

Mininet을 활용한 5G 네트워크에서의 TCP 혼잡제어 성능평가 : AQM 적용에 따른 Bufferbloat의 완화

남윤지, 이재용*, 김병철

충남대학교 전파정보통신공학과

202201891@o.cnu.ac.kr, *jyl@cnu.ac.kr, byckim@cnu.ac.kr

Performance Evaluation of TCP Congestion Control in 5G Networks Using Mininet: Bufferbloat Mitigation with AQM

Nam Yun Ji, Lee Jae Yong*, Kim Byung Chul

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

요약

본 논문은 Mininet 기반 에뮬레이션을 활용한 5G 네트워크의 가변 대역폭 환경에서 TCP 혼잡제어 알고리즘인 CUBIC과 BBR의 성능을 평가하고, AQM(CoDel) 적용 여부에 따른 성능 차이를 분석하였다. DropTail 방식의 대역폭 버퍼 환경에서 대역폭 변동이 매우 클 경우 bufferbloat 현상이 두드러졌으며, CUBIC은 지연 증가와 처리량 저하가 뚜렷하게 나타났다. 반면 BBR은 대역폭 변화에 신속하게 적응하며 상대적으로 안정적인 성능을 보였다. CoDel을 적용한 경우 두 알고리즘 모두 지연이 정상 수준으로 억제되고 전송 효율이 개선되는 경향을 확인하였다. 이러한 결과는 5G와 같이 대역폭 변동이 심하고 대역폭 버퍼가 존재하는 환경에서 bufferbloat 문제를 완화하기 위해 AQM 적용이 필수적임을 실험적으로 입증한다.

I. 서론

5세대(5G) 이동통신은 초고속·초저지연·초연결 특성을 기반으로 다양한 서비스와 응용을 가능하게 한다. 그러나 5G 무선 채널은 장애물로 인한 LOS(Line of Sight)와 NLOS(Non-Line of Sight) 전환, 자원의 동적 할당 등으로 인해 순간적으로 대역폭이 급격히 변한다. 이 과정에서 대형 버퍼가 존재할 경우 bufferbloat 현상이 발생할 수 있다[1].

Bufferbloat은 패킷 손실 대신 큐 적체로 인해 지연이 과도하게 증가하는 문제로[2], 실시간성이 중요한 응용 서비스 품질에 큰 영향을 미친다. 기존 손실 기반 혼잡제어 알고리즘인 TCP CUBIC[3]은 손실 신호를 통해 전송률을 조절하지만, 버퍼가 큰 환경에서는 손실이 늦게 발생하여 심각한 지연을 유발한다. 반면 구글이 제안한 BBR[4]은 병목 대역폭과 RTT 추정을 바탕으로 전송률을 설정하여 bufferbloat을 완화할 수 있으나, 큐를 발생시키는 동작 특성으로 인해 지연이 완전히 억제되지 않는다.

이러한 문제 해결을 위한 방법으로 AQM(Active Queue Management)이 연구되고 있으며, 특히 CoDel[5]은 큐 지연을 제어하는 대표적인 기법으로 알려져 있다. 따라서 본 연구에서는 Mininet[6] 기반 5G 에뮬레이션 환경을 구축하고[7], TCP CUBIC과 BBR의 성능을 DropTail 및 CoDel 환경에서 비교·분석하여 bufferbloat 문제 해결 가능성을 검증하였다.

II. 기존 연구 동향

5G 네트워크 환경에서는 대역폭 급변, 긴 RTT, 대역폭 버퍼로 인한 bufferbloat 현상이 자주 발생한다[2]. 이러한 문제를 완화하기 위해 다양한 TCP 혼잡제어 알고리즘과 큐 관리 기법이 제안되어 왔다[1].

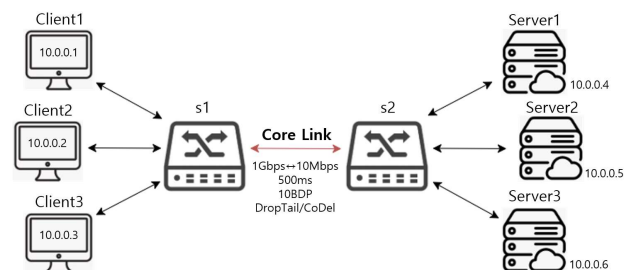
TCP CUBIC은 손실 신호를 기반으로 혼잡 윈도우를 조절하는 알고리즘으로, 고속·고지연 네트워크에 적합하도록 설계되었다[3]. 그러나 버퍼가 큰 환경에서는 손실 발생이 지연되어 큐 적체가 누적되고, 이로 인해 지연이 크게 증가하는 문제가 발생할 수 있다. TCP BBR은 손실에 의존하지 않고, 병목 대역폭과 최소 RTT를 추정하여 전송률을 제어하는 모델 기반

알고리즘이다[4]. 이 방식은 buffer bloat 완화에 유리하지만, 대역폭 탐색 과정에서 일시적으로 큐가 형성되어 지연이 증가하는 한계를 가진다.

큐 관리 방식 측면에서, DropTail은 큐가 가득 찼을 때까지 패킷을 수용하다가 초과 시점에 폐기하는 기본 정책으로, 구현이 단순하지만 buffer bloat을 억제하지 못한다. 반면, CoDel(Control Delay)은 큐 지연을 기준으로 패킷을 능동적으로 드롭하여 지연을 일정 수준 이하로 유지시키는 방식으로, bufferbloat 문제 완화에 효과적인 것으로 알려져 있다[5].

III. 동적 5G 네트워크에서의 TCP 혼잡제어 실험 구성

본 논문에서는 Ubuntu 22.04 LTS 환경에서 네트워크 에뮬레이션 도구인 Mininet[6]과 GUI 기반의 MiniEdit을 활용하여 5G 네트워크 환경을 가상으로 구현하였다[7]. 링크 특성은 NetEm과 HTB로 설정하였으며, 각 Client의 트래픽은 Server1~3으로 전송되도록 구성하였다.



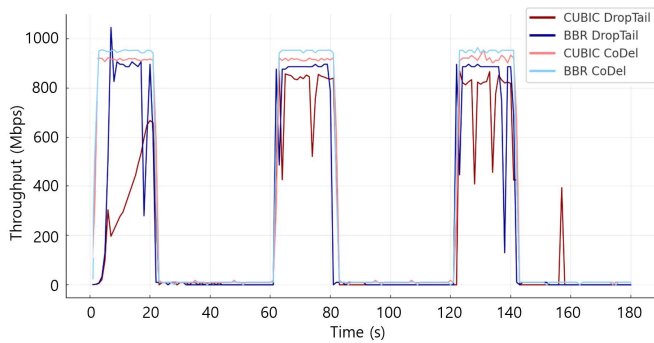
<그림 1> 5G 네트워크 환경을 반영한 실험망 구성

스위치 간 링크(s1-s2)에 큐 관리 정책을 적용하고, 알고리즘별로 5회 반복 측정 후 평균값을 성능 지표로 사용하였다. 전체 실험 시간은 180초이며, 링크 대역폭은 1Gbps(20초)와 10Mbps(40초)를 주기적으로 변동시켜 급변하는 네트워크 상황을 모사하였다. 왕복 지연은 500ms, 버퍼 크기는 1Gbps 기준 10BDP로 고정하였고, 패킷 손실은 버퍼 포화로 인한 손실만 반영하였다. 혼잡제어 알고리즘은 TCP CUBIC[3]과 TCP

BBR[4], 큐 관리 방식은 DropTail과 CoDel[5]를 적용하였다. 성능 측정 지표는 처리량(Throughput)과 왕복 지연(RTT)이며, 측정에는 iperf3 도구를 사용하였다.

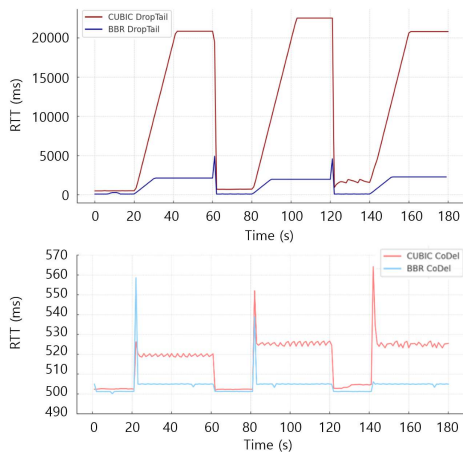
IV. 동적 5G 네트워크에서의 TCP 혼잡제어 성능 평가

<그림 2>는 TCP 혼잡제어 알고리즘이 5G 네트워크의 대역폭 변동에 따라 보이는 처리량 변화를 나타낸 것이다. DropTail 환경에서는 CUBIC이 링크 대역폭을 충분히 활용하지 못하고 처리량 회복 속도가 느리며, 순간적으로 급격한 하락 구간이 반복적으로 나타났다. BBR은 상대적으로 빠르게 적응하여 평균적으로 높은 처리량을 유지했으나, 대역폭 급변 구간에서 일정 수준의 불안정성이 관찰되었다. 반면, CoDel을 적용한 경우 두 알고리즘 모두 처리량이 안정적으로 유지되었으며, 대역폭 급변 상황에서도 큰 성능 저하 없이 지속적인 전송이 가능함을 확인할 수 있었다.



<그림 2> CUBIC과 BBR의 처리량 비교

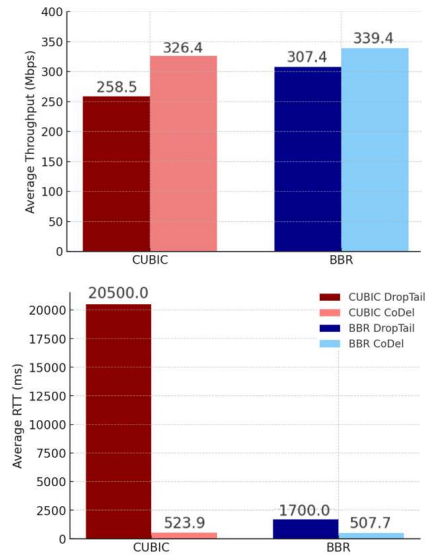
<그림 3>은 bufferbloat 현상을 보여주는 RTT 결과이다. DropTail 환경에서 CUBIC은 평균 20.5초까지 폭증하며 극심한 지연이 발생하였고, 이는 큐 적체가 손실 없이 누적된 전형적인 bufferbloat 현상이다. 같은 조건에서 BBR은 평균 1.7초 수준으로 억제되었으나 여전히 정상 범위를 크게 벗어난 지연을 보였다. 반면, CoDel을 적용한 경우 두 알고리즘 모두 RTT가 약 500ms 수준으로 안정화되어, 지연 제어 효과가 뚜렷하게 나타났다. 이를 통해 CoDel이 bufferbloat 완화에 효과적임을 실험적으로 확인할 수 있다.



<그림 3> CUBIC과 BBR의 RTT 비교 (DropTail 및 CoDel 적용)

<그림 4>는 실험 전체 구간의 평균 처리량과 지연이 발생한 구간에서의 평균 RTT를 비교한 결과이다. CoDel 적용 시 CUBIC은 처리량이 326.4Mbps로 약 26% 향상되고 RTT가 523.9ms로 크게 개선되었으며, BBR 역시 처리량이 339.4Mbps로 약 10% 향상되고 RTT가 507.7ms로 안정화되었다. 즉, CoDel 적용이 두 알고리즘 모두에 대해 성능 향상을

가져옴을 실험적으로 보여준다.



<그림 4> 평균 처리량, RTT 비교 (DropTail vs CoDel, CUBIC vs BBR)

V. 결론

본 논문에서는 Mininet을 활용해 5G 네트워크의 가변 대역폭 환경에서 TCP 혼잡제어 알고리즘인 CUBIC과 BBR의 성능을 DropTail 및 CoDel 조건에서 평가하였다. DropTail 환경에서는 bufferbloat 현상이 심각하게 발생하였으며, 특히 CUBIC은 처리량 저하와 극심한 지연을 보였다. BBR은 상대적으로 높은 처리량을 유지했으나, 지연 억제에는 한계가 있었다. 반면, CoDel을 적용한 경우 두 알고리즘 모두 처리량과 지연이 안정화되었으며, CoDel 적용 시의 BBR이 가장 우수한 성능을 기록하였다. CUBIC 또한 CoDel 적용으로 처리량과 지연 면에서 뚜렷한 개선 효과를 보였다.

이 결과는 5G와 같이 대역폭 변동이 심하고 버퍼 크기가 큰 환경에서 AQM의 적용이 필수적임을 보여준다. 향후에는 다양한 AQM 기법과 최신 혼잡제어 알고리즘을 대상으로 확장 연구를 진행할 예정이다.

참 고 문 헌

- [1] Khorov, E., Krasilov, A., Susloparov, M., and Kong, L., "Boosting TCP & QUIC performance in mmWave, Terahertz, and Lightwave wireless networks: A survey," IEEE Commun. Surveys Tuts., vol. 25, no. 4, pp. 3064-3096, 2023.
- [2] Gettys, J., and Nichols, K., "Bufferbloat: Dark buffers in the Internet," Commun. ACM, vol. 55, no. 1, pp. 57-65, Jan. 2012.
- [3] Ha, S., Rhee, I., and Xu, L., "CUBIC: A new TCP-friendly high-speed TCP variant," ACM SIGOPS Operating Systems Review, vol. 42, no. 5, pp. 64-74, Jul. 2008.
- [4] Cardwell, N., Cheng, Y., Gunn, C. S., Yeganeh, S. H., and Jacobson, V., "BBR: Congestion-Based Congestion Control," ACM Queue, vol. 14, no. 5, pp. 20-53, Oct. 2016.
- [5] Nichols, K., and Jacobson, V., "Controlling Queue Delay," Communications of the ACM, vol. 55, no. 7, pp. 42-50, Jul. 2012.
- [6] B. Lantz, B. Heller, N. McKeown, "A Network in a Laptop: Rapid Prototyping for Software-Defined Networks," ACM SIGCOMM HotNets-IX, Monterey, USA, pp. 1-6, Oct. 2010.
- [7] Crichigno, J., Network Tools and Protocols Lab Series, University of South Carolina, Book Version: 01-25-2021.