

# CDMA망에서 모바일 단말에 기반한 TCP 성능 개선

권오근<sup>o</sup> 차호정  
연세대학교 컴퓨터과학과  
{kok,hjcha}@cs.yonsei.ac.kr

## A Terminal-based TCP Performance Improvement in CDMA Networks

Ohkeun Kwon<sup>o</sup> Hojung Cha  
Dept. of Computer Science, Yonsei University

### 요약

본 논문은 CDMA2000 1x 망에서 이동단말 측의 측정정보와 TCP 동작 수정을 통하여 TCP 윈도우 크기를 제한하고 RTT (Round Trip Time)을 일정 수준으로 안정화 시키는 방안을 제시한다. CDMA2000 1x망은 망 구성의 특성으로 인하여 큰 RTT 값을 가지며 변화가 큰 전송률을 제공한다. TCP는 이러한 CDMA2000망에서 지나치게 큰 전송 윈도우로 인하여 변동이 큰 RTT를 보이며 베이스 스테이션의 버퍼링 부하를 증가시킨다. 본 논문에서는 CDMA2000 1x망에서 사용되는 PPP프로토콜을 이용하여 무선 최대 전송률을 측정하고, TCP timestamp option을 이용하여 RTT를 측정하는 방안을 제시한다. 또한 이 측정 값을 적용하여 TCP 수신측에서 RTT와 베이스 스테이션의 버퍼링을 일정수준으로 유지시키는 방안을 제시하고, 리눅스 프로토콜 스택 수정을 통해 적용 결과를 보인다.

### 1. 서론

CDMA2000 1x 망은 최대 144Kbps의 전송률을 가져 최근 상용화 되고 있는 IEEE 802.11과 같은 무선 연결 기술보다 낮은 전송률만을 제공한다. 하지만 넓은 지역에 서비스가 가능한 특성 때문에 고속의 다른 무선 연결 기술과 공존할 가능성이 크다. 반면에 CDMA2000망은 음성통화와 다수의 사용자를 고려하여 설계되었다. 따라서 데이터 통신에서 복잡한 물리 계층, MAC 계층 처리가 필요하며 따라서 긴 RTT 값을 가진다. 또한 사용자의 위치와 무선 상태에 따라 대역폭이 크게 변하게 된다. TCP는 일반적으로 순간적인 전송량이 크게 변하는 특성을 가진다. 전송률이 제한된 CDMA망에서 이런 전송량의 큰 변화는 TCP RTT를 크게 증가시켜 사용자가 느끼는 서비스 지연시간을 크게 한다. 또한 베이스 스테이션에서의 과도한 큐잉으로 인한 부하가 있으며 잦은 RTO (Retransmission Time Out) 발생을 일으키게 된다. TCP RTT 제한에 대하여 Hsiao[1]는 비디오 스트리밍에 TCP를 이용하기 위하여 중간 라우터의 버퍼링을 최소화하는 방안을 제시하였다. 이 방안은 라우터에서의 복잡한 큐 관리와 예측이 필요한 단점이 있다. 또 Koga[2]는 이론적인 CDMA2000망의 전송률과 RTT를 이용하여 TCP advertised window 크기를 조정하는 방법을 제안하였다. 하지만 전송률과 RTT 값을 측정하는 방안을 제시하지 않았고 시뮬레이션을 통해 이상적인 상황에서의 성능 측정을 하였다.

본 논문에서는 CDMA2000 1x 데이터 서비스에서 사용되는 PPP 프로토콜의 특성과 TCP timestamp option을 이용하여 무선 연결의 최대 전송률과 RTT를 측정하는 방안을 소개한다. 또한 이 두 값을 입력으로 하여 TCP advertised window 크기를 계산하여 TCP 수신자 측에서 전송률을 일정하게 제한하는 방안을 제시한다. 또한 리눅스 프로토콜 스택을 수정하여 CDMA2000 1x망에서의 성능평가 결과를 제시한다.

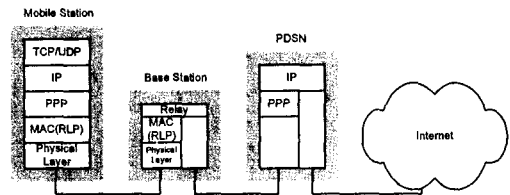
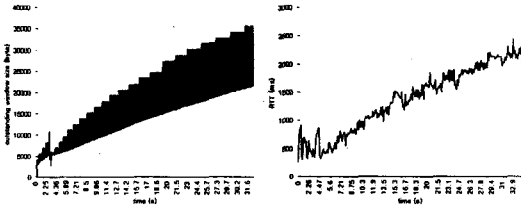


그림 1 CDMA2000 1x 패킷 데이터 서비스 모델

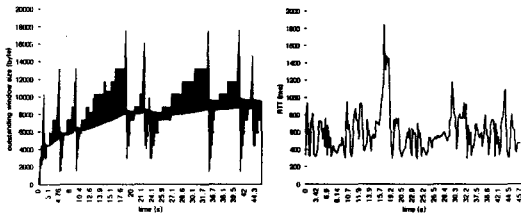
### 2. CDMA2000 1x 망에서 TCP 전송 특성

그림 1은 CDMA2000 망에서 IP 패킷 전송 서비스 지원을 위해 사용되는 프로토콜의 구조이다[3]. CDMA2000 무선 연결에서 IP 패킷 전송 시 생기는 지연은 크게 패킷 처리를 위한 고정 지연 시간과, 무선연결을 통해 패킷이 전송되는 전송시간으로 나눌 수 있다. CDMA2000 망의 물리계층은 다수의 음성통화를 위주로 설계되었다. 전파 대역 공유를 위해 사용되는 CDMA 방식은 한정된 전파 대역 이용의 효율성을 높이기 위해 복잡한 인코딩과 필터링이 필요하다. 기본적인 오류 처리로 물리계층에서 FEC(Forward Error Correction)를 이용하여 비트 오류를 검출하고 복구 한다. 데이터 서비스의 경우 음성과 달리 비트 오류에 민감하므로 추가적으로 MAC(Medium Access Control) 계층에서 RLP(Radio Link Protocol)을 채용하여 한정된 헷수의 재전송을 이용한 오류 복구를 하고 있다[3]. 또한 하나의 베이스스테이션에서 셀 안에 존재하는 많은 이용자를 지원하기 위해 복잡한 큐 관리와 무선 전파 세기 정책이 적용된다. 따라서 하나의 IP 패킷이 전송되기 위해서는 물리, MAC 계층에서의 처리와 베이스스테이션에서의 큐 관리 등의 과정을 거쳐야 하며, 이런 특성으로 인하여 무선 연결에서의 데이터 전송이 일어날 때 패킷 처리를 위한 고정 지연 시간이 일반적인 유선 연결에 비해 크다. 한 사용자의 가용 대역폭에서도 큰 변화를 보인다. 하나의 무선 단말에서 사용 가능한 전송 대역폭은 셀 안에 위치하는 사용자의 수, 사용자와 베이스스테이션 간의 거리와 무선 상태 등을 고려하여 베이스스테이션에서 결정된다. 또한 사용자의

• 본 연구는 한국과학재단에서 지원하는 특정기초연구사업으로 수행하였음 (과제번호 : R01-2002-000-00141-0).



(a) 전송자가 사업자 망 내부에 위치하는 경우



(b) 전송자가 외부 망에 위치하는 경우

그림 2 TCP 전송자의 위치에 따른 RTT와 outstanding 윈도우 크기 변화

이동이나 전파 간섭으로 인하여 비트 오류가 증가하는 경우 MAC 계층 재전송으로 인하여 순간적인 지연이 발생될 수 있다. 따라서 패킷 자체의 전송 시간 역시 크게 변화하게 된다.

그림 2는 상용 서비스 중인 CDMA2000 1x망에서의 TCP 데이터 전송 시 전송 측에서의 측정 결과이다. 그림 2(a)는 사업자의 망 내부에서 실험한 결과이다. 가장 작은 RTT값이 250ms 이상으로 일반적인 유선 연결에 비하여 큰 값을 가지는 것을 볼 수 있다. 또한 전송이 진행되면서 congestion window 크기가 증가되어 RTT 값이 지속적으로 증가하는 것을 볼 수 있다. 이것 양단간의 네트워크 상에서 데이터 패킷이 버퍼링 되고 있음을 의미한다. CDMA2000 1x망의 액세스 망과 무선 연결의 전송 대역폭을 고려할 때 버퍼링이 일어나는 병목지점은 무선연결일 가능성이 크며, 이 경우 TCP 데이터 패킷은 PDSN에 할당된 버퍼에 저장된다. 이런 베이스스테이션에서의 버퍼링은 수많은 단말을 관리하는 베이스스테이션에 큰 부담이 된다. 또한 베이스스테이션에서의 한 단말에 대한 패킷 큐잉 정책이 FIFO라고 가정하면 TCP 데이터 전송이 일어나는 동안 해당 단말로 전송되는 모든 패킷이 큰 지연시간을 가지게 된다. 그림 2(b)는 TCP 전송 호스트가 일반적인 유선 네트워크에 위치하는 경우이다. 큰 outstanding 윈도우 크기를 보이며, outstanding 윈도우 크기에 따라 RTT값이 증감하는 것을 확인할 수 있다. 또한 RTT 증가에 따라서 TCP RTO가 발생하면서 outstanding 윈도우 크기를 초기화하는 것을 볼 수 있다. 이렇게 Outstanding 윈도우가 증가된 상태에서 RTO가 발생하는 경우 베이스스테이션에 버퍼링 되어 있는 TCP 데이터 패킷이 소진될 때까지 긴 시간이 걸리면서 순간적으로 길어진 RTT 때문에 불필요한 TCP 재전송이 일어날 가능성이 크다.

이런 TCP 전송의 문제점을 해결하기 위해 본 논문에서는 TCP 수신 측에서의 무선 상태 측정과 RTT 측정을 이용한 전송률 조정 방법을 제시한다.

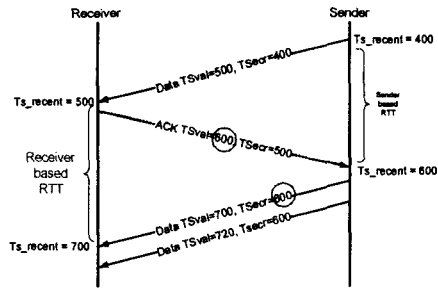


그림 3 TCP timestamp 동작의 예

### 3. 모바일 호스트 측정 정보를 이용한 TCP의 RTT 개선

TCP는 전송률 조절을 위해 송신 측에서의 전송 윈도우 크기 조절을 이용한다. 전송 윈도우의 크기는 네트워크 상태에 따라 전송 측에서 결정되는 congestion 윈도우와 수신 측의 수신 버퍼 상태를 나타내는 advertised 윈도우 크기 중 작은 값으로 결정된다. 일반적으로 TCP 전송을 최적화하기 위한 TCP 수신 버퍼 크기는 bandwidth-delay product 값이 권장된다. 하지만 이런 수신 버퍼 설정은 네트워크 연결 매체에 따라 크게 다르다. CDMA2000의 경우 지속적으로 RTT와 대역폭이 변화되므로 고정된 값을 설정하기는 힘들다. 본 논문에서는 식 (1)과 같이 무선의 전송률과 RTT 값을 측정하여 최적의 advertised window 크기를 결정하는 방안을 제시한다.

$$\text{buffer\_size} = \text{무선최대 전송률} \times (\text{RTT} + a) \quad (1)$$

무선 최대 전송률은 측정된 무선 연결의 최대 전송률을 의미한다. 유선 연결과 무선 연결의 전송률 차이로 인하여 bandwidth-delay product에 적용되는 병목 연결은 무선 연결일 가능성이 크다. 또한 전체 연결의 전송률을 무선의 최대 전송률을 넘지 못하므로 advertised window 크기 계산에 무선의 최대 전송률을 반영하였다. 따라서 TCP 전송률은 무선 최대 전송률 이하로 유지된다. 최대 전송률을 측정하기 위해 데이터 서비스에 사용되는 PPP 프로토콜을 수정하였다. 수신측에서 PPP 프로토콜은 CDMA2000 망의 MAC 계층을 통해 전송된 바이트 스트림을 구분하여 IP패킷으로 분리하는 역할을 하므로 각 패킷의 전송에 필요한 시간을 정확히 계산할 수 있다. 이 패킷 전송 시간과 패킷 크기를 이용하여 패킷 한 개의 순간 전송률을 계산할 수 있다. 순간측정값의 오류를 줄이기 위해 순간 전송률에 EWMA (Exponentially Weighted Moving Average) 필터를 적용하여 동적 평균 값을 이용하였다. TCP 수신측에서 RTT 값을 측정하기 위해 TCP timestamp 옵션을 이용하였다. TCP timestamp 옵션은 송신측에서 RTT를 전송 윈도우의 한변 측정하는 기존의 TCP RTT 측정에 비해 정확히 측정하기 위해 제안된 옵션이다. 하지만 일반적인 timestamp 옵션의 동작에 의해 그림 3과 같이 수신측에서도 RTT를 측정할 수 있다. 이 측정 RTT 값에 송신측과 같은 EWMA 필터를 적용하여 smoothed RTT 값을 계산하였다. 식1에 반영된 a값은 패킷 손실 발생이나 수신자의 TCP delayed ack 적용을 고려한 초과 RTT이다.

설정	값
TCP timestamp	사용
TCP window scale	사용 안함
TCP selective ack	사용
MTU	1500 Byte
PPP 헤더 압축	사용 안함
전송량	200 KB (4KB × 50)
threshold α (식1)	100 ms

표 1 실험 설정

4. 실험

제시한 기법의 성능평가를 위하여 리눅스 프로토콜 스택을 수정하여 알고리즘을 구현하였다. 무선 최대 전송률 측정을 위해 모바일 호스트 측의 PPP를 수정하였다. 측정값의 전달은 프로토콜 스택의 각 계층간에 협조가 필요하므로 큰 부하가 생긴다. 따라서 무선 최대전송률이 특정 경계 값에 도달하는 경우 ICMP 메시지를 생성하여 상위 계층으로 전송률 변화를 통보하도록 하였다. RTT 측정은 리눅스에 기본적으로 구현되어 있는 TCP timestamp option 부분을 수정하였다. 현재의 범용 OS는 TCP 수신 버퍼로 32KB~64KB의 메모리를 사용한다. advertised window 크기의 설정은 이미 할당된 TCP 수신 버퍼의 크기를 줄이지 않고 TCP ack 패킷이 생성될 때 계산된 advertised window 크기이하를 유지하도록 하였다. TCP 수신 버퍼는 TCP 세그먼트 손실이 발생하는 경우 채워지게 된다. 따라서 적용된 알고리즘에 의해 advertised window 크기가 제한된 상황에서 패킷 손실이 발생하는 경우 TCP 수신 버퍼 사용량이 증가되면서 손실에 대한 재전송과 이후 데이터 전송이 지연될 수 있으므로 패킷 손실이 발생한 경우 재전송이 일어날 때까지 일시적으로 advertised window 크기를 실제 수신 버퍼 크기로 조정하였다. 실험은 상용 서비스 중인 CDMA2000 1x망에서 모바일 호스트로 리눅스가 설치된 PC와 유대단말을 이용하였다. TCP 전송을 위해서 네트워크 성능평가에 쓰이는 nttcp를 사용했으며 해석에는 tcpdump와 tcptrace를 사용했다. 프로토콜 설정과 상수는 이 값은 IETF에서 제시한 2.5G 3G 망을 위한 설정을 반영하여 표 1과 같이 설정하였다[4].

그림 1은 제시한 알고리즘이 적용된 경우 advertised window 크기의 변화이다. 측정은 송신측에서 tcptrace를 이용하였다. 알고리즘의 계산 결과가 적용되어 11000 바이트 정도로 advertised window 크기를 유지하는 것을 볼 수 있다. 패킷 손실이 발생하는

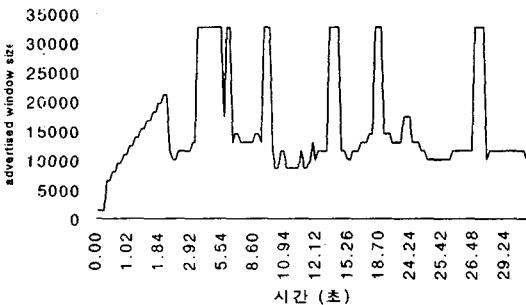


그림 4 advertised window 크기 조정 결과

	RTT 평균 (ms)	RTT 표준편차 (ms)	응용프로그램 전송률 (Byte/sec)
알고리즘 적용	608	188	6955
미적용	718	346	7075

표 2 알고리즘 적용에 따른 전송률 / RTT 변화

경우 advertised window 크기를 순간적으로 리눅스의 기본 수신 버퍼 크기인 32KB로 조정하는 것을 볼 수 있다. 표 2는 5회에 걸친 전송에서 평균을 구한 값이다. 제시된 알고리즘은 수신 측에서의 계산에 의해 TCP 전송률을 적정수준으로 제한한다. 따라서 알고리즘이 적용되지 않는 경우에 비해 TCP 전송률이 낮아지지 않아야 한다. 실험 결과와 제시된 알고리즘을 적용한 경우도 전송률을 유지하는 것을 볼 수 있다. 또 목표한 것처럼 RTT의 평균이 감소하고 RTT 값의 변화가 줄어든 것을 볼 수 있다. 추가적으로 구현한 알고리즘에서 EWMA 필터의 gain값이나 식 1의 α값에 대해서는 최적의 값을 실험을 통해 결정되어야 한다.

5. 결론

본 논문에서는 RTT와 전송률 변화가 심하고 낮은 전송률을 가지는 CDMA2000 1x 망에서 TCP 데이터 전송의 문제점을 실험을 통해 제시하였다. TCP의 기본적인 데이터 전송은 전송 오류가 발생할 때까지 지속적으로 전송률을 증가시키기 때문에 RTT 증가와 RTO 발생과 같은 문제점을 유발한다. 수신측에서의 측정과 advertised window 크기 조절을 통해 수신측에서 TCP의 전송률을 적정수준으로 제한하여 이런 RTT 증가와 RTO 발생의 문제를 해결할 수 있다. 본 논문에서는 수신측에서의 PPP와 TCP 프로토콜 수정만으로 전송률을 제한을 하는 방안을 제시하였다. 실험 결과를 통해 제시한 알고리즘이 데이터 전송률을 제한하지 않으면서 RTT 값을 적정하게 유지하는 것을 보였다. CDMA2000 EV-DO 등 고속의 3G 무선망의 경우 더욱 다단계의 전송률을 가진다. 또한 권고되고 있는 window scaling 옵션이 적용되는 경우 RTT 값과 전송률 변화가 커지므로 제시된 알고리즘의 효과가 커질 수 있다.

참고문헌

- [1] Hsiao, P. H., Kung, H. T., and Tan, K-S. "Streaming Video over TCP with Receiver-based Delay Control," *IEICE Transactions on Communications*, Vol. E86-B, No. 2, pp. 572-584, February 2003.
- [2] Hiroyuki Koga, Kenji Kawahara, and Yuji Oie, "TCP Flow Control Using Link Layer Information in Mobile Networks," *Proceedings of SPIE Conference of Internet Performance and Control of Network Systems III*, Boston, Massachusetts, Vol. 4865, pp.305-315, July 2002.
- [3] 3GPP2, *Data Service Options for Spread Spectrum Systems Addendum 2*, C.S0017-0-2, August 2000.
- [4] H. Inamura, Ed., G. Montenegro, Ed., R. Ludwig, and A. Gurtov, F. Khafizov, *TCP over Second (2.5G) and Third (3G) Generation Wireless Networks*, IETF RFC 3481, February 2003.