

대규모 VOD 시스템을 위한 효율적인 P2P 기반의 프록시 패칭 기법

권 춘 자[†] · 최 황 규^{††}

요 약

대규모 VOD 시스템에서는 동시에 다수의 클라이언트 요청에 따른 서버 저장장치 및 네트워크 대역폭 병목현상이 발생하며, 이를 해소하기 위한 효율적인 서버 대역폭 감소 기법이 필요하다. 패칭 기법은 멀티캐스트를 활용한 대표적인 VOD 시스템의 병목현상 해결 기법 중의 하나이다. 본 논문은 기존 패칭 기법의 성능 향상을 위하여 프리픽스 캐싱과 P2P 프록시를 결합한 새로운 P2P 프록시 패칭 기법을 제안한다. 제안된 기법에서 각각의 클라이언트는 자신이 받고 있는 스트림을 동일한 비디오를 요청한 다른 클라이언트들에게 멀티캐스트 함으로써 다른 클라이언트를 위한 프록시 역할을 수행한다. P2P 프록시와 프리픽스 캐싱의 사용으로 인하여 패칭 윈도우 크기를 벗어나는 사용자 요청들은 새로운 정규 채널을 생성하는 대신 이전의 패칭 그룹에 속한 다른 클라이언트로부터 정규 스트림을 전송받음으로써 서버로 집중되는 부하를 줄일 수 있다. 시뮬레이션을 통하여 프리픽스 크기와 요청 간격에 따른 서버의 대역폭 요구량을 측정한 결과에서 제안된 기법이 기존의 패칭 방법보다 서버의 평균 대역폭 요구량을 전체적으로 약 33% 감소시킬 수 있음을 보인다.

키워드 : 멀티미디어, 패칭, P2P, 프리픽스 캐싱, 프록시 서버, VOD 서버

An Efficient P2P Based Proxy Patching Scheme for Large Scale VOD Systems

Chun Ja Kwon[†] · Hwang Kyu Choi^{††}

ABSTRACT

The main bottleneck for large scale VOD systems is bandwidth of storage or network I/O due to the large number of client requests simultaneously, and then efficient techniques are required to solve the bottleneck problem of the VOD system. Patching is one of the most efficient techniques to overcome the bottleneck of the VOD system through the use of multicast scheme. In this paper, we propose a new patching scheme, called P2P proxy patching, for improving the typical patching technique by jointly using the prefix caching and P2P proxy. In our proposed scheme, each client plays a role in a proxy to multicast a regular stream to other clients that request the same video stream. Due to the use of the P2P proxy and the prefix caching, the client requests that arrive out of the patching window range can receive the regular stream from other clients in the previous patching group without allocating the new regular channels from the VOD server to the clients. In the performance study, we show that our patching scheme can reduce the server bandwidth requirement about 33% less than that of the existing patching technique with respect to prefix size and request interval.

Key Words : Multimedia, Patching, Peer To Peer, Prefix Caching, Proxy Server, VOD Server

1. 서 론

최근 인터넷 활용의 폭발적인 증가와 디지털 콘텐츠 제작 기술의 발전에 힘입어 멀티미디어를 기반으로 하는 인터넷 서비스가 널리 확대 보급되고 있다. 멀티미디어를 기반으로 하는 서비스는 다수의 클라이언트가 서버에 접속하여 정보

를 요청하고 수신하는 서비스로 VOD(Video on Demand), NOD(News on Demand), AOD(Audio on Demand) 등 다양하다. 그 중 대표적인 예가 실시간 VOD 서비스로 원격 교육, 온라인 오락, 디지털 비디오 도서관, 전자상거래 등 인터넷을 기반으로 하는 모든 응용 분야에 폭넓게 활용되고 있다. VOD 서비스는 많은 클라이언트들이 각자 원하는 시간에 인터넷이나 전용 통신망을 통하여 미디어 데이터를 저장하고 있는 VOD 서버에 접근하여 원하는 비디오를 실시간으로 끊임 없이 연속적으로 볼 수 있게 하는 서비스이다. 이러한 VOD 서비스는 인터넷 속도의 향상과 무선 인터넷

* 본 논문은 2005년 정보통신기초기술연구지원사업(과제번호 04-기초-004) 연구결과와 일부임.

† 준 회 원 : 강릉영동대학 사이버경찰과 초빙교수

†† 정 회 원 : 강원대학교 전기전자정보통신공학부 교수

논문접수 : 2005년 4월 28일, 심사완료 : 2005년 7월 19일

의 출현 등으로 인하여 앞으로도 그 응용분야가 간단한 광고 비디오 클립 전송에서 시작하여 기존의 TV나 유선 방송을 대체할 정도로 그 범위를 더욱 넓혀 갈 것이 예상된다[24].

특히 VOD 서비스를 요구하는 클라이언트 수는 기하급수적으로 증가하는 추세로 많은 네트워크 자원과 큰 저장 용량을 가진 시스템 자원을 필요로 한다. 그러나 고도의 컴퓨팅 기술과 통신 기술의 발전에도 불구하고 각 클라이언트들에게 충분한 자원을 할당해 주기는 어렵다. 각 클라이언트가 VOD 서버로부터 비디오 스트림을 전송해 보기 위해서는 일반적으로 높은 서버 입출력 대역폭(bandwidth)과 네트워크 입출력 대역폭을 필요로 한다. 따라서 네트워크의 병목현상(bottleneck)으로 인하여 서버의 부하(load)가 증가하고 서비스 시간이 지연되는 현상을 발생시킨다. 이것은 하드웨어 자원과 네트워크 비용의 증가로 전체 시스템 관리 비용 증가를 초래한다. 클라이언트 수가 증가하여도 모든 클라이언트는 고품질로 신속하게 항상 동일한 서비스를 받기를 원한다. 한정된 자원으로 자원 전체 관리 비용을 줄여 시스템의 성능을 향상시키는 것이 무엇보다 필요하다. 통신망의 특성과 상태, 데이터의 전송률 등이 일정하게 유지될 때, VOD 서버의 입출력과 네트워크 대역폭을 줄이면 시스템의 성능 향상과 서비스의 질(QoS: Quality of Service)을 높일 수 있다.

이에 따라 지금까지 VOD 서버의 대역폭을 줄이기 위한 많은 연구가 이루어졌으며, 최근에는 패칭을 통한 VOD 시스템의 성능 향상과 프록시 서버를 활용하는 방법에 대한 연구가 활발히 이루어지고 있다[10, 11]. 패칭(patching) 기법은 VOD 서비스를 위해 멀티캐스트(multicast)를 이용한 가장 효율적인 방법 중의 하나이다[8]. 패칭 기법에서는 동일한 비디오를 요청하는 클라이언트 사이에서 하나의 패칭 윈도우 안에 임의의 시간차로 도착하는 클라이언트들이 비디오의 앞부분을 재생하는 동안 서버로부터 멀티캐스트되는 정규 스트림을 버퍼링하여 사용함으로써 VOD 서버의 네트워크 대역폭을 감소시킨다. 이렇게 하여 클라이언트들에게 스트림의 프리픽스(prefix)를 전송하기 위한 채널만 추가함으로써 시간 지연 없이 즉시 서비스할 수 있도록 한다. 그런데 이 경우에도 하나의 패칭 윈도우마다 서버와 클라이언트 사이에 새로운 통신 채널을 생성해야 하므로 부하 집중에 따른 VOD 서버의 대역폭의 한계는 여전히 존재한다.

서버로 집중되는 부하를 줄이기 위한 또 다른 방법으로써는 프록시(proxy) 서버를 이용하여 VOD 서버의 역할 중 일부를 대신하도록 하는 기법이 있다[7, 17, 21, 22, 27]. 프록시 서버는 클라이언트 가까이서 클라이언트가 원거리의 VOD 서버로부터 비디오 데이터의 전송에 드는 시간 지연과 대역폭을 절약하기 위해 사용된다. 특히 프록시 프리픽스 패칭 방법은 인기 있는 비디오(popular video)의 프리픽스 스트림들을 프록시 서버에 저장하여 즉시 서비스함으로써 빠른 서비스를 제공한다. 만일 클라이언트로부터 비디오에 대한 요청이 들어오면, 프록시는 클라이언트에게 프리픽스 스트림의 전송을 시작하는 동시에 나머지 스트림을 서버

로부터 전송 받는다. 그렇지만 이 경우에도 프록시 서버 용량의 한계 등으로 VOD 서버에 대한 부하의 집중을 궁극적으로 해결하지는 못한다.

한편, 최근에는 VOD 서버의 중앙 집중 방식에 따른 한계를 극복하기 위하여 분산 환경 하에서 중앙 집중 서버 대신 클라이언트를 통해 미디어 스트림을 전송받는 P2P(peer to peer) 기반의 미디어 스트리밍 방법에 대한 연구가 활발히 진행되고 있다[5, 6, 18, 19]. P2P 미디어 스트리밍 방식에서는 빠르고 신속한 서비스를 위해 WAN 환경에 속한 다수의 피어(peer)들을 트리 또는 선형 구조로 구성하여 피어들 간에 스트림을 공유함으로써 클라이언트 수의 증가에 따른 VOD 서버의 네트워크 대역폭을 감소시킬 수 있다. 본 논문에서는 주어진 VOD 서버 대역폭의 한계를 극복하여 대규모 VOD 시스템의 실현을 위해 P2P 환경 하에서 패칭 기법을 적용하는 방법에 대하여 기술한다.

이를 위하여 본 논문은 기존의 프록시 프리픽스 패칭 기법과 클라이언트 시스템의 스트림 버퍼링 및 재전송 방법을 사용하여 서버가 아닌 클라이언트로부터 정규 스트림을 전송받는 **P2P 프록시 패칭 기법**을 제안한다. 제안된 기법은 패칭 윈도우의 크기를 벗어나는 사용자 요청에 대하여 VOD 서버가 새로운 정규 채널을 생성하는 대신, 이전의 패칭 그룹에 속한 클라이언트로부터 정규 스트림을 전송받아 사용함으로써 VOD 서버로 집중되는 부하를 줄이는 방법이다. 또한 기존의 P2P 방법이 WAN 환경 상에서 멀티캐스트 트리 구조를 유지하기 위하여 여러 차례에 걸쳐 메시지를 교환해야 하는 오버헤드의 단점을 줄이고자 한다. 본 논문에서는 프록시 서버를 기반으로 하는 LAN 환경 하에서 클라이언트들을 패칭 윈도우 단위로 묶어 선형 구조의 패칭 그룹으로 구성한다. 또한 모든 클라이언트들의 상태를 알 수 있도록 프록시 서버 내에 인덱스를 유지한다. 이 인덱스는 패칭 그룹을 형성하여 서비스 받고 있는 모든 클라이언트들의 현재 상태 정보가 저장되며, 상태 변화 발생 시 즉시 갱신되어 유지된다. 클라이언트의 상태 변화로 인해 인덱스 구조를 재구성하여야 할 때 프록시는 클라이언트와 한번의 메시지 교환으로 인덱스를 갱신한다. 이렇게 유지되는 인덱스는 클라이언트의 서비스 요청 시 참조되어 서비스 효율을 높인다. 그리고 P2P 스트리밍은 불안정한 클라이언트로부터 정규 스트림의 재전송을 수행하므로 클라이언트 이탈시의 스트림 전송을 위한 복구 과정이 필요하다. 따라서 본 논문은 클라이언트 이탈시 스트림 재전송을 위한 복구 과정도 함께 제안한다. 성능 평가를 통해 프리픽스 크기와 평균 요청 간격, 그리고 비디오의 접근 패턴에 따른 서버의 대역폭 요구량을 측정하여 기존의 패칭 방법과 비교함으로써 우수함을 입증한다.

본 논문의 구성은 다음과 같다. 먼저 2장에서는 P2P 미디어 스트리밍 방법과 이와 관련된 지금까지의 연구결과들에 대하여 살펴봄으로써 본 논문의 연구 배경을 설명한다. 3장에서는 본 논문에서 제안한 P2P 프록시 패칭 기법을 기본 동작 개념, 인덱스 구조와 클라이언트 이탈에 따른 복구 과

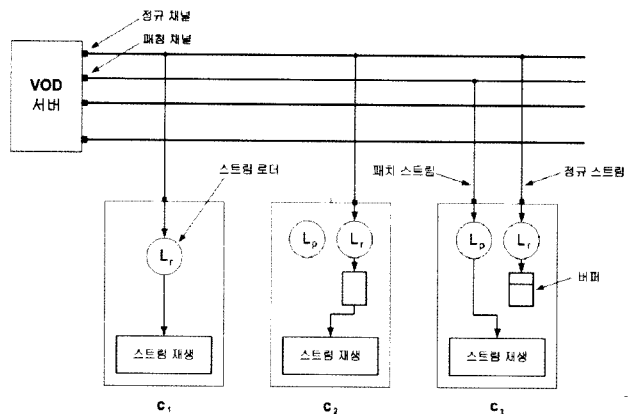
정 등에 대하여 상세하게 기술한다. 4장에서는 제안된 기법이 기존의 패칭 기법보다 성능 향상을 이루었음을 시뮬레이션 결과를 통해 보인다. 그리고 마지막으로 5장에서는 제안된 기법에 대한 결론을 기술한다.

2. 관련 연구

2.1 패칭 기법(Patching Scheme)

VOD 시스템은 다수의 클라이언트가 동시에 서비스를 요청하므로 VOD 서버의 적은 자원(resource)으로 보다 많은 클라이언트들에게 높은 QoS를 제공할 수 있어야 한다. 인기 있는 비디오의 경우 특히 요청하는 클라이언트들이 많은 것에 착안하여 이를 가장 효율적으로 이용할 수 있는 방법이 멀티캐스트(multicast)이다. 기존의 유니캐스트(unicast) 방법을 사용하면 요청하는 클라이언트마다 서버와의 연결을 설정하여 스트림을 전송받아야 하므로 클라이언트 수에 따라 그만큼 채널이 생성된다. 따라서 멀티캐스트 전송 방법은 한 번의 스트림 전송으로 다수의 클라이언트가 공유 가능하므로 유니캐스트 방법에 비해 네트워크 자원을 효율적으로 사용할 수 있다. 멀티캐스트를 통한 미디어의 전송은 다수의 클라이언트에게 연속 미디어를 제공하는 VOD 서버의 부하를 줄이기 위한 효과적인 방법이다. 그러나 클라이언트의 요청에 따른 미디어 데이터의 전송을 위해 매번 서버와 연결할 필요가 없으나, 멀티캐스트를 이루기 위한 시간 지연이 존재하므로 대기 시간 없는 서비스를 위해 멀티캐스트를 활용한 패칭 방법이 제안되었다[8].

패칭 방법은 멀티캐스트 되는 정규 채널과 각 클라이언트 요청의 시간적인 차이를 클라이언트의 버퍼를 통하여 보완하는 방법이다[1, 10, 11]. (그림 1)은 패칭의 기본 개념을 나타낸다. 서버는 큐에 도착하는 모든 요청을 저장하였다가 채널이 사용가능하게 되었을 때 요청들을 검사하여 다음 요청을 처리한다. 우선, 클라이언트 C_1 의 요청이 VOD 서버로 들어오면 정규 채널이 생성되어 클라이언트 C_1 의 버퍼에 멀티캐스트 스트림이 전송되면서 스트림 로더에 의해 비디오 스트림이 재생된다. 이때 클라이언트 C_2 가 같은 비디오 스트림을 요청하였으므로 C_1 을 위해 생성된 정규 채널을 통해 전송되어지는 멀티캐스트 정규 스트림을 C_2 의 로더에 의해 전송받아 재생해 본다. 또한 일정 시간이 지난 후 C_2 이후로 클라이언트 C_3 이 동일 비디오를 요청해 오면 이미 생성된 정규 채널을 통해 정규 스트림을 전송받아 버퍼에 저장하면서, 역시 지나간 프리픽스의 패치 스트림을 별도의 채널인 패칭 채널을 통해 전송받아 재생한다. 따라서 이러한 패칭 방법을 이용한다면 다수의 클라이언트 요청이 들어올 때 초기 지연시간 없이 바로 프리픽스 스트림만을 전송받고 이전에 생성된 정규 채널을 통해 멀티캐스트 정규 스트림을 공유하므로 서버의 부하를 크게 줄일 수 있다[4]. 이때 연속된 클라이언트 요청을 하나의 그룹으로 묶는 일정 시간 구간을 패칭 윈도우(patching window)라 한다[8].



(그림 1) 패칭의 기본 개념

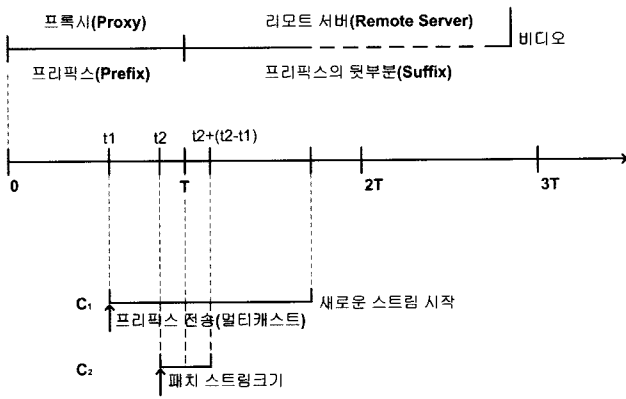
2.2 프록시 캐싱(Proxy Caching)

최근에는 기존의 연구와 더불어 클라이언트 가까이에서 VOD 서버의 역할을 대신하는 프록시(proxy) 서버를 설치하여 VOD 서버로의 네트워크 대역폭 요구를 줄여 VOD 서비스의 성능을 개선하고자 하는 연구도 꾸준히 진행되고 있다[19]. 프록시는 LAN 환경에서 VOD 서버의 미디어 데이터를 캐싱(caching)하여 클라이언트에게 바로 서비스하도록 하여 VOD 서버 역할의 일부를 대신함으로써 서버에 집중되는 부하의 분산 효과를 갖는다. 특히 프록시 서버는 클라이언트와 LAN으로 연결되어 있어 네트워크 지터나 전송 어려움에 민감하지 않다[24]. 프록시를 이용한 분산 시스템은 초기 지연시간(startup delays) 감소, 대역폭(bandwidth) 증대, 유연성(flexibility)의 특성을 가지므로 서버-프록시와 프록시-클라이언트 간에 효율적인 서비스가 이루어지게 한다.

위와 같은 장점으로 인하여 Proxy Prefix Caching 기법과 같이 프록시 서버에 비디오 파일의 프리픽스 부분을 저장해 두어 초기 지연시간을 줄이는 기법이 제안되었다[17][27]. 또한 Proxy Prefix Caching 기법의 최적화를 통하여 네트워크의 대역폭 사용 비용을 최소화 하는 기법에 대한 연구도 진행되었다[21, 23]. 최근에는 프록시 서버를 캐싱 에이전트(caching agent)로 활용하여 하나의 프록시가 다른 프록시 서버에게 미디어 스트림의 일부 또는 전체를 전송하여 서버의 역할을 대신하는 기법도 제안되었다[7].

VOD 서비스의 QoS는 비디오의 재생 지연이나 지터와 같은 전송 패킷의 손실 비율에 큰 영향을 받는다. 이러한 문제를 극복하여 QoS를 보장하기 위한 프리픽스 캐싱 기법이 제안되었다. 이 기법은 네트워크상에서의 전송률을 최소화하고 클라이언트 레벨의 QoS를 유지하기 위해 멀티캐스팅 스케줄을 서버와 프록시 사이에 적용하고 서버와 클라이언트 사이의 프록시에 프리픽스 혹은 데이터의 일부분 스트림(partial stream)을 캐싱하여 서비스하도록 한다[21].

(그림 2)는 프리픽스 캐싱과 패칭의 개념을 나타낸다. 클라이언트 C_1 은 t_1 시간에 도착하여 비디오를 요청하여 스트림을 재생하여야 하는데 원거리의 서버로부터 비디오의 프리픽스 스트림을 요청할 필요 없이 가까운 프록시에 프리픽스



(그림 2) 프록시 프리픽스 패칭

스가 캐싱되어 있으므로 프리픽스 스트림을 전송받아 즉시 재생한다. 이후 동일한 비디오를 요청한 클라이언트 C_2 는 이미 전송받고 있는 클라이언트 C_1 과 함께 t_2 시간부터 t_2-t_1 만큼 전송받는 동시에 그 크기만큼 프록시로부터 프리픽스를 전송받음으로써 스트림 전송에 최대 두 개의 채널만을 이용한다.

2.3 P2P 스트리밍(Peer to Peer Streaming)

P2P 환경 하에서 데이터 공유 프로그램을 통해 별도의 서버를 구축하지 않고도 파일 공유 등의 서비스뿐만 아니라 인터넷상에서 미디어 스트리밍 서비스를 받을 수 있다[9][14, 16]. 특히 P2P 미디어 스트리밍 시스템은 향후, 개인컴퓨터의 고성능화와 고속 네트워크의 보편화 그리고 다양한 멀티미디어 콘텐츠의 발달에 따라 그 활용이 더욱 커질 것이다.

그러나 P2P 네트워크를 구성하는 각 클라이언트는 서로 다른 네트워크 대역폭, 업로드 속도, 컴퓨터 성능을 가지고 VOD 서버에 접속하므로 서버에 접속 횟수가 증가할수록 QoS를 만족하지 못할 가능성이 높다. 또한 P2P 네트워크 특성상 클라이언트는 네트워크 이탈이 잦으며 연속성을 갖는 미디어 데이터의 속성상 이탈에 따른 스트림 전송 실패 시 서버에 접속하여 전송이 계속되어야 한다. 다른 클라이언트로부터 스트림을 전송받음으로써 서버와의 접속을 피하고 아울러 클라이언트들에게 초기 서비스 응답 시간을 줄일 수 있는 P2P 미디어 스트리밍 서비스가 이루어져야 한다. 따라서 한정된 대역폭에 의한 품질저하를 최소화하고 빠른 서비스를 위한 연구가 필요하다.

이러한 문제를 해결하기 위하여 지금까지 이루어진 P2P 미디어 스트리밍 기법과 관련된 연구는 다음과 같다. 먼저 Chaining 기법은 다수의 클라이언트를 선형 구조로 연결한 후 버퍼링을 통하여 다른 클라이언트에게 재전송하는 방식이다[18]. 또한 ZIGZAG 기법과 같이 전체 시스템의 클라이언트들을 멀티캐스트 트리 구조로 구성하여 클라이언트들을 통한 미디어의 전송을 수행한다[20]. 그러나 이러한 기법은 트리구조를 유지하기 위해 매번 클라이언트의 상태 변화에 대한 메시지를 교환해야하는 오버헤드가 따른다. 또한 미디어

어 스트림이 여러 클라이언트를 거쳐 전송되므로 스트림 전송에 대한 지연이 존재한다. 이와 함께 네트워크 대역폭 소모를 줄이기 위해 기존의 패칭 기법을 P2P 미디어 스트리밍에 적용하고자 한 P2CAST 기법도 제안되었다[5]. 여기에서 클라이언트들은 P2P 형식의 유니캐스트로 연결되어 있으며 패칭 기법에 의해 클라이언트들이 스트림을 공유하도록 하는 기법이다. 그러나 이러한 기법 역시 VOD 서버가 정규 채널을 주기적으로 생성하므로 정규 스트림 생성에 대한 부하가 요구된다. 최근의 연구로 스트리밍 전송을 위해 클라이언트들을 멀티캐스트 트리로 구성하고, 빠르고 효율적인 복구를 위해 가까이 위치한 클라이언트가 대신 전송하도록 하는 지역성을 이용하고 버퍼에 저장된 스트림을 다음 클라이언트에게 연속적으로 전송하는 P2P 방식의 P2VOD 기법이 제안되었다[3].

3. P2P 프록시 패칭(P2P Proxy Patching) 기법

본 논문에서는 주어진 VOD 서버 대역폭의 한계를 극복하여 대규모 VOD 시스템의 실현을 위해 P2P 환경 하에서 패칭 기법을 적용하는 방법에 대하여 기술한다. 즉, 다수의 클라이언트들에게 빠르고 나은 QoS를 제공하기 위해 프록시 프리픽스 패칭을 이용한 향상된 패칭 방법인 P2P 프록시 패칭 기법을 제안한다. 제안된 기법은 P2P 미디어 스트리밍 환경에서 클라이언트가 다른 클라이언트에게 재전송하기 위해 재생한 스트림을 계속 버퍼링하여 저장하며 여기에 프록시 프리픽스 패칭 기법을 적용함으로써 서버의 대역폭을 줄인다. 여기에서 클라이언트들은 서버와 클라이언트의 기능을 동시에 갖도록 하고 가까이 있는 클라이언트들이 서버의 기능을 함께 수행하도록 함으로써 클라이언트들 간의 미디어 스트림 데이터의 공유가 빠르고 안정성 있게 유지된다.

본 장에서는 제안된 P2P 프록시 패칭 방법에 대한 기본 구조, 프록시 서버의 역할, 새로운 클라이언트의 참여 방법, 클라이언트 버퍼의 버퍼링을 통한 재전송 과정, 그리고 클라이언트의 불규칙한 행동에 의한 이탈에 따른 복구 방법 등 클라이언트의 동작관리에 대하여 상세하게 기술한다.

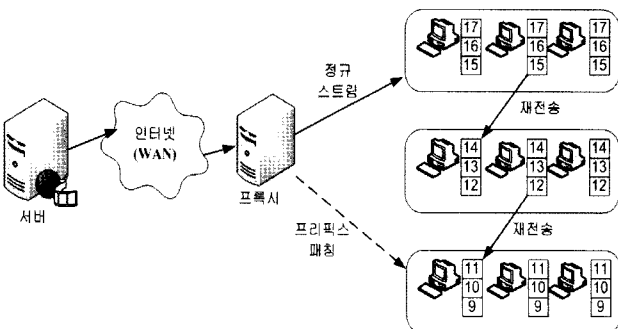
3.1 프록시 서버를 이용한 P2P 패칭 기법

P2P 프록시 패칭 기법이란 P2P 방식을 이용하여 프록시 패칭의 효과를 향상시킴으로써 VOD 서버와의 대역폭 요구량을 줄이기 위한 방법이다. 제안 기법의 시스템 구성은 (그림 3)과 같다. VOD 서버와 프록시는 인터넷이나 WAN을 통한 네트워크로 연결되어 있으며, 비디오 스트림을 요청하는 클라이언트들은 자신과 거리상 가까운 프록시 서버와 LAN으로 연결되어 있다. 또한 프록시 서버에는 인기도가 높은 비디오 파일의 시작부분인 프리픽스(prefix) 스트림을 저장하여 바로 서비스를 수행하며, 이 크기에 의해 클라이언트들이 그룹으로 묶여지며 패칭을 수행하기 위한 패칭 윈도우 크기가 된다. VOD 서버는 프록시에 없는 뒷부분(suffix) 스트림을 제공하고 모든 요청을 처리하며, 그리고 자신과

가장 가까운 프록시 서버에 연결되어 스트림을 재생하고 버퍼링을 수행하는 다수의 클라이언트가 있다. 일반적으로 프록시 프리픽스 패칭에서 클라이언트들이 자신이 보고자 하는 비디오를 VOD 서버에 요청하면 VOD 서버는 패칭 윈도우 우안의 클라이언트들을 위해 패칭 윈도우 크기마다 새로운 정규 채널을 생성하여야 하며, 요청 빈도가 높은 비디오의 경우는 주기적인 정규 채널의 생성으로 많은 부하가 요구된다. 따라서 본 논문에서 제안한 방법은 프리픽스 패칭의 효과를 극대화하고, VOD 서버의 빈번한 정규 채널 생성으로 인한 대역폭을 줄이기 위해 클라이언트 버퍼에 멀티캐스트되는 정규 스트림을 버퍼링하였다가 패칭 윈도우 크기 이후에 요청하는 다음 그룹의 클라이언트에게 스트림을 재전송하는 방법이다.

기존의 그레이스 패칭 방법[1]을 적용하는 VOD 시스템에서 패칭 윈도우 크기는 클라이언트의 버퍼 크기와 일치하고, 프록시로부터 전송된 프리픽스 스트림이 재생되는 동안 이 스트림과 근접한 정규 멀티캐스트 스트림이 클라이언트의 버퍼에 저장된다. 본 논문은 이점에 착안하여 패칭 윈도우 크기를 벗어나는 새로운 요청이 들어온 경우 VOD 서버로부터 새로운 정규 채널을 생성하는 대신 이전의 패칭 윈도우에 속하는 클라이언트 버퍼에 버퍼링된 정규 스트림의 비디오 데이터를 재전송 받아 사용하도록 한다.

즉, 하나의 프록시 서버에 연결된 다수의 클라이언트들이 동일한 비디오를 요청할 때, 각 요청의 시간적인 순서에 의하여 패칭 윈도우 단위로 클라이언트 그룹이 형성되며 이러한 그룹은 연속적으로 생성될 수 있다. 이렇게 패칭 그룹이 연속적으로 생성되는 경우 기존의 패칭 방법은 패칭 윈도우 크기마다 정규 채널을 생성하는 반면, 제안 방법은 클라이언트 버퍼에 버퍼링된 정규 스트림의 지연 전송을 이용하므로 연속적인 정규 채널을 생성하지 않는다. 연속적인 패칭 그룹이 지속되는 동안 정규 스트림에 대해 매번 VOD 서버로부터의 정규 채널을 생성하지 않고 이전의 패칭 그룹에 속한 클라이언트 버퍼에 버퍼링된 스트림을 전송받음으로써 정규 채널의 생성으로 인한 서버의 부하를 줄일 수 있다. 다만 안정적인 서버가 아닌 불규칙한 클라이언트로부터 스트림을 제공받는 것이므로 클라이언트의 이탈에 따른 복구 방법과 동적으로 움직이는 다수의 클라이언트들에 대한 정보 관리가 필요하다.



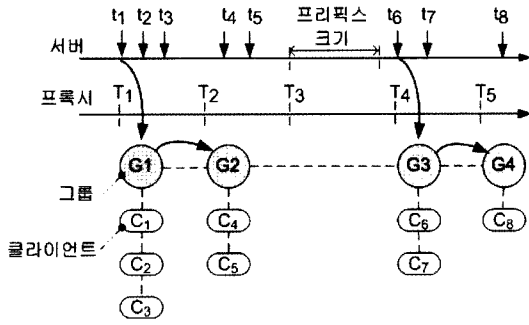
(그림 3) P2P 프록시 패칭 기법의 시스템 구성

3.2 프록시 서버의 역할

일반적으로 프록시 서버를 활용하는 VOD 시스템의 경우 클라이언트에게 전송되는 스트림은 프록시 서버를 경유하여 전송된다. 즉, 프록시 서버는 클라이언트와의 세션 정보를 유지하고 클라이언트들의 상태 변화를 갱신하는 역할을 수행한다. 따라서 불안정한 클라이언트 시스템을 정규 스트림을 전송하기 위한 하나의 프록시 서버의 역할을 수행하도록 하는 본 논문의 제안 방법을 위해 프록시 서버에 유지되는 인덱스 정보를 이용한다. 인덱스는 클라이언트의 불규칙적인 행동으로 인한 대체 및 선택과정을 수행하기 위한 정보이다. 따라서 프록시 서버는 프리픽스의 패치 스트림을 제공하는 역할과 더불어 인덱스 서버로서의 역할도 함께 수행한다. 이러한 인덱스는 기존의 프록시 서버가 스트림 전송을 위하여 유지하는 각 클라이언트의 연결 세션 정보를 패칭 윈도우와 그에 속한 클라이언트 리스트로 재구성한 것이다.

(그림 4)는 자신과 연결된 모든 클라이언트들에 대한 세션 정보를 유지·관리하며 전송중인 클라이언트가 이탈한 경우 이를 대체하여 새로운 클라이언트로 선택하기 위한 프록시상의 인덱스 구조를 나타낸 것이다. 즉, 시간의 변화에 따른 클라이언트들의 요청과 프록시 서버에 형성되는 인덱스를 나타낸다. 인덱스는 IP 주소, 호스트 이름 등 재생을 위해 동기화되는 클라이언트의 정보와 그룹의 상태 정보가 된다. 또한 여기에서 프리픽스 크기를 단위로 생성되는 그룹들은 그 순서에 따라 리스트를 형성하고, 각 그룹의 패칭 윈도우 크기는 프리픽스 크기와 동일하며 그룹에 속한 클라이언트 리스트로 이루어진다. 그룹에 대한 상태 정보는 그룹의 시작 시간과 더불어 최소한 하나 이상의 클라이언트 인덱스를 포함하며 클라이언트 인덱스는 현재 재생중인 클라이언트를 가리킨다. 또한 각 그룹에 대한 정규 스트림이 서버로부터 전송되는 것인지 이전에 형성된 클라이언트 그룹에 의하여 전송된 것인지에 대한 정보도 포함한다.

(그림 4)에서 각 클라이언트 C_1, C_2, \dots, C_8 은 같은 비디오 스트림을 보기위해 각각 t_1, t_2, \dots, t_8 시간에 요청한다고 가정한다. 또한 각 클라이언트는 프리픽스 패칭의 수행에 필요한 동일한 버퍼 공간을 가지고 있다. 그리고 클라이언트 요청이 프리픽스 크기를 초과할 때 각 패칭 그룹 G_1, G_2, G_3, G_4 는 새로이 생성되고, 각 그룹은 이전의 패칭 그룹이나 VOD 서버로부터 그들의 정규 멀티캐스트 스트림을 전송받는다. 처음 클라이언트 C_1 으로부터 t_1 시간에 요청이 도착하면 하나의 패칭 그룹 G_1 이 생성되고, 클라이언트 C_1 은 VOD 서버로부터 정규 채널을 통해 정규 스트림을 전송받기 시작한다. 이후 연속된 클라이언트 요청들이 t_1, t_2, t_3 시간에 도착하여 첫 번째 패칭 그룹 G_1 안에 포함된다. 그룹 G_1 안의 C_1, C_2, C_3 클라이언트들은 VOD 서버로부터의 정규 채널을 통해 멀티캐스트 스트림을 전송받아 버퍼에 저장하면서 패칭 채널을 통해 프록시 서버로부터 프리픽스 스트림을 전송받아 재생한다. 패칭 윈도우 크기를 벗어나 도착한 C_4 와 C_5 는 새로운 패칭 그룹 G_2 에 포함되어 프록시



(그림 4) P2P 프록시 패칭 기법의 인덱스 구성

서버로부터 새로운 패칭 채널을 통해 프리픽스 스트림을 전송받는다. 그리고 정규 멀티캐스트 스트림은 이전 그룹 G1 안의 클라이언트로부터 스트림을 전송받게 된다. 이 경우 프록시 서버는 그룹 G2에게 정규 스트림을 전송하기 위해 그룹 G1안의 하나의 클라이언트를 선택하고 인덱스를 갱신한다. VOD 서버는 그룹 G2에게 정규 스트림을 전송하기 위해 추가적인 채널을 생성할 필요가 없다. 패칭 그룹 G3은 VOD 서버로부터 정규 멀티캐스트 스트림을 전송받아야 한다. 마지막으로 패칭 그룹 G4는 그룹 G3과의 시간 간격이 프리픽스 크기보다 작으므로 VOD 서버와 새로운 정규 채널을 생성하지 않고 그룹 G3안의 클라이언트로부터 버퍼링된 정규 스트림을 재전송 받을 수 있음을 보여준다.

이와 같이 제안된 방법에서 프록시 서버는 클라이언트들과 그룹의 상태 정보에 대한 인덱스를 유지한다. 프록시 서버안의 인덱스는 그룹 생성 시간, 각 클라이언트의 요청 시간, 비디오 스트림을 재생하고 있는 클라이언트들의 정보 및 정규 멀티캐스트 스트림의 전송 위치 등의 정보를 포함한다. 따라서 각 클라이언트가 자신이 원하는 요청을 프록시 서버에 전달하면 프록시 서버는 클라이언트의 요청을 수행하는 동시에 프록시 서버에 유지되는 인덱스를 갱신한다.

그러므로 시스템의 구조를 유지하기 위한 메시지의 교환이 단 한번 이루어지며, ZIGZAG 방법[20]과 같은 트리 구조 상의 클라이언트들이 여러 차례 메시지를 교환하여야 하는 오버헤드를 줄일 수 있다. 또 멀티캐스트 트리를 구성하는 P2P 시스템에서는 임의의 클라이언트의 이탈에 따른 스트림 복구를 위해 다수의 클라이언트들 간에 메시지 교환에 드는 오버헤드와 그동안에 스트림 전송에 대한 지연이 발생한다. 그에 비하여 제안한 방법은 네트워크 거리가 가까운 프록시에서 관리하고 한 번의 메시지 교환으로 스트림 복구를 위한 지연이 매우 짧다. 또한 이렇게 프록시 서버에 유지되어야 하는 인덱스 정보는 기존의 패칭 방법에서 사용된 클라이언트와의 세션 정보를 제안된 방법에 맞게 선형 구조로 재구성한 것이므로 인덱스 정보를 유지·갱신하기 위한 프록시 서버의 오버헤드 또한 매우 적다.

3.3 클라이언트들의 동작 관리

본 논문에서의 프리픽스는 프록시 서버로부터 패치되며 이것은 클라이언트 버퍼에 버퍼링 되는 정규 스트림의 크기

와 같다. 그리고 하나의 그룹 안에 포함된 클라이언트들의 버퍼 내용은 서로 동기화를 이룬다. 이렇게 함으로써 패칭 그룹 안의 클라이언트가 이탈하면 이 클라이언트의 역할을 위해 그룹 안의 다른 클라이언트로 대체될 수 있도록 한다. 지연 전송을 수행하는 한 클라이언트가 떠났을지라도 프록시 서버는 이 클라이언트의 인덱스 정보를 갱신하고 비디오 스트림의 전송을 위해 다른 클라이언트로 대체하여 전송하도록 하거나 VOD 서버로부터 스트림이 계속 전송되도록 처리한다. 그러므로 동적으로 변화하는 클라이언트들의 세션 정보와 그룹의 상태 정보를 동기화시키기 위해 프록시 서버에 인덱스를 유지한다. 다음에서 클라이언트들의 동적인 행동에 따른 처리 과정을 설명한다.

3.3.1 새로운 클라이언트의 참여

기존의 패칭 방법에서 정의되는 패칭 윈도우 크기는 클라이언트의 버퍼 크기에 의해 결정된다. 제한된 P2P 프록시 패칭 방법에서 프록시 서버로부터 패칭되는 프리픽스 크기와 클라이언트 버퍼에 버퍼링되는 정규 스트림의 크기는 동일하다. 즉, 하나의 패칭 윈도우안의 클라이언트 그룹에 의해 버퍼링 되고 재전송되는 정규 스트림의 크기는 프리픽스 크기와 동일하다. 예를 들어, 새로운 클라이언트가 비디오를 재생해 보기위해 요청하면 프록시 서버는 우선 클라이언트의 연결 세션을 처리한다. 새로운 클라이언트의 요청에 대해 가장 최근에 생성된 그룹의 시작 시간과 현재 클라이언트의 요청 시간을 비교하여 새로운 정규 채널을 사용하는 새로운 그룹을 생성할 것인지, 혹은 최근의 그룹으로부터 정규 스트림을 재전송 받는 그룹을 생성할 것인지, 아니면 기존의 그룹에 참여할 것인지 결정하여야 한다. 즉, 최근 그룹의 시작 시간과 클라이언트의 요구 시간차가 프리픽스 크기보다 작다면 기존 그룹에 참여시키는 과정을 수행하고, 반대로 최근 그룹의 시작 시간과 현재 요청 시간의 차이가 프리픽스 크기보다 크다면 새로운 패칭 그룹을 생성하여 새로운 클라이언트를 선두로 시작한다. 또한 새로운 그룹을 생성하여 시작하는 경우에도 이전 그룹의 종료 시간과 현재 요청 시간을 비교하여 이전 그룹의 클라이언트로부터 정규 스트림을 전송받도록 할 것인지 또는 VOD 서버로부터 새로운 정규 채널을 요구할 것인지 결정한다. 이렇게 결정되는 패칭 그룹과 새로운 클라이언트의 관계는 프록시 서버 상에 유지되는 인덱스 정보와 일치한다.

앞에서 설명한 (그림 4)의 인덱스 구조의 예를 보자. 동일 비디오에 대해 클라이언트들의 연속적인 요청이 도착한다고 가정하였다. 처음 시간 t_1 에 클라이언트 C_1 의 요청이 들어왔을 때 이전 그룹이 존재하지 않으므로, VOD 서버에 새로운 정규 채널을 요청한 후 새 패칭 그룹 G1을 생성한다. 계속하여 VOD 서버로부터 스트림이 멀티캐스트 되는 동안 클라이언트 C_2 와 C_3 의 요청이 도착하면, 클라이언트들의 도착 시간 t_2, t_3 과 그룹 G1의 시작 시간 t_1 과의 차이가 프리픽스 크기보다 작으므로 프록시는 그룹 G1에 클라이언트 C_2, C_3 을 참여시킨다. 또한 패칭을 위하여 그룹의 시작 시간 T_1

과 각 클라이언트의 요청 시간과의 차이 만큼에 해당하는 프리픽스를 각 클라이언트에 전송한다. 그룹 G_1 의 시작 시간 T_1 로부터 프리픽스 크기만큼의 시간이 흐른 T_2 가 되면 G_1 그룹에 대한 추가 작업을 종료한다.

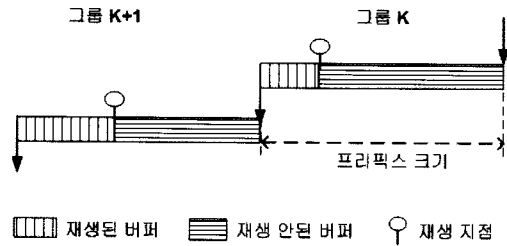
위와 같이 T_2 의 시점이 지나면 그룹 G_2 가 생성되고 새로운 클라이언트 C_4 와 C_5 는 그룹 G_2 안에 참여한다. 클라이언트 C_4 와 C_5 는 이전 그룹 G_1 의 클라이언트로부터 정규 스트림을 재전송 받아 사용할 수 있다. 왜냐하면 그룹 G_2 는 그룹 G_1 이 종료된 시간 T_2 로부터 프리픽스 범위이내의 시간에 이루어졌기 때문이다. 이때 프록시는 C_4 에게 T_2 에서 t_4 의 분량에 해당하는 프리픽스를 전송한다. 또한 그룹 G_2 는 그룹 G_1 에 버퍼링 된 스트림을 받아 사용하는 것이므로 미디어 스트림의 연속성을 위하여 이전 그룹 G_1 의 종료시간인 T_2 가 그룹 G_2 의 시작 시간이 된다. 즉, G_1 과 G_2 의 클라이언트 버퍼에 버퍼링되는 스트림이 패칭 윈도우 크기를 단위로 연속성을 가지며, 이것은 연속된 패칭 그룹에서 초기에 필요한 하나의 정규 채널만을 사용하여 VOD 서비스가 가능하도록 한다. 그룹 G_2 는 시작 시간 T_2 로부터 프리픽스 크기의 시간이 지난 T_3 에 종료된다.

반대로 이전 그룹이 이미 생성되어 있는 경우라 하더라도 이전 그룹의 종료 시간과 새로운 요청이 도착한 시간과의 차이가 프리픽스 크기보다 큰 경우 프록시로부터 패칭을 수행할 수 없으므로 이전 그룹으로부터 정규 스트림을 재전송 받을 수 없다. (그림 5)에 나타난 바와 같이 새로운 클라이언트 C_6 과 C_7 은 새 그룹 G_3 안에 포함되고 VOD 서버로부터 새롭게 정규 채널을 할당 받아 사용한다. 이때 그룹 G_3 의 시작 시간 T_4 는 클라이언트 C_6 의 요청이 이루어진 시간 t_6 이며 이전의 그룹과는 단절된 상태이다. 그룹 G_3 안의 클라이언트 C_6 과 C_7 은 이전 그룹 G_2 로부터 정규 멀티캐스트 스트림을 전송받지 못한다. 왜냐하면 이전 패칭 그룹 G_2 안의 마지막 요청 클라이언트 C_5 와 클라이언트 C_6 과의 요청 시간차가 패칭 윈도우인 프리픽스 크기보다 커 재전송할 수 있는 범위를 벗어나기 때문이다. 또 새로운 클라이언트 C_8 은 t_8 시간에 그룹 G_4 안에 참여하게 되고 그룹 G_3 으로부터 정규 스트림을 재전송 받는 경우를 보여준다.

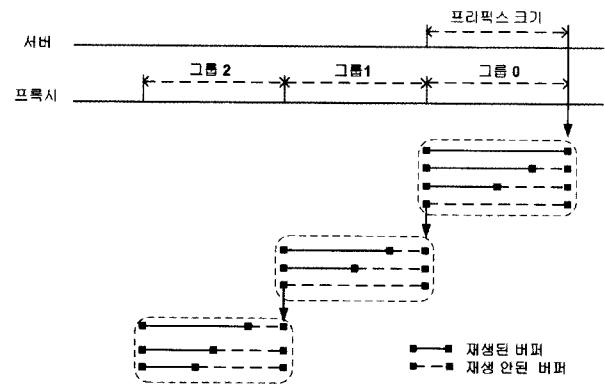
위와 같은 과정을 거쳐 새로운 클라이언트들의 참여로 그룹이 생성되면 각 그룹은 클라이언트들 중 하나를 선택하여 이후의 그룹에 대한 프록시의 역할로 활용한다. 이때 각 클라이언트 시스템은 정규 스트림을 저장하기 위해 버퍼링 과정을 수행하며 그에 대한 내용은 다음과 같다.

3.3.2 클라이언트의 버퍼링과 재전송 과정

제한된 패칭의 성능 향상을 위한 P2P 프록시 패칭은 스트림의 지연 전송을 이용하여 정규 스트림을 재전송하는 방법이다. 프록시를 이용한 프리픽스 패칭 시스템 상에서 클라이언트 버퍼에 정규 스트림을 버퍼링하여 이후의 그룹에 지연 전송하므로, 클라이언트는 자신의 패칭 그룹에 속한 클라이언트들과 정규 스트림을 재전송하기 위해 패칭 윈도



(그림 5) 클라이언트 그룹의 버퍼 구조

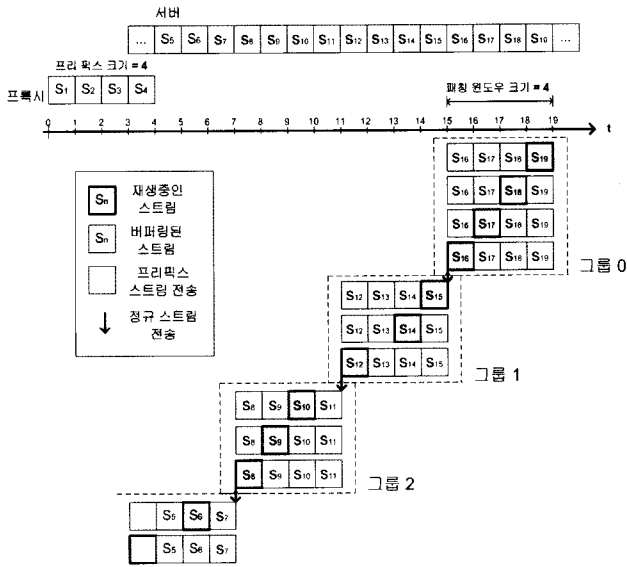


(그림 6) 패칭 그룹에 의한 정규 스트림 재전송

우 크기에 해당하는 분량의 스트림을 버퍼링 한다. 기존의 패칭 방법에서 재생을 위하여 정규 스트림의 시작 시간과 요청 시간과의 차이에 해당하는 분량만을 버퍼링하는 것에 비하여 제안된 방법은 재생을 위한 버퍼링과 재전송을 위한 버퍼링을 함께 수행한다. 즉, 기존의 패칭 기법의 경우 정규 스트림과의 시간 차이만큼을 재생을 위하여 버퍼링하고 재생한 스트림은 소멸시켜버리지만, 제안된 방법에서는 재생된 스트림이라고 하더라도 재전송을 위하여 일정시간 버퍼에 남겨둔다. 따라서 각 패칭 그룹에 참여하는 모든 클라이언트는 패칭 윈도우 크기의 정규 스트림을 저장하였다가 그만큼의 지연을 가지고 재전송을 수행한다.

(그림 5)는 본 논문에서 제안된 방법을 위한 클라이언트의 버퍼 구조를 보여주고 있다. 그룹 K 와 $K+1$ 은 연속적인 선후관계를 가지는 패칭 그룹으로서 각각 자신의 그룹에 속한 클라이언트들 중 하나를 선택하여 이어지는 그룹에게 멀티캐스트를 통하여 정규 스트림을 전송하도록 하고 있다. 이때 각 클라이언트 버퍼는 스트림의 재생이 이루어지는 지점을 경계로 재전송을 위하여 대기 중인 버퍼와 재생을 위하여 대기 중인 버퍼 두 부분으로 나뉜다. 아직 재생이 이루어지지 않은 부분의 크기는 일반적인 패칭 기법에서 재생을 위하여 버퍼링을 수행하는 정규 스트림의 시작 시간과 요청 시간 사이의 차이 값과 일치한다.

(그림 6)은 이전의 패칭 그룹과 그에 속한 클라이언트에 의하여 연속되는 다음 그룹으로 정규 스트림이 재전송되는 과정을 보여준다. 여기서 하나의 그룹은 최소한 하나 이상의 클라이언트를 가지며, 클라이언트들 중 하나를 선택하여 이



(그림 7) 스트림과 버퍼의 현황도

어지는 다음 그룹에게 멀티캐스트를 통하여 정규 스트림을 재전송하는 프록시 서버의 역할을 수행한다. 또 같은 그룹에 속한 클라이언트라 하더라도 서로의 요청 시간에 따라 재생 지점이 다르고 전송을 위한 버퍼와 재생을 위한 버퍼 공간의 비율이 다르게 된다. (그림 7)에서 실선은 재전송을 수행하기 위해 스트림을 버퍼링하고 있는 부분이며, 점선은 재생을 위하여 저장된 스트림을 나타낸다. 그러나 두 버퍼 공간의 비율과 상관없이 같은 그룹에 속한 모든 클라이언트는 프리픽스 크기와 동일한 분량의 정규 스트림을 버퍼링하고 있다. 따라서 정규 스트림을 재전송하기 위해 프록시의 역할을 수행하는 클라이언트가 예기치 못한 행동으로 인하여 그룹에서 이탈한 경우 그룹 내의 다른 클라이언트가 동일한 스트림을 버퍼링하고 있으므로 다른 클라이언트를 선택하여 프록시 서버의 역할을 대신 수행하게 할 수 있다.

(그림 7)은 재전송을 수행하는 그룹 안에 포함된 클라이언트들의 실제 버퍼 내용을 나타낸다. 프리픽스 크기가 4이므로 패칭 윈도우 크기도 4이다. 현재 그룹 0의 첫 번째 클라이언트는 정규 스트림을 멀티캐스트로 전송하면서 스트림 S_{16} 를 재생하고 있으면서 이미 재생이 끝난 스트림도 자신의 버퍼에 버퍼링하고 있다. 두 번째로 들어온 클라이언트는 스트림 S_{18} 를 재생하고 있으면서 다음에 재생할 스트림인 S_{19} 와 이미 재생이 끝난 두 스트림을 버퍼링하고 있다. 또한 그룹안의 모든 클라이언트들의 버퍼에는 동일한 스트림 내용이 버퍼링되어 있다. 그리고 패칭 윈도우 크기 이후에 들어온 그룹 1의 첫 클라이언트는 그룹 0의 클라이언트로부터 정규 스트림을 멀티캐스트로 전송받아 재생과 버퍼링을 수행한다. 역시 그룹 1안의 클라이언트들도 동일한 스트림 내용을 자신의 버퍼에 저장하고 있다. 시간차를 두고 들어온 다음 그룹의 클라이언트들도 이전 그룹의 클라이언트로부터 스트림을 재전송 받으며 재생과 버퍼링을 수행하므로 VOD 서버로부터 스트림을 전송받을 필요가 없다. 또

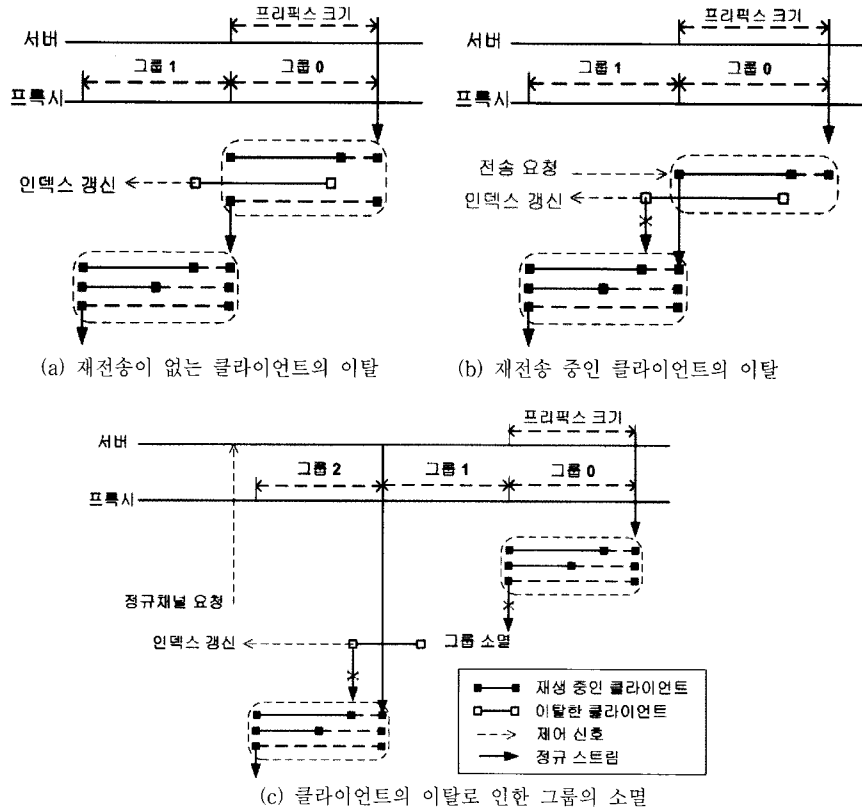
한 프리픽스 스트림 S_1, S_2, S_3, S_4 는 프록시에 있으므로 즉시 패칭이 이루어져 재생을 수행한다. 만일 이전 그룹으로부터 재전송을 받을 수 없을 만큼 시간차가 길게 들어올 경우는 서버로부터 스트림을 전송받고, 그 이후에 도착한 다음 그룹에게 재전송을 수행하게 된다. 이러한 모든 클라이언트에 대한 정보는 프록시의 인덱스에 저장되어 재전송 시 이용되고 클라이언트의 이탈에 따른 정보도 저장되어 정규 스트림 전송을 위한 복구 시에 이용될 수 있게 한다. 즉, 각 클라이언트의 행동에 따라 프록시 서버에 유지되는 인덱스 정보도 갱신된다. 이러한 인덱스 정보는 정규 스트림을 전송중인 클라이언트가 이탈한 경우 스트림을 복구하기 위한 대체 및 선택과정을 수행하는데 이용된다. 이와 관련된 내용은 다음절에서 상세하게 기술한다.

3.3.3 클라이언트 이탈에 따른 복구 과정

제안된 P2P 프록시 패칭 방법은 클라이언트의 버퍼를 정규 스트림 전송을 위한 프록시로 활용하므로 안정적인 서버와는 달리 클라이언트의 불규칙한 행동이나 재생이 종료되어 멀티캐스트 중인 클라이언트가 그룹에서 이탈하는 경우가 발생한다. 따라서 스트림을 전송중인 클라이언트가 그룹에서 이탈하는 경우 그룹 내의 다른 클라이언트를 선택하여 정규 스트림을 전송하도록 대체해야 한다. 또한 그룹 내의 모든 클라이언트가 그룹에서 이탈하여 재전송이 불가능하면 VOD 서버에 새로운 정규 채널을 요청하고 그룹을 소멸시켜야 한다.

클라이언트의 이탈에 따른 복구 과정은 프록시 서버에 유지되는 인덱스를 이용하여 이루어진다. 즉, 각 클라이언트는 재생 중에 그룹으로부터 이탈을 원하는 경우 프록시 서버에 자신이 그룹을 떠남을 알린다. 이때 프록시는 이탈을 원하는 클라이언트가 재전송을 수행하던 클라이언트라면 자신의 인덱스를 검색하여 동일 그룹 내의 클라이언트를 선택하여 스트림 전송을 대체한다. 그러나 동일한 그룹 안에 스트림 전송을 대체할 클라이언트가 존재하지 않는 경우 해당 그룹을 소멸시키고 VOD 서버로부터 서비스 중인 비디오의 현재 시점부터 새로운 정규 채널을 생성할 것을 요청한다.

(그림 8)은 클라이언트의 이탈에 따른 정규 스트림의 복구 과정을 나타낸다. (그림 8)(a)는 재전송을 수행하지 않던 클라이언트가 그룹에서 이탈을 원하는 경우이다. 이탈을 원하는 클라이언트가 프록시 서버에 이탈 신호를 보내면 프록시는 이에 대한 인덱스를 갱신하여 클라이언트의 상태와 인덱스의 내용을 동일하게 유지시킨다. 또 (그림 8)(b)는 재전송을 수행 중인 클라이언트가 이탈하는 경우, 그룹 내의 새로운 클라이언트를 선택하여 스트림 전송을 대체하는 과정을 보여준다. 이때 프록시 서버는 이탈을 요청한 클라이언트 신호에 따라 인덱스를 갱신하고, 동시에 스트림 전송을 대체할 클라이언트를 검색하여 스트림을 전송할 것을 요청한다. 마지막으로 (그림 8)(c)는 그룹 내의 모든 클라이언트가 이탈한 경우를 나타낸다. 이탈을 희망하는 클라이언트들이 프록시 서버에 이탈 신호를 보내면 프록시는 이탈에 따



(그림 8) 클라이언트의 이탈과 복구 과정

른 인덱스를 수정한다. 이때 해당하는 그룹의 클라이언트들 모두가 이탈하였으므로 스트림을 대체할 클라이언트를 선택할 수 없다. 따라서 프록시는 VOD 서버에게 비디오의 남은 분량에 대한 정규 스트림을 전송할 것을 요청한다.

이렇게 모든 클라이언트에 대한 상태 정보가 클라이언트와 가까이 있는 프록시내의 인덱스에 존재하므로 인덱스를 이용하여 복구 과정이 이루어지게 한다. 만약 클라이언트의 이탈이 빈번할 지라도 즉시 인덱스는 재구성되고 재구성된 인덱스 정보에 의해 스트림 재전송이 신속히 이루어질 수 있다. 따라서 클라이언트 이탈 시에도 클라이언트들의 연결 정보를 찾기 위해 여러 번 메시지를 교환할 필요 없이 복구 과정에 의해 스트림 재전송이 빠르게 수행될 수 있다.

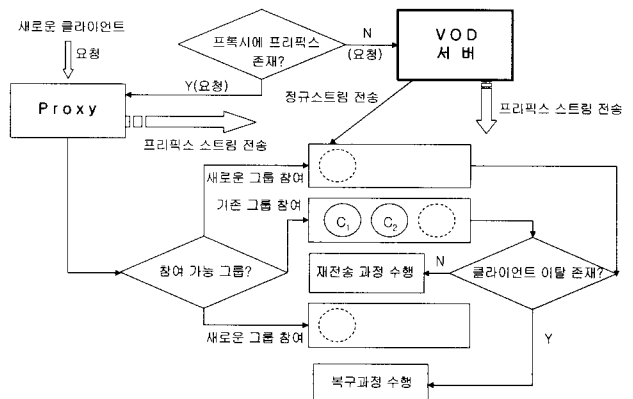
4. 성능평가

본 논문에서 제안한 패칭의 성능 향상을 위한 P2P 프록시 패칭 기법에 대한 성능 평가를 위해 시뮬레이션을 수행하였다. 시뮬레이션 모델 및 환경 설정은 Pentium IV PC의 Windows 환경에서 Microsoft Visual C++ 프로그램을 이용하였다.

4.1 시뮬레이션 모델

제안된 방법의 성능 평가를 위한 시뮬레이션 모델은 (그림 9)와 같다. 우선 VOD 서비스를 받기위한 클라이언트는

임의의 시간 간격으로 도착한다. <표 1>에 나타난 바와 같이 90분 분량의 비디오 파일 100개를 서비스하는 VOD 서버와 7개의 프록시 서버를 포함하는 네트워크 모델을 가정한다. 또한 각 프록시 서버는 인기도가 높은 상위 10%의 비디오의 프리픽스를 저장하여 전송한다. 또 프록시에 패칭 기법이 적용되며 패칭 윈도우 크기는 프리픽스 크기와 일치한다고 가정한다. VOD 서버에 대한 클라이언트의 요청은 λ 의 발생 빈도를 가지는 포아송 분포를 따른다고 가정할 때, VOD 서버의 평균 요청 간격은 $1/\lambda$ 로 10초를 기본 값으로 한다. 그리고 프록시 서버에 저장된 비디오 파일의 프리픽스 크기는 10분이다. 마지막으로, 전체 비디오에 대한 요청



(그림 9) 시뮬레이션 모델

〈표 1〉 성능 평가를 위한 파라미터

| Parameter | Default | Variation |
|----------------------------------|---------|-----------|
| 비디오 개수 | 100 | N/A |
| 비디오의 길이(minutes) | 90분 | N/A |
| 평균 요청 도착시간 $1/\lambda$ (seconds) | 10 | 5-100 |
| 프리픽스 크기(minutes) | 10분 | 0-20 |
| 비디오의 재생률 b (Mbps) | 1.5 | N/A |
| 편이 계수(skew factor) θ | 0.271 | 0.0-1.0 |
| 프록시 서버의 개수 | 7 | N/A |
| 시뮬레이션 시간(hours) | 10 | N/A |

패턴이 편이 계수(skew factor) θ 값을 갖는 Zipf-like 분포를 따른다고 가정할 때, 편이 계수의 기본 값은 $\theta=0.271$ 이라 한다. 편이 계수 값이 1일 때는 들어오는 요청이 균등(uniform) 분포를 갖고, 0일 때는 급격한 편이(heavy skew) 분포를 갖는다.

N 개의 비디오 중 비디오 i 에 대한 요청 확률은 $P_i = \frac{c}{i^{(1-\theta)}}$, $1 \leq i \leq N$ 로 계산된다. 이때 편이계수 θ 값과 함께 c 는 표준 상수(normalized factor)로 $c = \frac{1}{H_N^{(1-\theta)}}$, $H_N^{(1-\theta)} = \sum_{i=1}^N \frac{c}{i^{(1-\theta)}}$ 이며, 조화 수(harmonic number) H 의 역수이다[2].

편이 계수인 θ 값이 작을수록 특정 비디오에 대한 요청이 집중적으로 이루어진다. 새로운 클라이언트가 서비스를 요구하면 현재 서비스를 받고 있는 그룹 중에 현재 요청한 시간과 비교하여 조건에 해당하는 그룹에 포함이 되도록 하여 멀티캐스트 정규 스트림을 전송받을 수 있게 한다. 이때 비디오의 시작부분인 프리픽스를 프록시로부터 받거나 서버로부터 전송받아 재생을 한다. 그런데, 클라이언트 이탈에 의해 정규 스트림을 재전송 받을 수 없을 때에는 이탈에 따른 복구 과정을 수행하여 서비스가 계속되도록 한다.

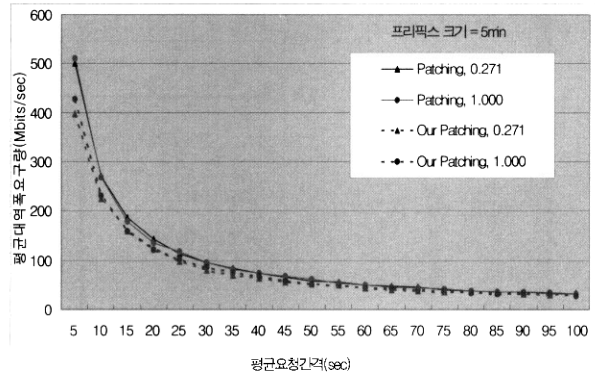
성능 평가는 위와 같은 조건으로 VOD 서버와 프록시에 적용한 기존의 패칭 방법과 P2P 방식을 함께 적용한 제안 방법에 대해 다음의 세 가지 항목으로 나누어 서버의 평균 대역폭 요구량을 비교한다. 서버의 대역폭 요구량은 10시간 분량의 요청이 들어왔을 때 VOD 서버에서 사용한 전체 대역폭 소모량을 시뮬레이션 수행시간에 대한 평균값으로 구하였다. 신뢰성을 위해 이 값은 20회 반복한 결과의 평균값을 최종 수치로 구하였다.

4.2 P2P 프록시 패칭의 성능 평가 결과

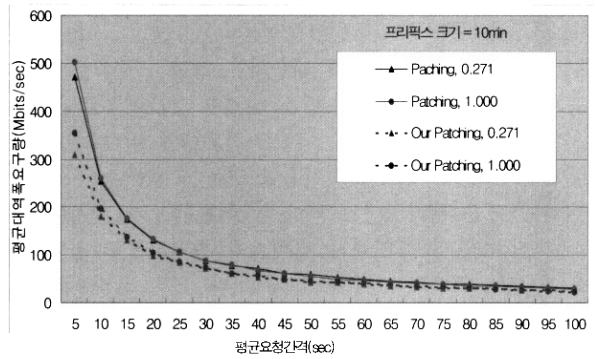
4.2.1 프리픽스 크기에 따른 성능 비교

프록시 서버에 저장된 프리픽스 크기 값에 따른 VOD 서버의 대역폭 요구량을 비교한다. 성능 평가를 위한 성능 모델에서 전체 100개의 비디오들 중에서 인기가 높은 10%의 비디오만이 프록시에서 서비스 된다고 가정하였고, 클라이언트들의 평균 요청 도착 간격은 10초이다. 또한 프록시에 저장된 프리픽스 크기는 패칭 윈도우의 크기와 일치한다.

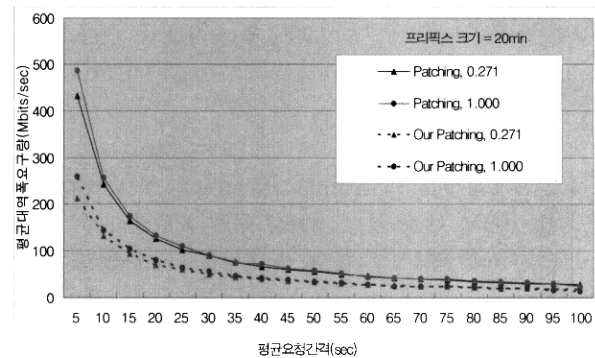
(그림 10)은 프리픽스 크기가 각각 5분, 10분, 20분일 때 평균 요청 간격에 따라 요구되는 평균 대역폭 요구량을 나타낸다. 그래프에서 실선은 기존의 패칭 방법을 적용했을 때의 결과이며, 점선은 본 논문에서 제안한 P2P 프록시 패칭 방법을 적용한 결과이다. 또한 각 방법에 대해 편이 계수 값이 0.271과 1.0일 때 각각의 평균 대역폭 요구량을 측정하였다. 프리픽스 크기를 증가시키에 따라 대역폭 요구량이 작아지는 것을 볼 수 있다. 이는 프리픽스 크기가 클수록 평균 대역폭 요구량이 작아짐을 알 수 있다. 평균 요청 간격이 15초일 때 프리픽스 크기는 10분이고 편이계수



(a) 프리픽스 크기가 5분일 때 평균 요청 간격에 따른 대역폭 요구량

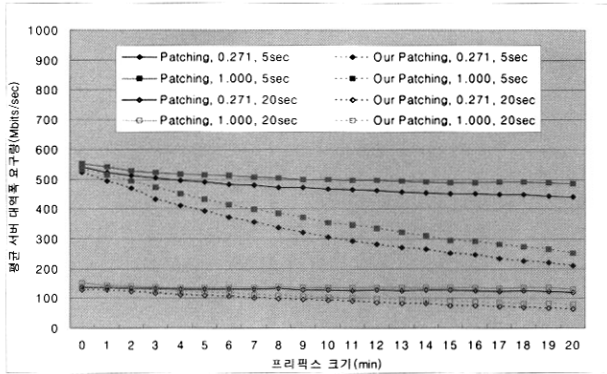


(b) 프리픽스 크기가 10분일 때 평균 요청 간격에 따른 대역폭 요구량

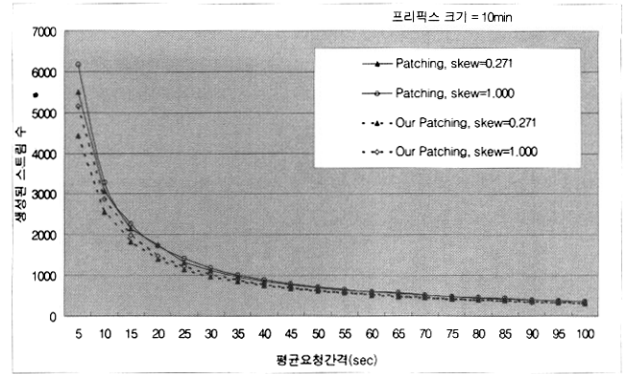


(c) 프리픽스 크기가 20분일 때 평균 요청 간격에 따른 대역폭 요구량

(그림 10) 프리픽스 크기와 요청 시간 간격 변화에 따른 대역폭 요구량



(그림 11) 프리픽스 크기 변화에 따른 대역폭 요구량



(그림 12) 평균 요청 시간 간격의 변화에 따른 생성 스트림의 수

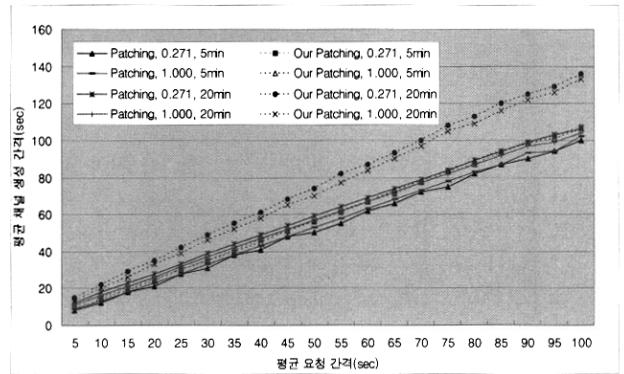
값이 0.271일 때 패칭만 적용한 경우와 제안한 기법을 비교하면 대역폭 요구량의 성능 이득이 약 25% 향상되었다. 이는 패칭만 적용하였을 때보다 제안된 기법은 이전 그룹의 클라이언트의 재전송에 의해 새로이 정규 채널을 생성하지 않으므로 대역폭 요구량이 증가하지 않는다. 같은 조건에서 프리픽스 크기가 20분으로 증가하였을 때 약 43%의 성능 이득을 얻을 수 있었다.

(그림 11)은 프리픽스 크기를 변화시키면서 평균 요청이 빈번할 때(5초)와 그렇지 않을 경우(20초)에 따른 대역폭 요구량을 두 가지 θ 값 0.271과 1.0에 대해 측정하였다. 그래프의 실선은 기존의 패칭에 대한 결과이며, 점선은 제안된 방법에 대한 결과이다. 그래프의 결과에서 평균 요청 간격이 빈번하지 않은 20초일 때 편이 계수와 프리픽스 크기에 따라 요구되는 대역폭 양이 제안된 방법에서는 현저하게 작게 요구되는 것을 볼 수 있다. 또한 빈번한 요청 시간 간격인 5초에 대해서도 제안된 방법은 기존의 패칭 방법에 비해 프리픽스가 증가함에 따라 대역폭 요구량이 크게 감소함을 볼 수 있다. 즉, 기존의 패칭 방법과 비교하여 볼 때 제안된 방법은 클라이언트를 정규 스트림의 재전송을 위해 일시적인 캐싱 서버로 사용하고 프록시로부터 프리픽스를 바로 전송받으므로 프리픽스 크기의 증가에 따라 평균 대역폭 요구량을 훨씬 적게 요구한다. 특히 프리픽스 크기가 10분일 때 편이계수 값이 0.271이고 5초 단위로 더욱 빈번하게 요청이 들어올 때 패칭만 적용한 경우 보다 제안 방법의 대역폭 요구량에 대한 성능 이득은 약 35% 크게 얻을 수 있었다.

4.2.2 평균 요청 간격에 따른 성능 비교

VOD 서버에 대한 클라이언트의 평균 요청 시간 간격 $\frac{1}{\lambda}$ 의 변화에 따른 서버의 대역폭 요구량을 측정하였다. 이때 인기도가 높은 상위 10%의 비디오에 대한 프록시 서버의 프리픽스 크기는 10분으로 하였다.

(그림 12)는 서비스를 위하여 VOD 서버에서 생성되는 전체 스트림의 수를 편이 계수 값 0.271과 1.0에 대해 프리픽스 크기 10분일 때 평균 요청시간 변화에 따라 측정하였다. 실선의 패칭만 적용한 방법보다 점선으로 표시된 제안된 P2P 프록시 패칭 방법에서 스트림의 수가 적게 생성됨



(그림 13) 요청 간격 시간의 변화에 따른 채널 생성 간격

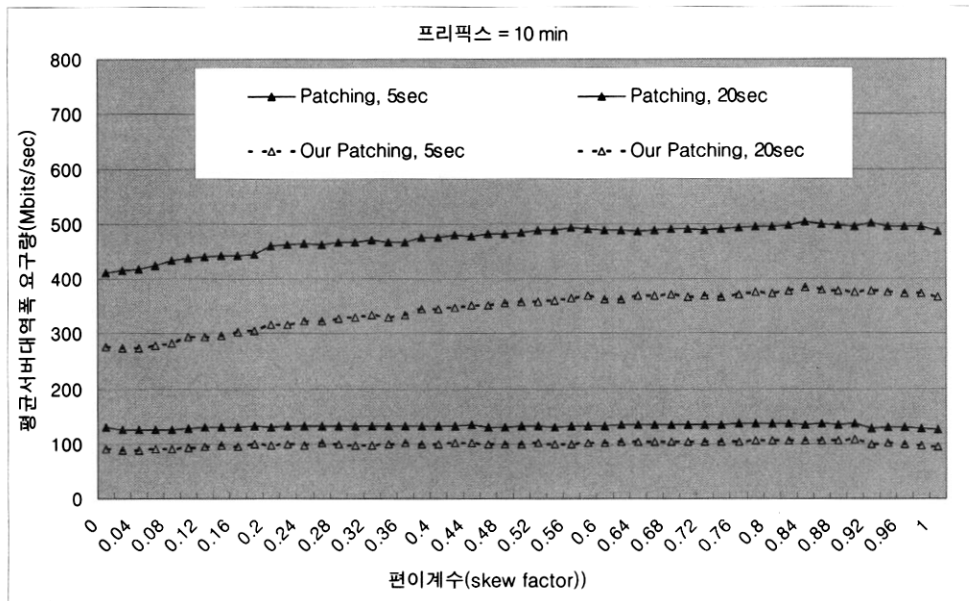
을 볼 수 있다. 특히 프리픽스 크기가 10분일 경우 편이 계수 값에 상관없이 제안 방법의 효과가 크게 나타남을 알 수 있다. 또한 기존의 패칭 방법보다 본 논문에서 제안한 방법이 평균 요청 간격이 같을 때 생성된 스트림 수가 작게 나타나며, 특히 요청 빈도가 높을수록 제안된 방법이 스트림 수를 적게 생성하였다.

(그림 13)은 VOD 서버에서 정규 채널이 생성되는 평균 시간 간격을 편이 계수 값, 프리픽스 크기 값과 평균 요청 시간 간격에 따라 측정하였다. 정규 채널이 생성되는 시간 간격이 큰 경우 그만큼 VOD 서버와의 대역폭 요구량을 적게 필요로 하므로 시스템의 성능을 높일 수 있음을 의미한다. 점선으로 표시된 것은 제안한 방법을 나타내며 실선은 기존의 패칭 방법을 적용한 경우를 나타낸다.

특히, 제안 방법의 경우 프리픽스 크기가 클수록 기존의 패칭 방법보다 정규 채널의 평균 생성 시간 간격이 넓은 것을 볼 수 있다. 결과 그래프에서 프리픽스의 효과와 함께 편이 계수가 작을수록 성능은 좋게 나오는 것을 볼 수 있다. 또한 평균 요청 간격이 넓을수록 채널 생성 간격은 더욱 크게 나온다. 따라서 채널 생성 간격이 넓은 만큼 VOD 서버에 대한 부하를 줄여 대역폭의 감소 효과를 얻을 수 있다.

4.2.3 요청 패턴에 따른 성능 비교

전체 비디오에 대한 요청 패턴이 편이 계수 값을 비율로 갖는 Zipf's 분포를 따를 때 편이계수 θ 값의 변화에 따른



(그림 14) 편이 계수 값의 변화에 따른 대역폭 요구량

대역폭 요구량을 측정해 본다. 이때 θ 의 범위는 0에서 1이며 θ 의 값이 작을수록 특정 비디오에 대한 요청 비율이 높은 것을 나타낸다. 또 θ 의 값이 1인 경우는 모든 비디오에 대한 요청 확률이 균등함을 뜻한다.

(그림 14)는 편이 계수 값이 변화할 때 요청 간격에 따른 대역폭 요구량을 구하였다. 프리픽스 크기가 10분일 때 평균 요청 간격이 짧은 5초인 경우와 긴 20초인 경우 제안한 방법과 기존의 패칭 방법을 비교하였다. 점선으로 표시된 제안 방법인 P2P 프록시 패칭 방법이 실선의 기존 패칭 방법보다 평균 대역폭을 훨씬 적게 요구한다.

그래프에 나타난 결과에 의하면 편이계수의 값이 작을수록 서버의 대역폭 요구량이 작다. 이는 인기도가 높은 10%의 비디오가 프록시 서버에 의하여 패칭 되므로 편이계수 값이 작아 프록시 서버에 대한 접근이 많아질수록 VOD 서버에 대한 부하가 적어지기 때문이다. 또한 제안된 기법이 기존의 패칭 기법보다 동일한 편이계수 값에서 VOD 서버의 대역폭을 훨씬 작게 요구하는 것을 볼 수 있다. 특히 5초의 요청간격의 경우 제안 기법이 약 30%의 성능 이득을 얻을 수 있었다.

5. 결 론

최근 컴퓨터 기술과 통신 기술의 향상으로 VOD 서비스에 대한 요구가 급속히 증가하고 있다. 이러한 증가는 우수한 VOD 서버 시스템의 성능을 요구하고 있으며, 특히 다수의 클라이언트들에게 고품질의 서비스를 빠르고 효과적으로 제공하여야 한다. 현재의 VOD 시스템 상에서 이러한 요구를 수용하기 위해서는 더욱 효율적인 서버의 기능을 필요로 하며, 현재까지의 주요 연구에서는 VOD 서버의 네트워크 대역폭을 줄임으로써 이를 해결하고자 하였다.

본 논문은 패칭의 성능을 증가시키고 VOD 서버와의 접속을 줄여 네트워크 대역폭 사용을 최대한 줄이기 위해 P2P 미디어 스트리밍 방식을 패칭에 적용한 P2P 기반의 향상된 패칭 기법을 제안하였다. 제안된 기법에서 클라이언트는 스트림 재생 및 버퍼링과 재전송을 수행하도록 하고 클라이언트 요청들은 패칭 윈도우 단위로 그룹화 된다. 동일한 패칭 윈도우 내의 클라이언트들은 같은 스트림 내용으로 버퍼링되어 재생이 이루어지며, 이 스트림은 다음 패칭 윈도우의 그룹에게 재전송을 수행함으로써 패칭 방법에서 갖는 주기적인 정규 채널 생성에 따른 서버의 대역폭을 감소시킬 수 있었다. 또한 프록시 안에 프리픽스 스트림을 캐싱하여 패칭 시킴으로써 서비스 지연시간을 줄여 서비스의 질을 향상시킬 수 있도록 하였다. 다수의 클라이언트들이 연속적으로 요청하는 빈도가 높을수록 제안 방법은 재전송에 의해 새로운 정규 채널을 생성하지 않으므로 더욱 효과를 높일 수 있었다. 또한 새로운 클라이언트의 참여나 재전송 중인 클라이언트가 이탈하여도 재전송을 계속 수행하도록 복구하기 위해 모든 패칭 그룹과 클라이언트의 상태 정보를 유지하는 인덱스 구조를 제안하였다.

제안된 방법의 성능을 평가하기 위하여 프리픽스 크기에 따른 대역폭 요구량과 평균 요청 간격에 따른 대역폭 요구량, 그리고 비디오 요청 패턴에 따른 대역폭 요구량 등을 다양한 파라미터 값을 바탕으로 시뮬레이션을 수행하였다. 성능 평가의 결과로 프리픽스 크기가 10분이고 요청간격이 짧으며 요청 빈도도 높을 때 기존의 패칭 만을 적용한 경우와 비교하였을 때 제안 방법이 약 35%의 성능 증대 효과를 얻었다. 프리픽스 크기가 더 증가하였을 때는 더욱 우수함을 보였다. 실험 결과 모든 경우에 대하여 제안된 방법이 기존의 패칭 방법에 비하여 우수한 성능을 나타내었으며 전체적으로 약 33%의 성능 향상을 보여주었다.

참 고 문 헌

[1] Y. Cai, K. Hua, and K. Vu, "Optimizing patching performance," In Proc. of SPIE/ACM Conf. on Multimedia Computing and Networking, pp.204-215, Jan., 1999.

[2] A. Dan, D. Sitaram, and P. Shahabuddin. "Dynamic Batching Policies for an on-Demand Video Server," *Multimedia Systems*, Vol. 4, No. 3, pp.112-121, Jun., 1996.

[3] T. Do, K. Hua, and M. Tantaoui, "P2VoD: Providing Fault Tolerant Video-on-Demand Streaming in Peer-to-Peer Environment," In Proc. of IEEE Int. Conf. on Communications, Jun., 2004.

[4] C. Griwodz, M. Zink, M. Liepert, G. On, and R. Steinmetz, "Multicast for Savings in Cache-Based Video Distribution," *Multimedia Communication and Networks*, Jan. 2000.

[5] Y. Guo, K. Suh, J. Kurose, and D. Towsley, "P2Cast: Peer-to-Peer Patching Scheme for VoD Service," In Proc. of Int. Conf. on World Wide Web, pp.301-309, May, 2003.

[6] K. Hua, D. Tran, and R. Villafane, "Caching Multicast Protocol for on-Demand Video Delivery," In Proc. of ACM/SPIE Conf. on Multimedia Computing and Networking, pp.2-13, Jan., 2000.

[7] K. Hua, S. Sheu, and D. Tran, "A New Caching Architecture for Efficient Video Services on the Internet," In Proc. of IEEE Symposium on Applications and the Internet, pp.27-31, Jan., 2003.

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

[9] X. Jiang, Y. Dong, D. Xu, and B. Bhargava, "Gnustream: A P2P Media Streaming System Prototype," In Proc. of IEEE ICME, pp.325-328, Jul., 2003.

[10] C. J. Kwon, C. K. Choi, and H. K. Choi, "An Improved Patching Scheme for Video-On-Demand Servers," In Proceedings of the Int. Conf. on Parallel and Distributed Processing Techniques and Applications, pp.1161-1166, Jun., 2004.

[11] C. J. Kwon, C. K. Choi, and H. K. Choi, "An Efficient Patching Scheme Based on Proxy Prefix Caching and Buffer Expanding for Video-On-Demand Services," In Proc. of ACIS Int. Conf. on Computer and Information Science, pp.374-380, Aug., 2004.

[12] 권춘자, 최치규, 최황규, "VOD 시스템 상에서 P2P 프록시 기반의 패칭 기법," 강원대학교 산업기술연구소 논문집, 제24권 B호, pp.177-186, 2004.

[13] 권춘자, 최치규, 최황규, "VOD 시스템을 위한 P2P 프록시 기반의 효율적인 패칭 기법," 한국정보처리학회 추계 학술발표 논문집, pp.221-224, 2004.

[14] J. Lee, "On a Unified Architecture for Video-on-Demand Services," In Proc. of the IEEE Trans. Multimedia, Vol.4, pp.38-47, Mar., 2002.

[15] R. Rejaie, M. Handley, H. Yu, and D. Estrin, "Proxy Caching Mechanism for Multimedia Playback Streams in the Internet," In Proc. of Int. Web Caching Workshop, Mar., 1999.

[16] S. Saroiu, P. Gummadi, and S. Gribble, "A Measurement Study of Peer-to-Peer File Sharing Systems," In Proc. of Multimedia Computing and Networking, 2002.

[17] S. Sen, J. Rexford, and D. Towsley, "Proxy Prefix caching for Multimedia Streams," In Proc. of IEEE INFOCOM, pp.1310-1319, Mar., 1998.

[18] S. Sheu, K. Hua, and W. Tavanapong, "Chaining: A Generalized Batching Technique for Video-On-Demand Systems," In Proc. of IEEE Int. Conf. on Multimedia Computing and Systems, pp.110-117, Jun., 1997.

[19] D. Tran, K. Hua, and T. Do, "A Peer-to-Peer Architecture for Media Streaming," In Proc. of IEEE JSAC Special Issue on Advances in Service Overlay Networks, 2003.

[20] D. Tran, K. Hua, and T. Do, "ZIGZAG: An Efficient Peer-to-Peer Scheme for Media Streaming," In Proc. of IEEE INFOCOM'03, Apr., 2003.

[21] O. Verscheure, C. Venkatramani, P. Frossard, and L. Amini, "Joint Server Scheduling and Proxy Caching for Video Delivery," *Journal of Computer Communications*, Vol.25, No.4, pp.413-423, Jun., 2001.

[22] B. Wang, S. Sen, M. Adler, and D. Towsley, "Optimal Proxy Cache Allocation for Efficient Streaming Media Distribution," In Proc. of IEEE INFOCOM, pp.1726-1735, Jun., 2002.

[23] B. Wang, S. Sen, M. Adler, and D. Towsley, "Optimal Proxy Cache Allocation for Efficient Streaming Media Distribution," *Journal of IEEE Transaction on Multimedia*, Vol.6, No.2, pp.366-374, Apr., 2004.

[24] D. Wu, Y. Hou, W. Zhu, Y. Zhang, and J. Peha, "Streaming Video Over the Internet: Approaches and Directions," *Journal of IEEE Transactions on Circuits and Systems for Video Technology*, Vol.11. No.3, pp.1-20, Mar., 2001.

[25] Z. Xiang, Q. Zhang, W. Zhu, Z. Zhang, and Y. Zhang, "Peer-to-Peer Based Multimedia Distribution Service," In Proc. of IEEE Transactions on Multimedia, Vol.6, No.2, pp.343-355, Apr., 2004.

[26] D. Xu, M. Hefeeda, S. Hambruch, and B. Bhargava, "On Peer-to-Peer Media Streaming," In Proc. of IEEE Int. Conf. on Distributed Computing Systems, Vol.1, pp.363-371, Jul., 2002.

[27] L. Zhu, Z. Sahinoglu, G. Cheng, A. Vetro, N. Ansari, and H. Sun, "Proxy Caching for Video on Demand Systems in Multicast Networks," in Proc. of Int. Conf. on Information Sciences and Systems(CISS), Mar., 2003.

권 춘 자



e-mail : kwoncj@mail.kangwon.ac.kr
1986년 한양대학교 전자공학과(학사)
1991년 한양대학교 전자계산학과
(공학석사)
1991년 4월~1994년 4월 한국과학기술
정보연구원 데이터베이스운영실
연구원

2002년 3월~2005년 2월 한림성심대학 컴퓨터정보과 초빙교수
2005년 2월 강원대학교 대학원 컴퓨터정보통신공학과(공학박사)
2005년 3월~현재 강릉영동대학 사이버경찰과 초빙교수
관심분야: 멀티미디어 VOD 시스템, 데이터베이스 시스템, 클러스터 웹 서버 시스템 등

최 황 규



e-mail : hkchoi@kangwon.ac.kr
1984년 2월 경북대학교 전자공학과(학사)
1986년 2월 한국과학기술원 전기및전자
공학과(석사)
1989년 8월 한국과학기술원 전기및전자
공학과(박사)

1994년 7월~1995년 7월 Univ. of Florida 방문교수
1999년 3월~2001년 2월 강원대학교 전자계산소 소장
2002년 7월~2003년 7월 Univ. of Minnesota 방문교수
1990년 3월~현재 강원대학교 전기전자정보통신공학부 교수
관심분야: 멀티미디어 시스템, 데이터베이스 시스템, Intelligent Storage System 등