

# VOD 서비스를 위한 채널 예약 배치-패칭 방법의 설계 및 평가

하속정<sup>†</sup>, 이경숙<sup>\*\*</sup>, 배인한<sup>\*\*\*</sup>

## 요 약

VOD 시스템에서 비디오 서버가 지원할 수 있는 채널의 개수는 서버의 통신 대역폭에 의해 결정되므로 비디오 서버의 가용채널 수는 제한되어 있다. 멀티캐스트 데이터를 공유함으로써 비디오 서버의 네트워크 입출력 대역폭을 절감하기 위해 일괄처리, 패칭, 배치-패칭과 같은 멀티캐스트 방법들이 제안되었다. 본 논문에서는 인기 비디오 요청들에 대해 먼저 배치 방법을 적용한 후 배치들에 대해 패칭 방법을 수행하는 채널 예약 배치-패칭 방법을 제안한다. 그리고 제안하는 방법은 많은 인기 비디오 요청을 위하여 비디오 서버의 채널 용량 중 일부를 예약하므로 인기 비디오 요청들을 이탈없이 서비스할 수 있다. 제안하는 방법의 성능을 시뮬레이션을 통하여 평균 서비스 지연 시간, 이탈율, 공평성, 프레임 절감을 항목에 대해 패칭 방법과 배치-패칭 방법의 성능과 비교한다.

## Design and Evaluation of a Channel Reservation Batch-Patching Technique for VOD Services

Sook-Jeong Ha<sup>†</sup>, Kyung-Sook Lee<sup>\*\*</sup>, Ihn-Han Bae<sup>\*\*\*</sup>

## ABSTRACT

In VOD systems the number of channels available to a video server is limited since the number of channels, the video server can support, is determined by the communication bandwidth of the video server. Multicast techniques such as batching, patching and batch-patching have been proposed to reduce I/O demand on the video server by sharing multicast data. In this paper, we propose a channel reservation batch-patching technique that first applies the batching technique to hot video requests and then applies the patching technique to the batches of the hot video requests. And the proposed technique reserves a part of the channel capacity of the video server for many hot video requests, so that the hot video requests can be served without deflection. The performance of the proposed technique is compared with those of Patching and Batch-Patching techniques in terms of average service latency, deflection rate, fairness and frame reduction rate by simulations.

**Key words:** VOD, Communication bandwidth(통신 대역폭), sharing of data(데이터 공유), Multicast(멀티캐스트), Batching(배치), Patching(패칭), Batch-Patching(배치-패칭), Channel Reservation(채널 예약), hot video(인기 비디오)

※ 교신저자(Corresponding Author): 하속정, 주소: 대구광역시 북구 산격3동 1370번지(702-701), 전화: 053)950-7334, FAX: 053)940-8582, E-mail: sjha@m80.knu.ac.kr  
접수일: 2003년 4월 28일, 완료일: 2003년 8월 27일

<sup>†</sup> 경북대학교 전자전기컴퓨터학부 조빙교수

<sup>\*\*</sup> 경북대학교 전자전기컴퓨터학부 조빙교수

(E-mail: lks8682@hotmail.com)

<sup>\*\*\*</sup> 대구가톨릭대학교 컴퓨터정보통신공학부 교수

(E-mail: ihbac@cu.ac.kr)

## 1. 서 론

고속 네트워크 하부구조와 최첨단 디지털 비디오 기술은 라이브 비디오 방송, 원격교육 제한지역 유선 방송(narrowcast), 비디오 클립 스트리밍과 같은 네트워크 상에서 지원되는 여러 가지 멀티미디어 응용들의 실현을 가능하게 하였다[1]. 특히 VOD(Video on Demand)는 디지털 비디오 도서관, 주문형 뉴스,

원거리 교육, 전자 상업과 같은 많은 멀티미디어 응용을 위한 중요한 기술이다[1,2]. 전형적인 VOD 서비스는 서버 상에 대량의 비디오들이 저장되어 있으며, 원거리의 사용자가 장소와 시간에 제한 받지 않고 원하는 비디오를 서버에 요청함으로써 서버로부터 전송되는 비디오를 재생해서 보도록 해준다. 비디오 스트림은 비록 압축이 되었다라도 대역폭 요구량이 높으며, 재생시간이 길고, 엄격한 응답시간을 요구하며, 연속적으로 재생되어야 하므로 전송 비용이 많이 든다. 하나의 비디오 스트림을 유지하는데 필요한 시스템 자원을 비디오 채널이라고 하는데 일반적으로 VOD 시스템에서 비디오 서버의 가용채널 수는 통신 대역폭에 의해 결정되므로 한정되어 있다[2,3]. 비디오 서버의 가용채널수가 한정되어 있으므로 여러 사용자들의 비동기적 비디오 요청에 대해 경제적이며 짧은 서비스 지연시간으로 보다 많은 요청을 서비스할 수 있는 기법이 요구되었으며, 이를 위해 여러 사용자들이 멀티캐스트를 통해 비디오 스트림을 공유하는 방법들이 제안되었다[2-6]. 특히 배치-패칭[4]은 배치와 패칭을 혼합한 방법으로 사용자들의 비디오 요청은 일단 짧은 배치 윈도우 기간동안 모아졌다가 동일 비디오에 대한 요청 집합인 배치 단위로 패칭[2] 방법이 적용된다.

본 논문은 VOD 서비스에서 뉴스, 상품 광고, 의학, 텔레비전 쇼, 필수적인 원격 교육 콘텐츠 등과 같은 인기 비디오의 공유에 초점을 둔다. 사용자들의 비디오 접근 형태를 살펴보면 모든 비디오를 균등하게 접근하는 것이 아니라 접근 빈도에 차이가 있는 것을 알 수 있다. 즉, 특정 인기 비디오의 접근 빈도는 높고 인기가 없는 비디오에 대한 접근 빈도는 낮다. 이와 같은 특정 비디오의 인기도 편중으로 짧은 기간에도 인기 비디오에 대한 새로운 요청은 많이 도착하게 된다. 본 논문은 이와 같이 자주 도착하는 인기 비디오 요청의 이탈율을 낮추어 VOD 시스템의 처리력을 향상시키며 배치-패칭에서 발생하는 비인기 비디오 요청의 불필요한 서비스 지연시간을 개선시키기 위한 채널 예약 배치-패칭 방법을 제안한다.

제안하는 방법은 사용자가 요청한 비디오를 인기와 비인기 비디오로 구분하여 요청별로 다른 서비스 방법을 제공한다. 배치 윈도우 동안 도착가능한 요청이 한 개보다 많은 즉 인기 비디오 요청에 대해서는 배치 윈도우 기간동안 요청을 모아두었다가 배치별로 패칭 방법을 적용함으로써 채널 상의 비디오 스트

림의 공유와 서버의 네트워크 대역폭 요구량을 개선시킨다. 반면에 도착 간격이 긴 즉 배치 윈도우 동안 하나 이하의 요청이 도착하는 비인기 비디오 요청들은 일단 배치 윈도우 동안 모아두는 것이 아니라 도착하는 즉시 패칭 방법을 적용하여 서비스 지연시간을 개선한다. 그리고 사용자 요청의 대부분을 차지하는 인기 비디오 요청들이 서버가 과부하인 상태에서도 이탈하지 않고 서비스를 받을 수 있도록 비디오 서버의 전체 채널 용량 중에서 일부를 인기 비디오 전용으로 예약한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구로 배치, 패칭, 배치-패칭에 관하여 소개하고, 3장에서는 제안하는 채널 예약 배치-패칭 방법을 설계하고, 4장에서는 시뮬레이션을 통해 제안하는 방법의 성능을 평가한 후, 5장에서 결론을 맺는다.

## 2. 관련 연구

VOD 서비스의 성능을 개선하기 위해 비디오 서버의 네트워크 채널을 효율적으로 사용하기 위한 여러 가지 방법들이 제안되었다. 여러 비디오 요청자들에게 하나의 채널로 동일 비디오 스트림을 멀티캐스트함으로써 여러 요청자간에 비디오 스트림을 공유하는 대표적인 멀티캐스트 방법들로 배치, 피기백킹(piggybacking), 패칭, 배치-패칭 등이 있다.

배치[5]은 사용자의 비디오 요청을 도착하는 즉시 서비스하는 것이 아니라 배치 윈도우에 해당하는 일정 기간동안 모아둔다. 이렇게 모아진 배치 윈도우안의 요청 중에서 동일 비디오에 대한 사용자 요청들의 집합을 배치라고 한다. 비디오 서버는 하나의 배치에 포함된 여러 사용자들의 요청에 대해 한 채널로 하나의 비디오 스트림만 멀티캐스트하여 서비스함으로써 채널을 효율적으로 사용할 수 있다. 그러나 배치는 사용자의 요청을 도착하는 즉시 서비스하지 않으므로 서비스 지연시간이 생기게 된다. 그림 1은 비디오 서버가 충분한 가용채널을 가지고 있으며 배치 윈도우의 크기가  $BW$ 일 때 비디오 서버에 차례대로 도착한 비디오  $i$ 에 대한 다섯 개의 요청  $R_1, R_2, R_3, R_4, R_5$ 가 배치에 의해 서비스되는 예를 보여준다. 배치 윈도우,  $BW$  시간 동안 모아진 요청  $R_1$ 과  $R_2$ 를 포함한  $Batch_1$ 을 위해 가용채널  $CH_1$ 으로 비디오  $i$ 의 전체 데이터가 멀티캐스트됨으로써 두 요청의 사용자가 동시에 서비스를 받게 된다. 새로운 배치 윈

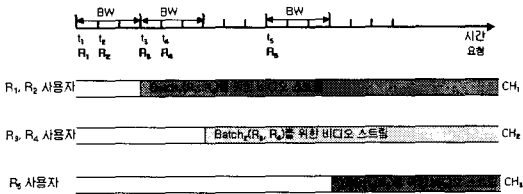


그림 1. 배치의 예

도우 동안 모아진 요청  $R_3$ 와  $R_4$ 를 포함한  $Batch_2$ 를 위해 가용채널  $CH_2$ 로 비디오  $i$ 가 멀티캐스트되며  $t_5$ 에 도착한 요청  $R_5$ 의 경우는 배치 윈도우 동안 다른 요청이 도착하지 않았으므로 요청  $R_5$ 만 포함한  $Batch_3$ 에 대해 멀티캐스트가 수행된다.

피기백킹[6]은 진행중인 비디오 스트림의 재생율을 조정하여 동일 비디오를 요청한 사용자들을 위한 스트림들을 하나의 스트림으로 합병하는 방법이다. 피기백킹은 배치보다 서비스 지연시간이 짧지만, 피기백킹의 변화가  $\pm 5\%$  이내여서 합병할 수 있는 스트림의 수가 제한적이며 특별한 하드웨어 환경이 필요하다.

최근 사용자 스테이션은 고성능화되어 서버로부터 동시에 두 개의 비디오 스트림을 전송받을 수 있을 만큼의 대역폭을 가지고 있으며, 몇 분의 비디오 스트림을 버퍼링할 수 있을 만큼의 충분한 디스크 공간을 가지게 되었다[3]. 패칭[2,3]은 이러한 사용자 스테이션의 고성능을 기반으로 사용자에게 지연시간 없이 서비스를 할 수 있으며 서버의 채널을 효율적으로 사용할 수 있도록 제안된 방법이다. 패칭은 새롭게 서비스할 사용자를 기존의 멀티캐스트에 합병(join)함으로써 멀티캐스트되고 있는 비디오 스트림의 공유를 증가시킨다. 서버의 네트워크 채널은 한 개의 비디오 데이터 전체를 멀티캐스트하는 정규 채널이나 비디오의 시작부터 일부분만을 멀티캐스트하는 패칭 채널로서 사용될 수 있다. 정규 채널과 패칭 채널을 통해 사용자가 수신하는 데이터 스트림들을 각각 정규 스트림과 패칭 스트림이라고 한다. 정규 멀티캐스트가 시작된 이후 일정 기간 안에 새로 도착한 동일 비디오에 대한 요청에 대해서는 패칭 멀티캐스트가 수행되는데 이 기간을 패칭 윈도우라고 한다. 즉 동일 비디오에 대해 최근의 정규 멀티캐스트가 시작된 후 패칭 윈도우 안에 도착하여 서비스될 요청에 대해서서 비디오 서버는 이 정규 멀티캐스트에서 이미 전송해버린 비디오 앞부분만을 새로운 채

널 즉 패칭 채널을 통해 전송하며, 이와 동시에 사용자가 뒷부분의 비디오 스트림을 연속적으로 상영할 수 있도록 이 새로운 사용자를 진행중인 최근의 정규 멀티캐스트에 동적으로 합병시킨다. 이와 같이 패칭은 많은 사용자들을 기존의 멀티캐스트에 합병시킴으로써 멀티캐스트의 효율을 증가시키며 더욱이 비디오 요청을 일정기간 동안 모아두지 않고 도착하는 즉시 서비스하도록 함으로써 서비스 지연시간을 감소시킬 수 있다.

그림 2는 패칭 윈도우의 크기가  $PW$ 이며 그림 1과 동일한 조건에서 패칭 방법이 적용될 때 각 요청의 서비스 과정을 보여준다. 첫 번째 요청  $R_1$ 에 대해서는 진행중인 정규 멀티캐스트가 없으므로 비디오  $i$ 의 전체 데이터가 채널  $CH_1$ 상으로 정규 멀티캐스트된다. 요청  $R_2, R_3, R_4$ 의 경우는 모두가 최근의 요청  $R_1$ 을 위한 정규 멀티캐스트의 패칭 윈도우 안에 있으므로 각자의 스큐에 해당하는 패칭 스트림만이 패칭 멀티캐스트된다. 여기서 스큐란 정규 멀티캐스트의 시작 시간과 새 요청의 서비스 시작 시간과의 시간 차이이다. 요청  $R_2, R_3, R_4$ 의 사용자는 패칭 스트림을 다운로드받는 동시에 패칭 스트림 이후의 비디오를 연속해서 상영할 수 있도록 비디오  $i$ 의  $R_1$ 을 위한 정규 스트림을 미리 버퍼링한다. 패칭 스트림을 모두 상영한 후에는 각자의 버퍼에 미리 저장해둔 정규 스트림을 상영한다. 마지막으로 요청  $R_5$ 는 최근의 요청  $R_1$ 을 위한 정규 멀티캐스트의 패칭 윈도우를 벗어났으므로 새롭게 비디오  $i$ 의 전체 데이터를 전송하는 정규 멀티캐스트가 발생한다.

배치-패칭[4] 방법은 배치의 서비스 지연시간을 개선시키기 위해 배치와 패칭을 조합한 방법이다. 사용자들간에 비디오 스트림의 공유를 증가시키기 위

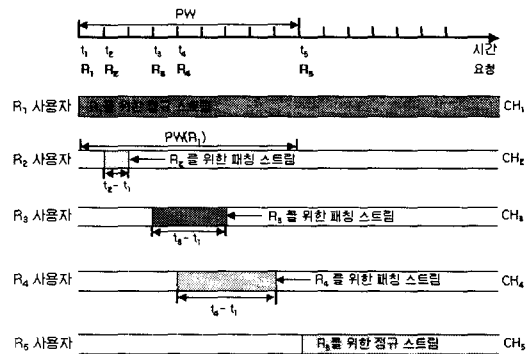


그림 2. 패칭의 예

하여 단일 사용자 요청 단위로 패칭 방법을 적용하는 것이 아니라 배치 단위로 패칭 방법을 적용한다. 비디오 서버는 사용자의 비디오 요청이 도착하자마자 서비스하는 것이 아니라 짧은 배칭 윈도우 기간동안 모아진 배치에 대해 하나의 비디오 스트림을 동일 채널상으로 멀티캐스트하여 서비스한다. 그리고 동일한 비디오에 대한 새로운 배치의 서비스 시점이 가장 최근에 개시된 기존의 정규 멀티캐스트의 패칭 윈도우 안에 속한다면 배치에 포함된 모든 사용자에게 패칭 멀티캐스트를 수행한다. 그러므로 하나의 패칭 채널로 서비스할 수 있는 사용자의 수가 증가된다.

그림 3은 패칭 윈도우의 크기가  $PW$ 이며 그림 1과 같은 조건에서 배치-패칭이 적용될 때 각 요청의 서비스 과정을 보여준다. 배칭 윈도우 동안 모아진 비디오  $i$ 에 대한 요청  $R_1$ 과  $R_2$ 를 포함한  $Batch_1$ 에 대해 선 진행중인 정규 멀티캐스트가 존재하지 않으므로 배칭 윈도우 기간이 끝나자마자 채널  $CH_1$ 으로 비디오  $i$ 의 전체 데이터를 요청  $R_1$ 과  $R_2$ 의 사용자에게 정규 멀티캐스트함으로써 두 개의 요청을 동시에 서비스한다. 다음 배칭 윈도우 기간동안 모아진 요청  $R_3$ 와  $R_4$ 를 포함한  $Batch_2$ 의 서비스 시점은  $Batch_1$ 을 위한 정규 멀티캐스트의 패칭 윈도우 안에 있으므로 스큐인  $t_3-t_1$ 동안 비디오의 앞부분이 새로운 채널  $CH_2$ 로 전송된다. 그러나 요청  $R_5$ 만을 포함한  $Batch_3$ 의 서비스 시점은  $Batch_1$ 을 위한 최근의 정규 멀티캐스트의 패칭 윈도우를 벗어났으므로 비디오  $i$ 의 전체 데이터가 채널  $CH_3$ 를 통해 정규 멀티캐스트된다.

패칭을 이용하도 다른 기법으로  $P2Cast$ [7]는 유니캐스트 연결에만 의존하는 동등계층(peer-to-peer) 접근에 기반을 두고 있다. 비디오를 요청하는 사용자 스테이션은 비디오 서버의 부하를 줄일 수 있도록 계산 능력과 비디오 데이터를 캐시하기 위한 충분한 저장 공간을 가져야 한다. 패칭 윈도우 안에 도착한 사용자 요청들은 하나의 세션을 형성하고

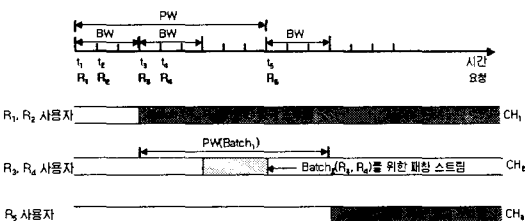


그림 3. 배치-패칭의 예

이 세션 안의  $P2Cast$  사용자들은  $Best-Fit$  알고리즘을 사용하여 비디오 서버를 루트로 하는 응용 수준의 멀티캐스트 트리를 형성한다. 이 멀티캐스트 트리의 루트 바로 밑의 사용자는 서버로부터 정규 스트림을 전송받으며, 이 정규 스트림은 나머지 사용자에게 전송된다. 정규 멀티캐스트가 아닌 패칭 멀티캐스트로 서비스되어야 하는 사용자들은 정규 스트림 외에도 패칭 스트림을 전송받아야 하며 패칭 서버는  $Best-Fit$  알고리즘을 사용하여 현재 서비스할 사용자에게 가장 적절한 것으로 선택된다.  $P2Cast$ 는 패칭과 달리 사용자가 요청 스트림을 수신하는 것뿐만 아니라 다른 사용자에게 정규 스트림과 패칭 스트림을 전송해야 하며, 패칭 서버의 역할을 하기 위해 비디오 시작 부분의 데이터를 캐시하고 있어야 하며, 한 세션 안에 새로운 사용자가 추가될 때마다 멀티캐스트 트리 형성을 위한 처리과정을 거쳐야 한다. 사용자가 비디오 데이터의 시작 부분을 10% 이상 캐시할 수 있을 때 그리고 서버의 부하가 높을 때 패칭보다 요청 거절율이 조금 우수한 것으로 결과가 나왔으며, 한 세션 안의 사용자 스테이션에서 결합이 발생하는 경우 멀티캐스트 트리 안의 다른 사용자 스테이션에서 비디오를 연속적으로 재생할 수 없게 되므로 이를 위해 결합 복구 능력이 요구된다.

#### 4. 채널 예약 배치-패칭 방법의 설계

배치-패칭에서 비디오 서버의 모든 비디오를 배칭 윈도우  $BW$ 동안 요청이 도착 가능한 확률이 1 이상인 비디오 집합  $V_1$ 과 비디오 요청이 도착 가능한 확률이 1 이하인 비디오 집합  $V_3$ 로 구분해서 두 종류의 비디오별로 서비스 지연시간을 분석하면 다음과 같은 결과를 얻을 수 있다.  $V_1$ 에 속한 비디오 요청들은 배칭 윈도우 동안 모아진 배치에 여러 개의 요청이 존재하므로 여러 개의 비디오 요청들에 대해 하나의 비디오 스트림을 멀티캐스트함으로써 비디오 스트림의 공유가 가능하다. 즉 그림 4와 같이 배칭 윈도우 동안 모아진 요청  $R_1, R_2, R_3$ 를 포함한 배치에 대

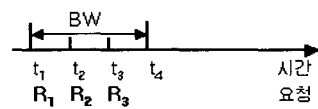


그림 4. 배칭 윈도우 안에 동일 비디오 요청이 3개 있는 경우

해 하나의 채널로 요청한 비디오 데이터만 멀티캐스트하면 되는 것이다. 그러나  $V_s$ 에 속한 비디오 요청들은 배치 윈도우 동안 모아진 배치 안에 많아야 하나의 비디오 요청만 존재하므로 비디오 스트림의 공유가 불가능하다. 그림 5와 같이 비디오 요청  $R_i$ 는 동일한 비디오를 요청하는 다른 요청이 도착하지 않음에도 불구하고 배치 윈도우가 끝날 때까지 대기해야만 한다. 즉  $V_s$ 에 속한 비디오 요청들은 배치로 인한 이득을 전혀 얻을 수 없음에도 불구하고 사용자의 요청은 최악의 경우 배치 윈도우 동안 대기한 후에 서비스가 이루어지는 것이다. 그러므로 사용자의 요청 중에서  $V_i$ 에 속한 비디오 요청에 대해서만 배치 후에 패칭 방법을 적용하고  $V_s$ 에 속한 비디오 요청은 도착하자마자 패칭 방법을 적용한다면 배치에서의 서비스 지연시간을 개선할 수 있을 것이다.

배치-패칭 방법의 경우 패칭 윈도우가  $PW$ 일 때 현재 시점  $t$ 에서 새로운 배치를 서비스하기 위해 패칭 멀티캐스트가 수행되기 위해서는  $t-PW$ 와  $t$  사이에 시작된 동일 비디오에 대한 정규 멀티캐스트가 존재해야 한다. 패칭 멀티캐스트는 정규 멀티캐스트가 존재하는 경우에만 발생할 수 있기 때문이다. 그러나 이러한 진행중인 정규 멀티캐스트가 존재하여 새로운 배치에 대해 패칭 멀티캐스트가 가능하더라도 비디오 서버에 가용채널이 없다면 이 새로운 배치는 즉시 스케줄되지 못하고 대기 큐에서 기다려야 하며, 이 후에 가용채널이 생겨 스케줄되더라도 이 시점이 이미 최근의 정규 멀티캐스트의 패칭 윈도우를 벗어난 경우에는 새로운 배치에 대해 다시 정규 멀티캐스트가 발생되어야 하는 상황이 될 수 있다. 그러므로 특정 비디오에 대한 요청들이 패칭 멀티캐스트됨으로써 이득을 획득하기 위해서는 이 비디오에 대한 배치가 즉시 스케줄되어 서비스될 수 있도록 가용채널이 존재해야 한다. 그러나 비디오 서버의 채널은 한정되어 있으므로 배치를 즉시 서비스할 수 있는 가용채널이 항상 존재하지 않을 수 있다. 이에 대한 대안으로 요청 확률이 높은 비디오 요청들에 대해서만 항상 서비스될 수 있도록 비디오 서버의

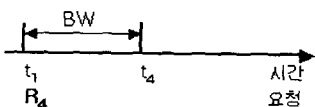


그림 5. 배치 윈도우 안에 동일 비디오 요청이 1개만 있는 경우

전체 채널 용량 중에서 일부를 미리 예약하는 방법을 고려할 수 있다.

그러므로 본 논문에서는 사용자가 요청한 비디오를 인기와 비인기 비디오로 구분하고 비디오 서버의 일부 채널을 인기 비디오 요청 전용으로 예약하는 채널 예약 배치-패칭 방법을 제안한다. 배치 윈도우 동안 한 개 보다 많은 요청이 도착하는 비디오 즉 비디오 집합  $V_i$ 에 속하는 비디오를 인기 비디오로 간주하며, 인기 비디오 요청에 대해서는 배치-패칭과 같이 배치 윈도우 기간동안 요청을 모아두었다가 배치별로 패칭 방법을 적용한다. 반면에 도착 간격이 긴 즉 배치 윈도우 동안 최대 하나의 요청이 도착하는 비디오 집합  $V_s$ 에 속하는 비디오를 비인기 비디오로 간주하며, 비인기 비디오 요청들은 도착하는 즉시 패칭 방법을 적용하여 배치 방법에서의 서비스 지연시간을 개선한다. 그리고 비디오 스트림의 공유가 많이 일어나는 인기 비디오 요청들에 대해서는 서버가 과부하인 상태에서도 이탈없이 서비스되어 VOD 시스템의 처리력을 개선시킬 수 있도록 비디오 서버의 전체 채널 용량 중에서 일부를 인기 비디오 요청 전용으로 예약한다.  $BW$ 는 배치 윈도우의 크기(분),  $L$ 은 비디오의 길이(분),  $p_i$ 는 전체 비디오 요청 중에서 비디오  $i$ 가 요청될 확률,  $PW$ 는 패칭 윈도우의 크기(분),  $\lambda$ 는 요청 도착률(분당 도착한 사용자의 비디오 요청수)이라고 하자. 배치 윈도우  $BW$ 분 안에 도착 가능한 비디오  $i$ 의 요청수는  $p_i \times \lambda \times BW$  이므로 인기 비디오는 식 1을 만족하는 비디오이며 나머지 비디오는 비인기 비디오가 된다.

$$p_i \times \lambda \times BW > 1.0 \tag{1}$$

정규 멀티캐스트가 시작된 후 패칭 윈도우만큼의 시간이 지나면 새로운 정규 멀티캐스트가 수행되어야 하므로 비디오 전체 재생시간동안 발생할 수 있는 정규 멀티캐스트의 수는 비디오 길이  $L$ 분 안에 포함된 패칭 윈도우 수인  $(L/PW)$ 이다. 하나의 정규 채널이 비디오 길이  $L$ 분 동안 사용된 후에는 가용 상태가 되어 다시 정규 멀티캐스트에 사용될 수 있으므로 비디오  $i$ 에 대한 배치를 즉시 서비스하기 위해 필요한 최대 정규 채널의 수는  $(L/PW)$ 이다. 그러므로  $k$ 개의 인기 비디오를 위해 예약해야 할 최대 정규 채널 용량은 식 2와 같다.

$$k \times \left(\frac{L}{PW}\right) \tag{2}$$

패칭 윈도우  $PW$  안에는 배칭 윈도우  $BW$ 가 ( $PW/BW$ ) 번 포함될 수 있다. 배칭 윈도우 동안 생성된 비디오  $i$ 에 대한 배치마다 하나의 패칭 채널이 필요하며, 새로운 패칭 윈도우의 시작지점에서는 정규 멀티캐스트가 발생하므로, 패칭 윈도우 기간동안 배치들이 즉시 서비스되기 위해 필요한 패칭 채널의 수는  $(PW/BW)-1$  개가 된다. 그러므로  $k$ 개의 인기 비디오를 위해 예약할 패칭 채널의 용량은 식 3과 같으며, 정규 채널과 패칭 채널을 합한 총 예약 채널의 용량은 식 4와 같다.

$$k \times \left(\frac{PW}{BW} - 1\right) \quad (3)$$

$$k \times \left(\frac{L}{PW} + \frac{PW}{BW} - 1\right) \quad (4)$$

제안하는 채널 예약 배치-패칭을 구현하기 위해 비디오 서버와 비디오를 요청하는 사용자 스테이션에서 실행하는 알고리즘은 패칭[2]과 유사하다. 서버의 채널은 정규 채널이나 패칭 채널 역할을 할 수 있다. 사용자 스테이션은 정규 스트림과 패칭 스트림을 다운로드받기 위해 각각 버퍼를 가진 다운로더  $L_r$ 과  $L_p$ 를 가지며, 버퍼에 저장된 비디오 프레임을 조정하여 스크린에 상영하는 기능을 가진 *VideoPlayer*를 가진다. 사용자 스테이션에서는 비디오 서버에게 사용자와 요청 비디오의 식별자로 구성된 요청 토큰 ( $VideoId, UserIdList$ )를 전송한다. 비디오 서버는 도착한 요청 토큰의  $VideoId$ 가 비인기 비디오라면 비인기 비디오 대기 큐에 저장하며, 인기 비디오라면 배칭을 위해 배칭 큐에 저장하였다가 배칭 윈도우가 끝나면 큐 안에 있는 동일 비디오  $VideoId$ 를 요청한 모든 사용자를  $UserIdList$ 에 포함한 요청 토큰 ( $VideoId, UserIdList$ )를 인기 비디오 대기 큐에 저장한다. (비)인기 대기 큐에 요청이 존재하고 적어도 하나 이상의 사용가능한 (비)예약 채널이 존재하는 서버는 그림 6의 알고리즘을 실행한다.

서버는 가용채널  $FreeChannel$ 이 디스패치되었을 때, 이 채널이 예약된 채널이라면 인기 비디오 대기 큐  $HWQ$ 에서 비예약 채널이라면 비인기 비디오 대기 큐  $WQ$ 에서 스케줄링 정책(예로  $FCFS$ )에 따라 다음에 서비스할 요청 토큰을 선택하여 서비스할 비디오  $v$ 를 선택한다. 이 때 서버는 현재 시점  $t$ 와 가장 최근에 채널  $LatestRegular$ 로 진행중인 비디오  $v$ 에 대한 정규 멀티캐스트의 시작 시점  $t_r$  간의 차이 시간  $(t-t_r)$ 이 패칭 윈도우  $PW$ 보다 작은지를 검토한다. 만

*Algorithm: Video Server Main Routine*

1. Dispatch a free channel  $FreeChannel$ ;
2. If the dispatched channel  $FreeChannel$  is a reserved channel, then  
 Select the request token ( $VideoId=v, UserIdList=vClients$ ) from  $HWQ$  according to  $FCFS$  policy;  
 else  
 Select the request token ( $VideoId=v, UserIdList=vClients$ ) from  $WQ$  according to  $FCFS$  policy;
3. Initialize the service token as ( $PID=null, RID=null$ );
4. If there is no regular multicast of video  $v$  in progress, then  
 Call  $MulticastPortion(FreeChannel, null)$ ;  
 else  
 set  $PID=FreeChannel$  and  $RID=LatestRegular$ , where  $LatestRegular$  is the latest regular channel for video  $v$ ;  
 Call  $MulticastPortion(FreeChannel, LatestRegular)$ ;
6. Because all requests for the same video  $v$ , which are already in the waiting queue, can be served by multicasting on the same channel, if the dispatched channel  $FreeChannel$  is a reserved channel, then  
 do the following for each request token ( $v, vClients$ ) in  $HWQ$ ;  
 else  
 do the following for each request token ( $v, vClients$ ) in  $WQ$ ;  
 If  $PID$  is null, then  
 append  $vClients$  to the client list of the channel  $RID$ ;  
 else  
 append  $vClients$  to the client lists of both channels  $PID$  and  $RID$ ;  
 Send the service token ( $PID, RID$ ) to every clients in  $vClients$  ;  
 Delete the request token from the corresponding waiting queue;
7. Activate the multicast on  $FreeChannel$ ;

그림 6. 비디오 서버의 알고리즘

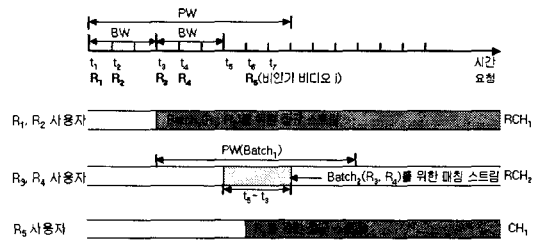
일 패칭 윈도우보다 크다면 새로운 정규 멀티캐스트를 수행해야하므로 요청 토큰의  $UserIdList$ 에 포함된 사용자들에게 패칭 채널은 사용되지 않으며 현재 가용채널  $FreeChannel$ 이 정규 채널로 사용되어 서비스됨을 알리기 위해 서비스 토큰 ( $PID=null, RID=FreeChannel$ )을 전송한다. 그러나 차이 시간이 패칭 윈도우보다 작다면 즉  $(t-t_r) < PW$  라면 패칭 멀티캐스트를 수행해야 하므로 요청 토큰의  $UserIdList$ 에 포함된 사용자들에게 정규 채널이

*Algorithm: MulticastPortion(FreeCh, RegularCh)*  
 $t$  : current time  
 $t_r$  : start time of the latest regular multicast on regular channel *RegularCh* for video  $v$   
 $v$  : the video currently multicast on channel *RegularCh*  
*PW* : size of patching window(i.e. size of the client buffer in playback time unit)  
*D* : portion of video data which should be multicast on channel *FreeCh*(i.e. workload for channel *FreeCh*)  
 $l$  : playback duration of video  $v$ (i.e. video length)  
 $v[t_d]$  : video data of video  $v$  from start to  $t_d$  time unit  
 If  $((RegularCh = null) \text{ or } (t - t_r \geq PW))$ , then  
      $D = v[l]$ ;  
     Modify the service token as  $(PID=null, RID= FreeCh)$  so that the video server can do a new regular multicast of video  $v$  on channel *FreeCh* ;  
 else // in the case of  $(t - t_r < PW)$   
      $D = v[t - t_r]$  ;

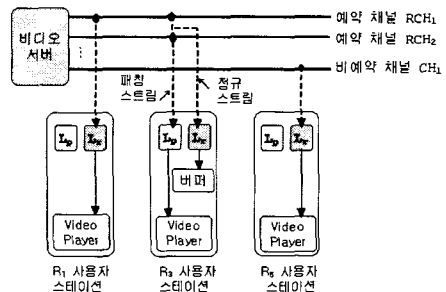
그림 7. 채널 *FreeCh*로 전송할 비디오 데이터 결정 알고리즘

*LatestRegular*이며 패칭 스트림이 전송될 패칭 채널이 *FreeChannel*임을 알리기 위해 서비스 토큰  $(PID=FreeChannel, RID=LatestRegular)$ 를 전송한다. 그리고 정규 채널 *RID*로 진행중인 정규 멀티캐스트의 사용자 목록에 *UserIdList* 안의 사용자들을 추가함으로써 정규 멀티캐스트를 동적으로 확장시킨다. 서버는 서비스 토큰을 전송한 후 채널 *FreeChannel* 상으로 새로운 멀티캐스트를 활성화시킨다.

그림 8(a)는 채널 예약 배치-패칭에서 비디오 서버에 도착한 사용자의 요청을 서비스하기 위해 사용된 서버의 채널과 전송되는 비디오 스트림을 보여준다. *PW*는 패칭 윈도우의 크기, *BW*는 배칭 윈도우의 크기, 채널 *RCH1*, *RCH2*는 인기 비디오를 위해 예약된 채널, 채널 *CH1*은 비예약 채널,  $R_1, R_2, R_3, R_4$ 는 차례대로 시각  $t_1, t_2, t_3, t_4$ 에 도착한 인기 비디오  $i$ 에 대한 요청,  $R_5$ 는  $t_5$ 에 도착한 비인기 비디오  $j$ 에 대한 요청을 나타낸다.  $t_1$  시점부터 배칭 윈도우 *BW* 동안 모아진 인기 비디오  $i$ 에 대한 배치 *Batch1*은  $t_3$  시점부터 서비스되며 예약된 정규 채널 *RCH1*으로 비디오  $i$ 의 데이터 전체가 전송되고 있다. 새로운 배칭 윈도우 동안 모아진 *Batch2*는 *Batch1*을 위한 정규 멀티캐스트의 패칭 윈도우 안에 있으므로 예약된 패칭 채널 *RCH2*로  $t_5-t_3$  만큼의 비디오  $i$ 의 앞부분을 전송한다.  $t_5$ 에 도착한 요청은 비인기 비디오  $j$ 를 요청하



(a) 비디오 요청을 서비스하기 위한 비디오 스트림



(b)  $t_7$  시점에서 사용자 스테이션의 상태

그림 8. 채널 예약 배치-패칭 방법의 예

였으므로 곧바로 사용 가능한 비예약 채널 *CH1*으로 비디오  $j$ 의 전체 데이터를 전송하여 서비스한다. 그림 8(b)는  $t_7$  시점에서 비디오를 요청한 사용자 스테이션의 상태를 보여준다. 요청  $R_1$ 과  $R_2$  각각의 사용자 스테이션에서는  $L_r$ 이 예약된 정규 채널로부터 다운로드받은 비디오  $i$ 의 정규 스트림을 *VideoPlayer*가 상영하고 있다. 요청  $R_3$ 와  $R_4$  각각의 사용자 스테이션에서는  $L_p$ 가 예약된 패칭 채널로부터 패칭 스트림을 다운로드받아서 이를 *VideoPlayer*가 상영하고 있으며 이와 동시에 요청  $R_1$ 과  $R_2$ 를 위해 예약된 정규 채널 *RCH1*으로 전송되는 정규 스트림을  $L_r$ 이 미리 버퍼에 저장하고 있다. 마지막으로 요청  $R_5$ 의 사용자 스테이션에서는 비예약 채널로부터 다운로드 받은 비디오  $j$ 의 정규 스트림을 *VideoPlayer*가 상영하고 있다.

### 5. 시뮬레이션 및 성능 평가

본 논문에서 제안한 채널 예약 배치-패칭 방법의 성능을 시뮬레이션을 통하여 패칭과 단순 배치-패칭에 대해 비교 평가한다. 고려된 성능 평가 항목은 비디오 요청 도착율에 따른 사용자의 이탈율, 평균 서비스 지연시간, 공평성, 전송된 비디오 프레임 절감

올이다. 시뮬레이션은 IBM PC-Pentium IV에서 Visual C++로 작성한 프로그램을 사용하여 수행되었으며 사용된 파라미터는 표 1과 같다. 비디오 관람 패턴에 대한 통계에 의하면 사용자들이 요청하는 비디오는 특정 비디오에 편중되며 이런 지역성은 Zipf 법칙을 사용하여 표현할 수 있기 때문에 비디오  $i$ 의 요청 확률  $p_i$ 는 식 5와 같이 Zipf 법칙을 따른다고 가정한다[8,9].

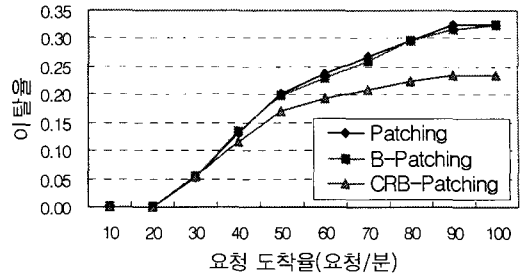
$$p_i = \frac{1}{i^z \sum_{j=1}^N \frac{1}{j^z}} \quad (5)$$

여기서  $N$ 은 전체 비디오의 수이며  $z$ 는 스큐 인자 (skew factor)로서 VOD 응용에서 일반적으로 나타나는 0.7을 사용하였다[3]. 비디오의 길이는 일반적인 영화 길이인 90분으로 설정하였으며 비디오를 요청한 사용자가 서비스를 기다리다가 이탈하는 시간은 관람자들의 일반적인 행동을 반영하여 1분~5분으로 랜덤하게 설정하였다. 사용자는 되감기, 정지, 빨리 감기 등의 요청을 하지 않고 단순히 비디오의 시작부터 끝까지 순차적으로 본다고 가정한다. 시뮬레이션은 360분 동안의 사용자 요청에 대해서 수행되었으며 성능 평가는 시뮬레이션 중단으로 서비스 되지 못한 사용자들을 제외시키기 위해 시작부터 350분 사이에 도착한 요청들에 대해 각 방법의 성능을 평가하였다.

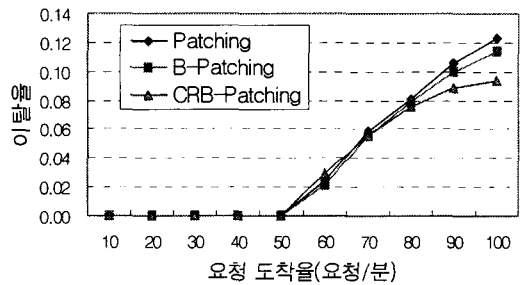
그림 9는 각 방법의 비디오 요청 도착율에 따른 이탈율 결과를 보여준다. Patching은 패칭 방법, B-Patching은 배치-패칭 방법, CRB-Patching은 채널 예약 배치-패칭 방법을 나타낸다. 이탈율은 비디오 서비스를 요청한 후 대기하다가 서비스 받지 않고 이탈한 사용자의 요청수를 서버에 도착한 전체 요청수로 나눈 값으로 비디오 서버의 처리력을 나타낸다. 패칭과 단순 배치-패칭은 이탈율이 거의 같지

표 1. 시뮬레이션 파라미터

파라미터	값
비디오 개수, 길이	100개, 90분
서버 네트워크 채널	900, 1200 채널
요청 도착율/분	10~100 (포아송)
사용자 이탈시간	1분~5분 (랜덤)
시뮬레이션 기간	360분
패칭 윈도우, 배치 윈도우	5분, 1분



(a) 비디오 서버 채널 수 = 900



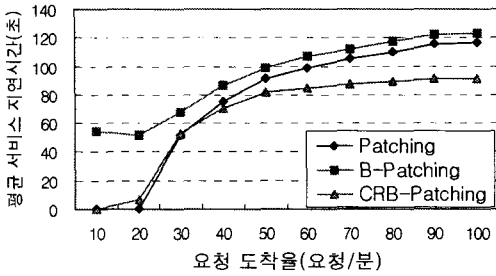
(b) 비디오 서버 채널 수 = 1200

그림 9. 요청 도착율에 따른 이탈율

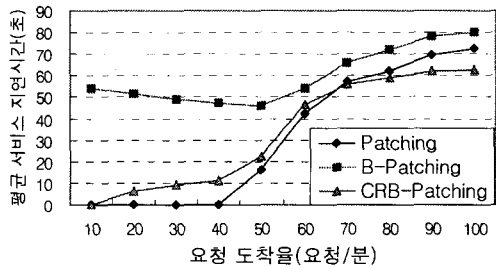
만 요청 도착율이 높아질수록 단순 배치-패칭의 이탈율이 낮으며, 제안한 채널 예약 배치-패칭은 모든 경우에 이들보다 낮은 이탈율을 보인다. 특히 도착율이 증가할수록 도착율이 이탈율에 미치는 영향이 적어짐을 볼 수 있다. 이것은 요청 도착율이 증가할수록 배치 윈도우 내에 존재하는 동일 비디오 요청수가 증가하여 하나의 채널로 서비스할 수 있는 요청수가 증가함으로써 상대적인 이탈자 수가 감소하기 때문이다. 그리고 도착율의 증가로 서버의 부하가 높아졌을 때 채널 일부를 미리 예약하지 않은 단순 배치-패칭과 패칭은 인기 비디오와 비인기 비디오 요청 모두가 블록화될 수 있는 반면에 채널 예약 배치-패칭은 인기 비디오를 위하여 예약된 채널로 인해 많이 도착하는 인기 비디오 요청들이 블록화되지 않고 서비스됨으로써 이탈하지 않기 때문이다.

그림 10은 각 방법의 비디오 요청 도착율에 따른 평균 서비스 지연시간을 보여준다. 평균 서비스 지연시간은 비디오 서비스를 요청한 후 대기하다가 이탈한 사용자의 대기시간도 포함한 모든 사용자 요청에 대한 총 대기시간을 총 요청수로 나눈 것이다. 단순 배치-패칭은 그림 9에서 패칭보다 서비스한 사용자의 수가 많았지만 요청 도착율이 낮아질수록 평균 서비스 지연시간이 패칭보다 훨씬 더 길어짐을 볼 수 있



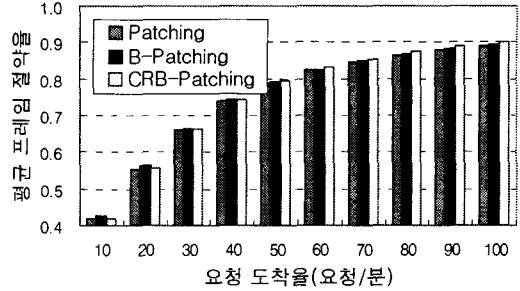


(a) 비디오 서버 채널 수 = 900

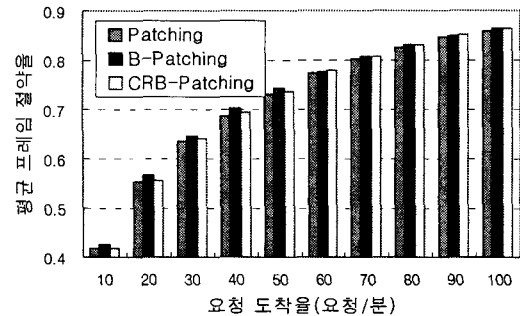


(b) 비디오 서버 채널 수 = 1200

그림 10. 평균 서비스 지연시간



(a) 비디오 서버 채널 수 = 900



(b) 비디오 서버 채널 수 = 1200

그림 11. 요청 도착율에 따른 프레임 절감율

다. 이는 비디오 서버의 부하가 낮은 경우에는 도착한 요청을 즉시 서비스할 수 있음에도 불구하고 단순 배치-패칭이 배칭 윈도우 동안 요청을 모아두었다가 서비스하기 때문이다. 제한한 방법은 요청 도착율이 낮을 때는 패칭보다 평균 서비스 지연시간이 조금 길게 나타났지만 도착율이 높을 때는 패칭보다 더 짧아짐을 볼 수 있다. 이것은 요청 도착율이 증가할수록 배칭 윈도우 내에 존재하는 동일 비디오 요청수가 증가하며 많이 도착하는 인기 비디오에 대한 요청을 예약된 채널을 통해 빠르게 서비스할 수 있기 때문이다.

그림 11은 각 방법에 대해 비디오 요청을 위해 전송한 비디오 프레임의 평균 절감율을 보여준다. 프레임 절감율은 하나의 비디오를 서비스하기 위해 정상적으로 전송해야 하는 전체 프레임 수를 얼마나 많이 감소시켰는가를 나타내는데 사용된다. 프레임 절감율이 높을수록 서버의 네트워크 부하가 감소된다고 할 수 있다. 평균 프레임 절감율은 식 6과 같이 계산된다.

$$1 - \frac{\sum_{i=1}^s f_{R_i}}{s \times f} \quad (6)$$

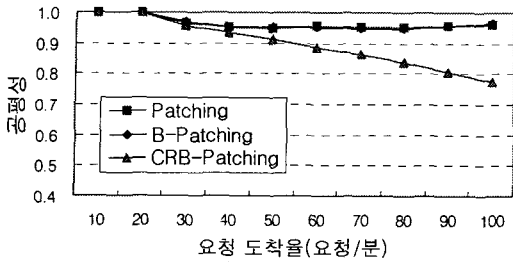
여기서  $s$ 는 서비스된 총 사용자 요청 수,  $f$ 는 한

개의 비디오를 위해 전송해야 하는 총 프레임 수,  $f_{R_i}$ 는 각 방법에 의해 사용자 요청  $R_i$ 를 위해 실제로 전송된 비디오 프레임 수를 나타낸다. 모든 방법들이 요청 도착율이 증가할수록 프레임 절감율이 증가됨을 볼 수 있다. 이것은 도착율이 증가할수록 배칭 윈도우와 배칭 윈도우 안에 존재하는 동일 비디오 요청수가 증가됨으로써, 배칭과 패칭으로 인한 비디오 스트림의 공유가 증가하기 때문이다. 그림에서 볼 수 있듯이 요청 도착율이 낮을 때는 단순 배치-패칭이 가장 우수하나 요청 도착율이 높아질수록 제안하는 방법의 프레임 절감율이 가장 높음을 알 수 있다.

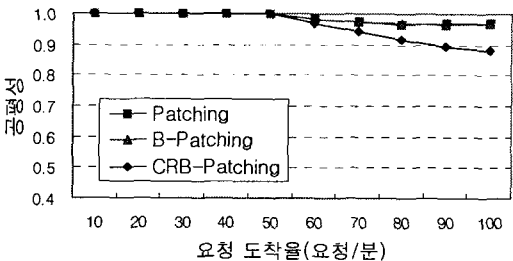
그림 12는 각 방법마다 사용자가 요청한 비디오에 대하여 서비스의 공평성(fairness)을 계산한 결과를 보여준다. 모든 비디오 요청에 대한 평균 이탈율이  $d_i$ , 각 비디오  $i$ 에 대한 요청의 이탈율이  $d_i$ , 전체 비디오 수는  $N$ 이라고 할 때, 공평성은 식 7과 같이 계산된다[3].

$$1 - \sqrt{\frac{\sum_{i=1}^N (d_i - d)^2}{N-1}} \quad (7)$$

그림 12에서 볼 수 있듯이 배치-패칭과 패칭은 공



(a) 비디오 서버 채널 수 = 900



(b) 비디오 서버 채널 수 = 1200

그림 12. 요청 도착율에 따른 공평성

평성이 거의 비슷하며 채널 예약 배치-패칭 방법보다 우수함을 알 수 있다. 이것은 요청 도착율이 높아져서 비디오 서버가 모든 요청을 서비스하지 못하는 경우에도 제한한 채널 예약 배치-패칭 방법에서는 인기 비디오 요청들이 서버의 예약된 채널을 통해 서비스됨으로써 많은 인기 비디오 요청들의 이탈율이 패칭에 비해 최소화되었고, 비인기 비디오 요청의 이탈율은 패칭보다 증가됨으로써 비디오간의 이탈율의 차이가 커졌기 때문이다. VOD 서비스에 있어서 서버의 과부하로 인해 많은 요청들이 이탈될 수밖에 없는 상황에서 훨씬 더 많은 요청들을 서비스하며 평균 서비스 지연시간을 감소시키는 것이 공평성보다 더 중요하다고 생각한다면 패칭에 비해 비디오 전체에 대한 공평성이 저하된 것은 큰 문제가 되지 않을 것이다.

지금까지의 시뮬레이션 결과를 비교해본 결과 제안한 채널 예약 배치-패칭 방법은 비디오 서버의 부하가 높아질수록 이탈율, 평균 서비스 지연시간, 전송된 프레임 절감율이 다른 방법보다 우수하므로 서버의 부하가 높을 때 좋은 방법이라고 할 수 있다. 그림 10에서 비디오 요청 도착율이 낮을 때는 패칭의 평균 서비스 지연시간이 가장 우수하였는데 이것은 배치-패칭이나 채널 예약 배치-패칭이 사용자의 비디오 요청이 도착하는 즉시 서비스하는 것이 아니라

배칭 윈도우 기간동안 모으기 때문이다. 그러므로 비디오 서버에 가용채널이 충분한 경우에는 도착한 요청을 배칭 윈도우 동안 모아두는 것이 아니라 도착하는 즉시 서비스를 수행하여 서비스 지연시간을 제거하는 방법을 고려할 수 있다. 도착한 요청을 배칭 윈도우 동안 모아두지 않기 위해서는 단순히 비디오 서버의 배칭 윈도우 크기를 0으로 설정하기만 하면 될 것이다. 즉 주기적으로 비디오 요청 도착율과 평균 서비스 지연시간을 검토하여 배칭 윈도우의 크기를 동적으로 증감시킴으로써 VOD 시스템의 성능을 향상시킬 수 있을 것으로 생각된다.

## 6. 결 론

비디오를 요청하는 여러 사용자들에게 보다 빠른 VOD 서비스를 제공하기 위해서는 비디오 서버의 네트워크 대역폭에 의해 결정되는 제한된 자원인 채널을 효율적으로 사용할 수 있는 방법이 필요하다. 이를 위해 동일한 비디오를 요청한 여러 사용자들을 단일 스트림으로 서비스함으로써 서버의 네트워크 대역폭 요구를 절감시키려는 여러 가지 멀티캐스트 방법이 제안되었다. VOD 서비스를 요청하는 사용자들의 비디오들을 분석해보면 모든 비디오가 균등하게 요청되는 것이 아니라 특정 인기 비디오가 집중적으로 요청되는 특징을 가지고 있다. 본 논문에서는 이러한 사용자의 요청 특성에 근거하여 배칭과 패칭을 조합한 배치-패칭의 성능을 개선시키기 위해 인기 비디오 요청만 배칭 윈도우 동안 모아두었다가 패칭 방법을 적용하며 비인기 비디오 요청은 도착하는 즉시 패칭 방법을 적용하며, 도착율이 높은 인기 비디오 요청의 서비스 보장으로 VOD 서버의 처리력을 증가시킬 수 있도록 비디오 서버의 채널 일부를 예약하는 채널 예약 배치-패칭을 제안하고 성능을 평가하였다.

시뮬레이션 결과 제안한 채널 예약 배치-패칭은 인기 비디오 전용의 비디오 서버 채널 예약으로 공평성이 다른 방법보다 낮았다. 그리고 서버의 부하가 작을 때는 제한한 채널 예약 배치-패칭의 평균 서비스 지연시간이 패칭보다 우수하지 않았으나 부하가 높을 때는 이탈율, 평균 서비스 지연시간, 전송된 프레임 절감율 모두가 가장 우수하여 서버의 부하가 높을 때 좋은 방법임을 알 수 있었다.

향후 배칭 윈도우의 크기를 요청 도착율과 평균

서비스 지연 시간에 따라 동적으로 조절하여 부하가 작을 때도 패칭과 같이 평균 서비스 지연시간이 개선될 수 있는 방법에 대해 연구할 계획이다.

참 고 문 헌

[1] Jani Huoponen and Thorsten Wagner, "Video on Demand A Survey," telecommunication Networks Project, 1, [http://fiddle.visc.vt.edu/courses/ee4984/Projects1996/huoponen\\_wagner/huoponen\\_wagner.html](http://fiddle.visc.vt.edu/courses/ee4984/Projects1996/huoponen_wagner/huoponen_wagner.html), 1996.

[2] Y. Cai, K. Hua, and K. Vu, "Optimizing Patching Performance," *In Proc. SPIE/ACM Conference on Multimedia Computing and Networking*, pp. 204-215, 1999.

[3] K. Hua, Y. Cai, and S. Sheu, "Patching: A Multicast Technique for True Video-on-Demand Services," *In Proc. ACM Multimedia*, pp. 191-200, 1998.

[4] P. White, J. Crowcroft, "Optimized Batch Patching with Classes of Service," *In ACM SIGCOMM Computer Communication Review*, pp. 21-28, October 2000.

[5] Dan, A., Sitaram, D., and Shahabuddin, P., "Scheduling Polices for an On-Demand Video Server with Batching," *In Proc. of the 2nd ACM Multimedia Conference*, pp. 25-32, 1994.

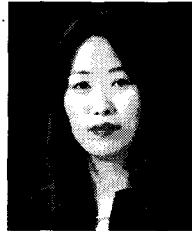
[6] C. C. Aggarwal, J. L. Wolf, and P. S. Yu, "On Optimal Piggyback Merging Policies for Video-On-Demand Systems," *In Proc. 1996 ACM SIGMETRICS Conf. on Measurement and Modeling of Computer Systems*, pp. 200-209, 1996.

[7] A. Yang Guo, Hyoungwon Suh, Jim Kurose, and Don Towsley, "P2Cast: Peer-to-peer Patching Scheme VoD Service," *In Proceedings of the 12th International Conference of World Wide Web*, pp. 301-309, 2003.

[8] A. Chervenak, D. Patterson, and R. Katz, "Choosing the Best Storage System for Video Service," *In Proc. of ACM Multimedia 95*, pp.

109-119, Aug. 1995.

[9] Christer Samueleson, "Relating Turing's Formula and Zipf's Law," *In Proc. of the Fourth Workshop on Very Large Corpora*, 1996.



하 숙 정

1988년 계명대학교 전자계산학과(학사)  
 1990년 중앙대학교 전자계산학과(석사)  
 1998년 대구가톨릭대학교 전산통계학과(박사)  
 2001년~현재 경북대학교 전자전기컴퓨터학부 초빙교수

관심분야: 모바일 컴퓨팅, 멀티미디어 시스템 등



이 경 속

1990년 대구가톨릭대학교 수학교육학과(학사)  
 1993년 대구가톨릭대학교 전산통계학과(석사)  
 2000년 대구가톨릭대학교 전산통계학과(박사)  
 2002년~현재 경북대학교 전자전기컴퓨터학부 초빙교수

관심분야: 모바일 컴퓨팅, 멀티미디어 시스템 등



배 인 한

1984년 2월 경남대학교 전자계산학과(학사)  
 1986년 2월 중앙대학교 대학원 전자계산학과(석사)  
 1990년 8월 중앙대학교 대학원 전자계산학과(박사)  
 1996년~1997년 Computer and Information Science, The Ohio State University (Post-doctoral)

2002년~2003년 Department of Computer Science, Old Dominion University (Visiting Professor)  
 1989년~현재 대구가톨릭대학교 컴퓨터정보통신공학부 교수  
 관심분야: 이동 무선망, M-Commerce, 멀티미디어 시스템 등