

전송오류가 큰 이동통신 환경에서 TCP 성능 개선에 관한 연구

정회원 장재신*

A study on improving TCP performance in mobile communication systems with a high transmission error

Jae-Shin Jang* *Regular Member*

요 약

기존 TCP 프로토콜은 전송오류가 거의 발생하지 않는 유선망에서 효율적으로 동작하도록 설계된 프로토콜이었다. 그러나 무선망에서는 경로손실, 페이딩, 간섭 등의 특징에 의해 전송오류 발생률이 유선망에 비해 훨씬 높다. 이런 환경에서 TCP의 성능저하를 방지하기 위한 연구가 많이 진행되었는데 그 중에서 링크계층 접근방법으로 SNOOP 프로토콜이 많이 알려져 있다. 그러나 기존 SNOOP 프로토콜이 핸드오프가 발생하는 상황에서는 불필요한 패킷 재전송을 수행할 가능성이 있다. 따라서 본 논문에서는 핸드오프가 자주 발생하는 이동통신 환경에서 불필요한 패킷 재전송을 방지하고 TCP의 성능을 향상시키기 위한 방법을 제안하고 성능이 개선됨을 모의실험을 통해서 제시하였다.

Key Words : M-SNOOP, TCP, Duplicate ACK, Handoff, Mobile communications, MH

ABSTRACT

Conventional TCPs are designed to work efficiently on wired network where any transmission errors rarely take place. However, the probability of transmission error in a wireless network is much higher than in a wired network, due to pass loss, multipath fading, and many kinds of interference. There were many researches on preventing the degradation of TCP performance in these wireless networks with bad channel condition. One of these researches is the SNOOP protocol which is link-layer solution for achieving high throughput of TCP. However, this SNOOP protocol is apt to retransmit some TCP segments unnecessarily, which could cause some increased traffic loads in a wireless network. In this paper, we propose a new algorithm to prevent the unnecessary retransmission of TCP segments and to achieve increased performance of TCP.

* 인제대학교 공과대학 전자정보통신공학부 (icjoseph@inje.ac.kr),

논문번호 : 030172-0424, 접수일자 : 2003년 4월 22일

※ 본 논문은 2002년도 인제대학교 학술연구조성비 보조에 의한 것임 (This work was supported by the 2002 Inje University research grant.)

I. 서론

이동통신의 발달은 급속히 이루어지고 있다. 동시식 3세대 이동통신망은 상용서비스를 시작한지 오래되었고, 비동시식 3세대 이동통신망도 부분적으로 상용서비스를 벌써 시작한 상태이다. 대부분의 경우 통신은 음성서비스에서 출발하였지만 인터넷의 확산으로 데이터 통신, 특히 인터넷을 통한 데이터 통신을 고려하지 않고서는 통신이 이제 무의미해졌다. 그래서 이동통신에서도 인터넷을 통한 데이터 통신에 주력한지 오래되었으며, 국내에서도 서비스 공급업체들이 다양한 데이터 서비스를 시행하고 있다.

한편 인터넷은 TCP/IP 프로토콜이 중심이며, TCP 프로토콜처럼 도입된 이후 오랫동안 생존하면서 각광을 받고 있는 프로토콜도 드물다. TCP 프로토콜은 유선망과 같이 전송오류가 거의 발생하지 않는 환경(예를 들면 BER이 10^{-10} 이하)에서 효율적으로 동작하도록 설계된 프로토콜이다. 그러나 이동통신 환경은 무선링크를 통해서 단말기와 기지국이 통신하는 구조이며, 통화 중에 단말기가 이동하기 때문에 기지국 사이에 핸드오프가 발생하게 된다. 무선링크의 채널 상태는 변화무쌍하여 단말기가 정지해 있더라도 수신신호의 크기가 계속 변화하며 더구나 단말기가 사용 중에 이동하는 상황에서는 매우 다이내믹한 특성을 가지고 있다. 따라서 이동통신 망에서는 경로손실, 다중경로 페이딩, 신호간섭 등으로 인해 패킷전송 오류가 클 뿐만 아니라(예를 들면 BER이 10^{-3} 정도), 오류가 발생하는 형태도 랜덤오류가 아닌 군집오류(Burst error) 성격이 강하다. 즉 핸드오프 과정이나 단말기가 전파음영지역에 접근할 경우에는 수백 msec에서 수 초 동안 통신이 제대로 이루어지지 않을 수도 있다. 이 동안에 전송된 패킷은 모두 전송오류가 되기 때문에 유선망과 달리 군집오류가 많이 발생하게 된다. 데이터 통신에서 전송오류는 링크 계층과 트랜스포트 계층에서 재전송 기법으로 해결하고 있다. 그래서 전송오류가 상대적으로 큰 무선링크에서 TCP 프로토콜의 성능을 개선하기 위한 연구가 많이 있었다. 본 논문에서는 이들 연구동향과 맥락을 같이하며 핸드오프가 발생하는 이동통신 환경에서 무선 TCP의 성능을 향상시킬 수 있는 방법을 제안하고, 제안된 알고리즘의 성능결과를 모의실험을 통해서 제시한다.

제 2장에서는 무선 TCP 프로토콜 성능 개선을 위한 각종 연구동향에 대해서 살펴보고, 기존 방법 중 SNOOP 방식이 가지고 있는 문제점을 제 3장에서 기술한다. 제 4장에서 SNOOP 방식의 문제점을 해결한 새로운 방식인 M-SNOOP 알고리즘을 소개하며, 5장에서 모의실험을 통해서 제안된 방법의 성능 결과를 제시한 뒤, 6장에서 결론을 맺었다.

II. 관련 연구동향

TCP 프로토콜은 전송오류가 거의 발생하지 않아 신뢰도가 높은 유선망에 적합하도록 설계된 트랜스포트 프로토콜이다. Slow start 기능과 Fast retransmit 기능을 가진 Tahoe TCP가 1988년에 소개되었고^[1], Fast recovery 기능이 추가된 Reno TCP가 1990년에 소개되었다^[2-3]. Reno TCP는 윈도우 내에 1개의 세그먼트 전송오류가 있을 경우에는 효율적으로 동작한다. 그러나 윈도우 내에 2개 이상의 오류가 존재할 경우에는 오류를 복구하는데 시간이 오래 소요되는 문제점을 갖는다. 그래서 이 문제를 해결하기 위해 NewReno TCP^[4]와 Vegas TCP^[5]가 소개되었다. 윈도우 내에 두 군데 이상의 위치에서 전송오류가 발생한 경우, TCP 송신 측이 전송오류가 발생한 세그먼트를 선별적으로 재전송하도록 하기 위해 TCP 헤더 내의 SACK 옵션을 사용하여 동작하는 SACK TCP^[6]가 표준으로 채택되었다. 다양한 버전의 TCP의 성능을 비교하는 연구도 많이 수행되었다. 그 중에서 다양한 패킷 손실 상황 하에 모의실험을 통해서 Tahoe, Reno, NewReno, 그리고 SACK TCP의 성능을 비교한 연구가 있었다^[7]. 이 연구결과에서는 한 윈도우 내에서 다수 개의 패킷 손실이 발생할 경우에는 SACK TCP가 성능이 가장 우수함을 보였다.

일반적으로 무선통신이 유선통신에 비해서 갖는 특징과 데이터 통신관련 3세대 이동통신 시스템이 갖는 특징들을 열거하면 다음과 같다.^[8-10]

- 대역폭이 제한되어 있다.
- 양방향 전파시간이 길다.
- 핸드오프나 열악한 무선링크 환경 때문에 패킷 손실률이 높다.
- 단말기의 이동성이 제동된다.
- Flow의 길이가 짧아 대부분 TCP의 Slow start 구간에 Flow가 종료된다.

- 배터리 때문에 낮은 소비전력이 요구된다.
- 오류제어 기법과 인터리빙 기법들의 추가로 Latency가 증가되었다.
- 데이터 전송속도의 증가로 LFN (Long fat network)의 특징을 가지게 되었다.
 - 순방향과 역방향 데이터 전송이 비대칭이다.

이런 특징을 지닌 무선링크를 가진 무선 랜이나 이동통신 시스템처럼 단말기가 통화 중에 이동할 수 있는 환경에서는 전송오류 확률이 높고 신뢰도가 유선망에 비해서 많이 떨어지기 때문에 기존 TCP 프로토콜을 수정 없이 그대로 적용할 경우에는 Throughput이 크게 떨어지는 문제가 발생한다. 왜냐하면 기존 TCP에서는 무선링크에서 핸드오프나 전파환경이 열악하여 발생한 패킷 전송오류들 망 내에서 Congestion이 발생했다고 판단하고 송신 측에서 전송속도를 줄이거나 또는, 버스트 오류 시에 재전송 타이머가 종료되면 Slow start 단계로 천이하여 전송속도가 떨어지는 현상이 발생하기 때문이다. 무선링크에서 TCP 성능향상 방법에 대해서는 많은 연구가 진행되었으며 각 접근 방법들에 대한 비교분석연구도 많이 진행되었다^[10-11]. 이미 진행되었던 연구 결과들을 정리해보면 다음과 같이 몇 개의 분야로 대별할 수 있다.

1. 연결 분할 프로토콜을 이용한 접근 방법 [12-15]

유선성이 공존하는 환경에서 연결 분할 프로토콜의 필요성을 참고본헌^[12]에서 언급하였고, 이 분야에 대한 선구적인 연구결과는 바로 Indirect-TCP (I-TCP) 프로토콜^[13]이다. 이 방법은 단단한 TCP 연결구간을 FH (Fixed Host)와 기지국 (BS : Base Station) 사이의 정규 TCP 연결구간 (Regular TCP connection)과 기지국과 MH (Mobile Host) 사이의 무선 TCP 연결구간 (Wireless TCP connection)으로 나눈다. FH에서 전송한 TCP 세그먼트에 대한 응답은 BS에 있는 유선 TCP 구간의 수신 TCP 프로토콜이 수행하며, 무선구간에서 발생하는 문제점들을 유선구간 TCP에게 감출 수 있다는 장점이 있다. 하지만 FH에서 전송한 세그먼트가 최종 목적지인 MH에게 도착되기 전에 이 세그먼트에 대해 BS에서 응답한 ACK이 FH에 도착할 수도 있어 End-to-end semantics가 깨어진다는 치명적인 단점을 갖는다. 연결 분할 프로토콜 접근방식을 이용한

연구동향으로는 I-TCP 외에도 WTCP^[14], M-TCP^[15] 등이 제안되어 있다.

2. 링크계층 접근 방법 [16-18]

링크계층이 물리계층의 특성을 잘 알고 있고 물리계층을 가장 잘 제어할 수 있다는 특징을 가지고 있으며, 속도가 빠르다는 장점을 이용한 접근방법이다. 이 방법은 ARQ 기법을 기본으로 활용하며 상위 계층의 TCP와 연계하여 동작하는데, 독자적으로 동작하느냐에 따라서 TCP-aware LL 방식과 TCP-unaware LL 방식으로 구분한다. TCP-aware LL 방식으로 대표적인 기법이 SNOOP^[16] 프로토콜이다. SNOOP 프로토콜은 BS에 SNOOP 모듈을 설치하여 MH (Mobile Host)로 전달되는 패킷들을 캐쉬에 저장하였다가 MH로부터 응답을 받지 못하면 자체적으로 재전송을 수행하고, Duplicate ACK 들을 필터링하는 기능을 가지고 있다. TCP-unaware LL 방식으로는 DDA^[17]과 TULIP^[18] 등이 있다. 한편 링크계층 접근방법에서는 패킷에 대한 재전송이 TCP와 링크계층에 중복해서 발생할 수 있으므로 두 계층의 프로토콜들이 긴밀하게 연계되어 동작해야만 부초화를 방지할 수 있다.

3. 기존 TCP 프로토콜 수정에 의한 접근 방법 [4-6, 19-23]

무선링크의 가장 큰 특징은 군집오류가 발생하는 성질이다. 그래서 군집오류에 강한 특성을 갖도록 기존 TCP를 수정하는 방법에 대한 연구도 많이 진행되었다. 대표적인 것이 NewReno^[4], Vegas TCP^[5], SACK TCP^[6]들로서 이 범주에 속하는 접근 방법의 단점은 상용으로 운영하려면 기존 유선망 TCP 프로토콜이 수정된 새로운 프로토콜로 모두 교체되어야 한다는 것이다. 참고문헌^[21]은 WAP용으로 제안된 TCP 프로토콜로 기존 TCP를 수정했다기 보다는 RFC를 통해서 제안되어 있는 표준들의 기능 중에서 무선링크 전송에서 장점을 가질 수 있는 기능들을 조합해서 만든 것이다. 참고문헌^[22]은 핸드오프와 같은 특이상황에서 발생한 세그먼트 전송오류를 재전송 타이머가 종료될 때까지 기다리지 않고 MH에서 Duplicate ACK 3개를 빨리 전송함으로써 전송오류에 빨리 대처할 수 있는 방법이다.

4. 새로운 TCP 프로토콜 개발에 접근 방법^[24]

무선링크에 적합한 전혀 새로운 무선 TCP 프로토콜을 개발하려는 접근 방법으로 대표적인 것이 WTCP^[24]이다. 연결관리와 흐름제어는 기존 TCP와 유사하지만 수신 측에서 비율에 근거하여 전송 속도를 제어하는 방법(Rate-based transmission)을 채택하였고, 오류제어를 위해 SACK과 probe를 사용하며 재전송 타이머를 사용하지 않았다. 그러나 이 접근 방법은 기존 TCP와 호환이 되지 않기 때문에 운용되기 위해서는 기존의 유선 TCP가 WTCP로 교체되어야 한다는 단점을 갖고 있다.

III. 기존 SNOOP 방법의 문제점

링크계층 해법으로 제안된 방법 중의 하나인 SNOOP 프로토콜^[16]은 그림 1처럼 BS (또는 BSC)에 SNOOP 모듈을 설치하여 운영하며 무선링크 상에서의 패킷 손실에 의해 불필요한 TCP의 혼잡제어가 발생하지 않게 하고, TCP의 단대단 전송 의미를 유지하면서 FH의 TCP를 수정하지 않고 운영할 수 있게 하는 접근방법이다. SNOOP 모듈에서는 데이터 세그먼트와 ACK 세그먼트의 흐름을 관찰하며, FH에서 MH로 전송되는 데이터 세그먼트를 캐쉬에 저장한다. 무선링크 상에서 패킷 손실에 의해 MH로부터 Duplicate ACK이 도착하면 (동일한 ACK을 2회 수신하면) 캐쉬에 저장되어 있는 데이터 세그먼트를 재전송하며, 해당 Duplicate ACK은 필터링되어 FH로 전달하지 않는다. SNOOP 모듈에서 패킷을 재전송하면 Van Jacobson 방식으로 계산된 RTT(Round trip time) 값을 기반으로 재전송 타이머를 운용하며 재전송 타이머가 종료되면 재전송을 수행한다. Persist 타이머는 TCP 송신 측과 수신 측으로부터 200 msec 동안 아무런 동작이 없을 때 종료되며, 캐쉬에 저장된 세그먼트가 최초 200 msec 동안 ACK을 받지 못했을 경우에는 Persist 타이머 동작에 의해 해당 세그먼트를 재전송하고 난 후 재전송 타이머를 활성화한다. 한편 MW-TCP 알고리즘^[25]에서는 MW-TCP 모듈에서 MH로부터 전송된 ACK을 수신했을 경우 해당 윈도우 내에 아직 응답을 받지 못한 세그먼트가 캐쉬 내에 존재하면 Duplicate ACK을 기다리지 않고 해당 세그먼트를 곧바로 전송하게 하여 성능향상을 시도했다. 200 msec로 고정된 Persist timer는 MW-TCP 모듈과 MH 사이의 RTT 값을 제대로 만

영하지 못하고 있다는 단점 때문에 MW-TCP 모듈에서 MH로 전송한 모든 패킷에 대해 Van Jacobson 방식을 기반의 재전송 타이머를 운용하는 것으로 수정하였다.

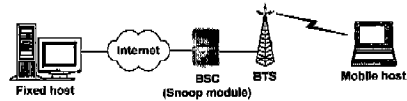


그림 1. SNOOP 모듈을 이용한 성능개선 방안.

그러나 MW-TCP 방식이 무선링크에서의 패킷 손실에 대해서 빠르게 대응할 수 있다는 장점은 가지고 있지만, 이로 인해서 불필요한 재전송을 유발할 가능성이 매우 크다. 인터넷에서 네트워크 계층 프로토콜인 IP 프로토콜은 순서에 맞게 전송하는 것을 보장하지 못하기 때문에 SNOOP 모듈에 도착하는 IP 패킷의 순서가 뒤바뀔 가능성은 항상 존재하며, 더구나 이동통신에서 단말기가 핸드오프 시에는 (Inter-SNOOP 핸드오프나, Mobile IPv6에서 다루는 Fast 핸드오프^[26] 등) 일부 IP 패킷의 전송 순서가 뒤바뀌어 SNOOP 모듈에 도착할 수 있다.

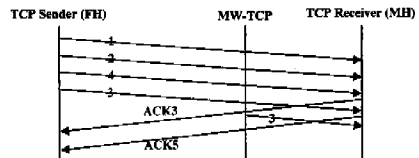


그림 2. MW-TCP 방식에서 불필요한 재전송 예.

예를 들어 그림 2와 같이 4번 세그먼트가 3번 세그먼트보다 먼저 MW-TCP 모듈에 도착한 경우를 살펴보자. 나중에 도착한 3번 세그먼트를 MH로 전송하고 응답을 기다리고 있는 상태이며 무선링크에서 전송오류는 발생하지 않았다. 그러나 순서가 뒤바뀌어 도착한 4번 세그먼트 때문에 3번 세그먼트를 전송한 직후에 ACK 3을 수신하게 된다. 여기서 ACK 3의 의미는 2번 세그먼트까지는 순서에 맞게 제대로 수신했으며, 3번 세그먼트를 수신할 차례라는 것을 의미한다. 3번 세그먼트를 이미 전송했던 MW-TCP는 해당 세그먼트에 전송오류가 생겼다고 판단하고 3번 세그먼트를 불필요하게 재전송한다.

한편 SNOOP의 경우에 있어서도 유사한 경우가

발생할 수 있는데 적절한 예는 그림 3과 같이 3번 세그먼트가 4, 5번 세그먼트보다 늦게 SNOOP 모듈에 도착했을 경우이다. SNOOP 모듈이 늦게 도착한 3번 세그먼트를 MH로 전송했고, 전송 오류는 발생하지 않은 상황에서 응답을 기다리고 있다. 그러나 순서가 맞지 않는 4번과 5번 세그먼트 때문에 SNOOP 모듈은 Duplicate ACK(ACK 3)을 수신하게 된다. 기존 SNOOP 모듈의 경우는 Duplicate ACK를 수신하는 즉시 3번 세그먼트를 불필요하게 재전송하거나, TCP 송신 측에서 재전송한 세그먼트로 판단한 경우에는(즉, Sender transmitted) Duplicate ACK이 FH로 전달되어 Fast retransmit에 의해 TCP 송신 측에서 재전송이 발생할 수도 있다.

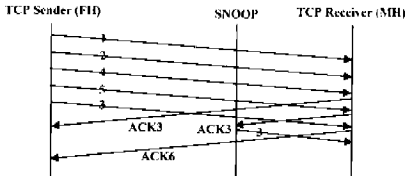


그림 3. SNOOP 방식에서 불필요한 재전송 예.

불필요한 재전송이 이동통신망에 미치는 영향을 무시할 수 없다. 동기식 CDMA 이동통신망에서는 TCP 세그먼트에 20바이트의 IP 헤더를 추가한 뒤, PPP 계층에서 헤더를 덧붙여서 RLP(Radio link protocol) 계층으로 전달한다. RLP 계층에서는 필요에 따라 다수 개의 패킷으로 세그먼트이전을 수행하고 오류제어 알고리즘에 따라 재전송 동작을 통해서 MH로 전달한다. 이런 과정을 고려해볼 때 불필요한 TCP 세그먼트의 재전송이 무선링크와 이동통신 시스템에 미치는 영향을 무시할 수는 없다. 따라서 불필요한 재전송을 줄이는 것이 제한된 대역폭을 가진 무선링크의 부담을 줄이는데 아주 중요한 고려사항이 된다. 또한 유선망 부분에서 IP 프로토콜에 의해 발생한 패킷 순서 뒤바뀔 문제는 종단간의 TCP가 해결하는 것이 옳은 방법이지만, 무선망의 특성에 의해서 발생한 IP 패킷의 전송 순서 뒤바뀔 문제는 (예를 들면 핸드오프에 의한 경우) 무선망 부분에서 해결하는 것이 타당한 해법이라고 생각된다. 따라서 본 논문에서는 핸드오프에 의해서 발생한 IP 패킷의 전송 순서 뒤바뀔 문제를 해결하기 위한 방법을 제시하고, 이 방법을 본 논문에서는

M-SNOOP (Modified SNOOP)라고 명하겠다. 그리고 Persist 타이머를 사용하면 무선링크에서 패킷 손실이 발생했을 경우, 경우에 따라서는 초기에 재전송을 시도할 수도 있기 때문에 M-SNOOP 프로토콜에서는 SNOOP 프로토콜과 동일하게 Persist 타이머를 사용한다.

IV. 제안된 M-SNOOP 프로토콜 구조

기존 SNOOP 모듈과 MW-TCP가 불필요한 재전송을 야기했던 이유는 순서가 뒤바뀌어 도착한 세그먼트들이 Duplicate ACK을 발생시켰고 해당 데이터 세그먼트에 대한 ACK은 잠시 후에 도착할 예정이지만 이런 조건을 고려하지 않고 도착된 Duplicate ACK에 따라 처리하는 데서 비롯된다. M-SNOOP 프로토콜에서는 M-SNOOP 모듈 간의 핸드오프가 발생한 것을 인지할 수 있다고 가정한다. 제안된 M-SNOOP 모듈의 동작을 흐름도로 나타내어 설명하면 다음과 같다. 핸드오프 직후 FH로부터 데이터 세그먼트를 수신했을 경우는 기존 SNOOP 모듈의 동작과 차이가 거의 없다(그림 4 참조).

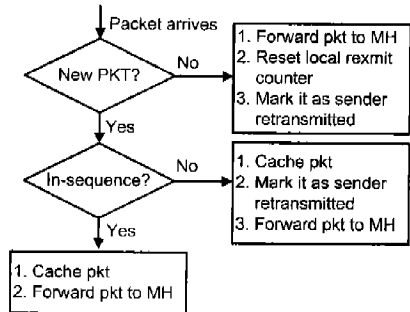


그림 4. 핸드오프 직후 TCP 세그먼트 수신 시.

한편 핸드오프 직후 MH로부터 ACK을 받았을 경우에는 (그림 5 참조) 수신한 ACK이 새로운 ACK인지 검사한다. 만약 새로운 ACK인 경우는 기존 SNOOP 기능에 부가하여 핸드오프 지연 타이머를 구동하며, 타이머는 d 동안 작동한다. 여기서 핸드오프 직후 핸드오프 지연 타이머를 구동하는 조건은 다음과 같다.

- 캐쉬 내에 해당 세그먼트가 빠져있는 경우, 즉 전송해야 할 세그먼트가 순서가 뒤바뀌어 아직 도착하지 않은 경우.
- 캐쉬 내에 해당 세그먼트가 저장되어 있지만 M-SNOOP 모듈에 순서가 뒤바뀌어 도착했던 경우.

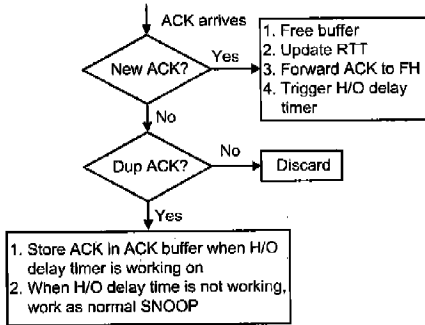


그림 5. 핸드오프 직후 ACK 세그먼트 수신 시.

그리고 핸드오프 지연 타이머가 동작하고 있는 중에 도착한 Duplicate ACK은 모두 ACK 버퍼에 저장하며, 새로운 ACK 패킷이 도착하면 핸드오프 지연 타이머를 중지하고 ACK 버퍼에 있는 모든 Duplicate ACK들을 폐기하며 새로운 ACK 패킷에 의해 캐쉬 내에 응답된 데이터 세그먼트들도 모두 폐기한다. 새로운 ACK이 아니고 Duplicate ACK도 아닌 경우는 Spurious ACK에 해당하므로 폐기한다. 그리고 핸드오프 지연 타이머가 동작하지 않을 경우에는 기존 SNOOP처럼 동작한다. 한편 핸드오프 지연 타이머가 종료되었을 경우는 해당 세그먼트에 대한 ACK이 시간 d 동안 도착하지 않은 경우이며 (그림 6 참조), 캐쉬 내에 손실된 세그먼트가 저장되어 있을 경우에는 해당 세그먼트를 재전송하고, ACK 버퍼에 저장되어 있던 Duplicate ACK은 모두 폐기한다. 그러나 캐쉬 내에 손실된 세그먼트가 저장되어 있지 않을 경우에는 ACK 버퍼에 저장된 모든 Duplicate ACK을 FH로 전송하고 ACK 버퍼를 모두 비운다. 이런 절차에 따라서 M-SNOOP는 Inter-SNOOP 핸드오프 시에 SNOOP 방식이나 MW-TCP 방식이 야기할 수 있는 불필요한 재전송 문제를 해결한다.

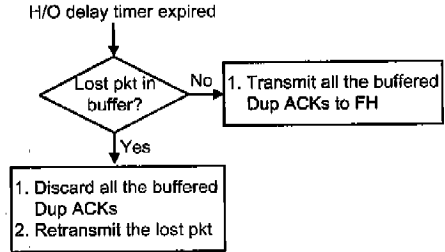


그림 6. 핸드오프 지연 타이머 종료 시.

요약하면 M-SNOOP 모듈 간의 핸드오프 직후 상태에서는 Duplicate ACK을 수신한 경우, 해당 TCP 세그먼트가 캐쉬에 저장되어 있지 않더라도 핸드오프로 인해 순서가 뒤바뀌어 늦게 도착할 수도 있기 때문에 MH로부터 수신한 Duplicate ACK을 곧바로 FH로 전송하지 않고 일정한 시간, d 동안 ACK 버퍼에 저장해놓고 대기한다. 만약 해당 세그먼트가 캐쉬 내에 있더라도 해당 세그먼트가 핸드오프에 의해 순서가 뒤바뀌어 나중에 도착했고, 늦게 MH로 전송된 경우이기 때문에 d 동안 정당한 ACK이 도착하기를 기다린다. 그러면 그림 3의 예제를 M-SNOOP 모듈에 적용하여 설명해보자. 그림 3의 SNOOP 모듈에서는 Duplicate ACK 3을 수신 후 캐쉬에 3번 세그먼트가 있는 것을 확인하고는 3번 세그먼트를 (불필요하게) 곧바로 재전송했지만, M-SNOOP 모듈간 핸드오프 직후라면 3번 세그먼트가 늦게 도착한 뒤, MH로 늦게 전송된 경우므로 Duplicate ACK 3을 수신했을 경우 캐쉬에 있는 3번 세그먼트를 재전송하지 않고 핸드오프 지연 타이머를 사용하여 시간 d 동안 정당한 ACK(ACK 6)을 기다린다. 이상의 과정은 그림 7에 나타나 있다. 즉 MH에서 4번 세그먼트를 수신한 후 이에 대한 응답으로 ACK 3을 전송한다. ACK 3을 수신한 M-SNOOP에서는 캐쉬에 3번 세그먼트가 빠져있다는 것을 확인하고, 핸드오프 직후라는 것을 인지하면 ACK 3을 FH로 전송한 뒤, 핸드오프 지연 타이머를 구동한다. 그 후 d 동안 기다리다가 ACK 6을 수신하면 ACK 6을 FH로 전송하고 핸드오프 지연 타이머를 중지시키며, ACK 버퍼에 저장된 Duplicate ACK들과 캐쉬에 저장된 3,4,5 번 세그먼트를 모두 폐기한다.

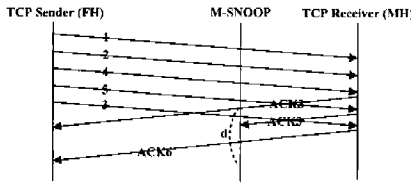


그림 7. 핸드오프 직후 M-SNOOP 방식의 동작 예.

V. 성능분석 결과

M-SNOOP 프로토콜에 대한 성능평가는 NS-2^[27]를 이용하여 모의실험을 수행하였으며 성능평가를 위한 망 구조는 그림 8과 같다. M-SNOOP 모듈의 성능평가에서 필요한 환경은 Inter-SNOOP 핸드오프 시에 발생하는 패킷의 전송 순서가 뒤바뀔 현상이며 이 현상을 모의실험에서 수행하기 위해서 그림 8의 B 노드에 패킷 전송순서 뒤바뀔을 야기하는 모듈을 만들어서 연결하였고 노드 D에 M-SNOOP 모듈을 연결하였다. 그림 8의 망 구조와 관련된 파라미터들은 표 1에 기술되어 있다.

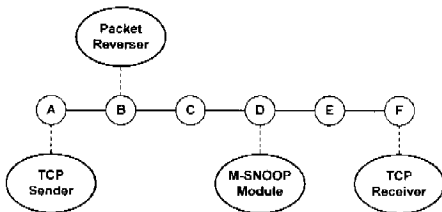


그림 8. 모의실험을 위한 망 구조

표 1. 망 구조에서 링크 특성.

링크 명	종류	링크속도	링크지연	버퍼관리	손실여부
A-B	전 이중	10Mbps	0 msec	Drop/Tail	<
B-C	LAN	11Mbps	2 msec		>
C-D	전 이중	10Mbps	0 msec	Drop/Tail	>
D-E	LAN	11Mbps	2 msec		O
E-F	전 이중	10Mbps	0 msec	Drop/Tail	>

M-SNOOP 모듈은 기존의 SNOOP 모듈 (~NS-2/TCP/snoop.cc) 을 수정하여 만들었고, 패킷

전송순서를 뒤바꿔게 하는 모듈도 SNOOP 모듈을 기초로 대폭 수정하여 만들었다. 그래서 B-C 링크가 D-E 링크와 같이 LAN으로 구성되어 있지만, 본 논문에서 제한한 방식을 모의실험에 의해 검증하는 과정에 별다른 영향은 없다. D-E 링크는 유선 LAN이지만, 무선링크를 모델링하기 위해 손실이 발생하는 링크로 구성하였으며 패킷 손실 확률 값을 변수로 전송오류를 발생시킨다. LAN 링크를 제외한 모든 링크는 전 이중 (Full duplex) 통신 링크이며, 편의상 링크에서의 전송지연시간은 무시하였다. A노드에(FH에 해당) TCP 송신 측이 연결되어 있고, F노드에(MH에 해당) TCP 수신 측이 연결되어 있다. NS-2의 기본 TCP인 Tahoe TCP를 사용하였고, 응용프로그램은 FTP를 사용하였으며, 전송하고자 하는 데이터가 항상 존재하는 모델을 사용하였다. 따라서 FTP 세션은 모의실험 시작부터 설정되어 10초 동안 끊임없이 데이터를 전송하는 모델이다. 패킷 순서를 뒤바꾸는 모듈은 해당 시간 () 이후 노드 B에 FH로부터 처음으로 도착하는 TCP 세그먼트 한 개를 ... 동안 지연시켰다가 전송한다. 한편 모의실험에서 사용된 시나리오는 다음과 같다.

- 시나리오 1 : 무 손실,
 $T_1 = 7.1[\text{sec}], T_2 = 0.01[\text{sec}], d = 0.2[\text{sec}]$
- 시나리오 2 : 손실 발생,
 $T_1 = 7.1[\text{sec}], T_2 = 0.01[\text{sec}], d = 0.2[\text{sec}]$

그림 9,10,11은 시나리오 1에 대한 결과로 그림 9는 노드 D에 기존 SNOOP 모듈을 사용하였을 경우의 예이다. NS-2를 이용한 모의실험 결과에서 ACK N의 의미는 N번 세그먼트까지를 순서에 맞게 정상적으로 수신했다는 의미로, TCP 헤더 내에 있는 ACK 시퀀스 번호와 조금 의미 차이가 있음을 상기하기 바란다. 본 성능평가에서는 6889번 세그먼트가 노드 B에서 패킷 순서 뒤바뀔 모듈에 의해 10 msec 만큼 지연되었다가 6897번 세그먼트 이후에 SNOOP 모듈로 전송된다. 한편 MH에서는 6888번 이후 곧바로 6890 ~ 6908번 세그먼트(그림 9에서 A) 부분에 해당)가 도착하기 때문에 Duplicate ACK (ACK 6888)을 8번 전송한다. SNOOP에 Duplicate ACK들이 도착한 시점에 캐시 내에 6889번 세그먼트가 아직 도착하지 않았거나, 해당 세그먼트가 나중에 도착했다라도 순서가 뒤바뀌어 도착했기 때문에 SNOOP모듈은 송신 측이 6889번 세그먼트를 재전송해서 다시 도착했다고 판단하고 표기

(Sender transmitted) 를 해 놓은 상태이기 때문에 Duplicate ACK들이 필터링되지 않고 곧바로 FH로 전송된다.

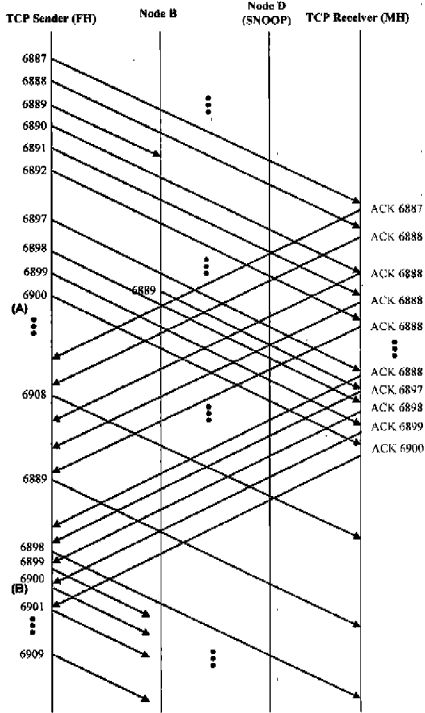


그림 9. SNOOP을 사용한 경우 동작 예.

Duplicate ACK을 세 번 수신한 TCP 송신 측은 Fast retransmit에 의해 6889번 세그먼트를 재전송하며 응답을 기다리던 중 (Tahoe TCP는 이 경우 Slow start를 시작함), 순서가 뒤바뀌어 늦게 MH에 도착했던 6889번에 대한 응답, ACK 6897을 수신한다. TCP 송신 측은 ACK 6897을 재전송한 6889번 세그먼트에 대한 응답으로 오인하고 6898번 세그먼트부터 Slow start 방식에 따라서 전송을 시작한다(그림9에서 (B)부분에 해당). 그 후 (B) 시점에서 재전송한 세그먼트들이 MH에 도착하면 6908번 세그먼트까지는 이미 받았으므로 Duplicate ACK (ACK 6908)을 계속 보낸다. FH에서 전송한 6909번 세그먼트가 SNOOP에 도착한 후(그림 10 참조) MH로부터 ACK 6908을 받은 SNOOP 모듈이 이

제는 6909번 세그먼트가 D-E 링크에서 손실되었다고 판단하고, 6909번 세그먼트를 재전송한다. 왜냐하면 기존 SNOOP 프로토콜은 MH로부터 Duplicate ACK을 받았을 경우 캐시 내에 해당 세그먼트가 있으면 무선헤링크에서 전송오류가 발생한 경우라고 판단하고 재전송을 하기 때문이다(그림 3 참조).

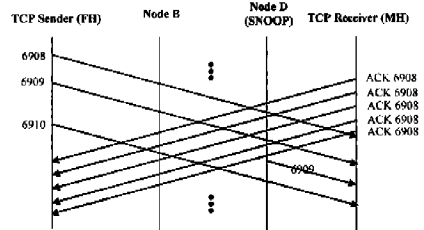


그림 10. SNOOP모듈에서 불필요한 재전송이 발생하는 경우.

그러나 제안된 방법인 M-SNOOP 모듈에서는 불필요한 재전송이 발생하지 않는다. 즉 핸드오프 직후에 ACK 6888번을 받은 M-SNOOP는 6889번 세그먼트가 캐시에 없는 것을 확인하고는 핸드오프 과정에서 순서가 뒤바뀌었다고 판단하고 핸드오프 지연 타이머를 구동한다. 타이머가 구동된 이후에 수신하는 Duplicate ACK 6888들은 모두 ACK 버퍼에 저장하며 FH로 전송하지 않는다. 6897번 세그먼트 이후에 6889번 세그먼트가 순서가 뒤바뀌어 도착하고, 이에 대한 응답으로 ACK 6897을 M-SNOOP가 수신하면 핸드오프 지연 타이머를 중단시키고, ACK 버퍼를 비운 뒤, ACK 6897을 FH로 전달한다. 따라서 SNOOP의 경우와 달리 6889번 세그먼트에 대한 재전송이 발생하지 않으며, 불필요한 Duplicate ACK(ACK 6888)들이 FH로 전달되지 않는다. 그리고 그림 10에서와 같이 SNOOP 모듈에서 6909번 세그먼트와 같은 불필요한 세그먼트 재전송도 발생하지 않는다. 이런 장점 때문에 전체 Throughput이 증가하고 불필요한 재전송이 줄어들어 이동통신망에서의 부담도 줄어든다. 시나리오 1에서 SNOOP를 사용한 경우 10초 동안 9588개의 세그먼트를 전송했지만, M-SNOOP에서는 9661개의 세그먼트를 전송했다. 물론 이 차이는 Inter-SNOOP 핸드오프 발생 빈도가 높을수록 커질 것으로 예상된다.

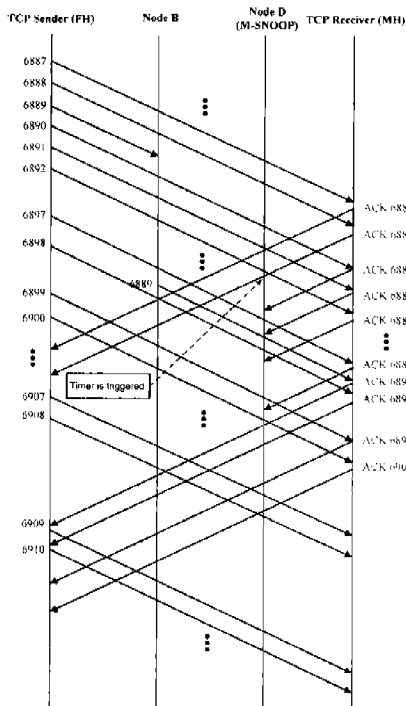


그림 11. M-SNOOP모듈을 사용한 경우의 동작 예.

표 2. 성공적으로 전송된 TCP 세그먼트의 개수 변화.

손실률 (%)	0	1	2	3	4	5
SNOOP	9588	8368	7357	6740	5410	4292
M-SNOOP	9661	8374	7547	6816	5410	4292

표 2는 시나리오 2에 대해서 D-E 링크의 전송 손실 값을 변화시키면서 FH에서 MH로 성공적으로 전송된 실제 TCP 세그먼트 개수를 구한 값이다. M-SNOOP 알고리즘이 SNOOP에 비해 성능이 뛰어난 것을 알 수 있고, D-E 링크에서 세그먼트 손실률이 4% 이상이 될 경우에는 Throughput의 차이가 거의 없음을 알 수 있다. 왜냐하면 이 범위에서는 무선링크에 해당하는 D-E 링크에서의 패킷 손실률이 시스템의 성능을 주도하고 있기 때문이다.

VI. 결론

본 논문에서는 무선 TCP의 성능을 개선하기 위한 방법 중에서 링크계층 해법으로 제시된

SNOOP^[16] 방식을 개선한 M-SNOOP 알고리즘을 제안하고 NS-2^[27]를 사용한 모의실험을 통해서 제안된 알고리즘이 기존 방식에 비해서 성능이 개선됨을 보였다. M-SNOOP 방식은 이동통신 환경 중 에서 중요한 개념 중의 하나인 핸드오프가 발생하였을 경우 FH에서 SNOOP로 도착하는 패킷의 전송 순서가 뒤바뀔 수 있는 상황에서 불필요한 재전송을 방지하기 위한 방법으로 제안하였다. 성능분석 결과를 통해서 M-SNOOP 방식이 불필요한 재전송 횟수를 많이 줄임을 확인하였으며, 제안된 방식은 기존의 SNOOP 방식과 동일하게 유선망에서의 변경 없이 기존 TCP와 연동하여 사용할 수 있다는 점이 또한 큰 장점이다. 본 논문에서는 모의 실험하는 과정에서 M-SNOOP 알고리즘을 검증하기 위해 무선링크 구간을 랜덤오류률 갖는 유선 랜 환경으로 구성하여 모의실험을 수행하였다. 그러나 이 구조가 M-SNOOP 알고리즘의 성능 결과에 미치는 영향은 거의 없으며 차후 연구에서는 본 논문의 결과를 기반으로 그림 8의 D-E 링크 또는 E-F 링크를 무선링크로 바꿔 단말의 이동성을 추가한 뒤 실제 환경에 근접한 모델로 변경하여 성능평가를 수행할 계획이다.

참고 문헌

- [1] V. Jacobson and M. J. Karels, "Congestion avoidance and control," Computer communication review, November 1988.
- [2] M. Allman, et al., "TCP congestion control," IETF RFC 2581, April 1999.
- [3] W. Richard Stevens, *TCP/IP Illustrated, Volume 1 : The protocols*, Addison Wesley, 1994.
- [4] S. Floyd and T. Henderson, "The NewReno modification to TCP's fast recovery algorithm," IETF RFC 2582, April 1999.
- [5] J. S. Ahn, et al., "Evaluation of TCP vegas : emulation and experiment," in *Proc. of SIGCOMM'95*, pp. 185-195, August 1995.
- [6] M. Mathis, et al., "TCP selective acknowledgment options," IETF RFC 2018, October 1996.
- [7] K. Fall and S. Floyd, "Simulation-based comparisons of Tahoe, Reno, and SACK TCP," *ACM computer communication review*, vol. 26, no. 3, July 1996.
- [8] H. Inamura, G. Montenegro, R. Ludwig, A.

Gurtov, and F. Khafizov, "TCP over second (2.5G) and third (3G) generation wireless networks," *IETF RFC 3481*, February 2003.

[9] G. Xylomenos, et al., "TCP performance issues over wireless links," *IEEE Commun. Mag.*, pp. 52-58, April 2001.

[10] K. Pentikousis, "TCP in wired-cum-wireless environments," *IEEE Communications Surveys*, pp. 2-14, Fourth quarter 2000.

[11] H. Balakrishnan, V. N. Padmanabhan, S. Seshan and R. H. Katz, "A comparison of mechanisms for improving TCP performance over wireless links," *IEEE/ACM Trans. on networking*, vol. 5, no. 6, pp. 756-769, December 1997.

[12] G. Montenegro, et al., "Long thin networks," *RFC 2757*, January 2000.

[13] A. V. Bakre and B. R. Badrinath, "Implementation and performance evaluation of indirect TCP," *IEEE Trans. on computers*, vol. 46, no. 3, pp. 260-278, March 1997.

[14] Karunaharan Ratnam and Ibrahim Matta, "WTCP : An efficient mechanism for improving TCP performance over wireless links," in *Proc. of 3rd IEEE Symposium on Computers and Communications (ISCC'99)*, 1999, Athens Greece.

[15] K. Brown and S. Sigh, "M-TCP : TCP for mobile cellular networks," *ACM computer communication review*, vol. 27, no. 5, October 1997.

[16] H. Balakrishnan, S. Seshan, and R. H. Katz, "Improving reliable transport and handoff performance in cellular wireless networks", *ACM Wireless Networks*, vol. 1, December 1995.

[17] N. H. Vaidya, et al., "Delayed duplicate acknowledgements: A TCP-unaware approach to improve performance of TCP over wireless," *Technical Report 99-003*, Computer science department, Texas A&M university, February 1999.

[18] C. Parsa, et al., "Improving TCP performance over wireless networks at the link layer," *Mobile Networks and Applications*, 1999.

[19] J. C. Hoe, "Improving the start-up behavior

of a congestion control scheme for TCP," in *Proc. of SIGCOMM'96*, August 1996.

[20] M. Mathis and J. Mahdavi, "Forward acknowledgment : Refining TCP congestion control," in *Proc. of SIGCOMM'96*, August 1996.

[21] WAP, "Wireless profiled TCP," *WAP forum*, <http://www.wapforum.org/>, March 2001.

[22] R. Caceres and Liviu Iftode, "Improving the performance of reliable transport protocols in mobile computing environments," *IEEE J. on selected areas in communications*, SAC-13, no. 5, June 1995.

[23] M. Allman, S. Floyd, and C. Partridge, "Increasing TCP's initial window," *IETF RFC 2414*, September 1998.

[24] Prasun Sinha, et al., "WTCP : A reliable transport protocol for wireless wide-area networks," in *Proc. of ACM Mobicom'99*, August 1999.

[25] 조준상, 최명환, "전송 오류율이 높은 무선환경에서의 TCP 성능저하 극복방안," *Telecommunications review*, 제10권 6호, pp. 1220-1232, 11-12 2000.

[26] R. Koodli, "Fast Handovers for Mobile IPv6," *IETF Mobile IP WG Internet draft*, draft-ietf-mobileip-fast-mipv6 -06.txt, March 1 2003.

[27] NS-2 network simulator, URL : <http://www.isi.edu/nsnam/ns/>.

장 재 신 (Jae-Shin Jang)

정회원



1990년 2월 : 동아대학교 전자공학과 졸업
 1992년 2월 : 한국과학기술원 전기및전자공학과 석사과정
 1998년 2월 : 한국과학기술원 전기및전자공학과 박사과정
 1997년 7월~2002년 2월 : (주)

삼성전자 네트웍사업부 책임연구원
 2002년 3월~현재 : 인제대학교 공과대학 전자정보통신공학부 전임강사

<주관심분야> 이동통신, 무선 랜, 무선인터넷, QoS control in wireless networks