

패킷음성 프로토콜의 구현

正會員 李 相 吉* 正會員 申 炳 喆** 正會員 金 潤 寬*

Implementation of Packet Voice Protocol

Sang Kil Lee*, Byung Cheol Shin**, Youn Kwan Kim* *Regular Members*

要 約

Ethernet에 음성신호의 전송을 위한 패킷음성 프로토콜을 PC 구현하였다. 구현한 패킷음성 프로토콜은 CCITT의 G.764 패킷음성 프로토콜을 수정하여 사용하였다.

이를 위하여 ethernet을 통한 음성통신을 하기 위한 하드웨어 시스템을 설계하였는데, 전화기 인터페이스부분, 음성처리부분, PC인터페이스부분 및 제어부분으로 나누어진다. 소프트웨어는 OSI 7계층에 맞추어 설계하였으며 ethernet device driver, 전화기 인터페이스 하드웨어 driver 그리고 패킷음성 프로토콜 처리 부분으로 구성되어 있다. 실제 실험결과 ethernet을 통하여 전화를 연결하고 만족한 음성통화를 할 수 있었다.

ABSTRACT

In this paper, the packet voice protocol for the transmission of voice signal onto ethernet is implemented in a personal computer (PC). The packet voice protocol used is a modified one from CCITT G.764 packetized voice protocol.

The hardware system to facilitate the voice communication onto ethernet is divided into telephone interface, speech processing, PC interface and controllers. The software structure of the protocol is designed according to the OSI seven layer architecture and is divided into three routines: ethernet device driver, telephone interface, and processing routine of the packet voice protocol.

Experiments through ethernet with telephone interface show that this packet voice communication achieves satisfactory quality when the network traffic is light.

I. 서 론

지금까지 음성은 회선교환방식을 이용한 공중전화망에서 주로 서비스하였다. 그러나 패킷교환망에 대한 기술이 발전하고 정보통신분야에 광범위하게 사용됨에 따라 제공하는 서비스의 종류를 늘릴 필요가 생겨 중

합정보통신망을 향한 궁극적인 목표아래 패킷교환망에 음성을 서비스하고자 하는 욕구가 생겨났다. 패킷교환망에 음성을 서비스하는 것은 그 경제적, 기술적인 잇점으로 인해 많은 사람들의 관심이 되어 왔으며, 현재 오디오, 비디오까지도 패킷 단위로 보내려는 연구가 활발히 진행되고 있다[1-3].

일반적으로 음성은 음성구간과 묵음구간으로 나눌 수 있다. 음성구간은 40% 정도이기 때문에 그 구간만 패킷화해서 보낸다면 전송효율을 두 배 이상 향상시킬

*金星情報通信(株)

**韓國科學技術院 電氣 및 電子工學科

論文番號: 93-185

수 있으며, 복음구간을 데이터 트래픽이나 다른 음성 트래픽이 이용할 수 있으므로 채널의 이용도를 높일 수 있다. 그러나 이러한 방법은 종래의 수과수 분할 다중화 방식이나 시분할 다중화 방식에서는 구현하기가 어렵지만 패킷교환방식에서는 간단히 구현할 수가 있다. 또한 회선교환방식에서는 호(call)가 설정되면 호가 해제될 때까지 그 채널을 다른 가입자가 사용할 수 없게 된다. 그러나 패킷교환방식은 회선교환방식과 달리 데이터를 서비스할 경우 일반적으로 delay보다는 에러 없는 전송을 중요시하기 때문에 별로 문제되지 않는다. 반면에 음성은 coder의 종류에 따라 다르지만 약 1% 이하의 손실은 알아듣기에 큰 지장이 없어서 어느 정도의 채널 에러를 허용하지만 음성 패킷의 end-to-end delay가 200ms 이상이면 통화에 지장이 생긴다. 또 일반적으로 데이터 전송시에는 에러가 없는 전송을 하기 위해 재전송 알고리즘이 필요하다. 그러나 음성 전송시에는 실시간으로 전송을 하면서 어느 정도의 에러를 허용하므로 재전송 알고리즘이 필요하지 않는 대신 음성의 질을 높이기 위한 음성 재생 알고리즘을 필요로 한다.

본 연구에서는 CCITT의 G.764 패킷음성프로토콜을 ethernet 환경에 맞게 수정하여 ethernet 상에서 음성통신을 구현해 보았다. II 장에서는 패킷 음성 통신에 대해 살펴보고 III 장에서 G.764 패킷음성 프로토콜과 패킷음성 프로토콜을 ethernet에 접속시킬 때의 방안을 설명하고 IV 장에서는 직접 구현한 것으로 ethernet 상에서의 음성통신을 위해 필요한 소프트웨어와 하드웨어의 구조에 대해 살펴보고 V 장에서 실험결과를 보이고 VI 장에서 결론을 맺는다.

II. 패킷 음성 통신에 대한 고찰

데이터를 패킷 단위로 보내는 패킷교환이 발달함에 따라 패킷교환기능을 제공하는 common carrier와 사설 LAN에서 패킷교환의 활용이 폭발적인 성장을 보였다. 패킷교환의 주요한 잇점은 대부분의 데이터 트래픽의 bursty한 성질을 이용함으로써 데이터가 존재하지 않는 동안에는 망의 자원을 다른 traffic이 이용하도록 할 수 있고 따라서 전송설비들을 보다 효과적으로 사용할 수 있다는 것이다. 최근에는 컴퓨터를 이용하여 멀티미디어에 관한 정보를 처리하려는 욕구가 높아져 하나의 단일화된 패킷교환망에 데이터 뿐만 아니라 음성과 비디오 트래픽까지도 공급하고자 하였다. 특히 LAN

에서 음성을 서비스하기 위한 연구가 활발히 진행되었다[4-12]. 패킷교환망에서 음성 전송시 패킷의 전송 delay는 패킷의 크기, 음성의 코딩율, 채널 용량, throughput 등의 영향을 받는다. 따라서 이러한 파라미터들을 적절히 조화시키는 것이 패킷교환망에서 음성을 서비스하는 데 매우 중요하다. 또한 수신단에서 수신된 음성 패킷을 적절히 이용하여 음성의 질을 크게 떨어뜨리지 않는 음성 재생알고리즘과 상호간에 통화를 하기 위한 호 설정/해제 과정 및 전송효율을 높이기 위한 복음구간 검출 알고리즘 등이 필요하다.

음성과 같은 실시간 트래픽의 주기적인 성질과 네트워크 내에 패킷의 비주기적인 전송사이의 본질적인 차이로 인해 그것을 보상해야 하는 부가적인 일이 필요하다. 그리하여 음성을 패킷 단위로 전송시에 수신단에 도착하는 각 패킷의 delay는 균일하지 않다. 이것은 정확한 동기를 요구하는 음성통신에는 치명적인 약점이 될 수 있다. 이러한 점의 해결책으로 음성재생알고리즘을 사용하는 것이며 이의 주요 역할은 도착하는 패킷을 buffering한 후 재생할 시간을 결정해서 내보내는 기능과 도착한 패킷에 대한 재생이 끝난후 다음 패킷이 도착할 때까지의 동작을 결정하는 기능이다.

도착한 패킷의 재생이 끝난 후에 더 이상 재생할 패킷이 없는 경우는 다음 패킷이 손실되었거나 수신단에서 delay가 커져 미처 도착하지 못했을 수가 있으며 음성구간이 끝나고 복음구간이 시작되어 도착할 패킷이 없을 수도 있다. 이러한 경우들을 수신단에서는 구별하지 못한다. 그리하여 다음 패킷이 도착할 때까지 수신단이 적절한 동작을 취해야 한다.

이러한 fill-in방법의 선택은 음성 encoder의 종류, 패킷의 크기, network의 delay variance 특성을 고려해서 선택한다. 음성 패킷의 재생 알고리즘에는 null timing information(NTI), incomplete timing information(ITI), complete timing information(CTI), adaptive multi-rate timing information(AMTI), adaptive multi-rate acoustic information(AMAI), G.764에서 사용된 방법, real-time adaptive 방법 등 여러가지가 있다[12].

복음구간 검출 알고리즘을 사용하는 목적은 통신채널의 효율적인 사용이다. 한 예로, 전화통화에서 speech activity는 단지 40%밖에 되지 않는다는 사실은 잘 알려져 있다[19]. 따라서, 네트워크 환경에 적절한 알고리즘의 사용은 수용할 수 있는 traffic 양을 두 배로 늘릴 수도 있는 것이다. 이 알고리즘은 오래전부터

많은 연구가 진행되었으며, 간단한 예로는 음성의 에너지와 영교차율을 사용하는 방법이 있다.

III. 패킷음성 프로토콜의 구현

1. 개요

패킷음성 프로토콜은 기존의 데이터용 프로토콜과는 달리 음성재생 알고리즘과 목음구간 검출 알고리즘이 필요하다.

본 연구에서는 사용된 프로토콜의 기반이 되는 음성프로토콜로 G.764 패킷음성 프로토콜에 대해 살펴보고 ethernet에 적합한 프로토콜에 대해 수정사항들을 살펴보겠다.

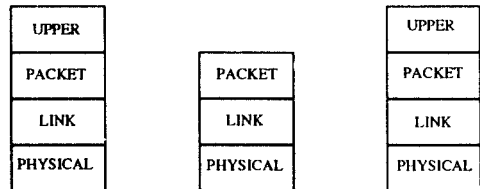
2. G.764 패킷음성 프로토콜(PVP)[16]

G.764 PVP는 CCITT 권고안으로 영구가상회선에 적용하기 위해서 만든 패킷음성 프로토콜이다. 음성은 125 μ s마다 샘플링을 하고 128개의 샘플들을 모아 16ms마다 하나의 패킷을 생성한다. 음성구간에만 패킷을 전송하고 목음구간에는 패킷을 전송하지 않는다. 그리고 폭주 발생시에는 블록버림기능을 적용할 수도 있다. 정해진 재생시간보다 먼저 도착한 패킷은 큐에서 대기 후 재생 알고리즘에 의해 재생하고, 보다 늦게 도착한 패킷은 버린다.

G.764 PVP의 계층 구조를 살펴보면 physical layer, link layer, packet layer로 나누며 그림1에 나타나 있으며, packet layer는 보통 OSI 참조모델에서는 network 계층에 해당하고, 이러한 계층구조는 데이터 통신에서의 기본사항이므로 더 이상의 설명은 생략한다. 그림 2와 3에는 각각 음성프레임과 신호프레임의 형태가 나타나 있다. G.764 PVP의 link layer에서는 기존의 데이터용 프로토콜과는 달리 재전송을 요구하는 error recovery 과정을 제공하지 않고자 high-level data link control(HDLC)에서 보는 바와 같이 unnumbered information(UI) frame의 구조를 사용하였다. Voice frame의(UIH) frame을, signaling frame의 경우에는 UI frame을 사용하고 있다. Signaling frame의 경우에는 frame 전체에 대해 CRC(frame heder and packet header check sequence)를 적용하고 음성 샘플들에 대해서는 CRC를 적용하지 않는 것이 다르다. 그리고 frame header의 address field는 permanent virtual circuit상에서 동작하기 때문에 미리 정해져 있으며 각 노드에서 정해진 값으로 바뀌어

frame이 전송된다.

그림2에서 'M(More)' 비트는 본 프레임에 이어 다음 프레임에 연속된다는 뜻으로 talkspurt 기간이 연속됨을 의미한다. 'M₂M₁' 필드는 음성 코딩으로 embedded coding을 채택한 경우에 네트워크가 congestion되었을 때 버릴 수 있는 블록의 총 수를 나타내고, 'C₂C₁' 필드는 중간노드에서 블록을 버릴 때마다 1만큼 감소되며 아직도 버릴 수 있는 블록의 총수를 나타낸다.



- Originating endpoint : User data가 PVP packet으로 만들어지는 노드
- Intermediate node : PVP packet 수신시 packet format을 변환시키지 않고 전달하는 노드
- Terminating endpoint : PVP packet의 목적지 노드

그림 1. G.764의 계층구조

Fig. 1. Layered architecture of G.764 PVP

Address (upper subfield)		0	0
Address (lower subfield)		1	
UIH Control field			
1	1	1	1
P	1	1	1
Protocol discriminator			
0	1	0	0
0	1	0	0
Block Dropping Indicator			
R	R	M ₂	M ₁
R	R	C ₂	C ₁
Time Stamp			
M	R	R	
Coding Type			
Sequence number		Noise	
Non-droppable blocks			
Optionally droppable blocks			
Header check sequence			

- M : More bit
- R : Reserved bit
- SEQ : 0, 1-15
- P : Poll bit
- M₂M₁ : 버릴 수 있는 블록 총수
- C₂C₁ : 버릴 수 있도록 남아 있는 블록 총수

그림 2. G.764의 음성프레임 형태

Fig. 2. Voice frame format of G.764 PVP

Address (upper subfield)							0	0
Address (lower subfield)							1	
UI Control field								
0	0	0	P	0	0	1	1	
Protocol discriminator								
0	1	0	0	0	1	0	0	
Block Dropping Indicator								
R	R	0	0	R	R	0	0	
Time Stamp								
M	R	R	R	R	R	R	N/A	
0	0	0	0	A	B	C	D	
Frame Check Sequence								

- N/A(Normal/Alarm) : Virtual circuit 상의 에러 표시
- ABCD signaling bits : 전송 채널상의 현재신호 상태 표시

그림 3. G.764의 신호프레임 형태
Fig. 3. Signaling frame format of G.764 PVP

다음으로 수신단에서 음성을 재생하기 위한 알고리즘에는 먼저 build-out delay(BOD) time(패킷 허용 최대 지연시간)이 설정되고 수신한 프레임 내의 time stamp 값에 따라서 재생을 하게 되는 데, 음성구간의 첫 음성패킷과 모든 신호 패킷 그리고 손실된 패킷의 다음 패킷에 대해서는 패킷최대허용 지연시간에서 내트릿을 통한 지연시간(time stamp)의 차만큼 지연한후 재생한다. 이와 달리 순서대로 전송된 패킷에 대해서는 곧바로 재생한다.

3. Ethernet에 적합한 패킷 음성 프로토콜

Ethernet에서 음성을 서비스하려면 ethernet 환경에 적합한 음성프로토콜을 사용하여야 한다. 그리고 그 프로토콜에서도 가장 중요하게 고려할 점은 트래픽이 많아지면 ethernet의 delay의 분산이 크게 되기 때문에 수신단에서의 음성 재생 알고리즘을 적절하게 선택하는 것이라 할 수 있겠다.

이 절에서는 먼저 필요한 패킷들의 종류, 음성프로토콜의 각계층에서 필요한 필드, ethernet에 적용시 수신단에서의 재생알고리즘 선택시 고려해야 할 사항 그리고 ethernet에 적절하다고 생각되는 알고리즘을 살펴보겠다.

첫째로, ethernet에 음성을 서비스하기 위해 필요한

패킷들은 가입자간에 음성 샘플들을 전송하기 위한 음성 패킷과 호의 설정/해제를위한 call control 패킷 그리고 호가 설정된 상태인데도 불구하고 time out 으로 호가 해제되지 않도록 일정한 시간마다 call refresh 패킷을 보내는 것이 필요하다.

다음으로 각 계층에서 필요한 필드로는 G.764 패킷음성 프로토콜과 비슷하다. 우선 사용할 프로토콜을 구분하는 protocol identifier field, 음성 패킷과 call control 패킷 및 refresh 패킷들을 구별하는 packet identifier field가 필요하며, call control 패킷의 경우 call control type field가 필요하다. 그리고 음성 재생시에 필요한 sequence number, time stamp 등이 필요하며 이것은 사용되는 음성 재생 알고리즘에 따라 사용유무가 결정된다.

패킷음성 프로토콜을 ethernet에서 구현할 경우 고려할 사항을 살펴보면 수신단에서 음성 패킷 수신시 어떤 음성 재생 알고리즘을 적용하는가가 관건이 된다. 일반적으로 time stamp field와 sequence number field를 사용해서 음성재생 알고리즘을 구현한다. Time stamp는 수신단의 음성구간사이의 가변 delay에도 불구하고 정확한 복음구간의 gap duration을 계산하여 음성을 재생하고자 하기 위한 것이다. 부정확한 gap duration은 특히 syllables 또는 한 phrase내의 단어들 사이의 짧은 gap에 대해 재생된 음성에서 분명한 질적손상을 가져올 수가 있다. 다음으로 time stamp는 패킷 손실과 복음구간을 구별하지 못하므로 손실된 패킷의 김출을 위해 sequence number를 사용한다.

Time stamp의 정확한 값은 패킷이 생성되고 난 후의 음성 패킷이 전송되기까지의 누적된 delay가 되는 데 ethernet은 데이터를 전송할 목적으로 만들어졌기 때문에 전송시 곧바로 전달될 수도 있지만 충돌이 일어나면 전송이 되지 않고 truncated binary exponential backoff algorithm을 사용하여 재전송을 시도한다. 이 경우 프레임 전송을 맨 처음 시도한 경우가 실제로 그 프레임이 전송된 시간이 아니라 충돌시 재전송 알고리즘을 거쳐 시간이 지연된 후 충돌이 없는 후에야 실제로 전송이 된 것이다. 이러한 충돌시 재전송에 의해 지연되어 실제로 전송되기까지의 시간을 포함하여 time stamp값으로 하여야 하는데 일단 처음 전송 후에는 ethernet MAC 프로토콜이 처리하므로 이것이 불가능하다. 그리하여 ethernet 환경에서는 G.764 프로토콜의 음성재생 알고리즘은 적당하지 않다. 위와 같은 사항을 고려한다면 ethernet환경에서 송신단에서 프레임을 송

신하기까지 겪는 정확한 누적된 delay를 기록하는 것이 불가능하므로 time stamp field를 사용하지 않는 방법을 사용해야만 하겠다. 즉 본 연구에서는 sequence number를 잘 활용하는 방법으로 null timing information방법을 사용하였다.

IV. 패킷 음성 프로토콜의 구현

1. 개요

지금까지 패킷 음성 프로토콜에 대하여 살펴보았다. 본 논문에서는 III장에서 설명한 내용을 고려해서 적절한 프로토콜을 구현하고 ethernet을 통해 음성통신을 하였다.

음성 터미날로는 상용의 2선식 DTMF전화기를 사용하였다. 그리고 전화기와의 인터페이스 및 음성의 A/D, D/A변환을 위한 하드웨어를 구현하고 패킷 음성 프로토콜과 하드웨어 인터페이스에 관한 소프트웨어를 구현하였다.

본 논문에서는 PC-to-PC 음성 통신을 하였다. PC-to-PC 음성 통신으로는 PC내에 ethernet adaptor와 전화기 인터페이스 카드를 장착해서 ethernet을 통해 음성 통신을 하였다.

다음 절에서 본 논문에서 구현한 패킷 음성 프로토콜과 구현된 하드웨어와 소프트웨어의 구조에 대해서 서술하겠다.

2. 구현한 패킷 음성 프로토콜

III장의 3절에서 ethernet 적용시 고려해야 할 사항들을 살펴보았다. 이를 바탕으로 적절한 프로토콜을 구현하였으며 이를 물리, 링크, 패킷, 상위 계층으로 나누어 설명하겠다.

1) 물리계층

Ethernet상에서 음성통신을 하므로 ethernet의 물리계층과 동일하며 채널 용량은 10Mbps이다.

2) 링크계층

LAN 프로토콜의 경우 링크계층은 MAC 부계층과 LLC 부계층으로 나누어진다. 본 연구에서는 ethernet MAC 프로토콜을 사용하며 LLC 부계층은 사용하지 않는다.

3) 패킷계층

상위 계층으로부터 입력된 음성 샘플과 목음구간검출결과에 대한 정보를 받아 패킷을 만들 것인지를 결정한다. 이 음성 샘플들이 목음구간이 아니면 패킷 헤더를 붙여 전송한다. 패킷 형식은 G.764 패킷음성 프로토콜을 기본으로 하였으며 사용하지 않는 필드는 0으로 하였다. 음성 재생 알고리즘시 사용되는 필드로는 sequence number 필드만을 사용하였다. Call 패킷의 경우는 상위 계층으로부터 받은 정보에 근거하여 적절한 call 패킷을 만들어 전송하며 수신한 경우에도 그에 해당하는 행동을 취한다. 패킷의 제일 앞 부분에는 packet identifier field로 1byte를 추가해 사용하였고 다음으로 protocol discriminator 필드가 있다.

본 연구에서는 음성 프레임은 G.764 프로토콜의

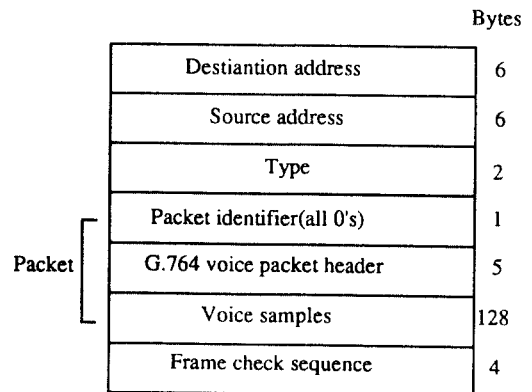


그림 4. 음성프레임 형태
Fig. 4. Voice frame format

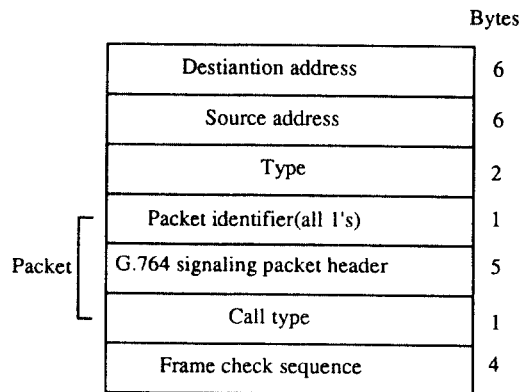


그림 5. Call 프레임 형태
Fig. 5. Call frame format

음성프레임 형태를, call 프레임은 G.764 프로토콜의 신호프레임의 기본 형식을 크게 바꾸지 않고 수정하여 사용하였으며 각 프레임의 형태가 그림 4, 5에 나타나 있다. 음성 패킷의 경우에는 원래의 프로토콜과 비슷하며 음성 샘플들은 128bytes이고 call control 패킷의 경우에는 신호패킷형태에서 패킷의 맨끝에 call control type 필드로 1byte를 추가하였다. Call control type의 종류에는 call request, call accept, call connect 그리고 call release가 있다. Call refresh 패킷을 특별히 사용하지 않고 sequence number field가 0일 때마다 전송하게끔 하였다.

4) 상위계층

음성들을 샘플링하고 일정크기의 샘플들에 대해 복음구간 검출을 행하여 그에 해당하는 정보를 패킷계층으로 보내고 반대로 수신된 음성 샘플들을 재생한다. 또한 패킷계층에서 각 state에 따라 그에 해당하는 tone이 이 계층에 제공된다. 전화기의 state를 검출해서 패킷계층에 전달한다.

5) 음성재생 알고리즘

그림 6에 음성재생 알고리즘의 state diagram을 나타 내었다. 본 논문에서 사용한 음성재생 알고리즘은 sequence number field만을 사용한다. Call이 이루어지고 난 후 수신단은 IDLE state가 되며 수신단의 상태변수 RSEQ(그림 2의 'sequence number' field)는 0이 된다. RSEQ는 수신단의 순서 상태 변수로써 도착할 패킷의 sequence number가 RSEQ와 같아야 하는데 그렇지 않은 경우 loss가 일어나게 된다. 즉 패킷의 loss를 판단하는데 사용된다. 수신단에서 처음으로 패킷이 도착하면 TALKSPURT상태가 되며 규정된 크기의 지연 후에 재생된다. 다음패킷 재생시 도착한 패킷의 sequence number field와 상태 변수인 RSEQ를 비교한다. 같으면 1을 증가시키고 재생을 시작한다. RESQ가 작으면 LOSS상태로 들어간다. LOSS 상태에서 패킷이 도착하면 패킷이 손실된 것이므로 다시 TALKSPURT state로 가서 buffer의 음성 샘플들을 재생한다. 그렇지 않고 buffer가 비어 있으면 송신단을 묵음구간으로 보고 SILENCE state로 전이한다. 그 후 패킷이 도착하여 buffer가 차게 되면 다시 TALKSPURT state로 가서 음성을 재생한다. 이 때에도 위와 같이 RSEQ와 도착한 패킷의 sequence number를 비교하여 재생한다.

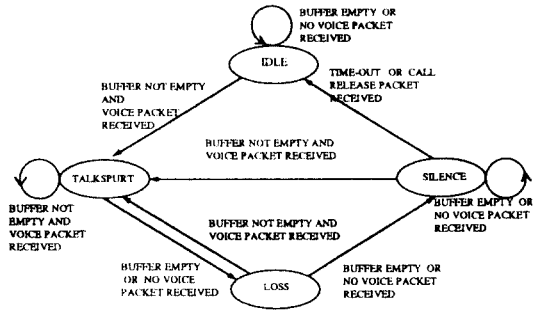
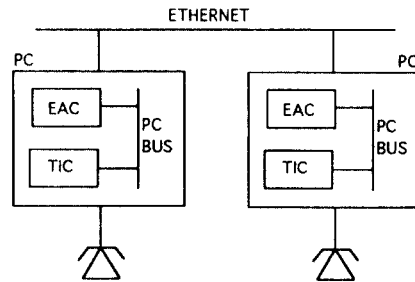


그림 6. 음성재생 알고리즘의 상태도
Fig. 6. State diagram of voice reconstruction algorithm

3. 하드웨어의 구조

PC 내에 ethernet adaptor와 전화기 인터페이스 카드를 통해 음성통신을 한 경우에 대해서 설명하겠다.

PC-to-PC 통신에 대한 시스템블록 다이어그램은 그림 9와 같다.



EAC : Ethernet Adaptor Card TIC : Telephone Interface Card

그림 7. 시스템 블록 다이어그램
Fig. 7. System block diagram

여기서 ethernet adaptor는 기존의 것을 사용하였고 전화기 인터페이스 카드는 제작하였다. 전화기 인터페이스 카드는 SLIC chip을 이용한 전화기와 인터페이스에 관련된 부분과 음성의 A/D, D/A변화에 관련된 부분과 PC와의 인터페이스 및 제어에 관련된 부분으로 나누어지며 그림 8에 블록다이어그램이 나타나 있다.

다음에 이들의 구체적인 구조와 기능에 대하여 살펴해보겠다.

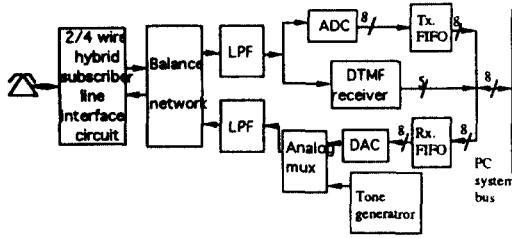


그림 8. 전화기 인터페이스 카드의 블록 다이어그램
Fig. 8. Block diagram of telephone interface card

3.1 Telephone interface part

전화기를 시스템에 연결하려면 전화기에 power를 공급해야 하고 2W/4W hybrid기능을 해야 하는 등 여러가지 문제점이 많다. 이 부분은 이러한 전화기 인터페이스에 관계된 제반기능을 담당한다.

1) SLIC(Subscriber Line Interface Circuit)

전화기 인터페이스시 주요 구성요소는 SLIC chip이다. SLIC chip으로는 HC5504를 사용하였으며 주요 기능은 다음과 같다.

가) Power feeding

전화기에 dc power를 공급해 준다. 이 경우에 48V의 DC 전원이 필요한 데 이는 PC에서 제공되는 +12V로 current PWM controller chip UC3845와 transformer를 사용하여 -48V를 만들어 사용하였다.

나) Hook status detection

Loop impedance(200-1200ohm)의 변화를 통한 loop current를 감지하여 on-hook과 off-hook의 상태를 감지하며 이를 CPU에서 읽어들인다.

다) Ring injection

전화기가 on-hook 상태에서 CPU에 의해 ring command를 주면 SLIC chip내의 ring relay driver에 의해 ring generator로부터 들어오는 ring신호를 전화기에 공급하여 bell이 울리게 해준다. 전화기의 bell은 75Vrms, 20 Hz의 sinewave를 20IPM(1초 접속, 2초 단절)[18]에 의해 울린다. Ring generator는 transformer를 사용하였다.

라) 2W/4W hybrid

상용의 전화기는 ring, tip의 2선에 송, 수신 신호가 섞여서 흐른다. 교환기와 연결시에는 교환기가 hybrid기능을 제공해서 상호간에 echo가 안 생기게 하나 본 시스템에서는 그러한 기능이 없으므로 직접 제공해주어서 상호간에 echo로 인한 통화 불능을 해결해야만 하는데 SLIC chip에 이러한 2선과 4선과의 변환을 담당하는 hybrid기능이 제공된다.

2) Balance network

SLIC chip의 hybrid회로의 부분과 연결되어 2W/4W변환기능을 한다. SLIC chip이 2W/4W로 변환해주는 데 balance network에서 부하(전화기) 임피던스와의 정합을 통해 송신 신호 전류에 섞여 흘러 나오는 수신 신호 전류를 상쇄시켜서 echo가 생기지 않게 해준다.

3.2 Speech processing part

음성신호의 A/D, D/A에 관계된 모든 처리를 담당한다. 또한 DTMF 전화기로부터 dialing을 감지해서 이를 CPU에 알려주는 역할을 하며 FIFO를 이용하여 송·수신할 일정량의 샘플들을 저장한다.

1) Filter

Op-amp를 이용한 Sallen and key low pass filter를 사용하였으며 대역폭은 3.4kHz로 하였다.

2)ADC

음성신호의 디지털 변환은 A/D Converter를 사용하였다. 샘플당 bit수는 8bit이고 샘플링속도는 8kHz이며 ADC chip으로는 ADC0804를 사용하였다.

3)DAC

8bit의 음성 디지털 신호를 음성 아날로그 신호로 바꾸어준다. 또한 전화기와 관련된 미리 저장된 데이터를 이용해서 tone을 DAC를 통하여 제공하게 하였다.

4)Dialing detector

DTMF 전화기의 keypad를 누르면 특정 주파수가 조합된 출력이 나온다. 이러한 특정주파수의 조합을 감지해서 그에 해당하는 ethernet address로 바꿀 수가 있다. 이를 위해서 DTMF receiver용 IC 인 MT8870을 사용하였다. 이는 각 주파수의 조합을 감지해서 이에 해당하는 디지털 코드로 바꾸어준다.

5)FIFO

송신 FIFO의 경우 ADC로부터 샘플링된 데이터가 저장되며 128개의 음성샘플들이 모이면 인터럽트가 걸려 패킷 계층에서 패킷을 만들어 전송한다. 또한 수신FIFO의 경우 수신된 음성 패킷이 있을 경우 음성 샘플들을 저장하고 주기적인 시간마다 DAC로 재생된다.

6)Tone generator

전화기에 필요한 각종 tone을 제공한다. Dial tone, ringback tone, busy tone, congestion tone을 생성한다.

3.3 PC와의 interface 및 controller part

PC내의 확장슬롯을 통해 address decoder를 통하여 해당하는 입출력 포트를 통하여 원하는 데이터를 읽거나 쓴다. 또 DTMF receiver의 데이터를 읽기 위한 입력포트도 있다.

사용된 clock은 PC내의 확장슬롯에서 제공되는 14.31818MHz의 clock을 이용해서 ADC에 8kHz를 DTMF receiver에 3.5795MHz의 clock을 카운터로 분주해서 제공하였다.

4. 소프트웨어의 구조

필요한 소프트웨어 모듈로는 ethernet과의 access를 위한 ethernet MAC driver, 음성 및 call 패킷의 packetization, depacketization, 호 설정/해제를 위한 호 처리 루틴 그리고 전화기와의 인터페이스에 관련된 루틴 등이 필요하다.

음성 프로토콜 루틴은 음성과 call 패킷에 관련된 제반사항을 다루며 음성 패킷에 대해서는 에러복구를 하지 않고 call 패킷에 대해서는 에러복구를 한다.

다음 절에서는 소프트웨어의 구성을 구체적으로 알아보겠다.

4.1 Ethernet MAC driver

Ethernet상에 프레임을 보내기 위해 PC의 확장슬롯에 장착된 ethernet adaptor를 구동해야 한다. 이 driver는 크게 초기화 루틴, 프레임수신 루틴 그리고 프레임송신 루틴으로 구성되어 있다.

초기화루틴은 메모리 초기화, 82586의 초기화 그리고 자체진단으로 나누어진다. 메모리 초기화로는 ethernet 프레임을 송·수신하기 위한 공유메모리 영역의 초기화로서 receive buffer descriptor, transmit buffer des-

criptor의 설정으로 구성되어 있고 82586초기화는 LAN coprocessor에게 미리 정의된 address인 SCP주소를 통해 mail box기능을 하는 data structure(SCB)의 위치를 가르켜 주는 일을 한다. SCB가 지정되면 이를 통해 individual address 설정 등을 수행하여 82586 LAN coprocessor내에 있는 모든 register들을 설정한다. 모든 과정이 끝나면 command unit과 receive unit를 정상 상태의 동작을 수행하게끔 한다. 프레임수신 루틴은 프레임을 수신하였을 경우에 대한 데이터처리를 담당하며 프레임전송 루틴은 송신할 음성 패킷의 경우에 프레임을 만들어 전송하는 일을 수행한다.

4.2 패킷계층

패킷계층에서 핵심적인 기능을 담당하는 데 주요기능은 다음과 같다.

- 음성의 packetization 및 depacketization 기능
- Ringing과 각종 tone을 포함한 call의 시작과 종료 기능
- Call 패킷에 대한 error recovery

패킷 음성 프로토콜의 특징은 가입자의 reaction에 의존한다는 점에 있다. 패킷계층의 state는 이러한 가입자의 reaction에 따른 call의 유지에 대한 제반사항들로 구성된다. 주고 받는 call 프레임과 이에 따른 state transition diagram을 그림 9에 도시하였다.

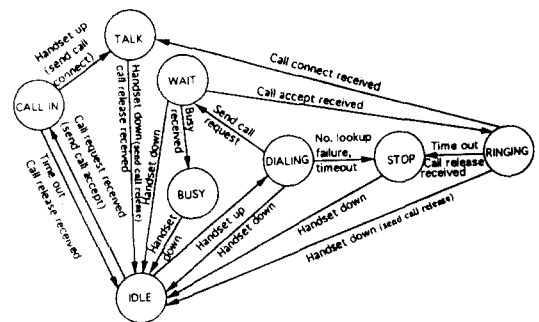


그림 9. 전화기 상태도
Fig. 9. Telephone state diagram

음성 통화시 정의한 state는 다음과 같다.

1) IDLE state

송·수신측 모두 수화기가 내려져서 call이 이루어

지지 않은 상태로 초기상태이다. 이 상태에서는 call request packet을 수신하거나 수화기를 들어 dialing을 할 경우 다른 state로 이동한다.

2) DIALING state

송신측에만 존재한다. IDLE state에서 전화기를 사용하고자 수화기를 들면 전화기 인터페이스 회로에서 hook상태가 off임을 감지하여 packet layer로 전달되어 DIALING state로 오게 되며 이때부터 전화번호가 입력된다. 전화번호가 입력될 때마다 DTMF receiver chip (MT8870)으로부터 어느 button이 눌렸는지를 알 수 있다. 전화번호의 입력이 다 끝나게 되면 전화번호 table에서 해당되는 곳의 ethernet address를 찾아 그 address로 call request 패킷을 보낸다.

3) WAIT state

송신측에만 존재한다. DIALING state에서 call request 패킷을 보내고 call의 accept를 기다린다. 만약 상대방이 통화중일 경우 busy 패킷을 송신하게 되는 데, 이 때 송신측은 busy 패킷을 수신하게 되며 busy state로 전환한다. 또한 call accept 패킷을 수신하면 송신측은 RINGING state로 전환하게 된다. 송신측이 수화기를 놓는 경우에는 IDLE state로 전환한다.

4) BUSY state

송신측이 call request 패킷을 보냈을 때 상대방이 통화중일 경우 busy 패킷을 수신하게 된다. 그러면 송신측은 WAIT state에서 BUSY state로 전환하고 전화기에 busy tone을 울린다. 이 상태에서 수화기를 놓으면 IDLE state로 가게 된다.

5) RINGING state

송신측에 존재한다. Call이 accept된 후에 수신측 전화기의 bell을 울려주는 상태로서 이때 송신측의 수화기에는 ring back tone이 울리게 된다. 이 상태에서 송신측이 수화기를 놓으면 call release 패킷을 수신측에 보내며 IDLE state로 간다. 또한 call connect 패킷을 수신하게 되면 TALK state로 가게되며 음성 패킷을 주고 받아 통화를 할 수 있는 상태가 된다.

6) CALL IN state

수신측에 존재하는 state로서 송신측이 보낸 call request 패킷을 수신하게 된 경우로서 전화기에 bell이

울리게 된다. 전화기를 들면 call connection 패킷을 보내 call의 성립을 알리고 통화를 할 수 있는 TALK state로 전환한다. 또 송신측으로부터 call release 패킷을 받거나 수신측이 수화기를 놓으면 IDLE state로 전환하며 일정한 시간동안 수화기를 들지 않으면 time-out되어 송신측에 call release 패킷을 보내고 IDLE state로 전환한다.

7) STOP state

송신측에 존재하는 state로서 송신측이 DIALING state에서 입력된 전화번호와 대응하는 것을 table에서 찾지 못했을 경우와 WAIT state와 RINGING state TALK state에서 clear request 패킷을 수신하였을 경우 이 state로 오게 되며 이 상태에서는 수화기가 내려지기를 기다린다.

8) TALK state

송·수신측에 존재하며 수신측에서 call connect 패킷을 보내고 또 송신측에서 이 패킷을 받게 되면 양쪽 다 TALK state로 가게 되며 실제적인 음성 패킷을 주고 받는 상태가 된다. 음성 패킷에는 sequence number field가 있어 음성 재생시 이를 참조한다. Buffer overflow가 나는 경우에는 이전의 샘플들을 버린다. 이 상태에서는 어느 한쪽이 수화기를 놓으면 call release 패킷을 보내 call의 종료를 알리며 또한 이를 받은 측은 이를 수락하여 IDLE state로 가게 된다.

지금까지 패킷 음성 프로토콜의 packet layer의 송·수신측의 state와 그 전환에 대하여 알아보았다. 이러한 state들은 call의 성립과 종료에 관련된 아주 기본적인 것들인데 이외에 가입자에게 향상된 서비스를 제공하고 확실한 call의 유지를 위해 보조적인 state를 둘 수도 있다. 예를 들면 다양한 음성 코더와 코딩비율, 패킷의 크기 등을 서비스하기 위해 WAIT state대신 NEGOTIATION state를 두어 이러한 파라미터들에 대한 협상을 할 수도 있고 TALK state에서 어느 한쪽에 문제가 발생해서 음성 패킷을 받을 수 없을 때 HOLD state라는 보조 state로 전환하여 이의 해결을 기다릴 수도 있다.

원하는 패킷들의 손실로 인해 발생하는 dead lock 현상을 방지하기 위하여 timer를 동작시켜 timer가 끝날 때까지 원하는 패킷이 오지 않으면 time-out이 되어 WAIT state와 RINGING state 그리고 TALK state에서는 STOP state로, CALL IN state에서는 IDLE state로 전환한다.

사용되는 call 패킷의 종류를 표시하면 다음과 같고 call을 setup/release하는 mechanism은 그림 10에 나타내었다.

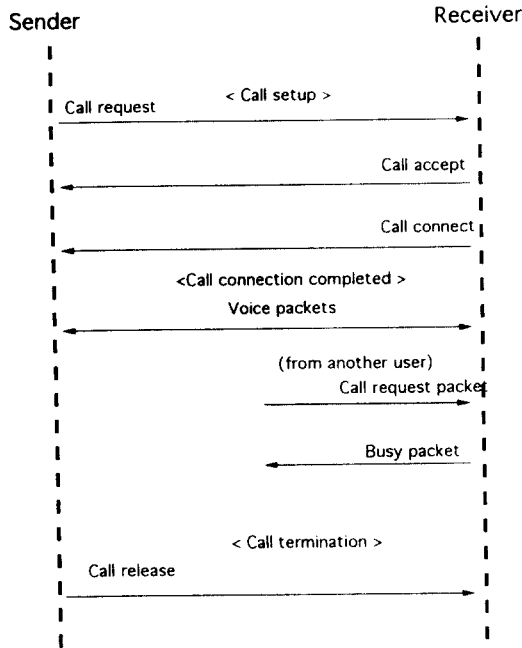


그림 10. 호 설정/해제 과정
Fig. 10. Call setup/release mechanism

- 1) Call request packet - 송신측이 상대방에게 call의 성립을 요구할 때 사용한다. 자신의 address와 상대방의 address가 frame에 포함되고 call type field를 call request로 set하고 전송한다.
- 2) Call accept packet - 수신측이 call request packet를 받았을 때 통화중이 아닐 경우 call accept packet을 송신측에 보낸다.
- 3) Busy packet - 수신측이 call이 성립된 상황에서 다른 가입자로부터 call request packet을 받았을 때 통화중이어서 call이 성립될 수 없을 경우 이 packet을 보내게 된다.
- 4) Call connect packet - 수신측의 전화기에 bell이 울리고 있는 상태에서 수화기를 들어 off-hook의 상태로 들어 갔을 때 call connect packet을 송신측에 보낸다.
- 5) Call release packet - 개설된 call의 취소를 요구할

때 사용한다. CALL IN state, TALK state, STOP state의 경우 call의 취소시 call release packet을 보내야 한다.

4.3 PAD(packet assembly and deassembly)

음성이나 데이터를 패킷교환망에서 서비스하기 위해서는 이들을 패킷화해야 하며 수신단에서는 이를 다시 풀어서 원래의 정보를 복원해야 한다. PAD는 이와같은 기능을 위해 존재한다.

음성은 일단 talk state로 들어오면 음성의 코딩비율에 따라 규칙적으로 만들어진다. 음성 PAD는 만들어지는 음성 데이터가 한 패킷의 크기가 되면 이부분이 음성 구간인지 묵음구간인지를 판단해서 음성구간이면 이를 패킷화하며 상대방으로 부터 전달되어 온 패킷에 대해서는 음성 재생 알고리즘에 따라 재생한다. 다음에 음성 PAD의 기능을 송신단과 수신단으로 나누어 설명하겠다.

1) 송신단의 기능

음성은 125 μ s마다 샘플링되어 128번의 샘플링후에 계산된 에너지로부터 묵음구간인지 아닌지를 판별하고 음성구간으로 판명되던지 SSEQ가 0이던지 하면 음성 패킷을 만들어 전송한다. 여기서 SSEQ는 송신단의 순서 상태 변수로서 한 패킷 전송시 패킷헤더의 SEQ field에 복사된 후 하나씩 증가한다. 상대방으로부터 맨처음으로 음성패킷을 수신하기 전까지는 묵음구간 검출을 행하지 않고 생성된 패킷마다 전송하며 첫 패킷을 수신하게 되면 그 후부터는 묵음구간 검출을 행하여 전송한다.

2) 수신단의 기능

수신된 음성프레임은 buffering후에 재생되는 데 음성구간의 첫 패킷에 대해서만 지연을 시키고 그 이후의 패킷은 코딩비율에 맞춰 재생을 한다. Buffer가 비어 있게 되면 묵음구간으로 판명되어 패킷 길이 만큼 동안 정지한 다음에 buffer를 살펴 재생할 음성 패킷이 도착하였는지를 점검한다.

4.4 묵음구간 검출 알고리즘

본 논문에서 구현된 알고리즘의 순서도는 그림 11에서 보여진다. 음성의 시작점과 끝점을 정확하게 찾는 문제는 그리 쉬운 일은 아니다. 또한 끝점 검출의 알고리즘이 정확해지고 복잡해 짐에 따라 전체 시스템에 주는 영향이 커지게 되어 delay와 같은 문제가 심각해질

수도 있으므로 정확도와 time delay 사이에는 적절한 타협이 필요하다. 본 패킷음성 프로토콜에서는 생성된 음성 패킷은 곧 바로 전송된다. 그리하여 패킷이 생성된 경우 곧바로 판단을 해서 전송 또는 버린다. 사용된 끝점 검출 알고리즘은 시스템 전체에 지나친 부담을 주지 않게 하기 위하여, 에너지만을 사용하여 음성과 묵음을 구별한 후에 인식할 수 없는 묵음구간의 연결을 위해 hangover time을 사용하는 비교적 간단한 알고리즘을 사용하였다. 음성 샘플들은 패킷단위로 전송되므로 에너지를 구하는 짧은 구간은 한 패킷 내의 음성 샘플들의 갯수가 128인 경우 즉 16ms가 된다. 이 구간은 음성의 특성이 급격히 변하지 않는 짧은 시간이라고 가정하며 한 패킷의 에너지는 다음과 같이 계산된다.

$$E(n) = \sum_{i=1}^{127} |s(i)|$$

여기서 $s(i)$ 는 i 번째 샘플을 나타낸다.

각 샘플의 제곱의 합이 아니라 절대값의 합으로 정의한 이유는 곱셈을 피하여 계산량을 줄이고, 음성 신호의 진폭 차이에 의한 시간축상의 변화를 감쇄시키기 위한 것이다. 에너지를 사용하여 음성과 묵음을 구별하기 위해서 에너지 경계치가 필요하다. 이 경계치는 미리 정해질 수도 있고 입력 신호의 처음 얼마 동안의 구간을 묵음구간이라 가정하여 계산되어 질 수도 있는데, 여기서는 편의상 미리 값이 주어진다. Hangover time은 160ms(10개의 voice packet)로 정했다.

5. 프로토콜 검증 및 트래픽 수용량

끝으로 프로토콜의 정확성 검증 및 LAN의 수용능력 등에 관하여 검토해 보기로 한다. 본 연구에서 설계한 프로토콜은 그림 9에 표시되어 있다. 일단 설계된 프로토콜은 정확성을 검증하여야 하는데, 검증방법에는 프로토콜이 페트리네트로 표시된 경우에는 Deductive Inference 방법으로, 본 연구에서와 같이 유한 자동 기계(Finite State Machine)으로 표시된 경우에는 Reachability Analysis 방법 등으로 분석이 가능하다[20]-[21]. 그러나 본 연구에서는 구현이 주 목적이고 state 수가 적은 점을 고려하여, 직접 음성통화를 테스트하여 봄으로써, 입력 사건에 따라 각 state에 잘 도달하고 설계된 프로토콜에 오류가 없음을 확인하였다.

Ethernet에서 지원할 수 있는 음성 트래픽의 양에 관해서는 [22]-[23]의 참고문헌에서 연구된 것을

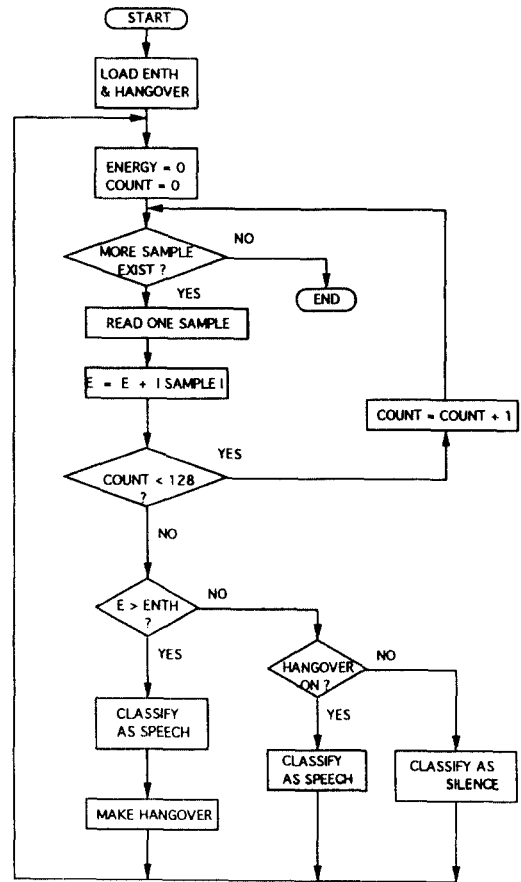


그림 11. 묵음구간 검출 알고리즘의 순서도

Fig. 11. Flow chart of silence detection algorithm

기초로 검토해 보기로 한다. 문헌 [22]에서는 데이터가 없을 때, 5.75ms 길이의 음성 packet에 대한 CSMA/CD 지연시간을 시뮬레이션을 통하여 연구한 결과, 약 60개의 대화(또는 120개의 음성 트래픽)를 분기점으로 하여 트래픽이 증가하면 지연시간이 급격히 상승함을 보여주고 있다. 또한 [23]에서는 overhead(6 bytes로 가정)를 제외한 음성의 패킷 크기가 96 bytes인 경우 throughput이 약 60% (즉 6Mbps)를 넘으면 packet loss가 1% 이상 급격히 증가함을 보여주고 있다. 따라서 음성이 6Kbps이며 talkspurt가 전체 음성의 약 반이라고 가정하고, 또한 G.764에서는 overhead가 그림 2에서 보듯이 9 bytes이고 이 외에도 Ethernet frame 구성시 overhead가 그림 4에서 보듯이 19 bytes 임을 고려하면, 음성 트래픽수는 대략 $(6000/64) * 2 * [(96+6)/(96$

+9+19)]=142과 같이 산정된다. 이 두가지 계산 결과로부터 10Mbps Ethernet에서 음성 전송능력은 대략 100여개 정도로 추산할 수 있다.

V. 실험결과

음성 패킷의 end-to-end delay는 2음성 패킷의 digitization에서 재생되기까지의 시간으로 정의되는데, 여기서는 T_e 라고 하겠다. 음성 frame이 수신단에 도착하여 queue에 대기하는 시간 $T_{QUEUING}$ 이 0이라고 가정하고 채널상의 트래픽이 매우 적은 경우에 time들을 측정해 보았다. 한 패킷당 음성 샘플들의 크기는 64, 128, 256byte의 경우에 대하여 측정하였으며 그 결과를 표 1에 나타내었다. $T_{COLLECT}$ 는 sample collection time이고, $T_{RECONSTRUCT}$ 는 수신단에서 음성 패킷의 재생시간으로 이 둘은 서로 같은 값이 된다. $T_{PROC-TX}$ 및 $T_{PROC-RX}$ 는 각각 송·수신단에서 processing time으로 측정된 값이며, $T_{TRANSIT}$ 는 송·수신단 사이의 전송시간이다.

표1. end-to-end delay의 측정

Samples in a packet	64	128	256
Delay times			
$T_{COLLECT}$	8ms	16ms	32ms
$T_{PROC-TX} + T_{PROC-RX}$	80.625 μ s	196.625 μ s	308.75 μ s
$T_{TRANSIT}$	436.875 μ s	490.625 μ s	577.5 μ s
$T_{RECONSTRUCT}$	8ms	16ms	32ms
T_e	16.5175ms	32.68725ms	64.88625ms

측정과정을 살펴보면 다음과 같다. 먼저, end-to-end delay 측정시 $T_{QUEUING}$ 을 0이라 가정하였고, $T_{COLLECT}$ 와 $T_{RECONSTRUCT}$ 는 미리 결정되므로 그 나머지 시간을 측정하면 된다. 송신단에서 패킷을 전송한 후 수신단에 도착하고 다시 수신단에서 송신단으로 패킷을 보내어 송신단에 도착하는 시간을 8kHz의 clock으로 측정하여 2로 나누었다. 다음으로 processing time의 측정방법으로 위의 측정과정에서 processing time에 해당하는 과정을 수행하지 않고서 측정된 결과를 얻어 end-to-end delay에서 빼 값을 사용하였다. 위의 실험결과는 network의 traffic이 거의 없을 경우를 택하여 측정을 한 결과이다. 그러나 network의 traffic이 많아지게 되면 전송시의 delay time이 매우 커지게 된다.

음성의 품질은 다소 잡음이 섞여있는 상태로 통화

품질이 일반전화보다 다소 떨어지는 상태였으나, 상대방의 음성을 알아듣는 데는 아무 지장이 없었다.

VI. 결론 및 차후 연구 방향

지금까지의 통신망은 음성전송을 위한 공중전화망, 데이터전송을 위한 패킷 교환망 등과 같이 각기 독특한 서비스에 적합하도록 발전되어 왔다. 그러나 컴퓨터와 통신기술이 급속도로 발전하여 통신설비의 가격이 낮아지고 데이터 전송량이 상대적으로 중요해지고 서비스의 집적을 위한 전송용량의 증가로 점차 여러통신망이 통합되어 가고 있는 상황이다. 이러한 추세에 맞춰 본 연구에서는 ethernet에서 음성을 패킷단위로 전송하여 기존의 공중전화망처럼 음성통신을 시도하였다.

이를 위하여 G.764 패킷 음성 프로토콜을 기본으로 하고 이를 수정하여 ethernet에 적합한 패킷 음성 프로토콜을 구현하고 이에 필요한 소프트웨어 및 하드웨어를 제작하였다.

하드웨어는 상용의 ethernet card를 사용하고 음성 타이밍은 상용의 2선식 전화기를 사용하였으며 전화기와 인터페이스 및 음성 처리 기능을 담당하는 전화기 인터페이스 카드를 제작하여 사용하였다. 소프트웨어는 ethernet device driver 부터, 전화기 인터페이스 카드용 driver 부터 그리고 패킷 음성 프로토콜 부터들로 구성되어 있다.

패킷 네트워크에 음성을 전송한다는 것은 음성은 다양한 서비스 중 가장 기본이 되는 서비스로 다른 여러 서비스들과 함께 사용될 수 있다는 점에서 의미가 크다. 우선 음성 서비스에 대한 응용으로 다자간 음성 통화를 생각해 볼 수가 있었고 또한 오디오와 비디오 등의 트래픽도 동시에 서비스하는 데 기본 연구로서 중요하다. 또 국내 및 외부 전화와도 통신이 가능하도록 PABX gateway의 개발도 필요하고 ISDN의 사용이 중대될 경우를 고려하여 ISDN전화기와와의 접속도 필요하다.

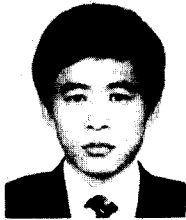
본 논문은 과거 저점단 요소 기술과제 제정 NN09010으로 부터 (부분적으로) 지원 받았음.

참 고 문 헌

1. J. Zdepski, K. Joseph, D. Raychaudhuri and D. G. Daut, "Prioritized Packet Transport of VBR CCITT H.261 Format Compressed Video on a CSMA/CD LAN", IEEE VISICOM, pp.20-25, 1990.
2. Steve Casner, "Data packet header formats for real-time audio and video", Audio/Video Transport WG Meeting Report, San Diego, Mar. 1992.
3. Henning Schulzrinne, "A transmission for audio and video conferences and other multiparticipant real-time applications", Univ. of Massachusetts, July 1992.
4. T. A. Gonsalves, "Packet Voice communication on an Ethernet local computer network : an experimental study", Communications of the ACM, pp.178-185, 1983.
5. J. Van de Lagemaat, J. M. A. Daeman and I. G. Niemegeers, "Voice transmission in a priority CSMA/CD LAN : an efficient protocol using hybrid switching", ACM SIGCOMM symposium, pp.102-110, 1986.
6. Eluzar Friedman, Chaim Ziegler, "Real-time voice communications over a token-passing ring LAN", ACM SIGCOMM symposium, pp.52-57, 1986.
7. C. J. Adams and Stephen Ades, "Voice experiments in the UNIVERSE Project", Nov. 1984.
8. CBM Mishra and R. Madhusudhan, "Voice data integration on FDDI Network implementation and analysis", ICCS, pp.220-228, 1990.
9. 전 홍범, Packet voice/data terminal의 개발과 성능분석, KAIST 석사논문, 1987년 1월.
10. 남 세현, 토큰링 LAN에서의 음성/데이터 집적에 관한 연구, KAIST 석사논문, 1987년 1월.
11. E. Friedman and C. Ziegler, "Packet voice communication over PC-based Local Area Networks", IEEE JSAC., vol. 7, no. 2, Feb. 1989.
12. 이 상길, PC용 패킷음성 프로토콜의 구현, KAIST 석사논문, 1993년 2월.
13. Jayant and Christensen, "Effects of packet losses in waveform coded speech and improvements due to an odd-even sampleinterpolation procedure", IEEE Trans. on Communications, vol. COM-29, no. 2, Feb. 1981.
14. L. R. Rabiner and M. R. Samber, "An algorithm for determining the end points of isolated utterances", Bell System Technical Journal, vol. 54, no. 2, Feb. 1975.
15. CCITT draft Recommendation G.764-Voice packetization : Packetized voice protocol, July 1990.
16. 우 준석, 은 종관, "Bit-dropping에 의한 overload control 방식을 채용한 packet voice multiplexer의 성능분석에 관한 연구", 한국통신학회 논문지, vol. 8, no. 1, 1993년 1월, pp.110-122.
17. 전화회로도 보는 법, 그리는 법, 도서출판 세운, p.12.
18. P.T.Brady, "A technique for investigating on-off pattern in its conversations", "Bell System Tech. Journal, vol. 47, pp.73-91, Jan. 1968.
19. Tat Y. Choi, "Formal Techniques for the specification, verification and construction of communication protocols", IEEE Commun. Mag., vol. 23, no. 10, p.46-52, Oct. 1985.
20. 전 주경, 허 기택, 이 동호, "근거리 통신망과 ISDN의 연동을 위한 게이트웨이 모델링에 관한 연구", 한국통신학회논문지, 제17권, 제8호, 1992년 8월호, p.823-836.
21. 조 동호, "음성/데이터 집적 근거리 통신망에 관한 연구", 정보통신기술, 제3권, 제2호, 1989년 12월, p.53-65.
22. Jeffrey M. Musser and et al., "A local area network as a telephone local subscriber loop", IEEE J. Sel. Area Commun., vol. SAC-1, no. 6, Dec. 1983, p.1046-1053.
23. 신 병철, "멀티미디어 통신기술", 대한전자공학 회지, 제20권, 제2호, 1993년 2월, p.110-131.



李相吉(Sang Kil Lee) 정회원
1991년 2월 : 충남대학교 전자공학과 졸업
1993년 2월 : 한국과학기술원 전기 및 전자공학과 석사졸업
1993년 1월 ~ 현재 : 삼성정보통신(주) 연구원



金潤寬(Youn Kwan Kim) 정회원
1978년 : 서강대학교 전자공학과(학사)
1983년 : 한국과학기술원 전기및 전자공학과(석사)
1989년 : Northwestern 대학(박사)
1978년 ~ 1987년 : 삼성전기(주) 연구원

1987년 ~ 현재 : 삼성정보통신(주) 책임연구원



申炳喆(Byung Cheol Shin) 정회원
1952년 11월 2일생
1975년 2월 : 서울대학교 전기공학과 졸업
1977년 2월 : 한국과학기술원 전기 및 전자공학과 석사졸업
1984년 8월 : 한국과학기술원 전기 및 전자공학과 박사졸업

1977년 2월 ~ 1980년 2월 : 한국전자통신연구소 연구원
1987년 3월 ~ 1988년 2월 : SRI International에서 Post Doc. 인수

1984년 9월 ~ 현재 : 한국과학기술원 전기및 전자공학과 부교수

※주관심분야 : Multimedia 통신, High Speed Network, Wireless Communication, ATM Network 분야