

# 무선 랜 고속전송을 위한 최대버퍼한계 기반 TCP 버퍼튜닝

## TCP Buffer Tuning based on MBT for High-Speed Transmissions in Wireless LAN

문성곤\*                      이홍석\*\*                      추현승\*\*\*                      공원영\*\*\*\*  
Sung-Gon Mun              Hongseok Lee              Hyunseung Choo              Wonyoung Kong

### 요 약

무선 랜(IEEE 802.11)은 신뢰적인 데이터 전송을 위해 기존 TCP를 사용하고 무선 신호의 단절, 간섭, 감쇠 영향으로 인하여 혼잡하지 않은 상태에서 의도하지 않은 패킷손실을 유발한다. 무선 랜에서의 TCP는 패킷손실을 혼잡으로 판단하고, 혼잡제어 알고리즘을 작동하므로 전반적인 전송률이 감소된다. 본 논문은 무선 랜에서의 전송률 증가를 위해 유선 네트워크 고속전송 기법인 TCP 버퍼튜닝을 적용하여 TCP 버퍼크기와 전송률과의 상관관계를 분석한다. 분석을 통해 버퍼크기 증가에 관계없이 더 이상의 전송률이 증가되지 않는 특정 버퍼 한계점을 찾고, 이를 우리는 최대버퍼한계(Maximum Buffer Threshold, MBT)라 정의한다. 최대버퍼한계를 산출하기 위해 실제 연주를 통해 음악파일을 생성한 후 이를 이용하여 실험을 한다. 최대버퍼한계 기반 TCP 버퍼튜닝은 운영체제 기본 버퍼크기에서의 무선 전송률과 비교하여 송수신간 RTT가 5ms 구간에서 20.3%, RTT 10ms에서 21.4%, RTT 20ms에서 45.4% 전송률이 향상된다. 또한 특정 RTT 구간에서 최대 전송률을 97%의 정확도로 예측하고, 최대버퍼한계 이상으로 TCP 버퍼크기를 설정하는 것은 전송률 증가에 영향이 없음을 확인한다.

### Abstract

Wireless LAN (IEEE 802.11) uses traditional TCP for reliable data transmission. But it brings the unintentional packet loss which is not congestion loss caused by handoff, interference, and fading in wireless LAN. In wireless LAN, TCP experiences performance degradation because it consumes that the cause of packet loss is congestion, and it decrease the sending rate by activating congestion control algorithm. This paper analyzes that correlation of throughput and buffer size for wireless buffer tuning. We find MBT (Maximum Buffer Threshold) which does not increase the throughput through the analysis. For calculation of MBT, we experiment the throughput by using high volume music data which is created by real-time performance of piano. The experiment results is shown that buffer tuning based on MBT shows 20.3%, 21.4%, and 45.4% throughput improvement under 5ms RTT, 10ms RTT, and 20ms RTT, respectively, comparing with the throughput of operation system default buffer size. In addition, we describe that The setting of TCP buffer size by exceeding MBT does not have an effect on the performance of TCP.

☞ Keyword : TCP, Buffer Tuning, Wireless Network, Music data

## 1. 서론

- \* 준 회 원 : (주)엠텔로 근무  
msgon@ece.skku.ac.kr
- \*\* 정 회 원 : 성균관대학교 컨버전스연구소 연구원  
juspeace@hanmail.net
- \*\*\* 중신회원 : 성균관대학교 정보통신공학부 부교수  
choo@ece.skku.ac.kr
- \*\*\*\* 정 회 원 : 수원대학교 음악대학 피아노학과 겸임교수  
kongchoo@kornet.net

IEEE 802.11 무선 랜(Wireless LAN, WLAN)은 지형지물로 인한 일시적인 신호의 단절, 다른 주파수에 의한 신호 간섭 및 감쇠의 영향으로 불안정한 연결성을 가진다. 그리고 무선 랜은 신뢰적인 데이터 전송을 위해 기존의 TCP를 그대로 사용하고 있으며, 무선 랜의 불안정성으로 인하여

[2006/06/08 투고 - 2006/07/03 심사 - 2006/09/12 심사완료]

패킷손실이 발생된다[1]. TCP는 패킷손실을 네트워크 혼잡이라 가정하므로 패킷손실이 있을 때마다 혼잡제어 알고리즘이 작동되고, 결과적으로 의도하지 않은 전송률의 감소를 가져온다.

무선 랜에서의 TCP 문제점을 해결하기 위해서 여러 학회와 업체는 다양한 방법의 연구를 했다. 지금까지의 연구 동향은 크게 TCP 혼잡제어 알고리즘 자체를 개선하는 방법과 중간 게이트웨이 장치를 더 지능적으로 만드는 방법으로 분류된다. 이러한 연구는 TCP 혼잡제어에 기반하고 있기에 실제 무선 랜에 적용될 때까지 많은 시간과 비용이 필요하다.

한편 흐름제어에 기반한 TCP 버퍼튜닝 기법은 송수신 호스트 및 네트워크가 허용할 수 있는 데이터의 크기를 TCP 버퍼크기로 설정하고 데이터를 전송하는 방법으로 유선 네트워크에서 성능이 입증된 우수한 고속전송 기법중 하나이다. 이러한 흐름제어 방법은 혼잡제어 알고리즘 자체를 수정하는 것보다 비교적 빠르고 간단하게 전송률을 향상시킨다. 하지만 부적절한 TCP 버퍼크기의 설정은 시스템에 악영향을 미칠 수 있으므로 항상 조심스러운 접근이 필요하다. 따라서 유선 네트워크의 고속전송 기법인 TCP 버퍼튜닝 기법을 무선 랜에 적용하기 위해서 TCP 버퍼크기와 무선 전송률과의 관계 분석이 필요하다.

본 논문에서는 더 효율적인 흐름제어 기법을 수행하기 위해서 다양한 RTT 구간에서 TCP 버퍼크기에 대한 전송 실험을 한다. 이 때, 실험은 송신자에서 실제 연주로 음악 파일을 생성하여 수신자가 이를 받는 실험을 통해서 TCP 버퍼크기를 증가시켜도 더 이상의 전송률 상승이 없는 지점을 찾고, 이를 무선 랜에서의 TCP 버퍼튜닝 기준으로 설정한다. 무선 랜에서의 기본 전송률과 TCP 버퍼튜닝을 이용한 전송률과의 비교를 통해서 무선 랜에서의 TCP 전송률 향상을 확인하고, 버퍼크기와 전송률과의 관계를 정리한다. 또한 음악파일 실시간 서비스 실험으로 무선랜 환경에서 고음질 음악파일의 실시간 제공을 통해 청중과

연주자가 다른 장소에 있어도 고음질의 음악을 제공할 수 있는 새로운 형태의 음악회를 구성할 수 있음을 확인한다.

무선 랜 환경에서 TCP 전송률을 향상시키기 위한 본 논문은 다음과 같은 구성된다. 2장에서 유선 네트워크에서 사용되는 여러 가지 고속전송 기법과 무선 랜의 특성 및 개선 방향에 대하여 설명한다. 3장에서는 무선 랜 TCP 버퍼튜닝 실험을 통한 최대버퍼한계를 제시하고, 4장에서 최대버퍼한계 기반 TCP 버퍼튜닝의 효율성을 입증한다. 마지막으로 결론과 향후 연구방향을 설명한다.

## 2. 관련연구

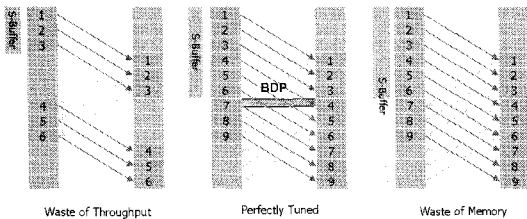
### 2.1 유선 네트워크 고속전송을 위한 TCP 버퍼튜닝

현재 유선 네트워크 고속전송 기법의 큰 연구방향으로는 각 연결에 필요한 TCP 버퍼 크기를 결정하여 조절하는 기법과 TCP 버퍼 제한을 피하기 위해 단일전송에 다중 스트림을 이용하는 병렬전송 기법으로[2] 크게 분류된다. TCP 버퍼튜닝은 네트워크 관리자가 네트워크 성능측정 도구를 이용해 얻은 파라미터들을 이용하여 적절한 버퍼 크기를 설정하는 수동튜닝과[3] 관리자의 개입 없이 튜닝데몬이나[4] TCP 스택 내의 튜닝 알고리즘을[5][6] 통해 동작하는 자동튜닝으로 구분된다.

또한 자동튜닝은 튜닝시점에 따라 연결 설정 시에만 튜닝을 실시하는 정적튜닝과[4] 네트워크 상황의 변화에 따라 지속적으로 동작하는 동적튜닝으로[5][6] 나누어진다. 이러한 튜닝 기법들은 유선 네트워크 고속전송에 충분한 성능이 검증되고[7], 특히 수동튜닝은 가장 우수한 성능을 나타낸다. 본 논문에서는 수동튜닝을 이용한 TCP 버퍼 조절 기법을 줄여서 TCP 버퍼튜닝 기법이라 한다.

TCP 전송률은 전송 종단에 위치한 TCP 네트워킹 스택에 의해 사용되는 운영체제의 시스템 파라미터를 조절함으로써 성능이 향상된다. TCP 버퍼튜닝은 일반적으로 TCP가 링크를 가득 채우

기 위해 필요한 충분히 큰 윈도우를 사용할 수 있도록 메모리 버퍼의 크기를 증가시키는 것이다. TCP는 송신자가 네트워크로 내보내는 데이터양을 조절하기 위해 윈도우 개념을 사용하며 이를 이용한 튜닝의 원리는 그림 1과 같다. 그리고 이는 수신자의 가용한 용량과 다른 링크의 트래픽에 의존적이다. 윈도우는 송신자가 전송했지만 아직 확인응답을 받지 않아 송신자와 수신자 사이에 존재하는 패킷의 양을 결정한다.



〈그림 1〉 TCP 버퍼튜닝 원리

링크의 용량이 항상 채워져 있도록 하기 위해, TCP는 Bandwidth-Delay Product(BDP)[4]라는 이름으로 Round Trip Time(RTT)과 대역폭을 곱한 값보다 크게 윈도우 크기를 유지해야 한다. BDP는 이론적인 최적의 버퍼크기이지만 실제 전송 시에는 네트워크 혼잡이 있기 때문에 일반적으로 TCP는 최대 전송률을 얻기 위해 BDP 보다 같거나 큰 수신 버퍼가 필요하다. 또한 일반적으로 에러로부터 회복하기 위해  $2 * BDP$ 의 송신 버퍼가 요구된다.

## 2.2 무선 랜

IEEE 802.11 무선 랜에서의 TCP는 지형지물로 인한 일시적인 전파장애 및 신호의 간섭, 감쇠로 인하여 유선 네트워크에 비하여 상대적으로 높은 비트 에러율(Bit Error Rate)을 가진다[1]. 이러한 무선 랜의 특성은 TCP가 가정하고 있는 중요한 사실을 유배하는데, 그것은 바로 혼잡판단의 기준이다. TCP는 패킷전송 후 확인응답이 없으면 네트워크 혼잡이라 판단하고 혼잡제어 알고리즘을

수행한다. 하지만 무선 랜에서는 혼잡하지 않은 상태에서 무선 랜 특성으로 인하여 의도하지 않은 혼잡제어 알고리즘을 수행하게 되며, 결과적으로 전체적인 전송률의 감소를 유발한다.

이러한 문제점을 해결하기 위해서 여러 학회와 업체에서 무선 랜에서의 TCP 성능개선 연구를 시작하고, 여러 가지 결과를 제시한다. 거의 모든 연구는 무선 랜에서 TCP 성능을 감소시키는 무선 패킷손실이 네트워크 혼잡으로 오판되는 점에서 시작한다. 연구는 크게 두 가지 방향인 TCP 혼잡제어 알고리즘 자체를 수정하는 방법(TCP-Peach[8], TCP-Peach Plus[9], TCP-Westwood[10])과 중간 게이트웨이(AP) 기술을 수정하는 방법(RED[11], ECN[12])으로 구분된다. 추가적으로 이 두 가지 방향을 혼합하여 사용하는 방법(TCP-Jersey[13])도 연구된다.

TCP 혼잡제어 알고리즘을 수정하는 방법은 새로운 버전의 TCP를 개발하는 것과 거의 동일한 작업이며 무선 랜에 적용되기까지 많은 시간이 걸린다. 또한 게이트웨이 기술을 수정하는 방법은 기존에 배치된 게이트웨이를 교체해야 하는 비용적인 측면의 문제점이 있다.

## 3. 무선 랜 TCP 버퍼튜닝 실험

본 장에서는 유선 네트워크 고속전송 기법인 TCP 버퍼튜닝 기법이 무선 랜 환경에 적용될 경우 전송률과 TCP 버퍼크기와의 상관관계를 분석한다. 다양한 특성을 가지는 무선 랜 환경에서의 TCP 버퍼크기와 전송률에 대한 여러 가지 RTT에서의 실험은 무선 네트워크 특성 분석을 위해서 아주 중요하다. 특히, 실제 대용량 데이터로 실험을 진행하기 위하여 실제 데이터는 음악 연주를 통해 이를 녹음한 고음질의 대용량 음악 파일을 실험에 이용한다. 실험을 통해 우리는 무선 랜의 몇 가지 특성을 발견하고, 특히 TCP 버퍼크기를 증가시켜도 더 이상의 전송률 증가가 없는 한계점을 제시한다. 또한 고속전송을 통해 실시간으로 음악 파일을 다른

장소에 제공할 수 있음을 보여 새로운 음악제공 서비스를 구축할 수 있음을 보인다.

### 3.1 실험환경 및 절차소개

실험을 위하여 송신측 무선 호스트에서 수신측 무선 AP로 IEEE 802.11g를 지원하는 최대 전송률 54Mbps의 무선 네트워크를 구성한다. 무선 호스트와 AP는 리눅스가 설치되어 있으며, 전송률 측정을 위해서 초당 최대한으로 데이터를 전송하는 테스트 툴인 iperf[14]를 사용한다. iperf에서 더미 데이터가 아닌 “차이코프스키 : 피아노 협주곡 1악장”을 실제 피아노 연주를 녹음하면서 고음질의 음악파일로 변환하는 과정 후 그 파일을 실시간으로 수신자에게 전송 한다. 실시간 고품질 고속 전송 실험을 위해 기존에 있던 음악파일이 아닌 실제로 연주하고 이를 실시간으로 녹음하여 변환 과정을 거쳐 생성된 파일을 통하여 실험한다. 본 실험으로 연주회 도중에 연주되는 음악을 무선랜을 통하여 실시간으로 다른 장소에 있는 청중에게도 고음질 음악의 서비스가 가능한지를 알 수 있다. 또한 TCP 버퍼크기를 설정하기 위해서 iperf의 -w 옵션을 사용하고, 이는 리눅스 커널 변수인 SO\_RCVBUF, SO\_SNDBUF를 통한 setsockopt() 함수 접근으로 TCP 버퍼크기를 조절한다. 무선 호스트와 AP간 다양한 지연시간을 구성을 위하여, 네트워크 에뮬레이터인 NetEm[15]을 이용하여 무선 호스트에서 지연시간을 구성한다. 표 1은 실험에 사용된 호스트 정보이다.

〈표 1〉 무선 호스트 및 AP 정보

	Sender (wireless host)	Receiver (wireless AP)
OS	Fedora Core 5 v2.6.16	Redhat 9 v2.4.25
CPU	Intel P4 3GHz	Geode by NSC 233Mhz
Memory	1024MB	64MB
TCP Buffer Size	From 16KB to 1MB	64KB
NIC	802.11 g, Max 54Mbps	802.11g, Max 54Mbps
NIC model	D-Link DWL-AG530	D-Link DWL-G650
Linux AutoTuning	Off	Off

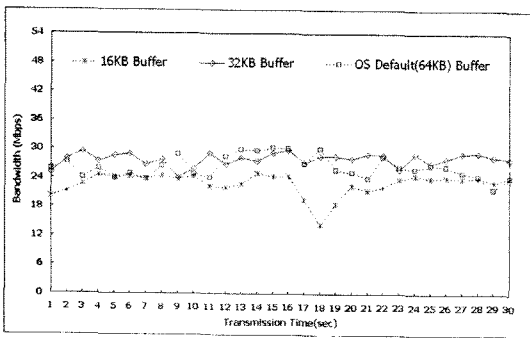
TCP 버퍼튜닝 기법이 무선 랜 환경에서 미치는 영향을 명확하게 분석하기 위해서 리눅스 v2.4 이상부터 지원되는 리눅스 오토튜닝(Linux AutoTuning) 기능을 비활성화 한다. 무선 AP의 TCP 버퍼크기는 다른 AP와 비슷하게 운영체제 기본값인 64KB로 설정하고, 무선 호스트는 최소 16KB에서 최대 1MB까지의 TCP 버퍼크기를 설정한다. 무선 호스트와 AP간의 물리적인 거리는 약 10ms 정도이며 30초 동안 3회 데이터 전송하여 평균을 구한다. 무선 AP의 평균 노이즈는 약 -100dBm로 주변에 다른 무선 신호가 거의 없음을 확인한다. 또한 신호레벨은 -32dBm로 신호대 잡음비(Signal-to-Noise Ratio, SNR)는 평균 68dBm 정도이다. 청취 가능한 최소 신호대잡음비가 20dBm 임을 가만할 때 실험에 사용된 무선 호스트와 AP사이의 신호감도는 상당히 좋다.

### 3.2 RTT 1ms 구간에서 TCP 버퍼튜닝

무선 호스트와 AP사이의 RTT가 1ms인 경우 무선 호스트의 TCP 버퍼크기를 16KB, 24KB, 32KB, 운영체제 기본값(64KB)으로 설정하고 데이터를 전송한다. 리눅스를 포함한 대부분의 운영체제는 기본값으로 64KB의 TCP 버퍼를 보유한다. TCP 버퍼가 16KB인 경우에는 평균 22.8Mbps의 전송률을, 24KB인 경우에는 평균 25.4Mbps의 전송률을 획득한다. 16KB인 경우 18초 구간에서 낮은 전송률을 나타내는데, 이는 무선 랜 환경의 일시적인 혼잡에 의해서 발생한 것으로 여기에 대한 원인은 향후 연구에서 다룰 계획이다. 본 논문에서는 오직 무선 랜 환경에서 TCP 버퍼튜닝을 통해서 얻을 수 있는 최적 전송률에만 초점을 맞춘다.

TCP 버퍼크기가 32KB인 경우 최대 29.8Mbps와 평균 27.8Mbps의 전송률을, 운영체제 기본 버퍼크기로 설정할 경우 최대 30.3Mbps와 평균 26.3Mbps의 전송률을 획득한다. 중요한 것은 TCP 버퍼크기가 32KB인 경우와 운영체제 기본값인 경우의 전송률 차이가 없다는 것이다. 따라서

32KB 크기로 TCP 버퍼를 설정하고 데이터를 전송할 때부터 무선 랜 환경에서 허용하는 최대 전송률을 수렴한다는 것을 알 수 있다. 또한 오히려 32KB인 경우보다 두 배의 버퍼크기를 할당해서 데이터를 전송하는 운영체제 기본값인 경우 불안정한 전송 상태를 보인다. 이번 실험을 통하여 RTT가 1ms 구간에서 운영체제 기본 버퍼크기로 데이터를 전송할 때 불필요한 메모리 낭비와 불안정한 전송률을 나타냄을 확인한다.



〈그림 2〉 RTT 1ms 구간에서 TCP 버퍼크기에 따른 전송률 비교

### 3.3 RTT 5ms, 10ms, 20ms 구간 TCP 버퍼튜닝

이번에는 네트워크 애플레이터인 NetEm을 사용하여 무선 호스트와 AP사이의 RTT를 5ms, 10ms, 20ms로 증가하면서 32KB, 64KB, 80KB, 96KB, 128KB, 256KB로 TCP 버퍼를 설정하고 실험한다. NetEm은 리눅스 커널 v2.6.8 이상부터 내

장되어 있는 모듈로 리눅스 tc(traffic control) 명령어와의 연동되어 사용된다. NetEm은 커널의 TCP/IP단과 네트워크 장치단 사이에 새로운 모듈인 Queuing Discipline를 제공하고 나가는 방향의 패킷에 대하여 지연시간, 손실률, 패킷 중복 및 재순서화의 기능을 추가한다. 실험에서는 무선 호스트에서 NetEm을 활성화하고 무선 AP로의 패킷에 지연시간을 추가한다. 이때 사용된 큐잉 정책은 기본설정인 FIFO(First In First Out)이다.

RTT를 5ms로 구성하고 총 6가지 TCP 버퍼크기(32KB, 64KB, 80KB, 96KB, 128KB, 256KB)로 데이터 전송 실험을 수행한 결과, 96KB에서 최대 27.6Mbps, 평균 25.5Mbps의 전송률을 보인다. TCP 버퍼크기 96KB 이하로는 네트워크 수용력을 전부 활용할 수 없고, 그 이상으로는 메모리의 낭비를 유발한다. RTT를 10ms로 설정한 경우 버퍼크기 96KB에서 최고 24.6Mbps, 평균 22.1Mbps로 최적의 전송률을 보장하는 TCP 버퍼크기임을 확인한다. 마지막으로 RTT를 20ms로 구성한 경우 역시 96KB에서 최고 18.5Mbps, 평균 17.3Mbps의 최적의 전송률을 나타내며, 오히려 256KB로 TCP 버퍼를 설정한 경우 전송률이 낮아지는 현상을 보인다. 과도한 크기의 TCP 버퍼는 무선 네트워크와 AP에 과부하를 유발하고, 이는 불안정한 데이터 전송의 원인이 된다. 결과적으로 데이터 전송률은 감소한다.

실험을 통해서 IEEE 802.11 무선 랜 환경에서 TCP 버퍼크기와 전송률 증가에 관한 두 가지 특성을 확인한다. 첫 번째는 RTT가 증가할수록 확

〈표 2〉 RTT 5ms, 10ms, 20ms 구간에서 TCP 버퍼크기에 따른 전송률 비교

Buffer Size \ RTT	32KB		OS Default (64KB)		80KB		96KB		128KB		160KB		256KB	
	Max.	Avg.	Max.	Avg.	Max.	Avg.	Max.	Avg.	Max.	Avg.	Max.	Avg.	Max.	Avg.
5ms	20.2	17.9	24.8	21.2	25.2	24	27.6	25.5	28.2	25.9	N/A	N/A	28.2	25.1
10ms	N/A	N/A	19.3	18.2	N/A	N/A	24.6	22.1	24.2	22.2	23.2	21.6	23.3	21.8
20ms	N/A	N/A	12.6	11.9	N/A	N/A	18.5	17.3	18.7	17.3	N/A	N/A	18.7	16.8

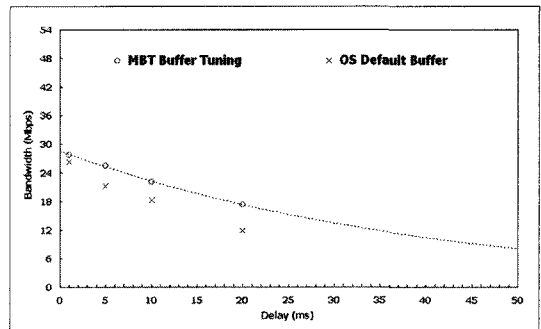
득 가능한 최대 전송률은 낮아진다는 것과 두 번째는 일정한 크기 이상으로 TCP 버퍼를 설정하여도 최대 전송률의 증가는 없다는 것이다. 이러한 두 가지 현상은 유선 네트워크에서도 동일하게 발생된다. 유선 네트워크에서 최대 전송률을 보장하기 위한 TCP 버퍼튜닝 기법은 BDP에 기반하여 계산된 버퍼크기로 설정하고, 양단간의 RTT가 증가함에 따라 TCP 버퍼크기도 증가해야 한다. 따라서 TCP 버퍼튜닝을 적용한 호스트는 모든 RTT 구간에서의 최대 전송률이 같아야 한다. 하지만 이는 이론적인 공식일 뿐 실제 네트워크에서는 TCP 버퍼크기가 커질수록 CPU, 랜카드, I/O 인터럽트와 같은 호스트 성능의 영향을 받기에 RTT당 최대 전송률은 점점 낮아진다. 무선 랜 환경에서는 위와 같은 영향뿐 아니라 추가적으로 무선 신호의 간섭 및 감쇠 등의 영향에 의해 RTT당 전송률 감소는 더 크게 발생한다. 본 논문에서 우리는 TCP 버퍼크기 증가에 관계없이 더 이상의 전송률이 증가되지 않는 지점을 최대버퍼한계(Maximum Buffer Threshold, MBT)라 정의하고, 무선 랜 환경에서 TCP 버퍼튜닝의 기준으로 제시한다.

#### 4. 무선 랜 환경에서의 최대버퍼한계점

최대버퍼한계는 무선 랜 환경에서 최대 전송률을 얻기 위한 최적의 TCP 버퍼크기를 의미한다. 따라서 최대버퍼한계 이상으로 TCP 버퍼크기를 설정해도 전송률은 더 이상 증가하지 않는다. 이번 장에서는 최대버퍼한계 기반으로 TCP 버퍼크기를 설정한 경우 운영체제 기본 전송률에 비하여 얼마나 많은 전송률의 증가가 있는지 알아본다. 또한 RTT 증가에 따른 전송률 감소를 바탕으로 특정 RTT 구간에서 이용 가능한 전송률을 예측한다. 마지막으로 특정 RTT 구간에서 최대버퍼한계 이상으로 TCP 버퍼크기를 설정해도 전송률의 증가가 없음을 확인하고, 이를 통해서 최대버퍼한계점의 정당성을 검증한다.

#### 4.1 최대버퍼한계 기반 TCP 버퍼튜닝

BDP 기반 유선 네트워크 고속전송 기법인 TCP 버퍼튜닝 기법을 무선 랜 환경에 그대로 적용할 경우 전송률은 증가되지만 상당한 양의 불필요한 자원 낭비를 유발하고, 불안정한 전송을 보인다. 무선 랜에 BDP 기반 TCP 버퍼튜닝 아주 비효율적이고, 다른 기준의 TCP 버퍼튜닝이 필요하다. 우리는 이러한 기준점을 최대버퍼한계로 구성하고, 최대버퍼한계에 기반한 TCP 버퍼크기를 설정하고 추가적인 실험을 한다. 실험환경은 이전과 동일하고 RTT 1ms, 5ms, 10ms, 20ms 구간에서 최대버퍼한계 기반 TCP 버퍼튜닝 전송률과 운영체제 기본 전송률을 비교한다.



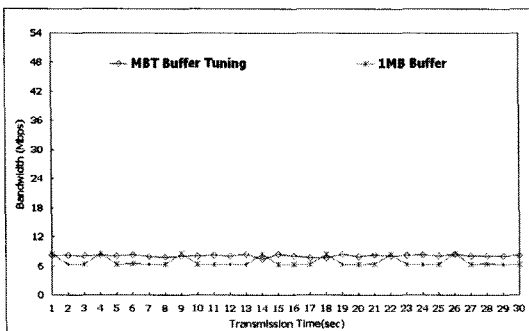
〈그림 3〉 MBT 기반 버퍼튜닝 vs. 운영체제 기본 전송률 비교

실험 결과 RTT 1ms 구간에서 최대버퍼한계 기반 TCP 버퍼튜닝과 운영체제 기본 버퍼에서의 획득된 전송률은 거의 비슷하다. 하지만 최대버퍼한계점이 32KB임을 가만할 때 운영체제 기본 버퍼인 경우보다 50% 메모리를 절약한다. RTT 5ms 구간에서 최대버퍼한계 기반 TCP 버퍼튜닝은 20.3%의 전송률이 향상되고, RTT 10ms와 20ms 구간에서는 운영체제 기본 전송률 보다 각각 21.4%, 45.4%의 전송률 향상을 보인다. 무선 랜 환경의 모든 RTT 구간에서 최대버퍼한계 기반 TCP 버퍼튜닝은 운영체제 기본 버퍼보다 안정적이고 우수한 전송률을 나타냄을 입증한다.

## 4.2 전송률 증가 기반 특정 RTT 전송률 예측

그림 3은 RTT 1ms, 5ms, 10ms, 20ms 지점에서 MB 기반 TCP 버퍼튜닝의 최대 전송률을 나타낸다. 이 수치를 기반으로 회귀분석을 적용하고, 하나의 회귀식을 찾아 그 결과를 그림 3의 점선(추세선)으로 표현한다. 이러한 식은 단일 회귀분석 도구를 제공하는 응용프로그램을 이용하면 쉽게 획득 가능하다. 이를 통해서 우리는 회귀식  $y = 29.372e^{-0.0253x}$ 를 산출하고, 이 식을 기반으로 RTT 50ms 지점에서 평균 전송률을 8.32Mbps로 예측한다. 실제 실험을 통한 검증을 위해서 네트워크 에뮬레이터인 NetEm을 이용하여 RTT 50ms 구간을 구성하고 iperf를 이용한 전송 실험결과, 평균 8.08Mbps의 전송률을 획득한다. RTT 50ms 구간에서 회귀식을 통한 예측 정확도가 약 97% 정도로 RTT 증가에 따른 전송률 예측이 상당히 정확함을 보인다.

또한 이전과 동일한 실험 환경에서 TCP 버퍼 크기를 1MB로 설정하고 데이터를 전송한 결과, 평균 6.83Mbps의 전송률을 보인다. 이를 통해서 최대버퍼한계 이상의 버퍼크기는 더 이상의 전송률을 증가시키지 못함을 확인하고, 오히려 과도한 크기의 버퍼는 불안정한 전송을 나타내며 전반적인 전송률 감소를 유발한다.



〈그림 4〉 RTT 50ms 구간에서 전송률 비교

## 5. 결론

본 논문은 IEEE 802.11 무선 랜 환경에서의 TCP 전송률 증가를 위해서, 유선 네트워크의 고속전송 기법인 TCP 버퍼튜닝 기법을 무선 랜 환경에 적용하고 전송률과 버퍼크기와의 상관관계를 분석한다. 분석을 통해 TCP 버퍼크기가 증가해도 더 이상의 전송률 향상이 없는 지점인 최대버퍼한계를 제시한다. 최대버퍼한계 기반 TCP 버퍼튜닝은 운영체제 기본 전송률보다 RTT 1ms 구간에서 50%의 메모리 절약하고, RTT 5ms 이상의 구간에서 최소 7%이상의 전송률을 향상한다. 또한 특정 RTT 구간에서의 전송률을 3% 오차로 예측하고, 최대버퍼한계 이상으로 버퍼크기를 설정하는 것은 전송률 증가에 영향이 없음을 확인한다. 또한 연주를 통해 생성된 고음질의 음악 파일 전송 실험은 고음질의 대용량 음악파일을 실시간으로 무선랜을 통하여 장소에 구애받지 않고 제공할 수 있음을 알 수 있다.

향후 계획으로 무선 네트워크에서의 의도하지 않은 패킷손실을 구별하기 위해 무선 AP에서 발생하는 비콘(beacon) 신호 강도를 판단기준의 요소로 접목하는 연구를 수행할 것이다. 또한 비콘 신호 강도를 최대버퍼한계 기반 TCP 버퍼튜닝의 인자로 설정하여 무선 랜 환경을 더 정확하게 반영하는 무선 랜 고속전송 모듈 개발을 계획하고 있다.

## Acknowledgment

본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음. IITA-2006-(C1090-0603-0046).

교신저자 : 추현승

## 참고 문헌

- [1] G. Xylomenos and G. C. Polyzos, "TCP Performance Issues over Wireless Links," *IEEE Communications Magazine*, vol. 39, pp. 52-58,

- April 2001.
- [2] H. Sivakumar, S. Bailey, and R. L. Grossman, "PSockets: The Case for Application-level Network Striping for Data Intensive Applications using High Speed Wide Area Networks," the IEEE/ACM SC2000, pp. 38-43, Nov. 2000.
- [3] B. Tierney, "TCP Tuning Guide for Distributed Applications on WAN," In USENIX&SAGE Login, <http://www-dicd.lbl.gov/tcp-wan.html>, Feb. 2001.
- [4] B. L. Tierney, D. Gunter, J. Lee, and M. Stoufer, "Enabling Network-Aware Applications," IEEE-HPDC, pp. 281-288, Aug. 2001.
- [5] J. Semke, J. Mahdavi, and M. Mathis, "Automatic TCP Buffer Tuning," ACM SIGCOMM, vol. 28, no. 4, pp. 315-323, Oct. 1998.
- [6] M. K. Gardner, W. Feng, and M. Fisk, "Dynamic Right-Sizing in FTP (drsFTP): Enhancing Grid Performance in User-Space," IEEE-HPDC, pp. 42-49, July 2002.
- [7] E. Weigle and W. Feng, "A Comparison of TCP Automatic Tuning Techniques for Distributed Computing," IEEE-HPDC, pp. 265-272, July 2002.
- [8] I. F. Akyildiz, G. Morabito, and S. Palazzo, "TCP-Peach: A new congestion control scheme for satellite IP Networks," IEEE/ACM Trans. Networking, vol. 9, pp. 307-321, June 2001.
- [9] I. F. Akyildiz, X. Zhang, and J. Fang, "TCP-Peach+: Enhancement of TCP-Peach for satellite IP Networks," IEEE Commun. Lett., vol 6, pp. 303-305, July 2002.
- [10] C. Casetti, M. Gerla, S. Mascolo, M. Y. Sanadidi, and R. Wang, "TCP Westwood: Bandwidth estimation for enhanced transport over wireless links," ACM Mobicom, pp. 287-297, July 2001.
- [11] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," IEEE/ACM Trans. Networking, vol. 1, pp. 397-413, Aug. 1993.
- [12] S. Floyd, "TCP and explicit congestion notification," ACM Comput. Commun. Rev., vol. 24, pp. 10-23, Oct. 1994.
- [13] K. Xu, Y. Tian, and N. Ansari, "TCP-Jersey for Wireless IP Communications," IEEE JSAC., vol. 22, no. 4, pp. 747-756, May 2004.
- [14] M. Gates, A. Tirumala, J. Dugan, and K. Gibbs, Iperf, <http://dast.nlanr.net/Projects/Iperf/>
- [15] <http://linux-net.osdl.org/index.php/Netem>



◎ 저자 소개 ◎



**문 성 곤(Sung-Gon Mun)**

2005년 동서대학교 정보통신공학과 졸업(학사)  
2007년 성균관대학교 대학원 컴퓨터공학과 졸업(석사)  
2007년~현재 (주)엠텔로 근무  
관심분야 : Wireless TCP, High-speed Data Transfer etc.  
E-mail : msgon@ece.skku.ac.kr



**이 흥 석(Hongseok Lee)**

1998년 성균관대학교 수학과 졸업(학사)  
2002년 성균관대학교 대학원 전기전자 및 컴퓨터 공학부 졸업(석사)  
2004년 성균관대학교 대학원 전기전자 및 컴퓨터 공학부 박사과정 수료  
2005년~현재 성균관대학교 컨버전스연구소 연구원  
관심분야 : 유무선 네트워크, 고성능컴퓨팅 etc.  
E-mail : juspace@hanmail.net



**추 현 승 (Hyunseung Choo)**

1988년 성균관대학교 수학과 졸업(학사)  
1990년 University of Texas 컴퓨터공학과 졸업(석사)  
1996년 University of Texas 컴퓨터공학과 졸업(박사)  
1997년 특허청 심사4국 컴퓨터심사담당관실(사무관)  
1998년~현재 성균관대학교 정보통신공학부 부교수  
2001년~현재 한국인터넷정보학회/한국시뮬레이션학회 이사  
2004년~2006 대통령직속 교육혁신위원회 전문위원  
2004년~현재 한국인터넷정보학회 논문지편집위원장  
2005년~현재 건강보험심사평가원 전문위원  
2005년~현재 한국정보과학회 논문지편집위원  
2005년~현재 정보통신부ITRC 지능형HCI융합연구센터장  
관심분야 : 유/무선/광네트워킹, 모바일컴퓨팅, 임베디드SW, 그리드컴퓨팅  
E-mail : choo@ece.skku.ac.kr



**공 원 영(Wonyoung Kong)**

1989년 서울대학교 음악대학 졸업(학사)  
1994년 Southern Methodist University 졸업(석사)  
1999년 University of North Texas 졸업(박사)  
2004년~현재 수원대학교 음악대학 피아노학과 전임교수  
E-mail : kongchoo@komet.net