

## 특 집

# IEEE1394 버스 상에서의 IP(Internet Protocol) 기반 홈 네트워크 기술

김 재 화, 나 일 주, 조 재 문

삼성전자 DM 연구소

### I. AV 홈 네트워크의 개요

홈 네트워크라는 단어는 인터넷이 보급되는 광범위하게 보급되는 시점과 맞물려서 가정내의 모든 기기들을 네트워크를 통해 유기적으로 연결시키고자 하는 개념으로부터 출발한 개념이다. 기존에 지칭되던 홈 네트워크에서의 시나리오는 외부 기기에서 가정내의 기기들을 켜고 끄고 또는 원격 지침과 같이 상태 점검 등의 역할을 수행하는 것과 같은 단순 제어에 초점이 맞추어져 있었다.

그러나 현재의 홈 네트워크 환경은 가정 내 가전기기들이 고도로 지능화되고 다기능화 됨에 따라 기존의 단순 제어 중심의 네트워크 모델로는 기기간의 유기적 관계를 충분히 설정할 수 없는 시점에 이르렀다. 특히 가정내의 환경의 주요 변화로, PC의 광범위한 보급 및 디지털 방송의 개시에 따라 고용량 고품질의 디지털 콘텐츠를 저장/재생/전송하는 기기들이 가정내의 주요 가전 기기로 자리잡고 있으며, 외부적 환경의 변화로는 인터넷의 인프라의 폭넓은 구축에 따라 가정내의 로컬 네트워크가 외부 인트라 넷의 하나의 부분으로 인식되기 시작했으며 다양한 종류의 멀티미디어 콘텐츠들을 쉽게 얻을 수 있게 되었다.

이와 같은 환경의 변화를 수용하기 위해서는 기존의 단순 제어에 치중하던 홈 네트워크의 모델보다는 멀티미디어 콘텐츠의 유기적 통합을 핵심으로 하는 AV 기기 중심의 홈 네트워크 모델의 수립이 필요하다. 이에 본 고에서는 기존의 단순 제어 중심의 홈 네트워크 모델과 대비되는 개

념으로 AV 콘텐츠 중심의 가정 내 네트워크 모델을 'AV 홈 네트워크'라 지칭하고자 한다.

AV 콘텐츠의 전송은 AV 홈 네트워크를 구성하는 핵심적인 기능을 담당한다. 디지털 방송의 콘텐츠 양식인 MPEG2-TS의 경우 HD(High Definition)급 영상을 전송하는 데에는 20Mbps 이상의 전송 대역폭을 요구하며 디지털 캠코더의 데이터 양식인 DV의 경우에도 25Mbps 이상의 대역폭을 보장해 주어야만 실시간 전송이 가능하다. IEEE1394<sup>[1]</sup> 인터페이스는 현재 규격으로 최대 400Mbps의 전송속도를 보장하면서도 실시간 전송의 QoS를 보장하여 주기 때문에 실시간 데이터의 전송을 생명으로 하는 AV 홈 네트워크를 구성하기에 유망한 인터페이스라 할 수 있다. 이뿐만 아니라 1394 인터페이스는 TCP/IP 통신을 지원하기 위한 프로토콜 스택이 RFC 2734(IP over 1394)<sup>[2]</sup>로 규격화 되어있어 기존의 인터넷 인프라와의 유연한 연동이 가능하다는 장점을 지닌다.

본 고에서는 IEEE1394로 연결된 AV 기기간의 홈 네트워크를 구성하는 프로토콜인 IPHN(IP-based Home Network) 표준<sup>[3]</sup>에 의한 AV 홈 네트워크의 구조에 대해 설명하고자 한다. IPHN은 1394 네트워크 상에서 HTTP/TCP/IP 프로토콜을 통해 AV 홈 네트워크를 구성하는 미들웨어로서 국내 1394 포럼을 중심으로 국내의 다수 기업들이 참여하여 제정한 홈 네트워크 표준안이다. 여기서 미들웨어란 기존의 HTTP/TCP/IP 등과 같은 프로토콜을 하위로 하여 상위 어플리케이션을 연결시켜주는 또 다른 프로토콜 계층을 의미한다.

지금까지 제안된 홈 네트워크의 미들웨어로서는 마이크로소프트의 UPnP(Universal Plug and Play)<sup>[4]</sup>, 소니를 비롯한 일본 및 유럽 8개사의 HAVi(Home AV Interoperability)<sup>[5]</sup>, 선 마이크로시스템의 JINI 등이 있으며 국내에서는 삼성전자에 의해 Home Wide Web<sup>[6]</sup>이 제안된 바 있다. HAVi를 제외한 대부분의 미들웨어들은 Inter Networking을 위해 TCP/IP 프로토콜을 하부에 사용하고 있으며, HAVi는 IEEE1394를 기반으로 설계된 별도의 프로토콜 스택을 사용한다.

이러한 미들웨어들의 특징을 살펴보면, UPnP는 Device Architecture 규격에서 IP 주소 할당, 기기의 기술(Description), 기기의 발견(Discovery), 기기의 제어(Control), 이벤트 처리(Eventing), 그리고 제시(Presentation) 메커니즘의 큰 틀을 제공하고, 각 워킹그룹(WG: working group)에서는 기기들의 형태(Type)와 해당 기기들에 적용될 서비스(Service)들을 정의하여 각 기능들을 XML(eXtended Markup Language) 문서로 표현하여 HTTP 프로토콜을 이용하는 구조를 정의한다. 한편 JINI는 JAVA application을 분산 컴퓨팅 환경에서 수행하기 위해 Java Virtual Machine, Look Up Server, Proxy라는 개념을 사용한다. UPnP와 JINI는 주로 PC 및 PC 주변기기들 사이의 네트워킹을 위해 고안되었다. 디지털 가전 기기간 네트워킹 기술로는 HWW와 HAVi가 있으며 모두 IEEE1394 인터페이스가 제공하는 핫 플러그이나 실시간 데이터 전송기능을 사용할 수 있도록 설계되었다. HWW은 UPnP와 유사하나 1394기술에 기반을 둔 Device Discovery, Automatic IP address Resolution, Home Network Directory 등의 기술을 제안하고 있다. HAVi 또한 IEEE1394기술에 그 바탕을 두고 있으며 기본 아키텍처는 CORBA(Common Object Request Broker Architecture)와 매우 유사하다.

기존 미들웨어 솔루션들에 비하여 IPHN은

HTTP/TCP/IP 프로토콜을 직접적으로 적용하여 별도의 프로토콜의 정의를 최소화한다는 전제 하에 표준화 작업이 이루어졌다. 이에 따라 IPHN은 기존의 웹브라우저 및 웹서버만으로도 구현 가능하여 미들웨어 규격화에서 가장 큰 문제로 제기되는 기기간의 호환성 문제를 효과적으로 해결하고 구현 코스트 측면에 있어서도 경쟁력 있게 대처할 수 있는 가능성을 열어 두었다. 이러한 IPHN의 구체적인 구조와 동작 메커니즘에 대하여 II장에 상세히 기술하고, III장에 구현 예제를 제시한 후, 향후 보완되거나 고려해야 할 점들을 IV장에서 지적하고자 한다.

## II. IPHN 표준

IPHN은 삼성전자를 포함한 1394 기기 제조자들의 연합인 1394 포럼의 이름으로 2001년 6월에 제정된 AV 홈 네트워크 표준이다. IPHN은 기본적으로 HTTP/TCP/IP 프로토콜을 이용한 미들웨어 프로토콜이라고 할 수 있다. IPHN과 UPnP는 모두 TCP/IP 기반의 기술이라는 점에서는 유사하다고 할 수 있으나 UPnP의 경우는 SOAP(Simple Object Access Protocol), GENA(general Event Notification Protocol), SSDP(Simple Service Description Protocol) 등과 같은 HTTP 확장 프로토콜을 새로이 정의하여 홈 네트워크에서 요구되는 기기대 기기(Device-to-Device) 제어 메커니즘을 주로 정의하고 있다. 이에 비하여 IPHN은 사용자 대 기기(User-to-Device) 메커니즘을 주요 대상으로 하여 정의되었으며 기존의 웹브라우징을 하듯이 기기를 제어하도록 하여 사용자의 편의성과 기기의 호환성을 극대화하는 방향으로 구성되었다.

이러한 IPHN에 대하여 본 절에서는 IPHN에서 정의하는 기기들이 어떠한 구조로 네트워크를 구성하고, 기기를 제어하고, 기기의 정보를 얻

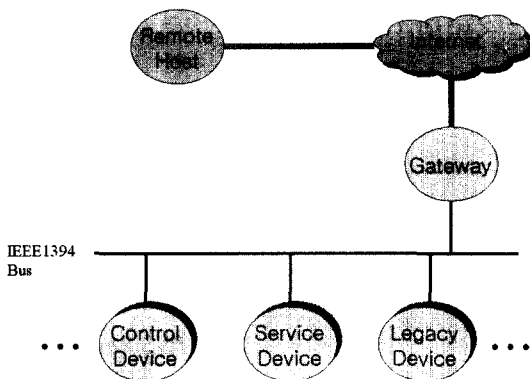
어오는지에 대하여 기술하고자 한다.

### 1. 네트워크 모델

〈그림 1〉에 IPHN을 구성하는 기기들의 연결 구조를 나타내었다. 각 기기들은 IPHN내에서의 기능에 따라 control device, service device, legacy device, 그리고 gateway device로 구분되며, IPHN 표준에서는 이들을 단순히 C\_dev (control device), S\_dev(service device), L\_dev(legacy device)로 각각 표시하고 있다.

C\_dev는 사용자 인터페이스를 제공할 수 있는 출력장치를 가지고 있어서 네트워크 내의 각 기기들을 제어할 수 있는 기기를 의미하며, S\_dev는 C\_dev의 요청에 따라 동작을 수행하는 기기를 의미한다. L\_dev는 IPHN을 지원하지 않는 기기를 의미하며 이러한 기기는 C\_dev의 proxy 기능을 이용하여 제어를 수행할 수 있도록 규정하고 있다. DTV와 같이 출력장치를 포함하는 동시에 채널의 튜닝과 같은 서비스를 제공할 수 있는 기기의 경우는 C\_dev와 S\_dev의 기능을 동시에 구현하고 있을 수 있다. IPHN의 규격화 과정에서는 DTV나 PC와 같이, 모니터 등의 출력 장치를 포함하고 웹서버 등을 구동 할 수 있을 정도의 높은 프로세싱 파워를 가지는 기기들이 C\_dev와 S\_dev 기능을 동시에 구현하는 기기가 될 것이라는 전제를 염두하고 진행되었다.

앞서 언급한 모든 홈 네트워크 표준들에 내포된 근본적인 패러다임은 하나의 중심되는 기기를



〈그림 1〉 IPHN 네트워크 구성도

통해 가정내의 모든 기기를 제어하고자 하는 것이며, 이러한 대표적인 홈 네트워크 중심기기의 후보들로는 DTV나 PC 등이 있다. IPHN에서도 이러한 제어 구조를 기본 전제로 하여 가정내에서 하나의 C\_dev가 홈 네트워크내의 모든 기기의 목록을 제공하고 이들 목록을 기반으로 각 기기를 제어하기 위한 인터페이스를 공급받는 구조를 가지고 있다. 여기에 더해 IPHN에서는 인터넷 너머의 외부 클라이언트를 통해서도 가정내의 기기들을 제어할 수 있도록 하고 있는데, 이러한 기능은 〈그림 1〉에 나타낸 gateway device를 통해 수행된다. IPHN에서 정의하는 gateway device는 기본적인 라우터(router)로서의 기능인 가정내의 네트워크를 외부 인터넷에 연결하여주는 역할뿐만 아니라, 외부의 제어 요구를 가정내의 IPHN기기들에 전달하고 이의 응답을 다시 외부 클라이언트에 전달하는 홈 네트워크의 접속 포인트로서의 역할을 수행한다. 이에 대한 상세한 메커니즘은 2. 2절에 기술한다.

상기 각 기기들에 대한 역할을 요약하면 다음과 같다.

- Control device : 사용자 인터페이스를 제공하여 가정 내 모든 기기를 제어한다. 각 기기들의 고유 명령들은 각 S\_dev가 제공하는 HTML 문서를 통해 얻는다.
- Service device : C\_dev의 제어 명령에 따라 기기 고유의 동작을 수행한다. C\_dev의 제어 명령은 S\_dev가 HTML 형태로 제공하는 CGI를 통해 수행된다.
- Legacy device : IPHN을 프로토콜을 지원하지 않는 기기들을 의미하며 C\_dev의 proxy 기능을 통해 제어할 수 있다. Proxy 기능을 하는 C\_dev는 AV/C CTS<sup>[7]</sup>와 같은 IPHN 프로토콜과는 별도의 제어 프로토콜을 이용하여 기기를 조작한다.
- Gateway device : 기본적으로 IP router의 역할을 수행하며, NAPT(Network Address Port Translation) 기능을 포함해야 한다. 규격에 따르면 IPHN gateway는

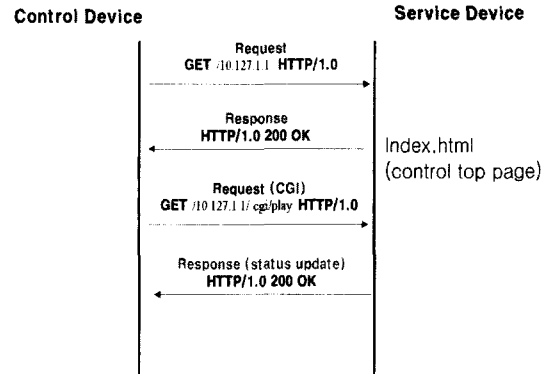
C\_dev 기능을 가져야 한다.

## 2. Control

IPHN의 기기제어는 C\_dev에서 제공되는 사용자 인터페이스(웹브라우저)를 통해 네트워크 상의 다른 기기들을 제어하는 구조이다. IPHN에서의 제어는 하나의 네트워크로 구성된 가정내 기기간의 제어시나리오인 내부제어(Internal Control) 시나리오와, 인터넷 상의 외부 클라이언트가 가정내의 기기를 제어하는 외부제어(External Control) 시나리오로 구성된다. 두 경우 모두의 제어 메커니즘은 표준적인 HTTP 프로토콜의 요구/응답(Request/Response) 구조와 CGI를 기반으로 하고 있다. 이러한 구조를 따르면 C\_dev가 각 S\_dev들의 독특한 제어구조나 기능에 대한 정보를 알 필요가 없고 해당 기기가 제공하는 HTML 문서 형태의 제어 인터페이스만을 표시하여 주기만 하면 되기 때문에, 기기의 형태라던가 제공될 서비스의 형태들을 따로 정의할 필요가 없어진다.

내부제어의 과정을 설명하면, C\_dev가 HTTP GET 메소드를 이용하여 S\_dev의 home page를 요구하면 S\_dev는 자신의 제어와 관련된 CGI 명령들로 구성된 HTML 문서를 전송한다. 이때 제공되는 제어 인터페이스를 포함하는 HTML 문서를 해당 기기의 "Control Top Page"라 부른다. C\_dev는 Control Top Page의 아이콘 등을 이용하여 CGI 호출을 수행하고 이를 수신한 S\_dev는 해당 CGI 명령을 수행한 후, 상태의 변화 등을 반영한 HTML 문서를 전송함으로써 제어가 이루어진다. 이와 같은 내부제어 시나리오에 대한 플로우를 <그림 2>에 나타내었다.

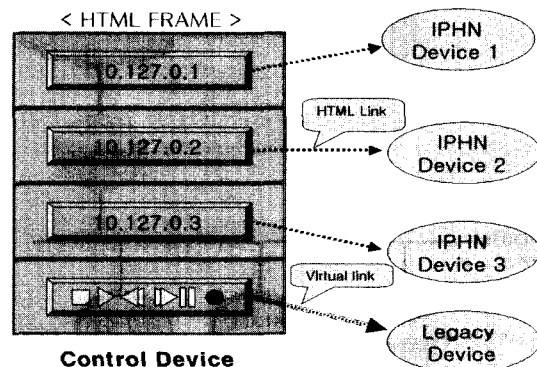
플로우에 따르면, 기기를 제어하기 위해서는 대상 기기의 IP 주소를 알아야한다는 전제가 있다. C\_dev가 S\_dev들의 IP 주소를 알아내는 메커니즘은 이후 기술할 device discovery 메커니즘에서 기술되며, 이 discovery 메커니즘을 통해 C\_dev는 로컬 네트워크 내의 모든 기기들의 IP 주소를 수집하고 이를 HTML "프레임(frame)" tag에 포함되는 링크로 만들어 제공함



<그림 2> 내부제어 시나리오의 흐름도

으로써 사용자가 네트워크 내의 모든 기기를 접속할 수 있도록 한다. 이와 같이 C\_dev가 모든 S\_dev들의 링크를 HTML frame으로 구성하여 사용자에게 제공되는 HTML 문서를 Home Network Directory Page 또는 Directory Page라고 정의한다. Directory Page는 IPHN 호환기기 뿐만 아니라 L\_dev에 대한 제어 인터페이스를 선택적으로 제공할 수 있다. 이러한 Directory Page의 형태와 연결 구조를 <그림 3>에 나타내었다.

그림에서 legacy device를 제어하는 메커니즘은 AV/C 등과 같이 HTTP 이외의 모든 프로토콜의 적용이 가능하다. Legacy device가 제공할 수 있는 기능들의 발견은 IPHN 규격밖의 일로서, 해당 프로토콜을 지원할 것인가 말 것인가의 여부는 C\_dev의 제조자의 선택 사항으로



<그림 3> Directory Page의 구조

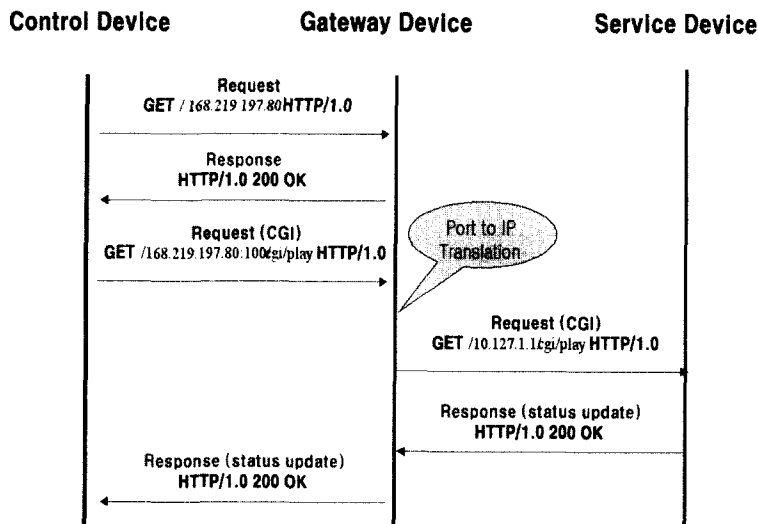
남겨 두고 있다.

외부제어의 경우는 외부클라이언트와 내부 네트워크를 매개 해주는 gateway device가 중간에 존재하게 된다. IPHN에서는 gateway device가 기본적으로 C\_dev의 기능을 가지고 있다고 전제를 하는데, 이는 gateway 기기가 Directory Page를 제공하는 역할을 수행하여야 하기 때문이다. Gateway device는 public domain의 IP 주소를 가지고 있고 이 IP 주소는 집안의 로컬 네트워크를 대표하는 유일한 IP 주소로서, 외부의 클라이언트는 이를 홈 네트워크의 유일한 접속 점으로 인식하여 이 주소를 접속하여 Directory Page를 얻고 이를 통해 집안 내의 기기들을 제어하게 된다.

외부제어에서의 중요한 문제는 HTML frame 내의 링크로 표현된 집안내의 기기들의 IP 주소가 private IP 주소라는데 있다. 외부 클라이언트가 gateway를 통해 얻어온 Directory Page의 링크는 private IP로 구성되어 있으므로 이를 직접 외부 클라이언트 기기에 제공하게 되면, 외부 클라이언트 사용자가 frame 내의 링크를 클릭한다 해도 해당 기기의 목적지 주소를 결정하지 못하게 된다. 이를 해결하기 위해 IPHN에는 NATP(Network Address Port Transla-

tion) 프로토콜을 gateway 기기가 구현하도록 규정하고 있다. Gateway device는 자신의 Directory Page를 외부에 제공할 때, 가정 내 모든 기기들의 private IP들을 자신의 public domain IP와 포트번호의 조합으로 변환하여 프레임 내의 링크를 다시 구성하여 주어야 한다. 이렇게 하면 외부의 클라이언트는 가정내 기기들을 제어하는 CGI 명령을 마치 gateway device의 특정 포트를 통해 전송하는 것과 같은 효과를 얻을 수 있고, 자연스럽게 각 포트별로 독립된 기기들이 대응되어 있는 것처럼 보이게 된다.

예를 들어, 외부 클라이언트가 168.219.197.80의 public domain의 IP 주소를 대상으로 Directory Page를 요구하면 gateway device는 포트 변환된 frame으로 구성된 Directory Page를 송신한다. 사용자는 Directory Page에 표시된 어떤 기기에 대하여 play에 해당되는 아이콘을 클릭하게 되면, IP주소 168.219.197.80에 해당 기기의 포트번호 100을 연장하여 CGI 명령을 전송한다. 이를 수신한 gateway device는 해당 포트 100번에 대응시켜놓은 10.127.1.1이라는 기기에 해당 CGI 명령을 전송한다. 전송된 명령이 수행된 후에 응답 메시지도 역의 과정을 통해 외부 클라이언트에게 전송된다. 이러한 일



〈그림 4〉 외부제어 시나리오의 흐름도

련의 과정을 <그림 4>에 나타내었다.

### 3. Description

앞서 기술한 제어구조는 C\_dev가 네트워크 내의 모든 기기의 IP 주소와 해당 Control Top Page의 URL(Uniform Resource Location)을 알고 있다는 가정으로부터 출발하고 하고 있다. IPHN에서는 이와 관련하여 IP 주소의 할당 방법, IP 주소의 위치, 그리고 Control Top Page를 얻기 위한 URL을 정의하고 있다. 또한 gateway device를 위한 별도의 레지스터들과 기기의 기본적인 제어와 이벤트 처리를 위한 레지스터들을 정의하고 있다. 각 레지스터들은 1394 트랜잭션 계층에서 정의되는 것이고, URL들은 상위 계층인 HTTP 계층에서 정의되는 사항들을 구분하여 이해할 필요가 있다.

#### 1) Unit Directory 및 CSR 레지스터

IPHN 기기는 1394 configuration ROM에 unit directory에 자신이 IPHN 기기임을 알리고 여기서 정의하는 CSR의 offset 정보를 포함하고 있어야 한다. 이를 나타내는 IPHN unit directory의 형식을 <표 1>에 나타내었다.

Unit\_Spec\_ID는 본 unit directory가 IPHN 규격을 나타내는 정보임을 나타내며 향후 IEEE

<표 1> IPHN Unit Directory의 형식

Directory length	CRC
0x12	Unit_Spec_ID
0x13	Unit_SW_Version
0x54	Unit_CSR_Base
0xF8	Adjacent_IP_Devices (optional)

RA에 의해 결정될 사항이다. 현재는 등록 진행 중에 있다. Unit\_SW\_Version은 현재 규격의 버전을 나타내며 Unit\_Spec\_ID와 함께 결정될 사항이다. Unit\_CSR\_Base가 IPHN의 동작을 결정하는 레지스터들의 시작 위치를 나타내는데, 특이한 점은 IPHN이외의 unit directory들은 relative offset 값을 나타내는 경우가 대부분인데 비하여 IPHN은 절대 offset을 사용한다는 것이다. 상기의 Unit\_CSR\_Base의 값을 offset으로 하는 영역에는 <표 2>에 나타낸 IPHN unit register들이 정의되어 있다

각 레지스터의 필드에 대한 자세한 설명은 IPHN 규격<sup>[8]</sup>을 참조하기 바란다.

#### 2) IP 주소의 할당 : HNCP(Home Network Configuration Protocol)

IPHN은 가정내의 네트워크를 대상으로 하므로 DHCP server와 같은 자동 IP 할당 기능이 제공되리라는 기대를 하기 어렵다. 따라서 IPHN에서는 별도의 사용자 조작이나 서버의 도움이 없이 모든 기기가 private domain의 IP address를 할당받도록 하는 프로토콜인 HNCP(Home Network Configuration Protocol)를 정의하고 있다. HNCP는 버스리셋이 일어난 후 버스의 안정화를 위해 1초를 기다리도록 규정하고 있다. 1초 이후에는 기본적으로 각 기기들은 10.127.1.1~10.127.255.255 구간의 임의의 IP 주소를 자신에게 할당한다. 그러나 무작위로 선정한 IP 주소는 주변의 기기들과 충돌을 일으킬 확률이 매우 높기 때문에 이를 조정할 필요가 있다. IP 주소의 충돌을 알아내는 전통적인 방법은 IP-probing 방법 즉, 자기 자신의 IP 주

<표 2> IPHN Unit 레지스터

오프셋	기 호	설 명
0x00	IP_CONTROL	기기의 IP 관련 기능의 제어 및 상태
0x04	IP_ADDRESS	기기에 할당된 IP 주소
0x08	IP_EVENT	이벤트 수신을 위한 레지스터
0x0C~0x1F	Reserved	확장을 위한 예약영역

소를 정보로 하여 ARP(Address Resolution Protocol) packet을 broadcast하는 방법이 있으나, IP 충돌이 검출된 후 새로 임의의 IP를 할당하고 이의 과정을 계속하는 하는 과정을 반복하다 보면 IP 주소의 초기화 시간이 길어질 수 있는 위험을 내포하므로 홈 네트워크와 같이 hot-plugging을 요구하는 환경에는 적합하지 않다. 이를 위해 IPHN에서는 IP-probing 대신에 deterministic 한 IP 충돌 회피방법을 정의하고 있다.

먼저 어떤 기기가 IP 충돌의 검출하기 위하여 네트워크내의 모든 기기에 대하여 <표 2>에 표시한 IP\_ADDRESS를 읽어 들인다. 만일 IP 충돌이 발생하지 않았다면 현재 설정된 각각기기의 IP 주소를 그대로 사용하도록 IP\_CONTROL 레지스터의 IE(IP Enable) 비트를 '1'로 설정한다(1394 Write 트랜잭션을 이용한다). 만일 IP 주소의 충돌이 발생했다면 충돌되는 기기들의 IP 주소를 새로이 10.127.1.1~10.127.255.255 영역의 임의의 주소로 할당하여 해당 기기들의 IP\_ADDRESS 레지스터에 써넣고 IE 비트를 '1'로 설정함으로써 IP 충돌 문제를 해결한다. 이러한 기능을 수행하는 기기를 HNCP manager 하고 하고 이러한 기기는 IP\_CONTROL 레지스터의 필드 H(HNCP manager capable) 비트가 '1'로 설정되어 있어야 한다.

네트워크 내에서 이러한 기능을 수행하는 기기는 유일해야 하는 것이 기본 조건이다. 만일 HNCP manager capable한 기기가 네트워크 내에 다수가 존재할 경우에는(다수의 HNCP manager의 존재 여부는 IP 할당 전에 모든 기기의 IP\_CONTROL 레지스터의 H 필드를 확인함으로써 알 수 있다) 각 기기의 GUID의 비트역순이 가장 큰 기기가 HNCP manager로 선정된다. GUID 값을 직접 비교하지 않고 비트역순 크기를 비교하는 이유는 GUID의 상위 비트들이 vendor ID 필드로 구성되므로 vendor ID가 높은 기기가 언제나 HNCP manager로 고정되는 것을 막기 위한 고려이다.

### 3) Description URLs(Uniform Resource Location)

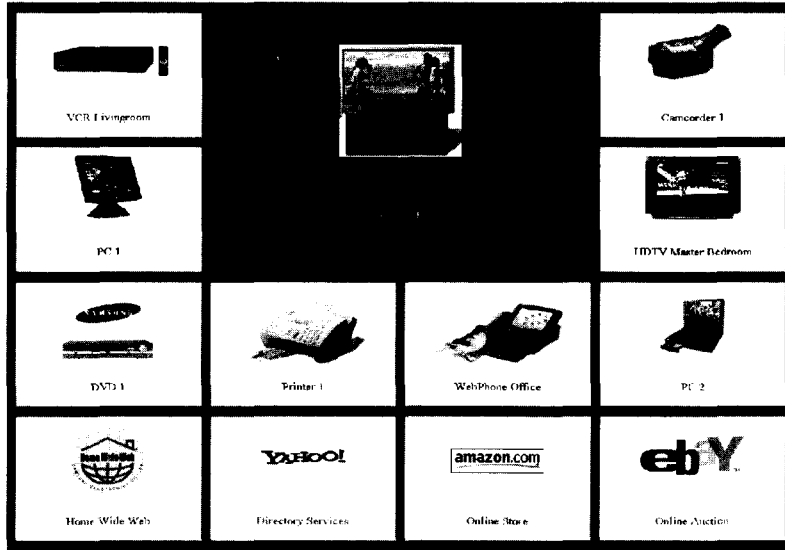
IPHN 제어구조는 앞서 기술한 바와 같이 HTML/HTTP 프로토콜을 사용한다. 따라서 C\_dev는 각 기기의 IP 주소 정보뿐만 아니라 Control Top Page를 얻을 수 있는 URL 정보를 알아야만 Directory Page를 구성하여 사용자에게 제공할 수 있다. IPHN에서는 네트워크 빠른 구성을 위하여 URL 정보를 방송하는 별도의 프로토콜을 정의하지 않고 고정된 URN(Uniform Resource Name)을 Control Top Page를 접근하도록 하고 있다.

Directory Page를 구성할 때는 frame tag를 이용하여 각 frame의 내부에 S\_dev의 URL을 링크하여야 한다. 이러한 링크로 이용될 수 있도록 S\_dev는 "/icon.htm" 페이지를 제공하여야 한다. Icon.htm 페이지의 내용물은 S\_dev 제조자의 결정사항이나, 이름에서 알 수 있듯이 frame 내에 표시할 기기의 아이콘 이미지를 포함하도록 구현하는 것을 IPHN에서는 추천하고 있다. 이 이미지는 다시 해당 기기의 Control Top Page로 링크되도록 하여 사용자는 아이콘 클릭을 통해 기기의 제어메뉴를 얻을 수 있다.

Icon.htm에 더불어 IPHN에서는 "/name.htm"의 페이지를 정의하고 있다. 이는 icon.htm이 제고하는 아이콘 이미지에 부가적으로 기기의 이름 등을 텍스트 형태로 표시하는 기능을 제공한다.

## III. 홈 네트워크 구현 예

IPHN 프로토콜로 구성된 1394 홈 네트워크의 구현 예시를 <그림 5>에 나타내었다. 예시 그림은 DTV가 홈 네트워크의 중심 기기라는 가정 하에서 거실의 DTV가 제공할 수 있는 IPHN Directory Page의 예를 보인 것이다. 이를 살펴보면, 중앙부에 가장 큰 영역에 DTV 자신의 제어 인터페이스를 접속하기 위한 아이콘을 제공하



〈그림 5〉 IPHN Directory Page의 예

고 있으며 주변에 가정 내에서 1394 버스에 연결될 수 있는 다양한 기기들의 목록을 배치해 보여주고 있다. 가장 밑에 영역에는 기기들과는 관계 없이 favorite한 인터넷 사이트들의 링크를 들어서 네트워크 게이트웨이의 기능을 수행하고 있음을 보여주고 있다.

〈그림 6〉에 특정기기의 “/icon.htm”의 구현 예를 보였다. 주어진 예시는 IPHN 규격에서 제안하는 기본 양식이다. 기본적인 “title” tag에는 기기를 설명하는 text가 들어가고 하나의 이미지 파일을 단순히 제공하고 이를 Control Top Page인 “index.htm”에 링크시켜 놓았다.

```

<HTML>
<HEAD>
  <TITLE>Device Icon</TITLE>
</HEAD>
<BODY BGCOLOR="#FFFFFF" TEXT="#000070" LINK="#0000ff"
  ALINK="#FF0000" VLINK="#007986">
  <CENTER>
    <A HREF= index.htm TARGET= _blank >
    <IMG SRC="icon.gif" BORDER=0</A>
  </CENTER>
</BODY>
</HTML>

```

〈그림 6〉 Service device의 “icon.htm” 예시

#### IV. 향후 과제

IPHN이 제공하는 홈 네트워크 구성은 기존의 TCP/IP 및 HTTP 등의 널리 알려진 프로토콜을 직접적으로 이용하기 때문에 기기간의 호환성을 위한 별도 프로토콜의 구현이 요구되지 않는다는 장점을 가지고 있다. 그러나 기존 프로토콜만을 기반기술로 하여 별도의 프로토콜의 정의를 최소화한다는 규격제정의 전제가 호환성의 문제를 효과적으로 해결하고 있음에도 불구하고 이에 따른 기능상의 제약과, 홈 네트워크 환경이라는 제한적 환경에서 야기되는 문제점들이 발생한다. 향후 규격에서 고려해야 할 사항들을 기술하면 다음과 같다.

- 1) 현재 규격에서 정의하고 있지 않고 향후 보완되어야 할 대표적인 문제점이 기기 대 기기의 제어문제이다. 기본적으로 HTML/HTTP 프로토콜은 브라우저를 통해 제공되는 GUI를 통해 사용자의 명령을 전달하는 구조이므로 디스플레이 기능을 가지지 않는 기기를 통한 제어는 현재의 제어구조로는 불가능하다. 이를 위해서는 Control Top Page에서 특정한 tag를 예약하고 이를 통해 최소한의 제어 기능



- 만이라도 제공하는 방법을 생각해 볼 수 있다.
- 2) 외부제어에 있어서 사용자의 인증 등과 관련된 보안 문제를 해결해야한다. 현재의 규격에는 별도의 인증 절차 등을 포함하고 있지 않기 때문에 임의의 외부 클라이언트들이 gateway 기기의 IP 주소를 알기만 하면 가정내의 모든 기기를 접근할 수 있는 구조이다. 이를 해결하기 위해서는 별도의 포탈 사이트를 구축하여 이 사이트를 경유하여야만 접속할 수 있도록 하는 등의 방법을 생각할 수 있다.
  - 3) 현재 규격에서는 가정 내에 gateway 기기가 유일하게 존재하고 있다는 전제를 하고 있으므로 gateway 설정에 문제를 발생시키지 않으나 다수의 gateway 기기가 존재할 경우 어떤 기기를 기본 gateway 기기로 설정하여야 하는지의 방법이 필요하다. 현재 규격에서는 IP\_CONTROL 레지스터에 R(Router Capable) 비트를 제공하고 있으나 이를 이용한 기본 라우터 설정 방법을 기술하고 있지 않다.
  - 4) 이상의 고려사항 이외에도 버스리셋 후 IP 재설정 문제, 이벤트 처리문제 등에 대하여 기기 구현자가 임의로 처리할 수 있는 문제들이 포함되어 있어 향후 규격에서 보완되어야 할 사항이라 보인다.

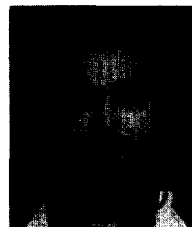
참 고 문 헌

- (1) IEEE, IEEE Standard for a High Performance Serial Bus(IEEE Std 1394-1995), Microprocessor and Microcomputer standards committee of the IEEE computer society, Dec. 1995.
- (2) IETF, RFC2734, IP over IEEE1394, Dec. 1999
- (3) 1394 Forum, Internet Protocol(IP) based Home Network over IEEE1394 Bus, Ver 0.9, Jun. 2001.
- (4) Microsoft, Universal Plug and Play Device Architecture, Ver. 1.0, Jun. 2000.
- (5) Sony, Matsushita, Philips, Thomson, Hitachi, Toshiba, Sharp, Grundig,

Specification of the home audio/video interoperability (HAVi) architecture, Jan. 2000

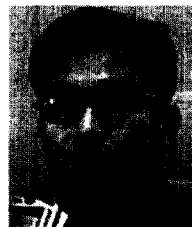
- (6) EIA, EIA-775.1, Web-enhanced DTV 1394 interface specification, Jan. 2000
- (7) 1394TA, AV/C Command Set General Specification, version 3.0.

저 자 소개



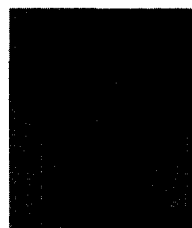
金 載 和

1996년 2월 중앙대학교 제어계측 공학과 학사, 1998년 2월 중앙대학교 제어계측학과 석사, 2001년 2월 중앙대학교 제어계측학과 박사, 2000년 12월~현재: 삼성전자 DM연구소 선임연구원, <주 관심 분야: 홈 네트워크 프로토콜, 디지털 통신 시스템, 디지털 신호처리>



羅 日 柱

1994년 2월 서울대학교 제어계측 공학과 학사, 1996년 2월 서울대학교 제어계측공학과 석사, 1996년 2월~현재: 삼성전자 DM연구소 책임연구원, <주 관심 분야: IEEE1394 전송 기술, 디지털 홈 네트워크>



趙 在 汶

1984년 2월 서울대학교 전기공학과 학사, 1986년 2월 한국과학기술원 전기 및 전자공학과 석사, 1991년 8월 한국과학기술원 전기 및 전자공학과 박사, 1986년 3월~1987년 2월: University of California, Irvine 연구원, 1989년 3월~현재: 삼성전자 DM연구소 수석연구원, <주 관심 분야: 홈 네트워크 인터페이스, DTV 시스템, 영상처리 알고리즘>