

# VOD 서버에서 효율적인 연속미디어 서비스를 위한 동적 버퍼 분할 기법

권 춘 자<sup>†</sup> · 최 창 열<sup>††</sup> · 최 황 규<sup>††</sup>

## 요 약

VOD 서버는 비디오와 같은 연속미디어를 여러 사용자에게 매끄럽게 서비스하기 위해 데이터의 일부를 미리 서버 버퍼로 읽어 들여 정교하게 관리한다. 이때 서버 버퍼로 읽어 들인 연속미디어 데이터를 다른 사용자가 재사용하면 저장장치로의 접근 횟수가 줄어들고 사용자의 대기 시간이 감소된다. 본 논문에서는 서버의 버퍼를 그룹으로 분할하고 각 그룹에서 활용도가 가장 낮은 버퍼 공간을 동적으로 재분할하여 새로운 사용자에게 할당하는 동적 버퍼 분할 기법을 제안하고, 이를 다중 스트림과 클립 데이터, 그리고 VCR 기능을 처리하도록 확장한다. 제안된 기법에서는 버퍼 활용도가 높아져 병행 사용자의 수가 증가되고, 저장장치의 대역폭 한계에 서서히 도달하여 사용자의 평균 대기시간이 감소된다. 시뮬레이션을 통해 동적 버퍼 분할 기법과 기존의 버퍼 분할 기법에 대한 사용자 평균 대기시간과 병행 사용자의 수를 관찰한 결과, 평균 대기 시간은 최소 50% 이상 감소되고 병행 사용자의 수는 1~5% 증가함을 확인하였다.

## Dynamic Buffer Partitioning Technique for Efficient Continuous Media Service in VOD Servers

Chun Ja Kwon<sup>†</sup> · Chang Yeol Choi<sup>††</sup> · Hwang Kyu Choi<sup>††</sup>

## ABSTRACT

In VOD server, in order to guarantee playback of continuous media, such as video, without hiccups for multiusers, the server has to manage its buffer sophisticatedly by prefetching a part of the data into the buffer. As the continuous media data buffered by one user can be used again by the others, the number of disk accesses is reduced and then the latency time for the users is also reduced. In this paper, we propose a new buffer management technique for continuous media in VOD server. Our basic algorithm partitions the buffer into groups and then a group of buffer which has the lowest utilization is chosen and partitioned again for a new user. The basic algorithm is extended for supporting multiple streams and clip data and for providing VCR functions. Our proposed technique is able to increase in the number of concurrent users as increasing the utilization of the buffer and to minimize the average waiting time for multiuser accesses as the bandwidth of storage is slowly reached to the limit. In the simulation study for comparing the performance of our technique with that of the existing techniques, we show that the average waiting time is reduced more than 50% and the number of concurrent users increases by 1~5% as compared with those of the exiting techniques.

키워드 : 멀티미디어(multimedia), VOD 서버(VOD server), 버퍼관리(buffer management), 동적 버퍼분할(dynamic buffer partitioning), 연속미디어(continuous media)

## 1. 서 론

최근 들어 통신 기술과 비디오 압축 기술의 급속한 발전에 힘입어 네트워크를 통한 실시간 비디오 서비스가 보편화되고 있다. 사용자들은 비디오 데이터를 저장하고 있는 시스템에 접근하여 원하는 비디오를 끊임 없이 연속적으로 서비스 받고자 한다. 사용자가 요청한 비디오 프로그램을 네트워크를 통해 제공하는 시스템을 보통 VOD(Video-On-Demand) 서버라

부른다. VOD 서버에 대한 연구는 대용량성과 실시간 처리 같은 멀티미디어 데이터의 특성에 초점을 맞춰 저장 서버, 버퍼 관리, 실시간 전송 및 스케줄링 기법 등 다양한 관점에서 다양하게 이루어져 왔다[1-4, 11-13].

VOD 서버에서 비디오처럼 연속 재생을 필요로 하는 대량의 연속미디어(continuous media) 데이터는 일반적으로 대용량의 디스크에 저장되고, 사용자가 요청한 비디오 데이터의 일부를 서버 버퍼로 미리 읽어 들여 연속 서비스가 이루어지도록 관리된다. 이때 저장장치로부터 서버 버퍼로 읽어 들인 연속미디어 데이터를 다른 사용자가 재사용하면 저장장치에 새롭게 접근하지 않고도 사용자의 대기시간을

※ 본 논문은 강원대학교 BK21 사업단 지원에 의한 연구 결과의 일부임.

† 준 회 원 : 강원대학교 대학원 컴퓨터정보통신공학부 박사과정

†† 정 회 원 : 강원대학교 전기전자정보통신공학부 교수

논문접수 : 2002년 2월 8일, 심사완료 : 2002년 6월 3일

줄일 수 있다. 즉, 크기가 매우 큰 비디오 데이터를 메모리에 유지하기 어려우므로 비디오 데이터의 일부를 블록 단위로 버퍼로 읽어들이 동일한 비디오를 요청한 사용자에게 서비스하고, 버퍼에 있는 데이터 블록이 모두 서비스되기 전에 다음 블록을 버퍼로 읽어들이 연속적인 서비스를 보장한다. 궁극적으로는 서버 버퍼의 정교한 관리를 통해 사용자의 평균 대기시간, 버퍼 크기, 디스크 대역폭 요구를 최소화하거나 병행 사용자의 수를 증가시키고자 한다[2, 3, 5, 14-16].

최근의 VOD 서버의 버퍼 관리는 정적 기법의 문제점을 해결하기 위해 동적인 방법으로 연구가 이루어지고 있다 [6-8]. [6]에서는 시스템의 완전 부하와 불완전 부하 상태에서 최소 크기 버퍼를 동적으로 할당하는 기법을 제시하였다. Rotem과 Zhao[5]는 사용자 그룹에 대응되는 수 개의 섹션으로 버퍼를 분할(buffer partitioning)하여 저장장치로부터 검색한 데이터 블록을 그룹에 속한 모든 사용자들이 재사용함으로써 끊임 없이 비디오를 재생하면서 평균 대기시간을 최소화하고자 하였다. 그러나 이 기법은 단일 비디오만을 대상으로 하고 있으며 한번 형성된 그룹은 상황에 따라 변경되지 않는다. 버퍼 분할 기법은 동적 버퍼 분할[7]과 버퍼의 동적 그룹화(dynamic grouping)[8]로 확장되었다. 동적 그룹화에서는 LRU와 유사한 버퍼 관리 기법은 VOD 응용에서는 버퍼 공간을 효율적으로 사용하지 못함을 밝히고, 그룹들을 동적으로 결합하거나 분해하여 메모리 공간과 입출력 대역폭을 최적화 하고자 하였다.

본 논문에서는 연속미디어에 대한 접근이 순차적으로 이루어지는 특성에 기반하여 서버의 버퍼를 그룹으로 분할하고 각 그룹에서 활용도가 가장 낮은 버퍼 공간을 동적으로 재분할하여 새로운 사용자가 사용하는 동적 버퍼 분할 기법을 제안한다. 하나의 비디오 스트림을 대상으로 제안된 기본 알고리즘은 전체 버퍼를 하나의 그룹으로 관리하다가 각 그룹에서 활용도가 가장 낮은 버퍼 공간을 재분할하여 사용한다. 따라서 버퍼의 활용도가 높아져 병행 사용자의 수가 증가되고 저장장치의 대역폭 한계에 서서히 도달하여 사용자의 평균 대기시간이 감소된다. 또한 기본 알고리즘을 인터벌 캐싱[9] 등과 결합시켜 다중 스트림을 지원하고 클립(clip) 데이터와 VCR 기능을 처리하도록 확장한다.

본 논문은 다음과 같이 구성된다. 서론에 이어 2장에서 기존의 버퍼 분할 기법들의 동작과 장단점, 인터벌 캐싱(Interval Caching)[3, 4] 등의 관련 연구에 대해 알아본다. 3장에서는 제안된 동적 버퍼 분할 기법의 기본 알고리즘과 확장 기능의 지원에 대해 기술하고, 4장에서 제안된 기법들의 성능 분석을 다룬다. 마지막으로 5장에서는 결론 및 향후 연구과제에 대해 기술한다.

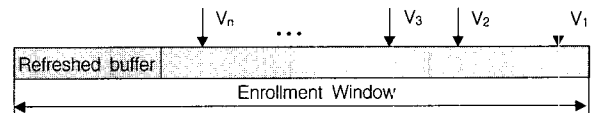
2. 관련 연구

2장에서는 동적 버퍼 분할 기법의 제안 배경이 되는 기

존의 버퍼 분할 기법들에 대한 동작과 장단점을 살펴보고, 제안된 동적 분할 기법의 기능을 확장하는데 응용되는 인터벌 캐싱 기법에 대해 기술한다.

2.1 버퍼 분할 기법

연속미디어를 위한 버퍼 관리 방법 중의 하나인 버퍼 선인출(prefetch)은 연속미디어의 접근이 순차적으로 이루어지는 특성에 근거하여 앞으로 접근될 데이터를 미리 버퍼로 읽어온 후에 사용자 요구를 버퍼에서 서비스함으로써 재생 지연시간을 최소로 한다. 즉, (그림 1)과 같이 사용자  $V_1$ 에서  $V_n$ 까지가 동일한 비디오에 접근하는 경우에 대체로 크기가 큰 하나의 버퍼를 첫 번째 사용자에게 할당하면 후속 사용자들인  $V_2$ 에서  $V_n$ 까지는 더 이상 저장장치에 접근하지 않고 버퍼로부터 서비스를 받을 수 있다[4, 5].



(그림 1) 사용자 등록 창(Viewer Enrollment Windows) 정책

Rotem과 Zhao[5]는 단일 비디오를 연속적으로 서비스하면서 평균 대기시간을 최소화하기 위한 버퍼관리 기법으로 버퍼 비분할(No Partitioning), 정적 분할(Static Partitioning), 적응 분할(Adaptive Partitioning)을 제안하였다.

버퍼 비분할은 (그림 1)과 같이 버퍼 전체를 하나로 관리한다. 그림에서 갱신된 블록(Refreshed Block : RB)은 연속 재생을 보장하기 위해  $V_1$ 의 입장에서 한 주기 동안에 디스크로부터 읽어와야 하는 블록의 개수를 말한다. 비분할에서는 등록 창(Enrollment Window)이 닫히기 전에 도착한 사용자들에게만 승인이 허락되므로 동일 비디오에 대한 병행 사용자 수가 제한되고 버퍼 크기만큼의 사용자들은 원하는 데이터가 이미 버퍼에 들어 있으므로 평균 대기시간 없이 비디오를 서비스를 받는다. 그러나 등록 창이 닫힌 순간에 서비스를 요청한 사용자는 승인이 허락된 사용자들에 대한 서비스가 끝날 때까지 기다려야 하므로 대기시간이 증가한다. 극단적인 경우로, 등록 창이 닫히기 전에 도착한 사용자가 한 명이라면 두 번째 사용자부터 나머지 사용자들은 첫 번째 사용자에게 대한 서비스가 끝날 때까지 모두 기다려야 한다. 따라서 비분할은 사용자의 서비스 요청 간격을 짧을 때는 효과적이지만, 일반적으로 사용자들의 서비스요청 간격이 매우 다양하므로 비분할에서의 버퍼 활용도는 낮아지게 된다. 갱신된 블록의 크기 ( $b_c$ )와 등록 창의 크기 ( $w$ ), 그리고 버퍼 갱신이 일어나기 전에 버퍼에 있어야 하는 블록의 최소 개수(Refresh Slack :  $y$ )에 대해서는 4장에서 다룬다.

정적 분할에서는 버퍼를 일정한 크기의 부버퍼로 분할하

고, 비디오의 시작부분에 가까운 선두 부버퍼에서 끝 부분에 가까운 후미 부버퍼까지 상대적인 위치에 따라 부버퍼를 정렬시킨 후 각각의 부버퍼는 비분할 방법으로 관리한다. 이때 비분할 방법으로 관리되는 부버퍼를 그룹이라 하며, 모든 그룹에 대한  $w, b_c, y$  값을 미리 결정해야 한다. 그리고 각 그룹에서 필요로 하는 블록들을 저장장치로부터 갱신해야 하므로 저장장치의 대역폭을 충족시키는 범위 내에서 그룹의 개수를 결정해야 한다. 그룹의 개수는 우선 각 그룹에서 필요로 하는 최소한의 버퍼 크기 ( $B_{min}$ )를 계산한 후 버퍼의 전체 크기에 대한 비율로 결정한다. 정적 분할에서는 동시에 서비스를 받는 병행 사용자 수가 증가되고 평균 대기 시간은 감소된다. 그러나 각 부버퍼에서 필요로 하는 블록들이 연속적이지 않기 때문에 디스크탐색시간이 증가하고 비분할로 관리되는 각 그룹의 버퍼 활용도가 낮아진다.

적용 분할은 정적 분할의 각 그룹에서 활용도가 낮은 버퍼 공간을 재분할하여 새로운 사용자가 다시 사용하게 함으로써 각 그룹의 버퍼 활용도를 높인다. 즉 각 사용자가 필요로 하는 최소 버퍼 크기  $B_{min}$ 이 서로 연속적으로 연결되면 이 블록들을 하나의 그룹으로 서비스하고 연속적이지 않으면 다른 그룹으로 분리해서 서비스한다. 적용 분할은 사용자의 서비스 요청 간격이 짧은 경우에는 버퍼 재분할이 드물게 일어나기 때문에 비분할과 정적 분할에 비해 효과가 크지 않다. 그러나 사용자의 서비스 요청 간격이 각 사용자에게 필요한 최소 버퍼 크기  $B_{min}$ 에 비해 상대적으로 큰 경우에는 버퍼의 재분할이 자주 일어나 효과적으로 된다. 그러나 버퍼 재분할이 많이 일어나면 그룹의 개수가 늘어나 어느 순간부터는 저장장치의 대역폭 한계에 도달하여 더 이상 그룹을 만들 수 없게 된다. 따라서 버퍼의 크기가 커진다 하더라도 사용자들의 평균 대기시간은 더 이상 줄어들지 못한다.

## 2.2 인터벌 캐싱 기법

인터벌 캐싱은 연속되는 스트림들 사이의 작은 인터벌을 캐싱한다. 스트림  $S_i$ 가 읽은 블록을 스트림  $S_j$ 가 다음에 읽는다면  $S_i$ 와  $S_j$ 는 연속적이라 하고 연속적인 스트림 각각을 선행 스트림과 후행 스트림이라 할 때, 선행 스트림이 가져온 블록을 버퍼에 유지함으로써 후행 스트림은 필요한 블록을 디스크에 접근하지 않고 버퍼에서 읽게 된다.

GIC(Generalized Interval Caching)는 인터벌 캐싱 정책을 확장하여 비디오처럼 크기가 큰 연속미디어뿐 아니라 크기가 작고 산발적으로 요청되는 클립(clip) 데이터가 함께 있는 작업부하 환경을 다룬다[4, 9]. 클립 데이터는 길이가 짧으므로 클립 데이터의 재생이 끝나기 전에 같은 요구가 들어올 확률이 낮다. 따라서 GIC에서는 재생이 끝난 클립 데이터에 대한 정보를 유지한 후, 같은 클립으로의 서비스 요청이 들어오면 인터벌을 측정하여 캐싱한다.

## 3. 동적 버퍼 분할 기법

### 3.1 기본 알고리즘

동적 버퍼 분할 기법은 기본적으로 전체 버퍼를 하나로 관리하다가 사용자의 참조 간격이 일정한 크기 이상으로 되는 버퍼 공간을 회수하여 그룹으로 분할해서 관리한다. 이를 위한 기본 알고리즘에서는 다음과 같은 가정을 한다.

- 버퍼에 있는 데이터는 MPEG 프레임으로 압축되어 있다.
- 하나의 GOP를 한 블록으로 한다.
- 사용자 측에는 압축을 해제할 수 있는 장비가 있다.
- 버퍼를 갱신하는 시간은 디스크에서 버퍼로 데이터를 전송하는 시간과 같다.
- 모든 사용자는 동일한 재생률을 요구한다.
- 모든 사용자는 하나의 동일한 스트림을 서비스 받는다.

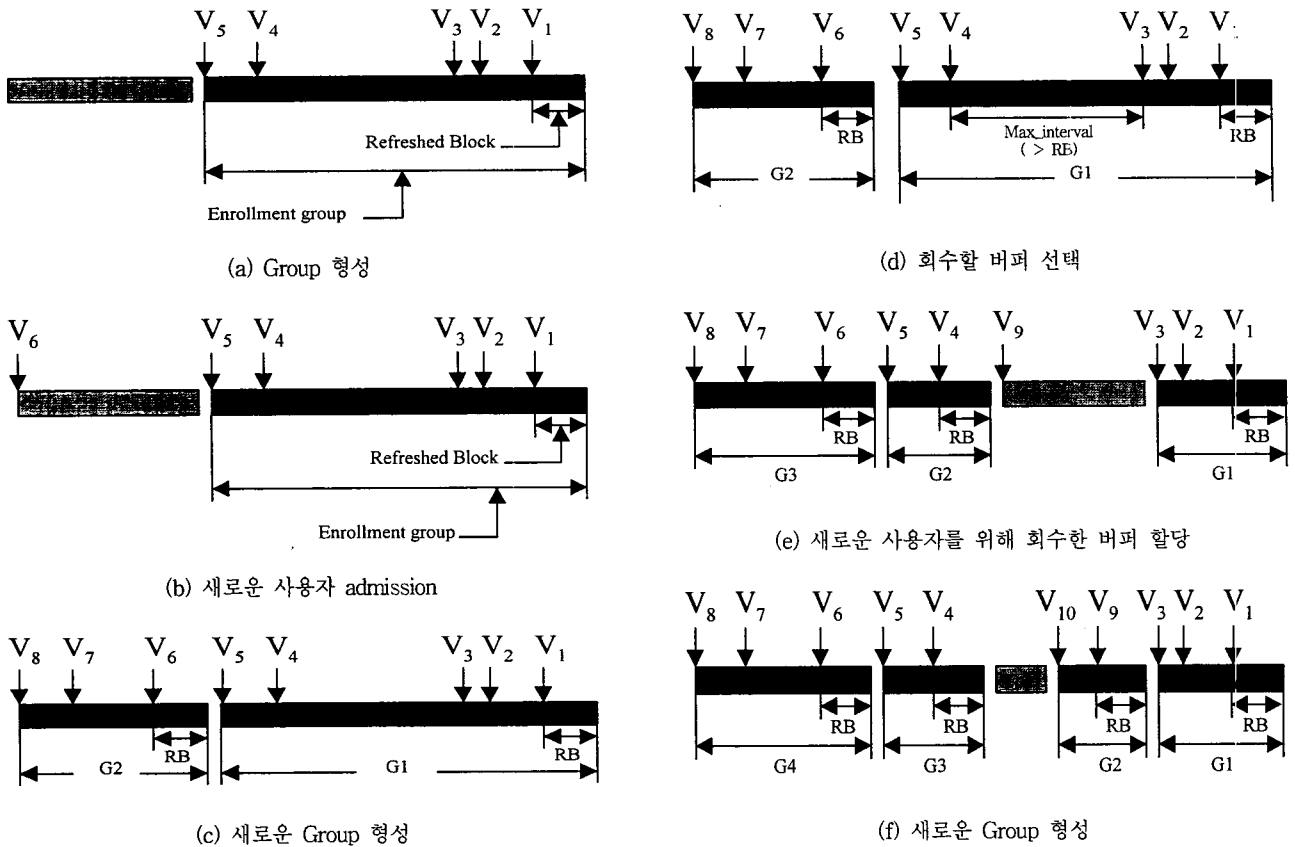
하나의 비디오를 위한 동적 버퍼 분할 기법에서의 버퍼 관리는 (그림 2)와 같으며, 각 단계에서 이루어지는 작업은 다음과 같다.

제 1단계(그룹의 형성): 첫 번째 사용자가 영화를 보기 위해 참여( $V_1$ )하여 영화를 보고 있는 동안 동일한 영화를 보고자하는 다른 사용자들이 임의로 도착해서 참여한다. 이때 승인이 허락된 사용자(enrolled user)들이 필요로 하는 블록들이 이미 모두 버퍼에 들어 있으므로 디스크에 접근하지 않고, 즉 재생 지연시간 없이 곧장 서비스를 받게 된다. 여기서 첫 번째 사용자가 버퍼에 있는 블록들을 모두 서비스 받은 후에 다음 주기를 재생하기 위해 버퍼에 없는 블록을 원하면, 처음 사용자부터 마지막 사용자까지를 하나의 그룹(Enrollment Group)으로 만든다.

제 2단계(새로운 사용자 참여): 첫 번째 등록 창이 닫혀 첫 번째 그룹이 형성된 바로 다음에 도착한 사용자( $V_0$ )는 앞으로 생성될 두 번째 그룹의 선두 사용자가 되며 첫 번째 그룹에서 사용하지 않는 버퍼를 이용하여 두 번째 그룹을 형성한다.

제 3단계(새로운 그룹의 형성): 제 2단계에서 도착한 사용자와 그 이후에 도착한 사용자는 제 1단계를 반복하여 두 번째 그룹  $G_2$ 를 형성한다. 같은 방법으로 제 2단계와 제 3단계를 반복하여 3, 4, 5, ..., n 번째 그룹을 형성한다. 이때 형성될 수 있는 그룹의 최대 개수 ( $n_{max}$ )는 무한히 큰 값이 아니고 저장장치의 대역폭 한계를 만족해야 하는 값이다. 그룹 내의 사용자들은 마지막 사용자부터 차례로 다음 블록을 버퍼에서 재참조하며, 선두 사용자는 마지막 사용자가 사용한 버퍼를 비우고 저장장치로부터 미리 읽어온 새로운 블록을 비운 버퍼에 위치시킨다.

제 4단계(회수할 버퍼 선택 및 할당): 이미 형성된 그룹들이 모든 버퍼를 사용하고 있으면 참조 시간 간격이 RB



(그림 2) 동적 버퍼 분할에서의 버퍼 관리

보다 큰 두 사용자 사이의 버퍼를 회수하여 새로운 그룹을 형성한다. (d)의  $V_3$ 와  $V_4$ 처럼 각 그룹에서 참조 시간 간격이 가장 큰 두 사용자를 선택하여 이들이 속한 그룹을, 그림 (e)의  $G_1$ 과  $G_2$ 와 같이 두 개의 서로 다른 그룹으로 분리한 후 두 사용자 사이에 존재하는 버퍼들을 회수한다. 회수한 버퍼는  $V_9$ 와 같은 새로운 사용자에게 할당하여 새로운 그룹을 형성한다.

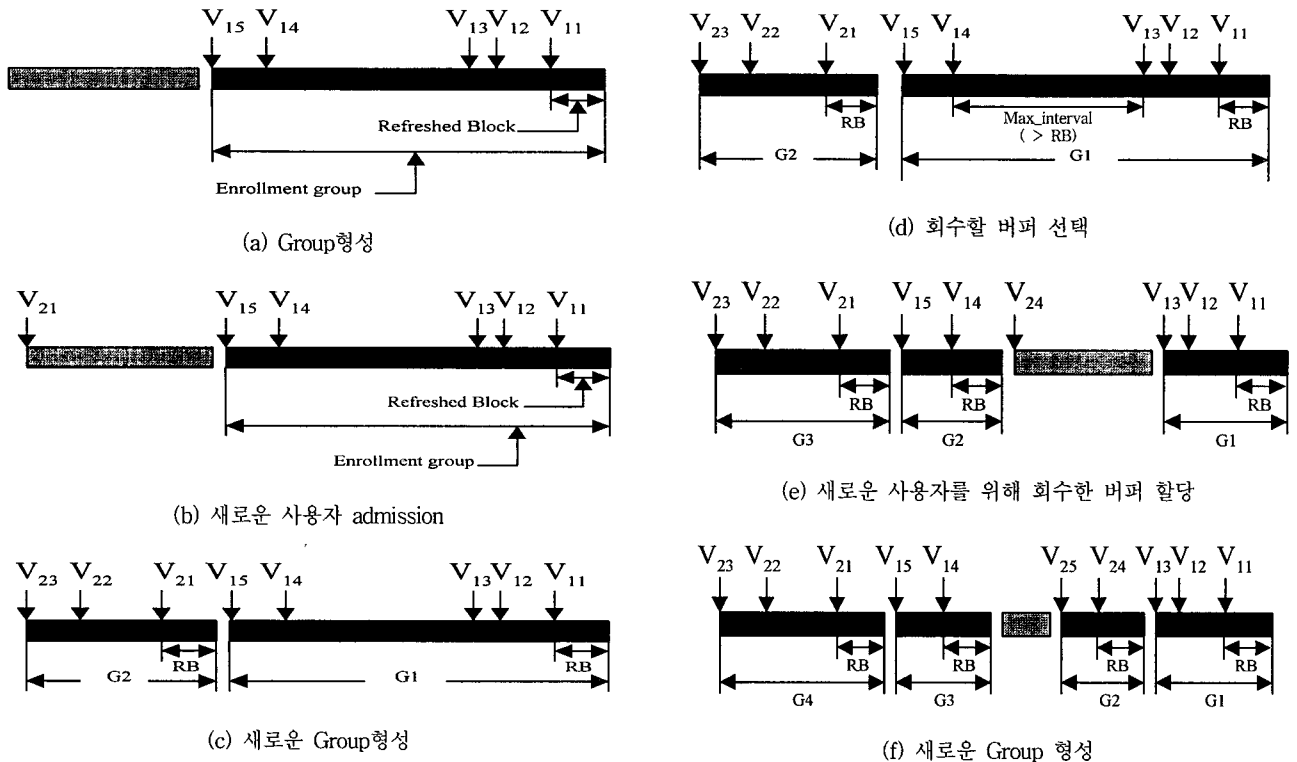
제 5단계(새로운 사용자 참여): 2, 3, 4 단계를 반복하여 회수한 버퍼를 이용하여 새로운 그룹을 형성한다.

제안된 동적 분할 기법은 처음에는 버퍼 전체를 하나의 그룹으로 관리하다가 가장 활용도가 낮은 버퍼 공간을 재분할하여 사용한다. 따라서 저장장치의 대역폭 한계에 서서히 도달하게 되어 버퍼 재분할이 많이 늘어나더라도 사용자의 평균 대기시간은 줄어들게 된다. 또한 동적 분할에서는 앞으로 참여할 사용자들의 재참조가 고려되므로 사용자의 서비스 요청 간격이 클 경우에 동적 분할을 통한 버퍼의 재사용이 늘어나 버퍼의 활용도가 높아진다. 이것은 각 사용자에게 필요한 최소 버퍼 크기  $B_{min}$ 을 기준으로 그룹을 생성하여 사용자의 서비스 요청 간격이  $B_{min}$ 에 비해 상대적으로 커 재분할이 많이 일어나면, 저장장치의 대역폭 한계에 빨리 도달되는 적응 분할과 비교된다.

### 3.2 다중 스트림 지원

단일 스트림을 위한 기본적인 동적 분할에 인터벌 캐싱을 응용하여 다중 스트림의 처리를 지원한다. 동적 분할과 유사하게 각 그룹을 관리하고 회수할 버퍼를 선정하지만, 참조 시간 간격이 가장 큰 두 사용자를 그룹에 상관없이 선택하여 이들이 속한 그룹을 두 개로 분리한 후 그 사이에 존재하는 버퍼 공간을 모두 회수해서 새로운 사용자에게 할당한다.

다중 스트림을 처리할 수 있도록 기본 알고리즘을 확장한 동적 분할은 (그림 3)과 같다. 비디오 번호를  $n$ 이라 하고 동일한 비디오를 서비스 받는 사용자들의 순차 번호를  $x$ 라 하면 각 사용자는  $V_{mx}$ 로 나타낼 수 있다. 서로 다른 비디오를 나타내는  $n$  값에 관계없이, (d)의  $V_{13}$ 과  $V_{14}$ 처럼 두 사용자 사이의 참조 시간 간격이  $RG$ 보다 큰 버퍼 공간을 회수하여 2번 비디오 사용자인  $V_{24}$ 에 할당하고 새로운 그룹을 형성한다. 동일한 비디오를 요청한  $n$  값이 같은 사용자들만 동일한 그룹으로 묶일 수 있으므로 그림에서는 1번 비디오와 2번 비디오를 위한 그룹이  $G_1$ ,  $G_3$ 와  $G_2$ ,  $G_4$ 로 각각 2개씩 생성되었음을 보여준다. 즉, 연속미디어에 대한 데이터 재참조를 예상해서 그룹을 형성함으로써 하나의 그룹은 동일한 비디오를 보는 사용자들로만 이루어지게 되어 서로 다른 스트림은 서로 다른 그룹을 형성하게 된다.



(그림 3) 동적 버퍼 분할을 통한 다중 스트림 지원

3.3 클립 데이터 처리

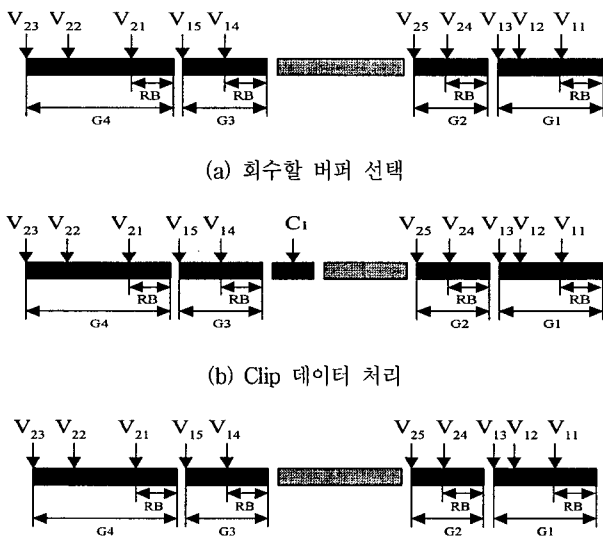
클립 데이터의 처리는 동적 분할과 GIC 정책을 결합하여 (그림 4)와 같이 이루어진다. 클립 데이터를 위한 버퍼 공간은 동적 분할과 동일하게 마련된다. 즉, 그룹에 상관없이 참조 시간 간격이 가장 큰 두 사용자를 선택하여 이들이 속한 그룹을 두 개로 분리한 후, 그 사이에 존재하는 버퍼를 모두 회수해서 새로운 사용자에게 할당한다. 할당된 버

퍼 공간 중에 (그림 4)(b)와 같이 클립 데이터의 길이 만큼만 사용하고 나머지는 다음 사용자를 위해 남겨둔다. 그리고 클립 데이터를 전송한 후에는 (그림 4)(c)처럼 클립 데이터 처리에 사용된 버퍼를 제일 먼저 반환한다.

3.4 VCR 기능 처리

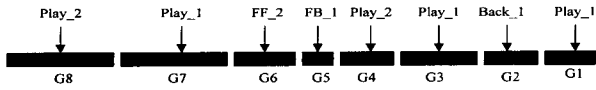
VCR 기능은 다중 스트림 처리와 비슷하게 이루어진다. 후진 기능은 또 다른 스트림으로 간주하여 동일한 비디오에 대해 후진 기능을 요청하는 사용자들은 다른 그룹으로 관리한다. 고속 전진과 고속 후진을 요청하는 사용자들은 요청한 배속에 따라 동일한 배속을 같은 스트림으로 간주하고 다른 배속은 또 다른 그룹으로 관리한다. 사용자가 일시정지를 요청했을 때, 만약 그 사용자가 그룹의 선두이거나 마지막 사용자이면 그가 소유한 버퍼를 반환하고 그를 그룹에서 제외시켜 버퍼의 활용도를 높인다. 그러나 그 사용자가 그룹의 선두이거나 마지막 사용자가 아니면 버퍼의 재참조를 기대하여 버퍼를 분할하지 않고 그대로 둔다. 나중에 재개(resume) 요청이 들어왔을 때, 원하는 데이터가 버퍼 공간에 있으면 그 버퍼 공간이 있는 그룹에 사용자가 들어가게 되고 그렇지 않으면 새로운 그룹을 만든다.

(그림 5)는 각각의 VCR 기능을 위해 분할되어있는 그룹의 상태를 보여준다. 1번 비디오의 재생 기능은 G1, G3, G7 그룹에서 요청되어있고, G4, G8 그룹은 2번 비디오의 재생 기능을 위해 사용되고 있다. 또 1번 비디오의 후진 기



(그림 4) 클립 데이터 처리 과정

능은 G2 그룹이, 고속 후진 기능은 G5 그룹이 요청하고 있음을 보여주고 있다. 그리고 G6는 2번 비디오에 대한 고속 전진을 보인다.



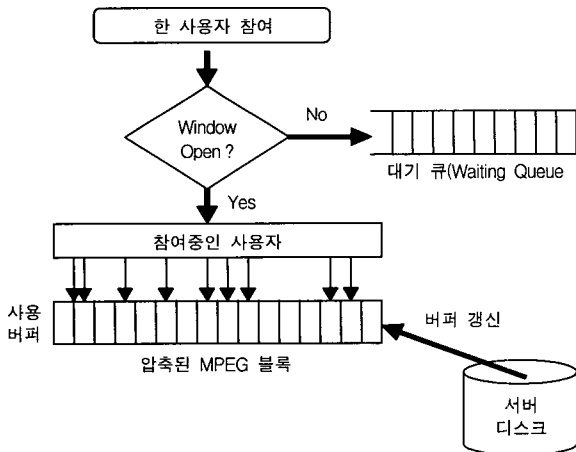
(그림 5) VCR 기능의 처리

4. 성능분석

4.1 시뮬레이션 환경

4.1.1 시뮬레이션 모델

제안된 동적 버퍼 분할 기법의 성능을 분석하기 위한 시뮬레이션 모델과 환경은 SMLP[10]로 구축되었으며, 시뮬레이션 프로그램은 Visual C++로 작성하였다. 시뮬레이션 모델은 (그림 6)과 같으며, 간단한 동작 흐름은 다음과 같다. 사용자의 서비스 요청은 임의의 시간 간격으로 도착하며 서로 독립적이다. 서버에 도착한 사용자 요청은 등록 창이 열려있으면 이미 서비스를 받고 있는 사용자들 사이에 참여하게 되고, 아니면 대기 큐에서 대기한다. 서버 디스크에는 MPEG으로 압축된 데이터가 저장되어 있으며, 서비스에 참여중인 사용자를 위한 MPEG 블록은 서버 버퍼로 선인출된다.



(그림 6) 성능분석을 위한 시뮬레이션 모델

시뮬레이션을 간단히 하기 위해 대기 큐에서 대기중인 사용자는 서비스를 중단하고 빠져나갈 수 없게 하였으며, 사용자 요청의 시간 간격은 지수 확률 분포를 따르고 전체 시뮬레이션 시간은 8시간으로 하였다.

4.1.2 시뮬레이션 파라미터

시뮬레이션에 사용되는 파라미터들과 초기 값은 <표 1>과 같다. <표 1>에서 보는 바와 같이 제안된 동적 버퍼 분

할 방법과 적응 버퍼 분할 방법 등과의 성능을 비교 평가하기 위해서는 등록 창의 크기와 그룹의 최대 개수 등이 결정되어야 하며, 다음과 같이 유도된다[5].

<표 1> 시뮬레이션 파라미터 및 초기값

기호	의 미	값
$B$	버퍼의 전체 크기	250 (MB)
$b$	압축된 블록의 크기	0.06 (MB)
$d$	재생률	4 (블록/초)
$r$	디스크 속도(디스크 지연시간 포함)	2 (MB/초)
$M$	영화 재생 시간	14400 (초)
$s$	디스크 지연시간 (디스크탐색시간과 회전지연시간)	0.015 (초)
$x$	사용자의 평균 서비스 요청 간격	300 (초)
$y$	Refresh Slack의 크기	(블록)
$b_c$	주기당 갱신되는 블록 수	(블록)
$w$	등록 창(Enrollment Window)의 크기	(블록)
$n_{max}$	그룹의 최대 개수	(개)

디스크 대역폭 한계 내에서 연속 재생을 보장하면서 병행 사용자들의 평균 대기시간을 최소화하는 그룹  $n$ 의 최대 값을 구하기 위해서는, 버퍼 선인출에서 최대 사용자 수를 위한 등록 창의 크기  $w$ 와 병행 사용자 수를 최대화하면서 한 주기당 갱신되는 MPEG 블록의 최소 개수인  $b_c$ 를 알아야 한다.  $w$ 는  $w = \frac{B}{b} - b_c - y$ 로 주어지고,  $y$ 는  $y = d \cdot \frac{b}{r}$ 이다. 그리고 연속 재생이 보장되기 위해서는  $b_c$ 값이 최소한 1보다 크거나 같아야 하므로  $b_c = \max(1, \frac{rds}{r-bd})$ 로 된다. 한편, 그룹의 최대 개수  $n_{max}$ 는 버퍼의 전체 크기  $B$ 를 그룹별 최소한의 버퍼 크기인  $B_{min}$ 으로 나눈 값이므로 다음 식과 같다.

$$n_{max} = \frac{-Bbd + \sqrt{Bbd(Bbd + 4r^2s)}}{2bdrs}$$

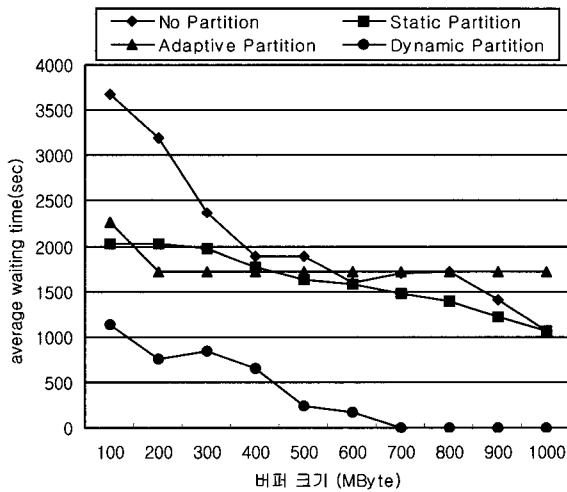
4.2 결과 및 분석

4.2.1 사용자 평균 대기시간

기존의 버퍼 분할 기법과 본 논문에서 제안한 동적 분할 기법에 대해 버퍼 크기, 사용자의 평균 서비스 요청 간격, 그리고 디스크 속도의 변화에 따른 사용자 평균 대기시간을 측정하였다. 평균 대기시간은 디바이스의 종류와 모델에 따라 달라지는 디스크 속도 같은 물리적인 값에 영향을 받으며, 동일한 조건에서 서로 다른 버퍼 분할 기법들을 상호 비교하는 것이 목적이므로 결과 그래프에서 평균 대기시간은 절대 값이 아닌 상대 값으로 표현한다.

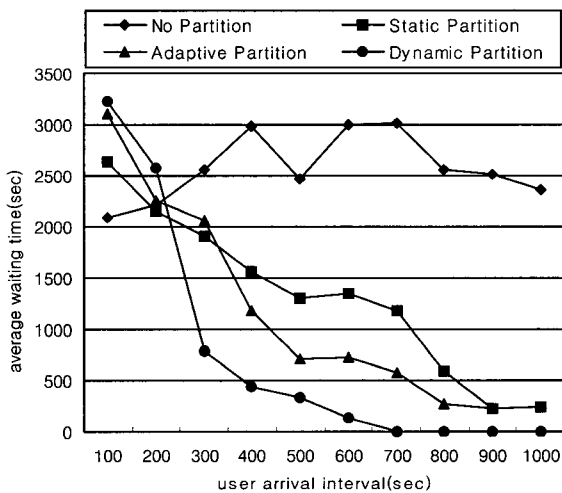
버퍼 크기에 따른 사용자 평균 대기시간의 변화는 (그림 7)과 같다. 동적 분할에서는 버퍼가 커질수록 대기시간이

지속적으로 감소되어 700MB부터는 거의 없게 되며, 평균적으로 기존 방법에 비해 최소한 50%이상 짧아짐을 볼 수 있다. 이것은 저장장치의 대역폭 한계에 서서히 도달하기 때문인 것으로 분석된다. 한편 적응 분할에서는 버퍼 크기가 커짐에 평균 대기시간이 감소하다가 저장장치의 대역폭 한계에 도달하는 버퍼 크기 200MB부터는 더 이상 감소되지 않으며, 버퍼 비분할은 버퍼 크기 600MB 이상에서는 정적 분할이나 적응 분할에 비해 대기시간이 짧은 것으로 나타났다.



(그림 7) 버퍼 크기에 따른 평균 대기시간

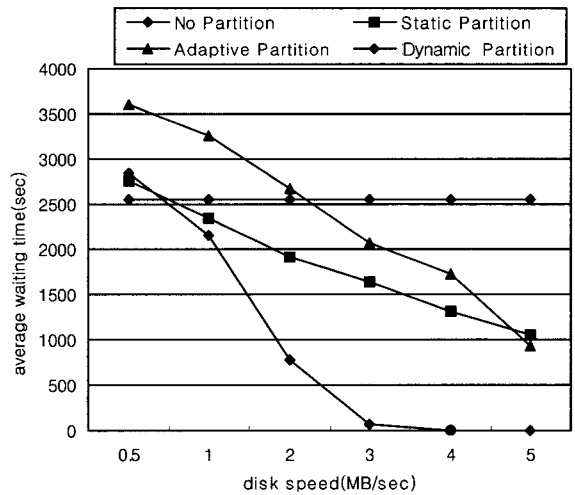
사용자의 평균 서비스 요청 간격에 따른 평균 대기시간은 (그림 8)과 같다. 비분할 방법은 평균 서비스 요청 간격에 무관하게 평균 대기시간이 변하며, 비분할 방법을 제외한 세 가지 버퍼 분할에서는 서비스 요청 간격이 길어질수록 대기시간이 짧아진다. 사용자의 평균 서비스 요청 간격이 작을 때는 적응 분할이 저장장치의 대역폭 한계에 느리게 도달하므로 동적 분할 보다 우수하다가 요청 시간 간격



(그림 8) 평균 서비스 요청 간격에 따른 평균 대기시간

이 길어지면 동적 분할의 대기시간이 짧아짐을 볼 수 있다. 즉 동적 분할은 서비스 요청 간격이 300초보다 긴 경우에는 최소한 50% 정도로 대기시간이 단축됨을 알 수 있다.

디스크 속도에 따른 사용자 평균 대기시간을 나타낸 (그림 9)로부터 버퍼를 분할하지 않고 하나로 사용하는 비분할 기법은 디스크 속도에 무관하게 일정한 대기시간을 갖지만 다른 세 가지 방법은 디스크 속도가 빠를수록 대기시간이 줄어드는 것을 볼 수 있다. 동적 분할에서는 디스크 속도가 빨라질수록 대기시간이 급속히 줄어든다.

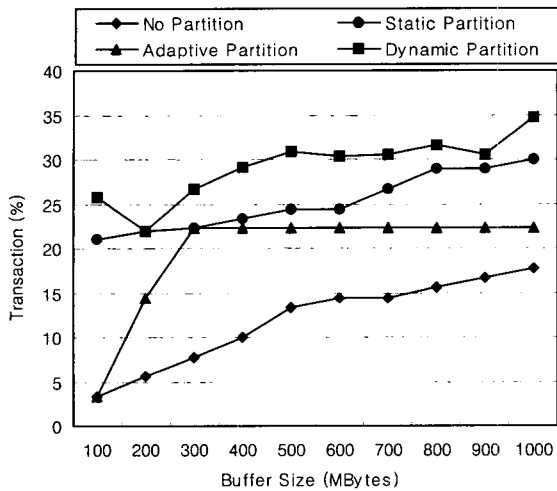


(그림 9) 디스크 속도에 따른 평균 대기시간

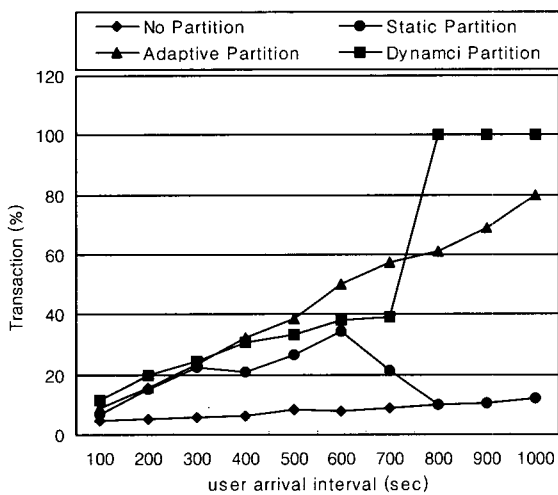
#### 4.2.2 병행 사용자의 수

사용자들은 비디오가 재생되는 동안에 발생하는 화면의 끊김 현상보다 서비스가 시작되기 전에 기다려야 하는 초기 대기시간에 대해 일반적으로 덜 민감하지만, 초기 대기시간이 길어지면 불만족스럽게 생각할 수도 있다. 각 버퍼 분할 기법이 초기 대기시간 없이 지원할 수 있는 병행 사용자의 수를 버퍼 크기, 사용자 평균 서비스 요청 간격, 디스크 속도를 변화시키면서 관찰한 결과는 각각 (그림 10)~(그림 12)와 같다. 그래프에서 세로 축은 전체 사용자 중에서 초기 대기시간 없이 서비스될 수 있는 최대 병행 사용자 수의 비를 나타낸다.

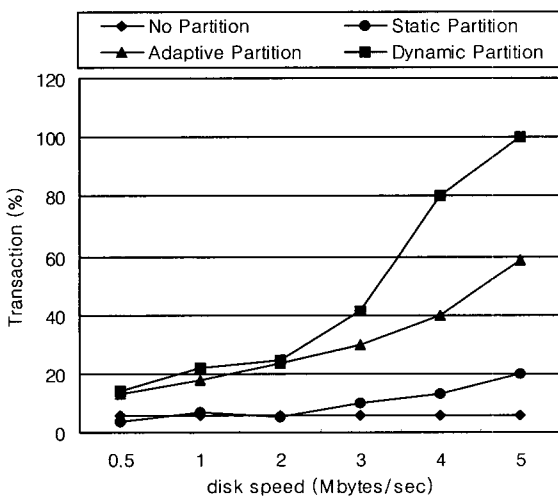
동적 분할에서는 버퍼 크기에 따라 전체 사용자의 22~35%가 초기 대기시간 없이 비디오를 볼 수 있는 반면에 기존의 분할 기법에서는 제일 좋은 경우에도 21~30%로 나타나 1~5%의 차이를 보인다. 사용자 평균 서비스 요청 간격에 따른 병행 사용자의 수는 서비스 요청 간격이 늘어날수록 적응 분할의 경우에는 거의 선형적으로 늘어나지만 동적 분할에서는 요청 간격이 800초 이후에서는 거의 모든 요청이 초기 대기시간 없이 서비스된다. 디스크 속도에 따른 병행 사용자의 수도 디스크 속도가 빨라짐에 따라 서서히 증가하다가 4MB에서는 80%가 초기 대기시간 없이 서비스를 받을 수 있게 된다.



(그림 10) 버퍼 크기에 따른 병행 사용자 수



(그림 11) 평균 서비스 요청 간격에 따른 병행 사용자 수



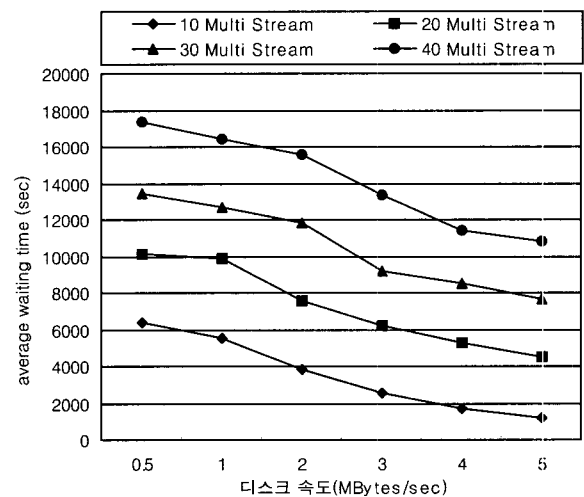
(그림 12) 디스크 속도에 따른 병행 사용자 수

#### 4.2.3 확장 기능의 성능

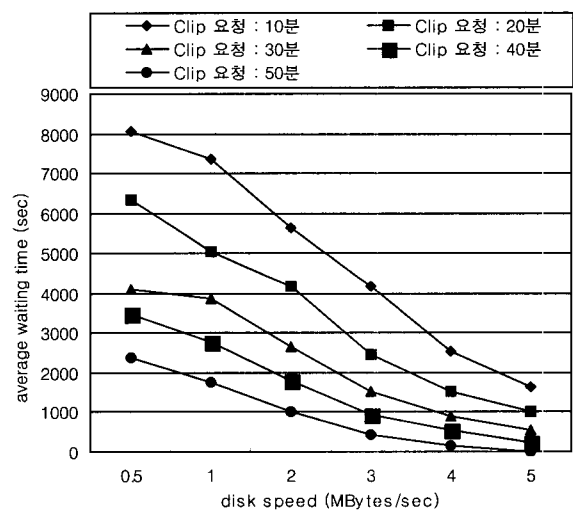
기본 동적 버퍼 분할 기법을 확장하여 다중 스트림, 클립

데이터, VCR 기능을 처리할 때의 성능을 본다. 시뮬레이션에 사용된 파라미터로는 버퍼 크기 250MB, 사용자 평균 서비스 요청 간격 300초, 그리고 디스크 속도는 2MB/sec이며, 전체 시뮬레이션 시간은 8시간이다. 여기서도 버퍼 분할 방법의 비교에서와 마찬가지로 결과를 나타내는 평균 대기시간은 절대 값이 아닌 상대 값이다.

디스크 속도와 다중 스트림의 개수를 변화시켰을 때의 사용자의 평균 대기시간은 (그림 13)과 같다. 다중 스트림의 개수에 무관하게 전체적으로 디스크 속도가 증가할수록 평균 대기시간이 짧아지며, 동일한 디스크 속도에서는 스트림이 10개일 때의 대기시간은 40개인 여우에 비해 최소 10%에서 최대 34%로 나타났다. 다중 스트림의 수가 증가할수록 병행 사용자들의 평균 대기시간이 증가한다.



(그림 13) 다중 스트림 개수와 디스크 속도에 따른 평균 대기 시간



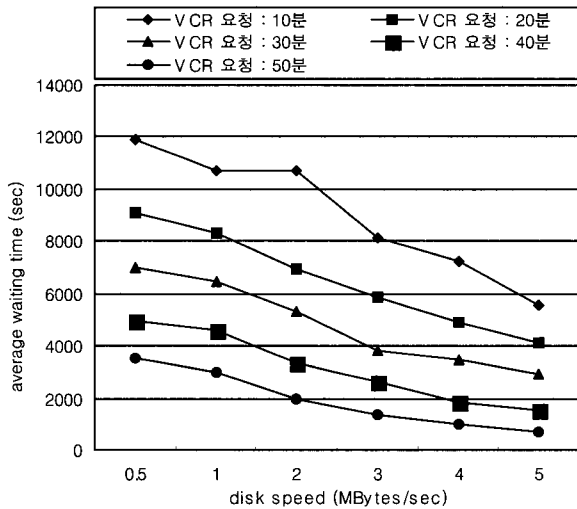
(그림 14) 클립 데이터 요청 간격과 디스크 속도에 따른 평균 대기시간

클립 데이터의 크기를 5초에서 1분으로 하고 10개의 연



속 미디어 스트림을 클립 데이터와 함께 처리할 때, 클립 데이터를 요청하는 평균 시간 간격에 따른 사용자 평균 대기시간의 변화는 (그림 14)와 같다. 클립 데이터를 요청하는 시간 간격이 짧으면, 즉 클립 데이터 요청이 빈번할수록 새로운 그룹의 형성이 많아지게 되므로 병행 사용자들의 평균 대기시간은 증가된다.

VCR 기능들이 후진, 일시정지, 재개, 고속 전진, 고속 후진의 평균 서비스 요청 간격을 10분에서 50분까지 변화시키면서 디스크 속도에 따른 평균 대기시간을 측정된 결과는 (그림 15)와 같다. 이때 각 기능의 유지 시간은 2분으로 하여 2분이 지나면 다시 재생 연산이 실행된다. (그림 15)에서 디스크 속도가 빠를수록 그리고 VCR 기능을 요청하는 시간 간격이 클수록 병행 사용자들의 대기시간이 작아짐을 볼 수 있다.



(그림 15) VCR 기능 요청과 디스크 속도에 따른 평균 대기시간

### 5. 결 론

통신 기술과 비디오 압축 기술의 급속한 발전에 힘입어 네트워크를 통한 실시간 비디오 서비스가 보편화되고 있다. 사용자가 요청한 비디오 데이터를 네트워크로 제공하는 VOD 서버는 크기가 큰 데이터의 일부를 미리 서버 버퍼로 읽어 들여 연속적인 서비스가 보장되도록 관리한다. 이때 저장장치로부터 서버 버퍼로 읽어 들인 연속미디어 데이터를 다른 사용자가 재사용하면 저장장치로의 접근 횟수가 줄어들고 사용자의 대기시간은 감소된다.

본 논문에서는 서버의 버퍼를 그룹으로 분할하고 각 그룹에서 활용도가 가장 낮은 버퍼 공간을 동적으로 재분할하여 새로운 사용자가 사용하도록 하는 동적 버퍼 분할 기법을 보였다. 먼저 하나의 비디오를 대상으로 전체 버퍼를 하나의 그룹으로 관리하다가 각 그룹에서 활용도가 낮은 버퍼 공간을 재분할하여 사용하는 기본 알고리즘을 제시하고, 이를 다중 스트림에 적용하고 클립 데이터와 VCR 기능을 처리할 수 있도록 확장하였다. 제안된 기법은 각 그룹

에서 활용도가 낮은 버퍼 공간을 동적으로 재분할하여 다시 사용함으로써 버퍼 활용도가 높아져 병행 사용자의 수가 증가되고 저장장치의 대역폭 한계에 서서히 도달하여 사용자의 평균 대기시간이 감소된다.

버퍼 크기, 사용자 평균 서비스 요청 간격, 디스크 속도를 변화시키면서 시뮬레이션을 통해 동적 버퍼 분할 기법과 기존의 버퍼 분할 기법에 대한 사용자 평균 대기시간과 초기 대기시간 없이 지원할 수 있는 병행 트랜잭션의 수를 관찰하였다. 실험 결과에서 평균 대기시간은 최소한 50% 이상 감소되고 초기 대기시간 없이 지원할 수 있는 병행 사용자의 수는 1~5% 증가함을 확인하였다.

### 참 고 문 헌

- [1] D. Jadav and A. Houdhary, "Design Issues in High Performance Media-on-Demand Servers," IEEE Parallel and Distributed Technology Systems and Applications, Summer 1995.
- [2] B. Ozden, R. Rastogi, and A. Silberschatz, "Buffer Replacement Algorithms for Multimedia Storage Systems," IEEE International Conference on Multimedia Computing and System '96, June, 1996.
- [3] A. Dan and D. Sitaram, "Buffer Management Policy for an On-Demand Video Server," Technical Report RC19347, IBM Research Report, 1994.
- [4] D. Sitaram and A. Dan, "Multimedia Servers : Applications, Environment, and Design," Morgan Kaufmann Publishers, 2000.
- [5] D. Rotem and J. L. Zhao, "Buffer Management for Video Database Systems," Proceedings of International Conference on Data Engineering, Taipei, Taiwan, pp.439-448, March, 1995.
- [6] 이상호, 문양세, 황규영, 조완섭, "주문형 비디오 시스템에서의 동적 버퍼 할당 기법", 정보과학회논문지 : 시스템 및 이론, 제28권 제9·10호, pp.442-460, 2001.
- [7] 장은정, 최황규, "A Dynamic Buffer Partitioning Technique for Continuous Media Data in Multimedia Storage Server," ITC-CSCC '98, 대한전자공학회, 1998.
- [8] S. Sheu, K. A. Hua, and W. Tavanapong, "Dynamic Grouping : An Efficient Buffer Management Scheme for Video-on-Demand Servers," Technical Report CS-TR-97-02, University of Central Florida, Orlando Florida, Feb., 1997.
- [9] A. Dan and D. Sitaram, "A Generalized Interval Caching Policy for Mixed Interactive and Long Video Workloads," Proceedings of Multimedia Computing and Networking, SPIE, 1996.
- [10] M. H. MacDougall, Simulating Computer Systems, The MIT Press.
- [11] A. L. N. Reddy and J. C. Wyllie, "I/O Issues in a Multimedia System," IEEE Computer, Vol.27, No.3, pp.69-74, March, 1994.
- [12] B. Ozden, R. Rastogi, and A. Silberschatz, "Research Issues in Multimedia Storage Servers," ACM Computing Survey,

December, 1995.

- [13] M. Chen, D. D. Kandlur, and P. S. Yu, "Support for Fully Interactive Playout in Disk-Array-Based Video Server," Proceedings of the Second ACM International Conference on Multimedia '94, San Francisco, CA, pp.391-398, October, 1994.
- [14] 장은정, 최황규, "동적 버퍼 분할에 의한 연속미디어 데이터 버퍼 관리 방법", 98 하계 종합학술대회논문집, 한국통신학회, 1998.
- [15] 권택근, 이석호, "연속미디어를 위한 FFU 버퍼 재배치 알고리즘", 한국정보과학회논문지, 제22권 제10호, 1995.
- [16] D. J. Makaroff and R. T. Ng., "Schemes for Implementing Buffer Sharing in Continuous-Media Systems," Information Systems, pp.445-464, 1995.



**권 춘 자**

e-mail : kwoncj@mail.kangwon.ac.kr

1986년 한양대학교 전자공학과(학사)  
 1991년 한양대학교 전자계산학과(석사)  
 2000년~현재 강원대학교 컴퓨터정보통신  
 공학과 박사과정  
 2002년~현재 한림정보산업대학 전산정보  
 처리과 초빙교수

관심분야 : 멀티미디어 VOD 시스템, 데이터베이스 시스템, 클러스터 웹 서버 시스템 등

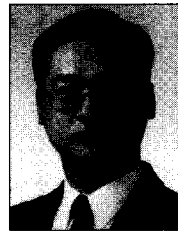


**최 창 열**

e-mail : cychoi@cc.kngwon.ac.kr

1979년 경북대학교 전자공학과(학사)  
 1981년 경북대학교 전자공학과(석사)  
 1995년 서울대학교 컴퓨터공학과(박사)  
 1984년~1996년 한국전자통신연구원 컴퓨  
 터연구단 책임연구원/연구실장

1996년~현재 강원대학교 전기전자정보통신공학부 교수  
 관심분야 : 컴퓨터구조, 멀티미디어시스템, 이동컴퓨팅 등



**최 황 규**

e-mail : hkchoi@kangwon.ac.kr

1984년 경북대학교 전자공학과(학사)  
 1986년 한국과학기술원 전기및전자공학과  
 (석사)  
 1989년 한국과학기술원 전기및전자공학과  
 (박사)

1994년~1995년 Univ. of Florida Database R&D Center 방문  
 교수

1999년~2001년 강원대학교 전자계산소 소장  
 1990년~현재 강원대학교 전기전자정보통신공학부 교수  
 관심분야 : 멀티미디어 시스템, 데이터베이스 시스템, 멀티미디  
 어 정보검색, 병렬 I/O 시스템 등