

MPEG4를 이용한 다자간 멀티미디어 채팅 프로그램 설계 및 구현

이봉하^o 윤교철 김영만
국민대학교 컴퓨터학부

{lyoon74, charlie, ymkim}@kookmin.ac.kr

Design and Implementation of the Multi-Point Chatting Program

Bong Ha Lee^o Kyo Chul Yoon Young Man Kim
School of Computer Science, Kookmin University

요약

본 논문에서는 MPEG4 표준에 의거한 멀티미디어 응용프로그램인 다자간 채팅 프로그램을 설계, 구현하였다. 다자간 멀티미디어 채팅 프로그램의 영상모듈과 음성모듈은 데이터 성격상 큰 차이를 가지고 있어서 별개의 정보처리 및 전송 모델을 적용하여 설계, 구현하였다. 영상모듈은 분산처리 모델을 적용하여 실제 채팅 시에는 영상 정보를 멀티캐스트로 클라이언트간 직접 통신을 하도록 하였다. 멀티캐스트 방식은 송신자 측에서 여러 수신자 측에게 한 번에 메시지가 전송되도록 하여, 데이터의 중복전송으로 인한 네트워크 자원의 낭비를 최소화할 수 있게 된다. 반면 음성모듈은 Client-Server 모델을 적용하여 설계, 구현하였다. 음성처리 서버는 영상처리 서버와 달리 서버에서 각 클라이언트의 음성정보들을 합성하는 mixing작업을 수행하여 그 결과를 각 클라이언트로 전송하게 된다. 즉, 각 데이터 특성에 적합한 정보모형을 사용하여 다자간 채팅 프로그램을 설계, 구현하였다.

2. 멀티미디어 채팅 프로그램 설계

MPEG4 표준에 의거한 다자간 멀티미디어 채팅 프로그램은 크게 음성모듈과 영상 모듈로 구성되어 있는데 각각 다른 정보전달 모델을 적용하여 설계 하였다. 음성 모듈은 전형적인 Client-Server모형을 사용하였고, 영상 모듈은 완전 분산처리 모델을 이용하여 설계 하였다.

2.1. 다자간 멀티미디어 채팅 프로그램 구성 개요

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

그림 1은 다자간 멀티미디어 채팅 프로그램 중 클라이언트 부분의 전체 구조를 보여주는 구성도 이다. 오디오, 비디오 데이터를 입력받아 부호화처리를 하게 되고, 그 결과를 Multiplexer를 통해 전송한다. 수신자는 부호화 처리된 오디오, 비디오 데이터를 입력받아 Demultiplexer를 통해 디코더로 보내고 디코더 처리된 데이터를 해당 인터페이스를 통하여 출력하도록 설계하였다. 전체적인 데이터 흐름의 제어는 시스템 제어부에서 다루어진다.

1. 서론

최근 들어서 영상을 중심으로 여러 형태의 멀티미디어 정보를 결합하여 저장하거나 전송하는 소위 '멀티미디어응용도구'들이 전세계적인 열풍을 불러일으키고 있다.

고속 저손실의 광케이블이 일반 가정까지 도달하여 TV 수상기 위의 컨버터인 Set Top Box (STB) 혹은 고성능 PC에 연결됨으로써, 안방에서 대형 스크린과 하이파이 입체 음향을 갖는 고선명 TV(HDTV)를 통해 영화관에서와 같은 감동을 그대로 느끼고, 화상전화를 통해 멀리 떨어져 있는 사람과 통화하며, 상품정보를 컴퓨터 화면을 통해 보면서 구매하고 (홈쇼핑), 은행 계좌도 집에서 하며 (홈뱅킹), 원하는 TV 프로나 음악이나 게임을 언제든 주문하여 즐길 수 있고 (주문형 비디오/오디오/게임: Video/Audio/Game On Demand), 학교의 강의도 집에서 들으며 (원격강의: Teleteaching), 기업 간의 상거래 문서도 전자화되고 (전자 데이터 교환(EDI: Electronic Data Interchange) 혹은 전자교역(EC: Electronic Commerce)), 나아가서 재택 근무까지 가능한 세상이 되었다. 이때 기존의 방송, 가전, 컴퓨터, 통신 분야가 각기 멀티미디어 정보를 처리할 수 있도록 디지털화 됨으로써 궁극적으로 이 모두가 융합되어 진정한 멀티미디어 시대가 이루어 질 것이다[2][3].

본 논문에서는 MPEG4 표준[1]에 의거한 멀티미디어 응용프로그램인 다자간 채팅 프로그램을 설계, 구현하였다. MPEG4에서는 기존의 영상, 오디오 신호의 압축 부호화에 추가해서 정지영상, 컴퓨터 그래픽스, 분석 합성계의 음성 부호화, MIDI등에 의한 합성 오디오 및 텍스트도 포함하는 종합 멀티미디어 부호화 규격을 목적으로 하고 있다. 다자간 멀티미디어 채팅 프로그램의 영상모듈과 음성모듈은 데이터 성격상 큰 차이를 가지고 있어서 별개의 정보처리 및 전송모형을 적용하여 설계, 구현하였다.

본 논문의 2장에서는 멀티미디어 채팅 프로그램의 설계에 대해서 살펴보고, 3장에서는 실제적으로 구현한 주요 부분에 대해서 기술한다. 마지막으로 4장에서는 본 논문의 결론을 맺는다.

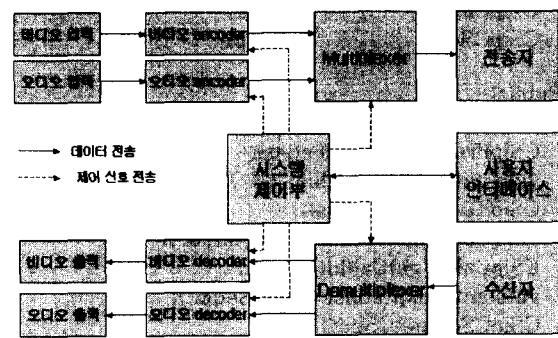


그림 1 다자간 멀티미디어 채팅 프로그램 클라이언트부분 구성도

2.2. 음성 모듈 설계

음성모듈 부분에서는 Client-Server 모델을 사용하였다. 음성 데이터 성격상 각 클라이언트의 음성정보들을 합성(mixing) 하는 작업이 중복 없이 효율적으로 이루어지도록 서버에서 합성과정을 담당하여 그 결과를 모든 클라이언트에게 전송하도록 하였다.

Client-Server 모델은 정보의 집중처리를 통하여 시스템 효율을 올릴 수 있는 응용분야에 적합하다. 예를 들면 요즘 음 널리 사용되는 인터넷 뱅킹을 들 수 있다. 고객이 자기의 컴퓨터로 자신의 예금 상황을 조회할 경우, 그 고객의 컴퓨터에 있는 클라이언트 프로그램이 잔액 조회 요청을 은행 컴퓨터에 있는 서버 프로그램에게 하며 서버는 데이터베이스를 조회하여 해당 정보를 제공하게 된다. Client-Server 모델은 지금 네트워크 컴퓨팅의 중심 개념으로 되어 있다. 인터넷 프로토콜인 TCP/IP 내의 응용프로토콜(예,FTP, NFS, DNS등)들도 Client-Server 모델을 사용하고 있다. 그림 2는 음성 모듈에 적용된 Client-Server 모델을 도시하고 있다. 클라이언트들이 서버로 음성을 보내면 서버는 클라이언트들로부터 전송 받은 음성들을 처리(mixing작업)하여, 다시 클라이언트들에게 합성 음성을 전송하게 된다.

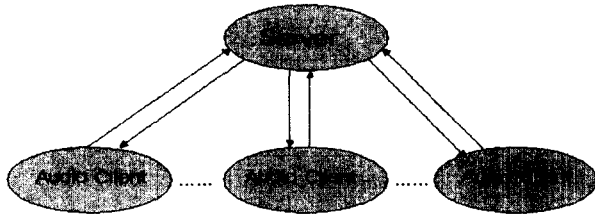


그림 2 음성 모듈에 적용된 Client-Server 모델

2.3. 영상 모듈 설계

음성 모듈과는 달리 영상 모듈은 분산처리 모델이 사용된다. 분산처리 모델에서는 영상스트림처리 기능이 모든 노드에 균등하게 분산되어 수행되어지며, 사용자가 필요로 하는 데이터만이 네트워크를 통해 전달되는 방식이다.

영상모듈은 분산처리 모델을 적용하여 설계하였는데 채팅 클라이언트는 채팅그룹에 가입하거나 탈퇴할 때에 한하여 서버와 관리 정보를 주고 받으며 실제 채팅 시에는 멀티캐스트 방식으로 클라이언트 간 직접 통신을 하게 된다. 멀티캐스트 방식은 송신자 측에서 여러 수신자 측에게 동시에 메시지가 전송되도록 하여, 데이터의 중복전송으로 인한 네트워크 자원의 낭비를 최소화해준다. 따라서 멀티캐스트 방식은 인터넷 화상 회의 등의 응용에서 많이 사용된다.

그림 3은 영상 모듈에 적용된 분산처리 모델을 보여주고 있다.

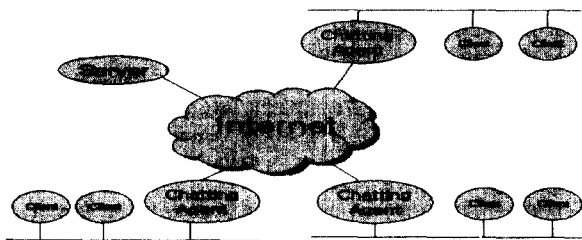


그림 3 영상 모듈에 적용된 분산처리 모델

다.

3. 멀티미디어 채팅 프로그램 구현

데이터 특성에 적합한 구현을 위해서 음성모듈은 Client-Server 모델을 사용하고 있고, 영상모듈은 완전한 분산처리 모델이 사용되었다. 이 장에서는 본 프로그램 구현시의 환경 및 방법에 관해서 서술한다.

3.1. 구현 환경

본 논문에서 구현한 다자간 멀티미디어 채팅 프로그램은 클라이언트 및 서버프로그램으로 이루어져 있으며, 윈도우98이상 환경의 운영체제에서 Visual C++ 6.0 Tool을 이용하여 MPEG4 표준[1]을 적용하여 구현하였다.

3.2. 음성 모듈 구현

Client-Server 모델을 사용하고 있는 음성모듈의 서버프로그램에는 Demultiplexer, Multiplexer, Audio-Decoder 클래스와 Audio-Encoder, Audio-Mixer 클래스들이 있다. 최대 12명 클라이언트로부터 음성 정보를 받아들이는 부분은 Demultiplexer 클래스이고, Audio-Decoder 클래스에서는 받은 음성 정보를 각각 해독하여 해당 버퍼에 저장하며, Audio-Mixer 클래스에서는 디코딩 버퍼내의 음성정보들을 합성하게 된다. Audio-Encoder 클래스는 합성음성정보를 부호화하며 Multiplexer 클래스에서 각각의 클라이언트들에게 전송하게 된다. 그림 4에서는 다자간 멀티미디어 채팅 프로그램 음성모듈 내부 클래스들간의 관계를 나타낸다.

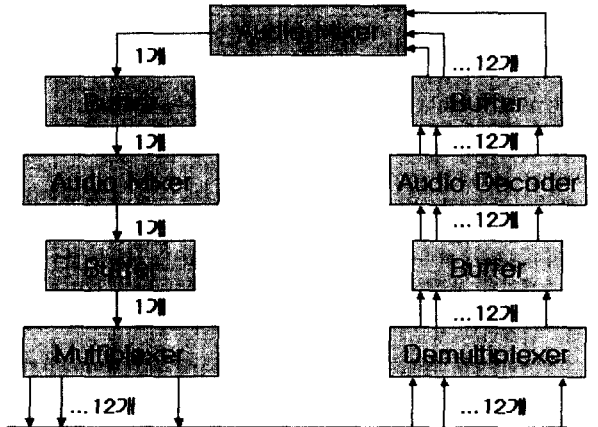


그림 4 서버 프로그램 내 음성 모듈 내부 클래스 구성도

Client 프로그램내의 음성모듈에서는 Audio-Mixer 클래스 대신 Audio-Input, Audio-Renderer 클래스 부분이 추가되어 마이크로폰으로부터 음성 정보를 받아들이고, 사용자 스피커로 출력을 하는 작업을 담당하도록 하였다.

3.3. 영상 모듈 구현

분산처리 모델과 멀티캐스트를 사용하는 영상모듈에는 Demultiplexer, VideoDecoder, VideoRenderer, Multiplexer, VideoInput, VideoEncoder 클래스가 있다. Demultiplexer 클래스에서는 네트워크로부터 영상정보를 받아들이게 되고, VideoDecoder 클래스는 이 영상정보들을 해독하게 된다. VideoRenderer 클래스는 해독된 영상을 사용자 화면으로 출력시

킨다. 한편, VideoInput클래스는 카메라를 통하여 사용자 영상 정보를 입력받는 부분이고, VideoEncoder클래스는 부호화를 하게 되며, Multiplexer클래스는 부호화 영상 정보를 네트워크로 전송한다[2][3]. 그림 5에서는 다자간 멀티미디어 채팅 프로그램 내의 영상모듈 클래스간 관계를 나타낸다. 화살표는 데이터의 흐름을 나타낸다.

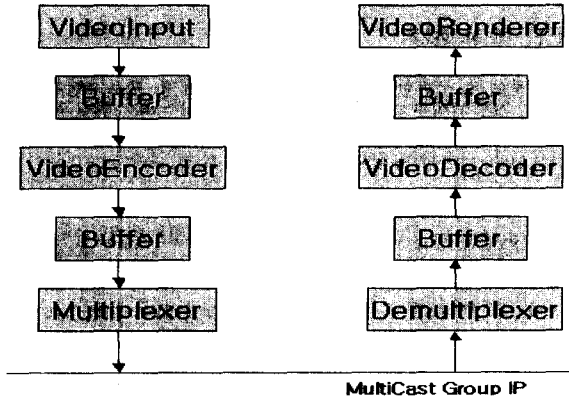


그림 6 영상 모듈 내부 클래스 구성도

3.4. RTP/RTCP 채널

실시간 멀티미디어 전송용 프로토콜인 RTP(Real-time Transport Protocol)와 그 제어 정보를 전달하는 RTCP(RTP Control Protocol)가 본 프로그램에서 사용되고 있다.

RTP는 멀티캐스트 또는 유니캐스트 상에서 음성, 화상, 또는 모의 데이터와 같은 실시간 데이터를 전송하는 용도에 적합한 단대 단 트랜스포트 기능을 제공한다.

RTCP는 회의 참여멤버 간에 분실된 패킷 수, 지터 간격, 앞의 패킷과의 지연시간 등의 QoS 정보를 교환하여 adaptive encoding을 구현하는데 사용된다. 또한 RTCP는 많은 참여자들의 스케일을 위해서 패킷 송신율을 계산하고 사용자 인터페이스의 참여자 ID를 지칭하는 최소한의 세션 제어 정보를 전달하는데 사용되는 프로토콜이다[4][5].

그림 6은 영상 모듈의 멀티캐스트 구성도를 보여준다. 서버는 멀티캐스트 그룹의 IP를 생성하게 되고 서버로부터 IP정보를 전달받은 클라이언트들은 멀티캐스트 그룹 IP에 참가를 해서 클라이언트간의 직접 통신이 이루어진다.

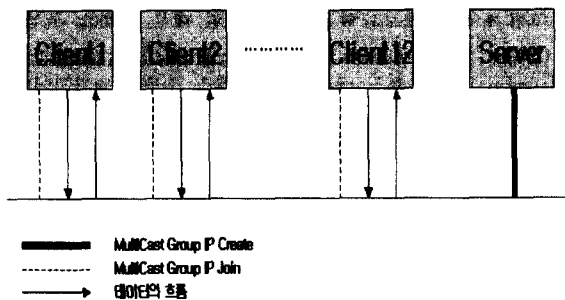


그림 6 영상 멀티캐스트 구성도

영상모듈은 분산처리 모델을 적용하여 설계하였는데 채팅 클라이언트는 채팅그룹에 가입하거나 탈퇴할 때에 한하여 서버와 관리 정보를 주고 받으며 실제 채팅 시에는 멀티캐스트 방식으로 클라이언트 간 직접 통신을 하게 된다. 멀티캐스트 방식은 송신자 측에서 여러 수신자 측에게 동시에 메시지가 전송되도록 하여, 데이터의 중복전송으로 인한 네트워크 자원의 낭비를 최소화 해준다. 그림 7은 본 프로그램에서 쓰이고 있는 음성 RTP채널 구성도를 보여주고 있다.

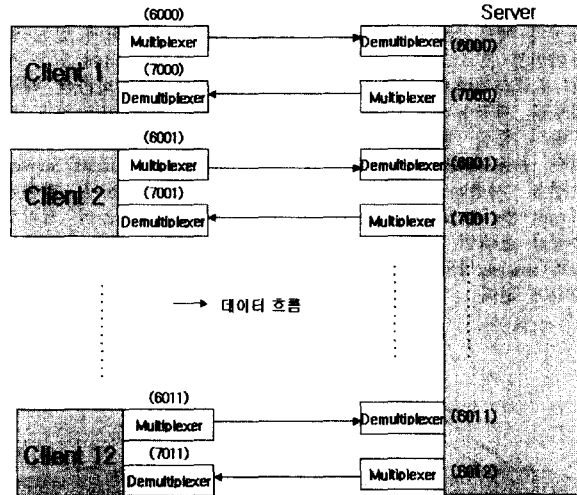


그림 7 음성 RTP 채널 구성도

4. 결론

본 논문에서는 MPEG4 표준에 의거한 다자간 멀티미디어 채팅 프로그램을 설계, 구현하였다. 최대 12명의 다자간 채팅이 가능한 프로그램이며, 음성, 영상모듈을 나누어서 각기 다른 모델을 적용하여 설계, 구현 하였다. 음성모듈은 Client/Server 모델을 적용하였으며, 영상 모듈은 분산처리 모델을 이용하여 설계 하였다.

향후 연구 방향으로 음성 모듈의 옵션으로서 영상 부분에 적용한 분산 처리 모델을 사용하여 분산구조에 적합한 분야(예를 들면 LAN응용)에서 사용하도록 할 예정이다.

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, William 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".