주문형 멀티캐스팅 기법을 사용한 새로운 MOD 시스템의 구현

김 영 준† · 황 태 준† · 권 기 섭† · 김 익 수††

요 약

본 논문은 멀티캐스트 전송을 이용한 MOD 시스템을 구현하였다. 기존의 시스템은 서버 지향적인 멀티캐스트 시스템을 제공하였지만 구현된 시스템은 사용자 요구 지향적인 멀티캐스트 시스템을 제공한다. 스케줄러는 사용자의 요구를 수집하고 요구된 비디오 아이템과 서비스 시간에 따라서 멀티캐스트 그룹 주소와 포트 번호를 발생한다. 그러면 스케줄러는 MOD 서버와 서비스를 요구한 사용자에게 즉시 멀티캐스트 주소와 포트 번호를 전송한다. 그리고 MOD 서버는 멀티캐스트 그룹 주소로 요구한 스트림을 전송하고, 사용자는 자동으로 그룹 주소에 가입한다. 스케줄러는 같은 스케줄링 시간 안에 있는 다른 사용자들이 동일한 비디오를 요구하였을 때 같은 멀티캐스트 그룹 주소를 할당한다. 구현된 MOD 시스템은 동시에 많은 사용자에게 서비스할 뿐 아니라 서버의 부하를 크게 줄일 수 있음을 확인하였다.

Implementation of New MOD System Using On-demand Multicasting Technique

Youngjune Kim† · Taejune Hwang† · KiSeop Kwon† · Iksoo Kim††

ABSTRACT

This paper implements MOD system using multicast delivery. Conventional system provide server-based system in multicast delivery but implemented system provides on-demand client-based multicast system. The scheduler aggregates clients' request and it generate multicast group addresses and port numbers according to requested video items and service request time. Then it transmits immediately multicast address to MOD server and client who request service. And then MOD server transmits requested streams with a multicast group address and the client joins the group automatically. The scheduler assigns the same multicast group address when other clients request an identical video within the same scheduling duration. The system can reduce load of server and support many clients at the same time.

키워드 : MOD, 멀티미디어, 멀티캐스팅, On-Demand

1. 서 론

통신기술과 컴퓨터 기술의 발전은 고속 네트워크의 구현을 가능하게 하였으며 대용량 멀티미디어 전송과 멀티미디어 시스템 구축에 대한 관심을 높여주고 있다. 고속 네트워크는 이러한 대용량 멀티미디어를 전송할 수 있는 대역폭을 보장할 수 있게 되었는데, ADSL과 VDSL은 사용자 단말환경의 좋은 경우라고 할 수 있다.

주문형 멀티미디어 시스템은 네트워크를 통하여 사용자가 원하는 시간에, 원하는 정보를, 원하는 품질로 서비스를 제공받는 사용자 중심의 서비스로서 단순한 텍스트 데이터에서 동영상, 음악, 이미지 등 다양한 미디어를 통합한 멀티미디어 데이터를 요구하기에 이르렀다. 따라서 네트워크 망은 보다 대용량의 데이터를 전송해야 할 필요가 있게 되었다[1, 2].

현재 주문형 멀티미디어 시스템에서 서버 및 네트워크의 사용 효율을 높이면서 사용자들에게 다양한 주문형서비스를 제공하기 위하여 프락시 캐싱 기법과 멀티캐스트 방식이 제안되었다.

프락시 캐싱 기법은 프락시를 클라이언트 부근에 두어 종단간 지연과 네트워크 부하도 줄일 수 있으며, 한번 요청된 데이터를 프락시에 저장하여 이후에 프락시에 캐쉬되어 있는 비디오를 재차 요청할 경우에는 서버를 통하지 않고 프락시에만 접속하여 서비스를 제공받는 방식이다. 그렇지만 특정 인기 있는 비디오가 저장되어 있는 프락시에 항상 과부하가 발생하는 문제점이 있다[5-8].

멀티캐스트 전송방식은 동일한 비디오 컨텐트를 요청한 사용자들의 서비스 요청을 미리 설정된 시간 단위로 그룹

※ 본 연구는 한국과학재단 지정 인천대 멀티미디어 연구센터의 지원에 의한 것입니다.
† 준 회 원 : 인천대학교 대학원 정보통신공학과
†† 정 회 원 : 인천대학교 정보통신공학과 교수
논문접수 : 2003년 5월 27일, 심사완료 : 2003년 10월 9일

화하여 서비스를 제공하는 방식으로, 그룹화는 수 분단위로 이루어지는 것이 일반적이다[3, 4, 9]. 그러나 기존의 멀티캐스트 전송 방식은 서버-중심으로 사용자 요구에 의한 스케줄링이 아니라 사전에 게시시점을 미리 설정하여 그 시점에서만 사용자들이 서비스를 제공 받을 수 있는 방법으로 구현되고 있다.

본 논문에서 제안하는 시스템은 사용자가 원하는 아이템에 대한 서비스 요청을 하면 서버측의 멀티캐스트 스케줄러가 즉시 멀티캐스트 그룹주소와 포트번호를 발생하여 이를 미디어 서버와 사용자에게 전송하여 사용자들은 자동적으로 멀티캐스트 그룹에 참여(join)하여 서비스를 제공받는 사용자-중심의 멀티캐스트 전송을 사용한다[1, 2]. 이 서비스는 대규모 아파트 단지나 최신의 비디오 컨텐츠 서비스 같은 로컬 네트워크에서 사용자들이 동일한 아이템에 대해서 서비스 시점이 비슷한 경우나, 사내 망과 같은 인트라넷 상에서 보다 많은 사용자의 요구를 충족시킬 수 있는 주문형 특성을 살려 사용자-중심의 멀티캐스트를 사용한 MOD 시스템 구현을 하는데 적합하다.

2. MOD 시스템 구성요소

2.1 서비스 스케줄러

멀티캐스트 전송 방법은 동일한 아이템에 대한 욕구가 많을수록 자원을 보다 효율적으로 사용하게 된다. 그러나 기존의 멀티캐스트 전송을 사용한 서비스는 사용자들이 원하는 시간에 서비스를 받기보다는 정해진 시간에 접속하여 서비스를 받도록 하였다. 따라서 사용자가 원하는 시점이 아닌 서비스 제공자의 입장에서 데이터를 전송하므로 원천적으로 주문형 서비스를 제공할 수 없다는 단점을 갖고 있다.

그러나, 본 논문에서 제안한 시스템은 기존의 멀티캐스트의 단점을 제거하여 주문형 서비스 특성이 제공되는 시스템으로서, 멀티캐스트 스케줄러는 동일한 데이터를 동시에 요청한 사용자들에게 단일 채널을 사용하여 서비스를 제공할 수 있도록 하기 위해 사용자 요구를 스케줄링 한 결과로 발생된 각 그룹별 멀티캐스트 주소와 포트번호를 할당하게 된다.

사용자들의 서비스 요청이 스케줄러에 의해 동일한 그룹에 속하기 위한 조건은, 동일한 아이템을 신청한 사용자이어야 하고, 사용자들이 납득할 만한(약 20초에서 1분) 대기시간으로 설정된 멀티캐스트 그룹 설정 시간(Grouping Interval)의 범위 내에 서비스를 신청한 사용자들이어야 한다. 스케줄러는 사용자의 요청을 받아들인 후 사용자의 요구를 서비스 할 수 있는지 시스템 자원을 확인한다. 시스템에서 서비스에 사용되는 채널의 여유가 없을 경우에는 사용자의 요청을 거절하거나 대기시켜야 한다. 그러나 채널의 여유가 있을 경우 멀티캐스트 스케줄러는 즉시 채널을 확보한 후 멀티캐스트 그룹화 작업을 수행하게 된다. 이와 함께 스케줄러는 현재 제공되고 있는 아이템의 목록과 관리를 위한 정보와 서비스를 요구한 모든 사용자들의 정보, 아이템별 타이머, 영화별로 동일한 그룹에서 서비스 받고 있는 사용자들에 대한 목록 등을 항상 유지하며 관리하여야 한다.

2.2 미디어 서버

미디어 서버는 사용자가 요청할 대용량의 데이터를 저장하고 스케줄러로부터 멀티캐스트 그룹 주소와 포트번호를 수신한 다음 네트워크를 통해 멀티캐스트 그룹에게 멀티미디어 스트림 데이터를 실시간으로 전송한다.

본 논문에서 제안한 시스템의 경우, 스케줄러를 통해 서비스 요청을 하게 되므로 서버는 자신이 서비스 할 사용자들의 세부 정보들을 유지할 필요가 없다.

2.3 클라이언트 프로그램

클라이언트 프로그램은 네트워크를 통하여 전송된 데이터 스트림을 수신하고 이를 디코딩하여 사용자에게 보여준다. 사용자가 서비스를 제공받기 위해서는 스케줄러에게 원하는 아이템을 요청해야 한다. 사용자가 요청한 데이터가 서버에 존재하면 스케줄러는 멀티캐스트 그룹 주소를 발생시켜서 클라이언트 프로그램과 서버에게 알려준다. 클라이언트 프로그램은 스케줄러로부터 멀티캐스트 그룹 주소를 얻게 되면 스케줄러와 연결되었던 소켓을 닫고, 서버와의 통신을 위하여 새로운 멀티캐스트 소켓을 생성한다. 또한 서버는 멀티캐스트 그룹 주소를 이용해서 클라이언트 프로그램에게 데이터를 전송하기 위하여 데이터그램 소켓을 생성한다.

2.4 단말 네트워크

단말 네트워크 환경은 사용한 시스템이 가장 알맞게 동작을 할 수 있는 지역적으로 인접하면서, 다수의 사용자가 같은 라우터에 접속되어 있는 사용자 밀집형 구조이다. 예를 들어, 아파트 단지에서 비디오나 영화 서비스를 한다든지, 학교 LAN 망에서 교육용 컨텐츠를 지정해서 제공하는 경우, 또는 회사에서 사내 교육 자료를 제공하는 경우 등이 될 수 있다. 또한 웹상에서 멀티캐스트 전송의 범위가 확대 될 수 있다. 기존의 방송국에서 많은 인기를 끌었던 지난 프로그램이나, 엄청난 관객을 모았던 개봉영화 등을 전국적으로 스트리밍 할 수 있는 대규모 서비스 네트워크로 확대될 수 있다.

3. MOD 시스템의 구현 및 동작

본 논문에서 구현하는 시스템은 자바 JSDK 버전 1.4와 Java Media Framework(JMF)로 제작되었다. 클라이언트 프로그램과 스케줄러는 서로 소켓을 만들어 통신을 할 준비를 하고 사용자가 보낸 요청을 처리하여 원하는 아이템의 번호와 사용자의 주소를 스케줄러에 저장한다. 사용자와 스케줄러는 일대일 관계가 아니라 다대일(n-to-one) 관계이

므로 사용자가 서비스 요청을 하게 되면 스케줄러의 ServerSocket은 하나의 요청당 하나의 쓰레드를 발생시켜 각각의 연결 설정을 유지한다. 발생된 그룹 멀티캐스트 주소는 사용자들과 서버에 전달되고, 서버는 멀티캐스트 그룹에게 서비스를 제공하게 되며 이때 사용자들은 스케줄러로부터 전송되어온 그룹 멀티캐스트 주소 그룹에 join하여 요청된 비디오를 시청하게 된다.

```
step 1 : 스케줄러 접속
step 2 : 서비스 받고자 하는 아이템 ID 입력
step 3 : if (사용자가 요청한 아이템이 스케줄링 중인
           세션이 있으면)
         else {
         Grouping Interval 동안 그룹핑 처리 ;
         멀티캐스트 그룹 주소와 포트 생성 ;
         }
step 4 : 멀티캐스트 주소와 포트를 서버와 사용자에게 전송
step 5 : 수신한 멀티캐스트 그룹 주소와 포트로 멀티캐스트
         그룹에 Join
step 6 : 요청했던 데이터를 서비스 받는다.
step 7 : 서비스 종료
```

(그림 1) 서비스 Procedure

3.1 스케줄링

서비스 스케줄러 구현부는 두 부분으로 나뉘어 진다. 스케줄러는 서버 소켓을 사용하여 사용자의 요청을 받고 요청 아이템별로 멀티캐스트 그룹을 만들고, 관리하는 부분과 사용자와의 연결을 가지고 있는 소켓 설정 쓰레드 부분으로 나누어진다. 명칭을 각각 메인 스케줄러와 스케줄러 쓰레드로 표기하겠다. (그림 2)는 스케줄러의 동작 순서를 나타내었다.

메인 스케줄러 부분에서는 서비스가 시작되면 서버 소켓을 만들어 사용자의 요청을 기다리고, 요청이 왔을 경우 서버 소켓 객체의 accept() 함수에서 넘겨지는 소켓 객체를 받아 이를 인자로 하여 스케줄러 쓰레드 객체를 생성한다.

메인 스케줄러는 사용자의 요청을 받아서 그룹핑을 담당하는 쓰레드로서, 소켓 연결을 넘기는 동작은 무한 루프를 통해 구현되고 서버가 서비스되는 동안 멈추지 않는다. 사용자 입력을 받아들여 스케줄러 쓰레드 객체의 인자로 보내진 소켓 객체는 나중에 스케줄링이 끝나게 되면 종료하여 스케줄러에서 불필요하게 생성되어 있는 소켓 연결들이 유지되지 않도록 한다.

```
step 1 : 스케줄러 시작
step 2 : while (사용자 접속) {
step 3 : Scheduler Thread 실행
step 4 : 사용자가 원하는 아이템의 ID값을 입력받음
step 5 : if (동일 아이템 서비스를 위한 Scheduling이 진행 중)
         else
         새로운 멀티캐스트 그룹 주소와 포트 생성 ;
         발생된 주소와 포트를 서버와 사용자에게 전송
         }
step 6 : 스케줄링 종료
```

(그림 2) 스케줄러 Procedure

메인 스케줄러에서 이루어지는 그룹핑을 거쳐 서비스에 쓰일 멀티캐스트 주소가 결정되면 멀티캐스트 주소를 미디어 서버와 클라이언트 프로그램에게 전송한다.

클라이언트 프로그램과 연결된 스케줄러 쓰레드는 자신과 통신하는 사용자의 주소, 신청한 아이템의 구별번호를 받아들여 공통 데이터 객체를 담고 있는 CommonData 클래스에 정의 되어있는 멀티캐스트 주소 발생 함수와 포트 발생 함수, 현재 스케줄링 중인 아이템에 사용될 멀티캐스트 주소 저장 배열, 사용 가능한 그룹 주소인지 확인하는 플래그 정보 저장 배열 등을 엑세스하여 스케줄링을 수행한다.

스케줄러 쓰레드는 사용자와 연결을 만들기 위해 소켓을 생성하고 소켓간에 데이터를 보내고 받기 위한 스트림을 만든다. 일단, 사용자에게서 전송되어진 서비스 요청 아이템에 대한 정보와 기타 다른 정보들도 메인 스케줄러로 보내져 스케줄링을 거치게 되고, 이때 만들어진 멀티미디어 아이템을 사용자에게 전달하는 것으로 임무가 완료된다. 따라서 사용자가 많아질 때 그에 따른 스케줄링을 순차적인 프로그램 구조에서 수행하는 것 보다 서버부하를 줄이기에 용이한 구조를 갖고 있다.

3.2 서비스 제공

본 논문에서 제안하는 시스템의 미디어 서버는 사용자와 연결을 간접적으로 유지하는 데이터그램 소켓유지부분과 직접 사용자가 요구한 아이템을 전송하는 데이터 전송부분으로 나뉜다.

사용자의 요청에 따라 그룹핑을 거친 후 발생된 멀티캐스트 그룹 주소를 미디어 서버가 받게 되면 사용자에게 요청한 데이터를 전송할 준비를 하게 된다. 이때 서버는 (그림 3)과 같은 순서로 동작을 수행하게 된다. .

```
step 1 : 스케줄러에서 서버 호출(서버 시작)
step 2 : 멀티캐스트 그룹 주소와 포트를 입력 받음
step 3 : 데이터의 압축 포맷과 전송률 점검
step 4 : 패킷의 목적지 필드에 멀티캐스트 그룹 주소와 포트 입력,
         패킷 전송
step 5 : 서버 프로그램 종료
```

(그림 3) 서버 Procedure

3.3 클라이언트 프로그램

(그림 4)에서 알 수 있듯이 사용자가 원하는 아이템을 서버에게 요청하려면 클라이언트 프로그램과 스케줄러가 소켓을 생성하고 데이터를 전송해야 한다. 따라서 우선 서비스를 신청하기 전에 원하는 아이템을 선택한 후 소켓을 생성하여 원하는 데이터의 내용을 스케줄러의 메인 스케줄러로 전송하면 스케줄러 쓰레드 부분에서 생성된 멀티캐스트 그룹 주소를 돌려받아 그 주소로 멀티캐스트 소켓을 만들고, 미디어 서버에 접속해서 다른 동일한 아이템을 요청한 사용자들과 동시에 서비스를 받는 전송 세션을 유지한다.

클라이언트 프로그램 구현부에서는 전송된 멀티캐스트 주소가 목적지 주소 필드에 삽입되어 온 패킷을 감지하면 된다. 이렇게 전송된 패킷은 클라이언트 프로그램의 버퍼에 저장되면서 사용자에게 서비스된다.

step 1 : 클라이언트 프로그램 시작
step 2 : 소켓 생성, 스케줄러와 연결 준비
step 3 : 연결 후 원하는 아이템의 ID를 스케줄러에 요청
step 4 : 스케줄러에서 전송된 멀티캐스트 그룹 주소와 포트를 전달받음
step 5 : 스케줄링을 위한 소켓 닫음
step 6 : 스케줄러가 정해준 그룹에 Join하여 데이터를 전송 받음
step 7 : 아이템 재생
step 8 : 플레이어 끝

(그림 4) 클라이언트 프로그램 Procedure

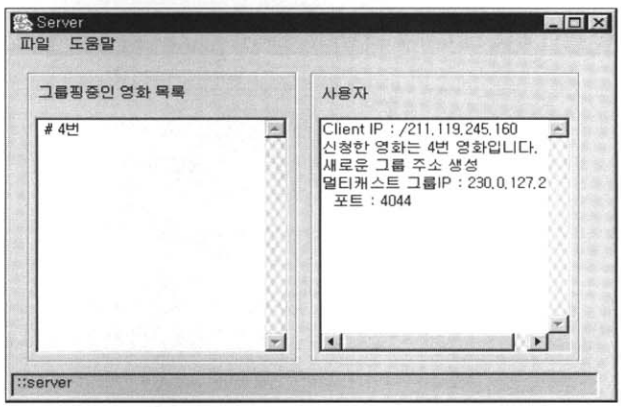
3.4 클라이언트 프로그램과 서버 프로그램의 그래픽 사용자 인터페이스

(그림 5)에 나타낸 바와 같이 서버 측에서는 스케줄러를 먼저 실행시켜야 하므로 명령 행에서 응용프로그램을 구동시킨다.

(그림 5) 서버 프로그램과 스케줄러 실행 화면

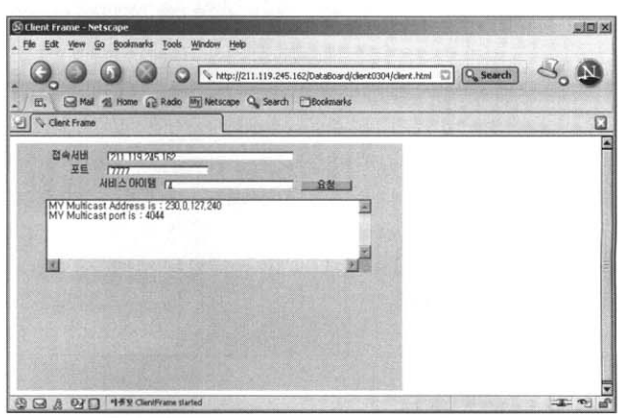
본 논문에서 구현된 프로그램은 서버가 데이터를 전송하기 전에, 클라이언트 프로그램과 스케줄러가 소켓으로 연결을 설정하고 서비스에 필요한 멀티캐스트 그룹 주소와 포트를 발생시키고 이들을 사용자와 미디어 서버에 전송하므로, 사용자는 서비스 요청을 위해 서버에 직접 접속을 하지 않고 스케줄러에 접속을 하게 된다.

응용프로그램을 실행시키면 명령 행에 사용자의 인터넷 접속주소를 출력하고 이후 JMF를 통해 구현된 플레이어에서 데이터 설정에 관한 프로파일(profile)을 체크하는 과정이 나타난다.



(그림 6) 스케줄러 접속 화면

(그림 6)에 스케줄러 프레임이 나타나 있다. 좌측에는 현재 그룹핑 중인 아이템의 목록이 나열된다. 새로운 사용자가 접속하여 데이터를 요청할 때마다 공통 데이터를 저장하고 있는 CommonData class에 존재하는 그룹핑 중인 아이템 배열을 조사하여 각 아이템에 해당하는 필드가 set 상태이면 그룹핑 리스트에 추가한다. 우측에는 스케줄러에게 서비스를 요청한 사용자들의 인터넷 주소와 요청한 아이템의 ID, 스케줄링 결과로 발생된 멀티캐스트 그룹 주소 및 포트를 표시한다.



(그림 7) 클라이언트 프로그램 실행화면 및 접속화면

(그림 7)과 같이 브라우저 창에 실행된 클라이언트 프로그램 프레임에는 접속할 스케줄러의 DNS Name이나, 인터넷 주소를 입력하고 시스템에서 사용하는 공통 접속 포트 7777(시스템에서 고정한 포트 번호)을 입력한다.

미디어 서버에서 전송을 시작하면 사용자 측에는 (그림

8)과 같이 컨트롤러를 가진 플레이어가 생성되어 아이템을 재생한다. 20초 이내에 다른 사용자에서 동일한 데이터를 신청할 경우, 같은 멀티캐스트 그룹 주소와 포트를 사용하고 동기화가 된 아이템을 동시에 볼 수 있다.

(그림 8) 영화 상영 화면

4. MOD 서버 효율 및 시뮬레이션 분석

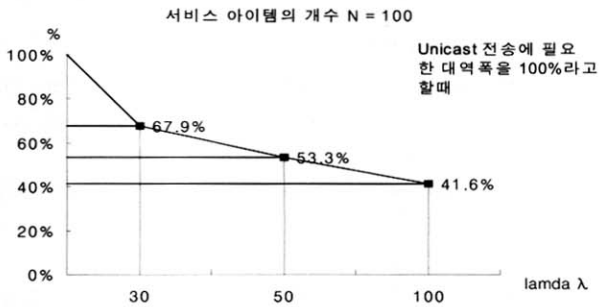
사용자가 각 아이템을 요구할 확률은 아이템의 인기도에 따라 다르다. 인기가 제일 많은 아이템은 i번째 아이템이 된다. 미디어 서버가 제공할 수 있는 N개의 아이템 ID 가운데 i번째 인기 있는 아이템을 선택할 확률을 구하기 위하여 Zipf 분포를 사용한다. N개의 아이템 ID 가운데 i번째로 자주 선택할 확률은 Z/i로 나타낼 수 있으며, 여기서 $Z = 1/(1 + 1/2 + 1/3 + \cdots + 1/N)$이다. Zipf 분포를 사용하면 사용자가 i번째 아이템을 요청할 확률은 ρ_i이며 $_i$에 관한 함수 $\rho = Z/i$로 나타낼 수 있다[10]. 서비스 요청율을 λ라 하면 i번째 아이템의 서비스 요청율은 $\lambda_i = \lambda\rho_i$이며, 시뮬레이션에서 ρ_i는 가중치 파라미터이다. 따라서 이러한 서비스 요청 패턴을 가지고 있는 사용자들에게 멀티캐스트 전송 방식을 이용한 스케줄링 시스템을 사용하여 서비스를 제공하면 서버의 부하와 소요되는 네트워크 대역을 크게 절감할 수 있다.

시뮬레이션에서 사용되는 미디어 서버가 제공할 수 있는 아이템의 수는 N = 100개이고, 서비스 요청율 λ는 사용자가 서비스를 요청하는 횟수를 나타내고 Poisson 분포를 따른다.

이를 식으로 나타내면 다음과 같다.

$$f_x(X=k) = \lim_{n\to\infty} p^k (1-p)^{n-k} \cong \frac{\lambda^k}{k!} e^{-\lambda}$$

(그림 9)은 멀티캐스팅 기술을 사용한 알고리듬을 사용하고, 서비스 요청율 $\lambda = 30$에서 $\lambda = 100$인 경우로서 한 개 아이템에 대하여 유니캐스트 서비스에 비해 채널 수가 평균 54.26%로 줄어들 수 있음을 확인하였다. 사용자의 서비스 요청율(λ)에 따른 서비스 대역은 유니캐스트 전송과 비교하였을 때, $\lambda = 30$인 경우 32.1%의 채널 감소 효과를 가져왔고, $\lambda = 50$인 경우 46.7%, $\lambda = 100$인 경우 58.4%에 달하는 채널 감소 효과를 보였다.



(그림 9) 서비스 요청율에 의한 전송대역폭

5. 결 론

본 논문에서는 새로운 멀티캐스트 스케줄링을 이용하여 사용자-중심의 MOD 시스템을 구현하였다.

기존의 시스템은 서버-중심의 서비스였지만, 구현된 시스템은 사용자가 동일한 아이템을 일정시간에 요청할 때 즉시 사용자 요구에 의해 반응하는 사용자-중심의 주문형서비스이다. 즉 특정 인기 있는 아이템에 집중되는 환경에서 근소한 시간 간격을 두고 서비스 요청을 한 사용자들에게 동일한 채널을 통해 서비스를 제공한다.

정해진 대기시간동안 신청자 그룹을 정하고 스케줄링을 통해 멀티캐스트 그룹 주소와 포트를 사용자 그룹이 공유함으로써 서버의 부하와 네트워크 채널 사용을 줄일 수 있음을 확인하였다.

참 고 문 헌

[1] K. Almeroth and M. Ammar, "Providing a scalable, interactive Video-On-Demand service using multicast communication," in ICCCN'94, San Francisco, CA, September, 1994.

[2] D. Eager, M. Vernon and J. Zahorjan, "Optimal and Efficient Merging Schedules for Video-on-Demand Servers," Proc. ACM Multimedia 99, ACM, 1999.

[3] R. Braudes and S. Zabels, "Requirement for Multicast Protocol," RFC 1458, May, 1993(Status : Informational).

[4] S. Armstrong, A. Freier and K. Marzullo, "Multicast Transport Protocol," RFC1301, February, 1992(Status : Informational).

[5] R. Rajaie, H. Yu, M. Handley and D. Estrin, "Multimedia Proxy caching mechanism for Quality Adaptive streaming Application in the Internet," Proc. IEEE Infocom, March,

2000.

[6] C. Shahabi and M. H. Alshayeji, "Super-streaming : a New Object Delivery Paradigm for Continuous Media Servers," Journal of Multimedia Tools and Applications, Vol.11, No.1, pp.275-298, May, 2000.

[7] L. Fan, P. Cao, J. Almeida and A.Z. Broder, "Summary Cache : A Scalable Wide-Area Web Cache Sharing Protocol," Proceedings of ACM SIGCOMM'98, Technical Report 1361, Computer Sciences Department, Univ. of Wisconsin-Madison, pp.254-265, Feb., 1998.

[8] S. Sen, J. Rexford and D. Towsley, "Proxy prefix caching for multimedia streams," in *Proceedings of IEEE Infocom 99*, NewYork, USA., 1999.

[9] S. Hares and D. Katz, "Administrative Domains and Routing Domains : A model for routing in the Internet," RFC 1136, December, 1989(Status : Informational).

[10] P. Cao, L. Fan and G. Philips, "Web Caching and Zipf-like Distributions : Evidence and Implications," IEEE Infocom 1999.

김 영 준

e-mail : yjkim@incheon.ac.kr

2001년 인천대학교 정보통신공학과 (공학사)

2003년 인천대학교 정보통신공학과 대학원 (공학석사)

2003년~현재 (주)KBS 인터넷(사원)

관심분야 : Multicast, MOD, 멀티미디어

황 태 준

e-mail : tjhwang@incheon.ac.kr

1997년 인천대학교 전자계산학과(공학사)

1999년 인천대학교 전자계산학과 대학원 (공학석사)

1999년~2001년 인천대학교 정보통신공학과 대학원 박사과정 수료

관심분야 : Multicast, VOD, Webcaching, 데이터베이스, 멀티미디어

권 기 섭

e-mail : kkscom@netian.com

1998년~현재 인천대학교 정보통신공학과 재학

관심분야 : Multicast, Network, 멀티미디어

김 익 수

e-mail : iskim@incheon.ac.kr

1977년 동국대학교 전자공학과(공학사)

1981년 동국대학교 전자공학과 대학원 (공학석사)

1985년 동국대학교 전자공학과 대학원 (공학박사)

1988년~1993년 인천대학교 정보통신공학과 부교수

1993년~1994년 North Carolina State Univ. 객원교수

1994년~현재 시립 인천대학교 정보통신공학과 교수

관심분야 : Multicast, Network, VOD, Webcaching