전송률 편중을 완화시키기 위한 TCP 비례 증가 혼잡 제어

이종민° 차호정
연세대학교 컴퓨터과학과
{jmlee,hjcha}@cs.yonsei.ac.kr

A Proportionally Increasing TCP Congestion Control for Enhancing Throughput Bias

Jongmin Lee° Hojung Cha
Dept. of Computer Science, Yonsei University

요 약

본 논문은 TCP에서 사용되는 혼잡 제어 방식의 문제점인 전송 시간에 대한 전송률 편중 현상을 완화시키기 위한 비례 증가 혼잡 제어 방식을 제안한다. TCP는 네트워크의 혼잡 상황을 제어하기 위하여 ACK 기반의 혼잡 윈도우 조정 방식을 사용한다. 그러나 기존 TCP에서 사용되는 혼잡 제어 방식은 그 자체적으로 네트워크의 혼잡 상황을 유도하며, 전송 시간이 다른 두 개의 TCP 연결이 하나의 병목 지점을 공유할 경우, 전송 시간의 차이에 따라 전송률의 편중 현상이 나타난다. 본 논문에서는 이러한 네트워크 혼잡 상황 유발과 전송 시간의 차이에 따른 전송률 편중 현상을 완화시키기 위한 송신자 기반의 네트워크 대역폭 측정 방법 및 비례 증가 혼잡 제어 알고리즘을 제안하며, 구현 및 실험을 통해 기존 TCP의 문제점을 해결할 수 있음을 보인다.

1. 서론

대표적인 인터넷 전송 프로토콜인 TCP는 신뢰성 있고 연결 지향적인 서비스를 제공하기 위한 전송 프로토콜로 네트워크의 혼잡 상황을 제어하기 위해 윈도우 기반의 AIMD (Additive Increase Multiplicative Decrease) 혼잡 제어 방식을 사용한다[1]. TCP에서 사용되는 AIMD 방식의 혼잡 제어는 네트워크의 대역폭을 가늠하기 위한 방법으로 송신자에서 자료의 전송률을 서서히 증가시키다가 패킷 손실이 발생하였을 경우 전송률을 급격히 줄이는 방식이다. 그러나 AIMD 방식의 혼잡 제어의 경우 네트워크 중간 라우터의 버퍼링 부하를 가중시키고 네트워크의 혼잡 상황을 유발하는 문제점이 있다. 또한, 네트워크 혼잡 상황을 회피하기 위한 혼잡 회피 과정에서의 전송률 증가율이 너무 작아서 네트워크의 대역폭을 충분히 사용할 수 없다는 단점이 있다[2]. TCP 전송률 조정 방법의 또 다른 문제점으로는 전송률 편중 현상이다. TCP 송신자의 전송률 제어는 수신자로부터의 ACK 패킷을 수신하였을 때 수행되며, 이러한 ACK 기반의 전송률 조정 방법으로 인해 전송 시간이 서로 다른 두 개 이상의 연결이 하나의 병목 지점을 공유할 경우 전송 시간의 차이에 따라 전송률의 차이가 발생하는 전송률 편중 현상이 나타난다[3].

본 논문에서는 기존 TCP에서 사용되는 AIMD 혼잡 제어 방식의 문제점인 네트워크 혼잡 상황 유발과 네트워크의 대역폭을 충분히 활용하지 못하는 것, 그리고 전송 시간의 차이에 따른 전송률 편중 현상을 해결하기 위한 송신자 기반의 비례 증가 혼잡 제어 방식을 제안한다. 또한, 제안된 방식을 실제 시스템에서 구현하고 성능 분석을 함으로써 제안하는 방식의 효율성을 검증한다. 논문의 구성은 다음과 같다. 2장에서는 논문에서 제안하는 비례 증가 혼잡 제어 방식에 대해 기술한다. 3장에서는 실험 방법 및 결과에 대해 기술하고 4장에서 결론을 맺는다.

2. 비례 증가 혼잡 제어

기존 TCP에서 송신자는 네트워크의 대역폭을 가늠하기 위해 전송률을 지속적으로 증가시키면서 네트워크의 혼잡 상황에 의한 패킷 손실을 유발한다. 본 논문에서는 TCP 송신자에서 네트워크의 대역폭을 측정하고 측정된 대역폭에 따라 전송률을 비례적으로 증감시킴으로써 네트워크의 혼잡 상황 발생을 감소시키고 전송 시간 차이에 따른 전송률 편중 현상을 완화시키는 방법을 제안한다. 송신자에서의 네트워크 대역폭 측정은 패킷 전송 후 송신자에서 수신된 ACK 패킷의 시간 간격과 수신자가 수신한 자료의 양으로 식 1에 의해 구할 수 있다.

$$MB_i = \frac{S_{rcv}}{IT_{ack}} \quad (1)$$

식 1에서 MB_i는 i 번째 측정된 네트워크 대역폭을 의미하고, IT_{ack}는 수신된 ACK간의 간격, S_{rcv}는 수신자가 수신한 자료의 양을 의미한다. 일반적으로 TCP에서는 Delayed ACK가 사용되며 Delayed ACK의 사용으로 수신자는 패킷 수신시 수신된 패킷에 대한 ACK를 바로 전송하지 않고 일정 시간동안 대기함으로써 전송되는 트래픽의 양을 줄일 수 있으나, 수신자에서의 ACK 패킷 전송 지연으로 송신자에서의 측정되는 네트워크 대역폭은 실제 네트

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

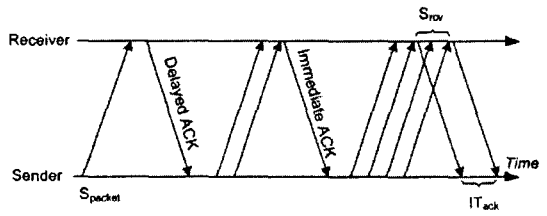

그림 1. ACK를 이용한 대역폭 측정

워크 대역폭과 크게 다를 수 있다. 네트워크 대역폭 측정시 Delayed ACK에 대한 영향을 최소화하기 위해 패킷을 여러 개로 묶어서 전송하는 Packet Bunch 방식을 사용한다[4]. Packet Bunch의 크기는 최소 4개로 하며, 이는 수신자가 연속된 두 개의 최대 크기의 세그먼트를 수신하였을 경우 ACK를 지연시키지 않고 즉시 전송하기 때문이다[5]. 따라서 송신자가 4개의 최대 크기의 세그먼트를 전송할 경우 수신자가 Delayed ACK를 사용하더라도 송신자는 두 개의 연속적인 ACK를 받을 수 있으며, 송신자에서 보다 정확한 네트워크 대역폭 측정이 가능하다. 그림 1은 Packet Bunch를 이용한 대역폭 측정 방법을 보여준다.

가변하는 네트워크 상황에 적절히 적응하기 위하여 측정된 대역폭은 식 2에서와 같은 방법으로 보정한다.

$$SB_i = \alpha SB_{i-1} + (1-\alpha) MB_i \quad (2)$$

SB_i는 보정된 네트워크 대역폭으로 과거의 보정된 네트워크 대역폭과 현재 측정된 네트워크 대역폭을 반영함으로써 네트워크 대역폭 변화의 경향을 반영하는 동시에 측정된 네트워크 대역폭의 오류를 보정할 수 있도록 한다. 대역폭 지역 산출 (BDP: Bandwidth Delay Product)은 TCP에서 최적의 전송 성능을 얻기 위한 TCP 송신자의 전송 윈도우의 크기를 의미하며 식 3과 같다.

$$BDP = Bandwidth \times RTT \quad (3)$$

TCP 송신자는 식 2에서의 보정된 네트워크 대역폭과 ACK 수신시 계산되는 전송 시간(RTT)을 이용하여 식 3에 의해 최적의 전송 윈도우의 크기를 결정할 수 있다. 계산된 최적 전송 윈도우 크기를 이용하여 TCP 송신자의 전송률을 제어함으로써 과도한 전송률 증가로 인한 네트워크의 혼잡 상황을 피할 수 있다.

기존 TCP에서의 전송률 증가율은 느린 시작 구간에서 한 패킷의 전송시간(RTT)에 혼잡 윈도우(S_C)가 두 배 증가한다고 했을 경우 S_C/RTT^2 Bytes/sec 이며, 혼잡 회피 구간에서는 하나의 RTT에 하나의 패킷 크기(S_{packet})만큼 증가한다고 했을 경우 S_{packet}/RTT^2 Bytes/sec 이 된다[3]. 현재의 패킷 크기나 혼잡 윈도우 크기에 비례하는 전송률 증가율은 매우 큰 BDP를 가지는 네트워크 환경에서는 최대 네트워크 대역폭을 사용하기 까지 매우 긴 시간이 걸리게 한다. 또한, 전송률 증가율이 전송 시간의 제곱에 반비례하기 때문에 서로 다른 전송 시간을 가지는 트래픽이 하나의 병목 지점을 공유할 경우 전송률의 편중 현상이 나타나게 된다.

비례 증가 혼잡 제어 방식에서는 매우 큰 BDP를 가지는 네트워크 환경에서 전송률 증가율이 느린 문제점과 전송 시간의 차이에 따른 전송률 편중 현상을 해결하기 위해 식 2에 의해 보정된 네트워크 대역폭과 현재의 혼잡 윈도우의 크기, 그리고 식 3에 의해 계산된 BDP를 고려하여, 매 RTT마다 증가할 혼잡 윈도우의 양을 식 4에 의해 계산한다.

$$\beta = \gamma RTT^2 (SB \times RTT - S_C) \quad (4)$$

비례 증가 혼잡 제어 방식의 전송률 증가율은 매 RTT마다 혼잡 윈도우가 β만큼 증가하므로 γ*(SBxRTT-S_C) Bytes/sec 가 된다. 즉, 비례 증가 혼잡 제어 방식의 전송률 증가는 전송 시간에 관계없으며, 현재의 혼잡 윈도우의 크기와 BDP와의 차이가 큰 트래픽에 대해 전송률 증가의 우선순위를 주게 된다. 따라서, 전송 시간의 차이에 의한 전송률 편중 현상을 해결할 수 있으며 BDP가 큰 네트워크 환경에서도 네트워크 대역폭을 효율적으로 사용할 수 있다. 비례 증가 혼잡 제어 방식에서의 최종 전송률을 결정하기 위한 γ 값은 기존 TCP와 네트워크를 공유할 경우를 고려하여 기존 TCP와 친근성을 갖도록 조정해야 한다.

$$\frac{S_C}{RTT^2} = \gamma (SB \times RTT - S_C), \; where \quad S_C = \frac{SB \times RTT}{2} \quad (5)$$

식 5에 의해 혼잡 윈도우의 크기가 BDP의 반일 경우 기존 TCP의 전송률 증가율과 비례 증가 혼잡 제어 방식의 증가율을 같도록 γ값을 조정한다. γ는 $1/RTT^2$이 되며 인터넷의 평균 RTT 값인 200ms를 사용함으로써 다양한 네트워크 환경에서 기존 TCP와의 친근성을 보장할 수 있도록 한다[6].

식 4에 의해 하나의 RTT에서 증가하여야 할 혼잡 윈도우의 크기가 결정되면, 해당 RTT에서 송신자가 수신해야 할 ACK의 갯수를 예상하여 매 ACK마다 전송해야할 패킷의 갯수를 계산할 수 있다. Delayed ACK가 사용될 경우의 예상된 ACK의 갯수는 $S_C/2S_{packet}$ 이며 Delayed ACK가 사용되지 않을 경우는 S_C/S_{packet} 이다. 따라서 Delayed ACK가 사용되고 Packet Bunch가 4개의 패킷으로 구성되어 있을 경우, 매 ACK를 받을 때마다 전송해야할 Packet Bunch의 갯수(N_{pb})는 식 6으로 계산할 수 있다.

$$N_{pb} = \left\lfloor \frac{S_C + \gamma RTT^2 (SB \times RTT - S_C)}{4 S_{packet}} \right\rfloor \quad (6)$$

3. 실험

본 논문에서 제안한 비례 증가 혼잡 제어 방법은 리눅스 2.6.7 커널 코드를 수정하여 구현되었다. 실험을 위해 두 대의 시스템을 사용하였으며, 다양한 네트워크 환경에서의 실험을 위해 네트워크 에뮬레이터인 NIST Net을 사용하였다[7]. 실험 자료의 분석은 tcpdump[8], tcptrace[9] 프로그램을 사용하였다. 실험에서 사용한 TCP 패킷의 크기는 이더넷에서 패킷의 단편화가 이루어지지 않는 최대 크기인 1448 Bytes를 사용하였으며, 100 MBytes의 자료를 전송하는 실험을 수행하였다. 이더넷의 대역폭은 100 MBps이며, Nist Net을 사용하여 RTT를 50ms, 250ms, 500ms의 세 가지로 구분하여 총 10회의 실험을 수행하였다. 그림 2와 3은 각각 현재 가장 많이 사용되는 TCP-Reno와 논문에서 제안하는 TCP-Pi의 RTT에 따른 전송률의 변화를 보여준다.

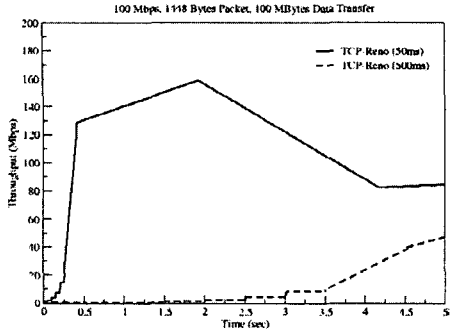

그림 2. RTT에 따른 TCP-Reno 전송률 변화

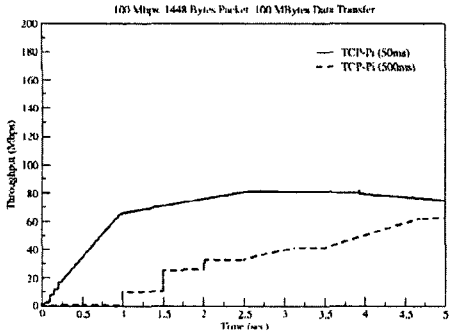

그림 3. RTT에 따른 TCP-Pi 전송률 변화

TCP-Reno의 경우 RTT가 작을 경우 전송률 증가가 급격하게 이루어지며, RTT가 클수록 전송률 증가가 작음을 알 수 있다. 이로써, TCP-Reno의 경우 RTT에 따른 전송률 편중 현상이 심각함을 알 수 있다. 또한, 네트워크 대역폭 이상으로 전송률이 급증하여 네트워크 혼잡 상황을 유발함을 알 수 있다. TCP-Pi의 경우는 RTT의 차이가 큼에도 불구하고 전송률 증가의 정도가 비슷함을 볼 수 있다. 이는 TCP-Pi의 경우 전송 시간의 차이에 따른 전송률 편중 현상이 완화되었음을 의미한다. 연결 초기 RTT가 큰 경우 전송률 증가가 작은 이유는 대역폭 측정에 따른 지연 때문이다.

그림 4는 TCP-Reno와 TCP-Pi의 전송 시간에 따른 전송 성능 비교이다. TCP-Reno의 경우 전송 시간이 커질수록 전송 성능이 급격히 감소함을 알 수 있다. TCP-Reno의 경우 패킷 손실이 발생하였을 경우 시작되는 혼잡 회피 과정의 전송률 증가율이 너무 작아서 네트워크의 대역폭을 충분히 활용하지 못하기 때문이다. 반면, TCP-Pi의 경우 네트워크의 대역폭을 측정하고 BDP를 고려한 전송률 조정으로 전송 성능이 높음을 알 수 있다. 전송 시간이 50ms일 경우 TCP-Pi의 전송 성능이 TCP-Reno보다 약간 작게 측정되었다. 이는 전송 시간에 대한 전송률 편중 현상을 제거하기 위한 전송률 조정 정책이 전송 시간이 짧은 네트워크에서는 오히려 전송률을 감소시키게 되기 때문이다. 그러나 측정된 네트워크 대역폭으로 전송률을 일정하게 유지함으로써 네트워크 대역폭의 활용도가 높아지게 되므로, 전송 되는 자료의 양이 많을수록 전송 성능의 차이는 줄어들게 된다. TCP-Pi에서 전송 시간이 길수록 전송 성능이 약간씩

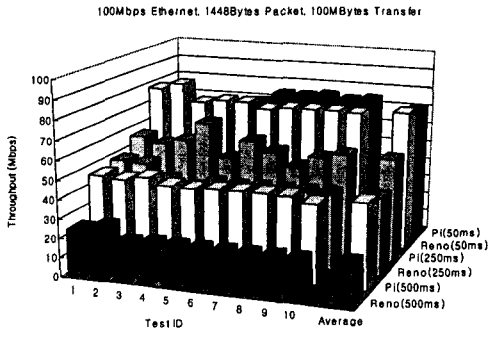

그림 4. 전송 성능 비교

감소하는 이유는 초기 네트워크 대역폭 측정 과정에 드는 시간이 전송 시간이 길수록 길어지기 때문이다.

4. 결론

본 논문에서는 TCP의 혼잡 제어 방식을 개선한 비례 증가 혼잡 제어 방식을 제안하였다. 제안된 방식은 네트워크의 대역폭을 동적으로 측정하고 측정된 대역폭에 따라 송신자의 전송률을 조정한다. 전송률 조정시 현재 세션에 대한 전송 시간을 고려하여 기존 TCP에서의 문제점으로 지적되었던 전송 시간에 대한 편중 현상을 해결하였다. 또한, 전송률 조정시 BDP를 고려함으로써 큰 BDP를 가지는 네트워크에서 기존 TCP가 네트워크의 대역폭을 효율적으로 사용하지 못하는 단점을 해결하였다.

제안된 비례 증가 혼잡 제어 방식은 송신자의 변경만을 필요로 하며 실제 시스템으로 구현되었다. 기존 TCP와의 성능 비교 분석을 통해 전송 시간에 대한 전송률 편중 현상이 기존 TCP보다 작으며, 전송 시간이 클수록 전송 성능이 보다 향상됨을 보였다. 향후 연구 과제로는 측정되는 네트워크 대역폭의 정확도를 높이는 방법에 대한 연구와 다른 변형 TCP 들과의 성능 비교 분석이 있다.

참고문헌

[1] V. Jacobson, "Congestion Avoidance and Control," Proceedings of ACM SIGCOMM, 1998, pp. 314-329.

[2] K. Lai and M. Baker, "Measuring Bandwidth," Proceedings of IEEE INFOCOM '99, New York, NY, USA, March 1999, pp. 235-245.

[3] Floyd S. and Van. Jocobson, "On Traffic Phase Effects in Packet-Switched Gateways," Journal of Internet Working: Research and Experience, Vol. 3, No. 3, September 1992, pp. 115-156.

[4] V. Paxson, "End-to-End Internet Packet Dynamics," IEEE/ACM Transactions on Networking, Vol. 7, No. 3, June 1999, pp. 277-292.

[5] R. Braden, "Requirements for Internet Hosts - Communication Layers," RFC1122, IETF, October 1989.

[6] Network Service & Colsulting Coperation, Internet Traffic Report, http://www.internettrafficreport.com

[7] NIST Internetworking Technology Group. NIST net network emulation package, *http://www.antd.nist.gov/itg/nistnet*, June 2000.

[8] V. Jacobson, C. Leres, and S. McCanne, tcpdump, *http://www.tcpdump.org*, 1989

[9] S. Ostermann, tcptrace, *http://www.tcptrace.org*, 1994.