

True VOD 시스템을 위한 채널 예약 패칭 방법의 설계 및 평가

이 주 영[†] · 하 숙 정^{††} · 배 인 한^{†††}

요 약

비디오 서버가 제공할 수 있는 채널의 개수는 서버의 통신 대역폭에 의해 결정되므로 비디오 서버의 가용 채널 수는 제한되어 있다. 멀티캐스트 데이터를 공유함으로써 비디오 서버에 대한 I/O 요구를 절감시키기 위하여 일괄처리, 피기백킹 그리고 패칭과 같은 여러 방법들이 제안되었다. 특히 패칭은 VOD 시스템을 위한 비용에 있어서 효율적인 것으로 보여지고 있다. 패칭은 일반적인 멀티캐스트 기법과 달리 새로운 요청을 진행중인 멀티캐스트에 합병할 수 있게 하는 동적 멀티캐스트 방법이다. 더욱이, 새로운 요청은 다음 멀티캐스트를 기다리지 않고 즉시 서비스될 수 있으므로 true VOD를 이룰 수 있다. 본 논문에서, 우리는 고정 채널 예약 패칭과 가변 채널 예약 패칭이라는 두 가지 채널 예약 패칭 알고리즘을 제안한다. 인기 비디오에 대한 요청들을 즉시 스케줄하기 위하여, 제안하는 알고리즘들은 고정된 개수의 인기 비디오 또는 서버의 부하에 따라 동적으로 결정되는 가변 개수의 인기 비디오를 위하여 비디오 서버의 채널들을 예약한다. 제안하는 알고리즘들의 성능은 시뮬레이션을 통하여 평가되었고, 단순 패칭의 성능과 비교하였다. 사용된 성능 척도는 비디오 서버 부하에 따른 평균 이탈율, 평균 대기 시간, 서비스 공평성 그리고 버퍼량이다. 시뮬레이션 결과는 우리가 제안하는 채널 예약 패칭 알고리즘들이 단순 패칭 보다 성능이 더 나음을 보였다.

Design and Evaluation of a Channel Reservation Patching Method for True VOD Systems

Joo-Yung Lee[†] · Sook-Jeong Ha^{††} · In-Han Bae^{†††}

ABSTRACT

The number of channels available to a video server is limited since the number of channels a video server can support is determined by its communication bandwidth. Several approaches such as batching, piggybacking and patching have been proposed to reduce I/O demand on the video server by sharing multicast data. Patching has been shown to be efficient in the matter of the cost for VOD systems. Unlike conventional multicast techniques, patching is a dynamic multicast scheme which enables a new request to join an ongoing multicast. In addition, true VOD can be achieved since a new request can be served immediately without having to wait for the next multicast. In this paper, we propose two types of channel reservation patching algorithm : a fixed channel reservation patching and a variable channel reservation patching. To immediately schedule the requests for popular videos, these algorithms reserve the channels of video server for the fixed number of popular videos or for the variable number of popular videos which is determined dynamically according to the load of video server. The performance of the proposed algorithms is evaluated through simulations, and compared with that of simple patching. Our performance measures are average defection rate, average latency, service fairness and the amount of buffered data according to video server loads. Simulation results show that the proposed channel reservation patching algorithms provide better performance compared to simple patching algorithm.

키워드 : True VOD, 멀티캐스트(Multicast), 비디오 데이터 공유(Video data sharing), 서버의 통신 대역폭(Communication bandwidth), 패칭(Patching), 인기 비디오, 일괄 처리(Batching), 고정 채널 용량 예약(Channel reservation), 가변 채널 용량 예약

1. 서 론

멀티미디어의 대표적인 형태인 VOD(Video-on-Demand) 서비스는 향후 가장 각광받게 될 서비스중 하나이다[1]. VOD

서비스는 비디오 서버에 저장된 비디오를 사용자들이 직접 선택해서 원하는 시간에 볼 수 있는 사용자의 주문에 의한 영상 서비스로서 방송국에서 방송하는 TV 프로그램을 일방적으로 수신하는 기존의 방송 방식에서 벗어난 형태이다. 본 논문은 오디오나 비디오 파일과 같은 대량의 데이터 파일들을 사용자가 요청한 즉시 제공하는 실시간 주문형 전송 서비스를 위한 방법에 대해 연구한다. MPEG(Moving Pic-

[†] 정 회 원 : 엠비즈 네트워크

^{††} 정 회 원 : 경북대학교 전자전기컴퓨터학부 교수(BK21)

^{†††} 정 회 원 : 대구가톨릭대학교 컴퓨터정보통신공학부 교수

논문접수 : 2002년 7월 3일, 심사완료 : 2002년 8월 31일

ture Experts Group)-1과 MPEG-2 같은 동영상 압축 기술의 발달과, 고속의 전송 기술인 ATM(Asynchronous Transfer Mode) 기술과 RAID(Redundant Arrays of Inexpensive Disks)와 같은 디스크 배열을 이용한 고속 대용량 저장 서버의 구현 기술 등의 발달로 VOD 서비스가 가능해지고 있다. 그러나 비디오 스트림은 압축이 되더라도 높은 대역폭을 요구하며(MPEG2의 경우 4~6Mbps), 재생시간이 일반적으로 길며, 엄격한 응답 시간을 요구하며 연속적으로 전송되어야 하므로 전송을 위한 I/O 스트림 예약과 채널 할당이 보장되어야 한다. 그러므로 대용량의 비디오 서비스가 가능하며 새로운 사용자를 위한 처리비용이 최소가 되도록 서버의 네트워크 대역폭을 절약하는 효과적인 기법이 요구된다. 최근의 멀티캐스트와 방송은 멀티미디어 스트림을 사용자에게 즉시 서비스하기 위한 연구의 큰 흐름이다. 멀티미디어 스트림의 전송에 드는 대부분의 비용이 네트워크 비용이므로 두 기법 모두 서버의 네트워크 대역폭을 절약하기 위하여 스트림의 공유를 기초로 하고 있다. 최근 두 방법과 관련된 새로운 연구들이 활발히 연구되고 있다.

본 논문은 VOD 서비스에서 뉴스, 상품 광고, 의학, 텔레비전 쇼, 필수적인 원격 교육 콘텐츠 등과 같은 인기 비디오의 공유에 초점을 둔다. 사용자들의 데이터 접근 형태를 살펴보면 모든 비디오를 균등하게 접근하는 것이 아니라 접근 빈도수에 차이가 있는 것을 알 수 있다. 즉, 인기가 있는 비디오의 접근 빈도수는 많고 인기가 없는 비디오에 대한 접근 빈도수는 적다. 이러한 비디오에 대한 접근 빈도수는 일반적으로 Zipf 분포[2]를 따른다. 이와 같은 특정 비디오의 인기도 편중으로 짧은 순간에 인기 비디오에 대한 사용자들의 새로운 요청이 많이 도착하게 된다. 본 논문은 이러한 비동기적인 사용자들의 요청에 대해 비디오 스트림을 멀티캐스트하면서 서버의 네트워크 대역폭을 감소시키는 패칭(patching) 정책[3-6]을 기반으로, 인기 비디오에 대한 요청들이 즉시 서비스되도록 서버의 채널 일부를 인기 비디오 서비스를 위한 채널로 미리 예약해 두는 채널 예약 패칭 방법을 제안한다. 제안하는 방법은 서버의 네트워크 대역폭 요구를 절감시키면서 사용자에게 대기 시간 없이 서비스를 할 수 있을 뿐만 아니라 사용자의 버퍼를 기초로 VCR 기능을 구현 할 수 있기 때문에 true VOD 서비스가 가능하다.

본 논문의 구성은 다음과 같다. 2장에서는 주문형 비디오 시스템에 대해 간략히 기술하고, 3장에서는 패칭과 캐칭에 관하여 소개하고, 4장에서는 제안하는 채널 예약 패칭 방법을 설계하고, 5장에서는 시뮬레이션을 통해 제안하는 방법의 성능을 평가한다. 그리고 마지막으로 6장에서 결론을 맺는다.

2. 주문형 비디오 시스템

전자 기술의 발달로 케이블 TV와 VCR(Video Cassette

Recorder)의 보급이 늘어가고 있지만 소비자의 요구를 충분히 만족시키지 못하고 있다. 이는 사용자에게 보고 싶은 프로그램이나 시간을 선택할 수 있는 권한이 제공되지 않기 때문이다. 이에 반해 VOD 서비스는 컴퓨터의 처리 속도 증가와 광대역 네트워크의 실현, 멀티미디어 데이터 압축 기술, 대용량 데이터 저장 기술, 멀티미디어 데이터 처리를 위한 운영체제 및 응용 프로그램 기술 발달 등에 힘입어 비디오 데이터를 컴퓨터에 저장하여 원거리의 사용자가 원하는 시간에 비디오 스트림을 제공하도록 제안된 서비스이다. 이러한 VOD 서비스는 사용자가 비디오와 시간을 선택하여 서비스받을 수 있게 하므로 대표적인 멀티미디어 정보 서비스로 자리 잡을 것으로 전망되고 있다[7, 8].

VOD 서비스는 주문형 비디오, 홈쇼핑, 화상회의, 원격 교육 및 자료, 전자 도서관, 주문형 신문과 같은 응용 분야에 사용될 수 있다. VOD 서비스를 상용화하기 위해서는 많은 비디오 데이터를 저장할 대용량의 저장 능력과, 전송 능력을 최대한 활용하여 많은 사용자를 서비스하고, 사용자의 이탈율과 평균 대기 시간을 최소화하고, 서비스되는 비디오의 공정성을 유지하는 방법이 필요하다. 특히 true VOD 서비스는 사용자에게 빨리 감기, 되감기, 일시정지 등 VCR의 모든 기능을 제공하면서도 실시간으로 서비스를 제공하는 VOD 서비스를 말한다. 이러한 true VOD 서비스는 가상 VCR과 같은 강력한 서비스를 제공하지만 이를 실현하기 위해서는 많은 비용이 든다. 현재 VOD의 표준화를 앞당기기 위해 DAVIC(Digital Audio-Visual Council)을 중심으로 VOD 서비스의 표준화 작업이 진행되고 있으며, 여러 사용

<표 1> 상호 대화성에 따른 VOD의 분류

종 류	특 징	장 점	단 점
No-VOD	공중과 TV와 같은 공급자의 일방적인 서비스 형태	여러 사용자에게 서비스 제공	단방향성
PPV (Play-Per-View)	일정액의 돈을 지불하고 원하는 영상을 받아 보는 형태		양방향성 불가능
SVOD (Staggered VOD)	수분 정도의 짧은 시간 간격을 두고 여러 채널에 동일한 프로그램을 시간 간격에 따라 송신함으로써 시청자가 채널간을 이동하면서 필요한 부분을 볼 수 있게 한 시스템	시청자는 VCR의 일부 기능을 채널 이동으로 실현 할 수 있다. 단방향이면서도 양방향의 효과를 낼 수 있음	채널의 소요가 많음
NVOD (Near VOD)	시차를 30분 정도로 해서 동일한 프로그램을 송신함으로써 시간 간격을 통해 영상 데이터를 시청할 수 있음	채널의 수요가 크지 않음(예: 90분 영화를 30분 간격으로 보낼 때 3 채널 소요)	양방향성 미흡
TVOD (True VOD)	가입자의 개별 요구에 따라 영상 데이터에 대해 완전히 대화적인 가상 VCR 기능을 제공하는 시스템	완전 양방향 가능	기술적으로 복잡하고 어렵다.

자에게 동시에 안정적인 비디오 스트림을 제공하기 위한 연구가 활발히 진행되고 있다[9, 10]. <표 1>은 다양한 VOD 서비스의 특징과 장단점을 보여준다.

현재 PC의 메인 메모리나 디스크의 바이트 당 저장 비용이 극도로 감소하였고, 액세스 시간도 감소하고 있으며, 시스템 버스 속도는 수 GHz, 그리고 디스크 공간은 수십 GB에 이르고 있다. 이러한 추세에 의해, 사용자 스테이션은 장시간의 고품질 비디오를 버퍼하기에 충분한 대용량 저장 공간과 동시에 여러 전송 채널로부터 데이터를 받아들이기에 충분한 I/O와 디스크 대역폭을 가지게 되었다. 그러나 일반적으로 비디오 서버는 네트워크 대역폭이 제한되어 있으므로 true VOD를 구현하기 위해 서버의 네트워크 부하를 감소시키기 위한 여러 가지 방법들이 연구되었다[3, 5, 11-15].

3. 관련 연구

True VOD 서비스를 위해서는 중요한 자원인 채널이 최대한 많은 사용자들과 공유될 수 있어야 한다. 채널을 공유하기 위한 대표적인 멀티캐스트 기법으로 일괄처리[11]와 피기백킹(piggybacking)[13, 14] 기법이 있다.

일괄처리는 짧은 시간 동안의 요청을 모아 두었다가 동일한 비디오에 대한 여러 요청에 대해 하나의 멀티캐스트 스트림으로 서비스하는 기법으로 채널을 많이 공유할 수 있지만 사용자의 대기 시간이 증가될 수 있다. 피기백은 동일 비디오에 대한 여러 스트림이 하나로 병합될 때까지 진행 중인 스트림의 디스플레이율을 조정하는 기법이다. 피기백은 일괄처리보다 서비스 대기 시간이 감소되지만, 피기백 율의 변화가 $\pm 5\%$ 이내여서 병합할 수 있는 스트림의 수가 제한적이며 특별한 하드웨어 환경이 필요하다[14]. 이러한 채널의 공유와 대기 시간에 대한 상반관계를 해결하기 위해 멀티캐스트를 이용하는 패칭 방법[3, 5]과 방송을 사용하는 skyscraper[15, 16]가 제안되었다.

본 논문은 멀티캐스트를 통해 사용자가 대기 시간 없이 서비스를 받으며 서버의 대역폭 요구량을 감소시키는 패칭 방법을 기반으로 한다. 패칭은 사용자의 요청에 대해 언제나 비디오 전체를 전송하는 것이 아니라 이미 한 비디오의 전체 스트림이 멀티캐스트되고 있는 채널을 여러 사용자가 공유하도록 하는 것이다. 즉 이미 비디오 전체를 전송 중인 멀티캐스트가 수행되고 있을 때, 이후에 발생한 사용자의 요청에 대해서는 진행 중인 최근의 멀티캐스트의 시작 시간과 현재 사용자 요청에 대한 서비스 시작 시간과의 차이에 해당하는 비디오의 앞부분만을 새로운 채널을 통해 전송한다. 서버가 비디오 전체를 전송하는 채널은 정규 채널, 비디오의 특정 앞부분만을 전송하는 채널은 패칭 채널, 사용자가 정규 채널을 통해 수신하는 비디오 데이터는 정규 스트림, 패칭 채널을 통해 수신하는 비디오의 앞부분 데이터는

패칭 스트림이라고 한다. 현재 서비스될 요청이 이미 서비스 중인 최근의 정규 멀티캐스트를 패치할 수 있는 최대 시간을 패칭 윈도우라 한다. 서버는 패칭 윈도우를 사용하여 새로운 사용자의 요청에 대해 정규 채널을 통해 비디오 전체에 대한 정규 멀티캐스트를 수행할 지 아니면 시간차만큼의 비디오 앞부분만을 패칭 채널을 통해 패칭 멀티캐스트를 수행할 것인가를 결정한다. 현재 서비스할 요청과 최근의 정규 멀티캐스트간의 시간차가 패칭 윈도우 시간을 벗어났다면 항상 새로운 채널을 통해 비디오 전체를 전송한다. 그러나 패칭 윈도우 안에 있다면 정규 스트림을 기준으로 패치된다.

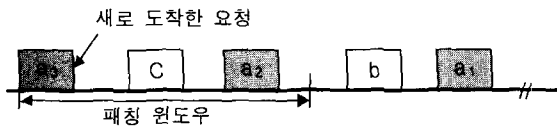
패칭 방법은 일괄처리 방법에 비해 훨씬 성능이 좋다. 예를 들어 일괄처리 방법에서 비디오 A에 대한 배치 윈도우 3분 동안의 요청들을 일괄처리하여 배치 B_i 를 채널 C_i 로 멀티캐스트한다고 하자. 다음의 3분 동안의 비디오 A에 대한 요청의 배치 B_j 는 채널 C_j 로 멀티캐스트된다. 비디오의 길이가 90분이라고 가정한다면 채널 C_i 와 C_j 는 동일 비디오 A에 대한 B_i 와 B_j 를 각각 90분동안 전송할 것이다. 그러나 패칭 방법에서는 C_j 는 동일 비디오에 대한 마지막 정규 멀티캐스트의 시작 시간과 현재 스케줄된 요청의 서비스 시작 시간과의 차이시간으로 정의되는 경사(skew) 동안만 패칭 스트림을 전송하면 된다. 즉 패칭 방법은 채널 C_j 를 비디오 A의 전체 재생시간 동안 점유하는 것이 아니라 경사 부분만을 패치하는동안 사용한다. 새로운 정규 멀티캐스트를 시작해야할지 패칭을 해야할지를 결정하기 위해서 서버는 경사를 계산한다. 이 예에서 일괄처리 기법에서는 채널 C_j 를 90분동안 점유하는데 반해 패칭 윈도우가 3분인 패칭 기법을 사용하면 단지 3분만 채널 C_j 를 점유한다는 것을 알 수 있다.

이와 같이 패칭은 채널의 처리력을 일괄처리보다 훨씬 많이 개선시켰을 뿐만 아니라 사용자의 요청에 대해 즉시 서비스함으로써 대기시간을 제거할 수 있지만, 사용자는 정규 스트림과 패칭 스트림을 동시에 받기 위해서 파일 재생율의 두 배의 대역폭이 필요하고, 패칭 스트림을 버퍼링하기 위해 추가적인 버퍼 공간이 요구된다. 그러나 현재 시스템의 성능이 발달하여 사용자들이 대용량의 디스크 공간을 가지고 있으며 고대역 전송을 받기에 충분한 뿐만 아니라, 사용자의 버퍼를 이용하여 true VOD를 위한 VCR 기능을 구현할 수 있다는 장점이 있다[17-19].

4. 채널 예약 패칭 방법의 설계

본 논문에서는 사용자 스테이션의 버퍼 공간과 채널이 일괄처리에 비해 두배로 필요하지만, 사용자에게 대기 시간 없이 서비스를 제공함으로써 true VOD를 구현하기 위한 채널 예약 패칭 기법을 제안한다. 제안하는 채널 예약 패칭 방

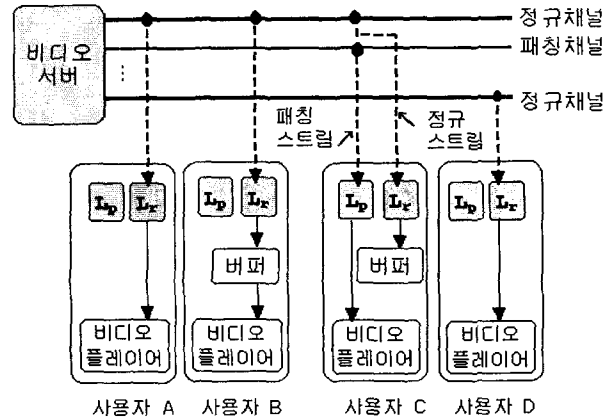
법은 (그림 1)과 같은 시나리오로부터 얻어진다. 비디오 a는 인기 비디오이며 비디오 b와 c는 인기가 없는 비디오이다. 비디오 a에 대한 새로운 요청이 도착했을 때, 이미 사용자 1의 비디오 a에 대한 요청이 서비스되어 a_1 스트림이 전송중이며 다음으로 사용자 2의 비디오 a에 대한 요청이 서비스되어 a_2 스트림이 전송중인 상태라고 하자. 새로운 요청이 스케줄되었을 때 패칭 윈도우 안에 이미 전송중인 정규 스트림 a_2 가 있으므로 이 새 요청은 정규 스트림 a_2 를 공유하도록 패치될 수 있다. 그러나, 비디오 서버에 가용 채널이 없어서 새로운 요청이 스케줄되지 못하고 대기 큐에서 기다려야 한다면 패칭 기회를 놓치게 된다. 요청율이 높은 인기 비디오에 대한 요청들은 자주 발생되므로 항상 패칭 윈도우 안에 전송중인 정규 스트림이 존재하기 쉽다. 그러므로 인기 비디오의 요청이 항상 패치되도록 하기 위해서는 서버의 일부 채널을 인기 비디오만을 위해 사용되도록 예약해두면 될 것이다.



(그림 1) 인기 비디오 a 요청들간의 관계

채널 예약 패칭 방법은 (그림 2)와 같은 구조에서 수행된다. 비디오 서버의 통신 채널은 비디오 전체를 전송하는 정규 멀티캐스트를 위한 정규 채널과, 경사 부분만 전송하는 패칭 멀티캐스트를 위한 패칭 채널로 사용될 수 있다. 사용자는 각 채널로부터 정규 스트림 또는 패칭 스트림을 다운로드 받기 위해 두 개의 다운로드 L_r 와 L_p 를 가지며, 버퍼에 저장된 비디오 프레임들을 조정하여 스크린에 상영시켜주는 기능을 하는 VideoPlayer를 가진다. (그림 2)는 사용자 A가 먼저 비디오 요청을 한 후, 사용자 B, C, D가 차례대로 같은 비디오에 대한 요청을 하였으며, 사용자 A는 사용자 B와 C의 패칭 윈도우 안에 있으나 사용자 D의 패칭 윈도우는 벗어난 경우이다. 사용자 A는 가장 먼저 요청하였으므로 서버는 정규 채널을 통해 비디오 전체를 전송하며, 사용자는 정규 스트림을 전송받아 비디오 플레이어가 이를 스크린에 상영한다. 사용자 C는 사용자 A에 대한 멀티캐스트가 패칭 윈도우 안에 있으므로 서버는 경사만큼의 비디오 스트림을 패칭 채널을 통해 전송하고, 사용자 C의 비디오 플레이어는 수신한 패칭 스트림을 스크린에 상영한다. 이때 사용자 C는 동시에 정규 채널의 정규 스트림을 버퍼에 저장하여 패칭 스트림을 모두 상영한 후 버퍼에 저장된 그 이후의 비디오 스트림을 연속하여 상영하도록 한다. 사용자 C보다 먼저 서비스된 사용자 B에서는 비디오 플레이어가 이미 패칭 스트림을 모두 스크린으로 상영한 후 버퍼에 저장된 정규 스트림을 스크린에 상영하면서 계속해서 정규 스트림을 버퍼에 저장하고 있는 상태이다. 사용자 D의

경우는 사용자 A에 대한 정규 스트림이 패칭 윈도우 밖에 있으므로 서버가 새로운 정규 채널을 할당하여 비디오 전체를 전송하고 있다.



(그림 2) 패칭 시스템

인기 비디오 요청을 위해 서버의 채널을 예약할때 채널을 너무 많이 예약하면 낭비가 발생하고, 너무 적게 예약하면 인기 비디오에 대한 예약 채널이 부족하여 인기 비디오 요청자의 이탈이 많아질 것이다. 본 논문의 채널 예약 패칭 방법에서는 요청 빈도수가 높은 인기 비디오의 개수 k 에 따라 서버의 채널 일부를 예약하는 두 가지 정책을 제안한다. 첫 번째는 고정 채널 예약 방법으로서 비디오 요청 도착율에 관계없이 인기도 순위에 따라 인기 비디오를 결정한다. 여기서 각 비디오의 요청 확률 즉 인기도가 Zipf 법칙을 따른다고 가정하면, 매우 자주 요청되는 비디오는 소수에 불과하고 이런 요청 빈도가 높은 비디오는 이 비디오에 대한 요청 빈도가 어느 정도 줄더라도 순위가 잘 내려가지 않지만 중하위 순위의 비디오에서는 그 비디오 요청 빈도가 조금만 줄어도 순위가 많이 내려가게 되는 'hyperbolic' 그래프 모양의 요청 빈도를 보인다. 따라서 우리는 인기 비디오의 개수 k 를 2로 결정한다. 두 번째는 가변 채널 예약 방법으로서 비디오 요청 도착율에 따라 인기 비디오의 개수를 동적으로 변경시킨다. p_i 가 Zipf 법칙에 의해 전체 비디오 중에서 비디오 i 가 요청될 확률이고 λ 는 비디오 요청 도착율이라고 할 때 비디오 i 에 대한 분당 도착 가능한 요청 수는 $p_i \times \lambda$ 개라고 할 수 있다. 패칭의 이점을 얻기 위해서는 패칭 윈도우 안에 여러 개의 동일 비디오 요청이 존재해야 한다. 일반적으로 패칭 윈도우는 몇 분 정도이므로 가변 채널 예약 방법에서는 분당 요청 도착수가 1개 보다 더 많은 비디오 즉 $p_i \times \lambda > 1.0$ 를 만족하는 비디오들을 인기 비디오로 간주하여 도착율에 따라 k 가 결정되도록 하였다.

제안하는 채널 예약 패칭에서 인기 비디오를 위한 서버의 채널 예약 용량은 다음과 같이 결정할 수 있다. p_i 는 비디오 i 의 요청 확률, L 은 비디오의 길이, PW 는 패칭 윈도우

우의 크기, λ 는 도착율, 그리고 k 는 인기 비디오의 개수라고 하자. 비디오 i 에 대해 정규 멀티캐스트가 시작된 후 패칭 윈도우 만큼의 시간이 지나면 새로운 정규 채널을 통해 정규 멀티 캐스트가 시작되어야 하므로 비디오 길이 L 분 동안 일어날 수 있는 정규 멀티 캐스트의 회수는 (L/PW) 이며 k 개의 인기 비디오를 위해 필요한 정규 채널은 식 (1)에 의해 얻어진다.

$$k \times \frac{L}{PW} \quad (1)$$

패칭 윈도우 PW 내에 도착 가능한 비디오 i 의 요청 회수는 $p_i \times \lambda \times PW$ 이다. 패칭 윈도우 시작지점의 요청에 대해서는 정규 멀티캐스트가 수행되며 그 다음 요청부터는 패칭 윈도우를 벗어나지 않는 동안은 패칭 채널을 통해 패칭 멀티캐스트가 수행된다. 이 패칭 채널은 스큐에 해당하는 패칭 스트림을 전송한 후 가용상태가 되어 다음 패칭 윈도우안의 요청들에 대한 패칭 멀티캐스트를 수행할 수 있으므로 k 개의 인기 비디오를 위해 필요한 패칭 채널의 수는 식 (2)에 의해 얻어진다.

$$\sum_{i=1}^k (p_i \times PW \times \lambda - 1) \quad (2)$$

그러므로 k 개의 인기 비디오를 위하여 정규 채널과 패칭 채널을 모두 합한 총 채널 예약 용량은 식 (3)과 같다.

$$k \times \frac{L}{PW} + \sum_{i=1}^k (p_i \times PW \times \lambda - 1) \quad (3)$$

채널 예약 패칭 방법을 사용하는 VOD 시스템에서 서버와 클라이언트의 실행 과정은 다음과 같다. 서버는 먼저 인기 비디오를 위한 정규 채널과 패칭 채널을 예약해둔다. 식별자가 *ClientID*인 클라이언트는 특정 비디오 *VideoID*를 요청하기 위해 요청 토큰 (*ClientID*, *VideoID*)를 서버로 전송한다. 클라이언트의 요청 토큰은 서버의 대기큐에 저장되고, 가용 채널이 있을 때 스케줄링 정책에 의해 스케줄링 되면 서버는 클라이언트에게 서비스 토큰 (PID, RID)를 전송한다. 클라이언트는 서비스 토큰을 통해 정규 스트림은 채널 RID를 통해, 패칭 스트림은 채널 PID를 통해 전송됨을 확인한다. (그림 3)은 이러한 채널 예약 패칭 방법을 사용하는 서버의 알고리즘을 보여준다. 스케줄된 요청의 *VideoID*가 인기 비디오라면 예약 채널을 할당하고, 인기가 없는 비디오라면 비 예약 채널을 할당한다. 이때 패칭 여부는 스케줄된 현재 요청의 패칭 윈도우 안에 이미 진행중인 *VideoID*에 대한 정규 멀티캐스트가 있느냐에 따라 정해지며 채널로 멀티캐스트될 비디오 데이터 부분은 (그림 4)의 알고리즘에 의해 구해진다. (그림 3)에서 서버는 *VideoID*에 대한 정규 멀티캐스트가 없는 경우에는 *VideoID*에 대한 새로운

정규 멀티캐스트가 수행되어야 하므로 가용 채널 *FreeChannel*을 정규 채널 RID로 할당하여 정규 멀티캐스트를 수행하며, 정규 멀티캐스트가 있는 경우는 다시 패칭 윈도우를 기준으로 두 가지 경우로 나뉜다. 먼저 *VideoID*에 대한 가장 최근의 정규 멀티캐스트가 정규 채널 *LatestRegular* 상으로 수행되고 있고 정규 멀티캐스트가 패칭 윈도우 안에 있다면 가용 채널인 *FreeChannel*을 패칭 채널 PID로 할당하여 PID를 통해 패칭 멀티캐스트를 수행하며 *LatestReg-*

Algorithm: Server Main Routine

1. Reserve the I/O channels of the video server for hot videos ;
2. Select the next video, say v , to serve according to FCFS policy ;
3. If the video v is hot video, then
 - Dispatch a free channel *FreeChannel* from reserved channels ;
 - else Dispatch a free channel *FreeChannel* from not reserved channels ;
4. Initialize the service token as ($PID = null, RID = null$) ;
5. If there is no regular multicast of the video v in progress, then
 - set $RID = FreeChannel$.
 - else
 - Set $PID = FreeChannel$ and $RID = LatestRegular$, where *LatestRegular* is the latest regular channel for the video v ;
 - call *MulticastData(FreeChannel, LatestRegular)* to determine the portion of video data which should be multicast on channel *FreeChannel* ;
6. For each request token ($vClient, v$) in a *WaitingQueue*, do the following :
 - If PID is null, then
 - append $vClient$ to the client list of channel RID ;
 - else it is appended to the client lists of both channels PID and RID ;
 - Send the service token (PID, RID) to the client $vClient$;
 - Delete the request token from the *WaitingQueue* ;
7. Activate the multicast on *FreeChannel* ;

(그림 3) 비디오 서버의 알고리즘

Algorithm : MulticastData(FreeID, RegularID)

- t : Current time
 t_s : Start time of the regular multicast on channel *RegularID*
 V : The video currently multicast on channel *RegularID*
 $|V|$: Playback duration of the video V
 $V[l]$: The workload for a channel, that is the amount of video data which should be transmitted on the channel. $V[l]$ represents the data of video V during l , where l is the desired playback duration for the video V .
 PW : size of patching window
1. If $t - t_s \leq PW$, then the workload for channel *FreeID* is $V[t - t_s]$.
 2. Otherwise, *FreeID* is designated to start a new regular multicast as follows :
 - Modify the service token as ($PID = null, RID = FreeID$).
 - Set the workload for the new regular channel *FreeID* as $V[|V|]$.

(그림 4) 멀티캐스트 데이터 결정 알고리즘

ular를 RID로 지정함으로써 정규 스트림을 공유하도록 한다. 두 번째로 정규 멀티캐스트가 패칭 윈도우 밖에 있다면 새로 비디오 전체를 전송하기 위해 가용 채널 *FreeChannel*을 정규 채널 RID로 할당하여 RID상으로 정규 멀티캐스트를 수행한다.

(그림 5)는 채널 예약 패칭 방법에서 클라이언트 측에서 수행되는 알고리즘을 보여준다. 서버가 전송한 서비스 토큰(PID, RID)를 수신한 클라이언트는 PID가 null일때와 아닐때로 구분되어 수행된다. PID가 null이라면 정규 스트림만 수신하면 되므로 데이터 로더 L_r 만 활성화하여 정규 스트림을 다운로드받고, *VideoPlayer*는 수신한 비디오 프레임들을 스크린에 상영한다. 만일 PID가 null이 아니라면, 서버가 패칭 채널 PID상에 패칭 스트림을 전송할 것이므로 클라이언트는 RID와 PID로부터 동시에 다운로드받기 위하여 로더 L_r 과 L_p 모두를 활성화한다. *VideoPlayer*는 먼저 PID로부터 수신한 패칭 스트림을 재생하면서 RID로부터의 정규 스트림을 국부 버퍼에 저장한다. 패칭 멀티캐스트가 끝나면 *VideoPlayer*는 L_r 의 국부 버퍼에 미리 저장되어 있는 비디오의 다음 부분을 상영한다. 이때 데이터 로더 L_r 은 계속해서 RID로부터 정규 스트림을 다운로드받아 국부 버퍼에 저장함으로써 비디오를 연속하여 재생할 수 있도록 한다.

Algorithm: Client Main Routine

1. Send a request token (*ClientID*, *VideoID*) to the video server.
2. Wait until the service token (*PID*, *RID*) from the server arrives.
3. If *PID* is not null, start the data loader L_p .
4. Start the data loader L_r .
5. Start the video player *VideoPlayer*.

Algorithm: Loader L_p

1. Do the following until no more data arrive on the patching channel *PID* :
 - Download one data packet on channel *PID* ;
 - Store the data packet in *PatchBuffer*.
2. Terminate L_p .

Algorithm: Loader L_r

1. Do the following until no more data arrive on the regular channel *RID* :
 - Download one data packet on channel *RID* ;
 - Store the data packet in *RegularBuffer*.
2. Terminate L_r .

Algorithm: VideoPlayer

1. Do the following until no more data in *PatchBuffer* :
 - Fetch one playback unit from *PatchBuffer* ;
 - Free the disk space for the fetched data ;
 - Reassemble the fetched data into frames and render them onto the screen.
2. Do the following until no more data in *RegularBuffer* :
 - Fetch one playback unit from *RegularBuffer* ;
 - Free the disk space for the fetched data ;
 - Reassemble the fetched data into frames and render them onto the screen.
3. Terminate *VideoPlayer*.

(그림 5) 클라이언트의 알고리즘

5. 시뮬레이션 및 성능 평가

본 논문에서는 시뮬레이션을 통하여 서버의 채널 용량을 예약하지 않는 단순 패칭과 인기 비디오 수에 따라 서버의 채널 용량을 예약하는 채널 예약 패칭 방법의 성능을 평가하고 비교한다. 여기서 고려된 성능 평가 항목은 비디오 요청 도착율에 따른 사용자의 이탈율, 평균 대기 시간, 공정성, 버퍼량 등이다.

5.1 실험 환경

시뮬레이션은 IBM PC-Pentium에서 Turbo-C 언어를 사용하여 시뮬레이션 프로그램을 작성하였으며 사용된 파라미터는 <표 2>와 같다. 여기서 비디오 *i*의 인기도는 Zipf 법칙을 따른다고 가정한다. 이것은 비디오 관람 패턴에 대한 통계에 의하면 특정 비디오에 대해 인기도가 편중되는 것으로 나타나며 이런 지역성은 Zipf 법칙을 사용하여 표현할 수 있기 때문이다[20, 21]. 그리고 이탈 시간은 관람자들의 일반적인 행동을 반영하여 3분~7분으로, 비디오의 길이는 일반적인 영화 길이인 90분으로 설정하였다.

<표 2> 시뮬레이션 파라미터

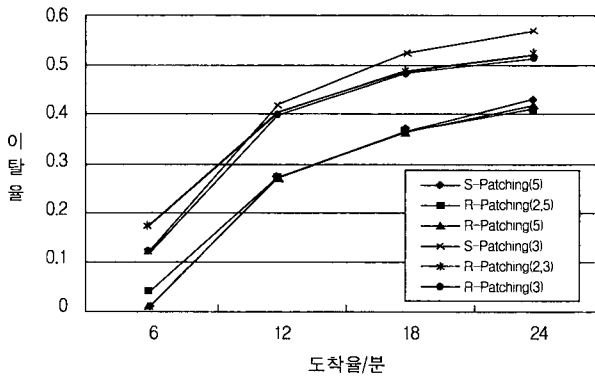
파라미터	값
비디오 개수	50개
서버 네트워크 채널	300 채널
패칭 윈도우	3, 5분
도착율/분	6, 12, 18, 24(포아송)
비디오 길이	90분
이탈시간	3분~7분(랜덤)

사용자는 패칭을 위한 충분한 버퍼 공간을 가지며, 되감기, 정지, 빨리 감기 등의 요청을 하지 않고 단순히 비디오의 시작부터 끝까지 순차적으로 본다 가정한다. 시뮬레이션은 540분동안의 사용자 요청에 대해서 수행되었으며 공정성을 위해 서비스를 시작한 후 100분과 500분 사이에 도착한 요청들에 대해 각 정책의 성능을 평가하였다.

5.2 성능평가 결과

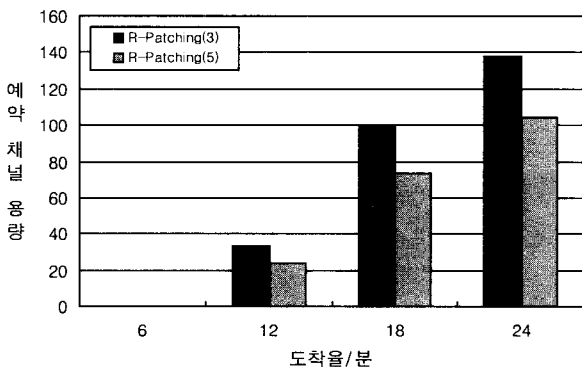
(그림 6)은 비디오 요청 도착율에 따른 패칭 정책별 이탈율을 보여준다. R-Patching(x)과 S-Patching(x)는 각각 x분의 패칭 윈도우를 갖는 가변 채널 예약 패칭과 단순 패칭을 나타내고, R-Patching(k, x)는 패칭 윈도우 크기가 x분이고 인기 비디오의 수가 k개인 고정 채널 예약 패칭을 나타낸다. 이탈율은 이탈 회수를 전체 요청 회수로 나눈 값을 사용한다. 도착율이 6일때 고정 채널 예약 패칭의 이탈율은 다른 방법보다 높게 나타났다. 이것은 도착율이 낮음에도 불구하고 k=2로 고정되어 있어서 서버의 채널을 너무 많이 예약함으로써 채널의 낭비가 발생하기 때문이다. 그러

나 나머지의 경우에는 패칭 윈도우 크기가 클수록 이탈율이 낮으며, 비디오 요청 도착율이 증가할수록 제안하는 채널 예약 패칭 방법들이 단순 패칭보다 낮은 이탈율을 보이며 고정 채널 예약보다는 가변 채널 예약 패칭이 이탈율이 더 나음을 알 수 있다. 이것은 단순 패칭 정책에서는 서버의 과부하 상황에서 모든 비디오 요청들이 블록되는데 반해 채널 예약 패칭 방법은 인기 비디오를 위한 서버 채널 용량이 예약되어 있기 때문에 서버가 과부하인 상황에서도 요청 수가 많은 인기 비디오에 대한 요청들이 서비스될 수 있기 때문이다.



(그림 6) 요청 도착율에 따른 이탈율

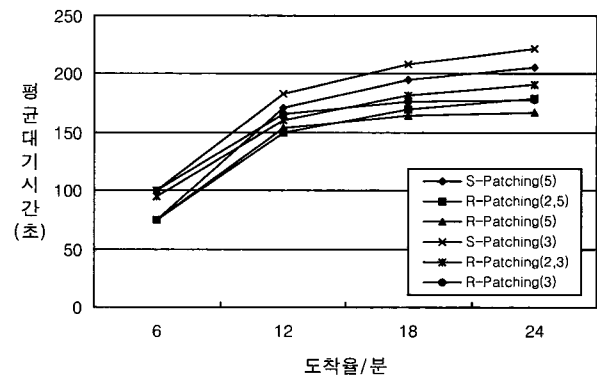
(그림 7)은 가변 채널 예약 패칭에서 패칭 윈도우의 크기와 요청 도착율에 따라 인기 비디오를 위하여 예약되는 서버 채널의 용량을 보여준다. 비디오 요청 도착율이 증가할수록 채널 예약 용량은 증가함을 알 수 있다. 그리고 패칭 윈도우가 3분인 경우보다 패칭 윈도우가 5분일 경우 예약되는 서버 채널 용량이 더 작음을 볼 수 있다. 이것은 패칭 윈도우가 크면 동일 비디오에 대한 요청이 많이 나타나므로 패칭 채널의 예약 용량은 증가하지만 상대적으로 정규 채널의 예약 용량은 감소하기 때문에 전체 예약 채널의 용량이 감소하기 때문이다.



(그림 7) 요청 도착율에 따른 가변 예약 채널 용량

(그림 8)은 비디오를 요청한 사용자가 대기하다가 이탈한

경우도 포함한 모든 요청의 총 대기시간을 총 비디오 요청 수로 나눈 평균 대기시간을 보여준다. 제안하는 채널 예약 패칭 방법이 단순 패칭 방법보다 평균 대기시간이 감소된 것을 알 수 있다. 또한 요청 도착율이 증가할수록 가변 채널 예약 패칭이 고정 채널 예약 패칭보다 평균 대기시간이 감소되는 것을 알 수 있다. 이것은 단순 패칭 방법에서는 인기 비디오와 비인기 비디오의 구분이 없으므로 가용 채널이 없어서 대기하게 되는 많은 인기 비디오의 요청들이 채널 예약 패칭 방법에서는 예약된 채널을 통해 즉시 스케줄되어 서비스될 수 있기 때문이다.



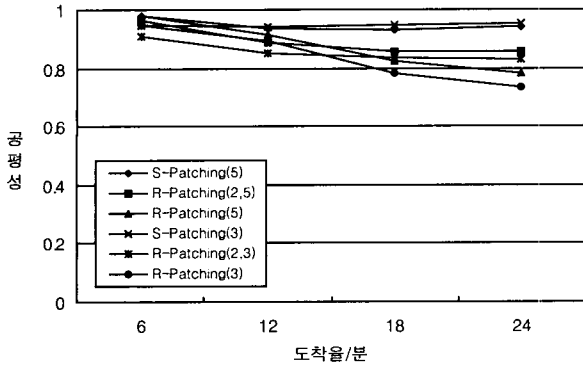
(그림 8) 모든 요청에 대한 평균 대기 시간

(그림 9)는 전체 비디오에 대한 서비스의 공평성을 보여주며 (그림 10)과 (그림 11)은 각각 인기 비디오와 비인기 비디오에 대한 평균 대기시간을 단순 패칭의 모든 비디오에 대한 평균 대기시간과 비교한 결과를 보여준다. 비디오 i 의 이탈율은 d_i , 평균 이탈율은 d , 전체 비디오 수는 N 이라고 할때, 공평성은 식 (4)와 같이 계산된다. 이 식은 Hue [3]의 연구에서 정의된 불공평에 대한 식을 공평성을 나타내는 식으로 바꾼 것이다.

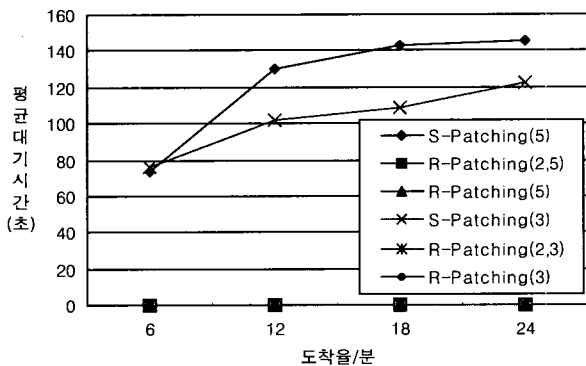
$$1 - \sqrt{\frac{\sum_{i=1}^N (d_i - d)^2}{N-1}} \quad (4)$$

그림에서 볼 수 있듯이 채널 예약 패칭 방법이 단순 패칭보다 공평성이 좋지 않으며 고정 채널 예약 방법보다는 가변 채널 예약 방법이 더 좋지 않음을 알 수 있다. 이것은 채널 예약 패칭 방법이 인기 비디오를 위해 서버의 채널 일부를 예약해두기 때문에 인기 비디오 요청들의 대기시간은 단순 패칭보다 더 작지만 비인기 비디오에 대한 요청들의 대기 시간은 더 커지기 때문이다. 그러나 (그림 10)을 살펴보면 채널 예약 패칭 방법에서의 인기 비디오의 평균 대기 시간은 대부분이 0으로 단순 패칭과는 비교할 수 없을 만큼 작으며 (그림 11)을 살펴보면 채널 예약 패칭 방법에서의 비인기 비디오의 평균 대기시간은 20초 이하의 차이가 있음을 알 수 있다. 즉 채널 예약 패칭 방법의 공평성이

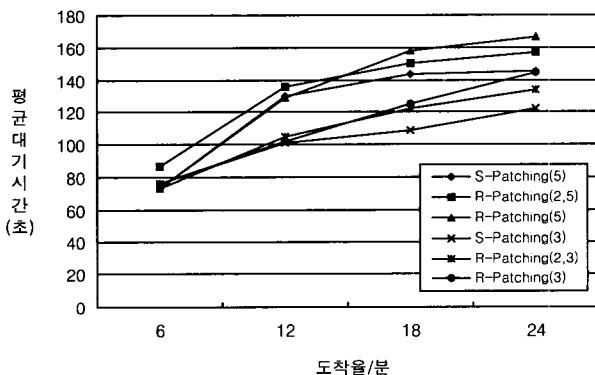
떨어진 것은 비인기 비디오의 평균 대기시간은 아주 조금 증가했지만 인기 비디오의 평균 대기시간이 최소화 되었기 때문이다. 그러므로 이러한 분석과 채널 예약 패칭 방법의 전체 평균 대기시간과 이탈율이 개선된 것을 고려할 때 공평성의 저하는 실제적으로 큰 문제가 되지 않는다.



(그림 9) 요청 도착율에 따른 공평성



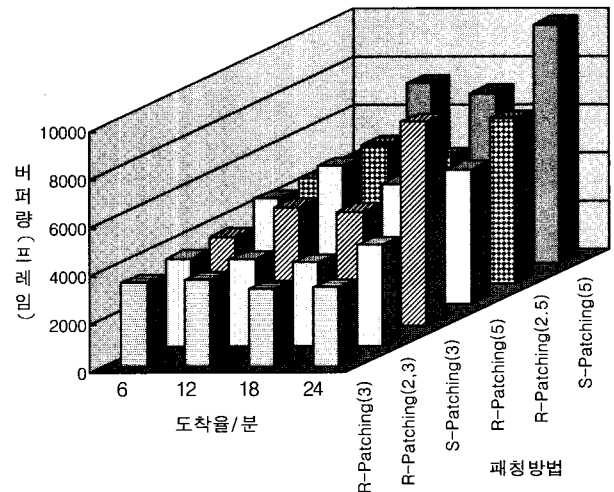
(그림 10) 인기 비디오의 평균 대기 시간



(그림 11) 비인기 비디오의 평균 대기 시간

(그림 12)는 모든 사용자 스테이션에 대한 평균 버퍼량을 보여준다. 패칭 윈도우의 크기가 클수록 사용자에 많은 버퍼 공간이 필요함을 알 수 있다. 이것은 패칭 윈도우의 크기가 클수록 패칭 시간이 늘어나서 정규 스트림을 버퍼링하는 양이 많아지며, 요청 도착율이 증가할수록 즉시 스케

줄되지 못하고 대기 큐에 있다가 서비스되는 요청이 증가함으로써 패칭 스트림의 양이 증가하게 되어 버퍼링이 많이 되기 때문이다. 그림에서 볼 수 있듯이 단순 패칭에 비해 채널 예약 패칭이 평균 30% 정도의 버퍼량의 감소를 보였으며, 가변 채널 예약 방법이 고정 채널 예약 방법보다 7.6% 정도 버퍼량이 감소됨을 알 수 있다. 이것은 채널 예약 패칭 방법에서는 인기 비디오를 위해 정규 채널과 패칭 채널을 예약해둠으로써 인기 비디오의 요청이 대기시간 없이 즉시 서비스될 수 있기 때문이다. 이와 같이 앞의 여러 가지 평가 항목에 대한 성능평가 결과 단순 패칭보다는 채널 예약 패칭 방법이 우수하며, 고정 채널 예약 방법보다는 가변 채널 예약 방법이 우수함을 보였다.



(그림 12) 도착율에 따른 버퍼 사용량

6. 결 론

VOD 서비스를 위한 멀티캐스트 기법은 동일한 비디오 요청들에 대하여 단일 스트림으로 전송하여 서비스하는 방법으로 서버의 네트워크 대역폭을 절감시키는 우수한 기법이다. 그러나 동일한 비디오 요청들을 모아서 전송하기 때문에 대기시간이 길어진다. 그러므로 멀티캐스트 기법은 대역폭은 감소시킬 수 있지만 대기시간의 증가로 true VOD 서비스를 제공할 수 없다. 본 논문에서는 이러한 대기시간과 멀티캐스트의 상반관계를 해결하기 위하여 패칭 기법을 기반으로 한 채널 예약 패칭 방법을 연구하였다.

패칭은 멀티캐스트를 사용하여 대역폭을 감소시키면서도 대기시간 없이 사용자에게 비디오 스트림을 전송하는 기법이다. VOD 서비스를 요청하는 사용자들의 데이터 접근 형태를 살펴보면 모든 비디오를 균등하게 접근하는 것이 아니라 인기 비디오에 대한 요청은 자주 발생되고 인기가 없는 비디오에 대한 요청은 비교적 적은 특징을 가진다. 이와 같이 VOD 서비스에서 특정 비디오에 대한 인기가 편중

되어있기 때문에 대부분의 인기 비디오의 경우 짧은 순간에 새로운 사용자 요청이 많이 도착하고 서버는 이 요청들에 대해 실시간으로 사용자들에게 요청한 비디오를 제공해야 한다. 그러므로 본 논문에서는 인기 비디오를 위하여 채널 용량을 예약해둠으로써 자주 발생하는 인기 비디오에 대한 서비스를 보장하는 채널 예약 패칭 방법을 제안하였다. 그리고 제안하는 방법의 성능을 시뮬레이션을 통하여 평가하고 그 결과를 비교 분석하였다. 성능 평가 결과 비디오 요청 도착율에 따른 이탈율, 평균 대기시간, 공평성, 버퍼 사용량 모두에 대해 제안한 채널 예약 패칭 방법이 단순 패칭 방법보다 성능이 우수함을 알 수 있었다. 또한 제안한 두 가지 채널 예약 방법인 고정 채널 예약과 가변 채널 예약의 성능을 비교해 본 결과 고정 채널 예약보다는 가변 채널 예약의 성능이 우수하였으며, 두 방법 모두 패칭 윈도우의 크기가 클수록 버퍼되는 양이 증가함을 알 수 있었다.

결론적으로 본 논문이 제안한 채널 예약 패칭 방법은 자주 발생하는 인기 비디오에 대한 서버 용량을 예약해둠으로써 서버의 과부하 상태에서 발생할 수 있는 사용자 이탈을 감소시키고, 패칭 방법을 기반으로 멀티캐스트 데이터를 공유함으로써 네트워크 부하를 감소시키며, 사용자의 요청을 대기시간 없이 즉시 서비스할 수 있으므로 true VOD 서비스의 구현을 가능하게 한다. 향후 연구 과제는 서버의 여러 가지 상태에 따라 인기 비디오를 결정하기 위한 보다 좋은 기준과 예약 채널 용량을 최적으로 결정하기 위한 기준을 결정하는 것이다.

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, 1996.
- [2] Christer Samueleson, "Relating Turing's Formula and Zipf's Law," Proceedings of the Fourth Workshop on Very Large Corpora, 1996.
- [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] S. Sen, L. Gao, J. Rexford, and D. Towsley, "Optimal Patching Schemes for Efficient Multimedia Streaming," Tech. Rep. 99-2, Department of Computer Science, University of Massachusetts Amherst, p.18, 1999.
- [5] Y. Cai, K. Hua, and K. Vu, "Optimizing Patching Performance," *In Proc. SPIE/ACM Conference on Multimedia Computing and Networking*, pp.204-215, 1999.
- [6] D. L. Eager, M. K. Vernon, and J. Zahorjan, "Optimal and Efficient Merging Schedules for Video-on-Demand Servers," *In Proceedings ACM Multimedia 99*, pp.199-202, 1999.
- [7] T. Little and D. Venkatesh, "Prospect for Interactive Video-on-Demand," *IEEE Multimedia Magazine*, pp.14-24, 1994.
- [8] D. Delodere, W. Verbiest, and H. Verhile, "Interactive Video On Demand," *IEEE Communication Magazine*, pp.82-88, 1994.
- [9] DAVIC Specification 1.0, <http://www.davic.org>.
- [10] F. A. Tobagi et al., "Streaming RAID : A Disk Storage System for Video and Audio Files," *In Proc. ACM Multimedia 93*, pp.393-400, 1993.
- [11] Dan, A., Sitaram, D., and Shahabuddin, P., "Scheduling Policies for an On-Demand Video Server with Batching," *In Proc. of the 2nd ACM Multimedia Conference*, pp.25-32, 1994.
- [12] D. L. Eager, M. K. Vernon, and J. Zahorjan, "Optimal and Efficient Merging Schedules for Video-on-Demand Servers," *In Proc. of the 7-th ACM International Multimedia Conference*, pp.199-203, 1999.
- [13] 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.
- [14] L. Golubchik, J. Lui, and R. Muntz, "Adaptive Piggybacking: Arrival Technique for Data Sharing in Video-on-Demand Service," *ACM Multimedia Systems*, Vol.4, No.3, pp.140-155, 1996.
- [15] K. Hua and S. Sheu, "Skyscraper broadcasting : A New Broadcasting Scheme for Metropolitan Video-on-Demand Systems," *In Proc. ACM SIGCOMM*, pp.89-100, 1997.
- [16] D. Eager and M. Vernon, "Dynamic Skyscraper Broadcasts for Video-on-Demand," *In Proc. 4th Int. Workshop on Advances in Multimedia Information Systems*, pp.18-32, 1998.
- [17] J. Korst, "Random Duplicated Assignment: An Alternative to Striping in Video Servers," *In Proc. of ACM Int'l Multimedia Conference*, pp.219-226, 1997.
- [18] M. S. Chen and D. D. Kandlur, "Stream conversion to support interactive video playout," *IEEE Multimedia magazine*, Vol.3, No.2, pp.51-58, 1996.
- [19] W. Feng, F. Jahanian, and S. Sechrest, "Providing VCR Functionality in a Constant Quality Video-on-Demand Transportation Service," *In Proc. of the IEEE Int'l Conf. on Multimedia Systems'96*, pp.25-32, 1996.
- [20] 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.
- [21] A. Dan and D. Sitaram, "A Generalized Caching Policy for Mixed Interactive and Long Video Workloads," *IBM Research Report, RC 20206*, Yorktown Heights, NY, 1995.



이 주 영

e-mail : chu_young@hanmail.net

2000년 대구가톨릭대학교 컴퓨터정보통신
공학부(학사)

2002년 대구가톨릭대학교 전산통계학과
전산전공(석사)

2002년~현재 엠비즈 네트워크

관심분야 : 무선 인터넷 서비스, 멀티미디어 시스템 등



하 숙 정

e-mail : sjha@m80.knu.ac.kr

1988년 계명대학교 전자계산학과(학사)

1990년 중앙대학교 전자계산학과(석사)

1998년 대구가톨릭대학교 전산통계학과
전산전공(박사)

2001년~현재 경북대학교 전자전기컴퓨터
학부 조교수(BK21)

관심분야 : 모바일컴퓨팅, 분산컴퓨팅, 멀티미디어 시스템 등



배 인 한

e-mail : ihbae@cuth.cataegu.ac.kr

1984년 경남대학교 전자계산학과(학사)

1986년 중앙대학교 대학원 전자계산학과
(석사)

1990년 중앙대학교 대학원 전자계산학과
(박사)

1996년~1997년 Computer and Information Science, The Ohio
State University(Post-doctoral)

2002년~현재 Department of Computer Science, Old Dominion
University(Visiting Professor)

1989년~현재 대구가톨릭대학교 컴퓨터정보통신공학부 교수

관심분야 : 이동무선망, 무선인터넷, M-Commerce, 멀티미디어
시스템 등