
고속 대용량 라우터의 성능 향상을 위한 R-IPC 프로토콜 성능분석

김수동* · 강태규** · 조경록***

A New R-IPC Protocol for a High-speed Router System to Improve the System Performance

Soo-Dong Kim* · Tae-kyu Kang** · Kyoung-Rok Cho***

요 약

급증하는 인터넷 이용자로 인한 인터넷 트래픽의 폭발적인 증가는 라우터에서 패킷 전달의 병목 현상을 일으켜 망의 성능에 큰 영향을 미치고 있다. 이러한 문제점을 해결하기 위해 고속 대용량 라우터 시스템은 분산형 시스템 구조를 갖는다. 분산형 라우터 시스템은 메인카드 프로세서에서 라우팅 테이블(Routing Table)을 관리하고 IPC(Inter Processor Communication)를 통해 라인카드 프로세서로 포워딩테이블(Forwarding Table)을 전달하게 함으로써 패킷처리가 분산되어 wire-speed로의 포워딩 기능을 가능하게 하여 망의 성능을 개선시키는 효과를 갖는다. 이와 같은 프로세서의 분산은 각종 정보의 원활한 교환을 위해 IPC 기능을 필요로 하며, 특히 이더넷을 이용한 IPC의 구성은 비용 대비 효과 측면에서 주로 사용되고 있다. 그러나 IPC를 통해서 처리하여야만 하는 OAM(Operation, Administration and Maintenance) 및 상위 프로토콜 관련 패킷의 증가로 IPC의 처리에서 병목현상이 발생하게 되었다. 본 논문에서는 기존 IPC의 TCP/IP(또는 UDP/IP)를 통한 2-계층 처리를 단일계층에서 처리할 수 있는 구조인 R-IPC(Reduced IPC) 프로토콜을 제안함으로써 평균 10%이상의 패킷처리 성능개선을 가져왔다.

ABSTRACT

By a tremendous expansion of Internet users, there's a number effects that cause the phenomenon of bottlenecked switching packets from routers. In order to tear down this problem, distributed system is applicable to almost every highly performed router systems. The main processor of distributed system, which manages routing table, commands IPC to delivering the forwarding table line processor that eases functionalities of the router. This makes the system having wired-speed forwarding function based on the hardware so that the performance of the network can be enhanced. Therefore, IPC, which assign a part of router, is necessary to exchange data smoothly and the constitution of IPC using Ethernet is widely adapted as a method for saving investment. In this paper, R-IPC mechanism improve the packet-processing rate over 10% through changed from defect of conventional Ethernet IPC, that is, 2 layer processing to TCP/IP or UDP/IP into 1 layer processing for efficient packet forwarding.

키워드

Router, IPC, R-IPC, 분산형 시스템

*쌍용정보통신

**ETRI

***충북대학교 정보통신공학과

접수일자 : 2004. 07.19

I. 서 론

초기 라우터의 구조는 메인 프로세서가 공유된 버스를 이용하여 입, 출력되는 패킷에 대한 라우팅 계산 및 포워딩에 대한 처리를 하는 구조로써, 집중된 패킷 처리로 인한 메인 프로세서의 과중한 부하는 전체 시스템의 효율을 떨어뜨리는 문제를 발생시켰다. 이러한 비효율적인 성능 한계를 벗어나기 위해 현재의 고속 패킷 스위칭 및 라우팅을 구현할 수 있는 네트워크 전용 프로세서를 사용한 라우터 시스템의 개발이 보편화 되어지고 있다. 이것을 바탕으로 네트워크의 고속 대용량화는 현재 테라비트(Tera bps)의 속도를 지원하기에 이르렀다.[1]

고속 대용량 라우터의 구조는 일반적으로 입력된 패킷 포워딩을 위한 네트워크 프로세서(NP, Network Processor) 제어 및 관리 기능을 담당하는 라인 프로세서(LP, Line Processor)를 갖는 라인 카드(Line Card)와 라우팅 관련 프로토콜 처리 및 시스템 처리 등을 전담하는 라우팅 프로세서(RP, Routing Processor)를 갖는 메인카드(Main Card), 입출력 패킷을 스위칭 해주는 스위치 패브릭(Switch Fabric)을 갖는 스위치 패브릭 카드(Switch Fabric Card)로 구성된다.[2][3]

이러한 고속 대용량 라우터는 정보의 처리량이 수십만 개의 라우팅 엔트리에 까지 이르며, 이것은 시스템의 성능과 밀접하게 관련되기 때문에 얼마나 신속하게 갱신하여 빠른 시간 내에 전달하느냐에 따라 효율이 달라진다. 따라서 시스템의 성능향상을 위해서는 라우팅 프로세서에서 생성된 RIB(Routing Information Base) 정보를 라인 프로세서 내의 모든 네트워크 프로세서로 전달하여 네트워크 프로세서가 보다 정확한 FIB(Forwarding Information Base)를 이용하여 룩업(Lookup) 할 수 있도록 실시간 처리가 되어야 하며, 아울러 OAM 처리를 위해서도 신속한 정보가 전달되어야 한다. 이를 위해 라우터가 처리해야 할 정보교환을 위한 보다 빠른 라인 프로세서와 라우팅 프로세서간의 통신을 위해 IPC 구조를 필요로 한다.[4][5]

최근 QoS, 보안 등 다양한 서비스를 지원하기 위한 OAM 및 상위 프로토콜과 같이 IPC에서 처리해야 하는 패킷의 증가로 IPC에서 병목현상이 발생하게 되었다.

본 논문에서는 고속 대용량 라우터를 위한 효율적인 IPC 통신을 위한 소프트웨어 메커니즘을 제안 한다.

II장에서는 고속 대용량 라우터 구조와 성능향상을 위한 효율적인 R-IPC 통신 메커니즘에 관해 기

술 한다. III장에서는 기존의 UDP/IP를 이용한 IPC 통신 메커니즘과 제안한 R-IPC 통신 메커니즘의 성능을 비교 분석하고, IV장에서 타당성을 결론을 맺도록 한다.

II. 본 론

1. 고속 대용량 라우터 시스템의 구조

고속 대용량의 라우터의 구성은 네트워크 프로세서와 라인 프로세서를 갖는 라인카드와 스위치 패브릭을 갖는 스위치 패브릭카드, 라우팅 프로세서와 이더넷 IPC를 갖는 메인카드 구성된다. 그림1은 고속 대용량 라우터 시스템의 기본 구조를 나타낸다.

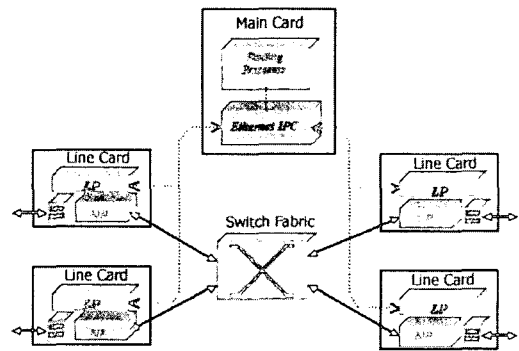


그림 1. 라우터 시스템 구조

라인 카드는 입출력 포트 물리계층 정합 및 미디어 제어 기능을 제공하는 MAC/PHY 블록과 이더넷 패킷 인식, 분석, 어드레스 룩업 및 포워딩 기능을 wire speed로 제공하기 위하여 하드웨어로 구성되는 네트워크 프로세서 블록으로 구성된다. 이것은 입력된 패킷을 포워딩하기 위해 포워딩 테이블을 통해 룩업하여 적절한 포트에 패킷을 포워딩 한다. 스위치 패브릭은 라인카드의 입력 포트에 연결된 MAC/PHY 블록으로부터 전달된 패킷의 목적지를 확인하여 해당 출력 포트의 MAC/PHY 블록으로 패킷을 스위칭하게 한다. 메인 카드는 시스템의 구성 관리, 제어 및 장애 관리, 라우팅 테이블 관리, 망 관리 기능을 수행하는 곳이며, 라우팅 프로세서 블록과 이더넷 IPC 블록으로 구성된다. 라우터가 부팅 될 때 운영체제 및 라우팅 프로토콜을 주 메모리에 적재하고서 라우팅 프로토콜에 따라서 주변 라우터와 데이터교환을 통해서 라우팅

경로 및 변경 정보를 얻어 음으로써 라우팅 기능을 수행한다. 라인 프로세서의 주기능은 라우팅 정보의 교환을 통해서 라우팅 테이블을 만드는 것이다. 라우팅 테이블이 만들어지면 이것을 포워딩 기능을 수행하는 곳에서 사용하기 알맞은 형태인 포워딩 테이블로 변경해서 각 입출력 포트로 전송을 한다. 입출력 포트에서는 라인 프로세서로부터 받은 포워딩 테이블을 이용해서 들어온 패킷의 출력포트를 결정 한다. 여기서 라우팅 프로세서와 라인 프로세서 사이에는 라우팅 정보 및 OAM 등의 정보를 교환하기 위해 이더넷 기반의 IPC 채널을 사용한다.[6][7]

이러한 IPC는 주로 시스템의 내부정보를 처리하기 위한 내부 IPC와 시스템의 외부정보를 처리하기 위한 외부 IPC의 두 종류로 나뉜다. 다음절에서 시스템의 성능을 향상하기 위한 두 가지 IPC에 대한 R-IPC의 소프트웨어 구조를 제안 및 구현한다.

2. R-IPC 메시지

TCP/IP(또는 UDP/IP) 계층을 이용한 방법은 네트워크 계층인 IP 주소를 사용하여 패킷 전달을 처리하는 것으로 동일한 네트워크에서 통신을 하는 시스템 내부에서는 IP 주소를 이용한 통신이 비효율적이며, 두 계층을 거치기 때문에 시간도 많이 걸리는 단점을 갖는다. R-IPC에서는 이러한 단점을 보완하여 하나의 프로토콜 계층으로 처리한다. 즉, R-IPC는 MAC주소를 이용하여 목적지, 소스 주소를 설정하고, 이더넷 프레임에 데이터를 인캡슐하는 방식으로 TCP/IP 또는 UDP/IP 기능을 사용하지 않는다. 이는 IPC를 처리하기 위해 2-계층을 통과하는 패킷 프로세싱 절차를 단일 계층을 통해 패킷 프로세싱 함으로써 시간 단축을 얻을 수 있다는 것을 의미한다. 그림2는 R-IPC의 헤더를 나타낸다.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
daddr				saddr				type	D	S	dport	sport	len	undef									
Ethernet Header											R-IPC Header												

그림 2. R-IPC 헤더

R-IPC 메시지의 이더넷 타입 필드에는 R-IPC패킷을 구분하기 위하여 내부적으로 정의한 값을 사용한다. 소스 포트 번호(sport)는 해당 근원지 응용 프로그램의 포트 번호이며, 목적지 포트 번호(dport)는 목적지 응용 프로그램의 포트 번호로서

메시지가 수신될 응용 프로그램을 명시한다.

3. 내부 R-IPC

내부 R-IPC의 목적은 라우팅 프로세서와 라인 프로세서간의 내부정보에 대한 메시지 교환을 위해 사용된다. 그림3은 내부 R-IPC 소프트웨어 구조 및 흐름을 나타낸다.

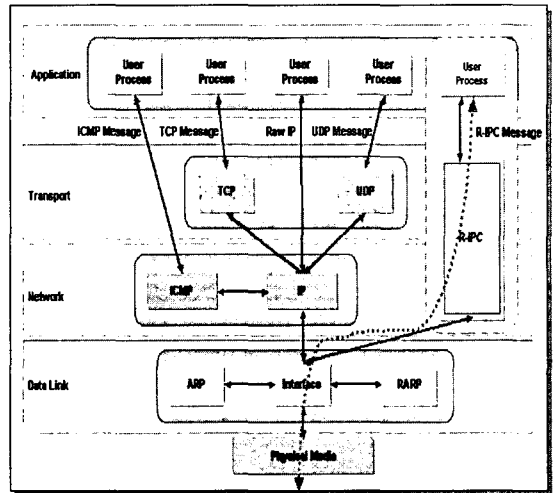


그림 3. 내부 R-IPC 소프트웨어 구조 및 흐름

내부 R-IPC 메시지 전송동작은 다음과 같다. 우선 R-IPC socket이 생성되는데 이때 사용자가 설정한 목적지 MAC 주소가 저장되고 R-IPC header가 생성되어 목적지 MAC주소와 포트 번호로 헤더가 완성된 후에 목적지 MAC주소로 OIF(Outgoing Interface)를 결정(eth0)하고 메시지를 Ethernet Interface Queue에 전달하는 과정을 거치게 된다. 그리고 R-IPC 메시지 수신 과정은 먼저 Ethernet Interface로부터 R-IPC 메시지를 수신한 후 헤더 검사를 통해 목적지 주소, 소스 주소, 소스 MAC주소 확인을 통해서 수신여부 결정하고 목적지 포트와 소스 포트, 목적지 포트번호를 확인하여 해당 블록 프로세스로 전달한다.

4. 외부 R-IPC

외부 R-IPC의 목적은 네트워크 프로세서를 통해 포워딩 되지 않고 라인 프로세서에서 라우팅 프로세서로 전달되는 패킷(Self-packet : IP 목적지 주소가 라우팅 프로세서 주소와 일치하는 패킷, 브로드캐스트 패킷, 멀티캐스트 패킷, 네트워크 프로세서에서 처리할 수 없는 에러 보고가 요구되는 패킷 등)과 라우팅 프로세서에서 라인 프로세서를 거쳐

서 외부로 전달되어야 하는 패킷(My-packet : ARP, OSPF, BGP, IS-IS 등 프로토콜 관련 송신 패킷)을 처리하기 위한 기능이다.

그림4는 외부 R-IPC의 소프트웨어 계층도 및 흐름을 나타낸다.

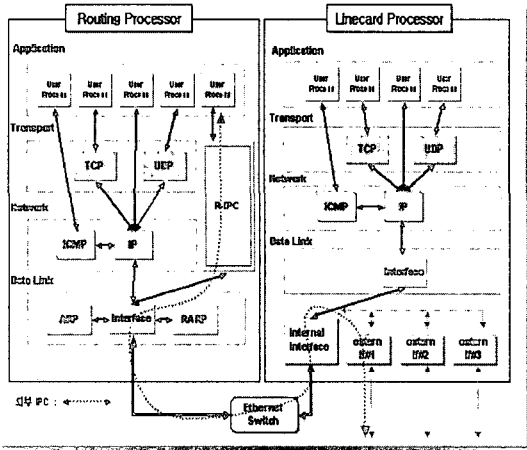


그림 4. 외부 R-IPC 소프트웨어 계층 및 흐름

외부 R-IPC 메시지는 라인 프로세서에서 라우팅 프로세서로 전송하는 경우와 라우팅 프로세서에서 라인 프로세서로 전송하는 2가지의 경우가 있다.

전자는 라인 프로세서에서 보내진 패킷이 라우팅 프로세서에 도착하게 되면 실제 네트워크 디바이스에서 인터럽트를 발생시킨다. 그러면 네트워크 디바이스 드라이버는 헤더를 분석하고 구조체를 초기화 한 후에 패킷을 상위계층(IP Layer)으로 넘겨주는 역할을 하게 되는데 이곳에서 type 정보를 분석하여 가상 네트워크 디바이스 드라이버로 넘겨줘야 할 패킷을 판단한다. 즉, type 정보가 외부 IPC 타입인 0x1002이면 가상 디바이스로 전달되어야 할 패킷임을 알고 뒤따라오는 2바이트의 port정보로 어느 디바이스로 전달할 것인지 결정해서 가상 디바이스 드라이버로 패킷을 전송하게 된다. 그러므로 상부의 모든 Layer는 패킷이 실제 가상 디바이스에서 직접 들어온 것으로 인식하고 처리하게 된다.

후자는 라우팅 프로세서의 상위계층에서 처리된 패킷은 각각의 가상 디바이스로 전달된다. 이는 앞에서 기술한 바와 같이 패킷이 실제 디바이스에서 온 것이 아니고 가상 디바이스를 통해서 왔다고 생각하기 때문이다. 상부에서 보내진 패킷은 가상 디바이스 드라이버에서 헤더가 더해져서 다시 실제 네트워크 디바이스 드라이버로 보내져 라인 프로

세서로 전송되게 된다. 이때 필요한 모든 헤더는 가상디바이스 드라이버에서 추가 된다.

III. 성능 비교 결과

본 장에서는, 먼저 UDP/IP를 이용한 IPC와 R-IPC를 적용한 IPC에 대한 성능을 측정한다. 그림5는 IPC의 비교 측정을 위한 시험 구성이다.

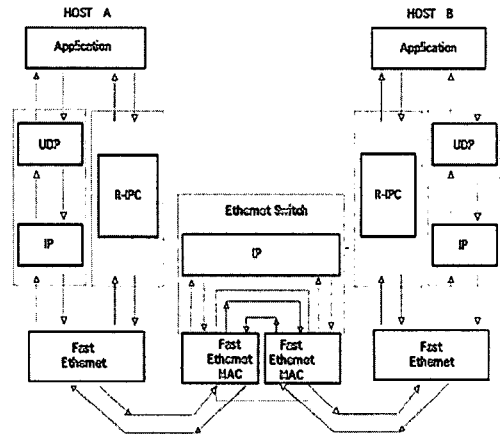


그림 5. UDP와 R-IPC를 이용한 IPC 시험 구성도

먼저, 성능을 측정하기 전 호스트 A와 호스트 B의 사이를 연결하는 100Mbps 이더넷 스위치의 수율(throughput)을 구하는 계산식은 식(1)과 같다. 이더넷 패킷은 8 바이트 (7바이트 Preamble + 1바이트 SFD) 오버헤드와 64~1518 바이트의 데이터로 구성되어 있고 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 존재한다. 여기서 실제 100%의 utilization을 갖는다고 가정할 경우에도 IPG나 Preamble 때문에 실제 수율은 100Mbps가 되지 못하는데, 이때의 수율은 패킷의 길이에 영향을 받게 된다.

$$Packet/sec = \frac{FrameLength[DA+SA+EtherType+데이터길이+FCS]}{FrameLength+Preamble[8Byte]+IPG[12Byte]} \quad \text{-- (1)}$$

그림6은 계측기를 이용한 100Mbps 이더넷 스위치의 실제 측정치를 나타낸다.

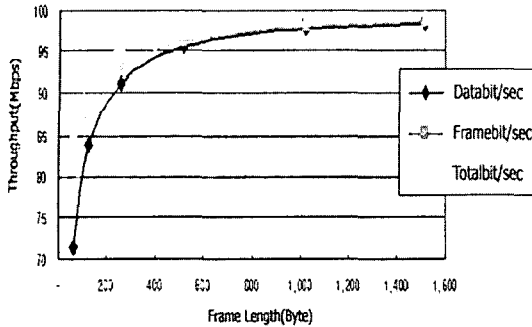


그림 6. 패킷의 길이에 따른 최대 수율 측정값

각각의 정의에 따라서 값의 차이가 있지만, 최소 패킷인 64 byte에서는 최대 수율이 72~86 Mbps 정도가 되며, 최대 패킷인 1518 byte에서는 거의 100Mbps에 도달하고 있다는 것을 알 수 있다.

위와 같은 IPC 채널을 이용하여 측정된 결과, 제안한 R-IPC의 프로토콜은 UDP/IP에 비해 RTT(Round Trip Time)가 64바이트에서 1500바이트 까지 모두 줄어들어 결과적으로 같은 패킷을 처리할 경우 보다 더 빠르게 처리함으로써, 처리속도가 향상되는 결과를 가져왔다. 실험에 적용한 IPC 메시지 길이는 64바이트에서 1500바이트 까지 다양하게 시험 하였다. 성능 측정은 서버 어플리케이션을 이용하였다.[8] 표1은 UDP/IP와 R-IPC의 성능에 대한 결과 값을 보여준다.

표 1. UDP/IP와 R-IPC의 RTT 측정 결과

TYPE	UDP/IP				
송신 Bytes수	64	256	512	1000	1500
Send Packet 수	10000	10000	10000	10000	10000
RTT (msec)	2830.3	4786.1	5399.7	8297.3	10938.9
TYPE	R-IPC				
송신 Bytes수	64	256	512	1000	1500
Send Packet 수	10000	10000	10000	10000	10000
RTT (msec)	2573.1	4321.6	4913.5	7533.6	9735.1
시험결과 (%)	9.0874	9.7052	9.0042	9.2042	11.0048

RTT 측정결과를 보면 UDP/IP를 이용한 경우에 비해 R-IPC를 이용한 경우, 64바이트에서 9.08%, 256바이트에서 9.70%, 512바이트에서 9.00%, 1000바이트에서 9.2%, 1500바이트에서 11.00%가 줄어

들을 알 수 있다.

그림7은 호스트A와 호스트B 사이에 UDP/IP를 통한 수율과 R-IPC를 이용한 수율의 측정 결과를 보여준다. 그림의 결과처럼 패킷 처리 능력에서도 R-IPC가 평균 13.68%정도 향상됨을 확인할 수 있다. IPC 메시지 길이가 64바이트일 경우 15%, 256바이트일 경우 14.6%, 512바이트일 경우 12.5%, 1000바이트일 경우 13.1%, 1500바이트일 경우 13.2%의 성능 개선효과를 보여준다.

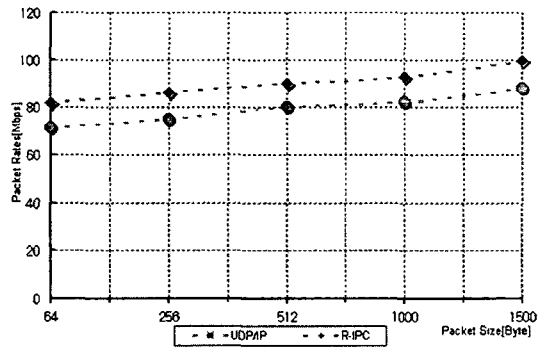


그림 7. UDP/IP와 R-IPC의 수율 비교

IV. 결 론

최근 인터넷 트래픽의 폭발적인 증가와 보안 및 QoS(Quality of service) 관련 어플리케이션이 점점 더 많은 대역폭을 요구함에 따라 네트워크의 고속화를 필연적으로 수반하게 됨으로써 고속 대용량 라우터 시스템이 등장하는 가운데 본 논문에서는 시스템의 성능향상을 위한 방법으로, R-IPC 프로토콜을 이용한 단일계층 처리로 기존의 TCP/IP나 UDP/IP를 통한 2-계층 처리보다 시간을 단축함으로써, IPC 메시지의 전송률을 높이는 방안을 제시 하였다.

측정결과 R-IPC 프로토콜이 UDP/IP를 이용할 때 보다 RTT는 평균 9.6%, 전송률은 평균 13.68%의 성능 향상을 가져왔다. 결과적으로 고속 대용량 라우터 시스템에서는 단일계층으로 처리하는 R-IPC 프로토콜 방식이 전송률의 향상을 통한 시스템 성능향상 측면에서 적합하다는 결론을 얻었다.

참고문헌

- [1] S.Keshav, R.Sharma, "Issues and trends in router design," IEEE Communications Magazine, Vol.36, pp.144-151, May 1998.
- [2] Linley Gwennap and Bob Wheeler, "A Guide to Network Processors", Micro DesignResources, 1st Edition, 2000.
- [3] V.P.Kumar, T.V. Lashman, D. Stiliadis, Beyond Best Effort: Router Architectures for the Differentiated Services of Tomorrow's Internet, IEEE Communications Magazine, Vol.36, pp.152-164, May 1998.
- [4] Bup Joong Kim, et al, "Design and Implementation of IPC Network in ATM SwitchingSystem" ICATM 2001, pp.148-152
- [5] J. Furnuas, et al, "A prototype for interprocess communication support, in hardward," Ninth Euromicro Workshop on Real-Time Systems, pp.18-24, 1997.
- [6] Wang-Bong Lee, et al, "An Architecture of Distributed Multi-Gigabit IP Router,"AIC 24th Conference, Seoul, Nov. 2000.
- [7] S.Keshav and R.Sharma, "Issue and Trends in Router Design" IEEE Communications Magazine Vol. 36 No. 5 pp.144-151, June. 1998
- [8] A.Bharagava and B. Bhargava, "Measurements and quality of service issues in electric commerce software," in Proc. Application-Specific System and Software and Technology, pp.26-33, 1999.

저자소개

김수동(Soo-Dong Kim)



2000년 경기대학교 전자공학과 학사
 2003년~현재 충북대학교 대학원 정보통신공학과 석사과정
 쌍용정보통신 SI기술연구소 선임연구원

※관심분야 : L4스위치, L7 스위치, 멀티 캐스트 프로토콜, Linux 시스템

강태규(Tae-kyu kang)



1998년 전북대학교 전자공학과 학사
 2000년 전북대학교 전자공학과 석사
 2000년~현재 ETRI 광대역통합 망연구단라우터연구부 10GE팀

근무(연구원)

※관심 분야 : 고속 LAN, 통신시스템, 디지털 변복조, 오류정정부호

조경록(Kyoung-Rok Cho)



1977년 경북대학교 전자공학과 학사
 1989년 동경대학교 전자공학과 석사
 1992년 동경대학교 전자공학과 박사

1979년~1986년 금성사 TV 연구소 선임연구원
 1992년~현재 충북대학교 공과대학 정보통신공학과 교수

※관심분야 : VLSI 시스템 설계, 통신시스템용 LSI 개발 및 고속 마이크로프로세서 설계