

MPEG4를 이용한 다자간 멀티미디어 프로그램에서의 효율적인 데이터 전송 방법

윤교철⁰, 김영만
국민대학교 컴퓨터학부
{charlie⁰, ymkim}@kookmin.ac.kr

Efficient Data Transmission of Multi-Point Multimedia Chatting Program using MPEG4

Kyo Chul Yoon⁰, Young Man Kim
School of Computer Science, Kookmin University

요 약

본 논문에서는 MPEG4 표준 코덱을 사용하여 다자간 멀티미디어 프로그램을 구현하는데 있어서 효율적으로 멀티미디어 데이터를 전송하는 방법과 그 구현에 대해서 연구한다. 다자간 멀티미디어 프로그램은 크게 음성, 영상, 텍스트의 멀티미디어 요소를 가지고 있으며 각각의 데이터는 네트워크를 통해 여러 사용자에게 전송이 되는데 네트워크를 통해 전송되어지는 데이터를 최소화하기 위한 여러 가지 방법을 제시하고 각 방법에 대한 전송효율에 대하여 분석한다. 본 논문에서는 서버-클라이언트 모델을 사용하여 이 방법들을 구현하였으며 현재 8명의 사용자가 동시에 접속하여 다자간 멀티미디어 프로그램을 사용할 수 있도록 하였고 그 이상의 사용자 수에 대해서도 변경할 수 있도록 하였다.

1. 서 론

최근 들어 화상채팅과 같은 다자간 멀티미디어 서비스를 제공하는 사이트들이 증가하고 있으며 가정이나 사무실, 인터넷 PC 방 등에서 손 쉽게 접속하여 여러 사용자들과 영상 및 음성, 텍스트 등을 주고 받을 수 있게 되었다. 그러나 이러한 사이트들에서 제공하는 다자간 멀티미디어 서비스의 어느 한 이용자에게 최대의 권한을 부여하여 데이터 전송을 주도하고 다른 사용자들의 경우 멀티미디어 데이터 중 일부만을 전송할 수 있도록 하고 있다. 특히 음성의 경우, 한 이용자(주로 방장)만이 권한을 소유하고 있어서 다른 이용자는 최고 권한 이용자의 허락 없이는 음성을 전달할 수 없으며 단지 최고 권한 이용자의 음성 데이터만을 수신 할 수 있다. 따라서, 본 논문에서는 다자간 멀티미디어 프로그램을 이용하는 모든 사용자에 대해 똑 같은 권한을 부여해서 각각의 사용자들이 모든 멀티미디어 데이터를 전송할 수 있도록 MPEG4 표준[1][2][3] 을 만족하는 코덱을 사용하여 영상 데이터를 압축 전송하도록 하고 음성 데이터의 경우 ADPCM를 사용하여 음성 뿐만 아니라 MP3 와 같은 고품질 음악파일을 전송할 수 있도록 하며 또한 효율적인 데이터 전송 방법에 대해서 설명하도록 한다.

2. 다자간 멀티미디어 프로그램 설계

다자간 멀티미디어 서비스를 구현하는 방법에 있어서 영상 데이터에 대해 멀티 캐스팅을 통한 네트워크 전송 횟수를 줄이는 방법과 음성 데이터에 대해서는 서버-클라이언트 모델을 사용하여 각 클라이언트들은 음성 데이터를 서버에게 전송하고 서버는 모든 클라이언트들의 음성을 합성하여 각 클라이언트들에게 재전송하는 방식을 구현한 바 있다[6]. 그러나, 이 방법은 여러 가지 문제점을 발생 시키는데 본 논문에서는 이러한 문제점에 대해서 설명하고 이를 보완하기 위한 설계 및 구현을 하였다.

2.1. 영상 데이터

멀티 캐스팅을 통해 각각의 클라이언트가 자신을 제외한 나머지 클라이언트에게 영상 데이터를 전송하면 피어-투-피어(Peer-To-Peer) 방식에 비해서 네트워크 전송 횟수는 획기적으로 줄일 수 있지만 사무실이나 학교, 관공서 등 로컬 랜으로 연결되어있는 경우에만 적용되어지며 가정에서 다자간 멀티미디어 서비스를 사용하고자 하는 사용자의 경우는 멀티캐스팅을 통한 영상 데이터 전송을 할 수 없다. 이를 보완하기 위해 멀티 캐스팅을 통한 분산 모델을 음성과 같은 방식의 서버-클라이언트 모델로 변경하여 각각의 클라이언트는 자신의 영상 데이터를 서버에게 보내고 서버는 전송 받은 영상 데이터들을 클라이언트에게 재전송함으로써 로컬 랜(Local LAN)이 아닌 클라이언트들에 대해서도 정상적인 영상 데이터를 주고 받을 수 있도록 하였다.

2.2. 음성 데이터

음성의 경우 서버에 연결된 모든 클라이언트는 많은 시간 동안 노이즈를 포함한 유효 음성 데이터와 상관없는 데이터를 계속해서 서버에게 전송하게 되는데 서버는 연결된 모든 클라이언트에서 보내온 음성 데이터를 합성 후 다시 모든 클라이언트에게 재전송하게 된다. 즉 클라이언트는 불필요한 음성 데이터에 대해서도 네트워크를 통해 서버로 전송하게 되고 서버는 유효 음성 데이터(실제 사용자가 말을 하여 발생하게 되는 데이터)가 아닌 음성 데이터에 대해서도 매번 합성 작업을 하게 된다. 또한 하나의 클라이언트가 전송한 음성데이터는 서버에 의해서 모든 클라이언트에게 재전송됨으로 자신의 음성을 다시 듣게 되는 메아리 현상이 발생하게 된다. 이를 보완하기 위해 클라이언트에서 발생하는 음성 데이터를 유효 영역과 노이즈 영역으로 구분하여 유효 음성 데이터만을 서버에게 전송하고 서버는 음성 데이터를 전송한

클라이언트를 제외한 나머지 클라이언트에게 음성 데이터를 전송하여 모든 성 클라이언트에서 발생하는 음성 데이터에 대해 네트워크를 통한 전송 횟수를 획기적으로 줄였다.

2.3. 사설 IP 사용

인터넷 서비스 제공 업체로부터 NAT(Network Address Translator)를 통해 사설 IP를 제공 받아 사용하는 클라이언트의 경우 다른 클라이언트로부터 직접 멀티미디어 데이터를 전송 받을 수 없으므로 서버에게 멀티미디어를 전송하고 서버는 각각의 클라이언트들에게 멀티미디어 데이터를 전송하도록 하여 사설 IP를 이용하는 클라이언트와도 정상적인 데이터 전송을 할 수 있도록 하였다.

3. 다자간 멀티미디어 프로그램 구현

본 논문의 다자간 멀티미디어 프로그램은 클라이언트 프로그램과 서버 프로그램으로 이루어져 있다.

[그림 1]은 다자간 멀티미디어 프로그램 중 클라이언트 프로그램의 전체 구조를 보여주는 구성도이다. 음성과 영상은 Encoder를 통해 압축을 하고 텍스트 데이터는 순수하게 Multiplexer를 통해 전송한다. 수신 측에서는 압축된 음성과 영상 데이터를 입력 받아 Demultiplexer를 통해 Decoder로 보내 압축된 데이터를 복원하고 텍스트 데이터는 그대로 각각 해당 인터페이스를 통하여 출력하도록 구현하였다. 전체적인 데이터 흐름의 제어는 시스템 제어부에서 다루어진다.

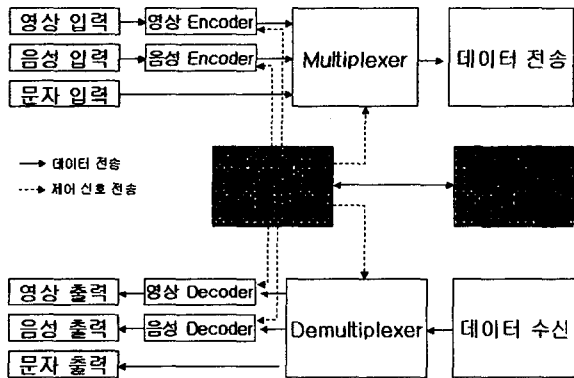


그림 1 클라이언트 프로그램 구성도

서버 프로그램의 경우 각각의 클라이언트가 보내온 멀티미디어 데이터 들을 해당 클라이언트들에게 재전송하게 된다.

3.1. 영상 데이터

[그림 2]는 영상 데이터에 대한 서버와 클라이언트의 관계를 나타낸다. 앞서 말한 바와 같이 멀티 캐스트를 통한 영상 데이터의 송수신은 사설 IP를 사용하는 클라이언트에서는 할 수 없으므로 사설 IP 클라이언트의 경우 서버에게 접속하고 각각의 소켓 포트에 대해 연결 상태를 유지해 자신의 영상 데이터를 전송하고 다른 클라이언트의 영상 데이터를 서버로부터 수신 받아 영상을 출력할 수 있도록 한다. 그러나, 클라이언트들이 서버에 접속할 때 발생하게 되는 문제점은 MPEG4를 사용한 영상 데이터 압축을 높이기 위해 영상 데이터의 초기 프레임(I-Frame)에 영상의 사이즈, 전체 영상 데이터 등의 정보를 넣고 다음 프레임(P-Frame)부터는 앞선 영상에서

변경되어지는 영상 부분만을 압축하게 된다. 이 때, 서버에 새롭게 접속하는 클라이언트는 기존에 서버에 연결된 클라이언트의 영상 데이터의 초기 프레임을 서버로부터 얻지 못하여 기존 클라이언트의 영상 데이터를 정상적으로 전송 받지 못하게 된다. 이를 보완하기 위하여 서버는 새로운 클라이언트의 접속에 대해 기존 클라이언트들에서 알려주고 기존 클라이언트들은 초기 프레임을 재전송 하도록 하여 새롭게 접속된 클라이언트가 기존의 클라이언트의 영상 데이터를 정상적으로 재생할 수 있도록 하였다. 각각의 영상 데이터의 구분은 RTP(Real-time Transport Protocol)를 사용하여 RTP 헤더에 해당 클라이언트의 인덱스를 삽입하여 전송함으로써 전송 받은 영상 데이터가 어떤 클라이언트의 것인지 구분할 수 있도록 하였다.

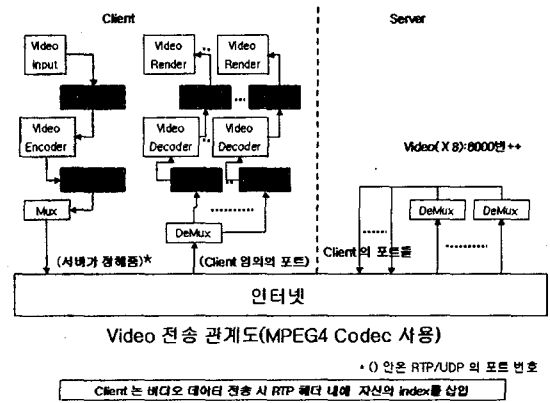


그림 2 영상 데이터에 대한 서버와 클라이언트의 관계

3.2. 음성 데이터

[그림 3]은 음성 데이터에 대한 서버와 클라이언트의 관계를 나타낸다. 앞서 말한 바와 같이 서버에 연결된 모든 클라이언트가 노이즈를 포함한 무의미한 음성 데이터를 서버에게 전송하고 서버는 모든 음성 데이터에 대해 합성을 통해 다시 모든 클라이언트에게 재전송하는 것은 서버의 부하 및 네트워크 사용량을 증가시키게 된다. 음성 데이터는 다음의 공식에 의해

$$\begin{aligned} \text{평균 초당 바이트 수} &= \text{채널 수} * (\text{샘플링 당 비트 수}/8) * \\ &\quad \text{초당 샘플링 수} \\ &= 1 \text{ Channel} * (16/8) * 8000 \\ &= 16000 \text{ byte} \end{aligned}$$

초당 약 16kbyte 가 생성되며 480byte 가 입력이 되어질 때마다 ADPCM 방식을 통해 32kbps 의 비트율로 부호화하여 전송이 되어진다. 기존의 방식을 사용하였을 경우 8개의 클라이언트가 서버와 음성 데이터를 송수신 하면 다음 공식에 의해 모든 클라이언트와 서버의 분당 전송 횟수는

$$\begin{aligned} \text{총 분당 전송 횟수} &= (\text{총 클라이언트의 수} * \text{분당 클라이언트가} \\ &\quad \text{서버에게 보내는 음성 데이터 전송 횟수}) \\ &\quad + (\text{총 클라이언트의 수} * \text{분당 서버가} \\ &\quad \text{클라이언트에게 보내는 음성 데이터 전송} \end{aligned}$$

$$= (8 * (16000/480*60)) + (8 * (16000/480*60))$$

$$= 32000$$

가 발생하게 된다. 이는 노이즈를 포함한 음성 데이터이며 유효 음성 데이터가 존재하지 않더라도 발생하게 되는 총 분당 전송 횟수이다. 또한 서버는 각각의 클라이언트가 보내온 음성 데이터에 대해 합성을 한 후 전송하게 되므로 시간차 및 서버 부하를 가져오게 된다. 이에 전송 횟수를 줄이고 서버의 부하를 줄이기 위하여 각각의 클라이언트에서 발생한 음성 데이터에 대해 유효 음성 데이터와 노이즈를 분리하여 유효 음성 데이터 발생시만 서버에게 전송하게 하고 어느 한 순간에는 오직 한 클라이언트의 음성 데이터만이 유효하다고 가정하면 다음의 공식에 의해 총 분당 전송 횟수는

총 분당 전송 횟수 = 분당 클라이언트가 서버에게 보내는 음성 데이터 전송 횟수
 + (총 클라이언트의 수 - 1) * 분당 서버가 클라이언트에게 보내는 음성 데이터 전송 횟수
 = (16000/480*60) + (7*(16000/480*60)) = 16000

가 나온다. 이는 클라이언트의 요청에 의해 서버가 한 사람의 음성 데이터만을 전송하도록 하여 나온 결과이며 유효 음성 데이터가 발생하지 않을 경우 전송 횟수는 훨씬 더 줄어들게 된다. 또한 음성 데이터를 전송한 클라이언트를 제외한 나머지 클라이언트들에게 데이터를 보내므로 메아리 현상이 발생하지 않게 된다.

음성 이외에 자신이 보유한 음악 데이터를 상대방에게 전송하고자 하는 경우 해당 클라이언트는 서버에게 자신이 음악을 전송함을 알리고 음악 재생 플레이어를 가동하면 서버에 연결된 모든 클라이언트에게 음악을 전송할 수 있도록 하였다. 이를 위해서는 마이크를 통해 입력되는 음성 신호를 받아들이는 Audio Mixer의 MIC 디바이스에 해당 부분을 음악 데이터 재생 시 발생되는 신호를 받아들이는 WAVE 디바이스로 변경함으로써 인해서 WAVE 디바이스에서 발생하는 신호를 부호화하여 전송하게 된다.

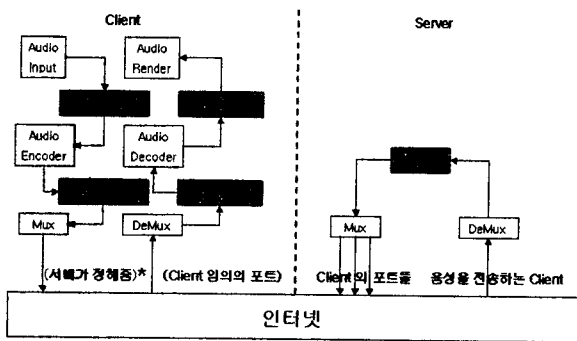


그림 3 음성 데이터에 대한 서버와 클라이언트의 관계
 * () 안은 RTP/UDP 의 포트 번호

그림 3 음성 데이터에 대한 서버와 클라이언트의 관계

3.3. 사설 IP 사용

[그림 4] NAT 구성도이며 앞서 말한 바와 같이 사설 IP를 사용하는 클라이언트에 대해서는 해당 클라이언트는 서버와의 CallMessage 를 위한 TCP 포트 뿐만 아니라 멀티미디어 데이터에 대한 UDP 포트들에 대해서 미리 연결을 하여 다른 클라이언트가 보낸 멀티미디어 데이터를 서버로부터 정상적으로 전송 받을 수 있도록 구현하였다.

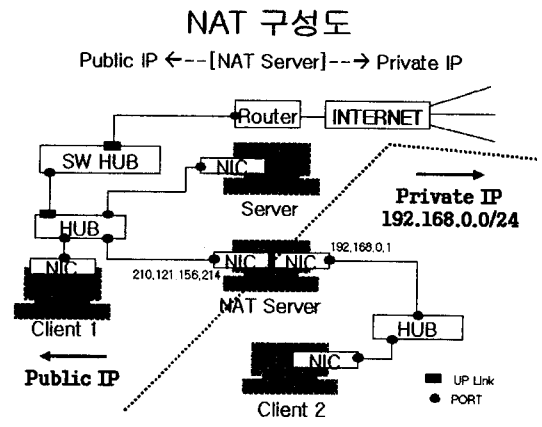


그림 4 NAT 구성도

4. 결론

본 논문에서는 MPEG4 표준에 의거한 다자간 멀티미디어 프로그램을 구현하는데 있어서 발생 할 수 있는 문제점을 노출시키고 그 해결 방안으로서 네트워크의 전송 횟수를 줄이고 서버의 부하를 줄이는 방법을 제안하며 제안방법에 기반한 소프트웨어를 설계, 구현하였다.

참고 문헌

[1] ISO/IEC, "Information Technology - Generic Coding of Audio-Visual Objects", ISO/IEC 14496-1 ~ ISO/IEC 14496-5, Dec.1998.
 [2] Weiping Li, Jens-Rainer Ohm, Mihaela van der Schaar, Hong Jiang, and Shipeng Li, "MPEG-4 Video Verification Model version 17.0", July.2000.
 [3] Joan L.Mitchell, Willam B. Pennebaker, Chad E. Fogg, and Didier J. LeGall, "MPEG video compression standard", 1996.
 [4] RFC 1889 "RTP - A Transport Protocol for Real-time Applications".
 [5] RFC 1890 "RTP Profile for Audio and Video Conferences with Minimal Control".
 [6] 한국정보과학회, 이봉하, 윤교철, 김영만, MPEG4를 이용한 다자간 멀티미디어 코딩 프로그램 설계 및 구현 (2001, 10,20)