IP 기반의 멀티미디어 스트리밍 전송을 위한 객체 모델 설계

유형석*, 유인호*, 김태형*, 송제호*, 김형진*
*전북대학교 IT응용시스템공학과
e-mail : kim@jbnu.ac.kr

A Design of Object Model for Multimedia Streaming Transmission based on IP

Hyoung-Seok Ryu*, In-Ho Ryu*, Tae-Hyoung Kim*, Je-Ho Song*, Hyoung-Jin Kim*
*Dept. of IT Applied System Engineering, Chonbuk National University

요 약

최근 들어 데이터, 음성 위주의 통신 산업과 영상중심의 방송 산업이 빠르게 융합되고 있다. 따라서 본 논문에서는 IP 기반의 멀티미디어 스트리밍 전송이 가능하도록 네트워크 대역폭의 사용을 최소화하여 특정 채널에 대한 멀티미디어 서비스뿐만 아니라 여러 채널을 동시에 사용자가 서비스 받을 수 있도록 멀티 접근 방법을 제안하고자 한다. 또한 네트워크의 지연을 흡수 할 수 있는 버퍼링 기법, 안정적인 채널 대역폭 할당 및 유지를 위한 객체 모델을 설계 하고자 한다.

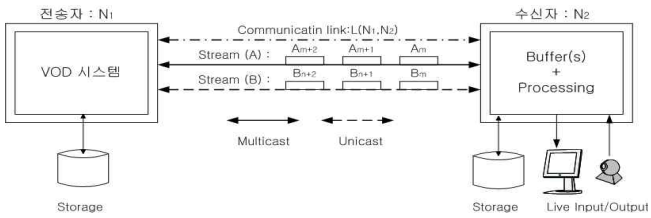
1. 서론

최근 멀티미디어 컨텐츠의 빠른 확산과 초고속 광대역 IP 네트워크 기술이 빠르게 진화되면서 방송과 통신의 기능을 결합한 통방 융합 서비스에 대한 관심이 집중되고 있다. 현재 통방 융합 서비스 형태는 주파수 기반의 CATV (Cable Television), 위성 TV 와 IP 기반의 IPTV 가 대표적이다.

특히 IP 기반의 IPTV는 멀티캐스팅에 대한 모노 접근방식으로 여러 개의 데이터 서비스를 할 수 없다. 이러한 기존의 데이터 방송 시스템은 대역폭 제한 때문에 한 채널 내에 다수의 데이터 방송 서비스 내용을 담을 수 없어 채널 종속적으로 특정 데이터 방송 서비스만을 특정 채널에서 제공하는 구조이다. 따라서 사용자는 채널 시청 중 특정의 데이터 서비스를 사용하기 위해 그 서비스가 가능한 다른 채널로 이동해야 하는 불편함을 겪어야만 한다.

이와 같은 문제점 해결을 위해서 본 논문에서는 IP 기반의 멀티캐스팅 및 유니캐스팅을 혼합한 멀티접근 방법을 제안하고자 한다. 또한 IP 기반의 통신망을 통해 멀티미디어 스트리밍 서비스를 위해 영상과 음성 등과 같은 동적 미디어를 실시간에 수신자에게 전송하는 서비스가 가능하도록 하고자 한다. 수초 단위 이상의 지연으로 전송되어야 할 시간에 데이터가 전달되지 못하게 되면 실시간 데이터로서의 의미를 상실하게 되고, 실시간 서비스도 이루어질 수 없게 된다. 따라서 본 논문에서는 IP 기반의 멀티미디어 스트리밍 전송이 가능하도록 네크워크 지연을 흡수 할 수 있는 버퍼링 기법, 안정적인 채널 대역폭 할당 및 유지를 위한 객체 모델을 설계 하고자 한다.

2. 시스템 설계



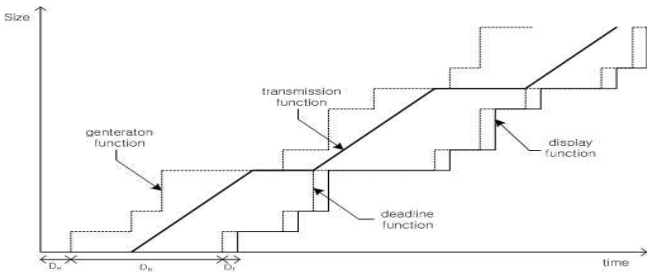
[그림 1] VOD 트래픽 흐름도

그림 1은 멀티케스트 및 유니케스트 방식으로 전송하는 VOD 트래픽 흐름도를 보이고 있다. VOD 어플리케이션은 전송측에서 사용자가 선택한 VOD를 멀티캐스팅 및 유니캐스팅 방식으로 전송한다.

전송측과 수신측은 세션이 결정되기 전에 실시간성을 보장할 수 있는 네트워크 자원을 할당해야 한다.

본 논문에서는 세션이 유지되는 동안 연속 미디어 스트림의 출력 연속성을 보장할 수 있도록 그림 2에서와 같이 시간에 따라 연속 미디어 데이터가 생성되고

소비되어야 한다. 그림 2는 비디오 데이터와 같은 프레임 단위로 생성되는 생성함수(generation function)와 수신측에서 출력되는 디스플레이 함수(display function)를 나타내고 비디오 출력의 연속성을 보장할 수 있는 전송속도로 전송되는 전송함수(transmission function)를 보이고 있다. 전송함수는 가변 비트율의 전송속도를 갖는 비디오 데이터와 네트워크 상태로 인하여 발생되는 지터 때문에 다음 두 구간으로 나눌 수 있다. 그 첫 번째는 수신측에서 요구한 전송속도로 전달하는 전송구간과 전송측에 버퍼에 충분한 데이터가 채워지지 않아서 전송을 할 수 없는 유휴구간으로 이루어져 있다. 전송속도는 전송측에서 패킷의 크기를 결정하는데 이용된다. 요구된 패킷을 전송하기 위해서 네트워크의 가용대역폭을 할당한다.



[그림 2] 연속 미디어 스트림의 유연화 과정

전체 시스템의 성능은 QoS 파리미터 P_q와 자원 할당 시 뮤텍스, 세마포어 사용 자원 할당 블록킹에 인한 오버헤드 P_o, 그리고 자원의 이용률 P_u와 관계 있다. 이때 전체 시스템에서 각 파라미터의 중요도에 따라 가중치를 부여하여 다음과 같은 식이 나올 수 있다.

$$T = P_q + \alpha \cdot P_o + \beta \cdot P_u \qquad (식\ 1)$$

$$P_q = A_{blocking} + \mu \cdot A_{ave} \qquad (식\ 2)$$

두 송신자와 수신자간의 동기성을 보장하기 위한 방법은 흐름 제어 변환기법을 통하여 수행한다. 이러한 동기성을 보장할 수 있는 방법은 피드포워드 제어 기법과 피드백 제어 기법이 있다. 전자는 전송버퍼의 버퍼상태를 기반으로 전송률을 조정한다. 두 번째, 수신버퍼버퍼상태를 통하여 전송측의 전송률을 변경하도록 한다.

본 논문에서는 수신자 기반의 버퍼 상태 정보를 기반한 전송속도를 변경하는 제어 방법으로 전송자와 수신자간의 동기성을 유지 할 수 있도록 하고자 한다. 이를 위해서 수신버퍼는 TH_h, TH_m, TH_l와 같이 세 가지 수준의 임계값을 갖는다. 수신버퍼에 도착한 데이터가 세 가지 수준의 임계값에 이를 때 전송률 조정정보를 피드백 한다. 이러한 버퍼임계값과 전송속도의 사용은 연속 미디어 스트림의 실시간성을 보장하기 위한 것이다.

또한 각 미디어 데이터들은 네트워크를 통하여 전송될 때 미디어 데이터의 전송시간을 결정하여야 한다. 미디어 데이터의 전송 시간을 결정하는 궁극적인 이유는 해당 미디어 데이터의 검색 시간을 결정하기 위해서이다. 예를 들어, 크기가 M인 어떤 이미지 데이터를 전송하기 위해서 $T_t(M)$시간이 요구된다면, 이 이미지 데이터가 요구된 시간에 정상적으로 전송되기 위해서는 적어도 식 7의 $T_{transfer}$ 시간에는 전송에 적합한 형태로 변환되어 네트워크를 통해 전송되어야 할 것이다(식 3). 또한, 저장된 미디어 데이터를 전송하기 위해서는 그것이 저장된 저장 위치에서 검색되어야 한다. 크기가 M인 미디어 데이터를 모두 검색하는데 소요되는 시간을 $T_r(M)$이라 하면, 미디어 데이터의 검색을 시작해야 할 검색 시점은 적어도 식 4의 $T_{retrieval}$이어야 한다[4].

$$T_{transfer} = T_{playout} - T_t(M) \qquad (식\ 3)$$

$$T_{retrieval} = T_{playout} - T_{transfer} - T_r(M) \qquad (식\ 4)$$

따라서 해당 미디어를 전송하는데 요구되는 순수 데이터 전파 시간과 네트워크의 상태에 따라 변하는 전송 지연 시간으로 구성되어 전송될 미디어 데이터, 패킷의 크기 및 채널 용량에 종속적이거나 고정적인 시간을 요구하며, 네트워크 트래픽 상태의 변화에 따라 패킷당 지점간 전송 지연 시간 $D_{end-to-end}(pkt)$은 다음과 같다. 단대단 패킷 지연시간은 데이터를 디스크에서 읽어오는 시간, 이 데이터를 패킷화 시키는 시간 등 일정하게 발생하는 지연시간 D_{fix}와 네트워크에서 라우터를 경유하면서 라우터에서의 불규칙한 대기시간과 네트워크 전송 도중에 발생하는 지연시간 D_{var}의 합으로 나타낼 수 있다.

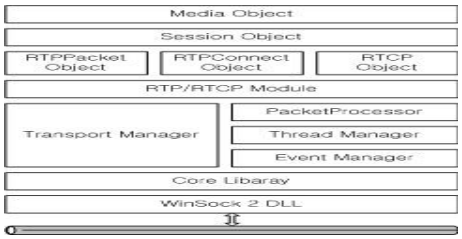
$$\begin{aligned} D_{end-to-end}(pkt) &= D_{fix} + D_{var} \\ &= (D_{transfer_setup} + D_{propagation}) + D_{jitter} \end{aligned} \qquad (식\ 5)$$

3. 객체 모델

객체 모델에는 다양한 미디어의 특성을 고려한 정보를 담고 있는 MediaObject를 응용 프로그램에서 요구하게 된다. 또한, MPEG, H.261, JPEG 등 미디어 타입과 미디어의 시간, 우선 순위 등의 정보를 포함하고 있다. 미디어 타입에 따라 미디어 특성에 적합한 특정 처

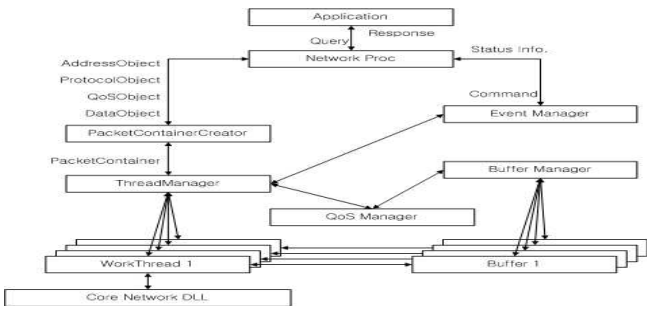
리가 필요하지만 추상화를 위해 MediaObject에서 PacketObject에 범용적으로 담겨질 DataObject로 변환되어 PacketObject에 담긴다.

AddressObject는 호스트 주소나 호스트 이름, 포트 번호, 서비스 번호를 관리하며, ProtocolObject에서는 프로토콜에 따른 소켓 초기화 작업을 담당하는 객체이다. 미디어의 특성과 프로토콜에 따라 응용프로그램에서 QoS 파라미터 정보를 담는 객체이다. 그림 3에서 객체 모델을 보여주고 있다.



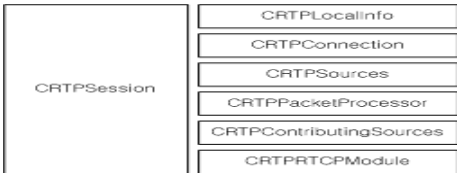
[그림 3] RTP 객체 모델

이렇게 Transport Manager는 PacketContainer에 담긴 객체들 중 DataObject와 QoSObject를 이용하여 미디어의 특성 정보를 얻고, 전송지연 허용시간을 얻어 RTP의 초기화 작업을 한다. 다수의 클라이언트의 접속을 가능하도록 PacketContainer에 하나의 쓰레드가 할당되며 ThreadManager에서 WorkThread를 생성하게 된다. WorkThread에서는 QoS 처리, 버퍼 처리를 위하여 QoSObject를 이용하여 Container에 있는 소켓에 QoS 옵션을 할당하게 된다. 그림 4에서 ThreadManager는 WorkThread에서 사용하게될 버퍼를 할당하게 되며 WorkThread 자신에게 할당된 버퍼에만 액세스가 가능하며 버퍼가 full이거나 empty일 경우, BufferManager에서 Event를 발생시키게 된다.

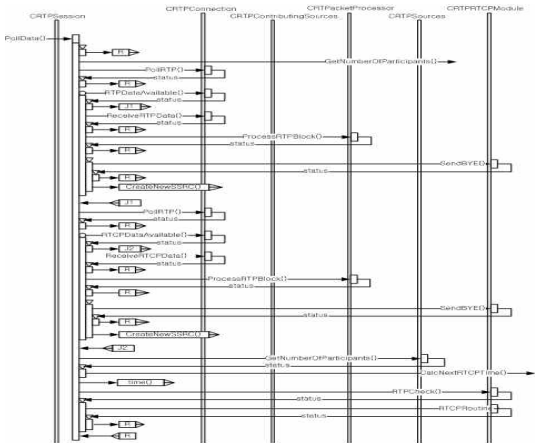


[그림 4] 이벤트 모델

RTPSession 객체는 하나의 세션이 열리면 생성되는 객체로서 접속이 이루어지면 하나의 세션이 생성되고 이 세션이 소멸되면 하나의 접속도 끊어지게 된다. 즉, 데이터의 접속에서부터 데이터 전송을 위한 모듈과 네트워크 관리 모듈이 포함되어 있다. 그림 5와 같이 CRTPSources, CRTPPacketProcessor, CRTPRTCPModule, CRTPContributingSources, CRTPConnection, CRTPLocalInfo 객체를 갖고 있다.



[그림 5] CRTPSession 구성



[그림 6] CRTPSession의 Polling 흐름 처리

CRTPSession 객체 내에서 데이터를 전송 버퍼와 수신 버퍼로 polling하는 부분의 알고리즘으로 그림 6에서 보여주고 있다. PollData() 함수에서 세션이 유효한지를 검사하고 전송할 데이터를 버퍼에 보내기 위해 PollRTP() 함수로 버퍼 상태를 확인한다. 수신한 패킷이 존재 할 수 있으므로 먼저 수신 패킷을 수신하고 송신하기 위해 유효하다면 전송하고자 하는 데이터를 전송한 다음 수신한 데이터에 대해 확인을 하고, RTCP 패킷을 확인하게 된다.

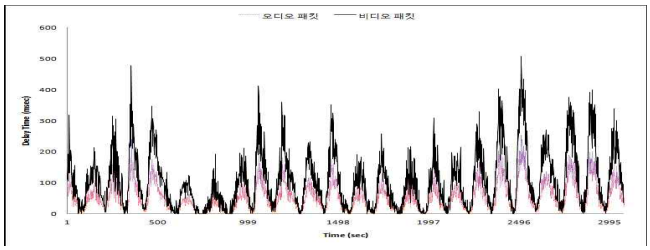
CRTPContributingSources 객체에서 RTP의 CSRC, CNAME 등을 처리할 수 있으며, 각 패킷들의 소유자를 구별하는 객체이며, CRTPSource 객체에 의해서 수신된 패킷에서 지터, 지연, 패킷번호, 왕복시간, 리소스 타입, 리소스 정보 등을 분석하게 된다. 또한 지터를 분석하여 패킷 전송률을 재설정하게 된다.

CRTPRTCPModule에서 각 패킷의 네트워크 상태를 파악하고, RTP 데이터 패킷과, RTCP의 SR, RR를 생성, 전송하게 된다.

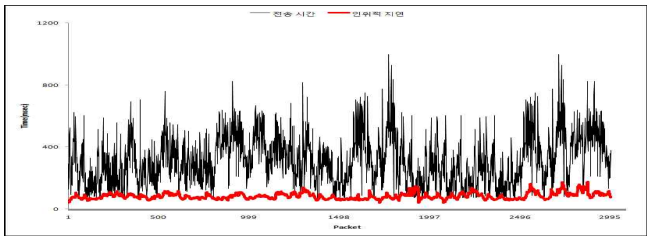
4. 실험 및 성능평가

시스템1(INTEL CORE i7 950 3.06GHZ/8MB, Windows 7, 210.117.155.131)과 시스템2(INTEL CORE i7 975 3.33 GHZ/8MB, 8MB, LINUX 9, 210.117.155.133)의 두 시스템 왕복시간이 분당 12~333ms 정도 소요되는 상태에서 시뮬레이션 하였다. 비디오, 오디오 데이터를 전송하기 위하여 비디오 전송 소켓, 오디오 전송 소켓을 만든 후, 서버의 로컬 포트를 송신할 RTP 패킷과 SR 패킷에 3000, 수신할 RTP 패킷과 RTCP의 RR 패킷에 3001을 할당하였고. 알고리즘을 적용한 클라이언트에는 각각 3002, 3003을 할당하였으며, 적용하지 않은 클라이언트에는 각각 3004, 3005를 할당하였다. 동일한 조건에서 실험하기 위하여 비디오 데이터 16Kbytes의 패킷과 오디오 데이터 8Kbytes의 패킷을 UDP로 각각 3000개씩 하나의 호스트에 알고리즘을 적용시킨 클라이언트와 적용시키지 않은 클라이언트 프로그램으로 전송하였다.

RTP/RTCP를 구현하는 모듈을 구성하고, 지연을 제어할 수 있도록 인위적인 지연을 주어 데이터를 전송하였다. 모듈은 RTP/RTCP에서 다중 전송을 위해 필요한 기본 기능으로 구현하여 패킷 처리의 오버헤드를 감소시켰다.



[그림 7] 실시간 전송에 따른 오디오와 비디오 패킷



[그림 8] 불규칙한 전송에 대한 인위적 지연

오디오와 비디오 스트림의 처리에 대한 차이점은 오디오는 데이터가 발생하는 주기마다 첫 패킷에 대해 인위적인 지연 시간을 재설정하고, 비디오 데이터에 대해서 가장 처음 전송되는 패킷만 인위적인 지연을 설정한다. 음성의 경우, 지연 평균값이 0.25초일 때 표준편차는 0.0345초이고, 이때 수신측에서 허용 가능 에러율($< 10^{-1}$)에 따라 전체 지연시간은 $\mu + 1.282\sqrt{2}\sigma$이므로 0.3125초로 예측값을 얻을 수 있었다. 시뮬레이션 결과 그림 8과 같이 지연시간이 일정하게 유지는 것을 알 수 있다.

수신측에서 인위적인 지연시간을 설정하지 않을 경우 그림 7과 같이 오디오 데이터를 전송 한 결과 비디오 데이터의 전송시간보다 낮으며 불규칙한 전송시간을 보이는 것을 볼 수 있다. 지연 시간의 변동으로 인하여 출력하고자 하는 패킷이 도착하지 않았을 때, 도착할 때까지 기다려야 하며 그 다음 패킷들이 이미 도착하여 버퍼에 쌓여 있을 가능성도 높아진다. 인위적인 지연시간을 둘 경우, 지연되는 패킷을 폐기한다 해도 허용하는 손실률을 유지 할 수 있기 때문에 패킷을 기다릴 필요가 없다.

4. 결론

본 논문은 IP 기반의 멀티미디어 스트리밍 전송이 가능하도록 멀티캐스팅 및 유지캐스팅을 혼합한 멀티접속이 가능하도록 네트워크 지연을 흡수 할 수 있는 버퍼링 기법과 객체 모델을 설계하였다. 또한 성능평가를 통해 수신측에서 연속 출력을 보장하기 위하여 인위적 지연을 설정 하므로 연속성을 보장 할 수 있었다. 따라서 본 논문은 IP 기반의 멀티미디어 통신 서비스가 가지고 있는 다양한 컨텐츠 서비스를 급격한 시스템의 변화와 네트워크의 변화에도 사용자가 원하는 고급 서비스를 충족할 수 있을 거라 생각한다.

참고문헌

[1] Edward M Schwalb, “TV Handbook Technologies and standards” PTR, in New Jersey, 2004

[2] S. Deering, RFC 1112: Host Extensions for IP Multicasting, IETF, Aug, 1989.

[3] T. D. C. Little and F. Kao, "An Intermedia Skew Control System for Multimedia Data Presentation", The 3rh Lnt'l workshop on Network and Operating System Support for Digital Audio and Video, Nov. 1992.

[4] T. D. C. Little and A. Ghafoor, "Distributed Multimedia Objects Composition and Communication", IEEE Network Magazine, Nov. 1990.