VoIP Howto

Roberto Arcomano

berto (at) fatamorgana.com

도운이

조지완

ai (at) kldp.org

v1.4, 10 December 2001

Voice Over IP 는 거의 돈을 내지 않고서도 인터넷을 통해 전화를 할 수 있도록 해주는 새로운 통신수단이다. 이 하우투 문서에서는 어떻게 이것이 가능한지, 어떤 시스템이 사용되는지, 어떤 것이 표준인지를 다룬다. 웹사이트 http://www.fatamorgana.com/bertolinux 에서 이 문서의 최신 버전을 찾아볼 수 있다.

고친 과정

고침 0.12002-02-20고친이 조지완

최초 번역, 원문 V1.4, 2001-12-10 에 대해

1. 들어가며

1.1. 개요

이 문서는 VoIP 시스템에 대해 설명한다. 저렴한 비용으로 인터넷이 보급되고, 음성 압축 전용 프로세서가 통합되는 최근의 추세는 일반적인 사용자들의 필요를 변화시켜 VoIP 표준이 확산시키고 있다. 이 하우투 문서는 VoIP 아키텍처의 기본 개념에 대한 설명을 할 것이다.

제안이나 비평은 나의 이메일 (mailto:berto (at) fatamorgana.com) 로 해주기 바란다.

1.2. 저작권

Copyright (C) 2000,2001 Roberto Arcomano. This document is free; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version. This document is distributed in the hope that it will be useful, but

WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You can get a copy of the GNU GPL here (http://www.gnu.org/copyleft/gpl.html)

1.3. 번역

이 문서 번역은 자유로우며(free) 아래의 사항만 지키면 된다 :

1. 이미 해당하는 언어권의 LDP 에 번역본이 있지 않은지 살펴보라.
2. 소개, 저작권, 번역, 크레딧을 포함한 모든 'Introduction' 부분을 유지하라.

당신의 번역에 감사드린다!

1.4. 크레딧

하드웨어 장비와 실험 기회를 제공한 Fatamorgana Computers (http://www.fatamorgana.com) 에 감사드린다.

나의 문서를 빠르게 알릴 수 있도록 도와준 Linux Documentation Project (http://www.linuxdoc.org) 에 감사드린다.

도움을 준 David Price (mailto:dprice (at) intercorp.com.au) 에게 감사드린다.

2. 배경

2.1. 과거

20~30 년 전에는 인터넷이 존재하지 않았다. 쌍방향의 통신방식은 PSTN [1] 회선에 비용을 지불하는 전화에 의한 방법 밖에 없었다.

데이터 교환은 (장거리 전화일 경우) 높은 비용이 필요했고, 쌍방향의 화상통신은 아무도 생각할 수 없었다. (알다시피 오직 TV 만이 존재했는데, 이는 쌍방향이 아니다.)

2.2. 어제

몇 해 지나지도 않은 일 중에서 몇가지 재미있는 것들이 있는데: 일반 대중에게 PC 가 보급되고, 휴대전화와 같은 새로운 통신기술이 개발되었으며, 마침내 커다란 망인 인터넷을 형성하게 된 것이다. 사람들은 e-mail, chat 등의 새로운 서비스를 통해 통신하기 시작하였고, 회사들은 고객들에게 "클릭" 한 번으로 구매할 수 있는 웹을 제공하게 되었다.

2.3. 오늘

오늘날 우리는 통신계의 대변혁을 느낄 수 있다. 누구나 PC 와 인터넷을 사용하여 일자리를 구하고, 서로간 통신을 하며 여가시간을 보내거나 자료 (이미지나 음악, 문서 등과 같은) 를 교환하고, 때때로 MS Netmeeting이나 Internet Phone 과 같은 어플리케이션을 사용하여 대화하고 있다. 특히 통신계의 미래가 될 것이며 실시간 음성 통신 (vocal communication) 을 가능하게 할 수단이 무엇인가에 대해 공통된 의견이 확산되고 있는데, 그것이 바로 VoIP 이다.

2.4. 미래

우리는 미래가 모습이 될 지를 알 수는 없지만, 많은 컴퓨터와 고속 인터넷이 어디에나 마련되어 있는 환경에서 사람들은 실시간으로 대화 (음성, 화상 모두의 방법으로) 하게 될 것으로 상상할 수는 있다. 우리는 어떤 수단을 통해 이를 실현할 것인지 알기만 하면 된다 : UMTS [2] , (화상에까지 확장된) VoIP, 이외에 또 다른 것이 있는가? 어쨌든 최근 몇 년간 인터넷은 매우 빠른 속도로 성장해 왔고 그것은 (적어도 국제간의 수단으로서는) 자유로우며 (free) 차세대의 틀림없는 통신매체가 될 것임을 우리는 인식하고 있다.

3. 개관

3.1. VoIP 란 무엇인가?

VoIP 는 ’V’oice ’o’ver ’I’nternet ’P’rotocol 을 의미한다. 단어에서 나타난 바와 같이, VoIP 는 인터넷을 통해 IP 패킷에 (주로 사람의) 음성을 담아 보내려고 한다. VoIP 는 이러한 목적을 달성하기 위해 가속장치를 사용할 수 있으며 PC 환경도 사용할 수 있다.

3.2. 어떻게 동작하는가?

신호를 멀리 떨어진 목적지에 보내는 것은 디지털의 방법으로 가능하다는 것은 오래 전에 알려져 있는데 : 신호를 보내기 전에는 ADC (analog to digital converter) 를 통해 디지털화 시키고, 그것을 전송한 다음, 수신 측에서는 DAC (digital to analog converter) 를 사용하여 다시 아날로그 신호로 바꾸면 된다.

VoIP 는 음성을 디지털화 시켜서 데이터 패킷에 담아 그것을 전송하고, 수신측에서는 다시 음성으로 재변환시키는 방법으로 동작한다.

디지털 형태는 제어하기가 쉽다 : 압축할 수도 있고 경로를 지정할 수도 있으며 더 좋은 새로운 형태로 변형시킬 수 있는 등 장점이 많다. 또한 디지털 신호는 아날로그 신호보다 잡음에 잘 견딘다는 것을 알고 있다. (GSM [3] 과 TACS [4] 를 비교해보라.)

TCP/IP 망은 헤더 (통신을 제어하기 위한 부분) 와 데이터 전송을 위한 페이로드로 이루어진 IP 패킷으로 구성된다. VoIP 는 IP 패킷을 이용하여 망을 건너가 목적지에 도착한다.

```
음성 (송신지) - - ADC - - - - 인터넷 - - - DAC - - 음성 (목적지)
```

3.3. PSTN 에 대해 VoIP 가 가지는 장점은 무엇인가?

당신이 공중전화 회선을 사용하는 경우 망 사업자에게 통상 시간당의 사용요금을 지불해야 한다 : 따라서 더 많이 전화를 사용할 수록 더 많은 요금을 내야한다. 게다가 당신은 한 번에 한 사람 밖에 통화할 수 없다.

반면 VoIP 를 사용한다면, 요금에 무관하게 당신이 원하는 한 모든 사람들과 언제나 이야기할 수 있으며 (상대방도 같은 시간에 인터넷에 접속해 있어야만 한다.) 게다가 그들 모두와 동시에 대화할 수도 있다.

그럼에도 불구하고 당신이 이러한 사실을 고려하지 않는다 하더라도, 당신은 사람들과 대화하면서 동시에 이미지나 도표, 동화상 등의 데이터를 교환할 수도 있다는 것이다.

3.4. 그런데 왜 사람들은 아직도 VoIP 를 쓰지 않나?

불행하게도 VoIP 아키텍처와 인터넷의 통합에는 약간의 문제가 있다. 쉽게 상상할 수 있듯이 음성 데이터의 통신은 실시간 스트림이어야만 한다. (당신은 이야기하지 않은 채 몇 초를 기다려 상대방의 응답을 듣는 경우도 있다.) 이는 인터넷의 균질하지 않은 아키텍처와 충돌하게 되는데, 인터넷을 통해 패킷이 전달되기 위해서는 20~30 개의, 혹은 그보다 많은 수의 라우터 (패킷의 경로를 결정해주는 장치) 를 거쳐야 하므로 매우 높은 round trip time (RTT) [5] 을 갖게 된다. 따라서 제대로 동작하기 위해서는 이러한 상황을 수정해 줄 필요가 있다.

다음 절에서는 이 중대한 문제를 해결하는 방법에 대해 이해하게 될 것이다. 일반적으로 VoIP 어플리케이션이 필요로 하는 대역폭을 인터넷에서 보장한다는 것은 매우 어려운 일임을 알게된다.

4. VoIP 에 대한 기술정보

이제 우리는 VoIP 를 이해하는데 필요한 기술적인 내용을 다루게 된다.

4.1. VoIP 연결에 대한 개요

VoIP 통신을 위해 필요한 것은 :

1. 우선 아날로그 음성을 디지털 신호 (bits 형태) 로 바꾸기 위해 ADC 가 필요하다.
2. bits 형태의 정보는 전송에 유리한 형태로 압축되어야 한다 : 나중에 살펴보겠지만 이를 처리하는 프로토콜은 매우 많다.
3. real-time protocol (통상 IP 에 얹은 UDP 위에 얹은 RTP 를 의미한다.) 를 통해 음성 패킷을 데이터 패킷에 담는다.
4. 사용자를 호출하기 위해 신호 프로토콜이 필요하다 : ITU-T H323 가 있다.
5. RX (수신측) 에서는 패킷을 해체하고 데이터를 추출한 다음, 그것을 아날로그 음성 신호로 변형한 다음 사운드카드 (또는 전화) 로 넘겨준다.
6. 이 모든 동작은 실시간으로 이루어져야만 하는데, 그렇지 않다면 상대방의 목소리가 응답하는 데 너무 오랜 시간을 기다려야만 할 것이다. (4.6절 을 참조)

```
                              기본구조

음성 )) ADC -    압축 알고리듬    -    TCP/IP 에 RTP 를 삽입  -------
                                                          ---->    |
                                                          <----    |
음성 (( DAC - 압축해제 알고리듬 -  TCP/IP 에서 RTP 를 추출 -------
```

4.2. 아날로그 신호를 디지털 신호로 변환하기

이 과정은 하드웨어를 통해 이루어지며, 통상 ADC 기능을 통합한 카드가 사용된다.

요즘의 모든 사운드카드는 16 bit, 22050 Hz 의 처리능력이 있으므로 (나이키스트 정리에 의해 샘플링에는 44100 Hz 가 필요하다.) 단위시간당 처리량은 2 bytes * 44100 (초당 샘플 수) = 88200 Bytes/s 가 되며, 스테레오 스트림에 대해서는 176.4kBytes/s 의 결과를 얻게 된다.

VoIP 에서는 22 kHz 대역폭은 필요로 하지도 않는데 (16 bit 역시 필요하지 않다) : 다음에서 그러한 코딩 방식을 보게 될 것이다.

4.3. 압축 알고리듬

이제 우리가 가진 것은 빠른 전송을 가능하게 하는 표준형태로 변형해야 할 디지털 데이터이다.

PCM, Pulse Code Modulation, Standard ITU-T G.711

- 음성 대역폭은 4 kHz 이므로 샘플링 대역폭은 8 kHz 이어야 한다. (나이키스트 정리)
- 각각의 샘플 데이터는 8 bit 로 표시한다. (256 단계의 값을 가진다.)
- 시간당 처리량은 8000 Hz * 8 bit = 64 kbit/s 으로서 통상의 디지털 전화선 정도가 된다.

- 실제 적용되고 있는 mu-law (북미방식) 와 a-law (유럽방식) 에서는 아날로그 신호를 로그스케일로 코딩할 때 8 bits 이 아닌 12 bits, 13 bits 를 사용하는 것으로 변형되었다. (Standard ITU-T G.711 참조)

ADPCM, Adaptive differential PCM, Standard ITU-T G.726

- 이 방식은 이전 음성 패킷과 실제 음성 패킷의 차이점만을 32 kbps 로 변환한다. (Standard ITU-T G.726 참조)

  LD-CELP, Standard ITU-T G.728
  CS-ACELP, Standard ITU-T G.729 and G.729a
  MP-MLQ, Standard ITU-T G.723.1, 6.3kbps, Truespeech
  ACELP, Standard ITU-T G.723.1, 5.3kbps, Truespeech
  LPC-10, 2.5 kbps 까지 가능!!

- 이 최신의 프로토콜들은 원 신호 코딩에 매우 낮은 최소 대역을 보장해주기 때문에 매우 중요하다. 또한 G.723.1 코덱은 매우 높은 MOS (Mean Opinion Score, 음성의 충실도를 평가하기 위해 사용됨) [6] 를 가지게 된다. 하지만 이러한 프로토콜이 요구하는 프로세스 연산량은 매우 높아서 26 MIPS 정도에까지 이른다!

4.4. RTP 실시간 전송 프로토콜

이제 우리가 가지고 있는 것은 TCP/IP 스택에 인캡슐레이션 시킬 원형 데이터이다.

```
 VoIP 데이터 패킷
       RTP
       UDP
       IP
(하부) I,II 레이어
```

VoIP 데이터 패킷은 RTP (Real-Time Transport Protocol) 패킷 내부에 존재하며, 한편 RTP 패킷은 UDP-IP 패킷 내부에 존재한다.

첫째, VoIP 는 TCP 를 사용하지 않고 UDP (datagram) 를 사용하는데, 이는 실시간 어플리케이션에서 TCP 가 너무 거추장스럽기 (heavy) 때문이다.

둘째, UDP 는 목적지에 도착한 패킷들의 순서를 제어하지 않고 목적지에 도착하는데 걸린 시간에 대해서도 상관하지 않는다 (datagram 개념). 이 두 특징은 전반적인 음성 품질 (다른사람이 말하는 것을 얼마나 잘 알아들을 수 있는지) 과 대화 품질 (대화를 하기가 얼마나 용이한지) 에 매우 중요하다. RTP 는 이러한 문제를 두 가지 방법으로 해결하는데 : 수신자가 받은 패킷을 올바른 순서로 정리하도록 해주며, 손실된 패킷이나 너무 오래 걸려서 도착하는 패킷을 기다리느라 너무 많은 시간을 대기하도록 하지 않도록 한다. (우리에게 필요한 것은 음성 패킷의 모든 부분이 아니라 그 순서가 제대로 갖춰진 패킷의 연속적인 흐름이다.)

```
                    Real Time Transport Protocol
```

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|V=2|P|X|  CC   |M|     PT      |       sequence number         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           timestamp                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           synchronization source (SSRC) identifier            |
+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
|            contributing source (CSRC) identifiers             |
|                             ....                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

위에 표시된 것은 :

- V 는 RTP 의 버전을 나타낸다.
- P 는 padding, 즉 무결성 검사를 위해 패킷 내에 사용되지 않는 바이트를 표시한다.
- X 는 확장 헤더의 존재를 의미한다.
- CC 는 고정된 헤더 다음에 붙는 CSRC 지시자의 수를 나타낸다. 예를들어 회의같은 경우라면 CSRC 필드가 사용된다.
- M 은 marker bit 이다.
- PT 는 페이로드 타입을 나타낸다.

RTP 프로토콜에 대한 완전한 설명과 그 응용에 대해서는 관련 RFC 1889 (http://www.ietf.org/rfc/rfc1889.txt) 와 1890 (http://www.ietf.org/rfc/rfc1890.txt) 를 참고하라.

4.5. RSVP

VoIP 에는 RSVP 와 같이 QoS 를 처리할 수 있는 다른 프로토콜도 사용된다.

RSVP is a signaling protocol that requests a certain amount of bandwidth and latency in every network hop that supports it. RSVP 는 그것이 지원되는 모든 네트웍 합 (hop) 에서 특정한 양의 대역폭과 레이턴시 (latency) 를 요구하는 신호 프로토콜이다.

RSVP 에 대한 자세한 정보는 RFC 2205 (http://www.ietf.org/rfc/rfc2205.txt?number=2205) 를 보라.

4.6. Quality of Service (QoS)

사용자는 쌍방 대화식의 음성 교환을 원하기 때문에 VoIP 어플리케이션은 실시간 데이터 스트리밍을 요구한다는 것을 수차례 이야기하였다.

불행하게도 TCP/IP 는 이러한 목적을 보장해 줄 수가 없고, 단지 "힘 닿는대로" 처리해 줄 뿐이다. 따라서 우리는 패킷이 지나가는 모든 라우터에서 패킷을 관리해 줄 수 있도록 하는 책략과 정책을 새울 필요가 있다.

어떻게 하느냐 하면 :

1. IP 프로토콜의 TOS 필드는 서비스 종류 (type of service) 를 표시하는데 : 이 값이 높으면 별로 긴급하지 않다는 것을, 낮으면 낮을수록 점점 더 실시간의 긴급함을 나타낸다.
2. 패킷을 큐에 집어넣을 때 :
   a. FIFO (나중에 온 패킷이 가장 먼저 서비스) 는 가장 멍청한 방법으로, 패킷이 도착한 순서대로 서비스하는 것 보다도 나쁘다.
   b. WFQ (가중치를 둔 공정한 대기열) 는 패킷의 공정한 (fair) 서비스를 위해 노력한다. (예를들어 FTP 서비스가 모든 가용 대역폭을 잠식하도록 내버려두지 않는다.) 이는 데이터 흐름의 종류를 판단하여, 통상 UDP 패킷을 하나 처리하고 TCP 패킷을 하나 처리하는 공정한 방법에 의해 이루어진다.
   c. CQ (사용자 정의 대기열) 은 사용자가 우선순위를 결정한다.
   d. PQ (우선순위를 가지는 대기열) 은 몇 개 (통상 4 개) 의 대기열로 이루어지는데 각각은 자신의 우선순위를 가지고 있다 : 패킷은 높은 우선순위의 대기열에서 우선적으로 서비스되며, 높은 우선순위의 대기열이 비면 그 아래의 대기열을 서비스하기 시작한다.
   e. CB-WFQ (단계에 기반해 가중치를 주는 공정한 대기열) 은 WFQ 와 비슷하지만, 추가적으로 64 까지의 단계와 각각에 해당하는 대역폭 값을 가지고 있다.
3. 쉐이핑 (shaping[7]) 을 통해 아래와 같은 두 가지 스트림에 자원을 배분할 때 각각에 대한 대역폭 제한을 두는 방법으로
   a. 수신
   b. 송신
4. RED (Random Early Detection) 와 같이 체증을 회피하는 방법으로 [8]

QoS 에 대한 방대한 정보는 IETF 에서 Differentiated Services (http://www.ietf.org/html.charters/diffserv-charter.html) 를 참조하라.

4.7. H323 신호 프로토콜

예를들어 MS Netmeeting과 같은 프로그램은 VoIP 를 위해 H323 프로토콜을 사용한다.

이 프로토콜은 상호간 대화를 하도록 여러 요소를 허용하는데 :

1. VoIP 연결을 초기화하는 클라이언트인 터미널을 제공한다. 비록 터미널들은 다른 요소가 없더라도 상호간 대화할 수 있으나, 확장성의 측면에서 몇 가지 추가적인 요소를 필요로 한다.
2. Gatekeepers 가 제공된다. 이것의 본질적은 동작은 :
   a. IP 주소 대신에 이름 [9] 을 사용할 수 있도록 주소 변환을 수행한다.
   b. 호스트들과 사용자들에 대해 연결을 허용하고 거부하는 등의 통제를 수행한다.
   c. 대역폭을 관리한다.
3. 게이트웨이가 제공된다. 이는 TCP/IP 와 PSTN 의 변환 기준점이다.
4. 회의 기능을 위해 다중점 제어장치 Multipoint Control Units (MCUs) 가 제공된다.
5. 또한 프락시 서버가 사용된다.

H323 은 VoIP 뿐만 아니라 화상통신, 데이터 통신도 지원한다.

H323 은 VoIP 에 대하여 오디오 코덱인 G.711, G.722, G.723, G.728, G.729 를 전송할 수 있고, 비디오 코덱에 대해서는 H261 과 H263 을 지원한다.

H323 에 대한 더 많은 내용은 OpenH323 Standards (http://www.openh323.org/standards.html) 와 H323 에 대한 이 사이트 (http://www.cs.columbia.edu/~hgs/rtp/h323.html), 그리고 표준을 기술하고 있는 ITU H-series Recommendations (http://www.itu.int/itudoc/itu-t/rec/h/) 에서 찾아볼 수 있다.

실제적으로 구현된 여러 상용 어플리케이션은 Microsoft Netmeeting (http://www.microsoft.com), Net2Phone (http://www.net2phone.com), DialPad (http://www.dialpad.com) ... 등이 있고, 특히 프리웨어로서 개발된 어플리케이션은 OpenH323 Web Site (http://www.openh323.org) 에서 찾을 수 있다.

5. 요구사항

5.1. 하드웨어 요구사항

소규모 VoIP 시스템을 구축하기 위해 당신이 필요한 하드웨어는 아래와 같다 :

1. 386 또는 그보다 좋은 PC
2. 전이중 (full duplex) 방식이 가능한 사운드카드
3. 네트웍 카드나 인터넷 망, 혹은 2 대의 PC 사이의 통신을 가능하게 하는 인터페이스

정상적인 통신을 위해서는 위의 세트가 두 꾸러미 있어야 한다.

위에 나열한 조건은 VoIP 연결을 위한 최소한의 요구사항이다 : 다음으로 우리는 실제적인 상황에서 구현하기 위해서는 (인터넷을 통해 구현하기 위해서는) 추가적인 하드웨어가 필요함을 알게 된다.

말하는 동안에 상대방의 목소리를 듣기 싫은 것이 아니라면 사운드카드는 반드시 전이중 (full duplex) 의 것이어야 한다.

또한 압축된 형태(4.3절 참고)로 데이터 스트림을 처리해 줄 수 있는 하드웨어 카드 (다음 절에 설명) 가 추가적으로 필요하다.

5.2. 하드웨어 가속장비

우리는 하드웨어 가속능력을 가지고 있는 특별한 카드를 사용할 수 있다. 그 중 두 가지 (또한 현재 리눅스 커널에서 직접적으로 지원되는 것들) 는 아래와 같다 :

1. Quicknet PhoneJack
2. Quicknet LineJack
3. VoiceTronix V4PCI
4. VoiceTronix VPB4
5. VoiceTronix VPB8L

Quicknet PhoneJack 은 G723.1 (4.3절 참고) 처럼 오디오 스트림을 4.1 Kbps 로 압축해 주는 표준 알고리듬을 사용하는 사운드 카드이다.

이 제품은 phone (POTS 단자 [10]) 나 마이크-스피커 셋에 직접 연결할 수 있다.

또한 ISA 나 PCI 두 가지 방식으로 모두 제공된다.

Quicknet LineJack 은 PhoneJack 과 비슷하게 동작하지만 몇 가지 추가적인 기능이 있다. (다음 절에 설명)

VoiceTronix V4PCI 는 Quicknet LineJack 과 매우 유사한 PCI 카드이지만 phone 단자가 4 개 존재한다.

VoiceTronix VPB4 는 V4PCI 와 동일한 ISA 방식의 제품이다.

VoiceTronix VPB8L 는 8 개의 포트와 음성 기록 (logging) 기능을 가진다.

더 자세한 내용은 Quicknet web site (http://www.quicknet.net) 와 VoiceTronix web site (http://www.voicetronix.com.au) 를 참고하라.

5.3. 하드웨어 게이트웨이장비

Quicknet LineJack 와 VoiceTronix 카드는 PSTN 회선에 연결하여 VoIP 게이트웨이 기능을 지원한다.

이를 위해서는 게이트웨이를 관리하기 위한 소프트웨어가 필요하다. (다음에 설명)

5.4. 소프트웨어 요구사항

사용할 수 있는 운영체제는 다음과 같다 :

1. Win9x
2. Linux

Win9x 에서는 MS Netmeeting, Internet Phone, DialPad 등을 사용하거나 Quicknet 카드를 위한 Internet Switchboard (Quicknet web site (http://www.quicknet.net) 에서 제공) 를 사용할 수 있다.

경고!! : 최근의 Switchboard 를 사용하는 Quicknet 카드 (구형 버전도 마찬가지로) 는 Microtelco 계정 (유료) 에의 접근을 위해 인터넷 연결이 반드시 필요하다. 따라서 당신이 당분간 인터넷에서 고립된 상태로 남아있을 예정이라면 OpenH323 software (http://www.openh323.org) 을 설치해야 한다.

VoiceTronix 카드에 필요한 소프트웨어는 VoiceTronix web site (http://www.voicetronix.com.au) 에서 찾을 수 있다.

리눅스에서는 simph323 또는 ohphone 와 같은 프로그램이 Quicknet 하드웨어 가속장치를 지원하므로 OpenH323 (http://www.openh323.org) 웹사이트에서 그것을 가져다 쓰기만 하면 된다.

주의 : 모든 OpenH323 소스코드는 유저 디렉터리에서 컴파일되어야만 한다 (몇몇 환경 변수들의 수정이 필요하지 않다면). 그리고 컴파일 시간이 매우 오래 걸린다는 경고가 발생할 것인데, 왠만한 시간 내에 컴파일을 마치려면 매우 많은 메모리가 있어야만 한다.

5.5. 게이트웨이 소프트웨어

게이트웨이 기능 (TCP/IP VoIP 를 PSTN 회선에 접속) 을 관리하기 위해 아래와 같은 소프트웨어가 필요하다 :

- Internet SwitchBoard (http://www.quicknet.net) (인터넷에 연결되었을 때만 필요) 는 윈도우 시스템에서 H323 터미널처럼 동작한다.
- PSTNGw 는 리눅스와 윈도우 시스템에서 동작하며 OpenH323 (http://www.openh323.org/code.html) 에서 다운로드 받을 수 있다.

5.6. Gatekeeper 소프트웨어

gatekeeper 로는 아래와 같은 프로그램을 선택할 수 있다 :

1. Opengatekeeper, opengatekeeper web site (http://www.opengatekeeper.org) 에서 리눅스용과 Win9x 용을 다운로드할 수 있다.

2. Openh323 Gatekeeper (GK), 이쪽 (http://www.willamowius.de/openh323gk.html) 에서 다운로드할 수 있다.

5.7. 그 외의 소프트웨어

추가적으로 H323 를 따르는 유용한 소프트웨어가 있는데 :

- Phonepatch 는 NAT 방화벽 아래에서의 문제를 해결할 수 있다. 이것은 사용자가 (방화벽 외부와 내부에서 볼 수 있는) 웹 페이지를 호출하도록 만들어서 : 웹 어플리케이션이 원격지 호스트가 준비되었음을 인식하면 송신지 측으로 호출하여 (H323) 모든 준비가 이루어졌음을 알리고 연결을 설정하게 된다. Phonepatch 는 소유권이 있는 소프트웨어이며 (3분만 대화 연결이 가능한 데모버전도 있다.) 이쪽 (http://www.equival.com/phonepatch) 에서 다운로드 받을 수 있다.

6. 카드 설치

여기서는 리눅스와 윈도우 환경에서의 특별한 하드웨어 설정에 대해 알아본다.

6.1. Quicknet PhoneJack

알아본 바와 같이 Quicknet Phonejack 은 VoIP 가속기능을 가지고 있는 사운드카드이다. 이 장비가 지원하는 것은

- G.711 normal 과 mu/A-law, G.728-9, G.723.1 (TrueSpeech) 과 LPC10.
- 폰 커넥터 (전화기를 통해 직접 연결할 수 있도록 한다.) 또는
- 마이크와 스피커 잭

Quicknet PhoneJack 은 ISA (또는 PCI) 카드로서 PC 에 설치할 수 있다. 이 장비는 IRQ 없이 동작한다.

6.1.1. 소프트웨어 설치

윈도우 시스템에서는 아래를 설치해야 한다 :

1. 카드 드라이버
2. Internet Switchboard 프로그램 (신형의 Quicknet 카드를 사용하여 인터넷이 있어야 동작할 수 있다.)

모든 프로그램은 Quicknet web site (http://www.quicknet.net) 에서 다운로드 받을 수 있다.

Switchboard 를 설치한 다음 Quicknet 에 등록하면 장비의 모든 기능을 사용할 수 있다.

수화기를 들면 Internet Switchboard 가 작동하여 전화번호 입력을 기다린다. (전화기에서 직접 입력하면 된다.) 이때 가능한 동작은 :

1. 별표 "*" 를 누른 다음 IP 주소를 입력한 다음 (점 "." 은 별표 "*" 를 사용) 우물 정 "#" 을 누르면 된다.
2. (국제전화 국가코드를 포함하여) PSTN 전화번호를 입력하여 기존 전화가입자에게 전화를 걸 수 있다. 이때는 시간당 요금을 지불해야 할 게이트웨이 업체에 등록이 필요하다.
3. 미리 기억시켜둔 (두 자리까지의) 단축다이얼을 입력하면 IP 나 PSTN 을 통한 전화를 걸 수 있게 된다.

Internet Swichboard 는 H323 을 준수한다. 따라서 당신은 다른 사용자, 예를들자면 MS Netmeeting 사용자와도 통화할 수 있다.

경고!! Internet Switchboard 는 반드시 신형의 Quicknet 카드를 사용하여 인터넷에 연결되어야만 한다.

Internet Switchboard 대신에 OpenH323 어플리케이션인 openphone (http://www.openh323.org/code.html) (GUI 기반) 이나 ohphone (http://www.openh323.org/code.html) (커맨드라인으로 작동) 을 사용할 수 있다.

리눅스 시스템에서는 아래를 설치해야 한다 :

1. 카드 드라이버를 Quicknet web site (http://www.quicknet.net) 에서 받은 다음 컴파일한다. (반드시 현재 리눅스 소스 디렉터리를 `/usr/src/linux` 로 소프트링크나 하드링크 걸어두어야만 한다.)
2. 어플리케이션 openphone (http://www.openh323.org/code.html) 또는 ohphone (http://www.openh323.org/code.html) 을 설치한다.
3. 만약 당신이 개발자라면 SDK (ftp://ftp.quicknet.net/Developer/Linux/Docs/) 를 받아서 직접 자신만의 어플리케이션을 작성할 수도 있다. (윈도우용도 가능)

6.1.2. 설정

Internet Switchboard (또는 다른 어플리케이션) 을 통해 이런 것이 가능하다 :

1. 선호하는 압축 알고리듬을 변경
2. 지터 (jitter) 딜레이를 튜닝
3. 볼륨을 조절
4. 에코 소거수준을 조절

6.2. Quicknet LineJack

이 카드는 방금 설명한 모뎀과 매우 유사하며, 역시 게이트웨이 기능을 지원한다.

PSTNGx 어플리케이션 (리눅스와 윈도우 버전) 을 이쪽 (http://www.quicknet.net/code.html) 에서 다운로드받거나 게이트웨이 기능을 위해 Internet Switchboard 를 사용한다.

6.3. VoiceTronix 제품

1. 우선 이쪽 (http://www.voicetronix.com.au/vpb-driver-2.1.8.tar.gz) 에서 파일을 다운로드 받는다.
2. 압축을 풀어준다.
3. 'src/vpbreglinux.cpp' 를 README 파일의 지시대로 수정한다.
4. 'make' 라고 입력한다.
5. 'make install' 라고 입력한다.
6. src 디렉터리로 이동한다.
7. 'insmod vpb.o' 라고 입력한다.
8. 콘솔에서 'dmesg' 명령의 결과를 보고 major number 를 찾는다. 이것을 MAJOR 라 하자.
9. 위에서 찾은 MAJOR 를 적용하여 'mknod /dev/vpb0 c MAJOR 0' 라고 입력한다.
10. unittest 디렉터리로 이동해서 './echo' 라고 입력한다.

더 상세한 도움말은 README 의 지시대로 따른다.

나는 개인적으로 VoiceTronix 제품은 테스트해보지 않았으므로 VoiceTronix web site (http://www.voicetronix.com.au) 로 지원을 요청해 보도록 한다.

7. 설치

이 장에서 우리는 VoIP 시스템을 설치할 것이다. 처음에는 간단하게 시작하고 점점 더 복잡한 과정으로 진행해보자.

7.1. 간단한 연결 : IP 에서 IP 로

```
A (Win9x+Sound card)   - - -   B (Win9x+Sound card)

     192.168.1.1       - - -        192.168.1.2
```

```
192.168.1.1 가 192.168.1.2 를 호출
```

A 와 B 는 아래와 같아야 한다 :

1. MS Netmeeting (또는 유사 소프트웨어) 이 설치되어 있고 제대로 설정되어 있을 것
2. 네트웍 카드를 가지고 있거나 다른 종류의 TCP/IP 인터페이스로 상호간 연결되어 있을 것

이런 상황에서는 A 는 H323 호출을 B 의 IP 주소를 이용해서 B 에게 시도할 수 있다 (물론 B 가 Netmeeting 을 동작시키고 있는 상황에서). 그때 B 측에서 응답할 생각이 있다면 호출에 응하게 되고, 이후에는 VoIP 데이터 패킷이 전송되기 시작한다.

7.2. 이름을 사용하기

MS 윈도우 시스템을 사용한다면 랜상의 다른 사용자에게 NetBIOS 이름을 사용하여 호출할 수 있다. NetBIOS 는 저수준 프로토콜 NetBEUI 와 함께 작동할 수 있으며 또한 TCP/IP 와도 연동하 수 있다. 상대방으로의 연결을 이루기 위해 필요한 것은 오직 "컴퓨터 이름" 뿐이다.

```
        A              - - -             B

   192.168.1.1         - - -        192.168.1.2

      John             - - -           Alice


              John 이 Alice 를 호출
```

이것은 John 이 Alice 에게 보낸 호출 요청은 NetBIOS 프로토콜에 의해 IP 주소로 바뀌기 때문이다.

위에서 설명한 두 가지 예제는 구현하기에 매우 쉽지만 확장성이 없다.

인터넷과 같이 매우 큰 규모에서는 상대방의 IP 주소를 모르는 경우가 일반적이기 때문에 직접적인 호출을 사용하는 것은 불가능하다. 게다가 NetBIOS 는 메세지를 브로드캐스팅하는 것으로 동작하는데, 이것은 통상 ISP 의 라우터를 통과하지 못하므로 NetBIOS 를 이용해 컴퓨터 이름을 호출하는 것도 불가능하다.

7.3. WINS 서버를 이용한 인터넷 호출

NetBIOS 이름을 호출하는 아이디어는 WINS 서버를 사용하여 인터넷 환경에서 구현될 수 있다 : NetBIOS 클라이언트는 컴퓨터 이름을 해독하기 위해 WINS 서버에 질의하도록 구성될 수 있다.

같은 WINS 서버를 사용하는 PC 들 간에는 상호간 이름을 통한 직접 호출이 가능하다.

```
A (WINS 서버는 S) - - - - - - | - - - - -  B (WINS 서버는 S)
                              |
                              |
                              인  - - - - -   S (WINS 서버)
C (WINS 서버는 S) - - - - - - 터
                              넷
                              | - - - - -  D (WINS 서버는 S)
                              |
```

인터넷에서의 연결

A, B, C 와 D 는 서로 다른 서브넷에 존재하지만 상호간에 NetBIOS 이름을 호출하는 방법으로 동작할 수 있다. 오직 필요한 조건은 S 가 동일한 WINS 서버인 것이다.

참고 : WINS 서버는 NetBIOS 기능을 사용하며 얼마 안되는 서브넷을 통합하는데만 사용되기 때문에 아주 높은 퍼포먼스를 보이지는 않는다.

7.4. 큰 문제거리인 NAT [11]

IP 가 얼마 없는 문제는 가리기 (masquering, 또는 NAT, 즉 network address translation) 를 통해 해결할 수 있다 : 공인 IP 주소 (즉 인터넷에서 직접적으로 "접근" 할 수 있는 주소) 가 하나만 있는 경우에, 모든 다른 호스트들은 이 IP 를 사용하여 "가려질 (masqueraded)" 수 있다.

```
A  - - -

B  - - -   NAT 기능이 있는 라우터  - - -  인터넷

C  - - -
```

이런 방법으로는 곤란해요

예제에서 A, B 와 C 는 인터넷으로 웹 브라우징을 하거나, ping 을 보내거나, 메일이나 뉴스 서비스를 받을 수는 있지만 VoIP 호출을 할 수는 없다**.** 이것은 H323 프로토콜이 어플리케이션 레벨에서 IP 주소를 보내므로 그에 대한 응답이 사설 IP 주소를 사용하고 있는 송신자에게 도착하지 않기 때문이다.

해결책은 :

- H323 패킷이 이러한 문제를 피해가도록 수정된 리눅스 모듈이 있다. 그 모듈은 이쪽 (http://www.coritel.it/coritel/ip/sofia/nat/nat2/nat2.htm)에서 다운로드 받을 수 있다. 이것을 설치하기 위해 당신은 지정된 소스 디렉터리에 파일을 복사하고, Makefile 을 수정하여 컴파일한 뒤에 "modprobe ip_masq_h323" 이라는 명령으로 모듈을 올릴 수 있다. 불행히도 이 모듈은 현재 ohphone 소프트웨어에 대해서는 좀처럼 동작하지 않는다. (이유는 모르겠네요.)

```
A  - - -   NAT 기능의 라우터
```

```
B  - - -          +          - - -  인터넷

C  - - -  ip_masq_h323 모듈
```

이렇게 하면 동작한다

- 이런 문제를 해결하는 어플리케이션이 또 하나 있는데 : 5.7절 을 보라

```
A  - - -

B  - - -   PhonePatch   - - -  인터넷

C  - - -
```

이렇게 해도 동작한다

7.5. 리눅스 환경에서는

리눅스 시스템에서의 H323 터미널은 위에서 설명한 모든 기능을 사용할 수 있다.

7.5.1. Ohphone 프로그램의 문법

문법은 아래와 같다 :

```
"ohphone -l|--listen [options]"
"ohphone [options]... address"
```

- "-l", 표준 포트 (1720) 로 접속을 기다린다. (listening)
- "address", 호출을 기다리는 주소가 아니라 연결하고 싶은 상대 호스트의 "주소" 를 의미한다.
- "-n", "--no-gatekeeper", gatekeeper 가 없다면 이 옵션을 사용한다.
- "-q num", "--quicknet num", Quicknet card 를 /dev/phone(num) 장치로 작동한다.
- "-s device", "--sound device", /dev/device 사운드 장치를 사용한다.
- "-j delay", "--jitter delay", 지연 버퍼를 "delay" 만큼 세트한다.

또한 ohphone 을 실행했을 때 명령어 해석기에 직접 커맨드를 입력할 수도 있다. (decrease AEC 로써 에코 소거수준을 내릴 수 있다.)

7.6. gatekeeper 를 설치하기

당신은 gatekeeper 기능도 사용할 수 있다.

예제

```
(H323 터미널) A  - - -
                      \
(H323 터미널) B  - -  - D (Gatekeeper)
                      /
(H323 터미널) C  - - -

             Gatekeeper 설정
```

1. 호스트 A, B 와 C 는 D 에 gatekeeper 가 설치된 것으로 설정한다.
2. 각각의 호스트는 처음에 D 에게 자신의 주소와 이름 (그리고 별명도) 을 알려주어 D 에 도착하는 호출자가 그 정보를 이용할 수 있도록 한다.
3. 어떤 터미널이 호스트를 찾기 위해 D 에 요청하게되면, D 는 올바른 IP 주소를 알려줌으로써 연결을 성사시킨다.

여기서 gatekeeper 는 이름을 IP 주소로 해석하는 능력만을 가지고 있으며, 상호간 (IP 레벨에서) 접근할 수 없는 호스트들을 접속해주는 기능은 하지 않음을 유의해야한다. 다시 말하자면 gatekeeper 는 NAT 라우터처럼 기능하는 것이 아니다.

gatekeeper 코드는 이쪽 (http://www.opengatekeeper.org)에서 얻을 수 있으며 : 또한 openh323 library (http://www.openh323.org/code.html) 가 필요하다.

프로그램은 -d (데몬으로 동작) 옵션이나 -x (실행모드로 동작) 옵션으로만 시동하면 된다.

추가적으로 설정파일 (.ini) 을 찾을 수 있는 곳은 이쪽 (http://www.opengatekeeper.org/opengate.ini)이다.

7.7. 게이트웨이를 설치하기

앞에서 설명한 바와 같이 게이트웨이는 VoIP 를 PSTN 회선에 접속하여 인터넷을 통해 기존 방식의 전화가입자에게 전화를 할 수 있도록 기능한다. 따라서 추가적으로 PSTN 회선을 관리할 수 있는 카드가 필요하게 되는데 : Quicknet LineJack 이 있다.

OpenH323 web site (http://www.openh323.org) 에서 다음을 다운로드 받으면 된다 :

1. Linejack 용 드라이버
2. 게이트웨이 구축을 위한 PSTNGw 어플리케이션.

만약 실행파일이 동작하지 않는다면 소스코드와 openh323 library (http://www.openh323.org/code.html) 를 다운로드 받은 다음, 이들을 사용자 홈 디렉터리에 설치하면 된다.

이제 H323 게이트웨이를 작동하기 위해 할 일은 PSTNGw 를 실행하는 일 뿐이다.

8. 대역폭의 고려

지금까지 논의한 내용을 살펴볼 때 우리는 아직도 대역폭에 관련된 문제점을 해결하지 못했으며, 실시간 데이터 스트림을 만들어내는 문제를 가지고 있음을 주목하게 된다.

사실상 우리가 (우리의 패킷이) 지나가는 각각의 라우터에서 실시간 관리 프로토콜이 가능하지 않다면 절대 해결책이 없을 것이다. 그렇다면 어떻게 해야 하는가?

우선 가능한 한 가장 높은 수준의 압축 알고리듬을 사용하는 것이다. (LPC10 은 단지 2.5 kbps 대역폭만을 소비한다. 이는 약 313 bytes/s 에 해당한다.)

그리고 발생되는 패킷을 분류하여 TOS 필드에 가장 높은 우선순위를 기록함으로써 모든 라우터들이 긴급하게 우리 패킷을 서비스해 주도록 할 수 있다.

중요 : 이 모든 것을 처리한다 하더라도 통화 품질이 항상 양호하도록 보장하기에는 불충분하며, 기반시설의 확충과 (VoIP 를 위한) 대역폭 예약 등의 수단이 없다면 또한 불가능할 것이다. TCP/IP 는 실시간 프로토콜이 아니기 때문이다.

한 가지 가능한 솔루션이 있다면 그것으 대역폭을 보장하는 소규모 WAN 에서 시작하여 단계적으로 규모를 늘려가는 방법일 것이다.

결론적으로 우리가 알게된 것은 오직 하나인데 : PSTN 회선과 같은 소위 보장받는 서비스 (guaranteed services) 는 가입자 모두를 수용할 수가 없다는 것이다. 예를 들어 GSM 과 같은 서비스는 수 십만 명 이상의 클라이언트를 감당할 수는 없다.

어쨌든 이제 막 싹트기 시작하여 사용하는 사람들이 별로 없긴 하지만, VoIP 는 기존 PSTN 서비스를 대체할 유효한 대안책이 될 것이다.

9. 유용한 사이트

9.1. 오픈 소프트웨어

- Voxilla (http://www.voxilla.org)

- Linux Telephony (http://www.linuxtelephony.org)
- Open H323 web site (http://www.openh323.org)
- Speak Freely (http://www.speakfreely.org)

9.2. 상용 소프트웨어

- Fatamorgana Computers (http://www.fatamorgana.com)
- International Communication Union (http://www.itu.org)
- Voicetronix web site (http://www.voicetronix.com.au)
- Quicknet Web site (http://www.quicknet.net)
- Cisco Systems (http://www.cisco.com)
- www.metropark.com
- www.nbxsoftware.com

주석

1. 역주 : Public Switched Telephone Network, 공중전화망
2. 역주 : Universal Mobile Telecommunications Systems
3. 역주 : Global System for Mobile, ETSI에서 제정한 디지털 셀룰러 이동 통신 시스템 표준
4. 역주 : Total Access Coverage Service, 영국 표준의 아날로그 셀룰러 이동 통신 시스템 표준
5. 역주 : 송신지와 수신지를 왕복하는 데 걸리는 시간
6. 역주 : 전화통화품질에 대한 소비자의 주관적 평가를 평균한 수치
7. 역주 : traffic shaping 을 이야기하는 것 같습니다.
8. 역주 : 네트워크 상의 혼잡을 피하기 위하여 사용하는 알고리즘
9. 역주 : 컴퓨터 이름을 의미합니다. 자세한 내용은 7.2절 을 참고.
10. 역주 : Plain Old Telephone Service, 일반 전화단자를 의미합니다.
11. 역주 : 원문은 masquering 으로 되어 있으나 의미 전달상 NAT 라고 했습니다.