

IEEE 802.11 기반 무선랜에서 TCP 인지 서브 계층 TAS

(TAS: TCP-Aware Sub-layer over IEEE 802.11-based wireless LANs)

최낙중[†] · 정하경[†] · 류지호[†]

(Nakjung Choi) (Hakyung Jung) (Jiho Ryu)

석용호^{**} · 최양희^{***} · 권태경^{****}

(Yongho Seok) (Yanghee Choi) (Taekyoung Kwon)

요약 본 논문은 IEEE 802.11 기반 무선랜 환경에서 TCP 성능을 향상시키기 위한 새로운 분할 TCP 기법을 소개한다. IEEE 802.11 기반 무선랜 환경에서는 유선 환경과는 달리 TCP 데이터 흐름(flow)이 많은 데이터를 보내려고 시도하지 않기 때문에 TCP 데이터 흐름 성능 저하의 주요한 원인이 된다. 본 논문에서는 이런 문제를 완화하여 TCP 데이터 흐름의 성능을 향상시키기 위한 TAS (TCP-Aware Sub-layer) 기법을 제안한다. TAS 기법은 하나의 TCP 데이터 흐름을 AP 등의 분할 지점을 기점으로 두 개의 TCP 데이터 흐름으로 나누는 기존의 분할 연결(split-connection) 기법을 확장한 개념이다. TAS를 기반으로 동작하는 무선 노드는 실제로 TCP ACK을 수신하는 것이 아니라, 수신된 MAC ACK을 이용하여 TCP ACK을 에뮬레이션한다. NS2 모의 실험을 통하여 제안된 기법인 TAS 기법의 성능을 기존 TCP 기법, I-TCP (Indirect TCP) 기법의 성능과 비교하였으며 모의 실험 결과는 TAS 기법이 다른 기법들에 비하여 시간당 처리량과 자원 할당의 형평성 측면에서 더 좋은 성능을 보인다는 것을 증명한다. 또한 절전 모드의 경우, 전송 지연 시간도 줄일 수 있다.

키워드 : 분할 연결, IEEE 802.11, TCP, 데이터 흐름 제어, 혼잡 제어, 형평성

Abstract This paper introduces a new split-TCP approach for improving TCP performance over IEEE 802.11-based wireless LANs. TCP over wireless LANs differently from wired networks is not aggressive, which is a fundamental reason for poor performance. Therefore, we propose TAS (TCP-Aware Sub-layer) to mitigate this problem. Our scheme extends the split-connection approach that divides a connection into two different connections at a split point such as an access point (AP). Using TAS, a wireless node emulates TCP ACK packets using MAC ACK frames, instead of receiving real TCP ACK packets. We compared TAS with both normal TCP and I-TCP (Indirect TCP) by NS2 simulation. Results show that TAS achieves higher throughput, more fair resource allocation and, in power-saving mode, shorter delays.

Key words : split-connection, IEEE 802.11, TCP, flow control, congestion control, fairness

[†] 학생회원 : 서울대학교 전기컴퓨터공학부
fomula@mmlab.snu.ac.kr
hkjung@mmlab.snu.ac.kr
jhryu@mmlab.snu.ac.kr
^{**} 정회원 : 서울대학교 전기컴퓨터공학부
yhseok@mmlab.snu.ac.kr
^{***} 종신회원 : 서울대학교 전기컴퓨터공학부 교수
yhchoi@mmlab.snu.ac.kr
^{****} 정회원 : 서울대학교 전기컴퓨터공학부 교수
tk@mmlab.snu.ac.kr
논문접수 : 2005년 12월 1일
심사완료 : 2006년 5월 25일

1. 서론

TCP(Transmission Control Protocol)는 ACK (Acknowledgement) 메커니즘을 통해 신뢰성있는 스트림 서비스를 제공하기 위한 전송 계층 프로토콜로서 현재 인터넷에서 가장 널리 사용되고 있는 프로토콜 중 하나이다. 그러나 TCP는 초창기 유선망에 적합하도록 설계되었기 때문에 무선 환경에서는 최적의 성능을 달성할 수 없다는 것이 이미 알려져 있다. 이것은 TCP 데이터 흐름

(flow)이 모든 패킷 손실을 네트워크 혼잡(congestion)에 의한 것이라고 가정하고 보내는 데이터량을 조절하지만, 무선 환경에서는 무선 채널 에러에 의한 패킷 손실도 발생하기 때문이다. TCP 데이터 흐름은 이를 구별할 수가 없기 때문에 모든 패킷 손실을 네트워크 혼잡으로 간주하여 데이터 흐름 및 혼잡 제어 알고리즘에 의해서 TCP 데이터 흐름의 성능 저하가 발생한다. 이러한 문제를 해결하기 위해 분할 TCP(split-TCP), 지역적인 재전송(local retransmission), FEC(Forward Error Correction)와 같은 다양한 기법들이 연구되었다. 그러나 제안된 기법들은 일반적인 무선 네트워크가 아닌 IEEE 802.11 기반 무선랜 환경에 적용하였을 때는 만족할 만한 성능 향상을 보여 주지 못한다. 이것은 IEEE 802.11 링크 계층이 자체적으로 손실된 프레임에 대한 재전송을 수행하기 때문에 전송 계층의 입장에서 대부분의 패킷이 성공적으로 전송되기 때문이다. 따라서 최근 들어 무선랜 환경에서 TCP 데이터 흐름 성능 저하를 완화하고 향상시키기 위한 연구들이 진행되고 있다. 무선랜 환경에 초점을 맞추고 있는 최근 연구들이 제안하는 대부분의 해결책은 링크 계층 프로토콜이나 전송 계층 프로토콜을 수정하는 기법으로, 전송 무선 노드 간의 형평성이나 업로딩 TCP 데이터 흐름과 다운로드 TCP 데이터 흐름 사이의 형평성 측면에서는 성능 향상을 보이지만 TCP 데이터 흐름 자체의 처리량은 크게 향상시키지 못하는 한계가 있다.

IEEE 802.11 기반 무선랜에서 나타나는 TCP 데이터 흐름 성능 저하의 근본적인 원인은 전송 계층에서는 전송할 데이터가 있지만 링크 계층에서는 전송할 데이터가 없는 경우가 발생하기 때문이다. 이러한 현상은 TCP 데이터 흐름 및 혼잡 제어 알고리즘과 모든 무선 노드에게 동일한 무선 채널 접근 기회를 부여하는 IEEE 802.11 링크 계층의 특성이 결합하여 발생하게 된다. 이런 TCP 데이터 흐름 성능 저하 문제는 링크 계층 프로토콜과 전송 계층 프로토콜의 중복된 기능으로 인하여 더욱 심각해진다. 즉, 중복된 기능 수행을 위한 불필요한 전송으로 인하여 프레임 사이에 더 많은 충돌이 발생하게 되고, 실제 필요한 프레임 전송이 실패하는 상황이 발생하게 되어 더욱 TCP 데이터 흐름 성능 저하를 가속화시키게 된다. 이런 근본적인 문제가 해결되어야지만 IEEE 802.11 기반 무선랜 환경에서의 TCP 데이터 흐름 성능 저하를 막고, 궁극적으로 TCP 데이터 흐름의 성능을 향상시킬 수 있을 것이다. 또한 절전 모드(PSM: Power Saving Mode)인 경우, 무선 구간간의 패킷 전송 지연 시간 증가가 TCP 데이터 흐름의 심각한 성능 저하를 야기한다는 것이 널리 알려져 있다. 본 논문에서는 앞서 지적한 문제들을 해결할 뿐만 아니라

동시에 절전 모드 무선랜 환경에서도 TCP 데이터 흐름의 성능 향상을 가져올 수 있는 기법을 제안한다.

본 논문은 다음과 같이 구성되어 있다. 2장에서는 TCP 데이터 흐름 성능 향상과 관련된 기존 관련 연구들이 지적하고 있는 문제점과 해결책을 처리량과 형평성 측면으로 나누어 살펴보고, 기존 관련 연구의 한계점을 지적한다. 3장에서는 IEEE 802.11 기반 무선랜 환경에서의 TCP 성능 저하 원인을 규명하고, 이런 문제를 해결하기 위하여 TAS(TCP-Aware Sub-layer) 기법을 4장에서 기술하고, 제안된 기법을 실제 구현 및 적용할 때 발생하는 몇 가지 이슈 및 해결책도 간략히 언급한다. 5장에서는 모의 실험을 통하여 제안된 기법의 성능 평가를 수행하고, 마지막으로 6장에서 결론을 맺는다.

2. 관련 연구

본 장에서는 무선랜에서의 TCP 성능 향상을 위한 관련 연구들을 TCP 데이터 흐름 처리량 향상과 TCP 데이터 흐름 형평성 향상 측면으로 나누어 살펴본다. 기존 관련 연구들의 장단점을 파악하여 본 논문에서 해결하려고 하는 새로운 문제점을 정의한다.

2.1 IEEE 802.11 무선랜에서 TCP 데이터 흐름 처리량 향상

전통적으로 인터넷에서는 혼잡 및 에러 제어를 상위 계층(중단간)에서 처리하도록 하였으나 IEEE 802.11 무선랜과 같이 링크 계층에서 지역적인 복구를 시도하는 경우에는 전송 계층과 링크 계층에서 동시에 재전송을 하기에 무선 링크의 대역폭이 낭파되는 문제점을 낳기도 한다[1]. 관련 연구 [2]에서 제안하는 TCP 스누핑(snooping) 기법은 AP 단에 TCP 스누핑 에이전트 모듈을 삽입하여 네트워크 계층의 변형을 시도하였다. TCP 스누핑 모듈은 AP를 통하여 이동 노드로 전송했으나 아직 TCP ACK을 받지 못한 TCP DATA를 버퍼에 임시로 저장한다. 만약 중복된 TCP ACK이 도착하거나 전송 계층의 타임 아웃에 의한 TCP DATA 손실이 감지되면, TCP 스누핑 모듈은 버퍼에 저장해 둔 TCP DATA를 무선 링크에서 지역적(local)으로 재전송하게 된다. 또한 AP가 수신한 중복된 TCP ACK을 전달하지 않고 버림으로써 TCP 송신측이 불필요한 빠른 재전송(fast retransmission)을 수행하지 않도록 할 수 있다. 관련 연구 [3]은 관련 연구 [2]에서 제안한 TCP 스누핑 기법의 변형으로, 관련 연구 [2]와 달리 AP에서 타임아웃 매커니즘을 사용하지 않고, 수신 무선 노드가 직접 재전송을 요청하는 경우에만 지역적 재전송을 수행하게 된다. 이를 통해 무선랜에서 일시적으로 TCP 데이터 흐름이 단절되었을 때 트래픽이 축적되는 문제를 해결할 수 있다는 장점이 있다. 관련 연구 [4]에서는

TCP 데이터 흐름이 자주 단절되는 무선 환경을 위하여 M-TCP 기법이 제안되었다. 이 기법은 전체 네트워크를 단순한 이동 노드와 하나의 셀(cell)을 담당하는 이동 지원 노드(MSS: Mobile Support Station), 여러 개의 이동 지원 노드를 통제하는 감시 노드(SH: Supervisor Host)의 3 계층으로 구분하여 AP 간 핸드오프 발생시 이동 지원 노드 사이에서 상태 정보를 교환하는 것이 아니라 상위의 감시 노드 사이에서 교환이 이루어짐으로써 빈번한 핸드오프 발생시 상태 정보 전송에 드는 오버헤드를 줄일 수 있는 기법이다.

관련 연구 [5]에서는 무선 채널 에러가 빈번한 무선망에서 TCP 성능 향상을 위하여 분할 연결 방식(split-connection approach)에 기반하여 I-TCP(Indirect TCP)를 제안하였다. 즉, 하나의 TCP 데이터 흐름을 유선 노드부터 AP 사이 유선 구간과 AP로부터 이동 노드 사이 무선 구간의 두 TCP 데이터 흐름으로 분할하는 것이다. AP가 유선 구간의 TCP 데이터 흐름을 통하여 유선 노드가 보낸 TCP DATA를 수신하면 AP는 TCP ACK을 유선 노드에게 보내고, 동시에 무선 구간의 TCP 데이터 흐름을 통하여 TCP DATA를 무선 노드에게 전달한다. 이 기법의 장점은 무선 링크에서 발생한 전송 오류를 유선 구간의 TCP 송신측은 알 필요가 없다는 것이고, 무선 구간의 TCP 데이터 흐름은 유선 구간의 TCP 데이터 흐름과는 독립적으로 무선 링크에 최적화될 수 있다는 것이다. 반면, 중단간 연결이라는 TCP 데이터 흐름의 기본 개념을 위반한다는 단점이 존재한다. 예를 들면, 실제 TCP 수신측인 무선 노드가 TCP DATA를 수신하기도 전에 TCP 송신측인 유선 노드에게 해당 TCP DATA에 대응하는 TCP ACK이 전달되는 경우가 발생할 수 있다.

관련 연구 [6]은 무선랜에서 TCP 성능을 향상시키기 위한 최근 연구 중 하나로 DCF+ 기법을 제안하였다. IEEE 802.11 DCF 표준은 MAC DATA를 전송한 무선 노드는 수신 노드로부터 MAC ACK을 받아야만 한다고 규정하고 있다. DCF+는 수신 노드의 MAC ACK이 RTS 역할을 하고 RTS를 수신한 송신 노드가 CTS로 응답하도록 하여, 수신 노드가 즉시 송신 노드에게 데이터를 전송할 수 있도록 한다. 즉, 수신 노드에서 송신 노드로의 역방향 전송을 하기 위한 무선 채널 경쟁없이 즉각적으로 전송이 가능하게 하기 때문에 성공적인 전송을 위해 필요한 시간이 줄어드는 이득을 얻게 된다. 이 기법은 무선 노드들이 항상 서로에게 보낼 데이터를 가지고 있고, TCP 데이터 흐름의 경우에 TCP DATA와 TCP ACK이 이에 해당된다고 가정하고 있다.

2.2 IEEE 802.11 무선랜에서 TCP 데이터 흐름 형평성 향상

관련 연구 [7]에서는 무선랜 환경에서 존재하는 다양한 TCP 데이터 흐름 시나리오에 대한 모의 실험 결과를 바탕으로 TCP 데이터 흐름 사이의 형평성 문제를 지적하고 해결책을 제시하고 있다. 대표적으로 동일한 수의 업로딩 TCP 데이터 흐름과 다운로드 TCP 데이터 흐름이 존재할 경우, 업로딩 TCP 데이터 흐름과 다운로드 TCP 데이터 흐름 사이의 형평성 문제를 지적하고, 업로딩 TCP 데이터 흐름에서 TCP ACK을 전송할 때 TCP ACK 내의 수신 가능 윈도우(receiver advertised window) 값을 조절하여 업로딩 TCP 데이터 흐름과 다운로드 TCP 데이터 흐름 사이의 형평성 향상 결과를 보였다. 또한, 무선 채널 에러나 TCP 데이터 흐름별 RTT 값에 의한 요인도 간과할 수 없다는 점을 지적하면서 제안한 분석을 확장할 필요가 있음을 시사하였다. 관련 연구 [8]에서는 다운로드 TCP 데이터 흐름과 업로딩 TCP 데이터 흐름 사이의 형평성 문제를 AP가 무선 채널 경쟁에서 좀더 우선 순위를 가지도록 함으로써 해결하는 방법이 제안되었다. 즉, IEEE 802.11 DCF 표준과 달리 AP는 DIFS가 아닌 PIFS 만큼만 기다린 후, 백오프 매커니즘 없이 무선 채널을 획득함으로써 다운로드 TCP 데이터 흐름이 업로딩 TCP 데이터 흐름만큼 무선 채널을 획득할 수 있도록 한다.

관련 연구 [9]와 [10]에서는 무선랜 환경에서 업로딩 TCP 데이터 흐름 사이의 형평성 문제를 지적하고 있다. 즉, 여러 개의 업로딩 TCP 데이터 흐름들이 존재하는 경우, 업링크 상의 TCP DATA와 다운링크 상의 TCP ACK 간의 충돌이 발생할 수 있고, 이 때 TCP ACK이 손실되면 이 TCP ACK을 받아서 다음 TCP DATA를 전송해야 하는 무선 노드는 큰 손해를 보게 된다는 것이다. 그러나 실제로는 IEEE 802.11 기반 무선랜에서 업로딩 TCP 데이터 흐름들 사이의 형평성 문제는 실제 환경에서는 발생하지 않는다. 이것은 업로딩 TCP 데이터 흐름을 생성하는 무선 노드가 수가 늘어나더라도 링크 계층에서 실제 경쟁에 참여하는 무선 노드의 수는 크게 증가하지 않기 때문에 TCP DATA와 TCP ACK 사이의 충돌 횟수 역시 크게 늘어나지 않는다.

3. 문제 정의

본 장에서는 IEEE 802.11 기반 무선랜에서 TCP 성능 하락의 주요 원인을 살펴본다. 본 논문에서 제안하는 알고리즘은 이런 문제점들을 해결하거나 성능 하락에 미치는 영향을 최소화하는 것을 목표로 한다.

3.1 전송 계층과 링크 계층의 중복된 기능

전송 계층 프로토콜인 TCP와 링크 계층 프로토콜인 IEEE 802.11은 모두 신뢰성 서비스 제공을 위하여 해당 계층에서 ACK 및 재전송 매커니즘을 채용하고 있다.

TCP 송신 호스트는 TCP DATA를 전송하고 일정 시간 내에 TCP 수신 호스트로부터 TCP ACK을 수신하게 되면, 다음 TCP DATA 패킷을 전송할 수 있다. 만약 일정 시간 내에 TCP ACK을 수신하지 못하면 해당 TCP DATA를 재전송하게 된다. IEEE 802.11도 마찬가지로 MAC DATA를 전송하고 일정 시간 내에 MAC ACK을 수신하지 못하면, 전송이 실패한 것으로 간주하고 MAC DATA 길이에 따라 정해진 회수만큼 해당 MAC DATA를 재전송하게 된다. 따라서 TCP DATA를 포함하는 MAC DATA 전송을 전송하고 MAC ACK을 성공적으로 수신하였다면, 이것은 성공적인 TCP DATA 전송을 암시적으로 의미하기 때문에, TCP 수신 호스트가 TCP ACK을 전송하는 것은 일종의 희소한 무선 네트워크 자원의 낭비라고 볼 수 있다.

3.2 IEEE 802.11 기반 무선랜에서 TCP 데이터 흐름 및 혼잡 제어

IEEE 802.11 기반 무선랜¹⁾에서는 최적의 성능을 달성하기 위해서 경쟁에 참여하는 무선 노드들의 수가 적절하게 유지되어야 한다. 하나의 무선 노드가 계속 무선 채널을 획득하는 것을 막기 위해 IEEE 802.11 표준[11]에 정의된 포스트 백오프(post backoff)에 의해서 하나의 무선 노드로는 무선 채널이 낭비되고, 많은 수의 무선 노드가 경쟁에 참여하는 경우에는 무선 채널 사용을 극대화할 수는 있지만 충돌로 인한 무선 자원의 낭비가 있기 때문이다. 모든 TCP 송신 호스트는 TCP ACK을 수신할 필요 없이 혼잡 윈도우 크기만큼의 TCP DATA를 전송할 수 있지만 그 이후에는 TCP ACK을 수신해야지만 다음 TCP DATA를 전송할 수 있기 때문에, 전송 계층에서는 보낼 데이터가 있지만 링크 계층에서는 아직 전송할 데이터가 없는 무선 단말이 존재할 수도 있다.

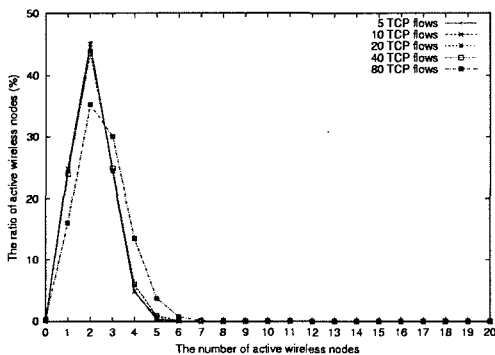


그림 1 IEEE 802.11 관점에서의 경쟁 무선 노드의 비율 [12]

일정 시간이 지나면 대부분의 무선 노드들이 자신의 혼잡 윈도우만큼의 TCP DATA를 전송하고 TCP ACK을 기다리는 상황이 발생하게 된다. 이 때 AP (Access Point)가 TCP ACK을 전송하게 되면 해당 TCP ACK을 받은 무선 노드는 다음 TCP DATA를 전송하기 위해서 채널 경쟁에 참여하게 된다. 물론 AP도 다른 무선 노드를 향하는 TCP ACK을 전송하기 위해서 채널 경쟁에 참여하지만, IEEE 802.11은 모든 무선 노드에게 동일한 무선 채널 접근 기회를 보장하기 때문에 무선 노드가 무선 채널을 획득하여 TCP DATA를 보낼 확률이 높다. 해당 무선 노드가 혼잡 회피 단계였다면 TCP DATA를 전송하고 다시 TCP ACK을 기다리는 상황이 되기 때문에 다음 채널 경쟁에는 참여하지 않게 되어 AP가 무선 채널을 획득할 확률이 높아지게 된다. 그림 1은 링크 계층 관점에서 다양한 개수의 업로딩 TCP 데이터 흐름에 따른 경쟁에 참여하는 무선 노드의 비율을 나타내며, TCP 데이터 흐름 수와는 무관하게 평균적으로 2개의 무선 노드(AP와 TCP ACK을 받은 무선 노드)가 경쟁에 참여하는 경우가 절반에 가까운 비율을 차지하고 있는 것을 볼 수 있다.

3.3 TCP 데이터 흐름 내 경쟁

단방향성인 UDP와는 다르게 TCP는 양방향성으로 하나의 TCP 데이터 흐름은 정방향의 TCP DATA 데이터 흐름과 역방향의 TCP ACK 데이터 흐름 두 개로 구성된다. 대역폭이 충분하고 에러가 거의 발생하지 않는 유선 네트워크에서는 큰 무리 없이 동작하지만 업링크와 다운링크의 구별없이 무선 채널을 공유하는 IEEE 802.11 기반 무선랜의 경우, 하나의 TCP 데이터 흐름 내에서도 TCP DATA와 TCP ACK은 서로 경쟁을 하게 되고, TCP DATA와 TCP ACK 충돌로 인해서 전체 시스템의 성능이 저하될 수 있다. 이 문제는 자기 경쟁(self-contention) 혹은 자기 간섭(self-interference)으로 널리 알려져 있다.

3.4 절전 모드 무선랜에서의 긴 지연 시간

IEEE 802.11은 배터리로 동작하는 무선 노드의 에너지를 절약하여 동작 가능 시간을 늘리기 위해서 절전 모드(PSM: Power Saving Mode)를 정의하고 있다. 그러나 IEEE 802.11 절전 모드는 무선랜 상에서 TCP 성능을 더욱 저하시킨다[13]. 예를 들어 업로딩 TCP 데이터 흐름의 경우, 무선 노드가 TCP DATA를 전송하고 TCP ACK이 AP에 도착하기 전에 비콘 주기(beacon interval)가 끝나면 무선 노드는 에너지 절전 모드로 동작하게 되고, 이 때 TCP ACK이 AP에 도착하면 단말 노드에게 데이터를 전송할 수 없기 때문에 AP는 해당 TCP ACK을 임시로 저장하고 다음 비콘 주기까지 기다려야만 한다. 무선 단말이 깨어났을 때 TCP ACK을

1) 대부분의 IEEE 802.11 기반 무선랜은 DCF (Distributed Coordination Function)로 동작하기 때문에 본 논문에서는 PCF (Point Coordination Function)은 고려하지 않는다.

전송하게 하기 때문에 TCP 데이터 흐름, 특히 짧은 수명(short-lived)의 TCP 데이터 흐름에 대하여 RTT를 증가시켜 무선 채널의 대역폭을 충분히 사용하지 못하는 결과를 낳는다.

4. 제안된 알고리즘

본 장에서는 이전 장에서 지적되었던 IEEE 802.11 기반 무선랜에서의 TCP 성능 저하 문제를 해결하기 위한 알고리즘을 제안한다. 기본적으로 분할 연결(split connection) 개념을 기반으로, AP나 라우터 등의 분할 지점(split point)과 무선 노드에 서브 계층을 추가하여 동작한다.

4.1 기본 아이디어

앞에서도 언급하였듯이 IEEE 802.11 기반 무선랜에서 TCP 데이터 흐름이 동작할 경우, 신뢰성 서비스를 위하여 전송 계층 프로토콜과 링크 계층 프로토콜이 독립적으로 자신만의 ACK 메커니즘을 사용하기 때문에 기능이 중복되고 TCP 데이터 흐름 및 혼잡 제어가 TCP ACK 기반으로 동작한다는 문제점이 있었다. 본 논문에서는 무선 구간에서의 중복된 TCP ACK 메커니즘을 우회할 수 있는 기법을 제안한다. 즉, 무선 구간의 신뢰성 서비스 제공을 IEEE 802.11에게 일임하고 TCP ACK 시그널링을 제거하여 불필요한 메시지 전송을 줄여 TCP 성능 향상을 꾀한다. 무선 구간에서 송신 노드가 TCP DATA를 포함하는 MAC DATA 전송 후 MAC ACK을 수신하면, TCP ACK을 수신하지 않고도 TCP DATA가 성공적으로 전송한 것으로 간주하여 TCP ACK을 생성하여 TCP 송신 노드의 상위 계층에게 보낸다. 이 기법을 통하여 무선 노드가 TCP DATA를 포함하는 MAC DATA 전송 후 MAC ACK을 수신

하면 TCP ACK을 수신한 것과 동일한 효과가 나기 때문에 전송 계층에서 보낼 데이터가 있지만 링크 계층에서는 보낼 데이터가 없는 경우를 방지하여 좀 더 적극적으로 무선 노드들이 무선 채널 경쟁에 참여하게 되어 무선 채널 사용을 극대화할 수 있다. 따라서 기존 연구 [9]에서 지적되었던 새로운 TCP 데이터 흐름의 진입 장벽 문제가 어느 정도 해결 가능하다. 이런 개념을 좀 더 일반화시켜 적용하기 위하여 본 논문에서는 분할 연결 개념을 채용하여 분할 지점인 AP를 기준으로 무선 구간에서의 TCP를 위한 무선 단말의 동작을 정의한다. 그림 2는 전송 계층의 관점에서 기존에 제안되었던 I-TCP와 본 논문에서 제안하는 기법의 차이점을 간략히 보여준다.

본 논문에서 제안하는 기법은 절전 모드 무선랜에서 더욱 효율적으로 동작할 수 있을 것으로 예상된다. 절전 모드 무선랜에서 업로딩 TCP 데이터 흐름이 활성화된 경우, TCP DATA를 전송하고 TCP ACK이 AP에 도착하기 전에 비콘 주기가 끝난다면, AP는 버퍼링하여 다음 비콘 주기에 TCP ACK을 전송하게 되어 TCP ACK 전송 시간을 증가시킨다. 이런 동작은 해당 TCP 데이터 흐름의 RTT를 증가시켜 전체 성능이 하락되는 문제를 발생시키지만, 본 논문에서 제안하는 기법은 무선 노드가 무선 채널 상에서 실제로 TCP ACK과 무관하게 TCP DATA를 전송하기 때문에 TCP ACK으로 인한 TCP 성능 저하 현상을 막을 수 있다. 다음 절에서 본 논문에서 제안하는 기법을 적용한 프로토콜의 자세한 동작 과정을 설명한다.

4.2 프로토콜 설계

UDP는 데이터 흐름이나 혼잡 제어 메커니즘이 없기 때문에 분할 연결 개념을 적용하더라도 큰 이득을 얻을

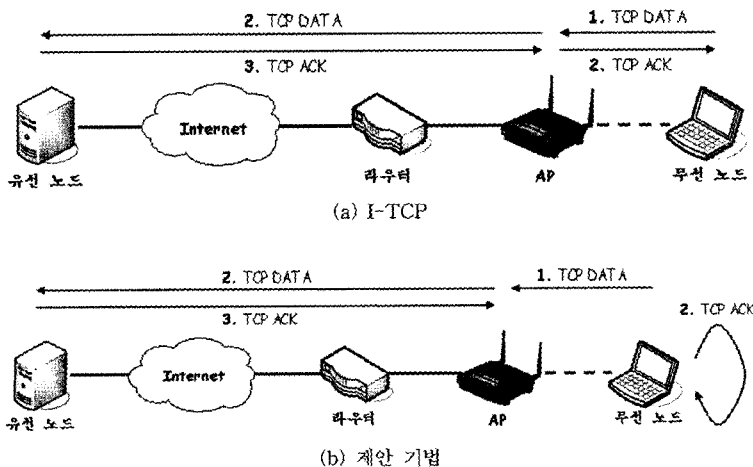


그림 2 I-TCP와 제안 기법의 동작 방식

수 없으나, TCP는 분할 연결 개념을 도입하여 유선 구간과 상이한 특성을 지니는 무선 구간에서 동작의 최적화를 통하여 성능 개선의 이득을 취할 수 있다. 본 논문에서 제안하는 기법은 전송 계층과 링크 계층 사이에 새로운 TCP 인지 서브 계층(TAS: TCP-aware sub-layer)을 삽입하는 계층간 최적화(cross layering) 방식을 기반으로 하기 때문에 TCP 자체의 수정은 필요하지 않다. 분할 지점인 AP와 무선 노드의 수정된 프로토콜 스택은 그림 3과 같다.

제안된 알고리즘으로 동작하는 무선 노드 역시 기존의 TCP와 마찬가지로 유선 노드와 TCP 데이터 흐름을 생성하거나 종료할 때 TCP SYN 혹은 TCP FIN 메시지를 주고 받는다. 그러나 이 메시지는 제안된 알고리즘이 분할 연결 개념을 채택하고 있기 때문에 실제로는 분할 지점인 AP와 주고 받게 된다. 이 때 AP와 무선 노드의 TAS는 이 메시지들을 스누핑(snooping)하여 해당 TCP 데이터 흐름 정보를 저장하거나 필요없는 TCP 데이터 흐름 정보를 삭제한다.

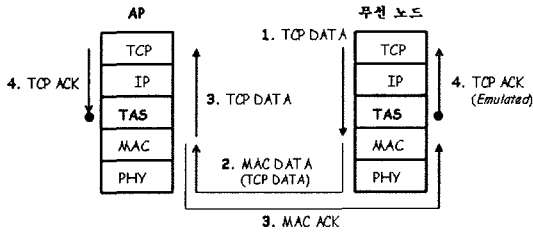


그림 3 수정된 프로토콜 스택

특히 초기의 TCP SYN 혹은 TCP ACK 메시지는 초기 시퀀스 번호 등 TCP 데이터 흐름 유지를 위한 정보를 포함하고 있고, 이는 TAS가 TCP ACK을 에뮬레이션 하기 위해서는 필요한 정보이다. TCP 데이터 흐름의 생성 과정 후 TCP DATA를 보낼 때 무선 노드의 TAS는 해당 TCP DATA 전송 성공시에 수신해야 하는 TCP ACK 시퀀스 번호를 갱신한다(그림 3-1. TCP DATA). 무선 노드의 링크 계층에서는 TCP DATA를 포함하는 MAC DATA를 전송하고(그림 3-2. MAC DATA), AP의 링크 계층은 성공적인 MAC DATA 수신을 알리는 MAC ACK을 전송한다(그림 3-3. MAC ACK). 그리고 동시에 수신한 TCP DATA를 전송 계층에게 전달한다(그림 3-3. TCP DATA). MAC ACK을 수신한 무선 노드의 링크 계층은 성공적인 MAC DATA 전송에 대하여 TAS에게 알리게 되고, TAS는 해당 MAC DATA에 포함된 TCP DATA가 성공적으로 전송된 것으로 간주하고 대응하는 저장된 TCP 데이터 흐름 정보를 바탕으로 TCP ACK을 생

성하여 전송 계층에게 넘겨주게 된다(그림 3-4. emulated TCP ACK). 또한 AP의 전송 계층이 TCP DATA를 수신하고 TCP ACK을 보내게 되는데, 무선 노드의 TAS가 동일한 TCP ACK을 생성하여 전송 계층으로 전송했기 때문에 AP의 TAS는 해당 TCP ACK을 버린다(그림 3-4. TCP ACK).

이전에도 언급했듯이 유선 환경과는 달리 무선 환경에서는 패킷 전송 중에 다양한 원인으로 에러가 발생하기 쉽다. 비록 링크 계층에서 4번 혹은 7번 재전송을 하지만 결국 전송에 실패하는 경우가 발생할 수도 있다. 이 때는 링크 계층에서 TCP DATA를 포함하는 MAC DATA를 버리게 되고, 전송 실패를 상위 계층에게 알리게 된다. TAS는 이 신호를 가로채어 아무런 동작을 취하지 않는다. 따라서 전송 계층의 타이머가 경과하면 데이터 손실을 인지하고 해당 TCP DATA를 재전송하게 되어 위의 과정을 반복하게 된다. 이런 동작 방식을 통하여 기존의 TCP 동작을 직접 수정하지 않고도 제안하는 기법을 구현하는 것이 가능하다.

그러나 제안된 기법으로 동작하는 경우, 모든 무선 노드의 링크 계층이 항상 보낼 MAC DATA를 가지고 있기 때문에 이전 관련 연구에서 지적되었던 TCP 데이터 흐름 사이의 형평성 문제가 대두된다. AP를 제외한 모든 무선 단말은 각각 자신의 업로딩 TCP 데이터 흐름을 위한 MAC DATA를 전송하면 되지만, AP는 모든 다운로드 TCP 데이터 흐름을 위한 MAC DATA를 전송해야만 한다. 그러나 IEEE 802.11은 AP를 포함하는 모든 무선 노드에게 동일한 무선 채널 접근을 보장하기 때문에 실제 업로딩 TCP 데이터 흐름이 다운로드 TCP 데이터 흐름에 비하여 무선 채널을 획득할 확률이 크다. 따라서 제안된 기법으로 동작하는 무선랜의 경우, 업로딩 TCP 데이터 흐름과 다운로드 TCP 데이터 흐름 사이에 형평성 문제가 발생한다. 또한 AP 큐 길이가 무한하지 않기 때문에 특정 다운로드 TCP 데이터 흐름으로부터 도착한 다수의 TCP DATA가 큐에서 버려질 수도 있고, 이 때 다운로드 TCP 데이터 흐름 사이에서도 형평성 문제가 발생할 수도 있다. 제안된 기법을 적용할 때 발생하는 업로딩 및 다운로드 TCP 데이터 흐름 사이의 형평성 문제를 해결하기 위해서 본 저자들이 이전 관련 연구 [12]에서 제안하였던 제로 슬롯(zero-slot) 기법을 추가적으로 적용할 수 있다. 예를 들어, M개의 업로딩 TCP 데이터 흐름과 N개의 다운로드 TCP 데이터 흐름이 있을 때, N개의 다운로드 TCP 데이터 흐름의 TCP DATA 전송을 책임지는 AP는 일정 시간 (T) 간격 동안(T_i) 측정된 무선 채널 사용의 비율(α)

$\frac{N}{M+N}$ 이보다 작다면 다음 일정 시간 간격(T_{i+1}) 전에

특정 시간(AP_Ti) 동안 해당 무선 채널 사용 비율을 맞추기 위하여 AP가 우선적으로 무선 채널 획득한 후, 전송 기회를 획득하지 못한 다운로드 TCP 데이터 흐름

개수인 $\lfloor \left(\frac{N}{M+N} - \alpha \right) \times (M+N) \rfloor$ 만큼의 TCP DATA

를 연속적으로 전송한다. 그 외에도 이런 형평성 문제가 발생하는 상황을 가정하고, 이를 해결하기 위한 기존 관련 연구들[7-9]이 존재한다.

4.3 추가 고려 사항

본 논문에서 제안된 기법을 실제 환경에 적용하기 위해서는 몇 가지 추가로 고려해야 하는 사항들을 살펴본다. 추가 고려 사항은 모의 실험에서는 구현되지 않았다.

4.3.1 구현 관련

실제 구현과 관련하여 분할 지점이나 무선 노드에서 MAC DATA를 성공적으로 수신하여 MAC ACK을 전송하더라도 TCP DATA를 전송 계층에 전달하는 과정에서 소프트웨어적인 문제로 TCP DATA가 손실되거나 에러가 발생할 가능성도 존재한다. 따라서 제안된 기법을 실제 구현할 때는 TAS에서 TCP DATA를 버퍼링하는 기능이 필요하다. 만약 전송 계층에서 TCP DATA를 정상적으로 수신한다면 TAS는 해당 TCP DATA에 대응하는 TCP ACK을 수신할 수 있을 것이고, 이 때 버퍼링하고 있던 TCP DATA와 수신한 TCP ACK을 버릴 수 있다. 그러나 일정 시간 동안 대응하는 TCP ACK을 수신하지 못하면 해당 TCP DATA를 다시 전송 계층에게 전달해 주는 기능을 수행하면 자체 소프트웨어 에러에 의한 TCP DATA 손실을 방지할 수 있다.

4.3.2 배포 관련

실제 배포(deployment) 시나리오와 관련하여 분할 지점의 선택이 중요한 문제 중에 하나이다. 본 논문에서 제안된 기법은 분할 지점인 AP와 무선 노드의 프로토콜 스택에 서버 계층을 추가해야 한다. 무선 노드의 경우는 소프트웨어 갱신을 통하여 서버 계층을 추가할 수 있지만, 값싼 비용으로 배포되고 있는 AP의 경우는 종종 추후 소프트웨어 갱신이 불가능하다. 따라서 본 기법을 적용하기 위해서는 갱신된 소프트웨어를 포함하는 AP를 재배포해야 비용이 추가로 든다. 만약 AP와 연결된 첫 번째 라우터를 분할 지점으로 선택한다면 이런 문제가 어느 정도 해결 가능하다. 즉, 대부분의 라우터가 소프트웨어 갱신 기능을 제공하고 있기 때문에 제안된 기법을 추가하는 것이 용이하고, 일반적으로 AP와 첫 번째 라우터 사이의 링크는 충분한 대역폭을 제공하고 무시할 정도의 링크 지연 시간만을 야기시킬 것이므로 제안된 기법의 동작에 거의 영향을 미치지 않을 것으로 예상된다. 다른 방법으로는 제안된 기법의 모든 기

능을 무선 노드의 프로토콜 스택 상에서 구현하는 방법이 가능하다. 즉, 다운로드 TCP 데이터 흐름의 경우 TAS 계층에서 전송 계층으로부터 내려오는 TCP ACK을 필터링하고, 업로딩 TCP 데이터 흐름의 경우 링크 계층이 TCP ACK에 해당하는 MAC DATA를 성공적으로 수신한 후 전송 계층으로 건내줄 때 TAS 계층에서 가로채어 필터링한다. 그러나 이 방법은 업로딩 TCP 데이터 흐름의 경우, 3.1절에서 지적하였던 무선 채널 낭비 문제가 그대로 남아 있다.

4.3.3 혼잡 제어 및 형평성 문제

제안된 기법으로 동작하는 무선랜에서 무선 노드가 멀리 떨어져 있는 유선 구간의 서버와 통신하기 위하여 TCP 데이터 흐름을 생성하는 경우, 무선 노드와 분할 지점인 AP 사이의 TCP 데이터 흐름은 충분한 대역폭과 1~2 홉 거리로 매우 작은 RTT 값을 가지게 될 것이고, 높은 TCP 데이터 흐름 처리량을 달성할 수 있다. 이에 반해 AP와 유선 노드 사이의 TCP 데이터 흐름은 인터넷을 통하여 연결되는 다수의 홉으로 구성되는 경로를 가지기 때문에 경로의 길이에 따라 다양한 범위의 RTT 값을 가지게 된다. 만약 유선 구간의 경로가 길어서 RTT 값이 크다면, TCP 데이터 흐름 제어 매커니즘에 의해서 TCP 데이터 흐름은 많은 TCP DATA를 전송할 수 없다. 따라서 무선 구간의 TCP 데이터 흐름의 처리량이 유선 구간의 TCP 데이터 흐름의 처리량보다 큰 경우가 발생할 수 있다. 즉, 두 개의 분할 TCP 데이터 흐름 사이에서 전송 속도 차이가 차이나는 경우에 AP나 무선 노드의 큐나 버퍼 길이에 제한이 있기 때문에 AP에서 두 개의 분할 TCP 데이터 흐름 사이에서 TCP DATA를 건내줄 때 지속적으로 TCP DATA가 버려지는 상황이 발생할 수 있다. 이런 과부하를 피하기 위해서 AP나 무선 노드는 자신의 큐나 버퍼 길이가 임계값을 초과하는 경우에는 고의적으로 TCP DATA를 포함하는 MAC DATA에 대응하는 MAC ACK를 전송하지 않는다. 링크 계층은 몇 번의 재전송 후에 TCP DATA를 버리게 되고 전송 계층의 타이머가 경과하게 되면, 패킷 손실을 인지하고 TCP 데이터 흐름 및 혼잡 제어 매커니즘이 호출되어 두 개의 분할 TCP 데이터 흐름 사이의 전송 속도를 균형을 맞출 수 있다.

4.3.4 IPsec 사용 관련

제안된 기법에서는 TAS가 TCP ACK을 에뮬레이션하기 위해서 초기의 TCP SYN 혹은 TCP ACK 메시지에 포함된 초기 시퀀스 번호 등 TCP 데이터 흐름 유지를 위한 정보를 획득해야 하며, TAS를 통과하는 메

2) IEEE 802.11b, IEEE 802.11a/g. 표준화가 진행 중인 IEEE 802.11n 무선랜의 경우, 최대 11(Mbps), 최대 54(Mbps), 최소 200(Mbps) 전송 속도를 지원한다.

시지들을 스누핑함으로써 필요한 TCP 데이터 흐름의 정보를 유지한다. 만약 네트워크 계층에서 IPsec을 사용하는 경우 적재물(payload)이 암호화되므로 하위 계층인 TAS에서 메시지를 스누핑하더라도 수신해야 하는 TCP ACK 시퀀스 번호 등 필요한 정보를 추출할 수가 없다. 따라서 본 논문에서는 IPsec을 사용하지 않는 상황을 가정하도록 한다.

5. 성능 평가

본 장에서는 본 논문에서 제안하는 알고리즘의 성능 평가를 위하여 수행한 모의 실험 결과를 설명한다. 널리 사용되고 있는 NS2[15]를 사용하여 TAS를 구현하였으며 다양한 큐 길이, 패킷 에러 비율과 PSM 사용 여부에 따른 TAS 성능을 기존의 TCP, I-TCP 성능과 비교하였다. 성능 측정 기준은 시간당 네트워크 처리량(aggregate throughput), 형평성 지표(fairness index), 평균 지연 시간으로 다양한 무선 노드 수에 따른 변화를 측정하였다.

모의 실험에 사용된 네트워크 구성은 그림 4와 같다. N개의 무선 단말, AP, 라우터와 N개의 고정 단말이 차례대로 연결되어 있으며, N은 5에서부터 80까지 값을 가진다. 각 무선 단말은 유선 구간에서 대응하는 고정 단말을 향하여 업로딩 TCP 데이터 흐름을 생성하며, FTP 세션을 통하여 100(s) 동안 1000(byte) TCP DATA를 지속적으로 보낸다. 모든 라우터와 고정 호스트 사이의 링크 대역폭은 100(Mbps), 링크 지연 시간은 10(ms)로 설정되어 있고, 라우터와 AP 사이의 링크 대역폭은 100(Mbps), 링크 지연 시간은 5(ms)로 설정되어 있다. 무선 구간의 대역폭은 11(Mbps)이며, 그 외의 특별한 언급이 없으면 802.11b 기본 설정을 따른다.

5.1 분할 지점인 AP 큐 길이에 따른 시간당 네트워크 처리량

그림 5는 제안된 기법(TAS)이 기존 TCP 방식이나 I-TCP 방식보다 네트워크 처리량 측면에서 더 좋은 성

능을 보인다는 것을 증명한다. 본 모의 실험은 분할 지점인 AP 큐 길이에 따라 다양한 수의 무선 노드가 존재할 때 업로딩 TCP 데이터 흐름 총 처리량을 측정하였다. TCP 데이터 흐름의 수가 작거나 적절한 경우에는 AP 큐 길이에 무관하게 제안된 기법이 더 좋은 성능을 보인다. 이것은 AP나 무선 노드 내에서 TCP ACK 에물레이션을 통해서 TCP 데이터 흐름이 무선 링크 상에서 TCP ACK을 수신하지 않고도 지속적으로 TCP DATA를 전송할 수 있기 때문이다. 기존 TCP 방식이나 I-TCP 방식의 TCP 데이터 흐름 총 처리량은 무선 노드의 개수가 증가하면서 약간 증가하기는 하지만 큰 변화는 없다. 이것은 이전에 언급했듯이 IEEE 802.11 기반 무선랜에서는 TCP 데이터 흐름 및 혼잡 제어 메커니즘의 영향으로 링크 계층에서는 무선 노드의 개수가 늘어나더라도 보낼 데이터가 없는 무선 단말이 대부분이기 때문이다. 제안된 기법으로 동작하는 TCP 데이터 흐름의 경우, 무선 노드의 개수가 증가함에 따라 지속적으로 네트워크 처리량이 감소한다. 이것은 TCP 데이터 흐름 개수만큼의 무선 노드가 실제 링크 계층의 관점에서 무선 채널을 획득하기 경쟁을 하여 그만큼 패킷 충돌이 많아지게 되고, 무선 채널 사용을 떨어뜨리는 효과를 가져오기 때문이다. 만약 링크 계층에서 경쟁하는 무선 노드의 수에 따른 최적의 경쟁 윈도우(contention window) 크기에 도달하는 것을 목표로 하는 S-EDCF 기법[16]과 같은 추가적인 메커니즘이 함께 사용된다면, 제안된 기법에서 많은 경쟁 노드 수에 따른 빈번한 패킷 충돌의 효과를 줄여 TCP 데이터 흐름 총 처리량을 떨어뜨리지 않고 유지할 수 있다.

5.2 분할 지점인 AP 큐 길이에 따른 업로딩 TCP 데이터 흐름 사이의 형평성 지표

그림 6은 제안된 기법이 기존 TCP 방식이나 I-TCP 방식보다 형평성 측면에서 더 좋은 성능을 보인다는 것을 증명한다. 본 모의 실험 역시 분할 지점인 AP 큐 길이에 따라 다양한 수의 무선 노드가 존재할 때 업로딩

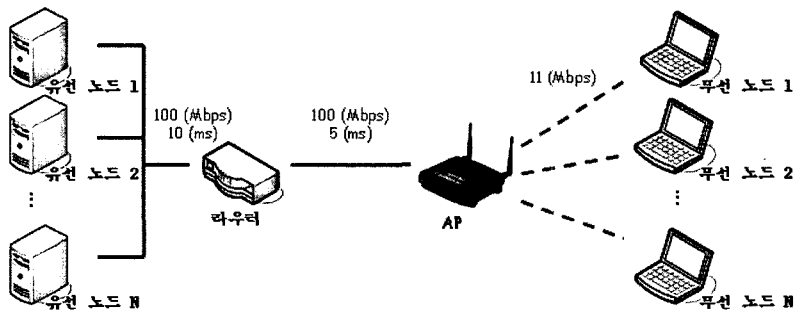
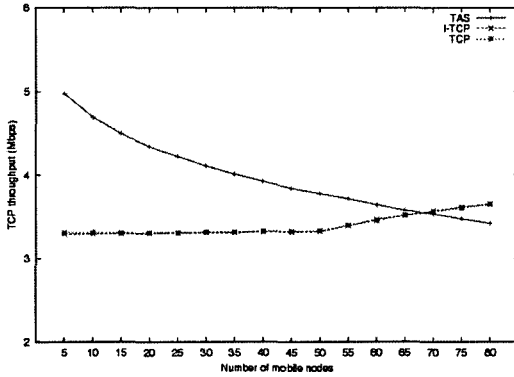
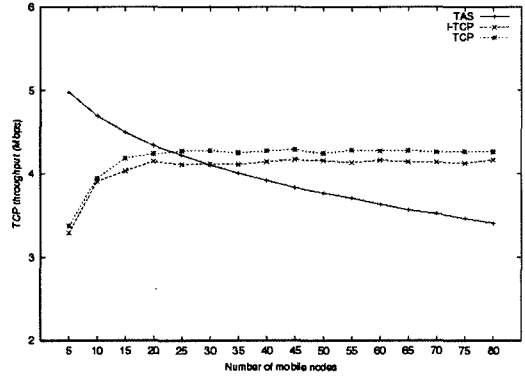


그림 4 네트워크 구성도

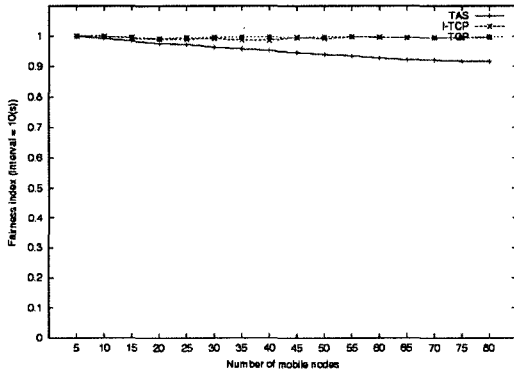


(a) AP 큐 길이가 1000인 경우

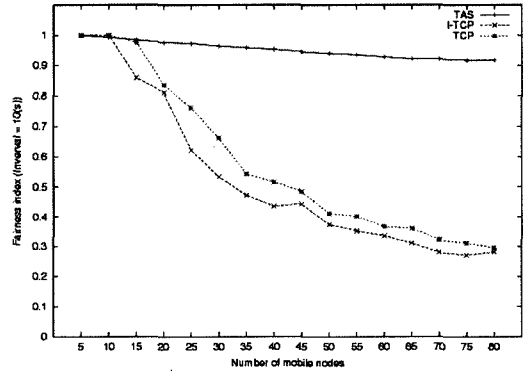


(b) AP 큐 길이가 100인 경우

그림 5 무선 노드 수에 따른 업로딩 TCP 데이터 흐름의 총 처리량



(a) AP 큐 길이가 1000인 경우



(b) AP 큐 길이가 100인 경우

그림 6 무선 노드 수에 따른 업로딩 TCP 데이터 흐름 형평성 지표

TCP 데이터 흐름 사이의 형평성 지표를 계산하였다. 관련 연구[17]에서 제안한 형평성 지표는 아래와 같이 정의된다.

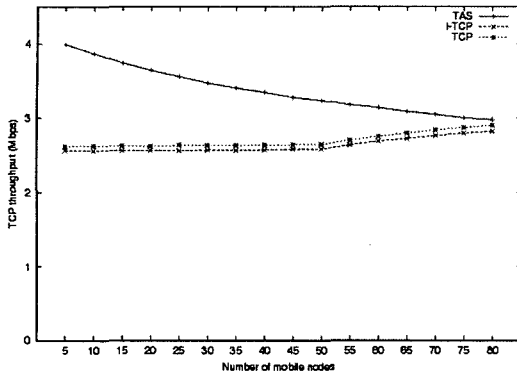
$$\text{형평성 지표} = \frac{\sum_f T_f / \phi_f}{\# \text{ of flows} \times \sum_f (T_f / \phi_f)^2}$$

여기서 T_f 는 데이터 흐름 f 의 처리량, ϕ_f 는 데이터 흐름 f 의 가중치, 형평성 지표는 얼마나 공평하게 자원이 할당되고 있는지를 나타낸다. 그림 6에서 볼 수 있듯이 AP 큐 길이가 충분히 큰 경우(즉, 1000일 때)에는 무선 노드의 수에 무관하게 제안된 기법, 기존 TCP 방식과 I-TCP 방식 모두 거의 공평하게 자원을 할당하고 있다. 제안된 기법이 무선 노드의 수가 증가함에 따라 형평성 지표가 약간 감소하지만 큰 영향을 미치지 않는다. 그러나 AP 큐 길이가 작은 경우(즉, 100일 때)에는 무선 노드의 수가 증가함에 따라 기존 TCP 방식이나 I-TCP 방식으로 동작하는 TCP 데이터 흐름은 형평성 지표가 크게 감소하는 것에 반해 제안된 기법으로 동작하는 TCP 데이터 흐름의 형평성 지표는 거의 유지되고

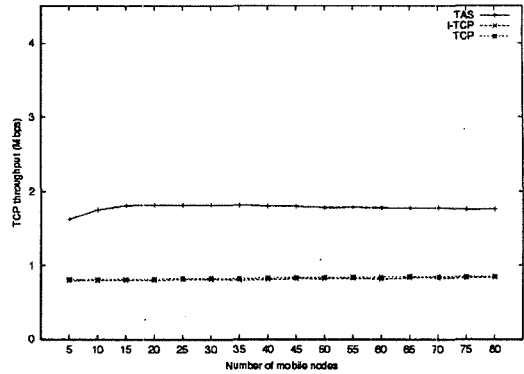
있다. 이것은 기존 TCP 방식이나 I-TCP 방식으로 동작하는 TCP 데이터 흐름의 경우, AP 큐에서 TCP ACK이 버려지는 상황이 발생하게 되어 TCP 데이터 흐름 및 혼잡 제어 메커니즘에 의해서 TCP 데이터 흐름 사이에 형평성 문제를 야기시키지만, 제안된 기법은 실제 무선 링크 상에서 TCP ACK 전송을 제거하였기 때문에 AP 큐에서 버려지는 패킷에 의해서 형평성의 영향을 받지 않는다.

5.3 패킷 에러 비율에 따른 시간당 네트워크 처리량

그림 7은 무선 채널 에러가 존재하는 환경에서 무선 노드의 개수에 따른 제안된 기법, 기존 TCP 방식과 I-TCP 방식의 TCP 데이터 흐름 총 처리량을 측정된 결과를 보여준다. 제안된 기법은 무선 구간의 분할 TCP 데이터 흐름은 중복된 전송 계층의 신뢰성 서비스 기능을 제거하고 링크 계층의 신뢰성 서비스에만 의존하고 있기 때문에 무선 채널 에러가 발생하는 경우 신뢰성 서비스 제공에 문제가 발생할 수도 있다. 따라서 패킷 에러 비율이 제안된 기법의 신뢰성 서비스 제공에 어떤 영향을 미치는지 알아보기 위하여 무선 채널 에러



(a) 패킷 에러 비율이 0.1인 경우



(b) 패킷 에러 비율이 0.4인 경우

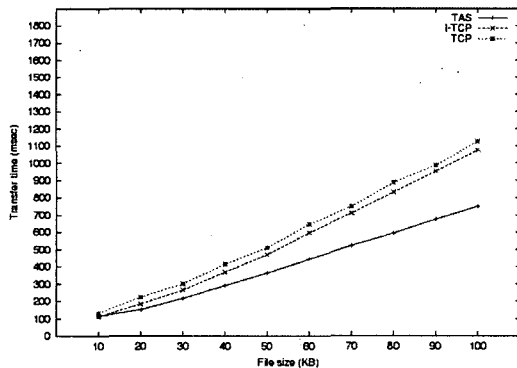
그림 7 무선 노드 수에 따른 업로딩 TCP 데이터 흐름 총 처리량

가 존재하는 환경에서 무선 노드의 개수에 따른 제안된 기법, 기존 TCP 방식과 I-TCP 방식의 TCP 데이터 흐름 총 처리량을 측정하였다. 그림 7에서 볼 수 있듯이 패킷 에러 비율이 증가하면서 TCP 데이터 흐름 총 처리량은 감소하며, 기존 TCP 방식이나 I-TCP 방식의 TCP 데이터 흐름의 경우 무선 노드의 개수와 무관하게 감소한 TCP 데이터 흐름 총 처리량을 유지하고 있다. 비록 제안된 기법이 무선 단말의 개수 증가에 따른 추가적인 패킷 충돌 오버헤드로 인해 TCP 데이터 흐름 총 처리량이 다소 감소하긴 하지만 여전히 기존 TCP 방식이나 I-TCP 방식의 TCP 데이터 흐름 총 처리량보다는 좋은 성능을 보이고 있다.

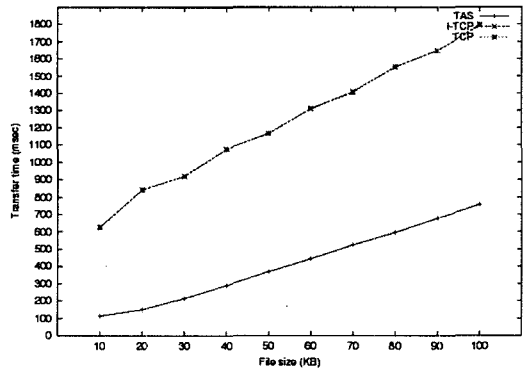
5.4 절전 모드 무선랜에서 평균 지연 시간

절전 모드 무선랜의 경우, 무선 노드의 에너지 소모를 줄일 수는 있지만 좋은 성능을 보이지 못한다는 문제가 널리 알려져 있다. 더욱이 절전 모드에서 동작하는 업로딩 TCP 데이터 흐름은 자신에게 향하는 TCP ACK이 AP에서 버퍼링 되어 다음 주기까지 기다려야 하는 상

황이 발생하여 수명이 짧은 일반 TCP 데이터 흐름에 대하여 RTT를 증가시켜 성능 저하를 유발한다. 절전 모드 무선랜에서 제안한 기법의 성능 측정을 위하여 5개의 무선 노드가 5개의 대응하는 유선 노드에게 FTP 세션을 이용하여 10(KB)에서 100(KB) 크기의 파일을 전송한다. 그림 8은 절전 모드가 아닌 무선랜, 비콘 주기가 50(ms)인 절전 모드 무선랜, 비콘 주기가 200(ms)인 절전 모드 무선랜에서의 파일 전송이 완료되는 시간 측정 결과를 보여준다. 절전 모드 무선랜이 아닌 경우, 제안된 기법은 기존 TCP 방식이나 I-TCP 방식의 TCP 데이터 흐름보다 더 높은 TCP 데이터 흐름 처리량이 가능하기 때문에 파일 전송 완료에 더 짧은 시간을 필요로 한다. 절전 모드 무선랜의 경우, 기존 TCP 방식이나 I-TCP 방식의 TCP 데이터 흐름은 AP 큐에서 TCP ACK 지연에 따라 파일 전송에 좀 더 많은 시간을 필요로 하고, 비콘 주기가 증가하면서 이 현상은 더 심해진다. 그러나 제안된 기법의 경우, AP 큐에서 TCP ACK 지연이 무선 구간의 분할 TCP 데이터 흐름



(a) 절전 모드가 아닌 경우



(b) 비콘 주기가 200(ms)인 절전 모드인 경우

그림 8 전송 파일 크기에 따른 전체 전송 시간

의 성능에 영향을 미치지 않기 때문에 비콘 주기 증가에 따른 영향을 거의 받지 않는다.

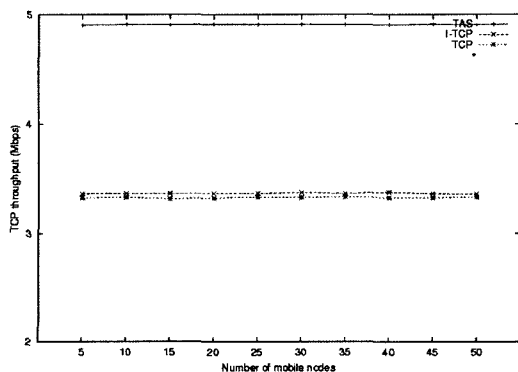
5.5 다운로드 TCP 데이터 흐름의 성능

이전 업로딩 TCP 데이터 흐름에 대한 모든 실험을 다운로드 TCP 데이터 흐름에 대해서도 동일하게 수행하였고, 대부분의 모의 실험에서 업로딩 TCP 데이터 흐름과 유사한 결과를 보였다. 따라서 지면 상의 제약 문제로 다운로드 TCP 데이터 흐름의 측정 결과 중 업로딩 TCP 데이터 흐름과 차이가 있거나 제안된 기법, 기존의 TCP 방식, I-TCP 방식의 다운로드 TCP 데이터 흐름의 동작 과정에서 주목해야 할 모의 실험 결과만을 살펴본다.

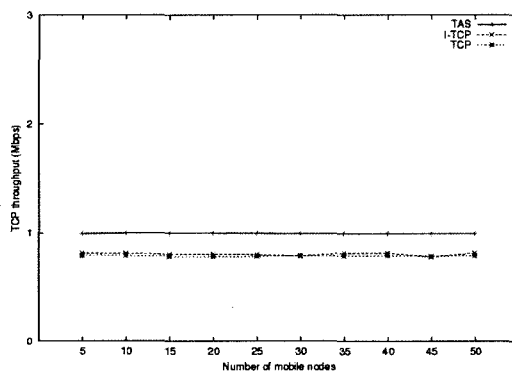
그림 9는 무선 노드의 개수에 따른 제안된 기법, 기존 TCP 방식과 I-TCP 방식의 다운로드 TCP 데이터 흐름 총 처리량을 측정된 결과를 보여준다. 이 때 AP 큐 길이가 미치는 영향을 최소화하기 위해서 1000으로 설정한다. 무선 채널 에러가 발생하지 않는 경우와 달리 무선 채널 에러가 자주 발생하는 경우에는 I-TCP 설계 목적에 따라 무선 구간에서 패킷 손실이 발생한 경우 유선 구간의 불필요한 패킷 재전송을 방지하기 때문에 기존 TCP 방식의 다운로드 데이터 흐름보다 I-TCP 방식의 다운로드 데이터 흐름이 더 좋은 성능을 보일 것으로 예상할 수 있다. 그러나 그림 9(b)에서 볼 수 있듯이 패킷 에러 비율이 상당히 높은 경우에도 I-TCP 방식과 기존 TCP 방식의 다운로드 TCP 데이터 흐름이 거의 비슷한 수준으로 유지된다. 이런 현상의 주요 원인은 링크 계층에서 프레임 손실이 있을 경우, 대부분의 프레임이 4번 혹은 7번의 재전송을 통하여 성공적으로 전송되기 때문에 전송 계층의 관점에서는 패킷 손실이 자주 발생하지 않기 때문이다. 이에 반해 제안된 기법의 경우, 무선 구간에서 불필요한 TCP ACK 전송을 제거하여 실제 TCP DATA 전송을 위해 무선 채널을 더욱

효율적으로 사용함으로써 다운로드 TCP 데이터 흐름의 성능을 향상시키고 있다. 여기서 주목할 것은 그림 7(b)와 그림 9(b)에서 볼 수 있듯이 패킷 에러 비율이 0.4인 때 제안된 기법으로 동작하는 다운로드 TCP 데이터 흐름이 업로딩 TCP 데이터 흐름보다 총 처리량이 다소 낮은 것을 알 수 있다. IEEE 802.11이 모든 경쟁 무선 노드에게 동일한 채널 접근 기회를 부여하기 때문에 업로딩 TCP 데이터 흐름의 경우 하나의 TCP ACK이 전송될 때 여러 TCP DATA가 전송되고, 다운로드 TCP 데이터 흐름의 경우 하나의 TCP DATA가 전송될 때 여러 TCP ACK이 전송된다. 따라서 패킷 에러가 동일한 확률로 발생한다면 업로딩 TCP 데이터 흐름의 경우 TCP DATA, 다운로드 TCP 데이터 흐름의 경우 TCP ACK이 손실된다. TCP ACK 손실이 TCP DATA 손실보다 TCP 데이터 흐름 및 혼잡 제어 알고리즘에 미치는 영향이 더 크기 때문에 다운로드 TCP 데이터 흐름의 총 처리량이 더 낮다.

그림 10은 무선 노드 수에 따른 제안된 기법, 기존 TCP 방식, I-TCP 방식의 다운로드 TCP 데이터 흐름의 형평성 지표를 보여준다. AP 큐 길이가 큰 경우에는 분할 지점에서 패킷이 버려지는 경우가 거의 발생하지 않기 때문에 모든 업로딩 및 다운로드 TCP 데이터 흐름들이 공평하게 무선 자원을 할당받는다. 그러나 AP 큐 길이가 작은 경우에는 무선 구간에서 업로딩 TCP 데이터 흐름의 TCP ACK이 분할 지점에서 버려지게 된다. 따라서 기존 TCP 방식과 I-TCP 방식의 업로딩 TCP 데이터 흐름은 분할 지점에서 버려지는 TCP ACK에 따라 무선 자원이 불공평하게 할당되지만, 제안된 기법에서는 실제 TCP ACK 전송을 제거함으로써 AP 큐에서 버려지는 TCP ACK이 존재하지 않아 형평성 문제가 발생하지 않는다. 그러나 업로딩 TCP 데이터 흐름과는 달리 다운로드 TCP 데이터 흐름의 경우에

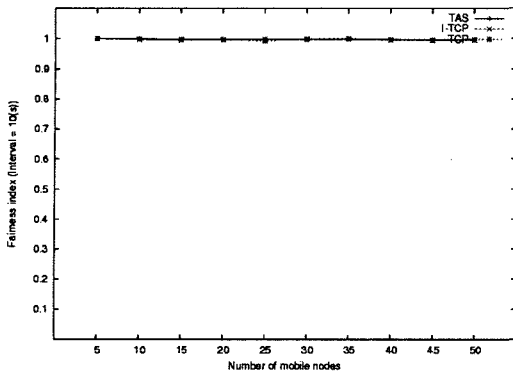


(a) 패킷 에러가 없는 경우

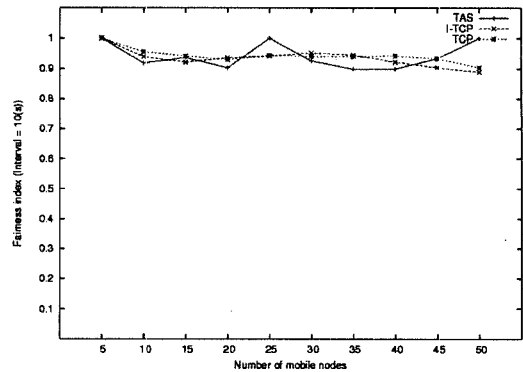


(b) 패킷 에러 비율이 0.4인 경우

그림 9 무선 노드 수에 따른 다운로드 TCP 데이터 흐름 총 처리량



(a) AP 큐 길이가 1000인 경우



(b) AP 큐 길이가 100인 경우

그림 10 무선 노드 수에 따른 다운로드 TCP 데이터 흐름 형평성 지표

는 AP 큐 길이가 작아서 패킷 손실이 발생하더라도 각각의 무선 노드로 향하는 TCP DATA가 확률적으로 동일하게 손실되고, 업로딩 TCP 데이터 흐름의 TCP ACK 손실보다는 데이터 흐름 및 혼잡 제어 알고리즘에 미치는 영향이 크지 않기 때문에 기존 TCP 방식이나 I-TCP 방식의 다운로드 TCP 데이터 흐름의 총 처리량은 감소하지만 다운로드 TCP 데이터 흐름 사이의 형평성 문제는 심각하게 발생하지 않는다. 제안된 기법의 업로딩 TCP 데이터 흐름 경우에는 분할 지점에서 손실되는 패킷이 없었지만, 다운로드 TCP 데이터 흐름 경우에는 TCP DATA가 손실되기 때문에 약간의 변동이 있기는 하지만 대체적으로 무선 자원이 공평하게 할당되고 있다.

6. 결론

인터넷에 많은 응용들이 신뢰성 서비스를 제공하는 TCP 기반으로 동작하고 있고 무선 네트워크 환경에서 TCP 데이터 흐름의 성능 향상을 위한 다양한 연구가 있었지만, 현재 가장 널리 사용되고 있는 무선 네트워크 중의 하나인 IEEE 802.11 기반 무선랜에서는 충분한 성능 향상을 보여주지 못하기 때문에 IEEE 802.11 기반 무선랜 환경에서 TCP 데이터 흐름 성능 향상 연구가

우선적으로 수행되어야 한다. TCP 데이터 흐름 성능 저하의 주요 원인은 전송 계층의 TCP 데이터 흐름 및 혼잡 제어 알고리즘이 IEEE 802.11 링크 계층과의 상호 작용에 의해서 실제 무선 채널 획득을 위해 경쟁에 참여하지 않기 때문이다. 이런 문제를 해결하기 위해서 논문에서는 링크 계층과 전송 계층의 중복된 기능을 제거하여 TCP 데이터 흐름의 성능을 향상시키는 새로운 TAS 기법을 제안하였다. 즉, 무선 채널에서 실제 TCP ACK을 전송하지 않고, TCP DATA를 전송한 무선 노드가 MAC ACK 기반으로 TCP ACK을 에뮬레이션한다. 제안된 기법의 성능 평가를 위해 NS2 구현을 통한 모의 실험을 수행하여, 기존 TCP 방식과 I-TCP 방식의 TCP 데이터 흐름과 비교하였다. 모의 실험 결과는 패킷 에러가 있는 환경에서도 TCP 데이터 흐름의 총 처리량 및 무선 자원 할당의 형평성 관점에서 다른 기법에 비해 성능 향상이 있음을 보여 준다. 또한 절전 모드인 경우, 제안된 기법은 수명이 짧은 TCP 데이터 흐름의 경우에도 전송 지연 시간을 단축할 수 있다. 현재 실제 제안된 기법의 성능 측정을 위한 구현이 진행 중이고, 기존 TCP 방식의 TCP 데이터 흐름과 혼재할 경우 발생하는 형평성 문제를 해결하기 위한 연구가 진행 중이다.

부 록

A. TAS 간략(pseudo) 구현 코드

```

Class Tas: public Agent {
    ...
    Packet* TcpData_;
    NsObject* tas_uptarget_;
    NsObject* tas_downtarget_;
}

Tas::Tas(MobileNode* node, Mac802_11* mac): Agent(PT_NTYPE) {
    tas_uptarget_ = node->entry_;
    
```

```

tas_downtarget_ = mac;
...
}

void Tas::recv(pkt) {
...
if direction(pkt) == DOWN then
if type(pkt) == TCP_DATA then
if checkTasMgtTable(pkt) == ADMITTED then
    TcpDatatx = pkt→copy();
    tas_downtarget_→recv(pkt, NULL);
else if type(pkt) == TCP_ACK then
    drop(pkt);
else if direction(pkt) == UP then
    tas_uptarget_→recv(pkt, NULL);
...
}

void Tas::emulateTcpAck() {
    Packet* p = Packet::alloc();
    setTcpAck(p, TcpDatatx); // 직전에 전송한 TcpDatatx로부터 TCP ACK 정보 설정 (패킷 p)

    tas_uptarget_→recv(p, NULL);
}

Class Mac802_11: public Mac {
...
    Tas* tas_;
}

Mac802_11::Mac802_11(): Mac(), ... {
...
    tas_ = new Tas(node, this);
}

void Mac802_11::recvACK(pkt) {
    if type(MSDUtx) == TCP_DATA then
        tas_→emulateTcpAck();
...
}

```

참고 문헌

- [1] A. DeSimone, C. Mooi Choo, and Y. On-Ching, "Throughput performance of transport-layer protocols over wireless LANs," in IEEE in Houston. GLOBECOM '93. IEEE Global Telecommunications Conference, including a Communications Theory Mini-Conference. Technical Program Conference Record (Cat. No.93CH3250-8), 1993, pp. 542-9 vol.1.
- [2] H. Balakrishnan, S. Seshan, and R. H. Katz, "Improving reliable transport and handoff performance in cellular wireless networks," Wireless Networks, vol. 1, pp. 469-481, 1995.
- [3] K. Ratnam and I. Matta, "WTCP: an efficient mechanism for improving TCP performance over wireless links," in Proceedings Third IEEE Symposium on Computers and Communications. ISCC'98. (Cat. No.98EX166), 1998, pp. 74-78.
- [4] K. Brown and S. Singh, "M-TCP: TCP for mobile cellular networks," Computer Communication Review, vol. 27, pp. 19-43, 1997.
- [5] A. Bakre and B. R. Badrinath, "I-TCP: indirect TCP for mobile hosts," in Proceedings of the 15th International Conference on Distributed Computing Systems (Cat. No.95CH35784), 1995, pp. 136-143.
- [6] W. Haitao, P. Yong, L. Keping, C. Shiduan, and M. Jian, "Performance of reliable transport protocol

over IEEE 802.11 wireless LAN: analysis and enhancement," in Proceedings. IEEE INFOCOM 2002 Conference on Computer Communications. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No.37364), 2002, pp. 599-607 vol.2.

- [7] S. Pilosof, R. Ramachandran, D. Raz, Y. Shavitt, and S. Prasun, "Understanding TCP fairness over wireless LAN," in IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies(IEEE Cat. No.03CH37428), 2003, pp. 863-72 vol.2.
- [8] K. Sung Won, K. Byung-Seo, and F. Yuguang, "Downlink and uplink resource allocation in IEEE 802.11 wireless LANs," IEEE Transactions on Vehicular Technology, vol. 54, pp. 320-327, 2005.
- [9] D. J. Leith and P. Clifford, "Using the 802.11e EDCF to achieve TCP upload fairness over WLAN links," in Third International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks, 2005, pp. 109-118.
- [10] C. H. N. Anthony, M. David, and J. L. Douglas, "Experimental evaluation of TCP performance and fairness in an 802.11e test-bed," in Proceeding of the 2005 ACM SIGCOMM workshop on Experimental approaches to wireless network design and analysis. Philadelphia, Pennsylvania, USA: ACM Press, 2005.
- [11] IEEE Computer Society, "802.11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification," Jun. 1997.
- [12] N.Choi, J.Ryu, Y.Seok, Y.Choi, and T.Kwon, "Throughput enhancement for uploading TCP flows in IEEE 802.11 wireless LANs," Proc. ICCE 2006, Jan. 2006.
- [13] R.Krashinsky and H.Balakrishnan, "Minimizing energy for wireless web access with bounded slow-down," Proc. Mobile Computing and Networking, 2002.
- [14] IEEE P802.11E/D9.0 Draft Standard, Aug. 2004.
- [15] NS-2, URL <http://www.isi.edu/nsnam/ns>.
- [16] J.Ryu, Y.Seok, Y.Choi, T.Kwon, and J.-M.Bonnin, "S-EDCF: EDCF based on superslot and pseudo collision," ACM MobiCom Student Poster Session, 2005.
- [17] R.Jain, G.Babic, B.Nagendra, and C.Lam, "Fairness, call establishment latency and other performance metrics," Tech. Rep. ATM Forum/96-1173, ATM Forum Document, Aug. 1996.



최 낙 중

2002년 2월 서울대학교 전기컴퓨터공학부 졸업. 2004년 2월 서울대학교 전기컴퓨터공학부 석사. 2004년 3월~현재 서울대학교 전기컴퓨터공학부 박사과정. 관심분야는 멀티홉 무선 네트워크, 무선랜 MAC 프로토콜 개선. 이기종 망의 연동



정 하 경

2005년 2월 서울대학교 전기컴퓨터공학부 졸업. 2005년 3월~현재 서울대학교 전기컴퓨터공학부 석사과정. 관심분야는 무선네트워크 MAC 프로토콜 개선, 게임 이론, 유비쿼터스 네트워킹



류 지 호

2005년 2월 한국과학기술원 전산학과 졸업. 2005년 3월~현재 서울대학교 전기컴퓨터공학부 석사과정. 관심분야는 무선랜 QoS 보장, 게임이론, 네트워크 이동성



석 용 호

2000년 2월 한동대학교 전산전자공학부 졸업. 2002년 2월 서울대학교 전기컴퓨터공학부 석사. 2005년 8월 서울대학교 전기컴퓨터공학부 박사. 관심분야는 인터넷 트래픽 측정, 트래픽 엔지니어링, 무선랜 QoS, 멀티홉 무선 네트워크



최 양 호

1975년 서울대학교 전자공학과 공학사 1977년 한국과학기술원 전자공학과 석사 1984년 프랑스 국립전기통신대학 컴퓨터공학 박사. 1991년~현재 서울대학교 컴퓨터공학과 교수. 2005년~현재 한국과학기술원 학원 회원. 관심분야는 무선 모바일 네트워크, 미래 인터넷, IP기반 멀티미디어 트래픽



권 태 경

1993년 서울대학교 컴퓨터공학과 공학사 1995년 서울대학교 컴퓨터공학과 석사 2000년 서울대학교 컴퓨터공학과 박사 2004년~현재 서울대학교 컴퓨터공학과 조교수. 관심분야는 센서 네트워크, 무선 네트워크, IP 이동성, 유비쿼터스 컴퓨팅