# Embedded Linux 기반의 UPnP를 사용한 홈-네트워크 서버 구현

論 文
53D-9-4

# Implementation of Home-Network Server using UPnP based on the Embedded Linux

鄭 震 珪* · 陳 善 一* · 李 熙 靜* · 黃 仁 永** · 洪 錫 敎†
(Jin-Kyu Jung · Sun-Il Jin · Hee-Jung Lee · Yin-Young Hwang · Suk-Kyo Hong)

**Abstract** - Middleware enables different networking devices and protocols to inter-operate in ubiquitous home network environments. The UPnP(Universal Plug and Play) middleware, which runs on a PC and is based on the IPv4 protocol, has attracted much interest in the field of home network research since it has versatility. The UPnP, however, cannot be easily accessed via the public Internet since the UPnP devices that provide services and the Control Points that control the devices are configured with non-routable local private or Auto IP networks. The critical question is how to access UPnP network via the public Internet. The purpose of this paper is to deal with the non-routability problem in local private and Auto IP networks by improving the conventional Control Point used in UPnP middleware-based home networks. For this purpose, this paper proposes an improved Control Point for accessing and controlling the home network from remote sites via the public Internet, by adding a web server to the conventional Control Point. The improved Control Point is implemented in an embedded GNU/Linux system running on an ARM9 platform. Also, this paper implements the security of the home network system based on the UPnP (Universal Plug and Play), adding VPN (Virtual Private Network) router that uses the IPsec to the home network system which is consisted of the ARM9 and the Embedded Linux.

**Key Words** : UPnP, VPN, Middleware, Control Point, Embedded Linux

## 1. 서 론

차세대 기술이라 일컬어지는 유비쿼터스 컴퓨팅 환경에 발맞추어, 공장 자동화, 사무 자동화에 이어, 가정에서 사용되는 가전기기들이 지능화 되어감에 따라, Digital 혹은 Intelligent Appliance라 불리우는 인터넷 정보가전기기들의 상호연동과 제어/관리 장치의 중요성이 새로운 Issue로 떠오르고 있다. 인터넷 정보가전기기의 통신 매체로는 전화선, 전력선, RF(Radio Frequency), Blue Tooth, Wireless LAN, IEEE1394(Firewire), Ethernet 등이 사용되지만, 각각의 통신매체마다 프로토콜이 다름으로 인해서, 이들 정보가전기기들의 상호연동을 기대하기는 힘든 상황이다. 이러한 이종 프로토콜간의 상호연동을 가능케 하는 것이 바로 미들웨어이며, 대표적인 미들웨어로는 멀티미디어를 위한 HAVi, JAVA의 RPC 기능을 이용한 JINI, 네트워크로 연결된 Plug and Play 개념의 UPnP(Universal Plug and Play)등이 있는데, 그 중에서도 PC와 IPv4를 기반으로 동작하는 UPnP 미들웨어는 강력한 bridging 기능을 지니고 있어서 실제적용에 있어 많은 편의를 제공하고 있다[1]. 그러나 유비쿼터스 컴퓨팅의 목적이 그러하듯 핸드폰, PDA, 인터넷 등을 이용하여 언제 어디서나 홈 네트워크 시스템에 접속하여 원하는 기기들을 접근 및 제어 할 수 있어야 하지만, UPnP 기반의 홈 네트워크 자체로는 외부로부터의 접근을 허용하기가 쉽지 않다

이의 해결을 위하여 본 논문에서는 별도의 추가적인 장비 없이도 기존의 UPnP 홈 네트워크 시스템의 외부에서 접근 및 제어가 가능하도록 Control Point 구조의 개선점을 제안하고, Embedded Linux를 이용하여 이를 구현함으로써, 그 가능성을 검증하고자 한다. 또한 공공의 인터넷 망을 이용하기 때문에 발생하는 정보의 보안을 위하여 VPN을 이용하여 공공 인터넷을 하나의 사설망처럼 사용할 수 있는 가능성을 보여주고자 한다.

* 學生會員 : 亞洲大 情報通信大 大學院 · 碩士課程
** 正 會 員 : LG電子 홈넷사업팀 시스템개발Gr. · 責任硏究員
† 교신저자, 正會員 : 亞洲大 情報通信大 敎授 · 工博
E-mail : skhong@ajou.ac.kr
接受日字 : 2004年 5月 4日
最終完了 : 2004年 6月 15日

## 2. UPnP Middleware

그림1 에서는 다양한 통신매체로 연결된 전형적인 홈 네트워크의 구성을 보여주고 있으며, 이러한 구성이 가능하기 위해서는 다양한 프로토콜의 상호연동을 가능케 하는 middleware가 필수요소이다. 그리고 표1에서는 대표적인 middleware 세 가지를 비교하고 있는데, HAVi는 1998년 소니, 톰슨 ,필립스, 도시바 등 8개 가전업체가 참여하여 만들어 IEEE 1394를 사용하여 통신하고 있으며, Jini는 1998년 썬 마이크로 시스템즈사에서 발표한 분산 환경의 홈 네트워

표 1 Middleware의 비교
Table 1 Comparison of Middleware

| | HAVi | JINI | UPnP |
|---|---|---|---|
| Method | Home Audio/Video Interoperability | Java Intelligent Network Infrastructure | Universal Plug & Play |
| Medium | IEEE1394 | IP based Java based | IP based (Bridging) |
| Target | AV Device | Intelligent Appliance | Intelligent Appliance |
| Company | SONY, PHILIPS | SUN | MS, Intel |

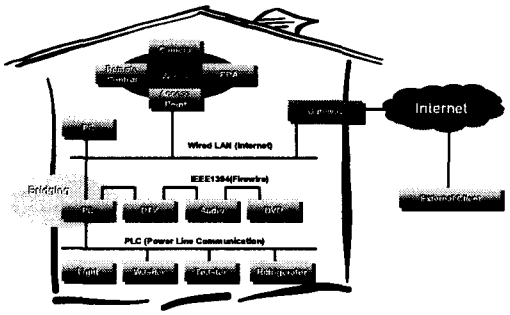


그림 1 홈 네트워크의 구성
Fig. 1 Home Network Topology

크 공유 플랫폼으로 JVM(Java Virtual Machine)기반 위에서 사용된다 그리고 UPnP는 1999년 MS, 인텔, Compaq등 150여개 업체가 PC 중심의 가전기기 제어 S/W표준으로 정의하고 있다 본 논문에서는 IP를 기반으로 Plug and Play의 개념을 갖고 기존의 TCP/IP, HTTP등 많은 표준 프로토콜을 사용할 수 있고 개방형 표준을 정의하고 있는 UPnP 미들웨어를 사용하였다.

### 2.1 UPnP의 구성

UPnP는 Device, Service, Control Point로 구성되며 각 기능은 다음과 같다.

1) Device

UPnP Device는 Service 및 embedded device 들을 내포하고 있다. 예를 들어 VCR는 테이프 전송 서비스, 튜너 서비스 및 시계 서비스로 구성되어 있을 수 있고, TV/VCR 콤보 장치는 서비스 뿐만 아니라 다른 내장형 장치들로 구성되어 있을 것이다. UPnP Device가 포함 될 수 있다.

2) Service

Service는 UPnP 네트워크의 소규모 제어 단위이다 Service는 동작을 나타내고 상태변수를 통하여 자신의상태를 모델링한다. UPnP Device가 제공하는 Service는 상태 테이블, 제어 서버 및 이벤트 서버로 구성된다. 상태 테이블은 상태변수를 활용하여 Service 상태를 업데이트 한다. 제어 서버는 동작요청을 수신하여 실행하고 있는 상태 테이블을 업데이트 하며, 그 결과를 반환한다. 이벤트 서버는 Service의 상태가 변경 될 때 마다 이벤트를 관계되는 가입자들에게 항상 알려준다.

3) Control Point

UPnP 네트워크의 Control Point는 다른 Device를 검색하여 제어하는 능력을 가진 컨트롤러이다. Device를 검색한 후에 Device Description 문서를 검색하여 Service 목록을 확보하고 관련된 Service의 SCPD 문서를 검색하여 확보하며 Service 제어 활동을 실행한다

표 2 UPnP Middleware 네트워크 주소 지정방식
Table 2 UPnP Middleware Network Addressing

| Private Network | Link Local |
|---|---|
| • DHCP<br>• 192.168.0.0 / 16<br>• **Non-Routable** | • Auto IP<br>• 169.254.0.0 / 16<br>• **Non-Routable** |

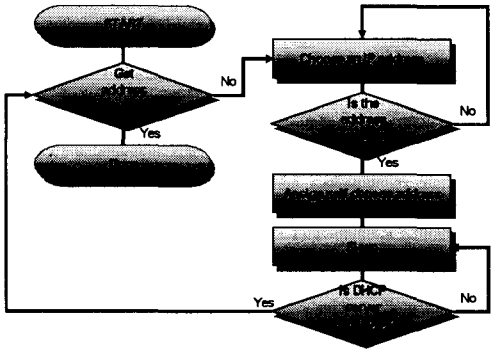


그림 2 주소 지정 흐름도
Fig. 2 Addressing Flowchart

### 2.2 Addressing

UPnP 네트워크는 지역 사설망을 구성하는 방식으로 어디서나 자신의 독립적인 네트워크를 구성하는데, Private Network을 구성하기 위해 그림 2의 흐름도에 나타난 바와 같이 먼저 DHCP 서버를 검색하고, 이를 발견하지 못할 경우, Auto IP 방식의 Link Local Network을 구성한다[3].

위 표2에 있는 UPnP 네트워크의 두 가지 구성방식은 모두 Non-Routable 즉, 공중망에서 접근이 불가능한 특정 IP 주소 대역이므로, 외부에서 UPnP 네트워크 내부로의 접근이 차단된다는 문제점을 갖고 있다[4].

## 3. 개선된 Control Point

UPnP 네트워크를 외부망에서 제어한다는 것은 Control Point가 외부망에 존재함을 의미하는데, 현재 사용되고 있는 표2와 같은 UPnP 네트워크 구성방식으로는 불가능하다. 하지만, UPnP의 정보 전달은 XML 메시지를 사용하고 있으

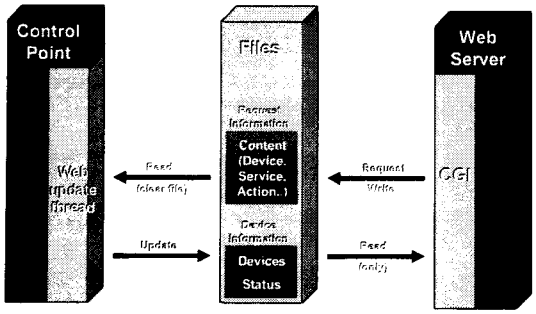


그림 3 개선된 Control Point
Fig. 3 Improved Control Point

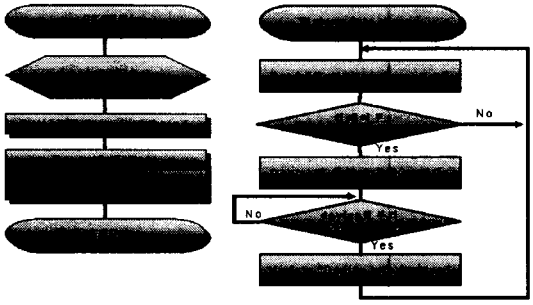


그림 4 개선된 Control Point의 Flow Chart
Fig. 4 Flow Chart of Improved Control Point

며, 구성된 UPnP 네트워크의 모든 정보는 Control Point를 통해 알 수 있으므로, Control Point의 정보를 외부망과 연결을 시켜주는 매개체만 있다면 외부망에서 Control Point를 접속하는 것과 동일한 효과를 지니게 된다.

따라서 본 연구에서는 UPnP 내부망에 연결된 Control Point에 Web Server 기능을 추가함으로써, Non-Routable 문제를 우회하여 해결할 수 있도록 기존 Control Point를 그림3과 같은 구조로 개선하였다.

개선된 Control Point 에는 Web Server 외에도 UPnP 네트워크의 정보와 외부망에서 입력되는 정보를 저장하고 교환해주는 파일 구조가 추가 되었다.

그림3은 개선된 Control Point의 내부동작을 나타내고 있다. 기존의 Control Point 기능은 그대로 수행하면서 정보파일을 통해 Web Server와 정보를 주고받는 Web update thread가 Control Point와 동시에 실행된다. 또한, Web Server는 CGI(Common Gate Interface)를 통해서 정보파일을 통해 웹 페이지를 갱신하고, 사용자의 입력을 정보파일에 기록함으로써, UPnP Device의 구동을 가능케 한다. 이상의 과정을 흐름도로 나타내면 그림 4와 같이 된다.

## 4. VPN

VPN(Virtual Private Network)은 네트워크 상의 여러 노드들(Nodes)사이의 통신을 가능하게 하는 Network 특성, 사설 망의 강력한 보안 기능을 그대로 가지는 Private 특성, 기존에 존재하는 공공망인 인터넷 상에 새로운 연결 통로를 만드는 Virtual 특성 등을 가지는 보안 망의 개념이다. 즉

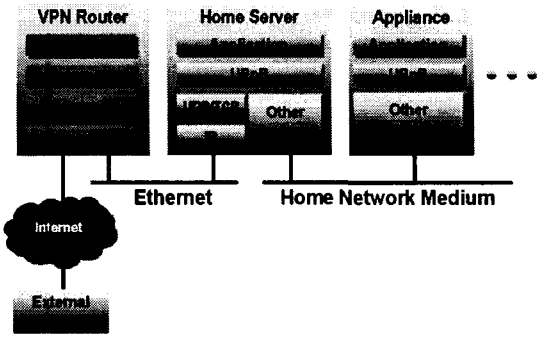


그림 5 VPN 라우터가 추가된 홈 네트워크 시스템
Fig. 5 VPN router added home network system

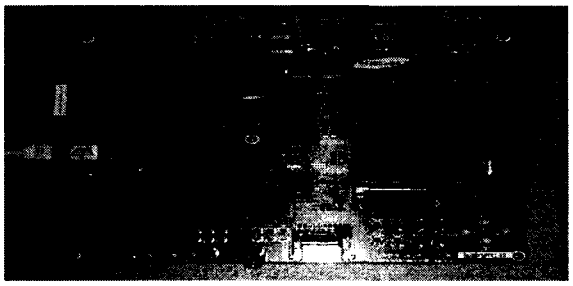

그림 6 개발에 사용된 실제 플랫폼 (ARM9)
Fig. 6 Target Platform (ARM9)

VPN을 이용함으로써, 사용자는 인터넷과 같은 공중망을 사설망과 같은 보안성을 가지고 이용할 수 있다는 장점을 가지게 된다. 그리고 VPN Standard 중에 IPsec은 IETF에서 설계하고 IPv6에서 기본으로 제공되는 보안 프로토콜로써 현재 많은 VPN 장비에 사용되고 있고, 본 논문에서도 IPsec을 이용하여 VPN을 구현하였다. 그림 5는 Super Free S/WAN, CryptoAPI, IPTABLES 등을 이용하여 구현된 VPN 라우터를 앞서 구현한 홈 네트워크 시스템에 추가한 구성도이다.

## 5. 실험 및 결과

### 5.1 시스템 구성

Embedded Linux 상에서 Home Server를 구현하기 위하여 H/W로는 그림6과 같은 ARM9 플랫폼을 사용하였으며, S/W로는 Linux Kernel을 Porting하여 구성하였다. 이를 실현하기 위하여 ARM9 플랫폼에 Linux Kernel, Intel Linux UPnP Library를 porting 하고, 개선된 Control Point code를 작성하였다. Web Server는 BOA를 사용하여 porting 하였고, Control Point와 Web Server 사이의 정보 교환을 위한 파일 구조는 이진구조의 데이터 파일을 사용하였다. ARM9 플랫폼에 개선된 Control Point를 탑재시키고, UPnP Device를 Private Network 방식으로 구성하여 실험을 하였다. UPnP Device는 Intel에서 제공하는 Intel Tools for UPnP Technology Utility를 각각의 PC에서 구동하여 시뮬레이션 하였다. 그림7은 실제 연구에 사용된 전체 시스템 구성도를 나타낸다. VPN Router는 포트 80(HTTP)에 대하여 홈 서버로 Port Forwarding을 하도

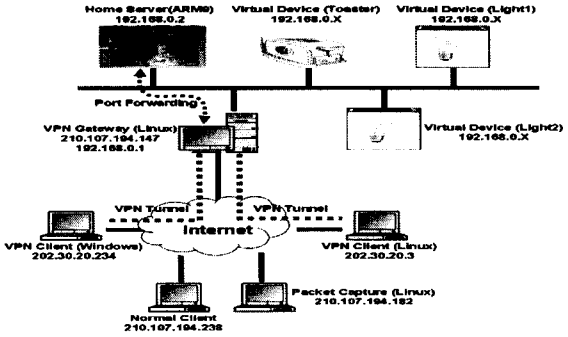


그림 7 전체 시스템 구성도

Fig. 7 Whole System Construction

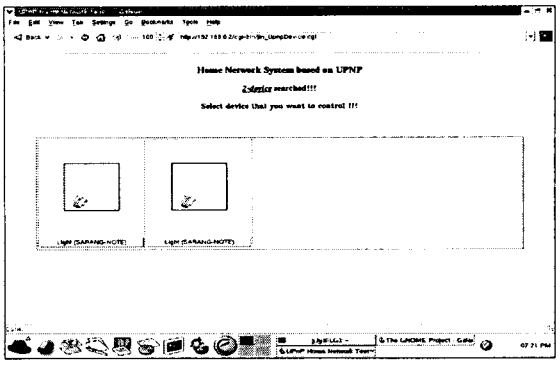


그림 8 Web Server - 추가된 Device를 인식하기 전

Fig. 8 Web Server - Before Recognition of a added Device

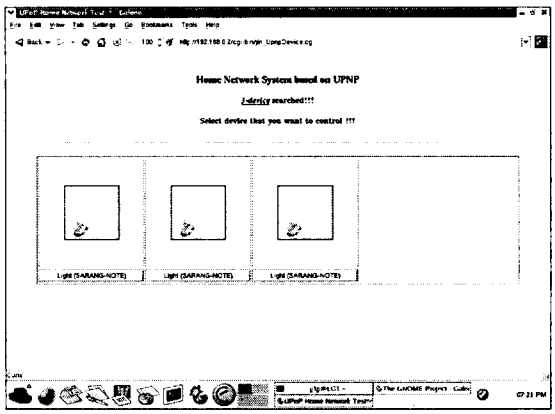


그림 9 Web Server - 추가된 Device를 인식한 후

Fig. 9 Web Server - After Recognition of a added Device

록 설정하였다. 또한 VPN Router는 Linux 기반의 Client (202.30.20.3) 및 Windows 기반의 Client (202.30.20.234)와 VPN 터널을 구축하여 IPsec을 이용한 암호화된 데이터로 통신한다. 그리고 패킷 캡쳐(Packet Capture) 컴퓨터는 별도로 두고, 그 컴퓨터 상에서 패킷 캡쳐 프로그램인 Ethereal을 이용하여 홈 서버로 오고 가는 패킷을 캡쳐하여 분석하였다. 각각의 UPnP 장치들은 홈 네트워크에 추가될 때, 홈 서버에 존재하는 DHCP 서버로부터 자동으로 192.168.0.X의 IP 주소를 할당 받게 된다.

### 5.2 실험 결과

1) UPnP Device 제어

그림8은 Web Server 상의 초기 화면으로 내부 Network 상에 UPnP Device(전등) 2개가 연결 되어 있는 모습을 나타내고 있다. 그리고 디바이스를 하나 추가 하였을 때 그림9와 같이 Device가 추가된 모습을 볼 수 있다

동시에, 그림10은 Intel에서 제공하는 Virtual Device Simulation이며, ON 명령이 수신되어 상태가 변한 보습을 그림11에 나타내었다. 따라서 외부 Client로 부하의 명령이 수행됨을 알 수 있다.

그림 12를 통해 현재 선택된 디바이스의 상태가 OFF 상태임을 웹 페이지의 내용에서 확인 할 수 있으며, 화면 아래의 명령입력 부분(라디오 버튼)을 통해서 "ON" 명령을 홈 서버에 전달한다. 그림 13은 "ON" 명령을 전달 받은 UPnP 디바이스는 명령을 수행하고, 해당 디바이스의 변경된 상태는 웹을 통해서 확인 할 수 있다.

2) VPN

다음의 그림들은 VPN Router와 VPN Client 사이에 암호화된 데이터 교환이 이루어지고 있음을 보여준다. 이 과정을 확인하기 위해 Packet Capture 컴퓨터(210.107.194.182)에서 홈 서버로 오고 가는 패킷을 캡쳐하였다.

그림14는 일반 Client가 VPN Router를 통해 홈 서버에 접속할 때 주고 받는 패킷을 Paket Capture 컴퓨터(210.107.194.182)에서 Ethereal을 이용하여 캡쳐한 화면이다. 그림14-①은 주고 받는 데이터가 프로토콜 HTTP를 이용한다는 것을 나타내며, 그림14-②는 실제 주고 받는 데이터의 내용을 보여준다. 이처럼 보안을 이용하지 않고 홈 서버에 접속하게 되면 Client와 서버가 주고 받는 모든 데이터의 내용을 제3자가 그대로 볼 수 있다는 문제점을 안고 있다. 따라서 이러한 문제를 본 논문에서는 VPN을 이용하여 해결하였고 다음그림은 그러한 결과를 보여준다.

그림15에서 알 수 있듯이 VPN Client가 VPN Router를 통해 웹 서버에 접속하게 되면, 그림15-②와같이 ESP라는 암호화 프로토콜만이 보이게 된다. 따라서 이 패킷을 가로챈 제 3자는 Client가 서버에 HTTP를 이용하여 접속했는지, TELNET를 이용하여 접속했는지, 혹은 다른 어떤 프로토콜을 이용하여 접속했는지에 대해 전혀 알 수 없게 된다. 또한 Client와 서버는 ESP를 이용한 암호화된 데이터로 통신하기 때문에, 그림15-③과 같이Client와 서버가 주고 받는 어떤 데이터의 내용도 볼 수 없게 된다. 그림15-①은 이 프로그램이 동작하는 인터페이스를 지정한 것이다. 이렇듯 VPN을 이용하여 Client와 서버가 통신하게 되면 공중망인 인터넷을 이용하면서도 안전한 통신을 할 수 있게 된다.

다음 그림16은 VPN Client (202.30.20.3)에서 VPN Server와 VPN Tunnel을 구축한 후 생성되는 새로운 인터페이스 IPsec0에 대해 패킷 캡쳐 프로그램인 Ethereal을 실행시킨 결과이다. 그림16에서 보듯이, Client와 서버는 VPN Server와 VPN Client 사이에 형성된 VPN Tunnel을 통해실제의 HTTP 데이터를 주고받고 있음을 확인할 수 있다. 즉, 공중망인 인터넷 상에서는 암호화된 데이터 형태만이 보이

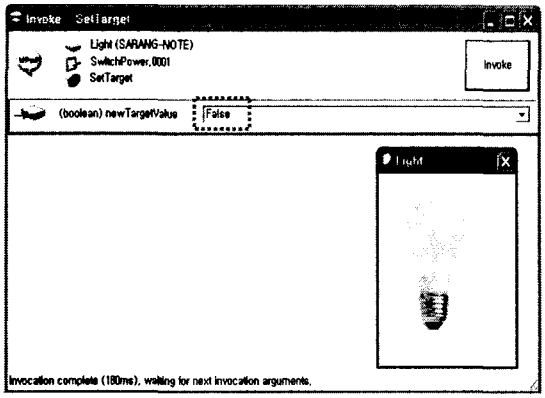


그림 10 Device 시뮬레이션 (ON명령 도착 전 : OFF)
Fig. 10 Device Simulation (Before receive "ON" : OFF)

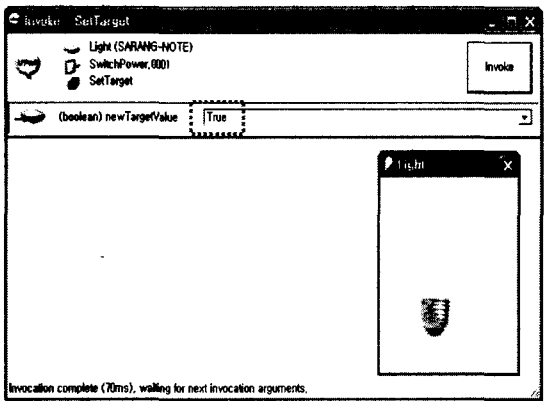


그림 11 Device 시뮬레이션 (ON 명령 도착 후 : ON)
Fig. 11 Device Simulation (After receive "ON" : ON)

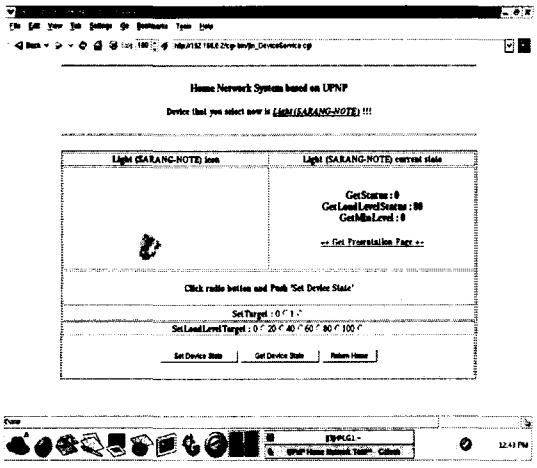


그림 12 장치의 상태정보 출력 및 서비스 제어
Fig. 12 State information display of device and Service check

고 VPN Tunnel을 통해서 보았을 때에만 Client와 서버가 주고 받는 실제 데이터를 확인할 수 있는 것이다.

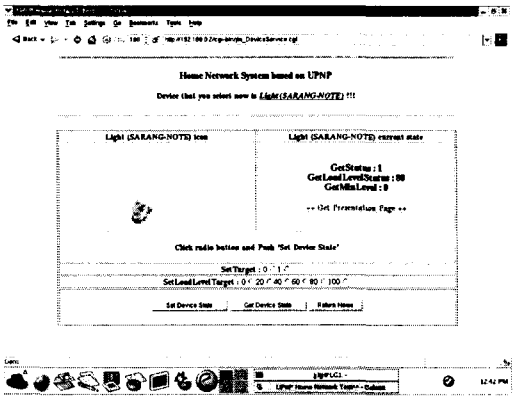


그림 13 제어 이후의 장치의 상태정보
Fig. 13 State information display of device after control

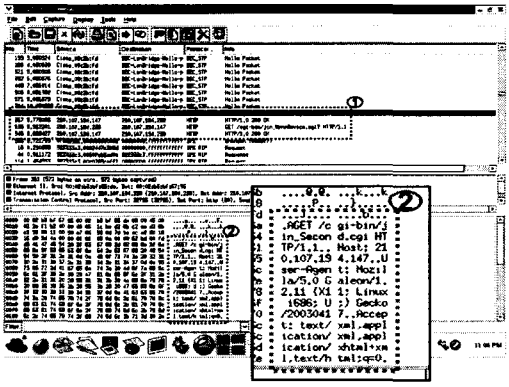

그림 14 일반 사용자가 홈 서버에 접속할 때의 패킷 캡쳐
Fig. 14 Packet capture, when normal client connects to Home Serve

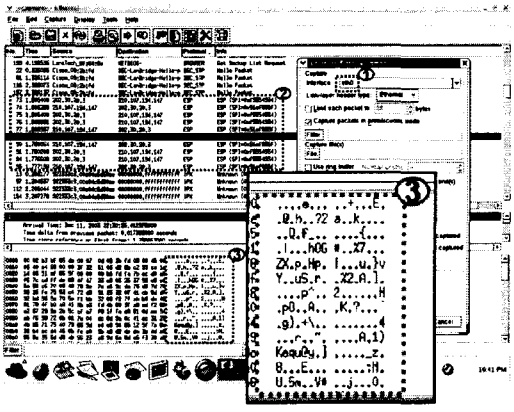

그림 15 VPN Client가 홈 서버에 접속할 때의 패킷 캡쳐
Fig. 15 Packet capture, when VPN Client connects to Home Server

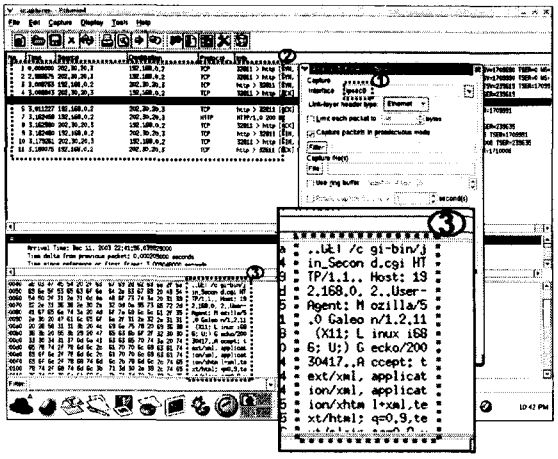

그림 16 VPN Client와 VPN Server의 VPN Tunnel에서의 패킷 캡쳐

Fig. 16 Packet capture in VPN tunnel of VPN Client and VPN Server

본 논문에서 UPnP 네트워크의 단점인 Non- Routable 문제를 "개선된 Control Point"를 사용함으로써 해결하여 외부망에서도 UPnP의 Private Network 네트워크에 접속하여 Device 들을 제어하고 모니터링 할 수 있는 방법을 제안하였다. 이 제안한 방법을 ARM9 플랫폼에 Embedded Linux를 사용하여 실현하였고 UPnP 장치는 Intel에서 제공하는 Simulation Tool을 사용하여 외부 인터넷상에서 부하를 제어하고 모니터링이 됨을 보여 주었다. 또한 외부 침입으로부터의 보안을 위하여 VPN을 첨가하여 사용된 자료가 유출되지 않음을 보여주었다.

## 참 고 문 헌


[1] M. Jeronimo and J. Weast, "UPnP Design by Example," Intel Corporation, April 2003.

[2] T. Fout, "Universal Plug and Play in Windows XP," Microsoft Corporation, July 2001.

[3] Microsoft, "Universal Plug and Play Device Architecture", Microsoft Corporation, June 2000.

[4] IANA, "Special-Use IPv4 Address", IANA, Sep, 2002.

[5] K. Yaghmour, "Building Embedded LINUX SYSTEMS," O'Reilly & Associates Inc., April 2003.


## 저 자 소 개


**정 진 규 (鄭 震 珪)**

1976년 7월 15일생 2002.2 아주대학교 전자공학과 졸업 2004.2 아주대학교 대학원 전자공학과 졸업 (석사)

Tel : 017-513-3566

E-mail : fire0715@hanmail.net

**진 선 일 (陳 善 一)**

1976년 11월 17일생 2002.2 아주대학교 전자공학과 졸업 2004.2 아주대학교 전자공학과 졸업 (석사)

Tel : 019-326-1625

E-mail : jjj333jjj@hotmail.com

**이 희 정 (李 熙 靜)**

1980년 3월 3일생. 2003년 국립 목포 해양대학교 기관 공학 졸업. 2003년~현재 아주대학교 대학원 석사과정

Tel : 016-664-5870

E-mail : sarang5870@hanmail.net

**황 인 영 (黃 仁 永)**

1964년 3월 1일생 1987년 서울대학교 전기공학과 졸업, 2000년 경북대학교 대학원 전자공학과 졸업(석사) 현재 LG전자 홈넷사업팀 시스템개발Gr. 책임연구원

관심분야 :홈넷 솔루션, 센서 네트웍

Tel: 02-526-4403

E-mail : yyhwang@lge.com

**홍 석 교 (洪 錫 敎)**

1948년 8월 2일생 1971년 서울대학교 전기공학과 졸업 1981년 서울 대학교 대학원 전기공학과 졸업 (박사) 1976년~현재 아주대학교 전자공학부 교수. 관심분야: 로봇제어, 컴퓨터 응용, 지능제어

Tel : 031-219-2478

Fax : 031-212-9531

E-mail : skhong@ajou.ac.kr