

전송률과 공평성 향상을 위한 타이머 기반 TCP 혼잡 제어

이종민[○] 차호정
연세대학교 컴퓨터과학과
{jmlee[○], hjcha}@cs.yonsei.ac.kr

A Timer-based TCP Congestion Control Mechanism for Enhancing Throughput and Fairness

Jongmin Lee[○] Hojung Cha
Dept. of Computer Science, Yonsei University

요약

본 논문은 TCP의 무선 환경에서의 전송 성능 감소와 패킷 왕복 시간에 따른 대역폭 선점 문제를 해결하기 위한 타이머 기반의 혼잡 제어 방식을 제안한다. TCP는 패킷 손실 확률이 많은 무선 환경에서 네트워크 혼잡에 의한 패킷 손실을 방지하기 위한 느린 전송률 증가로 인해 전송 성능이 크게 떨어진다. 또한, TCP 송신자는 전송률을 결정하는 전송 윈도우를 수신자로부터 응답 메시지를 받을 때만 조정시키므로, 패킷의 왕복 시간의 차이에 따른 전송률 편중 현상과 다수개의 응답 메시지에 의한 과도한 트래픽 발생의 문제가 발생한다. 본 논문에서 제안하는 타이머 기반의 TCP 혼잡 제어 방식은 패킷의 전송 시간 간격을 타이머로 조정함으로써 무선 환경에서 전송 성능을 향상시키고 패킷의 왕복 시간 차이에 따른 전송률 편중 현상을 완화시키며 다수개의 응답 메시지에 의한 과도한 트래픽의 발생을 제한한다. 제안하는 방법은 실제 환경에서 구현되었으며, 다양한 네트워크 환경에서의 실험을 통해 그 성능을 입증하였다.

1. 서론

TCP는 대표적인 전송 프로토콜로서 신뢰성 있고 연결 지향적인 서비스를 제공하기 위해 네트워크의 혼잡 상황을 제어하기 위한 윈도우 기반의 AIMD (Additive Increase Multiplicative Decrease) 혼잡 제어 방식을 사용한다 [1]. TCP 송신자의 전송률은 수신자로부터의 응답 메시지 수신 없이 전송 가능한 자료의 양을 나타내는 전송 윈도우의 크기로 결정된다. TCP는 패킷 손실의 원인을 네트워크의 혼잡 상황이라 가정하여 패킷 손실이 발생할 때마다 전송률을 감소시킨다. 그러나, 무선 네트워크의 경우 패킷 손실은 링크의 오류로 인해 발생할 수 있으므로 TCP의 잦은 전송률 감소와 혼잡 상황을 피하기 위한 느린 전송률 증가는 무선 환경에서 TCP 전송 성능을 크게 감소시킨다 [2]. 따라서, 무선 네트워크에서의 TCP 전송 성능 향상을 위해서는 네트워크의 혼잡 상황으로 인한 패킷 손실과 링크 오류로 인한 패킷 손실이 서로 다른 방식으로 처리되어야 한다 [3].

윈도우 기반의 혼잡 제어 방식에서 송신자의 전송 윈도우는 전송 패킷의 시간 초과나 송신자가 수신자로부터의 응답 메시지를 받았을 경우 조정된다. 따라서, 패킷 왕복 시간이 서로 다른 두 개 이상의 연결이 하나의 병목 지점에서 경쟁할 경우 패킷 왕복 시간의 차이에 따라 공유 네트워크 대역폭 분배의 불공평 현상이 발생하게 된다 [4]. 또한, 송신자가 수신자로부터 다수개의 응답 메시지를 연속적으로 수신할 경우 전송 윈도우의 급작스런 증가로 과도한 트래픽이 생성될 수 있다. 과도한 트래픽은 라우터의 버퍼링 부하를 증가시키며, 패킷 왕복 시간의 증가와 네트워크 혼잡을 발생시킨다.

본 논문에서는 기존 AIMD 혼잡 제어 방식의 문제점인 패킷 왕복 시간에 대한 전송률 편중 현상과 다수개의 응답 메시지로 인한 과도한 트래픽을 줄이고, 무선 환경에서의 전송 성능을

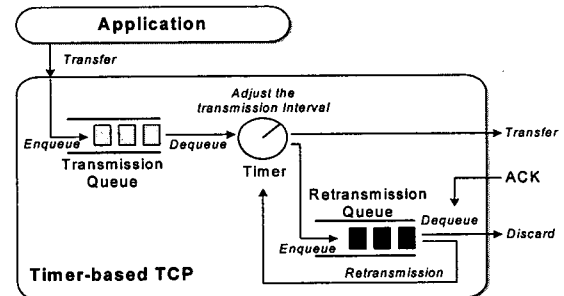


그림 1. 타이머 기반 TCP 구조

높일 수 있는 타이머 기반의 혼잡 제어 방식을 제안한다. 또한, 제안된 방식을 실제 시스템에서 구현하고 성능 분석을 함으로써 제안하는 방식의 효율성을 검증한다. 논문의 구성은 다음과 같다. 2장에서는 논문에서 제안하는 타이머 기반의 혼잡 제어 방식에 대해 기술한다. 3장에서는 실험 방법 및 결과에 대해 기술하고 4장에서 결론을 맺는다.

2. 타이머 기반 혼잡 제어

타이머 기반 TCP는 송신자가 타이머를 유지하며, 전송되는 패킷의 시간 간격을 조정한다. 타이머 기반 TCP에서의 패킷 전송은 타이머 시간이 초과 되었을 경우 수행되며, 응답 메시지의 수신 여부와 관계없이 패킷이 전송되기 때문에 동시에 다수의 응답 메시지가 수신되었을 경우 발생하는 과도한 트래픽의 문제점이 없다. 그림 1은 타이머 기반 TCP의 구조를 보여준다. 응용 프로그램이 자료를 전송하면, 패킷은 전송 큐로 삽입된다. 타이머는 시간 주기를 가지며, 시간이 초과 되었을 경우 전송 큐의 패킷을 네트워크로 전송한다. 이때, 전송되는 패킷은 재전송 큐에 삽입되어 향후 패킷 손실에 대해 재전송을 수행할 수 있도록 한다. 수신자로부터 응답 메시지가 오면 재전송 큐에서 응답 메시지에 상응

• 본 연구는 과학기술부에서 지원하는 국가지정연구실사업으로 수행하였음 (과제번호 : 2005-01352)

하는 패킷이 제거되며, 패킷이 손실된 경우는 해당 패킷을 전송 큐의 앞부분에 삽입하며 우선적으로 전송될 수 있도록 한다. 패킷이 손실된 경우, 기존 TCP는 혼잡 윈도우의 크기를 반으로 줄이며, 현재 네트워크로 응답 메시지를 받지 않고 전송중인 패킷의 갯수가 혼잡 윈도우의 크기가 될 때까지 전송을 일시 중지한다. 그러나, 타이머 기반 TCP는 현재 응답 메시지 없이 전송중인 패킷의 갯수가 줄 때까지 기다리지 않고 전송을 수행한다. 따라서, 불필요한 패킷 전송의 중지가 없으며, 패킷 손실 확률이 높은 무선 네트워크에서 전송 성능을 높일 수 있다.

TCP는 느린 시작 단계와 혼잡 회피 단계의 두 가지 단계를 가지며, 각각의 단계에 따라 다른 전송률 증가를 보인다. 느린 시작 단계에서, 혼잡 윈도우가 매 패킷 왕복 시간마다 두 배로 증가한다고 가정할 경우, 전송률 증가는 식 1과 같다 [4].

$$\frac{W_{cn}}{RTT^2} \text{ bytes/sec} \quad (1)$$

식 1에서와 같이 느린 시작 단계에서의 TCP 전송률은 패킷 왕복 시간 RTT의 제곱에 반비례 하며 혼잡 윈도우의 크기 W_{cn} 에 비례하여 증가한다. 따라서, RTT가 서로 다른 트래픽이 하나의 병목 링크에서 경쟁할 경우 RTT가 작은 트래픽이 전송률 증가를 빠르게 하고, 결과적으로 공유 네트워크 대역폭을 선점하게 된다. 이러한 문제는 RTT에 대한 전송률 편중 현상을 발생시킨다. 타이머 기반 TCP는 타이머의 주기와 매 주기마다 전송해야 할 자료의 양으로 전송률을 결정한다. 타이머 기반 TCP는 느린 시작 단계와 혼잡 회피 단계를 가지며 느린 시작단계에서는 현재의 전송률을 유지하는 타이머의 주기 N_t 마다 전송률을 두 배로 증가시킨다. 타이머의 주기를 T_p , 매 주기마다 전송해야 할 자료의 양을 S_t 라 하면 타이머 기반 TCP의 느린 시작 단계에서의 전송률 증가는 식 2와 같다.

$$\frac{S_t}{T_p N_t} \text{ bytes/sec} \quad (2)$$

수식 2와 같이 타이머 기반 TCP의 전송률 증가는 패킷 왕복 시간 RTT와는 관계가 없으며, 오직 타이머 주기 T_p , 매 주기마다 전송해야할 자료의 양 S_t , 그리고 현재 전송률을 유지하는 기간 N_t 에 의해서만 결정된다. 따라서, RTT가 서로 다른 트래픽이 하나의 병목 링크에서 경쟁하더라도 전송률이 동일하게 증가하므로 특정 트래픽에 전송률이 편중되는 현상이 없다.

타이머 기반 TCP는 패킷 전송이 시간 초과되거나 송신자가 수신자로부터 세 개의 중복된 응답 메시지를 수신하였을 때 패킷이 손실되었다고 판단한다. 또한, 패킷 손실의 원인을 명확히 파악하기 위한 검증 단계를 거친다. 만약 패킷 손실이 발생하였을 때 전송 중이던 자료의 양이 이전 패킷 손실이 발생하였을 때 전송 중이던 자료의 양과 비슷하다면 패킷 손실은 네트워크의 혼잡 상황으로 발생할 확률이 크며, 그 조건은 식 3에 의해 계산된다.

$$\left| S_{pflight} - S_{cflight} \right| \leq \alpha, \text{ where } \alpha = \frac{S_{pflight} (RTO - RTT)}{4RTT} \quad (3)$$

$S_{cflight}$ 는 현재 수신자로부터의 응답 메시지 수신 없이 송신자가 전송중인 자료의 양이며, $S_{pflight}$ 는 과거 패킷 손실이 발생했을 당시 송신자가 수신자로부터 응답 메시지를 수신하지 않고 전송중인 자료의 양이다. 패킷 왕복 시간 RTT의 변화는 전송 시간 초과 RTO에 반영되며, RTT는 네트워크 대역폭과의 곱으로 표시되는 대역폭 지연 산출 [5]에 직접적으로 영향을 끼치므로, RTT의 변화는 네트워크로 응답 메시지를 받지 않고 전송 가능한 자료

양의 변화를 반영한다. 즉, RTT의 변화를 네트워크 혼잡 상황을 검증할 때 반영함으로써 네트워크의 혼잡 상황을 정확히 파악하도록 한다.

타이머 기반 TCP는 패킷 손실의 원인이 링크 오류라고 판단되면, 느린 시작 단계로 동작하고, 네트워크 혼잡이라고 판단되면 혼잡 회피 단계로 동작한다. 혼잡 회피를 위한 전송률 감소는 네트워크의 혼잡 발생 빈도에 따라 식 4에 의해 결정한다.

$$S_{t(i)} = \alpha S_{t(i-1)} \text{ or } T_{p(i)} = \frac{T_{p(i-1)}}{\beta}, \text{ where } \alpha = \frac{N_c + 1}{N_c + 2} \quad (4)$$

식 4에서 N_c 는 네트워크 혼잡 상황 발생 빈도이다. 즉, 혼잡 발생 횟수가 많을수록 전송률 감소의 폭을 줄여서 현재의 전송률을 유지하도록 한다. 전송률 감소의 폭이 줄 경우 네트워크 혼잡 상황 발생 확률이 오히려 증가할 수 있으나, 현재의 전송률을 유지하는 기간을 늘림으로써 네트워크 혼잡 상황 발생 확률을 줄이고 안정적인 전송 상태로 유지할 수 있도록 한다. 네트워크 혼잡 상황을 감지했을 경우의 전송률 유지 기간은 식 5에 의해 정한다.

$$N_{t(i)} = \alpha N_{t(i-1)}, \text{ where } \alpha = \frac{N_c + 1}{N_c + 2} \quad (5)$$

현재의 전송률을 유지하기 위한 주기의 갯수 $N_{t(i)}$ 는 과거의 주기 갯수 $N_{t(i-1)}$ 에 네트워크 혼잡 발생 횟수를 고려하여 계산하며, 네트워크 혼잡 발생 확률이 클수록 $N_{t(i)}$ 를 크게 함으로써 오랜 시간 동안 현재 전송률을 유지할 수 있도록 한다. 타이머 기반 TCP는 패킷 손실이 발생하였을 경우 식 3에 의해 패킷 손실의 원인을 분류하고 패킷의 손실이 네트워크의 혼잡상황 때문이라고 판단되면, 전송률을 네트워크 혼잡 상황 발생 횟수에 비례하여 감소시키고, 전송률을 느리게 증가시킨다. 만약 패킷 손실의 원인이 링크 오류 때문이라고 판단되면, 전송률을 반으로 줄이고 느린 시작 단계를 시작하게 된다. 네트워크의 대역폭이 변하는 경우나 패킷의 손실 원인을 잘못 판단한 경우, 혼잡 회피 단계의 느린 전송률 증가는 전송 성능을 크게 감소시킨다. 따라서, 현재 송신자가 수신자로부터의 응답 메시지 없이 전송중인 자료의 양이 식 3의 범위를 넘을 때는 느린 시작단계로 전환함으로써 전송률 증가를 빠르게 하여 대역폭 변화에 대처할 수 있도록 한다.

3. 실험

본 논문에서 제안한 타이머 기반 TCP는 리눅스 2.6.11 커널 코드를 수정하여 구현되었다. 실험을 위해 두 대의 시스템을 사용하였다. 다양한 네트워크 환경에서의 실험을 위해 네트워크 에뮬레이터인 NIST Net을 사용하였고, 실험 자료의 분석은 tcpdump, tcptrace, netperf 프로그램을 사용하였다 [6]. 또한, 성능 비교를 위해 무선 네트워크를 위한 프로토콜인 TCP Westwood+ [7]와 고속 네트워크를 위한 프로토콜인 BIC TCP [8]도 같이 실험하였다. 그림 2는 패킷 왕복 시간에 따른 전송되는 패킷의 개수를 보여준다.

그림 2에서 기울기는 전송률을 의미한다. 기존 TCP의 경우 RTT에 따라 기울기가 다른 것을 볼 수 있으며, 이는 기존 TCP가 RTT에 따른 전송률 편중 현상을 가지고 있음을 증명한다. 타이머 기반 TCP의 경우 응답 메시지의 수신여부와 상관없이 타이머에 의해 패킷의 전송이 결정된다. 따라서, 타이머 기반 TCP의 전송률 증가는 그림 2에서 보듯이 RTT와 상관없이 일

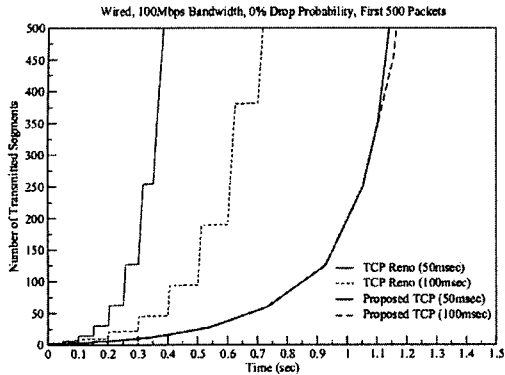


그림 2. RTT에 따른 패킷 전송

정하다. 이는 타이머 기반 TCP가 RTT에 대한 전송률 편중 현상을 완화했음을 증명한다. 기존 TCP는 패킷의 전송이 계단식으로 증가한다. 이는 TCP 송신자가 다수개의 응답 메시지를 수신했을 경우 전송 윈도우를 급작스럽게 증가시키기 때문이며, 이는 과도한 트래픽을 발생시켜 중간 라우터의 버퍼링 부하를 가중시키고 네트워크 혼잡을 발생시킨다. 그러나, 타이머 기반 TCP는 패킷의 전송이 완만한 곡선을 보이며, 이는 타이머에 의한 패킷 시간 간격 조정으로 과도한 트래픽을 발생시키지 않기 때문이다.

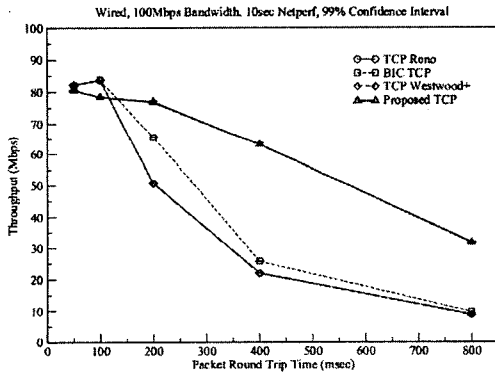


그림 3. RTT에 따른 전송 성능

그림 3은 RTT에 따른 전송 성능을 보여준다. 기존 TCP의 경우 RTT에 대한 전송률 편중 현상이 있으며, RTT가 클수록 전송률의 증가가 느리다. 따라서, RTT가 클수록 전송 성능이 급격히 떨어진다. TCP Westwood+ 또한 기존 TCP의 혼잡 제어 방식을 계승하므로 RTT에 대한 전송률 편중 현상으로 인해 RTT가 증가할수록 전송 성능이 떨어진다. BIC TCP의 경우 고속 네트워크에서의 전송률 향상을 위해 전송률 증가를 빠르게 하지만 RTT에 대한 전송률 편중 현상 때문에 RTT가 클수록 전송 성능이 떨어진다. 타이머 기반 TCP의 경우 RTT에 대한 전송률 편중 현상이 없으므로 RTT가 증가할수록 전송 성능이 급격히 떨어지지 않는다. 그러나, 수신자에서의 Advertised 윈도우의 제한으로 RTT가 클 경우 전송 성능이 약간 떨어진다. 그러나, Advertised 윈도우에 의한 전송 성능 감소는 전송 시간이 길어짐에 따라 없어진다.

그림 4는 패킷 손실 확률의 변화에 따른 기존 TCP와의 성능 향상 정도를 보여준다. TCP Westwood+와 BIC TCP는 무선 네

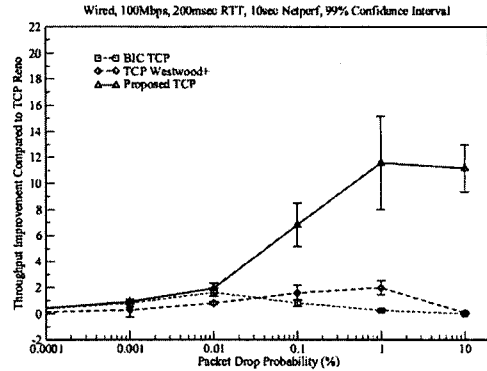


그림 4. 패킷 손실 확률에 따른 전송 성능 향상

트워크를 고려하므로 기존 TCP보다 약간의 전송 성능 향상을 보이고 있다. 타이머 기반 TCP는 패킷 손실 원인 분류에 따른 적응적 전송률 조정과 패킷 손실시 전송되는 자료의 양이 줄어들기를 기다리는 일시 전송 중지가 없으므로 패킷의 손실률이 클수록 전송 성능이 급격히 향상되는 것을 확인할 수 있다.

4. 결론

본 논문에서는 타이머 기반 TCP 혼잡 제어 방식을 제안하였다. 제안한 방법은 타이머로 전송되는 패킷의 시간 간격을 조절하여 다수개의 응답 메시지에 의한 과도한 트래픽과 RTT에 대한 전송률 편중 현상을 완화하였다. 또한, 패킷 손실의 원인 분류에 따른 적응적 전송률과 패킷 손실시 전송률의 감소를 기다리지 않는 지속적인 전송으로 무선 환경에서의 전송 성능을 크게 향상시켰다.

제안된 타이머 기반 TCP 혼잡 제어 방식은 송신자의 변경만을 필요로 하며 실제 시스템으로 구현되었다. 다양한 TCP와의 성능 비교 분석을 통해 전송 시간에 대한 전송률 편중 현상이 완화되었으며, 패킷 손실률이 높을수록 전송 성능이 향상됨을 보였다. 향후 연구 과제로는 패킷 손실 원인 분류의 정확도 향상과 서비스 자동화를 위한 자원 관리 정책과의 연계 방안 연구가 있다.

참고문헌

- [1] V. Jacobson, "Congestion Avoidance and Control," *Proceedings of ACM SIGCOMM*, pp. 314-329, 1998.
- [2] Kevin Lai, Mary Baker, "Measuring Bandwidth," *Proceedings of IEEE INFOCOM'99*, pp. 235-245, New York, 1999.
- [3] Song Cen, Pamela C. Cosman, "End-to-End Differentiation of Congestion and Wireless Losses," *IEEE/ACM Transactions on Networking*, vol. 11, no. 5, pp. 703-717, October 2003.
- [4] Floyd S. and Van. Jacobson, "On Traffic Phase Effects in Packet-Switched Gateways," *Journal of Internet Working: Research and Experience*, Vol. 3, No. 3, September 1992, pp. 115-156.
- [5] Brian L. Tierney, "TCP Tuning Guide for Distributed Applications on Wide Area Networks," *Usenix: login: journal*, vol. 26, no. 1, pp. 33-39, 2001.
- [6] S. Parker, C. Schmechel, "Some Testing Tools for TCP Implementers," *Internet Engineering Task Force*, RFC 2398, August 1998.
- [7] L.A. Grieco, S. Mascolo, "Performance evaluation and comparison of Westwood+, New Reno, and Vegas TCP congestion control," *ACM SIGCOMM Computer Communication Review*, vol. 34, no. 2, pp. 25-38, April 2004.
- [8] Lisong Xu, Khaled Harfoush, and Injong Rhee, "Binary Increase Congestion Control (BIC) for Fast, Long Distance Networks," *Proceedings of IEEE INFOCOM'04*, Hong Kong, 2004.