

개선된 SNOOP 기법을 이용한 무선 TCP 성능향상 방안

(Wireless TCP Enhancement by Modifying SNOOP)

문 영 성 [†] 강 인 석 ^{**}
(Youngsong Mun) (Insuk Kang)

요 약 현재 가장 널리 쓰이는 수송계층 프로토콜인 TCP는 패킷 손실의 원인을 망의 혼잡 때문에 일어난다고 가정하고 있으므로 기존의 유선망과 고정 호스트로 이루어진 전통적인 네트워크에 적합하다. 그러나 무선 링크에서의 패킷 손실은 대부분 혼잡에 의해서가 아니라 높은 에러율과 핸드오프에 의해 발생하게 되므로 기존의 TCP를 그대로 사용하게 되면 불필요한 혼잡제어 메커니즘의 호출로 성능의 저하를 가져온다.

현재까지 무선환경에 적합한 TCP를 위한 많은 방안이 제시되고 있다. 이 가운데 SNOOP 기법은 무선 링크에서 패킷 손실이 생겼을 경우 송신측에서 재 전송하는 것이 아니라, 지역 재전송에 기반을 두고 있기 때문에 무선 링크의 손실로 인해 유선 링크가 영향을 받는 것이 아니므로 매우 적절한 해결책이다. 하지만 SNOOP은 캐쉬에 재전송할 데이터가 없으면 성능의 저하를 초래하는 것이 단점이다.

본 논문에서는 SNOOP의 단점을 보완하는 기법을 제안한다. 제안하는 기법은 패킷손실의 원인을 판별하여 불필요한 혼잡제어 메커니즘의 호출을 줄여 성능을 개선한다.

키워드 : 무선TCP, SNOOP, 혼잡제어

Abstract Reliable transport protocols such as TCP are tuned to perform well in traditional networks where packet losses occur mainly because of congestion. In a wireless network, however, packet losses will occur more often due to reasons such as the high bit error rate and the handoff rather than due to congestion. When using TCP over wireless network, TCP responds to losses due to the high bit error rate and the handoff by invoking congestion control and avoidance algorithms, resulting in the degraded end-to-end performance in the wireless network. There have been several schemes for improving TCP performance over wireless links. Among them, SNOOP is a very promising scheme because of the localized retransmission.

In this thesis, an efficient scheme is proposed by modifying SNOOP scheme. The invocation of congestion control mechanism is now minimized by knowing the cause of packet loss.

Key words : Wireless TCP, SNOOP, Congestion Control

1. 서 론

최근 휴대용 컴퓨터와 무선통신 기술의 발달로 이동 컴퓨팅 환경에 대한 관심이 날이 높아지고 있다. 특히, 무선 통신 이용자가 장소에 구애받지 않고 어느 때나 원하는 정보를 주고받을 수 있다는 것은 상당히 매

력적이다. 오늘날의 인터넷은 대부분이 유선망에 기반을 두고 있으며 이러한 유선망에 무선망을 도입한 유·무선 통합망의 구현이 현실화되고 있다. 유·무선 통합 망에서 우선적으로 직면하는 문제는 단말기의 이동성이며, 이에 따라 어드레싱과 라우팅 기능을 보장하여 단말기의 이동성에 대응할 수 있는 Mobile IP에 대한 연구가 최근 깊이 이루어져 왔다. 그러나, 이러한 네트워크 계층에 대한 문제 이외에 신뢰성 있는 연결을 보장하기 위한 TCP(Transmission Control Protocol)의 유효성에 대한 검증은 해결해야 할 과제로 남아 있다. 오늘날 통신환경에서 널리 사용되고 있는 TCP는 유선망과 고정 호스트(FH: Fixed Host)로 이루어진 전통적인 네트워크

· 본 연구는 한국 과학 재단 목적 기초 연구(RP1-2001-000-00362-0) 지원으로 수행되었음

[†] 종신회원 : 숭실대학교 컴퓨터학부 교수
mun@computing.ssu.ac.kr

^{**} 비 회 원 : 팜택인큐리텔 연구원
kang.inseok@curitel.com

논문접수 : 2003년 8월 12일

심사완료 : 2004년 10월 8일

크에 적합하며 유선망의 특성을 적용한 것이기 때문에, 무선 구간의 제한된 대역폭, 높은 지연, 산발적인 비트 에러, 일시적인 연결 두절 및 핸드오프 등과 같은 특징을 가진 무선망에 이를 그대로 적용하게 되면 오히려 불필요한 메커니즘의 호출로 인해 성능 저하를 가져온다.

이러한 성능 저하는 TCP가 유선망의 낮은 비트 에러율 때문에 통신상에서 발생하는 패킷 손실을 기본적으로 혼잡에 의한 것으로 생각하여 혼잡 제어 메커니즘으로 패킷 손실을 처리하기 때문에 발생한다. TCP 송신자는 패킷 손실을 발견하면 먼저 전송 윈도우 크기를 줄이고 손실된 패킷을 재전송한다. 또한 혼잡 제어나 회피 메커니즘을 초기화하고 재전송을 위한 타이머의 값을 증가시킨다[1,2]. 하지만 무선망에서의 패킷 손실은 대부분 혼잡에 의해서가 아니라 핸드오프 시 발생하는 이동 호스트(MH: Mobile Host)와 기지국(BS: Base Station)간의 연결 끊김(disconnection)과 오랜 지연 시간, 그리고 무선망 자체의 높은 비트 에러율에 의해서 발생한다. 그러므로 앞에서 언급한 TCP의 패킷 손실 회피 메커니즘을 그대로 무선망에 적용시키면 오히려 불필요한 함수(혼잡 제어나 회피 메커니즘)호출로 인해 처리율의 성능 저하를 가져오게 된다.

이동 컴퓨팅 환경을 지원하기 위해서는 망에서의 접속 지점을 수시로 바꾸는 MH의 위치를 찾아내고, 통신 중에도 위치 변화가 잦은 MH와의 연결을 유지해줘야 한다. 유선망 간의 통신 시에는 기존의 TCP에 큰 변화를 주지 않으면서 유선망과 무선망간의 통신 시에는 무선망의 특징을 고려하는 개선된 TCP를 구현하는 방법이 제안되었고, 현재 많은 연구가 진행중이다.

본 논문에서는 패킷손실이 유선링크에서 일어나는 것인지, 무선링크에서 일어나는 것인지를 판별하여 패킷손실이 유선링크에서 일어난 경우는 기존의 혼잡제어 메커니즘을 호출하여 재전송하고, 무선링크에서 일어난 경우는 혼잡제어 메커니즘을 호출하지 않고 재전송 하여 성능을 개선한다.

2. 관련 연구

무선망에서의 End-to-End throughput을 향상시키기 위한 여러 가지 방법이 각자 다른 방향으로 이루어지고 있다. 이런 방법들은 End-to-End 프로토콜, split-connection 프로토콜, 링크 계층 프로토콜의 세 가지로 크게 나누어 볼 수 있다[3]. End-to-End 프로토콜은 송신자가 무선망의 존재 여부를 알고 있는 프로토콜로, 패킷의 재전송 시 조잡한 타임아웃에 의존하지 않고 하나의 윈도우에 여러 개의 패킷 손실을 회복할 수 있도록 하는 selective acknowledgement 방법과 패킷 손실

의 원인이 혼잡에 의한 것이 아니라 다른 원인(무선망 특성 관련)으로 발생하였음을 알려서 불필요한 혼잡 제어 메커니즘의 호출을 방지하는 ELN(Explicit Loss Notification) 방법을 주로 이용한다[3]. Split-connection 프로토콜은 용어 자체에서 알 수 있듯이, 유선망과 무선망의 연결을 2개의 독립된 연결로 분리하고 각자에게 적합한 프로토콜을 적용하는 방법이다. 즉, FH와 BS간의 유선망 연결과 MH와 BS간의 무선망 연결로 분리한다. 링크 계층 프로토콜은 local reliability를 제공하는 프로토콜로써 TCP와 같은 네트워크 계층에 무선망과 관련된 손실을 숨김으로써 성능 향상을 가져오는 ARQ(Automatic Repeat reQuest)와 FEC(Forward Error Correction)의 결합을 이용한다.

2.1 제안된 기법들

2.1.1 End-to-End fast retransmission scheme

기존의 TCP에서는 핸드오프 시 패킷 손실이 있으면 네트워크의 상태가 좋아질 때까지 일정시간(재전송 타임아웃)동안 기다려야 한다. 하지만 패킷 손실의 원인이 혼잡에 의한 것이 아니므로 재전송 타임아웃동안 기다리는 것은 무의미하기 때문에 핸드오프가 완성된 후 곧바로 FH는 손실된 패킷을 MH에게 재전송 함으로써 delay 시간을 줄이고자 했다. 이미 TCP Tahoe 등에서는 fast retransmission을 구현하고 있지만, 이 프로토콜에서는 단지 이전의 TCP와 Mobile IP에 약간의 수정을 가하여 핸드오프가 완성되었음을 FH에게 알리고, FH는 그 즉시 fast retransmission을 호출하게 된다. 그러므로, delay 시간이 길어질수록 기존의 TCP보다 더 나은 성능을 가져온다. 즉, 패킷 손실이 혼잡에 의한 것인지 이동에 의한 것인지를 구분하여, 혼잡에 의한 것이면 이전의 프로토콜을 적용하고, 이동에 의한 것이면 [4]에서 제안하는 프로토콜을 적용하여 이동 컴퓨팅 환경에 더 나은 성능을 제공한다.

2.1.2 I-TCP(Indirect-TCP)

MH가 FH와 통신을 하는 경우 무선네트워크에서는 MH의 이동성과 무선링크의 높은 데이터 손실률에 의해 전송 성능 저하가 발생하게 되고 유선네트워크에서는 데이터의 폭주에 의해 전송성능저하가 발생한다. 이와 같이 유선네트워크와 무선네트워크의 성질이 다른 점에 착안하여 유선네트워크와 무선네트워크의 데이터 전송 정책을 분할하여 적용하려는 연구가 진행되어 왔다.

Indirect-TCP에서는 BS(Base Station)를 MSR(Mobile Support Router)로 언급하고 있다. MH는 MSR과 wireless-TCP를 이용해서 연결하고, MSR은 FH와 기존의 TCP를 이용하여 연결한다. 이러한 I-TCP의 장점은 MH가 핸드오프 시 MSR간의 소켓정보 전송을 통해 소켓의 재설정이 필요하지 않고, 무선 링크에 적합한

TCP를 사용함으로써 전송 성능을 향상시킬 수 있다는 점이다. 그러나, I-TCP의 치명적인 단점은 전송 계층의 연결 분리로 인해 TCP의 종단 간 작동의미(end-to-end semantics)에 위배된다는 것이다. 이것은 FH에서 MH로 TCP패킷을 전송할 때 TCP ack이 MH가 아닌 MSR에서 발생함으로써 MSR의 고장이나 오랜 시간의 연결두절 시에는 중대한 문제가 초래된다. 또한 I-TCP는 새로운 소켓 인터페이스가 적용되므로 기존의 인터넷 어플리케이션은 재 컴파일이나 요구되며, MH의 핸드오프가 발생하면 이전 MSR과 현재 MSR의 전송층간에 연결정보를 유지하고 전송해야 하는 오버헤드도 존재한다. 그러므로 FTP, WWW등의 어플리케이션 계층에서 ack를 제공하는 경우에만 I-TCP가 적합하고, TELNET 등의 전송 계층에서 ack를 사용하는 것은 기존의 일반적인 TCP가 적합하다.

2.1.3 SNOOP

Berkeley 대학에서 제안한 SNOOP 프로토콜[5]은 무선망에서 높은 비트 에러율을 해결하기 위해서 MSR에 SNOOP 모듈을 적용하였으며, 핸드오프 시 발생하는 데이터 손실을 줄이기 위해서 새로운 방식의 라우팅 프로토콜을 제시하였다. SNOOP 모듈의 수행 방법은 다음과 같다. FH에서 MH로 데이터를 전송할 경우, FH가 전송한 패킷이 새로운 패킷이라면 이를 MSR에 저장하고 MH로 전송한다. SNOOP 모듈은 MH에서 보내는 모든 ack를 기억하고 있다가 패킷 손실이 발견되면 이미 MSR에 저장해 둔 패킷을 MH로 재 전송한다. MSR은 MH에서 보낸 중복 ack를 FH로 보내지 않음으로써 FH에서의 불필요한 혼잡 제어 메커니즘 호출을 방지할 수 있다. HA는 primary MSR 뿐만 아니라 이동 호스트의 이웃 MSR에게도 데이터를 포워딩 함으로써 핸드오프가 발생할 때, 다른 프로토콜들과는 달리 이전의 primary MSR에서 새로운 primary MSR로의 데이터를 포워딩하는 과정이 필요 없게 된다. 그 결과, 핸드오프 사이에 발생하는 지연을 없앨 수 있다. SNOOP 프로토콜의 적용은 비트 에러율이 낮은 경우에는 기존의 TCP와 거의 같은 성능을 보이며 비트 에러율이 높은 경우에는 상당한 성능 향상을 나타내고 있다. 또한, I-TCP와는 달리 중간에 인위적인 ack를 발생시키지 않기 때문에 TCP acknowledgment의 semantics를 위반하지 않는다.

2.1.4 링크 계층 프로토콜

무선망과 유선망으로 구성된 망은 유선망에 비해 MH가 갖는 제한된 전력, 작은 대역폭[6], 처리 능력의 차이로 인해 비대칭(asymmetry)을 이룬다. AIRMAIL[7] 프로토콜에서는 이런 차이를 그대로 인정하고 BS에 intelligent한 타이머를 두며, 복잡한 상태 메시지 처리

부분과 결정권을 모두 BS에 일임한다. AIRMAIL 프로토콜에서는 데이터 전송 시의 reliability와 성능향상을 위해서 ARQ와 FEC를 결합하여 이용하고 있다. 데이터 전송 시에 여분(redundant)의 패킷을 함께 전송시킴으로써 수신 측에서 정확하게 받은 패킷과 여분의 패킷에 근거를 두고 손상되거나 손실된 패킷의 재생성을 가능하게 하는 FEC[7] 기술은 패킷의 재전송을 줄임으로써 성능 향상을 가져온다. [7]의 시뮬레이션에서는 narrow-band outdoor처럼 대부분의 블록이 에러를 가진 byte로 되어 있으며 그 수가 두드러지게 변하는 경우에는 bit-, byte-level의 FEC가 적합하며, 대부분의 byte가 에러가 없는 wideband indoor인 경우에는 블록 중에 에러가 있으며 그 수가 많기 때문에 패킷-level의 FEC가 적절하다. 다만, 이러한 링크 계층 프로토콜에서의 재전송은 TCP와 같은 전송 계층에서의 재전송과의 경쟁에 의해 오히려 역효과를 가져올 수도 있다. 또한 링크 계층에서의 타이머와 전송 계층에서의 타이머간에도 충돌이 일어날 수 있다. 이 두 계층이 서로 적절하게 조화를 이룬다면 최고의 throughput을 제공할 수 있다[8].

2.1.5 TCP/IP 헤더 압축 기법

Van Jacobson은 무선 링크와 같은 저속의 링크를 위해 TCP/IP 헤더 압축 기법을 제안하였고[9], Degermark은 무선 통신망을 위한 TCP/IP 및 UDP/IP 헤더 압축 기법을 제안하였다[10]. 헤더 압축 기법은 연속적인 TCP/IP 패킷 흐름에서 헤더의 대부분 내용이 바뀌지 않는다는 점에 착안하여, 송신측에서 보내는 첫 번째 TCP/IP 패킷은 완전한 헤더 정보를 포함해서 전송하고, 두 번째 패킷부터의 헤더에는 최소한의 정보만 포함해서 전송한다. 수신측에서는 첫 번째 패킷의 완전한 헤더 내용을 저장하고 있다가 헤더가 압축된 패킷을 받으면 저장해 둔 헤더정보와 받은 패킷의 헤더로부터 패킷의 전체 헤더를 복구한다. 이 기법에는 헤더 정보를 전송하는데 사용되는 대역폭이 줄어들기 때문에 무선링크의 대역폭을 절약할 수 있는 장점이 있다.

그러나 송신측에서 전송한 완전한 헤더 정보가 포함된 패킷이 전송 중에 손실되면 수신측에서는 그 이후에 수신된 모든 패킷을 복원할 수 없는 문제점이 발생할 수 있을 뿐만 아니라 무선 통신망에서 패킷 손실로 인한 TCP 프로토콜의 성능저하에 대한 해결 방안은 제시하지 못하였다.

3. 제안하는 LIW(Loss In Wireless) flag를 이용한 혼잡 제어 메커니즘

지역 재전송 기법을 기반을 둔 SNOOP 기법[11]에서는 MSR의 캐쉬가 오버플로우 되면 전체 연결의 성능이 떨어진다. MSR에서 저장하지 못한 패킷이 무선 링

크에서 손실되어 재전송이 요청되면 혼잡에 의한 패킷 손실이 아님에도 불구하고 송신 쪽에서는 윈도우 사이즈를 줄이는 혼잡 제어 메커니즘을 호출한다. 이 단점을 극복하는 방안으로 이 장에서는 새롭게 정의된 LIW flag를 이용해서 불필요하게 혼잡 제어 메커니즘을 호출하지 않는 기법에 대해서 설명한다.

3.1 MSR에서의 동작

MSR의 캐시는 SNOOP 캐시처럼 데이터 패킷만 저장하는 것이 아니라 TCP connection에서 전송되는 패킷과 Sequence Number(SN)를 각각 저장하여 MSR에서 재전송할 데이터가 없더라도 SN이 저장되어 있으면 유선 상에서는 손실이 없음을 알고 무선 상에서 일어나는 손실에 대해서 기존의 혼잡 제어 메커니즘이 수행되지 않도록 한다.

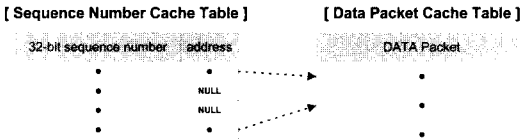


그림 1 Cache table의 자료구조

그림 1에서는 Sequence Number Cache Table(SNCT)

과 Data Packet Cache Table(DPCT)의 구조를 보여주고 있다. SNCT의 두 번째와 세 번째 항목이 null로 되어 있는 것은 DPCT에서 오버플로우가 발생하여 데이터를 저장하지는 못하고 SN 값만을 저장했기 때문이다.

MSR에서는 MH가 보내는 ack패킷과 자신이 저장해 놓은 SN을 비교하여 손실이 없는 경우인지, 유선링크 상에서의 손실인지, 무선링크 상에서의 손실인지를 판단하게 된다. 그림 2는 이를 판별하는 알고리즘을 도식화하였다.

- ① 오류가 있는 ack의 경우, SNCT에 저장되어 있지 않은 SN에 대한 ack이고 중복된 ack도 아닌 경우로 거의 발생하지 않는 상황이다. 이 ack는 MSR에서 마지막으로 수신된 ack보다 작은 ack에 해당하며, 이 ack를 폐기시키고 계속 진행한다.
- ② 유선 링크상의 손실인 경우, SNCT에 저장되어 있지 않은 SN에 대한 중복된 ack를 받은 것은 FH가 보낸 패킷이 MSR까지 전송되지 않은 경우이다. 이는 패킷이 유선 링크 상에서 손실된 것을 말하고, 이 중복된 ack를 그대로 송신측에 전달하여 기존의 혼잡 제어 메커니즘을 호출한다. 그러면 FH에서는 윈도우 사이즈를 줄이고 손실된 패킷을 재전송 하게 된다.
- ③ 손실이 없는 정상적인 경우, 저장된 SN에 대한 ack이고 중복된 ack가 아니면 유선 링크와 무선 링크

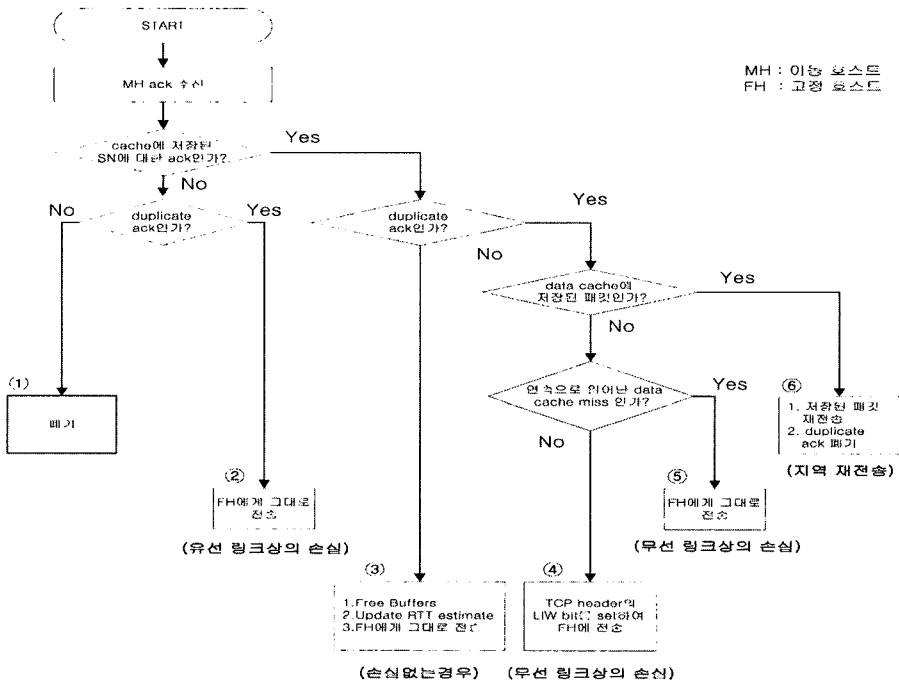


그림 2 Decision 알고리즘

모두 손실이 없었음을 나타내는 것이므로, SNCT과 DPCT에서 ack된 모든 패킷들의 항목을 삭제한다. 무선 구간에 대한 RTT 측정도 이때 갱신된다. 또한 이 ack는 유선망의 고정 호스트로 그대로 전송된다. 마지막으로, SNCT에 저장되어 있는 SN에 대한 중복된 ack인 경우는 무선 링크 상에서 손실이 발생한 경우로 다시 다음의 세 가지로 세분된다.

- ④ 중복된 ack를 통해서 재전송이 요구되는 패킷이 MSR의 DPCT에 저장되어 있지 않고 연속되는 cache miss가 아닌 경우, 이 경우는 DPCT에 오버플로우가 생겨서 송신 측으로부터 전송된 패킷을 DPCT에 저장할 수 없었고, 무선 링크 상에서 그 패킷이 손실된 경우이다. 하지만 SNCT를 통해 유선까지는 손실되지 않은 패킷임을 알 수 있으면, FH는 기존의 혼잡 제어 매커니즘을 호출하지 않고 재전송한다. 이를 유선망의 FH에게 알리기 위해서는 중복된 ack의 TCP 헤더에 그림 3과 같이 새롭게 정의한 LIW(Loss In Wireless) flag를 1로 세팅하여 보낸다.

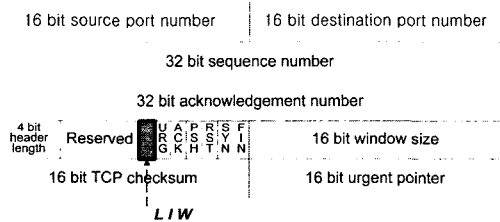


그림 3 추가된 LIW flag

송신측에서는 중복된 ack를 수신하게 되면 기존의 fast retransmit 매커니즘을 작동시키게 되지만, 그 이전에 LIW flag를 조사하여 1로 세팅되어 있으면 윈도우 사이즈와 slow start threshold 값을 기존의 값으로 유지한 채 손실된 패킷을 재전송 한다.

- ⑤ 중복된 ack를 통해서 재전송이 요구되는 패킷이 MSR의 DPCT에 저장되어 있지 않고 연속된 cache miss인 경우, 이 경우도 위의 ④의 경우와 마찬가지로 무선 링크 상에서 패킷이 손실된 상황이지만, 송신 쪽에서 윈도우 사이즈를 계속해서 줄이지 않는 방법은 유선 링크 상에서 더욱 심한 혼잡을 유발시킬 수 있다. 또한 cache miss가 계속 일어나고 있는 상황에서 윈도우 사이즈를 계속해서 늘린다면 cache miss율은 더욱 증가하게 된다. 그러므로 어느 시점에서는 윈도우 사이즈를 줄이는 방법이 필요하다. MSR의 DPCT에서 cache miss가 연속적으로 일어난다면 LIW flag를 1로 세팅하지 않고 그대로 FH에 중복된 ack를 전달하여 기존의 혼잡 제어 메커니

즘을 호출하고 손실된 패킷을 재전송 한다.

- ⑥ 중복된 ack를 통해서 재전송이 요구되는 패킷이 MSR의 DPCT에 저장되어 있는 경우, 이 경우 역시 무선 링크 상에서 패킷 손실이 일어났으나 재전송할 패킷이 DPCT에 존재하므로 MSR은 중복된 ack를 유선망의 FH로 전달하지 않고 폐기시키고 손실된 패킷을 재전송 한다. 중복된 ack를 받지 못한 FH는 패킷 손실이 있어 낮음을 알지 못하게 되고 불필요한 혼잡 제어 매커니즘을 호출하지 않은 채 계속 다음 패킷을 전송할 수 있다.

기존의 SNOOP 기법에서는 데이터 캐쉬만을 유지하며 FH가 보낸 패킷을 중간에 MSR가 캐쉬에 저장하고 MH으로 포워딩하는데 만일 응답이 없는 경우 MSR에서 MH으로의 재전송을 시도한다. 즉, 세션 장애에 대한 재전송은 로컬 영역 내부에서만 행해진다. 만일 Cache-miss가 발생하는 경우 로컬에서 재전송할 정보가 없으므로 FH로 재전송을 요청하게 되며 이때 혼잡상황을 처리하기 위해 FH에서는 혼잡제어 알고리즘이 실행되고 MH과 FH간의 세션 성능은 감소하게 된다.

본 논문에서는 데이터 캐쉬 테이블에서 오버플로우 발생으로 인한 Cache-miss 환경을 고려하고 있다. 만일 MSR 영역 내에 많은 MH이 존재하고 여러 개의 TCP 세션 연결을 통해 대량의 트래픽이 발생하는 경우 이를 각 세션별로 수신되는 패킷을 캐쉬에 저장해야 하는데 문제는 캐쉬의 크기는 한정되어 있으므로 세션의 증가로 인한 오버플로우가 발생할 수 있다. 그러나 이는 MSR 내부에서의 문제로서 세션의 혼잡 상황과는 무관하다. 이 경우 기존의 SNOOP 방식에서는 MSR에서 MH으로 재전송할 패킷을 분실한 경우이므로 FH로 재전송을 요구하고 이는 혼잡제어의 실행을 초래하므로 이후 세션은 혼잡 제어에 의해 영향을 받는다. 본 논문에서는 데이터 캐쉬 이외에 2차 캐쉬를 둔다. 2차 캐쉬에는 FH로부터 받은 응답 패킷에 존재하는 시퀀스 번호들로 구성되므로 데이터 캐쉬 miss가 발생해도 만일 시퀀스 번호 캐쉬가 존재한다면 이는 세션 장애로 인한 손실이 아닌 MSR의 오버플로우로 인한 문제이므로 재전송을 요청하지만 혼잡상황은 아님을 FH로 알려주기 위해 TCP 헤더의 Reserved 영역에 있는 플래그(LIW)를 사용한다. 재전송 요청을 수신하면 FH은 플래그 값을 검사해서 만일 설정된 경우 재전송만 처리하며 설정되지 않은 경우 재전송과 흐름제어를 함께 처리한다. 따라서 기존의 SNOOP방식에 추가해야 할 부분은 시퀀스 번호 캐쉬 테이블 및 FH에서 LIW 플래그 처리를 위한 별도의 TCP 개선 작업이 요구된다.

4. 성능 분석

4.1 성능평가 모델

본 논문에서 제안하는 기법에 대한 성능평가 모델은 그림 4와 같다. 유무선 통합 망에서 FH가 MH로 데이터를 전송할 때 제안하는 기법을 사용했을 경우와 기존의 TCP만을 사용했을 경우, SNOOP을 사용했을 경우의 throughput의 변화를 실험하였다.

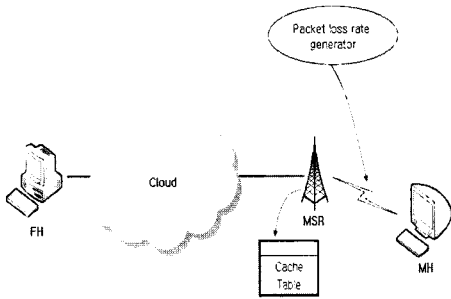


그림 4 성능평가 모델

무선 링크의 에러율은 BER를 평균으로 하는 포아송 분포를 따라 발생한다고 가정하고 Link Layer의 에러 수정 정책은 없다고 간주하여 각 패킷이 손실될 것인지 아닌지를 결정한다. 패킷의 손실은 데이터 패킷 뿐 아니라 ack 패킷도 손실될 수 있다고 보았다. 또한 유선 링크상의 라우터가 갖는 버퍼 크기에도 제한을 두어 네트워크의 혼잡에 의한 손실도 고려하였다. 성능평가에 쓰인 파라미터는 표 1과 같다.

표 1 성능분석 파라미터

파라미터	값	설명
S_t	512 bytes	TCP 데이터 패킷 크기
S_a	40 bytes	ACK 데이터 패킷 크기
$B_{wireless}$	2 Mbps	무선 링크 대역폭
B_{wired}	10 Mbps	유선 링크 대역폭
S_r	30~50 Kbytes	라우터의 버퍼 크기
W_{max}	32 segments	최대 윈도우 사이즈
W	1~32 segments	현재 전송중인 윈도우 사이즈
m	10~36 Kbytes	MSR의 캐쉬 사이즈
R	4~14 %	packet loss rate

SNOOP 프로토콜에서 TCP의 throughput은 지역 재전송을 위한 cache 메모리 크기에 따라 영향을 받는다. 재전송이 필요한 손실된 패킷의 경우, SNOOP cache의 크기를 m 이라고 하고 현재 전송중인 윈도우 사이즈를 W 라고 하면 SNOOP cache에 재 전송할 패킷이 있을 확률은 $p = \frac{m}{W}$ 로 정의된다. B_h 를 cache hit일 때의 throughput이라고 하고 B_m 를 cache miss일 때의

throughput이라고 하면 TCP의 전체 throughput은 $p \times B_h + (1-p) \times B_m$ 가 된다.

MSR이 중복된 ack를 수신하였을 때 cache miss일 확률을 $1 - \frac{m}{W}$ 이라 놓고 랜덤한 환경으로 가정한 확률값을 발생시켜 현재 상태에서 cache miss인지 hit인지를 결정한다.

4.2 결과 및 분석

각 그래프에서 제안하는 기법을 LIW라고 표시하였고, SNOOP기법을 사용한 경우와 기존의 TCP만을 사용했을 때의 throughput을 비교하였다.

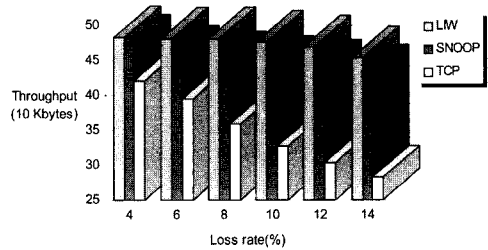


그림 5 Loss rate에 따른 throughput의 변화

그림 5는 loss rate에 따른 throughput의 변화량에 대한 그래프이고, MSR의 cache size는 24 Kbytes로 고정시켰다. 그림에서 loss rate가 증가할수록 기존의 TCP를 사용하는 경우는 성능이 급격히 감소하지만, SNOOP과 제안하는 기법은 loss rate의 변화에 따라 성능이 급격히 감소하지는 않는다.

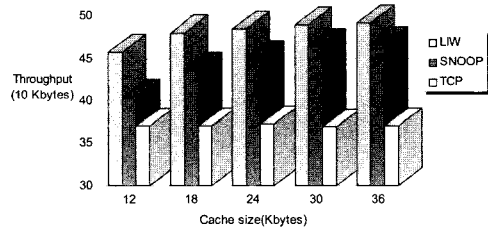


그림 6 Cache size에 따른 throughput의 변화

그림 6은 cache size에 따른 throughput의 변화량에 대한 그래프이고, loss rate는 8 %로 고정시켰다. 기존의 TCP를 사용하였을 때는 cache size에 영향을 받지 않으므로 일정한 값을 유지하나 SNOOP과 제안하는 기법은 cache size가 증가함에 따라 성능이 향상됨을 보인다. cache size가 작을 때는 SNOOP이 제안하는 기법과 비교하여 throughput의 차이가 많이 나지만, 이 실험에서는 cache size가 30 Kbytes일 때부터 성능 향상 폭이 줄고 안정적인 상태로 진입하였음을 볼 수 있다.

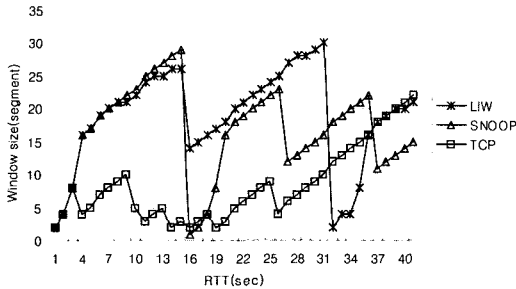


그림 7 LIW, SNOOP, 기존의 TCP에서의 윈도우
사이즈 변화

그림 7은 LIW와 SNOOP, 그리고 기존의 TCP에서 윈도우 사이즈가 변화하는 모습을 나타낸 그래프이다. loss rate는 12% 이고, cache의 크기는 10 Kbytes로 평균 miss rate가 36% 일 때의 상황이다.

기존의 TCP는 무선 상에서 일어난 손실로 혼잡 제어 메커니즘을 호출한 결과 윈도우 사이즈가 증가하지 못하는 반면, LIW와 SNOOP은 무선 상의 손실로 윈도우 사이즈가 불필요하게 줄어들지 않았다. LIW와 SNOOP은 계속되는 윈도우 사이즈 증가로 인해 1번씩의 유선 상의 패킷 손실을 유발 시켰지만, LIW는 SNOOP에서 cache miss로 인해 윈도우 사이즈가 불필요하게 감소되는 현상을 막아 총 전송량은 LIW가 457,000 bytes, SNOOP이 421,376 bytes, 기존의 TCP가 369,930 bytes로 LIW가 기존의 TCP보다는 23.5%, SNOOP보다는 11%의 성능향상을 가져 왔다.

4.3 향상된 SNOOP 기법들과 제안 방식과의 성능 비교

향상된 SNOOP 기법들로는 MH과 MSR사이의 SNACK를 적용한 다중 분실패킷의 재전송, Rate-control 및 버퍼 관리를 통한 혼잡 제어 방식들이 있다 [12,13]. 먼저 SNACK를 이용한 방식은 무선 링크 상에서 여러 개의 패킷 손실에 대한 명시적인 정보를 제공하기 위해 SNACK를 사용하는데 FH가 MH로 보내는 다중 패킷에 대한 분실이 발생하는 경우 MSR로부터 패킷 손실을 알리는 SNACK 옵션 메시지를 수신하면 한꺼번에 손실된 패킷을 재전송한다. SNOOP이 한 번에 한 ACK만을 처리하는데 비해 SNACK 방식은 명시적으로 표시된 SNACK 정보를 이용해 한 번에 여러 패킷의 손실을 처리하여 재전송한다. 이는 MH에서 FH로의 패킷 분실인 경우에도 동일하게 적용된다. 따라서 패킷 손실률에 따른 높은 처리율을 제공한다. Rate-control 방식은 MSR에 있는 SNOOP Agent가 TCP 연결을 통해 지나가는 모든 패킷들을 감시하고, 수신지 링크로 보낸 후 응답을 받지 못한 TCP 세그먼트들에 대한 캐쉬를 유지한다. Agent는 패킷 사이의 지연 시간

을 관찰해서 최적의 전송률을 계산해서 MSR가 FH와 MH간의 세션에 대한 Rate-control을 실행하는 방법이며 SACK 기반의 재전송 구조 및 버퍼의 임계값을 이용한 전송률 제어기능들이 포함된다.

제안하는 기법에서는 앞에서 설명한 바와 같이 패킷의 시퀀스 번호를 함께 저장하므로 분실 패킷에 대한 다중 재전송 요청이 가능하고 버퍼의 임계치 즉, 버퍼 오버플로우 발생 시점에 도달해도 즉시 흐름제어를 실행하지 않고 2차적으로 시퀀스 번호 캐쉬 항목 존재 여부를 통해 흐름제어 여부를 판단한다. 따라서 혼잡 발생 시 SNACK와 비슷한 성능을 가지며 2차 캐쉬를 둬으로써 버퍼 오버플로우 발생시 2차 캐쉬를 검색한 후 흐름 제어 여부를 판단하므로 Rate-control 방식에 비해 2차 캐쉬에 시퀀스 번호가 존재하는 경우 흐름제어 없이 FH로 재전송만 요청하게 되므로 Rate-control 방식에 비해 TCP 세션의 성능은 향상된다.

5. 결론 및 향후 과제

최근 휴대용 컴퓨터와 무선통신 기술의 발달로 이동 컴퓨팅 환경에 대한 관심이 나날이 높아지고 있다. 오늘날의 인터넷은 대부분이 유선망에 기반을 두고 있으며 이러한 유선망에 무선망을 도입한 유·무선 통합망의 구현이 현실화되고 있다. 그러나 가장 널리 쓰이는 수송 계층 프로토콜인 TCP는 패킷 손실의 원인을 망의 혼잡 때문에 일어난다고 가정하고 있으므로 기존의 유선망과 고정 호스트로 이루어진 전통적인 네트워크에 적합하며 패킷 손실이 주로 에러율과 핸드오프에 의해 발생하는 무선망에서는 적합하지 않다.

본 논문에서는 패킷손실이 유선링크에서 일어나는 것인지, 무선링크에서 일어나는 것인지를 판별하는 기법을 제안하였다. 그래서 패킷손실이 유선링크에서 일어난 경우는 기존의 혼잡제어 메커니즘을 호출하여 재전송하고, 무선링크에서 일어난 경우는 새롭게 정의한 LIW flag를 이용하여 혼잡제어 메커니즘을 호출하지 않고 재전송을 하여 윈도우 사이즈가 무선상의 손실로 불필요하게 줄어드는 요인을 제거하였고 이에 따른 성능을 평가하였다.

향후 과제로는 핸드오프시의 문제점에 대한 해결책을 제시하고 이에 대한 성능을 평가하는 것이다.

참고 문헌

- [1] W.R. Stevens, TCP/IP Illustrated, vol. 1. Reading, MA: Addison-Wesley, Nov. 1994.
- [2] V. Jacobson, "Congestion avoidance and control," in Proc. ACM SIGCOMM '88, Aug. 1998.
- [3] H. Balakrishnan, V.N. Padmanabhan, S. Seshan,

and R.H. Katz, "A Comparison of Mechanisms For Improving TCP Performance Over Wireless Links," IEEE/ACM Transactions on Networking, vol. 5, no. 6, Dec. 1997.

[4] 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.

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

[6] H. Balakrishnan, V.N. Padmanabhan, and R.H. Katz, "The effects of asymmetry on TCP performance," Proc. 3rd IEEE/ACM Int'l. Conference on Mobile Computing and Networking (MobiCom), Budapest, Hungary, Sept. 1997.

[7] E. Ayanoglu, S. Paul, T.F. LaPorta, K.K. Sabnani, and R.D. Gitlin, "AIRMAIL: A link-layer protocol for wireless networks," ACM/Baltzer Wireless Network J., vol. 1, pp. 47-60, Feb. 1995.

[8] H. Balakrishnan, V.N. Padmanabhan, S. Seshan, and R.H. Katz, "A Comparison of Mechanisms For Improving TCP Performance Over Wireless Links," IEEE/ACM Transaction on Networking, vol. 5, no. 6, Dec. 1997.

[9] V. Jacobson, "Compressing TCP/IP Header for Low-Speed Serial Links," Network Working Group, RFC 1144, Feb. 1990.

[10] M. Degermark, M. Engan, B. Nordgen, and S. Pink, "Low-loss TCP/IP Header Compression for Wireless Network," MOBICOM '96, pp. 1-14, Rye, New York, Nov. 1996.

[11] R.Yavatkar and N. Bhagwat, "Improving End-to-End performance of TCP over mobile internet-networks," Mobile 94 Workshop on Mobile Computing Systems and Applications, Dec. 1994.

[12] F. Sun, V.Li and S.C.Liew, "Design of SNACK Mechanism for Wireless TCP with New Snoop," IEEE Communication Society, WCNC 2004.

[13] Y.J. Song and Y.J. Shu, "Rate-control Snoop: A Reliable Transport Protocol for Heterogeneous Networks with Wired and Wireless Links," IEEE Communication Society, 2003.



강 인 석

1992년 숭실대학교 전자계산학과 입학
 1999년 숭실대학교 전자계산학과(학사)
 2001년 숭실대학교 대학원 컴퓨터학과(석사). 2001년~현재 팬택엔큐리텔 전임 연구원으로 재직중. 관심분야는 이동통신, mobile ip



문 영 성

1993년 연세대학교 전자공학과(학사). 1986년 Unive. of Alberta 전자공학과 졸업(석사). 1993년 Univ. of Texas, Arlington 전산학과 졸업(박사). 1994년~현재 숭실대학교 컴퓨터학부 부교수. 관심분야는 Mobile IP, IPv6, 무선 TCP, GRID,