

VoIP 기술을 이용한 P2P 멀티미디어 채팅용 통신 프로그램의 구현

윤현님, 김양우, 이필우*
동국대학교 정보통신공학과
*한국과학기술정보연구원
{yhnim, ywkim}@dongguk.edu

Implementation of P2P Communication Program for Multimedia Chatting Using VoIP Technology

Hyunnim Yoon, Yangwoo Kim, Pilwoo Lee*
Dept. of Information Communication Engineering, Dongguk University
*Korea Institute of Science and Technology Information

요 약

이제 인터넷은 우리 실생활에서 기본적인 도구로 이용하게 되었다. 이러한 변화로 기존 음성통화에서 사용되던 OFF-LINE 전화통신이 ON-LINE 상으로 빠르게 대체되어 가고 있다. 이에 따라 음성데이터들을 인터넷망을 통하여 효율적으로 전송하는 기술들이 연구되어 왔고, VoIP 기술은 이러한 시도들에 대한 구체적인 방법론이라고 할 수 있다. 본 논문에서는 문자채팅 및 VoIP 기술을 기반으로 일대일 직접 통화가 가능한 음성채팅 프로그램을 구현하였다. 본 논문은 유무선 통합메시징시스템(UMS)의 기술 진척에 기여할 수 있다.

1. 서론

고속의 통신망이 보급되면서 인터넷은 우리의 생활 속으로 더욱 깊게 자리잡아가고 있다. 이에 따라 다양한 종류의 응용프로그램들이 개발되고 기술의 진보를 거듭하면서 문자와 음성을 동시에 메시징의 수단으로 사용할 수 있게 되었다. 이러한 변화로 기존의 음성통화에서 사용되던 off-line 전화통신이 on-line 상으로 빠르게 대체되어 가고 있다.

인터넷망을 통하여 음성 데이터들을 효율적으로 전송하는 기술들이 연구되어 왔고 VoIP(Voice over Internet Protocol) 기술은 이러한 시도들에 대한 구체적인 방법론이라고 할 수 있다.

본 논문에서는 문자채팅은 물론 VoIP 기술을 기반으로 일대일 직접 통신이 가능한 음성채팅이 포함된 멀티미디어 채팅 프로그램을 구현하였다. 또한 클라이언트 서버 중심의 정보교환이 대부분인 현재의 서비스 구조를 탈피하여 사용자들간의 직접적인 정보교환을 가능하게 하는 P2P(Peer-to-Peer) 방식을 적용하였다. 본 논문은 추후 연구를 통하여 유무선 통합메시징시

스템(UMS : Unified Messaging System)의 기술 진척에 기여할 수 있다.

2. 관련연구

인터넷을 통하여 영상, 음성 및 팩스 메시지를 전달하는 기술로 VoIP가 있다. 인터넷을 이용하고자 하는 사용자는 PC에서 인터넷에 접속하거나 인터넷 프로토콜이 적용된 독립적인 디바이스를 이용하여 접속하거나 기존의 PSTN 단말기에서 게이트웨이로 전화를 걸어 접속할 수가 있다. 이와 같은 여러 경우에 VoIP는 음성과 비디오와 같은 실시간 미디어를 전송한다.

2.1 RTP/RTCP

2.1.1 RTP(Realtime Transport Protocol)

RTP는 오디오, 비디오 및 시퀀셜 데이터와 같은 실시간 데이터를 멀티캐스트 또는 유니캐스트 네트워크를 이용해서 전송하는 응용 서비스에 알맞은 단말-대-단말 네트워크 전송 기능을 제공한다[2]. 자원예약

을 수행하지 않으며, 적시전달이나 순차전달과 같은 서비스 품질도 보장하지 않는다. 따라서 전송의 의미는 실시간 데이터의 특성에 중점을 두어 제정한 표준이라고 할 수 있다.

2.1.2 RTCP(RealTime Control Protocol)

RTP와 관련된 프로토콜로 RTCP 데이터 전송을 사용한다. 멀티캐스트 섹션에서 RTCP 패킷은 RTP 패킷과 동일하게 주기적으로 멀티캐스트된다. 따라서 실제 데이터가 없어도 멤버선택과 연결이 지속되고 있음을 알려주게 된다. 또한 애플리케이션이 조건에 적응할 수 있도록 현 네트워크 상태와 수신 상태에 대한 우수한 품질을 제공한다.

2.2 SIP/SDP

2.2.1 SIP(Session Initiation Protocol)

기존의 VoIP 서비스의 기반이 되었던 프로토콜인 H.323의 단점을 보완하기 위하여 개발한 것이 SIP이다. SIP는 인터넷 자체에 좀 더 적합하고 높은 유연성을 가지며 동시에 구현이 쉽도록 만들어진 세션의 생성, 수정, 종료를 위한 신호 프로토콜로 단일 혹은 그 이상의 미디어 타입으로 이루어진 멀티미디어 회의, 인터넷 폰 및 이와 유사한 응용이 가능하다[3].

2.2.2 SDP(Session Description Protocol)

SDP는 단순히 멀티캐스트 세션 디렉토리용 뿐 만 아니라 다양한 네트워크 환경이나 응용을 위해 설계되었다. 그러나 전송 프로토콜을 포함하고 있지 않기 때문에 SDP의 전달을 위하여 SAP(Session Announcement Protocol), SIP, RTSP(RealTime Streaming Protocol), e-mail, HTTP 등과 같은 전송 프로토콜을 사용한다.

2.3 음성데이터의 처리

여러 가지 음성데이터의 변환방식 중에 PCM(Pulse Code Modulation) 방식은 가장 널리 사용되는 방식이다. 이 방식은 음성을 아날로그에서 디지털로 변환하여 양자화한 데이터를 그대로 저장한 후 재생할 때에는 그 데이터를 디지털에서 아날로그로 재변환하여 음성 파형을 만든다. PCM을 사용하여 동영상 비디오, 음성, 음악, 원격측정, 그리고 가상현실 등을 포함한 모든 형태의 아날로그 데이터를 디지털화하는 것이 가능하다.

2.4 소켓 및 네트워크 프로그래밍

소켓은 네트워크 상에서 클라이언트 프로그램과 서버 프로그램 사이의 통신 방법으로 소켓 API라고 불리는 일련의 프로그래밍 요청이나 function call로 만들어지고 사용된다. 소켓 프로그래밍은 연결형(TCP) 소켓을 개설하고 서로 연결한 다음 데이터를 송수신하고 소켓을 닫는 절차를 말한다.

네트워크 프로그래밍은 그림 1에서 보는 것과 같이 계층별로 분류할 수 있고, TCP/IP 계층을 이용하는 소켓 API와 연관되는 계층은 트랜스포트 계층으로 호스트 사이의 연결관리와 패킷 단위의 데이터 송수신을 다룬다.

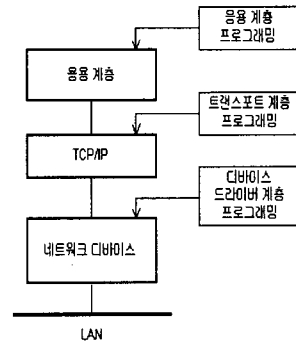


그림 1. 네트워크 프로그래밍의 계층별 분류[5]

2.5 인스턴트 메시지의 분류

인스턴트 메시징 프로그램의 주된 기능은 실시간으로 텍스트 메시지를 신속히 상대방에게 전달하는 것으로 구현 형태에 따라 크게 두 가지로 구분된다.

2.5.1 P2P(Peer-to-Peer) : 동등계층

P2P는 각 컴퓨터가 동등한 능력을 가지고 있어 어떤 컴퓨터에서라도 통신 세션을 시작할 수 있는 통신 모델을 지칭한다. 네트워크 관점의 접근으로 리소스를 분산하여 처리하고 저장하는 분산컴퓨팅기술이라고 볼 수 있다. P2P 통신 방식은 특정한 컴퓨터에 의존하지 않을 수 있으며, 어느 한 지역과 지역의 통신 회선이 파괴되어 회선을 통한 상호연락이 두절되었을 때 문제없이 사용할 수 있는 통신체계이다. 이 체계는 통신 네트워크 내에 어떠한 통신의 중심이 마스터와 슬레이브의 관계가 존재할 수 없으며 각 노드에서 다른 노드로 매우 다양한 경로를 통하여 상호간 통신

을 할 수 있다는 특징을 가지고 있다. 서버가 필요 없으므로 동종 프로그램이 실행되고 네트워크 자원 사용이 가능하다면 인스턴트 메시징 기능을 수행할 수 있다.

2.5.2 Client/Server : 클라이언트/서버

클라이언트는 다른 프로그램에게 서비스를 요청하는 프로그램이며, 서버는 그 요청에 대하여 응답을 해주는 프로그램이다. 단일 컴퓨터 내에서도 적용될 수 있지만, 네트워크 환경에서 더 큰 의미를 가진다. 네트워크 상에서 클라이언트/서버 모델은 여러 다른 지역에 걸쳐 분산되어 있는 프로그램들을 연결시켜주는 편리한 수단을 제공한다. 장점으로는 지역적인 제약 없이 접속해 사용자가 원하는 사람과 대화를 나누거나 파일을 쉽게 주고받을 수 있으나 단점으로는 중앙 서버가 모든 관리를 통합, 통제함으로써 시스템 관리와 유지보수에 많은 시간과 비용이 필요하며, 네트워크 단절 또는 서버의 불안정으로 인하여 서버 접속이 허용되지 않을 경우에는 모든 기능 수행이 불가능하다.

3. 멀티미디어 채팅 프로그램의 분석 및 설계

3.1 기존 인스턴트 메시징의 분석

일반적인 인스턴트 메시징은 메시지 전달의 기본적인 기능과 함께 여러 가지 통합된 기능을 제공함으로써 사용자에게 편리성을 제공하고 있다. 그러나 서버가 역할을 못하게 되는 경우가 발생된다면 메시징의 기능도 발휘되지 못한다. 또한 해킹이나 바이러스 침입, 네트워크 통신 두절 등의 심각한 문제가 서버에서 발생된다면 사용자에게 사용상의 불편은 물론 개인 정보의 유출뿐 만 아니라 사용자의 시스템에도 유해성분이 확산될 우려가 크다.

본 논문에서는 이러한 클라이언트/서버 기반 인스턴트 메시징의 단점을 고려하여 P2P 방식을 적용한 VoIP 기술 기반의 멀티미디어 채팅 프로그램을 구현하였다.

3.2 멀티미디어 채팅 프로그램의 설계

다음 그림 2는 VoIP 기술 기반의 멀티미디어 채팅 프로그램의 설계 구조를 간단하게 표현한 것이다.

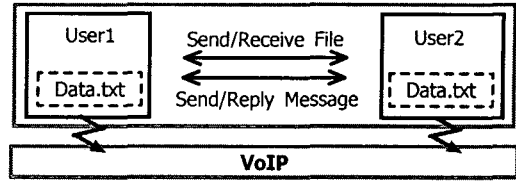


그림 2. 멀티미디어 채팅 프로그램의 실제 구조

사용자의 버디에 대한 정보는 텍스트형태의 데이터베이스로 생성되어 네트워크 상에 버디의 On/Off 상태를 파악할 수 있으며 각 사용자간에 문자 및 음성 형태의 메시지와 파일을 전송할 수 있다.

4. 멀티미디어 채팅 프로그램의 구현

Visual C++ MFC와 윈도우즈 API 함수를 사용하여 구현하였고 운영체제는 Windows 98과 Windows 2000을 사용하였다.

4.1 문자 메시지 처리 및 파일 전송

TCP를 기반으로 두 개의 포트를 정의하여 문자 메시지와 파일을 전송하는데 사용하였다. 서로 다른 포트를 사용하므로 문자 메시지의 전송과 동시에 파일 전송이 가능하고 문자 메시징과 파일간의 전송속도에는 영향을 미치지 않는다. 변수에 저장된 메시징과 네트워크 및 파일 정보는 변수 값에 따라 정의한 포트를 통하여 전송된다.

4.2 음성데이터 처리 및 전송

윈도우즈 멀티미디어 API 함수들을 이용하여 사운드 디바이스로부터 실제 데이터를 얻어와 음성데이터를 처리하였다. 윈도우즈 멀티미디어 API 함수들은 Visual C++가 인스톨된 include 디렉토리에 있는 mmsystem.h 라는 헤더파일 내부에 정의되어 있다. 실제 라이브러리는 lib 디렉토리 내 winmm.lib 파일이다. 이 파일을 프로젝트에 추가하여 컴파일한다.

입력된 음성데이터인 PCM 데이터를 사운드 디바이스로부터 받아오는 모듈과 출력하는 모듈을 통합하여 음성데이터를 처리하였다.

입력된 음성데이터를 사운드 디바이스로부터 받아오는 과정을 정리하면 표 1과 같다.

표 1. 입력된 음성데이터 받아오는 과정

과정	사용된 함수	함수설명
①	waveInOpen()	input device를 연다.
②	waveInPrepareHeader()	버퍼를 준비한다.
③	waveInAddBuffer()	준비된 버퍼를 input device에게 보낸다.
④	waveInStart()	input device를 작동시킨다.
⑤	실제 input device에게 보내준 버퍼가 오디오데이터로 채워진 경우, MM_WIM_DATA 이벤트 발생	
⑥	waveInAddBuffer()	⑤의 이벤트가 발생한 경우, 또 다른 버퍼를 input device에게 보내고 ③과정에서 보낸 버퍼에서 데이터를 읽어온다.
⑦	원할 때까지 ⑤, ⑥과정을 계속 반복한다.	
⑧	waveInReset()	input device를 멈춘다.
⑨	waveInUnprepareHeader()	버퍼를 해제한다.
⑩	waveInClose()	input device를 닫는다.

입력된 음성데이터를 사운드 디바이스로 출력하는 과정을 정리하면 표 2와 같다.

표 2. 입력된 음성데이터 출력하는 과정

과정	사용된 함수	함수설명
①	waveOutOpen()	out device를 연다.
②	waveOutPrepareHeader()	버퍼를 준비한다.
③	waveOutWrite()	out device를 작동시킨다.
④	실제 버퍼의 내용이 모두 재생되었을 경우, MM_WOM_DATA 이벤트 발생	
⑤	waveOutWrite()	④의 이벤트가 발생한 경우, 또 다른 버퍼를 out device에게 보내고 ③과정에서 보낸 버퍼에서 데이터를 읽어온다.
⑥	원할 때까지 ④, ⑤과정을 계속 반복한다.	
⑦	waveOutReset()	out device를 멈춘다.
⑧	waveOutUnprepareHeader()	버퍼를 해제한다.
⑨	waveOutClose()	out device를 닫는다.

4.4 문자채팅 및 음성채팅의 통합

Visual C++ MFC를 사용하여 문자채팅 모듈과 음성채팅 모듈을 통합하였다. 문자채팅과 동시에 음성채팅을 실행할 수 있으며 P2P 방식을 적용하였기 때문에 서버의 유지보수와 관리에 드는 비용 절감과 서버와 클라이언트간의 네트워크 단절로 인한 비접속문제를 해결하였다.

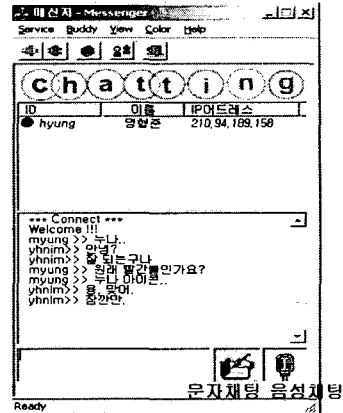


그림 3. 멀티미디어 채팅 프로그램 실행 화면

5. 결론 및 향후 연구

본 논문에서는 Visual C++ MFC와 윈도우즈 멀티미디어 API를 사용하여 문자를 도구로 하는 통신은 물론 VoIP 기술을 기반으로 하여 음성을 도구로 하는 통신 또한 가능한 멀티미디어 채팅용 통신 프로그램을 구현하였다. 또한 클라이언트/서버 중심인 기존의 서비스 구조를 탈피하여 사용자간의 직접적인 정보교환을 가능하게 하는 P2P 방식을 적용하였다.

본 논문에서 구현한 프로그램은 다자간 통신과 QoS를 보장하면서 멀티미디어를 활용한 통신이 가능하도록 멀티미디어의 통합에 대한 연구도 계속되어야 할 것이다. 이러한 연구를 통해 추후 통합메시징시스템(UMS)의 연구개발에 기여할 수 있게 될 것이다.

참고문헌

- [1] Debashish Mitra, "Network Convergence and Voice over IP," TATA Consultancy Services, March 2001
- [2] Zhang Zhanjun, Han Chengde, "A RTP-Based Architecture of Multimedia Communications for Wireless Networks," IEEE, 2001
- [3] Bostjan Vlaovic, Zmago Brezocnik, "Packet Based Telephony," IEEE, 2001
- [4] Josef Glasmann, Wolfgang Kellerer, "Service Development and Deployment in H.323 and SIP," IEEE, 2001
- [5] <http://redalert.java.rp2.co.kr/packet/network.htm>, "네트워크 프로그래밍의 분류"