

무선 셀룰러 네트워크에서 TCP 처리율 향상을 위한 기술

조용범*, 고봉진**, 조성준*

* 한국항공대학교, ** 창원대학교

목 차

I. 서 론	III. TCP 성능 향상 기술
II. 무선 네트워크에서의 TCP	IV. 결 론

I. 서 론

최근 들어 무선 통신과 무선 컴퓨팅이 나날이 인기를 얻고 있다. 무선 네트워크들이 미래의 통신 시스템에 있어서 매우 중요한 역할을 할 것이라는 것은 자명해 보인다. 무선 네트워크 기술들은 사용자들로 하여금 무선 단말들을 사용하여 정보 검색, 화상 전화, E-mail, 뉴스 검색, 증권 거래와 같은 여러 서비스들을 언제 어디서나 이용하는데 불편함도 없도록 해줄 것이다. 현재 인터넷을 포함한 대부분의 네트워크에서는 TCP를 전송 계층(transport layer) 프로토콜로 사용하고 있는데, 이것은 유선 네트워크를 기준으로 설계되었기 때문에 유선 네트워크에서는 좋은 성능을 발휘하지만, 네트워크 중간에 하나의 무선 링크라도 존재하면 TCP 처리율(throughput)은 현저히 떨어지게 된다. 이론적으로는 TCP 프로토콜을 이동 호스트(MH: Mobile Host)와 유선 네트워크간의 통신에 사용할 수 있지만, 무선 이동 통신에서 필요한 이동성(mobility), 핸드오프(handoff)와 같은 특성들을 만족시킬 수 없어 성능의 저하가 발생할 수밖에 없다. 이러한 문제점을 해결하기 위해 다양한 프로토콜이나 기법들이 개발, 제안되어 왔다.

본 논문에서는 무선 환경이 TCP 성능에 미치는 영향에 대해 알아보고, 무선 셀룰러 시스템(wireless cellular system)에서 TCP의 성능이 저하되는 문제를 해결하기 위해 제안된 여러 기술들에 대해 알아본다.

II. 무선 네트워크에서의 TCP

TCP는 유선 링크와 정지 호스트(FH: Fixed Host)들로 구성된 네트워크에서 잘 동작하도록 발전되어왔다. TCP는 종단간(end-to-end) 패킷의 전송 시간(RTT: Round Trip Time)을 이용하여 네트워크의 상태를 살펴 피게 된다. 즉, 패킷 전송 시간이 급격하게 증가되는 경우 네트워크에서 혼잡(congestion)으로 인한 패킷 손실이 발생한 것으로 간주하여 송신단에서 TCP 패킷 전송률을 급격히 낮추어 네트워크가 혼잡 상태에서 벗어날 수 있도록 한다[1]. 이런 혼잡 제어 방식은 인터넷과 같은 네트워크에서 전체적인 TCP 성능을 향상시킬 수 있음이 확인되었다. 유선 네트워크에서는 대부분의 패킷 손실이 네트워크 혼잡으로 인해 발생하기 때문에 이런 방식이 효과적으로 적용 가능하다. 하지만, 무선 네트워크에 있어서는 네트워크 혼잡과는 관련 없는 패킷 전송 지연이나 패킷 손실이 발생하게 되는데 그 이유는 다음과 같다.

- 한 셀(cell)에서 다른 셀로의 핸드오프 도중에는 통신이 중단되며, 완전히 핸드오프가 완료되어 MH로의 다른 경로가 설정된 후에야 패킷 전송이 재개된다.
- MH가 무선 통신이 가능한 영역 밖으로 벗어났을 경우 패킷이 손실된다.
- 무선 링크에서의 잡음 및 페이딩의 영향으로 인한 전송 에러가 발생하게 되어 빈번한 패킷 손

실이 발생하게 된다.

위와 같은 전송 지연과 패킷 손실로 인해 얼마간의 TCP 성능 저하는 피할 수 없다. 무선 네트워크에서는 네트워크 혼잡이 아닌 다른 이유 때문에 패킷 손실이 발생하므로 유선 네트워크에서와 같이 무조건 혼잡 제어를 수행하게 되면 더 심각한 성능 저하가 초래되게 된다.

TCP는 항상 ACK(acknowledgment)가 얼마 만에 되돌아오는지를 측정하고, 이것의 평균을 취하여 다음 ACK 도착 시간을 예측하게 된다. 만약 현재의 패킷 전송 시간이 예측된 전송 시간의 2배를 초과하게 되면, TCP는 패킷이 네트워크에서 손실된 것으로 판단하여 TCP는 손실된 패킷을 재전송하고, 혼잡 제어 과정을 수행함으로써 네트워크가 혼잡에서 벗어날 수 있도록 배려한다. 또한 재전송 타이머의 시간 값을 exponential backoff 설정 방법에 따라 재설정한다.

만약 네트워크 혼잡 때문에 발생된 패킷 손실이 아닌데 이와 같은 혼잡 제어를 수행하게 되면, TCP 처리율은 급격히 낮아지고 패킷 전송 지연이 더욱 커지게 된다. 이렇듯 TCP를 무선 네트워크에서 사용하는 경우에는 처리율에 심각한 저하를 가져오게 되는데 그 이유는 다음과 같다[3].

2.1 제한된 용량 (Limited capacity)

무선 통신에서 스펙트럼은 얼마 안 되는 귀한 자원이다. 무선 링크에서의 제한된 주파수 대역폭 (frequency bandwidth)과 용량은 결과적으로 무선 링크에서의 낮은 최대 데이터 전송 속도를 초래하게 된다. 높은 비트 에러율(BER: Bit Error Rate) 문제는 FEC(Forward Error Coding)와 ARQ(Automatic Repeat reQuest) 기술 덕분에 부분적으로 해결되었지만, 에러 정정이 필요 없을 경우에도 귀중한 무선 대역폭을 낭비하는 문제점을 내포하고 있다.

2.2 높은 전송 에러율 (High loss probability)

무선 링크는 높은 BER로 인하여 패킷 손실이 많이 발생하게 된다. ACK가 정해진 시간 안에 TCP 송신단에 도달하지 않은 경우 송신단은 패킷을 재전송하고 재전송 시간을 지속적으로 증가시키며, 혼잡 윈도우

(congestion window)를 1로 초기화 시킨다. 따라서, 무선 링크에서의 반복적인 전송 에러는 송신단에서의 혼잡 제어를 유발하게 되어 TCP 처리율이 낮아지게 된다. 또한 무선 링크에서의 패킷 에러는 페이딩(fading)에 의해 산발적으로 발생하는 것이 아니고 연 집적(burst)으로 발생하게 된다. 패킷이 심각하게 연 집적으로 손실되는 경우에는 FEC를 사용하여도 에러를 복구할 수가 없어서 더욱 심각한 TCP 성능 저하를 초래하게 된다.

2.3 높은 종단간 지연 (High end-to-end delay)

무선 링크는 대역폭이 제한되어 있기 때문에 유선 링크에서의 데이터 전송률만큼 빠를 수는 없다. TCP 연결 중간에 무선 링크가 존재하는 경우에는 종단간 지연(end-to-end delay)이 증가하게 되고 데이터 전송률은 저하되게 된다. 그리고 무선 링크에서 높은 에러율을 대비하기 위한 코딩(coding)과 인터리빙(interleaving)은 무선 링크에서의 지연을 더욱 증가시켜, 무선 링크에서의 지연이 100ms에 이를 만큼 커질 수 있다.

2.4 빈번한 연결 절단 (Frequent disconnections)

절단(disconnection)이란 링크 품질이 순간적으로 나빠져서 MH가 데이터 수신이 불가능하게 되는 것을 의미한다.

핸드오프는 패킷 전송 지연을 초래하게 되는데, 그 기간은 수 개의 패킷 시간으로부터 수 개의 프레임들에 해당하는 시간이다. 만약 한 셀에 다수의 사용자가 존재하고 있을 때 새로운 사용자가 셀로 진입하는 경우, 대역폭을 할당 받기 위해서는 굉장히 오랜 시간을 기다려야 하기 때문에도 패킷 전송 시간 지연이 발생할 수 있다. 호 블로킹(call-blocking) 또한 절단의 한 종류로 볼 수 있으며, 이러한 절단들은 TCP의 타임아웃(timeout)과 혼잡 윈도우의 저하를 초래하여 처리율이 저하되게 만든다. MH가 다시 연결이 되더라도 데이터 수신을 위해서는 송신단이 타임아웃 되기를 기다려야 한다. 이런 문제는 피코 셀(pico-cell)과 같이 작은 사이즈의 셀이 도입될 앞으로의 무선 시스템에서는 더욱 심각해질 것이다.

III. TCP 성능 향상 기술

무선 통신에서는 무선 링크에서의 높은 전송 에러로 인하여 많은 패킷 손실이 발생하게 된다. 무선 링크와 같이 높은 에러율을 가지는 시스템에서 사용하기 위한 새로운 여러 TCP 버전들이 제안되어왔다. SACK은 선택적 ACK(selective acknowledgment) 옵션을 사용하여 송신단으로 하여금 손실되거나 전송 지연된 패킷들을 보다 효과적인 방법으로 재전송을 가능하게 한다. New-Reno는 중복 ACK를 수신한 후에 새로운 데이터에 대한 ACK를 수신했을 때 수정된 혼잡 제어 알고리즘을 사용하여 송신단의 재전송 효율을 높인다. 이 두 방법은 무선 링크에서 테스트 결과 TCP 성능을 상당히 향상시킬 수 있음이 확인되었지만, 이 방법들은 유선 네트워크에 존재하는 수많은 FH들의 TCP 프로토콜의 수정을 필요로 하기 때문에 실제 적용하기에 어려움이 많다. 본 논문에서는 FH 측 TCP 프로토콜에 어떠한 수정도 없이 TCP 성능을 향상시키는 알고리즘에 대해 소개하고 각 방식에 대한 간략한 동작 원리를 알아보도록 한다. 특히, MH와 유선 네트워크 사이에 무선 링크가 가장 마지막 링크가 되는 무선 셀룰러 시스템(wireless cellular system)에서 사용되는 알고리즘에 대해 살펴보도록 한다.

3.1 종단간(End-to-end) 방식

(1) 빠른 재전송 방식 (Fast retransmit)

빠른 재전송 방식은 핸드오프 이후에 송신단에서의 타임아웃을 기다리는 것을 피하기 위해 제안되었다. 핸드오프 중에는 패킷들은 손실되거나 전송 지연되게 되는데, 이것은 TCP 송신단에서 타임아웃을 초래하게 된다. 만약 재전송 타이머 설정 시간이 핸드오프 시간보다 훨씬 길 경우에는 수신단은 송신단으로부터 패킷이 재전송 되기까지 오랜 시간을 기다려야 한다. 이를 방지하고자 제안된 이 기법은 MH가 핸드오프를 수행하자마자 가장 마지막 ACK를 3차례 송신단으로 전송한다. 이것은 송신단으로 하여금 빠른 재전송/복구(fast retransmit/recovery)를 수행하게 한다. 이 방식은 짧은 단절이 많은 경우에 추천할만하다. 장점은 핸드오프에 인한 단절 시간을 줄일 수 있으며, 종단간 semantic을 유지하면서 TCP를 모바일 컴퓨팅 환경에

적용할 수 있다는 것이다. 반면, 이 방식은 MH에서의 TCP 코드를 수정해야하고, 무선 링크의 에러 특성을 전혀 고려하지 않는다는 것이 단점이다[2].

3.2 연결 분할 방식 (Splitting the connection)

네트워크에서 유선 구간과 무선 구간은 전혀 다른 링크 성질을 가지고 있다. 연결 분할 방식들은 네트워크를 유선 구간과 무선 구간으로 분리시켜서 각 구간에서 따로 손실 복구(loss recovery)를 수행하게 한다.

(1) Indirect-TCP (I-TCP)

I-TCP는 연결 분할 방식에서 가장 먼저 제안된 프로토콜 중 하나이다. TCP 연결은 FH에서 BS(Base Station)까지와 BS와 MH 사이 두 개의 부 연결(sub-connection)로 분리된다. 이렇게 분리하고 BS가 무선 링크에서 발생한 에러들에 대한 재전송을 책임짐으로써 TCP 송신단에 패킷 에러를 감추어 송신단의 종단간 패킷 재전송을 방지할 수 있다. 게다가 BS가 프락시(proxy) 역할을 수행함으로써 RTT가 줄어들게 되어TCP 처리율이 높아지게 되고, 혼잡 윈도우가 줄어들었을 때 빨리 윈도우 크기를 증가시킬 수 있다 [5].

I-TCP는 연결을 분리하기는 하지만 여전히 무선 링크에서 TCP를 사용한다. 그러나 무선 링크에서는 보다 무선에 최적화가 된 프로토콜이 더 유리하다. 이렇게 무선에 최적화된 프로토콜을 사용한 방식이 M-TCP인데, 이 방식은 유선 구간에서는 TCP를 사용하고 무선 구간에서는 선택 반복(selective repeat) 프로토콜을 사용하게 된다.

이 두 가지 방식(I-TCP, M-TCP)의 큰 단점은 종단간 semantic을 유지하지 못한다는 점이다. 즉, FH에서 보내진 패킷에 대한 ACK가 MH에 도착하기 전에 BS에서 먼저 ACK를 생성해 보내게 된다. 또한 동시에 많은 MH가 유선 네트워크에 동시에 접속하는 경우 BS의 버퍼가 쉽게 오버플로(overflow)가 되어서 유선 링크 쪽으로 이미 ACK를 전송한 패킷을 손실하여 MH로 다시 재전송 할 수 없는 경우가 발생하기도 한다. 이것은 BS가 무선 링크 상태를 고려하지 않고 FH로부터의 패킷을 수신하기 때문에 발생하는 문제이다.

(2) M-TCP

M-TCP 방식은 셀룰러 네트워크를 위해 설계된 프

로토크로 무선 링크가 핸드오프와 같은 단절을 겪을 때 송신단이 slow start를 수행하지 않도록 방지하기 위해 고안되었다. M-TCP는 연결을 분리하는 방식을 사용하고 있지만, 분리는 BS에서 수행되는 것이 아니라 Supervisor Host(SH)라는 더 높은 레벨에서 수행된다. TCP 송신단과 SH 사이의 유선 링크에서는 TCP가 사용되고, SH와 MH사이의 무선 구간에서는 M-TCP가 사용된다.

중단간 semantic을 유지하기 위해서 패킷들 수신단에 도착하기 전에는 SH에서 ACK를 생성해 전송하지 않는다. SH는 항상 MH로부터 수신한 마지막 ACK를 곧장 TCP 송신단으로 전송하지 않고 저장해 둔다. 이런 버퍼링(buffering)의 목적은 MH가 핸드오프와 같은 이유 때문에 연결이 끊어졌을 경우 SH는 이 사실을 인지하여 이전에 버퍼에 저장해 두었던 마지막 ACK를 송신단으로 전송한다. 이 ACK의 advertisement 윈도우의 크기는 0으로 설정되어 있어서 TCP 송신단으로 하여금 persist mode로 진입하도록 하여 혼잡 제어가 수행하는 것을 방지한다. MH가 다시 연결되는 경우에는 중간 노드들이 새로운 ACK(new ACK일 필요는 없음)을 송신단 측으로 전달하여 TCP 송신단이 persist mode를 빠져나와 이전의 속도로 데이터를 전송한다.

M-TCP는 주로 핸드오프와 같은 연결 단절로 인해 발생하는 문제점을 해결하기 위해 제안되었기 때문에 시간에 따른 무선 링크의 변화와 링크에서의 높은 에러율에 대해서는 고려하지 않았다는 단점을 가지고 있다[6].

(3) WTCP (Wireless TCP)

BS는 TCP와 WTCP 두 개의 프로토콜 스택을 가지면서 TCP 연결을 모니터링 한다. FH와 MH에서의 TCP 수정은 필요 없다.

BS가 송신단으로부터 데이터 패킷을 수신하면, 각 패킷의 수신 시간과 순차 번호(sequence number)를 저장해둔다. 각 패킷이 MH로 보내질 때마다 WTCP는 해당 패킷이 WTCP 버퍼에서 머물렀던 시간을 패킷 헤더에 저장된 송신단에서 전송한 시간인 타임 스탬프(time stamp)에 더하게 된다. 이런 방법으로 송신단은 패킷이 실제로 무선 링크로 전송된 시간을 알 수 있게 된다. 이렇게 타임스탬프를 수정하여 송신단을 속이는 것은 RTT 측정에 왜곡이 발생하지 않게 하

기 위함이다. 왜냐하면 BS에서 몇 번의 재전송 끝에 전송된 패킷의 경우, RTT 값이 굉장히 커지게 된다. 더군다나 TCP는 smoothed RTT를 사용하는데, 이런 경우 커졌던 RTT 값이 다시 작아지는데 많은 시간이 걸리게 된다. BS는 MH가 실제로 해당 패킷을 수신하고 ACK를 전송해 왔을 때에만 FH측으로 ACK를 전송하기 때문에 중단간 semantic은 유지된다.

BS는 중복 ACK나 지역 타임아웃에 의해 손실된 패킷들은 지역 재전송(local retransmission)을 통해 복구된다. 지역 타임아웃의 경우 혼잡 윈도우를 1로 줄이게 되는데, 이는 손실된 패킷이 무선 링크에서의 연결된 패킷 손실의 첫 번째 패킷일 수 있기 때문이다. 반면, 중복 ACK를 수신하는 경우 패킷을 곧장 재전송 하는데, 이는 중복 ACK를 수신했다는 것은 무선 링크의 품질이 좋다는 증거이기 때문이다.

RTT에 관련해 송신단을 속인다는 것은 장점이자 단점이 될 수 있다. 즉, RTO가 RTT 추정과 밀접한 관계에 있기 때문에, 링크 품질이 나빠서 BS의 버퍼에 오랫동안 머물렀던 패킷들은 RTT 추정값이 작아짐으로 해서 RTO도 작아져 결과적으로 송신단에서의 타임아웃을 보다 자주 발생하도록 할 수 있다. 또한, 잘못된 추정된 RTT는 처리율의 저하를 가져올 수 있다 [4].

일반적으로 연결 분할 방식들은 BS에서 두 가지 프로토콜 스택을 가져야 하고, 큰 버퍼 용량을 필요로 하기 때문에 구현함에 있어서 소프트웨어에 과부하가 걸리게 된다.

3.3 링크 레이어 방식 (Link layer solutions)

높은 BER의 문제는 무선 링크에 관련된 것이기 때문에, 무선 링크에서의 패킷 손실을 지역적으로 해결하는 것이 바람직할 수 있다.

(1) 링크 레이어에서의 ARQ

ARQ 방식은 UMTS의 RLC 레이어와 같은 MAC (Medium Access Control) 레이어에서 동작하는데, 무선 셀룰러 시스템들에서 많이 구현되어 있다. 이 방식은 BER을 성공적으로 줄일 수 있으나, 재전송 타이머 값을 설정함에 있어서는 매우 신중해야 한다. 즉, 적절히 TCP와 ARQ 타이머 값이 설정되지 않을 경우에는 두 레이어에서 재전송을 서로 다투게 되어 TCP 성능 저하를 불러올 수 있다. 게다가 ARQ는 무선 연결

이 끊어졌을 때에는 사용할 수 없기 때문에, MH가 연결이 끊어져서 제시간에 ACK가 도착하지 않는 경우 TCP 송신단에서 타임아웃이 발생하게 된다[7].

(2) FEC (Forward Error Coding)

FEC는 물리 계층에서 적용될 수 있으나, 그 사용이 제한적이다. 왜냐하면 FEC는 정해진 개수만큼의 에러 난 비트를 복구할 수 있지만 연산을 위한 시간이 필요하게 되어 패킷 지연이 발생하게 되기 때문이다. 또한 FEC는 에러 정정이 필요하지 않을 경우에도 귀중한 대역폭을 낭비할 수 있다는 문제점도 가지고 있다. 몇몇 구현된 FEC는 BER에 따라 적응적으로 동작하게 설계되어 있다. 즉, 코딩율(coding rate)은 무선 링크의 BER이 낮은 경우에는 높아지고, BER이 높은 경우에는 낮아지도록 설정된다. 이렇게 코딩 정도를 적응적으로 수행함으로써 채널이 나쁜 경우에는 코딩을 사용하고, 채널이 좋은 경우에는 코딩을 사용하지 않으므로써 대역폭 낭비를 줄일 수 있다[8].

ARQ와 FEC 방식은 전송 계층에서의 패킷 손실을 효과적으로 줄일 수 있고, 프로토콜 스택의 계층적 구조에도 잘 맞는다. 그러나 핸드오프 동안에 발생하는 연결 절단에 대해서는 고려하지 않고 있다는 단점이 있다.

(3) Snoop-TCP

Snoop-TCP는 전송 계층 레이어인 TCP의 정보를 이용하는 방식이긴 하지만 종종 링크 레이어 방식으로 구분되어진다. TCP 연결은 연결 분할 방식들과 마찬가지로 두 개의 부 연결로 나뉘지만, BS에서는 전송 계층 레이어가 동작하지 않는다.

Snoop 프로토콜은 Snoop agent라는 모듈을 BS에 도입하고 있다. Snoop agent는 양측으로부터의 전송된 수신단에서 수신 확인이 되지 않은 모든 패킷들을 버퍼에 저장한다. BS가 무선 링크에서의 패킷 손실을 검출하면(중복 ACK를 수신하던지 지역 타임아웃이 발생하던지), Snoop agent는 MH로 패킷을 재전송하고 중복 ACK를 폐기함으로써 TCP 송신단에서의 불필요한 재전송 및 혼잡 제어 수행을 방지한다. Snoop 프로토콜은 다른 방식들에 비해 구현이 용이하고, 종단간 semantic을 유지하며 TCP 성능 향상도 크다. 하지만, 무선 링크에서의 패킷 손실을 완전히 감출수가 없어 TCP 송신단에서의 혼잡제어가 발생할 수 있다

는 단점을 가지고 있다[3].

IV. 결론

높은 에러율로 특징이 지어지는 무선 환경에서 TCP를 적용할 때 발생하는 여러 문제들을 해결하기 위한 많은 방법들이 연구되어지고 있다. 본 논문에서는 지금까지 제안되어온 여러 알고리즘에 대해 간단히 정리하였다. 특히 유선 네트워크에 존재하는 수많은 FH에서의 TCP 프로토콜의 변경 없이, BS와 MH에서의 코드 수정만으로 무선 링크에서의 패킷 손실을 보완하는 방식들에 대해 알아보았다.

TCP 프로토콜을 무선 링크에 적용하였을 때 처리율을 저하를 방지하기 위한 방법들로서는

- TCP에게 패킷 손실이 무선 링크에서의 전송 에러 때문에 발생한 것이라는 것을 알리는 방법
- BS나 MH의 프로토콜들을 수정하여 TCP 송신단을 속여 혼잡 제어를 수행하지 않도록 하는 방식이 있다.

위 두 방식을 동시에 사용하는 것이 무선 네트워크에서 TCP 처리율을 향상시킬 수 있는 좋은 방법이 될 것이다.

참고문헌

[1] Richard Stevens, *TCP/IP Illustrated*. vol. 1, Addison-Wesley, 1994.
 [2] R. Caceres and L. Iftode, "Improving the performance of reliable transport protocols in mobile computing environments," *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 5, June 1995.
 [3] H. Balakrishnan, S. Seshan, and R. Katz, "Improving reliable transport and handoff performance in cellular wireless networks," *ACM Wireless Networks*, December 1995.
 [4] K. Ratnam and I. Matta, "WTCP: An efficient mechanism for improving TCP performance over wireless links," *Proc. IEEE Symposium Computers and Communications*, June 1998.
 [5] A. Bakre and B. Badrinath, "I-TCP: Indirect TCP for

mobile hosts," ICDCS, October 1994.

[6] K. Brown and S. Singh, "M-TCP: TCP for mobile cellular networks," *Computer Communication Review* (a publication of ACM SIGCOMM), vol. 27, no. 5, October 1997.

[7] D. Huang, J. Shi, "Performance of TCP over radio

link with adaptive channel coding and ARQ," *Proc. of VTC Spring 1999*.

[8] R. Yavatkar and N. Bhagawat, "Improving end-to-end performance of TCP over mobile internetworks," *IEEE Workshop on Mobile Computing Systems and Applications, 1994*.

저자소개

조용범



1996년 한국항공대학교 항공통신정보공학과 졸업 (공학사)

1998년 한국항공대학교 대학원 항공통신정보공학과 졸업 (공학석사)

2003년 9월~현재 한국항공대학교 대학

원 정보통신공학과 박사과정

1996년~2003년 (주)스탠더드텔레콤 CDMA 개발실

※관심 분야: Wireless TCP, 모바일 인터넷, WMAN

조성준



1969년 한국항공대학 항공통신공학과 졸업 (공학사)

1975년 한양대학교 대학원 전자통신공학과 졸업 (공학석사)

1981년 오사카대학 대학원 통신공학과 졸

업 (공학박사)

1972년~현재 한국항공대학교 대학원 항공전자 및 정보통신공학부 교수

※관심 분야: 무선통신, 이동통신, 환경전자공학

고봉진



1995년 한국항공대학교 대학원 항공전자공학과 졸업 (공학박사)

1994년~1996년 인하공업전문대학 통신과 조교수

1997년 한국전자통신연구원 초빙연구원

1996년~ 현재 창원대학교 전자공학과 교수

2003년~2005년 창원대학교 산업정보대학원 부원장

2004년~2005년 창원대학교 산학협력중심대학사업단장

※관심 분야: 4G 이동통신, D-TV, HAPS