

## 무선 랜 환경에서 Cross-Layer기반의 TCP 성능향상 기법

김제훈<sup>o</sup>, 이선현, 최웅철, 이승형, 정광수

광운대학교 전자공학부 컴퓨터통신연구실

(jhkim<sup>o</sup>, sunlee}@adams.kw.ac.kr, {wchoi, shrlee}@daisy.kw.ac.kr, kchung@kw.ac.kr

### A Crosslayer approach to Improving TCP Performance in Wireless LANs

Jaehoon Kim<sup>o</sup>, Sunhun Lee, WoongChul Choi, Seung Hyong Rhee, Kwangsue Chung  
School of Electronics Engineering, Kwangwoon Univ.

#### 요 약

Snoop 프로토콜은 무선 랜 환경에서 발생하는 TCP 패킷 손실을 효과적으로 보상하여 TCP 전송률을 향상시킬 수 있는 효율적인 프로토콜이다. 하지만, 무선 링크에서 Burst loss가 발생하는 경우에는 지역 재전송을 효과적으로 수행하지 못하여 전송 효율이 떨어진다는 문제점이 있다. 본 논문에서는 Snoop 프로토콜의 이러한 문제점을 개선하기 위해 cross layer 기법을 적용한 지역 재전송 기법인 A<sup>2</sup>Snoop (ARQ Assistance Snoop) 프로토콜을 제안한다. A<sup>2</sup>Snoop 프로토콜은 현재 무선 랜 환경에서 가장 널리 사용되는 IEEE 802.11 MAC 프로토콜 기반의 지역 재전송 매커니즘으로서, MAC 계층의 ARQ 기법의 메시지와 새로이 제안된 지역 재전송 타이머에 의해 효율적인 재전송을 수행한다. ns-2 시뮬레이터를 이용한 실험을 통해 A<sup>2</sup>Snoop의 지역 재전송 기법은 무선 구간의 Burst loss에 대해 효율적인 보상을 수행하며, 이동 단말의 에너지 효율성을 향상시키는 것을 확인할 수 있다.

#### 1. 서 론

TCP는 지난 20년을 통틀어 가장 성공적인 전송 계층 프로토콜이다. 이러한 TCP의 성공 요인으로서 신뢰성 있는 데이터 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 최적화 되도록 진화해온 TCP는 무선망이 가지는 불안정한 링크에 의한 데이터 손실을 유선망에서처럼 네트워크의 혼잡으로 인한 손실로 오해한다. 그 결과 혼잡 제어 메커니즘이 수행되어 전송율을 줄이므로 네트워크 성능이 저하되는 문제점을 초래 한다[1].

이러한 이유로 최근 몇 년간 TCP의 성능을 향상시키기 위한 연구가 활발히 이루어지고 있다. 무선 환경에서 TCP의 성능을 향상시키기 위한 방법은 크게 링크계층 매커니즘, 전송 계층 매커니즘으로 분류 할 수 있다. Snoop, Delayed Duplicate Acknowledgments, TULIP등이 링크계층 매커니즘 이며, I-TCP, M-TCP, ELN등이 전송계층의 매커니즘 이다.

위의 다양한 매커니즘 중에서 Snoop 프로토콜은 가장 좋은 성능을 보여준다[1]. 이는 무선구간의 Random Loss에 대해 BS(Base Station)가 빠른 응답을 보이기 때문이다. 그러나 무선구간의 손실은 Random loss가 아닌 Burst loss가 대부분이다. 따라서 각 Local-RTT(Round Trip Time)마다 하나의 손실된 패킷만을 복구하는 Snoop 프로토콜은 Burst loss의 복구에 \*본 연구는 한국 과학재단 특정기초연구[R01-2005-0000-10934-0(2006)]의 지원에 의해 수행되었음.

많은 한계를 가진다. 이러한 Snoop의 Burst loss 복구 한계를 효율적으로 해결하기 위해 무선구간에 SACK 옵션 또는 SACK과 유사한 추가옵션을 사용하는 방식이 제안되었다[2,3].

하지만 기존의 Snoop 프로토콜과 같이 전송계층의 중복 ACKs기반의 재전송을 사용한다는 점과 ACK 패킷에 SACK 옵션의 사용하는 점은 무선구간의 Burst Loss를 효율적으로 복구하는데 있어 많은 한계를 가진다.

본 논문에서는 현재 무선 랜 환경에서 가장 보편적으로 사용되는 IEEE 802.11 MAC 프로토콜을 기반으로 이러한 재전송 과정의 문제점 및 무선 단말의 에너지 효율을 개선하기 위해 새로운 Cross layer 기법을 제안한다.

본 논문은 다음과 같이 구성된다. 2장에서는 무선구간에 추가 옵션을 사용하는 기법인 SNACK(Selective Negative Acknowledgement) 프로토콜의 동작방법 과 문제점에 대해 기술하고, 3장에서는 본 논문이 제시하는 새로운 재전송 기법에 대해 설명한다. 4장에서는 제안한 재전송 기법의 성능을 평가하기 위한 실험과 5장에서는 결론 및 향후 과제에 대해 기술한다.

#### 2. 관련 연구

##### 2.1. SNACK 프로토콜

SNACK는 Snoop 프로토콜 매커니즘에서 Burst Loss 복구의 한계를 해결하기 위해 제안된 기법이다. 기본적인 동작 방법은 Snoop프로토콜과 유사하다. 단, 무선 구간의 Burst Loss를 빠르게 복구하기 위해 ACK 패킷의 헤더에 SACK과 유사

한 SNACK 옵션을 사용하여 한 Local-RTT안에 손실된 모든 패킷을 재전송 하는 방법을 사용한다. 그림 1은 SNACK 프로토콜의 MH(Mobile Host)에서 FH(Fixed Host)로 데이터 전송과 FH에서 MH로 데이터를 전송하는 과정을 그림으로 나타낸 것이다.

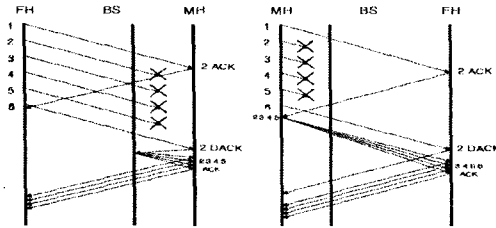


그림 1. SNACK 프로토콜의 에러 복구 과정

기존의 Snoop 프로토콜의 재전송 기법은 무선구간의 연속된 패킷 손실을 복구하는 과정에서 많은 Local-RTT를 소모한다. 이러한 재전송 과정의 많은 지연시간은 FH의 타임아웃을 발생시켜 심각한 성능 저하를 가져오는 원인이 된다. 하지만 그림 1과 같이 SNACK 프로토콜은 무선구간의 연속되는 패킷손실을 빠르게 복구하기위해 전송계층의 ACK에 추가 옵션을 사용한다. 따라서 하나의 Local-RTT에 손실된 모든 패킷을 복구하게 되고, FH의 타임아웃 발생을 방지할 수 있게 된다.

2.2 SNACK 프로토콜의 문제점

SNACK 프로토콜은 2.2절에서 언급한 장점들로 인해 기존의 Snoop 프로토콜보다 상대적으로 뛰어난 성능을 보인다. 그러나 Snoop 프로토콜과 같이 SNACK 프로토콜 또한 무선구간의 Burst Loss를 완벽히 복구하는데 있어서 몇 가지 문제점을 가지고 있다. 이러한 문제점은 크게 두 가지로 나눌 수 있다.

첫 번째는 기존의 Snoop 프로토콜과 같이 전송계층의 중복 ACK에 의한 재전송 기법이라는 점이다. 전송 손실율이 높은 무선구간에서 ACK 패킷의 손실로 인한 재전송 시간의 지연은 TCP의 타임아웃을 유발하여 성능 저하를 가져올 수 있다. 이러한 문제점은 상대적으로 TCP의 전송 윈도우의 크기가 작거나 전송방향이 MH에서 FH일 경우 더 큰 성능저하를 발생시킨다.

두 번째는 무선구간에서 SACK 옵션과 같은 추가 데이터 전송이다. 이러한 추가 옵션의 사용은 무선망의 대역폭과 무선 단말의 한정된 에너지 자원을 불필요하게 낭비하는 문제를 초래하게 된다.

이러한 SNACK 프로토콜의 문제점은 무선구간의 Burst Loss를 효율적으로 복구하는데 있어서 한계를 가진다. 따라서 전송계층의 중복 ACK 손실에 대한 성능 저하를 완화시킴과, 무선 단말의 한정된 에너지와 무선구간의 제한적인 가용대역폭을 효율적으로 사용할 수 있는 메커니즘이 필요하다.

3. A<sup>2</sup>Snoop(ARQ Assistance Snoop)

현재 IEEE 802.11 MAC 프로토콜은 무선 랜 환경에서 가장

보편적으로 사용되는 전송 프로토콜이다. IEEE 802.11 MAC 프로토콜은 데이터 전송의 신뢰성을 보장하기 위해 Stop & Wait 방식의 ARQ 기법을 이용한다. A<sup>2</sup>Snoop은 IEEE 802.11 MAC 프로토콜을 기반으로 하는 무선 랜 환경에서 BS와 MH에 Cross layer 기법을 적용한 새로운 지역 재전송 프로토콜이다[4].

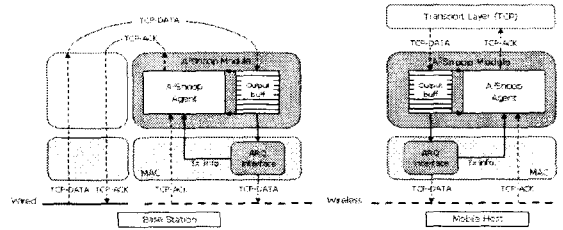


그림 2. A<sup>2</sup>Snoop 프로토콜의 시스템 아키텍처

그림 2는 Cross layer 기법이 적용된 BS와 MH의 시스템 아키텍처를 나타낸 것이다. 기존의 재전송 문제점을 개선하기 위해 그림 2에서 보는 바와 같이 각 시스템에 재전송 기능을 담당하는 A<sup>2</sup>Snoop agent(IP 계층)와 패킷의 전송 정보를 담당하는 ARQ interface를 추가하였다. 그림 2와 같은 시스템 아키텍처가 가지는 장점은 기존의 지역 재전송 기법과 달리 표준화된 전송 메커니즘의 수정이 불필요하며, 손실된 패킷을 복구하는 과정에서 BS와 MH가 자신의 ARQ 정보를 이용하여 독립적으로 재전송을 수행함으로써 효율적인 전송 성능향상을 기대할 수 있다. A<sup>2</sup>Snoop 프로토콜의 독립적인 재전송 기법은 전송 방향에 따른 성능 향상의 불균형을 제거한다[5]. 또한, 어느 한 쪽이 A<sup>2</sup>Snoop 프로토콜을 지원하지 않아도 A<sup>2</sup>Snoop을 지원하는 방향에서 전송이 이루어질 경우 높은 전송 성능 향상을 기대할 수 있다.

A<sup>2</sup>Snoop의 기본적인 동작 방법은 기존의 Snoop 프로토콜, SNACK 메커니즘 등과 같이 지역 재전송을 사용하여 성능 향상을 한다는 점에서 기본적인 동작 방법은 유사하다. 따라서 본 논문에서는 새롭게 A<sup>2</sup>Snoop에서 제안하는 기법인 계층 간의 상호작용과 재전송 타이머 산출방법에 대해서만 언급하고자 한다.

3.1 MAC 계층과의 상호 작용

IEEE 802.11 MAC 프로토콜은 기본적으로 무선 구간에서 신뢰할 수 있는 데이터 전송을 보장하기 위해서 MAC 계층의 ARQ 기법을 사용한다. MAC 계층은 무선구간의 손실을 가장 빠르게 판단할 수 있는 계층이다. 그러므로 A<sup>2</sup>Snoop은 패킷 전송관련 정보를 A<sup>2</sup>Snoop agent에 빠르게 통보하기 위해서 MAC 계층의 ARQ 기법에 ARQ interface를 추가 하였다. 패킷 전송관련 정보의 통보를 위해서 ARQ interface에 다음과 같은 2가지 이벤트를 아래와 같이 정의한다.

1) TX\_SUCCESS 이벤트: 이 이벤트는 목적지 노드로 패킷의 전송을 성공하였을 경우 발생된다. 목적지 노드에 의해서 MAC 계층의 ACK 프레임이 정확히 수신하였을 경우에 ARQ

interface에 의해 발생된다.

2) TX\_FAILED 이벤트: ARQ 기법에 미리 정의된 수만큼 재전송을 수행한 후 더 이상 패킷을 전송할 수 없을 경우에 발생된다. 즉, MAC 계층의 Timeout이 발생된 경우에 ARQ interface에 의해 발생되는 이벤트다.

3.2 상위 계층 또는 유선구간의과의 상호 작용

A<sup>2</sup>Snoop agent는 전송계층 또는 유선구간으로부터 수신되는 패킷을 조사하여 큐잉 지연시간을 포함한 필요한 모든 정보를 저장한다. 또한 하위계층으로 전달되기 직전에 전송할 패킷을 저장한다. 패킷이 하위 계층으로 전달된 이후에 A<sup>2</sup>Snoop agent는 ARQ interface로부터 전송관련 이벤트가 발생되기를 기다린다. 만약, TX\_SUCCESS 이벤트가 ARQ interface에 의해 발생된 경우 A<sup>2</sup>Snoop agent는 해당 패킷을 버퍼에서 삭제하고, 전송 지연시간을 계산하여 재전송 타이머 값을 갱신한다. 또한 이전의 전송 실패로 인해 저장한 패킷이 같은 목적지를 가지고 있을 경우, 가능한 빨리 손실된 패킷을 재전송 한다. 반대로 TX\_FAILED 이벤트가 발생한 경우 재전송을 위해 손실된 패킷을 저장하고 재전송 타이머로 손실된 패킷을 관리한다.

그림 3은 ARQ interface의 정보를 이용한 기본적인 재전송 과정을 나타낸 것이다.

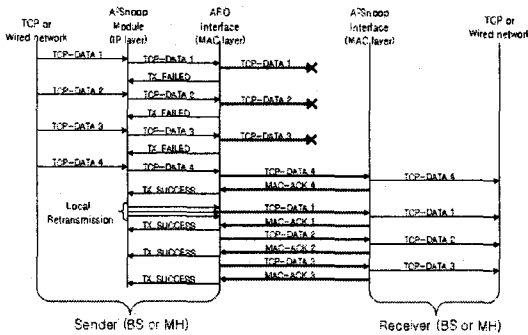


그림 3. A<sup>2</sup>Snoop의 ARQ interface 기반의 에러 복구 과정

3.3 재전송 타이머

IEEE 802.11은 CSMA/CA(Carrier Sense Multiple Access with Collision Avoidance)를 기본적인 매체 접근 메커니즘으로 사용한다. 또한, 하나의 추가 옵션으로써 MAC 계층의 데이터 패킷과 응답 패킷을 주고받기 전에 RTS/CTS(Request-To-Send/Clear-To-Send) 패킷을 교환한다. 이러한 매체 접근 방법은 무선구간에서 패킷 충돌을 최소화 하는 효율적인 방법으로 알려져 있다. 그림 4는 이러한 매체 접근 방법을 이용한 데이터 패킷의 전송과정을 나타낸 것이다.

T<sub>in</sub>에서 소스 노드는 데이터를 전송하기 위해서 매체 접근 메커니즘을 수행하며, 매체가 전송 범위 내의 다른 노드에 의해 점유되었음을 감지하고 Backoff 윈도우의 초기 크기와 함께 지수급수 백오프(Exponential backoff) 알고리즘을 수행한다. 그 후에 소스 노드는 매체가 다른 노드에 의해 점유되지 않음을 감지하고, 매체를 점유하기 위해 RTS 프레임용 목적지 노

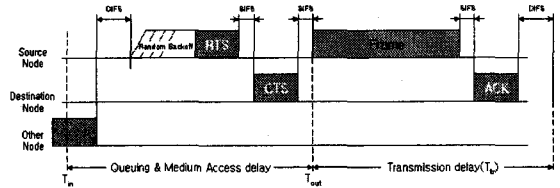


그림 5 IEEE 802.11의 매체 접근 및 데이터 전송 과정

드로 전송한다. RTS 프레임을 수신한 노드는 SIFS(Short Inter-Frame Space)의 시간 후에 자신의 전송범위 내에 있는 모든 노드의 NAV(Network Allocation Vector) 갱신을 위해 CTS 프레임으로 응답한다. T<sub>out</sub>에서 소스 노드는 데이터 패킷을 전송하며, 그 패킷을 정확히 수신한 목적지 노드는 ACK 패킷을 전송한다.

A<sup>2</sup>Snoop은 손실된 패킷을 전송계층의 Timeout 발생 이전에 손실된 패킷을 복구하기위해서 지역 재전송 타이머를 사용한다. 타이머를 산출하기위해 앞서 언급한 기본적인 매체 접근 및 데이터 패킷 전송 과정을 고려하여 무선 구간의 RTT(W<sub>RTT</sub>)를 식(1)과 같이 산출한다.

$$W_{RTT} = T_{in} - T_{out} + T_{tr} \tag{1}$$

T<sub>in</sub> - T<sub>out</sub>은 노드가 매체에 접근하기 위해서 지연되는 모든 시간을 포함한다. 이와 유사한 산출 방법은 Giuseppe Bianchi 에 의해 처음으로 표현되었다[6].

손실된 패킷을 효율적으로 복구하기 위해서 A<sup>2</sup>Snoop은 전송 계층에서 사용되는 RTO(Retransmission Timeout)의 계산 방법과 유사하게 W<sub>RTT</sub>를 기반으로 무선구간의 RTO(W<sub>RTO</sub>)를 산출한다. W<sub>RTO</sub>는 A<sup>2</sup>Snoop agent가 ARQ interface로부터 TX\_SUCCESS 이벤트를 수신한 경우 계속해서 새롭게 갱신되며, 산출 과정은 식 (2)와 같다.

$$\begin{aligned} SW_{RTT} &= (1 - \alpha) \times SW_{RTT} + \alpha \times W_{RTT} \\ RW_{RTT} &= (1 - \beta) \times RW_{RTT} + \beta \times |SW_{RTT} - W_{RTT}| \\ W_{RTO} &= SW_{RTT} + 4RW_{RTT} \end{aligned} \tag{2}$$

A<sup>2</sup>Snoop agent는 ARQ interface로부터 TX\_FAILED 이벤트를 수신한 경우에, 재전송을 위해 저장된 패킷에 타이머를 적용하여 관리한다. A<sup>2</sup>Snoop의 재전송 타이머는 FH의 Timeout의 발생 이전에 손실된 패킷을 복구하기 위해서 W<sub>RTO</sub>를 기반으로 식 (3)과 같이 산출한다.

$$Local\ Retransmission\ Timer = W_{RTO} - T_Q \tag{3}$$

T<sub>Q</sub>는 매체 접근 과정에서 기본적으로 발생하는 지연 시간 뿐만 아니라, MAC 계층에서 미리 정의 되어 사용되는 재전송 횟수(Retry limit)동안의 지연 시간을 포함한 값이다.

A<sup>2</sup>Snoop의 재전송 타이머는 무선구간의 RTT를 기반으로 하여 전송계층의 RTO 계산방법과 유사하게 산출되기 때문에

FH의 Timeout 이전에 손실된 패킷을 ARQ interface의 이벤트 발생 없이 복구할 수 있으며, 잦은 타이머 발생으로 생기는 시스템의 부하를 완화할 수 있는 장점을 가지고 있다.

4. 실험 및 성능 평가

본 장에서는 새로이 제안한 A<sup>2</sup>Snoop 프로토콜의 성능 평가를 위해 무선 구간의 Burst loss가 발생하는 무선 랜 환경에서 전송 성능 및 이동 단말의 에너지 효율성 실험을 LBNL(Lawrence Berkely National Laboratory)의 ns-2(network simulator)를 사용하여 수행하였고, 기존의 지역 재전송 메커니즘인 Snoop, SNACK 메커니즘과 비교, 평가 하였다 [7].

무선 랜 환경은 신뢰성이 상대적으로 낮은 전송 매체인 무선 채널로 구성되어있어 전송 에러에 의한 패킷 손실이 빈번히 발생한다. 따라서 본 절에서는 전송 에러에 의해 패킷 손실이 발생하는 무선 채널을 가진 환경에서 A<sup>2</sup>Snoop 프로토콜의 성능 평가를 위해 Snoop, SNACK 메커니즘과 전송 성능 및 이동 단말의 에너지 효율을 비교하였으며, 결과를 통해 A<sup>2</sup>Snoop 프로토콜이 기존의 연구에 비해 무선 랜 환경에서 더 효율적으로 동작한다는 것을 확인 하였다.

4.1. 실험 환경

무선 랜 환경에서 새롭게 제안한 A<sup>2</sup>Snoop 프로토콜의 성능을 평가하기 위해 그림 4과 같은 실험 환경을 구성하였다. 실험을 위한 파라미터 값은 표 1과 같다.

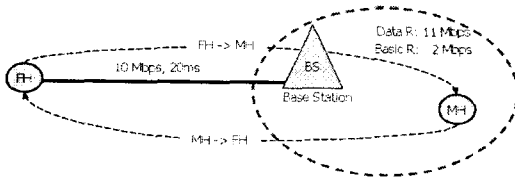


그림 4. 실험 환경

표 1. 파라미터 설정 정보

Parameter Name	Value
Slot	20μs
SIFS	10μs
DIFS	50μs
PLCP Preamble + header	96μs
Data Rate	11Mbps
Basic Rate	2Mbps

성능 평가를 위해 채널 오류에 의한 Burst loss rate를 0%~10%로 각각 나누어 실험 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 데이터를 전송

하므로 패킷의 크기에 송·수신 에너지 소모율이 비례하도록 설정하였다.

4.2. 전송 방향에 따른 전송률 실험

그림 5와 6은 무선 구간의 Burst loss rate를 5%로 두어 100ms동안 두 가지 전송 방향(FH→MH)에 대해 실험한 결과이다. 그림 5와 6에서 알 수 있듯이 SNACK 메커니즘에 비해 빠른 지역 재전송을 수행하고 무선 구간의 연속된 패킷 손실 구간에서 ACK 패킷의 손실에 영향을 받지 않는 A<sup>2</sup>Snoop 프로토콜의 성능이 전반적으로 높은 전송률을 보이는 것을 확인할 수 있다.

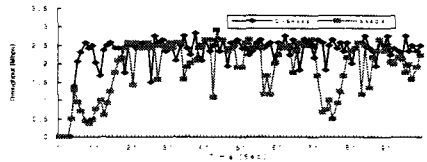


그림 5. 5% Burst loss에 따른 전송률(FH→MH)

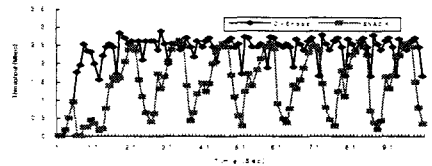


그림 6. 5% Burst loss에 따른 전송률(MH→FH)

또한, 그림 7과 8은 이러한 5%의 Burst loss rate에 따른 시퀀스 번호의 증가율을 나타내었다.

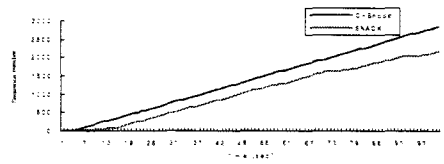


그림 7. 5% Burst loss에 따른 시퀀스 번호 증가율 (FH→MH)

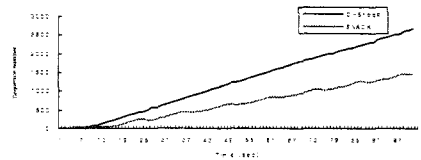


그림 8. 5% Burst loss에 따른 시퀀스 번호 증가율 (MH→FH)

그림 9와 10은 무선 구간의 Burst loss rate를 0%~10%까지

변화 시켜가며 100초 동안 Snoop, SNACK과 A<sup>2</sup>Snoop 프로토콜의 평균 전송률을 두 가지 전송 방향에 대해 실험한 결과다. 앞서 설명한 바와 같이 A<sup>2</sup>Snoop 프로토콜은 두 가지 전송 방향에 대해 고른 성능 향상을 보이는 것을 확인할 수 있다. 기존의 Snoop과 SNACK과 같은 지역 재전송 메커니즘은 Burst loss rate가 높아짐에 따라 한 전송 윈도우 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 A<sup>2</sup>Snoop 프로토콜은 Burst loss rate의 증가에 심각한 전송 성능 저하가 발생하지 않는 것을 확인할 수 있다.

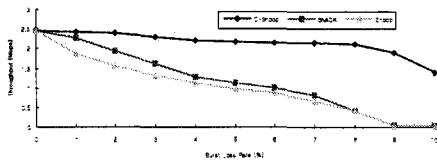


그림 10. 패킷 손실률에 따른 평균 전송률 (MH→FH)

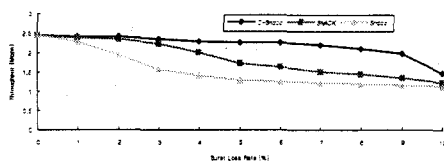


그림 9. 패킷 손실률에 따른 평균 전송률 (FH→MH)

### 4.3. MH의 에너지 효율성 실험

각 프로토콜의 에너지 효율성을 실험하기 위해 그림 4의 실험 환경에서 무선 구간의 Burst loss rate를 0%~10%까지 변화 시켜가며 100초 동안 수행하였다. 단, 에너지 효율성은 이동 단말인 MH에만 적용되므로 MH에서 FH로의 전송 방향에 대한 실험만 수행하였다. 식 (4)는 송신자가 전송한 데이터의 양과 그에 의해 소비된 에너지를 통해 이동 단말의 에너지 효율을 나타내는 식으로써 일정 에너지로 서비스할 수 있는 데이터의 양을 나타내게 된다. 식 (4)를 바탕으로 Snoop, SNACK 그리고 A<sup>2</sup>Snoop 프로토콜의 에너지 효율성을 평가한다.

$$EnergyEfficiency(\eta) = \frac{Throughput}{ConsumedEnergy} (Kb/s.J) \quad (4)$$

그림 11은 Snoop, SNACK 그리고 A<sup>2</sup>Snoop 프로토콜의 Burst loss rate에 따른 MH의 에너지 효율을 나타내는 결과 그래프이다. A<sup>2</sup>Snoop 프로토콜은 0%~10%의 Burst loss rate의 전체에서 Snoop과 SNACK 메커니즘 보다 좋은 성능을 보이게 된다.

무선 랜 환경에서 이동 단말의 에너지 효율은 시간당 전송한

데이터의 양과 이 때 소비된 에너지에 의해서 결정된다. 시간당 전송한 데이터의 양이 많더라도 그 데이터를 전송하는 에너지의 소비가 많다면 상대적으로 에너지 효율이 낮게 되며, 시간당 전송한 데이터의 양이 적더라도 에너지의 소비가 적다면 상대적으로 에너지 효율이 높을 수도 있다. 이와 같은 관점에서 A<sup>2</sup>Snoop 프로토콜은 ARQ interface의 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.

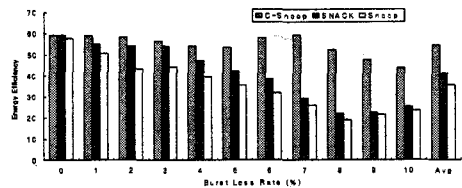


그림 11. 패킷 손실률에 따른 에너지 효율

### 5. 결론 및 향후 과제

본 논문에서는 Snoop과 SNACK이 가지는 문제점을 분석하고, 이를 해결하기 위해 MAC-ACK 기반의 재전송 기법을 제안하였다. 제안된 기법은 빠른 재전송을 수행하고, BS와 MH의 연관성을 제거하여 기존 프로토콜의 문제점을 해결하였다.

향후에는 기존 재전송 프로토콜의 또 다른 취약점인 BS의 버퍼관리 및 WPAN(Wireless Personal Area Network)과 같은 다른 네트워크 환경에 대해서도 고려할 것이다.

### 6. 참고문헌

- [1] H. Balakrishnan, V. Padmanabhan, S. Seshan, and R. H. Katz, "A Comparison of Mechanisms for Improving TCP Performance over Wireless Links," *IEEE/ACM Transactions on Networking*, December 1997.
- [2] Vangala, S. Labrador, M.A.Vehicular, "The TCP SACK-aware snoop protocol for TCP over wireless networks," *Technology Conference, 2003. VTC 2003-Fall. 2003.*
- [3] Fanglei Sun, Li, V.O.K., Liew, S.C., "Design of SNACK mechanism for wireless TCP with new snoop," *Wireless Communications and Networking Conference, 2004.*
- [4] V. Raisinghani and S. Iyer, "Cross-layer Design Optimization in Wireless Protocol Stack," *Computer Communications(Elsevier)*, May 2004.
- [5] 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.
- [6] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE Journal on Selected Areas in Communications*, March 2000.
- [7] The network simulator ns-2, <http://www.isi.edu/nanarn/ns/>