

SOHO 환경을 위한 효율적인 VoIP 단말 관리 기법 설계 및 구현

진 상 우[†] · 경 계 현^{††} · 고 광 선^{†††} · 엄 영 익^{††††}

요 약

인터넷 전화는 VoIP를 이용하는 전화 방식으로서 VoIP 단말이 인터넷에 연결되어 있다면 저렴한 요금과 다양한 부가서비스가 가능한 방식이다. 하지만 인터넷 전화를 관리하기 위한 시스템들은 대규모의 VoIP 서비스 사업자나 대기업에서의 사용을 목적으로 하여 개발되어 오면서 설치 및 관리의 편의성 부분이 결여되어 있으며, 이러한 문제점들은 VoIP 단말 설치를 어렵게 하여 인터넷 전화 보급에 장애가 되어 왔다. 본 논문에서는 이러한 문제를 해결하고자 VoIP 단말들의 설정 관리를 목적으로 하는 자동 프로비저닝 시스템을 설계하고 구현한다. 이 시스템은 SOHO 환경과 같이 소규모 VoIP 환경에서 VoIP 단말들을 쉽게 설치하고 관리할 수 있도록 지원한다.

키워드 : 인터넷 전화, VoIP, Auto provisioning

Design and Implementation of an Efficient Management Scheme for VoIP Terminals in SOHO Environments

Sangwoo Jin[†] · Gye-hyeon Gyeong^{††} · Kwang Sun Ko^{†††} · Young Ik Eom^{††††}

ABSTRACT

Internet telephone uses VoIP as its communication protocol, where it provides various additional services with inexpensive fare when the terminal is connected to the Internet. The existing auto provisioning systems that manage Internet telephone terminals are targeting large size VoIP service providers or enterprises, and has few consideration on the convenience of management for small size environments. The difficult installation and management procedures of the current auto provisioning systems prevents Internet telephone system from getting popular. Easy installation and management system is needed in order to spread out Internet telephone system. In this paper, we design and implement the auto provisioning system that provides easy installation and management services in small size environments such as SOHO environments.

Key Words : Internet telephone, VoIP, Auto provisioning

1. 서 론

VoIP란 지금까지 PSTN 네트워크를 통해 이루어졌던 음성 서비스를 인터넷 프로토콜(IP)을 이용하여 음성 서비스를 지원하며, 추가적으로 여러 가지 다양한 서비스를 제공하는 기술을 말한다. 이렇게 인터넷망을 이용함으로써 인터넷 전화는 전 세계 어디서나 저렴한 가격으로 통화할 수 있는 기능을 지원하며, 이를 기반으로 기존의 전화망에서 하지 못했던 많은 서비스들을 지원한다. 대표적인 응용들로서는 웹콜센터, Instant messaging, CTI(computer Telephony

Integration), UMS(Unified Messaging System) 등을 들 수 있으며, 현재도 다양한 서비스들이 개발되고 있다. 인터넷 전화를 사용하기 위해서는 VoIP 단말의 설치와 인터넷 전화를 사용하기 위한 설정을 수행해야만 한다. 이러한 불편함을 해결하기 위해, 네트워크 관리 시스템의 기능 중에서 VoIP 단말 설정 관리 기능만 특화된 자동 프로비저닝 시스템(auto provisioning system)이 도입되었다. 하지만 자동 프로비저닝 시스템의 수요처가 VoIP 서비스 사업자나 대기업으로 한정 되어 있으며 이러한 이유로 기존의 자동 프로비저닝 시스템들은 일반적으로 VoIP 서비스 사업자나 대기업에서의 사용을 대상으로 하여 개발되어 왔다. 이러한 시스템들은 편의성이 배제되어 있기 때문에 전문적 지식을 갖춘 관리자만이 관리할 수 있으며, 이러한 이유로

[†] 정 회 원: 성균관대학교 이동통신공학과 석사과정
^{††} 준 회 원: 성균관대학교 전자전기컴퓨터공학과 석사과정
^{†††} 정 회 원: 성균관대학교 이동통신공학과 연구교수
^{††††} 중신회원: 성균관대학교 정보통신공학부 교수
논문접수: 2007년 11월 26일, 심사완료: 2008년 2월 23일

일반인들은 설치와 관리가 어려워 인터넷 전화의 대중화를 막는 원인이 되고 있다.

기존의 자동 프로비저닝 시스템들은 HTTP [1] 서버나 TFTP [2] 서버를 이용해서 VoIP 단말들의 설정 데이터를 내려 받을 수 있도록 해주는 방식과 Cisco의 CallManager [3]라는 제품과 같이 이런 설정 데이터를 관리할 수 있는 부가 기능을 가지고 있는 방식이 주로 사용되어져 왔다.

본 논문에서는 SOHO 환경과 같은 소규모 환경에서도 별도의 관리자 없이 VoIP 단말의 설치 및 관리를 자동으로 수행해 주는 IP-PBX를 이용한 자동 프로비저닝 시스템을 설계한다. SOHO용 IP-PBX에 자동 프로비저닝 시스템을 추가한 서버를 구축하며, 이를 통하여 VoIP 단말들은 설치와 관리가 자동으로 이루어질 수 있다. 이러한 시스템은 전문적인 네트워크 관리자나 인터넷 전화 관리 전문가가 없는 소규모 환경에서도 VoIP 단말의 설치와 관리를 쉽게 수행할 수 있는 장점을 지니고 있다.

2. 자동 프로비저닝 기법

자동 프로비저닝 시스템은 VoIP 단말들의 설정 관리를 목적으로 하는 시스템이다. 본 절에서는 VoIP 환경에서 사용되는 자동 프로비저닝 기법 두 가지를 소개한다.

2.1 HTTP/TFTP 서버를 이용한 자동 프로비저닝 기법

일반적으로 VoIP 단말들은 HTTP 서버나 TFTP 서버로부터 자신의 설정 데이터를 내려 받아서 사용할 수 있는 기능을 가지고 있다. HTTP 서버나 TFTP 서버의 디렉토리에 각 VoIP 단말별 설정 데이터를 파일로 만들어서 저장해 놓으면 VoIP 단말은 부팅시에 자동 프로비저닝을 통하여 자동으로 설정을 수행하게 된다.

하지만 이렇게 설정데이터를 파일로 관리하는 방식은 일반적으로 관리에 어려움이 많게 된다. 파일의 이름은 VoIP 단말의 MAC 주소를 사용하여 정해야 하며, VoIP 단말에는 HTTP 서버나 TFTP 서버의 주소를 설정해 놓거나 DHCP [4] 서버로 받을 수 있도록 구성되어 있어야만 한다.

2.2 Cisco사의 자동 프로비저닝 기법

Cisco에서는 IP-PBX기능을 수행하는 CallManager를 이용하여 자동 프로비저닝 기법을 제공하고 있다. CallManager는 음성 통화 제어와 음성 응용 기능들을 제공하는 장비로서 이 장비의 단말인 VoIP 단말에 대한 관리 기능을 제공하고 있다. 먼저 CallManager는 등록된 VoIP 단말의 설정을 암호화한 후 VoIP 단말의 MAC 주소로 이름을 지정하여 파일로 저장하여 관리한다. VoIP 단말이 설정 파일을 요구하면 CallManager는 TFTP 프로토콜을 이용하여 그 파일을 VoIP 단말에게 전달하게 된다. 또한 VoIP 단말이 재설정이 필요할 시 CallManager는 VoIP 단말에게 리셋을 시킬 수 있는 메시지를 보낼 수 있으며, 리셋 메시지를 받은 VoIP 단말은 리셋 작업을 수행한다. 하지만 VoIP

단말에 기본적인 네트워크 설정이나 서버에 대한 정보를 사용자가 직접 입력해야 하고, 각 VoIP 단말별 개별 관리가 어려운 단점이 있다.

3. SOHO 환경을 위한 자동 프로비저닝 시스템 설계

현재 IP-PBX와 VoIP 단말을 설치하고 사용함에 있어서 가장 많이 제기되고 있는 문제점들은 다음과 같다.

- 설치의 어려움 및 반복적으로 해야 하는 VoIP 단말 설정의 필요성
- 자동 프로비저닝 시스템을 사용하는 경우에도 VoIP 단말에 기본적인 네트워크에 관련된 정보와 자동 프로비저닝 시스템이 구축된 서버에 대한 정보의 설정의 필요성

위의 문제를 해결하기 위하여 본 논문에서는 관리자 또는 사용자에게 의한 각각의 VoIP 단말 설정을 하지 않고도 자동 프로비저닝 서버(자동 프로비저닝 시스템이 구축된 서버)를 통하여 자동으로 구성되게 하는 S-APS(Auto Provisioning System for SOHO)를 설계하고자 한다. S-APS는 관리자가 존재하지 않는 SOHO 환경과 같이 5~50개 정도의 VoIP 단말을 사용하는 환경에서 유용하게 사용될 수 있다. S-APS가 적용된 VoIP 환경에서는 VoIP 단말의 S-APS 서버 검색방법, S-APS로 부터의 VoIP 단말 자동 설정 방법, S-APS의 설정 내용 변경 시 VoIP 단말의 적용 방법으로 이루어진 세 가지 메커니즘이 필요하다.

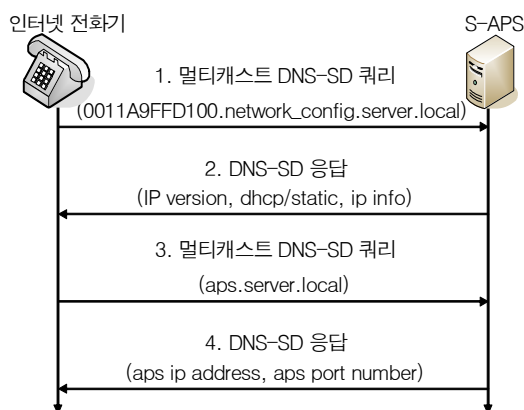
3.1 VoIP 단말의 S-APS 검색 방법

VoIP 단말이 S-APS와 연동하기 위해 필요한 기본적인 정보들의 획득을 자동화 한다. 이를 위하여 다음과 같은 특징을 지니고 있는 mDNS [5]/DNS-SD [6] 프로토콜을 적용한다.

- 멀티캐스트 방식을 사용하여 서버에 대한 정보가 없어도 자동으로 서버를 찾을 수 있음
- DNS를 이용하여 서비스를 검색하는 기능을 제공함

VoIP 단말은 mDNS/DNS-SD 프로토콜을 이용하여 (그림 1)에서 보이는 것과 같은 절차에 따라 S-APS 서버를 검색하고 S-APS 서버로부터 정보를 받아 네트워크를 설정할 수 있다.

(그림 1)에서 보이는 바와 같이, 먼저 VoIP 단말은 멀티캐스트 DNS-SD 쿼리를 이용하여 S-APS 서버에 네트워크 설정 정보를 요청해야 한다. 통신을 위해서는 IP 주소가 필요하며, VoIP 단말은 주소 생성을 위해 IPv6 프로토콜의 비상대 주소 자동설정(Stateless address autoconfiguration)기법을 사용한다 [7]. 이 기법은 추가적인 장치나 사용자에게 한 설정이 없이도 단말 스스로 주소를 생성하여 통신을 수



(그림 1) mDNS/DNS-SD를 이용한 S-APS 검색 및 네트워크 설정

행할 수 있는 주소 생성 기법이다. 이렇게 생성된 주소는 Link-local 주소로서 동일한 링크에서만 통신이 가능하므로, VoIP 단말을 사용하기 위해서는 다음 단계에서 S-APS 서버로부터 전송되는 주소 정보를 이용하여 IP 주소를 새롭게 설정해야 한다.

- <VoIP 단말의 MAC 주소>.network_config.server.local

멀티캐스트 DNS-SD 쿼리는 기본적으로 DNS 쿼리 형식을 취해야 하며, 위와 같은 형식으로 도메인 이름에 VoIP 단말의 MAC 주소를 포함하도록 한다. S-APS 서버에서는 이러한 쿼리를 받으면 MAC 주소를 이용하여 해당 VoIP 단말의 설정 데이터를 검색할 수 있다. S-APS 서버는 VoIP 단말의 네트워크 설정을 위하여 다음과 같은 정보를 포함하여 전송한다.

- IP 프로토콜 버전
- IP 설정 방식(DHCP/Static)의 선택
- IP 설정 방식이 static일 경우 static IP 주소의 정보

이 정보들은 DNS TXT Resource Record 방식 [8]을 사용하여 전송되며, VoIP 단말은 S-APS 서버로부터 받은 네트워크 설정 정보를 이용하여 네트워크 설정을 수행한다. VoIP 단말은 다시 멀티캐스트 DNS-SD 쿼리를 이용하여 S-APS 서버의 주소와 포트 번호를 요청한다. 이 경우의 도메인 이름은 서버의 주소와 포트 번호를 얻기 위한 방법인 DNS SRV [9]에 따라서 _aps._tcp.server.local와 같은 형식으로 도메인을 구성한다.

이러한 방식은 VoIP 단말이 동일 링크에서만 사용할 수 있는 Link-local 주소를 사용하기 때문에 S-APS 서버가 동일한 링크에 있어야만 사용 가능하다는 제약이 있다. 이러한 경우에는 S-APS Relay를 이용하여 해결할 수 있다. S-APS Relay는 VoIP 단말로부터의 mDNS/DNS-SD 요청을 수신하고, 이를 유니캐스트 DNS-SD로 변환하여 S-APS

서버로 전달하는 역할을 수행한다. 그리고 S-APS 서버로부터 응답이 오면, 이를 다시 VoIP 단말로 전달한다. 이러한 방식은 VoIP 단말들이 S-APS 서버와 동일 링크에 존재하는 것과 같은 환경을 만들어준다.

3.2 S-APS로 부터의 VoIP 단말 자동 설정 방법

다음으로 VoIP 단말은 S-APS에서 정보를 받아 VoIP 단말에 VoIP를 위한 설정을 자동으로 수행한다. 이 과정에서 S-APS에서는 VoIP 단말의 설정 데이터들을 관리하기 위하여 프로파일 항목을 사용한다. 프로파일은 설정 내용 중 동일한 부분을 하나의 그룹으로 묶어 관리할 데이터의 양을 최소화 하는데 목적을 갖으며, 아래와 같은 내용으로 프로파일이 구성된다.

- VoIP 패킷 처리를 위한 IP-PBX의 주소
- SIP 프로토콜에 대한 설정 값

설정 정보를 요청하고 받기 위해서 VoIP 단말은 네트워크 설정단계에서 얻은 S-APS의 IP 주소와 포트번호를 이용하여 S-APS에게 연결한다. 이때 VoIP 단말은 HTTPS [10] 프로토콜을 사용하여 VoIP 단말의 설정 데이터들을 다운로드 한다. VoIP 단말은 HTTPS의 URL로서 MAC 주소를 사용하며, S-APS는 MAC 주소를 이용하여 설정 정보를 검색하고 이 정보를 VoIP 단말에 전송한다.

3.3 S-APS의 설정 내용 변경 시 VoIP 단말의 적용 방법

마지막으로 S-APS에 등록된 설정 내용의 변경 시 변경 내용을 VoIP 단말에 자동으로 적용할 수 있도록 한다. S-APS에 VoIP 단말이 등록된 후 사용자가 VoIP 단말을 사용 중에 있을 때, S-APS에서 VoIP 단말의 설정을 수정하는 경우의 처리 방법은 S-APS에서 VoIP 단말에 변경된 설정이 있다는 이벤트를 전송하는 방법을 사용한다. 이를 위하여 S-APS는 다음과 같은 내용을 수행 할 수 있도록 설계되어야 한다.

- VoIP 단말 등록정보를 이용하여 VoIP 단말들의 주소를 추적하고 있어야 함
- VoIP 단말의 설정 값이 수정되는 경우에는 VoIP 단말로 이벤트를 전송해야 함

이벤트를 전송하기 위해서 IP-PBX에서 VoIP를 위하여 사용되고 있는 SIP의 Notify Request를 이용한다. VoIP 단말이 설정 변경 Notify Request를 수신하게 되면, S-APS에 연결하여 3.2장의 방법에 따라서 설정 데이터들을 다운로드 하여 재 적용하게 된다. 그리고 설정 변경뿐 아니라 VoIP 단말의 소프트웨어가 업그레이드 된 경우에도 SIP [11] 프로토콜의 Notify Request를 사용하여 VoIP 단말에게 소프트웨어가 업그레이드되었음을 알려주고, 업그레이드를 수행할 시간을 예약하여, VoIP 단말이 추후에 소프트웨어를 업그레이드 할 수 있도록 한다.

4. S-APS의 구현

본 장에서는 앞서 설명한 앞서 설명한 세 가지 메커니즘의 동작을 위하여 VoIP 단말, S-APS relay, S-APS 서버에 필요한 부분을 구성하고, 각 구성별로 자세한 설명을 한다.

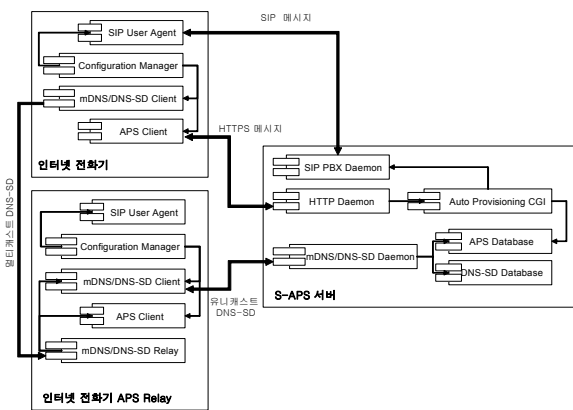
4.1 장비별 구조도

(그림 2)에서 보이는 바와 같이 S-APS를 이용하는 인터넷 전화 환경은 APS 서버, APS Relay, 그리고 VoIP 단말로 구성되며, 각 부분의 역할은 <표 1>에서 자세히 설명하고 있다.

(그림 2)에서 보이는 바와 같이, VoIP 단말에는 설치와 설정에 관련된 구성요소들이 추가로 구현되었다. 구성요소들은 S-APS 서버를 찾고 네트워크 설정을 위한 mDNS/DNS-SD Client, S-APS 서버로부터 설정 정보를 전달받는 APS Client, 받은 설정 정보를 이용하여 VoIP 단말에 설정작업을 수행하는 Configuration Manager, SIP User Agent로 구성된다.

S-APS 서버는 IP-PBX내부에 추가되므로 IP-PBX가 S-APS 서버 역할을 동시에 수행하게 된다. 앞 절에서 언급한 기능들을 구현하기 위하여 SIP PBX Daemon, HTTP Daemon, Auto provisioning Daemon, mDNS/DNS-SD Daemon, S-APS Database, DNS-SD Database들이 서로 상관관계를 갖고 동작하게 된다.

S-APS Relay의 기본 구조는 VoIP 단말의 구조와 동일하다. 단지 S-APS Relay를 추가하여 같은 네트워크에 있는 다른 VoIP 단말들이 S-APS와 같은 네트워크에 있는 것과 동일한 효과를 나타낸다. 이를 위해서 mDNS/DNS-SD Relay는 mDNS 서버로서 동작을 하도록 구성된다. mDNS/DNS-SD Relay는 VoIP 단말들로부터 멀티캐스트 DNS-SD 쿼리를 받게 되면 이를 유니캐스트 DNS-SD 쿼리로 변환하여 S-APS 서버로 전달한다. 그리고 S-APS 서버로부터 DNS-SD 응답을 받아서 멀티캐스트 DNS-SD 쿼리를 보낸 VoIP 단말로 다시 전달해주는 역할을 한다. 이런 S-APS 서버와의 통신을 위해서 mDNS/DNS-SD Client에 유니캐스트 DNS-SD 쿼리를 전송할 수 있는 기능이 추가 된다.



(그림 2) S-APS 구성요소 구조도

<표 1> VoIP 단말과 S-APS 관련 구성요소

구성요소	역할	
VoIP 단말	SIP 사용자 에이전트 (SIP User Agent)	<ul style="list-style-type: none"> SIP 프로토콜을 처리함 SIP 호에 대한 처리와 SIP Notify 요청으로 S-APS 서버로부터 VoIP 단말에 전달되는 S-APS 관련 통지를 구성 요소들에게 전달하는 역할을 수행함
	설정 매니저 (Configuration Manager)	<ul style="list-style-type: none"> VoIP 단말에 저장되어 있거나 S-APS 서버로부터 받은 설정 데이터를 이용하여, VoIP 단말의 각 S/W 구성요소 및 H/W의 초기화 역할을 수행함 각 단계 별로 필요한 데이터를 구하기 위하여 mDNS/DNS-SD Client와 APS Client를 사용함
	mDNS/DNS-SD 클라이언트 (mDNS/DNS-SD Client)	<ul style="list-style-type: none"> 멀티캐스트 DNS-SD 쿼리를 전송하고, DNS-SD 응답을 수신함 네트워크 설정에 대한 데이터와 S-APS 서버의 주소와 포트번호를 구함
	S-APS 클라이언트 (S-APS Client)	<ul style="list-style-type: none"> HTTPS 프로토콜을 사용하여 S-APS 서버로부터 VoIP 단말의 설정 데이터를 내려 받는 역할을 수행함 MAC 주소를 매개변수로 하는 HTTP Get 요청이 사용됨 S-APS 서버로부터 내려 받는 설정 데이터를 파싱하여 VoIP 단말에서 사용할 수 있는 형식으로 변환함
S-APS 서버	SIP PBX 데몬 (SIP PBX Daemon)	<ul style="list-style-type: none"> SIP 서버 역할을 하는 프로세스로서 VoIP 단말들의 SIP 요청을 처리함 VoIP 단말로의 호가 발생하는 경우에는 호를 VoIP 단말에게 전달하는 역할을 하는 IP-PBX의 역할을 수행함 VoIP 단말들의 SIP 등록 요청을 처리 하며, 이를 통하여 VoIP 단말들의 주소를 실시간으로 관리함 다른 프로세스들에서 VoIP 단말로 통지를 하고자 하면, SIP의 Notify 요청을 사용하여 이를 수행함 사용자의 요청으로 데이터의 수정이 발생하게 되는 경우, SIP PBX Daemon을 통하여 VoIP 단말에게 데이터 변경이 발생하였음을 통지
	HTTP 데몬 (HTTP Daemon)	<ul style="list-style-type: none"> 일반적인 HTTP 프로토콜을 처리하는 프로세스 각 URL별로 파일전송을 수행함 자동 프로비저닝 요청이 들어오면 Auto provisioning CGI [12]로 전달
	자동 프로비저닝 CGI (Auto provisioning CGI)	<ul style="list-style-type: none"> HTTP Daemon을 통하여 들어오는 VoIP 단말들의 자동 프로비저닝 요청을 받음 VoIP 단말의 MAC 주소를 키로 사용하여, S-APS Database에 저장되어 있는 VoIP 단말을 위한 설정 데이터를 VoIP 단말에게 전송함
	mDNS/DNS-SD 데몬 (mDNS/DNS-SD Daemon)	<ul style="list-style-type: none"> mDNS/DNS-SD의 서버 역할을 하는 프로세스로서 VoIP 단말들의 멀티캐스트 DNS-SD 쿼리와 S-APS Relay의 유니캐스트 DNS-SD 쿼리를 처리. VoIP 단말 네트워크 설정 쿼리와 서비스 서버의 주소에 대한 쿼리를 처리 쿼리에 포함된 MAC 주소를 이용하여 DNS-SD 데이터베이스를 검색
	S-APS 데이터베이스 (S-APS Database)	<ul style="list-style-type: none"> S-APS에 관계된 데이터들의 관리 S-APS 데이터의 추가, 수정 그리고 검색을 담당 S-APS관련 프로세스들의 APS 데이터에 대한 요청을 처리
DNS-SD 데이터베이스 (DNS-SD Database)	<ul style="list-style-type: none"> DNS-SD가 제공할 서비스 서버에 대한 정보에 대한 검색과 관리를 담당 	

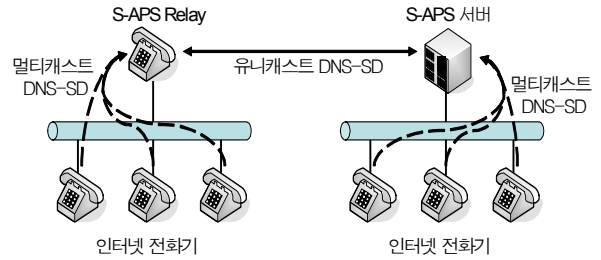
4.2 동작 시나리오

(그림 3)은 S-APS가 적용된 환경에서 VoIP 단말들이 관리되는 한 예를 보이고 있다. (그림 3)에서의 환경은 서로 다른 두 개의 네트워크가 존재하며 한 네트워크에는 S-APS 서버가 존재하며 다른 네트워크에는 존재하지 않는다. 하지만 S-APS 서버가 존재하지 않는 네트워크에는 릴레이가 존재하여 S-APS 서버로의 연결을 지원할 수 있다.

(그림 3)에서 보이는 바와 같은 환경에서, S-APS 서버가 존재하는 네트워크에 VoIP 단말이 새로 설치되면 다음과 같은 과정을 거치게 된다.

- VoIP 단말은 멀티캐스트 DNS-SD 쿼리를 보내어 S-APS 서버에 정보를 요청한다.
- S-APS 서버는 VoIP 단말의 쿼리를 받아 VoIP 단말에 필요한 네트워크 설정 정보와 자동 프로비저닝 서비스를 제공하는 서버의 주소와 포트번호를 검색한다. 검색된 정보는 쿼리를 보낸 VoIP 단말에 전송된다.
- VoIP 단말은 S-APS 서버로부터 받은 정보를 이용하여 VoIP 단말의 네트워크를 설정하고 서버의 주소와 포트번호를 이용하여 S-APS 서버에 접속한다.
- S-APS 서버는 VoIP 단말의 프로비저닝 요청을 인식하고 VoIP 단말을 설정하기 위한 설정 파일들을 HTTPS 프로토콜을 이용하여 VoIP 단말에 전송한다.

VoIP 단말이 설치되는 네트워크에 S-APS 서버가 없다 하더라도 S-APS Relay가 설치되어 있기 때문에 VoIP 단말의 설치는 정상적으로 완료될 수 있다. S-APS Relay는 멀티 캐스트 DNS-SD 쿼리를 유니캐스트 DNS-SD 쿼리로 변환하여 Relay에 지정된 S-APS 서버로 전송하는 역할을

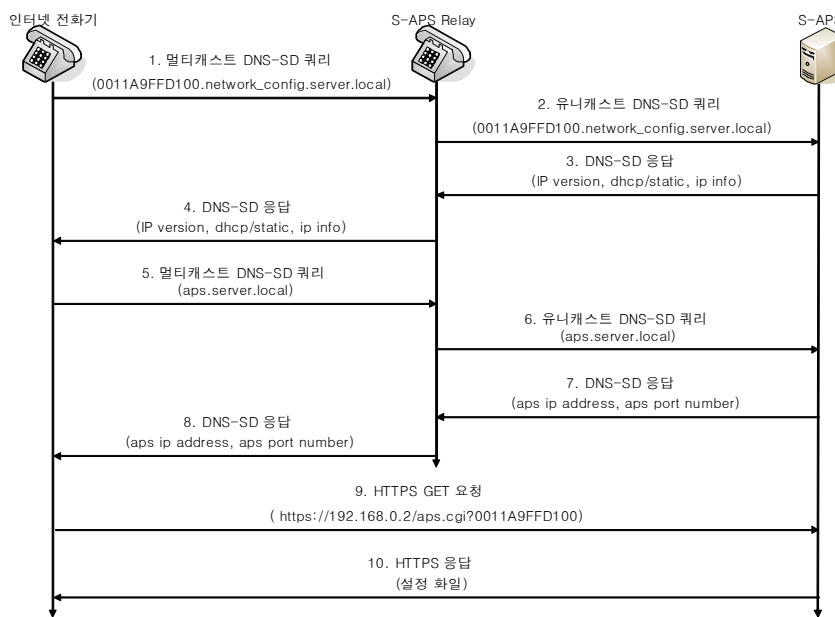


(그림 3) S-APS를 이용한 인터넷 전화 구성도

수행하며 VoIP 단말과 S-APS 서버간의 연결을 중계한다. 이러한 과정을 통하여 VoIP 단말은 설정 파일들을 내려 받은 후 VoIP 단말을 설정하고 이후 사용자는 VoIP 단말을 이용하여 통화를 할 수 있다. (그림 4)에서 이런 동작을 수행할 때의 네트워크 요청과 응답을 보여주고 있다.

5. 구현 및 비교

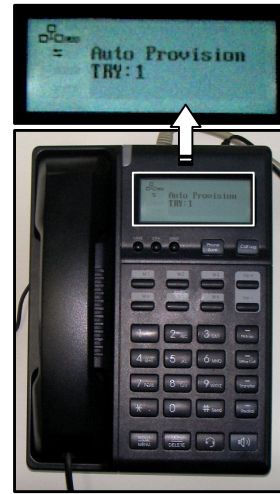
본 논문에 S-APS의 구현은 임베디드 환경으로의 이식성을 고려하여 C/C++ 언어로 구현되었다. S-APS를 위하여 사용되는 네트워크 프로토콜들이 모두 TCP/IP만을 사용하고 있기에 모든 네트워크 관련 기능들은 표준 socket 함수들만을 사용하여 구현하였다. 이는 socket 인터페이스를 지원하는 여러 운영체제로의 이식성을 위한 것이다. S-APS 서버 부분에 있어서 SIP PBX Daemon과 HTTP Daemon은 기존의 IP-PBX에 존재하는 코드를 재사용 했으며, S-APS 관련 부분만 새롭게 구현 하였다. Auto provisioning CGI는 Get 방식의 CGI로서 C++를 사용하여 구현하였다. 또한



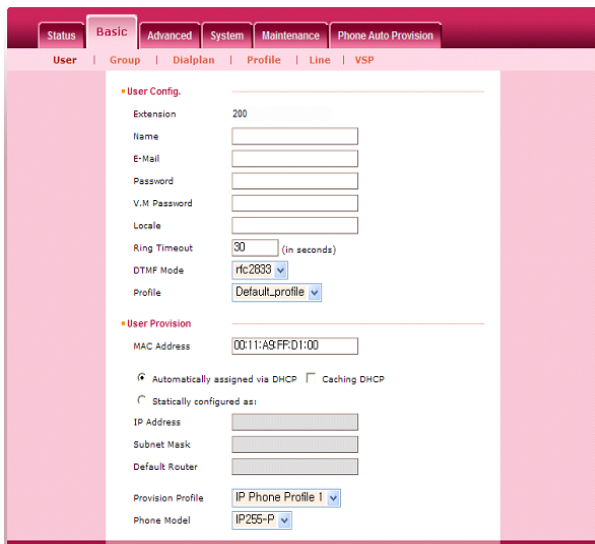
(그림 4) S-APS를 이용한 자동 프로비저닝 네트워크 흐름도

Multicast DNS를 처리하기 위한 mDNS/DNS-SD Daemon, Relay와 Client는 Open Source인 Avahi Project [13]의 라이브러리를 사용하여 구현하였다. Avahi는 Linux를 위해 mDNS/DNS-SD를 구현한 것이다. 이것은 Linux에 종속적이거나 임베디드 환경에 맞지 않는 부분들이 있기에 S-APS를 위하여 일부 라이브러리만을 추출하여 구현하는데 사용하였다. Avahi에서 추출한 부분은 표준 socket 인터페이스의 Multicast 옵션을 사용하여 구현된 Multicast socket 인터페이스 부분과 DNS 패킷을 처리하는 부분이다. mDNS/DNS-SD Client부분과 Relay는 VxWorks라는 상용 임베디드 운영체제를 사용하는 VoIP 단말에서 구현하였다.

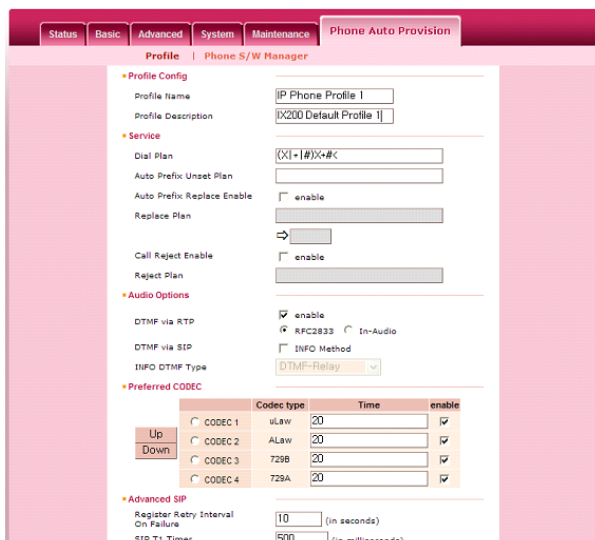
(그림 4)에서는 S-APS에서 단말 프로파일을 추가하고 사용자를 추가하는 유저 인터페이스 화면을 보이고 있다.



(그림 6) VoIP 단말 실행 모습



(2) VoIP 단말 추가 유저 인터페이스 화면



(1) VoIP 단말 프로파일 추가 유저 인터페이스 화면

(그림 5) S-APS 유저 인터페이스 화면



(그림 7) S-APS의 설정 관련 로그

VoIP 단말의 설정 데이터들에는 많은 항목들이 존재하지만, (그림 4)의 (1)에서 보이는 바와 같이 전화기별로 동일한 내용은 프로파일로 생성하여 각 단말별 설정에서 발생할 수 있는 반복 작업을 줄일 수 있다. (그림 5)의 (2)는 각 개별 VoIP 단말들을 설정할 수 있으며 프로파일 내용을 제외한 변경되기 쉬운 정보들만으로 구성되어 있다. 그 외의 정보들은 Provision Profile 항목을 통해서 설정되어지게 된다. 그리고 Phone Model항목은 VoIP 단말의 하드웨어 정보를 나타내는 것으로 VoIP 단말의 소프트웨어 자동 업그레이드를 위하여 사용된다. 이렇게 S-APS에 VoIP 단말에 대한 설정을 하면, VoIP 단말 설치 시 자동으로 설정을 수행할 수 있다. (그림 6과 7)은 자동으로 설정되는 모습을 보이고 있다.

<표 2> APS들간의 편의성 관련 비교

항목	기존 기법		제안 기법
	HTTP/TFTP APS	Cisco CallManager	S-APS
VoIP 단말에서의 네트워크 설정	필요	필요	불필요
VoIP 단말에서의 APS 설정	불필요(*)	불필요(*)	불필요
APS에서의 VoIP 단말 추가 방법	복사 후 수정	복사 후 수정	프로파일 선택
APS에서의 그룹별 수정 기능	없음	없음	있음
APS에서의 수정 즉시 적용 기능	없음	있음(**)	있음
VoIP 단말 S/W 업그레이드	단말 개별 지정	단말 개별지정	H/W종류별 지정

* 네트워크 주소 설정을 위해 DHCP 사용 및 DHCP 서버에 APS에 대한 등록 필요함.

** VoIP 단말을 개별적으로 리셋 시켜서 Auto Provisioning을 다시 시도하도록 하는 방식.

<표 3> HTTP APS와 S-APS의 VoIP 단말 설치시의 VoIP 단말과 APS에서의 설정 항목 수 비교

VoIP 단말 수	HTTP APS		S-APS	
	VoIP 단말	APS	VoIP 단말	APS
10	30	30	0	30
20	60	60	0	60
30	90	90	0	90
40	120	120	0	120
50	150	150	0	150

(그림 6)은 VoIP 단말 설치 시 단말에 표시되는 화면을 보이고 있다. VoIP 단말이 설치가 되면 부팅 과정을 거치면서 자동 프로비저닝 단계를 수행하게 된다. (그림 7)은 VoIP 단말의 설치 및 설정에 관한 정보를 배포하는 S-APS에서 VoIP 단말과 주고받는 정보들에 대한 로그를 보이고 있다. VoIP 단말의 자동 프로비저닝 단계가 종료되면 인터넷 전화를 사용할 수 있다. 이러한 방식으로 동작하는 S-APS를 HTTP/TFTP 서버를 이용한 APS 방식, Cisco CallManager 방식과 비교하여 보면 <표 2>와 같은 결과를 볼 수 있다.

각 항목을 자세히 살펴보면 먼저 S-APS는 VoIP 단말에서의 설정이 불필요하다. 기존의 방식들은 VoIP 단말에 네트워크 설정이나 APS 서버에 대한 정보를 설정해야만 사용이 가능하였다. 하지만 S-APS는 VoIP 단말과 같은 단말에서의 설정이 필요 없도록 하여 관리자의 관리 없이 VoIP 단말의 설치 및 관리를 자동화 하고 있다.

다음으로 S-APS는 APS 서버에서의 설정을 최소화 하고 있다. HTTP/TFTP APS와 Cisco의 CallManager에서 새로운 VoIP 단말을 등록하기 위해서는 등록되어 있는 VoIP 단말의 설정을 복사한 후에 필요한 부분을 수정하는 방식을 사용한다. 이 방식은 복사 대상 선정의 오류와 같은 사용자의 실수가 발생하기 쉽다는 단점이 있고 추후 관리에 있어서는 개별적인 수정을 필요로 하게 된다. S-APS에서는 프로파일을 선택하는 설정 방법을 통하여 각 VoIP 단말의 설정에 필요한

입력 항목의 반복적인 입력을 최소화 하고 있다.

다음으로 변경 사항의 적용 방법을 자동화 하고 있다. APS에서 VoIP 단말의 설정을 변경하는 경우, HTTP/TFTP APS 같은 경우에는 VoIP 단말이 주기적으로 APS에 접속하여 변경 여부를 검사하는 방식을 사용하고 있다. 또한 CallManager는 변경 사항을 적용하고자 할 때 사용자가 CallManager를 통하여 VoIP 단말을 리셋 시키는 방식으로 적용을 시키고 있다. 이 방식은 VoIP 단말을 개별적으로 리셋 시켜야 하는 단점이 있다. 본 논문의 S-APS에서는 변경 사항이 생기면, 이를 VoIP 단말에게 알려주어서 VoIP 단말이 변경 사항을 즉시 반영할 수 있도록 하고 있다.

마지막으로 VoIP 단말 소프트웨어 업그레이드 시 하드웨어별 그룹 관리가 가능하다. 일반적으로 소프트웨어 업그레이드는 VoIP 단말 전체를 대상으로 하게 된다. 본 논문의 S-APS에서는 VoIP 단말을 하드웨어별로 그룹 지어서 소프트웨어 업그레이드 할 수 있기 때문에 단말별 개별 관리에 비하여 VoIP 단말 전체를 업그레이드 하는데 필요한 입력 사항을 대폭 줄여줄 수 있다.

다음은 HTTP APS와 S-APS를 사용하여 VoIP 단말의 설치 시에 사용자가 VoIP 단말과 APS에서 수행해야 하는 최소 설정 항목의 개수를 설치해야할 VoIP 단말의 수를 증가시켜가며 비교를 하였다.

<표 2>에서 보여 지는 것과 같이 S-APS를 사용하는 경

우에 있어서는 VoIP 단말에서의 설정이 필요 없는 관계로 설치 시에 각 단말의 설치가 매우 쉬워 진다는 장점을 제공할 수 있는 것이다.

6. 결 론

본 논문에서는 VoIP 단말의 관리를 목적으로 하는 S-APS의 설계와 구현에 대하여 설명하였다. 본 논문의 S-APS는 SOHO 환경과 같이 적은 인원으로 운영되고, 네트워크 전문가가 없는 환경을 대상으로 하는 경우에 최적화된 서비스 제공을 목적으로 하였다. S-APS는 사용자의 편의성에 중점을 두고 설계를 하였고 기존의 APS들에 비하여 편의성 부분을 크게 개선하였다. 먼저 설치 과정에서 반복적으로 일어날 수밖에 없는 VoIP 단말들의 설정에 관련된 작업을 프로파일을 이용함으로써 관리 작업을 단순화 하였으며, S-APS 검색 방법을 자동화 하여 기존의 VoIP 단말들의 설치 과정에서 필수적으로 발생하는 관리자에 의한 초기 설정 작업을 자동화 하였다. 그리고 APS에서 VoIP 단말의 설정 변경을 그룹으로 적용할 수 있게 하여 사용 중에 발생할 수 있는 설정 변경 작업을 단순화 하였다. S-APS는 IP-PBX와 VoIP 단말 도입에 방해가 되어 왔던 설치 및 관리의 복잡함을 개선하여 인터넷 전화의 보급에 도움이 될 것이다.

참 고 문 헌

- [1] R. Fielding, R. Fielding, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee, RFC 2616, Hypertext Transfer Protocol -- HTTP/1.1., Jun. 1999.
- [2] K. Sollins, RFC 1350, The TFTP Protocol (Revision 2), Jul. 1992.
- [3] Cisco, Cisco Unified CallManager Administration Guide, Release 5.0, Jun. 2006.
- [4] R. Droms, RFC 2131, Dynamic Host Configuration Protocol, Mar. 1997.
- [5] S. Cheshire and M. Krochmal, Internet-Draft, draft-cheshire-dnsext-multicast-dns-06, Multicast DNS, Aug. 2006.
- [6] S. Cheshire and M. Krochmal, draft-cheshire-dnsext-dns-sd-04, DNS-Based Service Discovery, Aug. 2006.
- [7] S. Thomson, T. Narten, and T. Jinmei, RFC 4862, IPv6 Stateless Address Autoconfiguration, Sep. 2007.
- [8] P. Mockapetris, RFC 1035, Domain names - implementation and specification, Nov. 1987.
- [9] A. Gulbrandsen, P. Vixie, and L. Esibov, RFC 2782, A DNS RR for specifying the location of services (DNS SRV), Feb. 2000.
- [10] E. Rescorla, RFC 2818, HTTP Over TLS, May. 2000.
- [11] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, RFC 3261, SIP: Session Initiation Protocol, Jun. 2002.

[12] D. Robinson, K. Coar, RFC 3875, The Common Gateway Interface (CGI) Version 1.1, Oct. 2004.

[13] Avahi Project, <http://avahi.org>



진 상 우

e-mail : swjinjin@paran.com

1998년 성균관대학교 기계공학과 (학사)

1999년~2001년 (주)성운엔지니어링 부설 연구소 주임연구원

2001년~2003년 (주)텔레트론 부설연구소 선임연구원

2003년~현재 (주)모임스톤 부설연구소 책임연구원

2007년~현재 성균관대학교 이동통신공학과 석사과정

관심분야: VoIP, 분산 컴퓨팅, 운영체제, 내장형 시스템 등



경 계 현

e-mail : gyeheon@ece.skku.ac.kr

2006년 호서대학교 컴퓨터공학과 (학사)

2006년~현재 성균관대학교 전자전기 컴퓨터공학과 석사과정

관심분야: 이동 컴퓨팅, 이동 에이전트,

시스템 보안, 분산 컴퓨팅, 운영체제, 내장형 시스템 등



고 광 선

e-mail : rilla91@ece.skku.ac.kr

1998년 성균관대학교 정보공학과 (학사)

2004년 성균관대학교

전기전자및컴퓨터공학부(공학석사)

2007년 성균관대학교

전자전기컴퓨터공학과(공학박사)

2007년~현재 성균관대학교 이동통신공학과 연구교수

관심분야: 정보보호, 리눅스, 네트워크 등



엄 영 익

e-mail : yeom@ece.skku.ac.kr

1983년 서울대학교 계산통계학과 (학사)

1985년 서울대학교 전산학과 (이학석사)

1991년 서울대학교 전산학과 (이학박사)

2000년~2001년 Dept. of Info. and

Comm. Science at UCI 방문교수

2005년 한국정보처리학회 학회지 편집위원장

1993년~현재 성균관대학교 정보통신공학부 교수

관심분야: 분산 컴퓨팅, 이동 컴퓨팅, 이동 에이전트, 시스템 보안, 운영체제, 내장형 시스템 등