논문 2004-41TC-5-4

10기가비트 이더넷 인터페이스를 위한 프레임 다중화기/역다중화기와 IPC를 갖는 10기가비트 이더넷 시스템의 설계 및 구현

(Design and Implementation of 10Gigabit Ethernet System with IPC and Frame MUX/DEMUX Architecture)

조 규 인*, 김 유 진**, 정 해 원***, 조 경 록****

(Gyu-In Cho, You-Jin Kim, Hae-Won Jung, and Kyoung-Rok Cho)

요 약

최근 인터넷 트래픽의 폭발적인 증가에 따라, 매우 빠른 고속 네트워크 장비에 네트워크프로세서(NP)의 사용이 보편화 되고 있다. 이에 따라, 기존의 일반적인 마이크로프로세서를 이용한 네트워크 장비의 성능 한계를 벗어나 향상된 성능을 보이는 라우팅 기능과 패킷처리 기능을 분리하는 분산형 시스템 구조가 이용되고 있다. 본 논문에서는 10기가비트 이더넷 포트를 가지는 10기가비트 에지 스위치 시스템에 적용한 패킷 라우팅 처리와 OAM 처리를 위한 분산형 이더넷 IPC 통신 메커니즘과 10Gbps급 이더넷 데이터를 처리할 수 있는 프레임 방식의 MUX/DEMUX 구조를 설계하고 구현하는 방법을 기술한다. 본 논문에서 제안한 분산형 이더넷 IPC 통신 메커니즘 구조는 현재 진행되고 있는 10기가비트 이더넷 인터페이스를 갖는 320Gbps급의 백본용 이더넷 스위치 시스템에도 적용 하였다.

Abstract

In this paper, we propose the ethernet Inter-Processor Communication (IPC) network architecture and 10gigabit ethernet frame multiplexer/demultiplexer architecture for the edge switch system based on Linux that has 10 Gigabit Ethernet (10Gigabit Ethernet) port with 72Gbps capacities. we discuss the ethernet IPC with ethernet switch and we propose design and implementation of ethernet Inter-Processor Communication (IPC) network architecture and multiple gigabit ethernet frame multiplexing/demultiplexing scheme to handle 10gigabit ethernet frame instead of using 10gigabit network processor. And then ethernet Inter-Processor Communication (IPC) network architecture and 10gigabit ethernet frame MUX/DMUX architecture is designed, verified and implemented.

Keywords : IP Router, Routing Protocol, IPC, PRR

I. 서 론

LAN(Local Area Network)은 1973년에 이더넷이 개발된 이후 10/100Mbps 및 1Gbps를 거쳐 현재에는 10Gbps의 속도를 수용하는 단계에 와 있으며 기존 장거리 통신 사업자들이 시분할 기술을 이용한 10Gbps전송 장비를 출시하고 있는 시점에 와 있다. 일반적으로 에지급 중형라우터가 백본급 대형라우터로의 업링크를 위해서는, 10기가비트 이더넷 인터페이스를 갖는 이더넷 위주로 MAN (Metropolitan Area Network)/WAN(Wide

* 학생회원, 충북대학교 정보통신공학부
(School of Information and Communication Engineering, Chungbuk National University)

** 정회원, ETRI 디지털홈연구원
(Digital Home Research Division, Electronics and Telecommunications Research Institute)

*** ETRI , 광대역 통합망 연구단
(Broadband Convergence Network Research Division, Electronics and Telecommunications Research Institute)

**** 충북대학교 전기전자공학부
(School of Electrical and Electronics Engineering, Chungbuk National University)
접수일자: 2004년1월29일, 수정완료일:2004년4월12일

Area Network)에 접속 되는 것이 가장 유력시되기 때문에 에지급 시스템은 10기가비트 이더넷 인터페이스를 필요로 한다. 또한 기존의 일반적인 마이크로프로세서를 이용한 네트워크 장비의 성능 한계를 벗어나기 위해, 현재 고속의 패킷 스위칭 및 라우팅을 구현할 수 있고, Layer3 스위칭 및 Diffserv.를 하드웨어로 지원 가능한 네트워크 전용 프로세서인 NP(Network Processor)를 사용한 네트워크장비의 개발이 보편화 되어지고 있다[1]. 이러한 추세에 반해, 10기가비트 NP의 개발은 기존의 기가비트 이더넷에서 네트워크 프로세서가 1Gbps 데이터를 처리하는 것처럼 단순히 네트워크 프로세서 속도 상의 증가만을 의미하는 것뿐만 아니라 광전송 장치, 고속 인터페이스 기술, 새로운 칩에 대한 시장성 등이 요구되어지기 때문에, 상용 10Gbps 네트워크 프로세서의 출시는 쉽게 이루어지고 있지 않다.

NP를 이용한 시스템의 구조는 일반적으로 패킷처리 기능을 담당하는 NP와 제어 및 관리 기능을 담당하기 위해 사용되는 호스트 프로세서(LP: Line Processor)의 구조가 일반적이다. 이런 구조에서 사용되는 인터페이스로써, PCI 버스 등이 제공 된다[2]. 패킷처리를 위한 NP가 헤더 파싱, 패턴 매칭, 비트필트 조작, 테이블 룩업, 패킷 수정 및 트래픽 관리 등을 하드 와이어드 속도(hard -wired speed)를 지원하는 기능을 한다. 이에 반하여, LP는 시스템 초기화를 수행하고, 포워딩 테이블에 엔트리를 추가하고, 멀티캐스트 그룹들을 관리하고, 버퍼관리에서 임계값 등을 바꾸는 관리 및 제어 기능을 수행을 한다. 근래에는 NP의 부가 기능으로 IP 포워딩, 프로토콜 변환 기능, QoS, 보안, 트래픽 대역폭 할당 기능, VoIP 기능들이 추가되고 있는데, 이를 위해서는 상위 레이어(L4~L7)에 대한 패킷 프로세싱을 요구한다. 이를 위하여 LP가 외부 상위 프로토콜 스택(L4~L7)을 지원하는 기능도 한다.

NP를 사용한 근래의 일반적인 라우터의 경우, 라우팅 관련 프로토콜 처리 및 라우팅 테이블 유지 등을 전담하는 라우팅 프로세서, 입출력 패킷을 스위칭 해주는 스위치 패브릭, 네트워크 인터페이스 및 포워딩 기능을 처리하는 라인카드 모듈로 구성한다[3]. 이러한 근래의 일반적인 라우터 구조에서, 라우팅 기능을 1개의 LP내에 구현하여 PCI(Peripheral Component Interconnect)버스를 통한 n개의 NP를 제어하는 중앙집중식 시스템 구조와 각 라인카드별로 LP를 두어 라인카드에 있는 NP를 이 LP가 제어를 하고 n개의 LP를 다시 메인프로세서(MP)가 IPC 기능을 통하여 관리를 하는 다중 분산 시스템의 경우를 생각해 볼 수 있다. 후자의 경우 다수의 라인카드 모듈로 구성되는 대용량 다중 분산 시스템에서는 타 모듈의 프로세서와 필요한 정보를 송수신하기 위한 프로세서간 통신(IPC: Inter Processor Communication)이 필요하다[4].

따라서, 본 논문에서는 이러한 기술적인 사항을 고려하여 다중 분산 시스템에서의 이더넷 IPC 통신 메커니즘 구조와 이미 개발된 4기가비트 급의 NP를 3개 사용하여 10Gbps 데이터를 처리할 수 있는 프레임 다중화/역다중화기(Frame Multiplexer/Demultiplexer) 구조를 제안하며, 제안한 구조는 10기가비트 이더넷 스위치 시스템(10Gigabit Ethernet edge Switching System) 개발에서 설계 및 구현 하였다.

2장에서는 10기가비트 스위치 시스템 구조를 살펴본다. 3장에서는 10기가비트 스위치 구조에 따른 효율적인 IPC 통신 메커니즘과 프레임 방식의 10기가비트 이더넷 프레임 MUX/DEMUX 구조에 대한 설계 및 구현에 관해 기술하고, 구현한 시스템의 성능에 대한 신뢰성을 평가한다. 마지막으로 4장에서는 제안한 다중 분산 시스템에서의 이더넷 IPC 구조와 프레임 방식의 MUX/DEMUX 구조에 대한 타당성을 끝으로 결론을 맺는다.

Ⅱ. 10기가비트 이더넷 스위치 시스템 구조

1. 하드웨어 구조

10기가비트 이더넷 스위치 시스템의 하드웨어 구성은 MP(Main Processor), LP(Line Processor), NP(Network Processor), SF(Switching fabric)로 구성된다. 그림 1은 10기가비트 이더넷 스위치 시스템의 구조를 나타낸다.

MP는 라우팅 프로토콜을 이용한 최적의 라우팅 경로와 관리기능을 담당한다. 일반적으로 도메인 내의 라우팅 정보는 OSPF(Open Shortest Packet First), IS-IS (Intermediate System to Intermediate System)의 프로토콜에 의해서 만들어 지고 서로 다른 도메인은 BGP (Border Gateway Protocol) 프로토콜에 의해 생성된다[5][6]. 또한 관리 기능은Telnet등을 이용한 원격 로그인 기능, FTP, 장애관리, 보안 처리 등을 담당한다[7][8]. LP는 입력된 패킷을 포워딩하기 위한 FIB(Forwarding Information Base)를 룩업(Look-Up)하여 적절한 포트로 패킷을 포워딩하는 기능을 담당한다[9]. NP는 수신된 패킷의 포워딩 기능을 담당하며, SF는 입력포트와 출력포트를 연결한다.

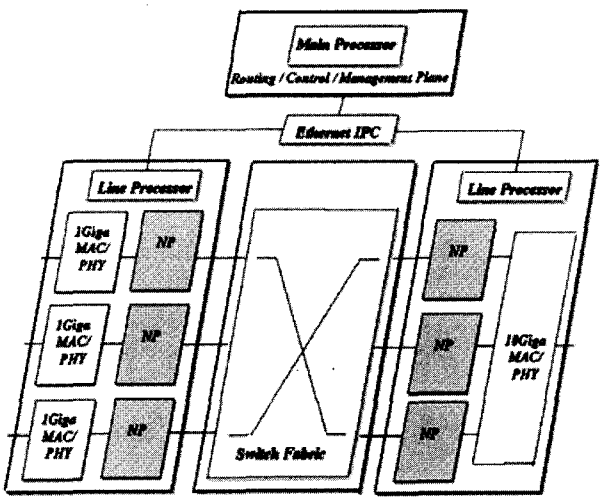


그림 1. 10기가비트 이더넷 시스템 구조
Fig. 1. The structure of 10gigabit ethernet system

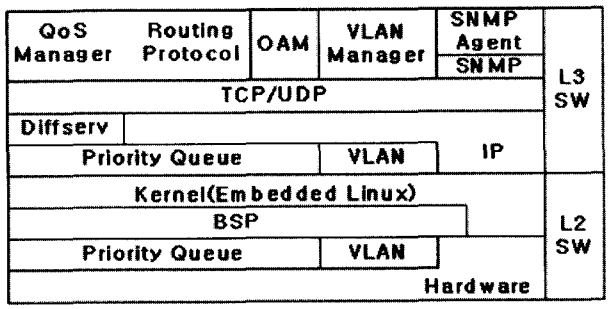


그림 2. 10기가비트 이더넷 리눅스 소프트웨어 구조
Fig. 2. The structure of 10gigabit ethernet Linux software

2. 소프트웨어 구조

10기가비트 스위치 시스템의 일반적인 소프트웨어 계층 구조는 하드웨어 기반위에 U-BOOT와 Linux Kernel, TCP/IP, Application 부분으로 나누어진다. 그림 2는 리눅스 소프트웨어 계층 구조도이다.

U-BOOT는 Kernel과 하드웨어 간의 인터페이스 기능을 담당하며, MP와 LP의 하드웨어 사양을 따른다. Kernel은 Embedded Linux를 사용한다. TCP/IP는 네트워크 모듈로 Linux에 포함되는 모듈이며, Application에는 L2/L3 프로토콜 처리 기능, 시스템 장애 및 시스템 운용 관리 기능, 망 관리 기능, L2/L3 QoS(Quality of Service) 관리 기능, L2/L3 VLAN(Virtual LAN) 기능을 수행하는 프로그램이 탑재되며, 이러한 응용 프로그램은 기능별 다수의 소프트웨어 블록으로 구성된다.

Priority Queue는 각 패킷에 있어서 우선순위를 보고 입출력 순서를 결정하며, L2 스위치에서는 MAC 어드레스로 나가는 곳의 주소를 판단하고 L3 스위치에서는 IP 어드레스로 판단한다. VLAN은 물리적인 네트워크 구성에 제한을 받지 않고, 네트워크 구성요소의 추가나 삭제 및 변경이 발생했을 경우 논리 네트워크를 구성함으로써 유연하게 대응할 수 있는 LAN 기능을 제공하며 VLAN Manager에서 관리한다. Diffserv는 Differential Service이며, QoS를 보장하기 위한 차등적 서비스이다. SNMP(Simple Network Management Protocol)는 망 관리 프로토콜이며, SNMP Agent에 의해 관리된다. QoS Manager는 Diffserv를 지원 및 관리하며, Priority Queue의 가입, 탈퇴, 우선권 부여 등 802.1p의 관리기능을 수행한다. 이러한 어플리케이션 프로그램의 모든 프로세싱은 OS(Operating System)를 통해CPU(Central Processing Unit)가 처리하며, 네트워크 입출력에 관련되는 프레임 데이터에 대한 프로세싱은 NP의 API (Application Programming Interface)를 이용해 NP가 처리한다.

그림 1과 같이 다중 노드로 구성되어 있는 시스템은 각 노드들 간에 정보 교환을 위해 프로세서간 통신인 IPC(Inter Processor Communication) 채널이 필요하다. 일반적으로 IPC는 셀버스(Cell Bus), HDLC(High Level Data Link), ATM(Asynchronous Transfer Mode) 스위치 버스, 이더넷 스위치 버스를 이용한 방식을 사용한다. 이중 셀버스, HDLC 방식은 공유버스 방식으로 일대일 통신 방법을 사용하는 ATM 스위치 버스, 이더넷 스위치 버스에 비해 처리 성능이 떨어지는 단점이 있어 주로 저가이면서 사용이 간편한 이더넷 스위치를 이용한 IPC 채널이 사용된다[10].

이와 같이 다중 노드 구조를 갖는 10기가비트 이더넷 스위치 시스템에서 필요로 하는 IPC 채널을 위한 통신 메커니즘은, 첫째 MP의 라우팅 테이블을 LP로 전달하고 OAM(Operation, Administration, Maintenance)을 통한 시스템 제어를 위해 사용하는 내부용IPC 통신 메커니즘과, 둘째 OSPF, RIP, BGP와 같이 외부 시스템과의 통신을 위한 외부용 IPC 통신 메커니즘으로 나누어 볼 수 있다.

또한, 10기가비트 라인카드와 같이 일반적으로 상위 라우터와 연결되는 업링크 포트는 다른 인터페이스에 비해서 빠른 속도를 가진다. 그림 1에서 스위치 패브릭과 NP의 인터페이스는 동일한 조건을 통해 제공해준다. 이 동일한 조건은 스위치 패브릭의 여러 포트를 논리적으로 하나의 단일포트로 구성하는 Link Aggregation 기능을 가능하게 함으로써, 업링크로 10Gbps를 인터페이스 할 수 있는 방법을 제공해준다.

3장에서는 이들 내부용 IPC와 외부용 IPC에 따른 효율적인 소프트웨어의 구조와 업링크로 10Gbps를 인터페

이스 하는 프레임 MUX/DEMUX 구조를 제안한다.

Ⅲ. 효율적인 IPC 통신 메커니즘 구조와 프레임 MUX/DEMUX 구조 설계

1. 시스템 내부용 IPC(IPC-II) 통신 메커니즘 구조 설계

리눅스 운영체제는 기본적으로 IP 프로토콜 위에 TCP(Transmission Control Protocol)와 UDP(User Da-tagram Protocol) 프로토콜을 제공한다. IP계층으로부터 처리된 패킷은 해당 패킷의 상위 프로토콜의 종류에 따라 해당 모듈로 전달된다. 상위 프로토콜이 TCP인 경우는 TCP 패킷 처리 모듈로 UDP인 경우는 UDP 처리 모듈로 전달된다. TCP 모듈은 신뢰성 있는 통신을 보장하기 위하여 패킷의 순서 제어, 흐름 제어를 수행하며 처리된 패킷은 소켓 인터페이스를 통해 해당 응용프로그램으로 전달된다. 패킷 전달의 경우 응용프로그램은 해당 전송 프로토콜을 결정하여 소켓 인터페이스를 사용하여 TCP로 패킷을 전달한다. TCP는 응용프로그램에서 전달한 패킷에 TCP 헤더를 붙인 후 IP에 전달한다. UDP는 비연결형의 통신을 사용하여 보다 간단한 통신 방법을 제공한다. 패킷의 흐름 제어를 하지 않으며 기본 헤더가 처리된 패킷은 소켓 인터페이스를 통해 해당 응용프로그램으로 전달된다. 패킷 전달의 경우 응용프로그램은 해당 전송 프로토콜을 결정하여 소켓 인터페이스를 사용하여 UDP로 패킷을 전달한다. UDP는 응용프로그램에서 전달한 패킷에 UDP 헤더를 붙인 후 IP에 전달한다. 그림 3은 TCP/UDP 계층을 이용한 소프트웨어 처리도이다.

그림 3과 같이 TCP는 데이터의 신뢰성 있는 전달을 보장하기 때문에 라우팅 정보와 같이 Reliable Transmi-ssion을 위해 사용하지만, 시스템 운용을 위한 OAM의 제어 메시지는 시스템 안정 동작을 위해 보다 빠른 처리를 요구한다. 이런 관점에서 보면, TCP/UDP 계층을 이용한 방법은 네트워크 계층인 IP 주소를 사용하여 패킷 전달을 처리하는 것으로 동일한 네트워크 망에서 통신을 하는 시스템 내부 망에서는 IP 주소를 이용한 통신은 비효율적이며, 2계층을 거치기 때문에 시간도 많이 걸리는 단점을 갖는다. 이러한 단점을 보완하기 위해 본 논문에서는 IPC-II 프로토콜을 이용하여보다 신속한 동작을 요구하는 메시지는, TCP/IP 또는 UDP/IP의 2계층을 사용하는 IPC 방법에서 하나의 프로토콜 계층으로 처리하도록 제안한다. 즉, IPC-II는 맥(MAC)주소를 이용하

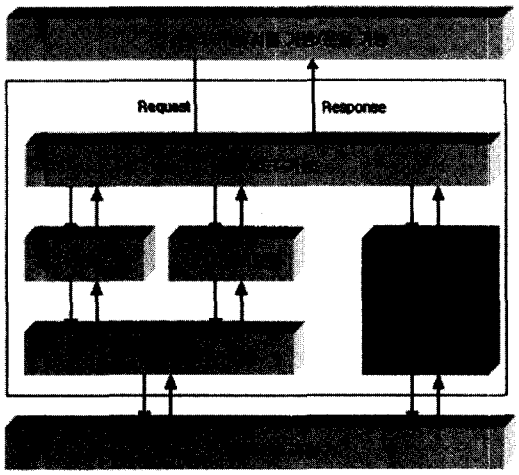


그림 3. TCP/UDP에 의한 처리 기법
Fig. 3. TCP/UDP process mechanism

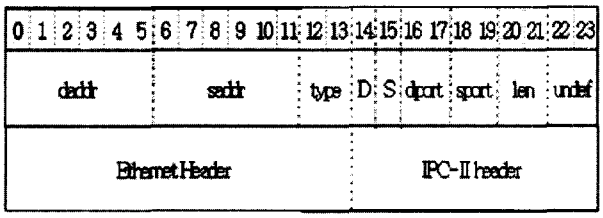


그림 4. IPC-II 헤더
Fig. 4. IPC-II header

여 목적지, 소스 주소를 설정하고, 이더넷 프레임에 데이터를 인캡슐하는 방식으로 TCP/IP 또는 UDP/IP 기능을 사용하지 않는다. 이는 시스템을 위한 내부 IPC를 위해 2계층을 통과하는 패킷 프로세싱 절차를 하나의 계층을 통해 패킷 프로세싱 함으로써 얻을 수 있는 시간 단축을 의미한다. 그림 4는 정의된 IPC-II 헤더를 나타낸다.

D는 목적지 슬롯(destination slot), S는 소스 슬롯(so-urce slot) 번호이며, dport(destination port)는 목적지 응용프로그램 포트 번호, sport(source port)는 소스 응용프로그램 포트 번호로 메시지가 송수신되는 응용프로그램을 명시한다.

2. 프레임 MUX/DEMUX 구조

네트워크 에지 부분에 있는 시스템들은 다음과 같은 이유로 고 가용성 보호 기능이 필요하다. 첫째, 에지 디바이스들이 종종 터미네이트 되면서 수많은 네트워크를 정지 시킬 수 있다. 이것은 서비스 공급자 네트워크의 경우에 하나의 에지 디바이스가 수십만 개의 연결된 연결부들을 터미네이트 시킴으로써 오랜 네트워크 정지를 가져오기 때문이다. 둘째, 네트워크 코어는 중복 회로에 의한 고장 난 네트워크 요소들을 완벽하게 우회할 수 있

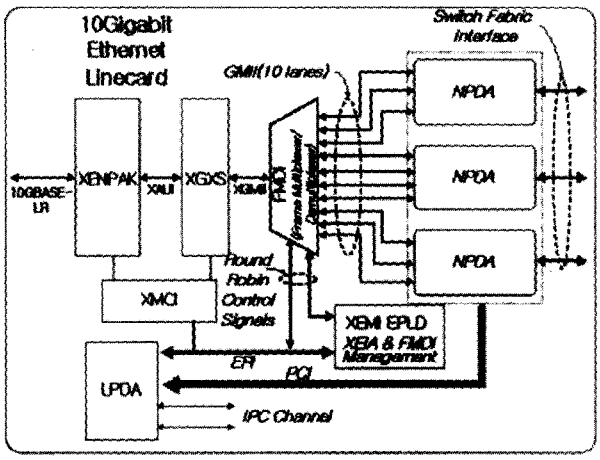


그림 5. 10기가비트 이더넷 라인카드의 구성도
Fig. 5. The structure of 10gigabit Ethernet linecard

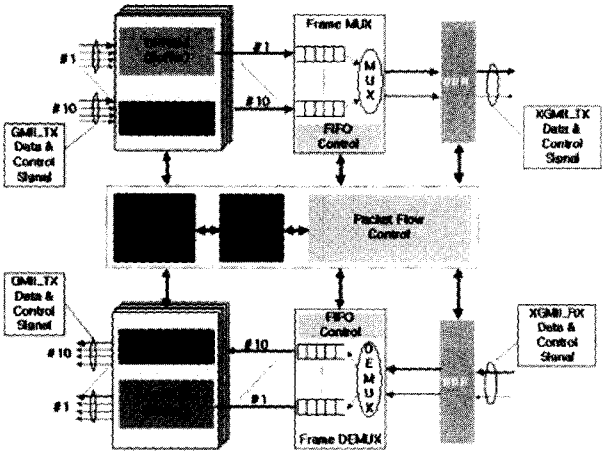


그림 6. 구현된 프레임 다중화/역다중화기 구성
Fig. 6. The structure of the implemented frame multi plexer/de multiplexer

는 반면에, 에지는 에지 디바이스에 대해 하나의 링크밖에 없으며 고객의 트래픽은 고장 난 디바이스를 우회해 갈 방법이 없다. 이러한 이유 때문에 에지 디바이스들을 중복하는 1+1 또는 N+1의 라인 카드 백업을 생각할 수 있으나, 이는 가용성 측면에서 좋은 방법이나 중복회로에 들어가는 비용이 높아지기 때문에 잘 사용하지 않는다. 이러한 가용성 측면과 10기가비트급 네크워크 프로세서가 일반화되지 않은 현 시점에서 기존에 사용되던 4기가비트급의 NP 3개를 이용해 10기가비트급 이더넷을 처리하고 가용성 측면에서 최대 NP 2개가 고장이 나더라도 보드의 교체 없이 고장 난 디바이스를 우회해 갈 수 있는 방안을 제시한다.

일반적으로 고속의 스위치 패브릭은 같은 속도와 같은 사양의 스위치 포트를 제공해 준다. 이는 포트를 묶어서 사용하는 Link Aggregation 기능을 말하는데, 논리적으로 단일포트처럼 인식하도록 하는 것이다. 이러한 Link Aggregation 기능을 프레임 방식의 다중화/역다중화기 설계에 적용하여 1Gbps의 GMII 인터페이스 10개의 채널을 논리적으로 10Gbps의 단일채널로 인식하도록 설계하였다. 그림 5는 10기가비트 이더넷 인터페이스 및 10Gbps 데이터를 처리할 수 있는 10기가비트 이더넷 라인카드 구성도이다. LP는 보드의 초기화 및 상태 감시 그리고 3개의 NP와 PCI버스를 통해 연결 되어지며, IPC 채널을 통해 MP와 연결 되어 FT를 받게 된다. 1개의 NP는 3개 혹은 4개의 GMII(Gigabit Media Independent Interface)인터페이스를 가지며, 10개의 GMII인터페이스는 FPGA(Field Programmable Gate Array)를 통해 구현된 프레임 다중화/역다중화기에 연결된다. 프레임 다중화/역다중화기는 1개의 XGMII(10 Gigabit Media Independent Interface) 인터페이스를 가지며, 이것은 XGXS(10Gigabit Extender Sublayer)인 10기가비트 이더넷 PHY블록으로 연결된다. XGXS는 10기가비트 이더넷 이더넷 광모듈인 Xenpak 모듈과 연결 된다.
그림 5의 10기가비트 이더넷 라인카드에서 MUX/DEMUX는 앞서 언급한 IEEE802.3ad Link Aggregation (LA) 기능을 이용하는데, 설계 시 고려되어야 할 사항은 다음과 같다.

첫째는 송신 시 프레임간 순서정렬을 위한 분배기능이며, 둘째는 MAC주소에 대한 주소생성(Addressing), 마지막으로 분배과정에서의 링크 변경 시에 대한 대응방안이다. 위의 3가지 고려사항은 하나이상의 링크를 LA그룹(Link Aggregation group)으로 모아 MAC 클라이언트가 LA그룹을 하나의 링크로 간주하는 경우에 해당한다. 제안한 프레임 MUX/DEMUX 설계에서는 1개의 10기가비트 이더넷의 물리적인 인터페이스로 링크가 되므로, LA의 MAC주소에 대한 주소생성 (Addressing)에서는 차이가 있다.

따라서, MUX/DEMUX 설계에서의 중요하게 고려해야 할 사항은 송수신시의 프레임간의 순서 정렬기능이라 할 수 있다. 특히, 스트리밍 지향적인 데이터인 경우, 프레임간의 잘못된 순서 정렬은 네트워크 효율을 상당히 저하시키는 요인으로 작용한다.

또한, GMII와 XGMII의 상호 변환 시에도 흐름제어를 통해 10Gbps 데이터양을 네트워크 프로세서 3개에 분산하여 처리하는 과정에서 1개의 NP에 부하가 집중되어 오버플로가 발생하지 않도록 해야 한다. 그림 6은 제안한 프레임 다중화/역다중화기를 FPGA로 구현한 블록도이다.

그림 6에서 송신시 1개의 GMII출력(GMII_Tx)은 송

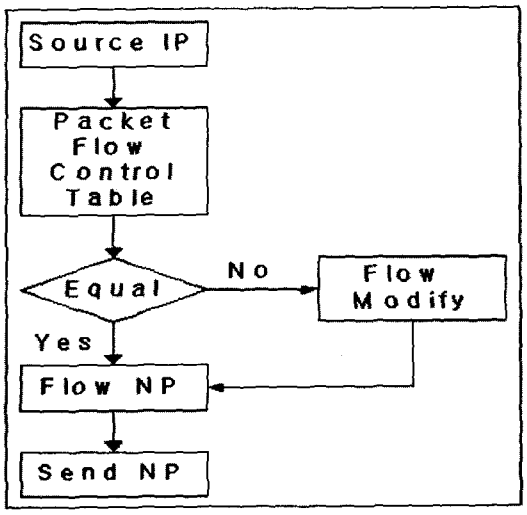


그림 7. PRR(Packet Round Robin)의 흐름도
Fig. 7. PRP mechanism

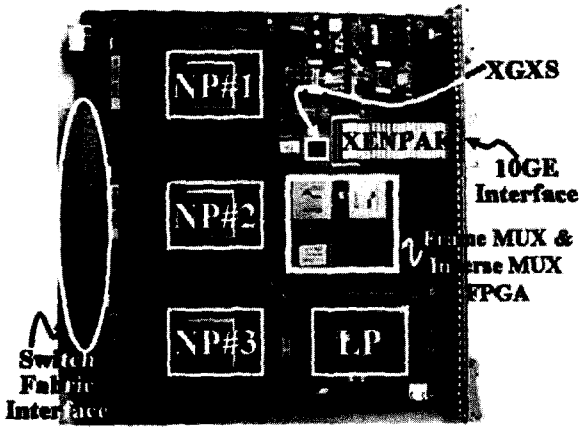


그림 8. 제안한 프레임 방식의 10기가비트 이더넷 라인 인터페이스 카드
Fig. 8. Implemented 10gigabit Ethernet linecard

신클럭인 TX_CLK(125MHz), 8비트 버스구조를 가지는 기가비트 이더넷 데이터(TXD<7:0>), 이러한 데이터의 유효구간을 알려주는 인에이블 신호(TX_EN) 및 에러의 유무를 지시해주는 에러신호(TX_ER)로 구성된다. 이러한 1개의 GMII_Tx는 FIFO에 입력되고, MUX/DEMUX의 제어를 받으면서 버퍼메모리(Buffer DPRAM)에 저장된다. 이와 같은 방법으로 모든 10개의 GMII_Tx 신호선 들이 각각의 버퍼메모리에 저장된다. 기가비트 이더넷에 대해 포트별로 버퍼메모리를 독립적으로 두는 이유는 독립 적인 프레임 변환 및 저장작업을 수행하기 위해서이다. 버퍼메모리의 크기는 최대 기가비트 프레임 크기보다 크게 설계 하였다.

특히, Packet Flow Control은 패킷 단위의 PRR(Pac-ket Round Robin) 알고리즘을 적용하였다. 기존의 시간단위 RR은 흐름마다 각각 분리된 큐를 사용하기 때문

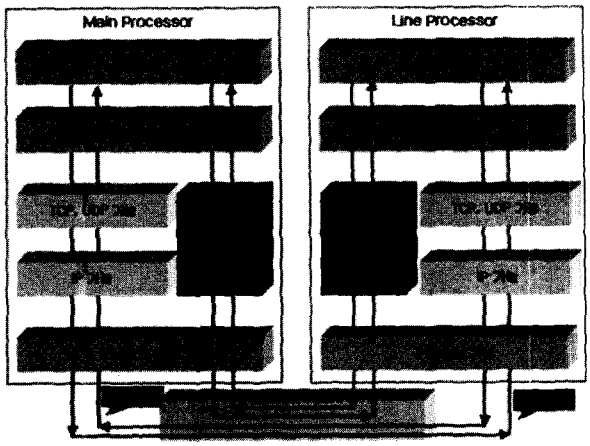


그림 9. 이더넷 IPC 측정을 위한 시험 구성도
Fig. 9. Test environment of Ethernet IPC measurement

에 흐름 간에 보호를 할 수 있으나, 서비스에 대한 보장에 제한을 가지게 된다. RR에서 각 흐름이 서비스 받는 일정 시간 간격은 고정된 시간을 사용할 수 있지만, 패킷 네트워크의 경우 패킷의 개수를 사용하게 되기 때문에 만일 패킷의 크기가 고정되지 않은 경우, 정확한 서비스 보장과 공평성이 유지되지 못하는 단점을 갖는다. 제안한 방법에서는 입력된 같은 패킷에 대해 플로우를 고려하여 같은SA(Source Address)를 갖는 패킷 스트림으로 정의하여, 같은 플로우는 같은 네트워크프로세서로 보내 처리하도록 하여 통과시 전달 순서가 바뀌지 않도록 설계 하였다. 그림 7은 PRR의 흐름도이다.

이상과 같이 10기가비트 이더넷 인터페이스 카드는 4기가비트 용량의 상용 NP 3개를 이용하여 10개의 기가비트 이더넷 포트를 갖으며, 각각의 포트에 대해 독립적인 처리를 위해 동일구조의 다중화부 및 역다중화부 메모리와 제어 서브블록이 사용 되었다. 설계한 MUX/DEMUX는 FPGA을 사용하여 구현하였으며 사용된 디바이스는 Xilinx사의 VERTEX II 계열로써, 내부 클록은 GMII 및 XGMII에서 사용되는 125MHz, 156.25 MHz 및 프로세서 인터페이스를 위한 33MHz를 사용한다. 그림 8은 프레임 방식의 MUX/DEMUX를 적용하여 설계한 10기가비트 이더넷 인터페이스 카드이다.

3. 제안한 내부 IPC 구조와 프레임 MUX/DEMUX에 대한성능비교 측정

본 장에서는, 먼저 UDP/IP를 이용한 IPC와 IPC-II를 적용한 IPC에 대한 성능을 측정하고, 1기가비트 라인카드의 NP에 대한 성능측정을 통해 가능한 MUX/DEMUX의 수율과 PRR 알고리즘에 대한 성능을 측정한다.

표 1. UDP/IP와 IPC-II의 성능 비교 결과값
Table 1. Test result between UDP/IP and IPC-II

Type	UDP/IP		IPC-II	
송신Bytes	800	1500	800	1500
송신Packet량	10000	10000	10000	10000
RTT(msec)	9991.23	10958.16	8277.41	10241.31

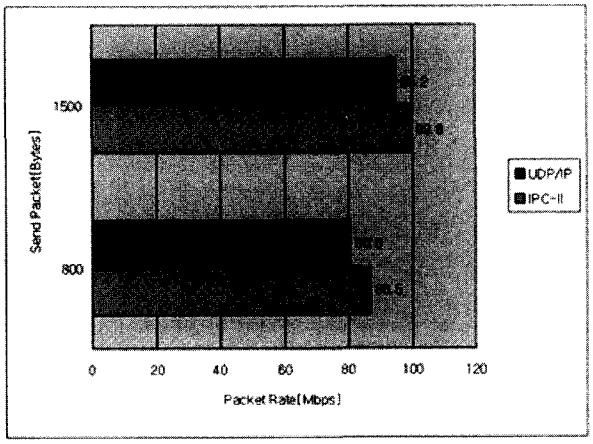


그림 10. UDP/IP와 IPC-II의 성능 비교 그래프
Fig. 10. Performance graph between UDP/IP and IPC-II

그림 9는 IPC 측정을 위한 시험 구성도이다. 특히 제안한 IPC-II의 프로토콜은 TCP/IP 또는 UDP/IP에 비해 RTT(Roud Trip Time)가 800바이트일 경우, 1500바이트일 경우 모두 줄어들어 결과적으로 같은 패킷을 처리할 경우 좀더 빠르게 처리함으로써, 처리속도가 향상되는 결과를 가져왔다. 실험에 적용한 IPC 메시지 길이는 OAM에서 필요로 하는 800바이트, 라우팅 테이블과 외부용 IPC에 이용되는 1500바이트를 기준으로 하였다. 시험은 라인카드에서 패킷을 생성하여 전송하는 어플케이션과 패킷을 수신하여 패킷의 소스로 다시 보내주는 서버 어플리케이션을 이용하였다[11]. 표 1은 UDP/IP와 IPC-II의 성능에 대한 결과 값을 보여준다.

그림 10은 MP와 LP 사이에 이더넷을 이용한 IPC에 대한 Throughput 측정 결과를 보여준다. 그림의 결과처럼 패킷 처리 능력에서도 평균 6%정도 향상됨을 확인할 수 있다. IPC 메시지 길이가 800바이트일 경우 7.3%, 1500바이트일 경우 4.7%의 성능 개선효과를 보여준다. 이것은IPC-II의 프로토콜을 이용하면, 시스템의 OAM 처리나 라우팅 프로토콜 패킷처리에서는 UDP/IP를 이용할 때 보다 처리시간과 패킷 처리 성능 면에서 각각 11%, 6%의 시스템 성능이 개선되었다.

마지막으로, 프레임 MUX/DEMUX에 대한 성능측정을 위한 수율을 구하면 다음과 같다. 이더넷 패킷은 8

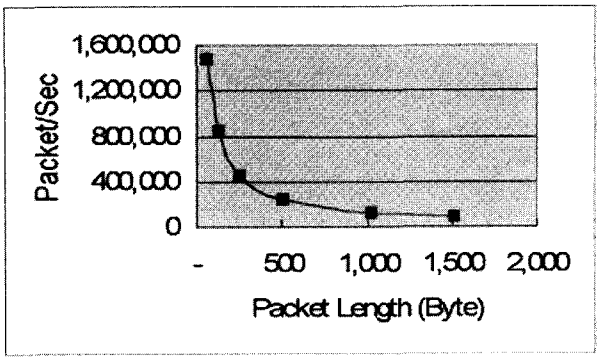


그림 11. 패킷 길이에 따른 전송 가능한 최대 패킷 수
Fig. 11. The maximum packet number of packet length

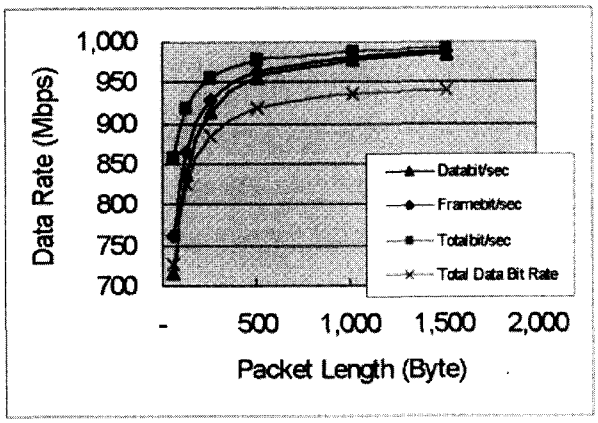


그림 12. 패킷 길이의 변화에 따른 최대 수율
Fig. 12. The maximum rate of packet length

바이트 (7바이트 Preamble + 1 바이트 SFD) 오버헤드와64~ 1518 바이트의 데이터로 구성되어 있다. 프레임의 내부는 46 ~ 1500바이트의 데이터, 14바이트의 MAC 헤더, 4바이트의 CRC로 이루어진 MAC 트레일러로 구성되어 있다. 또한 이더넷 패킷과 패킷 사이에는 IPG (Inter-Packet Gap)가 최소 12 바이트 (96ns) 존재한다. 따라서, 초당 전송 가능한 패킷의 수를 계산하면 아래의 수식과 같이 계산이 가능하다.

$$Packet/\sec = \frac{1}{FrameLength[ns] + Preamble(64ns) + IPG[ns]} \quad (1)$$

이 수식에서 IPG를 최소 단위인 12바이트로 가정하면, 각 패킷 길이에 따른 최대 패킷 전송률을 계산할 수 있다. 이를 나타낸 것이 그림 11인데, 패킷의 길이를 최소 64바이트에서 최대 1518바이트까지 변화시키면서 Packet/Sec의 최대 값의 추이를 나타낸 것이다. 또한, 초당 전송되는 패킷의 수에 패킷의 길이를 바이트나 비트 단위로 환산할 경우에 수율을 얻을 수 있다.

따라서, 실제 1Gbps나 10Gbps의 포트에서 100 %의

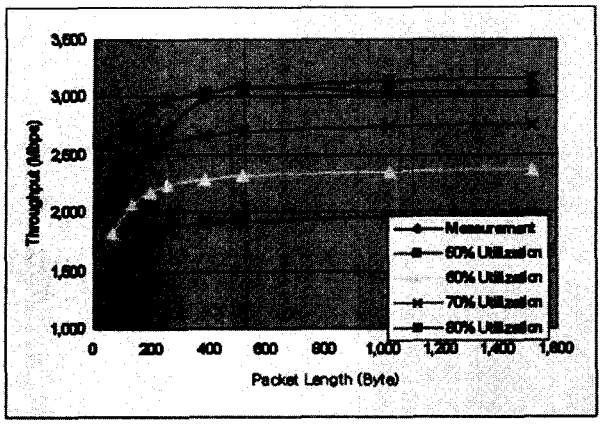


그림 13. 4개의 포트에 동일한 트래픽 입력시수율
Fig. 13. The maximum rate of IBM NP

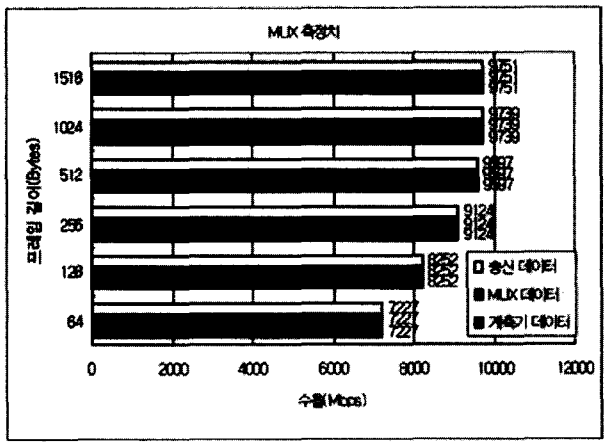


그림 14. 1Gbps 10포트와 10Gbps 1포트 트래픽 수율
Fig. 14. Test result between 1giga port and 10giga port

utilization을 갖는다고 가정한 경우에 IPG나 오버헤드 비트들 때문에 실제 수율(throughput)은 100 %의 utiliz -tion을 갖지 못하는 1Gbps 또는10Gbps가 된다. 이를 바탕으로 각각의 수율의 정의에 따른 최대 수율은 그림 12와 같이 측정된다.

각각의 수율에 대한 정의는 100%의 utilization을 갖는 것을 가정하여, 64 byte에서 1518 byte에 따른 수율의 변화를 보여주는 것이다. 이때 IPG는 12 byte (96ns), 이더넷 헤더 및 트레일러 (CRC) 까지를 포함한 수율 계산이다. 최소 패킷인 64 byte에서는 최대 수율이 700~850Mbps, 최대 패킷인 1518byte에서는 거의 1Gbps에 도달하고 있다는 것을 알 수 있다. 그림 13은 4개의 1기가포트를 통해서 동일하게 트래픽을 인가했을 경우에 나타나는 최대 수율의 변화를 측정한 것이다. 즉, 그림에서 나타난 입력 이상으로 트래픽이 인가될 경우에 시스템은 down된다. 따라서, 그림 13에서 보이는 수율은 현재 시스템에서 수용할 수 있는 최대 트래픽 전송 속도라고 할 수 있다. 패킷 길이에 따라서 수율이 변하기는 하지만, 대부분이 70-80%의 utilization에서 패킷 손실이 없이 거의 모든 패킷이 완전하게 전달됨을 확인하였다.

이상과 같은 방법으로 그림 13은 동시에 1Gbps 10포트에 데이터를 인가하였을 경우에 프레임 MUX /DEMUX를 통과하여 10Gbps 포트를 통과한 수율을 측정한 것이다.

그림 14는 1기가비트 10포트와 10기가비트 1포트에 대해 동시에 트래픽을 가했을 경우의 측정치이다. 먼저, MUX에 대한 결과는 IXIA의 1기가 10포트를 통해 송신한 패킷이설계한 MUX의 레지스터를 통해 처리된 패킷 카운트 수와 10기가비트 포트에 수신된 패킷 량이 같음을 보여준다. DEMUX에 대한 결과는 IXIA 10GE 포트를 통해 송신된 패킷이 설계한 DEMUX를 통해 1기가의 10포트로 거의 균일하게 처리됨을 보여준다. 따라서, QoS 기반의 10기가비트 이더넷 에지 스위치 시스템에 제안한 Frame MUX/DEMUX의 설계가 정상임을 확인할 수 있었다.

Ⅳ. 결론 및 향후 추진 방향

인터넷 이용자의 급속한 증가에 따른 인터넷의 데이터 증가는 일반적으로 라우터에서 데이터 전달의 병목현상을 일으켜 망의 성능에 큰 영향을 미치고 있다.

본 논문에서는 10기가비트 이더넷 스위치 시스템의 개발에 있어 상기와 같은 문제점을 해결하기 위해, 분산형 이더넷 IPC의 소프트웨어 구조와 기가비트급 네트워크 프로세서를 다중으로 사용하여 10기가비트급의 데이터 용량을 처리할 수 있는 프레임 MUX/DEMUX 구조의 방식을 제안한다. 분산형 이더넷 IPC는 MP에서 라우팅 테이블(Routing Table)을 관리하고 IPC를 통해 LP로 포워딩 테이블(Forwarding Table)을 전달하여 관리하게 함으로써, 라우터 기능의 일이 분산되어 하드웨어 기반의 wire-speed로의 포워딩 기능을 가능하게 하여, 망의 성능을 개선시킬 수 있음을 보여준다. 동시에 시스템 내부에서 처리되는 OAM 기능과 라우팅 프로토콜 패킷은 기존의 TCP/IP나 UDP/IP를 통한 2계층 처리에서 IPC-II 1계층 프로토콜 처리로 평균 7%의 패킷처리 성능개선을 가져와 보다 신속한 시스템의 제어를 가능하게 하였다. 마지막으로 10기가비트 이더넷 라인카드에 적용된 프레임 MUX/DEMUX 구조를 통해, 가용성 측면에서는 10Gbps로의 업링크 구성 시 패킷의 전달순서를 유지하며 3개의 NP를 사용함으로써 최대 2개의 NP

가 고장 시에도 라인카드의 교체 없이도 10Gbps로 서비스를 유지할 수 있으며, Load를 분산하여 하나의 NP에 오버플로가 발생하는 것을 방지할 수 있는 측면과 10기가비트급 네크워크 프로세서가 일반화되지 않은 현 시점에서 기존에 사용되던 NP를 이용해 10Gbps 인터페이스를 제공할 수 있는 계기를 마련하였다.

인터넷 트래픽의 증가로 인해 포워딩 패킷뿐만 아니라 내부 처리 패킷도 함께 증가 하고 있는 추세이며, 특히 시스템의 안정성이 중요시 되고 있다. 이에 따라 가용성 측면에서, 리눅스 기반의 효율적인 이중화 IPC와 분산처리에 대한 동작 절차의 연구를 지속적으로 수행할 예정이다.

참 고 문 헌

[1] 김봉완, 이형호, "네트워크 프로세서의 응용과 표준화 동향" 電子工學會誌, 第28卷, 第10號, 94~101쪽, 2001년10월.

[2] Linley Gwennap and Bob Wheeler, "A Guide to Network Processors", MicroDesign Resources, 1st Edition, 2000.

[3] 이형호, 김봉완, 안병준 "테라비트 라우터 기술", Telecommunications Review, 第11卷 第2號, 237~247쪽, 2001년4월.

[4] Bup Joong Kim, et al, "Design and Implemen tation of IPC Network in ATM Switching System," ICATM 2001, pp.148-152

[5] J. Moy, OSPF Version2, RFC2328, April. 1998.

[6] Christian Huitema, "Routing in the Internet," 2nd Edition, Prentice Hall, 1999.

[7] F. Baker, "Requirements for IP Version 4 Rout ers," RFC 1812, June 1995.

[8] Wang-Bong Lee, et al, "An Architecture of Distributed Multi-Gigabit IP Router," AIC 24th Conference, Seoul, Nov. 2000.

[9] S. Keshav and R. Sharma, "Issue and Trends in Router Design," IEEE Communications Magazine Vol. 36 No. 5 pp.144-151, June. 1998.

[10] J. Furnuas, et al, "A prototype for interprocess communication support, in hardward," Ninth Euromicro Workshop on Real-Time Systems, pp. 18-24, 1997.

[11] A. Bharagava and B. Bhargava, "Measurements and quality of service issues in electric commer ce software," in Proc. Application-Specific Syst em and Software and Technology, pp.26-33, 1999.

저 자 소 개

조 규 인(학생회원)

1996년 순천향대학교 전자공학과 공학사

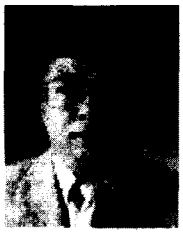
2003년 3월~현재 충북대학교 정보통신공학과 석사과정 재학 중

1995년 12월~2000년 4월 휴니드 테크놀러지스(구, 대영전자) 연구원

2000년 5월~현재 이스텔시스템즈(구, 성미전자) 선임연구원

2002년 4월~현재 한국전자통신연구원 파견원.

<주관심분야: 마이크로프로세서 응용, 네트워크 프로세서 응용, 10기가비트 이더넷 스위치 시스템 설계, 광다중화 시스템 설계>



정 해 원(정회원)

1980년 2월 한국항공대학원 항공통신정보공학과 공학사

1982년 2월 한국항공대학원 항공전자공학과 공학석사

1999년 2월 한국항공대학원 항공통신정보공학과 공학박사

1982년 3월~현재 ETRI , 광대역 통합망 연구단, 라우터 연구그룹, 그룹장

<주관심분야: 무선 LAN, 홈네트워킹, 기가비트 이더넷>

김 유 진(정회원)

2001년 2월 충북대학교 정보통신공학 공학박사 수료

1995년 12월~1999년 5월 LG반도체 MCU설계실 연구원

1999년 6월~현재 ETRI 디지털홈 연구단 선임연구원

<주관심분야는 ASIC설계, 무선 MAC설계, 이더넷시스템설계, 센서네트워킹, 스티리밍기술>

조 경 록(정회원)

1977년 경북대학교 전자공학과 학사

1989년 동경대학교 전자공학과 석사

1992년 동경대학교 전자공학과 박사

1979년~1986년 금성사 TV 연구소 선임연구원

1992년~현재 충북대학교 공과대학 정보통신공학과 교수

<주관심분야: VLSI 시스템 설계, 통신시스템용 LSI 개발 및 고속 마이크로프로세서 설계>