

큐 분리 및 패킷 분할을 이용한 효율적인 점보패킷 스케줄링 방법

정희원 윤민영*, 고남석*, 김환우**

Effective Scheduling Algorithm using Queue Separation and Packet Segmentation for Jumbo Packets

Bin-Yeong Yun*, Nam-Seok Ko*, Whan-Woo Kim** *Regular Members*

요 약

고속 네트워킹 기술 발전과 더불어 대용량의 데이터 처리는 컴퓨터의 CPU 사이클을 많이 소모하므로 컴퓨터의 성능을 저하시킨다. 따라서 고속의 네트워크 환경에서 컴퓨터 성능을 향상시키기 위해서는 데이터 처리로 소모되는 컴퓨터의 CPU 사이클을 최대한 억제해야 한다. 이러한 방법 중의 하나가 점보그램과 점보프레임 같은 패킷 길이가 긴 점보패킷을 사용하는 것이다. 그러나 점보패킷이 전달 지연에 민감한 VoIP 패킷들과 동시에 처리되는 경우 이 둘 서비스에 질적인 저하를 가져올 수 있다. 뿐만 아니라, 심각한 패킷 손실이 발생된다. 본 고에서는 점보패킷을 수용하는 경우에도 기존의 일반 패킷 전달 지연 및 손실을 거의 동일하게 유지 시킬 수 있는 스케줄링 방법을 제안한다.

Key Words : Jumbograms, Jumbo frame, IPv6, packet scheduling

ABSTRACT

With the advent of high speed networking technology, computers connected to the high-speed networks tend to consume more of their CPU cycles to process data. So one of the solutions to improve the performance of the computers is to reduce the CPU cycles for processing the data. As the consumption of the CPU cycles is increased in proportion to the number of the packets per second to be processed, reducing the number of the packets per second by increasing the length of the packet is one of the solutions. In order to meet this requirement, two types of jumbo packets such as jumbograms and jumbo frames have already been standardized or being discussed. In case that the jumbograms and general packets are interleaved and scheduled together in a router, the jumbograms may deteriorate the QoS of the general packets due to the transfer delay. They also frequently exhaust the memory with storing the huge length of the packets. This produces the congestion state easily in the router that results in the loss of the packets. In this paper, we analyze the problems in processing the jumbo packets and suggest a noble solution to overcome the problems.

1. 서 론

고속의 네트워킹 기술의 눈부신 발전에 의하여 점

퓨터가 처리할 수 있는 용량 이상의 데이터가 네트워크를 통하여 전달되고 있다. 이러한 고속 데이터는 컴퓨터의 CPU 부하를 증가시키므로 컴퓨터의 성능을 저하시킨다. 따라서 컴퓨터 부하를 줄이기

* 한국전자통신연구원(byyun@etri.re.kr), ** 충남대학교 전자공학과
논문번호 : 030298-0716, 접수일자 : 2003년 7월 8일

위해서는 데이터 처리로 소모되는 CPU 사이클을 최대한 억제해야 한다. 이러한 방안중의 하나가 패킷 길이를 증가^[1]시키는 것이다.

프로토콜 별로 패킷 길이에 관한 표준화 동향을 살펴보면, IPv4에서는 최대 64K 바이트 크기의 패킷이 사용될 수 있으며^[2], 기가비트 이더넷에서는 최대 9K 바이트 길이를 갖는 점보 프레임(Jumbo Frames)^[3]의 표준화가 진행되고 있다. 그리고 IPv6에서는 64K 바이트에서 최대 4 기가 바이트 길이를 갖는 점보그램(Jumbograms)^[4]을 표준으로 채택하고 있다. 본 고에서는 기가비트 이더넷의 점보 프레임과 IPv6의 점보그램을 모두 포함할 수 있도록 1518 바이트에서 최대 4G 바이트 길이를 갖는 점보패킷을 정의하여 사용한다.

백본망에서 사용되는 패킷 길이를 조사한 결과^[5] 1500 바이트 이상의 크기를 갖는 점보패킷들이 관찰되었다. 이것은 FDDI 망에서 4K 바이트 길이의 패킷과 ATM WAN에서 9K 바이트 길이의 패킷들이 백본망을 통해서 전달되기 때문이다. 또한 점보패킷은 전달되는 패킷의 개수가 적음에도 불구하고 한번에 전달되는 패킷 데이터양이 많기 때문에 약 50% 이상의 데이터가 점보패킷을 통해서 전달되었다. 따라서 점보패킷을 사용하면, 적은 수의 패킷으로 많은 데이터를 전달하므로 데이터 처리로 소모되는 CPU 사이클을 크게 향상시킨다. 이러한 점보패킷이 사용될 수 있는 대표적인 분야가 SAN(Storage Area Networks) 영역이며, 현재는 ETT/DTF, Internet2, DoE, 그리고 NASA 등의 네트워크를 이용한 점보패킷 연구^[6]가 활발하게 진행되고 있다. 그러나 많은 사람들이 점보패킷의 필요성을 공감하고 있음에도 불구하고 기존 시스템과의 호환성 때문에 점보패킷의 표준화 및 활성화에 어려움을 겪고 있다. 그러나 IPv6의 점보그램은 이미 IETF의 표준으로 채택되었고, 기가비트 이더넷의 점보 프레임이 대부분의 시스템에서 지원되기 때문에 점보패킷을 효율적으로 처리할 수 있는 시스템 개발의 필요성이 대두되었다.

현재 네트워크에서 사용되는 패킷(이하 일반패킷)과 점보패킷을 동시에 처리하는 시스템은 점보패킷의 긴 길이로 인하여 일반패킷에 심각한 전달 지연을 발생시킬 수 있다. 그러므로 VoIP와 같은 전달 지연에 민감한 서비스의 질이 크게 저하될 수 있다. 이러한 서비스 질의 저하는 점보패킷 길이와 물리 링크 속도에 따라 영향을 받는다. 즉 기가비트 이더넷은 100Mbps 이더넷에 비하여 10배 정도 링크 속

도가 증가했기 때문에, 10배 정도 길이가 긴 점보 프레임이 전달되어도 전달 지연이 크게 문제되지 않는다. 그러나 최대 4G 바이트 길이의 IPv6 점보그램을 전달하는 경우에는 이러한 영향을 심각하게 고려해야 한다. 또한 점보패킷은 패킷 저장을 위한 버퍼를 손쉽게 고갈시켜서 심각한 패킷 손실을 초래할 수 있다. 본 고에서는 점보패킷을 수용하는 경우에도 기존의 라우터에서 제공되는 서비스 질을 그대로 유지할 수 있도록 일반패킷의 전달 지연 및 손실을 최대한 억제 할 수 있는 방법을 제안한다. 그리고 제안된 방법과 점보패킷을 고려하지 않는 방법에 대한 시뮬레이션 결과를 비교한다.

II. 제안된 점보패킷 스케줄링

일반적인 패킷 스케줄링 방법은 패킷 길이 보다 는 주로 단위 패킷에 대한 우선순위와 공평성을 고려하여 패킷을 처리한다. 이러한 스케줄링 알고리즘 가운데 가장 기본적인 스케줄링 방법은 Round Robin 방식이며^[7], 그림 1과 같이 일반패킷(General Packet: GP)과 점보패킷(Jumbo Packet: JP)을 위한 별도의 큐를 사용하지 않고 패킷을 처리한다.

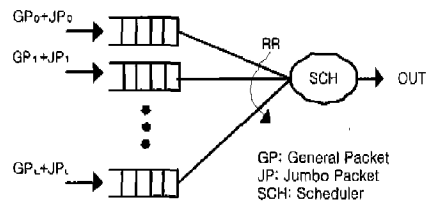


그림 1 점보패킷을 고려하지 않는 패킷 스케줄링

그러나 이러한 스케줄링 방법을 사용하여 점보패킷을 처리하게 되면, 다음과 같은 두 가지 문제가 발생될 수 있다. 첫 번째로 점보패킷에 의하여 일반패킷의 트래픽이 왜곡되는 현상이 발생된다. IPv6의 점보그램은 최대 4 기가 바이트 크기를 갖기 때문에 패킷을 스케줄링 할 때 일반패킷에 심각한 전달 지연을 발생시킬 수 있다. 두 번째로 점보패킷은 트래픽을 버스트(Burst)하게 만들기 때문에 버퍼를 손쉽게 고갈시킬 수 있다. 그러므로 폭주 상태가 자주 발생되어 일반패킷 손실을 심화시킬 수 있다. 그러므로 이와 같은 두 가지 문제점을 효율적으로 대처할 수 있는 스케줄링 방법을 제안한다.

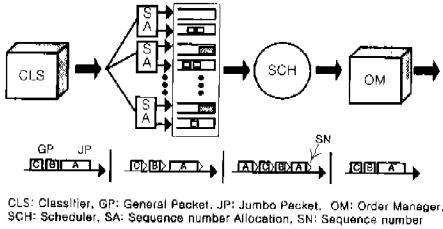


그림 2. 제안된 정보패킷 처리 블록도

제안된 방법에서는 첫 번째 문제를 해결하기 위해서 정보패킷을 분할하여 스케줄링하고 두 번째 문제를 해결하기 위해서 정보패킷과 일반패킷을 각각 독립된 큐에 저장하여 스케줄링 한다. 주요 기능은 그림 2와 같이 스트림을 구분하는 Classifier(CLS), 스트림 별로 패킷의 도착 순서를 표시해 주는 SA(Sequence number Allocation), 수신된 패킷을 저장하기 위해서 스트림 별로 정보패킷 큐와 일반 패킷 큐로 구분된 버퍼, 정보패킷을 효과적으로 처리하는 스케줄러, 그리고 패킷 순서를 보장해주는 Order Manager(OM)로 구성된다. 기능 동작을 살펴 보면 다음과 같다. 패킷이 수신되면, Classifier는 패킷헤더에 포함된 source address, destination address, 그리고 protocol field^[8]를 사용하여 스트림 ID를 결정한다. 그리고 수신된 패킷과 함께 SA로 전달한다. SA는 RED(Random Early Discard)와 같은 폭주 제어 방법^[9] 및 메모리 저장 공간에 따라서 패킷 폐기 및 수용 여부를 결정한다. 그리고 수용된 패킷에 대해서는 각 스트림 별로 SN(Sequence Number) 번호를 할당한다. 또한 SA는 패킷헤더의 total length 필드를 참조하여 수신된 패킷이 정보패킷인지 혹은 일반패킷인지 여부를 식별한다. Total length 필드의 위치는 사용되는 프로토콜에 따라서 다르기 때문에 이를 고려해야 한다^[2,4]. 그런 후 일반패킷이면 일반패킷 큐에 저장하고, 정보패킷이면 정보패킷 큐에 저장한다.

스케줄링은 그림 3과 같은 순서도에 의하여 일반패킷 큐에 대기된 패킷들을 정보패킷에 비하여 우선적으로 서비스한다. 만약 일반패킷 큐에 대기된 모든 패킷들의 서비스가 완료되면, 정보패킷 큐에 대기된 패킷을 서비스한다. 이때 일반패킷의 서비스 질을 보장하기 위해서 정보패킷은 세그먼트 단위로 분할되어 스케줄링 된다. 그리고 한 개의 정보패킷 세그먼트가 서비스될 때 마다 일반패킷 큐에 패킷이 대기되어 있는지 조사한다.

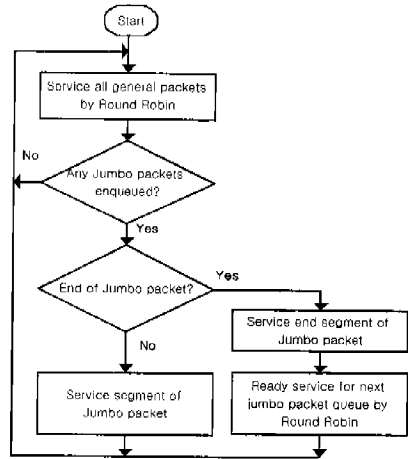


그림 3. 정보패킷 스케줄링 알고리즘

만약 일반패킷 큐에 대기된 패킷들이 있으면 이들을 먼저 서비스한다. 그런 후 대기된 모든 일반패킷들의 서비스가 완료되면, 일시적으로 전송이 중단된 정보패킷을 계속해서 서비스한다. 정보패킷의 분할 크기는 정보패킷이 일반패킷의 서비스를 유지할 수 있는 최대 길이로 정의된다. 즉 패킷 분할 시 정보패킷의 세그먼트가 크면 클수록 일반패킷의 전달 지연이 증가되기 때문에 서비스 저하가 초래된다. 이러한 전달 지연은 물리 링크의 전송 속도와 연관이 있기 때문에 물리 링크의 속도에 따라서 적절하게 패킷의 분할 크기가 결정되어야 한다. 따라서 정보패킷에 대한 세그먼트 분할 크기는 물리링크의 전송 속도를 고려하여 가변적으로 조정된다.

Order Manager는 패킷 순서를 보장해 주는 기능을 수행한다. 동일한 TCP 스트림 안에 일반패킷과 정보패킷이 혼란되어 전달되고 제안된 스케줄링처럼 일반패킷이 정보패킷에 비하여 우선순위를 갖는 스케줄링 방법을 적용하게 되면, 패킷 순서가 바뀌는 현상이 발생된다. 일반적으로 UDP 서비스의 경우에는 패킷 순서가 바뀌어도 아무런 문제가 발생되지 않으나, TCP 서비스 경우에는 패킷 순서가 바뀌지 않아야 한다. 따라서 Order Manager는 동일한 스트림 안의 패킷들의 순서가 유지되도록 SA에서 할당된 번호 순서대로 패킷 순서를 보장하는 기능을 수행한다. 또한 Order Manager는 스케줄링에 의하여 분할된 정보패킷을 재조립하여 출력한다.

III. 시뮬레이션

본 연구에서는 제안된 알고리즘 성능을 평가하기 위해서 기존의 스케줄링 방법과 제안된 스케줄링 방법에 대해서 자체 개발한 시뮬레이터를 사용하여 결과를 비교한다. 기존 스케줄링 방법을 위한 모델은 그림 1과 같이 점포패킷과 일반패킷을 위한 별도의 큐를 갖지 않는다. 그리고 수신된 패킷이 저장될 공간이 없는 경우, 즉 큐의 레벨이 threshold 이상이면, 수신된 패킷은 폐기된다. 그런 후 입력 큐에 대개 된 패킷은 Round Robin 방식에 의해서 서비스된다. 제안된 방법의 시뮬레이션 모델은 그림 4처럼 점포패킷과 일반패킷들이 구분되어 별도의 큐에 저장된다. 스케줄링은 그림 3과 같이 Round Robin 방식의 순서도로 동작하며, 점포패킷의 분할 크기는 이더넷 MTU 길이와 동일한 1518바이트를 사용한다. 이 두 방법의 성능을 평가하기 위해서 패킷의 전달 지연과 패킷 손실율을 계산한다.

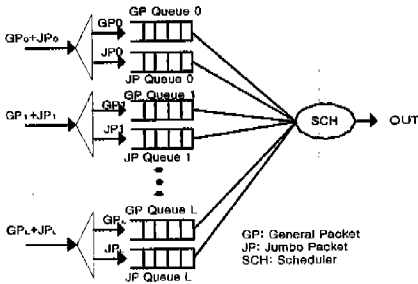


그림 4. 독립된 큐를 이용한 점포패킷 스케줄링

패킷 전달 지연은 입력된 패킷이 출력될 때 까지 걸리는 시간을 말하며, 64 바이트 길이를 기본 단위로 하여 다음과 같은 트래픽 모델을 사용하여 계산된다. 스케줄링 모델은 8개의 입력과 1개의 출력으로 구성되며, 각각의 입력 트래픽은 점포패킷과 일반패킷이 혼합되어 전달된다. 그리고 점포패킷 길이는 평균 32K 바이트이며, 최소 1.5K 바이트에서 최대 64K 바이트를 갖는 포아송 분포를 갖는다고 가정한다. 반면에 일반패킷 길이는 최소 64바이트 이면서 최대 1518 바이트 미만의 포아송 분포를 갖는다. 그림 5는 점포패킷 비율이 증가함에 따라서 64 바이트를 기본 단위로 계산된 일반패킷의 전달 지

연 변화를 보여준다. Case 1은 기존 방법을 사용한 경우이며, Case 2는 제안된 방법을 사용한 경우이다. 기존 방법에서는 점포패킷의 비율이 증가함에 따라서 일반패킷의 전달 지연이 크게 증가한다. 반면에 제안방법을 사용하는 경우에는 일반패킷이 우선적으로 서비스되기 때문에 점포패킷이 증가하면 할수록 일반패킷의 전달 지연이 오히려 감소한다. 그러므로 제안된 방법은 점포패킷에 의한 일반패킷의 전달 지연을 전혀 발생시키지 않는다.

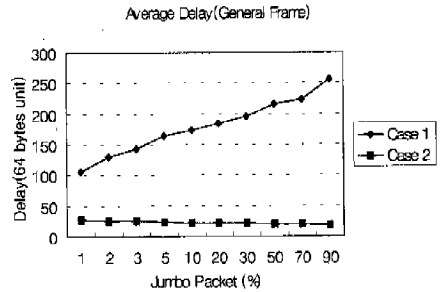


그림 5. 점포패킷 전달 시 일반패킷 전달 지연 비교

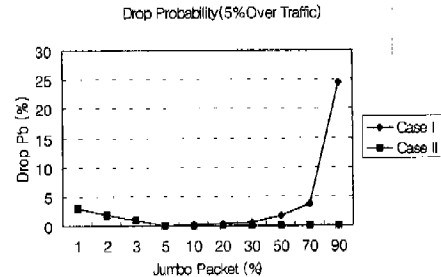


그림 6. 트래픽 폭주가 5% 초과 시 일반패킷 손실율 비교

그림 6은 입력트래픽이 출력트래픽을 5% 초과한 폭주 상태에서 입력된 트래픽의 점포패킷 비율을 증가시킬 때 일반패킷 손실율을 보여준다. 여기서 Case 1은 기존 방법을 사용한 경우이며, Case 2는 제안된 방법을 사용한 경우이다. 그림 6의 Case 1에서는 점포패킷 비율이 5% 미만일 때 모든 점포패킷과 일부의 일반패킷이 손실된다. 그리고 점포패킷이 5% 이상일 때는 점포패킷 비율이 증가함에 따라서 일반패킷 손실이 크게 증가한다. 그러나 제

안된 방법에서는 Case 2 처럼 정보패킷이 5% 미만 일 때 모든 정보패킷과 일부의 일반패킷이 손실된다. 그러나 정보패킷이 5% 이상일 때는 정보패킷만이 손실되고 일반패킷은 손실되지 않는다.

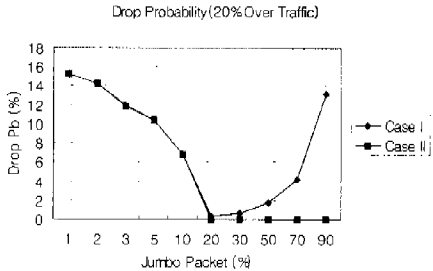


그림 7. 트래픽 폭주가 20% 초과 시 일반패킷 손실을 비교

그림 7은 입력트래픽이 출력트래픽을 20% 초과한 폭주 상태에서 정보패킷의 비율에 따라서 일반패킷의 손실율을 보여준다. 여기서도 이전의 결과와 유사한 형태의 패킷 손실율이 관찰된다. 그래서 제안된 방법을 사용하면, 정보패킷 트래픽이 20%에 이룰 때 까지 일부의 일반패킷이 폐기되나, 20% 이상이 되면 일반패킷의 손실이 발생하지 않는다. 결국 기존의 방법을 사용하는 경우에는 정보패킷 비율이 증가함에 따라서 일반패킷의 손실이 크게 증가함을 보여준다. 그리고 입력트래픽 속도가 동일하더라도 정보패킷 비율이 증가 할수록 버스트한 트래픽 특성으로 인하여 일반패킷 손실이 증가한다. 반면에 제안된 스케줄링 방법을 사용하는 경우에는 폭주 발생시 정보패킷이 우선적으로 폐기되기 때문에 일반패킷의 손실이 발생되지 않는다. 본 시뮬레이션은 최대 패킷 길이를 64K 바이트로 제한하였다. 그러므로 IPv6의 점보그램을 고려한다면, 제안된 방법이 패킷의 전달 지연과 패킷 손실율에 대해서 더욱더 효과적일 것으로 예측된다.

IV. 결론

고속의 네트워크에 연결되는 컴퓨터는 데이터 처리 용량의 증가로 인하여 막대한 컴퓨터 성능을 소모하고 있다. 이러한 환경에서 컴퓨터 성능을 향상시키는 방법 중의 하나가 패킷 길이를 증가시키는 방법이다. 그러나 패킷 길이의 증가는 패킷 전달 지연을 발생시켜서 실시간 서비스의 질을 저하시킬

뿐만 아니라, 패킷의 손실을 심화 시킨다. 본 연구에서는 최대 4기가 바이트 크기의 점보그램을 수용하는 경우에도 기존의 라우터가 제공하는 서비스 질을 그대로 유지할 수 있도록 정보패킷 스케줄링 방법을 제안했다. 제안된 방법에서는 패킷을 스트림 별로 구분하고 정보패킷과 일반패킷을 각각 독립된 큐를 사용하여 저장하고, 정보패킷을 분할하여 스케줄링 하였다. 이때 정보패킷의 분할 크기는 기존의 라우터에서 제공되는 서비스 질을 동일하게 유지할 수 있도록 불리링크의 전송 속도를 고려하여 가변적으로 조정된다.

제안된 방법의 성능을 평가하기 위해서 패킷의 전달 지연과 패킷 손실율을 시뮬레이션 하였다. 기존 방법을 사용하여 패킷 전달 지연을 제한한 결과, 입력트래픽의 정보패킷 비율이 증가함에 따라서 일반패킷의 전달 지연이 크게 증가하였다. 반면에 제안 방법을 사용하면, 일반패킷이 우선적으로 서비스되기 때문에 정보패킷이 증가하면 할수록 일반패킷의 전달 지연이 오히려 감소한다. 또한 기존 방법을 사용하여 패킷 손실을 계산하면, 정보패킷 비율이 증가함에 따라서 일반패킷 손실이 증가함을 보여준다. 그리고 동일한 트래픽 속도를 갖는다 하더라도 정보패킷 비율이 증가 할수록 버스트한 트래픽 특성으로 인하여 버퍼가 손쉽게 고갈되므로 일반패킷 손실이 증가한다. 반면에 제안된 스케줄링 방법을 사용하면, 폭주 발생시 정보패킷이 우선적으로 폐기 되기 때문에 일반패킷의 손실이 발생되지 않는다. 그러므로 제안된 방법은 정보패킷을 수용하는 경우에도 일반패킷의 전달 지연과 손실에 아주 효과적인 임을 발견하였다.

참고 문헌

- [1] Matthew Mathis, Jeffrey Semke, Jamshid Mahdavi, and Teunis Ott, "The Macroscopic Behavior of the Congestion Avoidance Algorithm", *Computer Communications Review*, volume 27, number 3, July 1997
- [2] Postel, J.B., "Internet Protocol," RFC791, September 1981.
- [3] Phil Hochmuth, Match 2002, "Revisiting Jumbo frames" Network-WorldFusion,

<http://www.nwfusion.com/news/2002/0311infra.html>

- [4] D.Borman, S.Deering, R.Hinden., "IPv6 Jumbograms", IETF RFC2675, August 1999
- [5] Phil Dykstra, "Gigabit Ethernet Jumbo frames and why we should care", <http://sd.wareonearth.com/~phil/jumbo.html>
- [6] Pittsburgh Supercomputing Center, "Raising the Internet MTU", <http://www.psc.edu/~mathis/MTU/>
- [7] Sanjay Jha, Mahbub Hassan, "Engineering Internet QoS", Artech House, Chapter3, 2002
- [8] Y. Bernet, S. Blake, D. Grossman, and A. Smith, "An Informal Management Model for Diffserv Routers", IETF RFC3290, May 2002
- [9] S.Keshav, "An Engineering Approach to Computer Networking", pp. 256-261, Addison Wesley, 1999

윤 빈 영(Bin-Yeong Yun)

정회원



1986년 : 중앙대학교 전자공학과 학사
 1991년 : 중앙대학교 전자공학과 석사
 1999년 ~ 현재 : 충남대학교 전자공학과 박사과정
 1991년 ~ 현재 : 한국전자통신연구원

구원

<주관심분야> 라우터 하드웨어, 네트워크 보안, Network Processor

고 남 석(Nam-Seok Ko)

정회원



1998년 : 전북대학교 컴퓨터공과 학사
 2000년 : 한국정보통신대학원 대학교 공학부 석사
 2000년 ~ 현재 : 한국전자통신연구원

<주관심분야> IP QoS, Network Processor, 패킷 스케줄링 알고리즘, 네트워크 보안

김 환 우(Whan-Woo, Kim)

정회원



1977년 : 서울대학교 전자공학과 학사
 1979년 : 한국과학기술원 전기 및 전자공학과 석사
 1988년 : University of Utah 전자공학과 박사
 1980년 ~ 현재 : 충남대학교

정보통신공학부 교수

<주관심분야> 디지털전송기술, 신호처리기술