

홈 네트워킹을 위한 UPnP 프록시 디바이스의 설계

김철민, *진성기, **한상숙, 은성배
한남대학교 정보통신공학과, *한국과학기술원 전산학과, **대전기능대학교 멀티미디어학과
전화 : 042-627-8886 / 핸드폰 : 016-779-3622

A Design of An UPnP Proxy Device for Home Networking

Chulmin Kim, Sungki Jean, Seongbae Eun
Dept. of Information & Communication Engineering, Hannam University
Email : cmkim@daniel.hannam.ac.kr

Abstract

We investigate the functions of UPnP protocol that would be used to provide easy and cheap access for home networking.

We suggest that a proxy device may reduce the cost of establishing UPnP network if it serves as an interface between UPnP and X10 protocol.

We also describe the design of UPnP proxy device and the service scenario.

I. 서론

정보 기술의 발달과 인터넷을 비롯한 네트워크 기술의 발달로 인해 산업에 주로 이용 되던 네트워크 기술이 가정까지 보급되고 있다. 홈 네트워킹은 가정에 컴퓨터가 두 대 이상으로 늘어나면서 연구가 시작되었으나 현재는 컴퓨터뿐만 아니라 가전제품에 까지 적용되어 가전제품이 정보 기기화 되는데 중요한 역할을 담당하고 있다.

홈 네트워킹이 가전 제품까지 적용 되면서 연구 개발되고 있는 것이 UPnP(Universal Plug and Play)이다. UPnP는 다양한 기기 제조 업체, 다양한 기능을 갖는 기기들을 네트워크에 연결하는데 최소한의 비용만을 이용하여 능동적인 홈 네트워킹을 구성할 수 있도록 하는 기술이다. UPnP 기술은 임의의 기기가 홈 네트워킹에 참여 하였을 때 그 기기가 제공하는 서비스를 참여한 네트워크에 속한 제어 포인트를 통해 쉽게 이용할 수 있도록 규정한 것이다.

현재까지 정의된 UPnP 기술의 문제점은 현재 사용되고 있는 많은 기기들과 UPnP 기능을 탑재 하기 어려운 기기들에 대한 처리 문제 이다. 이러한 기기들을 레거시 디바이스(Lagacy Device)라고 부르기로 한다. 따라서 레거시 디바이스가 UPnP 기능을 갖도록 만드

는 중간 단계의 디바이스가 필요하다. 이를 UPD(UPnP Proxy Device)라 부르기로 한다.

UPD를 설계할 때 고려되어야 할 사항이 UPD와 레거시 디바이스간의 인터페이스이다. 현재 이용 가능한 레거시 디바이스의 인터페이스는 매우 다양하여 본 논문에서는 X10 인터페이스를 제공하는 UPD를 설계하였다.

II. 배경

2.1 홈 네트워킹

네트워크의 급속한 발달은 가정까지 이어져 개인용 컴퓨터를 비롯한 가정에서 사용하는 여러 가전 기기까지 네트워크에 연결 되고 있다.

현재 사용되는 홈 네트워킹의 구성은 세 부분으로 나눌 수 있다.

첫째는 외부 인터넷 망과의 연결을 위한 부분으로서 우리나라 예서는 xDSL, Cable등이 많이 사용되고 있다. 이런 서비스는 주로 모뎀을 통하여 수행 되고 이 모뎀은 가정 내부의 네트워크와 외부의 네트워크를 연결하는 게이트웨이 역할까지 담당 한다.

둘째로 가정 내부의 네트워크를 위한 물리적 전송 방법으로 전화선, 전력선, 이더넷, IEEE 1394, USB, IEEE 802.11b, Home RF, 블루투스 등의 접속 방법이 있다. 이러한 물리적 접속 방법은 TCP/IP 프로토콜 지원을 권장하여 서로 다른 물리적 접속 방법 간에 통신 호환성을 갖춘다.

마지막으로 홈 네트워크에 연결된 다양한 기기들을 관리하기 위한 홈 서버가 있다. 홈 네트워킹의 발달로 가정 기기들이 점차 정보 기기화 되어 네트워크에 접속된 기기들을 쉽게 관리하기 위한 방법이 필요하다.

또한 홈 네트워크에서 사용될 수 있는 다양한 물리적 접속 방법 중 TCP/IP를 지원하지 못하는 기기에 접속 중계도 홈 서버를 통해 가능하도록 한다. 하지만 홈

서버의 개발은 다양한 접속 방법과 제어 방법을 지원해야 하므로 개발하는데 어려움이 따른다.

2.2 UPnP(Universal Plug and Play)

UPnP는 마이크로 소프트의 윈도우95가 발표된 이후로 제공한 PnP(Plug and Play)기능을 네트워크 범위까지 확장한 개념이다.

PnP기능을 통해 사용자는 개인용 컴퓨터에 더 쉽게 주변 장치를 추가해 사용할 수 있었다. 이와 같이 UPnP는 네트워크에 접속 될 수 있는 다양한 장치들이 네트워크에 접속 됐을 때 접속된 기기의 서비스를 사용자의 설정 없이 사용할 수 있도록 지원한다. 따라서 UPnP는 다양한 제조업체, 다양한 접속 방식을 지원하는 네트워크 정보 기기들이 네트워크에 참여하여 서비스를 제공하는 방법을 정의한 표준이다.

기본적인 통신 방법으로 TCP/IP를 제공하여 인터넷을 통해 다양하게 기기들을 관리, 제어할 수 있는 기능을 제공한다.

UPnP의 프로토콜 스택은 그림1과 같다.

UPnP Vendor				
UPnP Forum				
UPnP Device Architecture				
HTTPMU		HTTPU	SOAP	HTTP
GENA	SSDP	GENA	HTTP	GENA
UDP			TCP	
IP				

그림1 UPnP 프로토콜 스택

다음은 UPnP 서비스의 동작에 관한 설명이다.

ㄱ. 제어포인트와 UPnP 가능 디바이스

제어 포인트는 네트워크에 참여한 UPnP 가능 디바이스가 제공하는 서비스를 이용할 수 있는 지점이다. 제어 포인트를 통해서 네트워크에 존재하는 서비스를 사용한다. 일반적으로 제어 포인트는 UPnP를 지원하는 운영체제가 담당한다.

UPnP 가능 디바이스는 UPnP를 통해 서비스를 제공할 수 있는 디바이스이다. UPnP 규격을 지원하여 서비스를 제공하는 주체이다. 현재 UPnP Forum에서 인증한 UPnP가능 디바이스는 인텔의 AnyPoint(™) Net working Gateway[2], 마이크로 소프트의 Internet Connection Sharing[2]이 있다.

ㄴ. Addressing

UPnP 서비스는 네트워크에서 기본적으로 TCP/IP 프

로토콜을 이용하기 때문에 UPnP 가능 디바이스가 네트워크에 참가할 경우 IP를 획득하는 작업이 필요하다.

DHCP를 이용하여 획득을 시도하고 DHCP가 실패할 경우 다른 기기들과의 충돌을 피하는 임의의 IP를 획득하는 방법인 Auto-IP[6]를 이용하여 IP 주소를 획득한다.

ㄷ. Discovery

IP를 획득 후 UPnP 가능 디바이스는 SSDP(Simple Service Discovery Protocol), GENA(General Event Notification Architecture)를 이용하여 자신이 네트워크에 참가했음을 알린다.

GENA는 HTTP를 기반으로 어떤 이벤트의 발생을 알리는 방법이다.

SSDP는 HTTP를 기반으로 메시지를 주고 받으며 네트워크에 참가한 기기의 존재를 알리는 방법이다.

GENA와 SSDP를 이용할 때 멀티캐스트와 유니캐스트를 이용하기 때문에 HTTP의 확장된 형태로 멀티캐스트와 유니캐스트를 지원하는 HTTPMU(HTTP Multi cast over UD), HTTPU(HTTP unicast over UDP)를 이용한다.

ㄹ. Description

UPnP 가능 디바이스가 네트워크에 참가했음을 알린 후 제어 포인트는 접속된 기기로부터 자세한 서비스 내용을 얻어오는 작업을 한다. 이때 SSDP, GENA를 통해 요청한다. UPnP 가능 디바이스가 제공하는 서비스에 대한 설명, 사용 가능한 이벤트들을 XML[7]을 이용하여 기술 한다.

ㅁ. Control

제어 포인트는 UPnP 가능 디바이스에 명령을 내릴 수 있다. 이는 SOAP[8](Simple Object Access Protocol)을 이용한다. SOAP는 HTTP위에서 동작하고 네트워크를 통해 잘 정의된 이벤트를 전달 할 수 있는 기능을 갖는다.

ㅂ. Event

UPnP 가능 디바이스가 네트워크에 참가했을 때 제어 포인트는 다수가 될 수 있다. Event는 다수의 제어 포인트가 한 개의 서비스를 동시에 이용하고 있을 때 한 제어 포인트를 통해 이루어진 상태 변경을 다른 제어 포인트에 알리는 기능을 한다.

ㅅ. Presentation

Presenation은 제어 포인트를 통한 제어 외에 개인 컴퓨터에서 웹 브라우저를 통해 제어할 수 있다.

Presentation은 HTML로 구성되어 사용자가 쉽게 알아볼 수 있도록 사용자 인터페이스를 지원 한다.

o. 서비스 시나리오 및 UPnP의 장점

홈 네트워킹에 UPnP 가능한 디바이스인 개인 컴퓨터, 인터넷 게이트웨이 디바이스가 존재한다. 개인용 컴퓨터를 이용하여 인터넷을 이용하려 할 때 개인용 컴퓨터에 존재하는 제어 포인트를 이용해 인터넷 공유를 지원하는 서비스를 찾아 사용할 수 있다. 또한 프린터가 존재할 경우 프린터를 네트워크에 연결하기만 하면 개인 컴퓨터에서는 프린터 서비스를 찾아서 이용할 수 있게 된다.

기존의 방법은 프린터를 컴퓨터에 연결하고 프린터 드라이버를 설정하고 프린터를 사용하는 단계가 필요하고 네트워크에서 프린터를 이용할 경우는 프린터를 공유 하는 단계까지 필요하다. 하지만 UPnP를 이용할 경우 프린터 단독으로 네트워크에 연결 하는 작업만으로 프린터를 여러 컴퓨터에서 이용할 수 있게 된다.

2.3 X10

X10은 집안의 전기 배선을 통하여 명령을 주고 받아 전등, 스위치, 콘센트등을 제어할 수 있도록 하는 간단한 “통신 언어” 이다.

X10은 전선을 통해 전기와 다른 신호를 주고 받을 수 있는 것을 착안하여 제안 됐다.

X10은 X10신호를 보낼 수 있는 모듈, X10신호를 받을 수 있는 모듈, X10신호를 보내고 받을 수 있는 모듈의 세가지 모듈이 존재 한다.

X10신호를 수신할 수 있는 모듈을 전원에 연결하고 그 모듈에 전등을 연결 하면 X10신호를 송신할 수 있는 모듈에서 X10신호를 이용하여 전등을 조작 할 수 있다. X10지원 모듈의 경우 House Codes, Device Codes를 이용하여 구분할 수 있고 X10 송신 모듈을 통해 각각의 X10 수신 모듈을 모두 제어할 수 있다.

X10을 통해 On, Off, Dim, Bright, Status On, Status Off, Status Request등의 명령을 주고 받을 수 있다.

III. 설계

3.1 UPD(UPnP Proxy Device)의 개요

UPnP기능을 지원하는 디바이스의 경우 기본적으로 TCP/IP, HTTP, HTTPMU, HTTPU, GENA, SSDP, SOAP등의 다양한 프로토콜이 필요하다. 이런 프로토콜을 디바이스에 탑재하기 위해서는 디바이스의 주 기능을 제공하는 비용 외에도 추가 비용의 문제가 발생하게 된다. 따라서 간단한 기능을 갖는 전등, 선풍기 같은 레거시 디바이스의 경우 UPnP 기능을 포함하여

디바이스를 제작할 경우 기본 기능을 위한 비용 보다 UPnP 기능을 위해 포함되는 비용이 더 커지는 문제가 발생 한다.

이때 UPnP기능을 제공할 수 있는 프록시 디바이스를 통하면 한 개의 프록시 디바이스를 통해 다수의 레거시 디바이스를 UPnP기능을 하도록 해 줄 수 있어 비용상의 이득을 얻을 수 있다.

프록시 디바이스가 임의의 디바이스를 지원하기 위해서는 다양한 디바이스의 인터페이스 및 제어 가능한 기능들을 프록시 디바이스가 모두 알아야 하는 어려움이 있다. 따라서 본 논문에서는 전원을 이용한 간단한 인터페이스인 X10기능을 갖는 레거시 디바이스를 지원하는 프록시 디바이스를 설계 하였다.

3.1 UPnP 프록시 디바이스의 구조

UPnP 프록시 디바이스의 구조는 그림2와 같다. 임베디드 시스템은 홈 네트워크 접속 부분과 X10 모듈을 내장하고 있다.

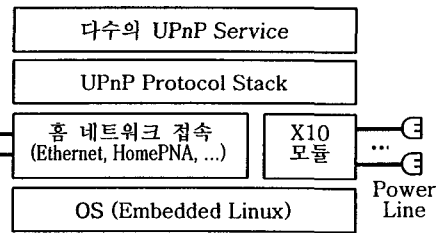


그림2 UPnP 프록시 디바이스의 구조

UPD에 포함될 UPnP 프로토콜 스택은 Intel에서 배포하는 리눅스용 UPnP SDK를 임베디드 리눅스에서 동작 할 수 있도록 포팅하여 사용한다. UPnP SDK는 HTTP, HTTPU, HTTPMU, SOAP, GENA, SSDP 등을 지원하는 API 이다.

UPnP SDK는 제어 포인트용 프로그램을 만들고 UPnP 서비스 에뮬레이션 프로그램을 만들 수 있다. 4~8개의 X10 모듈을 각각 UPnP 서비스처럼 만들기 위해서 서비스 에뮬레이션 프로그램을 만든다. 이때 X10모듈의 동작에 대한 Description과 Presentation을 기술한 템플릿을 두어 사용자가 UPD를 사용할 때 최소한의 설정을 통해 UPD를 사용 가능 하도록 구성한다. 서비스 Description으로 포함될 기본적인 템플릿의 예는 그림3과 같다.

```
<?xml version="1.0"?>
<root xmlns="urn:schemas-upnp-org:device-1-0">
<specVersion>
<major>1</major>
<minor>0</minor>
</specVersion>
```

```

<device>
<udn>uuid:UUID NUMBER</udn>
<friendlyName>INPUT DEVICE NAME</friendlyName>
<deviceType>urn:schemas-upnp-
org:device:lighting:NUMBER</deviceType>
<presentationURL>PRESENTATION.HTML</presentationURL>
<manufacturer>Microsoft</manufacturer>
<manufacturerURL>http://www.microsoft.com/</manufacturerURL>
<modelName>Upd-X10-module1</modelName>
<modelName>1</modelName>
<modelDescription>UPD-X10 control</modelDescription>
<upc>000000000001</upc>
<serialnumber>0000001</serialnumber>
<serviceList>
<service>
<serviceType>urn:schemas-upnp-
org:service:pwrdim:1</serviceType>
<serviceId>urn:upnp-org:serviceId:pwrdim</serviceId>
<controlURL>control.dll?pwrdim</controlURL>
<eventSubURL>control.dll?pwrdim </eventSubURL>
</service>
</serviceList>
</device>
</root>
    
```

그림3 Description 템플릿의 예

그림4는 X10 지원 레거시 디바이스 제어 서비스의 Presentation을 웹 브라우저를 통해 본 예이다.



현재 상태

현재 상태	
전원	ON
밝기	10

그림4 제어 화면의 예

3.2 서비스 시나리오

UPD는 다음과 같이 사용할 수 있다.

ㄱ. UPD를 네트워크에 접속한다. UPnP의 동작 규격에 따른 절차를 거쳐 UPD의 기본 서비스가 동작 한다.

기본 서비스는 현재 UPD가 제공하는 X10 지원 레거시 디바이스의 상태 확인 및 X10 지원 레거시 디바이스가 접속 됐을 때 동작 설정을 지원하는 서비스이다.

UPD의 기본 서비스는 네트워크 내의 제어 포인트를 통해 사용하거나 기본 서비스가 제공하는 Presentation을 통해 쉽게 사용 가능하다.

ㄴ. UPD에 X10 지원 레거시 디바이스가 접속 될 경우 UPD는 이를 감지하여 접속된 레거시 디바이스에 대한 서비스를 수행 시킬 수 있도록 상태를 변환한다. 이때

이미 설정이 완료된 X10지원 레거시 디바이스의 경우에는 바로 접속한 레거시 디바이스에 대한 서비스를 동작 시킨다.

ㄷ. 접속된 X10 지원 레거시 디바이스를 제어할 수 있는 기본 명령들을 설정 한다. 설정을 완료하면 접속된 X10 지원 레거시 디바이스는 UPD에서 UPnP 가능 디바이스의 서비스로서 동작 한다.

ㄹ. 설정 완료된 X10 지원 레거시 디바이스는 홈 네트워크 내부의 제어 포인트를 통해서 제어 할 수 있다.

IV. 결론 및 향후 연구 방향

본 연구에서는 활용도가 높아지고 있는 홈 네트워크 환경에서 적은 관리 비용을 제공하면서 사용하기 편리한 UPnP 기능에 대해서 알아보고 UPnP를 지원하는 홈 네트워크 환경으로의 진화 과정의 중간 과정에서 기존의 레거시 디바이스 및 비용상의 문제로 UPnP를 지원 할 수 없는 X10 지원 디바이스가 UPnP 지원 네트워크에 참여할 수 있도록 하는 UPD에 대한 설계를 하였다.

본 논문에서는 UPD와 레거시 디바이스간의 인터페이스 문제로 인해 기본적인 인터페이스와 서비스를 제공하는 X10 지원 레거시 디바이스에 한정해서 설계 하였으나 향후에는 USB, IEEE 1394, Serial Port, Parallel Port등으로 접속되는 디바이스까지 지원하는 UPnP 프록시 디바이스에 대한 지원 방안이 필요하다. 또한 본 논문에서 설계한 것을 토대로 UPD를 구현 할 수 있다.

참 고 문 헌

- [1] UPnP Forum의 기술 문서, <http://www.upnp.org/resources>
- [2] Certified UPnP Devices, <http://www.upnp-ic.org/certification/default.asp>
- [3] Intel SDK for UPnP Devices (Linux), <http://developer.intel.com/ial/upnp/>
- [4] Microsoft Device Kit for UPnP Devices, <http://www.microsoft.com/hwdev/upnp/>
- [5] X10 기술문서, <http://www.x10.org>, <http://www.smarthome.com>
- [6] Auto-IP, IETF draft
- [7] XML, W3C recommendation
- [8] SOAP, IETF draft & W3C Technical Report