유·무선 혼합망에서 Cross-Layer기반의 에너지 효율적인 전송 기법

(An Energy Efficient Transmission Scheme based on Cross-Layer for Wired and Wireless Networks)

김 재 훈 † 정 광 수 ††
(Jaehoon Kim) (Kwangsue Chung)

요 약 Snoop 프로토콜은 유·무선 혼합망에서 무선 링크에서 발생하는 TCP 패킷 손실을 효과적으로 보상하여 TCP 전송률을 향상시킬 수 있는 효율적인 프로토콜이다. 하지만, 무선 링크에서 연집한 패킷 손실이 발생하는 경우에는 지역 재전송을 효과적으로 수행하지 못하여 전송 효율이 떨어진다는 문제점이 있다. 이러한 Snoop 프로토콜의 문제점을 개선하기위해 무선 구간에서 TCP-SACK의 장점을 활용한 SACK-Aware-Snoop, SNACK 메커니즘 등이 제안되었다. SACK-Aware-Snoop, SNACK 메커니즘은 연집한 패킷손실 환경에서도 Snoop 프로토콜보다 높은 전송률을 보장하지만 전송 계층의 ACK 패킷을 기반으로 재전송을 수행한다는 점은 ACK 패킷의 손실에 심각한 전송 성능 저하를 가져오며, 무선 구간에서 SACK 옵션의 사용은 무선망의 대역폭과 이동 단말의 한정된 에너지 자원을 불필요하게 낭비하는 문제를 초래하게 된다. 본 논문에서는 이러한 문제점을 개선하기 위해 Cross-layer 기법을 적용한 지역 재전송 기법인 C-Snoop(Cross-layer Snoop) 프로토콜을 제안한다. C-Snoop 프로토콜은 현재 유·무선 혼합망에서 가장 널리 사용되는 IEEE 802.11 MAC 프로토콜 기반의 지역 재전송 메커니즘으로서, MAC 계층의 ACK 패킷과 새로이 제안된 지역 재전송 타이머에 의해 효율적인 지역 재전송을 수행한다. ns-2 시뮬레이터를 이용한 실험을 통해 C-Snoop의 지역 재전송 기법은 무선 구간의 연집적인 패킷 손실에 대해 효율적인 보상을 수행하며, 이동 단말의 에너지 효율성을 향상시키는 것을 확인할 수 있었다.

키워드 : TCP, Snoop, SNACK, Cross-layer, 프로토콜의 에너지 효율

Abstract Snoop protocol is one of the efficient schemes to compensate TCP packet loss and enhance TCP throughput in wired-cum-wireless networks. However, Snoop protocol has a problem: it cannot perform local retransmission efficiently under the bursty-error prone wireless link. To solve this problem, SACK-Aware-Snoop and SNACK mechanism have been proposed. These approaches improve the performance by using SACK option field between base station and mobile host. However in the wireless channel with high packet loss rate, SACK-Aware-Snoop and SNACK mechanism do not work well because of two reason: (a) end-to-end performance is degraded because duplicate ACKs themself can be lost in the presence of bursty error, (b) energy of mobile device and bandwidth utilization in the wireless link are wasted unnecessarily because of SACK option field in the wireless link. In this paper, we propose a new local retransmission scheme based on Cross-layer approach, called Cross-layer Snoop(C-Snoop) protocol, to solve the limitation of previous localized link layer schemes. C-Snoop protocol includes caching lost TCP data and performing local retransmission based on a few policies dealing with MAC-layer's timeout and local retransmission timeout. From the simulation result, we could see more improved TCP throughput and energy efficiency than previous mechanisms.

Key words : TCP, Snoop, SNACK, Cross-layer, Energy efficiency of mobile devices

· 이 논문은 2007년도 정부(과학기술부)의 재원으로 한국과학재단의 지원을 받아 수행된 연구임(R01-2005-000-10934-0)
† 학생회원 : 광운대학교 전자통신과
jhkim@adams.kw.ac.kr
†† 종신회원 : 광운대학교 전자통신과 교수
kchung@daisy.kw.ac.kr
논문접수 : 2006년 5월 15일
심사완료 : 2007년 9월 18일

: 개인 목적이나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.
정보과학회논문지 : 정보통신 제34권 제6호(2007.12)
Copyright@2007 한국정보과학회

1. 서 론

TCP는 지난 20년을 통틀어 가장 성공적인 전송 계층 프로토콜이다. 이러한 TCP의 성공 요인으로서 신뢰성 있는 데이타 전송, 효과적인 트래픽 관리, IP와의 효율적인 결합 등을 들 수 있다. 이와 같은 장점에 의해 대다수의 인터넷 응용 프로그램들은 전송 계층 프로토콜로서 TCP를 사용하고 있다. 하지만 유선망에 최적화되도록 진화해온 TCP는 무선망이 가지는 불안정한 링크에 의한 데이타 손실을 유선망에서처럼 네트워크의 혼잡으로 인한 손실로 오해한다. 그 결과 혼잡 제어 메커니즘이 수행되어 전송률을 줄이므로 네트워크 성능이 저하되는 문제점을 초래 한다[1].

이러한 문제를 해결하기위해 최근 몇 년간 TCP의 성능을 향상시키기 위한 연구가 활발히 이루어지고 있다. 유·무선 혼합망에서 TCP의 성능을 향상시키기 위한 방법은 크게 세 가지로 구분할 수 있다. Snoop 프로토콜과 같은 링크 계층 재전송 방식, I-TCP, M-TCP와 같은 연결 분할 접근(Split Connection) 방식 그리고 TCP-Vegas, Veno와 같은 종단간(End-to-End) 방식으로 분류 할 수 있다[2-6].

위의 다양한 메커니즘 중에서 Snoop 프로토콜은 TCP의 종단간 특성을 유지하면서도 무선 링크에서의 패킷 손실에 의한 송신단에서의 불필요한 TCP 혼잡 제어의 발생을 방지하여 TCP의 전송율을 매우 효과적으로 향상시킬 수 있는 프로토콜로 알려져 있다[7]. Snoop 프로토콜은 무선 링크와 유선 링크의 경계가 되는 BS(Base Station)에 존재하는 Snoop agent가 지역 재전송(Local Retransmission)을 수행하여 무선 링크에서의 패킷 손실을 보상하는데, 지역 재전송은 TCP의 수신단으로부터 중복 ACK 패킷을 수신하였을 경우에 수행되며, 재전송한 패킷에 대해 Snoop agent의 지역 재전송 타이머가 만료 되었을 경우 해당 패킷을 다시 재전송하게 된다. 또한, Snoop agent는 중복 ACK 패킷이 TCP 송신단으로 전달되는 것을 막아, 송신단에서 혼잡 제어를 수행하는 것을 예방하여 TCP 전송률이 저하되는 것을 방지한다. 그러나 패킷이 연속적으로 손실되는 무선 링크에서 Local RTT(Round Trip Time)마다 하나의 손실된 패킷만을 복구하는 Snoop 프로토콜은 에러 복구 과정에서 발생되는 지연시간으로 인해 전송 성능에 큰 영향을 받게 된다. 또한, Snoop 프로토콜은 중복 ACK 패킷을 수신하였을 경우 재전송을 수행하고, 지역 재전송 타이머는 재전송한 패킷에 대해서만 적용하므로 연집적인 패킷 손실(Bursty Packet Loss)이 발생하는 무선구간에서의 ACK 패킷 손실에 심각한 성능 저하가 발생한다. 이러한 Snoop의 연집에러 복구의 한계를 효율적으로 해결하기 위해 무선구간에 SACK 옵션 또는 SACK과 유사한 추가옵션을 사용하는 방식인 SACK-Aware-Snoop, SNACK 메커니즘 등이 제안되었다[8-10].

하지만 기존의 Snoop 프로토콜과 같이 전송 계층의 중복 ACK 기반의 재전송을 사용한다는 점은 전송 손실률이 높고, 패킷 손실의 연집성이 높은 무선구간에서 ACK 패킷의 손실에 의해 TCP 전송률의 저하를 가져올 수 있다. 또한, ACK 패킷에 SACK 옵션과 같이 추가 옵션을 사용하는 점은 무선망의 대역폭과 이동 단말의 한정된 에너지 자원을 불필요하게 낭비하는 문제를 초래하게 된다.

본 논문에서는 IEEE 802.11 MAC 프로토콜 기반의 유·무선 혼합망에서 발생하는 재전송 과정의 문제점을 해결하고, 무선구간의 한정된 자원과 이동 단말의 에너지를 효율적으로 사용하기 위해 새로운 Cross-layer 기반의 재전송 기법인 C-Snoop(Cross-layer Snoop) 프로토콜을 제안한다. C-Snoop 프로토콜은 기존의 지역 재전송 메커니즘과 같이 무선구간의 손실된 패킷에 대해 재전송을 통한 보상을 수행한다. 하지만, 재전송 기법은 기존의 Snoop 프로토콜이나 SNACK 메커니즘과는 달리 MAC 계층의 ACK 패킷(MAC-ACK)을 수신하거나, 새롭게 제안된 지역 재전송 타이머가 만료되었을 경우에 수행된다. 이러한 재전송 기법을 수행하는 C-Snoop 프로토콜은 전송 계층의 ACK 패킷 손실에 영향을 받지 않아 무선구간의 연속된 패킷손실에 효율적인 재전송을 가능하게 하며, MH(Mobile Host)와 BS(Base Station)가 각각 독립적인 재전송을 수행함으로써 전송 환경이나 전송 방향에 상관없이 효율적인 전송 성능 향상을 보장한다.

본 논문은 다음과 같이 구성된다. 2장에서는 기존의 Snoop 프로토콜과 무선구간에 추가옵션을 사용하는 기법인 SNACK 프로토콜의 동작방법과 문제점에 대해 기술하고, 3장에서는 본 논문이 제안하는 새로운 재전송 기법에 대해 설명한다. 4장에서는 제안한 재전송 기법의 성능을 평가하기 위한 실험과 5장에서는 결론 및 향후 과제에 대해 기술한다.

2. 관련 연구

본 장에서는 유·무선 혼합망에서 기존의 지역 재전송 메커니즘인 Snoop 프로토콜과 SNACK 메커니즘의 동작 기법과 문제점에 대해 기술한다.

2.1 Snoop 프로토콜

Snoop 프로토콜은 기존의 TCP에 아무런 수정없이, TCP의 종단간 특성을 유지하면서 BS에서의 지역 재전송을 통해 TCP 전송율을 향상시키는 방법이다. Snoop

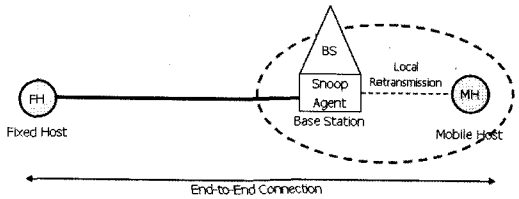


그림 1 Snoop 시스템 구성

프로토콜은 링크 계층에서 동작하는 프로토콜로서 그림 1과 같은 시스템으로 구성된다.

Snoop 프로토콜은 BS에 TCP 패킷과 ACK 패킷의 흐름을 관찰하는 Snoop agent를 두어 무선 링크에서의 패킷 손실에 의한 FH의 불필요한 TCP 혼잡제어가 발생하지 않도록 한다. BS에 위치한 Snoop agent는 FH에서 MH로 전송되는 TCP 패킷을 BS 내의 Snoop 버퍼에 저장한 후 MH로 전송한다. 그리고 MH로부터 수신되는 ACK 패킷의 종류에 따라서 재전송을 수행하며, 재전송한 패킷에 대해서는 지역 재전송 타이머의 만료에 의해 해당 패킷을 다시 재전송하여 무선 링크에서의 패킷 손실을 보상한다. 하지만 Local RTT 마다 하나의 손실된 패킷만을 복구하는 Snoop 프로토콜은 연집적인 패킷 손실이 발생하는 무선구간에서 에러 복구에 많은 지연시간을 소모하게 된다. 이러한 지연시간은 TCP의 혼잡제어 메커니즘을 호출하여 심각한 성능저하를 가져올 수 있다. 그림 2는 연집적인 패킷손실에 대한 FH가 전송하는 경우와 MH가 전송하는 경우의 두 가지 전송 방향에 대한 Snoop 프로토콜의 에러 복구 과정을 나타낸 것이다. 그림 2에 나타낸 것과 같이 Snoop 프로토콜은 에러 복구 과정에서 무선구간에서 손실된 패킷의 수 ×Local RTT 만큼의 지연 시간이 발생한다.

Snoop 프로토콜의 에러 복구 지연시간과 관련된 문제점을 해결하기 위해 SNACK 메커니즘과 SACK-Aware-Snoop 프로토콜이 제안되었다. SACK-Aware-Snoop 프로토콜과 SNACK 메커니즘의 동작방법은 ACK 패킷의 옵션 필드 크기의 차이만 있고, 거의 유사

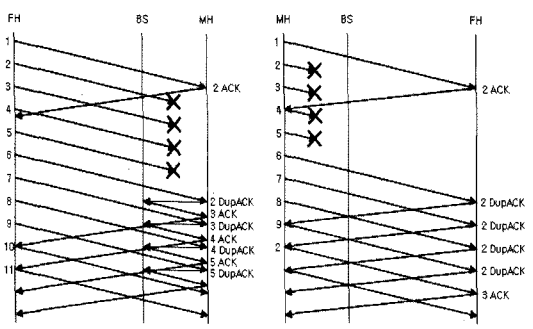


그림 2 Snoop 프로토콜의 에러 복구 과정

한 에러 복구 과정을 수행하므로 본 논문에서는 SNACK 메커니즘에 대해서만 기술하고자 한다.

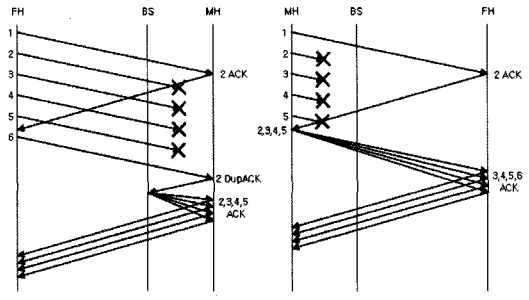
2.2 SNACK 메커니즘

기존의 Snoop 프로토콜의 재전송 기법은 무선구간의 연속된 패킷 손실을 복구하는 과정에서 많은 Local RTT를 소모한다. 이러한 재전송 과정의 지연은 FH의 타임아웃을 발생시켜 심각한 성능 저하를 가져오는 원인이 된다. Snoop 프로토콜의 연집 에러 복구의 한계를 해결하기 위해 SNACK 메커니즘은 무선 구간에서 추가 옵션을 사용한 새로운 재전송 기법을 제안한다. 기본적인 동작 방법은 Snoop 프로토콜과 유사하다. 단, 무선 구간의 연집 에러를 빠르게 복구하기 위해 ACK 패킷의 헤더에 SACK 옵션과 같이 추가 옵션을 사용한 SNACK을 이용하여 한 Local RTT안에 손실된 모든 패킷을 재전송 하는 방법을 사용한다. 이는 유선 구간에서 에러 복구가 가장 빠르고 효율적인 SACK의 장점을 무선 구간에 적용한 것이다.

SNACK 메커니즘은 효율적인 지역 재전송을 위해 BS의 SNACK-Snoop과 MH의 SNACK-TCP라는 두 개의 프로토콜 컴포넌트를 제안한다. FH에서 MH로 데이타 패킷을 전송한 경우 MH의 SNACK-Snoop은 다수의 손실된 패킷을 파악하여 SNACK 옵션 정보를 통해 BS로 전달하며, 이에 BS의 SNACK-Snoop은 무선 구간의 패킷 손실을 파악하고 빠른 재전송을 수행한다. 반대로 MH에서 FH로의 데이타 패킷을 전송한 경우 BS의 SNACK-Snoop은 FH에서 패킷을 전송한 경우의 SNACK-TCP 기능을 수행하며, MH의 SNACK-TCP 또한 FH에서 패킷을 전송한 경우의 SNACK-Snoop의 기능을 수행하게 된다.

2.2.1 FH에서 MH로의 데이타 전송

FH에서 데이타 패킷을 전송한 경우 BS의 SNACK-Snoop은 수신되는 모든 패킷의 시퀀스 번호(Sequence number)를 확인하고 손실된 패킷의 보상을 위해 버퍼에 저장한다. Snoop 프로토콜과 같이 이러한 시퀀스 번호의 확인과정은 BS가 패킷 손실의 원인이 네트워크의 혼잡 때문인지 아니면 무선 구간의 전송 손실인지를 판단하는 기준이 된다. 반면에 MH의 SNACK-TCP는 연속적으로 발생되는 패킷의 손실을 시퀀스 번호의 확인을 통해 찾고, SNACK의 추가 옵션에 손실된 패킷의 시퀀스 번호를 추가하여 BS로 전송한다. 이러한 추가옵션의 정보를 담은 SNACK 패킷을 수신한 SNACK-Snoop은 SNACK 옵션 정보에 포함된 시퀀스 번호를 확인하여 손실된 다수의 패킷을 하나의 Local RTT 안에 재전송 한다. 또한 SNACK-Snoop은 기존의 Snoop 프로토콜과 같이 송신단의 불필요한 재전송을 막기 위해 전송 계층의 중복 ACK 패킷을 차단한다. 그림 3(a)



(a) FH→MH로 데이타 전송 (b) MH→FH로 데이타 전송

그림 3 SNACK 메커니즘의 에러 복구 과정

는 이러한 동작과정을 나타낸 것이다.

2.2.2 MH에서 FH로의 데이타 전송

MH에서 FH로 데이타 패킷을 전송한 경우 SNACK-Snoop은 수신되는 패킷을 BS의 버퍼에 저장하지 않는다. 단지, 무선 구간의 전송 손실 여부를 판단하기 위해 MH로부터 수신되는 패킷의 시퀀스 번호만 확인하여 저장한다. 수신되는 시퀀스 번호를 확인하여 무선 구간의 전송 손실이 발생했다고 판단되는 경우, BS의 SNACK-Snoop은 FH로부터 수신되는 ACK 패킷에 손실된 패킷의 시퀀스 번호를 추가하여 MH로 전송한다. 시퀀스 번호가 추가된 SNACK 패킷을 수신한 MH의 SNACK-TCP는 옵션 정보의 시퀀스 번호를 확인하여 손실된 모든 패킷을 하나의 Local RTT안에 재전송한다. 또한 무선 구간의 손실에 대한 재전송이므로 MH의 불필요한 재전송이 발생하는 것을 방지한다.

그림 3(b)는 SNACK 메커니즘의 MH에서 데이타를 전송하는 경우의 에러 복구 과정을 나타낸 것이다. 앞서 설명한 바와 같이 BS로부터 SNACK 패킷을 수신한 MH는 하나의 Local RTT안에 손실된 패킷 모두를 재전송 한다. 이러한 SNACK 메커니즘의 지역 재전송 기법은 에러 복구과정에서 발생되는 지연시간을 줄이게 되어 송신단의 TCP에서 발생하는 불필요한 혼잡제어 메커니즘의 수행을 방지할 수 있게 된다.

2.3 SNACK 메커니즘의 문제점

SNACK 메커니즘은 2.2절에서 언급된 장점들로 인해 기존의 Snoop 프로토콜보다 상대적으로 뛰어난 지역 재전송을 수행한다. 그러나 SNACK 메커니즘 또한 Snoop 프로토콜과 같이 무선구간의 연집적인 패킷 손실을 효율적으로 복구하는데 있어서 몇 가지 문제점을 가지고 있다. 이러한 문제점은 크게 두 가지로 나눌 수 있다.

첫 번째는 기존의 Snoop 프로토콜과 같이 전송 계층의 ACK에 의한 재전송 기법이라는 점이다. Snoop 프로토콜은 전송 계층의 중복 ACK에 의해 손실된 패킷을 재전송 하고 재전송된 패킷에 대해서만 지역 재전송 타이머를 두어 관리한다. SNACK 메커니즘도 무선 구간에서 수정된 ACK 패킷을 사용하여 재전송하는 것 외에는 Snoop 프로토콜과 유사한 재전송 정책을 따른다. 타이머를 재전송한 패킷에 대해서만 적용하는 이유는 3.4절에서 자세히 설명한다. 이러한 재전송 메커니즘은 전송 손실률이 높고, 패킷의 손실이 연집적으로 발생되는 무선구간에서 ACK 패킷의 손실로 인해 심각한 전송 성능 저하를 가져오게 된다. 또한, FH에서 데이타를 전송한 경우에 비해 MH에서 데이타를 전송한 경우, 상대적으로 ACK 패킷의 수신에 의한 재전송 지연이 길어지게 됨으로 FH에서 데이타를 전송하는 경우만큼 좋은 성능을 기대하기는 어렵다[11].

두 번째는 무선구간에서 추가 옵션을 사용한 SNACK 패킷의 전송이다. SNACK 패킷과 같은 수정된 ACK 패킷의 사용은 기존에 널리 사용되는 전송 표준의 수정이 불가피 하며, 무선 구간에서 추가 옵션의 사용은 무선망의 대역폭과 이동 단말의 한정된 에너지 자원을 불필요하게 낭비하는 문제를 초래하게 된다. 무선망을 구성하는 이동 단말이 배터리 전원에 의존하여 동작하기 때문에, 이동 단말의 에너지 효율은 무선망에서 데이타 통신을 하는데 있어서 매우 중요한 고려 사항 중에 하나이다.

이러한 SNACK 메커니즘의 문제점들은 무선 구간의 연집적인 패킷 손실을 효율적으로 복구하는데 있어서 한계를 가진다. 따라서 전송 계층의 중복 ACK 손실과 전송 방향에 따른 성능 저하를 완화시키고, 이동 단말의 한정된 에너지와 무선구간의 제한적인 가용대역폭을 효율적으로 사용할 수 있는 새로운 지역 재전송 메커니즘이 필요하다.

3. C-Snoop 프로토콜

본 장에서는 2장에서 지적한 SNACK 메커니즘의 문제점을 개선하기 위해 새롭게 제안한 C-Snoop(Cross-layer Snoop) 프로토콜의 알고리즘에 대해 기술한다. C-Snoop 프로토콜은 SNACK 메커니즘이 갖는 문제점을 해결하기위해 IEEE 802.11 MAC 프로토콜을 기반으로 하는 유·무선 혼합망에서 BS와 MH에 Cross-layer 기법을 적용한 새로운 재전송 프로토콜이다[12]. 새롭게 제안된 C-Snoop 프로토콜은 무선 구간의 추가 옵션사용을 제거하여 이동 단말의 에너지 효율을 향상시켰으며, 무선 구간의 상태를 고려한 효율적인 재전송을 통해 무선구간의 전송률 또한 향상시켰다.

3.1 C-Snoop의 시스템 아키텍처

현재 IEEE 802.11 MAC 프로토콜은 유·무선 혼합망에서 가장 보편적으로 사용되는 전송 프로토콜이다. IEEE 802.11 MAC 프로토콜은 데이타 전송의 신뢰성

을 보장하기 위해 MAC-ACK를 사용한다. C-Snoop은 IEEE 802.11 프로토콜을 사용하는 MAC 계층과 IP 계층의 연동을 통해 유·무선 혼합망에서 효율적인 지역 재전송 기법을 수행하는 프로토콜이다. 효율적인 재전송을 위해 C-Snoop은 손실된 패킷과 같은 목적지를 갖는 데이타 패킷에 대한 MAC-ACK 패킷을 수신 하였거나, 새롭게 제안된 재전송 타이머가 만료되었을 경우에 손실된 패킷을 보상한다.

그림 4는 Cross-layer 기법이 적용된 BS와 MH의 시스템 아키텍처를 나타낸 것이다. 기존의 재전송 기법의 문제점을 개선하기 위해 그림 4에서 보는 바와 같이 각 시스템에 재전송 기능을 담당하는 C-Snoop Module (IP 계층)과 MAC-ACK의 수신 정보를 통보하는 C-Snoop Interface(MAC 계층)를 추가하였다. 그림 4와 같은 시스템 아키텍처가 가지는 장점은 기존의 지역 재전송 기법과 달리 표준화된 전송 메커니즘의 수정이 불필요하며, 손실된 패킷을 복구하는 과정에서 BS와 MH가 자신의 MAC 계층 정보를 이용하여 독립적으로 재전송을 수행함으로써 효율적인 전송 성능향상을 기대할 수 있다. C-Snoop 프로토콜의 독립적인 재전송 기법은 전송 방향에 따른 성능 향상의 불균형을 제거하며, 어느 한쪽이 C-Snoop 프로토콜을 지원하지 않아도 C-Snoop을 지원하는 방향에서 전송이 이루어질 경우 높은 전송 성능 향상을 기대할 수 있다.

C-Snoop은 기존의 Snoop 프로토콜, SNACK 메커니즘 등과 같이 지역 재전송을 사용하여 성능향상을 한다는 점에서 기본적인 동작 방법은 유사하다. 따라서 본 논문에서는 새롭게 C-Snoop에서 제안하는 기법에 대해서만 기술하고자 한다. 그림 5는 C-Snoop의 기본적인 MAC-ACK 기반의 재전송 과정을 나타낸 것이다. 그림 5

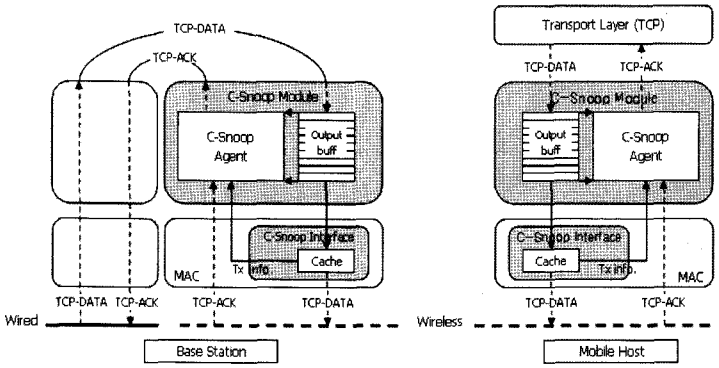


그림 4 C-Snoop 프로토콜의 시스템 아키텍처

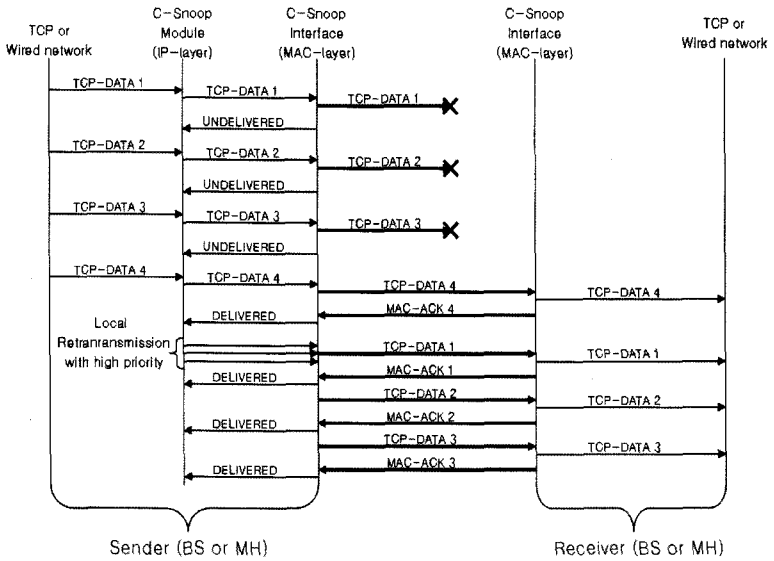


그림 5 C-Snoop의 MAC-ACK 기반의 에러 복구 과정

와 같이 C-Snoop agent는 상위 계층 또는 유선구간으로부터 수신된 TCP 데이타 패킷을 확인하고, MAC 계층의 C-Snoop Interface와 상호 연동을 통해 지역 재전송을 수행한다.

3.2 MAC 계층과의 상호 작용

IEEE 802.11 MAC 프로토콜은 기본적으로 무선 구간에서 신뢰할 수 있는 데이타 전송을 보장하기 위해서 ACK 패킷을 사용한다. 따라서 무선 구간의 전송 손실은 MAC 계층에서 가장 빠르게 판단할 수 있다. C-Snoop Interface는 이러한 MAC-ACK 패킷의 수신여부를 이용하여 무선구간의 전송 성공과 실패를 빠르게 판단하고, DELIVERED 또는 UNDELIVERED 이벤트를 발생시켜 패킷의 전송여부에 대한 정보를 C-Snoop agent로 통보하여 빠른 지역 재전송을 수행하게 한다. C-Snoop Interface에서 발생 시키는 두 가지의 이벤트에 대한 설명은 다음과 같다.

- DELIVERED 이벤트 : 전송을 성공한 경우에 발생되는 이벤트로서, 수신단으로부터 전송한 패킷에 대한 MAC-ACK 패킷을 수신한 경우 발생된다.
- UNDELIVERED 이벤트 : MAC 계층에서 해당 패킷을 전송하지 못하였을 경우 또는 전송한 패킷에 대한 MAC-ACK 패킷을 수신하지 못하여 타임아웃(Time-out)이 발생 하였을 때 C-Snoop agent에게 통보하는 이벤트이다.

3.3 상위 계층 또는 유선구간과의 상호 작용

C-Snoop agent는 기존의 Snoop agent와 같이 전송 계층 또는 유선구간으로부터 수신되는 패킷을 조사하여 재전송 타이머를 위한 큐잉 지연시간을 포함한 필요한 모든 정보를 저장한다. C-Snoop agent는 하위 계층으로 전달되기 직전에 전송할 패킷을 저장한다. 패킷이 하위 계층으로 전달된 이후에 C-Snoop agent는 MAC 계층으로부터 해당 패킷에 대한 전송 상태관련 이벤트가 발생되기를 기다린다. 만약 DELIVERED 이벤트가 MAC 계층으로부터 발생된 경우 C-Snoop agent는 해당 패킷을 버퍼에서 삭제하고, 무선 구간의 전송 지연시간을 계산하여 지역 재전송 타이머 값을 갱신한다. 또한 이전의 전송 실패로 인해 저장한 패킷이 같은 목적지를 가지고 있을 경우, 기존의 지역 재전송 메커니즘과 같이 높은 우선순위를 가지고 손실된 패킷을 재전송 한다. 전송한 패킷에 대해 C-Snoop Interface로부터 UNDELIVERED 이벤트가 발생한 경우에는 재전송을 위해 손실된 패킷을 재전송 버퍼에 저장하고 지역 재전송 타이머로 저장된 패킷을 관리한다.

3.4 C-Snoop 프로토콜의 지역 재전송 타이머

Snoop 프로토콜을 포함한 기존의 지역 재전송 메커니즘은 중복 ACK에 의한 재전송 이외에 지역 재전송 타이머를 두어 손실된 패킷을 관리한다. 기존의 지역 재전송 메커니즘은 전송 계층의 ACK 패킷을 기반으로 전송한 패킷의 전송 성공 여부를 판단하기 때문에 무선 구간의 전송 지연이 길어질 경우, 이미 MH에서 수신한 패킷에 대해 지역 재전송 타이머가 만료되어 불필요한 재전송을 수행하는 문제점이 발생할 수 있다. 이는 여러 사용자에 의해 공유되는 무선 채널의 효율성을 저하시켜, 무선 채널의 전체적인 전송 성능을 저하시킨다. 이러한 문제를 완화하기 위해서 Snoop 프로토콜은 재전송한 패킷에 대해서만 지역 재전송 타이머로 관리한다 [2,13]. 하지만 Snoop 프로토콜의 이러한 지역 재전송 타이머는 송신단의 전송 윈도우 크기가 작은 경우, 전송된 데이타 패킷과 ACK 패킷이 연속해서 손실될 경우 심각한 성능 저하를 발생 시킨다. 그림 6은 송신단의 전송 윈도우 크기가 작은 상황에서 연속된 패킷 손실이 발생하는 경우, Snoop 프로토콜의 지역 재전송 타이머 관리기법으로 인해 생기는 문제점을 나타낸 것이다.

그림 6은 FH의 전송 윈도우 크기가 4인 상황이며, 전송한 모든 패킷이 무선 구간에서 손실된 상황을 나타낸 것이다. 이러한 경우에 기존의 재전송 메커니즘은 전송 계층의 ACK수신이 불가능함으로 어떠한 재전송도 수행하지 않으며, FH의 타임아웃을 유발하여 전체 적인 전송 성능을 저하시킨다.

C-Snoop 프로토콜에서는 C-Snoop Interface가 BS 또는 MH로 데이타 패킷을 전송하지 못하였을 경우 발생시키는 UNDELIVERED 이벤트를 C-Snoop agent가 수신하였을 때 지역 재전송 타이머를 적용하기 때문에 불필요한 재전송이 발생하지 않는다. 제안하는 C-Snoop 프로토콜의 지역 재전송 타이머는 잦은 타이머 이벤트의 발생으로 인한 무선 채널의 성능 저하에 의한 문제를 완화하고, 송신단의 타임아웃 발생 이전에 해당 패킷을 재전송하기 위해 사용한다.

그림 7은 C-Snoop 프로토콜의 지역 재전송 타이머를 산출하기 위해 사용되는 큐잉 지연시간과 전송 지연시간을 나타낸 것이다. MH 또는 BS의 C-Snoop Module

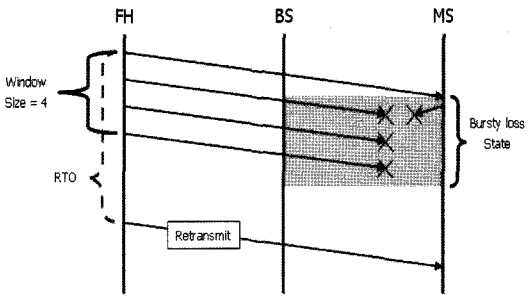


그림 6 기존 지역 재전송 타이머의 문제점

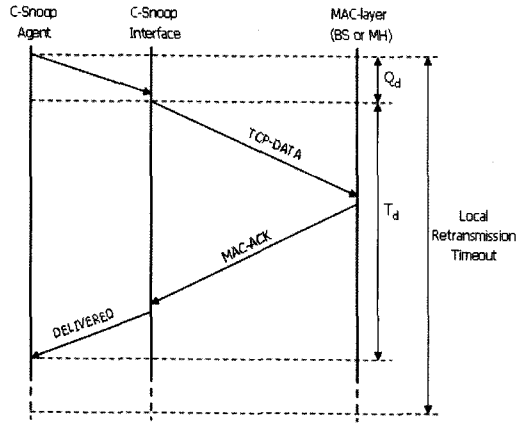


그림 7 C-Snoop의 Local RTT

에서 상위 계층 또는 유선 구간으로부터 수신되는 패킷의 큐잉 지연시간(Qd)과 전송을 수행하여 DELIVERED 이벤트가 수신되는 전송 지연시간(Td)의 합을 무선 구간의 RTT로 정한다.

이렇게 측정된 무선 구간의 RTT(WRTT)를 이용해서 식 (1)과 같이 현재 전송 계층에서 널리 사용되는 RTO(Retransmission Timeout)값의 산출방법과 동일하게 지역 재전송 타이머(*Local retransmission timer*)를 산출한다. a와 β의 값도 전송계층의 재전송 파라미터와 동일하게 0.125, 0.25로 한다. 산출된 지역 재전송 타이머는 무선 구간에서 손실된 재전송을 위해 저장된 각각의 패킷에 적용되며, MAC-ACK의 수신없이 타이머가 만료되었을 때 재전송을 수행한다.

$$W_{RTT} = Q_d + T_d$$
$$S_{WRTT} = (1-a)\times SW_{RTT} + a\times W_{RTT}$$
$$R_{WRTT} = (1-\beta)\times RW_{RTT} + \beta\times |SW_{RTT} - W_{RTT}| \quad (1)$$
$$Local\ retransmission\ timer = SW_{RTT} + 4RW_{RTT}$$

식 (1)과 같이 산출된 C-Snoop 프로토콜의 지역 재전송 타이머는 전송에 실패한 패킷에 대해서만 적용하므로 불필요한 지역 재전송이 발생하지 않으며, RTO의 산출방법과 동일하게 무선 구간의 지역 RTT를 기반으로 산출되므로 송신단의 타임아웃이 발생하기 이전에 해당 패킷을 재전송 한다. 그림 8은 송신단의 전송 윈도우 크기가 4인 상황에서 전송한 모든 데이타 패킷이 손실 되었을 경우, C-Snoop 프로토콜의 지역 재전송 타이머에 의한 재전송 과정을 나타낸 것이다.

그림 8에 나타낸 것처럼 C-Snoop은 무선 구간의 패킷 손실을 바로 알게 되고, 손실된 패킷을 지역 재전송 타이머를 두어 관리 하므로 ACK 패킷의 손실이나 전송 윈도우 크기가 작은 상황에서 성능 저하에 큰 영향을 받지 않는다. 또한, 기존 RTO의 산출방법을 사용함으로 잦은 타이머 이벤트의 발생으로 인한 무선 채널의 성능 저하에 의한 문제를 완화 하고, 송신단의 타임아웃을 발생시키지 않는다.

4. 실험 및 성능 평가

본 장에서는 새로 제안한 C-Snoop 프로토콜의 성능 평가를 위해 무선 구간의 연집적인 패킷 손실이 발생하는 유·무선 혼합망에서 전송 성능 및 이동 단말의 에너지 효율성 실험을 LBNL(Lawrence Berkely National

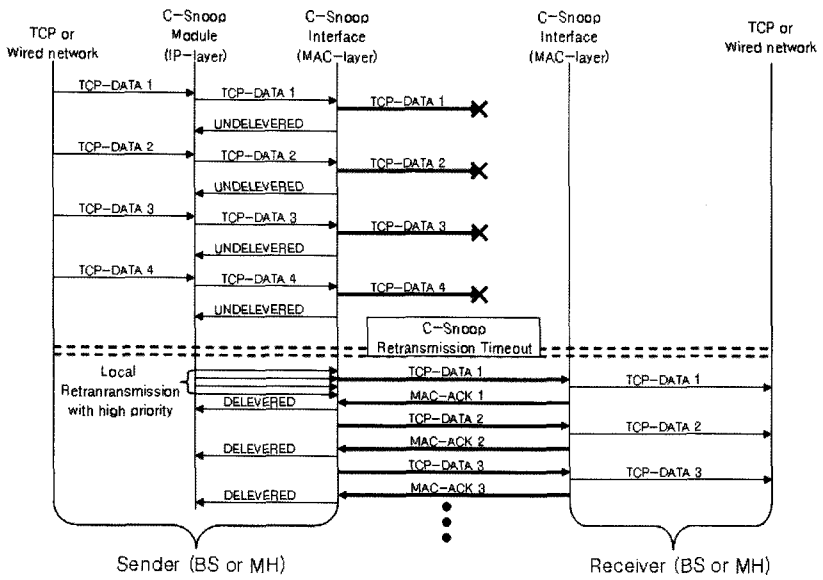


그림 8 C-Snoop의 재전송 타이머에 의한 에러 복구 과정

Laboratory)의 ns-2(network simulator)를 사용하여 실험을 수행하였고, 기존의 지역 재전송 메커니즘인 Snoop, SNACK 메커니즘과 비교, 평가하였다[14].

유·무선 혼합망은 신뢰성이 상대적으로 낮은 전송 매체인 무선 채널을 포함하고 있어 전송 에러에 의한 패킷 손실이 빈번히 발생한다. 따라서 본 절에서는 전송 에러에 의해 패킷 손실이 발생하는 무선 채널을 가진 유·무선 혼합망에서 C-Snoop 프로토콜의 성능 평가를 위해 Snoop, SNACK 메커니즘과 전송 성능 및 이동 단말의 에너지 효율을 비교하였으며, 결과를 통해 C-Snoop 프로토콜이 기존의 연구에 비해 유·무선 혼합망에서 더 효율적으로 동작한다는 것을 확인하였다.

4.1 실험 환경

유·무선 혼합망에서 제안한 C-Snoop 프로토콜의 성능을 SNACK, Snoop과 평가하기 위해 기존의 연구에서 사용되었던 실험 환경과 유사한 환경으로 그림 9와 같이 구성하였다. 유·무선 혼합망의 무선 채널에서 발생하는 연집적인 전송 에러에 따른 전송 성능 및 이동 단말의 에너지 효율성을 평가하기 위하여 이동 단말의 이동성을 배제한 환경으로 IEEE 802.11b MAC을 사용한 무선 링크를 설정하였다. 실험을 위한 파라미터 값은 표 1과 같다. 여기서, Slot은 MAC 계층의 슬롯 하나의 시간을 나타낸 것이며, SIFS와 DIFS는 패킷의 크기에 따라 적용되는 두 패킷간의 간격을 표현한 것이다.

성능 평가를 위해 사용되는 연집적인 패킷 손실률은 전체 전송한 패킷의 수에서 연속적으로 손실되는 패킷의 수를 비율로 나타낸 것으로 0%~10%로 각각 나누어 실험을 하며, 총 60초 동안 1Kbytes 크기의 패킷을 계속적으로 전송한다. 유선 구간의 혼잡에 의한 패킷 손실을 방지하기 위해 경쟁하는 트래픽을 배제하였다. 또한 이동 단말의 에너지 효율을 실험하기 위해 송신하는 이동 단말의 초기 에너지는 100J(Joule)로 설정하였으며, 각각 패킷 전송 시 0.6W(Watt)를 소비하며 ACK 수신시 0.3W를 소비하도록 설정 하였다. 단, SNACK 메커니즘의 경우 무선 구간의 패킷 손실이 발생하였을 때 ACK 패킷에 추가 데이타를 전송하므로 패킷의 크기에 송·수신 에너지 소모율이 비례하도록 설정하였다.

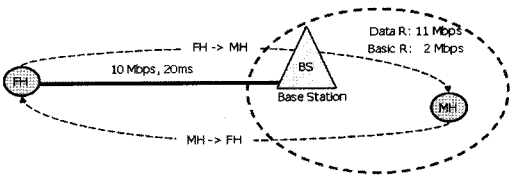


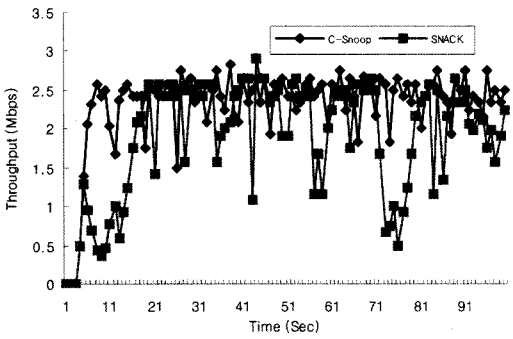
그림 9 실험 환경

표 1 실험 환경을 위한 파라미터 설정 정보

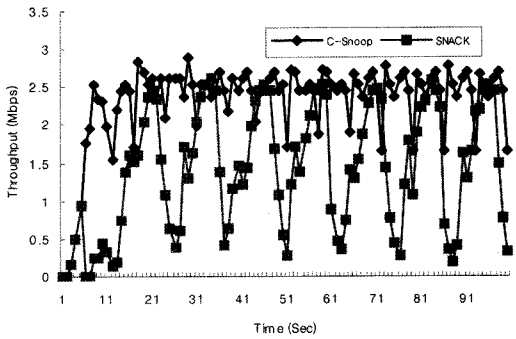
파라미터	값
Slot	20μs
SIFS(Short Interfram Space)	10μs
DIFS(DCF Interframe Space)	50μs
PLCP Preamble +header	96μs
Data Rate	11$Mbps$
Basic Rate	2$Mbps$
TCP Mechanism	TCP-Reno

4.2 전송 방향에 따른 전송률 실험

그림 10과 11은 무선 구간의 연집적인 패킷 손실률이 5%로 두어 100ms동안 두 가지 전송방향(FH↔MH)에 대해 C-Snoop과 SNACK 메커니즘의 전송률과 시퀀스 번호 증가율에 대한 실험 결과이다. 그림 10과 11에서 알 수 있듯이 SNACK 메커니즘에 비해 빠른 지역 재전송을 수행하고 무선 구간의 연속된 패킷 손실 구간에서 ACK 패킷의 손실에 영향을 받지 않는 C-Snoop 프로토콜의 성능이 전반적으로 높은 전송율과 시퀀스 번호의 증가를 보이는 것을 실험을 통해 확인할 수 있다. 또한, SNACK 메커니즘은 FH에서 MH로의 전송 방향에 비해 MH에서 FH로의 전송 방향의 성능이 저하되지

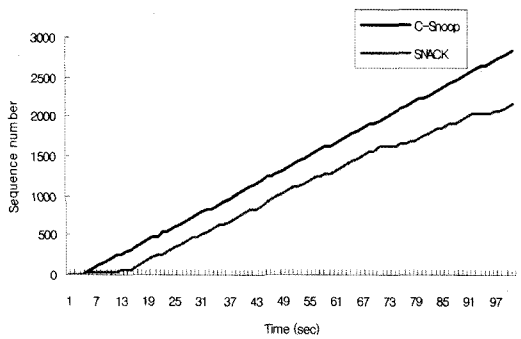


(a) FH→MH로 데이타 전송

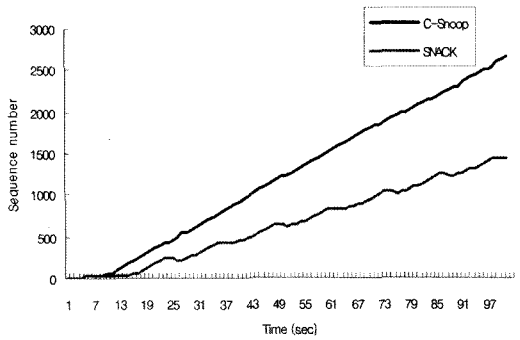


(b) MH→FH로 데이타 전송

그림 10 5%의 연집 손실률에 따른 전송률



(a) FH→MH로 데이타 전송



(b) MH→FH로 데이타 전송

그림 11 5%의 연집 손실률에 따른 시퀀스 번호 증가율

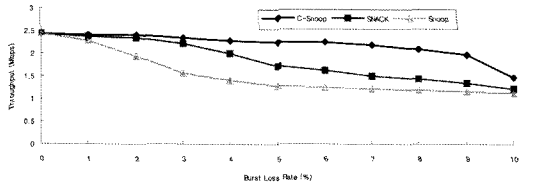


그림 12 패킷 손실률에 따른 평균 전송률 (FH→MH)

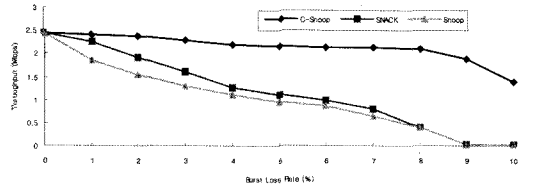


그림 13 패킷 손실률에 따른 평균 전송률 (MH→FH)

만, MAC 계층의 정보를 이용하여 빠르게 패킷의 손실 여부를 판단하는 C-Snoop 프로토콜은 전송 방향에 관계없이 고른 성능 향상을 보이는 것을 확인할 수 있다.

그림 12와 13은 무선 구간의 연집적인 패킷 손실 발생률을 0%~10%까지 변화 시켜가며 100초 동안 Snoop, SNACK과 C-Snoop 프로토콜의 평균 전송률을 두 가지 전송방향(FH↔MH)에 대해 실험한 결과다. 앞서 설명한 바와 같이 C-Snoop 프로토콜은 두 가지 전송 방향에 대해 고른 성능 향상을 보이는 것을 확인할 수 있다. 기존의 Snoop과 SNACK 같은 지역 재전송 메커니즘은 연집적인 패킷 손실률이 높아짐에 따라 한 전송윈도우 내의 대부분의 패킷이 손실되거나, 다수의 전송 계층의 ACK 패킷이 손실될 경우 심각한 성능 저하를 가져온다. 반면에 효율적인 지역 재전송 타이머를 두어 한 전송 윈도우 내에서 대부분의 패킷 손실이 발생하는 환경에서 큰 영향을 받지 않는 C-Snoop 프로토콜은 연집적인 패킷 손실율의 증가에 심각한 전송 성능 저하가 발생하지 않는 것을 확인할 수 있다.

4.3 MH의 에너지 효율성 실험

각 프로토콜의 에너지 효율성을 실험하기 위해 그림 9의 실험 환경에서 무선 구간의 연집적인 패킷 손실 발생률을 0%~10%까지 변화 시켜가며 100초 동안 수행하였다. 단, 에너지 효율성은 이동 단말인 MH에만 적용되므로 MH에서 FH로의 전송 방향에 대한 실험만 수행하였다. 식 (2)는 송신자가 전송한 데이타의 양과 그에 의해 소비된 에너지를 통해 이동 단말의 에너지 효율을 나타내는 식으로써 일정 에너지로 서비스할 수 있는 데이타의 양을 나타내게 된다. 식 (2)를 바탕으로 Snoop, SNACK 그리고 C-Snoop 프로토콜의 에너지 효율성을 평가한다.

$$Energy\ Efficiency(\eta) = \frac{Throughput}{Consumed\ Energy}\ (Kb/sJ) \quad (2)$$

그림 14는 Snoop, SNACK 그리고 C-Snoop 프로토콜의 연집적인 패킷 손실률에 따른 MH의 에너지 효율을 나타내는 결과 그래프이다. C-Snoop 프로토콜은 0%~10%의 연집적인 패킷 손실률 전체에서 Snoop과 SNACK 메커니즘 보다 좋은 성능을 보이게 된다. 또한 평균 에너지 효율은 C-Snoop 프로토콜의 경우 약 54Kb/sJ, SNACK 메커니즘의 경우 약 40Kb/sJ이며, Snoop 프로토콜의 경우 약 35Kb/sJ을 나타낸다. 패킷의 손실률이 5%일 때, Snoop, SNACK, C-Snoop의 전송률은 각

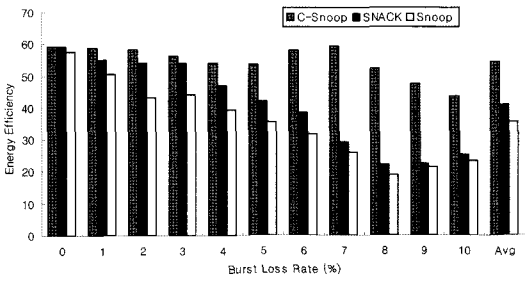


그림 14 패킷 손실률에 따른 에너지 효율

각 약 0.9Mbps, 1.2Mbps, 2.1Mbps이며, 이 때 소비된 에너지의 양은 각각 약 26J, 31J, 39J이다. Snoop 프로토콜은 소비된 에너지가 아주 작지만 전송한 데이타의 양이 작기 때문에 에너지 효율이 낮게 되는데 이는 곧 재전송에 의한 에너지 소비가 많았음을 의미하게 된다. SNACK 메커니즘과 C-Snoop 프로토콜은 Snoop 프로토콜에 비해 다소 많은 에너지를 소비했지만 전송한 데이타의 양이 SNACK 메커니즘에 비해 C-Snoop 프로토콜이 많기 때문에 더 좋은 에너지 효율을 보이게 된다. 이러한 결과는 C-Snoop 프로토콜이 MAC 계층의 정보를 이용하여 손실된 패킷을 빠르게 복구하고 연집적인 패킷 소실이 높은 환경에서 지역 재전송 타이머를 통해 효율적인 재전송을 수행하기 때문이다.

유·무선 혼합망에서 이동 단말의 에너지 효율은 시간당 전송한 데이타의 양과 이 때 소비된 에너지에 의해서 결정되게 된다. 시간당 전송한 데이타의 양이 많더라도 그 데이타를 전송하는 에너지의 소비가 많다면 상대적으로 에너지 효율이 낮게 되며, 시간당 전송한 데이타의 양이 적더라도 에너지의 소비가 적다면 상대적으로 에너지 효율이 높을 수도 있다. 이와 같은 관점에서 C-Snoop 프로토콜은 MAC 계층의 정보를 이용하여 망의 상태를 빠르게 판단하고, 망의 상태가 좋은 경우에 손실된 패킷을 빠르게 보상 하므로 불필요한 패킷의 재전송을 방지할 수 있어 소비되는 에너지의 양을 줄일 수 있었다. 또한 다수의 패킷이 무선 구간에서 연속 적으로 손실된 경우에도 지역 재전송 타이머에 의해 효율적으로 손실된 패킷을 보상 하게 된다. 따라서 C-Snoop 프로토콜은 높은 전송률을 유지하고 전송 에너지를 적게 소비하기 때문에 이동 단말의 에너지 효율 측면에서도 기존의 지역 재전송 메커니즘에 비해 높은 성능을 나타내게 된다.

5. 결론 및 향후 과제

기존의 지역 재전송 메커니즘인 Snoop 프로토콜과 SNACK 메커니즘은 유·무선 혼합망에서 무선 링크에서의 패킷 손실을 지역 재전송을 통해 보상하고 TCP 송신단에 알리지 않음으로써 TCP 전송률을 효과적으로 향상시켜주는 프로토콜이다. 하지만, 무선 링크의 연집적인 패킷 손실이 발생하는 경우에는 재전송 처리지연으로 인해 TCP의 전송률이 크게 저하된다는 문제점이 있다.

본 논문에서 제안한 C-Snoop 프로토콜은 이러한 지역 재전송 기법들의 단점을 보완하기 위해 Cross-layer 기반의 새로운 지역 재전송 기법을 제안하였다. C-Snoop 프로토콜은 전송 계층이 아닌 MAC-ACK 패킷을 이용하여 재전송을 수행함으로써 좀 더 빠른 재전송을 수행할 수 있을 뿐만 아니라, 지역 재전송 타이머 또한 효율적으로 손실된 패킷을 관리할 수 있도록 설계되어 기존의 지역 재전송 메커니즘들 보다 높은 TCP 전송률을 달성할 수 있었다. 또한 기존의 지역 재전송 기법과 달리 BS와 MH의 상호 의존성을 제거하여 FH에서 MH로의 전송뿐만 아니라, MH에서 FH로의 전송에서도 높은 성능 향상을 가지는 것을 확인 할 수 있었다. 이러한 상호 의존성 제거는 BS의 어떠한 도움없이 MH 만으로도 MH에 FH로 전송할 때 높은 성능 향상을 가져올 수 있으며, 반대 방향의 전송에서도 MH의 도움없이 BS만으로 무선 구간의 연집적인 패킷 손실을 효율적으로 보상하게 된다.

컴퓨터 시뮬레이션을 통해 유·무선 혼합망 환경에서 C-Snoop 프로토콜은 기존의 지역 재전송 메커니즘인 Snoop, SNACK 메커니즘보다 월등히 TCP 전송률을 높일 수 있음을 확인하였다. 또한 무선 구간의 연집적인 패킷 손실을 복구하는 과정에서 BS와 MH간의 의존적인 성향을 제거하여 전송 방향에 따른 성능 향상의 불균형성을 제거하였으며, MH의 에너지 효율성 또한 향상시켰다.

향후 연구과제로는 본 논문에서 제안하는 제어 방식을 실제로 구현하여 무선 LAN 등의 실제 환경에서 성능을 확인해 보아야 할 것이며 이를 위하여 핸드오프와 같은 상황에서 다른 곳으로 버퍼와 모듈의 상태를 넘겨주는 문제와 다양한 TCP 기반의 응용프로그램들에 대하여서도 적응력 있는 제어모듈이 될 수 있도록 하는 패킷 처리 알고리즘에 대한 집중적인 연구가 요구된다.

참 고 문 헌

[1] V. Tsaoussidis and I. Matta, "Open Issues on TCP for Mobile Computing," The Journal of Wireless Communications and Mobile Computing, pp. 3-20, February 2002.

[2] H. Balakrishnan, S. Seshan, and R.H Katz. "Improving Reliable Transport and Handoff Performance in Cellular Wireless Networks," ACM Wireless Networks, pp. 469-481, December 1995.

[3] A. Bakre and B. Badrinath, "I-TCP: Indirect TCP for Mobile Hosts," Proceedings of International Conference on Distributed Computing System, 1995.

[4] K. Brown and S. Singh, "M-TCP: TCP for Mobile Cellular Networks," Proceedings of ACM SIGCOMM Computer Communication Review, pp. 19-43, October 1997.

[5] L. Brakmo and L. Peterson, "TCP Vegas: End-to-end Congestion Avoidance on a Global Internet," IEEE Communication, pp. 1465-1480, October 1995.

[6] C. Peng Fu and S. Liew, "TCP Veno: TCP En-

hancement for Transmission over Wireless Access Networks," IEEE Journal of Selected Areas in Communications, pp. 216-218, February 2003.
[7] H. Balakrishnan, V. Padmanabhan, S. Seshan, and R. H. Katz, "A Comparison of Mechanisms for Improving TCP Performance over Wireless Links," Proceedings of ACM Transactions on Networking, pp. 756-769, December 1997.
[8] M. Mathis, J. Mandavi, S. Floyd, and A. Romanov, "TCP Acknowledgement option," IETF RFC 2018, 1996.
[9] S. Vagala and M. Vehicular, "The TCP SACK-aware Snoop Protocol for TCP over Wireless Networks," Proceedings of Vehicular Technology Conference, pp. 2624-2628, October 2003.
[10] F. Sun, Li. V.O.K. and S. Liew, "Design of SNACK Mechanism for Wireless TCP with New Snoop," Proceedings of IEEE Wireless Communications and Networking Conference, pp. 1051-1056, March 2004.
[11] G. Xylomenos and G. Polyzos, "Quality of Service Issues in Multi-service Wireless Internet Links," Proceedings of The International Workshop on QoS in Multi-service IP Networks, pp. 347-364, 2001.
[12] V. Raisinghani and S. Iyer, "ECLAIR: An Efficient Cross Layer Architecture for Wireless Protocol Stacks," Fifth World Wireless Congress, May 2004.
[13] K. Ratnam and I. Matta, "WTCP: An Efficient Mechanism for Improving TCP Performance over Wireless Links," Proceedings of Third IEEE symposium on Computer and Communications, pp. 74-78, June 1998.
[14] The network simulator ns-2, http://www.isi.edu/nanam/ns/

김 재 훈
2005년 광운대학교 전자공학부 학사. 2005년~현재 광운대학교 전자통신공학과 석사 과정. 관심분야는 인터넷 QoS, 유비쿼터스 컴퓨팅, Ad-hoc & Wireless 통신망

정 광 수
1981년 한양대학교 전자공학과 학사. 1983년 한국과학기술원 전기 및 전자공학과 석사. 1991년 미국 University of Florida 전기공학과 박사(컴퓨터공학전공). 1983년~1993년 한국전자통신연구원 선임연구원 1991년~1992년 한국과학기술원 대우 교수. 1993년~현재 광운대학교 전자공학부 교수(정보통신 연구원)