

리눅스 기반 이더넷-IEEE1394 게이트웨이

백갑천^U(gapdoli@cakra.dongguk.ac.kr), 오승현, 최우정, 안중석
(동국대학교 컴퓨터공학과 컴퓨터네트워크 연구실)

A Linux-based Ethernet-to-IEEE1394 Gateway

Gap-Chun Baek^U, Seun-Hyun Oh, Woo-Jung Choi, Jong-Suk Ahn
(Dept. of Computer Engineering, Computer Network Lab., University Of Dongguk)

요 약

IEEE1394는 다양한 디지털 기기들이 고속의 시리얼 버스를 통해 데이터를 송수신할 수 있는 홈네트워킹 기술 표준이다. 현재 이 기술을 이용하여 이더넷과 연결하기 위한 표준화가 IETF에서 진행 중이다. 본 논문에서는 리눅스 호스트에 장착된 IEEE1394 호스트 어댑터를 작동시킬 수 있는 디바이스 드라이버와 IETF에서 상정한 IPv4-over-1394 초안을 기초로 하여 이더넷과 IEEE1394를 연결하는 게이트웨이를 개발하였다. 본 논문에서는 IEEE1394용 리눅스 드라이버 모듈과 IPv4-over-1394를 구현한 커널 모듈의 구조를 설명하며 이더넷과 IEEE1394의 혼합 네트워크 환경에서 이더넷 응용 프로그램의 수행 실험으로 구현 모듈을 검증하였다.

1. 서 론

최근에 실시간 전송을 지원하는 다양한 멀티미디어 기기와 응용이 등장하고 있다. 또한 이들 기기간의 데이터 전송을 담당할 고속의 인터페이스 기술도 다양하게 개발되고 있다. PC나 디지털 기기들을 서로 연결해주는 인터페이스 기술로써 잘 알려진 표준들은 직렬/병렬 인터페이스, SCSI, USB 등이 있으나 멀티미디어 송수신을 위한 실시간 전송 능력, 확장성, 비용 등을 고려 할 때 미래 인터페이스 기술의 표준으로 IEEE1394[1]가 가장 유력시 되고 있다. 일 예로 미래 가정의 PC나 가전기기를 홈네트워크라는 단일 네트워크로 통합하려는 시도가 이루어지고 있는데, 세계 주요 가전 회사들은 자사의 오디오/비디오 기기들이 다른 회사의 제품들과 서로 연동될 수 있도록 하는 표준인 HAVi[7]등에서 IEEE1394를 인터페이스 표준으로 채택하고 있다. 이러한 IEEE1394가 각광받는 이유로는 고속을 필요로 하는 멀티미디어 데이터를 실시간 적으로 전송할 수 있고 각 기기를 연결하는 버스 구조의 유연성이 있으며, 64비트의 주소체계를 사용함으로써 기기의 메모리 하나 하나를 읽고 쓸 수 있다는 장점 때문이다.

현재 이더넷의 비약적인 성공은 다양한 네트워킹 기술들이 이더넷의 표준 프로토콜인 TCP/IP를 지원하도록 만들고 있다. 대표적으로 ATM 네트워크를 위한 IP-over-ATM이 그것이다. 따라서 이더넷의 표준화를 담당하는 IETF에서는 IEEE1394의 IP지원을 위한 표준안을 작성 중인데 현재 IPv4-over-1394-draft[2] 가 나와 있는 상태다.

본 논문은 IEEE1394 호스트 어댑터를 구동시키기 위한 리눅스 디바이스 드라이버와 IEEE1394 인터페이스를 이용한 IP통신을 규정한 IPv4-over-1394를 구현함으로써 이더넷과 IEEE1394를 연결하는 게이트웨이 개발에 관한 내용이다.

* 본 연구는 산업자원부의 "지능형 PCTV를 위한 가정용 네트워크 프로토콜 개발" 위탁과제로 수행 중임

2. IEEE1394와 IPv4-over-1394

2.1 IEEE1394 프로토콜 구조

IEEE1394-1995는 물리 층(Physical Layer), 링크 층(Link Layer), 그리고 트랜잭션 층(Transaction Layer)의 3계층으로 이루어진 하드웨어 및 소프트웨어 표준이다. 통상적으로 IEEE1394 호스트 어댑터는 물리 층과 링크 층을 수행하며, 호스트에서는 트랜잭션 층과 버스관리 기능을 담당한다. 물리 층은 주로 시리얼 버스를 사용할 수 있는 권리를 취득하는 중재 기능을 수행하며, 버스의 사이클 제어는 데이터 링크 층에서 담당한다. 트랜잭션 층은 네트워크 장치의 기본 기능-읽기, 쓰기-을 수행하며, 버스제어 기능에서는 동시성 전송에 필요한 자원관리 등의 기능을 담당한다. 또한 이들 3 계층이 제공하는 전송 서비스는 동시 및 비동기 전송이 있으며 전송 방법은 다음과 같다.

동시성 데이터는 패킷 내부에 특정 수신 노드의 주소를 포함하지 않고 채널번호를 기록하여 방송 모드로 송신한다. 노드들은 수신된 패킷을 검사하여 원하는 채널번호를 가진 패킷만을 수신한다. 이때 사용될 채널번호는 응용 프로그램 수준에서 정의되어야 한다. 참고로 방송 모드로 전송된 동시성 데이터에는 누락된 패킷에 대해 검색 및 복구 메커니즘이 제공되지 않는다. 동시성 전송이 완료되면 비동기 전송을 원하는 노드에서는 동시성 채널 사이의 간격이 예정보다 길어진 것을 감지하고 동시성 전송이 완료되었음을 알게된다. 비동기 전송은 매번 경쟁을 통해 버스 사용권이 사용되는데 비동기 전송에 앞서 노드들은 버스 마스터에게 사용 신청을 하게 되고 버스 마스터는 어느 한 노드에게 버스 사용권을 허락한다. 이 비동기 전송에는 목적지가 하나인 단방향 모드 전송과 버스의 모든 노드에게 송신할 수 있는 방송 모드가 가능하다. 단방향 모드 전송 후에는 수신 노드가 응답 패킷을 전송함으로써 신뢰성 있는 통신을 할 수 있다. 그러나 방송 모드 전송에 대해서는 응답 패킷이 발생하지 않으므로 패킷 손실을 복구할 수 없다.

비동기 전송은 한 사이클에서 등시성 데이터가 모두 전송된 후에 시작 될 수 있고, 따라서 등시성 데이터가 많은 경우에는 사이클마다 비동기 데이터가 전송되지 못하고 장시간 대기하는 경우가 발생할 수 있다. 비동기 전송에서는 하나의 신뢰성 있는 데이터 전송을 위해서 물리 층의 응답과 링크 층의 응답이 수반된다. 즉 데이터를 송신하고 나서는 상대방의 물리 층의 응답 패킷이 도착하게 되면 뒤이어 상대방의 링크 층의 응답 패킷이 뒤따르게 된다. 그리고 송신자의 측의 물리 층에서 이 링크 층의 응답 패킷에 따른 응답 패킷을 보내는 것으로 하나의 데이터 송신이 끝나게 된다.

2.2 IPv4-over-1394와 ARP(Address Resolution Protocol)

IPv4-over-1394는 IEEE1394버스 상의 노드들간의 IP데이터그램 송수신을 규정한 표준안이다. 표준안의 내용은 IP패킷과 IEEE1394 패킷을 구분하기 위한 인캡슐레이션(Encapsulation) 헤더 구조와 IP계층이 IEEE1394노드를 식별할 수 있도록 주소 변환 프로토콜을 위한 자료 구조와 방법을 명시하고 있다.

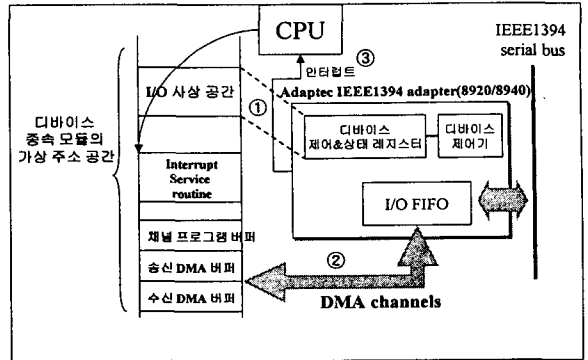
IEEE1394 버스에서 노드는 노드 ID를 통해 식별되므로 IP 주소로 지정된 목적지 노드로 패킷을 송신하기 위해서는 해당 IP 주소와 노드 ID의 쌍을 구해서 목적지 노드의 노드 ID를 결정하여야 한다. 기존의 TCP/IP 프로토콜 스택에서 ARP는 인터넷 장치를 중심으로 구성되어 있으므로 본 연구에서는 기존의 ARP 프로그램에 IEEE1394 장치를 추가하기 위해 새 하드웨어 장치 분류를 정의하였다. 또 ARP 요청 패킷은 버스에 방송 모드로 전송되므로 IEEE1394 버스의 방송 주소인 63번 노드 ID가 디바이스의 방송주소로 등록된다. ARP 응답 패킷은 버스에서 ARP 요청 패킷을 수신한 노드 중에서 자신의 IP 주소와 일치하는 노드에서 응답 패킷을 단방향 모드로 전송된다. ARP 요청/응답 패킷을 통해 얻어진 IP 주소와 노드 ID의 쌍은 인터넷과 동일하게 ARP 캐쉬(Cache)에 저장됨으로써 재 사용될 수 있다. 그러나 IEEE1394 버스에서는 노드의 가입/탈퇴로 버스 토폴로지가 변할 때마다 버스 리셋(Bus Reset) 신호가 발생하고 모든 노드의 ID가 다시 부여되어 IP 주소와의 쌍이 변경된다. 따라서 버스 리셋이 발생한 경우에 ARP에서는 ARP 캐쉬에 등록된 IP/노드 ID 쌍을 삭제하고, 필요에 따라 다시 ARP 요청/응답 과정을 실행하여야 한다.

3. IEEE1394어댑터 용 리눅스 디바이스 드라이버

3.1 IEEE1394 호스트 어댑터의 제어

본 절에서는 Adaptec사의 IEEE1394 어댑터인 8920/8940[6]용 리눅스 디바이스 드라이버 구현에 대해 소개한다. 드라이버와 어댑터는 인터럽트, 공유메모리, 그리고 DMA(Direct Memory Access) 방식에 의해 데이터를 교환하거나 어댑터를 제어하게 된다.

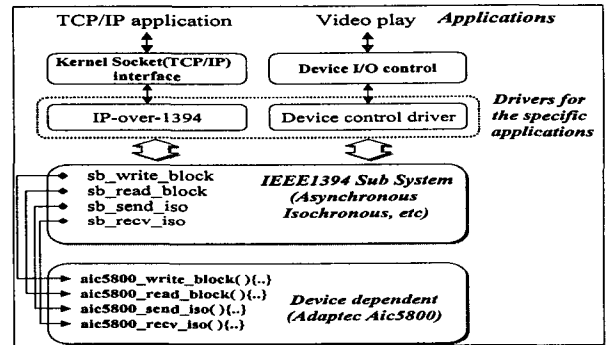
<그림 1>은 드라이버와 IEEE1394 어댑터 사이의 통신을 개념적으로 보여주고 있다. IEEE1394 어댑터 내의 제어 및 각종 상태 레지스터는 커널 메모리의 특정 영역에 매핑(Mapping)시키는 공유 메모리 방식(그림 안 ①)으로 접근할 수 있으며 이를 통해 어댑터 카드를 제어하게 된다. 실질적인 전송 데이터는 DMA(그림 안 ②)를 통해서 어댑터 카드에서 커널 드라이버의 DMA 버퍼로 또는 DMA 버퍼에서 어댑터 카드로 전송된다. 마지막으로 DMA완료, 버스 리셋(Bus Reset) 등의 하드웨어 사건(Event)은 인터럽트(그림 안 ③)를 일으켜 드라이버의 인터럽트 처리 루틴이 실행된다.



<그림 1> 드라이버 모듈의 IEEE1394 어댑터 제어

3.2 리눅스용 IEEE1394 디바이스 드라이버

커널에 동적으로 존재하는 IEEE1394 디바이스 드라이버는 <그림 2>과 같이 크게 세 가지 계층-응용 드라이버 모듈, IEEE1394 서브 시스템, 그리고 디바이스 종속-으로 구성되어 있다.

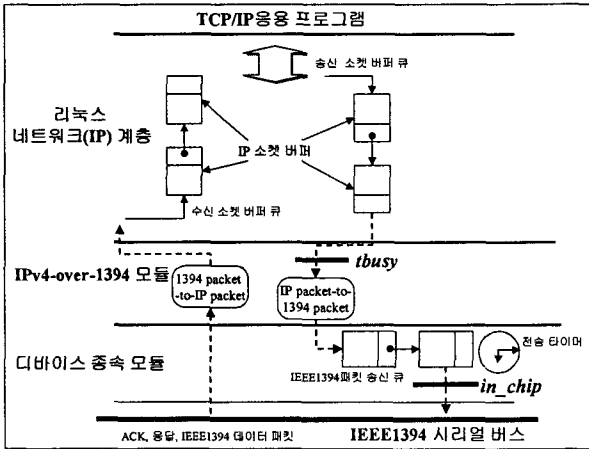


<그림 2> 리눅스 IEEE1394 디바이스 드라이버 모듈 구성도

최상위 계층인 응용 드라이버 모듈은 두 가지 부분으로 분리되어 있는데, 커널의 네트워크(IP)계층과 IEEE1394 서브 시스템 층을 연결해 주는 IPv4-over-1394 모듈과 응용 프로그램이 등시 전송 서비스를 이용할 수 있도록 서브 시스템과 응용 프로그램을 연결해주는 등시성 모듈이다. 이 계층은 응용 프로그램에게 투명한 서비스를 제공한다. 일 예로 IPv4-over-1394 모듈은 IP계층에게 서브 시스템의 특이 사항, 즉 IEEE1394의 특이 사항을 감추기 위한 계층이다. IP계층은 특정 IP 주소로 데이터를 보내기 위해 해당 네트워크 인터페이스가 인터넷이든, IEEE1394든 똑같이 응용 드라이버 모듈에 정의되어 있는 서비스 함수를 호출하면 된다. 호출된 서비스 함수는 서브 시스템을 거쳐 디바이스를 실질적으로 동작시키는 디바이스 종속 층의 함수를 호출하게 된다. 중간에 위치한 IEEE1394 서브 시스템의 역할은 상위 계층의 응용 드라이버 모듈에서 다중 IEEE1394 호스트 어댑터를 동일한 방식으로 이용할 수 있도록 해주는 것으로 본 연구에서 구현한 Adaptec 8920/8940 어댑터 외에 다른 어댑터를 사용하더라도 상위 계층들에게는 영향을 미치지 못하게 해준다. 마지막으로 디바이스 종속 층은 실제 IEEE1394 호스트 어댑터를 구동시키는 모듈로 디바이스와의 데이터의 송수신은 DMA를 이용하고 디바이스가 갖고있는 제어 레지스터들을 커널의 주소 공간으로 매핑시켜 디바이스를 제어한다.

3.3 IPv4-over-1394의 IEEE1394 비동기 전송 모듈 구성

IEEE1394를 통한 IP데이터그램의 전송은 IEEE1394의 비동기 전송 중 블록 송/수신을 이용한다. IP계층에서 하나의 IP패킷을 전송하게 되면 1394 계층은 정상적인 수신을 확인하는 신뢰성 있는 송신 트랜잭션을 수행하기 때문에 IPv4-over-1394를 구현한 모듈은 송신 트랜잭션이 완료될 때까지 IP패킷 전송을 연기시킨다.



<그림 3> IPv4-over-1394 모듈의 비동기 전송

<그림 3>에서 보듯이, IPv4-over-1394계층의 tbusy는 디바이스 종속 층의 모듈이 송신 트랜잭션을 수행할 수 있는가를 나타내며 물리 층의 송신 가능 여부는 in_chip을 이용한다. tbusy가 BUSY이면 송신할 데이터를 디바이스 종속 층의 송신 큐에 넣어 두며, 물리 층의 응답 패킷이나 링크 층의 응답 패킷이 수신되면 지연시킨 송신을 수행한다. 이때 물리 층의 응답 패킷은 in_chip을 IDLE로 만들어 전송 가능 상태로 만들고, 링크 층의 응답 패킷은 tbusy를 IDLE로 만들어 새로운 트랜잭션을 수행 할 수 있도록 만들어준다.

1394계층 내에서 물리적 ACK 및 링크 층의 응답 패킷이 오지 않을 경우 트랜잭션 층의 신뢰성 있는 전송을 확보하지 못한다. 이를 해결하기 위해 트랜잭션 층에서는 데이터 패킷 전송이 20ms의 타이머를 동작시켜, 전송 후 타임아웃까지 응답 패킷이 오지 않으면 5번의 재전송 시도 후, 이 때까지도 응답패킷이 오지 않을 경우 버스리셋을 한다. 이 버스리셋에 의한 새로운 망 구성으로 인해 변환된 노드ID를 상위 계층(TCP층)에 알려주어야 하며, 재전송으로 인한 중복 패킷 문제는 상위 계층(TCP층)에서 해결하도록 하고 있다. 앞으로, 타이머의 임계시간이 전체 성능에 미치는 영향을 고려해 보다 적합한 임계 시간의 정적 또는 동적 설정을 보완할 계획 이다.

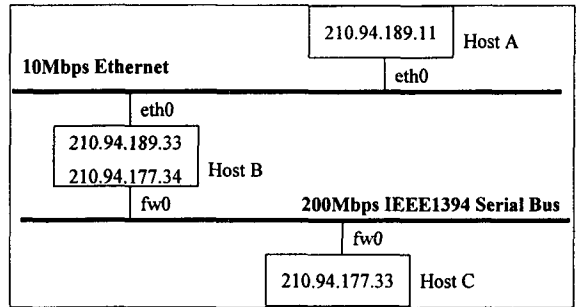
3.4 I/O제어 방식의 IEEE1394 동시 전송 이용

동시 전송 기능을 이용하기 위해서는 <그림 2>와 같이 디바이스 I/O방식을 이용하는데, 사용자 프로그램이 응용 드라이버 모듈과 연결되는 디바이스 파일을 오픈한 후 read(), write() 시스템 호출을 이용하여 송수신을 수행할 수 있도록 하였다. 또한 동시 모드 의 경우 비동기 모드와 달리 동시 채널 번호, 전송 대역폭 등의 리소스 예약이 필요한데 이는 ioctl() 시스템 호출을 이용 할 수 있도록 하였다.

4. 구현 모듈의 수행 실험

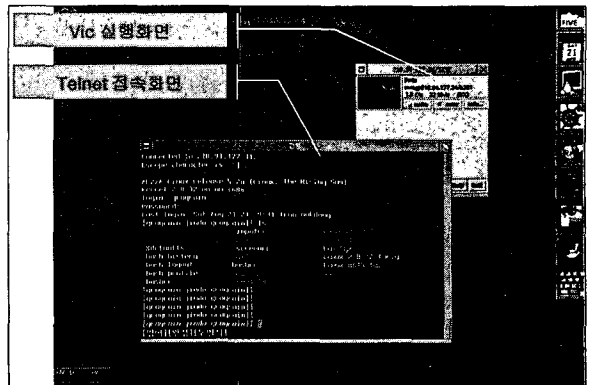
실험 응용 프로그램은 인터넷 영상의 도구인 Vic[3]과 Telnet

이 사용되었다. 실험 네트워크 토폴로지는 <그림 4>와 같이 10Mbps 이더넷과 200Mbps IEEE1394 버스의 두 개의 네트워크로 구성하여 Host B를 이더넷-IEEE1394 게이트웨이로 설정하였다.



<그림 4> 실험 환경

<그림 5>는 Host C에서 Vic과, Telnet을 수행한 모습으로 두 개의 응용 프로그램을 실행하면서 사용자 인터페이스를 지원하는 데 아무런 문제가 없었으며, 속도 저하도 느낄 수 없었다.



<그림 5> 실험 화면

5. 결론 및 향후 연구계획

IEEE1394는 미래의 PC나 가전기기를 통합하는 홈네트워킹의 표준 인터페이스로 예상되며 널리 사용되기 위해서는 인터넷과의 연동이 가장 큰 요구 사항으로 여겨지고 있다. 본 연구에서는 이더넷과 IEEE1394 간의 IP통신을 위한 이더넷-IEEE1394 게이트웨이를 개발하였으며, Vic[3]과 Telnet의 수행으로 구현 모듈을 검증하였다. 차후 연구 과제로는 동시 및 비동기 전송을 위한 통합 API의 개발과 성능 평가를 수행할 예정이다.

참고문헌

[1] IEEE 1394-1995, Std for High Performance Serial Bus, 1995
 [2] IETF Network Working Group, IPv4 over IEEE 1394, Internet-Draft. draft-ietf-ip1394-ipv4-15.txt
 [3] Steven McCanne and Van Jacobson, vic: A Flexible Framework for Packet Video. ACM Multimedia, November 1995
 [4] D. Anderson, "FireWire System Architecture: IEEE1394", Addison-Wesley, 1998
 [5] Alessandro Rubini, "Linux Device Driver", O'Reilly, 1998
 [6] Adaptec, "AIC5800 PCI-to-1394 Controller Chip Data Book and Design-In Handbook", 1997
 [7] Sony, "Specification of the Home Audio/Video-Interoperability (HAVi) Architecture", May 1998