

무선 네트워크에 적합한 Decoupled TCP

강운수^o, 모정훈
 한국정보통신대학교
 {kkamo^o, jhmo}@icu.ac.kr

Decoupled TCP: TCP for Wireless Networks

Moonsoo Kang^o, Jeonghoon Mo
 School of Engineering, Information and Communications University

요 약

무선망에서의 TCP 성능은 많은 연구가 되어 왔지만 손실율이 큰 경우의 TCP의 성능은 아직도 개선의 여지가 많다. 본 논문은 두 가지 점에서 새로운 TCP를 제안한다. 첫째, 혼잡제어(Congestion Control)와 손실제어(Loss Recovery)가 분리(Decoupled)되었다. 기존의 연구들은 무선손실과 혼잡손실을 구분하여야 한다는 점에는 많은 연구를 해왔지만 무선 손실에 대하여 어떻게 대처할 것 인가에 관한 문제는 상대적으로 간과하였다. 둘째, 기존의 TCP-Westwood와 TCP-Jersey에서 사용되는 ABE(Available Bandwidth Estimation)가 무선네트워크에 의해 발생하는 패킷 손실로 인해 부정확해지는 것을 지적하고 새로운 ABE방법을 제시한다. 시뮬레이션을 통하여 우수한 성능을 보여주었다.

1. 서 론

신뢰성있는 데이터 전송에 사용되는 TCP(Transmission Control Protocol)는 유선망과 달리 무선망에서 좋은 성능을 나타내지 못하고 있다.

TCP-Snoop과 같이 링크레벨 재전송을 통해 무선망의 에러를 차단하거나, TCP-Westwood[1]와 TCP-Jersey[2]와 같이 ABE를 이용하여 congestion시 cwnd(혼잡 제어 윈도우)를 결정하는 등, 무선망에서 TCP의 성능을 향상시키기 위한 연구들이 있다. 그 외에 무선에러에 의한 패킷 손실과 congestion에 의한 패킷 손실을 구분하여 원인에 따라 다른 처리를 함으로써 성능을 향상시키려는 시도도 있다.

이러한 시도에 대해, 무선에 의한 패킷손실과 congestion에 의한 패킷손실을 어떻게 구분할 것인가? 에 대한 많은 연구가 있지만 TCP가 어떻게 행동해야 하는가? 에 대한 연구는 상대적으로 적다. Biaz[3]와 Cen[4]의 송신자에서 ACK의 inter-arrival time을 이용하는 방법과 무선 AP나 라우터에서 explicit notification을 통해 손실 이유를 구분하는 방법은 무선에서 TCP의 성능의 향상을 보였지만 여전히 무선 bandwidth를 충분히 사용하지 못한다.

TCP의 cwnd는 ACK를 받지 않고 패킷을 보낼 수 있는 최대량으로 loss recovery와 congestion control에 사용된다. 수신자는 송신자에게 cumulative ACK를 알려준다. 송신자는 이 값을 lastAcked에 기록하고 lastAcked + cwnd 사이의 시퀀스 넘버를 가진 패킷을 전송한다. 패킷이 손실되거나 도착순서가 잘못되었을 경우, 송신자는 수신자로부터 duplicate ACK(중복응답)를 받으므로 송신자는 lastAcked를 변경시키지 않는다. 이 때문에 TCP는 손실된 패킷이 복구 되지 않는 한, 최소한 하나의 RTT동안 새로운 패킷을 보낼 수가 없다. 나은 성능을 얻기 위해, cumulative ACK를 이용한 sliding window방법에서 lastAcked가 진행되는 것이 중요하다.

무선 네트워크에서 낮은 성능은 TCP의 congestion control과 loss recovery의 이런 밀접한 연관, 즉, 패킷 손실과 congestion과 동일시하는 TCP의 전통적인 디자인 철학에 있다. 이 논문은 TCP가 성능을 증대시킬 수 있는 두가지 아이디어를 제시한다.

2. Decoupled TCP

이절에서는 Decoupled TCP의 2가지 특징인 (i)Loss Recovery와 Congestion Control의 분리, (ii)무선에러를 고려하여 가용 Bandwidth를 측정하는 방법인 WABE에 대해 설명한다.

2.1 Loss Recovery와 Congestion Control의 분리

TCP는 lastAcked와 lastAcked+cwnd 사이의 시퀀스 넘버를 가진 패킷만 전송할 수 있다. 패킷 손실이 발생하면 lastAcked가 진행될 수 없어 bandwidth를 충분히 사용할 수 없게 된다. Decoupled TCP는 이런 제약성을 제거하기 위해 패킷 전송에 lastAcked에 의존하지 않는다.

Decoupled TCP는 WAITQ 와 RETRXQ라는 두개의 큐를 가지고 있다. 패킷이 전송될 때 최초의 전송이던 재전송이던 expiration timer set을 가지고 WAITQ에 저장된다. ACK가 도착하면 그 패킷은 WAITQ에서 제거된다. 그렇지 않고 timer가 expiration될 때까지 ACK가 도착하지 않거나, 3DUP이 도착하면, 그 패킷은 RETRXQ로 옮겨지게 된다. 이 큐에 있는 패킷들은 재 전송되기 위해 대기하고 있는 패킷들이다. RETRXQ가 비어있지 않으면, 이 큐에 있는 패킷들을 새 패킷보다 먼저 전송한다. Decoupled TCP는 RETRXQ에 있는 패킷들에 새 패킷보다 높은 우선 순위를 부여 한다. RETRXQ에 있는 패킷들의 수가 cwnd보다 클 경우, TCP는 다음 RTT 시간동안 전송할 패킷들을 cwnd만큼 선택하게 된다. 그렇지 않으면, RETRXQ에 있는 패킷들을 전송한 뒤에, cwnd-RETRXQ의 패킷 수 만큼의 새 패킷들을 전송하게 된다. 성공적으로 패킷이 전송 될 때 까지, 패킷은 두개의 큐 중 하나에 존재하게 되고 전송이 끝난 뒤 제거된다.

Decoupled TCP의 cwnd관리방법은 TCP-Westwood나 TCP-Jersey와 같은 additive-increase와 adaptive-decrease이다. 하나의 RTT시간 마다 윈도우를 하나씩 증가시키고 congestion이 발생했을 때 가용한 bandwidth로 윈도우를 감소시킨다.

$$cwnd = \begin{cases} cwnd + \frac{1}{cwnd} & \text{upon ACK} \\ ABE \times RTT & \text{upon Congestion} \end{cases} \quad (1)$$

무선 패킷 손실이나 congestion이 발생하여 cwnd의 값이 변화해도, 패킷 전송에 영향을 미치지 않는다. 각 RTT 시간 동안, TCP는 대략 cwnd정도의 패킷들을 전송한다.

Decoupled TCP에서는 fast recovery와 fast retransmit이 필요 없다. Bandwidth의 가용성을 높이기 위한 cwnd의 일시적인 증가는 lastAcked에 대한 의존성을 제거함으로써 필요 없다. 3 DUP를 받으면 timeout이전에 패킷을 전송하는 faster transmission도 필요하지 않다. 또 패킷 손실 시, 패킷 전송을 lastAcked+cwnd로 제한하지 않는다는 점에서 전통적인 congestion control과는 달라진다. 기존의 TCP는, lastAcked가 중복 ACK을 받을 때는 멈추기 때문에, Error recovery에 걸리는 RTT 동안 패킷을 cwnd만큼 전송하지 못하지만 Decoupled TCP는 3 DUP이나 timeout이후에 패킷을 전송하는 것이 아니라 TCP는 매 RTT 시간 마다 cwnd정도의 패킷을 계속적으로 전송하려고 한다.

2.2. WABE(Wireless Available Bandwidth Estimation)

패킷 손실이 무선에러에 의해 발생하는 무선 네트워크에서는 ABE가 정확히 측정되지 않는다. 그 이유는 ABE가 ACK의 수신에 기반하고 있는데 무선 에러로 인해 발생하는 잦은 congestion control에 의해 데이터 송신이 원활하지 않으므로 Bandwidth 측정에 충분한 ACK도 송신자에게 수신이 되지 않는다.

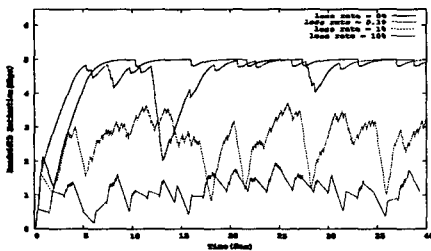


그림 1 TCP-Westwood의 패킷 손실률에 따른 ABE값

그림 1은 congestion없고 bottleneck link의 capacity가 5Mbps인 네트워크에서 무선에러율이 0, 0.1, 1, 10% 일 때 TCP-Westwood의 ABE값을 보여준다. ABE의 정확한 값은 5Mbps이나 1%과 10%의 경우는 오차가 심하다. 무선에러에 의한 패킷손실로 ABE값이 정확히 측정되지 못하는 것이다. 이것은 ACK의 도착 패턴을 살펴보면 그 이유를 알 수 있다. ACK들은 시간에 따라 일정 간격으로 균일하게 도착하는 것이 아니라 어느 부분에 집중적으로 도착하고 이것을 RTT 시간 간격을 단위로 반복한다. 같은 RTT 내에서 Sample Bandwidth Estimation에 의해 점차 증가하는 ABE이 값이 다음 RTT에 다시 감소한다. 이것은 low pass filter의 특징 때문이다. $ABE = (1-\alpha) \cdot ABE + \alpha \cdot BE$ 로 표현되는 low pass filter는 α 의 값이 0.1 미만으로 ABE를 계산할 때, 새로운 BE의 값을 적게 반영하고 기존의 값을 0.9이상으로 반영한다. BE의 값이

아주 적은 값이라고 하면 $\alpha \cdot BE$ 의 값이 거의 0에 가깝게 되고 새로운 ABE의 값은 직전 ABE의 값의 90%에 해당되므로 그 전 RTT동안 ABE값에 누적된 Sample Bandwidth Estimation은 새로운 RTT가 시작되면 버려지게 된다. 이런 일이 매 RTT마다 반복되므로 ABE는 안정된 값으로 수렴하지 못한다. RTT와 RTT사이에서 발생하는 ACK의 inter-arrival time은 packet pair알고리즘의 link의 패킷처리 시간에 의해 발생하는 시간 간격이 아니므로 ABE를 계산할 때 이런 형태의 ACK의 inter-arrival time을 제거해야 한다. 패킷 손실률이 0%인 경우 cwnd의 사이즈는 점차 증가하여 ACK의 수신 패턴이 RTT시간 간격 전체로 퍼지게 되어 RTT와 RTT사이의 inter-arrival time은 같은 RTT내의 ACK inter-arrival time과 거의 같아 때문에 ABE의 계산이 정확하다. 이것은 시뮬레이션을 통해 확인 되었다. 무선 에러에 의한 패킷 손실로 인해 cwnd의 값이 증가하지 못하고 줄어들어서 이러한 ACK의 수신 패턴은 ABE의 측정에 악영향을 미친다.

$$IA = (1-\alpha) \cdot IA + \alpha \cdot \frac{1}{ACKinterarrival}$$

$$\text{Average of } ACKinterarrival = \frac{1}{IA}$$

ACK의 RTT간 inter-arrival time을 제거 하는 방법은 ACK의 inter-arrival time의 역수에 대해 low pass filter를 적용한 역수의 평균값을 이용해 ACK의 inter-arrival time의 평균값으로 기존의 ABE 수식에 사용한다. 이 방법은 일시적, 비정상적으로 긴 ACK의 inter-arrival time을 제거한다. 이것에 대한 효과 및 정확도는 3. 성능평가 절에서 보인다.

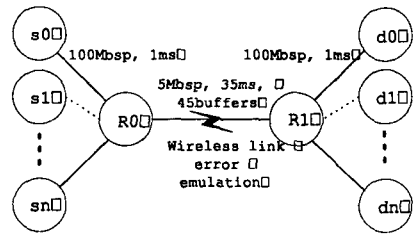


그림 3 NS-2 시뮬레이션에 사용된 네트워크 모델

3. 성능평가

Decoupled TCP를 TCP-Reno, TCP-Sack, TCP-Westwood와 비교했다. 시뮬레이션은 NS-2를 사용하고 Decoupled TCP의 송신자 Agent를 구현했다. 그림 3은 시뮬레이션에 사용된 네트워크 모델이다. R0와 R1은 5Mbps의 링크 용량을 가지고 있고 35ms의 delay를 가진다. R0에서 확률 p로 패킷을 버림으로써 wireless link error를 에뮬레이션하고 있다.

3.1 무선링크에서 Bandwidth 측정

WABE의 성능 측정을 위해 bottleneck link에 1~2Mbps의 UDP 트래픽에 따른 Decoupled TCP, TCP-Westwood, TCP-Jersey의 ABE를 조사하였다. 그림 4는 패킷 손실율을 0.1%인 경우, 각 TCP의 ABE를 보여 주고 있다. UDP 트래픽의 양에 따라 가용 bandwidth의 양을 Decoupled TCP가 비교적 정확하게 측정하고 있다. 그러나 TCP-Westwood의 ABE는 불안정한 값을 보인다. 이것은 무선 링크에 의한 패킷의 손실로 인해 비정상적인 ACK의

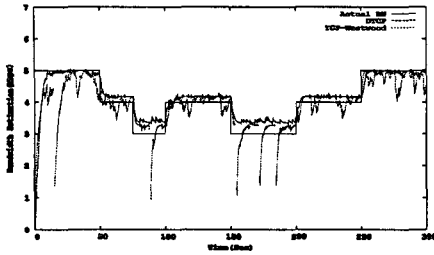


그림 4 패킷손실률 0.1%의 ABE

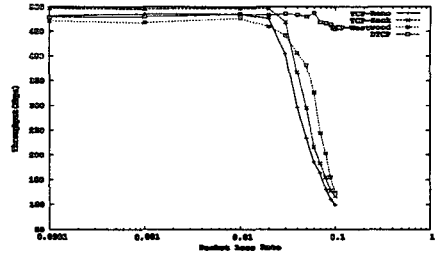


그림 6. 패킷 손실률에 따른 throughput

inter-arrival time이 발생하기 때문이다. WABE는 그에 비해 비교적 안정된 값을 보여 준다. 이러한 사실은 무선의 패킷 손실률이 1%일 경우 더욱 명확히 나타난다.

그림 5이 이러한 경우 각 TCP의 ABE를 보여 준다. TCP-Westwood의 ABE는 실제 Bandwidth 값과는 거리가 멀다. 증가된 패킷손실로 인해 비정상적인ACK의 inter-arrival time이 늘어나게 되고 ABE의 값을 부정확하게 만든다. 그에 비해 Decoupled TCP는 실제 Bandwidth값과는 오차가 있지만 실제 Bandwidth에 근접한 값을 보여 주고 있다. 그림 4,5를 통해 ACK inter-arrival time의 역수에 대한 평균을 이용한 ABE 측정 방법이 무선 네트워크의 패킷 손실에 견고하다.

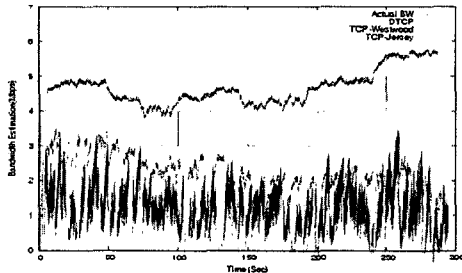


그림 5 패킷손실률 1%의 ABE

3.2 Throughput 비교

그림 6은 패킷 손실률의 변화에 따라 TCP-Reno, TCP-Sack, TCP-Westwood와 Decoupled TCP가 bottleneck link에 10개씩 존재할 때 각TCP의 throughput의 평균값이다. 무선 패킷 손실률이 증가함에 따라 TCP-Reno와 TCP-Sack은 throughput 이 감소하여 패킷 손실률이 10%에서는 매우 낮은 throughput을 보인다. TCP-Westwood의 경우, 패킷 손실률이 1%인 경우 TCP-Reno, TCP-Sack보다 나은 성능을 보이고 있다. Decoupled TCP는 loss recovery와 congestion control이 분리되어 있어 패킷 손실이 패킷 전송에 영향을 미치지 못하기 때문에 TCP-Reno, TCP-Sack, TCP-Westwood에 비해 패킷 손실이 상당히 높은 상황(10%)에서도 훌륭한 성능향상을 나타낸다. 이것으로 보아 loss recovery와 congestion control을 분리하는 것이 무선 네트워크에서 TCP의 성능 향상의 핵심이란 사실을 알 수 있다. 그림 7은 동일한 두개의 TCP가 bottleneck link를 공유하고 서로 다른 RTT(200ms, 50ms)를 가졌을 때, 긴 RTT를 가진 TCP가 짧은 RTT를 가진 TCP에 대해 얼마나 빨리 패킷을 전송하는 지 보여준다. 일반적으로 긴 RTT를 가진 TCP는 짧은 RTT를 가진

TCP에 비해 적은 Bandwidth를 점유하게 되고 패킷전송에서 손해를 보게 된다.TCP-Reno가, 50ms, 200ms의 차이가 가장 심하고 TCP-Westwood와, Decoupled TCP는 TCP-Reno에 비해서는 적은 차를 나타낸다. 이것으로 보아 Decoupled TCP가 TCP-Reno에 비해 Fairness란 측면에서 좋다.

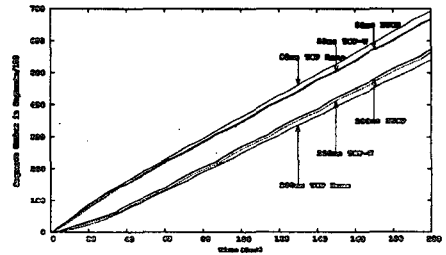


그림7 서로 다른 RTT를 가진 TCP의 전송속도

4. 결론

기존 TCP의 전통적인 구조인 loss recovery와 congestion control 간의 상호 의존적인 구조가 무선네트워크에서 성능향상의 걸림돌임을 지적하고 무선 네트워크의 패킷 전송의 높은 에러율에도 상당한 성능을 보이는 Decoupled TCP를 제안하였다. Decoupled TCP는 loss recovery와 congestion control을 분리하여 무선 네트워크 상의 잦은 패킷 손실로 인한 loss recovery가 congestion control의 패킷전송에 영향을 미치지 못하도록 하였다. 무선 패킷 손실로 인해 TCP-Westwood나 TCP-Jersey와 같은 기존의 ABE의 측정이 부정확해지는 것을 지적하고 그것에 대한 해결책으로 WABE을 제안하였다. 시뮬레이션을 통해 Decoupled TCP의 성능 및 WABE의 효과를 증명하였다.

5. 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] Kai Xu; Ye Tian; Ansari, N., TCP-Jersey for wireless IP communications, IEEE Journal on Selected Areas in Communications, Volume: 22, Issue: 4, May 2004.
- [3] Saad Biaz and Nitin H. Vaidya, Discriminating Congestion Losses from Wireless Losses using Inter-Arrival Times at the Receiver, Proceedings of the 1999 IEEE Symposium on Application - Specific Systems and Software Engineering and Technology, 1999.
- [4] Song Cen, Pamela C. Cosman, Geoffrey M. Voelker: End-to-end differentiation of congestion and wireless losses. IEEE/ACM Transactions on Networking, 11(5): 703-717 (2003).