

논문 2006-43TC-11-15

IEEE 802.11x 무선 랜에서의 데이터 다운스트림 성능 향상

(An Improvement of Performance for Data Downstream in IEEE 802.11x Wireless LAN Networks)

김 지 흥*, 김 용 현*, 홍 윤 식**

(Ji-Hong Kim, Yong-Hyun Kim, and Youn-Sik Hong)

요 약

IEEE 802.11x 무선 랜 기반 유무선 통합 네트워크에서 고정 호스트인 데스크톱 PC와 이동 호스트인 PDA 간 다운스트림 성능을 개선할 수 있는 방안을 제안하였다. 성능 차가 뚜렷한 이기종 단말기간 데이터 전송시 다운스트림은 업스트림에 비해 수신시간이 최대 20% 느리다. 그 이유는 송신단의 패킷 전송률에 비해 수신단의 패킷 처리율이 낮아 혼잡 윈도우 크기가 수시로 변하고, 이에 따라 컨트롤 패킷이 증가하기 때문이다. 본 논문에서는 이러한 성능 저하를 완화시키기 위한 2가지 방안을 제시하였다. 먼저 PDA의 응용 계층 버퍼 크기를 증가시키면, 소켓 수신 버퍼의 처리 속도가 개선되어 혼잡 윈도우 변화율을 낮출 수 있다. 그러나 파일 액세스 시간은 버퍼 크기와 상관없이 거의 일정하였다. 버퍼 크기를 32,768바이트로 설정했을 경우 512바이트일 경우에 비해 수신시간을 32%정도 단축할 수 있었다. 또 다른 개선 방안은 송신자의 패킷간 전송 지연 시간을 설정하는 것이다. 송신단의 패킷 전송 간격을 5ms 로 지정하였을 때 수신시간을 최대 7% 단축시킬 수 있었다.

Abstract

We proposed a method for improving a performance of TCP downstream between a desktop PC as a fixed host and a PDA as a mobile host in a wired and wireless network based on IEEE 802.11x wireless LAN. With data transmission between these heterogeneous terminals a receiving time during downstream is slower than that during upstream by 20% at maximum. The reason is that their congestion window size will be oscillated due to a significantly lower packet processing rate at receiver compared to a packet sending rate at sender. Thus it will cause to increase the number of control packets to negotiate their window size. To mitigate these allergies, we proposed two distinct methods. First, by increasing a buffer size of a PDA at application layer an internal processing speed of a socket receive buffer of TCP becomes faster and then the window size is more stable. However, a file access time in a PDA is kept nearly constant as the buffer size increases. With the buffer size of 32,768bytes the receiving time is faster by 32% than with that of 512bytes. Second, a delay between packets to be transmitted at sender should be given. With an inter-packet delay of 5ms at sender a resulting receiving time is faster by 7% than without such a delay.

Keywords : TCP, IEEE 802.11x 무선랜, 다운스트림, 혼잡윈도우, 내부 패킷 간격

I. 서 론

무선 랜(LAN) 서비스는 주로 제한된 공간 내에서 노트북 컴퓨터나 PDA(Personal Digital Assistants)와 같은 개인 이동 단말기를 사용하여 무선으로 인터넷에

접속할 수 있는 환경을 제공한다. 무선 랜을 사용할 수 있는 공간은 허브(hub)에 해당하는 AP(Access Point)를 중심으로 대략 반경 100m 이내이다. 무선 랜은 전송 주파수 대역에 따라 IEEE 802.11b/g(2.4GHz) 및 IEEE 802.11a(5GHz) 등 2가지 방식으로 나눌 수 있다. IEEE 802.11g는 IEEE 802.11b와 호환성은 유지한 채, IEEE 802.11a의 OFDM(Orthogonal Frequency Division Multiplexing) 기술을 접목시켜 802.11b에 비해 데이터 전송률을 5배 향상시킨 것이다. 유럽의 경우 ETSI (European Telecommunication Standard

* 학생회원, ** 정회원, 시립인천대학교 컴퓨터공학과 (Dept. of Computer Science & Eng., University of Incheon)

※ 본 논문은 인천대학교 2006년도 자체연구비 지원에 의하여 연구되었음

접수일자: 2006년10월10일, 수정완료일: 2006년11월18일

Institute)에서 5GHz 주파수 대역의 표준으로 채택한 HIPERLAN/2가 있다^[1]. 최근 출시되고 있는 노트북 컴퓨터의 경우 IEEE 802.11a/b/g 무선 랜 표준을 모두 지원할 수 있도록 설계된 모바일 프로세서를 장착함으로써, 별도의 외장형 무선 랜 카드를 부착하지 않고서도 무선 랜 서비스 이용이 가능해졌다.

무선 랜 구성은 인프라스트럭처 네트워크 (*infrastructure network*)와 ad-hoc 망으로 나눌 수 있다. 인프라스트럭처 네트워크(그림 3)는 일종의 유무선 통합 네트워크로써 유선 랜에 연결된 하나 이상의 AP를 포함하고 있으며, 이동 단말기는 AP를 거쳐 유선 랜에 연결된다. ad-hoc 망은 고정된 네트워크 구조를 갖지 않으며, 필요에 따라 지리적으로 인접한 이동 단말기 간에 직접 메시지를 주고받을 수 있다.

IEEE 802.11x 무선 랜 기반의 유무선 통합 네트워크에서는 유선 랜의 서버(*fixed host, FH*)와 무선 랜의 이동 단말기(*mobile host, MH*) 간 데이터 송수신이 이루어진다. FH와 MH 간 성능(프로세서, 메모리 용량 등)이 엇비슷할 경우 이들 간에 이루어지는 데이터 전송은, 무선 링크가 갖는 불안정성을 제외하면, 유선 랜에서의 클라이언트와 서버 간 데이터 전송과 크게 다르지 않다. 그러나 FH와 MH 간 성능 차가 뚜렷할 경우 이들 기기 간 성능 차를 완화시킬 방안들이 고려되어야 보다 효율적인 데이터 송수신이 가능할 것이다. 예를 들면, 고성능의 FH가 데이터를 전송하고 상대적으로 저성능인 MH가 데이터를 수신하는 경우, 패킷 전송률에 비해 MH측에서의 패킷 처리율이 상대적으로 낮아 전체 처리율(*throughput*)이 저하될 수 있기 때문이다. 무선 랜 환경에서 MH로 사용 가능한 기기 중 노트북 컴퓨터의 경우 전체적인 성능 면에서 데스크톱 컴퓨터(이하 PC)와 비교해서 큰 차이가 없다. 반면 사용의 즉시성이나 휴대성 면에서 노트북 컴퓨터에 비해 유리한 PDA의 경우, 전용 프로세서의 처리 속도가 206MHz에서 400MHz로 2배 이상 빨라지긴 했지만, 수GHz 급의 노트북 컴퓨터나 PC에 비해 여전히 성능 면에서 뒤처진다. 더욱이 무선 랜 환경에서 이동 단말기로 PDA를 사용하여 FH로부터 대용량 멀티미디어를 다운로드 받을 경우, 이들 기기 간 성능 차로 인한 전송 성능 저하는 더욱 커질 것이다. 특히 VoWLAN(Voice Over WLAN)^[3]과 같은 무선 랜 기반 실시간 음성 및 화상 데이터 전송이 일반화될 경우 이러한 문제는 더욱 부각될 것이다. 최근 표준화 확정 단계인 IEEE 802.11e (Wireless Media Extensions)는 범용 데이터(웹 브라우

징 및 이메일 등)에 비해 음성 및 화상 데이터에 전송 우선순위를 부여함으로써 QoS를 보장하는 표준안이다.

따라서 본 논문에서는 무선 랜 환경에서 고성능 FH와 저성능 MH간 멀티미디어 데이터 송수신시 전송 성능에 어떠한 영향을 미치는지 알아보고자 한다. 이를 위해 음성 메시지 전송 시스템(VMS, *Voice Messenger Systems*)을 구현하여 이를 직접 측정해 볼 것이다. 또한, 기기 간 성능 차로 인한 전송 성능 저하를 개선할 수 있는 방안을 제시할 것이다. 이러한 성능 차를 완화하기 위한 방안으로 수신 버퍼 크기 및 패킷 간 전송 간격 조절 등에 대해 살펴볼 것이다.

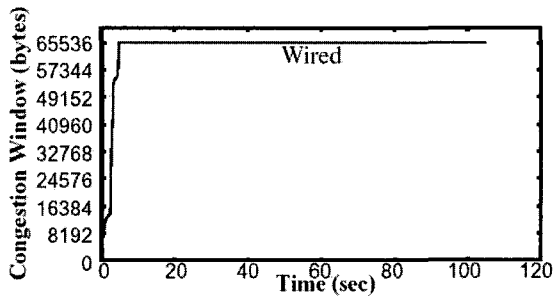
본 논문의 구성은 다음과 같다. I장 서론에 이어 II장에서는 관련 연구에 대해 언급하며, 무선 랜 환경에서 멀티미디어 데이터 전송 성능 분석에 대해 설명한다. III장에서는 II장의 성능 분석 결과를 토대로 성능 개선 방안을 제안한다. 마지막으로 IV장에서 결론 및 향후 연구 방향 등에 대해 언급한다.

II. 본 론

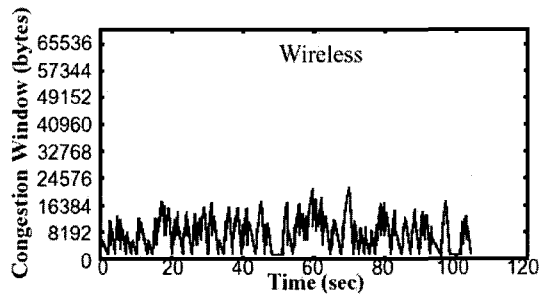
1. 무선 랜 환경과 TCP

무선 랜 환경에서는 유선 랜 환경에 비해 상대적으로 여러 발생 빈도가 높으며, 패킷 손실 등으로 인한 잦은 패킷 재전송 등으로 대역폭이 낮다. 그림 1(a)에 보인 것처럼, 유선 랜의 혼잡 윈도우는 짧은 지연 시간 경과 후, 그 크기가 거의 일정하게 유지된다. 반면에 무선 랜의 경우(그림 1(b)) 혼잡 윈도우 크기가 급격하게 변화할 뿐만 아니라, 그 크기도 유선 랜의 1/4선에 그치고 있다. 무선 랜에서의 이러한 혼잡 윈도우 크기의 급격한 변화는 무선 랜의 전송 성능을 저하시키는 요인이다^[4].

스트림 기반 프로토콜인 TCP^[5]에서는, 송신 및 수신 프로세스가 같은 속도로 데이터를 생성하거나 소비하지 않을 수 있으므로, 데이터 보관을 위한 버퍼가 필요하다^[6]. 이러한 버퍼는 데이터 흐름의 방향에 따라 송신 버퍼와 수신 버퍼로 나눌 수 있다(그림 2). 송신 버퍼에서는 보내고자 하는 데이터의 일부분을 저장하고 있다. 수신 프로세스의 상태나 망의 상태에 따라 전송한다. 전송한 데이터는 수신측으로부터 ACK 패킷이 올 때까지 계속 보관한다. 수신측에서 확인 응답이 오게 되면 전송된 데이터 크기만큼의 공간이 비워져서 송신 프로세스가 사용할 수 있는 순환 버퍼(*ring buffer*) 구조로 되어 있다. 수신 버퍼는 송신측에서 보낸 데이터를 수신프로세스가 데이터를 소비할 때까지 보관하는



(a) 유선 랜의 혼잡 윈도우
(a) the congestion window of a wired LAN



(b) 무선 랜의 혼잡 윈도우
(b) the congestion window of a wireless LAN

그림 1. 유무선 랜의 혼잡 윈도우 비교
Fig. 1. A comparison of the congestion window between a wired and a wireless LAN.

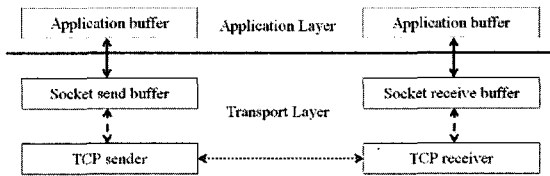


그림 2. TCP의 송수신단에서 사용하는 버퍼
Fig. 2. Buffer usages at the TCP sender and the TCP receiver.

역할을 한다^[4]. 응용 계층의 송신 및 수신 버퍼는 응용 프로그램에서 사용하는 데이터를 전송하기 전에 임시로 저장하기 위하여 사용자가 만든 버퍼이다. 반면 전송 계층의 소켓 송신 및 수신 버퍼는 데이터를 전송할 소켓에 있는 버퍼로 기본 버퍼의 크기는 사용하는 시스템에 따라 다르다(표 2 참조). 이러한 버퍼는 송신 및 수신 성능에 중요한 영향을 끼치기 때문에 최적 버퍼 크기 할당 등 효율적인 버퍼 관리를 위한 연구가 활발히 진행되고 있다^[7,8,9].

일반적으로 수신자의 패킷 처리율이 송신자의 패킷 전송률에 비해 현저히 낮을 경우 수신측에서는 버퍼 오버플로우가 발생한다. 이러한 버퍼 오버플로우를 막

기 위해 TCP에서는 슬라이딩 윈도우 방식을 사용하여 송신자의 패킷 전송 비율을 조정함으로써 흐름 제어를 수행한다. 즉, 수신자가 보낸 ACK 패킷에 따라 송신자의 윈도우 크기를 조절한다. 즉 송신자의 윈도우 크기는 수신자의 소켓 수신 버퍼 크기에 따라 동적으로 조절된다.

소켓 송신 버퍼와 수신 버퍼는 소켓 생성 시 자동으로 생성되지만, 사용자가 그 크기를 일정 범위 내에서 변경 할 수 있다. 유선 랜에서 이더넷(Ethernet)으로 연결된 2대의 워크스테이션 사이의 파일 전송 시, 송신 및 수신 버퍼의 크기를 기본 값(4,096바이트)에 비해 4 배 크게 설정했을 경우 처리율이 40% 향상된 사례가 있다^[13].

무선 랜의 경우 AP를 통하여 데이터를 주고받을 경우 업 링크와 다운 링크는 주변 환경에 따라 달라질 수 있다^[14]. 또한 무선 랜 환경에서 버퍼 크기와 성능은 밀접한 관계가 있다^[9]. 한편, 무선 랜은 유선 랜에 비해 제한된 대역폭, 산발적인 비트 에러, 핸드오프, 패킷 손실 등의 단점을 안고 있다. 따라서 무선 랜 환경에서 TCP 전송 성능을 개선시키기 위한 연구가 활발히 진행되었다.

무선망에서의 TCP 성능 향상을 위한 방법으로는 Indirect-TCP(I-TCP)와 Snoop TCP가 있다. I-TCP는 기지국을 중심으로 TCP 연결을 분리하여 종단 간 연결을 유선망과 무선망 두 부분으로 나누고, 기지국에서 각각의 연결을 독립적으로 제어함으로써 무선망에서의 오류를 송신측에 감추도록 하는 프로토콜이다^[15,16]. Snoop TCP는 무선 경로 부분만 재전송을 수행하므로 I-TCP의 목적과 같다. 즉, 종단간 TCP 시맨틱스(*semantics*)를 깨지 않고 손상된 패킷을 투명하게 재전송하도록 TCP 연결에서 스누핑이 수행되는 방식이다^[17,18].

이 외에도 TCP가 손실과 혼잡으로 유발된 손실을 구별할 수 있게 하여 각각의 경우에 적합하게 송신자가 작용하도록 허용하는 방식인 ELN^[19]이 있다. 또한, 마지막 패킷이 인식 된 후 패킷이 손실되었다고 송신자가 판단하여 추가적인 중복 ACK를 기다리지 않고 패킷을 재전송하는 SACK 옵션 방식^[20]이 있다.

2. 무선 랜의 다운스트림 전송 성능 분석

가. 음성 전송 시스템(VMS)

유선 랜의 서버(FH)인 PC와 무선 랜의 클라이언트(MH)인 PDA간 데이터 전송 성능을 측정하기 위하여 그림 3과 같은 음성 메시지 전송 시스템(VMS)을 구현

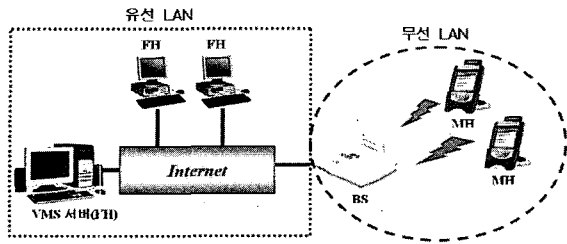


그림 3. 음성 메시지 전송 시스템 (VMS)
Fig. 3. A voice messenger system (VMS).

표 1. VMS의 FH 및 MH의 하드웨어 명세
Table 1. Detailed specifications of hardware of both FH and MH in a VMS.

	프로세서 (동작주파수)	메모리	NIC (전송률)
PC (FH)	인텔 펜티엄4 (2.4GHz)	1GB	PCI (100Mbps)
PDA (MH)	삼성 S3C2440 (400MHz)	128MB	CF (11Mbps)

하였다. 즉 이더넷 기반의 유선 랜과 IEEE 802.11b를 지원하는 AP를 통해 연결되는 무선 랜을 통합한 인프라스트럭처 네트워크를 구성하였다.

VMS의 동작은 다음과 같다. VMS의 송신 클라이언트가 사용자의 음성을 녹음하여, 녹음된 음성을 VMS 서버(FH)에게 전송한다. VMS 서버는 이 클라이언트가 보낸 음성 메시지를 수신하여 일단 하드디스크(캐시)에 저장한 다음, 수신 클라이언트가 현재 서버에 접속 중인지 여부를 확인한다. 만약 수신 클라이언트가 VMS 서버에 접속 중이라면 저장된 음성 메시지를 전달한다. VMS의 동작은 휴대 전화기의 SMS 메시지 전송 방식과 유사하다.

유선 랜의 PC와 PC간 데이터 송수신도 가능하지만, 본 논문에서는 PDA와 PC, PC와 PDA 및 PDA와 PDA 등 3 종류의 데이터 전송 성능만을 다룰 것이다. 또한, VMS 클라이언트인 PDA가 VMS 서버인 PC로부터 데이터를 다운로드(download) 받는 것을 다운스트림(downstream) 이라고 부르며, 반대로 PDA가 PC로 데이터를 업로드(upload)하는 것을 업스트림(upstream) 이라고 부르기로 한다.

VMS 구현을 위해 사용된 PC와 PDA의 하드웨어 명세는 표 1과 같다. PDA의 NIC (Network Interface Card)는 IEEE 802.11b를 지원하는 CF(Compact Flash) 방식의 랜 카드를 사용하였다. 표 1에서 알 수 있는 것처럼 PC와 PDA 간에는 모든 면에서 뚜렷한 성능 차가 존재한다.

나. 전송 방식별 성능 비교각 항 제목

(1) 데이터 크기에 따른 성능 비교

본 논문에서는 무선 랜 환경에서의 전송 프로토콜로 TCP를 사용하였다. 전송 성능 측정을 위해 음성 녹음 시간을 60초부터 최대 600초까지 변화시켜가며 샘플 데이터를 생성하였다. 녹음 시간이 60초인 경우 데이터 크기는 486KB이며, 600초일 경우 약 4.7MB이다. 데이터 크기를 변화시켜가며 업스트림, 다운스트림 및 PDA 간 전송 등 3가지 전송 방식별 수신 완료 시간을 측정 한 결과를 그림 4에 나타내었다.

PDA간 음성 데이터 전송의 경우, 업스트림(PDA 송신 클라이언트가 VMS 서버로 데이터 전송)에 이어 다운스트림(VMS 서버가 PDA 수신 클라이언트로 데이터 전송)이 연속해서 이루어지기 때문에 수신 시간은 업스트림과 다운스트림의 수신 시간을 합한 것과 비슷하다. 한편, 다운스트림의 경우 업스트림에 비해 수신 완료 시간이 최대 20% 느린 것을 알 수 있다. 특히 데이터 크기가 커질수록 두 전송 방식 간 수신 시간차가 커짐을 알 수 있다.

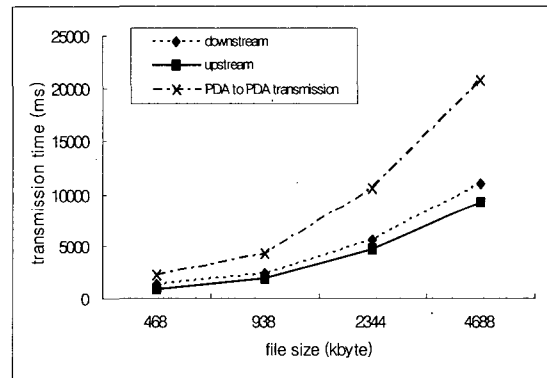


그림 4. 파일 크기에 따른 각 전송 방식별 수신 완료 시간

Fig. 4. A receive time with respect to its file size.

(2) 패킷 크기에 따른 성능 비교

MSS(Maximum Segment Size)는 TCP에서 전송할 수 있는 데이터 패킷의 최대 크기를 의미한다. 대용량 데이터를 전송할 경우 전송 패킷 크기를 응용 계층에서 MSS 보다 크게 지정하더라도 전송 계층에서는 이 패킷을 MSS보다 작은 패킷으로 단편화시킨다. 본 논문에서는 응용 계층에서의 전송 패킷 크기를 512바이트에서 7,168바이트까지 변화시켜 가며 전송 성능을 측정해 보았다. TCP의 성능은 송신지에서 수신지로 전송한 데이터의 양인 처리율로 평가한다. 즉 TCP의 처리율

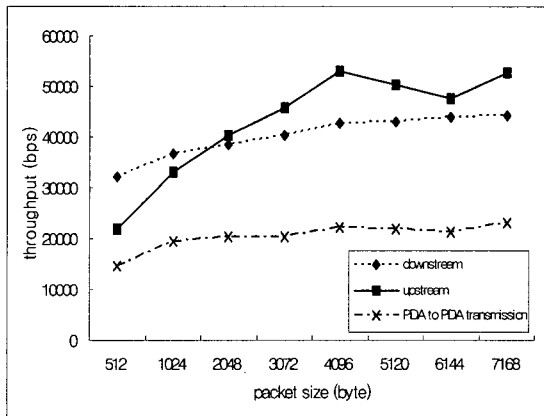


그림 5. 패킷 크기에 따른 전송 방식별 처리율 (2.34MB 크기의 데이터 사용)

Fig. 5. A throughput with respect to its packet size. (when the data size is set to 2.34MB)

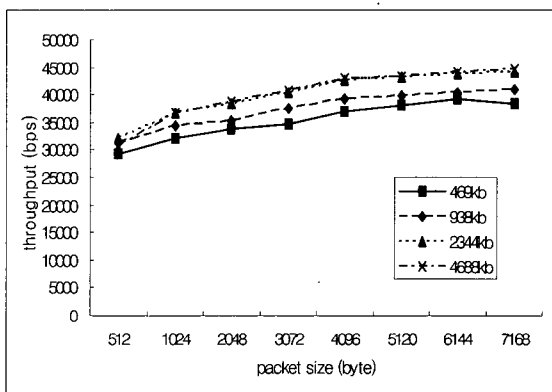


그림 6. 데이터 크기에 따른 다운스트림 처리율

Fig. 6. A throughput of downstream with respect to its data size.

(throughput)은 단위시간당 전송한 데이터 크기이다^[13]. 응용 계층의 패킷 크기에 따른 업스트림, 다운스트림 및 PDA간 전송 등 3가지 전송 방식별 처리율(bps)을 측정된 결과를 그림 5에 나타내었다.

패킷 크기를 4,096바이트에 이르도록 증가시킬 때까지는, 전송 방식에 무관하게 처리율이 계속해서 증가함을 알 수 있다. 일반적으로 전송 패킷 크기가 커질수록 전송해야 할 전체 전송 패킷의 개수는 작아지며, 이에 따라 수신해야 할 ACK 개수도 상대적으로 줄어들기 때문에, TCP 처리율이 증가한다. 전송 계층에서 TCP의 MSS는 전송매체가 이더넷일 경우에도 1,460바이트에 불과하다. 따라서 응용 계층의 패킷 크기를 MSS 이상으로 지정하더라도, 한 개의 패킷으로 MSS 이상 전송하지는 못한다. 그러나 TCP의 혼잡 제어 방식에 의해 수신자로부터 ACK 패킷을 수신할 때마다 연속해서 전송할 수 있는 패킷 개수가 점진적으로 늘어나기 때문

에, 응용 계층의 패킷 크기가 증가함에 따라 처리율이 높아지는 것으로 판단된다.

그러나 패킷 크기를 4,096바이트 이상으로 지정할 경우 처리율이 오히려 감소하거나 거의 변화가 없음을 알 수 있다. 다시 말해, 응용 계층에서 패킷 크기를 4,096바이트로 유지하는 것이 전송 방식에 무관하게 가장 높은 처리율을 보임을 알 수 있다. 그 이유는 표 2에 보인 바와 같이 PC이전 PDA이전 간에 소켓 송·수신 버퍼의 크기가 4,096바이트의 배수로 유지되는 것과 관련이 있을 것으로 추측된다.

또한, 패킷 크기가 1,024바이트 이하일 경우 오히려 다운스트림의 처리율이 업스트림에 비해 높게 나타났다. 그러나 이 경우를 제외하면 다운스트림의 처리율은 업스트림에 비해 현저하게 낮은 것으로 측정되었다.

그림 6은 데이터 크기에 따른 다운스트림 과정에서의 전송 처리율을 측정된 결과이다. 이 결과에서 알 수 있는 것처럼 일반적으로 데이터 크기가 커질수록 다운스트림의 처리율이 증가함을 알 수 있다. 또한, 응용 계층에서 전송 패킷 크기를 크게 할수록 처리율 역시 증가함을 알 수 있다.

(3) 다운스트림과 업스트림 성능 비교

패킷 분석 소프트웨어인 Analyzer^[21]를 사용하여 업스트림 과정 중 처음 11개의 패킷을 추출하여 분석한 결과를 그림 7에 나타내었다. 그림 7에서 굵은 선은 PDA 클라이언트가 VMS 서버에게 데이터를 송신하는 것이고, 가는 선은 VMS 서버가 PDA 클라이언트에게 데이터를 제대로 수신했음을 알리는 ACK 패킷을 보내

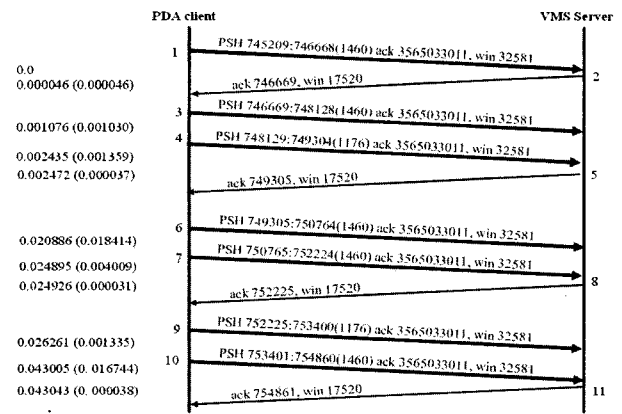


그림 7. 업스트림 전송 시 처음 11번째 패킷까지의 전송 과정

Fig. 7. The time line for the first 11 packets during the upstream.

는 것이다. PDA 클라이언트는 데이터를 송신할 때 처음엔 한 개의 패킷만 보내지만, ACK 응답을 받고난 직후에는 연속해서 두 개의 데이터 패킷을 전송한다. 이는 TCP가 AIMD (Additive Increase Multiplicative Decrease) 혼잡 제어 방식을 사용하기 때문이다^[22].

그림 7에서 왼쪽의 숫자는 VMS 서버에서 측정된 시간이며, 단위는 초(second)이다. 즉, 서버가 클라이언트로부터 데이터 패킷을 수신한 시간 및 서버가 클라이언트에게 ACK 패킷을 송신한 시간이다. 괄호 안의 숫자는 상대 시간을 가리킨다. 업스트림과의 비교를 위해 다운스트림 과정의 분석 결과를 그림 8에 나타내었다. 두 번째 패킷은 윈도우 업데이트를 위한 것으로서, 첫 번째 패킷에 비해 수신자 윈도우 크기가 업데이트되었음을 알리기 위한 것이다. 이는 빠른 송신자와 느린 수신자의 특징을 나타내는 것으로서, 윈도우 업데이트가 빈번할수록 전송 효율을 감소시키는 원인이 된다.

패킷 분석 결과, 전송 성능 면에서 업스트림과 다운스트림 간 뚜렷한 차이가 있음을 알 수 있다. 즉 업스트림 과정에서 수신자(VMS 서버)는 모두 7개의 데이터 패킷을 수신하였으며, 이때 걸린 시간은 0.043초였다. 반면 다운스트림 과정에서 수신자(PDA 클라이언트)는 모두 6개의 데이터 패킷을 수신하였으며, 이때 걸린 시간은 0.071초였다. 이렇게 차이가 나는 것은 다운스트림 과정에서 혼잡 상황으로 인한 윈도우 크기 변화 때문이다.

빠른 송신자와 느린 수신자의 특성으로 인한 혼잡 윈도우의 급격한 변화를 완화시키기 위한 방안은 패킷 처리율을 높일 수 있도록 느린 수신자에게 완충 기능을 제공하는 것이다. 본 논문에서는 수신자(PDA)의 소켓 수신 버퍼 크기를 조절하는 방안과 송신자(고정 호스트인 PC)의 패킷 전송 간격을 조절하는 방안 등에 대해 살펴보고자 한다.

(4) 다운스트림 성능 개선 방안

먼저 수신자인 PDA의 수신 버퍼 크기를 조절하는 방안에 대해 살펴본다. 무선 랜 환경에서 데이터 전송 성능 측정을 위한 PC(FH) 및 PDA(MS)의 기본 버퍼 크기는 표 2와 같다. 표 2에서 알 수 있는 것처럼 PDA의 송신 및 수신 버퍼 크기가 PC에 비해 모두 크다는 것을 알 수 있다. 그러나 PC의 버퍼 크기는 소켓 옵션을 사용해 사용자가 임의로 그 크기를 늘리거나 줄일 수 있는 반면에, PDA의 수신 버퍼는 제한된 메모리 용량으로 인해 그 크기를 마음대로 바꿀 수가 없다.

수신자인 PDA의 소켓 송수신 버퍼 크기를 변경하는

표 2. PC 및 PDA에서의 송·수신 버퍼 크기 (단위 바이트)

Table 2. Send-recv buffer size of PC and PDA.(byte)

	운영체제	송신 버퍼	수신 버퍼
PC (FH)	Windows 2000	8,192	8,192
PDA (MS)	Windows CE	16,384	32,768

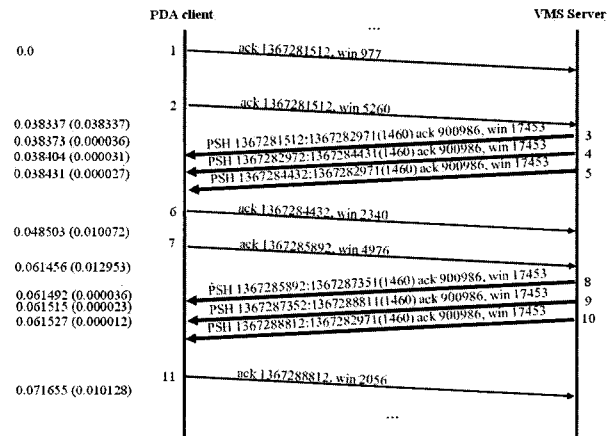


그림 8. 다운스트림 전송 처음 11번째 패킷까지의 전송 과정

Fig. 8. The time line for the first 11 packets during the downstream

대신, 응용 계층의 수신 버퍼 크기 변화에 따른 성능 변화를 분석해 보고자 한다. 수신자의 소켓 수신 버퍼의 여유 공간이 클수록 송신자의 윈도우 크기는 안정적으로 유지될 수 있다. 다시 말해 소켓 수신 버퍼에 저장된 데이터의 내부 처리율을 높이기 위해서는 소켓 수신 버퍼에 저장된 데이터를 최대한 빨리 응용 계층의 버퍼로 옮겨, 소켓 수신 버퍼의 여유 공간을 늘리는 것이 필요하다.

다음으로 송신자의 패킷 전송 간격을 조절하는 방안에 대해 살펴본다. 일반적인 관점에서 보면 패킷 간 전송 간격이 짧을수록 전송시간은 단축된다고 할 수 있을 것이다. 그러나 송신자의 패킷 전송률에 비해 수신자의 패킷 처리율이 현저히 낮을 경우에는 위의 사실이 적용되지 않을 수 있다. 따라서 패킷 간 전송 간격, 즉 전송 지연 시간을 삽입하여 송신자의 패킷 전송률을 조절함으로써 수신자의 패킷 처리율을 높일 수 있다면 전체적인 전송 성능을 개선할 수 있다. 그러나 지나친 패킷 전송 지연은 오히려 전송 성능을 저하시킬 수 있기 때문에 적절한 패킷 간격을 설정하는 것이 중요하다.

III. 실험

1. 수신 버퍼 크기

수신측, 즉 PDA 클라이언트는 소켓 수신 버퍼(크기 고정)와 응용 계층 버퍼(크기 조절 가능) 등 2 종류의 버퍼를 갖고 있다. 송신측의 윈도우 크기에 직접적인 영향을 미치는 것은 소켓 수신 버퍼이다. 소켓 수신 버퍼에 저장된 수신 데이터를 가능한 빨리 응용 계층 버퍼로 옮기면, 그만큼 소켓 수신 버퍼의 수신 용량이 늘어나게 되어 송신측 윈도우 크기가 커질 것이다. 윈도우 크기는 전송 데이터 크기와 직접적 관련이 있으므로, 윈도우 크기 증가는 처리율을 높이게 될 것이다. 따라서 대부분의 경우 응용 계층의 버퍼 크기는 클수록 유리하다. 그러나 응용 계층의 버퍼 역시 임시 저장 중인 데이터를 PDA의 파일 시스템(주로 플래시 메모리)에 빨리 옮겨 놓아야 버퍼의 여유 공간을 안정적으로 확보할 수 있을 것이다. 즉 응용 계층 버퍼 크기와 함께 PDA 내부에서 발생하는 파일 액세스(다운스트림의 경우 write 연산) 시간도 함께 고려해야 한다. 본 논문에서는 응용 계층 버퍼 크기를 512바이트에서 32,768바이트까지 증가시키며 다운스트림 과정의 수신 완료 시간을 측정하였으며, 측정 결과를 그림 9에 나타내었다. 한편, 이 실험에서 패킷 크기는 4,096바이트로 설정하였다.

그림 9에 보인 것처럼 수신측의 응용 계층 버퍼 크기가 커짐에 따라 수신 완료 시간이 단축됨을 알 수 있다. 버퍼 크기를 32,768바이트로 설정했을 경우, 512바이트로 설정했을 경우에 비해 약 32% 정도 단축되었다. 그러나 버퍼에 저장된 내용을 PDA의 플래시 메모리에 기록하는 데 걸리는 파일 액세스 시간은 버퍼 크기와

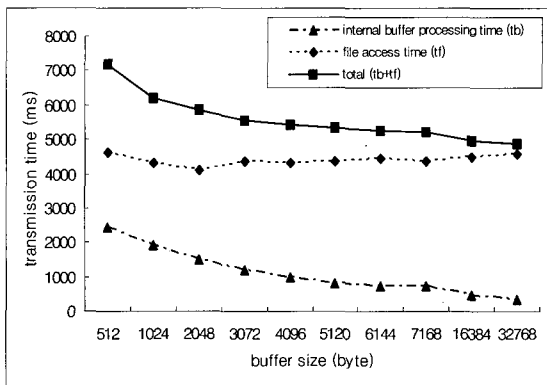


그림 9. 다운스트림 과정에서 수신측의 응용 계층 버퍼 크기에 따른 수신 완료 시간

Fig. 9. The transmission time with respect to the size of the receive buffer at application layer.

상관없이 거의 일정함을 알 수 있다. 다시 말해 버퍼 크기가 커지면 파일 액세스 횟수가 줄어들어 수신 완료 시간이 단축되는 것은 아니라는 뜻이다. 오히려 대용량 버퍼일수록 소켓 수신 버퍼에 저장된 수신 데이터를 신속히 응용 버퍼에 옮길 수 있게 되어, 윈도우 크기 증가로 인한 처리율 상승이 수신 완료 시간 단축에 직접적인 영향을 미치는 것으로 판단할 수 있다. 그림 9에 보인 것처럼 버퍼 크기 증가에 따라 버퍼 처리 시간이 감소함을 알 수 있으며, 버퍼 처리 시간의 변화는 수신 완료 시간 변화와 일치함을 알 수 있다.

2. 내부 패킷 간격에 따른 성능 측정

내부 패킷 간격은 Visual C++ 6.0에서 Sleep 함수를 사용하여 조절할 수 있다. Sleep 함수에서 설정할 수 있는 단위는 1/1000초(ms)이다. 내부 패킷 간격을 전혀 주지 않는 경우(함수를 호출하지 않음)에서부터 최대 100ms까지 설정한 경우까지 내부 패킷 간격에 따른 업스트림 및 다운스트림 과정에서의 전송 완료 시간을 측정하였다. 송신측에만 내부 패킷 간격을 지정한 경우 및 송신측 및 수신측 모두 내부 패킷 간격을 지정한 경우 2가지에 대해 전송 시간을 측정하였다. 이 실험에서 전송 패킷 크기는 4,096바이트로 설정하였으며, 전송 파일 크기는 2,344KB를 사용하였다.

먼저 업스트림 과정에서 내부 패킷 간격에 따른 전송 완료 시간을 측정한 결과를 그림 10에 나타내었다. 업스트림의 경우는 내부 패킷 간격을 지정하지 않았을 경우가 가장 좋은 성능을 보였다. 수신측인 유선 랜의 서버(PC)가 송신측인 PDA 클라이언트에 비해 훨씬 뛰어

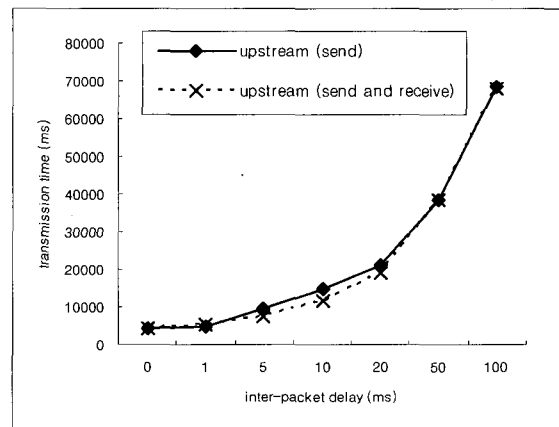


그림 10. 내부 패킷 간격에 따른 업스트림 과정의 수신 완료 시간

Fig. 10. The transmission time with respect to inter-packet delay during the upstream.

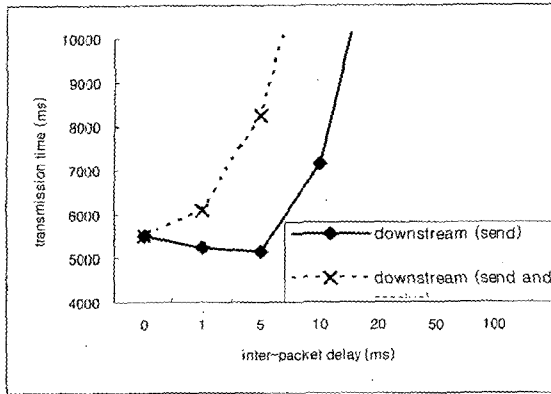


그림 11. 내부 패킷 간격에 따른 다운스트림 과정의 수신 완료 시간

Fig. 11. The transmission time with respect to inter-packet delay during the downstream.

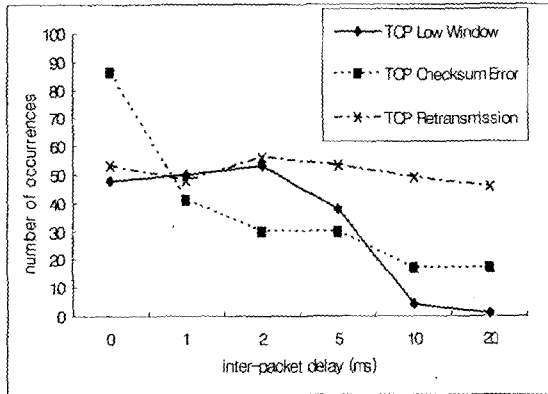


그림 12. 다운스트림에서 내부 패킷 간격에 따른 오류 패킷 발생 분석

Fig. 12. The packet losses with respect to inter-packet delay during the downstream.

난 성능을 갖고 있기 때문에, 수신측에서 수신 패킷 처리 지연이 발생할 가능성이 거의 없다.

다운스트림 과정에서 내부 패킷 간격에 따른 전송 완료 시간을 측정된 결과를 그림 11에 나타내었다. 업스트림의 경우와 달리 송신단의 패킷간 전송 간격을 5ms로 지정했을 경우 내부 패킷 간격을 전혀 지정하지 않았을 경우에 비해 7%가량 전송 성능이 개선됨을 확인할 수 있다. 또한, 송·수신측에서 모두 내부 패킷 간격을 설정하는 것에 비해 송신측에 한해 패킷간 전송 간격을 지연시키는 것이 효율적임을 알 수 있다. 즉 수신측인 PDA 클라이언트가 송신측인 유선랜의 서버에 비해 성능이 뒤처지므로, 수신 패킷을 처리할 여유 시간을 주기 위한 송신측 패킷 전송 지연이 전송 시간 개선에 뚜렷한 영향을 미침을 알 수 있다.

다만, 패킷간 전송 지연 시간을 지나치게 높게 설정하는 것은 전송 시간을 오히려 지연 시키게 된다는 것

이다. 실험 결과에 따르면, 패킷간 전송 지연 시간이 10ms를 넘지 않아야 하며, 최소한 1ms 이상 설정하는 것이 필요하다. 그러나 패킷간 전송 시간 지연으로 인한 성능 개선 효과는 응용 계층 버퍼 크기를 변화시키는 것에 비해 1/3~1/5 수준에 불과하다.

(그림 12)는 무선망에서의 패킷 분석 도구인 AiroPeek^[23]을 사용하여 PDA 다운스트림의 내부 패킷 간격에 따른 오류 패킷 발생을 분석한 결과이다. 송신측에 패킷 전송 간격을 지연시킬수록 오류 패킷 수가 뚜렷이 감소함을 알 수 있다. 즉 송신측 내부 패킷 간격 설정이 오류 패킷 발생 가능성을 낮춰 줌으로써 결과적으로 전송 성능 향상을 가져옴을 알 수 있다. 그림 12에서 Low Window란 수신단에서 수신 패킷을 제대로 처리하지 못하는 상황을 나타내며, Checksum error는 패킷의 헤더 또는 데이터에 오류가 발생(한 개 이상의 비트에서 오류 발생)했음을 가리킨다.

IV. 결 론

본 논문에서는 IEEE 802.11x 무선 랜 기반 인프라스트럭처 네트워크에서 고성능의 고정 호스트인 데스크톱 PC와 저성능의 이동 호스트인 PDA 간 다운스트림 성능을 개선할 수 있는 방안을 제안하였다. 특히 기존 응용 프로그램 대부분이 전송 계층 프로토콜로 TCP를 그대로 사용하고 있는 점을 감안하여, TCP 기반 트래픽 모델링은 그대로 사용하는 대신 응용 계층에서 TCP 성능을 개선할 수 있는 방안들을 제시하였다. 먼저 무선 랜에서 대용량 멀티미디어를 효율적으로 전송하기 위해서는 응용 계층에서 패킷 크기를 MSS보다 훨씬 크게 설정하는 것이 유리하다. 응용 계층에서 패킷 크기를 512바이트에서 최대 7,168바이트까지 단계별로 증가시키며 측정된 결과, 업스트림 및 다운스트림에 상관없이 패킷 크기가 커질수록 전송 효율이 향상됨을 확인하였다.

한편, 고성능 PC와 저성능 PDA간 다운스트림 프로세스에서 이들 기기 간 성능 차로 인해 업스트림 프로세스에 비해 혼잡 윈도우 크기가 수시로 변하고, 이에 따른 컨트롤 패킷 증가로 인해 전송 성능이 크게 저하됨을 확인할 수 있었다. 본 논문에서는 이러한 성능 차를 완화하기 위한 2가지 방안을 제시하였다. 먼저 PDA의 응용 계층 버퍼 크기를 증가시키면, 소켓 수신 버퍼의 처리 속도가 개선되어 혼잡 윈도우 변화율이 낮아지면서 컨트롤 패킷 전송 횟수도 함께 줄어 전송 성능이

개선된다. 버퍼 크기를 32,768바이트로 설정했을 경우, 512바이트로 설정했을 경우에 비해 약 32% 정도 단축되었다. 버퍼 크기가 커지면 파일 액세스 횟수가 줄어들긴 하지만, 파일 액세스 시간은 버퍼 크기와 상관없이 거의 일정하였다. 오히려 대용량 버퍼일수록 소켓 수신 버퍼에 저장된 수신 데이터를 신속히 응용 버퍼에 옮길 수 있게 되어, 윈도우 크기 증가로 인한 처리율 상승이 수신 완료 시간 단축에 직접적인 영향을 미치는 것으로 나타났다.

또 다른 개선 방안은 다운스트림 프로세스에서 송신 측, 즉 고정 호스트인 고성능 PC 측에서 데이터 송신 시 내부 패킷 간격을 설정하는 것이다. 패킷간 전송 지연 시간은 최소 1ms 이상 최대 10ms 범위 내에서 설정해야 한다. 특히 패킷간 전송 간격을 5ms로 지정했을 경우 내부 패킷 간격을 전혀 지정하지 않았을 경우에 비해 7%가량 전송 성능이 개선됨을 확인할 수 있다. 또한 송·수신측에서 모두 내부 패킷 간격을 설정하는 것에 비해 송신측에 한해 패킷간 전송 간격을 지연시키는 것이 효율적임을 알 수 있다.

참 고 문 헌

- [1] C. Murthy and B. S. Manoj, "Ad Hoc Wireless Networks: Architectures and Protocols", Chapter 2, Prentice Hall, 2004.
- [2] S. Pilosof, R. Ramjee, D. Raz, Y. Shavitt, and P. Sinha, "Understanding TCP fairness over Wireless LAN", *IEEE INFOCOM*, Vol.2, pp.863-872, April 2003.
- [3] ARC Group, Future Mobile Computing: Device Trends and Wireless Solutions, 2004.
- [4] W. R. Stevens, *TCP/IP Illustrated, Volume 1 : The Protocols*, Addison-Wesley, 1994.
- [5] J. Postel, "Transmission Control Protocol - Protocol Specification", *RFC 793*, September 1981.
- [6] Behrouz A. Rorouzan, "TCP/IP Protocol Suite", MC Graw Hill, 2003.
- [7] Razdan, A., Nandan, A., Wang, R., Sanadidi, M.Y., Gerla, M., "Enhancing TCP performance in networks with small buffer," *Computer Communications and Networks*, pp.39-44, October 2002.
- [8] 김주영, 정영준, "버퍼관리 기법을 응용한 무선 TCP 성능 개선 방안" 한국컴퓨터종합학술대회, 2005.
- [9] King-Tim To, Sammy Chan, "Buffer Sizing for Nonhomogeneous TCP Sources", *IEEE COMMUNICATIONS LETTERS*, pp.567-569, June 2005.
- [10] 김종덕, 임효준, 김종권, "무선망 위에서의 TCP 성능 향상 방법과 성능 분석", 한국통신학회 추계종합학술대회, 1999.
- [11] Caini. C, Firrincieli. R, "Packet spreading techniques to avoid bursty traffic in long RTT TCP connections", *VTC2004*, Vol.5, pp.2906-2910, May 2004.
- [12] Sung Won Kim, Byung Seo Kim, "Downlink and uplink resource allocation in IEEE 802.11 Wireless LANs", *IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY*, pp.320-327, January 2005.
- [13] W. R. Stevens, "TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms," *RFC2001*, 1994.
- [14] Dong Geun Jeong and Wha Sook Jeon, "CDAM/TDD System for wireless multimedia services with traffic unbalance between uplink and downlink", *IEEE Journal on selected AREAS IN COMMUNICATIONS*, pp.939-946, May 1999.
- [15] A. Bakre and B. R. Badrinath, "Implementation and Performance Evaluation of Indirect-TCP", *IEEE Transactions on Computers*, pp.250-278, March 1997.
- [16] A. Bakre and B. R. Bandrinath, "I-TCP : Indirect TCP for Mobile Hosts", *Proceeding of the International Conference on Distributed Computing Systems*, pp.136-143, June 1995.
- [17] H. Balakrishnan, S. Seshan, and R. H. Katz, "Improving Reliable Transport and Handoff Performance in Cellular Wireless Networks", *ACM Wireless Networks*, pp.469-481, 1995
- [18] H. Balakrishnan, Srinivasan Seshan, Elan Amir, Randy H. Katz. "Improving TCP/IP Performance over Wireless Networks", *Proc. 1st ACM Conf. On Mobile Computing and Networking, Berkeley, CA*, pp.146-152, November 1995.
- [19] H. Balakrishnan and R. Katz, "Explicit Loss Notification and Wireless Web Performance", *In IEEE CLOBECOM Internet Mini Conference*, November 1998
- [20] K. Fall and S. Floyd, "Simulation-based Comparison of Tahoe, Reno and SACK TCP", *ACM Computer Communication Review*, pp.5-21, July 1996.
- [21] Analyzer 웹 사이트, <http://analyzer.polito.it>
- [22] V. Jacobson, "Congestion Avoidance and Control", *Proceedings of ACM SIGCOMM*,

pp.314-329, 1998.

[23] WildPackets 웹 사이트 , <http://wildpackets.com>

저 자 소 개



홍 윤 식(정회원)
 1983년 한양대학교 전자공학과
 학사 졸업.
 1985년 한국과학기술원 전기 및
 전자공학과 석사 졸업.
 1989년 한국과학기술원 전기 및
 전자공학과 박사 졸업.

<주관심분야 : 모바일 컴퓨팅, USN>



김 지 홍(학생회원)
 2006년 시립인천대학교 컴퓨터
 공학과 학사 졸업
 2006년~현재 시립인천대학교
 대학원 컴퓨터공학과 석사
 과정

<주관심분야 : USN, 라우팅 프로
토콜>



김 용 현(학생회원)
 2006년 시립인천대학교 컴퓨터
 공학과 학사 졸업
 2006년~현재 시립인천대학교
 대학원 컴퓨터공학과
 석사과정

<주관심분야 : 센서네트워크, 라우
팅 프로토콜, 미들웨어>