스마트 홈 환경에서 UPnP를 이용한 동적인 센서 서비스 발견 기법*

박경환, 조진성

hicom@mesl.khu.ac.kr chojs@khu.ac.kr

경희대학교 컴퓨터공학과

A Dynamic Sensor Service Discovery Technique Using UPnP in Smart-Home Environment

Park Kyung Hwan Jinsung Cho

Dept. of Computer Engineering, Kyung Hee University

요 약

최근 센서 네트워크는 군사 및 과학 분야뿐만 아니라 스마트 홈 환경과 같은 다양한 분야에 적용하기 위한 연구가 활발히 진행되고 있다. 특히 스마트 홈 환경에서의 센서 네트워크는 가정 내부의 인터넷망과 연동하여 사용자에게 좀 더 유연한 서비스의 제공이 가능하게 된다. 그러나 많은 수의 센서 노드들을 사용자가 설치하고 관리하는 것은 상당한 불편함을 따르게 할 뿐만 아니라 현재까지 센서 노드들의 서비스 표준이 정해지지 않았기 때문에 서비스 발견 및 이용에 어려움이 있다. 따라서 본 논문은 대표적인 서비스 발견 미들웨어중 하나인 UPnP(Universal Plug and Play)를 이용하여 스마트 홈 환경에서 사용자가 동적으로 센서가 제공하는 서비스를 발견하고 이용할 수 있는 기법에 대해 제안하고 구현 및 시뮬레이션한다.

1. 서 론

우리 사회는 머지않아 유비쿼터스 환경을 구축하기 위한 다양한 기술들이 삶에 접목되어 생활 속의 편리함을 도모할 수 있게 될 것이다. 그중에서도 유비쿼터스 센서 네트워크(Ubiquitous Sensor Network)는 군사 및 과학 분야뿐만 아니라 스마트 홈 환경과 같은 다양한 분야에 적용하기 위한 연구들이 활발히 진행되고 있다. 앞으로의 스마트 홈은 저가이며 크기가 작은 초소형 센서 노드들이 가정 내부 구석구석에 내장되어 인터넷망과 연동하여 현재보다 더 높은 수준의 지능적인 서비스를 사용자에게 제공할 것이다. 또한 사용자간의 우선순위(Priority)를 두어 미리 Profiling된 값을 통하여 가족 구성원이 원하는 환경을 설정할 수 있는 등의 기능도 제공해야 한다. 이러한 센서를 응용한 User-Centric Smart-Home Environment를 구성하기 위해서는 사용자의 특별한 장치 설정 등의 과정 없이 시스템이 제공하는 서비스에 아주 쉽게 접근할 수 있도록 해야 한다. 하지만 아직까지 센서 노드상에서의 서비스 표준이 정해지지 않았기 때문에 개발자들은 센서 노드에 종속적인 프로그램을 개발해야 할 뿐 아니라 많은 수의 센서 노드를 사용자가 설치하고 설정 및 관리하는 것에는 큰 어려움이 따르게 될 것이다. 이러한 문제점을 해결하기 위한 대안으로 현재 대표적인 서비스 발견 미들웨어중 하나인 UPnP를 이용하여 동적인 센서들의 서비스를 발견하고 인지가 가능하도록 하며 동시에 사용자간의 우선 순위를 통한 동적인 환경을 제공하도록 한다. 본 논문의 구성은 다음과 같다. 2절에서는 본 논문을 위한 관련 연구를 설명하고 3절에서는 본 논문에서 제안한 센서 서비스의 동적인 발견 기법에 대해 기술한다. 4절에서는 제안한 시스템의 전체 구조를 설계하고 이를 구현한다. 또한 스마트 홈에 적용시키기 위한 데모 시뮬레이션을 보인다. 5절에서는 결론을 내리고 향후 과제에 대해 간략히 소개한다.

* 본 연구는 과학기술부 과학재단 목적기초연구(R01-2005-000-10267-0) 지원으로 수행되었음.

2. 관련 연구

본 논문에서는 가전기기와 같은 장비들이 제공하는 서비스들을 동적으로 발견하고 제어하기 위한 미들웨어로 마이크로소프트사를 중심으로 한 주요 업체들이 1999년에 발표한 UPnP (Universal Plug and Play) 미들웨어를 이용한다. UPnP는 가정이나 작은 사무실과 같은 소규모 네트워크 환경에서 사용자의 특별한 설정이나 작업 없이 전원에 연결만 함으로써 쉽게 장비간의 연결이나 서비스들을 이용할 수 있는 표준화된 방법을 제공한다. 그리고 장비간의 일대일 통신을 기반으로 하고 현존하는 인터넷 표준 프로토콜(TCP/IP, UDP, HTTP, XML 등)들을 이용하여 그 구조를 정의한다. [1][2] 또한 UPnP는 여러 장비들에 독립적인 인터페이스를 제공하기 때문에 개발에도 용이한 여러 장점을 지닌다. UPnP는 크게 서비스를 제공하는 장치인 Controlled Device와 그 서비스를 제어하는 Control Point 이렇게 두 가지로 나뉜다. 그림 1은 UPnP 장치들이 네트워크로 구성되었을 때의 구조를 간략하게 나타

낸 그림이다. 기본적으로 UPnP Control Point와 Controlled Device는 일대일 통신을 기반으로 하게 되나 하나의 Control Point는 여러 Controlled Device를 제어할 수 있음을 나타낸다. 그림 2는 Control Point와 Controlled Device간의 서비스 발견 및 제어를 나타내는 것으로 각 단계를 살펴보면 다음과 같다.

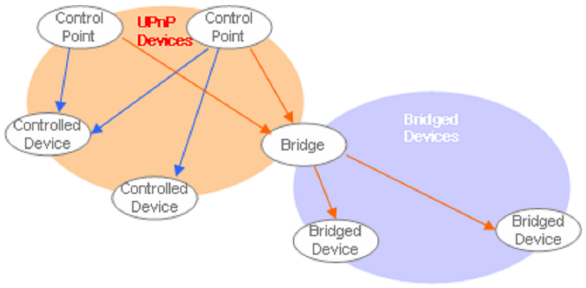

그림 1. UPnP 네트워크 구조

첫 번째 Addressing은 모든 Controlled Device와 Control Point가 DHCP 서버나 혹은 Auto IP를 통해 장치 자신의 IP를 획득하게 되는 단계이다. 이를 통해 장치들의 IP 통신이 가능하게 되면 두 번째로 Discovery단계이다. Controlled Device는 자신이 네트워크에 연결되었음을 특정 포트를 통해 Multicast하게 되고 Control Point는 해당 포트에 대기를 하여 장치들을 발견하게 된다. 이때 사용되는 프로토콜이 SSDP(Simple Service Discovery Protocol)이며 이로 인해 장치의 서비스 발견이 가능해진다. 세 번째는 Description단계이다. 모든 Controlled Device는 장치 자신의 설명 및 서비스의 종류를 XML 형태로 가지고 있어서 장치를 발견한 직후에는 이 문서를 Control Point에게 전송함으로써 Control Point는 장치가 제공하는 서비스를 인지 및 제어가 가능하게 된다. 마지막 단계로 Control 및 Event 처리 단계이다. Control Point는 SOAP(Simple Object Access Protocol)를 이용하여 Controlled Device에게 제어 명령을 내리며 Controlled Device는 GENA(Generic Event Notification Architecture)를 이용하여 자신의 상태정보를 Control Point에게 전달하게 된다.

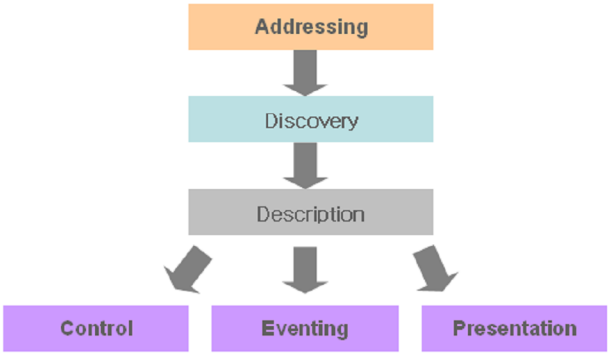

그림 2. UPnP의 네트워킹 동작

3. 동적인 센서 서비스 발견 기법

3.1 개요

현재 센서 네트워크와 관련한 다양한 연구가 활발히 진행되고 있지만 서비스 제공의 관점에서 타 네트워크와의 연동을 고려하여 동적으로 센서 서비스를 발견하고 이용하는 표준은 아직까지 구체화되지 않고 있다. 따라서 본 논문에서는 UPnP의 핵심인 서비스 발견 기능을 이용하여 스마트 홈 환경에서의 센서 서비스를 동적으로 발견하고 이를 인터넷망과 연동하여 사용자가 손쉽게 이용할 수 있는 기법을 제안한다. 먼저 제안한 기법을 위해서 고려해야 할 사항이 크게 두 가지로 나뉜다. 첫 번째는 센서 네트워크가 가지는 특성을 고려하는 것이다. 센서 네트워크는 일반적으로 비슷한 기능을 지닌 다수의 노드들로 구성이 되어 있다. 즉 모든 센서 노드의 데이터를 개별적으로 유지하고 이를 관리할 필요성이 없다. 따라서 이들이 가지는 서비스들을 잘 정의하여 UPnP Device(Controlled Device, Control Point)들 내에서 테이블로 유지하면 된다. 두 번째는 센서 서비스의 동적인 발견을 위해서 UPnP Device에 새로운 기능을 추가하여야 한다. 그림 3은 본 논문의 이러한 동적인 센서 서비스 발견 기법을 제안한 구조이다. 센서 노드와의 RF통신 및 IP 통신이 모두 가능한 장치가 Controlled Device가 되며 이 장치를 특별히 UPnP Sensor Device라고 부른다. UPnP Sensor Device는 센서 노드들의 데이터를 주기적으로 통합하여 이를 서비스 테이블로 유지한다. 그리고 UPnP Control Point는 UPnP Sensor Device가 네트워크에 접속되면 자동으로 주기적인 Action Invoking을 하는 새로운 기능을 추가하여 센서 데이터의 모니터링이 가능하며 해당 센서 서비스를 이용할 수 있게 된다.

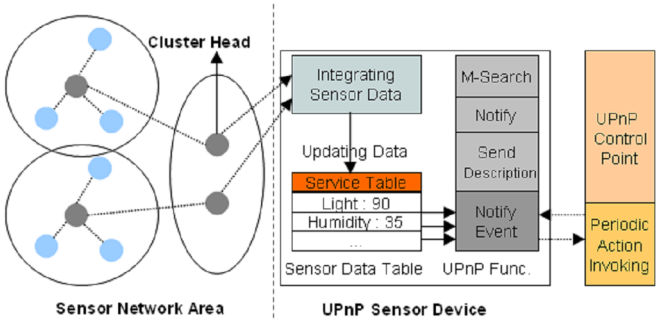

그림 3. Dynamic Sensor Service Discovery 구조

3.2 센서 서비스 정의

위에서 언급하였듯이 일반적으로 센서들은 동일한 종류의 많은 센서들을 사용하기 때문에 센서 노드마다 일일이 장치 설명 및 서비스들을 가지는 것은 비효율적이다. 따라서 이를 해결하기 위해서는 센서 노드들이 가지는 각각의 서비스들을 정의해야 한다. 표 1은 센서가 제공하는 서비스들을 최소한으로 정의한 테이블이다. 각각의 필드를 살펴보면 다음과 같다. 서비스 코드는 서비스 고유의 번호가 되어 센서가 가지는 서비스를 분류할 수 있게 되며 서비스 이름은 해당 서비스 코드의 이름이 된다. 최소값과 최대값은 센서가 획득한 데이터가 해당 상황에서 적절하지 못한 경우를 두어 그 값을 상회하거나 미치지 못할 경우 사용자에

게 경고를 할 수 있는 필드로 사용된다. 이와 같이 스마트 홈에서 사용될 센서들의 종류를 미리 정의를 하여 하나의 파일로 저장하고 UPnP Sensor Device 및 Control Point가 읽어 들여 테이블을 유지하게 되면 센서 노드들은 자신들의 서비스 코드를 포함한 데이터를 전송하게 되며 테이블 내에 코드 값과 일치여부를 조사하여 서비스를 알 수 있게 된다.

Service Code	Service Name	Min Value	Max Value
0x00	Temperature	-10	50
0x01	Humidity	0	100
0x02	Light	0	100
0x03	Vibration	10	70
•	•	•	•

표 1. 센서 서비스 테이블

3.3 센서 노드에서의 패킷 구조

센서 노드는 많은 부분에서 제한된 리소스로 동작하고 다수의 센서 노드들이 동시에 데이터를 전송하기 때문에 한 번에 많은 양의 데이터를 보낼 수 없다. 따라서 센서 노드상에서의 패킷 구조는 적용되는 상황에 맞도록 고려되어야 한다. 본 논문에서는 스마트 홈 환경 하에서 동작하는 센서 노드의 패킷 구조를 표 2와 같이 정의하였는데 패킷의 전체 크기는 5바이트로 표현이 가능하다. 각각의 필드를 살펴보면 센서 노드 개별적으로 인식하기 위한 아이디 필드, 서비스를 정의하기 위한 서비스 코드 필드, 그리고 확장성을 위한 두 바이트의 예약 필드, 마지막으로 센서 데이터를 위한 한 바이트로 전체 5바이트의 구조이다. 실제 구현에서는 예약 필드 두 바이트는 고려하지 않았다.

Sensor ID	Service Code	R1	R2	Sensor Data
0x01	0x03	•	•	0x60

표 2. 센서 데이터 패킷 구조

3.4 센서 데이터의 전송

UPnP Device는 자체적으로 이벤트가 발생하지 않는 한 UPnP Control Point에서 Action Invoking을 하게 되면 정보를 제공하게 된다. 이러한 이벤트 방식은 센서 노드와 같이 실시간 데이터 전송을 필요로 하는 방식에는 맞지 않다. 따라서 센서 노드들의 정보들을 실시간으로 전송하기 위하여 UPnP Control Point에서 UPnP Sensor Device를 발견하게 되면 주기적으로 Action Invoking을 일으키는 루틴을 추가함으로써 센서가 제공하는 데이터를 일정한 시간 간격으로 받을 수 있게 한다. 그림 5는 Control Point에서 UPnP Sensor Device를 발견하고 주기적인 Action Invoking을 하여 센서 데이터를 전송받는 흐름을 나타낸 것이다.

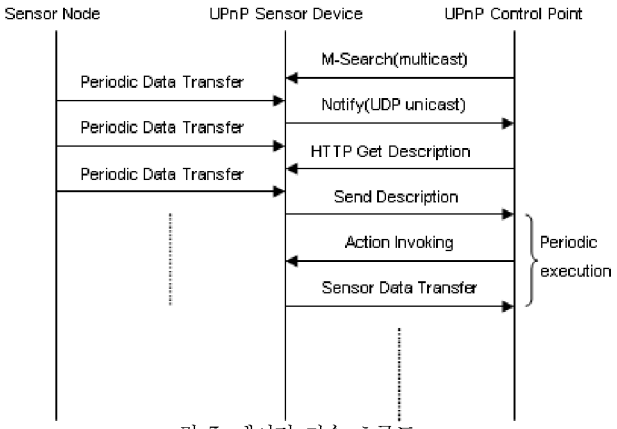

그림 5. 데이터 전송 흐름도

3.5 User Priority Profiling

본 논문에서는 스마트 홈에서의 구성원들 각자 자신들이 원하는 환경을 설정하고 지능적인 서비스를 제공받기 위하여 사용자간에 우선순위를 두는 방법을 사용하였다. 개인이 생활하는 공간은 자신의 취향에 맞게 설정을 할 수 있고 거실과 같이 주로 가족 구성원들 대부분이 함께 시간을 보내는 장소에서는 동일한 서비스를 우선 순위가 높은 구성원이 사용에 권한이 있도록 한다. 이 프로파일은 동적으로 추가 및 삭제가 가능하도록 한다. 이 우선순위 설정 기능은 UPnP Control Point에 하나의 함수로 추가 구현하여 가족 구성원 개개인의 모바일 단말기를 통해 동적인 환경을 구축할 수 있다.

3.6 시스템 전체 구조

그림 4는 본 논문에서 제안하는 시스템의 전체 구조이다. 센서 노드들의 데이터를 통합하고 서비스 테이블을 유지하며 Control Point에게 전송하는 역할을 하는 UPnP Sensor Device, 이를 제어하는 UPnP Control Point, 그리고 센서 노드 이렇게 크게 세 부분으로 나뉘게 된다. 그림에서 알 수 있듯이 주변의 여러 센서 노드들과 RF통신을 하는 장치는 IP통신이 가능한 UPnP Sensor Device이다. 이를 통해 여러 센서 노드들의 정보를 획득, 제어가 가능하게 되며 Control Point를 통해 사용자는 센서 서비스를 발견하고 제어를 할 수 있다.

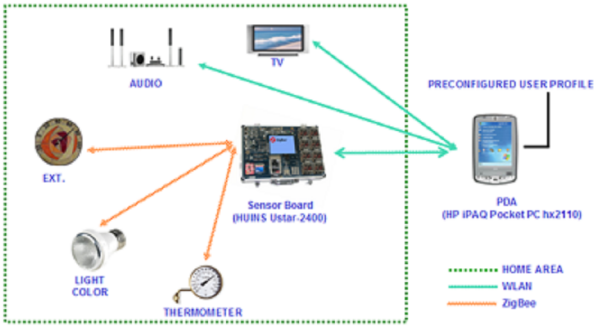

그림 4. 전체 시스템 구성도

4. 구 현

4.1 시스템 하드웨어 구성

본 논문에서 사용된 하드웨어를 살펴보면 다음과 같다. 센서 노드는 micaz와 호환이 되는 USS-2400 이라는 노드를 사용한다. 이 노드는 Atmega 128L Low Power 8-bit Microcontroller과 ZigBee RF모듈이 장착되어 무선 송/수신이 가능하며 TinyOS가 탑재되어 있다. 이 센서 노드들과 통신하며 Control Point와 통신을 하는 UPnP Sensor Device는 PXA-255(400MHz) CPU와 32Mbytes Flash Memory 및 128Mbytes의 SDRAM을 장착한 보드를 사용한다. 이 보드는 센서 노드들과 통신을 위한 USS-2400 노드가 함께 장착되어 있으며 UART로 보드와 시리얼 통신을 하여 센서 데이터의 송/수신이 가능하다. 이 보드에서 사용된 운영체제는 Embedded Linux이며 커널 버전은 2.4.22이다. Control Point를 위한 장비로는 PDA 단말기인 IPAQ hx2110이 사용되었다. IPAQ hx2110은 PXA270 312MHz CPU와 64MBytes의 ROM과 RAM이 장착되어 있으며 사용된 운영체제는 Pocket PC 2003이다. 위 하드웨어들은 UPnP 기능을 처리하기에 충분한 성능으로 구성되어 있다.

4.2 데모 시스템 구현

그림 6은 본 논문에서 제안하는 시스템 전체 구조를 스마트 홈 환경으로 시뮬레이션 하기 위한 용도로 윈도우 플랫폼에서 구현한 오디오 플레이어 및 TV 프로그램이다. 이들은 독립적으로 동작하는 프로그램으로 개발하였으며 각각의 장치 정보를 가지는 UPnP Device들을 생성하여 프로세스 간 통신을 하게 된다. 그리고 그림 7은 스마트 홈 환경을 시뮬레이션 하기 위하여 위와 마찬가지로 윈도우 플랫폼 상에서 개발한 데모용 프로그램이며 실제 데모를 수행한 모습을 나타낸 것이다. 이 프로그램은 UPnP Device(오디오 플레이어, TV, Sensor Device)들과 프로세스 간 통신을 하여 Control Point에 의해 이벤트가 발생하면 그 정보들을 이용해 화면상으로 나타내는 역할을 한다.

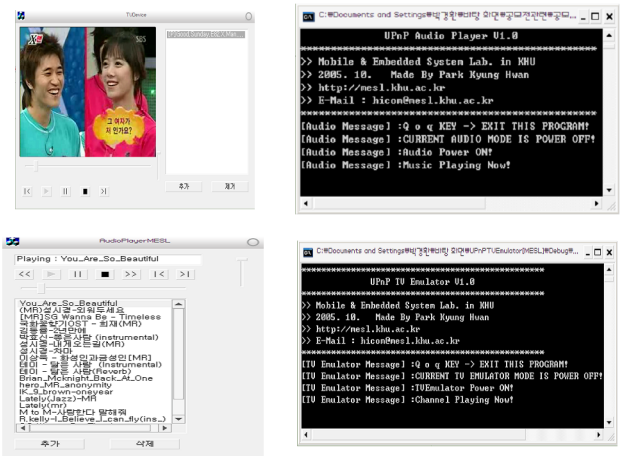

그림 6. UPnP 오디오 플레이어, TV

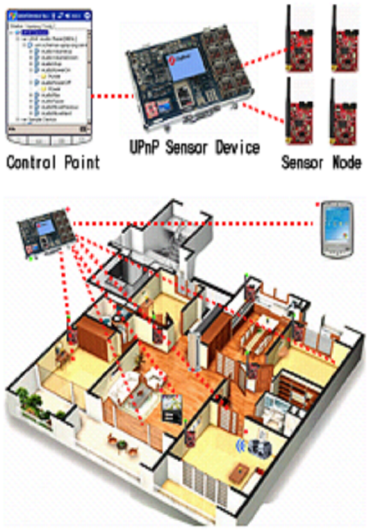

그림 7. 시뮬레이션 실행 화면

5. 결 론

본 논문은 스마트 홈 환경에서 서비스 발견 미들웨어로 대표적인 UPnP를 이용하여 동적인 센서 서비스 발견에 대한 기법을 제안하고 구현하였다. 이 기법을 이용하면 사용자는 센서 노드를 가정에 쉽게 설치할 수 있을 뿐 아니라 센서가 제공하는 서비스 또한 편리하게 이용할 수 있을 것이고 센서 노드 프로그램 개발자 입장에서도 플랫폼 독립적인 프로그래밍이 가능하다는 점에서 개발의 효율성을 가져올 것이다. 또한 이 기법은 센서 네트워크와 인터넷 망을 연동하는 하나의 방법을 제시하였기 때문에 타 네트워크(HomePNA, IEEE1394, PLC 등)와의 연동에도 쉽게 적용이 가능할 것이라 생각된다. 향후 과제로는 이러한 타 네트워크와의 효율적인 연동이 가능한 구조를 연구 및 개발해야 하겠다.

참 고 문 헌

[1] UPnP Forum, http://www.upnp.org

[2] UPnP Device Architecture Reference Specification Version 1.0, Microsoft Corporation, June 2000, http://www.upnp.org

[3] C. Lee and S. Helal, "Protocols for service discovery in dynamic and mobile networks", International Journal of Computer Research, vol. 11, no. 1, pp. 1-12, 2002.

[4] Sergio Marti, Venky Krishnan, "Carmen: A Dynamic Service Discovery Architecture", HP Laboratories Palo Alto, HPL-2002-257, September, 2002

[5] Yvonne Gsottberger, Xiaolei Shi, Guido Stromberg, Thomas F. Sturm, and Werner Weber, "Embedding Low-Cost Wireless Sensors into Universal Plug and Play Environments", EWSN 2004, LNCS 2920, pp. 291-306, 2004