

UPnP 기반의 홈-네트워크 원격제어를 위한 개선된 Control Point의 제안 및 구현

Suggestion and Implementation of Improved Control Point for Remote Control Home-Network based on the UPnP

정진규*, 진선일*, 안광혁*, 유영동*, 홍석교*

* 아주대학교 전자공학과 (전화 : (031)219-2489, 팩스 : (031)212-9531, E-mail : fire0715@hanmail.net)

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 study 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.

Keywords : Ubiquitous, Home Network, Middleware, UPnP, Control Point, Embedded Linux, ARM9

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 불리는 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 멀티

미디어를 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 실제 적용에 있어 많은 편의를 제공하고 있다.[1]

그러나 유비쿼터스 컴퓨팅의 목적이 그러하듯 핸드폰, PDA, 인터넷등을 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.

따라서 본 논문에서는 별도의 추가적인 장비 없이도 기존의 UPnP 홈 네트워크 시스템의 외부에서 접근 및 제어가 가능하도록 Control Point 구조의 개선점을 제안하고, 이를 구현함으로써, 그 가능성을 검증하는데 의의를 둔다.

II. UPnP Middleware

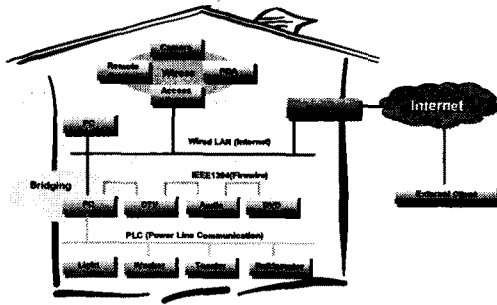


그림 1. 홈 네트워크의 구성
Fig 1. Home Network Topology

그림1에서는 다양한 통신매체로 연결된 전형적인 홈 네트워크의 구성을 보여주고 있으며, 이러한 구성이 가능하기 위해서는 다양한 프로토콜의 상호연동을 가능케 하는 middleware가 필수요소이다.

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

표1에서 대표적인 middleware 세 가지를 비교하고 있는데, 이중 가장 대표적인 UPnP의 동작과 구성은 다음과 같다.

- Service : UPnP 네트워크의 최소 제어단위로, 동작을 나타내고 상태변수를 통하여 모델링 된다.
- Device : 서비스 혹은 다른 내장형 장치로 구성된다.
- Control Point : UPnP 네트워크의 다른 장치를 검색하고 제어한다.[2]

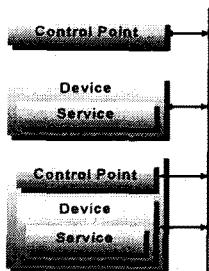


그림 2. UPnP 구성 예
Fig 2. UPnP Example

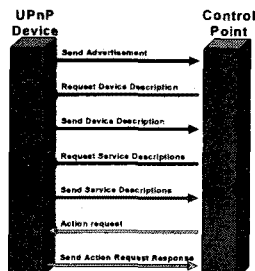


그림 3. 기존 Control Point
Fig 3. Established Control Point

UPnP 네트워크는 지역 사설망을 구성하는 방식으로 어디서나 자신의 독립적인 네트워크를 구성하는데, Private Network를 구성하기 위해 먼저 DHCP 서버를 검색하고, 이를 발견하지 못할 경우, Auto IP 방식의 Link Local Network를 구성한다.[3]

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

Private Network	Link Local
<ul style="list-style-type: none"> • DHCP • 192.168.0.0 / 16 • Non-Routable 	<ul style="list-style-type: none"> • Auto IP • 169.254.0.0 / 16 • Non-Routable

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]

III. 제안된 Control Point

UPnP 네트워크를 외부망에서 제어한다는 것은 Control Point가 외부망에 존재함을 의미하는데, 현재 사용되고 있는 표2와 같은 UPnP 네트워크 구성방식은 불가능하다.

하지만, UPnP의 정보 전달은 XML 메시지를 사용하고 있으며, 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 그림3의 기존 Control Point를 그림4와 같은 구조로 개선하였다.

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

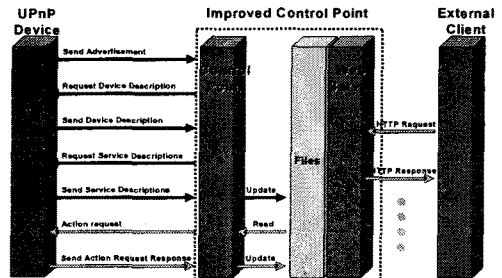


그림 4. 개선된 Control Point의 동작구조
Fig 4. Improved Control Point's Behavioral Architecture

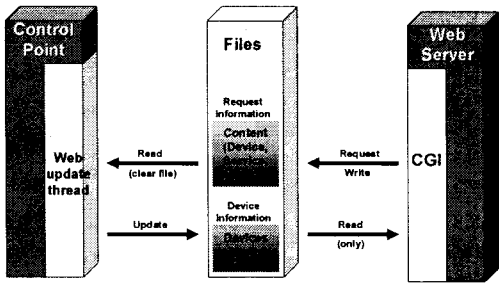


그림 5. 개선된 Control Point
Fig 5. Improved Control Point

그림5는 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.

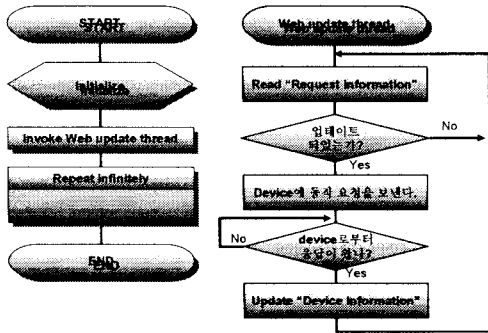


그림 6. 개선된 Control Point의 Flow Chart
Fig 6. Flow Chart of Improved Control Point

IV. 실험 및 결과

표 3. 소프트웨어 구성

Table 3. Component of Software

	내 용
OS	Embedded Linux (kernel v2.4.18)
Library	Intel UPnP SDK 1.2.1
UPnP Device	Intel Tools for UPnP Technology

표 4. 하드웨어 구성

Table 4. Component of Hardware

	내 용
CPU	S3C2410 (ARM9)
RAM	64 Mbyte
ROM (Flash)	64 Mbyte(Smart Media Card)
Network Interface	CS8900, PCMCIA

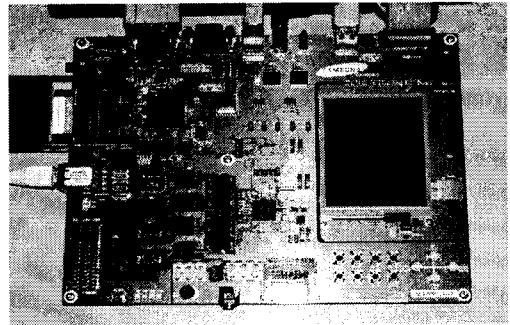


그림 7. 개발에 사용된 실제 플랫폼 (ARM9)
Fig 7. Target Platform (ARM9)

실험과정의 첫 단계는 ARM9 플랫폼에 Linux Kernel을 porting 하는 것이다.[5] 다음으로 Intel Linux UPnP Library를 porting 하고, 개선된 Control Point code를 작성하고, Web Server는 BOA를 사용하여 porting 하였고, Control Point와 Web Server 사이의 정보 교환을 위한 파일 구조는 이진구조의 데이터 파일을 사용하였다.

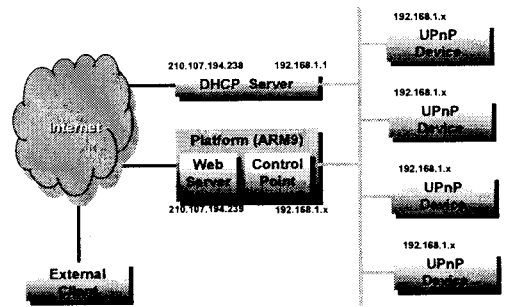


그림 8. 전체 시스템의 구성
Fig 8. System Architecture

그림8에 나타냈듯이 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.

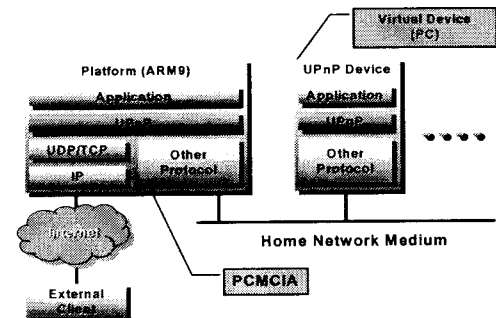


그림 9. 내부 통신망의 계층구조
Fig 9. System Structure

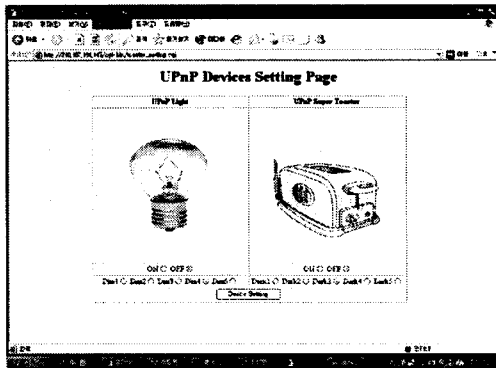


그림 10. ON 명령을 보내는 화면 (전등상태 : OFF)
Fig 10. Screen of Sending "ON"(Light Status : ON)

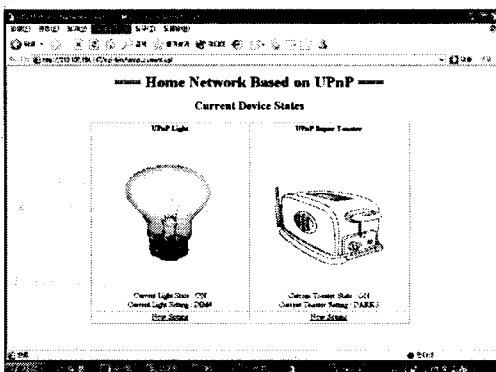


그림 11. 업데이트된 화면 (전등상태 : ON)
Fig 11. Updated Screen (Light Status : ON)

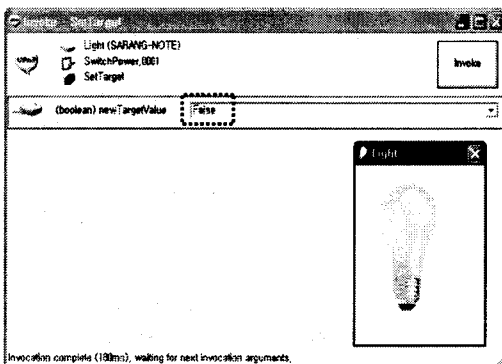


그림 12. Device 시뮬레이션 (ON명령 도착 전 : OFF 상태)
Fig 12. Device Simulation (Before receive "ON" : OFF)

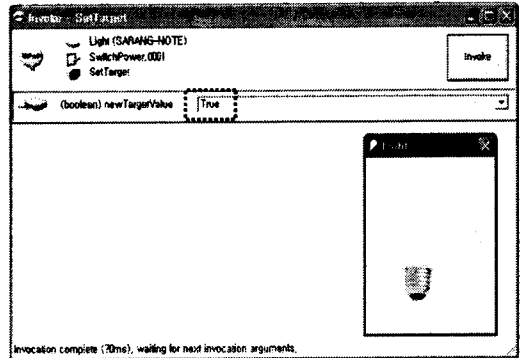


그림 13. Device 시뮬레이션 (ON 명령 도착 후 : ON 상태)
Fig 13. Device Simulation (After receive "ON" : ON)

그림10, 11은 외부망에서 개선된 Control Point에 접속하여 바라본 Device들이며, 그림10에서 Off 상태의 전등에 ON 명령을 전송하여 그 결과를 그림11의 업데이트된 화면에서 확인할 수 있다.

동시에, 그림12는 ON 명령이 수신되기 전의 Off 상태의 전등 시뮬레이션 모습이고, ON 명령이 수신되어 상태가 변한 모습을 그림13에 나타내었다.

V. 결 론

이로써, 본 논문에서 제안한 "개선된 Control Point"를 통하여 외부망에서도 UPnP의 Private Network 네트워크에 접속하여 Device 들을 제어하고 모니터링 할 수 있음을 검증 하였다.

본 논문에서 UPnP 네트워크의 단점인 Non-Routable 문제를 개선하여 외부망에서도 접근할 수 있는 방법을 제안하고, 검증을 하였다. 그러나, 홈 네트워크는 편리해야 할 동시에 외부의 침입으로부터 안전해야 할 것이다. 따라서 현재의 UPnP의 또 다른 문제점인 보안성에 대한 연구가 뒤따라야 할 것이다.

VI. 참고문헌

- [1] Michael Jeronimo, Jack Weast, "UPnP Design by Example" Intel Corporation, Apr, 2003
- [2] Tom Fout, "Universal Plug and Play in Windows XP", Microsoft Corporation, July, 2001
- [3] "Universal Plug and Play Device Architecture", Microsoft Corporation, Jun, 2000
- [4] "Special-Use IPv4 Address", IANA, Sep, 2002
- [5] Karim Yaghmour, "Building Embedded LINUX SYSTEMS", O'Reilly & Associates Inc, Apr, 2003