

논문 2006-01-05

CPC와 ZWSC를 이용한 무선 망에서의 TCP 성능 향상 방안 (TCP Performance Enhancement over the Wireless Networks by Using CPC and ZWSC)

이 명 섭, 박 영 민, 장 주 석, 박 창 현*

(Myung-Sub Lee, Young-Min Park, Joo-Seok Chang, Chang-Hyeon Park)

Abstract : With the original Transmission Control Protocol(TCP) design, which is particularly targeted at the wired networks, a packet loss is assumed to be caused by the network congestion. In the wireless environment where the chances to lose packets due to transmission bit errors are not negligible, though, this assumption may result in unnecessary TCP performance degradation. In these days, many papers describe about wireless-TCP which has suggested how to avoid congestion control when packet loss over the wireless network. In this paper, an enhancement scheme is proposed by modifying SNOOP scheme. To enhance the original SNOOP scheme, CPC(Consecutive Packet Control) and ZWSC(Zero Window Size Control) are added. The invocation of congestion control mechanism is now minimized by knowing the cause of packet loss. We use simulation to compare the overhead and the performance of the proposed schemes, and to show that the proposed schemes improve the TCP performance compares to SNOOP by knowing the cause of packet loss at the base station.

Keywords : TCP, SNOOP, 무선망(wireless network), CPC, ZWSC

1. 서 론

오늘날의 인터넷은 대부분이 유선망에 기반을 두고 있으며 이러한 유선망에 무선망을 도입한 유무선 통합망에서 우선적으로 직면하는 문제는 단말기의 이동성이며, 따라서 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 TCP의 유효성에 대한 검증은 해결해야 할 과제로 남아있다.

인터넷에서 가장 널리 사용되는 전송계층 프로토콜인 TCP(Transmission Control Protocol)는 송신측에서 일정 시간이 지나도록 ACK(acknowledgement) 패킷이 도착하지 않으면 타임아웃을 발생시키거나, 데이터 패킷의 손실을 알리는 중복 ACK 패킷을 받으면 네트워크에 혼잡이 일어났다고 가정하고 전송속도를 줄인다[1, 2].

* 교신저자 (Corresponding Author)

논문접수 : 2006. 05. 22, 채택확정 : 2006. 07. 06.

이명섭, 박창현 : 영남대학교 컴퓨터공학과

장주석, 박영민 : 경운대학교 모바일공학과

이와 같은 기능은 유선망과 고정 호스트(FH: Fixed Host)로 이루어진 전통적인 네트워크에 적합하며 유선망의 특성을 적용한 것이기 때문에 무선구간의 제한된 대역폭, 높은 지연, 산발적인 비트 에러, 순간적인 연결 단절 등과 같은 특성을 지닌 유무선 통합망에 TCP를 그대로 사용할 경우, 불필요한 메커니즘 호출로 인하여 전송속도가 감소되어 유선망 활용이 저조해지고 처리율(throughput)이 저하되게 된다.

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. 그러나 무선망에서의 패킷 손실은 대부분 혼잡에 의해서가 아니라 핸드오프 시 발생하는 이동 호스트(MH: Mobile Host)와 기지국(BS: Base Station)간의 연결 끊김과 오랜 지연, 그리고 무선망 자체의 높은 비트 에러율에 의해 발생한다. 따라서 앞에서 언급한 TCP를 그대로 무선망에 적용시키면 불필요한 제어 메커니즘(혼잡제어, 회피) 호출로 인해 성능저하를 가져오게 된다.

이에 무선망에서의 TCP 성능 향상을 위한 여러

가지 개선안들이 제안되었으며, 이들 가운데 가장 대표적인 프로토콜로 SNOOP를 들 수 있다. SNOOP은 기지국이 기본적으로 데이터와 ACK 패킷을 릴레이하는 역할을 담당하며, 무선 네트워크에서의 패킷 손실을 감추기 위해 중복 ACK을 스누핑하는 역할과 무선 링크 상에서의 오류에 의한 패킷 손실을 재전송에 의해 복구하는 역할을 담당한다[3, 4, 5, 6].

따라서 SNOOP은 무선망에서의 TCP 성능 향상을 지원하기 위한 다른 프로토콜에 비해 TCP의 중단간 연결 개념을 유지하며, 기지국에서의 버퍼요구량도 최소라는 장점이 있다. 그러나 기지국에서 지역적인 재전송을 통해 무선 링크에서의 패킷 손실을 복구하고 있는 동안 유선망에 접속되어 있는 송신측의 ACK 패킷을 받지 못하기 때문에 유선망에서의 가용성이 저하될 수 있다.

특히, 연속적인 패킷 유실이 발생할 경우, TCP persist timeout이나 재전송 timeout이 발생될 때까지 패킷을 재전송할 수 없는 유휴 상태가 되며, 이는 유휴시간 증가로 패킷 처리 속도를 지연시키는 결과를 가져오게 된다. 또한, 패킷 손실이 발생하고 복구가 지연될 경우 SNOOP에서는 재전송으로 인한 간섭현상이 발생한다.

본 논문에서는 패킷 손실이 연속적으로 발생할 경우 유휴시간을 줄이는 기법인 CPC(Consecutive Packet Control)를 기존에 제시된 SNOOP 프로토콜에 적용하여 무선 망에서의 TCP 성능향상에 더하여 연속적인 에러에 대한 성능향상을 제공한다. 개선된 SNOOP 프로토콜에서 ZWSC(Zero Window Size Control)는 타이머에 기반 한 재전송 간섭현상을 막아주는 기능을 제공한다.

II. 관련연구

무선망에서의 중단 간(End-to-End) 처리율을 향상시키기 위한 여러 가지 방법들이 소개되고 있다. 이런 방법들은 중단 간 프로토콜, 독립 연결(Split-Connection) 프로토콜, 링크계층 프로토콜의 세 가지로 크게 나누어 볼 수 있다[7].

중단 간 프로토콜은 송신자가 무선망의 존재 여

부를 알고 있는 프로토콜의 세 가지로, 패킷의 재전송 시 타임아웃 기법을 이용하지 않고 있다. 그리고 하나의 윈도우에 여러 개의 패킷 손실을 회복할 수 있도록 하는 선택적 재전송 방법과 패킷 손실의 원인이 혼잡에 의한 것이 아니라 다른 원인으로 발생하였음을 알려서 불필요한 혼잡제어 메커니즘의 호출을 방지하는 ELN(Explicit Loss Notification) 방법을 이용한다.

독립연결 프로토콜은 용어 자체에서 알 수 있듯이, 유선망과 무선망을 2개의 독립된 연결로 분리하고 각자에게 적합한 프로토콜을 적용하는 방법이다.

마지막으로 링크계층 프로토콜은 지역적 신뢰성을 제공하는 프로토콜으로써 TCP와 같은 네트워크 계층에 무선망과 관련된 손실을 숨김으로써 성능향상을 가져오는 ARQ (Automatic Repeat reQuest)와 FEC(Forward Error Correction)의 결합을 이용한다.

본 절에서는 본 논문에서 사용한 독립 연결 기반의 SNOOP 프로토콜에 대해 자세히 살펴본다. Berkeley 대학에서 제안한 SNOOP 프로토콜[3]은 무선망에서 높은 비트 에러율을 해결하기 위해서 MSR(Mobile Support Router)에 SNOOP 모듈을 적용하였으며, 핸드오프 시 발생하는 데이터 손실을 줄이기 위해 새로운 방식의 라우팅 프로토콜을 제시하였다.

SNOOP 모듈의 수행 방법은 다음과 같다.

- ① FH에서 MH로 데이터를 전송할 경우, FH가 전송한 패킷이 새로운 패킷이라면 이를 MSR에 저장하고 MH로 전송한다.
- ② SNOOP 모듈은 MH에서 보내는 ACK를 기억하고 있다가 패킷 손실이 발견되면 이미 MSR에 저장해 둔 패킷을 MH로 재전송한다.
- ③ MSR은 MH에서 보낸 중복 ACK를 FH로 보내지 않으므로써 FH에서의 불필요한 혼잡제어 메커니즘 호출을 방지할 수 있다.
- ④ HA는 primary MSR뿐만 아니라 이동 호스트의 이웃 MSR에게도 데이터를 포워딩함으로써 핸드오프가 발생할 때, 다른 프로토콜과는 달리 이전의 primary MSR에서 새로운

primary MSR로의 데이터를 포워딩하는 과정이 생략된다. 따라서 핸드오프 사이에 발생하는 지연을 없앨 수 있다.

이와 같이 동작하는 SNOOP 프로토콜은 TCP 종단간 연결개념을 유지, 핸드오프 시 지연시간 단축 등의 많은 장점이 있으나, 유선망에서의 가용성 저하, TCP persist timer 사용으로 인한 유희시간 증가, 재전송으로 인한 간섭현상 발생 등의 단점을 지니고 있다.

III. 제안하는 TCP 성능향상 방안

본 논문에서는 기존의 SNOOP에 비하여 좀 더 무선망에서의 패킷 전송율을 향상시키기 위하여 CPC를 사용하도록 SNOOP을 확장시키는 방안을 제안하며, FH에서 불필요한 혼잡제어가 일어나지 않도록 지원해주는 ZWSC를 제안한다.

제안된 기법들은 기존 SNOOP 모듈에 추가하여 리눅스 커널(Linux Kernel 2.4.18)상의 IP 계층에 구현되었다.

3.1. CPC(Consecutive Packet Control)

앞 절에서 살펴본 SNOOP의 단점 중 하나는 무선 링크에서 데이터 패킷이나 ACK 패킷이 연속적으로 유실될 경우, persist timeout이나 재전송 timeout이 발생될 때 까지 패킷을 재전송할 수 없는 유희 상태가 된다.

이와 같은 단점을 해결하기 위해 본 논문에서 제안한 CPC는 기존의 SNOOP 모듈과 연계하여 동작하도록 설계한다. 즉, 현재 버퍼에 캐싱되어 있는 패킷 중 가장 앞에 있는 ACK 패킷에 대한 중복 여부를 판단하여, 중복일 경우에는 CPC 모드로 동작하며 중복이 아닐 경우에는 기존 SNOOP 모드로 동작하도록 설계한다.

BS에 ACK 패킷이 도착한 경우 CPC의 처리 과정은 다음과 같다.

- ① 처음 ACK 패킷이 BS에 도착하면 새로운 ACK 패킷인가를 확인하고 새로운 ACK 패킷이고 "FLAG OFF"인 상태라면 SNOOP 모드로 동작한다.

- 해당 데이터 패킷을 버퍼에서 삭제
- 해당 무선 링크의 RTT를 계산
- 정상 수신한 ACK 패킷을 FH로 전송

- ② MH에서 전송된 ACK 패킷이 새로운 ACK이 아니라면
 - 버퍼에서 이미 삭제된 데이터 패킷 ACK 인지 확인
 - 그렇다면 그냥 버림(늦게 도착한 ACK 이기 때문)
 - 그렇지 않다면 첫 번째 중복 ACK인지 확인

- ③ 첫 번째 중복 ACK 패킷이면
 - 손실된 데이터 패킷을 MH로 지역 재전송
 - 중복 ACK 패킷 수를 계산하고 중복 ACK 을 버림
 - CPC를 처리하기 위해 "FLAG ON"으로 전환
 - last_seq 변수에 현재 버퍼에 캐싱된 마지막 데이터 패킷의 seq 번호를 저장

- ④ 무선 상태가 "FLAG ON"으로 등록되면, "FLAG OFF"상태가 될 때까지 ACK 패킷에 대한 즉각적인 로컬 재전송 발생
 - ACK 패킷이 도착할 때마다 "FLAG ON" 시점에 기록된 last_seq값이 현재 전송된 ACK 패킷의 seq보다 작은지를 비교하고, 작을 경우 "FLAG OFF"상태로 바꾸어 준다.

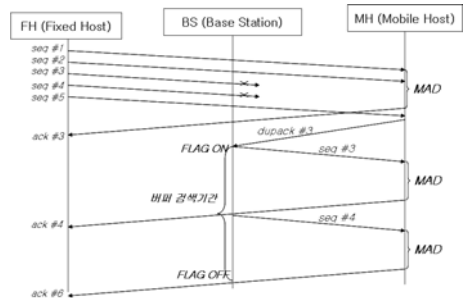


그림 1. CPC 패킷 흐름도
Fig. 1. Packet flow of CPC

이와 같은 과정을 패킷 흐름도를 통해서 보면 [그림 1]과 같다.

[그림 1]의 FH에서 데이터 패킷이 1 - 5번까지 보내졌고, 이 패킷들이 BS를 지나 MH로 가는 도중

3, 4번 패킷이 유실되었다고 가정하자.

일반적으로 TCP는 지연 ACK 방식을 사용하기 때문에 MH에서는 MAD(Maximum ACK Delay)시간 이후에 ACK 패킷을 보내게 된다. 그러므로 MH에서는 데이터 패킷 1, 2를 받고 3번을 요청하는 ACK #3패킷을 보낸다. ACK #3패킷이 BS에 도착하면 BS에서는 새로운 ACK 패킷인지에 대한 확인 과정을 거쳐 버퍼에 저장된 1, 2번 데이터를 삭제하고 송신측으로 전달한다.

이후에 MH에 5번 패킷이 도착하면 MH는 순서에 맞지 않는 패킷임을 감지하고 중복 ACK #3패킷을 보내게 된다. BS에 중복 ACK #3패킷이 도착하면, BS는 중복 ACK 패킷임을 감지하고 무선망이 CPC로 처리되어야 함을 알리는 "FLAG ON"상태를 설정하고, 동시에 버퍼에 들어있는 데이터 패킷 3을 로컬 재전송하게 된다. "FLAG ON"상태에서, 이후에 BS에서 들어오는 ACK 패킷에 대해서는 즉각적인 로컬 재전송이 일어난다.

무선망이 SNOOP 모드로 동작됨을 알려주는 "FLAG OFF"로 가기 위해 CPC는 "FLAG ON"시점에 last_seq를 기억하고 있다가 도착한 ACK 패킷의 번호를 비교하게 된다. 이때 last_seq값이 더 작게 되면 "FLAG OFF"상태로 되고, 이 경우 BS에서는 SNOOP 알고리즘을 따른다.

3.2. ZWSC(Zero Window Size Control)

일반적으로 SNOOP에서는 BS에서 중복 ACK 패킷을 받았을 경우 로컬 재전송이 일어나고, 이것이 여러 번 반복되어 패킷 처리가 느려지면 FH에서는 무선망에서의 패킷 손실을 혼잡으로 처리하는 오류를 범하게 된다.

이로 인해 FH에서 재전송이 일어나고 FH에서는 불필요한 혼잡제어가 일어나게 된다. 이와 같은 단점을 보완하기 위해 본 논문에서는 ZWSC를 제안한다.

본 논문에서 제안하는 ZWSC는 무선망에서 연속적인 패킷 손실이 일어날 경우 FH에서 timeout발생으로 무선 상의 패킷 손실을 감지하기 전에, BS에서 송신측으로 윈도우 크기를 zero(0)임을 알리는 메커니즘이다.

즉, 무선망에서 연속적인 패킷 손실로 인해 BS에서 로컬 재전송이 일어날 경우, 이를 FH가 감지하기 전에 MH의 윈도우 크기를 0으로 알림으로써 FH에서 더 이상 데이터 패킷을 보내지 못하도록 하는 메커니즘이다.

[그림 2]에서 ZWSC의 패킷 흐름을 보이며, 그 동작과정은 다음과 같다.

[그림 2]에서 중복 ACK #3번 패킷이 BS에 도착하였을 경우, SNOOP 동작에 의해 BS에서는 로컬 재전송이 발생하며, 이에 대한 ACK 패킷을 기다리는 timer를 설정한다.

이때, 로컬 타이머가 timeout되면 BS에서는 패킷이 유실되었다고 생각하고 패킷 재전송이 이루어진다.

두 번의 로컬 timeout이 발생될 때까지 해당 패킷에 대한 ACK 패킷을 받지 못하면 ZWSC 메커니즘에 의해 MH의 윈도우 크기가 0임을 알리는 ACK 패킷을 FH에 보내게 되고, 이로 인해 FH에서의 패킷 재전송을 막으로써 FH와 BS사이의 간섭을 방지할 수 있다.

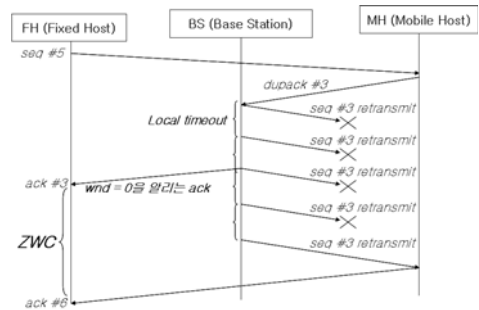


그림 2. ZWSC 패킷 흐름도

Fig. 2. Packet flow of ZWSC

만약 BS와 MH사이에 일정 횟수 이상 재전송이 일어날 때까지 해당 ACK 패킷을 받지 못하면 자동으로 윈도우 크기가 0인 상태에서 벗어나게 함으로써 낮아진 무선망의 활용도를 높일 수 있다.

IV. 성능 평가

본 논문에서 제안한 CPC&ZWSC메커니즘은 실제 리눅스 커널 상에 FH 1대, BS 1대, MH 1대로 구현되어 동작하며, 실측 실험을 수행하였으나 hop count가 1이기 때문에 실험에 필요한 노이즈, 페이딩, 간섭 등을 유도할 수 없었다.

따라서 제안하는 메커니즘과 기존의 SNOOP 프로토콜의 성능 비교를 위하여 시뮬레이션을 수행하였다. 4.1절에서 시뮬레이션에서 사용한 네트워크 트

래픽 모델에 대하여 설명하고, 4.2절에서는 시뮬레이션 결과를 분석한다.

4.1. 시뮬레이션 모델

시뮬레이션은 버클리 대학(U.C Berkeley)의 네트워크 시뮬레이터(Network Simulator) [8]를 이용하여 구현되었으며, 리눅스환경에서 수행되었다.

[그림 4]는 본 시뮬레이션의 네트워크 모델과 링크 특성을 보여준다.

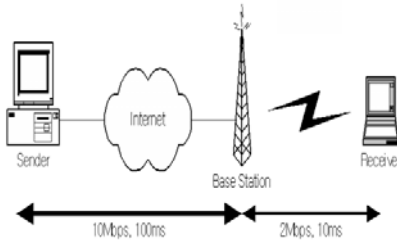


그림 4. 시뮬레이션을 위한 네트워크 모델
Fig. 4. Network model for simulation

유선 링크는 10Mbps의 대역폭을 가지고 있으며 전송지연은 100ms이다. 반면 무선 링크는 10ms의 전송 지연과 2Mbps의 대역폭 특성을 가지고 있다.

시뮬레이션 모델의 주요한 특성들을 요약해보면 먼저, 송수신측의 TCP는 Reno이며, 고정망에 연결된 FH는 FTP 혹은 TELNET 응용 프로그램에서 발생한 데이터를 무선망에 연결된 MH에게 전송한다. 무선 링크의 특성에 대한 TCP 성능을 평가하기 위하여 유선 링크에서의 혼잡 발생 확률은 0으로 설정하였다.

무선 링크의 비트 오류는 ‘좋은 상태’와 ‘중가 상태’ 그리고 ‘나쁜 상태’의 세 가지 상태로 구성되는 에러 모델에 의해 발생하도록 하였으며, 좋은 상태에서의 평균 비트 오류율은 10%로 설정하였고, 중간 상태에서는 비트 오류율을 15% 그리고 나쁜 상태에서의 평균 비트 오류율은 20%로 설정하였다.

그리고 좋은 상태가 지속되는 평균 기간은 0.8초, 중간 상태가 지속되는 시간은 0.5초 그리고 나쁜 상태가 지속되는 평균 기간은 0.2초로 설정하였다.

4.2. 시뮬레이션 결과 및 분석

본 논문의 목표인 무선 상에서의 TCP 성능향상을 검증하기 위해 시뮬레이션 결과를 그래프로 확인한다.

그래프에서 X축은 시간을 나타내며, Y축은 패킷의 시퀀스 번호를 나타낸다. 이것은 단위 시간당 증가하는 패킷의 시퀀스 번호를 보여줌으로써 각 프로토콜의 성능을 확인할 수 있다.

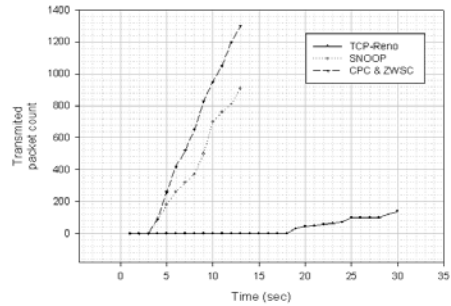


그림 5. 성능 분석 - 에러율 10%
Fig. 5. Performance analysis - Error rate 10%

[그림 5]는 에러율이 10%인 환경에서 TCP, SNOOP 그리고 본 논문에서 제안한 기법의 성능을 보여주고 있다.

그림에서 원으로 표시된 부분을 살펴보면, TCP의 경우 전송된 패킷이 없으며 SNOOP의 경우 700번 패킷까지 전송되었고 마지막으로 본 논문에서 제안한 기법은 950번까지 전송되었음을 확인할 수 있다.

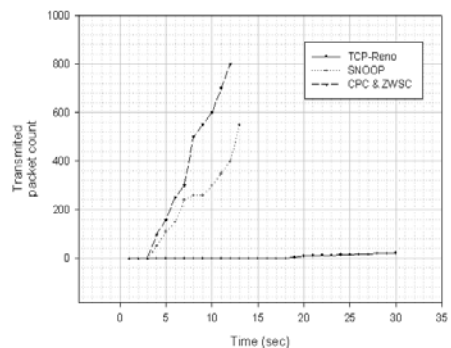


그림 6. 성능 분석 - 에러율 15%
Fig. 6. Performance analysis - Error rate 15%

즉, SNOOP의 경우는 TCP의 700배 그리고 CPC&ZWSC 기법이 SNOOP에 비해 1.5배 성능향상이 있음을 보인다.

[그림 6]은 에러율 15%를 적용한 결과이다. 이 실험결과에서도 에러율 10%인 경우와 마찬가지로 성능의 차이가 명확하며, 특히 SNOOP과 본 논문에서 제안한 CPC&ZWSC를 비교할 경우, 같은 시간동안 전송된 패킷의 수가 본 논문에서 제안한 기법이 약 2배 정도 처리량이 많음을 확인할 수 있다.

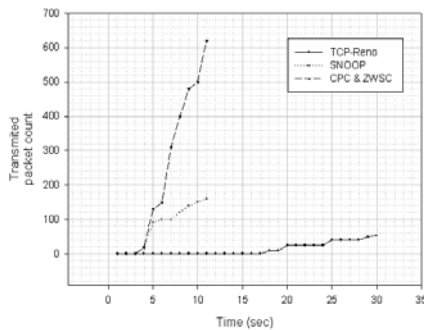


그림 7. 성능 분석 - 에러율 20%

Fig. 7. Performance analysis - Error rate 20%

[그림 7]의 경우도 기존 SNOOP의 단점을 보완한 CPC&ZWSC 기법의 경우 성능향상이 있음을 확인할 수 있다. 수치상으로 확인할 수 있듯이 동일 시간대에 패킷 전송량이 SNOOP에 비해 본 논문에서 제안한 CPC&ZWSC 기법이 약 4.5배 성능향상이 있음을 확인할 수 있다.

V. 결 론

본 논문에서는 기존의 SNOOP에 비하여 좀 더 무선 망에서의 패킷 전송율을 향상시키기 위하여 CPC를 사용하도록 SNOOP을 확장시키는 방안을 제안하며, FH에서 불필요한 혼잡제어가 일어나지 않도록 지원해주는 ZWSC를 제안하였다. 제안된 방안들은 기본적으로 SNOOP과 함께 동작하며, CPC의 경우 현재 버퍼에 캐스팅되어 있는 패킷 중 가장 앞에 있는 ACK 패킷에 대한 중복 여부를 판단하

여, 중복일 경우에는 CPC 모드로 동작하며 중복이 아닐 경우에는 기존 SNOOP 모드로 동작하도록 설계한다. 시뮬레이션을 통해 제안한 기법을 기존의 SNOOP과 비교한 결과 에러율의 증가율에 따라 제안기법의 성능이 향상됨을 확인할 수 있었다.

참고문헌

- [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 SIGMOD'98, Aug.1998.
- [3] H. Balakrishnan, S. Seshan, and R.H. Katz, "Improving Reliable Transport and Handoff Performance in Cellular Wireless Networks," ACM Wireless Networks, Dec. 1995.
- [4] H. Balakrishnan, Srinivasan Seshan, Elan Amir, Randy H. Katz, "Improving TCP/IP Performance over Wireless Networks," Proc. 1st ACM Conf. on Mobile Computing and Networks, Berkeley, CA, Nov. 1995.
- [5] Karu Ratnam and Ibrahim Matta, "WTCP: An Efficient Transmission Control Protocol for Networks with Wireless Links," Third IEEE Symposium on Computer and Communications (ISCC'98). 1998.
- [6] Karu Ratnam, Ibrahim Matta, "Effect of Local Re-transmission at Wireless Access Points on the Round Trip Time Estimation of TCP," Proceedings fo the The 31st Annual Simulation Symposium'98.
- [7] 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.
- [8] Network Simulator Document by U. Berkeley, <http://www.isi.edu/nanam/ns>

저 자 소 개

이 명 섭

1998년 경일대학교 컴퓨터공학 공학사. 2000년 영남대학교 대학원 컴퓨터공학과 공학석사. 2003년 영남대학교 대학원 컴퓨터공학과 공학박사. 현재, 영남대학교 전자정보공학부 객원교수. 관심 분야 : 망관리, 에이전트, Network QoS.

Email: skydream@yu.ac.kr

박 영 민

1993년 영남대학교 전산공학과 학사. 1995년 영남대학교 대학원 전산공학과 공학석사. 2005년 영남대학교 대학원 컴퓨터공학과 공학박사. 1995년 - 1999년 삼성전자 정보통신연구소 근무. 현재, 경운대학교 모바일공학과 조교수. 관심분야 : 컴퓨터비전, ASIC, 임베디드시스템

Email: ympark@ikw.ac.kr

장 주 석

1982년 영남대학교 전자공학과 학사. 1984년 영남대학교 대학원 전자공학과 공학석사. 1997년 영남대학교 대학원 컴퓨터공학과 공학박사. 1984년 - 1997년 대구과학대학 전자계산과 교수. 현재, 경운대학교 모바일공학과 교수. 관심 분야 : 영상처리, 패턴인식

Email: cschang@ikw.ac.kr

박 창 현

1986년 경북대학교 전자공학과 공학사. 1988년 서울대학교 계산통계학과 전산학전공 이학석사. 1992년 서울대학교 계산통계학과 전산학전공 이학박사. 1992년 - 1993년 서울대학교 컴퓨터신기술공동연구소 특별 연구원. 1998년 - 1999년 : University of Maryland, Institute of Advanced Computer Systems, Visiting Researcher. 현재, 영남대학교 컴퓨터공학과 교수. 관심분야 : 인공지능, 에이전트, 지능형 망관리

Email: park@yu.ac.kr