

오픈 소스 소프트웨어를 활용한 인터넷 전화 녹취 시스템

하은용†

요 약

인터넷 전화는 IP 기반 인터넷에서 VoIP(Voice over IP) 기술을 이용해서 음성 전화를 지원하는 인터넷 서비스다. 인터넷 전화는 영상통화, 메시징과 같은 인터넷 멀티미디어 서비스를 융합한 음성전화 서비스를 지원할 수 있는 장점을 갖고 있다. 특히 스마트폰을 통한 인터넷 소셜 네트워크 서비스가 보급되면서 기존의 전화망을 대체하는 서비스로써 인터넷 전화에 대한 연구와 개발이 활발히 진행되고 있다. 본 논문에서는 SIP(Session Initiation Protocol) 기반 인터넷 전화의 음성 통화 내용을 녹취하는 시스템의 설계 및 구현에 대해 설명한다. 인터넷 전화 녹취 시스템은 리눅스 기반으로 양방향 음성 스트림을 믹싱하는 기능, 라이브 패킷 스니핑 기능, 녹취 음성 파일 송신 기능은 공개 소프트웨어를 사용해서 구현하였다. 향후 개발된 시스템은 VoIP 기반 콜센터 시스템 등과 같은 복합 시스템을 구축하는데 있어 기반 기술로 활용될 계획이다.

주제어 : 인터넷 전화, VoIP, SIP, RTP, 오픈 소스 소프트웨어

An Internet Telephony Recording System using Open Source Softwares

Eun-Yong Ha†

ABSTRACT

Internet telephony is an Internet service which supports voice telephone using VoIP technology on the IP-based Internet. It has some advantages in that voice telephone services can be accompanied with multimedia services such as video communication and messaging services. Recently, the introduction of smart phones has led to a growth in social networking services and thus, the research and development of Internet telephony has been actively progressed and has the potential to become a replacement for the telephone service that is currently being used. In this paper we designed and implemented a recording system which records voice data of SIP-based Internet telephone's voice calls. It is developed on the linux system and has some features such as audio mixing of two in/out voice channels, live packet sniffing, and the ability to transfer mixed audio files to the log file server. These functions are implemented using various open source softwares. Afterwards, this VoIP recording system will be applied as a base technology to advanced services like a VoIP-based call center system.

Key Words : Internet Telephony, VoIP, SIP, RTP, Open Source Software

† 안양대학교 컴퓨터공학과 교수

논문접수: 2011년 7월 25일, 1차 수정을 거쳐, 심사완료: 2011년 10월 17일

1. 서론

인터넷 전화는 IP 기반 인터넷에서 VoIP 기술을 이용해서 음성 전화를 지원하는 인터넷 서비스다. 인터넷 전화는 영상통화, 메시징과 같은 인터넷 멀티미디어 서비스를 융합한 음성전화 서비스를 지원할 수 있는 장점을 갖고 있다. 특히 스마트폰을 통한 인터넷 소셜 네트워크 서비스가 보급되면서 기존의 전화망을 대체하는 서비스로써 인터넷 전화에 대한 연구와 개발이 활발히 진행되고 있다.

VoIP 인터넷 전화에서 호를 제어하기 위한 시그널링 프로토콜에는 IETF에서 제안한 SIP, MGCP(Media Gateway Control Protocol), IAX2(Inter-Asterisk eXchange)와 ITU에서 제안한 H.323이 있다. IAX의 경우 새로운 기능을 추가하기 위해서는 프로토콜 스펙에 정의해야 하기 때문에 확장이 용이하지 않고, H.323은 LAN 기반 비디오 회의를 지원하기 위해 IP 기반 음성, 영상, 데이터와 팩스 통신용으로 개발된 프로토콜로 최근까지 통신망 회사들이 서비스를 제공했지만 프로토콜이 복잡해서 서비스가 점차 축소되고 있는 상황이다. SIP는 프로토콜이 간단하고, 인터넷 응용 프로토콜인 HTTP 또는 SMTP와 같이 텍스트 기반 메시지 형식으로 가독성이 있어 VoIP 프로토콜의 신호제어 프로토콜로 널리 사용되고 있다[5].

인터넷 전화를 제공하는 통신 서비스 회사, 콜 센터를 운영하는 회사는 음성 통화 내역과 내용을 저장하는 서비스를 제공하기 위해서 VoIP 음성 통화를 녹취하는 시스템이 필요하다. 국내외에서 서비스되고 있는 SIP 기반 인터넷 전화는 초기 호 설정할 때 송수신되는 메시지를 연결경로 상에 패킷 캡처 기능을 수행하는 장치를 설치해서 SIP 메시지와 호 설정 후 음성 데이터를 전달하는 RTP(Realtime Transport Protocol) 프로토콜 데이터를 분석해서 음성 데이터를 수집하면 통화 내역과 내용을 녹취할 수 있다.

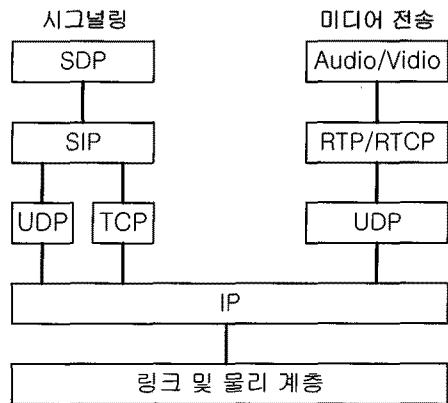
본 논문에서는 SIP 기반 인터넷 전화의 음성 통화 내용을 녹취하는 시스템의 설계 및 구현에 대해 설명한다. 인터넷 전화 녹취 시스템은 리눅스 기반으로 양방향 음성 스트림을 믹싱하는 기능, 라이브 패킷 스니핑 기능, 녹취 음성 파일 송신 기능은 공개 소프트웨어를 사용해서 구현하였다. 향후 개발된 시스템

은 VoIP 기반 콜센터 시스템 등과 같은 복합 시스템을 구축하는데 있어 기반 기술로 활용될 계획이다.

본 논문의 구성은 다음과 같다. 2장에서는 VoIP 시스템의 프로토콜 구조와 SIP, RTP 프로토콜에 대해 설명하고, 3장에서는 VoIP 녹취 시스템 설계에 대해 설명하고, 4장에서는 구현된 시스템의 실행 결과와 향후 연구 방향에 대해 설명하고, 5장에서는 관련 연구에 대해 설명하고, 마지막 6장에서는 결론을 맺겠다.

2. SIP기반 VoIP 프로토콜 스택 개요

SIP기반 VoIP 인터넷 전화 서비스를 지원하기 위한 프로토콜 스택은 [그림 1]과 같다. 호(call) 설정을 위한 시그널링은 SDP를 사용해 연결에 대한 속성들을 정의하고 SIP을 사용해 상호간 연결을 설정하고, 음성 또는 비디오 데이터를 전달하기 위해 RTP 프로토콜을 사용한다. SIP 메시지는 UDP 또는 TCP를 사용해서 전송하고, RTP 패킷은 UDP를 사용해서 전송한다.

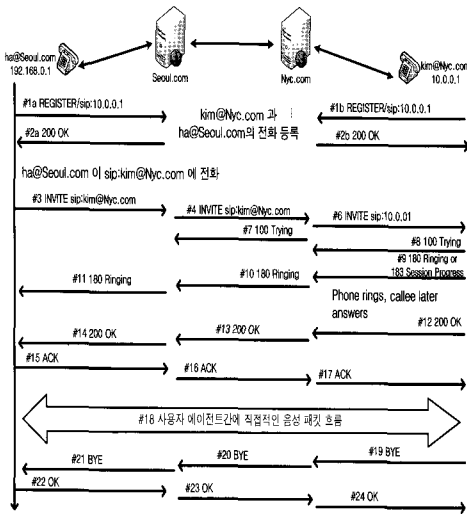


[그림 1] VoIP 프로토콜 스택 [3]

2.1 SIP 기반 인터넷 전화 메시지 흐름

일반적으로 SIP 기반 인터넷 전화는 [그림 2]와 같이 프로토콜 흐름을 따른다. 먼저 사용자 에이전트 역할을 하는 인터넷 전화기(소프트웨어 또는 하드웨어 폰)가 SIP 등록 서버(SIP registrar)에 자신의 이름, IP 주소, 연결 포트 등과 같은 정보를 등록한다(#1-#2). 전화기간에 호 설정은 caller 측에서 SIP

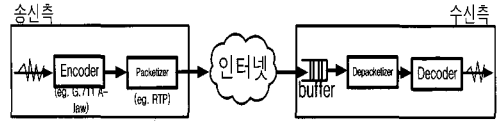
<INVITE> 메시지를 전송해서 호 설정을 요청하면, callee 측에서 <100 Trying> 메시지, <180 Ringing> 또는 <183 Session Progress> 메시지와 <200 OK> 메시지로 응답하고, caller 가 <ACK> 메시지를 전송함으로써 호 설정이 완료된다(#3-#17). 호 설정 후, 상호간에 미디어(음성, 영상) 통신은 RTP 패킷의 페이로드에 미디어를 실어서 보낸다(#18). 호 해제는 <BYE> 메시지와 <200 OK> 메시지 송수신을 통해 수행한다(#19-#24) [1][2].



[그림 2] SIP 기반 인터넷 전화 메시지 흐름

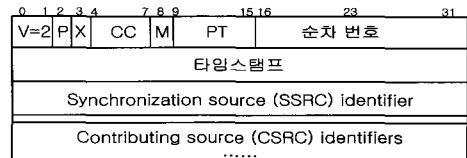
2.2 RTP를 통한 음성 데이터 전송

RTP는 음성, 비디오 등 실시간 데이터를 전송하기 위한 프로토콜이다[4]. 미디어 코덱은 SIP 호설정시 상호간에 협정된다. 예를 들어, 코덱으로 PCM a-law 방식을 사용하기로 협정되면, RTP를 통해 전달되는 음성 데이터는 PCM a-law 방식으로 인코딩되어 RTP 패킷의 페이로드 필드에 실리고, RTP 패킷은 UDP 세그먼트에 캡슐화되어 전송된다. [그림 3]은 송신측에서 오디오 데이터가 RTP 패킷으로 인코딩되어 네트워크를 통해 전달되고, 수신측에서 역으로 RTP 패킷을 수신하고 디코딩해서 원래의 오디오 데이터를 복원하는 과정을 설명하고 있다.



[그림 3] 음성 인코딩/디코딩

RTP 헤더는 [그림 4]와 같이 페이로드타입, 순서 번호, 타임스탬프, 동기화 출발지 식별자 등으로 구성된다. 예를 들어, 페이로드타입 필드 PT 값이 8이면, 패킷의 페이로드 필드에 ITU-T G.711 PCM A-law 64 kbps 오디오 데이터가 실려 있음을 뜻한다.



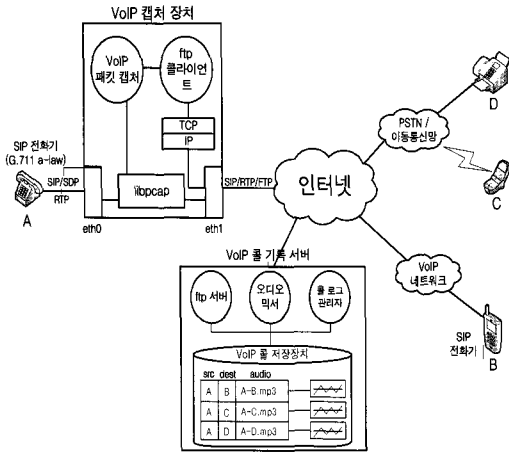
[그림 4] RTP 헤더

3. VoIP 인터넷 전화 녹취 시스템 설계

본 장에서는 VoIP 인터넷 전화 녹취 시스템을 포함한 통신시스템 전체적인 구성에 대해서 설명하고, VoIP 콜 녹취를 위한 모듈 설계 및 보조 기능을 수행하기 위해 적용된 공개 소스 소프트웨어 사용에 대해 설명하겠다.

3.1 VoIP 인터넷 전화 녹취 시스템 구성

인터넷 전화 시스템을 포함하는 통신 시스템은 [그림 5]와 같이 구성된다. 기존 유선 전화망(PSTN)과 셀룰라 이동통신망이 인터넷에 연결되어 있고, VoIP 인터넷 전화를 지원하는 VoIP 망이 인터넷에 연결되어 있다. 특별히 VoIP 전화 녹취를 위해 SIP 전화기가 연결된 VoIP 캡처 장치와 녹취된 내용을 보관하는 VoIP 콜 기록 서버가 인터넷에 연결된다.



[그림 5] VoIP 인터넷 전화 시스템 구성

3.2 VoIP 패킷 캡처 장치

VoIP 캡처장치는 임베디드 리눅스 박스로 이더넷 LAN 인터페이스를 통해 인터넷과 SIP 전화기에 연결된다. 패킷 필터링 기능을 제공하는 공개 라이브러리 libpcap[6]를 사용해서 패킷 캡처 박스를 통과하는 VoIP 콜 관련 이더넷 패킷 프레임을 캡처한다. VoIP 패킷 캡처 모듈은 콜 설정과 콜 해제와 관련된 SIP 프로토콜 메시지를 분석하고 처리해서 콜 관련 정보를 수집한다. 콜 관련 정보는 INVITE, RINGING, SESSION PROGRESS와 OK 메시지에 포함된 속성을 분석해서 얻는다. 속성에는 From, To 와 미디어 관련해서 오디오/비디오, 사용된 인코딩 방식, 전달 프로토콜, 포트 번호 속성들이 있다. 예를 들어, INVITE 메시지를 캡처한 내용을 [그림 6]에서 보면, 전화를 건 측의 송신자 정보 From은 'sip:07070168146@samsung070.com', 받는 측 수신자 정보 To는 'sip:0314670873@samsung070.com' 임을 알 수 있고, 미디어는 'm=audio 26284 RTP/AVP 8 101'에서 보면 audio를 RTP를 사용해서 포트번호 42656으로 PCMA 방식으로 전송함을 알 수 있다.

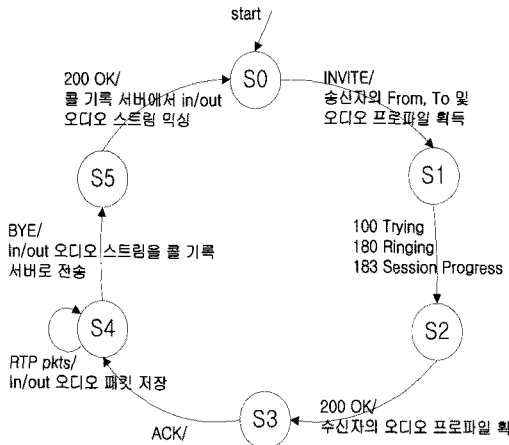
```

INVITE sip:0314670873@samsung070.com SIP/2.0
From: "Zilog"<sip:07070168146@samsung070.com>;
tag=95c26108-.....
To: <sip:0314670873@samsung070.com>
Call-ID:
95c19840-c0a80166-13c4-56-460e1c4e-56@samsung070.com
CSeq: 1 INVITE
Via: SIP/2.0/UDP 192.168.1.102:5060;
branch=z9hG4bK-56-150fc-370460cc
Max-Forwards: 70
Supported: replaces,timer
User-Agent: IRLink-ZiPhoneX-v3.3
Accept: application/sdp
Contact: <sip:07070168146@192.168.1.102>
Session-Expires: 1800, Min-SE: 90
Allow: INVITE,ACK,CANCEL,BYE,PRACK,REFER,
SUBSCRIBE,NOTIFY,UPDATE
Allow-Events: talk,hold
Content-Type: application/sdp, Content-Length: 190
v=0, o=- 56481 0 IN IP4 192.168.1.102
s=-, c=IN IP4 192.168.1.102
t=0 0
m=audio 26284 RTP/AVP 8 101
a=rtpmap:8 PCMA/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15, a=sendrecv
    
```

[그림 6] SIP INVITE 메시지

[그림 2]에서 설명한 SIP 기반 인터넷 전화 메시지 흐름도에 따라 녹취하는 소프트웨어의 엔진 부분을 설계하기 위한 녹취 엔진 상태 전이도는 [그림 7]과 같다. 녹취 엔진의 초기 시작 상태 S0에서는 전화를 거는 측에서 보낸 INVITE 메시지를 캡처해서 From, To 정보와 오디오 프로파일 정보를 획득하고, 콜 설정 진행 상태인 S1로 전이한다. S1에서는 Trying, Ringing 또는 Session Progress 메시지를 캡처한 후 S2로 전이한다. S2상태에서 전화를 받는 측에서 보낸 OK 메시지를 캡처하면 수신측의 오디오 프로파일 정보를 획득하고 S3상태로 전이한다. S3상태에서 수신측으로 보낸 ACK 메시지를 캡처하고 오디오 패킷 송수신 상태인 S4로 전이한다. S4상태에서는 RTP 패킷을 캡처하고 패킷에 실린 오디오 데이터를 추출해서 in/out 방향의 오디오 스트림에 맞게 지정한 오디오 파일에 기록한다. S4상태에서 BYE 메시지를 캡처하면 in/out 양방향 오디오 파일을 닫고, 두 오디오 파일을 콜 기록 서버로 전송한다. 파일 전송은 캡처 장치에서 오픈 소스 소프트웨어인 FTP 클라이언트를 사용하고, 콜 기록 서버 측에서 FTP 서버는 vsftpd[9]를 사용한다. S5상태에서 OK 메시지를 받으

변 초기상태인 S0로 전이해서 새로운 콜 캡처를 반복한다. 콜 기록 서버에서는 in/out 양방향 오디오 스트림을 믹싱해서 믹싱된 결과 파일인 mp3 또는 wave 파일을 콜 로그 데이터베이스에 저장한다. 오디오 믹싱 및 플레이어 기능을 갖는 오픈 소스 소프트웨어인 SoX[8]를 사용해서 믹싱을 수행한다.



[그림 7] VoIP 인터넷 전화 녹취 상태전이도

3.3 VoIP 콜 기록 서버

VoIP 콜 기록 서버는 리눅스가 탑재된 시스템으로 VoIP 캡처 장치와 연동된다. FTP 서버는 vsftpd를 사용하고, 오디오 믹싱은 앞에서 설명했듯이 오디오 변환 및 플레이어 기능을 갖춘 SoX 프로그램을 사용한다.

콜 로그 관리자는 VoIP 전화 통화 내역에 대한 데이터베이스 관리를 수행한다. 데이터베이스는 오픈 소스 데이터베이스인 MySQL DBMS[7]를 사용해서 구성한다. 전화 통화 내역 데이터베이스(DB_call_log)는 콜 로그 테이블(TBL_call_log)과 오디오 테이블(TBL_audio)로 구성된다. 각 테이블의 구성 필드는 <표 1>과 같다.

<표 1> 전화 통화 내역 데이터베이스 구성

a) TBL_call_log

필드설명	필드명	데이터타입
식별번호	call_ID	int
송신자	caller	char
수신자	callee	char
시작시간	start_time	char
종료시간	end_time	char
통화시간	calltime	int
in 오디오	in_audio	char
out 오디오	out_audio	char
mix 오디오	mix_audio	char

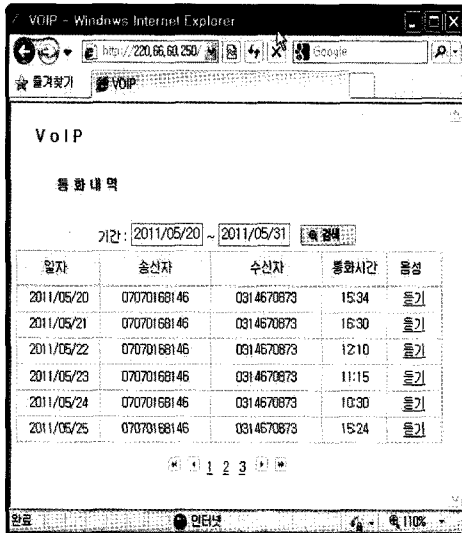
b) TBL_audio

필드설명	필드명	데이터타입
식별번호	audio_ID	char
오디오파일명	audio_file	char

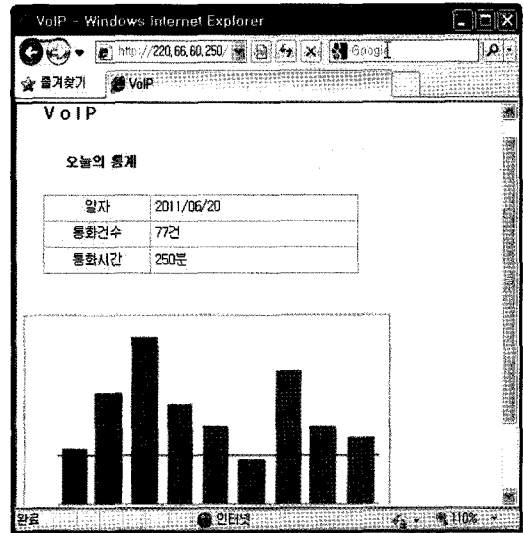
콜 로그 관리자는 스크립트 언어인 PHP를 이용해서 구현하고, 다음과 같은 기능을 수행한다.

- 리스트 기능 : 일별, 주별, 월별, 기간별 통화내역 리스트
- 검색 기능 : 송신자 또는 수신자를 검색 키로 통화 검색
- 통계 기능 : 일별, 주별, 월별 통화 통계

[그림 8]은 콜 로그 관리자의 각 기능에 대한 화면 설계다. 지면 관계상 몇 가지만 설명하기로 한다. 통화 내역은 시작일자와 끝 일자를 입력받아 주어진 기간에 전화 통화 내역을 화면에 출력한다 [그림 8a]. 검색된 리스트에서 음성 듣기를 클릭하면 오디오 음성을 재생하는 화면에서 통화 내용을 들을 수 있다.

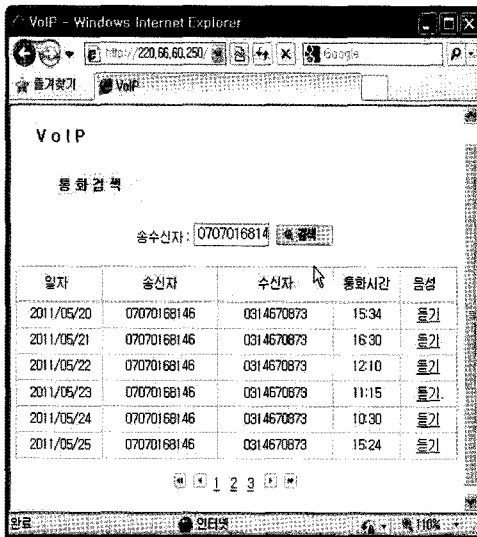


a) 통화 내역 리스트



c) 오늘 통계

[그림 8] 콜 로그 관리자 화면



b) 통화 검색

통화 검색은 송수신자 항목을 입력하고 검색을 실행하면 [그림 8b]와 같이 입력된 송수신자의 통화 내역 리스트를 출력한다. 검색된 레코드에서 듣기를 실행하면 음성을 들을 수 있다. 오늘 통계 기능은 [그림 8c]와 같이 오늘의 통화건수, 통화시간과 시간 대별 통화 건수를 그래프로 표현한다.

4. 구현 및 실행 결과

VoIP 인터넷 전화를 녹취하는데 가장 핵심적인 부분은 상태 전이도에 따른 녹취엔진 구현이다. 본 장에서는 패킷 캡처를 시작하기 전에 필요한 libpcap 라이브러리 관련 네트워크 장치와 패킷 필터 설정에 대해 설명하고, 캡처된 SIP 메시지와 RTP 패킷 처리에 대해 설명하고, 저장된 오디오 파일 믹싱과 파일 전송에 대해 설명한다.

4.1 캡처 인터페이스 설정

libpcap를 이용해서 실시간 패킷을 캡처하기 위해 네트워크 장치를 찾고(pcap_lookupdev), 네트워크 장치를 통해 송수신되는 모든 패킷을 캡처하기 위해 동작모드를 promiscuous 모드로 설정한다(pcap_open_live). 또한 네트워크 주소와 네트워크 마

스크를 알아내고 (pcap_lookupnet), 패킷 필터링 규칙을 적용해서 VoIP 관련 패킷을 캡처하도록 한다 (pcap_compile, pcap_setfilter). 설정이 끝나면 이후 로그 분석을 위해서 pcap 덤프 파일을 오픈한다 (pcap_dump_open). 이제 실시간으로 캡처된 이더넷 패킷 프레임에서 인터넷 전화관련 정보를 추출하기 위해 VoIP 패킷 처리를 위한 콜백 함수를 호출한다. 콜백 함수에서는 캡처 패킷을 읽고(pcap_loop), 덤프 파일에 기록한 다음에 VoIP 패킷 정보처리를 실행한다. 모든 처리가 끝나면 덤프 파일을 닫고 (pcap_dump_flush, pcap_dump_close), 패킷 캡처를 중지한다(pcap_close). 지금까지 설명한 VoIP 패킷 캡처를 위한 메인 알고리즘은 <알고리즘 1>과 같다.

<알고리즘 1> VoIP 패킷 캡처 메인 알고리즘

```
begin
    find an available network device:
        pcap_lookupdev()
    open the device for sniffing at the promiscuous mode:
        pcap_open_live()
    get the network address and mask of the device:
        pcap_lookupnet()
    set a packet filtering rule:
        pcap_compile();
        pcap_setfilter();
    open a pcap dump file:
        pcap_dump_open()
    call my pcap callback function for processing VoIP packets:
        pcap_loop(... VoIP_callback_function ...);
        pcap_dump();
    close the pcap dump file:
        pcap_dump_flush();
        pcap_dump_close();
    stop packet capturing:
        pcap_close();
end;
```

4.2 SIP 메시지와 RTP 패킷 처리

메인 알고리즘에서 실제로 VoIP 패킷을 처리를 위해 호출한 pcap_loop 함수에서 호출하는 VoIP 콜백 함수의 알고리즘은 <알고리즘 2>와 같다. 패킷에서 IP 헤더와 UDP 헤더 정보를 추출해서, IP 프로토콜 헤더 값이 UDP이고, UDP 헤더의 포트번호가 SIP

포트번호인 5060과 같은 패킷에 대해서 SIP 메시지 처리를 수행한다. 메시지 처리의 구체적인 내용은 <알고리즘 3>에서 설명된다. VoIP 처리 엔진의 상태가 S4이고 RTP 패킷이 수신된 경우에, RTP 헤더를 추출하고, 페이로드 부분에 실린 오디오 정보를 VoIP 녹취 파일에 저장한다.

<알고리즘 2> VoIP 콜백 함수

```
function VoIP_callback_function
begin
    get IP header information from the captured packet
    if protocol is UDP and UDP_port == SIP_port(5060)
        call process_SIP_message();
    if state is S4 and RTP packet is received
        get RTP header
        save the payload into the opened audio file
    end;
```

SIP 메시지 처리는 [그림 7]에서 설명한 녹취 상태 전이도에 따라 캡처한 SIP 메시지의 종류에 따라 처리한다. INVITE 메시지를 수신하면 caller의 오디오 프로파일을 추출하고 상태 S1로 전이한다. Trying, Ringing 또는 Session Progress 메시지를 수신하면 상태 S2로 전이한다. OK 메시지를 수신하면 callee의 오디오 프로파일을 추출하고 상태 S3으로 전이한다. ACK 메시지를 받으면 하나의 콜 연결 설정이 완료되는 것으로 VoIP 콜 녹취를 위해 양방향 녹취용 오디오 파일을 오픈하고 상태 S4로 전이한다. 일련의 SIP 메시지 처리 부분은 알고리즘 <알고리즘 3>과 같다.

<알고리즘 3> SIP 메시지 처리

```
function process_SIP_message()
begin
    case INVITE message
        get audio profile of the caller
        go to state S1
    case Trying, Ringing, Session Progress
        go to state S2
    case OK
        get audio profile of the callee
        go to state S3
    case ACK
        open in/out audio files for recording VoIP call
        go to state S4
    end;
```

하나의 VoIP 콜의 양방향 오디오 녹취 파일이 생성되면, 두 파일을 오픈 소스 소프트웨어를 이용해서 믹싱하고, 콜 기록 서버로 전송해서 저장한다. 믹싱과 전송의 일련의 작업은 외부 shell 스크립트로 <알고리즘 4>를 실행해서 처리한다. 오디오 믹싱은 SoX 프로그램을 사용했고, 기록서버의 ftp 서버는 vsftpd를 사용했다. sox 명령의 옵션 -m은 인자로 주어지는 두 파일(\$1, \$2)을 믹싱해서 결과 파일을 생성하라는 것이다. ftp 서버에 로그인하기 위해 IP 주소, 사용자 ID와 암호를 HOST, USER와 PASS라는 각각의 변수에 지정하고, ftp 명령을 호출한다. ftp 명령 처리과정에서 출력되는 메시지를 저장하기 위해 리눅스의 입출력 리다이렉션 기능을 사용한다. 두개의 양방향 녹취 오디오파일과 믹싱된 결과 파일을 ftp 서버로 전송하기 위해 일련의 ftp 내부 명령은 내부 도큐먼트 방식을 사용해서 처리한다.

<알고리즘 4> 믹싱 및 파일 전송

```
# mix_and_ftp.sh
sox -m $1 $2 $3
mv $1 ../log ; mv $2 ../log; mv $3 ../log
HOST='xxx.xxx.xxx.xxx'
USER='xxxxxxx'
PASS='xxxxxxx'
ftp -v -n << END_CMD >> ftpupload.log
open $HOST
user $USER $PASS
prompt off
bin
cd log
prompt off
lcd ../log
put $1
put $2
put $3
quit
END_CMD
```

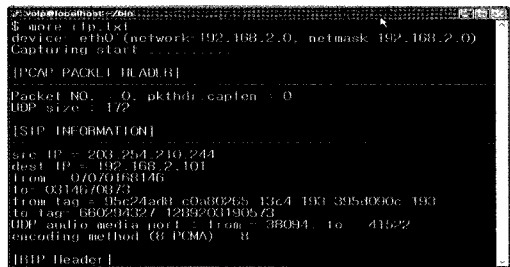
4.3 실행 결과

구현된 녹취 시스템의 실행 결과는 [그림 9]와 같다. [그림 9a]는 녹취 프로그램 실행 화면으로 첫 번째 인자 dump.pcap는 네트워크 인터페이스로 송수신되는 모든 패킷을 캡처해서 저장할 파일이고, 두 번째

인자 sip.pcap는 캡처된 패킷 중 SIP 콜과 관련된 패킷만 저장하는 파일이고, 세번째 인자 rtp.pcap는 in/out 오디오의 페이로드 부분을 저장할 파일이고, 마지막 rtp.txt는 SIP와 RTP와 관련된 정보를 텍스트 형태로 저장한 파일이다. [그림 9b]는 rtp.txt 파일에 저장된 텍스트의 일부를 캡처한 것으로 근원지/목적지 IP 주소, 송수신자 전화번호, 오디오 스트림 포트 번호 등의 정보를 보여준다. [그림 9c]는 in/out 오디오 파일과 믹싱된 오디오 파일을 보여준다.



a) VoIP 녹취 프로그램 실행



b) 캡처된 SIP와 RTP 정보



c) in/out 오디오와 믹싱 오디오

[그림 9] VoIP 녹취 프로그램 실행 결과

5. 관련 연구

SIP 기반 인터넷 전화의 음성 통화 내용을 녹취하는 기존 상용 제품에는 Avaya, Cisco, Nortel 등의 네트워크 제품 개발 업체에서 개발한 것들이 있다. Avaya 제품의 경우는 서버 및 클라이언트 시스템 모두 오픈 소스 개념이 아닌 Windows 시스템을 사용하고 있고, 음성 처리 관련해서는 G.711, G.723.1, G.729A, G.726/G.727, NetCoder 등의 코딩 기법을 지원하고 있다. Cisco와 Nortel 제품의 경우도 서버 및 클라이언트 시스템 모두 오픈 소스 개념이 아닌

Windows 시스템을 사용하고 있고, 음성 처리 관련해서는 G.711, G.723.1 등의 코딩 기법을 지원하고 있다. 하지만 본 논문에서 설계 구현한 인터넷 전화 녹취 시스템은 오픈 소스 소프트웨어를 활용해서 리눅스 운영체제를 탑재한 시스템에서 운영할 수 있는 시스템으로 공개성이 높은 특징을 갖는다.

6. 결 론

본 논문에서는 SIP 기반 인터넷 전화의 음성 통화 내용을 녹취하는 시스템의 설계 및 구현에 대해 설명했다. 인터넷 전화 녹취 시스템은 리눅스 기반으로 양방향 음성 스트림을 믹싱하는 기능, 라이브 패킷 스니핑 기능, 녹취 음성 파일 송신 기능은 오픈 소스 소프트웨어를 사용해서 구현하였다. 향후 개발된 시스템을 VoIP 기반 콜센터 시스템 등과 같은 복합 시스템을 구축하는데 있어 기반 기술로 활용될 것이다.

인터넷 전화 녹취 시스템을 다른 목적을 갖고 사용하게 되면 인터넷 전화 도청이라는 범죄적인 도구가 될 수 있다. 따라서 기존 상용 인터넷 전화 서비스를 제공하는 업계에서는 통신 보안을 위한 대책이 요구된다. 예를 들면, 콜 관련 SIP 메시지의 암호화 방법과 RTP에 실린 오디오 페이로드의 암호화를 통해 보안성을 높인 SRTP(secure RTP)를 인터넷 전화 서비스에 적용해서 악의적인 사용으로부터 통화 내용을 보호할 수 있는 보안 대책이 절실히 요구된다.

참 고 문 헌

[1] A. Johnston, S. Donovan, R. Sparks, C. Cunningham and K. Summers(2003), Session Initiation Protocol (SIP) Basic Call Flow Examples, RFC 3665, <http://www.ietf.org/rfc/rfc3665.txt>

[2] Dorgham Sisalem, John Florou, Jiri Kuthan, Ulrich Abend and Hemming Schulzrinne (2009), SIP Security, Wiley.

[3] Edward Oguejiofor, Philippe Bazot, Bruno Georges, Rebecca Huber, Callum Jackson, Jochen Kappel, Cameron Martin, Bala S. Subramanian and Abhijit Sur(2007),

Developing SIP and IP Multimedia Subsystem (IMS) Applications, ibm.com/redbooks

[4] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson(2003), RTP: A Transport Protocol for Real-Time Application, RFC 3550, <http://www.ietf.org/rfc/rfc3550.txt>

[5] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley and E. Schooler(2002), SIP: Session Initiation Protocol, RFC 3261, <http://www.ietf.org/rfc/rfc3261.txt>

[6] libpcap: a portable C/C++ library for network traffic capture, <http://www.tcpdump.org/>

[7] MySQL - The world's most open source database, <http://www.mysql.com/>

[8] SoX - Sound eXchange, <http://sox.sourceforge.net/>

[9] vsftpd - the most secure and fastest FTP server, <http://vsftpd.beasts.org/>

하 은 용



1986 서울대학교
전자계산기공학과(학사)
1988 서울대학교
컴퓨터공학과(석사)
1997 서울대학교
컴퓨터공학과(박사)

1997~현재 안양대학교 컴퓨터공학과 교수
관심분야: 정보통신, 정보보안, 인터넷 프로토콜, 임베디드 시스템, RFID/센서네트워크, 공개 소스 소프트웨어, 스마트 서비스
E-Mail: eyha@anyang.ac.kr