

무선 TCP를 위한 효율적인 재전송 기법

강재신* 홍충선* 이대영*

*경희대학교 전자정보학부

e-mail: jaesin12@hotmail.com

Adaptive Bulk Retransmission Mechanism for Wireless TCP

Jae-Sin Kang*, Choong-Seon* Hong, Dae-Yong Lee*

School of Electronics & Information, Kyung Hee University

요 약

TCP Westwood[1]는 송신측에 수신된 ACK를 통해서 대역폭을 계산하여 데이터를 전송하는 기법이다. 위성망 같이 손실률이 매우 높은 환경을 위해 TCP Westwood에서는 기존의 Bulk Retransmission[2] 기법을 보완 하였다. 하지만 Bulk 재 전송 기법은 불필요한 데이터도 함께 재전송 하기 때문에 성능이 저하되게 된다. 이 논문에서는 재 전송 패킷의 수를 적절히 조절하는 Adaptive Bulk Retransmission 기법을 제안하고자 한다.

1. 서론

무선 네트워크와 장비의 커다란 증가와 함께 mobile 사용자들은 무선상의 통신 뿐만 아니라 많은 데이터의 통신도 요구하게 되었다. 무선 네트워크는 높은 전송 에러를 보이기 때문에 유선상의TCP와 같은 신뢰성 있는 프로토콜을 필요로 한다. 하지만 일반적인 TCP는 유선 네트워크에 맞게 고안되었기 때문에 무선네트워크에는 상당히 낮은 성능을 보이게 된다. 무선 네트워크는 높은 비트 에러 율을 보인다. 유선 네트워크에서 대부분의 손실은 혼잡을 피하기 위해 중간 라우터에서 패킷을 drop시키는 경우에 발생하지만 무선 네트워크에서의 손실은 대부분이 전파 손실에 의해 발생한다. 따라서 기존 TCP처럼 congestion control 방법을 사용한다면 불필요한 congestion window size 감소로 인해 전송 속도가 줄어들게 된다. 또한 무선 네트워크는 장시간 끊어지거나 짧게 자주 끊어는 일이 발생하기 때문에 이

를 고려한 신뢰성 있는 전송을 위해 많은 wireless TCP가 제안되었다.

무선 TCP는 보통 3가지로 분류 할 수 있는데, 첫 번째는 TCP HACK[3]이나 Freeze TCP[4]같이 송신측과 수신측이 직접 통신하는 End-To-End 방법이다. 두 번째는 Snoop[5] 와 같이 중간의 Base Station에서 저장등의 간섭을 하는 Link Layer이고 마지막으로 세 번째 방법은 I-TCP[6]나 M-TCP[7]처럼 유선과 무선을 나누어서 통신하는 Split Connection 방법이 있다.

Wireless상에서의 손실은 일반적으로 발생하는 random error와 fading현상에 의한busty error로 나눌 수 있다. 특히 busty error는 RTO를 크게 증가시키고 윈도우 크기를 초기화시키기 때문에 전송속도에 크게 영향을 미치게 된다. 본 논문은 이러한 busty packet loss의 길이를 예상하여 retransmission packet을 조절하고자 한다.

논문 구성은 다음과 같다. 2장 관련연구에서는 bustyloss를 처리할 수 있는 SACK과 TCP HACK(Header Checksum) 및 TCP Westwood에서

본 논문은 한국과학재단 목적 기초연구(r05-200-000-12193-0)지원으로 수행되었음

의 Bulk Retransmission 방법에 대해서 간단히 살펴본다. 그리고 3장 제안 및 해결방안에서는 이 논문에서 제안하고자 하는 Adaptive Bulk Retransmission에 대해 알아본다. 4장에서는 시뮬레이션을 통한 결과를 제시하고 마지막으로 5장에서는 결론을 맺고 향후과제를 제시할 것이다.

2. 관련연구

TCP SACK(Selective Acknowledgement)는 하나의 윈도우에서 여러 개의 세그먼트가 drop 되었을 때 성능을 향상시키기 위한 방법이다. SACK를 이용하면 receiver는 sender에게 자신의 queue상태를 확인하여 loss되지 않은 segment에 대한 정확한 정보를 ACK에 추가하여 전달할 수 있으므로 sender는 이 ACK를 기반으로 실제로 loss가 발생한 segment만을 retransmit하는 것이 가능하다. 하지만 loss rate가 높을 경우에는 일반적인 TCP와 마찬가지로 timeout과 slow start로 인해 성능이 떨어지게 된다.

TCP HACK은 packet이 corruption되었을 때에 초점을 두고 있다. Wireless상에서는 보통 header가 아닌 데이터에 corruption이 일어나기 때문에 receiver는 corruption된 packet의 sequence number를 알 수가 있다. 이때 Header의 신뢰성을 높이기 위해 header checksum을 사용하게 되고 Receiver는 "special" ACK를 통해 packet이 무선상의 손실인지 유선상의 손실인지 sender측에 알리게 된다. 하지만 ACK가 손실될 경우에는 timeout으로 인해 전송률이 떨어지게 된다.

Bulk 재 전송은 TCP Tahoe와 Reno에서 이미 다룬바가 있다. 하지만 Bulk 재 전송은 네트워크의 혼잡을 가중시킬 수도 있다. TCP Westwood에서는 이러한 단점을 보완하기 위해 두 가지를 보완하였다. 하나는 고정된 Timeout를 쓰는 것이고 다른 하나는 손실이 Congestion때문인지 Error때문인지 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 Congestion때문인지 Error때문인지 구분하기 위해 Spike[8]방법과 RE값을 통한 Congestion 구분방법을 함께 사용하여 Congestion여부를 구분하고 있다. TCP Westwood는 이러 방법들로 기존의 Bulk 재 전송기법을 보완하고 있다.

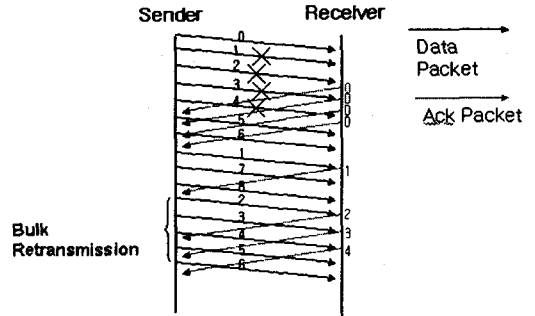


그림 1 Bulk Retransmission의 예

그림1의 왼쪽은 일반적으로 재 전송했을 때의 그림이고 오른쪽은 Bulk 재 전송 했을 때의 그림이다. 1번 데이터를 재 전송 후 송신측은 부분적인 ACK를 받게 된다. 그럼 송신측은 손실이 bursty하게 발생했다고 생각하고 Bulk 재 전송을 하게 된다. 하지만 Bulk 재 전송은 확인되지 않은 ACK에 해당하는 모든 데이터를 재 전송하기 때문에 약간의 비효율성을 보인다.

3. 제안 및 해결방안

무선네트워크에서 손실을Random loss와 Busty loss로 나눌 수 있다. Random loss rate는 BS와의 거리와 전파의 세기에 따라 변화하고 Busty loss rate는 주로 fading현상으로 인해 발생한다. bursty loss의 크기를 알 수 있다면 재전송 할 때 그만큼만 더 재전송하면 되기 때문에 더 효율적으로 재 전송할 수 있을 것이다.

Busty loss는 fading 현상등에 의해 일어나지만 그 길이는 신호의 세기에 의해서도 영향을 받는다. 신호의 세기가 작으면 fading현상에 의해 손실되는 packet의 수도 더 많아지기 때문이다. 신호의 세기는 random loss rate를 통해서 알 수 있고 fading

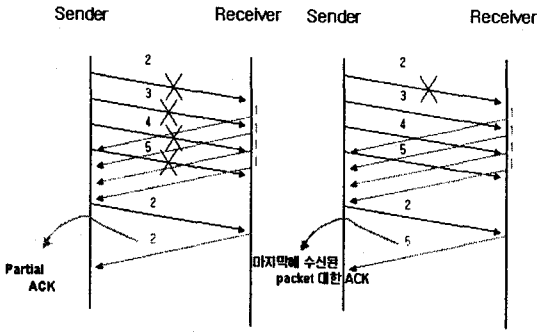


그림 2 partial ACK의 예

현상은 busy loss rate를 통해 알 수 있다.

random loss 와 busy loss는 그림2 에서 보듯이 재전송된 packet에 대한 ACK를 통해서 알 수 있다. Partial ACK인 경우는 busy loss가 생긴 경우고 마지막에 송신한 packet에 대한 ACK이면 random loss인 경우이다.

다음 수식은 busy한 손실이 발생하였을때 손실된 packet 개수를 예측하여 bulk 재전송하기 위한 공식이다.

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

여기서 $f(a)$ 는 random loss 을 a 에 따라 다음과 같은 상태를 가진다.

- $a < 0.01$ -> N(1)
- $0.01 < a < 0.05$ -> L(3)
- $0.05 < a < 0.20$ -> M(5)
- $0.20 < a$ -> H(8)

N(no), L(low), M(medium), H(high)는 bulk 재전송 상태이고 1, 3, 5, 8, 은 재전송 패킷의 개수 이다.

각각의 상태에 따라 bulk 재전송 패킷수를 1차로 결정하고 busy 손실률에 $\pm\beta$ 값이 더하여 최종적으로 재전송 패킷수를 구한다. 그림3은 a, β 값에 다른 상태도이다.

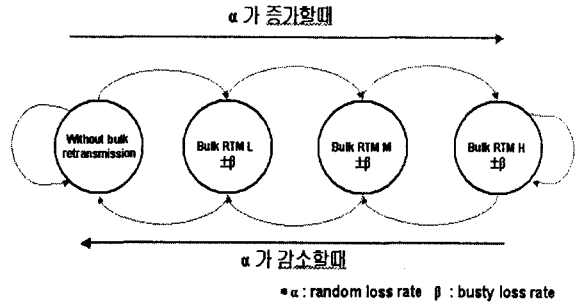


그림 3. adaptive bulk 재전송의 상태도

4. 실험 및 고찰

본 논문에서는 시뮬레이션을 통해 adaptive bulk재전송 기법을 성능을 평가하였다. 시뮬레이션은 한컴 리눅스 3.0버전의 OS를 사용하였고 ns-2(버전 2.26) 상에서 결과를 도출하였다. 또한 busy한 손실 환경을 설정하기 위해 multistate error 모델과 Markov error 모델을 사용하여 시뮬레이션을 하였다.

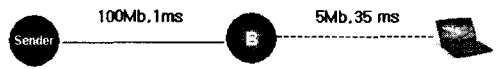


그림 4. 시뮬레이션 환경

Markov error 모델에서는 시뮬레이션 결과 값이 Newreno와 제안한 adaptive bulk 재전송이 120초 동안 연결을 설정하였을때 newreno는 12952개의 packet을 전송하였고 adaptive bulk 재전송기법을 사용하면 13532개의 packet을 전송하여 약 4.5%의 성능 향상을 보이고 있다.

아래의 그림 5는 여러 가지 파라미터를 설정하여 multistate error 모델에서의 cwnd와 그때의 스투드 홀드 값이다. 위의 그림이 제안한 방법에 의한 실험 값이고 아래 그림은 Newreno이다. 약 35초 정도에서는 Newreno가 조금 더 높은 값을 갖게 되는데 제한한 기법이 약간의 불필요한 재전송을 보낼때가 있기 때문이다. 그러나 60초에서 80초 사이의 값을 보면 제안한 방법이 더 높은 것을 알 수가 있는데 Newreno에서는 여러번의 timeout으로 인해 cwnd가 계속 작아지기 때문이다. 110초 동안 연결 하였을때 newreno는 6904개의 패킷을 제안한 기법은 8407개의 패킷 전송을 보이고 있어 약 21%의 성능 향상을

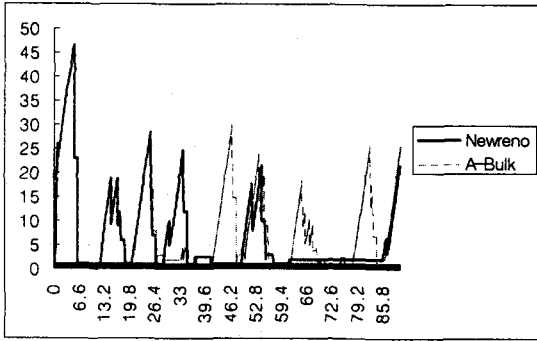


그림 5. multistate error 모델에서의 cwnd

보이고 있다.

5. 결과 및 향후 과제

busty한 packet 손실은 TCP의 성능을 크게 저하시킨다. 본 논문에서는 busty한 환경에 대처하기 위한 여러 가지 방법과 adaptive bulk 재전송 기법을 살펴 보았다. 또한 두 가지 에러 모델을 사용하여 시뮬레이션을 하였다. multistate error 모델과 Markov error 모델 둘 다 성능이 더 향상되었지만 예상했던 것보다는 약간 미흡하다. 아무래도 에러 모델에서 파라미터 설정과 bulk 재전송할 때 Timeout으로 인한 cwnd의 초기화 문제라 예상된다.

향후 과제로는 위의 문제점을 해결하고 α , β 값을 변화시켜가면서 시뮬레이션 결과를 최적화 하는 것이다. 또한 Newreno 이외의 다른 TCP 와 함께 시뮬레이션을 하여 제안한 adaptive bulk 재전송 기법의 성능을 다양하게 평가하고자 한다.

참고문헌

- [1] 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, pp 287-297, Rome, Italy, July 16-21 2001
- [2] 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
- [3] 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", InfoCom2001

[4] Tom Goff, James Moronski, D.S.Phatak, "Freeze-TCP: a true end-to-end TCP enhancement mechanism for mobile environment" InfoCom2001

[5] 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

[6] Ajay Bakre and B.R. Badrinath, "I-TCP: Indirect TCP for mobile hosts", Tech. Rep., Rutgers University, May 1995

[7] K.brown and S.Singh, "M-TCP: TCP for Mobile Cellular Networks", ACM computer Communications Review (CCR), vol.27, no.5, 1997

[8] 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 vol. 4673, pp. 1- 15, San Jose, CA, Jan. 2002