation (II).

$$\sum_{i=x+1}^{i=u_1} c_i' \ge [G^*(x) - G'(x)] + \sum_{i=x}^{i=u_1-1}[(r_i^* - r_i') \times T_i] + c_{u_1}^* \ldots\text{(II)}$$

(e) Assume that the time instance t_x is the last one buffer overflow that occurs between the time instance t_{-1} and t_{u_1}. According to the computation of the OC algorithm, it appears that $G^*(x) - G'(x) \ge 0$ and $\sum_{i=x}^{i=u_1-1}[(r_i^* - r_i') \times T_i] \ge 0$. From the equation (II), $\sum_{i=x+1}^{i=u_1} c_i' \ge c_{u_1}^*$ is derived. Additionally, according to the computation in the OC algorithm, $\sum_{i=-1}^{i=u_1} c_i^* = c_{u_1}^*$. Hence equation (III) is derived.

$$\sum_{i=-1}^{i=u_1} c_{u_1}' \ge \sum_{i=-1}^{i=u_1} c_{u_1}^* \ldots\text{(III)}$$

(f) If the buffer overflow doesn't occur between the time instance t_{-1} and t_{u_1}, we set $t_x = t_{-1}$ and equation (III) is also hold.

(g) Because equation (III) violates equation (I), we conclude that this new algorithm does not exist and $\sum_{i=-1}^{i=u_1} c_{u_1}^*$ is minimal.

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

(4) Given that $k = m$, we prove $\sum_{i=-1}^{i=u_m} c_i^*$ is minimal by contradiction.

(a) We assume that there is a new algorithm caching less than the OC algorithm from the time instance t_{-1} to t_{u_m}. Hence the equation (IIII) is formulated.

$$\sum_{i=-1}^{i=u_m} c_i' < \sum_{i=-1}^{i=u_m} c_i^* \ldots\text{(IIII)}$$

(b) To guarantee the playback quality and avoid the buffer underflow, it appears that $G'(u_m) \ge G^*(u_m)$ at the time instance t_{u_m} is true.

(c) Let t_x represent a time instance between the time instance $t_{u_{m-1}}$ and t_{u_m}. We now rewrite $G'(u_m) \ge G^*(u_m)$ as follows: $G'(x) + \sum_{i=x}^{i=u_m-1}(r_i' \times T_i) + \sum_{i=x+1}^{i=u_m} c_i' \ge G^*(x) + \sum_{i=x}^{i=u_m-1}(r_i^* \times T_i) + c_{u_m}^*$.

(d) By transposition, equation (V) is formulated.

$$\sum_{i=x+1}^{i=u_m} c_i' \ge [G^*(x) - G'(x)] + \sum_{i=x}^{i=u_m-1}[(r_i^* - r_i') \times T_i] + c_{u_m}^* \ldots\text{(V)}$$

(e) Assume that t_x is the last buffer overflow that occurs between the time instance $t_{u_{m-1}}$ and t_{u_m}. According to the computation in the OC algorithm, it appears that $G^*(x) - G'(x) \ge 0$ and $\sum_{i=x}^{i=u_m-1}[(r_i^* - r_i') \times T_i] \ge 0$. From the equation (V), $\sum_{i=x+1}^{i=u_m} c_i' \ge c_{u_m}^*$ is derived. From hypothesis (3), the following equation (VI) is derived.

$$\sum_{i=-1}^{i=u_m} c_i' \ge \sum_{i=-1}^{i=u_m} c_i^* \ldots\text{(VI)}$$

(h) If the buffer overflow doesn't occur between the time instance $t_{u_{m-1}}$ and t_{u_m}, we set $t_x = t_{u_{m-1}}$. We rewrite equation (V) as follows: $\sum_{i=-1}^{i=u_m} c_i^{'} \geq \sum_{i=-1}^{i=u_{m-1}-1} [(r_i^* - r_i^{'}) \times T_i] + \sum_{i=-1}^{i=u_m} c_i^{*'}$. According to the computation in the OC algorithm, it appears that $\sum_{i=-1}^{i=u_{m-1}-1} [(r_i^* - r_i^{'}) \times T_i] \geq 0$ and equation (VI) is also hold.

(i) Because equation (VI) violates equation (IIII), we conclude that this new algorithm does not exist and $\sum_{i=-1}^{i=u_m} c_{u_1}^{*}$ is minimal.

(5) Finally, we conclude that the cache storage computed by the OC algorithm is smaller than or the same as that computed by other algorithms.

Q.E.D.

4. EXPERIMENT RESULTS

Section 4 presents the results of simulations conducted to test the effectiveness of the proposed approach and compares its performance with that of previous methods. In this section, we test the OC algorithm and the conventional CC algorithm by several benchmark videos. Encoding parameters of benchmark videos and parameters used in our experiments are described in Table 1. Additionally, the statistics of four 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

The experimental results are evaluated according to the following performance indices.

(1) The proxy cache storage requirement
$= (|C| / |V|) \times 100\%$

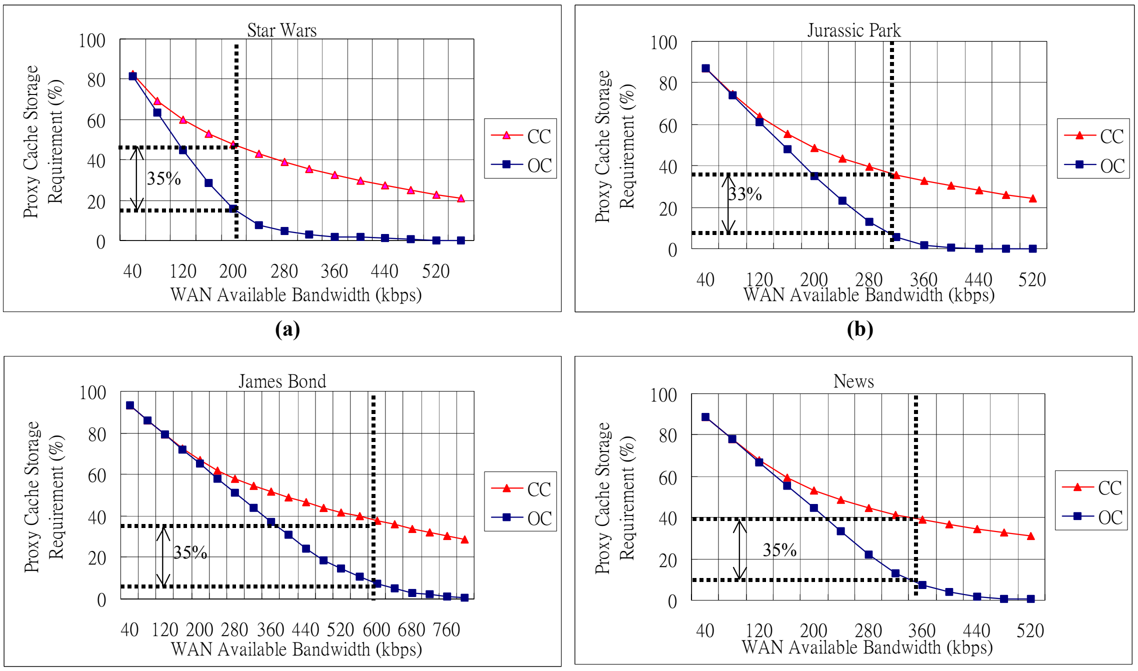
(2) The WAN bandwidth utilization
$= (\sum_{i=-1}^{i<n} [(f_i - c_i)/T_i] / [r_{WAN} \times \sum_{i=-1}^{i<n} T_i]) \times 100\%$

(3) The WAN available bandwidth requirement
$= r_{WAN} / (|C| / |V| \times 100\%)$

4.1 The Proxy Cache Storage Requirement

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 a video as possible in the video proxy. For different benchmark videos, cache storage requirement computed by the CC and OC algorithms are presented in Figure 6.

When the WAN available bandwidth increases, the cache storage requirement computed by the CC and OC algorithms both decrease. Experiments on these four benchmark videos show that the OC algorithm can averagely reduce the cache storage requirement in the video proxy by over 30% smaller than when computed by the CC algorithm, if we stream these benchmark videos with its average bit rate. Additionally, we observed that the decreasing slope computed by OC algorithm is sharper than that computed by the CC algorithm. Hence the OC algorithm can reduce the cache storage even more when the WAN available bandwidth is more sufficient. This improvement is significant.



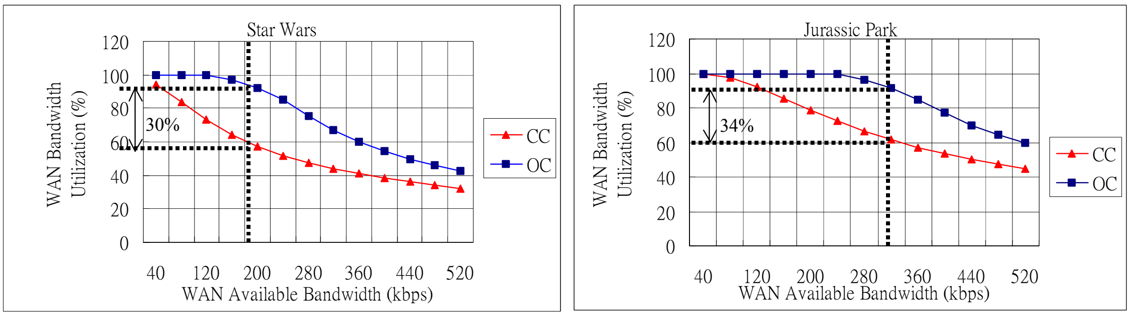
(a) (b)

(c) (d)

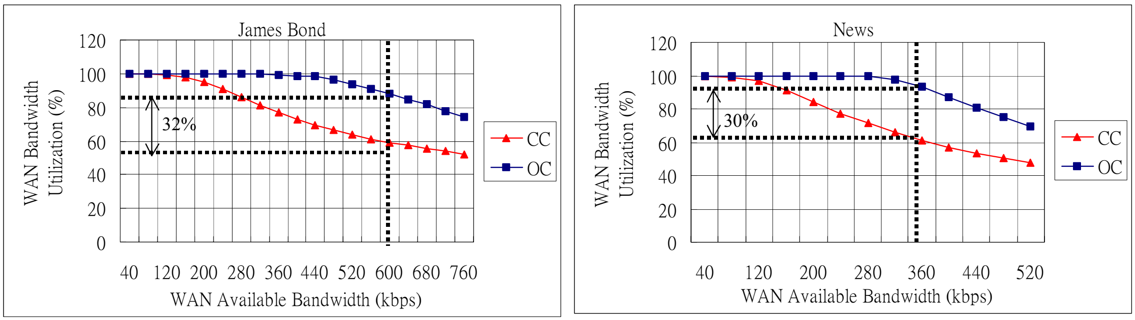
Figure 6. The proxy cache storage requirement.
(a) Star Wars (b) Jurassic Park (c) James Bond (d) News.

4.2 The WAN Bandwidth Utilization

Because the WAN bandwidth is costly, we must utilize it sufficiently at all times. In a distributed video streaming system, high bandwidth utilization implies that lots of video requests can be served at the same time. We use all of the above algorithms to compute the cached data in the video proxy. By simulation, we stream four benchmark videos with the streaming schedule computed by each algorithm distinctly. The experimental results are presented in Figure 7.



(a) (b)



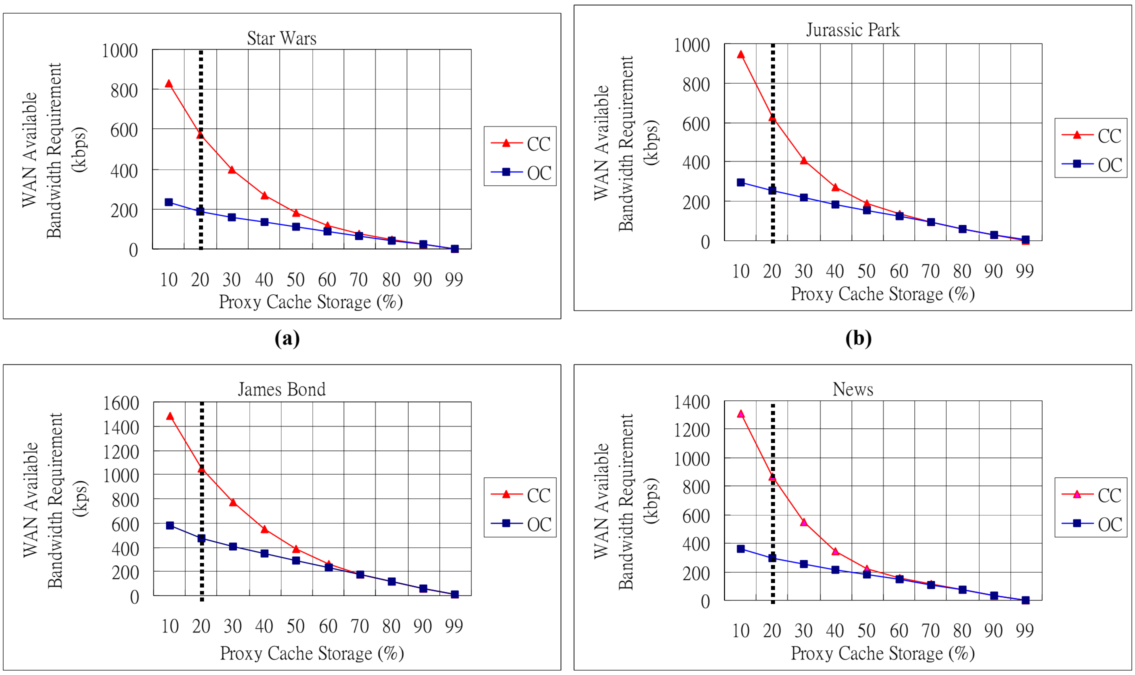
(c) (d)

Figure 7. Experiment results of WAN bandwidth utilization
(a)Star wars (b) Jurassic Park (c) James Bond (d) News.

In Figure 7, the utilization of the WAN available bandwidth is presented while streaming different benchmark videos. By the OC algorithm, experimental results show that the utilization of WAN available bandwidth can averagely increase with more than 30%, if these benchmark videos are streamed with its average bit rate. The improvement is significant.

4.3 The WAN Available Bandwidth Requirement

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 OC algorithm and CC algorithm when the proxy cache storage increases. Experimental results show that the OC algorithm can averagely reduce the WAN available bandwidth requirement with more than 50%, when the cached data of a video is less than 20%. The improvement is dramatic.



(a) (b)

(c) (d)

5. CONCLUSIONS

In this paper, we propose an Optimal Caching (OC) algorithm to handle Video Staging. We also theoretically prove that the cache storage requirement computed by the OC algorithm is minimal. From the basis of experimental results on several benchmark videos, the OC algorithm can averagely reduce the cache storage requirement in the video proxy by more than 30% less than when computed by the CC algorithm, if these benchmark videos are streamed with its average bit rate. By the OC algorithm, experimental results also show that the utilization of the WAN available bandwidth can averagely increase with more than 30%. The improvement is significant. Additionally, experimental results show that the OC algorithm can averagely reduce the WAN available bandwidth requirement with more than 50%, when the cached data of a video is less than 20%.

6. REFERENCES

[1] Ray-I Chang, Shin-Hung Chang, and Jam-Ming Ho, "*OSC: Optimal Selective Caching for video transmission with proxy servers, " in proc. of 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 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, H.Y. Yeom, 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] 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.

[9] 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.

[10] Jennifer Rexford, Subhabrata Sen, and Don Towsley, *"Online Smoothing for live, variable-bit-rate video," in proc. of NOSSDAV*, 1997.

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

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