

HAVi 홈 네트워크 미들웨어를 위한 IEEE1394의 Java API 구조

박동환^o 구태연 문경덕
한국전자통신연구원
(dhpark, kutai, kdmooon)@etri.re.kr

Research on IEEE1394 Java API for HAVi home network middleware

Dong-Hwan Park^o Tai-Yeon Ku Kyeong-Deok Moon
Dept. of Office, Electronics & Telecommunication Research Institute

요 약

IEEE1394는 디지털 멀티미디어 데이터의 전송의 실시간성이 보장되고 빠른 전송속도와 동적인 네트워크 재구성이 가능한 직렬 인터페이스로 현재 대표적인 홈 네트워크 프로토콜로 인식되고 있다. IEEE1394를 기반 프로토콜로 사용하는 홈 네트워크 미들웨어인 HAVi는 디지털 AV기기간의 상호운용성을 보장하는 미들웨어이다. HAVi의 구현은 어떠한 언어를 사용하든 관계가 없으나, 미들웨어의 구조상 HAVi 응용(Havlet)의 다운로드와 실행, 메시지의 전송방식 등에서 Java언어로의 구현이 가장 적절하며, HAVi에서는 또한 응용 개발을 위한 미들웨어의 Java API와 DTV와 셋탑박스등에 탑재될 때 사용할 수 있는 HAVi L2 ui를 제공한다. 본 논문은 HAVi 미들웨어를 Java언어로 구현함에 있어 필요한 IEEE1394의 Java API와 효과적인 데이터의 전달을 위한 콜백기법과 QoS의 보장을 위한 버퍼링 기법을 제안한다.

1. 서 론

IEEE1394는 고속의 전송속도를 가지며 프로토콜 상에서의 실시간성이 보장되므로 최근 디지털 TV, 셋톱박스, mp3 플레이어, 디지털 오디오 장비등에서의 디지털 오디오/비디오 스트림의 효과적인 전송 인터페이스로 채택되고 있다[1, 2]. 이러한 특성에 동적인 네트워크 재구성이 가능하고 사용이 편리하다는 장점이 덧붙여 저서 홈 엔터테인먼트 네트워크의 중요 네트워크 프로토콜로 인식되고 있다. 현재 IEEE1394는 디지털 캠코더와 노트북에 기본으로 장착되며 일반인에게도 친숙한 인터페이스로 다가오고 있으며, 점차 그 적용범위를 확장해가고 있다.

IEEE1394 기술을 적용한 홈 네트워크 미들웨어로는 HAVi (Home Audio/Video interoperability), VHN(Versatile Home Network), HomePnP(Home Plug and Play) 등이 있으며, IEEE1394기기를 지원하기 위한 연구로 썬 마이크로 시스템즈사에서 Jini의 Surrogate 기술을 적용한 1394 Surrogate 기술을 연구중이며, 마이크로 소프트의 UPnP측에서 UPnP AV 기술에서 IEEE1394 기기를 지원하고 있다. 이중 HAVi 미들웨어가 주요 가전업체의 지지를 등에 업고 HAVi 탑재 제품을 선보이는등 활발한 움직임을 보이고 있다.

HAVi 미들웨어는 구현에 자율성을 보장하고 있지만, 미들웨어 API로는 Java API를 제공하고 있다. 이러한 Java API와 제공은 Java의 플랫폼 독립적인 특성을 고려한 것으로, 한번의 개발로 어떠한 플랫폼에서도 수행이 가능한

장점을 지닌다. 이는 디지털 가전과 휴대용 정보가전 기기의 보급으로 인해 다양한 플랫폼이 연결되는 홈 네트워크 환경에서는 필수적이다.

본 논문에서는 정보가전용 임베디드 리눅스상에서 Java를 이용한 HAVi 미들웨어의 구현시 요구되는 IEEE1394의 Java API의 구현 방법과 효율적인 데이터의 전송과 수신 기법, 그리고 동적인 네트워크 구성을 지원하는 네트워크 관리 기법, 스트림의 효율적인 처리를 위한 버퍼의 구조와 버퍼제어 기법을 제안한다.

2. 관련 연구

2.1 IEEE1394의 개요

IEEE1394는 애플사에서 고안되어 IEEE위원회에서 1995년 표준으로 채택한 고성능 직렬 버스이다. 이는 100, 200, 400Mbps의 전송 속도를 지원하며, 신뢰성 있는 전송을 보장하는 비동기 통신과 스트림 전송의 실시간성과 QoS를 보장하는 동시성 통신을 동시에 지원한다. IEEE1394 프로토콜은 하나의 버스당 63개의 노드를 연결할 수 있으며, 1024개의 버스를 연결할 수 있는 구조를 가지고 있다.

현재 IEEE1394기술은 계속해서 발전하고 있는 기술이다. 1995년 IEEE1394표준이 확정된 이후, 이를 보완한 IEEE1394a-2000기술이 발표되었으며, 3.2Gbps까지 전송 속도의 향상을 도모한 IEEE1394.b기술이 올해안에 표준안으로 확정될 예정이며 이를 지원하는 칩셋과 제품이 판매중이다. 또한 무선 1394의 지원을 위한 그룹과 네트워크

기능을 정의한 IEEE1394.1 표준이 논의되고 있다. 이렇듯 세를 확장하고 있는 IEEE1394기술은 기존의 디지털 캠코더와 노트북에서 벗어나, 디지털 TV와 DVCR, 셋탑박스, 디지털 오디오 기기에 채택이 되고 있으며 향후 거의 모든 AV가전기기의 표준 인터페이스로 장착될 것으로 보인다.

2.2 HAVi의 개요

HAVi[3, 4]는 가정의 IEEE1394네트워크를 통해 연결된 다양한 제조사와 상표의 디지털 오디오와 비디오 장치 간의 상호 운용성(Interoperability)을 제공해주는 디지털 오디오 및 비디오 관련 전자 제품의 유연한 연동을 지원하기 위한 미들웨어에 대한 산업 표준의 하나이다. 이는 Sony, Philips등의 회사들이 주축이 되어 홈 네트워크용 제품을 만들기 위해 개발한 미들웨어 소프트웨어 구조로 독자적인 프로토콜과 API를 가지고 있고 IEEE1394[5, 6]를 기반으로 한다. HAVi 구조에서는 서비스들이 소프트웨어 요소(Software Element)라 불리는 객체로 모델링 된다. 모든 소프트웨어 요소는 80bit의 SEID(Software Element Identifier)로 접근되며 네이밍 서비스인 Registry 서비스를 이용해 다른 객체를 찾을 수 있다. 또한 모든 객체는 메시지 전달을 통해 통신을 하며, 목적 객체(target object)는 SEID에 의해 결정된다. 모든 소프트웨어 요소는 상호 운용 가능한 API를 제공하며, 이 API를 이용하여 홈 네트워크 상에서 분산 어플리케이션을 구현할 수 있다. HAVi 디바이스 사이의 상호 운용성을 제공하기 위해 필요한 소프트웨어 요소에는 메시징 시스템, 레지스트리, 이벤트 관리자, 자원 관리자, DCM(Device Control Module) 관리자 등이 있다.

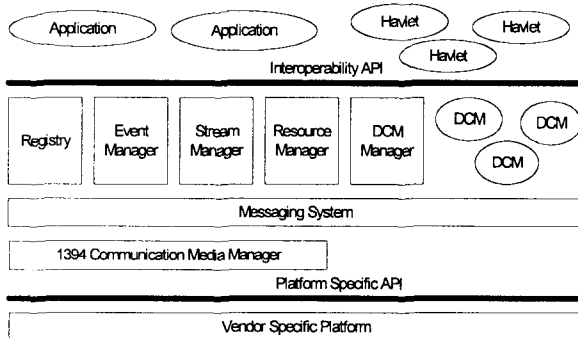


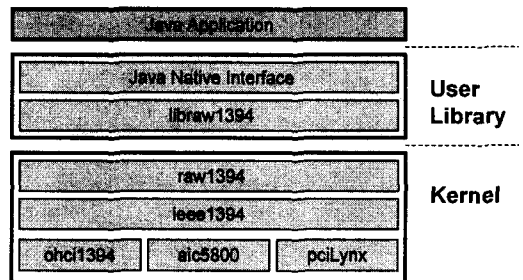
그림1은 HAVi의 시스템 구조에 관한 것으로, 1394CMM(Communication Media Manager)는 다른 소프트웨어 요소들이 IEEE1394를 통하여 비동기 통신과 등시성 통신이 가능하게 하며 메시징 시스템(Messaging System)은 소프트웨어 요소들 간의 메시지 전달을 담당한다. 이벤트 관리자(Event Manager)는 이벤트 전달 서비스를 제공하고 여기서의 이벤트는 홈 네트워크나 오브젝트의 상태 변화를 의미한다. 스트림 관리자(Stream Manager)는 각 컴포넌트 간의

AV스트림의 실시간 전송을 관리하며 레지스트리(Registry)는 홈 네트워크상의 다른 오브젝트를 찾는 디렉토리 서비스를 제공한다. DCM(Device Control Module)은 디바이스를 제어하는 소프트웨어 요소이며 DCM 관리자(DCM Manager)는 디바이스에 DCM코드를 설치하고 제거하는 기능을 한다.

3. IEEE1394의 Java API 구조

3.1 IEEE1394 리눅스 디바이스 드라이버

리눅스 디바이스 드라이버[5]는 AIC5800과 PCILynx 링크 컨트롤러 칩과 OHCI 호환 칩을 지원하기 위해 하드웨어와 인터페이스 하는 부분에 병렬적인 3개의 모듈을 구성하고 있다. 이는 해당하는 하드웨어를 직접 제어하기 위한 것으로 칩 고유의 레지스터와 인터페이스를 지원하도록 구성된다. 이 모듈의 위에 있는 IEEE1394 모듈은 특정 하드웨어에 종속적이지 않은 코드로 구성되며, 실질적인 트랜잭션 서비스와 패킷 처리등의 작업이 IEEE1394 프로토콜에 맞게 일반화되어 있다. 이와 같은 일반화된 IEEE1394 모듈을 이용하여 사용자 레벨에 단일 인터페이스를 제공하기 위해 raw1394 모듈을 만들고, 사용자 레벨에서는 최종 API로 libraw1394를 이용하여 활용하게 된다.



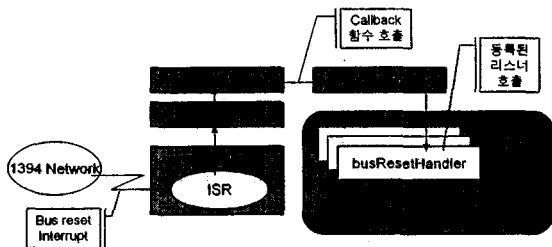
<그림 2> 리눅스 IEEE1394 라이브러리의 구조

위의 그림은 이러한 리눅스 디바이스 드라이버의 구조를 나타낸다. 사용자 레벨에서 사용할 수 있는 API는 libraw1394 라이브러리이다. IEEE1394의 Java API를 위해서는 JNI(Java Native Interface)라이브러리가 요구되며, 이는 user library로써 제공된다. 즉, Java API를 위한 JNI라이브러리, IEEE1394의 사용자 라이브러리, 커널영역의 raw1394, ieee1394, ohci1394등의 드라이버 모듈로 구성된다.

3.2 JNI (Java Native Interface) 의 구조

IEEE1394 디바이스 드라이버와 사용자 라이브러리에서는 IEEE1394 네트워크를 관리하고 데이터의 송수신을 위해 여러 종류의 콜백함수를 제공한다. 이러한 콜백함수의 호출은 IEEE1394 호스트 어댑터에서 인터럽트가 발생하였을 때, 리눅스 커널의 디바이스 드라이버의 ISR(Interrupt Service

Routine)에서 발생한 인터럽트의 처리시 호출되며, 연관된 데이터 구조를 콜백함수로 전달하며 호출하게 된다. 주된 콜백함수의 종류로는 Bus Reset Handler, FCP Handler, Tag Handler, Isochronous Handler가 존재한다. IEEE1394에서는 네트워크 기기가 연결되거나 제거될 때 Bus Reset 신호를 발생시키게 되고, 이는 네트워크에 연결된 모든 기기에 전달된다. 이에 따라 네트워크상의 모든 기기는 네트워크 상태의 변화를 인지하게 되고 정해진 루틴에 의해 네트워크를 재구성하게 된다. 이 경우에 발생하는 인터럽트가 Bus Reset 인터럽트이고 이는 호스트어댑터에서 수신되어 디바이스 드라이버의 ISR에서 처리되고 그 결과로 라이브러리의 Bus Reset Handler를 호출한다. 이렇게 네이티브 코드 영역에서 호출된 콜백함수는 Java 영역으로 전달하여야 한다. 제안된 구조에서는 JNI 영역에서 JVM(Java Virtual Machine)을 얻어와 Bus Reset 콜백을 처리할 리스너 클래스의 busResetHandler() 메소드를 직접 호출하여 데이터를 전달한다. 이러한 네이티브영역에서 Java영역으로의 invocation은 콜백이나 인터럽트를 처리하기 위해 요구되는 불필요한 thread의 생성을 줄여 자원을 효율적으로 사용할 수 있도록 하며, 지연없는 인터럽트의 전달이 가능하다.



<그림 3> 등록된 자바리스너 객체 호출 방식

효율적인 Isochronous 통신을 위해 제안하는 구조의 Java API에서는 네이티브 영역에서 사용될 버퍼의 구조와 Java 영역에서 사용할 버퍼의 구조를 따로 정의한다. 이 이유는 네이티브 영역에서 자바영역으로의 invocation이 많은 시간이 걸리기 때문에 적절한 크기의 버퍼를 네이티브 영역에 생성, 관리를 하고 인터럽트의 watermark를 지정하여 isochronous 리스너의 호출 주기를 조정할 수 있도록 한다. 또한 Java 영역에서의 버퍼는 Java 응용에서 처리 지연으로 인해 실시간성을 저해요소를 해결하기 위한 것으로, 네이티브에서 리스너로 전달되는 버퍼의 크기와 응용에서 처리되는 데이터의 크기를 비교하여 적절히 등시성 특성을 보장할 수 있도록 증대하는 역할을 수행한다. 이는 자바의 연산, 수행지연으로 인해 발생하는 지연 시간을 해소시키는 목적으로 사용된다.

4. 구현 및 고찰

제안된 HAVi 미들웨어를 위한 IEEE1394 디바이스 드라이버와 Java API는 정보가전 및 임베디드 시스템을 위한 운영체제인 Qplus 에서 구현되었다. IEEE1394 디바이스

드라이버와 사용자 라이브러리는 IEEE1394 네트워크를 관리하고 특정 주소영역의 등록 및 해제, 특정 메시지에 대한 응답등 미비한 기능을 보완하였으며, 또한 등시성 전송의 효율적인 데이터 처리를 위해 DMA기법을 사용하는 특정 드라이버를 사용할 수 있도록 하였다.

기존의 Java를 이용한 HAVi 미들웨어의 구현과 IEEE1394 응용의 개발에서는 Thread를 기반으로한 polling 방식의 리스너를 주로 사용하였기 때문에 자원의 낭비가 많았으며, 등시성 리스너의 경우에는 리스너 Thread가 polling을 할때만 데이터를 넘겨받아, 데이터의 polling 주기에 따른 지연시간이 큰 단점이 있었다. 따라서, 제안된 Java API는 네이티브영역에서 자바영역의 메소드를 직접 호출하여, 인터럽트나 이벤트가 발생할 때 마다 효과적인 데이터의 처리가 가능하며 DMA를 이용한 특정 드라이버의 사용이 가능하기 때문에 CPU의 간섭없이 데이터의 전송이 가능하므로 더욱 빠른 데이터의 전송이 가능하다.

5. 결론

초고속 통신망과 디지털 가전기기의 보급에 힘입어 최근 인터넷을 기반기술로 적용한 홈 오토메이션과 가전기기를 이용한 인터넷사용 등의 융합기술로써의 홈 네트워크가 현실화 되고 있다. 하지만 궁극적인 홈 네트워크의 요구사항과 서비스의 창출은 홈 엔터테인먼트 네트워크에서 이루어 질것으로 보이며, 이러한 홈 엔터테인먼트 네트워크 미들웨어로써 HAVi와 플랫폼 독립적인 특성을 제공하는 자바언어의 지원은 불가결한 것으로 여겨진다. 본 논문은 디지털 가전기기의 내장형 운영체제로 급부상하고 있는 임베디드 리눅스를 기반으로 IEEE1394인터페이스를 내장한 기기에 1394의 자바 API를 제공하고 효율적인 통신이 이루어지기 위한 콜백 및 버퍼링 기법을 제안하였다. 이러한 기법을 이용하여 미들웨어 계층과 응용계층에서의 패킷 전송의 지연을 최소한으로 하고, 프로토콜 레벨에서 보장하는 실시간성을 미들웨어와 응용에서도 어느정도 보장할 수 있다. 향후 실시간성이 보장되는 운영체제가 채택된다면, 현재 구조에 우선순위 역전현상을 방지하는 버퍼 스케줄링 기법이 적용될 수 있다.

[참고문헌]

- [1] IEEE1394, Std for High Performance Serial Bus, 1995.
- [2] FireWire System Architecture: IEEE1394a, Don Anderson, MindShare, Inc, 1999
- [3] Specification of the Home Audio/Video Interoperability (HAVi) Architecture, Version 1.1, May 15, 2001.
- [4] HAVi Example by Example Java Programming for Home Entertainment Devices, Roger Lea, S.Gibbs, R.Gaub, R.Balaraman, Prentice-Hall, 2002.
- [5] IEEE1394 for Linux, <http://www.linux1394.org>