

멀티미디어 정보 전송을 위한 오버레이 멀티캐스트 네트워크 시스템을 기반한 미들웨어 서버에 대한 연구

김중완*, 임노을, 김종율
부산광역시 사상구 주례2동 산69-1 동서대학교 컴퓨터정보공학부
connat@nate.com, immnoeul@nate.com, xmaskjr@gdsu.dongseo.ac.kr

Study on Middleware Server through Overlay Multicast Network System for Multimedia Data Transmission

Jong Wan Kim*, No Eul Lim, Jong Ryul Kim
Division of Computer & Information Engineering, Dongseo University

요 약

인터넷의 급격한 발전에 따라 멀티미디어 정보에 대한 요구가 증대되고 있으며 이와 관련된 이론 및 기술이 발전해 오고 있다. 이런 기술들 중 대표적으로 IP 멀티캐스트, CDN(Content Delivery Network) 등을 들 수가 있다. 하지만 이런 기술들은 사용자가 많이 위치한 곳곳에 하드웨어적인 장치들을 설치함으로써 초기 고정비용이 매우 크게 들어간다는 단점이 있다. 비교적 적은 초기비용을 들여서 유사한 성능의 멀티미디어 전송 서비스를 제공하기 위한 기술들에 대한 연구가 최근 주목 받고 있으며 이러한 연구들 중에 대표적으로 오버레이 멀티캐스트가 있다. 본 논문에서는 이러한 오버레이 멀티캐스트를 기반한 미들웨어 서버를 개발하여 멀티미디어 정보 전송 서비스를 제공할 수 있는 미들웨어 서버의 가능성에 대해 살펴보려고 한다.

1. 서 론

컴퓨터와 인터넷의 발전에 따라 인터넷 사용자의 서비스 요구 사항들도 변하고 있다. 몇 년 전만 하더라도 인터넷을 통해 단순한 텍스트와 이미지에 대한 정보 전송 서비스의 요구가 주가 되었지만 현재에는 텍스트와 이미지뿐만 아니라 다양한 멀티미디어 정보의 전송 서비스의 요구가 급증해오고 있다. 하지만 텍스트와 이미지 전송에 비해 고품질 멀티미디어 정보 전송은 서버에 많은 부하를 주게 되어 서비스에 제한을 가지게 되었다.

이러한 문제를 해결하고자 나온 기술이 IP멀티캐스트와 CDN(Content Delivery Network)이다. IP멀티캐스트와 CDN은 서버에 부하를 줄이면서 고품질에 멀티미디어 정보 전송이 가능하게 해주는 기술이지만 사용자가 많이 위치한 곳곳에 하드웨어적인 장치들을 설치함으로써 초기 고정비용이 매우 크게 요구된다는 단점을 가지고 있다. 이러한 단점을 개선하며 장점을 최대한 살리기 위한 연구가 최근 주목 받고 있으며 이러한 연구들 중에 대표적으로 오버레이 멀티캐스트가 주목 받고 있다. 오버레이 멀티캐스트는 IP멀티캐스트와 CDN에 비해 비교적 적은 초기비용을 들여서 유사한 성능의 멀티미디어 전송 서비스를 제공할 수 있는 기술이다 [3][6].

본 논문에서는 오버레이 멀티캐스트를 기반한

미들웨어 서버를 개발하여 멀티미디어 정보 전송 서비스를 제공함에 있어서의 미들웨어 서버의 가능성에 대해 살펴보려고 한다.

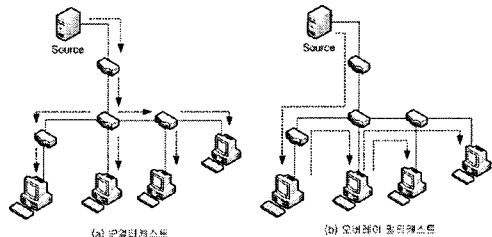


그림1. IP계층 멀티캐스트와 오버레이 멀티캐스트

II. 관련 연구

2.1 오버레이 멀티캐스트(Overlay Multicast)

오버레이 멀티캐스트는 온라인 게임, 실시간 인터넷 방송 등과 같은 동시에 많은 접속자들에게 실시간으로 정보를 송수신해야 하는 멀티캐스트 특성을 지니는 서비스에 대한 수요가 증대되고 있고 기존의 IP 멀티캐스트 기술이 인터넷 환경에서 널리 보급되지 않은 상황에서 기존 유니캐스트 네트워크를 기반으로 효율적으로 멀티캐스트 전송 서비스를 구현해 내기 위해서 제안된 기술이다 [5][6][7].

오버레이 멀티캐스트 서비스는 그림1과 그림 2와 같이 네트워크 층(멀티캐스트 라우터)의 도움 없이 멀티캐스트 서비스 실현하는 중단 호스트 기반의 라우팅 기술을 말한다. 패킷의 복제 및 재전송, 라우팅, 그룹 멤버십의 관리, 그룹 통신에 관한 모든 기능들을 멀티캐스트 라우터가 아니라 중단 시스템에서 수행한다.

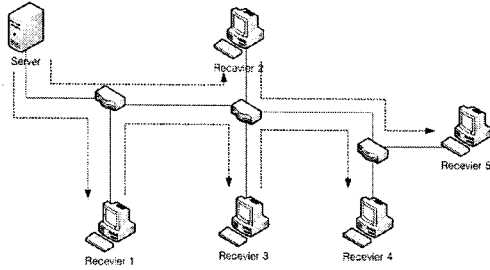


그림 2. 오버레이 멀티캐스트의 가상 토폴로지

오버레이 멀티캐스트의 IP 멀티캐스트와 구별되는 특징으로는 오버레이 멀티캐스트의 토폴로지는 하위의 물리적 토폴로지와는 별도로 방향성을 가진 가상 그래프(directed virtual graph)가 모든 그룹 멤버들 사이에서 유니캐스트 연결 형태로 생성되며 이때 가상 그래프를 최적으로 유지 관리하는 것이 상당히 중요하며 가상 그래프를 구성하는 방식에 따라 여러 가지로 분류할 수 있다. IP 멀티캐스트에서의 멤버십 정보 관리가 멀티캐스트 라우터들에 의해 수행되지만 오버레이 멀티캐스트는 그룹 멤버십 관리용으로 지정된 서버나 송신자 또는 멀티캐스트 세션의 모든 참가자들에 의해 그룹 멤버십 정보 관리가 수행된다.

오버레이 멀티캐스트의 장점은 현재의 멀티캐스트 틈새 시장의 수요를 충족시키면서 지역적으로 멀티캐스트가 지원되는 Intra-AS(Autonomous System)을 활용할 수도 있고 Inter-AS 멀티캐스트를 지원하는데 비교적 자유롭고 기존 인터넷의 라우터에 대한 변경 없이도 라우터에 부하를 주지 않으면서 멀티캐스트 서비스를 빨리 보급할 수 있고 적용하려는 응용에 따라 효율적인 멀티캐스트 전송 트리를 관리할 수 있다.

오버레이 멀티캐스트의 단점은 IP 멀티캐스트에 비해 대역폭 사용이 비효율적 물리적 토폴로지와 오버레이 토폴로지의 불일치에 기인하는 지연은 피할 수가 없으며 그룹에 가입, 탈퇴가 동적으로 이뤄지는 상황에서 안정성의 확보 노력이 필요하다. Intra-AS내의 지정된 서버가 아닌 중단시스템에서 릴레이 서비스를 지원하는 경우에는 보안이

나 개인의 네트워크 자원 사용에 따른 정산 문제 등이 발생할 가능성이 있다.

이런 단점에 대한 방지책은 오버레이 멀티캐스트는 적용하려는 응용에 따라 효율적으로 최적의 전송 트리를 생성, 유지, 관리하기 위한 알고리즘의 개발이 필요하다.

III. 시스템 구축과 구현내용

3.1 클라이언트 시스템 설계

Join과 미디어 요청 송신과 수신 동작 중 첫 번째로 서버에 접속 시에는 자신에 정보와 자신이 미디어를 가지고 있는지 유무를 보낸다. 서버에서는 미디어가 있을시에는 포트번호를 10000번으로 보낸다 하지만 미디어를 가지고 있지 않으면 -1번으로 보낸다.

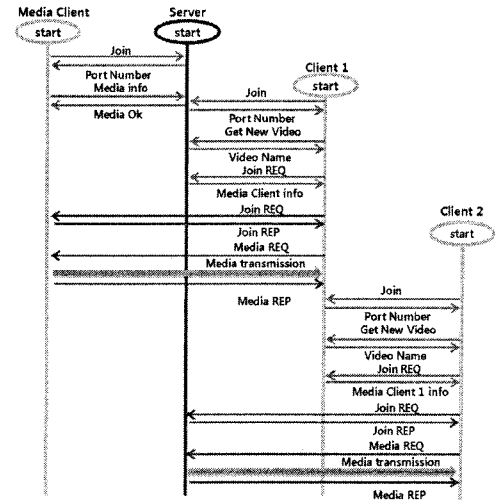


그림 3. 서버에 접속,미디어 요청,송신,수신 동작

미디어를 보고자할 때 GetNewVideo 버튼을 클릭할 경우 미디어가 무엇 있는지 확인하기위한 GetNewVideo 메시지를 보낸다. 서버에서 접속한 클라이언트 중에 미디어를 가진 멤버의 정보를 클라이언트에게 보내게 된다.

전송받은 미디어들 중에 선택하여 Request버튼을 클릭하면 선택한 미디어 이름에 정보를 보고자 메시지 JoinREQ를 보내게 된다. 서버에서는 메시지를 받아 미디어를 전송 할 수 있는 클라이언트를 찾아 미디어 클라이언트 정보를 보내게 된다. 클라이언트에서 받은 정보를 가지고 미디어를 가지고 있는 클라이언트에게 JoinREQ를 보내게 된다. 메시지를 받은 클라이언트는 메시지정보를 저장하고 자신 클라이언트 정보를 JoinREP 메시지 보내게 된다.

JoinREQ메세지를 보낸 클라이언트 받은 메시지를 가지고 멤버에 저장하고 MediaREQ를 보내게 된다. MediaREQ를 받은 메시지는 미디어를 송신을

처리하고 MediaREP를 보내게 된다. MediaREP 메시지를 받아서 미디어 수신 처리한다.

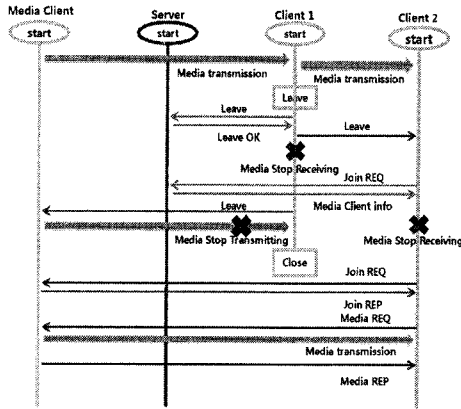


그림 4. Leave, 재요청 동작

Leave를 하는 경우 가장 먼저 서버에게 Leave 메시지를 보낸다. 그리고 자신이 미디어를 보내고 있는 클라이언트에게 Leave 메시지를 보낸다. 메시지를 보낸 후 받고 있는 미디어를 중지시킨다. Leave 메시지를 받은 클라이언트 해당 클라이언트의 정보를 삭제하고 서버에게 미디어를 재요청 JoinREQ 메시지를 보낸다. 서버에서는 요청한 미디어가 있는지 없는지 판단 후 미디어를 보낼 수 있는 클라이언트 정보를 보낸다. Leave하는 클라이언트에서는 미디어를 보내주고 있는 클라이언트에게 Leave 메시지를 보낸다. 미디어를 보내는 클라이언트에서는 미디어를 전송중지를 한다. Leave하는 클라이언트는 정보를 삭제하고 스레드를 종료 시키고 종료가 된다. 미디어를 재요청한 클라이언트는 서버에게 받은 정보로 미디어 전송 가능한 클라이언트에게 JoinREQ 메시지를 보낸다. 미디어 전송 가능한 클라이언트는 JoinREQ 메시지를 받고 정보를 저장하고 포트번호와 기타 정보를 JoinREP 메시지로 보내게 된다. 메시지를 받은 클라이언트는 받은 정보로 미디어 요청하는 MediaREQ 메시지를 보내고 받은 클라이언트는 미디어를 전송하고 MediaREP 메시지를 보낸다. MediaREP 메시지를 받아 미디어 수신을 처리한다.

3.2 미들웨어 서버 시스템 설계

미들웨어 서버 동작은 우선 서버의 시작은 시작 버튼을 통해서 이루어진다. 소켓 및 스레드를 생성하여 클라이언트의 접속을 기다리게 되며 클라이언트의 접속이 이루어지면 메시지를 주고받아 처리한다. 최초 접근시 클라이언트의 정보(아이피, 아이디, ...)를 얻어내고 해쉬테이블에 저장하기 위해 고유 키값을 얻게 된다. 그리고는 얻어낸 정보를 멤버리스트에 저장한다. 여기까지가 참여시의 처리 흐름이다.

참여를 마무리하게 되면 다음으로 받게 되는 메

시지가 미디어 소스를 가지고 있는 상황이면 미디어 정보를 전달 받게 된다. 이 과정은 미디어 소스가 존재시만 이루어지며 정보는 각각 분석되어 저장된다. 클라이언트가 소스의 미디어를 받기를 원하게 되면 제일 먼저 참여하고 있는 멤버의 정보를 요구한다. 이 메시지를 받게 되면 멤버리스트를 분석하여 소스를 가지고 있는 멤버의 정보를 추출하여 넘겨주게 된다. 이제 클라이언트가 소스에 접근을 요구하게 되면 해쉬테이블에 저장된 소스의 정보를 분석하여 불러야할 부모 클라이언트의 정보를 전송해주게 된다.

탈퇴 시에는 탈퇴의 결과 메시지를 반환해 주게 된다.

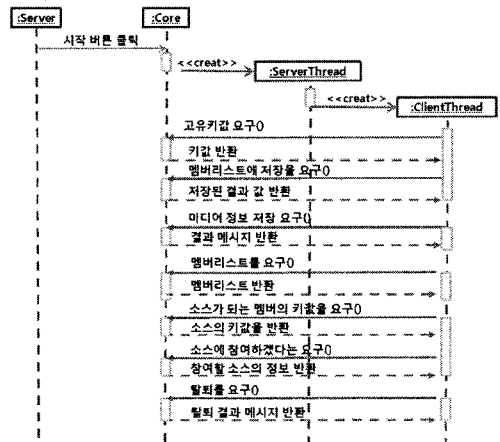


그림 5. 서버 가동 후 클라이언트의 접속/탈퇴의 가장 시나리오

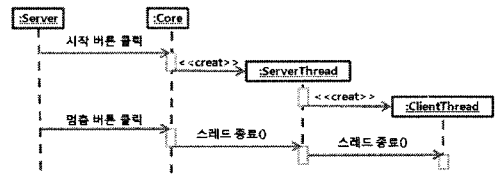


그림 6. 서버의 멈춤 버튼을 클릭시의 처리 과정

그림 6의 과정은 서버의 기능 중 멈춤 기능이다. 즉 서버의 가동을 중지할 때의 처리인데 현재 진행 중인 스레드의 작업을 종료 시키게 된다.

또한 두 개의 기능이 더 있는데 현재 해쉬테이블의 저장된 상황을 보여주는 기능과 프로그램을 완전 종료 시키는 기능이 추가로 구현된다.

IV. 연구 결과 및 실행과정

시스템을 개발하기 위해 본 연구에서는 Java의 JMF 라이브러리를 이용하였다. 멀티미디어의 정보 전송 기능은 JMF 라이브러리에서 지원하는 RTP를 이용하여 개발되었다 [8].

우선 그림 7에 나타난 것과 같이 미들웨어 서버를 실행 시키고 Star 버튼을 누르면 미들웨어 서버가 시작된다. 그리고 클라이언트의 접속을 기

다리게 된다.

클라이언트를 실행시키고 Join버튼을 누른다. 그리고 서버의 IP주소와 자신의 아이디 그리고 미디어의 유무를 체크한다. 체크 후 미디어를 보고자 하면 GetNewVideo 버튼을 클릭하고 원하는 미디어를 선택하면 미디어를 전송 받을 수 있다.

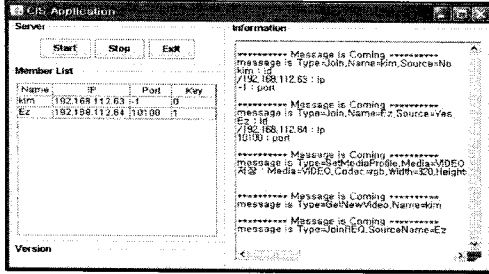


그림 7. 미들웨어 서버 동작 화면

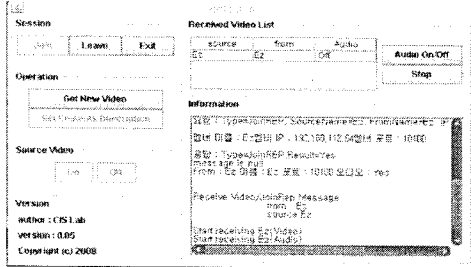


그림 8. 클라이언트 프로그램 실행화면

그리고 멀티미디어 정보를 송신하는 클라이언트에서 자신의 미디어를 보고자 하면 Source Video에 있는 On버튼을 클릭 하면 자신이 송신 중인 멀티미디어 정보를 화면에 표시하여 확인할 수 있다.

V. 결론

시스템의 구현 결과 하드웨어의 교체 비용을 들이지 않고 서버의 부하를 최소화 시키며 멀티미디어 전송에 성공 할 수 있었다. 하지만 많은 개선 사항이 남아있다. 먼저 JMF의 입출이 느리다는 점과 더 이상 Java 진영에서 JMF 개발을 하지 않는다는 점이다. 그리고 미들웨어 서버에서의 멀티미디어 전송 트리 구성에 문제점을 안고 있다. 이와 같은 점을 보완하는 추가 연구가 이루어지면 IPTV 및 화상회의 시스템 그리고 개인방송 시스템, 온라인 강의 시스템 등 다양한 곳에 응용 가능 할 것이다.

본 연구를 통하여 오버레이 멀티 캐스트에 기반을 둔 미들웨어 서버의 기술이 광범위한 응용 가능성을 엿볼 수 있었다.

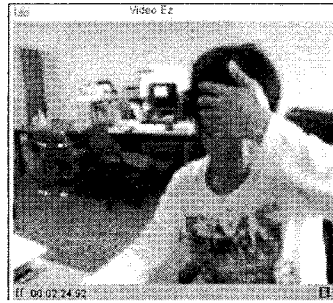


그림 9. 클라이언트에서 미디어 전송 받고 있는 화면



그림 10. 멀티미디어 정보의 송신 확인

참고문헌

- [1] Yi-Tae Cho, Gi-Sun Yang, Seung-sang Sun, Young-ik Eom "Overlay Multicast Scheme with Degree of direction" KIISE, pp.595-597 Vol.32, No.1, July, 2005.
- [2] Dong-Jae Park, Jae-Deuk Sung, Dong-Su Lee, Young "Dynamic Streaming Network using Distributed Clustering" KIISE, pp.413-416, Vol.32, No.2, Oct. 2007.
- [3] CDN service : <http://www.kisdi.re.kr/imagdata/pdf/10/1020080101.pdf>
- [4] K. A. Hua, M. A. Tantaoui, and W. Tavanapong, "Video Delivery Technologies for Large-Scale Deployment of Multimedia Applications," Proceedings of the IEEE, pp.1439-1451, Vol. 92, No. 9, Sep. 2004.
- [5] Relayed MultiCast Protocol (RMCP), <http://ectp.etri.re.kr/>
- [6] J. Park, "Overlay Multicasting," 11th Korea Internet Conference (KRNet 2003), June 2003.
- [7] J. Park, "Topology Management in Overlay Multicast," 12th Korea Internet Conference (KRNet2004), June 2004.
- [8] Java Media Framework (JMF) Library, <http://cs.uccs.edu/~cs525/jmf/jmf.html>