

무선망에서의 Bufferbloat에 따른 TCP 혼잡 제어 알고리즘 성능 비교

한유슬*, 김병철*, 이재용*

*충남대학교 전파정보통신공학과

yoursseull107@o.cnu.ac.kr, {byckim, jy1}@cnu.ac.kr

Performance Comparison of TCP Congestion Control Algorithms in Wireless Networks with Bufferbloat

Han Yoo Seul, Kim Byung Chul, Lee Jae Yong

Department of Radio and Information Comm. Eng., Chungnam National Univ.

요약

현대 인터넷에서 네트워크 데이터 전송의 중요성이 커짐에 따라, 버퍼 크기 확대에 의한 지연 문제인 Bufferbloat가 주목받고 있다. Bufferbloat 현상이 발생하면 불필요한 지연 시간이 증가하는 문제가 발생할 수 있다. 본 논문은 무선 네트워크 환경에서 Bufferbloat 문제에 따른 TCP 혼잡 제어 알고리즘 (Reno, Cubic, Vegas, BBR)의 성능을 Mininet-Wifi로 비교하였다. 실험 결과 Reno와 Cubic은 큰 버퍼에서 지연 시간이 많이 증가한 반면, BBR과 Vegas는 지연을 효과적으로 완화하였다. 특히 BBR이 낮은 지연 시간과 네트워크 자원 분배의 공정성에 있어서 가장 우수한 성능을 보였다.

I. 서론

최근 인터넷 트래픽은 주로 무선 단말을 통한 비디오 트래픽이며, 이 추세는 점점 더 증가하고 있다[1]. 이때 망을 구성하는 각 노드의 버퍼 크기가 작으면 대용량 데이터 전송 시 패킷 손실로 인해 사용자의 체감 품질이 낮아진다. 이를 해결하기 위해 각 노드의 버퍼 크기를 늘리는 방안이 메모리 가격 하락 추세와 맞물려 도입되었다.

그러나 과도하게 커진 버퍼 크기는 queuing delay 증가로 인한 사용자의 응답 시간 증가와 같이 전체 네트워크 성능에 부정적인 버퍼블로트(Bufferbloat)[2] 문제를 발생시킨다. 본 논문은 Mininet-Wifi[3]와 Mininet[4]을 이용하여 다양한 망 환경에서 Reno 기반 TCP 혼잡 제어와 최근 제안된 혼잡 제어 알고리즘의 버퍼블로트 문제에 대한 성능 비교를 수행한다.

성능 비교를 위해 RTT, Throughput 및 짧은 파일의 다운로드 완료 시간, 다중 세션 환경에서의 공정성을 Jain's fairness index[5] 기반으로 분석하였다.

II. 본론

1. 기존 연구 동향

최근 수년간 버퍼블로트 문제 해결을 위한 연구는 크게 적절한 AQM (Active Queue Management) 알고리즘을 적용하여 버퍼 크기가 적절한 수준에서 관리되도록 하는 방안[6][7]과 다양한 망에서의 버퍼블로트 현상이 일으키는 문제점 파악[8][9]의 두 방향으로 진행되었다.

그러나 다양한 topology와 실험 환경을 조성하여 쉽게 실험할 수 있는 이플래이터 기반으로 무선 환경에서의 버퍼블로트 문제를 다른 연구는 상대적으로 부족하다. 무선 네트워크의 특수성(신호 간섭, 패킷 유실, 대역폭 제약 등)을 고려하여 각 CCA 별로 버퍼블로트의 문제점을 파악하기 위해서는 테스트베드나 시뮬레이션보다는 에뮬레이터를 이용한 방법이

효과적이다.

2. 실험환경 및 구성

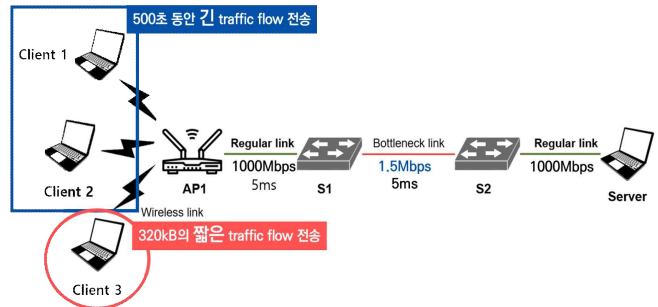


Figure 1. 실험 Topology

Figure 1과 같이 구성된 유무선 혼합 네트워크를 에뮬레이트하기 위해 Mininet-Wifi로 실험을 수행하였다. Ubuntu 20.04, Linux Kernel v5.15.72과 Iperf3[10], HTTP를 활용하여 TCP CCA의 성능을 비교하였다.

3개의 Client, 1개의 Server, 1개의 AP, 2개의 스위치로 네트워크를 구성하였고, Client와 AP는 Wi-Fi로, 나머지 노드는 유선으로 연결하였다. 유선 링크는 기본적으로 1,000Mbps, 5ms 지연을 가지며, 두 스위치 사이의 대역폭은 1.5Mbps로 설정하여 병목구간으로 설정하였다.

세 Client의 TCP CCA가 Reno[11], Cubic[12], Vegas[13], BBR[14]일 때 S1의 버퍼 크기를 2BDP, 10BDP로 변경하며 실험을 수행했다.

이때 Client 1과 2는 Server 쪽으로 500초간 Iperf3 테스트 트래픽(long traffic flow)을 보내고, 그 시간 동안 Client 3은 Server에 있는 320KB 파일(short traffic flow)을 반복 다운로드한다.

3. 실험 및 평가

Figure 2, 3, 4는 병목구간의 버퍼 크기가 2BDP일 때와 10BDP일 때 Client 1과 2의 평균 RTT, 평균 처리량, 평균·순간 처리량 기반 Jain's Fairness Index 그래프이다. 예측 기반 CCA인 Vegas와 BBR은 모든 경우에 낮은 RTT를 유지하며, 특히 BBR은 병목 대역폭과 RTT를 기반으로 전송 속도를 제어하기 때문에 버퍼 크기에 큰 영향을 받지 않고 안정적인 RTT를 보인다.

반면, 손실 기반 CCA인 Reno와 Cubic은 버퍼 크기 증가 시 RTT가 많이 증가한다. 특히, Cubic은 10BDP에서 RTT가 급격히 증가한다. 이는 Cubic이 CWND 조절에 큐빅 함수($W(t) = C(t - K)^3 + W_{max}$)를 사용하여 과도하게 전송 속도를 증가시킴에 따라 혼잡을 늦게 감지하여 많은 데이터를 전송하고 큰 버퍼 크기를 가지기 때문이다.

모든 알고리즘에서 버퍼 크기와 관계없이 유사한 처리량을 보이므로 Reno와 Cubic의 RTT 증가는 버퍼블로트 발생을 시사한다.

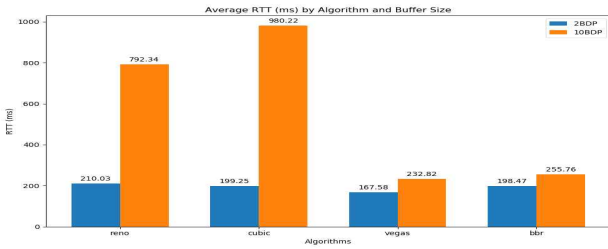


Figure 2. 버퍼 크기에 따른 Client 1, 2의 평균 RTT(ms)

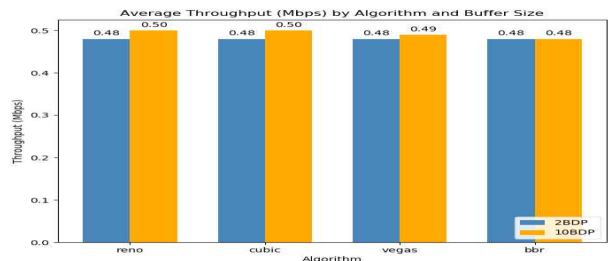


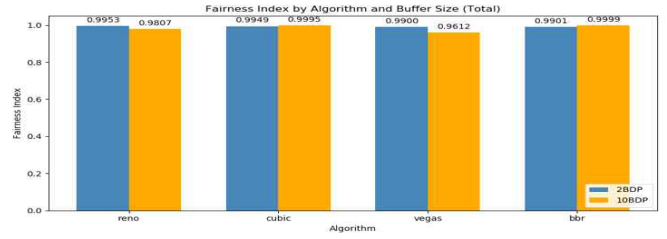
Figure 3. 버퍼 크기에 따른 Client 1, 2의 평균 처리량(Mbps)

Figure 4에 따르면 평균 처리량은 모든 경우에 매우 높은 공정성 지수를 보이며 자원을 공정하게 분배하는 것을 보여준다. 다만, Reno는 큰 버퍼에서 시간별 순간 처리량의 공정성이 많이 떨어지는데 각 TCP 세션이 독립적으로 운영되어 손실의 동기가 이루어지지 않고 특정 시점에서의 성능 차이가 크게 나기 때문이다.

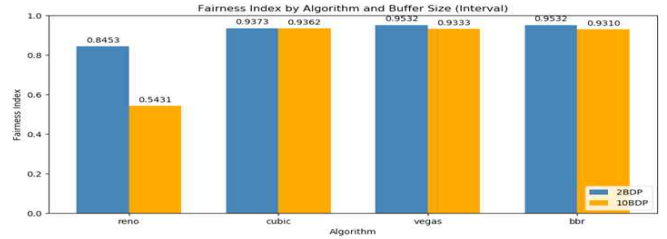
Figure 5는 Client 3에서 짧은 길이의 파일을 내려받을 때 소요되는 평균 시간 그래프이다. 긴 traffic이 이미 존재하는 상황에서의 완료 시간 변화로 버퍼블로트에 따른 queuing delay를 확인할 수 있다. RTT 추세와 유사하나, 10BDP에서 Reno와 Cubic의 완료 시간이 크게 증가하였다. BBR과 Vegas는 버퍼 크기와 관계없이 짧은 완료 시간을 보인다.

III. 결론

무선 환경에서의 트래픽 증가로 인해 버퍼블로트가 있는 무선망에서 각 TCP CCA 별 성능 비교를 수행하고 그 차이를 분석하는 것은 중요한 연구 과제이다. 본 논문에서는 실제 테스트베드가 아닌 Mininet과 Mininet-Wifi를 통한 성능 분석을 수행하였다.



(a) 평균 처리량



(b) 시간별 순간 처리량

Figure 4. 버퍼 크기에 따른 Client 1, 2의 전체, 구간별 Jain's Fairness Index

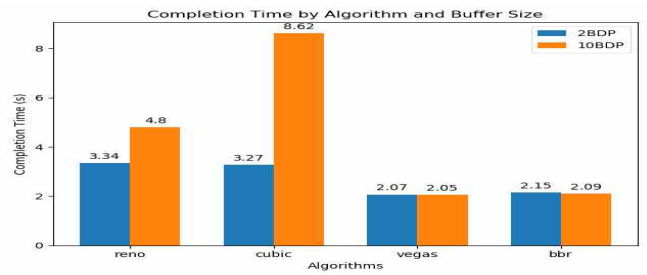


Figure 5. 버퍼 크기에 따른 Client 3의 평균 Completion time(s)

실험 결과 Reno는 전통적인 AIMD 방식으로 혼잡을 제어하기 때문에 큰 버퍼에서는 혼잡을 늦게 감지하여 지연 시간이 많이 증가한다. Cubic의 경우, 높은 처리량과 공정성 지수를 유지하지만, 큰 버퍼에서 CWND가 과도하게 커지면서 RTT와 완료 시간이 급격히 증가하는 모습을 보인다. Vegas와 BBR은 버퍼블로트에 영향을 덜 받으며, 특히 BBR은 병목 대역폭과 RTT를 지속적으로 평가하여 전송 속도를 조절함으로써 낮은 RTT와 높은 공정성을 유지하고, 버퍼블로트 현상을 효과적으로 완화하면서 안정적인 처리량을 제공할 수 있었다.

참고 문헌

- [1] Ericsson, "Ericsson Mobility Report 2024: Mobile Network Data Traffic Jumps 25% Year-On-Year Driven by 5G Migration and Data-Intense Services," 2024, (<https://www.ericsson.com/en/reports-and-papers/mobility-report/dataforecasts/mobile-traffic-forecast>)
- [2] Jim G., "Bufferbloat: Dark Buffers in the Internet," IEEE Internet Computing, vol.15, pp. 96-96, Apr. 2011.
- [3] Ramon R. F. et al., "Mininet-WiFi: Emulating software-defined wireless networks," 2015 11th International Conference on Network and Service Management (CNSM), Nov. 2015.
- [4] Mininet Walkthrough, "Mininet," (<https://mininet.org>)

- [5] Rajendra K. J. Dah-Ming W. C., William R. H., "A Quantitative Measure Of Fairness And Discrimination For Resource Allocation In Shared Computer Systems,"DEC Research Report, TR-301, Sep. 1984.
- [6] Gwangwoo P., Haneul K., Sangheon P., "Simulation study of bufferbloat problem on WiFi access point,"GCCE 2014, pp. 729-730, Feb. 2014.
- [7] Hao W., "Trade-off queuing delay and link utilization for solving bufferbloat," KICS ICT Express 6(2020), pp. 269-272, Jun. 2020.
- [8] Do T. T. H. et al, "Targeting Bufferbloat in Wi-Fi Networks: An Emulator-based Approach," 2019 ISCIT, Sep. 2019.
- [9] Stefan A. et al., "Impact of TCP congestion control on bufferbloat in cellular networks,"2013 IEEE 14th WoWMoM, Jun. 2013.
- [10] ESnet, "iPerf - The ultimate speed test tool for TCP, UDP and SCTP," (<https://iperf.fr/>)
- [11] Mark A. et al., "TCP Congestion Control,"RFC 2581, Apr. 1999.
- [12] Sangtae H. Injong R. Lisong X., "CUBIC: a new TCP-friendly high-speed TCP variant,"ACM SIGOPS Operating Systems Review, Vol. 42, pp. 64-74, Jul. 2008.
- [13] Lawrence S. B., Larry L. P., "TCP Vegas: end to end congestion avoidance on a global Internet," IEEE Journal on Selected Areas in Communications, Vol. 13, pp. 1465-1480, Oct. 1995.
- [14] Neal C. et al., "BBR: Congestion-Based Congestion Control,"ACM Queue, pp. 20-53, Sep. 2016