멀티캐스트를 이용한 사용자 기반의 비디오 주문형 시스템 구현

Client-Based Video-On-Demand System Implementation using Multicast

황 태 준* 김 백 현*** 김 익 수**
Hwang Tae June Kim Back Hyun Kim Ik Soo

요 약

본 논문은 멀티캐스트 전송을 이용한 사용자 기반의 VOD 서비스를 구현하였다. 기존의 서버 기반의 멀티캐스트 시스템을 제공하였지만 구현된 시스템은 사용자 요구 기반의 멀티캐스트 시스템을 제공한다. 멀티캐스트 에이전트 스케줄러(Multicast Agent Scheduler : MAS)는 사용자의 요구를 수집하고 요구된 비디오 아이템과 서비스 시간에 따라서 멀티캐스트 그룹과 포트 번호를 발생한다. 그러면 그것은 VOD서버와 서비스를 요구한 사용자에게 즉시 멀티캐스트 주소와 포트 번호를 전송한다. 그리고 VOD서버는 멀티캐스트 그룹 주소로 요구한 스트림을 전송하고, 사용자는 자동으로 그룹 주소에 가입한다. 멀티캐스트 에이전트 스케줄러는 같은 스케줄링 시간 안에 있는 다른 사용자들이 동일한 비디오를 요구하였을 때 같은 멀티캐스트 그룹 주소를 할당한다. 제안된 VOD 시스템은 동시에 많은 사용자에게 서비스할 뿐 아니라 서버의 부하를 크게 줄일 수 있음을 확인하였고, 이것을 마이크로 소프트의 WMT(Windows Media Technology)에 적용하였다.

Abstract

This paper presents implementation of client-based VOD service using multicast delivery. Conventional system provide server-based system in multicast delivery but implemented system provides on-demand client-based multicast system. The Multicast Agent 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 VOD server and client who request service. And then VOD server transmits requested streams with a multicast group address and the client joins the group automatically. The Multicast Agent Scheduler assigns the same multicast group address when other clients request an identical video within the same scheduling duration. The proposed system can reduce load of server and support many clients at the same time and applies it to WMT(window media technology) of Microsoft.

☞ Keyword : 멀티캐스트, 주문형, 스트리밍, 멀티미디어, 주문형 서버/ Multicast, On-demand, Streaming, Multimedia, VOD server

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, ADSL, VDSL은 사용자 단말환경의 좋은 경우라고 할 수 있다.

사용자 요구 서비스는 사용자가 데이터 서비스를 제공받는 데에만 목적을 두는 것이 아니라, 사용자가 원하는 시간에, 원하는 정보를, 원하는 품질로 서비스 받도록 하여 사용자 기반의 주문형 서비스로 변화되고 있으며 종류도 단순한 텍스트 데이터에서 동영상, 음악, 이미지 등 다양한 미디어를 통합한 비디오 데이터를 요구하기에 이르렀다. 따라서 네트워크 망은 보다 많은

* 정 회 원 : 인천대학교 정보통신공학과 대학원 박사수료
tjhwang@incheon.ac.kr(제1저자)
** 정 회 원 : 인천대학교 정보통신공학과 대학원 박사과정
hidesky24@incheon.ac.kr(공동저자)
*** 정 회 원 : 인천대학교 정보통신공학과 교수
iskim@incheon.ac.kr
[2005/02/21 투고 - 2005/03/09 - 1차 심사, 2005/10/12 - 2차 심사 - 2005/12/06 심사완료]

대용량의 비디오 데이터를 전송해야 할 필요가 있게 되었다.[1,2]

현재 이러한 사용자 요구 서비스와 주문형 비디오 시스템에서 서버 및 네트워크의 사용 효율을 높이면서 사용자들에게 다양한 대화형 서비스를 제공하기 위하여 프락시 캐싱 기법과 유니캐스트 및 브로드캐스트 전송 방식의 장점만을 취한 멀티캐스트 방식이 제안되었다.

프락시 캐싱 기법은 프락시를 클라이언트 부근에 두어 종단간 지연과 네트워크 부하도 줄일 수 있으며, 한번 요청된 데이터를 프락시에 저장하여 이후에 프락시에 저장되어 있는 비디오를 재차 요청할 경우에는 서버를 통하지 않고 프락시에만 접속하여 서비스를 제공받는 방식이다. 분산 프락시를 두어 보다 원활하게 양질의 서비스를 구현하고 있다. 그러나 분산 프락시를 구현하여 서비스를 제공하더라도 특정 프락시가 최근 인기 있는 비디오를 저장하고 있다면 많은 클라이언트들로부터 스트림 요청이 들어오게 된다. 이는 상대적으로 다른 프락시에 비해 트래픽이 폭주 될 가능성이 있기 때문에 여러 프락시에 스트림들을 분산 저장하게 된다. 하지만 기존의 분산 프락시들은 동일한 비디오의 복사본만을 저장할 뿐만 아니라 이들의 분산 정책은 클라이언트들의 서비스 환경을 고려하지 않고 동일한 품질의 스트림만을 전송한다.[4,7,8,9]

멀티캐스트 전송방식은 동일한 비디오 컨텐트를 요청한 사용자들의 서비스 요청을 미리 설정된 시간 단위로 그룹화하여 서비스를 제공하는 방식으로, 그룹화는 수 분단위로이루어지는 것이 일반적이다.[5,6] 멀티캐스트 전송방식은 하나의 전송 스트림을 사용하여 복수의 사용자들에게 데이터를 전송하므로 유니캐스트 전송방식에 비해 네트워크 자원을 효율적으로 사용하는 기술이다. 그러나 기존의 멀티캐스트 전송 방식은 서버기반으로 사용자 요구보다는 서버의 스케줄링에 우선하여 상당히 사용자에게 제한적인 방식으로 서비스할 수밖에 없었다. 즉 동일한 그룹 아이디를 통해 서비스 받는 사용자 그룹에게 서비스 제공자가 아이템을 정해서 미리 E-mail 등을 통해 서비스 개시 시간을 공지 한 후에 접속을 하여 서비스를 받는 방식이었던 반면, 본 논문에서 제안하는 사용자 기반의 주문형 서비스는 사용자가 원하는 아이템에 대한 서비스 요청을 하면 일정 시간동안 서비스를 잠시 지연하여 같은 아이템을 요구하는 사용자 그룹을 실시간으로 묶어서 서비스를 제공하므로 사용자 중심의 서비스를 하면서도 동시에, 멀티캐스트 전송을 제공할 수 있다.

본 논문은 2장에서는 사용자 기반의 비디오 주문형 시스템의 구조와 동작에 대해서 설명하고 있으며, 3장에서는 비디오 주문형 시스템 구현 및 알고리즘을 제안하며, 4장에서는 제안된 기법을 시뮬레이션하고 그 성능을 분석한다. 마지막으로 5장에서는 결론에 대하여 기술한다.

2. 비디오 주문형 시스템의 구조와 동작

2.1 멀티캐스트 에이전트 스케줄러(Multicast Agent Scheduler : MAS)

MAS는 서비스를 요구하는 사용자와 VOD 서버에게 멀티캐스트 그룹 주소와 포트 번호를 전송한다. MAS는 채널 효율을 증가시키기 위해 사용자들이 동일한 비디오 데이터를 요구할 때 동일한 멀티캐스트 그룹으로 그룹핑한다. 만약 어떤 사용자가 비디오 아이템에 대한 서비스를 요구한다면, MAS는 동일한 아이템으로 스케줄링 쓰레드가 존재하는 하지 않는지를 먼저 조사한다. MAS는 요구된 비디오 아이템에 대한 쓰레드가 존재한다면 사용자에게 동일한 멀티캐스트 그룹 주소와 포트 번호를 할당한다. 만약 그렇지 않다면 MAS는 사용자에게 또 다른 멀티캐스트 그룹 주소와 포트 번호를 생성한다. 따라서 MAS는 각각의 그룹핑 시간이 지나갔을 때 동일한 아이템을 요구하는 사용자에게 다른 멀

티캐스트 그룹 주소와 포트 번호를 생성하는 역할을 수행하게 된다.

VOD 서버에 처음 접속을 개시하고, 단말 노드(Head-and-Node : 이하 HEN)을 통해서 전달되는 사용자의 요구가 동일한 아이템인지 아닌지를 검사한다. MAS는 사용자들이 VOD 서버에 동일한 아이템을 요구할 때 멀티캐스트를 지원하기 위해 준비상태를 유지한다. VOD 서버는 유니캐스트로 각 사용자에게 텍스트 페이지를 전송한다. 만약 사용자 중에 하나가 멀티미디어 데이터를 요구한다면, MAS는 멀티캐스트 주소와 포트 번호를 발생시킨다. VOD 서버에서 멀티캐스트 에이전트(Multicast Agent : 이하 MA)와 HNET에 있는 MAS는 HNET와 VOD 서버 사이에서 멀티캐스트 터널 네트워크를 생성한다. 그러면 VOD 서버로부터 요구된 멀티미디어 스트림은 멀티캐스트 터널을 통해서 HEN에게 전송된다.

2.2 VOD 서버

VOD 서버는 사용자들로부터 요청된 비디오 데이터의 서비스를 제공하는 소스 장치로써 웹을 통하여 스트리밍 서비스를 즉시 수행한다. 서버는 스케줄러로부터 전달되어진 특정한 멀티캐스트 그룹에 요구된 멀티미디어 스트림을 전송한다. VOD서버는 스케줄러로부터 멀티캐스트 주소와 포트 번호를 받자마자 비디오 스트림을 전송하고, 사용자들은 멀티캐스트 주소와 포트 번호를 받은 후에 자동적으로 부여된 멀티캐스트 그룹에 참여한다.

2.3 사용자 프로그램

사용자는 비디오 주문형 서비스를 요구하기 위해서 HEN을 통하여 MAS에 접근하고, 또한 스케줄러로부터 멀티캐스트 주소와 포트 번호를 부여받는다. 그 다음 단계로 사용자는 VOD 서버로부터 전송된 비디오 스트림을 전송받기 위하여 멀티캐스트 그룹에 참여한다. 첫 번째 사용자는 WMS(Window Media Service)의 도움으로 미디어 플레이어를 구동한다. 동일한 멀티캐스트 그룹에 참여하는 다른 사용자는 자신의 버퍼에 스트림을 저장한다. 그러면 그것들은 사용자가 보기 원하는 즉시 버퍼로부터 요구된 비디오 스트림을 보기 위해 윈도우 미디어 플레이어를 구동시킨다.

사용자가 서비스를 요청하기 위해서는 스케줄러에게 원하는 아이템을 요청해야 한다. 사용자가 요청한 비디오가 서버에 존재하다면 스케줄러는 멀티캐스트 그룹 주소를 발생시켜서 사용자와 서버에게 알려준다. 사용자는 스케줄러로부터 멀티캐스트 그룹 주소를 얻게 되면 스케줄러와 연결되었던 소켓을 닫고, 서버와의 통신을 위하여 새로운 멀티캐스트 소켓을 생성한다. 또한 서버는 멀티캐스트 그룹 주소를 이용해서 사용자에게 데이터를 전송하기 위하여 데이터그램(datagram) 소켓을 생성하고, 사용자는 멀티캐스트 소켓을 소스(source) 주소 필드에 전달 받은 멀티캐스트 그룹 주소를 입력하고 수신을 기다리게 되며, 서버는 전송을 시작한다.

새로운 스케줄링을 사용하여 구현한 서비스는 멀티캐스트 그룹 주소를 스케줄러에게서 받은 사용자는 전송을 담당하는 서버에게서 데이터를 전송 받는 동안에는 서버 메인 프로그램과 세션을 유지하지 않고, 각 그룹에 할당된 전송을 담당하는 서버 쓰레드와 세션을 유지한다. 각 연결을 쓰레드로 분리하고 채널 관리 효율을 높일 수 있기 때문에, 자신이 받고자 하는 서비스가 끝나면 자동적으로 자신의 소켓을 파괴하고, 아이템을 전송했던 서버 쓰레드도 종료된다.

2.4 단말 네트워크

인터넷은 현재 멀티캐스트 전송을 완벽하게 지원하지 않는다. 따라서 본 논문에서는 멀티캐

스트를 지원하기 위해서 HNET를 제안한다. HNET란, ADSL/Cable modem 네트워크와 흡사하다. HNET는 MAS와 약간의 HENs(Head-End-Nodes)와 많은 수의 사용자로 구성되어 있다.

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 LAN망에서 교육용 컨텐츠를 지정해서 제공하는 경우, 또는 회사에서 사내 교육 자료를 제공하는 경우 등이 될 수 있다.

3. 주문형 비디오 시스템의 구현 및 동작

본 논문에서 멀티캐스트 전송을 사용하는 VOD 시스템 구현은 Microsoft사의 WMS(Windows Media Service)와 API을 사용한다. 시스템은 MAS(Multicast Agent Scheduler) 프로그램, VOD 서버, 사용자 프로그램 등 3개의 부분으로 구성되어 있다.

MAS는 메인 스케줄러와 소켓 쓰레드 두 부분으로 구성된다. 사용자의 요구를 받아들이기 위해 서버 소켓을 사용하는 메인 스케줄러는 각각의 요구된 비디오 아이템에게 멀티캐스트 그룹을 생성하고 그룹을 관리한다. 소켓 쓰레드는 사용자의 접속을 유지한다.

MAS 스케줄러는 초기화, 스케줄링, 종료의 3가지 동작 상태를 가지고 있다. 먼저, 초기화 단계에서 MAS는 WMS의 COM library를 초기화하기 위해 **PubPoint()**함수를 생성하고, 요구된 비디오 아이템을 찾는데 필요한 탐색 시간 줄이기 위해 해쉬 함수를 사용해 서버에 저장된 비디오 리스트인 **MovieInfoTable**를 생성한다. 그리고 MAS는 사용자의 요구가 발생할때까지 **Listen()**함수를 유지하고 사용자의 접속을 수용하기 위해 **ServerSocket**를 생성한다. 스케줄러 쓰레드는 스케줄러와 사용자 사이에서 접속을 개설하기 위해서 소켓을 생성하고, 스케줄러와 사용자 사이에 스트림을 교환하기 위해 필요한 준비를 한다. 스케줄링 상태는 새로운 접속을 생성하는 것과 기존 접속에 참여하는 것 두 가지가 있다. 사용자가 서비스를 요구할 때 스케줄러는 요구된 비디오 아이템 번호와 게시점 그리고 서비스를 요구하는 사용자 주소를 포함하는 멤버 함수인 **GenericMServerNsc**를 사용해서 전송하는 비디오 아이템에 대해 멀티캐스트 정보 파일(.nsc)을 생성한다.

3.1 멀티캐스트 에이전트 스케줄러(MAS)

MAS는 비디오 아이템에 대한 사용자의 요구

단계 1. 사용자는 MAS와 서비스 요구를 위해 VOD 서버에 접근한다.
단계 2. VOD 서버는 유니캐스트 전송으로 문자 기반의 페이지 서비스를 시작한다.
단계 3. MAS는 요구된 동일한 서비스에 대한 스케줄링 세션이 존재하는지 하지 않는지를 조사한다. 만약 존재하지 않으면 단계 5.로 분기한다.
단계 4. MAS는 그룹핑을 수행하고 멀티캐스트 그룹과 포트 번호를 생성한다.
단계 5. MAS는 동일한 비디오아이템을 요구한 사용자와 VOD 서버에게 멀티캐스트 그룹 주소와 포트 번호를 전송한다.
단계 6. VOD 서버는 문자 기반의 페이지내에서 비디오 아이템에 대한 요구를 받는다.
단계 7. 시스템은 서버에 있는 MA와 MAS사이에서 멀티캐스트 터널 네트워크를 게시한다.
단계 8. VOD 서버는 부연된 멀티캐스트 그룹에 요구된 아이템을 전송한다.
단계 9. 사용자는 부여된 멀티캐스트 그룹과 포트 번호를 가진 HEN에 조인한다.
단계 10. 만약 첫 번째 비디오 아이템을 요구하는 사용자가 있다면 아이템을 상영하기 위해 윈도우 미디어 플레이어를 구동한다. 그리고 단계 12로 분기한다. 그 밖에 다른 사용자는 그들의 버퍼에 아이템을 저장한다.
단계 11. 각각의 사용자는 적절한 시간에 아이템을 상영하기 위해 윈도우 미디어 플레이어를 구동한다.
단계 12. VOD 서버는 비디오 데이터 전송 서비스를 종료한다.

〈그림 1〉 VOD 서버에서 멀티캐스트 전송을 이용한 멀티미디어 서비스 개요

단계 1. MAS 초기화 : 게시점을 초기화하기 위해 PubPoint를 생성하고 COM library는 MovieInfoTable을 생성한다. ServerSocket을 생성한다.
단계 2. MAS는 사용자가 비디오 서비스를 요구할 때까지 기다린다.
단계 3. MAS는 HEN을 통해서 사용자의 요구 받는다.
단계 4. 스케줄러 쓰레드는 준비한다.
단계 5. MAS는 요구된 비디오 아이템에 대한 스케줄링 세션이 존재하는지 하지 않는지를 조사한다. 만약 존재한다면, 단계 7로 분기한다.
단계 6. MAS는 그룹핑 시간 동안 멀티캐스트 그룹 주소와 포트 번호를 생성하고 그룹핑을 수행하고, 게시점을 생성하고, 멀티캐스트 정보 파일(.nsc)을 생성한다.
단계 7. MAS는 동일한 비디오 아이템을 요청하는 사용자와 VOD 서버에게 멀티캐스트 그룹 주소와 포트 번호를 보낸다.
단계 8. MAS는 VOD 서버에 있는 MA와 멀티캐스트 터널 네트워크를 게시한다.
단계 9. 스케줄링 완료.
단계 10. MAS는 멀티캐스트로 전송된 비디오 스트림을 받고 HENs에게 스트림을 전송한다.

〈그림 2〉 HNET에서 MAS 동작 순서

를 받고 즉시 멀티캐스트 그룹 주소와 포트 번호를 생성하고, 서비스를 요구한 사용자와 VOD 서버에게 그룹 주소와 포트 번호를 전송 하지만 만약 다른 사용자가 VOD 서버에 전송을 종료한 후에 다시 동일한 비디오 아이템을 요구한다면 다른 멀티캐스트 그룹 주소와 포트 번호를 생성한다.

그림 2은 MAS의 수행 순서를 나타낸다. 그림 2에서 MAS 수행 순서에서 단계 5와 7은 MAS의 작동을 보여준다. 만약 동일한 비디오 아이템에 대한 다른 사용자로부터 요구가 없다면, MAS는 멀티캐스트 주소를 생성하지 않게 되고, VOD 서버는 문자 기반의 페이지와 멀티미디어 스트림을 유니캐스트로 전송한다.

3.2 VOD 서버

그림 4와 그림 5는 각각 VOD 서버와 사용자 동작 순서에 대해서 보여준다. VOD 서버는 **Datagram** socket을 생성하고 **Datagram** packet내에 멀티캐스트 그룹 번호와 포트 번호를 삽입하고 사용자에게 비디오 스트림을 전송한다. 서버는 두 부분으로 구성되어 있다. 한 부분은 사용자와 간접적으로 연결을 유지하는 **Datagram** socket과 다른 한 부분은 요구된 비디오 스트림을 보내기 위한 데이터 전송 부분이다.

VOD 서버는 서버 자원이 소비되는지 아닌지를 조사하고 요구된 비디오 스트림이 디스크에 저장되는지 아닌지를 조사하고, 요청된 비디오 스트림의 전송만 담당한다.

특정한 비디오 아이템을 요구하기 때문에, 사용자와 MAS는 소켓을 생성하고 사용자는 서비스 요청을 위해 주 스케줄러에게 데이터를 보낸다. 사용자는 스케줄러 쓰레드로부터 멀티캐스트 그룹 주소와 포트 번호를 받는다. 그러면, 사용자는 부여받은 멀티캐스트 그룹 주소로 멀티캐스트 소켓을 생성하고 VOD 서버로부터 멀티캐스트 터널 네트워크를 통해서 전송된 비디오 스트림을 받기 위해 그룹에 참여한다. 첫 번째 요청 사용자를 제외하고는, 동일한 비디오 스트림을 요구하는 모든 사용자는 각자 그들의 버퍼에 비디오 스트림을 저장하고 다른 사용자들이 비디오 스트림을 요청할 때 즉시 원도우 미디어 플레이어가 구동되고 스트림을 사용자 버퍼로부터 받게 된다. 그러나 첫 번째 요청 사용자는 멀티미디어 터널링 네트워크을 통해서 바로 비디오 스트림을 상영한다.

3.3 스케줄러 프로그램 인터페이스

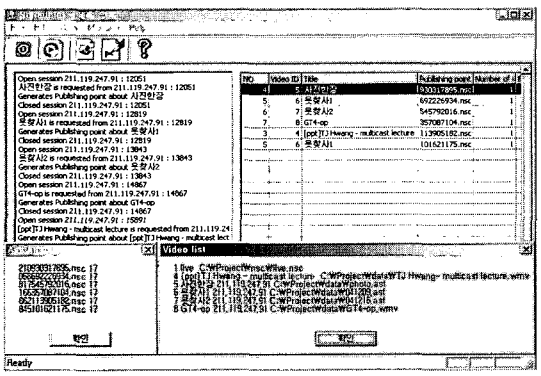
그림 5은 MAS을 관리하는 GUI 콘솔 화면이다. 콘솔은 4개의 정보로 구성되어 있다. 현재

단계 1. MAS는 VOD 서버를 접근한다. : VOD 서버 시작
단계 2. VOD 서버는 MAS 스케줄러로부터 멀티캐스트 그룹 주소와 포트 번호를 받는다. 만약 주소와 포트 번호를 받지 못한 다면, 단계 4로 분기한다; 요청된 비디오 스트림은 유니케스트에 의해 전달되어진다.
단계 3. VOD 서버는 MAS에게 멀티캐스트 터널링 네트워크가 게시되는 시점을 서버에 있는 MA에게 알려준다.
단계 4. VOD 서버는 압축 형태와 전송률을 조사한다.
만약 유니케스트 전송이라면, VOD 서버는 요청된 비디오 스트림을 전송하고 단계 6으로 이동한다.
단계 5. VOD 서버는 멀티캐스트 채널을 통해서 수신된 멀티캐스트 주소와 포트 번호로 비디오 스트림을 전송한다.
단계 6. VOD 서버는 비디오 전송을 종료한다. : 서버 프로그램 종료

〈그림 3〉 VOD 서버 동작 순서

단계 1. 사용자는 윈도우 미디어 플레이어를 시작하고 MAS에게 접속을 비교한다.
단계 2. 사용자는 MAS를 통해서 VOD 서버에 접근한다.
단계 3. 사용자는 접속 후에 MAS에게 비디오 아이템을 요구한다.
단계 4. 만약 MAS가 동일한 비디오 아이템에 대한 또 다른 요구를 받는다면 사용자는 멀티캐스트 정보 파일(.nsc)을 포함한 MAS로부터 멀티캐스트 그룹 주소와 포트 번호를 받는다. 그 밖에 경우는 단계8로 분기한다.
단계 5. 사용자는 MAS에게 접속을 종료한다.
단계 6. 사용자는 VOD 서버로부터 멀티캐스트 터널링 네트워크를 통해서 전송된 비디오 스트림을 받기 위해 멀티캐스트 그룹 주소에 참여한다. 만약 사용자가 비디오 아이템에 대한 첫 번째 요구자라면, 단계 8로 이동한다.
단계 7. 동일한 비디오 아이템을 요청하는 모든 사용자는 그들의 버퍼에 스트림을 저장한다.
단계 8. 윈도우 미디어 플레이어는 멀티캐스트 터널링 네트워크나 버퍼로부터 비디오 스트림을 상영한다. 또는 윈도우 미디어 플레이어는 유니캐스트 채널로부터 직접 비디오 스트림을 상영한다.
단계 9. 윈도우 미디어 플레이어는 상영을 중지한다.

〈그림 4〉 사용자 동작 순서



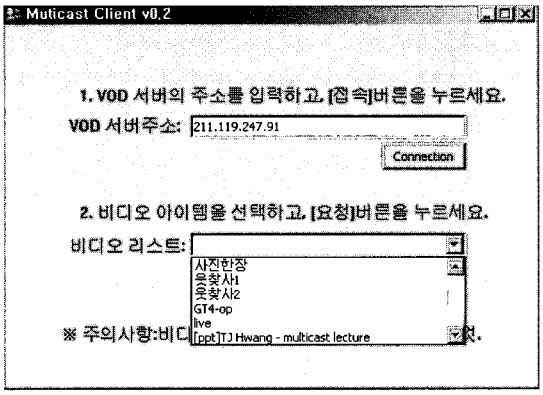
〈그림 5〉 MA-Scheduler

스케줄링되어 상영되고 있는 영화정보, 게시점 할당에 대한 정보, VOD 서버의 영화 리스트 정보, MAS의 동작 기록 정보 등이다. 원도우 미디어 서비스는 영화 아이템을 전달하기 위해 정보 파일(.nsc)와 게시점을 사용한다.

게시점 할당 콘솔 화면은 게시점이 현재 서비스되어지는 영화에 할당된 것을 보여준다. 영화 목록 콘솔 화면은 저장된 위치와 영화 제목을 보여준다. 디버그 콘솔 화면은 사용자 주소, 포트 번호, 사용자 정보와 같이 접속 세션 중에 하나인 세션 상태, 게시점 그리고 세션 종료를 표시한다. 또한 MAS는 서비스를 요청하는 사용자의 주소, 생성된 멀티캐스트 그룹 주소, 포트 번호를 표시하고 사용자 원도우에 요청된 비디오 아이템을 표시한다.

3.4 사용자 프로그램 인터페이스

그림 6은 사용자 응용 프로그램의 실행 결과 화면을 보여준다. 사용자는 사용자 콘솔 화면 그림 6에 멀티캐스트 스케줄러의 서버 IP주소나 DNS 이름을 입력하고 connection 버튼을 누르면 해당주소 서버와 접속이 이루어지고, 접속된 서버에 등록되어 있는 비디오 리스트 목록을 보고 사용자들은 원하는 비디오 아이템 번호를 선



〈그림 6〉 멀티캐스트 사용자 화면

택할 수 있다.

4. 결과 및 성능 분석

본 장에서는 멀티캐스트를 이용한 사용자 기반 VOD 서버의 효율 결과를 분석하였다. 각 비디오 요청 패턴은 VOD 서버로부터 제공되어진 비디오의 인기도에 의존한다. VOD 서버에서 제공되는 N개의 비디오 아이템들의 ID로부터 사용자들이 i번째로 인기가 있는 비디오 아이템을 선택하는 확률은 Zipf 분포를 사용한다.[3] 특정 비디오가 i번째로 선택될 확률 pi는 Z/i이고, Z는 다음으로 나타낼 수 있다.

$$Z = 1 / (1 + 1/2 + 1/3 + \cdots + 1/N) \quad (1)$$

단말-노드들을 통해서 VOD 서버가 서비스를 제공하는 전체 클라이언트들의 서비스 요청 율(request rate)이 λ(분당 서비스 요청 율)일 경우, i번째로 인기가 있는 비디오에 대한 서비스 요청 율은 다음으로 나타낼 수 있다.

$$\lambda_i = \lambda p_i \text{ (여기서 } p_i = Z/i) \quad (2)$$

비디오 서버가 제공하는 가장 인기가 있는 비디오(i=1)는 다른 비디오 아이템들에 비해서 보다 더 높은 가중치(weighting)를 두어야 하기 때문에 시뮬레이션에서 각 비디오 선택에 대한 가중치 매개변수로써 pi를 사용한다.[10]

인기 있는 비디오가 더 높은 확률로 요청될 수 있도록 하였다.[3] 실험에서 사용되는 기본값과 범위 그리고 시스템 매개변수는 (표 1)에 요약했다. 정확한 분석을 위해 이러한 매개변수를 다양화했다.

〈표 1〉 시뮬레이션에 사용된 매개변수

매개변수	기본값	범위
비디오 수	1,000개	N/A
비디오 길이(분)	100분	N/A
서버 대역폭(채널수)	10,000개	N/A
HEN 대역폭(채널수)	100개	N/A
요청율 λ(분당 요청율)	50	10 to 200
HEN의 캐쉬 크기(분)	100분	100 to 1000
HEN 수	10개	N/A

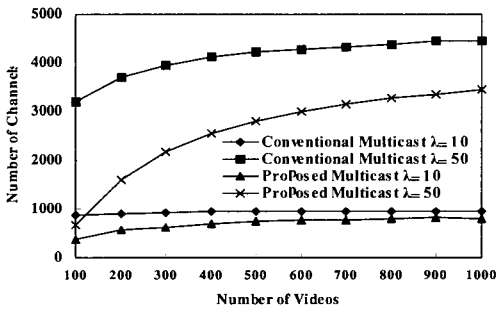
시뮬레이션에서 사용되는 VOD 서버가 제공할 수 있는 아이템의 수는 N=1,000개이고, 서비스 요청율 λ는 분당 10~200회로 가정한다. 여기서 λ는 사용자가 서비스를 요청하는 요청율을 나타내고 Poisson 분포를 따른다. 이를 식으로 나타내면 다음과 같다.

$$f_x(X=k) = \lim_{n\to\infty} \binom{n}{k} p^k (1-p)^{n-k} \cong \frac{\lambda^k}{k!} e^{-\lambda} \quad (3)$$

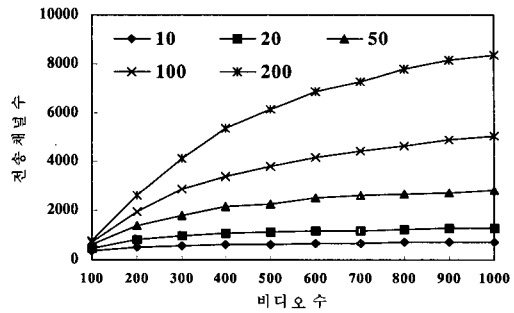
그림 7는 기존 멀티캐스트 기술과 제안한 멀티캐스트 기술에서 사용된 평균 전송 채널 수를 보여주고 있다. 이 경우 HEN의 수는 10개이며 서버는 최대 1000채널을 사용할 수 있으며 각 HEN은 100개의 채널을 사용할 수 있다. 시뮬레이션은 사용자로부터 발생되는 요청률 λ를 분당 10과 50으로 설정하여 수행하였다. 각 HEN은

100분 크기의 캐쉬를 갖고 있기 때문에 HEN에서 저장할 수 있는 비디오 데이터의 최대값은 1,000분이 된다. 전송 채널들은 HEN들에 저장된 비디오 데이터 전부를 재생한 후 생성된다. 따라서 비디오를 저장하는 HEN(nv)에 대한 블록들을 저장하고 있는 HEN의 수이며 nv ≥ 2의 값을 갖는 경우 각 비디오(v)에 대하여 전송 채널의 수를 1/nv로 줄일 수 있게 된다. 따라서 제안된 멀티캐스트 기법하에서 멀티캐스트 그룹 간격(lm)은 HEN에 저장된 블록 수(nv)에 따라 lm부터 nv · lm까지 다양한 값을 갖는다. (그림 8)의 결과로부터 멀티캐스트 기법만을 사용하는 경우와 비교하여 제안된 기법은 요청률 λ = 10인 경우 100편의 비디오에 대하여 약 59%, 1,000편의 비디오에 대하여 15% 정도 채널의 수를 감소시킬 수 있다. 요청률 λ = 50인 경우는 100편의 비디오에 대하여 약 80%, 1,000편의 비디오에 대하여 약 22% 정도 채널의 수가 감소되었다. 서버가 제공하는 비디오의 수가 100편 또는 1,000편인 경우 HEN들은 전체 비디오의 1%과 0.1% 정도의 데이터만을 저장할 수 있다.

그림 8은 요청률 λ = 10, 20, 50, 100, 200인 경우에 대하여 제공하는 비디오 수에 따라서 서버에서 생성되는 전송 채널의 수를 보여주고 있으며 다른 매개변수들은 그림 7의 경우와 동일한 값을 사용하였다. 실험 결과는 서비스 요청률의 증가량에 따라 서버로부터의 평균 전송



〈그림 7〉 서비스 요청률 λ = 10과 50에 대한 평균 서버 전송 채널 수

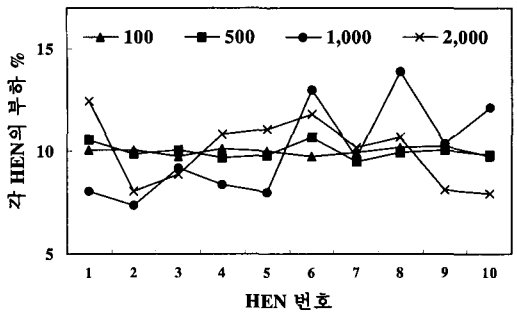


〈그림 8〉 요청률 λ=10, 20, 50, 100, 200 및 비디오 수 N에 따른 서버의 평균 전송채널수

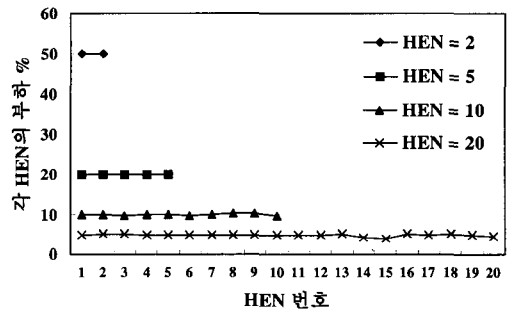
채널수는 선형적으로 증가하고 있음을 보여주고 있다. 동일한 요청률에서 비디오 수 N이 증가하는 경우 사용자들의 요청들은 많은 비디오에 분산되기 때문에 전송 채널의 수는 선형적으로 증가하게 된다. 그러나 비디오 수의 증가는 비인기 비디오의 수를 증가시키게 되며, 이러한 비디오들은 매우 낮은 요청률들을 갖으며 그 값이 거의 비슷하기 때문에 평균 전송 채널수가 지속적으로 증가할 수 없게 되기 때문에 일정이상에서 수렴하는 결과를 나타내고 있다.

그림 9는 요청률 λ=10, 20, 50, 100, 200 인 경우에 대하여 각 HEN에서의 평균 전송 채널수를 보여주고 있다. zipf분포로부터 서버가 적은 수의 비디오를 제공할수록 사용자들의 서비스 요청은 소수의 비디오들에게 집중되며, 제공되는 비디오 수가 커질수록 요청들은 넓게 분산 분포하는 특성을 갖고 있다. 서비스 요청이 집중되면 HEN은 지연 없는 서비스를 제공하기 위하여 다수의 패칭 채널들을 생성하기 때문에, 서버가 제공하는 비디오의 수가 적을수록 HEN에서 생성되는 전송채널의 수는 증가하게 된다. 또한 비디오 편수가 증가할 수록 멀티캐스트들의 병합이 적게 발생되기 때문에 평균전송 채널의 수는 선형적으로 증가함을 보여주고 있다.

그림 9와 그림 10은 HEN의 수와 비디오 수에 따라서 각 HEN의 부하를 보여준다. 그림 10에서 각 비디오에 저장되는 용량은 1분이고



〈그림 9〉 요청률 λ=50 일 때 비디오 수(100, 500, 1000, 2000)에 따르는 각 HEN의 부하



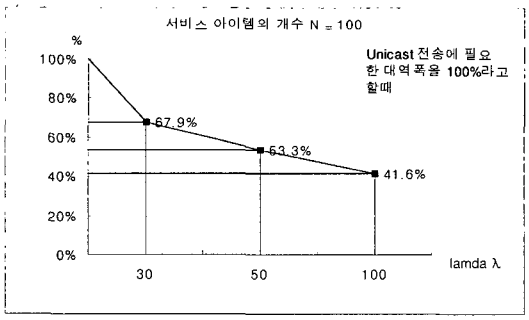
〈그림 10〉 요청률 λ=50 일 때(2, 5, 10, 20) HEN의 수에 따르는 각 HEN의 부하

HEN에 저장되는 전체 분량은 100분 정도이다. 비디오의 수가 100개 일 때 HEN의 부하는 10% 정도로 거의 비슷하다. 그러나 비디오의 수가 1,000개나 2,000개 일 때는 인기 있는 비디오에 따라서 변화한다. 비디오의 수가 더 많이 질수록, 인기가 없는 비디오의 수는 점점 증가된다. 이것은 HEN에서의 캐쉬 확률을 현저히 감소시킴을 의미한다. 결국 HEN에는 인기 없는 비디오가 저장되어 진다.

그림 10에서, 저장된 비디오의 전체 분량은 1,000분으로 고정되어 있다. 또한 각 HEN의 캐쉬 크기는 HEN의 수에 따라서 결정된다.(HEN=2 일때 500분, HEN=5일때 200분, HEN=10일때 100분, HEN=20일때 50분) 실험 결과는 HEN의 부하가 거의 1/(HEN의 수) 만큼 균등있게 분산되었다. 그러므로 제안된 멀티캐스트를 이용한 사용자 기반의 비디오 주문형 서비스 기술은 균형있게 부하를 분산시킬뿐만 아니라 서버의 대역폭을 현저히 줄일 수 있음을 알 수 있었다.

그림 11은 멀티캐스팅 기술을 사용한 알고리듬을 사용하고, 서비스 요청율 λ=30에서 λ=100인 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 대, λ=30인 경우 32.1%의 채널 감소 효과를 가져왔고, λ=50인 경우 46.7%, λ=100인 경우 58.4%에 달하는 채널 감소 효과를 보였다.

그룹 설정 시간이 증가할 경우 채널의 감소 효율이 증가하지만 클라이언트들의 대기시간이 길어지게 되므로 대기시간은 1분 이내로 할 경우 주문형 서비스의 특징을 그대로 유지할 수 있다고 본다.



〈그림 11〉 서비스 요청율에 의한 전송대역폭

5. 결 론

본 논문에서 제안된 시스템은 서버-기반의 주문형 서비스가 아닌 사용자-기반에서 운영되는 주문형 서비스를 구현하였다. 멀티캐스트 에이전트 스케줄러는(MAS)는 VOD 서버 시스템에서 사용자의 비디오 요구에 따라서 즉시 멀티캐스트 그룹 주소와 포트 번호를 발생하는 효과적인

스케줄링을 수행하였다. 또한 MAS는 서비스를 요구하는 사용자와 VOD 서버에게 멀티캐스트 주소와 번호를 전송하면 VOD 서버는 그룹에 요청된 비디오 아이템을 전송하기 시작하고, 사용자는 멀티캐스트 그룹에 자동적으로 조인하여 전송받은 비디오 스트림을 관람할 수 있다.

제안된 주문형 비디오 시스템에서 채널의 수는 서비스 요청율이 각각 30, 50, 100 일 때 유니캐스트를 사용하는 경우보다 32.1%, 46.7%, 58.4% 만큼 감소한다는 것을 알 수 있다.

본 논문에서는 VOD 서버의 부담을 제거한다는 것을 확인하였고, 멀티캐스트 그룹 주소와 포트 번호를 공유하기 위해 사용자-기반의 스케줄링을 통하여 효과적으로 네트워크 자원을 사용할 수 있다는 것을 확인하였다. 또한, 이 시스템은 ADSL/Cable 모뎀 네트워크에서 상업적인 비디오 서비스에 이용될 수 있으리라 기대된다.

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

[4] Carey Williamson, "On Filter Effects in Web Caching Hierarchies," ACM Transactions on Internet Technology, Vol. 2, No. 1, pp. 47-77, February 2002

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

[6] 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.

[7] C. Shahabi and M.H. Alshayeji, "Super-streaming: a New Object Delivery Paradigm for Continuous Media Servers", Journal of Multimedia Tools and Applications, V11, n1, 275-298, May 2000.

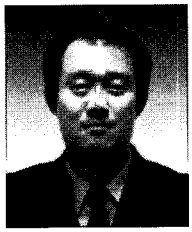
[8] L. Fan, P. Cao, J. Almeida and A.Z. Broder, "Summary Cache: A Scalable Wide-Area Web Cache Sharing Protocol", Proceedings of ACM SIGCOMM'98, pp. 254-265. Technical Report 1361, Computer Sciences Department, Univ. of Wisconsin-Madison, Feb. 1998.

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

[10] Carey Williamson, "On Filter Effects in Web Caching Hierarchies," ACM Transactions on Internet Technology, Vol. 2, No. 1, pp. 47~77, February 2002

◐ 저 자 소 개 ◑

황 태 준 (Hwang Tae June)
1997년 2월 인천대학교 전자계산학과 졸업(공학사)
1999년 2월 인천대학교 전자계산학과 대학원 졸업(공학석사)
1999년 3월~2001년 2월 인천대학교 정보통신공학과 대학원 박사과정 수료
관심분야 : Multicast, VOD, Webcaching, 데이터베이스, 멀티미디어
E-mail : tjhwang@incheon.ac.kr



김 백 현 (Kim Back Hyun)
1993년 2월 인천대학교 정보통신공학과 졸업(공학사)
1993년 8월~1997년 12월 삼성전자 전임연구원
2001년 2월 인천대학교 정보통신공학과 대학원 졸업(공학석사)
2000년 12월~2002년 12월 PNP네트워크 선임연구원
2003년 3월~현재 인천대학교 정보통신공학과 대학원 박사과정
관심분야 : QoS, MANET, 분산처리, 멀티미디어
E-mail : hidesky24@incheon.ac.kr

김 익 수 (Kim Ik Soo)
1977년 동국대학교 전자공학과 졸업(공학사)
1981년 동국대학교 전자공학과 대학원 졸업(공학석사)
1985년 동국대학교 전자공학과 대학원 졸업(공학박사)
1988년 3월~현재 : 인천대학교 정보통신공학과 교수
1993년 3월~1994년 2월 North Carolina State Univ. 객원교수
2004년 3월~2005년 2월 California State University Sacramento 객원교수
관심분야 : Multicast, Network, VOD, Webcaching
E-mail : iskim@incheon.ac.kr