

# 확장성을 고려한 Asterisk 기반 인터넷 전화 관리 방법

하은용  
안양대학교 컴퓨터공학과

## A Scalable Management Method for Asterisk-based Internet Telephony System

Eun-Yong Ha

Dept. of Computer Engineering, Anyang University

**요 약** 인터넷 전화망은 VoIP기술을 이용해서 음성 전화를 지원하는 인터넷 서비스다. 인터넷 전화는 영상통화, 메시징과 같은 인터넷 멀티미디어 서비스를 융합한 서비스를 지원할 수 있는 장점을 갖고 있다. 본 논문은 Asterisk를 기반으로 구축한 인터넷 전화망의 확장성을 고려한 효율적인 관리 방법을 제안한다. 기존 시스템은 SIP 사용자, 다이얼플랜, CDR, IVR 및 서버 연동 등의 기능을 관리하기 위해 텍스트 파일 형식의 설정 파일을 사용하였다. 본 논문은 관리의 효율성과 확장성을 위해서 DB 기반으로 여러 기능을 수행할 수 있는 관리시스템을 설계 구현하였고, 전반적인 관리를 웹을 통해서 할 수 있도록 Apache, MySQL, jQuery와 PHP 등의 오픈 소스 소프트웨어를 사용하여 구현하였다.

**주제어** : 인터넷 전화, 오픈 소스 소프트웨어, Asterisk, IVR, SIP, VoIP

**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. In this paper we suggested an Asterisk-based Internet telephony system which can be easily scalable. Most current systems use text files to manage their configuration: SIP users, dialplans, IVR service and etc. But we designed the management system which introduces database tables for efficiency and scalability. It also supports web-based functions developed by using Asterisk, Apache, MySQL, jQuery, PHP and open source softwares.

**Key Words** : Asterisk, Internet Telephony, IVR, Open Source Software, SIP, VoIP

### 1. 서론

인터넷 통신망의 저변확대로 인류가 정보통신 기술 중심으로 모든 산업, 사회, 정치, 경제, 문화 등의 대부분의 영역에서 혁명이라고 할 만큼 큰 변화가 이미 일어났

고 앞으로 계속 발전할 것으로 기대된다. 이제 스마트폰과 인터넷 연결은 우리 생활에서 필수가 되었다. 스마트폰으로 시작하고 스마트폰으로 끝나는 일과 속에서 카카오톡, 라인, 페이스북, 트위터 등과 같은 소셜 네트워크 앱 프로그램의 사용자는 문자 메시지, 이미지, 동영상 등

\* 본 논문은 안식년 기간 중 연구되었음

Received 27 May 2014, Revised 7 July 2014

Accepted 20 August 2014

Corresponding Author: Eun-Yong Ha(Anyang University)

Email: eyha@anyang.ac.kr

© The Society of Digital Policy & Management. All rights reserved. This is an open-access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0>), which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

ISSN: 1738-1916

으로 구성되는 멀티미디어 메시지를 교환하고 있다. 또한 메시지 교환 중에 VoIP(Voice over IP) 기술을 적용한 인터넷 음성 또는 영상 전화를 사용하고 있다. 전화 시스템은 인터넷 이전에도 정보를 교환하는 중요한 통신 수단으로 사용되어왔고, 지금도 인터넷 서비스 중에서 빠질 수 없는 핵심 통신 서비스다.

VoIP 인터넷 전화망에서 호를 제어하기 위해 시그널링 프로토콜에는 SIP(Session Initiation Protocol), MGCP(Media Gateway Control Protocol)와 H.323이 있다. SIP는 프로토콜이 간단하고, 인터넷 응용 프로토콜인 HTTP 또는 SMTP(Simple Mail Transfer Protocol)와 같이 텍스트 기반 메시지 형식으로 가독성이 있어 VoIP 프로토콜의 신호제어 프로토콜로 널리 사용되고 있다. 또한 현재 시스템에서 사용되고 있는 많은 소프트웨어들이 SIP 프로토콜을 표준규격으로 사용하고 있어 서버 시스템 및 인터넷과의 연동에서 높은 호환성을 갖는다[1].

본 논문에서 인터넷 전화망 시스템은 오픈소스 소프트웨어인 Asterisk를 기반으로 구축하였다. 리눅스 시스템에 구축된 Asterisk 서버는 기본적인 음성 전화 기능뿐만 아니라 통화 내역 기록 기능(CDR:Call Detail Record), 음성 녹취 기능, 대화식 음성응답 기능(IVR:Interactive Voice Response)을 구현하였다. 기존 시스템은 SIP사용자, 다이얼플랜, CDR, IVR 및 서버 연동 등의 기능을 관리하기 위해 텍스트 파일 형식의 설정 파일을 사용하였다. 본 논문은 관리의 효율성과 확장성을 위해서 DB 기반으로 여러 기능을 수행할 수 있는 관리시스템을 설계 구현하였다. 웹을 통한 사용자 관리 및 시스템 관리는 웹 서버는 Apache를 사용하였고, 스크립트언어는 PHP 등의 오픈 소스 소프트웨어를 사용해서 구현하였다.

본 논문의 구성은 다음과 같다. 2장에서는 인터넷 전화망의 구성과 오픈 소스 IP-PBX(Private Branch eXchange)인 Asterisk 서버 기능에 대해 설명하고, SIP와 RTP 프로토콜에 대해 설명한다. 3장에서는 Asterisk 기반 인터넷 전화망에서 사용자, 다이얼플랜, IVR 기능 등의 관리를 위한 기존 방법을 분석한다. 4장에서는 확장성을 고려한 제안한 방법에 대한 인터넷 전화망 관리에 대해 설명한다. 5장에서는 구현 시스템의 실행결과를 보여주고, 끝으로 결론을 맺겠다.

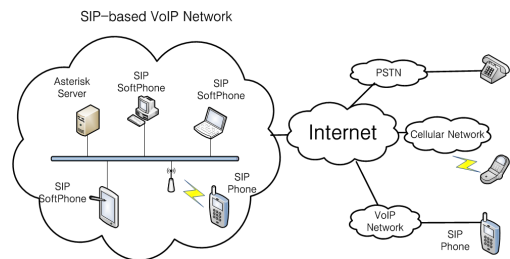
## 2. 인터넷 전화망 관련 연구

본 장에서는 오픈 소스 소프트웨어로 PBX 기능을 갖는 Asterisk 서버 중심의 VoIP 인터넷 전화망 구조와 Asterisk 서버의 기능에 대해 설명한다. VoIP 인터넷 전화를 위한 시그널링 프로토콜인 SIP 프로토콜에서 연결 설정 및 해제를 위한 메시지 흐름에 대해 설명한다. 또 오디오/비디오 데이터 전송에 사용하는 RTP 프로토콜에 대해 설명한다.

### 2.1 Asterisk 기반 인터넷 전화망 구조

Asterisk는 인터넷 전화망을 구축하는데 가장 중요한 구성 요소다. Asterisk를 기반으로 구성한 인터넷 전화망의 구조는 [Fig. 1]과 같다.

SIP Phone 상호간의 연결 설정 및 음성, 영상통화를 지원하기 위한 Asterisk 서버가 존재하고, 내부 네트워크는 보통 이더넷을 사용하여 구축된다. 인터넷을 통해 PSTN 및 이동 통신망에 스마트폰과 같은 다양한 통신 단말기가 상호 연결되고, VoIP 서비스를 제공하는 외부 VoIP Proxy Server를 통해 외부 통신장비가 Asterisk 서버와 통신을 한다.



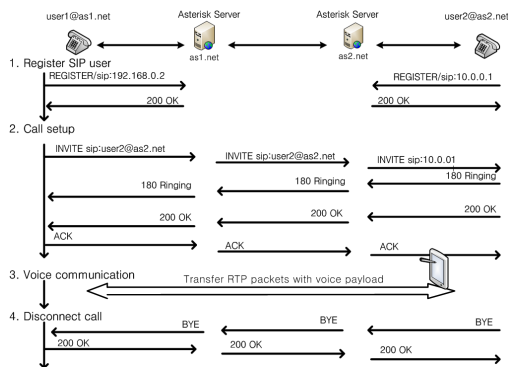
[Fig. 1] Asterisk-based Internet Telephony Network

Asterisk 서버는 SIP 전화 연결 설정, 통화 내용 기록, 음성 메일, 자동 응답 기능 등을 제공할 뿐만 아니라 새로운 응용을 개발해서 시스템에 접목할 수 있는 인터페이스를 제공한다[2].

### 2.2 SIP 기반 인터넷 전화 메시지 흐름

일반적으로 SIP 기반 인터넷 전화는 [Fig. 2]과 같이 프로토콜 흐름을 따른다. 먼저 SIP전화기(소프트웨어 또

는 하드웨어 폰)는 SIP 등록서버(SIP registrar)인 Asterisk 서버에 이름, IP 주소, 연결포트 등과 같은 정보를 등록한다. 전화기간에 호 설정은 caller 측에서 SIP <INVITE> 메시지 전송해서 호 설정을 요청하면, callee 측에서 <180 Ringing> 메시지와 <200 OK> 메시지로 응답하고, caller 가 <ACK> 메시지를 전송함으로써 호 설정이 완료된다. 호 설정 후, 상호간에 음성 또는 비디오 데이터는 RTP 패킷의 페이로드에 실어서 보낸다. 호 해체는 <BYE> 메시지와 <200 OK> 메시지 송수신을 통해 수행된다[3][4].



[Fig. 2] SIP message flow for internet telephony

### 2.3 RTP를 통한 음성 데이터 전송

RTP는 음성, 비디오 등 실시간 데이터를 전송하기 위한 프로토콜이다[5]. 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. 수신측에서는 역으로 RTP 패킷을 수신하고 디코딩해서 원래의 오디오 데이터를 복원한다.

## 3. Asterisk 기반 인터넷 전화망 분석

본 장에서는 Asterisk 기반 인터넷 전화망에서 사용자 및 여러 기능 관리할 때 사용하는 기존의 텍스트 파일을 사용한 설정 관리 방식에 대해 설명한다. 특히, 사용자 관리, 다이얼플랜 관리, IVR 관리 기능을 중심으로 설명한다.

### 3.1 기존 SIP 사용자 관리

Asterisk를 통해서 인터넷 전화를 사용하기 위해서는 사용자 등록이 필수적으로 이루어져야 한다. 사용자 설정과 관련된 사항들은 sip.conf 파일에 저장된다. [Fig. 3]와 같이 sip.conf 파일에 사용자 섹션 [1001]을 정의해서 SIP 사용자 계정을 생성한다. 사용자 속성에는 사용자 이름, 비밀번호, 타입, 호스트, 문맥(context), 호출자 ID, 메일박스 등이 있다.

```

[1001]
type = friend
host = dynamic
context = SIPphone
username = 1001
secret = xxxxxx
callerid = "Eun-Yong HA" <1001>
mailbox = 1001@default
nat = yes
qualify = yes
canreinvite = no
dtmfmode = rfc2833

[1002]
..... omitted
    
```

[Fig. 3] SIP user configuration: sip.conf

타입 값 friend는 SIP 사용자가 Asterisk 서버를 통해 전화를 걸 수도 있고 받을 수도 있음을 의미이고, 호스트 값 dynamic은 SIP 사용자가 임의의 IP 주소로 로그인 할 수 있음을 의미하고, 문맥은 SIP 사용자의 다이얼 플랜으로 extensions.conf 파일에 정의된다. nat 속성 yes는 SIP 사용자가 nat 기능을 하는 공유기 내부에 있을 수 있음을 의미한다. qualify 값 yes는 SIP 사용자의 접근가능 여부를 Asterisk 서버가 매 60초마다 점검한다는 의미다. 메일박스는 음성메일 사서함으로 사용된다. canreinvite 속성값은 no로 설정되면 오디오/비디오 미디어 데이터를 실은 RTP 패킷이 Asterisk 서버를 거쳐서 전달되고, yes로 설정되면 RTP 패킷이 Asterisk 서버를 거치지 않고 SIP 전화기간에 직접 전달된다. 따라서 서버의 녹취 기능을 수행하기 위해서 no로 설정한다.

### 3.2 기존 다이얼플랜 관리

내부 또는 외부에서 전화가 왔을 때 Asterisk 서버에서 취해야하는 동작에 대한 규칙을 정의하는 것을 다이

얼플랜(dialplan)이라 한다. 다이얼플랜의 설정은 extensions.conf 파일에 기술한다. 앞에서 설명한 SIP 사용자에게 문맥 SIPphone에 대해 [Fig. 4]와 같이 정의할 수 있다.

```
[SIPphone]
exten => _100X,1,Dial(SIP/${EXTEN},30,tTr)
exten => _100X,2,Congestion()
exten => _100X,3,Busy()
exten => _100X,4,Hangup()
..... omitted
```

[Fig. 4] Dialplan of SIP user : extensions.conf

사용자 설정에서 사용된 문맥 설정값 SIPphone의 내선번호에 대해 여러 exten이 정의된다. exten의 형식은 'exten => number, priority, application()' 으로 number은 전화번호, priority는 실행 우선순위, application은 해당 번호가 호출되었을 때 처리하는 응용 모듈로 구성된다. 전화번호 속성값 '\_100X'에서 X는 0~9까지 번호를 대체하는 부호다. 따라서 '\_100X'는 1000~1009 사이의 임의의 전화번호를 뜻한다. 정의된 우선순위에 따라 1번부터 4번까지 차례로 응용 모듈이 실행된다. 1번의 Dial()은 1001~1009 사이의 번호중 하나가 입력되었을 때 해당번호로 등록되어있는 SIP 폰에 다이얼링 기능을 수행한다. 2번 Congestion()은 혼잡 발생을 사용자에게 알려 상대방이 전화를 끊을 때까지 대기하게 된다. 3번 Busy()는 전화사용중이라는 신호를 보낸다. 4번 Hangup()은 호출을 끊는 역할을 한다.

### 3.3 기존 IVR 기능 관리

IVR(Interactive Voice Response) 기능은 안내 메시지를 재생해서 전화를 한 사람에게 정보를 제공해 주거나, 전화를 건 목적에 맞는 수신자에게 연결시켜 주는 작업을 실행한다. Asterisk에서는 IVR을 지원하며 기본적으로 extensions.conf 파일에 IVR 절차를 정의해서 구현이 가능하다.

[Fig. 5]는 전화번호 114에 대한 extensions.conf에 구현된 간단한 IVR 구분이다.

```
[114]
exten => 114,1,Answer()
exten => 114,2,Goto(ivr,114,1)
```

```
[ivr]
exten => 114,1,Answer()
exten => 114,2,Background(msg_top)
exten => 114,3,WaitExten(3)
exten => 114,4,Goto(ivr,114,2)

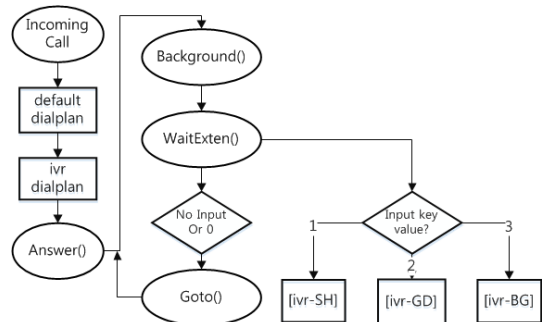
..... omitted

exten => 1,1,Goto(ivr-SH,114,1)
exten => 2,1,Goto(ivr-GD,114,1)
exten => 3,1,Goto(ivr-BG,114,1)
exten => 0,1,Goto(ivr,114,2)
..... omitted

[ivr-SH]
exten => 114,1,Background(msg_SH)
exten => 114,2,WaitExten(3)
exten => 1,1,Dial(SIP/1541,10)
exten => 1,2,Hangup()
exten => 2,1,Goto(ivr,114,1)
exten => 0,1,Goto(ivr-SH,114,1)
..... omitted
```

[Fig. 5] Dialplan for IVR : extensions.conf

[Fig. 5]에서 기본적으로 114라는 전화를 받게 되면 기본 다이얼플랜 문맥 [114]에서 Answer()를 실행하고, Goto()를 실행해서 문맥 [ivr]로 분기를 하게 된다. IVR 다이얼 플랜에 따른 대화식 음성 응답 서비스의 흐름은 [Fig. 6]과 같다.



[Fig. 6] Procedure of IVR dialplan

[Fig. 6]에서 사용자가 114로 전화를 하면 Asterisk 서버에서디폴트 다이얼플랜을 실행한 후, ivr 다이얼플랜을 실행한다. [Fig. 5]에서 설정한 절차에 따라 Answer()를 실행하고, Background()로 저장된 안내 음성 메시지를 재생하고, WaitExten()을 실행해서 사용자로부터 버튼입력을 기다린다. 입력된 번호에 따라 Goto()를 실행해서 다이얼플랜에 정의된 내선번호로 분기한다. 일정시간동

안 입력이 없으면 Goto()를 실행해서 다시 안내음성 메시지를 재생하는 명령으로 분기한다. 각각의 분기 문맥에서는 정의된 명령을 다시 실행한다.

예를 들어, 문맥 ivr-SH으로 분기하면 Background()를 실행해서 음성파일 mesg-SH을 재생하고, WaitExten()를 실행해서 3초 동안 입력을 기다리고, 입력된 값이 1이면 Dial()을 실행해서 SIP 사용자 1541으로 10초 동안 다이얼을 하고, 전화를 받지 않으면 Hangup()으로 호출을 끊는다.

#### 4. 제안한 확장성을 고려한 관리 기법

본 장에서는 기존의 텍스트 파일을 사용한 설정 관리 방식을 개선한 데이터베이스 중심의 기능 관리 기법에 대해 설명한다.

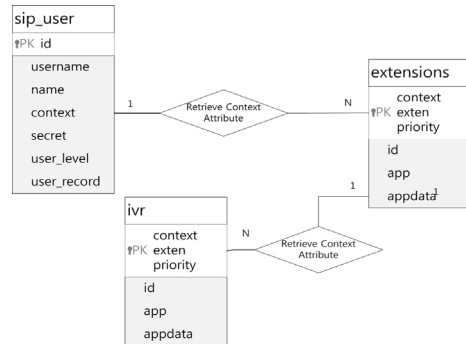
##### 4.1 전체 데이터베이스 구성

앞에서 설명했듯이 기존의 사용자, 다이얼플랜, IVR 기능 관리 방법을 보면 모두 텍스트 파일기반으로 이루어졌다. 따라서 설정을 변경하기 위해서는 텍스트 파일을 수정하고, 수정된 내용을 Asterisk 서버가 인식하도록 서버 프로세스를 재시작하는 과정을 반복해야한다. 즉, 사용자 등록, 다이얼플랜 정의, IVR 설정 등의 과정에서 서버 관리자가 직접 개입되어 파일을 수정하고 서버를 재시작해야 한다. 일반 사용자가 스스로 자신의 사용자 등록, 다이얼플랜 설계, IVR 설계 등의 작업을 할 수 없는 단점을 갖는다.

본 논문에서는 기존 관리 방법의 단점을 해결하기 위해 데이터베이스와 연동해서 Asterisk 서버를 관리하도록 하였다.

[Fig. 7]은 데이터베이스 테이블의 ER 다이어그램이다. 테이블 sip\_user는 SIP 전화 사용자 정보를 등록하는 테이블이고, 테이블 extensions는 사용자의 다이얼플랜을 저장하고, 테이블 ivr은 IVR 기능을 수행하기 위한 다이얼플랜을 저장한 테이블이다. 테이블 sip\_user의 필드 context를 통해 테이블 extensions과 1:N의 관계가 있고, context 값으로 문맥값 들을 추출한다. 또한 테이블 extensions의 필드 appdata를 통해 테이블 ivr과는 1:N의 관계가 있고, appdata 값으로 ivr 레코드를 검색해서 IVR

기능이 수행될 수 있도록 설계했다.



[Fig. 7] ER Diagram of Database Tables

##### 4.2 데이터베이스 테이블 구조

Asterisk와 연동된 데이터베이스의 테이블은 sip\_user, extensions, ivr 이며 테이블 구조는 <Table 1>과 같다.

<Table 1> Structure of DB tables

a) User table: sip\_user

Field Name	Description	Data type
id	identifier	int
username	user name	varchar
name	telephone No.	varchar
context	dialplan	varchar
secret	password	varchar
user_level	user level	enum
user_record	permission	enum

b) Tables of dialplan & IVR : extensions, ivr

Field Name	Description	Data type
id	identifier	int
context	IVR context	varchar
exten	input key val.	varchar
priority	priority	int
app	application	varchar
appdata	argument	varchar

SIP 사용자 정보를 저장하는 테이블 sip\_user는 7개의 필드로 구성된다. id는 사용자 고유번호로 기본키 역할을 하고, username은 사용자 이름, name은 SIP 전화번호, context는 사용자의 다이얼플랜 문맥으로 테이블 extensions에 정의되고, secret는 사용자 비밀번호, user\_level은 사용자 등급계정으로 관리자 또는 일반사용자로 구분되고, user\_record는 녹취록을 볼 수 있는 권한

을 지정한다.

사용자 다이얼플랜을 저장하는 테이블 extensions는 6개의 필드로 구성된다. id는 식별번호 역할을 하고, context는 IVR 다이얼플랜 문맥이름이다. exten은 사용자가 입력한 번호를 나타낸다. 예를 들어 사용자가 114를 누른 경우를 표현하는 경우 exten값은 114가 된다. priority는 다이얼플랜의 우선순위를 나타낸다. app은 IVR시스템에서 실행할 응용을 나타낸다. 응용에는 Answer(), Goto(), Dial(), Background() 등이 있다. appdata는 app 응용의 인자를 저장한다.

IVR 다이얼플랜을 저장하는 테이블 ivr은 테이블 extensions의 구조와 같다. 예를 들어, app이 'WaitExten' 이고 appdata가 '3'이면 대기 시간이 3초라는 것을 의미하고, app이 'Goto'고 appdata가 'ivr, 114, 2'이면 context가 [ivr], exten이 114, priority가 2인 곳으로 분기하는 응용을 실행하라는 의미이다.

### 4.3 IVR 서비스 절차

[Fig. 8]은 테이블 ivr에 저장된 레코드 스냅 샷의 일부분을 보여준다. 본 시스템에서 구현된 인터넷 전화망에서는 SIP 사용자는 기본적으로 [lan] 문맥으로 설정되어 있다. 사용자가 114로 전화를 하면 Asterisk 서버는 테이블 sip\_user에서 exten 값이 114인 사용자 계정을 찾아 다이얼플랜 문맥을 추출한다. 다음으로 테이블 ivr에서 추출한 문맥 값이 [lan]이고 exten이 114인 레코드를 검색해서 priority가 1인 레코드부터 app을 실행한다. 즉, Answer를 실행한 후 Goto 응용을 실행해서 "ivr, 114, 1"인 곳으로 실행을 넘긴다. 이제 [ivr] 문맥에서 exten이 114이고 priority가 1인 레코드를 찾아 Answer를 실행하고, 순차적으로 Background 응용을 실행해 음성 안내 메시지 Greeting을 재생하고, 안내음성에 따라 번호를 누르면 Goto 응용을 실행해서 IVR 서비스를 계속 진행한다. 이후 멘트에 따라 번호를 누르면 Dial을 실행해서 전화 연결을 하거나, Hangup을 실행해서 전화 연결을 종료하거나, Goto로 안내멘트를 처음부터 다시 듣기도 한다.

### 4.4 데이터베이스 연동

파일에 설정된 다이얼 플랜 대신 데이터베이스에 저장된 다이얼플랜을 수행하기 위해서는 해당하는 문맥을

데이터베이스와 연동해주어야 한다. 데이터베이스와 연동은 realtime 기능을 기본 문맥에 "switch => Realtime/[context]@table" 구문을 삽입함으로써 구현이 가능하다.

id	context	ext	p	app	appdata
1	[lan]	114	1	Answer	
2	[lan]	114	2	Goto	ivr,114,1
3	[ivr]	114	1	Answer	
4	[ivr]	114	2	Background	Greeting
5	[ivr]	114	3	WaitExten	3
6	[ivr]	114	4	Goto	ivr,114,2
7	[ivr]	1	1	Goto	ivr-SH,114,1
8	[ivr]	2	1	Goto	ivr-BG,114,1
..... omitted					
13	[ivr]	i	1	Goto	ivr,114,1
14	[ivr-SH]	114	1	Background	msg-SH
15	[ivr-SH]	114	2	WaitExten	3
16	[ivr-SH]	114	3	Goto	ivr-SH,114,1
17	[ivr-SH]	1	1	Dial	SIP/154,10
18	[ivr-SH]	1	2	Hangup	
19	[ivr-SH]	2	1	Goto	ivr,114,1
20	[ivr-SH]	i	1	Goto	msg-SH,114,1

[Fig. 8] Snapshot of ivr table

예를 들어, [Fig. 8]에 저장된 IVR 다이얼플랜을 연동하기 위해서 extensions.conf 파일에 테이블 ivr에 정의된 문맥 context를 실시간으로 참조할 수 있도록 다음과 같이 삽입한다.

```
[ivr]
switch => Realtime/[ivr]@ivr
[ivr-SH]
switch => Realtime/[ivr-SH]@ivr
```

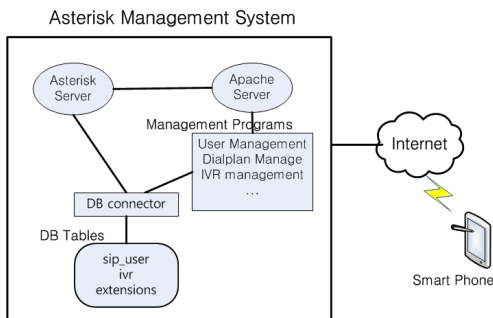
## 5. 구현 및 실행 결과

본 장에서는 IVR 서비스를 위한 동적 데이터베이스 설정 관리 및 서비스의 실행 결과에 대해 설명한다.

### 5.1 구현 환경

Asterisk 서버가 설정 관리를 위해 제공하는 CLI (command line interface) 방식은 설정 파일과 텍스트 명령을 사용하게 되어 있어서 일반 사용자가 접근하기 어렵다. 따라서 불편함을 해결하기 위해 웹 기반 GUI 관리

시스템으로 개발되었다 [2][6].



[Fig. 10] Web-based management system

웹 기반 GUI 관리 시스템의 구성은 [Fig. 10]과 같다. 웹 서버는 Apache 로 구축하고, 스크립트 언어는 PHP를 사용하고, 데이터베이스는 MySQL을 사용하였다 [7][8][9]. 사용자 등록과 IVR 서비스 설정을 웹 브라우저에서 실행하게 구현하였다.

### 5.2 SIP 사용자 등록

SIP 사용자 등록 화면은 [Fig. 11] 과 같다. 기본적인 항목만 입력해서 등록이 간단히 될 수 있도록 하였다. SIP 사용자 등록화면에서 아이디, 비밀번호, 내선번호와 연락처인 휴대폰번호와 전자우편을 입력해서 등록 버튼을 클릭하면 처리될 수 있도록 했다.

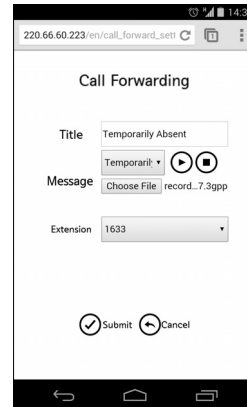


[Fig. 11] SIP user registration

### 5.3 사용자별 IVR 등록

SIP 사용자가 자신의 내선번호에 대해서 IVR 서비스

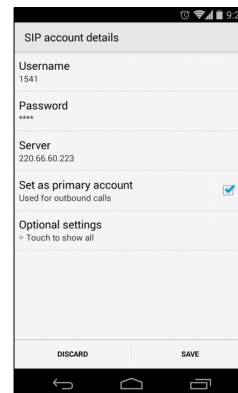
설정을 등록하는 화면은 [Fig. 12]와 같다. 제목 항목에는 자신이 전화를 받지 못하는 상황에 따라 음성 안내 메시지를 선택해서 업로드하고, 전화를 돌려받을 수 있는 내선번호를 입력하고 등록 버튼을 클릭하면 처리될 수 있도록 하였다.



[Fig. 12] User's IVR setup

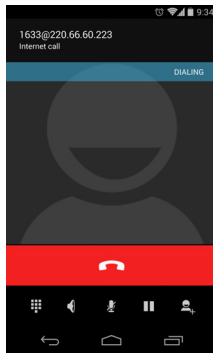
### 5.4 통화 실행 화면

SIP 소프트웨어를 사용해서 앞에서 등록한 사용자의 아이디, 비밀번호 과 Asterisk 서버 IP 주소를 소프트웨어에 설정하는 화면은 [Fig. 13]과 같다. 본 논문에서는 Linphone과 SmartPhone에 SIP 계정 설정을 사용하였다[10].



[Fig. 13] SIP account setting

[Fig. 14]는 IVR 서비스 실행으로 사용자가 1541 전화를 했을 때, 착신번호가 1541에서 1633으로 전환되어 통화연결을 한 결과를 보여준다.



[Fig. 14] Call forwarding state

## 6. 결론

본 논문에서는 오픈 소스 IP-PBX인 Asterisk를 기반으로 인터넷 전화망을 구축하는데 있어 SIP 사용자 등록이 간단하고, IVR 서비스 등록도 쉽도록 인터페이스를 설계 구현할 수 있도록 하였다. 특히 기존의 파일 중심의 관리를 데이터베이스로 전환하는 방법을 적용해서 웹 브라우저를 통해서 쉽게 관리할 수 있도록 하였다.

기존에 나와 있는 오픈 소스 소프트웨어 기반 구현이 일반적으로 텍스트 기반인 것을 데이터베이스화 하였을 뿐만 아니라, 기존의 데이터베이스로 구현된 사용자 인터페이스가 너무 전문적이고 복잡해서 일반인이 인터넷 전화망을 구성 운영 관리하기에 어려움이 있어서 일반인도 손쉽게 구성 운영할 수 있도록 단순화 하였다. 따라서 조그만 회사 혹은 조직에서도 인터넷 전화망을 경제적으로 구현할 수 있도록 하였다.

특히 요즘 정보통신 분야에서 화두가 되고 있는 오픈 소스 소프트웨어를 사용해서 전체적인 설계 구현을 하였다. 운영체제는 리눅스, IP-PBX는 Asterisk 서버, SIP 소프트웨어는 LinPhone 또는 스마트폰에 내장된 기본 앱, 웹서버는 Apache, 스크립트 언어는 PHP와 JavaScript 등, DBMS는 MySQL을 사용하였다.

향후 본 논문에서 구현한 결과를 소형 임베디드 장비로 사용되는 Raspberry Pi 와 OpenWRT를 지원하는 유무선 라우터에 포팅할 계획이다.

## ACKNOWLEDGMENTS

This paper was supported by Anyang University.

## REFERENCES

- [1] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., & Schooler, E. (2002), SIP : Session Initiation Protocol, RFC 3261.
- [2] Asterisk - The Open Source Telephony Projects, [www.asterisk.org](http://www.asterisk.org)
- [3] Johnston, A., Donovan, S., Sparks, R., Cunningham, C., & Summers, K. (2003), Session Initiation Protocol (SIP) Basic Call Flow Examples, RFC 3665.
- [4] Sisalem, D., Floroiu, J., Kuthan, J., Abend, U., & Schulzrinne, H. (2009), SIP Security, Wiley.
- [5] Schulzrinne, H., Casner, S., Frederick, R., & Jacobson, V. (2003), RTP: A Transport Protocol for Real-Time Application, RFC 3550.
- [6] Elastix - Open Source Unified Communications Server, [www.elastix.org](http://www.elastix.org)
- [7] Apache - The Apache HTTP Server Project, [httpd.apache.org](http://httpd.apache.org)
- [8] MySQL - The World's Most Open Source Database, [www.mysql.com](http://www.mysql.com)
- [9] PHP, Hypertext Preprocessor, [www.php.net](http://www.php.net)
- [10] Linphone - Open Source Video SIP Phone, [www.linphone.org](http://www.linphone.org)

## 하은용(Ha, Eun-Yong)



- 1986년 2월 : 서울대학교 전자계산기공학과(학사)
- 1988년 2월 : 서울대학교 컴퓨터공학과(석사)
- 1997년 2월 : 서울대학교 컴퓨터공학과(박사)
- 1997년 3월 ~ 현재 : 안양대학교 컴퓨터공학과 교수

· 관심분야 : 인터넷 서비스, 오픈소스 소프트웨어, 임베디드 시스템, IoT 서비스

· E-Mail : [eyha@anyang.ac.kr](mailto:eyha@anyang.ac.kr)