

확장된 ECN 메커니즘을 사용한 무선 링크에서의 TCP 성능 제어 기법

(TCP Performance Control Method for the Wireless Link by using Extended ECN Mechanism)

윤 여 훈[†] 김 태 윤^{**}
(You-Hun Yun) (Tai-Yun Kim)

요 약 오늘날 무선망의 출현으로 기존의 인터넷 환경은 유/무선이 통합된 단일 망으로 변화하고 있다. 그러나 현재 TCP는 전송상의 모든 패킷 손실을 혼잡으로 인한 손실로 판단하여 혼잡 윈도우를 줄이는 등의 혼잡 제어 메커니즘을 호출한다. 이것을 무선 구간에서 적용시켰을 때 핸드오프나 비트 에러로 인한 패킷 손실이 발생할 때조차 혼잡으로 인한 패킷 손실로 판단하여 혼잡 윈도우를 줄이기 때문에 종단간 TCP 처리량을 저하시킨다.

본 논문에서는 이러한 문제들을 해결하기 위해 기존의 유선망에서 혼잡 제어 메커니즘으로 사용되던 ECN(Explicit Congestion Notification)을 확장하여 무선 링크 상에서의 TCP 성능을 제어하는 기법을 제시한다. 이것은 패킷 손실이 혼잡에 의한 것인지, 무선 링크 구간에서 비트 에러 또는 핸드오프에 의한 것인지를 구분하는 기법으로 유/무선이 통합된 망에서 혼잡이 발생할 때만 혼잡 제어 메커니즘을 호출하도록 하는 기법이다.

키워드 : TCP, ECN, 무선, 혼잡 제어 메커니즘, 빠른 재전송, 빠른 회복

Abstract Nowadays, after appearance of wireless network the existent internet environment is changing into the united wire/wireless network. But the present TCP regards all of the packet losses on transmission as the packet losses due to the congestion. When it is applied on the wireless path, it deteriorates the end-to-end TCP throughput because it regards the packet loss by handoff or bit error as the packet loss by the congestion and it reduces the congestion window.

In this paper, for solving these problems we propose the method that controls the performance of TCP on the wireless link by extending ECN which is used as a congestion control mechanism on the existent wire link. This is the method that distinguished the packet loss due to the congestion from due to bit error or handoff on the wireless network, so it calls the congestion control mechanism only when there occurs the congestion in the united wire/wireless network.

Key words : TCP, ECN, Wireless, Congestion Control Mechanism, Fast Retransmission, Fast Recovery

1. 서 론

최근 무선 통신의 대중화로 무선망이 미래의 인터넷 정보 사회에서 중요한 역할을 담당하게 될 것으로 기대된다. 무선망의 출현으로 기존의 인터넷 환경은 유/무

선이 통합된 단일 망으로 변화되고 있다. 그러나 오늘날 통신 환경에서 널리 사용되고 있는 TCP(Transmission Control Protocol)은 유선망의 특성을 적용한 것이므로 유선망과 고정된 호스트로 이루어진 전통적인 네트워크에 적합하다. 현재 TCP 프로토콜은 전송도중 패킷이 손실되면 송신측과 수신측 사이의 경로상에 있는 중간 노드에서 혼잡(congestion)이 발생한 것으로 판단한다. 그 결과 송신측은 혼잡 제어 윈도우를 줄이는 등의 혼잡 제어 모드로 패킷 손실에 대해 반응한다. 이러한 TCP를

[†] 학생회원 : 고려대학교 컴퓨터학과
joy1223@hanmir.com

^{**} 종신회원 : 고려대학교 컴퓨터학과 교수
tykim@netlab.korea.ac.kr

논문접수 : 2001년 12월 8일
심사완료 : 2002년 3월 7일

어떠한 수정도 없이 무선 구간에서 적용하고 있어 종단간의 처리량(throughput) 저하를 가져오고 있다[1, 2]. 이 같은 성능 저하는 모바일 호스트가 다른 셀(cell)로 이동할 때 발생하는 패킷 손실 즉 핸드오프 시 연결 단절(disconnection)로 인한 패킷 손실과 무선망에서의 비트 에러율(BER : Bit Error Rate)에 의한 패킷 손실에 대해 TCP는 기본적으로 유선망에서의 혼잡에 의한 손실로 판단하여 혼잡 윈도우를 줄이는 등의 혼잡 제어 메커니즘을 호출하기 때문에 발생한다[3, 4, 5]. 어떠한 원인으로 인해 혼잡 윈도우를 줄인 다음 원래의 혼잡 윈도우로 복귀하기까지는 많은 RTT(Round Trip Time)가 필요하다. 따라서 혼잡에 의한 패킷 손실 이외의 원인으로 혼잡 제어 메커니즘을 자주 호출하는 것은 종단간의 처리량이 현저하게 떨어지는 문제를 가져오게 된다.

최근에 무선 링크 구간의 특성을 고려하여 TCP 성능을 향상시키기 위해 연결 분할 방식(split-TCP connection)을 사용하는 기법들이 제안되었고 그 예로 snoop[5, 6], I-TCP[4, 5, 6], M-TCP[5, 6]가 있다. 그러나 이 기법들은 무선 구간에서 비트 에러나 핸드오프가 발생했을 때 손실된 패킷들에 대해 BS(Base Station) 또는 SH(Supervisor Host)와 같은 중간 노드에서 지역적으로 재전송(local retransmission)하는 메커니즘을 사용하기 때문에 지역적 재전송의 특성상 전송 지연시간이 증가될 뿐만 아니라 중간 노드를 관리하는데 있어서의 오버헤드도 크다. 또한 종단간 의미(end-to-end semantics)를 유지하지 못하는 등의 문제를 일으킬 수 있다[4, 5, 6].

본 논문에서는 무선 링크 구간에서 TCP 성능을 향상시키는데 있어서 지역적 재전송 기법을 사용하지도 무선에서의 TCP 성능을 제어할 수 있고 종단간 의미를 유지할 뿐만 아니라 유선 링크 구간에서의 혼잡도 제어하는 기법을 제시한다. 이것은 기존의 유선 망에서 혼잡 제어 메커니즘으로 사용되던 ECN(Explicit Congestion Notification)을 확장하여 사용하는 기법으로 기존의 연결 분할 방식을 따르는 기법들과는 달리 패킷 손실의 원인이 혼잡에 의한 것인지 비트 에러에 의한 것인지 또는 핸드오프에 의한 것인지를 송신측이 직접 구분하여 처리하도록 하는 기법이다.

본 논문의 구성은 다음과 같다. 먼저, 2장에서는 무선 구간에서 TCP 성능 향상을 위해 기존의 유선 망에서 혼잡 제어 메커니즘으로 사용되던 ECN을 사용한 수정된 TCP 프로토콜을 제안한다. 3장에서는 제안한 기법의 시뮬레이션 결과를 나타낸 후 기존의 기법과 제안한 기법의 성능을 비교 분석했다. 마지막 4장에서는 결론을 나타낸다.

2. 무선구간에서 TCP 성능을 제어하는 확장된 ECN 메커니즘

본 장에서는 유/무선이 통합된 환경에서 무선 링크상의 TCP 성능을 제어하기 위해 기존의 유선 망에서 혼잡 제어 메커니즘으로 사용되던 ECN[3, 7, 8, 9]을 확장하여 사용한 기법을 제시한다. 이것은 송신측으로 하여금 패킷 손실의 원인이 혼잡에 의한 것인지, 비트 에러에 의한 것인지 또는 핸드오프 발생에 의한 것인지를 구분하는 기법으로 TCP가 ECN이 가능하다는 가정하에 기존의 ECN에 있어서 초기 연결 설정단계와 송신측의 역할을 수정한 것이다. 즉 기존의 ECN의 연결 설정단계와는 달리 송신측으로 하여금 현재 수신측의 연결이 유선 연결인지 무선 연결인지를 구분하도록 하였고, 수신측의 연결이 무선 연결일 때는 수신측으로부터 전송되어 오는 ACK 패킷을 이용하여 송신측으로 하여금 패킷 손실의 원인이 혼잡에 의한 것인지, 비트 에러에 의한 것인지 또는 핸드오프 발생에 의한 것인지를 구분하여 처리하는 기법이다. 따라서 본 장에서는 2.1절과 2.2절을 통해 제안한 확장된 ECN 메커니즘에서의 TCP 초기화와 수정된 송신측의 역할에 대해 각각 상세하게 설명할 것이다.

2.1 제안한 메커니즘에서의 TCP 초기화

제안한 기법은 TCP가 ECN이 가능하다는 가정을 전제로 하고있다. 이 기법은 ECN TCP 초기 연결 설정에서 송신측으로 하여금 수신측이 유선 연결인지 혹은 무선 연결인지를 구분한다. 수신측과의 연결에 대한 이러한 구분의 목적은 무선 구간에서의 TCP 성능을 제어하기 위한 기능 외에, 무선 구간에서의 TCP 성능 제어 기법이 유선 연결에도 적용되어 성능이 오히려 저하되는 것을 방지하기 위함이다.

• 유선 연결일 경우

기존의 ECN에서와 같이 송신측 호스트는 초기 연결 단계 시에 SYN 패킷(SYN packet: 연결을 시작하기 위해 시퀀스 번호를 동기화 하는 패킷)에 TCP 헤더의 예약 필드(reserved field) 중 9, 8 번째 비트에 각각 위치되어 있는 ECE(ECN-Echo)와 CWR(Congestion Window Reduced) 비트를 설정하여 수신측으로 보낸다. 이 SYN 패킷을 받자마자 수신자는 <그림 1>과 같이 SYN-ACK 패킷의 ECE 비트만을 설정한다. 이런 동기가 이루어지면 송신측은 ECN이 가능하다는 것을 네트워크에게 알리기 위해 IPv4 헤더에서는 TOS 필드의 6 번째 비트 또는 IPv6에서는 헤더의 10 번째 비트에 위치되어 있는 ECT(ECN Capable Transport

Indicator) 비트를 설정한다. 이와 같이 유선의 경우에 송신측은 기존의 ECN TCP와 동일한 혼잡제어 메커니즘을 사용한다.

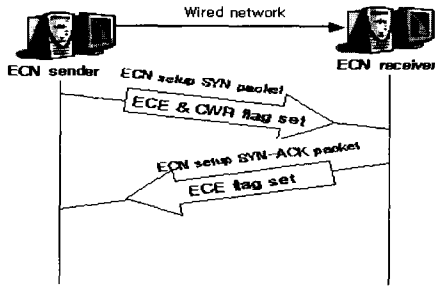


그림 1 유선 연결일 때의 연결 설정

• 무선 연결일 경우

송신측 호스트는 연결 설정 시에 SYN 패킷의 ECE와 CWR 비트를 설정하여 수신측으로 보낸다. 이 SYN 패킷을 받자마자 수신자는 <그림 2>와 같이 SYN-ACK 패킷에 ECE와 CWR 비트 모두를 설정하여 현재 목적지의 연결이 무선이라는 것을 송신측 호스트에게 알린다. 이런 동의가 이루어지면 송신측은 ECN이 가능하다는 것을 네트워크에 알리기 위해 데이터 패킷의 IP 헤더에 ECT 비트를 설정하여 보내고 무선 상황에 적합한 혼잡 제어 모드로 동작하게 된다. 무선 상황에서의 혼잡 제어 모드에 대해서는 2.2절 TCP 송신측의 역할 부분에서 상세히 설명하겠다.

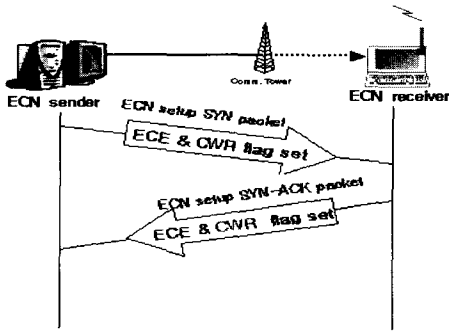


그림 2 무선 연결일 때의 연결 설정

2.2 제안한 메커니즘에서 송신측의 역할

앞 절에서 설명한 것과 같이 최초의 연결 설정에서 각각의 경우 즉 무선 연결, 유선 연결에 따라 송신측의 메커니즘을 달리한다.

<그림 3>은 각 연결 설정에 따른 송신자의 혼잡 제어를 나타내고 있다.

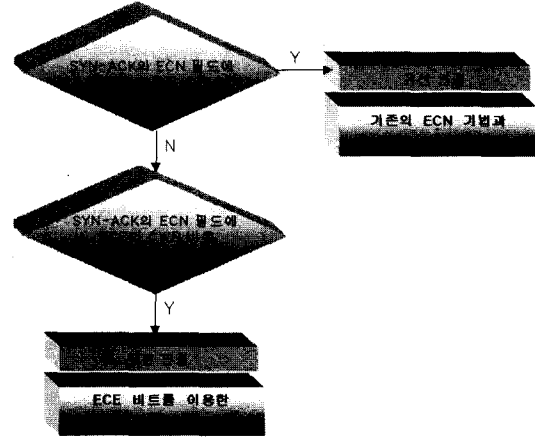


그림 3 연결 설정에 따른 송신측의 혼잡 제어

SYN-ACK 패킷의 ECN 필드의 ECE 플래그 비트만 설정되었다면 이는 유선망과의 연결을 나타내므로 기존의 ECN과 동일한 혼잡제어 메커니즘으로 동작할 것이다. 즉, TCP에서의 혼잡에 의한 패킷 손실 때의 빠른 재전송/회복 알고리즘과 동일한데, ECE가 설정된 ACK를 받았을 때 송신측은 혼잡 윈도우를 절반으로 줄이고 슬로우 스타트 경계값을 줄인다. 빠른 재전송/회복은 혼잡으로 인해 패킷이 손실되었다는 의미로 수신측으로부터 전송되어온 3개의 중복된 ACK를 받았을 때에도 동일하게 반응한다.

SYN-ACK 패킷의 ECN 필드의 ECE와 CWR 필드 모두가 설정되어 있다면 이는 무선망과의 연결을 나타내므로 이 경우에 기존의 메커니즘에서 중점적으로 수정되는 부분이며, 패킷 손실의 원인이 혼잡에 의한 것인지 무선에서의 비트 에러 또는 핸드오프에 의한 것인지를 구분한다. 패킷 손실에 대한 이러한 구분은 수신측으로부터 전송되어온 ECE가 설정된 ACK 패킷에 의해 이루어진다. <그림 4>는 수신측으로부터 전송되어온 ECE가 설정된 ACK 패킷을 통해 송신측이 패킷 손실의 원인이 혼잡에 의한 것인지, 무선 링크에서의 비트 에러 또는 핸드오프에 의한 것인지를 구분하는 판단 근거 및 각각에 대한 처리 기법을 나타낸 것이다.

먼저 <그림 4> (a)는 단지 ACK 패킷에 ECE 플래그 비트가 설정되어 있는 경우이다. 이 때 송신측은 송신측과 수신측 사이의 네트워크 경로 상에 단순히 혼잡이 발

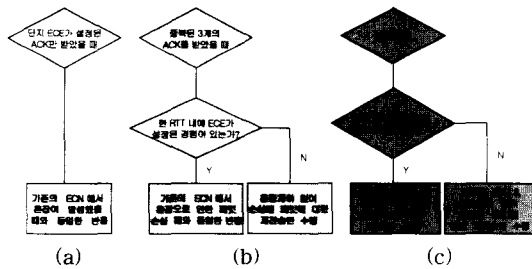


그림 4 무선 상황을 고려한 ACK에 따른 송신자의 반응

생했음을 알 것이다. 이와 같은 혼잡 지시에 대해 기존의 ECN에서의 혼잡이 발생했을 때와 동일한 반응을 한다. 즉 송신측은 혼잡 제어 윈도우를 절반으로 줄이고 스타트 경계값을 줄인다. 이 때 송신측 TCP는 ECE가 설정된 ACK 패킷을 받은 반응으로 혼잡 윈도우를 증가시키지 않아야 한다. TCP는 하나의 데이터 윈도우(패 RTT) 내에 ECE가 설정된 일련의 ACK 패킷들에 대해 1번 이상 반응하지 않는다. 즉, TCP 송신자의 혼잡 윈도우는 단일 데이터 윈도우에서 ECE가 설정된 일련의 ACK 패킷들에 대해 ACK 패킷 개수만큼 줄이는 것이 아니라 오직 한번만 줄여야 한다는 것이다. 또한 TCP 송신자는 ECE에 반응하여 자신의 혼잡 윈도우를 줄였을 때 다음에 수신측에 보낼 첫 번째 데이터 패킷의 CWR을 설정하여 더 이상 ECE를 설정하지 않을 것을 알린다. CWR을 설정하여 보내는 것은 ECE가 설정된 ACK에 대해 반응하는 모든 경우에 적용된다.

다음으로 <그림 4> (b)는 패킷 손실의 의미인 연속된 3 개의 중복된 ACK를 받은 경우이다. 이 경우를 설명할 때 Ramakrishnan 등이 [7]에서 기존의 ECN을 기술할 때에 각각 평균 큐 길이와 ECN 버퍼의 상위 경계값, 하위 경계값으로 정의한 기호인 avg , TH_{max} , TH_{min} 을 사용할 것이다. 3개의 중복된 ACK를 받은 경우 만약 현재까지의 한 RTT 내에 ECE가 설정된 경험이 있다면 이것은 유선 링크 구간에서 ECN 버퍼에서의 avg 가 TH_{max} 를 넘어서서 혼잡에 의한 패킷 손실이 발생한 경우이거나, avg 가 TH_{min} 과 TH_{max} 사이의 값이어서 단지 혼잡이 시작된 상태에서 무선 링크 구간의 비트 에러가 발생한 경우이다. 결과적으로 혼잡이 일어난 상태에서의 패킷 손실으므로 송신측은 이것을 혼잡으로 인한 패킷 손실로 간주하여 기존의 ECN에서 혼잡으로 인한 패킷 손실 때와 동일한 방법으로 동작할 것이다. 즉 손실된 패킷에 대해 빠른 재전송을 한 다음 혼잡 윈도우를 절반으로 줄이고 슬로우 스타트 경계값을

줄인다. 이 때 ECE가 설정된 ACK들에 대해 혼잡 윈도우와 슬로우 스타트 경계값을 중복하여 줄이지 않아야 한다. 3 개의 중복된 ACK를 받았으나 유선 구간에서 혼잡이 발생하지 않아 한 RTT 내에 ECE가 설정된 경험이 없다면 무선 환경의 특성을 고려하여 송신자는 이를 혼잡에 의한 손실로 간주하지 않고 무선 링크 구간에서의 비트 에러에 의한 패킷 손실로 판단한다. 이 경우는 혼잡 제어 메커니즘을 호출할 없이 단순히 손실된 패킷의 재전송만 수행할 것이다. 즉 송신자는 유선 연결 때와는 달리 자신의 혼잡 윈도우와 슬로우 스타트 경계값을 줄이지 않고 단순히 손실된 패킷에 대해 재전송만 수행한다.

마지막으로 <그림 4> (c)는 지형적인 영향으로 인한 페이딩 및 핸드오프가 자주 발생하는 무선 상황을 고려한 것으로 타임아웃이 발생했을 때의 경우이다. 타임아웃이 발생하였을 시 현재까지의 한 RTT 동안에 ECE 비트의 설정 유무에 따라 혼잡 경험이 있는지를 조사한다. 만약 한 RTT 내에 ECE가 설정되어 혼잡 경험이 있다면 일단 송신자는 전송을 중단하고 지속적 모드(persist mode)로 들어간다. 즉 혼잡 윈도우와 슬로우 스타트 경계값을 유지하고 재전송 타이머를 멈춘 채 패킷 전송을 중지한 후 주기적으로 탐색(probe) 패킷을 수신측으로 전송한다. 탐색 패킷에 대한 ACK가 왔을 때 기존의 TCP에서의 타임아웃 때와 동일한 방법으로 반응한다. 즉 혼잡 윈도우를 1로 줄인 채 슬로우 스타트를 시작하여 손실된 패킷들을 재전송하기 시작한다. 만약 한 RTT 내에 ECE가 설정되지 않아 혼잡 경험이 없는 것으로 판단했을 때 송신자는 지속적 모드로 들어간 다음 탐색 패킷에 대한 ACK가 왔을 때 혼잡 제어 없이 연결이 끊기기 전의 혼잡 윈도우와 슬로우 스타트 경계값으로 손실된 패킷들에 대한 재전송을 시작한다.

3. 실험 결과 및 성능 분석

3.1 실험 결과

본 연구에서 제안한 기법의 성능을 실험하기 위해 버클리 대학(U.C. Berkeley)에서 개발하여 배포한 네트워크 시뮬레이터 NS-2를 이용하였다. NS-2는 OTCL/TK를 사용하여 스크립트 파일로 작성하여 구동하도록 되어있다.

NS-2에서 제안된 기법이 구현된 기본적인 시뮬레이션 환경은 <그림 5>와 같다. 노드 ①, ②는 송신측 FH(Fixed Host)를 나타내고 노드 ③, ④는 게이트웨이, 노드 ⑤, ⑥는 수신측 MH(Mobile Host)를 나타낸다. 그리고 ②와 ③ 사이는 공유링크로 연결되어 있고, 링크

③-④, ③-⑤구간은 무선링크로 연결되어 있다. 유선 링크 및 무선 링크 구간의 대역폭과 전송지연을 1Mbps와 2ms로 설정하였고 공유링크의 대역폭과 전송지연은 1Mbps와 20ms로 설정하였다. 링크 ②-③구간이 ECN이 가능하도록 설정하였고, 버퍼 용량은 1500byte packets을 기준으로 25 packets으로 설정하였다. 성능 평가를 위해 고려된 파라미터는 송신자의 혼잡 윈도우, 무선 링크 구간의 에러율, 무선 링크 구간에서의 연결단절 시간, 송/수신 노드의 수(플로우 수), 송신측의 처리량이다. 송/수신측에서 사용되는 애플리케이션은 FTP이다.

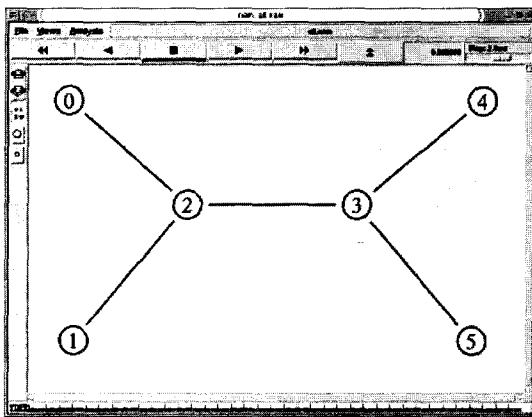


그림 5 시뮬레이션 기본 토폴로지

일반적으로 무선 링크 상의 비트 에러율이 $10^{-2} \sim 10^{-4}$ 으로 간주된다[2]. 따라서 <그림 6>은 비트 에러율을 10^{-3} 즉 0.001로 설정한 상태에서 ①에서 ④까지 데이터를 전송한 결과 시간 변화에 따른 혼잡 윈도우(cwnd)의 변화에 대해 일정시간의 구간을 확대한 그림이다. 제안한 ECN TCP는 유선 구간에서 혼잡이 발생할 때만 혼잡 윈도우를 줄이기 때문에 혼잡 윈도우의 변화(빈도수)가 가장 작게 나타났고 혼잡 윈도우의 사이즈 역시 크게 나타났다. 그러나 기존의 TCP 및 ECN TCP는 무선구간에서 0.001의 비트 에러율에 의한 패킷 손실조차도 혼잡으로 인한 패킷 손실로 간주한다. 따라서 중복된 3개의 ACK 또는 ECN이 설정된 패킷을 받을 때마다 혼잡 윈도우를 줄였기 때문에 그래프의 혼잡 윈도우의 변화가 제안한 ECN TCP에 비해 상대적으로 크게 나타났다. 뿐만 아니라 기존의 ECN TCP와 일반 TCP는 혼잡 윈도우를 자주 줄이기 때문에 혼잡 윈도우의 평균적인 크기가 제안된 ECN 보다 상대적으로 작게 나타났다.

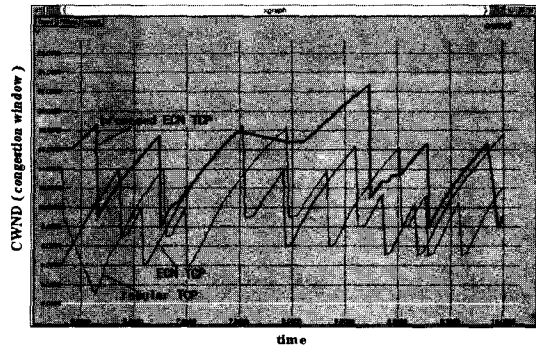


그림 6 송신측(노드 ①)의 시간에 따른 혼잡 윈도우의 변화

<그림 7>은 무선 링크 구간에서의 패킷 손실율에 따른 처리량을 실험한 결과이다. 플로우 수(flow number)는 ①에서 ④로 전송되는 것 하나이며 시뮬레이션 타임은 30초로 설정하였다. 제안된 TCP와 기존의 snoop이 무선 구간에서의 비트 에러에 의한 패킷 손실에 대해 혼잡 윈도우를 줄이지 않기 때문에 일반적인 TCP에 비해 성능이 우수한 것으로 나타났다. 이 경우는 플로우 수가 하나 밖에 없는 경우인데 제안한 TCP가 무선 구간에서 비트 에러에 효율적으로 반응하기 위한 목적으로만 고안된 snoop의 성능 보다 향상된 것으로 나타났다.

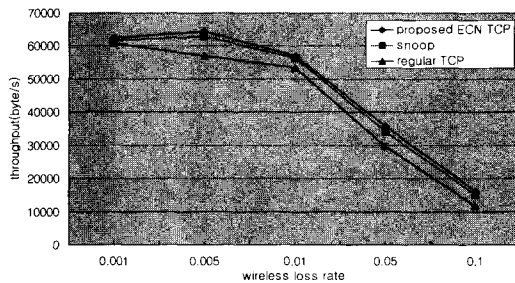


그림 7 무선 링크의 에러율에 따른 송신자(노드 ①) 처리량

<그림 8>은 노드 ①을 포함하여 FH와 FH가 전송하는 데이터를 받을 MH의 수 즉, 플로우 수를 증가시킴에 따른 ①에서 ④로 전송된 처리량을 나타낸 것이다. 이는 무선 링크의 비트 에러에 의한 패킷 손실율과 함께 노드 수(노드 ① 포함)가 증가함에 따라 공유 링크에서의 혼잡이 일어날 수 있는 상황을 고려하여 송신자의 처리량을 측정 한 실험이다. 시뮬레이션 시간은 30초이고

무선 링크 에러율을 0.01로 설정하여 플로우 수를 1에서 10까지 증가시키기에 따른 처리량을 측정한 결과이다. 플로우 수가 증가함에 따라 변동은 조금 있지만 전체적인 각각의 플로우 수에 따른 처리량은 제안된 TCP가 기존의 TCP와 snoop 보다 성능이 우수함을 볼 수 있었다. 이 때 혼잡이 증가될수록 제안된 TCP가 <그림 7>의 경우와 비교하여 snoop 보다 성능이 더욱 우수한 것으로 나타났는데 이것은 제안한 TCP가 유선구간에서의 혼잡도 제어할 수 있기 때문이다.

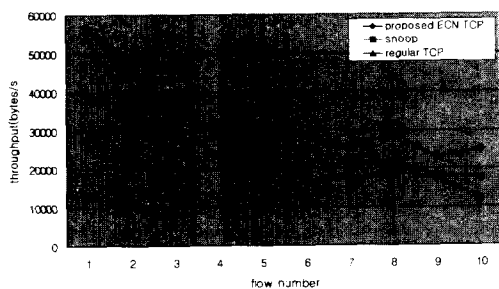


그림 8 플로우 수에 따른 처리량

<그림 9>는 핸드오프를 고려하여 연결 단절(disconnection) 때의 성능을 실험한 결과이다. 본 실험은 1M 크기의 파일을 FTP를 이용하여 ①에서 ④까지 전송할 때 무선 링크 구간에서 연결 단절 기간 변화에 따른 총 전송완료 시간을 측정할 것이다. snoop이 핸드오프 시 손실된 TCP 패킷들에 대해 혼잡 윈도우를 줄이기 때문에 핸드오프를 지원하지 못한다는 것과 M-TCP가 I-TCP의 단점 즉 빈번한 연결 단절(frequent disconnection) 때는 연속적인 타임아웃(serial timeouts)에 걸려 처리량이 상당히 줄어든다는 문제를 향상시켰고 중간 노드의 작업 오버헤드를 줄인 것이란 점을 감안하여 본 실험에서는 기존의 TCP와 함께 M-TCP와 제안한

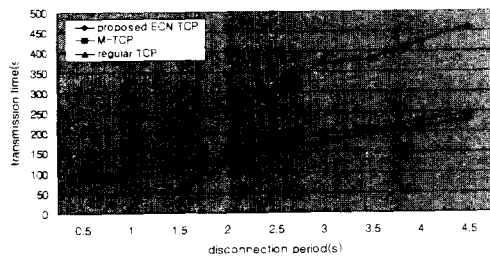


그림 9 연결단절 시간의 변화에 따른 총 전송완료 시간

TCP의 성능을 실험하였다. 기존의 일반적인 TCP가 연결 단절로 인해 발생한 패킷 손실들에 대해 무조건 슬로우 스타트를 시작할 뿐만 아니라 연속적인 타임아웃의 발생으로 인해 제안된 TCP와 M-TCP 보다 전송 완료 시간이 상대적으로 높게 나타났다. 그리고 지역적 재전송 기법을 사용하는 M-TCP 보다 제안한 TCP의 성능이 우수한 것으로 나타났다.

3.2 제안한 기법과 기존의 기법의 성능 비교 분석

[표 1]은 제안한 기법과 기존의 기법들과의 성능을 비교 분석한 것이다.

표 1 제안한 기법과 기존의 기법의 성능 분석

	snoop	I-TCP	M-TCP	제안한 TCP
중간 노드의 지역적인 재전송 기법 (local retransmission)	사용	사용	사용	사용하지 않음
종단간 의미 유지 (end-to-end semantics)	○	×	○	○
핸드오프	×	△	○	○
비트 에러	○	○	○	○
유선구간의 혼잡제어	×	×	×	○

(○ : 지원함, △ : 지원 못하는 경우가 있음, × : 지원하지 않음)

중간 노드의 지역적인 재전송 기법이란 송신측 FH에서 수신측 MH로 전송한 패킷들이 BS 또는 SH와 같은 중간 노드를 통과할 때 패킷의 복사본들을 중간 노드에 저장하였다가 무선 링크 상에서 TCP 패킷들이 손실되었을 시 손실된 패킷들을 중간 노드에서 감지하여 지역적으로 재전송하는 기법이다. 이러한 지역적인 재전송 기법은 현재 중간 노드를 통과하고 있는 모든 사용자 연결 각각에 대한 무선 링크 상에서의 TCP 패킷 손실의 구분 및 제어를 중간 노드에서 모두 해주어야 하기 때문에 중간 노드의 오버헤드가 상당히 크다. 또한 핸드오프 시 MH로부터 중간 노드로 ACK가 오지 않은 패킷들의 복사본 및 snoop, I-TCP, M-TCP 각 프로토콜의 상태 정보(state information)까지도 새로운 셀의 중간 노드로 복사해야 한다. 이 때 새로운 셀의 중간 노드로의 복사 작업은 상당히 느리게 진행되고, 이미 MH로 전송한 패킷들에 대한 손실 유무를 감지하여 손실된 패킷들이 있을 때 손실된 패킷들에 대한 재전송을 위해 새로운 중간 노드로의 복사 작업이 완전히 끝날 때까지 핸드오프가 완료되더라도 MH로의 패킷 전송은 중지되므로 전송 지연시간이 증가된다. 이러한 문제들은 셀에 접속해 있는 또는 셀간을 이동하는 사용자의 수가 많을

때 더욱 심각해진다[4, 5, 6].

종단간 의미 유지란 송신측에서 전송한 패킷들에 대한 ACK를 반대 끝단에 있는 송신측에서 받아야 한다는 TCP의 전통적인 규약을 말한다. 그러나 I-TCP는 FH에서 전송한 패킷에 대해 MH에서 BS로 ACK가 도착하기 전에 BS에서 FH로부터 받은 패킷에 대한 ACK를 직접 생성하여 FH로 전송하는 메커니즘을 사용하기 때문에 종단간 의미를 유지하지 못한다[4, 5]. 따라서 I-TCP의 경우에는 MH가 패킷을 받기 전에 무선구간의 연결이 끊겨 패킷들이 상실되었다더라도 FH는 BS로부터 생성된 ACK를 받았기 때문에 패킷이 안전하게 수신측 MH에 도착한 것으로 착각할 우려가 있다[4, 5].

핸드오프 지원에 있어서 snoop 기법은 핸드오프로 인한 패킷 손실에 대해 혼잡으로 인한 패킷 손실로 판단하여 송신측 FH가 혼잡 윈도우를 줄이는 등의 혼잡 제어 메커니즘을 호출하기 때문에 처리량이 저하되는 것을 막지 못하는 문제가 있다. I-TCP는 핸드오프가 빈번할 때 즉 빈번한 연결 단절이 일어날 때는 연속적인 타임아웃을 발생시키므로 핸드오프를 지원하지 못하는 경우가 발생한다[5, 6, 10].

비트 에러 지원에 있어서는 네 가지 기법 모두가 무선 구간에서의 비트 에러에 의한 패킷 손실을 구분할 수 있으므로 비트 에러에 의한 패킷 손실을 유선 구간에서의 혼잡으로 인한 패킷 손실로 판단하는 것을 방지할 수 있다.

유선구간의 혼잡 제어에 있어서는 제안한 기법이 유선망에서 혼잡 제어 메커니즘으로 사용되던 ECN을 확장하여 사용한 것이므로 유선 구간에서의 혼잡을 제어할 수 있지만 snoop, I-TCP, M-TCP는 혼잡을 제어하는 기능이 없다.

4. 결론

현재 TCP는 어떠한 수정 없이 무선망에 적용되고 있어 비트 에러나 핸드오프에 의한 패킷 손실이 발생할 때에도 이것을 유선구간에서의 혼잡으로 인한 손실로 판단한다. 따라서 모든 패킷 손실에 대해 혼잡 윈도우를 줄이는 등의 혼잡 제어 메커니즘을 호출하기 때문에 종단간 처리량이 현저하게 떨어진다는 문제점이 있다.

이러한 문제를 해결하기 위해 본 논문에서는 기존의 ECN을 무선 네트워크 상황에 맞도록 수정하여 송신측으로 하여금 패킷 손실의 원인이 혼잡에 의한 것인지, 비트 에러에 의한 것인지 또는 핸드오프 발생에 의한 것인지를 판단하는 기법을 제시하였다. 이것은 TCP가 ECN이 가능하다고 가정 하에 ECN 연결 설정단계에서 기존

의 설정단계를 수정하여 송신측으로 하여금 연결이 유선 연결인지 무선 연결인지를 구분하도록 한 것으로 무선 연결일 때는 송신측으로 하여금 혼잡으로 인한 패킷 손실의 경우만 혼잡 제어 메커니즘을 호출하는 기법이다. 본 기법은 무선 구간에서 TCP 성능을 제어하기 위해 연결 분할 방식을 따르는 기존의 snoop, I-TCP, M-TCP와는 달리 무선 구간에서의 비트 에러 또는 핸드오프로 인해 손실된 패킷들에 대해 BS와 같은 중간 노드에서 지역적으로 재전송하는 메커니즘을 사용하지 않고 송신측에서 직접 재전송하는 기법을 사용한다. 따라서 전송 지연시간을 불필요하게 증가시키지 않고 중간 노드를 통과하는 모든 사용자 연결 각각에 대한 무선 링크 상의 TCP 패킷 손실 구분 및 처리를 중간 노드에서 하지 않아도 되며 TCP 프로토콜 고유의 규약인 송신측과 수신측 양 끝단의 종단간 의미도 유지한다. 이러한 특성으로 인해 제안한 기법은 실시간 혹은 대화성 애플리케이션이 많이 사용되는 네트워크에 더욱 효율적이다. 하지만 이 방안이 적용되기 위해서는 송신측, 수신측, 라우터 모두가 ECN이 가능해야 한다는 단점이 있다.

참고 문헌

- [1] 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, Jun. 1995.
- [2] Aldar C. F. Chan, Danny H. K. Tsang, Sanjay Gupta, "TCP(Transmission Control Protocol) over Wireless Links," *IEEE In'l Conference*, 1997.
- [3] Rohit Ranami, Abhay Karandikar, "Explicit Congestion Notification (ECN) in TCP over Wireless Network," *IEEE In'l Conference*, pp.495-499, 2000.
- [4] Ajay Bakre, B. R. Badrinath, "I-TCP: Indirect TCP for Mobile Hosts," *Proceedings of 15th In'l Conference on Distributed Computing Systems*, Vancouver, Canada, pp136-143, May, 1995.
- [5] Kevin Brown, Suresh Singh, "M-TCP: TCP for Mobile Cellular Networks," *ACM Computer Communication Review(CCR)*, vol. 27, no. 5, pp.19-43. 1997.
- [6] Nachiket Deshpande, "TCP Extensions for Wireless Networks," July 2000. -www.cis.ohio-state.edu/~jain/cis788-99/tcp_wireless/index.html.
- [7] K. K. Ramakrishnan, Sally Floyd, D. Black. "The Addition of Explicit Congestion Notification(ECN) to IP," *IETF INTERNET DRAFT draft-ietf-*

tsvwg-ecn-03.txt.

- [8] Mark Handley, Curtis Villamizar, "ECN archive message 00009," IETF INTERNET DRAFT draft-jhsbns-ecn-00.txt.
- [9] H. Krishnan, "Analyzing Explicit Congestion Notification(ECN) Benefits for TCP," M.S. thesis, UCLA, 1998.
- [10] Jochen Schiller, "Mobile Communication," Addison-Wesley, 2000.
- [11] W. Stevens, "TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithm," RFC 2001, January 1997.
- [12] W. R. Stevens, "TCP/IP Illustrated, Vol. 1," Addison-Wesley Publishing Company, Nov. 1994.
- [13] William, Stallings. "High-speed networks TCP/IP and design principles," Prentice Hall, 1998.
- [14] V. Jacobson, "Congestion avoidance and control," in Proc. ACM SIGCOMM'88, Aug. 1988.
- [15] Sally Floyd, "TCP and Explicit Congestion Notification," Computer Communication Review, vol. 24, no. 5, pp.10-23, Oct 1994.
- [16] F. PENG, S. CHENG, J. MA, "An Effective way to improve TCP performance in Wireless/mobile networks," IEEE, pp. 435-441, 2000.



윤 여 훈

2000년 동국대학교 전산통계학과 졸업.
2000년 ~ 현재 고려대학교 컴퓨터학과 석사과정 재학중. 관심분야는 TCP, 무선, 라우팅 프로토콜, 유닉스 시스템



김 태 윤

1981년 고려대학교 산업공학과 학사.
1983년 미국 Wayne State University 전산과학과 석사. 1987년 미국 Auburn University 전산과학과 박사. 1988년 ~ 현재 고려대학교 컴퓨터학과 교수. 관심 분야는 전자상거래, 컴퓨터 네트워크, EDI, 이동통신, 멀티미디어 등