

무선환경에서 연속적인 패킷손실을 고려한 TCP 재전송 기법

홍 충 선* · 강 재 신** · 김 대 선***

요 약

무선환경상에서 TCP는 유선과는 달리 높은 비트 에러율을 가지기 때문에 이를 해결하기 위한 TCP가 필요하다. 위성망 같이 손실률이 매우 높은 환경을 위해 TCP Westwood에서는 기존의 벌크 재전송(bulk retransmission) 기법을 보완하였다. 하지만 벌크 재전송 기법은 불필요한 데이터도 함께 재전송하기 때문에 무절제한 패킷 전송으로 인해 혼잡을 가중시킬 우려가 있으며 또한 손실률이 높지 않은 네트워크 환경에서는 전송속도가 낮아진다. 본 논문에서는 네트워크 상황에 따라 Bulk 재전송 패킷수를 조절하는 적응 벌크 재전송 메커니즘 (adaptive bulk retransmission mechanism)을 제시하였고 Markov 에러 모델에서 기존의 TCP Westwood보다 높은 전송 속도를 보임을 NS-2를 이용하여 검증하였다.

Adaptive TCP Retransmission Mechanism for Continuous Packets Loss on Wireless Environment

Choong Seon Hong^{*} · Jae-sin Kang^{**} · Dae-sun Kim^{***}

ABSTRACT

We need to have an adaptive TCP protocol that can be tolerable on wireless network environment. TCP Westwood for use in the environment that have a very high loss rate like a satellite was proposed by modifying the existing bulk retransmission protocol. Bulk retransmission mechanism shows a highly enhanced performance on networks that have a very high loss rate but are prone to bursty loss networks. Also, it can experience less performance on low late transmission environment. This paper proposes Adaptive Bulk Retransmission Mechanism that adjusts the number of bulk retransmitted packets based on the network conditions. The proposed mechanism was evaluated by using NS-2.

키워드 : Wireless TCP, 벌크 재전송, BER

1. 서 론

TCP[1]는 신뢰성 및 혼잡제어(Congestion Control)를 제공하는 표준화된 전송 프로토콜로써 우리가 매우 자주 이용하는 Web(HTTP), e-mail(SMTP), FTP, Telnet 등의 응용뿐만 아니라 대부분의 응용들에서 많이 쓰이고 있다. 기존의 TCP는 패킷의 손실이 유선 환경에서의 혼잡(congestion)에 의해 일어났다고 가정하여 만들어진 알고리즘이기 때문에 무선 환경에서 발생하기 쉬운 링크 상에서의 패킷 에러에는 적절히 대응하지 못한다. 따라서 TCP에서 데이터의 흐름이 느려지게 되고, 이런 부적절한 대응으로 인해 전송속도가 전반적으로 떨어지게 된다. 또한 무선 네트워크는 bursty한 손실이 자주 발생한다. 기존의 TCP는 재전송시 ACK를 확인하면서 하나씩 전송하기 때문에 많은 RTT

(Round Trip Time)소요 및 타임 아웃(Time-Out)으로 인한 CWND(congestion window)의 초기화 때문에 성능이 저하되는데 특히 위성망과 같이 RTT가 클때에는 심각히 전송속도가 느려지게 된다.

최근 무선(wireless) TCP를 위한 연구는 3가지로 분류할 수 있는데, 첫 번째는 TCP HACK[2]이나 Freeze TCP[3]같이 송신측과 수신측이 중간 노드의 도움없이 직접 통신하는 End-To-End 방법이다. 두 번째는 Snoop[4]와 같이 무선 링크에 연결된 노드들의 link layer에서 재전송을 수행함으로써 성능을 향상시키려는 Link Layer 방법이고 마지막으로 세 번째 방법은 I-TCP[5]나 M-TCP[6]처럼 유선 구간과 무선 구간을 분리하여, 이 결합지점에서 특정한 서비스를 수행함으로써 성능을 향상시키는 Split Connection 방법이다. 본 논문은 첫 번째 End-to-End방법으로 Bursty한 손실에 대처하기 위한 방법에 초점을 맞추고 있다.

논문 구성은 다음과 같다. 2장에서는 TCP의 혼잡제어 및 무선 환경에서 TCP 문제점에 대하여 소개하고 3장 관련연

※ 본 연구는 정보통신부 ITRC Project 지원으로 수행되었음.
* 종신회원 : 경희대학교 전자정보학부 교수
** 준 회원 : 경희대학교 대학원 전자공학과
*** 준 회원 : 경희대학교 대학원 컴퓨터공학과
논문접수 : 2004년 9월 21일, 심사완료 : 2004년 12월 13일

구에서는 bursty한 손실을 처리할 수 있는 벌크 재전송(bulk retransmission)[11]방법 그리고 무선 네트워크에 효율적으로 적용할 수 있는 TCP Westwood에 대해서 간단히 살펴본다. 그리고 4장에서는 이 논문에서 제안하고자 하는 적응 벌크 재전송 기법을 소개한다. 5장에서는 시뮬레이션을 통한 결과를 제시한 후 마지막으로 6장에서 결론을 맺고 향후과제를 제시할 것이다.

2. TCP의 혼잡제어 및 Wireless TCP의 필요성

2.1 혼잡제어

TCP의 흐름제어는 수신측에서 버퍼에 저장되어 있는 데이터의 양에 따라 변하는 receiver-advertised 윈도우와 중간 네트워크의 혼잡에 따라 변하는 혼잡 윈도우에 의해 조절 된다. 두 윈도우 중 더 작은 크기로 TCP는 윈도우 크기를 정하게 된다. 과거의 네트워크에서는 컴퓨터의 처리 속도가 느리고 버퍼의 크기도 작았기 때문에 receiver-advertised 윈도우에 따라 전송속도가 좌우 되었지만 현재의 네트워크에서는 컴퓨터의 급격한 발달로 인해 receiver-advertised 윈도우에 거의 영향을 받지 않는다. 하지만 인터넷의 사용자가 증가하고 데이터의 종류도 텍스트에서 멀티미디어로 바뀌어 감에 따라 혼잡 윈도우가 전송속도를 좌우하게 된다. TCP Reno에서는 혼잡을 조절하기 위해 다음 네가지 알고리즘을 사용한다[7].

- Slow start
- Congestion avoidance
- Fast retransmit
- Fast recovery

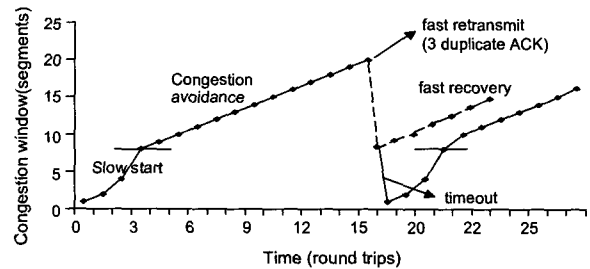
2.2 Wireless TCP의 필요성

1980년대 초반 TCP의 혼잡 제어가 처음 설계될 당시 패킷 전송은 신뢰성이 높은(reliable), 즉 전송중의 오류로 인해 패킷이 손실될 가능성이 매우 낮은 유선 링크상에서 주로 이루어 졌다. 따라서 TCP 혼잡 제어 메커니즘은 모든 패킷 손실을 망에서 혼잡(congestion)이 발생한 것으로 해석하고 이에 대응하는 즉 윈도우의 크기를 큰 쪽으로 감소시키거나 재전송 타임 아웃으로 인해 혼잡 윈도우(congestion window)는 1로 초기화 된다. 그러나 TCP가 무선 링크상에서 동작할 경우 이야기는 달라진다. 무선 환경의 특징을 살펴보자면 다음과 같다.

- 높은 비트 에러율(BER)
- Mobile node가 BS와 연결이 끊어짐으로 인한 에러
- 낮은 대역폭
- 페이딩 현상등으로 인한 bursty한 에러

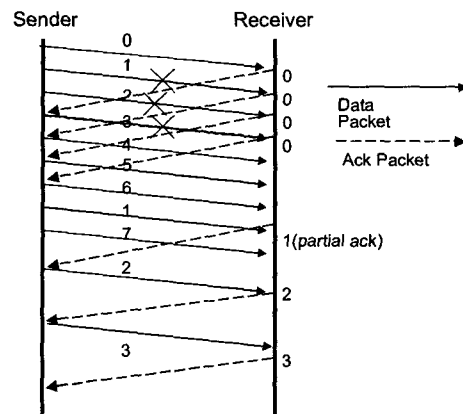
무선 링크의 비트오류 확률(Bit Error Rate : BER)은 유선 네트워크보다 높을 뿐 아니라 다중 경로 페이딩(Multi-Path Fading)으로 인해 연속적인 패킷손실을 가질 수도 있고 단말의 이동으로 인해 핸드오프가 발생할 때 종종 연결이

재 설정 되기도 한다[9]. 이런 요인들로 인해서 무선 환경에서 동작하는 TCP의 패킷 손실은 망의 혼잡과는 관계없이 발생한다는 점이 문제점의 출발이라고 할 수 있다. 흔히 비혼잡 패킷 손실(Non-Congestion Packet Loss)이라 불리는 이 패킷 손실들은 TCP로 하여금 불필요한 손실 복구 과정이나 RTO를 유발하도록 하여 CWND의 크기를 감소시키고 이는 결국 전송측의 전송 속도의 저하로 이어지게 되는 문제점을 가져온다. 특히 연속적인 패킷손실 즉 bursty한 에러는 TCP의 성능을 크게 떨어지게 한다.



(그림 1) 혼잡 제어 알고리즘

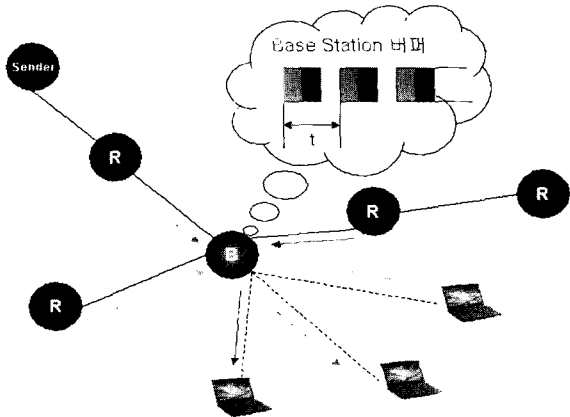
(그림 2)는 Bursty한 손실이 발생 하였을때 TCP의 전송 방법이다. (그림 2)에서처럼 3개의 bursty 손실을 처리하기 위해서는 RTT(Round Trip Time)의 세배의 시간이 필요하다. 그동안 전송측은 타임 아웃으로 인해 CWND의 초기화가 이루어 질수 있고 수신측은 1, 2, 3번 패킷을 재전송 받을 때까지 나머지 패킷을 처리하지 못하고 저장하기 때문에 윈도우의 크기가 작아지게 된다.



(그림 2) 버스트한 손실인 경우의 TCP전송

2.3 혼잡제어

TCP Westwood[10]는 혼잡 윈도우와 slow start 임계값을 직접 조절하기 위해 TCP ACK를 모니터링 함으로써 대역폭을 측정한다. 데이터는 BS(Base Station)이나 중간 라우터의 혼잡 정도에 따라서 지연되어 수신측에 도착하기 때문에 전송측은 ACK 간의 간격 t를 통해서 대역폭을 측정할 수 있다. (그림 3)은 BS의 Q버퍼에 FIFO(First In First Out)되는 것을 보여주고 있다.



(그림 3) TCP Westwood의 예

TCP Westwood의 대역폭 추정 방법은 BE(Bandwidth Estimation), RE(Rate Estimation), ABSE(Adaptive Bandwidth Share Estimation)[8]등이 있다. BE는 ACK들의 간격(t)을 통해서 대역폭을 추정하는 방법이고 RE는 일정 시간 동안 수신된 ACK의 평균 간격을 통해서 대역폭을 추정한다. 하지만 BE와 RE방법은 패킷 손실률에 따라 추정값이 많이 변한다. ABSE는 RE의 추정 시간 간격을 무선 네트워크의 상황에 맞게 조절함으로써 좀더 정확하고 공정한 대역폭을 추정한다.

다음은 3개의 중복(duplicate) ACK를 수신 하였을때 혼잡 윈도우의 재설정을 보여주고 있다. 패킷이 손실 되었을때 추정된 대역폭 ABSE를 이용하여 임계값을 설정하기 때문에 무선에서의 불필요하게 CWND를 줄이는 것을 막을 수 있고 공정하게 전송을 할 수가 있다.

```

if (3 DUPACKs are received)
    ssthresh = (ABSE * RTTmin) / seg_size ;
    if (cwin > ssthresh) /*congestion avoidance*/
        cwin = ssthresh ;
    endif
endif
    
```

다음은 타임 아웃이 되었을때 혼잡 윈도우의 재설정을 보여주고 있다. CWND는 1로 초기화 되지만 임계값은 대역폭으로 설정되었기 때문에 slow start를 통하여 빠르게 전송속도를 높일 수 있다.

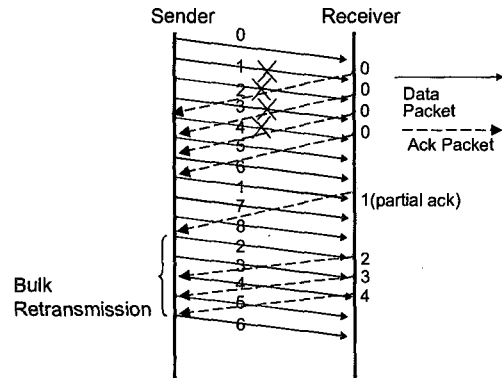
```

if (coarse timeout expires)
    cwin = 1 ;
    ssthresh = (ABSE * RTTmin) / seg_size ;
    if (ssthresh < 2)
        ssthresh = 2 ;
    endif ;
endif
    
```

3.2 벌크재전송

벌크 재전송 기법[11]은 busy 에러가 발생 하였을때 효율적으로 처리할 수 있는 알고리즘이다. (그림 4)는 Bulk 재전송 기법의 예를 보여 주고 있다. 1, 2, 3, 4번 패킷이 연속적으로 에러가 발생 하였을때 수신측은 재전송 요구를 위해

연속적인 0번 패킷에 대한 중복 ACK를 보내게 된다. 전송측에서는 3개의 중복 ACK를 수신하게 되면 1번 패킷을 재전송 하게 되고 수신측은 1번에 대한 ACK를 보내게 된다. (단일 패킷 손실 일때에는 6번 지금까지 수신한 마지막 패킷 즉 6번에 대해서 ACK를 보내게 된다.) 전송측은 이러한 부분적인 ACK를 수신하게 되면 전송하였으나 ACK가 수신되지 않은 모든 패킷을 재전송 하게 된다.



(그림 4) Bulk 재전송의 예

벌크 재전송은 TCP Tahoe와 Reno에서 이미 다룬바가 있다. 하지만 벌크 재전송은 네트워크의 혼잡을 가중시킬 수도 있다. TCP Westwood에서는 이러한 단점을 보완하기 위해 두 가지를 보완하였다. 하나는 고정된 타임 아웃을 쓰는 것이고 다른 하나는 손실이 혼잡 때문인지 에러 때문인지 구분하는 것이다. 기본적으로 TCP는 'binary exponential timeout backoff' 방식을 쓰고 있다. 이 방법은 손실이 일어날 때마다 Timeout 값을 2n 만큼씩 증가시키기 때문에 연속적인 에러가 생겼을 때는 상당히 낮은 전송률을 보이게 된다. TCP Westwood에서는 이러한 단점을 보완하기 위해 고정된 Timeout값을 사용하는데 새로운 ACK를 받을 때마다 n을 0으로 초기화시키기 때문에 무선에서 일어나는 에러로 인해 RTO를 계속적으로 증가시키지는 않는다. 또한 TCP Westwood에서는 에러가 혼잡 때문인지 에러 때문인지 구분하기 위해 Spike[12]방법과 RE값을 통한 혼잡 구분방법을 함께 사용하여 혼잡여부를 구분하고 있다. TCP Westwood는 이러한 방법들로 기존의 벌크 재전송기법을 보완하고 있다.

하지만 벌크 재전송은 확인되지 않은 ACK에 해당하는 모든 데이터를 재전송하기 때문에 약간의 비효율성을 보이고 네트워크의 혼잡을 가중시킬 수 있다.

3. 제안 및 해결방안

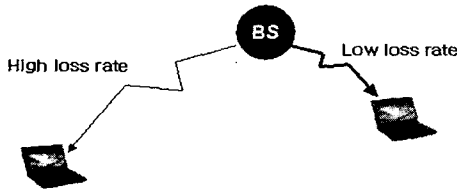
3.1 손실의 구분

무선네트워크에서의 손실은 랜덤 손실과 Bursty한 손실로 구분할 필요가 있다. bursty한 손실의 크기를 알 수 있다면 재전송 할 때 그만큼만 더 재전송하면 되기 때문에 더 효율적으로 재전송 할 수 있을 것이다. bursty한 손실은 페이딩 현상 등에 의해 일어나지만 그 길이는 신호의 세기에 의해

서도 영향을 받는다. 신호의 세기가 작으면 페이딩현상에 의해 손실되는 패킷의 수도 더 많아지기 때문이다. 신호의 세기는 랜덤 손실률을 통해서 알 수 있고 페이딩 현상은 bursty한 손실률을 통해 알 수 있다.

3.1.1 랜덤손실

랜덤 손실은 전파의 세기에 따라 변화하게 되는데 전파의 세기는 거리 및 장애물의 정도에 영향을 받는다. 왼쪽의 모바일 노드는 BS(Base station)와 거리가 멀어 전파의 세기가 약한 경우로서 랜덤 손실은 높게 나타난다. 오른쪽의 모바일 노드는 BS와 거리가 가까워 전파의 세기가 강한 경우로서 랜덤 손실은 낮게 나타난다.



(그림 5) 거리에 따른 손실률

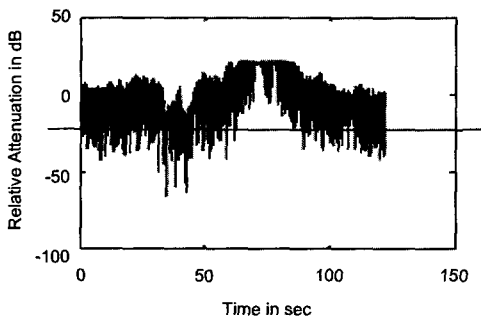
3.1.2 Bursty한 손실

bursty한 손실의 원인을 살펴보면 크게 세가지 볼 수 있다.

- 다중 경로 페이딩 : 중간에 장애물로 인하여 다양한 경로로 오는 전파들의 시간차로 인해 생기는 현상
- shadowing : 모바일 노드가 이동함에 따라 BS와 거리 변화에 의해 생기는 현상
- 핸드오프 : 모바일 노드가 셀(cell) 간에 이동함에 따른 현상

3.2 적응 벌크 재전송 기법

다음 (그림 6)은 Austin 지역에서 자동차로 운전을 하면서 측정된 신호의 감쇠를 나타낸 그래프이다[9].



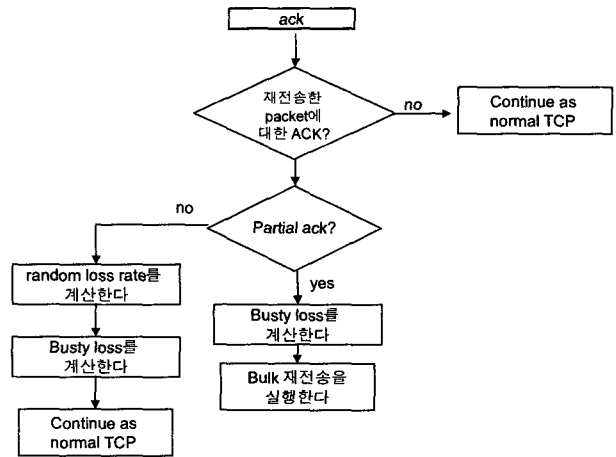
(그림 6) Austin 지역에서 측정된 신호 감쇠

-25dB에 있는 라인이 패킷손실이 일어나는 임계치라면 약 0~40초 에서는 전체적인 신호의 세기는 높지만 페이딩 현상으로 인해 패킷 손실이 짧게 빈번히 일어난다고 볼 수 있다. 40~50초 에서는 신호의 세기는 낮고 페이딩 현상으로 인해 발생하는 패킷 손실은 bursty하게 발생한다. 결과적으로

로 bursty한 패킷손실은 신호의 세기에 영향을 많이 받는다.

TCP는 전송계층이기 때문에 신호의 세기를 직접적으로 알 수는 없다. 하지만 앞에서 보았듯이 랜덤한 손실률을 가지고 신호의 세기를 알 수 있다. 또한 부분적인 ACK를 통하여 bursty한 손실을 알 수 있으므로 이를 이용하여 적응 벌크 재전송 기법을 소개하고자 한다.

(그림 7)은 전송측에서 새로운 ACK를 수신하였을때 전송 방법을 보이고 있는 흐름도(flow chart)이다.



(그림 7) 전송측에서의 flow chart

재전송 한 패킷에 대하여 부분적인 ACK를 받게 되면 bursty한 손실이 생긴 경우이고 부분적인 ACK가 아니면 랜덤 손실이 생긴 경우이다.

RPER(Random Packet Error rate)은 다음과 같이 구할 수 있다.

$$RPER = 100 / (\text{현재 손실된 seq_number} - \text{이전 손실된 seq_number})$$

Bursty한 손실 다음과 같이 구한다. $(0 \leq \beta \leq 5)$

Partial ACK를 수신 하였을때는 $++\beta$

Partial ACK를 수신 하지 않았을때 $--\beta$

(β 는 Bursty한 환경에 대한 가중치 값이다.)

다음 수식은 bursty한 손실이 발생 하였을때 손실된 패킷 개수(R_k)를 예측하여 적응 벌크 재전송을 하기위한 공식이다.

$$R_k = f(\alpha) \pm \beta$$

여기서 $f(\alpha)$ 는 랜덤 손실률인 α 에 따라 다음과 같은 상태를 가진다.

$$\alpha \leq 1\% \rightarrow N(1)$$

$$1\% < \alpha \leq 3\% \rightarrow L(3)$$

$$3\% < \alpha \leq 10\% \rightarrow M(5)$$

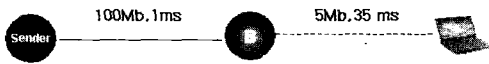
$$10\% < \alpha \rightarrow H(8)$$

N(no), L(low), M(medium), H(high)는 벌크 재전송 상태이고 1, 3, 5, 8은 실험을 통하여 얻은 재전송 패킷의 개수이다. 랜덤 손실률에 따라 벌크 재전송 패킷수를 1차로 결정

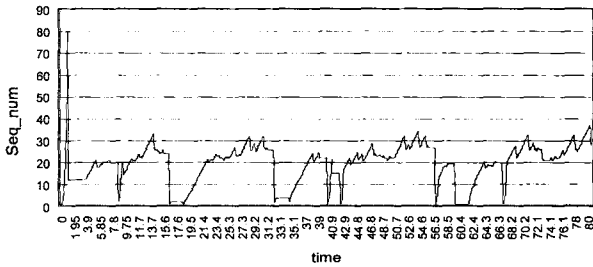
하고 bursty 환경 가중치 $\pm \beta$ 값을 더하여 최종적으로 재전송 패킷수를 구한다.

4. 실험 및 고찰

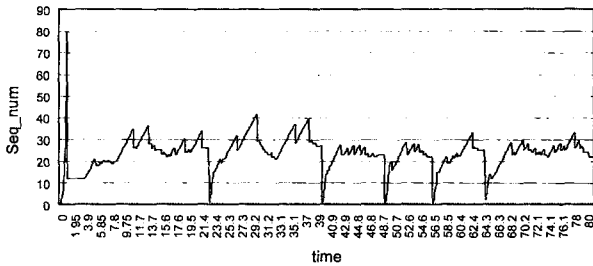
본 논문은 제안하는 적응 벌크 재전송 기법을 적용한 TCP와 적용하지 않은 TCP의 성능을 비교하기 위해 한컴 리눅스 3.0을 OS로 하여 캘리포니아 버클리 대학에서 개발된 NS-2(Network Simulator)을 이용하여 비교 평가 하였다. 그림은 시뮬레이션에서 사용된 네트워크 구조 및 링크 대역폭과 링크 지연시간을 보여주고 있다. 무선 링크에서 에러 모델로는 랜덤 에러 모델과 Markov 에러 모델(error model)을 사용하였다.



(그림 8) 시뮬레이션 환경



(a) TCP Westwood의 CWND



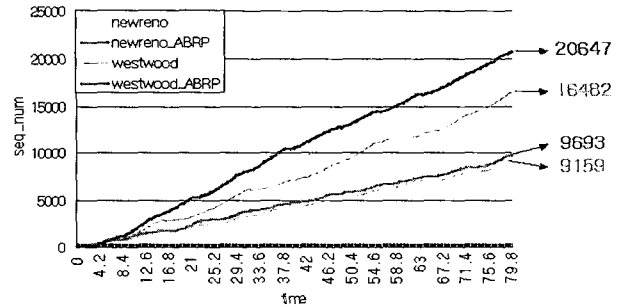
(b) TCP Westwood-ABRP의 CWND

(그림 9)

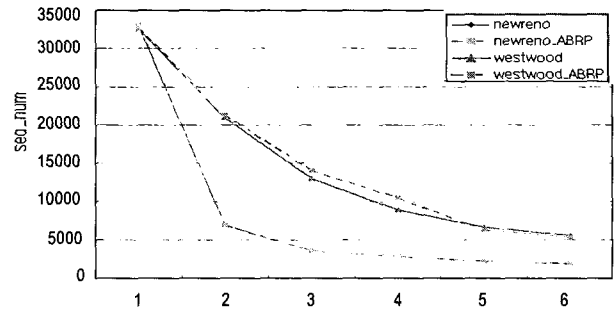
(그림 9)(a)와 (그림 9)(b)는 Markov 에러 모델에서 TCP Westwood에서의 CWND 결과를 보여주고 있다. (그림 9)(a)는 본 논문에서 제안한 재전송 기법을 적용하지 않은 경우이고 (그림 9)(b)는 적응 벌크 재전송 기법을 적용한 경우이다. TCP Westwood는 손실이 발생하더라도 추정된 대역폭으로 CWND를 감소시키기 때문에 무선상에서 불필요한 CWND의 감소는 보이지 않는다. 하지만 (그림 9)(a)에서 보이듯이 Bursty한 손실의 경우 타임 아웃이 되어 CWND를 초기화 시킨다. 하지만 제안된 기법에서는 (그림 9)(b)에서 보이듯이 벌크 재전송으로 인해 타임 아웃 현상을 막을 수 있다.

(그림 10)은 80초동안 수신된 패킷의 수를 보여 주고 있다. 가장 높은 성능을 보이고 있는 것은 TCP Westwood에

적용 벌크 재전송 기법을 적용 시켰을 경우로 20647개의 패킷을 수신하였다. 그리고 성능이 가장 낮게 보이는 것은 TCP Newreno로서 9159개의 패킷을 수신하였다. 그림에서 보이듯이 Newreno 보다는 Westwood가 무선 네트워크에서 적합하다는 것을 알 수 있다.



(그림 10) 80초 동안 수신한 패킷의 수



(그림 11) 랜덤 손실률에 따른 성능 평가

(그림 11)은 bursty한 손실이 아닌 랜덤한 손실이 발생하였을때 결과값을 보여 주고 있다. X축은 손실률이고 Y축은 80초 동안 수신한 패킷의 수이다. 손실률이 1~3.5%에서는 제안된 기법을 적용 하였을때 더 높은 성능을 보이지만 손실률이 4~5%에서는 조금 낮은 성능을 보이고 있다. 손실률이 높아 랜덤하지만 bursty하게 손실된 경우 불필요 하게 많은 재전송을 하기 때문이다. 하지만 5% 이상의 손실이 계속적으로 발생하면 연결을 해제하는 편이 나올 것이다.

<표 1>은 Markov 에러 모델에서 TCP Westwood에 벌크 재전송 기법과 적응벌크 재전송 기법 그리고 적용하지 않을 때 성능을 비교한 표이다.

<표 1> 80초 동안 수신한 패킷의 수

Protocol	수신된 패킷수(80초)
Original	16482
Bulk Retransmission	15268
Adaptive Bulk Retransmission	20647

<표 1>에서 보면 벌크 재전송 기법이 적용되지 않았을 때보다 적게 수신된 것을 알 수가 있다. 벌크 재전송 기법은 무선 네트워크 상황이 매우 안 좋을때만 성능이 높아지기 때문이다.

5. 결론 및 향후 과제

무선 네트워크에서는 페이딩 현상등으로 인한 bursty 패킷 손실이 빈번히 발생한다. 이러한 bursty 패킷 손실은 TCP의 성능을 크게 저하시킨다. 본 논문에서는 bursty한 환경에 대처하기 위한 여러 가지 방법과 적응 벌크 재전송 기법을 살펴보았다. 벌크 재전송 기법 기법은 확인되지 않은 ACK를 모두 재전송 함으로써 bursty한 환경에서는 높은 성능을 보이지만 다른 환경에서는 오히려 성능이 떨어진다. 또한 무절제한 패킷전송으로 네트워크에 혼란을 가중시킬 수가 있다.

본 논문은 랜덤 손실률과 bursty한 손실 가중치를 통하여 bursty 패킷 손실의 수를 예상하여 재전송 패킷 수를 조절하는 적응 벌크 재전송 기법을 제안하였다. 실험 결과 Markov 에러 모델을 사용하여 TCP Newreno에 적용 하였을 때는 5.8%의 성능향상을 보였고 TCP Westwood에 적용 하였을때는 25.2%의 성능 향상을 보여 무선 환경을 위한 TCP에서 더 높은 성능을 보이는 것을 알 수가 있다. Busty한 손실 환경이 아닌 랜덤한 손실 환경에서도 손실률이 3.5%까지는 더 높은 성능을 보이고 있다. 본 논문에서 제시한 적응 벌크 재전송 기법은 전혀 새로운 방법이 아닌 재전송 기법이기 때문에 다른 TCP 프로토콜에 적용 할 수가 있어 무선 환경에서 TCP의 성능을 향상 시킬수가 있다.

향후 과제로는 랜덤 손실률인 α 에 따른 재전송 값과 bursty한 손실 가중치인 β 값을 다양하게 변화시켜가면서 좀더 나은 성능을 보이는 것이다. 그리고 아직 핸드오프 상황에서도 실험을 할 필요가 있으며 Newreno와 TCP Westwood 이외의 다른 TCP에 적용하여 본 논문에서 제안한 적응 벌크 재전송 기법의 성능을 다양하게 평가하고자 한다.

참 고 문 헌

[1] Behronz A. Forouzan, "TCP/IP protocol suite," international editions 2000, pp.271-31.
 [2] R. K. Balan, B. P. Lee, K. R. R. Kumar, L. Jacob, W. K. G. Seah, A. L. Ananda, "TCP HACK : TCP header checksum option to improve performance over lossy links," InfoCom, 2002.
 [3] Tom Goff, James Moronski, D. S. Phatak, "Freeze-TCP : a true end-to-end TCP enhancement mechanism for mobile environment," InfoCom, 2000.
 [4] H. Balakrishnan, S. Seshan, Eamir and R. H. Katz, "Improving TCP/IP performance over Wireless Networks," In Proc.1st ACM Conf. on Mobile computing and Networking, November, 1995.
 [5] Ajay Bakre and B. R. Badrinath, "I-TCP : Indirect TCP for mobile hosts," Tech. Rep., Rutgers University, May, 1995.
 [6] K. brown and S. Singh, "M-TCP : TCP for Mobile Cellular Networks," ACM computer Communications Review(CCR), Vol.27, No.5, 1997.
 [7] Floyd, S. and Henderson, "The NewReno Modification to

TCP's Fast Recovery Algorithm," Internet RFC 2582, Experimental, April, 1999.

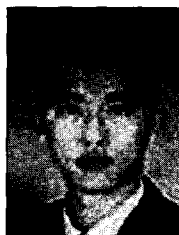
[8] Ren Wang, Massimo Valla, M. Y. Sanadidi, and Mario Gerla, "Adaptive Bandwidth Share Estimation in TCP Westwood," In Proc. IEEE Globecom 2002, Taipei, Taiwan, R.O.C., pp.17-21, November, 2002.
 [9] Young Yong Kim and San-qi Li, "Capturing Important Statistics of a Fading/Shadowing Channel for Network Performance Analysis," 888 IEEE Journal on Selected Areas in Communications, Vol.17, No.5, May, 1999.
 [10] C. Casetti, M. Gerla, S. Mascolo, M. Y. Sanadidi and R. Wang, "TCP Westwood : Bandwidth Estimation for Enhanced Transport over Wireless Links," In Proceedings of ACM Mobicom 2001, Rome, Italy, pp.287-297, July, 2001.
 [11] G. Yang, R. Wang, F. Wang, M. Y. Sanadidi and M. Gerla "TCP Westwood with Bulk Repeat for Heavy Loss environments," UCLA CSD Technical Report #020023, 2002.
 [12] S. Cen, P. C. Cosman and G. M. Voelker, "End-to-End Differentiation of Congestion and Wireless Losses," Proceedings of ACM Multimedia Computing and Networking 2002, SPIE, San Jose, CA, Vol.4673, pp.1-15, Jan., 2002.



홍 충 선

e-mail : cshong@khu.ac.kr
 1983년 경희대학교 전자공학과(학사)
 1985년 경희대학교 전자공학과(공학석사)
 1997년 Keio University 정보통신전공 학과(공학박사)
 1988년~1999년 한국통신 통신망연구소
 선임연구원/네트워크연구실장

1999년~현재 경희대학교 전자정보학부 부교수
 관심분야 : 인터넷 서비스 및 망 관리, 네트워크보안, 모바일 네트워크



강 재 신

e-mail : jaesin-1@hanmail.net
 2002년 경희대학교 전자공학과(공학사)
 2004년 경희대학교 일반대학원 전자공학과 (공학석사)
 2004년~현재 Zestim(연구원)
 관심분야 : Wireless TCP, 모바일 네트워크, IPv6



김 대 선

e-mail : dskim@khu.ac.kr
 2002년 경희대학교 컴퓨터공학과(공학사)
 2004년 경희대학교 일반대학원 컴퓨터공학과(공학석사)
 2004년~현재 경희대학교 일반대학원 컴퓨터공학과(박사과정)
 관심분야 : 모바일 네트워크, IPv6, SCTP