

멀티캐스트 기반의 RTP/RTSP를 이용한 멀티미디어 스트리밍 시스템에서 VCR 기능의 구현

이좌형[○] 최연욱 방철석 김병길 정인병
강원대학교 컴퓨터 정보통신공학과 시스템 소프트웨어 연구실
{jhlee[○], muchoi, csbang, bgkim }@snslab.kangwon.ac.kr, ibjung@kangwon.ac.kr

Implementation of VCR functions for Multimedia Streaming System with RTP/RTSP based on Multicast

Joa Hyoung Lee[○], MyunUk Choi, Cheolseok Bang, Byounggil Kim, Inbum Jung
Dept. Computer Information & Telecommunication Engineering, Kangwon Univ

요 약

VOD 시스템은 실시간으로 사용자의 요구를 받아들여 동영상 서비스를 제공하는 시스템으로서 크게 서버를 제공하는 서버와 사용자의 요구를 처리하는 클라이언트로 구성된다. VOD 서버와 클라이언트 설계 시 다양한 요소들을 고려하여야 하는데 그 중에서도 서버와 클라이언트를 연결해주는 네트워크가 큰 비중을 차지한다. 본 연구에서는 그 동안 제안된 다양한 멀티캐스트 기반의 VOD시스템들 중에서 네트워크 부하가 적은 Batching방식을 이용하는 VOD시스템을 설계하고 구현하고자 한다. 실시간 스트리밍 서비스를 위해 개발된 RTP와 RTSP를 도입하여 멀티캐스트 서비스의 안정성을 높이고자 하였다. 일반적인 재생을 위한 데이터와 빠른 재생을 위한 데이터 모두 멀티캐스팅 함으로써 클라이언트의 버퍼요구량을 줄이고자 하였다.

1. 서 론

최근 컴퓨터 기술이 빠르게 발전하고 널리 보급되면서 텍스트 위주로 작업을 하던 어플리케이션들의 비중이 낮아지고 멀티미디어 데이터를 처리하는 어플리케이션들의 비중이 점차 증가하고 있는 추세이다. 다양한 멀티미디어들 중에서 영화와 같은 동영상 멀티미디어를 다루는 프로그램들은 멀티미디어 응용 어플리케이션들 중에서 큰 비중을 차지하고 있으며 실생활에서 널리 사용되고 있다.

컴퓨터 기술의 발전과 더불어 급속하게 발전하는 네트워크 기술이 실생활에 보급되어 컴퓨터들이 독립적으로 존재하는 것이 아니라 서로 연결된 네트워크를 이루기 시작하면서 네트워크를 통한 멀티미디어 데이터의 교환이나 전송과 같은 서비스들이 활성화되고 있는 추세이다. 네트워크와 동영상을 이용하는 대표적인 어플리케이션으로서 실시간으로 동영상 스트리밍 서비스를 제공하는 VOD (Video-On Demand)가 있다. VOD 시스템은 실시간으로 사용자의 요구를 받아들여 동영상 서비스를 제공하는 시스템으로서 크게 서비스를 제공하는 서버와 사용자의 요구를 처리하는 클라이언트로 구성된다.

VOD 서버와 클라이언트 설계시 다양한 요소들을 고려하여야 하는데 그 중에서도 서버와 클라이언트를 연결해주는 네트워크가 큰 비중을 차지한다. 데이터양이 적은 텍스트에 비해 멀티미디어는 데이터양이 매우 방대하며 특히 VOD의 서비스 대상인 동영상의 경우 다른 멀티미디어 데이터보다도 훨씬 크기가 크다. 방대한 크기의 동영상데이터를 실시간으로 네트워크를 통해 전송시 많은 네트워크 대역폭을 필요로 하기 때문에 네트워크가 VOD시스템의 성능을 크게 좌우한다.

서버와 클라이언트간에 1:1로 설정된 채널을 통해 데이터를 전송하는 유니캐스트 방식의 경우 안정적인 서비스를 제공할 수 있으며 다양한 VCR기능을 쉽게 구현할 수 있지만 클라이언트의 수가 증가하면서 네트워크 사용량도 비례하여 증가하여 네트워크가 성능향상의 걸림돌이 되는 단점이 있다. 이에 반해

하나의 서버와 다수의 클라이언트가 1:N으로 연결되어 다수의 클라이언트가 동시에 같은 서비스를 제공받는 멀티캐스트의 경우 VCR기능 같은 부가적인 사용자의 요구를 충족시키기는 어렵지만 네트워크 사용량이 연결되는 클라이언트의 수와 상관없이 일정하다는 장점이 있다[1].

멀티캐스트를 이용하는 다양한 VOD시스템들이 제안되어져 왔으며 많은 경우 Batching 방식에 근간을 두고 있다고 볼 수 있겠다[1]. Batching방식은 동일한 동영상에 대한 여러 사용자들의 요구를 묶어서 서비스 하는 방식으로서 사용자가 서비스를 받기 위해 일정시간 기다려야 한다는 단점이 있지만 유니캐스트 방식에 비해 네트워크 사용량을 크게 줄여준다. Batching 방식의 단점을 해결하기 위해 Patching[2]과 같은 다양한 기법들이 제안되어졌는데 스트리밍의 시작부분은 유니캐스트 방식을 적용하고 일정시간이 지난후에는 멀티캐스트 방식을 적용하는 방식들이다. 이 경우 사용자에게 즉각적으로 서비스를 제공할 수 있다는 장점이 있지만 각각의 사용자별로 할당되는 초기의 유니캐스트 채널이 서버의 네트워크에 큰 부담을 줄 수 있다. 본 연구에서는 그 동안 제안된 다양한 멀티캐스트 기반의 VOD시스템들 중에서 네트워크 부하가 적은 Batching방식을 이용하는 VOD시스템을 설계하고 구현하고자 한다.

실시간 스트리밍 서비스를 위해 개발된 RTP[6]와 RTSP[5]를 도입하여 멀티캐스트 서비스의 안정성을 높이고자 하였다. RTP는 적은 오버헤드로 빠르게 실시간으로 멀티미디어 데이터를 전송하기 위해 개발된 프로토콜이다. RTP는 UDP를 기반으로 하여 네트워크의 오버헤드를 줄여주며 시퀀스번호와 같은 정보를 제공하여 안정적인 서비스가 가능하도록 해준다. 동영상 압축 방식이나 네트워크등에 따라 다양한 RTP 프로파일들이 있어 목적에 따라 선택하여 적용할 수 있다. 본 연구에서는 MPEG-1, 2로 압축된 동영상들을 서비스하는 것을 목적으로 하여 해당 프로파일을 적용하여 RTP를 사용한다. RTSP는 스트리밍 서비스의 초기화, 스트리밍 컨트롤 그리고 서비스 종료와 같이 스트리밍 서비스를 제어하는 기능을 제공한다.

멀티캐스트 방식의 큰 단점으로 지적되는 VCR기능을 구현하기 위해 많은 연구들이 있었지만 대부분 클라이언트의 버퍼를 이용하는 방식으로서 클라이언트 시스템에 다량의 저장공간을

- 본 연구는 한국과학재단 목적기초연구(R05-2003-000-12146-0)의 지원으로 수행되었음
- 본 연구는 강원대학교 ITRC 지원으로 수행되었음

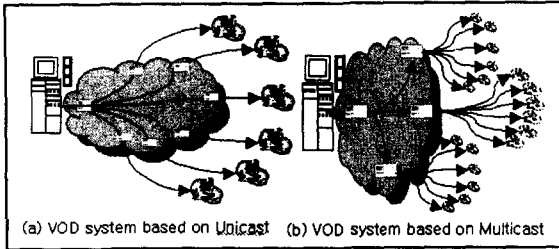


그림 1 유니캐스트(a)와 멀티캐스트(b)기반의 VOD시스템

필요로 한다[3][4]. 본 연구에서는 클라이언트에서의 버퍼 요구량을 줄이면서 VCR기능을 제공하기 위한 방안으로서 주기적으로 멀티캐스팅되는 일반적인 동영상 데이터를 처럼 VCR기능을 위한 데이터들도 주기적으로 멀티캐스팅하는 방식을 제안하고자 한다. 이 경우 네트워크에 부하가 증가하지만 클라이언트에 버퍼공간을 추가하지 않으면서 VCR기능을 제공하는 장점이 있다.

2. Batching 방식

VOD시스템은 크게 TVOD(True VOD)와 NVOD(Near VOD)로 구분된다. TVOD는 유니캐스트 방식(그림1의 (a))을 사용하여 사용자에게 VCR기능과 같은 다양한 부가적인 서비스를 제공한다. 이 경우 서비스되는 사용자의 수가 증가하면서 네트워크의 부하도 비례하여 증가하는 단점이 있다. NVOD는 멀티캐스트 방식(그림1의 (b))을 기반으로 하여 사용자별로 VCR과 같은 기능을 제공하기 어려운 단점이 있지만 네트워크 부하를 크게 줄여주는 장점이 크게 부각된다.

VOD시스템에서 멀티캐스트 방식이 적용 가능한 것은 사용자들이 특정시간대에 집중적으로 서비스를 요청하며 사용자들의 요구가 소수의 인기있는 동영상으로 집중되기 때문이다. 사용자들은 일반적으로 평일 저녁시간대나 주말에 집중적으로 서비스를 이용하며 포아송 분포에 따라 서버에 접속하는 것으로 알려져 있다. 서버에 접속하는 사용자들은 다수의 동영상들 중에서 극소수의 인기있는 동영상을 집중적으로 요구하며 이는 Zipf의 법칙을 따르는 것으로 간주된다[1]. 결론적으로 사용자들의 요구는 특정시간대에 특정 동영상으로 집중되는 것이다. 집중되는 다수의 사용자들을 그룹으로 묶어서 집중되는 그룹별로 채널을 할당하여 동영상 데이터를 멀티캐스팅 하게 되면 네트워크 자원을 크게 줄일 수 있게 된다.

Batching방식은 멀티캐스트를 기반으로 하는 대표적인 VOD시스템 방식으로 동일한 동영상에 대한 사용자들의 요구를 그룹화하여 그룹별로 채널을 할당하여 서비스를 제공한다. Batching 방식의 단점으로 지적되는 서비스 지연시간을 줄이기 위해 Patching방식과 같은 다양한 방안이 제시되어 왔지만 동영상의 시작부분을 사용자별로 별도로 전송하므로 서버에 부하가 가중되는 문제점이 있다. 본 연구에서는 Batching방식을 기본으로 하여 VOD시스템을 구성한다.

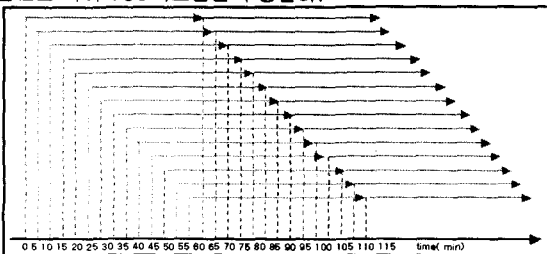


그림 2 주기적인 Batching방식

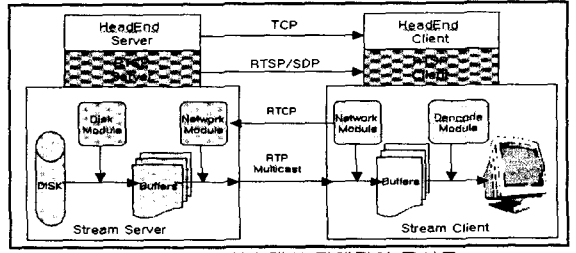


그림 3 VOD시스템의 전체적인 구성도

Batching방식에서 사용자의 요구를 빠르게 처리하기 위하여 MQLF(Maximum Queue Length First)나 FCFS(First Come First Served)와 같은 다양한 동적 스케줄링 방식이 제안되어졌다. 이러한 방식들은 서버에 여유 자원이 생기면 바로 새로운 채널을 할당하여 사용자가 서비스를 제공받기 위해 대기하는 시간을 줄여준다. 하지만 본 연구에서는 빠른 재생과 같은 VCR기능을 일반재생을 위한 채널과 연동하여 제공하기 위하여 동적 스케줄링 대신에 주기적으로 채널을 생성하는 정적인 스케줄링 방식을 채택하였다. 정적 스케줄링 방식에서는 그림 2에서 보듯이 일정한 주기(그림에서는 5분)로 새로운 채널을 할당하여 서비스를 제공한다. 이 경우 사용자가 대기하여야 하는 최대 대기시간(그림에서는 5분)을 미리 알 수 있어 사용자가 서비스 요청을 중단하는 확률을 낮출 수 있다.

3. VOD 시스템의 구성 및 동작

본 연구에서 설계한 VOD시스템의 전체적인 구성은 그림 1에서 보듯이 사용자 요구를 처리하는 HeadEnd Server와 HeadEnd Client, 동영상별로 채널을 관리하는 RTSP Server와 RTSP Client, 그리고 실질적으로 동영상 데이터를 전송하는 Stream Server와 Stream Client로 구성된다. Stream Server는 디스크로부터 데이터를 읽어와서 버퍼에 채우는 Disk Module과 버퍼에 있는 데이터를 네트워크로 전송하는 Network Module로 구성된다. Stream Client는 전송된 데이터를 버퍼에 저장하는 Network Module과 버퍼에서 읽은 데이터를 복호화하여 사용자에게 보여주는 Decode Module로 구성된다.

사용자가 HeadEnd Client를 통해 HeadEnd Server에 접속하면 사용자 인증을 거치게 되며 HeadEnd Servers는 사용자 요청에 따라 영화정보를 제공하게 된다. 사용자가 영화를 선택하여 서비스를 요청하면 HeadEnd Server는 해당 영화를 서비스하는 RTSP Server에 관한 정보를 제공한다. HeadEnd Server와 HeadEnd Client간에는 1:1로 TCP 연결을 설정하여 사용자별로 서비스를 제어할 수 있도록 한다. RTSP Server에 대한 정보를 받은 RTSP Client는 RTSP Server에 연결하여 채널에 대한 정보를 요청한다. RTSP Server는 동영상 데이터의 종류, 서비스 되는 주소, 포트번호, 서비스 속도 등과 같은 채널의 정보를 RTSP Client에게 제공한다. 이러한 정보들은 세션에 대한 정보를 제공하는데 사용되는데 SDP[7]를 이용한다.

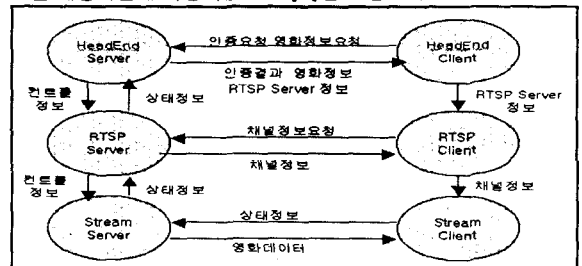


그림 4 VOD시스템의 자료흐름도

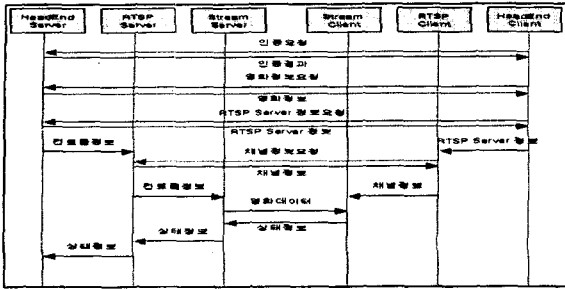


그림 5 VOD 시스템의 Sequence Diagram

Stream Client는 RTSP Server로부터 받은 채널에 관한 정보를 이용하여 해당 멀티캐스트 주소로 RTP세션을 설정하고 동영상 데이터를 수신하게 된다. Stream Server는 MPEG-1압축 방식에 따라 압축되어 디스크에 저장되어 있는 동영상 데이터 메모리상의 버퍼로 읽어 들어 네트워크로 전송한다. 버퍼는 네트워크나 디스크의 작업으로 인해 스트리밍 서비스에 장애가 발생하는 것을 막아주는 역할을 한다.

RTP의 경우 패킷 헤더에 시퀀스 번호를 제공하여 패킷 간에 순서를 맞출 수 있도록 해주며 타이밍 정보 필드에 저장되는 값을 이용하여 다수의 스트림 간에 싱크를 맞출 수 있다. RTP 세션의 전송상태 정보를 전달하고 컨트롤 하기 위해 RTP와 쌍으로 RTCP세션이 설정되며 RTCP세션을 통해 스트림의 품질, 세션에 참가하는 노드의 수와 같은 정보를 수집할 수 있다. RTCP를 통해 수집된 채널의 상태정보는 상위 계층인 RTSP Server로 전송되고 RTSP Server에서 취합된 각 채널별 상태정보는 다시 최상위의 HeadEnd Server로 전송되어져 시스템에 대한 전체적인 정보를 수집하고 컨트롤할 수 있도록 해준다.

그림 4는 UML로 설계된 VOD시스템의 자료흐름도를 보여주며 그림 5는 Sequence Diagram를 보여준다.

4. VCR기능의 제공

멀티캐스트 기반의 VOD시스템의 단점들 중 하나로 VCR기능의 구현이 어렵다는 것이 지적된다. 많은 연구들이 클라이언트의 버퍼를 이용하여 VCR기능을 제공하는 방안을 제시하고 있으나 이 경우 클라이언트에 추가적인 버퍼공간이 필요하다는 단점이 있다. 본 연구에서는 일반재생을 위한 데이터를 그룹화

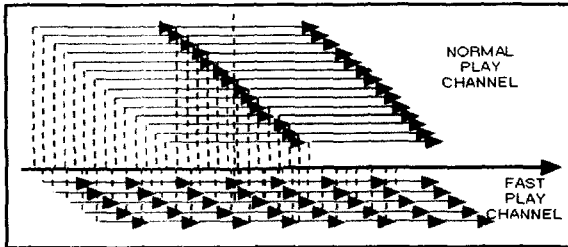


그림 6 일반재생과 빠른 재생의 멀티캐스팅

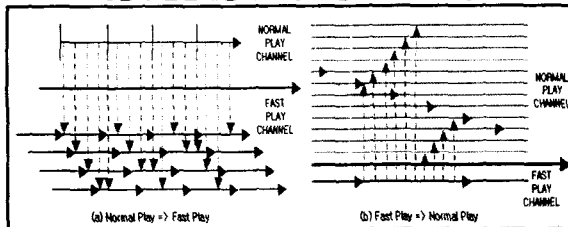


그림 7 일반재생과 빠른 재생간의 전환

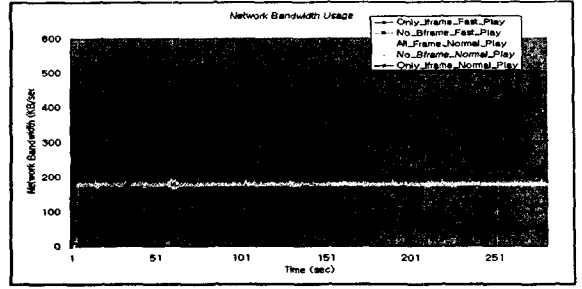


그림 8 플레이 상태에 따른 네트워크 사용량

하여 그룹별로 제공하는 방식을 적용하여 VCR기능을 구현한다. 그림 6에서 보듯이 서버는 일반재생을 위한 데이터와 빠른재생을 위한 데이터를 주기적으로 멀티캐스팅하여 클라이언트는 재생상태에 따라 적절한 채널을 선택하여 재생한다. 사용자가 일반재생에서 빠른재생으로 전환시에는 일반재생의 재생시간과 제일 근접하는 빠른재생용 채널을 선택하여 해당 채널로 변경한다. 반대로 빠른재생에서 일반재생으로 전환시에도 같은 방식으로 전환하면 된다. 그림 7에서 보듯이 채널의 수에 따라 이동할 채널을 결정하게 된다. 채널의 수가 많을수록 보다 나은 서비스를 제공할 수 있지만 그에 비례하여 네트워크를 사용하게 된다는 단점이 있으므로 적절한 절충이 필요하다. 그림 8은 MPEG D 1으로 압축된 동영상데이터를 서비스시 필요한 네트워크 대역폭을 측정할 것을 보여준다. 일반적인 재생의 경우 일정한 양의 대역폭을 사용하지만 빠른 재생을 위해 1 Frame만 전송할 경우 사용되는 대역폭이 크게 증가하고 변동이 있는 것으로 나타나 상황에 따라 적절한 스케줄링 기법이 적용되어야 한다.

5. 결론

본 연구에서는 멀티캐스트를 기반으로 RTSP와 RTP를 이용하는 VOD시스템을 설계하고 구현하고자 하였다. Batching 방식을 적용하여 전체적인 네트워크 부하를 줄이고자 하였으며 RTP와 RTSP를 이용하여 안정적인 서비스가 가능하도록 하였다. 그리고 VCR기능을 위한 데이터를 주기적으로 멀티캐스팅함으로써 클라이언트에 버퍼공간을 추가하지 않으면서 VCR 기능을 제공할 수 있도록 하였다.

6. 참고문헌

- [1] Huadong Ma, Kang G. Shin "Multicast Video-on-Demand Services," ACM Computer Communication Review, vol. 32, no. 1, January, 2002
- [2] Hua, K., Chai, Y. and Sheu, S., " Patching A Multicast Technique for True Video-on-Demand Services" , Proc. Of ACM Multimedia' 98, Sept, 1998.
- [3] Kevin C. Almeroth and Mostafa H. Ammer. "Providing a scalable, interactive video-on-demand service using multicast communication." Technical Report GIT-CC-94/36, Georgia Institute of Technology, Mar 1994.
- [4] A. Dan, P. Shahabuddin, D. Sitaram, and D. Towsley, " Channel Allocation Under Batching and VCR Control in, Movie-on-Demand Servers IBM Res. Rep.", Yorktown Heights, NY, RC19588, 1994.
- [5] <http://www.rtp.org/>
- [6] Colon Perkins, "RTP Audio and Video For the Internet", Addison Wesley
- [7] M. Handley, V. Jacobson. "SDP: Session Description Protocol", RFC2327. April 1998