

Ad hoc network 에서 TCP 성능 향상을 위한 혼잡 제어 방법

박매진, 엄두섭
전자전기공학과
고려대학교

TCP congestion control scheme to improve performance in ad hoc network

MAEJIN PARK, Doo-Seop Eom
Department of Electronic and Electrical Engineering
Korea University

Abstract - 무선 통신 기술의 발달로 최근 무선 device 들로 이루어진 ad-hoc network이 주요 issue가 되고 있으며, ad-hoc network 에서의 성능이 중요한 관심사가 되고 있다. 그러나 유선망에 최적화되도록 설계된 TCP 는 무선망이 가지는 링크의 불안정함으로 인한 손실을 네트워크 혼잡에 의한 손실로 인식하여 혼잡 제어 메커니즘이 수행함으로써 전송률을 줄이게 되므로 전체적으로 네트워크의 성능이 저하되는 문제점을 가진다. 본 논문에서는 무선망에서 TCP 성능을 향상시키기 위한 새로운 congestion algorithm을 제안하고 그 성능을 평가하였다. 성능 확인 결과 기존 TCP-vegas 에 비해 더 나은 성능을 보임을 알 수 있다.

1. 서 론

오늘날 무선 통신의 발달로 인해 무선망으로만 구성된 ad-hoc network 에 대한 관심이 증대되고 있다. 무선망으로 구성된 ad-hoc network 은 유선망에 비해 낮은 대역폭과 불안정한 링크를 제공하지만, 이동성과 위치 투명성이 있으며, 이동 device 들의 발달로 인해 활발한 연구가 진행되고 있다. 유선망에서와 마찬가지로 Ad-hoc network에서도 TCP를 일반적으로 사용하는데 무선망의 불안정한 링크로 인해 TCP 의 성능 확보가 주요 관심사가 되고 있다. 이러한 이유로 ad-hoc network에서 tcp 의 성능 문제를 평가하고 개선하기 위한 여러 가지 연구가 진행되고 있다. 본 논문에서는 TCP 성능향상을 위한 새로운 congestion 알고리즘에 대해 연구하고자 한다. [4],[6]

2. 본 론

인터넷에서 가장 널리 사용되는 프로토콜인 TCP는 네트워크로부터의 feedback을 이용하여 flow를 control 함으로써, 네트워크의 bandwidth를 효율적으로 사용하려는 방법이다. 그러나 TCP는 유선환경에 적합하게 설계된 프로토콜이라, 네트워크 환경이 수시로 변화하는 무선 환경에 그대로 적용할 경우 성능 저하를 야기하게 된다. 즉, 기존 TCP에서는 중복 ack packet를 받거나 time out 이 발생한 경우 network 혼잡으로 간주하여, contention window를 변경하여 flow를 control 함으로써 혼잡을 회피하게 된다. 그러나 무선망에서는 network 혼잡뿐만 아니라, 무선 링크의 error 에 의해 packet 의 전달이 실패하는 경우가 빈번히 발생한다. 이와 같은 링크 단절에 의한 packet 전달 실패를 기존의 TCP 에서서는 네트워크 혼잡과 구별하지 못하기 때문에, 네트워크 혼잡과 동일하게 contention window를 변경함으로써 flow를 control 하게 되며, 이로 인해 전체적인 network 의 성능이 감소하는 문제가 발생하게 되는 것이다.

이 문제를 해결하기 위해서 본 논문에서는 TCP-vegas 의 congestion control algorithm을 수정하고자 한다. 먼저 간단히 TCP-vegas를 살펴보면 아래와 같다.

TCP-vegas 에서서는 네트워크의 혼잡 상태를 예측하는 방법으로 기대되는 throughput 과 실제 throughput을 계산하여 이 두 값의 차이를 이용한다. 이 방법은 네트워크에 혼잡이 발생하지 않을 경우 실제 throughput 이 기대되는 throughput 에 가까울 것이라는 것에 근거한 것이다. 기대되는 throughput은 현재의 window size를 측정된 가장 작은 RTT인 baseRTT 로 나눈값으로 계산된다. 그리고 실제 throughput은 현재 window size를 한 패킷을 보내고 그에 대응하는 ack를 받을 때까지 실제로 걸린 RTT를 측정된 값으로 나눈 것이다. 정의에 의해 이 두 값의 차이값은 positive 값이 된다. TCP-vegas 에서서는 두 개의 threshold 값인 α, β ($\alpha < \beta$) 를 사용하는데, 그 차이값이 α 와 β 사이에 들어가는 경우엔 window size를 변화시키지 않고 그대로 유지한다. 반면 차이값이 α 보다 작은 경우엔 네트워크 혼잡이 없으므로 간주하여, window size를 증가시킨다. 그리고, 차이값이 β 보다 큰 경우엔 네트워크에 혼잡이 발생한 것으로 간주하여 window size를 줄이게 된

다 위의 각 상황에 따른 TCP-vegas 에서의 window size 변경은 아래와 같다.

$$\begin{aligned} \text{Diff} &= \text{expected throughput} - \text{actual throughput} \\ \text{CWND}(\text{contention window}) &= \begin{cases} \text{CWND} + 1 & \text{if diff} < \alpha \\ \text{CWND} - 1 & \text{if diff} > \beta \\ \text{CWND} & \text{if } \alpha < \text{diff} < \beta. \end{cases} \end{aligned}$$

위와 같은 방법으로 TCP-vegas에서는 네트워크의 queue 에 특정한 수의 packet을 유지함으로써 다른 방식의 TCP 보다 더 나은 throughput을 보여 주게 된다.

그러나, TCP-vegas를 무선망에 사용할 경우 다른 방식과 마찬가지로 link error 및 단절에 의한 packet 손실을 네트워크 혼잡으로 간주되기 때문에, link error 가 발생할 경우 전체적인 성능이 떨어지게 된다.

이에 본 논문에서는 아래와 같이 link error 와 네트워크 혼잡을 구별하는 방법을 제시하고 이를 TCP-vegas 에 추가한 ad-hoc-tcp를 제안한다.

본 논문에서 link error 와 네트워크 혼잡은 아래 두가지 방법을 이용하여 구별하게 된다.

첫째로 중복 ack를 받았을 경우이다.

송신자가 수신자로부터 ack packet을 받았다는 것은, data packet 이 수신자에게 제대로 전달되었으며, 네트워크가 차단되지 않았다는 것을 알려준다. 우리는 중복된 ack packet을 받았을 때 네트워크의 실제 throughput을 확인하게 된다. 만약 실제 throughput 이 기대되는 network throughput 의 절반이하를 차지하고 있다면 이는 네트워크 혼잡에 의한 것이라기 보다 bit error 에 의한 loss 로 볼 수 있다. 그리고 실제 throughput 이 기대되는 bandwidth 의 절반 이상을 차지하고 있는 경우에는 이를 network 혼잡으로 간주한다

두 번째 방법은 time out 이벤트가 발생한 경우이다.

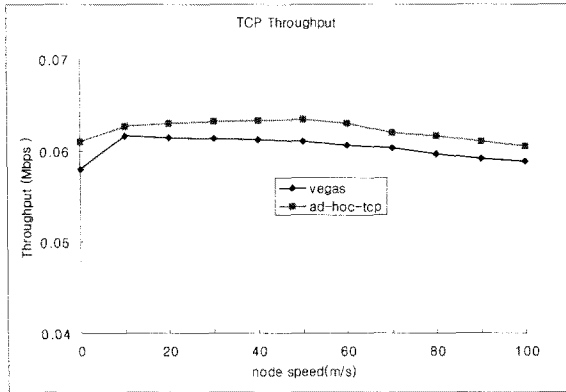
time out 이벤트의 발생은 데이터를 발송한 후 일정 시간 동안 응답이 없는 경우에 발생하는 것으로 어디에선가 data 가 분실된 경우이다. 만약 time out 이 발생할 때 RTT가 점점 증가하고 있었다면 이는 네트워크 혼잡으로 볼 수 있다. 반면에 RTT의 변화없이 time out 이벤트가 발생한 경우는 bit error 에 의한 링크의 단절로 볼 수 있다.

상기에 방법에 의해 우리는 네트워크 혼잡과 bit error를 구별하고 이에 맞추어 congestion control을 하게 된다. 만약 무선망의 bit error 에 의한 중복 ack 로 판단되는 경우엔, 네트워크 혼잡한 상황이 아니므로 네트워크의 bandwidth를 조정할 필요 없이 즉시 data packet을 재전송하게 된다. 따라서 congestion window 는 변경하지 않은 채로 유지하게 된다. 반면 네트워크의 혼잡으로 인한 것으로 판단되는 경우엔 기존 TCP-vegas 의 방식에 따라 처리하게 된다.

제안한 알고리즘의 성능 평가를 위하여 아래와 같은 네트워크 환경에서 ns-2를 이용하여 시뮬레이션 하였다. 전체 area 는 400m x 400 m 이며, 20개의 노드가 그 안에 존재하게 된다. Propagation model 은 TwoRayGround 모델을 사용하였으며, 각 노드의 통신 거리는 200m 로 설정하였다. 링크의 bandwidth 는 2Mbps 이며, 무선 채널의 bit error는 0.01 로 고려하였다. 20개의 노드는 2개씩 짝을 이루어 10개 그룹으로 나뉘며, 각각 TCP sender 와 TCP sink를 구성하게 된다. 따라서 전체적으로 10개의 tcp flow 가 존재하며, TCP packet 의 size 는 1000 byte 로 설정하였다. 시뮬레이션 시간은 100초이며, TCP vegas 와 성능을 비교 평가하였다.

성능 평가 결과는 아래 <그림 1 > 에 보여진다.

<그림 1> 에서 보는 바와 같이 무선망의 link error 가 네트워크 혼잡과 구별되기 때문에 제안한 알고리즘이 조금 더 나은 성능을 보임을 알 수 있다.



<그림 1> 노드 속도에 따른 TCP throughput 변화

3. 결 론

본 논문에서는 ad-hoc network에서 TCP 성능을 향상시키기 위한 새로운 congestion control algorithm을 제안하였다. 성능 평가 결과 본 논문에서 제안한 기법이 기존의 방법보다 더 나은 성능을 보임을 알 수 있었다.

[참 고 문 헌]

- [1] W. Stevens, "TCP Slow Start, Congestion, Avoidance, Fast Retransmit, and Fast Recovery Algorithm", RFC2001, 1997
- [2] B.M. Allman, V. Paxson, W. Stevens "TCP Congestion Control", RFC2581, 1999
- [3] Z. Jianxin, Z. Ling, S. Bingxin, "A Survey of TCP Performance in Wireless Network. Journal of Computer Research and development", China, vol. 41, 2004
- [4] AA. Hanbali, E. Altman, P. Nain, "A survey of TCP over Mobile Ad Hoc Networks" http://www-sop.inria.fr/mae-stro/personnel/AhmadAl_Hanbali/PUBLICATIONS/RR-5182.pdf, 2004
- [5] Lawrence S. Brakmo, Sean W. O'Malley, Larry L. Peterson. "TCP Vegas: New Techniques for Congestion Detection and Avoidance". In Proceedings of ACM SIGCOMM'94, pages 24-35, 1994.