MPEG-4 CELP를 이용한 실시간 다자간 통신시스템의 구현

Implementation of Real Time Multi-User Communication System with MPEG-4 CELP

김 헌 중*, 우 광 희*, 차 형 태*
(Hunjoong Kim*, Kwanghee Woo*, Hyungtai Cha*)

※ 본 연구는 정통부 초고속 정보통신 응용기술 개발 사업의 지원으로 이루어 졌습니다.

요 약

본 논문은 6~24kbit/s의 저 비트 율의 전송 율을 지원하는 MPEG-4 CELP CODEC과 실시간 처리를 위한 효율적인 알고리즘의 최적화를 통한 인터넷 환경에서의 PC-to-PC 실시간 양방향 다자간 동시 통화 시스템을 구현하였다. 현재 구현된 시스템은 MPEG-4 CELP Mode-I을 사용하여 음성신호 압축 비트 열을 생성하고 있으며, Mode-I에서 지원하는 비트율 중 18200bps 모드를 사용하고 있다. 이 경우 1프레임 당 처리하는 샘플 데이터 수는 160 샘플이고 현재 데이터 전송을 위한 데이터 package는 5 프레임이 1 package(117 byte)로 구성되어져 있으며, 동시에 4명의 사용자가 접속하여 실시간으로 다자간 양방향 통신이 가능하도록 구현되었다. 개발 환경은 Windows 운영체제 하에서 Microsoft Visual C++ 6.0을 사용하였다.
핵심용어: MPEG-4 오디오, CELP, 실시간 통신, 다자간 통신

ABSTRACT

In recent, the innovative improvement of a internet and computing environment make users desire the capability of processing information in real time. In this paper we implement a PC-to-PC real time multi-user communication system on the internet environment using the efficient algorithm for a real time processing and the MPEG-4 CELP codec which can be used for a low bit-rate coding from 6 to 24kbps. The implemented system produces a compressed bit-streams with the MPEG-4 CELP Mode-I 18200bps mode. There is 5 frames for a package and 1 frame has 160 samples. We can use this system to communicate with 4 users simultaneously in real time. The system is designed and examined on the Windows operating system.
Key words: Real-time communication, Multi-user communication, Internet telephony
투고분야: 음성처리(2.1)

I. 서 론

최근 컴퓨터 환경은 빠른 속도로 변화하고 있다. 이러한 빠른 변화 속에서도 두각을 나타내고 있는 분야가 인터넷을 통한 정보교환 분야라고 할 수 있다. 그 결과 사용자들은 기존의 정보 유지 및 획득과 같은 인터넷 정보의 최종 전달 도구였던 컴퓨터 환경에서 벗어나, 기존의 통신 시스템과 같이 모든 정보를 실시간으로 처리하기를 요구하게 되었다. 이러한 요구들은 급속한 인터넷 환경의 발전과 더불어 internet telephony service를 위한 기술개발을 통해 점차 충족되어지고 있으며, 이러한 상황 속에 그 동안 인터넷을 통한 음성통신에 대하여 많은 연구들이 수행되어져왔고 많은 시제품들이 출시되어 사용되고 있으나 통화 품질이 만족스럽지 못하거나 실시간으로 다자간 양방향 통신이 어려운 문제점들이 있었다.

본 논문은 이러한 문제점을 해결하기 위해 6~24kbit/s의 저 비트 율을 지원하고, 샘플링 율을 선택적으로 사용함으로써 8kHz 샘플링 율일 경우 300~3600Hz, 16kHz의 샘플링 율일 경우 500~7000Hz의 가변적인 대역폭을 제공하는 음성 압축 코덱인 MPEG-4 CELP를 사용하여 IP Network상에서의 다자간 실시간 통신을 위한 효율적인 알고리즘 처리방법을 적용하여 인터넷을 통한 PC-to-PC 실시간 다자간 동시 통화 시스템을 구현하였다[1][2].

II. 실시간 다자간 통신을 위한 시스템 구성

2.1. PC 기반의 실시간 디지털 오디오 처리

일반적으로 PC(Personal Computer)의 오디오 장치는 오디오 제어기에 연결되어진 A/D 변환기(녹음)과 D/A 변환기(재생)로 구성되어 있는데, 녹음과정을 예로 들 경우, 구동 프로그램은 오디오 제어기에 적절한 명령을 전달하

* 숭실대학교 전자공학과
접수일자 : 2000년 1월 19일

고 오디오 데이터를 메인 메모리에 저장하게 된다. 이 때 오디오 제어기는 DMA(Direct Memory Access) 채널을 사용하여 메인 메모리 상으로 오디오 데이터들을 전송하게 된다. 이때 다량의 데이터들을 전송하는데 있어서 DMA 채널을 이용하는 것이 CPU를 이용하는 방법보다 효율적이다.

이와 같이 녹음과 재생 과정을 통해 생성된 데이터 열은 네트워크 환경 하에서 전화와 같은 기능을 하는 실시간 기기에 적용될 수 있는데, 이러한 실시간 환경을 구축하기 위해서는 효율적인 알고리즘과 process scheduling, 충분한 하드웨어 처리 능력이 요구된다[3][4]. 이때 요구되는 오디오 장치는 오디오 데이터를 녹음하는 동안 버퍼내의 오디오 데이터를 재생할 수 있는 full-duplex를 지원해야 한다. 이와 같은 이유는 기존의 half-duplex 장치의 경우 각각의 처리 과정마다 재생 또는 녹음 모드 중 하나의 모드로만 오디오 장치를 제어할 수밖에 없는 반면, full-duplex장치는 녹음 과정과 재생 과정을 동시에 제어할 수 있는 장점을 가지고 있기 때문이다.

또한 본 논문에서와 같이 많은 양의 데이터를 연속적으로 처리하기 위해서는 streaming 방식으로 데이터를 처리해야 하는데, 이것은 녹음 과정을 예로 들 경우, 다음 그림 1에서와 같이 장치 드라이버가 녹음하는 시간과 동일한 시간에 처리되어지는 데이터를 처리하여, 장치 드라이버가 녹음 과정을 처리한 후에 application에 버퍼의 데이터를 넘겨 줄 때까지의 시간동안에 대한 데이터의 손실을 피함으로써 오디오 신호가 주기적으로 끊기는 현상을 방지하기 위함이다[5].

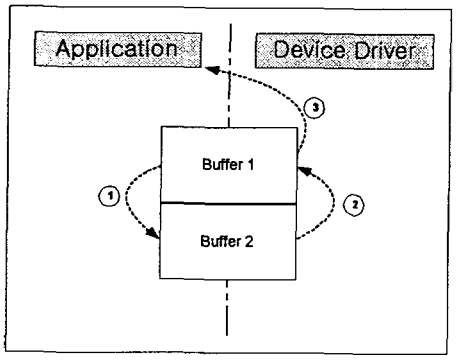

그림 1. 이중 버퍼 스트리밍
Fig. 1. Double buffer streaming.

다음의 그림 2는 이러한 full-duplex 오디오 처리능력을 통한 별도의 신호처리 과정을 통해 네트워크 환경 하에서 두 컴퓨터간의 데이터 열을 전송하는 전화와 같은 기능을 하는 적용 예를 나타내고 있다.

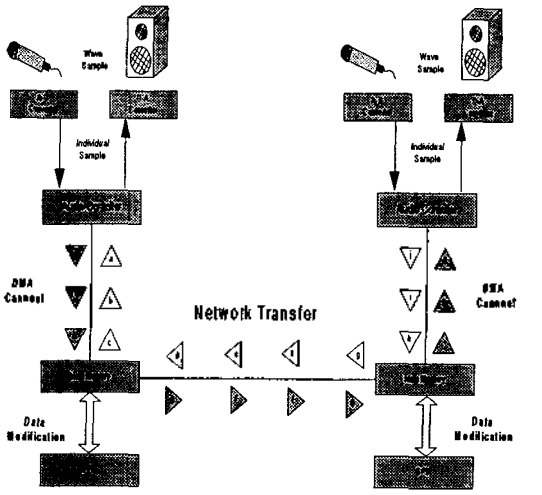

그림 2. 두 대의 컴퓨터를 통한 전 이중 오디오의 전화와 같은 예
Fig. 2. Full-duplex audio with two computers as a telephone-like application.

2.2. 실시간 처리 Scheduling

PC에서 디지털 오디오 신호의 처리 시에는 약간의 지연이 필연적으로 발생하게 되는데, 부가적인 신호 처리나 네트워크 전송의 경우를 제외하더라도 이러한 작은 지연은 버퍼에 데이터를 packing하는 것과 이러한 packing된 데이터들을 메모리로의 전송 또는 메모리로부터의 전송에서 발생한다.

결국 녹음 과정과 재생 과정사이에서 기본적으로 발생하게 되는 최소 지연 시간은

$$\text{최소지연시간} = \text{버퍼의 재생과 기록 시간} + 2 \times DMA \text{ 전송 시간} \quad (1)$$

가 된다. 결국 컴퓨터 프로그램상의 "real time"이란 일정 시간 내에 작업을 완전히 수행하는 것으로 설명되어질 수 있는데, 네트워크 전송을 예로 들 경우 실시간 오디오 처리란 녹음 과정과 재생 과정을 제한적인 지연 시간의 존재 하에서 (즉, 두 개체간 통신 환경에 지장을 주지 않을 정도) 두 작업을 제약 없이 수행하는 것이다.

결국 최종적으로 발생 가능한 지연 시간은

$$\text{최종 지연시간} = \text{최소 지연시간} + \text{신호처리 시간} + Network\text{전달 시간} \quad (2)$$

가 된다. 그러나 다음 그림 3에서 살펴볼 수 있듯이 기본적으로 발생하게 되는 최소 지연 시간에서 DMA 전송을 위한 시간은 CPU의 부하 없이 별도로 동작하는 DMA를 통한 데이터 처리 시간으로 초 당 수십 mega byte의 데이터를 처리하지 않는 한 현재의 컴퓨터 환경에서는 크게 고려대상이 되지 않으므로 결국 최소화해야 할 지연 시간은 CPU에 부하를 가져오는 신호 처리를 위한 시간,

네트워크 전송을 위한 시간과 버퍼에 저장되어 있는 데이터의 재생과 녹음을 위한 시간의 합이 된다. 그러나 참고적으로 PC상에서의 오디오 데이터의 재생과 녹음 처리 과정은 오디오 데이터가 저장되어 있는 메모리 번지를 통해 오디오 제어기에게 명령을 주면 사운드 장치는 오디오 제어기와 DMA 채널을 이용하여 CPU와는 별도로 구동되어 재생과 녹음 처리 과정을 수행한다. 그러므로 추가적인 지연 시간의 발생과 데이터의 손실 없이 네트워크 환경에서 원활한 데이터 통신이 이루어지기 위해서는 오디오 데이터들을 녹음하거나 재생하는 시간 내에 별도의 신호 처리 과정과 네트워크 전송 과정을 수행하면 된다.

버퍼의 재생과 기록시간≥(신호 처리시간+*Network* 전달시간) (3)

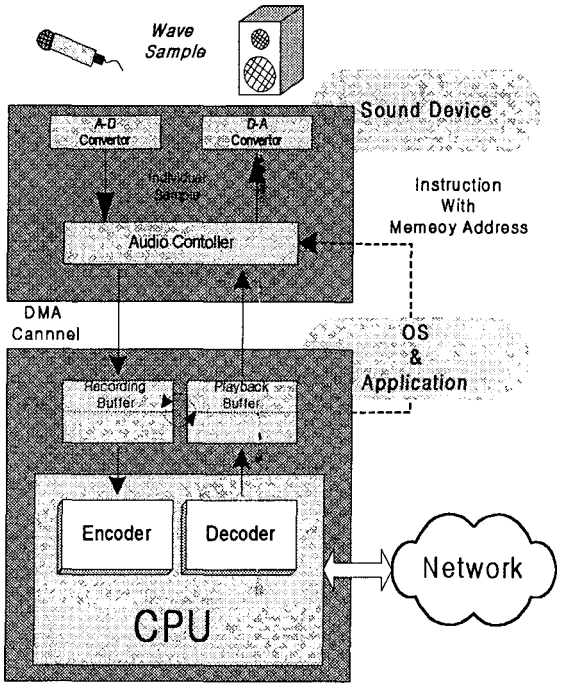

그림 3. 전 이중 방식 디지털 오디오 처리 과정
Fig. 3. Digital audio processing with full-duplex.

그러나 초기 상태에서 버퍼에 파형 오디오 데이터를 녹음할 만큼의 지연 시간이 발생하게 되는데, 이러한 초기 지연 시간을 최소화하기 위해서는 녹음이나 재생에 사용되어지는 버퍼의 크기를 줄여 feedback 시간을 줄일 필요가 있으나, 데이터 손실이나 추가적인 지연 시간의 발생 없이 별도의 신호 처리(encoding + decoding) 시간과 네트워크 전송을 위한 시간을 확보하려면, 이러한 처리 과정을 수행할 수 있을 만큼의 시간을 보장할 수 있을 정도의 재생 시간이나 녹음 시간을 가질 수 있는 버퍼 크기가 되어야 한다. 다음 그림 4는 이러한 예에 대한 1:4 통신의 경우에 대한 processing scheduling의 예를 나타내고 있다.

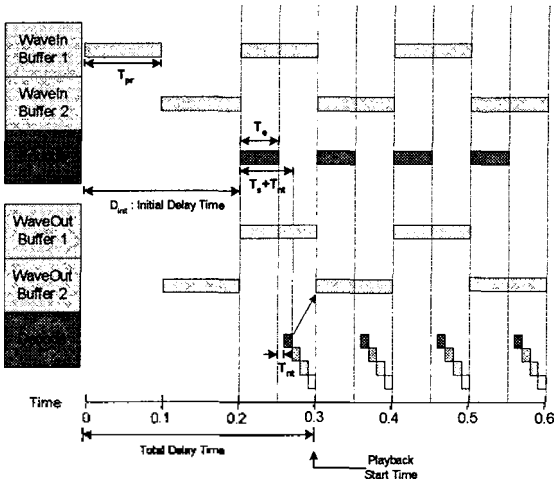

그림 4. 처리 예정 표
Fig. 4. Processing scheduling chart.

2.3. System 구성

다음 그림 5는 이와 같이 구현된 시스템의 전체 구성도이다.

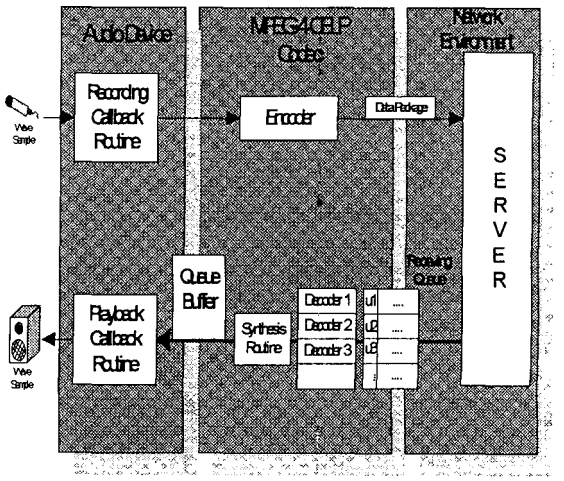

그림 5. 시스템 블록도
Fig. 5. System block diagram.

구현된 시스템에서 오디오 장치 드라이버를 이용한 재생 과정에서는 지속적으로 파형 오디오 데이터 stream을 녹음 및 재생하게 되고, 녹음 재생 과정에 포함되어 있는 MPEG-4 CELP 부호화기에서는 이러한 매 녹음 재생 때마다 생성되어진 파형 오디오 데이터 블록을 부호화하여 비트 열 package를 서버에 전송하게 되며, 서버 프로그램에서는 수신되어진 비트 열 package를 제외한 다른 사용자로부터 전송되어진 package들을 사용자에게 전송하게 된다. 이 때 각각의 사용자에 대한 데이터 package에는 Network 환경에 따라 jitter가 발생하게 되는데, 이러한

jitter로 인한 음성의 끊김 현상을 방지하기 위해 사용자별 수신 큐(queue) 버퍼를 구성하였으며, 이렇게 저장되어진 데이터 package들은 장치 드라이버에서 재생 신호가 발생할 때마다 각각의 복호화기를 통해 파형 오디오 데이터로 복원된 후에 합성 과정을 거쳐 재생되어 지게 된다.

그림 6은 이러한 복원 과정과 재생 callback 과정에서의 버퍼 처리 과정을 나타내고 있다.

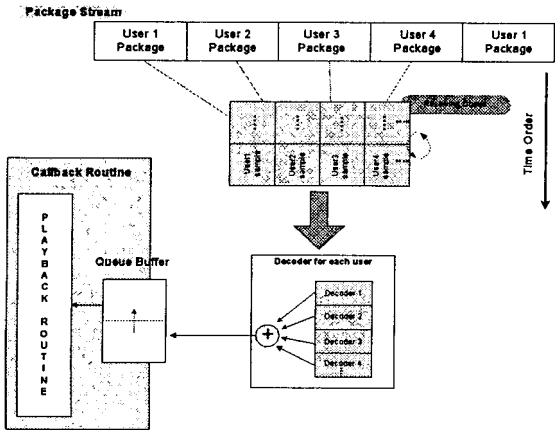

그림 6. 버퍼 제어
Fig. 6. Buffer control.

그림 7은 MPEG-4 CELP Mode-I을 사용하고, Mode-I에서 지원하는 비트 율 중 18200bps 모드를 사용하였을 경우에 대한 구성되어진 전송 package의 기본 구성 형식을 나타내고 있다. 그 구성을 살펴보면, 15 프레임의 비트열을 하나의 package로 구성하며, 2 byte의 header 정보가 추가되어진다. 이 때 2 byte의 header 정보 중 첫 번째 byte의 상위 4bit는 통화중인 총 인원을, 하위 4bit는 사용자 식별용 인덱스 정보를 포함하며, 두 번째 byte는 비트열에 대한 에러 점검을 위한 정보가 저장되어 진다.

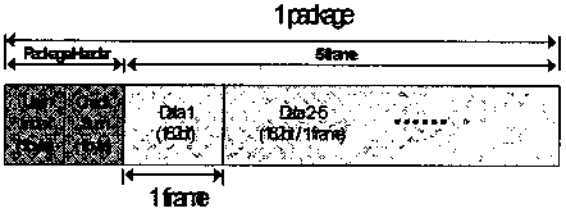

그림 7. 패키지 구성
Fig. 7. Package format.

III. 실험 및 고찰

구현된 시스템에 대한 실험은 시스템에 요구되어지는 자원에 대한 검증과 기존 제품들과의 특성 비교를 통해 이루어 졌으며, 실험을 위한 환경은

Operating System : Microsoft Windows 98
- Second Edition
CPU : Pentium Celeron 366A
RAM : 64M
Development Tool : Microsoft Visual C++ 6.0
Sound Card : Sound Blaster 64 PCI
LAN 환경 : T1
Sampling Frequency : 8kHz
Number of Channel : 1 (Mono)

과 같으며, MPEG-4 CELP Mode-I 을 사용하였고, Mode-I에서 지원하는 비트 율 중 18200bps 모드로 비트 열을 생성하였다. 이 때 CELP 코덱의 전체 신호 처리 시간을 줄이기 위해 8kHz로 샘플링된 파형 오디오 데이터를 적용하였는데, 이 경우 데이터 전송을 위한 package의 샘플 수 P_{sn}은

$$P_{sn} = F_{sn} * F_{pn} \tag{4}$$

단, F_{sn} : 프레임당 샘플 수 : 160
F_{pn} : *package*당 프레임 수 : 5

으로, P_{sn} 샘플 단위로 재생하거나 녹음하게 된다. 이 때 소요되는 재생 시간이나 녹음 시간, T_{pr}은

$$T_{pr} = \frac{P_{sn}}{F_s} * N_{ch}[\text{sec}] \tag{5}$$

단, F_s : *sampling freq.* : 8*kHz*
N_{ch} : *channel* 수 : 1(*Mono*)

로 0.1초가 된다. 그러므로 초기 지연 시간, D_{init}은

$$D_{init} = B_n * T_{pr}\ [\text{sec}] \tag{6}$$

단, B_n : 녹음 및 재생을 위한 버퍼 수

로 0.2초가 되며, 이 때 그림 4의 processing scheduling chart에서 살펴볼 수 있듯이 신호 처리를 위한 시간과 네트워크 전송 시간은 이러한 T_{pr}내에 이루어져야 하는데, 현재 사용하고 있는 MPEG-4 CELP 코덱에서는 비트 율을 18200bps 모드로 사용할 경우 1 프레임의 파형 오디오 데이터를 182bits로 코딩하는데 소요되는 신호 처리 과정에서 사용되어 지는 시간, T_s는

$$T_s = (T_e + T_d) * F_{pn} \tag{7}$$

단, T_e : *encoding time* for 1 *frame* : ≒ 11*ms*
T_d : *decoding time* for 1 *frame* : ≒ 2*ms*

으로 65ms가 된다.

이 때 요구되어지는 네트워크 전송 시간, T_{nt}은

$$T_{nt} < (T_{pr} - T_s) \quad (8)$$

인 조건을 만족할 만한 네트워크 환경이 보장되어야만 네트워크 전송 시에 발생하게 되는 지연 시간으로 인한 데이터 손실을 피할 수 있게 된다.

또한 여기서 다중 사용자가 사용하는 환경이라면, 추가되어지는 사용자 수에 따라 신호 처리를 위한 시간이 비트 열 package를 복원하기 위해, $T_d * F_{pn} = 10ms$씩 증가하게 되며, 추가된 사용자 package 전송을 위해, 그에 따른 네트워크 환경이 보장되어야만 한다.

다음 표 1은 구현된 시스템과 기존의 출시된 제품들과의 특징 비교이다. 기존의 제품들은 다자간의 음성 통신을 지원하기 위해 통화 품질이 저하되는 것을 감수하는 경향이 있으며, 실제 대화 환경과 같은 동시 다자간 통신을 지원하지 않는 것을 알 수 있다. 즉 다자간 통신을 지원하더라도 두 사람이 통화 중에 있으면 다른 사람은 그 대화에 동참하거나 대화하고자 하는 희망 조차 전달할 수 없다.

표 1. 기존 제품과의 특성 비교
Table 1. The comparison with other products.

제 품	CODEC	Sampling freq.	특 징
Netmeeting 3.01	G.723	8 kHz	● Silence suppression ● 1:N Communication → No Simultaneously
AOL Instant Massanger v. 3.5	G.723	8 kHz	● 1:1 Communication
CU-seeme v. 3.12	DigiTalk	8 kHz	● 1:N Communication →No Simultaneously ● Video Conference
Internet Phone v.5.0	Vocal tec VSC	8 kHz	● 1:1 Communication
Net2phone	Sound FDX	8 kHz	● 1:1 Communication ● IP Gateway (PC-to-Phone)
VDO Phone v. 3.03	G.723	16 kHz	● 1:1 Communication ● Video Conference
구현된 시스템	MPEG-4 CELP	8 kHz	● 1:N Communication → Simultaneously

구현된 시스템은 15명의 숙련된 평가자가 실제 대화 환경에서 MOS 테스트를 수행한 결과 4.5 정도의 우수한 성능을 가지고 있어 실제 음성과 별 차이가 없으며 4명이 동시에 말을 해도 다른 3명의 소리가 동시에 다른 화자에게 전달되는 기능을 확인하였다.

IV. 결 론

본 논문에서는 full-duplex 오디오 장치를 이용하여 인터넷을 통한 PC-to-PC 실시간 다자간 동시 통화 시스템을 구현하였다. 또한 통화 품질의 지하를 최소화하고 실시간으로 양방향 통신이 가능하도록 저 비트 율의 전송율을 지원하는 음성신호 압축 코덱과 실제 대화 환경과 같은 다자간 실시간 통신을 위한 효율적인 process scheduling 알고리즘에 대하여 연구하였다.

현재 구현된 시스템은 MPEG-4 CELP Mode-I 을 사용하여 음성신호 압축 비트 열을 생성하고 있으며 Mode-I 에서 지원하는 비트 율 중 18200bps 모드를 사용하고 있다. 이 경우 1 프레임 당 처리하는 샘플 데이터 수는 160 샘플로, 현재 데이터 전송을 위한 데이터 package는 5 프레임으로 117byte로 구성되어져 있으며, 동시에 4명의 사용자가 접속하여 양방향 통신이 가능하도록 구현되었다. 또한 실질적인 멀티미디어 환경을 구축하기 위해서는 현재까지 개발된 시스템을 기반으로 비디오 신호와의 전목을 통한 실시간 다자간 화상 통신 시스템 구현을 위한 연구와 streaming 전송 기술 개선을 위한 연구가 계속 수행되어져야 할 것이다.

참 고 문 헌

1. ISO/ITC 1/SC29/WG11 "Information Technology - Very Low Bitrate Audio Visual Coding. FCD 14496-3 Part3: Audio," May 1998.
2. Kondoz, "Digital speech coding for low bit rate communication systems," JOHN WILEY & SONS, 1994.
3. Dong Lin, "Real-Time Voice Transmissions over the Internet," M.S. Thesis, Dept. of Electrical and Computer Engineering, Univ. of Illinois, Urbana-Champaign, Dec. 1998.
4. G. Held, "Voice Over Data Networks. New York," McGraw-Hill, 1998.
5. Microsoft Development Library, 1998.

▲김 헌 중(Hunjoong Kim)

1997년 2월 : 관동대학교 전자통신공학과(공학사)
1999년 2월 : 숭실대학교 전자공학과(공학석사)
1999년 3월 ~ 현재 : 숭실대학교 전자공학과 박사과정 재학
※ 주관심분야 : 오디오 및 음성신호처리, 통신 신호처리, ASIC 설계

▲우 광 희(Kwanghee Woo)

1998년 2월 : 숭실대학교 전자공학과 (공학사)
1999년 3월 ~ 현재 : 숭실대학교 전자공학과 석사과정 재학
※ 주관심분야 : MPEG-4 Audio, ASIC 설계

▲차 형 태(Hyungtai Cha)

1985년 : 숭실대(공학사)
1988년 : The University of Pittsburgh (공학석사)
1993년 : The University of Pittsburgh (공학박사)
1993년 ~ 96년 : 삼성전자 신호처리 연구소 선임연구원
1995년 ~ 1998년 : 숭실대학교 전임 강사
1998년 ~ 현재 : 숭실대학교 조교수
※ 주관심분야 : Audio / Video Coding, Morphology, ASIC 설계