

논문 2009-46SP-4-11

# 효율적인 비디오 프록시 서버를 위한 캐시 관리 방법

## ( A Cache Management Technique for an Efficient Video Proxy Server )

이 준 표\*, 박 성 한\*\*

( Jun Pyo Lee and Sung Han Park )

## 요 약

사용자와 근거리에서 위치한 비디오 프록시 서버는 자주 요청되는 동영상 데이터들을 저장하고 사용자에게 직접 전송함으로써 초기 전송 지연과 네트워크 트래픽을 효과적으로 감소시킨다. 그러나 비디오 프록시 서버는 원격지의 중앙 비디오 서버에 비해 비교적 제한된 저장 공간을 가진다. 따라서 사용자들이 계속 요청하는 동영상만을 선별하여 저장하도록 하는 방법이 필요하다. 이를 위해 본 논문에서는 비디오 프록시 서버에서의 가상 메모리에 바탕을 둔 가상 캐싱 기법을 제안한다. 제안하는 알고리즘은 사용자가 요청한 동영상 데이터가 비디오 프록시 서버에 존재하지 않는 경우 원격지의 중앙 비디오 서버로부터 요청된 동영상 데이터를 전송받아 사용자에게 전송하고 가상 메모리에 저장한다. 저장된 동영상 데이터는 이후 사용자의 요청이 있는 경우 사용자에게 전송된다. 이때 가상 메모리에 저장된 동영상 데이터는 사용자의 요청의 상태에 따라 가상 메모리로부터 삭제되거나 비디오 프록시 서버에 저장된다. 또한 가상 메모리에서의 단편화를 막기 위하여 가상 메모리를 세그먼트 별로 영역을 구분한다. 실험을 통해 제안하는 방법이 기존의 방법들 보다 높은 적중률을 보이는 동시에 보다 적은 삭제 횟수를 보인다는 것을 확인한다.

## Abstract

Video proxy server which is located near clients can store the frequently requested video data in storage space in order to minimize initial latency and network traffic significantly. However, due to the limited storage space in video proxy server, an appropriate video selection method is needed to store the videos which are frequently requested by users. Thus, we present a virtual caching technique to efficiently store the video in video proxy server. For this purpose, we employ a virtual memory in video proxy server. If the video is requested by user, it is loaded in virtual memory first and then, delivered to the user. A video which is loaded in virtual memory is deleted or moved into the storage space of video proxy server depending on the request condition. In addition, virtual memory is divided into each segment area in order to store the segments efficiently and to avoid the fragmentation. The simulation results show that the proposed method performs better than other methods in terms of the block hit rate and the number of block deletion.

**Keywords :** video proxy server, virtual memory, video streaming, video-on-demand, media segmentation

## I. 서 론

사용자와 근거리에서 위치하는 비디오 프록시 서버(video proxy server)는 사용자들에 의해 요청되어지는 대용량의 동영상 데이터들을 복잡한 인터넷의 중간 경

로를 거치지 않고 다수의 사용자들에게 직접 전달함으로써 데이터의 손실과 초기 전송 지연(initial latency) 문제를 해결한다. 그러나 비디오 프록시 서버는 중앙 비디오 서버(central video server)의 저장 공간에 비해 비교적 제한된 용량을 가진다는 단점이 있다. 따라서 사용자에게 의해 앞으로 사용자에게 의해 계속 요청될 동영상 데이터만을 저장하도록 하는 방법이 필요하다. 이를 위한 연구에서 사용자가 주로 동영상의 시작 부분을 요청하는 접근 패턴을 가지며 비교적 짧은 지속시간을 가진

\* 학생회원, \*\* 정회원, 한양대학교 컴퓨터 공학과  
(Department of Computer Science and Engineering,  
Hanyang University)  
접수일자: 2009년5월26일, 수정완료일: 2009년6월22일

다는 점을 활용하여 사용자가 요청한 동영상의 전반부만을 비디오 프록시 서버에 저장 하는 전반부 캐싱(prefix caching) 기법이 제안되었다<sup>[6]</sup>.

전반부 캐싱 기법은 사용자가 동영상을 요청했을 때 비디오 프록시 서버에 저장 되어있는 해당 동영상의 전반부를 사용자에게 전송하여 초기 지연시간을 최소화하고 이와 동시에 원거리에 위치한 중앙 비디오 서버로부터 후반부를 전송받는다. 이와 같은 전반부 캐싱 기법을 기반으로 한 최근의 연구에서 사용자들에 의해 요청된 데이터의 전체 크기에 따라 정해진 인기도를 이용하는 PPC(Popularity-based Prefix Caching)<sup>[7]</sup> 방법이 제안되었다. 제안된 PPC 방법은 인기도에 따라서 비디오 프록시 서버에 저장될 동영상 데이터의 크기를 결정하고 해당 크기의 앞부분 동영상 데이터를 우선하여 저장한다. 그러나 이 방법은 접근 최근성(recentness)을 전혀 고려하지 않기 때문에 과거에 자주 요청되었던 동영상들로 인해 새롭게 요청되고 있는 동영상 데이터가 비디오 프록시 서버에 저장되지 못하는 문제점이 발생한다. 또 다른 연구에서 미리 설정된 횡수의 사용자 요청 시간 정보를 활용하는 LRU-k 알고리즘<sup>[8-9]</sup>과 사용자의 요청에 따른 우선 순위를 고려하여 비디오 프록시 서버에 동영상의 저장과 삭제를 결정하는 Greedy-Dual 알고리즘<sup>[10]</sup>이 제안되었다. 그러나 이러한 방법들은 접근 빈도수나 접근 최근성을 주로 활용한 방법으로 동영상 데이터의 사용자 요청과 관련된 정보들을 충분히 고려하고 있지 않다. 특히 사용자가 요청한 모든 동영상 데이터들이 비디오 프록시 서버에 저장되므로 한번 저장된 후 계속 요청되지 못하는 동영상 데이터 역시 비디오 프록시 서버에 저장되게 된다. 이는 곧 새로운 동영상 데이터의 저장을 위한 공간 부족 시에 과도한 동영상 데이터의 삭제가 수행되는 원인이 된다.

이러한 문제를 효율적으로 해결하기 위하여 본 논문에서는 사용자가 요청한 동영상 데이터를 중앙 비디오 서버로부터 불러와 곧장 비디오 프록시 서버에 저장하지 않고 가상 메모리에 일단 적재한 후 사용자의 요구가 계속되는 경우 이를 비디오 프록시 서버에 저장하는 가상 캐싱 기법(virtual caching technique)을 제안한다. 제안하는 방법은 사용자가 요청한 동영상 데이터들 중 사용자에게 의해 계속 요청될 데이터들만을 선별하여 비디오 프록시 서버의 저장 공간으로 이동하게 한다. 따라서 사용자들은 비디오 프록시 서버에 저장되어있는 동영상들을 빈번하게 요청하게 되며 비디오 프록시 서

버는 해당 동영상 데이터를 사용자에게 신속하게 전송할 수 있게 됨으로써 원거리에 위치한 중앙 비디오 서버로부터의 데이터 전송으로 인한 과도한 네트워크 트래픽이나 서비스 지연으로 인한 낮은 전송 품질의 문제점을 해결한다. 또한 한 번 요청되어 저장된 후 계속 서비스 되지 않는 동영상 데이터가 비디오 프록시 서버에서 우선 삭제하여 비디오 프록시 서버의 제한된 용량을 효율적으로 관리한다.

제안하는 방법을 위해 중앙 비디오 서버에서 동영상을 여러 개의 세그먼트(segment) 단위로 분할하여 전송하는 Chen 등이 제안한 피라미드식 미디어 분할(media segmentation) 기법<sup>[11]</sup>이 사용된다. 미디어 분할 기법을 사용하면 적은 비용으로 많은 데이터를 신속하게 삭제하거나 저장할 수 있으므로 초기 지연과 네트워크의 트래픽을 효과적으로 줄여 주는 장점이 있다. 또한 본 논문에서는 가상 메모리에서의 내부 단편화를 막기 위해 가상 메모리를 세그먼트 별로 영역을 나누고 분할 기법이 적용된 동영상 세그먼트들을 구분하여 저장한다.

## II. 가상 캐싱 기법

제안하는 가상 메모리를 이용한 비디오 프록시 서버의 관리는 비디오 프록시 서버에 자주 사용되지 않는 동영상 데이터가 저장 공간을 낭비함에 따른 비디오 프록시 서버에서의 블록 대체(block replacement)의 횡수를 줄이고 블록 적응률을 향상시키기 위한 것이다. 이를 위하여 제안하는 알고리즘은 사용자가 요청한 동영상 데이터를 가상 메모리에 저장하고 사용자에게 전송한다. 이때 가상 메모리에 저장된 동영상 데이터는 사용자의 요청의 상태에 따라 가상 메모리로부터 비디오 프록시 서버로 전송된다.

이를 위하여 본 논문에서는 요청된 비디오 세그먼트가 가상 메모리에서 사용자의 요청을 기다리는 시간인 대기 시간을 정의한다. 여기서 요청된 세그먼트의 대기 시간은 해당 세그먼트가 사용자들에 의해 얼마나 자주 요청되었는가에 의해서 결정되며 식 (1)과 같이 주어진다.

$$T_i(k) = T_i(k-1) \times \alpha_i(k) \quad (1)$$

여기서,  $\alpha_i(k)$ 는 임의의 동영상  $i$ 의 세그먼트  $k$ 와 세그먼트  $k-1$ 의 요청 확률의 비율 값으로  $P_i(k)/P_i(k-1)$ 으로 주어진다. 여기서 임의의 동영상의 주어진 세그먼트

의 요청 확률  $P_i(k)$ 는 요청된 동영상의 전체 세그먼트들에 대한 동영상의 해당 세그먼트의 비율이며 다음과 같은 특성을 가진다.

$$P_i(k) = P_i(k|k-1) \leq P_i(k-1) \quad (2)$$

즉, 세그먼트  $k$ 의 요청 확률은 항상 세그먼트  $(k-1)$ 의 요청 확률보다 적다. 이는 세그먼트  $k$ 는 세그먼트  $(k-1)$ 이 요청된 이후에 요청되기 때문이다. 따라서 식 (2)는 세그먼트  $k$ 의 대기시간이 세그먼트  $(k-1)$ 의 대기시간 보다 적거나 같은 것을 의미한다. 이는 요청 확률이 큰 세그먼트가 요청 확률이 작은 세그먼트에 비해 더 오래 동안 가상 메모리에 유지됨으로 더 자주 요청될 가능성이 큰 세그먼트가 중앙 비디오 서버로부터 전송되어지는 가능성을 줄일 수 있을 것이다.

위 식 (1)에서 요청된 임의의 세그먼트의 대기 시간을 구하기 위해서는 해당 동영상의 첫 번째 세그먼트의 대기 시간이 필요하다. 이를 위하여 임의의 동영상의 첫 번째 세그먼트의 초 당 평균 요청 횟수(*NOR*: Number Of Request)를 구한다. *NOR*은 식 (3)에 의하여 주어지며 첫 번째 세그먼트의 대기 시간은 *NOR*의 역수가 된다.

$$NOR_i(1) = \left[ M \times P_i(1) \times \frac{1}{N} \right] \quad (3)$$

여기서,  $M$ 은 사용자로부터 모든 비디오에 대한 초 당 세그먼트의 요청 횟수이며 이는 모든 동영상에 대하여 같다고 가정한다. 또한  $P_i(1)$ 은 첫 번째 세그먼트의 요청 확률이며  $N$ 은 동영상의 총 개수이다.

위와 같이 임의의 세그먼트의 대기 시간이 주어지면 제안하는 알고리즘에서 대기 시간을 활용한 가상 메모리에서 세그먼트의 삭제와 비디오 프록시 서버로의 전송은 다음과 같은 방법으로 수행된다.

- SET the first request time to T1
- SET the second request time to T2
- SET 대기시간 to  $\Delta$
- Request from user
- Request to central video server
- Store the delivered video segment in virtual memory

```

IF Not_Exist(second request for  $\Delta$ ) THEN
    Delete the stored video segment
ELSE
    IF Exist(another request for  $\Delta$ ) THEN
        Transfer the stored video segment
    ELSE
        Reset the  $\Delta$  at T2
        T1  $\leftarrow$  T2
    
```

A

Repeat the procedure "A" until the stored segment to be deleted or transferred to the video proxy server

END IF

END IF

위에서 세 가지 요청은 각각 요청된 세그먼트를 가상 메모리에 저장할 조건이며 세그먼트를 가상 메모리에 계속 저장하며 마지막으로 비디오 프록시 서버로 전송하기 위한 조건에 해당한다.

### III. 단편화를 고려한 가상 메모리 구조

세그먼트 단위로 동영상 데이터를 전송하는 방법 중 가장 효율적인 방법은 피라미드식 미디어 분할 방법이다. 피라미드 세그먼트 방법에서는 세그먼트  $k$ 에 포함되는 동영상 블록의 수는  $2^{(k-1)}$ 이다. 위 2장에서 살펴본 바와 같이 가상 메모리에는 계속하여 새로운 동영상 세그먼트들이 저장되고 삭제되는 것이 되풀이 된다. 이 경우 피라미드 세그먼트 방법은 제한된 저장 공간을 가지는 가상 메모리에 대하여 두 가지 문제점을 가진다. 첫째는 세그먼트 번호가 큰 동영상 데이터를 저장하기 위한 연속적 공간을 찾는 것이 어려울 수 있다. 다음으로 외부 단편화 문제가 일으킬 수 있으며 이는 가상 메모리를 이용한 비디오 프록시 서버의 관리의 성능에 큰 영향을 미칠 수 있다. 이러한 문제점을 해결하기 위하여 본 논문에서는 제한된 저장 공간을 가지는 가상 메모리에서 동영상 세그먼트들의 지속적인 삭제와 저장으로 발생할 수 있는 과도한 탐색 오버헤드와 단편화를 해결하기 위하여 세그먼트별로 저장 공간을 할당하도록 한다. 이를 위하여 가상 메모리는 그림 1과 같은 파일 구조를 가진다.

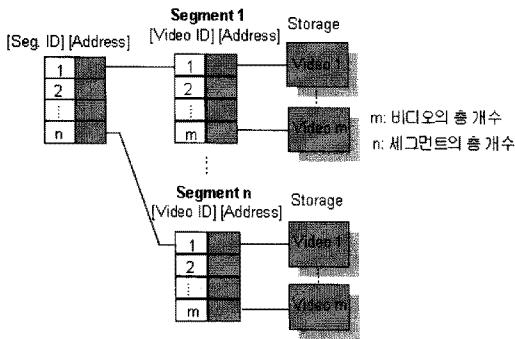


그림 1. 동영상 저장을 위한 세그먼트 기반 파일 구조  
Fig. 1. Segment-based file structure for video storing.

가상 메모리에서 세그먼트 별로 할당된 저장 공간을 최적화하기 위하여 다음과 같은 방법으로 공간 할당을 수행한다. 사용자들은 식 (2)에서 보는 바와 같이 동영상의 앞부분을 주로 요청하는 접근 패턴을 가진다. 따라서 동영상의 앞부분 세그먼트가 뒷부분의 세그먼트 보다 더 많이 저장 될 수 있도록 가상 메모리의 저장 공간을 할당하면 초기 지연 시간을 줄일 수 있을 것이다.

이를 위한 가장 간단한 방법은 가상 메모리에서의 세그먼트 별 저장 공간이 모두 동일한 크기를 가지도록 설정하는 것이다. 지금 동영상이 가지는 최대 세그먼트의 크기를  $K$ 라고 하는 경우 모든 세그먼트에는  $2^{(K-1)}$ 개의 블록의 정수배( $L$ )가 할당된다. 여기서 최대 세그먼트가 동시에 두 개 이상이 저장될 수 있도록 하기 위하여 정수 값은 최소 2 이상이 되도록 한다. 이러한 구조에서 가상 메모리로부터 비디오 프록시 서버로의 세그먼트의 전송은  $\Pi$ 장에서 주어진 방법에 의하여 수행된다. 반면 삭제는 각 세그먼트 별로 이루어진다. 즉 각 세그먼트 영역의 오버플로(overflow)를 방지하기 위하여 각 세그먼트의 저장 공간이 일정한 여유 공간을 확보하지 못하는 경우 가장 오래된 동영상부터 삭제한다. 여기서 일정한 여유 공간의 확보의 기준은 향후 대기 시간 동안 각 세그먼트의 저장 공간에 추가될 것으로 예측되는 세그먼트의 수가 현재 각 저장 공간에 저장 가능한 세그먼트의 수 보다 크게 되는 경우를 기준으로 한다. 그 조건은 다음 식 (4)와 같이 주어진다.

$$NRS_i(K) \leq L \times 2^{(K-1)} - SAR_i(K) \times T_i(K) \quad (4)$$

여기서,  $NRS_i(K)$ 는 각 세그먼트 저장 공간의 여유

세그먼트 값이며,  $SAR_i(K)$ 는 임의 세그먼트의 저장 공간에 축적되는 세그먼트의 평균 증가율이다.

#### IV. 실험

제안하는 동영상 데이터 삭제 알고리즘의 성능 평가를 위해서 기존의 알고리즘인 PLFU(Partial Least Frequently Used)<sup>[12]</sup>와 Distance-based 방법<sup>[13]</sup> 그리고 Reallocation Affinity<sup>[14]</sup> 방법을 대상으로 비디오 프록시 서버의 저장 공간 크기 변화에 따른 블록 적중률(block hit rate)과 블록 삭제 횟수(number of block deletion)를 비교한다. 이를 위하여 본 논문에서는 그림 2와 같은 동영상의 피라미드 분할 방법을 사용한다. 또한 실험을 위해 분 당 1,200회의 요청 횟수로 사용자가 원하는 동영상 데이터가 요청된다. 표 1은 실험을 수행하기 위한 조건이다.

그림 3은 제안하는 방법을 이용하여 가상 메모리의 크기 변화에 따른 블록 적중률을 비교하는 것을 보인다. 본 논문에서 가상 메모리의 크기는 비디오 프록시 서버 저장 공간의 10%, 15%, 20% 그리고 25%로 각각



그림 2. 미디어 분할: 피라미드식 분할

Fig. 2. Media segmentation approach: pyramid segmentation.

표 1. 실험 조건

Table 1. Simulation Parameters.

Parameter	Value
Simulation time	72 hours
Number of videos	1,200
Video size	655MB
Bit rate	1,024Kbps
Request rate per minute	1,200
Block play time	5 sec.
Storage size of video proxy server (GB)	50, 100, 150, 200, 250, 300, 350
Virtual memory size of video proxy server storage size	10%, 15%, 20%, 25%
Segmentation method	Pyramid segmentation
Number of segment per video	10 segments

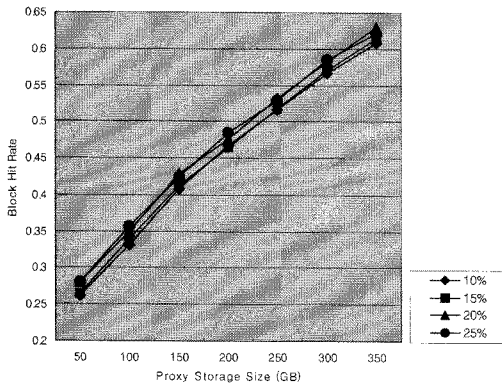


그림 3. 가상 메모리 크기 변화에 따른 적중률 비교  
Fig. 3. Comparison of block hit rate under various virtual memory size.

설정하여 실험한다. 수행된 실험 결과에서 가상 메모리의 크기를 15%, 20% 그리고 25%로 설정할 경우에는 비교적 유사한 블록 적중률을 보임을 알 수 있으나 10%로 설정할 경우에는 상대적으로 낮은 블록 적중률을 보이고 있음을 알 수 있다. 이것은 가상 메모리의 크기가 작을 때 해당 공간의 부족 현상으로 인한 삭제가 자주 일어나게 되고 이에 따라 사용자의 요청이 계속되는지 판단하기 위해 가상 메모리에 적재할 수 있는 동영상 데이터의 수가 적기 때문이다.

그림 4는 본 논문에서 제안하는 가상 캐싱 기법을 활용하여 기존의 방법인 PLFU와 Distance-based 방법 그리고 Reallocation Affinity 방법과 블록 적중률의 비교를 보인다. 실험 결과를 통해 기존의 방법들에 비해 비교적 높은 블록 적중률을 보임으로서 제안하는 비디오 프록시 서버에서의 가상 캐싱 기법이 사용자에 의해 향후 계속 요청될 동영상 데이터들을 선별하여 저장하고 있음을 알 수 있다. 실험을 위한 가상 메모리의 크기는 비디오 프록시 서버 공간의 20%로 설정한다.

제안하는 방법의 효율성을 보다 정확하게 확인하기 위해 그림 5와 같이 비디오 프록시 서버 저장 공간의 크기 변화에 따른 블록 삭제 횟수를 실험한다. 실험 결과를 통해 제안하는 방법이 기존 방법들에 비해 비교적 적은 블록 삭제 횟수를 보이는 것을 확인할 수 있다. 이를 통해 제안하는 방법이 사용자에 의해 요청될 가능성이 적은 동영상 데이터들만을 대상으로 삭제 알고리즘을 수행하여 자주 요청하는 동영상의 세그먼트들만을 비디오 프록시 서버에 유지하도록 함으로서 비디오 프

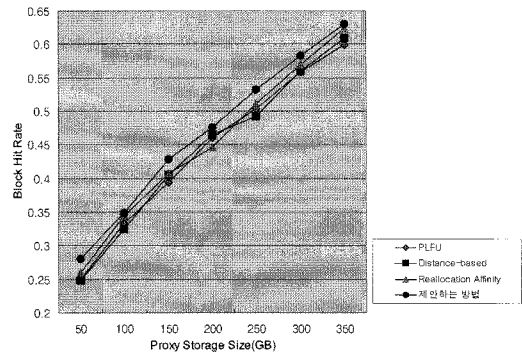


그림 4. 비디오 프록시 서버 저장 공간의 크기 변화에 따른 기존 알고리즘과의 블록 적중률 비교  
Fig. 4. Comparison of block hit rate under various video proxy server storage size.

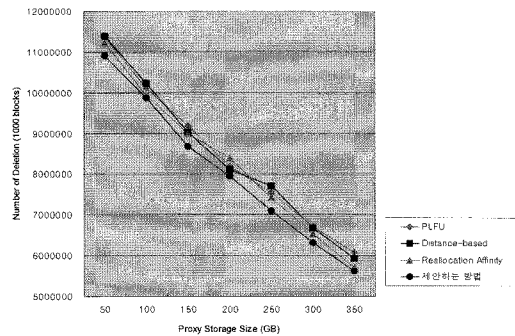


그림 5. 비디오 프록시 서버 저장 공간의 크기 변화에 따른 기존 알고리즘과의 블록 삭제 횟수 비교  
Fig. 5. Comparison of number of block deletion under various video proxy server storage size.

록시 서버의 저장 공간에서 수행되는 블록 삭제 횟수를 효과적으로 감소시키고 있음을 알 수 있다.

본 논문에서 수행된 실험 결과를 통해 기존 방법들에 비해 제안하는 방법이 높은 블록 적중률과 적은 블록 삭제 횟수를 나타낸다는 것을 확인한다. 이는 사용자가 계속해서 요청하는 동영상 데이터들이 비디오 프록시 서버에 미리 저장되며 사용자의 요구에 신속하고 안정적인 서비스를 제공할 수 있다는 것을 의미한다.

### V. 결 론

통신 기술의 급속한 발전에 힘입어 네트워크를 통한 실시간 동영상 서비스가 보편화 되고 있다. 현재 인터넷 상에서 이루어지는 동영상 데이터의 활용에 있어 가장 중요한 점은 그 품질을 보장 할 수 있어야 한다는

것이다. 현재의 동영상 전송은 손실과 지연에 상당히 민감하게 반응하여 이를 기반으로 한 여러 분야의 발전에 커다란 문제점으로 지적되고 있다. 이와 같은 문제를 해결하기 위해 사용자들과 근거리에서 위치한 비디오 프록시 서버를 활용하여 인터넷 VOD(Video-On-Demand) 서비스의 품질을 크게 개선시킬 수 있다. 그러나 비디오 프록시 서버는 비교적 제한된 용량을 가진다는 단점이 있다. 따라서 본 논문에서는 원격지에 위치하는 중앙 비디오 서버에 비해 상대적으로 제한된 저장 공간을 가지는 비디오 프록시 서버에 사용자들이 계속적으로 요청하게 되는 동영상 데이터만을 저장하는 가상 캐싱 기법을 제안한다.

제안하는 기법은 사용자가 요청한 동영상 데이터를 중앙 비디오 서버로부터 전송받아 가상 메모리에 일단 적재한 후 이후 사용자들이 요청한 시각의 최근성에 기초하여 해당 동영상 데이터를 비디오 프록시 서버에 이동하거나 삭제한다. 비디오 프록시 서버의 저장 공간과 가상 메모리의 크기를 변화시켜가며 수행한 실험을 통해 제안하는 방법은 기존의 방법들에 비해서 높은 블록 적응률과 적은 블록 삭제 횟수를 나타냄으로써 보다 더 효율적임을 확인한다.

### 참 고 문 헌

[1] Webin Jiang, Chong Huang, Hai Jin and Xiofei, "A New Proxy Scheme for Large-Scale P2P VoD System," *Proc. of IEEE/IFIP Int. Conf. EUC*, Vol. 1, pp. 512-518, Dec. 2008.

[2] Yun-Jung Cheng, Kuochen Wang, Rong-Hong Jan, Chien Chen, and Chia-Yuan Huang, "Efficient failover and load balancing for dependable SIP proxy servers," *Proc. of IEEE Symp. ISCC*, pp. 1153-1158, July 2008.

[3] Wathsala W.V., Siddhisena B., and Athukorale A.S., "Next Generation Proxy Servers," *Proc. of Int. Conf. ICACT*, Vol. 3, pp. 2183-2187, Feb. 2008.

[4] Wang. J. Z and Yu. P. S, "Fragmental Proxy Caching for Streaming Multimedia Objects," *IEEE Trans. on Multimedia*, Vol. 9, pp.147-156, Jan. 2007.

[5] Jun Pyo Lee and Sung Han Park, "A Cache Management Policy in Proxy Server for an Efficient Multimedia Streaming Service," *Proc. of IEEE Int. Conf. ISCE*, pp. 64-68, June 2005.

[6] Kaihui Li, Changqiao Xu, Yuanhai Zhang, and

Zhimei Wu, "Optimal Prefix Caching and Data Sharing Strategy," *Proc. of IEEE Int. Conf. Multimedia and Expo*, pp. 465-468, June 2008.

[7] Seong Ho Park, Eun Ji Lim, Ki Dong Chung, "Popularity-based partial caching for VOD systems using a proxy server," *Proc. of Parallel and Distributed Processing Symposium*, pp. 1164-1168, April 2001.

[8] E. J. O'Neil, P.E. O'Neil, G. Weikum. "The LRU-K Page Replacement Algorithm For Database Disk Buffering," *Proc. of the 1993 ACM SIGMOD Conference*, pp. 297-306, 1993.

[9] S. Acharya and B. Smith, "MiddleMan: A Video Caching Proxy Server", *Proc. of 10th International Workshop on Network and Operating Systems Support for Digital Audio and Video*, Chapel Hill, NC, June 2000.

[10] S. Jin and A. Bestavros, "Popularity-aware greedy dual-size web proxy caching algorithms," *Proc. of 20th IEEE Intl. Conf. Distributed Computing Systems (ICDCS)*, Taipei, Taiwan, pp. 254-261, April 2000.

[11] Songqing Chen, Haining Wang, Xiaodong Zhang Shen B, and Wee S, "Segment-based proxy caching for Internet streaming media delivery," *IEEE Trans. on Multimedia*, Vol. 12, pp. 59-67, Sept. 2005.

[12] Kuan-Sheng Hsueh and Sheng-De Wang, "A Packet-Based Caching Proxy with Loss Recovery for Video Streaming," *Proc. of 2002 Pacific Rim Int. Symposium on Dependable Computing*, pp. 185-190, Dec. 2002.

[13] Songqing Chen, Bo Shen, Wee. S, and Xiaodong Zhang, "Segment-based streaming media proxy: modeling and optimization," *IEEE Trans. on Multimedia*, Vol. 8, pp. 243-256, April 2006.

[14] Christian Spielvogel and Laszlo Boszormenyi, "Quality-of-Service based Video Replication," *Proc. of Int. Workshop on Semantic Media Adaptation and Personalization*, pp. 21-26, Dec. 2007.

저 자 소 개



이 준 표(정회원)  
 2001년 한성대학교 컴퓨터공학과  
 학사  
 2003년 한양대학교 컴퓨터공학과  
 석사  
 2003년~현재 한양대학교 컴퓨터  
 공학과 박사 과정

<주관심분야 : 영상처리, 멀티미디어 시스템>



박 성 한(정회원)  
 1970년 한양대학교 전자공학과  
 학사  
 1973년 서울대학교 전자공학과  
 석사  
 1984년 미국 텍사스 주립대 전기  
 및 컴퓨터공학과 박사

2003년 대한전자공학회 회장  
 2005년~2007년 WFEO 정보통신의장  
 1986년~현재 한양대학교 컴퓨터공학과 교수  
 <주관심분야: 영상처리, 컴퓨터 네트워크>