

오버레이 멀티캐스트를 위한 HDTV 중계전송 시스템 설계 및 구현

정회원 손승철*, 객용원*, 준회원 허권*, 정회원 이형옥*, 남지승*

Design and Implementation HDTV Relay Transmission System for Overlay Multicast

Seung-Chul Son*, Yong Wan Kwag* *Regular Members*, Kwon Heo* *Associate Member*
Hyung Ok Lee*, Ji-Seung Nam* *Regular Members*

요 약

최근 IP 멀티캐스트의 대안으로 제시되고 있는 오버레이 멀티캐스트는 응용계층에서 라우팅을 실시하는 기법으로서 하드웨어의 연산능력과 네트워크의 발전으로 많은 설득력을 얻고 있다. 이러한 오버레이 멀티캐스트 환경에서 HDTV급 영상을 실시간으로 서비스 해주기 위해서는 시스템 자원과 네트워크 대역폭을 효율적으로 사용해야 하며, 특히 응용계층에서 발생할 수 있는 지연시간과 지터를 고려하여 시스템을 설계해야 한다. 본 논문에서는 기존의 오버레이 멀티캐스트 프로토콜에 의해 구성된 세션 내에서 HDTV를 실시간으로 방송하기 위한 서버와 클라이언트를 설계 및 구현한다. 방송서버는 TV 튜너, HDTV 캠코더 그리고 MPEG-2 TS 파일등을 이용해 서비스를 수행하고 그룹을 구성하는 각 클라이언트들은 수신한 데이터를 자신이 관리하는 클라이언트들에게 중계 전송한다. 이때 클라이언트들은 자신의 지연시간과 네트워크 상태를 주기적으로 서버에게 보고하게 되는데 이러한 정보들은 오버레이 세션을 유지 보수하기 위한 중요한 정보로 사용된다. 제안된 시스템은 DirectShow 기반으로 구현되었고 LAN 테스트베드를 통하여 성능 평가 된다.

Key Words : Relay Transmission, Overlay Multicast, MPEG, Directshow, Transport Stream

ABSTRACT

The overlay multicast that has been recently presented as an alternative for the IP multicast has been getting much persuasion by the computing power of the hardware and the advancement of the network techniques to enforce Routing in application-level. In an overlay multicast, the system resource and the network bandwidth must be utilized efficiently to service real-time HDTV images. Specifically, the system must consider the delay and the jitter that can be incurred at the application-level. In this paper, we implement a server and a client to broadcast HDTV, in the session composed by the existing overlay multicast protocol. The broadcasting server performs the service using a TV tuner, An HDTV camcorder, and files, clients constituting a multicast group relay the received data to other clients. At this time, the information that the clients report periodically, including their delay and the network state, to the server is used as an important information to maintain an overlay session. The implementation is based on the DirectX and its performance is evaluated by the LAN test bed that has been set.

* 전남대학교 컴퓨터공학과 멀티미디어 데이터통신 연구실 (rodem94@empal.com)
논문번호 : KICS2006-11-499, 접수일자 : 2006년 11월 16일, 최종논문접수일자 : 2007년 1월 10일

I. 서론

인터넷 망에서의 고품질 HDTV급 비디오는 현실감 있는 화상회의, 실시간 방송, 고품질의 VOD (Video On Demand) 서비스 등 여러 분야에 적용될 수 있으며 이러한 점에서 최근 많은 주목을 받고 있다. 이러한 서비스들을 수행하기 위해 먼저 해결해야 할 대표적인 문제는 네트워크의 효율성을 높이기 위한 멀티캐스트 문제라고 할 수 있다. 네트워크 계층에서 멀티캐스트 라우터에 의해 수행되는 IP 멀티캐스트가 그러한 문제점의 해결책이 될 수 있으나¹⁾, 라우터의 구현 및 확장, 혼잡 제어, 신뢰성 있는 전송 등의 문제로 인해 실제 인터넷에서는 적용되어지지 못하고 있는 실정이다. 최근 오버레이 멀티캐스트(Overlay Multicast)가 IP 멀티캐스트의 대안으로 제시되고 있다. 오버레이 멀티캐스트는 IP 멀티캐스트와는 달리 기존 라우터들을 변경할 필요가 없이 응용 계층에서 라우팅을 실시하며 하드웨어와 네트워크의 급속한 발전으로 인해 클라이언트의 연산 능력과 네트워크 조건이 크게 향상되었기 때문에 더욱 설득력을 얻고 있다²⁻⁷⁾. 그러나 오버레이 멀티캐스트는 고려되어야 할 몇 가지 문제들을 가지고 있다. 그 중에서 대표적 것이 지연 시간과 지터라 할 수 있는데, 과도한 시간 지연과 지터는 동영상 전송에 있어 실시간성과 화질을 크게 떨어지게 하는 요인이 된다. 이와 같은 상황에서 최소한의 지연 시간과 지터로 실시간성과 고화질을 보장할 수 있는 응용계층에서의 중계 전송 기술은 반드시 필요한 핵심 기술이라 할 수 있다.

본 논문에서는 오버레이 멀티캐스트 상에서 25Mbps급 HDTV의 실시간 방송서비스를 수행하기 위한 중계전송 시스템을 설계 및 구현한다. 시스템은 방송 서버와 다수의 클라이언트로 구성되어 있다. 방송 서버는 HDTV 캡코더, TV 수신 튜너 그리고 파일에서 입력되는 MPEG-2 TS¹⁾ 스트림에 지연시간과 지터를 계산하기 위한 타임스탬프와 순서번호를 각각 추가하여 전송하고, 각 클라이언트들은 수신된 MPEG-2 TS의 비디오와 오디오 정보를 파싱하고 각각 렌더링 함과 동시에 다른 클라이언트들에게 중계 전송한다. 하나의 클라이언트는 소수의 다른 클라이언트들의 세션을 관리한다. TCP/IP

를 이용하여 신뢰적인 전송을 수행하며, 끊임없는 시청을 위해 원형 버퍼를 이용한다. 제안된 시스템은 윈도우 플랫폼상의 DirectX를 이용하여 구현하였으며, LAN 테스트베드의 실험을 통해 성능 평가된다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구를 기술하고 3장에서는 제안된 시스템의 개요와 동작 절차 및 구현된 구조를 설명한다. 4장에서는 테스트베드를 통한 실험 결과를 제시하고 5장에서 결론을 맺는다.

II. 관련 연구

오버레이 멀티캐스트의 기본 개념은 라우터가 수행하는 IP 멀티캐스트 대신, 종단 노드에서 패킷을 중계하는 방법으로 유니캐스트 기반으로 멀티캐스트 라우팅을 수행하는 것이다. 네트워크에서 멀티캐스트를 수행하는 것보다 지연 시간이나 동일한 패킷 수가 증가할 수 있지만, 멀티 캐스트 라우터를 필요로 하지 않아 기존의 유니캐스트망 기반에서 효율적인 멀티캐스트 통신을 제공할 수 있다²⁾. 오버레이 멀티캐스트 프로토콜 관련 연구들로는, DVMRP (Distance Vector Multicast Routing Protocol) 멀티캐스트 라우팅 프로토콜을 적용한 Narada³⁾, 네트워크 상에 scattercast 프록시(proxy)를 설치하여 프록시 간의 유니캐스트 통신을 이용한 ScatterCast⁴⁾, 그리고 "Core" 노드를 중심으로 오버레이 멀티캐스트 트리를 구성하는 CBT(Core Based Tree) 기반의 연구들이 진행 중에 있다. 이러한 CBT 기반의 오버레이 멀티캐스트 프로토콜로는, IP 멀티캐스트를 지원하는 HMTP(Host Multicast Tree Protocol)⁵⁾, DNS를 이용하고 노드들의 중계 능력과 거리를 고려하여 CBT를 구성하는 YOID(Your Own Internet Distribution)⁶⁾, 그리고 호스트들을 계층적으로 정렬하여 CBT를 구성하는 NICE(NICE is the Internet Cooperative Environment)⁷⁾ 등이 연구 진행 중이다. 이러한 오버레이 멀티캐스트 프로토콜을 사용하여 멀티캐스트 그룹에 참가하는 각 노드들은 트리의 구성, 에러복구, 재구성 등을 원활하게 수행하기 위한 기능을 포함하고 있다.

HDTV는 MPEG-2시스템 표준의 TS(Transport Stream)를 사용한다. MPEG-2 TS는 전송오류를 대비하여 짧고 고정된 길이의 패킷으로 구성된다⁸⁻¹⁰⁾. 제안하는 시스템에서는 이러한 MPEG-2 TS를 사용하고, 지연시간과 지터를 계산하기 위해 타임스탬프

1) MPEG-2 시스템의 일부로 전송 오류에 유연하게 대처할 수 있는 다중화 방식 및 동기화 관련 정보 등을 제공함으로써 방송 환경이나 네트워크 전송 등에 적합한 방식이다¹⁾.

와 순서번호 등을 MPEG-2 TS와 함께 패킷타이징하여 TCP/IP를 통하여 전송한다. 지연시간과 지터는 앞서 언급한 오버레이 멀티캐스트 트리를 구성할 때 트리의 깊이와 폭등을 조정하기 위한 중요한 정보로 사용된다.

ActiveMovie²⁾에서 발전한 DirectShow는 멀티미디어를 위해 Microsoft사가 개발한 하나의 아키텍처이다. 스트리밍 비디오 및 오디오를 위해 DirectShow는 시스템의 사운드 및 그래픽 카드에 데이터를 효율적으로 렌더링 하도록 DirectDraw 및 DirectSound를 사용한다¹³⁻¹⁵⁾. 다양한 소스와 포맷 그리고 하드웨어 장치를 지원하기 위해 DirectShow는 컴포넌트로 모듈화된 필터를 사용하며¹³⁾ 제안하는 시스템에서는 데이터의 송/수신 및 파싱을 위하여 이러한 필터들을 설계 및 구현한다.

III. 실시간 방송시스템 모델

3.1 개요

제안 시스템에서 방송 서버는 파일, 캡코더 그리고 TV 수신 안테나를 통하여 MPEG2-TS 패킷을 획득한다. 획득된 MPEG2-TS 패킷은 제안 시스템에서 설계한 MTP(MPEG-2 TS Transport Protocol)로 포장되어 각 클라이언트들에게 전송된다. 각 클라이언트는 수신된 MTP를 자신이 관리하는 클라이언트에게 중계전송하고, MTP 헤더를 떼어내 자신의 out-degree 상태를 세팅하고 서버로 피드백 한다. 이때, 피드백 되는 MTP 헤더에는 서버로부터 송신될 때의 시간정보가 포함되어 있고, 방송 서버는 이 시간 정보를 이용하여 RTT(Round Trip Time), 즉 지연시간을 측정하게 된다. 클라이언트는 MTP 헤더가 제거된 순수 MPEG2-TS 데이터를 역 다중화하고 렌더링 함으로써 오디오와 비디오를 시청하게 된다.

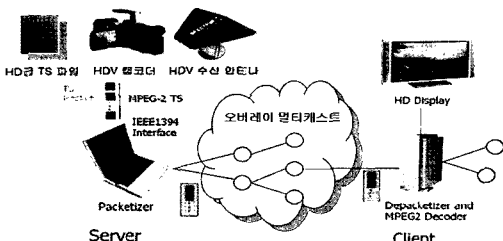


그림 1. 시스템 구조도

2) 마이크로소프트사에서 인터넷 익스플로러에 동영상 지원하기 위하여 개발한 프로그램이다¹³⁾.

게 된다. 그림 1은 앞서 설명한 방송 서버와 클라이언트의 모습을 표현한 그림이다. 각 클라이언트들은 상위 노드로부터의 데이터를 자식노드에게 중계전송함으로써 오버레이 멀티캐스트를 수행하게 된다.

3.2 전송프로토콜

제안 시스템에서는 HDTV의 전송표준으로 채택된 MPEG-2 TS를 사용하는데, 이는 압축된 오디오나 비디오 스트림뿐만 아니라 필요에 따라 사용자 데이터를 다중화 하여 전송에 적합하도록 포장된 비트 스트림이다. 전송을 위한 스트림이기 때문에 전송시 발생할 수 있는 오류에 대비하여 188 바이트의 짧고 고정된 길이의 패킷으로 구성된다¹⁸⁻¹⁰⁾. 오버레이 멀티캐스트로 다수의 사용자가 세션 트리를 구성하였다면 이 세션에 가입한 사용자들은 자신의 부모 노드로부터 받은 실시간 방송 데이터를 MPEG-2 역 다중화기³⁾에 입력시켜 비디오와 오디오를 렌더링 함과 동시에 자신의 자식노드들에게 중계 전송을 해주어야 한다. 이때, 패킷의 지연시간과 지터를 계산하고 MPEG-2 역 다중화기의 샘플 사이즈⁴⁾를 세팅 하기위해 방송서버는 MTP헤더를 MPEG-2 TS에 추가하여 송신한다.

그림 2는 MPEG-2 TS 패킷과 MTP패킷의 구조를 보여주고 있다. 각 필드에 대한 의미를 살펴보면, Client ID 필드는 MTP 헤더를 피드백하는 클라이언트의 식별자이며 Sample Size 필드는 MTP 헤더 이후에 적재될 MPEG-2 TS Payload의 길이를 나타낸다. Hop Count 필드는 피드백을 수행하는 클

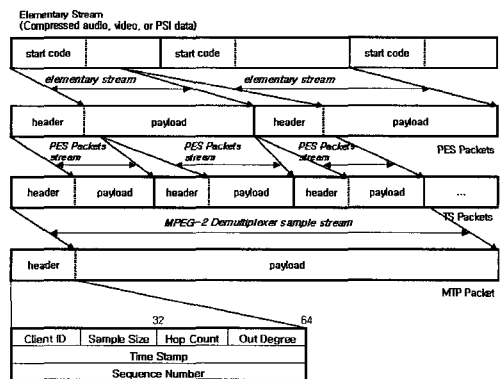


그림 2. 패킷의 구조

- 3) 다중화된 스트림을 비디오와 오디오로 분리시키는 장치로서 본 시스템에서는 DirectX의 기본 역 다중화기를 사용한다.
- 4) Directshow Demuxer 필터의 버퍼링 단위로서 demuxer에 스트림을 입력해주는 소스필터에 의해 정해지며 번들마다 각기 다른 크기를 사용하고 있다.

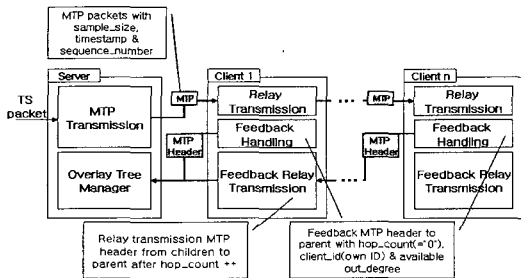


그림 3. 중계전송 및 피드백과정

라이언트로부터 서버까지의 거리를 위한 필드이고 Out Degree 필드는 그 클라이언트의 수용 가능한 자식 노드 수를 나타낸다. Time Stamp 필드는 서버로부터 송신 시점을, Sequence Number 필드는 패킷의 순서번호를 각각 나타낸다.

그림 3은 MTP 패킷의 중계전송 과정과 MTP 헤더의 피드백 과정을 보여주고 있다. 방송서버는 MTP 헤더의 Sample Size, Time Stamp, Sequence Number 필드를 세팅하여 전송한다. 여기서 Sample Size는 캠코더나 TV 튜너의 MPEG-2 TS 인코더에 의해 결정되고 MPEG-2 TS의 패킷 사이즈인 188 바이트의 배수로서 $2^{16}-1$ 바이트를 초과하지 않는 값으로 결정된다. Sequence Number는 패킷의 순서번호로써 단위시간당 패킷의 손실률을 측정하기 위한 필드이다.

클라이언트는 수신한 MTP 헤더의 Client ID, Hop Count, Out Degree 필드를 세팅하여 서버로 피드백한다. 이때 Hop Count는 피드백을 수행하는 클라이언트에 의해 '0'으로 세팅된 후 서버로 피드백 되는 과정에서 상위노드들에 의해 1씩 증가하게 된다. 오버레이 멀티캐스트 트리를 관리하기 위해 주기적으로 측정해야할 정보에는 클라이언트까지의 RTT와 대역폭에 따른 홉 카운트 등이 있으며, 일반적인 오버레이 멀티캐스트 프로토콜은 콘트롤 메시지를 송수신함으로써 필요한 정보를 획득한다^[2-7]. 제안 시스템은 콘트롤 메시지가 아닌 주기적 피드백 기법을 이용하여 오버레이 멀티캐스트 트리를 관리하기 위한 정보를 제공할 수 있다.

3.3 Windows 플랫폼 상에서의 시스템 구현

3.3.1 버퍼의 설계 및 구현

제안하는 모델에서 하나의 클라이언트는 소규모의 다른 클라이언트들을 관리하며, 각 클라이언트 간에는 유니캐스트 방식을 이용하여 데이터를 송수신한다. 전체적인 방송 시스템은 이러한 유니캐스트의 중첩 즉, 오버레이 멀티캐스트를 사용한다.

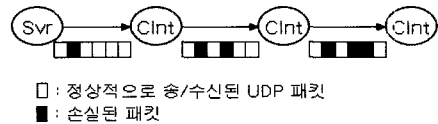


그림 4. UDP 패킷의 송/수신시 문제점

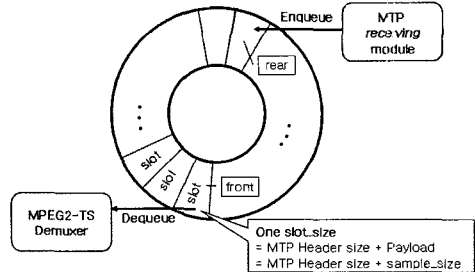


그림 5. 원형버퍼

방송이라는 측면에서 볼 때 전송 프로토콜로써 UDP가 적당하지만 오버레이 멀티캐스트를 적용한 제안 시스템에서 UDP를 사용할 경우 그림 4와 같이 패킷이 노드를 경유하는 동안 손실이 누적될 수 있다. 이러한 패킷 손실의 누적은 클라이언트의 수신 품질을 크게 저하시킨다. 이러한 이유로 제안 시스템에서는 TCP/IP를 채택하였다. 비록 TCP/IP의 사용이 앞서 언급한 패킷 손실의 문제점 해결할 수 있다 하더라도 네트워크에 따른 지터의 발생은 필연적이며 그러한 이유로 응용계층에서 제어할 수 있는 버퍼의 구현은 필수적이라 할 수 있다. 또한 네트워크의 상황이 좋지 않을 경우 버퍼링 시간만큼의 지연시간을 감수 하더라도 미디어를 끊김없이 시청하기 위해서는 꼭 필요한 기능이라 할 수 있다.

그림 5는 제안 시스템에서 구현한 버퍼의 모습을 보여준다. 원형 큐의 형태로 구현함으로써 메모리 복제가 빈번히 일어나는 선형 큐의 단점을 해결하였다. 큐에 입력되거나 큐에서 출력되는 데이터의 단위 즉, 슬롯의 크기는 MTP 패킷의 크기가 되고 큐의 전체 크기는 초단위의 시간으로 설정한다. 가령, 버퍼링하고자 하는 시간을 n초, 하나의 MTP 패킷의 크기를 slot이라고 했을 때, 설정되는 큐의 크기는 다음과 같다.

$$Qsize(MB) = n \times 3.125 - (n \times 3.125) \% slot \quad (1)$$

3.125는 렌더러에 입력해 줘야할 데이터의 양으로서 초당 25Mbit 즉, 초당 3.125MByte를 의미한다. 나머지 연산자를 통하여 큐의 크기를 slot의 배수로 맞추어주는 이유는 큐의 오버플로우와 언더플로우를 방지하기 위해서이다.

3.3.2 PSI Parser

TS 패킷에 포함된 PSI(Program Specific Information)는 수신기에서의 초기화 등을 위해 일정한 시간 간격 내에 주기적으로 전송된다. PSI에는 그림 6과 같이 TS에 포함된 비디오와 오디오 elementary 스트림 정보를 추출할 수 있는 PMT(Program Map Table)와 PAT(Program Association Table)가 포함되어 있다. PMT에는 각 elementary 스트림의 스트림 ID, 프로그램 ID 그리고 타임이 정의 되어 있다. PMT에도 다른 elementary 스트림과 마찬가지로 고유 프로그램 ID가 부여되는데 그러한 프로그램 ID들은 PAT에 정의 되어 있다¹¹⁾.

DirectShow 기반으로 구현된 PSI 파서는 방송 서버 및 클라이언트에 장착된 후 MPEG-2 역 다중화기와 연동되어 그림 7과 같이 PAT와 PMT에 포함된 정보를 이용하여 비디오와 오디오의 스트림을 추출하게 된다. 앞서 언급한대로 PSI 정보는 주기적으로 수신되며 기존에 저장된 PMT와 PAT정보가 새로 수신된 정보와 다르다면 PSI 파서는 이벤트를 발생시키게 되고 어플리케이션은 새로운 정보를 이용하여 비디오와 오디오를 렌더링하게 된다.

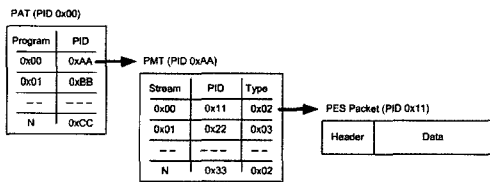


그림 6. PSI 테이블의 구조

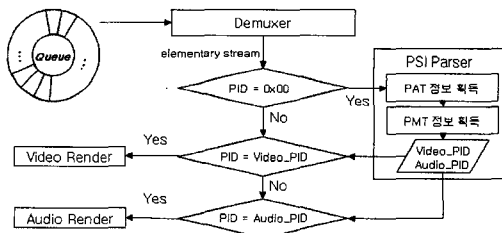


그림 7. 비디오 및 오디오 스트림 추출과정

3.3.3 방송서버

그림 8은 방송서버의 구조를 나타낸다. TV 튜너, HDTV 캡코더 그리고 파일등을 통하여 입력된 MPEG-2 TS 패킷은 MPEG-2 역 다중화기와 PSI 파서를 통하여 렌더링 된다. 이와 동시에 MTP Packetizer에 의해서 순서번호, 타임스탬프 그리고 샘플사이즈가 MPEG-2 TS 패킷에 추가되고 TCP/IP

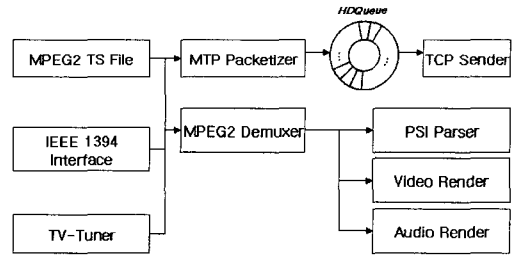


그림 8. 방송서버의 구조

송신모듈에 의해 클라이언트들에게 전송된다.

3.3.4 중계전송 클라이언트

그림 9는 클라이언트의 구조를 나타낸다. 모든 클라이언트들은 중계전송 되거나 방송서버로부터 전송된 MTP 패킷을 수신한다. 수신된 MTP 패킷은 수신즉시 자식노드들에게 전송되고 Status 모듈에 의해 지터가 계산된 후, Queue에 입력된다. MTP Depacketizer는 Queue로부터 MTP 패킷을 출력하고 MTP 헤더를 떼어낸 순수 MPEG-2 TS 패킷을 MPEG-2 역 다중화기에 입력한다. 방송서버와 마찬가지로 PSI 파서와 연동하여 비디오와 오디오를 렌더링 한다.

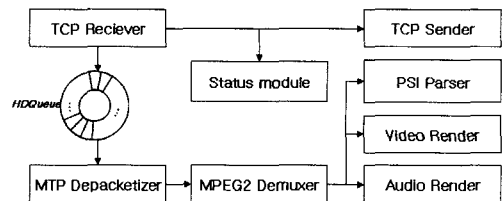


그림 9. 클라이언트의 구조

3.3.5 전송지연과 재생지연

방송 서버가 미디어 데이터를 전송한 후 클라이언트가 시청하기까지 어느 정도의 시간 지연은 불가피 하다. 클라이언트에서 발생할 수 있는 시간 지연은 아래 그림 10에서 보는바와 같이 네트워크에 의한 전송지연(Transport Delay)과 사용자 머신의 성능에 따른 재생지연(Playback Delay)으로 나눌 수 있다¹²⁾.

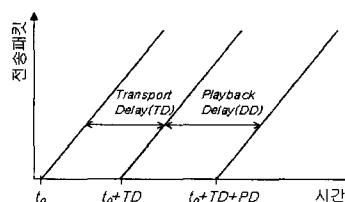


그림 10. 전송지연과 재생지연

방송 서버가 시간 t_0 에 데이터를 전송하였을 때, 클라이언트에서 발생할 수 있는 총 지연시간 D 는 다음과 같이 전송지연 TD 와 재생지연 PD 의 합으로 표현할 수 있다.

$$D = TD + PD \quad (2)$$

아래 그림 11은 서버로부터 2홉까지 거리의 클라이언트에서 발생하는 TD 와 PD 를 네트워크 계층으로 표현한 그림이다. PD 는 데이터를 큐에서 꺼내 역 다중화 하고 디코딩 및 렌더링을 수행하는 시간으로서 TD 에는 전혀 영향을 미치지 않는다.

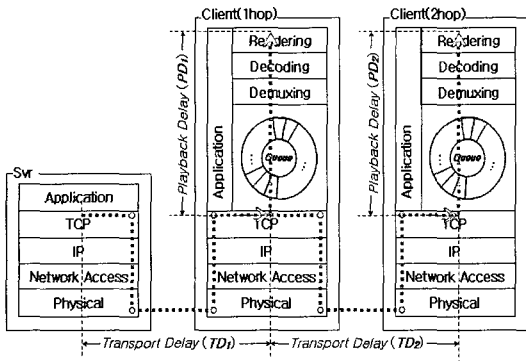


그림 11. 서버로부터 2홉까지의 지연

위의 식(2)을 이용하여 각 클라이언트에서의 총 지연시간을 수식으로 나타내면 다음과 같다.

$$D_1 = TD_1 + PD_1 \quad (3)$$

$$D_2 = TD_2 + PD_2 \quad (4)$$

서버로부터 거리가 n 홉인 클라이언트에서 발생하는 총 지연시간을 일반화 시키면 다음과 같다.

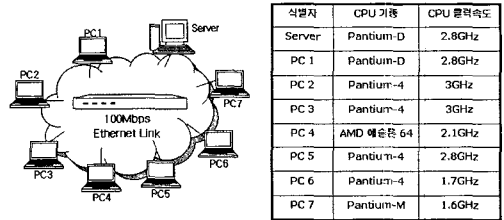
$$D_n = \left(\sum_{k=1}^n TD_k \right) + PD_n \quad (5)$$

전송지연은 방송 서버로부터 여러 상위 노드를 거쳐 중계 전송된 데이터를 클라이언트가 수신할 때까지의 시간이다. 이는 네트워크 상황에 따라 달라질 수 있으며 하위노드에 영향을 미친다. 재생지연은 데이터가 도착한 후부터 역 다중화 되고 디코딩 및 렌더링 과정을 거쳐 실제 시청하기까지의 시간이다. 이는 CPU와 디코더의 성능에 따라 달라질 수 있으며 중계 전송 시 발생하는 전송지연에는 어떠한 영향도 끼치지 않는다^[12].

IV. LAN 테스트베드 실험 결과

4.1 테스트베드의 구성

클라이언트에서의 수신품질을 확인하기 위해 실험용 LAN 테스트베드 환경을 구성하였다. 테스트베드는 방송 서버 한개, 클라이언트 7개, 100Mbps 스위치로 구성된다. 모든 노드는 100Mbps 이더넷 인터페이스로 스위치와 연결되어 있고 영상 및 음성 재생을 위해 윈도우 기반의 DirectX가 설치되어 있다. 각 노드의 CPU 기종과 클럭 속도는 그림 12와 같이 구성된다. 각 클라이언트의 큐 사이즈는 15Mbyte로 설정하였는데 이는 25Mbps의 HD급 영상을 약 5초간 버퍼링 할 수 있는 크기이다. 큐의 슬롯사이즈는 48128Byte 즉, 188×256 Byte로 설정하였다.



식별자	CPU 기종	CPU 클럭속도
Server	Pentium-D	2.8GHz
PC 1	Pentium-D	2.8GHz
PC 2	Pentium-4	3GHz
PC 3	Pentium-4	3GHz
PC 4	AMD 애슬론 64	2.1GHz
PC 5	Pentium-4	2.8GHz
PC 6	Pentium-4	1.7GHz
PC 7	Pentium-M	1.6GHz

그림 12. 테스트베드 및 사용된 PC

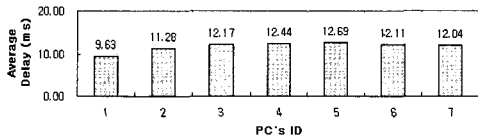
서버로부터 송출되는 테스트용 미디어는 MPEG-2 TS 형식의 파일로써 SD급과 HD급 두 가지를 사용하였다. SD급 영상은 720×480의 해상도, 29.97fps (frame per second), 5Mbps등의 특성을 갖고 HD급 영상은 1920×1080의 해상도, 29.97fps, 25Mbps등의 특성을 갖는다. 네트워크의 상태 측정 및 클라이언트의 피드백 주기는 1초로 설정하였고 큐 사용률 측정주기 또한 1초로 설정하였다. 단, 각 클라이언트의 초기 큐 사용률을 보다 자세히 확인하기 위하여 최초 5초간은 밀리초 단위로 측정주기를 설정하였다. 각 실험은 600초(10분)간 진행되었다.

4.2 실험결과

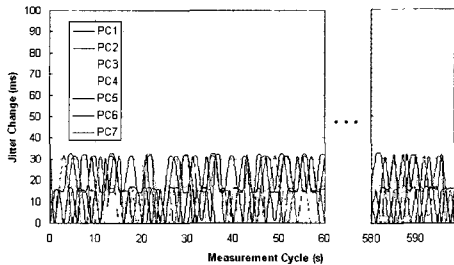
수신품질의 척도에는 일반적으로 손실, 지연 그리고 지터 등이 있다^[6]. 제안 시스템은 TCP/IP의 신뢰적인 전송을 통하여 최소한의 손실을 보장한다. 하지만 제안 시스템이 오버레이 멀티캐스트를 사용하기 때문에 중계 전송으로 인한 지연의 발생은 필연적이다. 또한 버퍼를 통하여 네트워크에서 발생하는 미세한 지터를 해결한다 할지라도 적은 대역폭으로 인해 발생하는 과도한 지터는 실시간 시청에

큰 장애 요소가 된다. 지연과 지터는 오버레이 멀티캐스트 트리를 구성하기 위한 중요한 정보가 되며, 이러한 지연과 지터의 측정을 위해 3가지의 실험을 진행하였다. 먼저 방송서버의 out-degree를 무제한으로 설정하여 모든 클라이언트들이 방송서버에 연결되도록 오버레이 트리를 구성한 후 네트워크의 상태를 측정하였다. 다음으로 모든 노드의 out-degree를 "1"로 설정하여 서버로부터 클라이언트들이 연속적으로 연결되도록 트리를 구성하여 네트워크의 상태를 측정하였다. 마지막으로 방송 서버와 클라이언트를 1:1로 연결하여 각 클라이언트에서의 큐 사용량을 통한 재생지연을 측정 하였다.

1) out degree: 무제한



(a) 평균지연시간

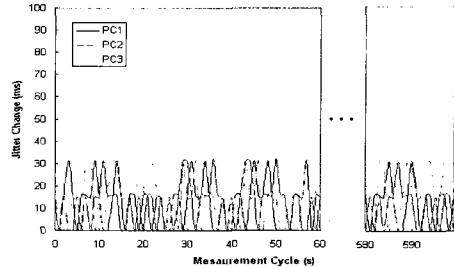


(b) 지터변화

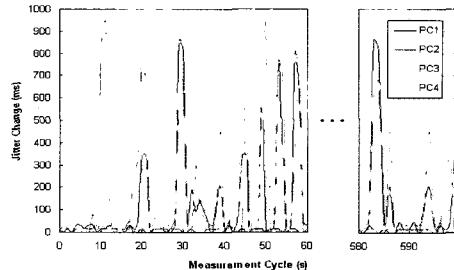
그림 13. SD급 사용자 네트워크 상태

그림 13은 SD급(5Mbps) 미디어를 사용한 경우의 네트워크 상태를 나타내고 있다. 모든 클라이언트는 그림 13.(a)와 같이 평균 전송지연시간이 10ms 내외로 확인 되었다. 이는 오버레이 멀티캐스트 트리 상에서 1홉 거리의 노드간 평균 전송지연을 의미한다. 그림 13.(b)에서 보는바와 같이 모든 클라이언트에서의 지터는 약30ms 이내에서 변화하므로 끊임없는 시청을 충분히 보장한다.

그림 14는 HD급 즉, 25Mbps를 이용한 경우이다. 이론상 방송 서버는 100Mbps의 대역폭을 사용할 수 있기 때문에 4개의 클라이언트에게 서비스 수행이 가능할 것으로 예상 되지만 그림 14.(b)에서 알 수 있듯이 4개의 클라이언트에게 서비스 할 경우 지터의 변화가 심한 클라이언트가 발생하게 되며, 그러한 클라이언트는 실제 미디어의 시청이 거



(a) 3개의 클라이언트



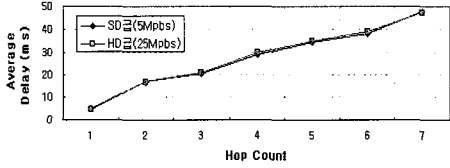
(b) 4개의 클라이언트

그림 14. HD급 미디어 사용자 지터변화(병렬연결)

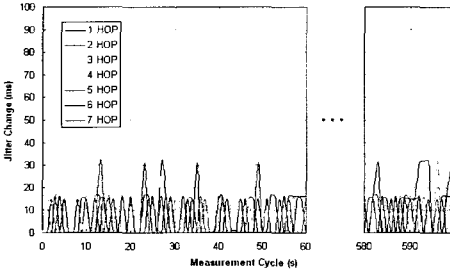
의 불가능하다. 오버레이 멀티캐스트 트리를 구성하는데 있어 이러한 점을 고려하여 out degree를 결정해야 할 것이다. 다시 말해 100Mbps의 대역폭을 지원하는 네트워크 일지라도 최대 out degree를 3으로 하여 최대 약 80%의 대역폭만 사용하도록 구성해야 할 것이다.

2) out degree: 1

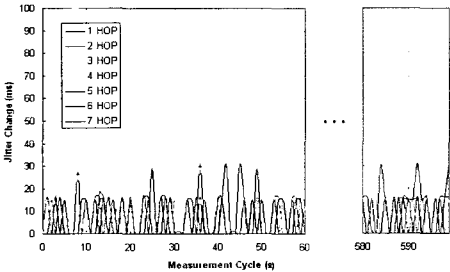
그림 15는 방송 서버로부터 시작해 7개의 클라이언트가 연속적으로 연결 되었을 때의 네트워크 상태를 나타내고 있다. 1홉부터 7홉까지 각 클라이언트의 평균 전송지연을 측정한 결과 SD급과 HD급 모두 한 홉을 거칠 때마다 선형적으로 증가하며 그 형태 또한 유사하다.(그림 15.(a)) 이는 오버레이 멀티캐스트 트리의 깊이와 관련된 전송지연을 대변한다. 깊이가 깊어질수록 전송지연 또한 당연히 증가할 것이므로 실시간성을 최대한 보장해야하는 경우, 이를테면 클라이언트가 방송시청 도중 서버로 시청 내용에 따른 리액션을 수행할 경우 등에는 이러한 전송지연을 고려하여 오버레이 트리를 구성해야한다. 그림 15.(b)와 그림 15.(c)에서 보는바와 같이 지터는 약30ms 이내에서 변화한다. 네트워크 상태나 대역폭이 보장된다면 서버로부터의 거리(홉카운트)에 의한 전송지연과 상관없이 모든 클라이언트는 미디어를 끊임없이 시청할 수 있다.



(a) 평균지연시간



(b) SD급 사용시 지터변화



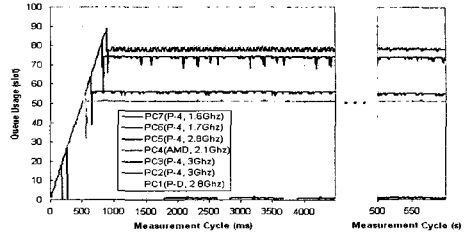
(c) HD급 사용시 지터변화

그림 15. 네트워크 상태(out degree:1)

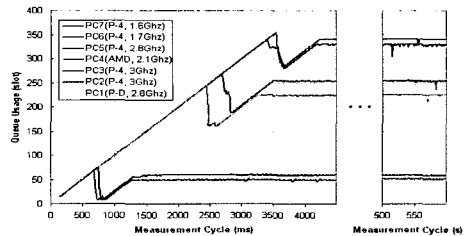
3) 각 클라이언트의 재생지연

그림 16은 각 미디어 데이터에 대한 각 클라이언트의 측정 시간당 큐 사용량을 나타내고 있다. 클라이언트들은 SD급과 HD급 모두 초기화 과정동안 큐에 입력만 실시하고 그 과정이 끝난 후부터 큐의 입출력이 균형을 이루며, 그 시점부터 클라이언트는 미디어를 시청할 수 있게 된다. 수신된 미디어 데이터가 재생되기 위한 초기화 과정에는 미디어에 적합한 Directshow 필터의 검색시간과 검색된 필터간의 연결 및 버퍼 협상시간 등이 포함된다. 초기화 시간은 SD급의 경우 그림 16.(a)와 같이 수 밀리초부터 1초 이내의 시간이 소요됨을 알 수 있다. 또한 큐의 슬롯은 최대 90개까지 사용되고 있다. HD급의 경우에는 그림 16.(b)와 같이 수 밀리초부터 약 4초간의 시간이 소요되고 큐의 슬롯은 최대 357개까지 사용되고 있다. 이처럼 미디어의 종류(SD급, HD급)에 따라 초기화 시간이 다른 이유는 필터를 검색하고 버퍼를 협상하는 시간이 미디어의 프레임률 및 재생률과 관계되기 때문이다. 표 1은 각 PC

와 미디어에 따른 큐의 최초 입출력 균형점을 정리한 표이다. 이러한 균형점이 PC의 성능과 미디어의 재생률(bps)과 관계된 ‘재생지연’이라 할 수 있다.



(a) SD급(5Mbps)



(b) HD급(25Mbps)

그림 16. 각 클라이언트의 큐 사용량

표 1. 각 PC의 재생지연(단위: 초)

	PC1	PC2	PC3	PC4	PC5	PC6	PC7
SD	0.01	0.19	0.28	0.59	0.66	0.85	0.92
HD	1.60	1.28	1.45	3.25	3.51	4.17	4.24

전송지연과 재생지연을 종합해 볼 때 재생지연이 가장 작은 PC1이 서버로부터 가장 먼 거리(7홉)에 위치해 있다하더라도 다른 어떠한 PC보다 실제 시청시간은 가장 빠를 수 있다.

V. 결론

본 논문에서는 오버레이 멀티캐스트 환경에서 HDTV급 고화질의 영상을 실시간으로 서비스 해 주기 위한 중계전송 시스템을 설계 및 구현 하였다. 오버레이 멀티캐스트의 트리를 구성하는 각 클라이언트들은 수신 큐를 이용한 버퍼링을 통하여 원활한 시청이 가능하고, 지식 노드들에게 중계 전송해주는 기능을 가지며, MTP 헤더를 통하여 지터를 측정한다. 또한 MTP 헤더의 피드백 메커니즘을 통하여 지연시간을 측정하고 이렇게 측정된 지연시간은 오버레이 멀티캐스트 트리의 관리를 위한 정보로 사용 될 수 있다.

구현된 시스템을 이용하여 LAN상에서의 방송

서비스는 충분히 수행 가능하다는 사실을 실험을 통하여 확인 하였다. 하지만 WAN으로의 확장을 위해서는 상당히 가변적일 수 있는 지연과 지터를 위해, 네트워크의 상태에 적절히 대응할 수 있는 버퍼의 구현이 필요하다.

참 고 문 헌

[1] P. Sharma, E. Perry, R. Malpani, "IP multicast operational network management: design, challenges, and experiences," Network, IEEE, Volume 17, April. 2003.

[2] A. Ganjam, H. Zhang, "Internet multicast video delivery," Proceedings of the IEEE, Volume 93, Jan 2005.

[3] Y. Chu, S. Rao, S. Seshan, H. Zhang, "A case for end system multicast," Selected Areas in Communications, IEEE Journal on, Oct. 2002.

[4] Y. Chawathe, S. Maccanne, E. A. Brewer, "An Architecture for Internet Content Distribution as an Infrastructure Service," unpublished paper, 2000.

[5] B. Zhang, S. Jamin, L. Zhang. "Host Multicast: A Framework for Delivering Multicast To End Users," In Proceedings of INFOCOM, 2002.

[6] P. Francis, "Yoid: Extending the Multicast Internet Architecture," Unrefereed report, <http://www.aciri.org/yoid>, April. 2000.

[7] S. Banerjee, B. Bhattacharjee, C. Kommareddy, "Scalable Application Layer Multicast," ACM SIGcomm'02, Pittsburgh, PA, August. 2002.

[8] ISO 13818-1:2000, "Information technology: Generic coding of moving pictures and associated audio information: Systems," 2000.

[9] HDV Format, <http://www.hdv-info.org/>

[10] Internet HDTV, <http://www.washington.edu/hdtv/>

[11] P. A. Sarginson, "MPEG-2: Overview of the system layer," BBC R&D Report, 1996.

[12] J Lee and K. Chon, "High-Definition Digital Video Over IP: System Design, Experiments and Applications," in Proc. APAN Network Research Workshop 2004, Cairns, Australia, July. 2004.

[13] DirextX Standard,

<http://www.microsoft.com/windows/directx/>

[14] 신화선, "Directshow 멀티미디어 프로그래밍," 한빛미디어, 2004.

[15] Kris Gray, "The Microsoft DirectX 9 Programmable Graphics Pipeline," Corporation Microsoft, July. 2003.

[16] B. Lowekamp, B. Tierney, L. Cottrell, R. Hughes-Jones, T. Kielmann, and M. Swany. "A hierarchy of network performance characteristics for grid applications and services," Proposed Recommendation Global Grid Forum (GGF), Network Measurement Working Group (NMWG), Jan. 2004.

손 승 철 (Seung-Chul Son)	정회원
2002년 2월 전남대학교 컴퓨터공학과 졸업	
2003년~2004년 (주) 금영 음향연구소	
현재 전남대학교 컴퓨터공학과 석사과정	
<관심분야> 컴퓨터 네트워크, 통신 프로토콜	
곽 용 완 (Yong Wan Kwag)	정회원
1999년 조선대학교 컴퓨터공학과 학사 졸업	
2001년 전남대학교 컴퓨터공학과 석사 졸업	
현재 전남대학교 전자컴퓨터공학과 박사과정	
현재 그리드인터넷 방송연구센터 연구원	
허 권 (Kwon Heo)	준회원
2005년 2월 전남대학교 정보통신공학과 졸업	
현재 전남대학교 컴퓨터공학과 석사과정	
<관심분야> 컴퓨터 네트워크, 라우팅 프로토콜	
이 형 옥 (Hyung Ok Lee)	정회원
2006년 2월 전남대학교 컴퓨터 정보통신공학과 졸업	
현재 전남대학교 컴퓨터 정보통신공학과 석사과정	
<관심분야> 통신 프로토콜, 컴퓨터 네트워크	
남 지 승 (Ji-Seung Nam)	정회원
1992년 Univ. of Arizona, 전자공학과(공학박사)	
1992년~1995년 한국전자통신연구소 선임연구원	
1999년~2001년 전남대학교 정보통신특성화 센터장	
2001년~현재 전남대학교 인터넷창업보육 센터장	
1995년~현재 전남대학교 컴퓨터공학과 교수	
<관심분야> 통신 프로토콜, 인터넷 실시간서비스, 라우팅	