

에뮬레이션 환경에서의 BBRv3 전송 성능 측정 및 비교 분석

홍원택

한국과학기술정보연구원 과학기술연구망센터

wthong@kisti.re.kr

Measurement and Analysis of Throughput for BBRv3 on Emulation-Based Testbed

Wontaek Hong

KREONET Center, KISTI

요 약

Google에서 제안된 BBR(Bottleneck Bandwidth and RTT) 혼잡제어 알고리즘은 최대 대역폭과 최소 RTT를 추구하는 Network path 모델을 기반으로 동작하고 현재까지 BBRv3가 발표되어 활용 및 검증되고 있다. 본 논문에서는 원거리 네트워크 환경을 위한 BBRv3 에뮬레이션 테스트베드 환경을 구축하고 RTT, 패킷 손실률의 변화에 따른 처리율을 측정하여 기존 손실기반 혼잡제어 알고리즘과의 성능 비교를 수행한다. 이를 통해 과학응용을 위한 고성능 네트워크 환경을 지향하는 과학기술연구망 분야에서 기존 손실기반 혼잡제어 알고리즘들을 대체할 수 있는 또 다른 선택지로서의 BBRv3 활용 가능성을 확인한다.

1. 서론

Google에 의해 제안된 BBR(Bottleneck Bandwidth and RTT) 혼잡제어 알고리즘(CCA)은 Cubic, Reno와 같은 기존 손실기반 CCA들이 패킷 손실을 혼잡의 시그널로 받아들여 혼잡 윈도우의 크기를 조정하는 방식을 따르지 않고 Network path 모델에 Bottleneck Bandwidth, RTT 등의 기본 정보를 측정값으로 받아들여 pacing rate, quantum, cwnd 등의 제어 파라미터를 기반으로 전송 메커니즘을 제공한다. 이러한 방식은 기존 손실기반 CCA들의 문제로 지적된 이용 가능한 네트워크 대역폭의 소극적 이용 및 스위치, 라우터 등의 네트워크 인터페이스의 큐 지연에 따른 bufferbloat 등의 문제를 해결한다.

BBR은 2016년에 BBRv1이 발표되었고 2019년 BBRv2 alpha 발표에 이어, 최근 2023년 7월 BBRv3가 공개되어 특정 커널 버전에서 이용이 가능한 상태이다 [1]. BBRv2 alpha는 BBRv1의 단점들인 공유 대역에서의 기존 CCA들과의 fairness, RTT가 다른 BBR 플로우들에 대한 공평성 문제, shallow 버퍼에서의 높은 재전송률 등의 문제들을 개선하였

고, BBRv3는 이러한 BBRv2 alpha의 개선 사항을 포함하여 Bandwidth proving 과정에 참여하는 플로우들의 지연된 Bandwidth convergence 현상 등에 대해 주요 파라미터들의 조정 및 Startup 단계에서의 성능 튜닝 등을 추구한다 [2][3].

본 논문에서는 최근 발표된 BBRv3를 Mininet 기반의 에뮬레이션 테스트베드 환경을 구성하고, 원거리 고성능 전송을 요구하는 네트워크 환경에서 전송 성능에 중요한 영향을 미치는 성능 메트릭인 패킷 손실률, RTT에 변화를 주면서 BBRv3 전송 처리율을 측정하고, 기존의 손실기반 CCA인 Cubic과의 처리율을 비교한다.

2. 본론

고에너지물리, 천문학, 기상기후 등의 과학응용들은 대용량 데이터의 공유 및 고속 전송 환경을 필요로 하고 이러한 요구 사항은 ScienceDMZ라는 고성능 원거리 네트워크 환경에서의 friction-free network 환경을 설계하고 제공하기 위한 노력으로 이어져 왔고, Mathis's equation에 따르면 이러한 망 환경에서의 전송 성능은 패킷 손실률, RTT, MSS(Maximum Segment Size)와 같은 성능 메트릭

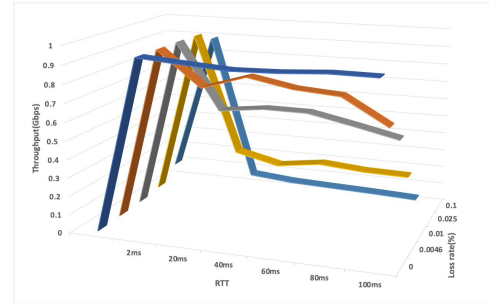
들이 전송 성능에 크게 영향을 주는 요소로 작용한다 [4]. 이러한 ScienceDMZ 환경에서는 여전히 손실기반 CCA들인 Cubic, Htcp와 같은 혼잡제어 알고리즘들이 적용되어 활용되고 있다. 물론 ScienceDMZ에서 추구하는 기본 철학은 패킷 손실이 거의 존재하지 않는 이상적인 friction-free network 환경을 추구하고 있지만, 이러한 가정은 종단간 경로 구성, 종단 시스템의 네트워크 스택 튜닝 등을 포함한 많은 엔지니어링 과정을 필요로 한다. 이러한 배경에서 최대 대역폭, 최소 RTT, 낮은 패킷 손실률 등을 추구하는 BBR의 장점은 기존 손실기반 CCA들을 대체할 수 있는 옵션이 될 수 있는지 파악하고자 한다.

본 실험에서는 BBRv3과 Cubic의 전송 성능을 측정하기 위해 가상머신 Mininet 기반의 에뮬레이션 테스트베드를 구성하고 TCP 송신자와 수신자를 포함하는 Dumbbell 토폴로지로 생성한다. 이용된 VM은 16 Xeon W-3235 cores@3.3GHz, 32GB RAM으로 구성되고 22.04 버전의 Ubuntu, 6.4.0+v3+7542cc7c41c0+GCE 버전의 BBRv3 커널 버전 및 Mininet(ver.2.3.0)이 적용된다. 비교 대상이 되는 CCA들은 BBRv3, Cubic이고, iperf3를 이용하여 60s 동안 전송 트래픽을 종단간에 전송하는 동일한 실험을 각 case 별로 10회 반복하여 평균 처리율을 산출한다.

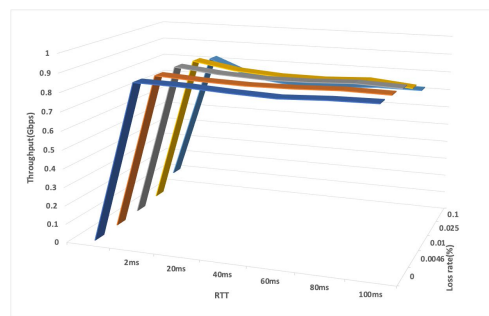
Dumbbell 토폴로지에서 송신측 스위치의 오른쪽 인터페이스에 TBF(Token Bucket Filter)를 이용하여 1Gbps의 병목 지점을 생성하고 network buffer는 1BDP(Bandwidth Delay Product)로 설정한다. 추가적으로 NetEm을 이용하여 RTT, 패킷 손실률의 변화를 설정한다. 이러한 변화는 원거리 ScienceDMZ 환경을 고려하여 RTT를 {2, 20, 40, 60, 80, 100}ms로 설정하고, 패킷 손실률은 {0, 0.0046, 0.01, 0.025, 0.1}%를 적용한 실험 케이스들을 생성한다.

실험 결과 아래 그림에서 볼 수 있듯이 BBRv3의 처리율은 손실기반 CCA인 Cubic의 처리율에 비해 변화하는 RTT 및 패킷 손실률에 덜 민감하게 반응한다. 모든 실험 케이스들에 대해 BBRv3는 대역폭 대비 70%이상의 안정적인 처리율을 보이는 반면 Cubic은 낮은 패킷 손실률 또는 낮은 RTT로 조합된 몇몇 케이스들을 제외하고 현저한 처리율 저하를 가져오는 것으로 확인된다. 이러한 부분은 현재 과학응용을 위한 고성능 네트워크 환경을 제공하기

위해 리눅스 Defalut CCA인 Cubic을 이용할 때 저지연, 낮은 패킷 손실률의 고품질 네트워크 환경을 추구해야 하는 부분과도 맥을 같이 하고, BBRv3는 이러한 제약을 우회할 수 있는 선택지가 될 수 있을 것으로 보인다.



(그림 1) RTT, 패킷 손실률에 따른 Throughput(Cubic)



(그림 2) RTT, 패킷 손실률에 따른 Throughput(BBRv3)

3. 결론

본 논문에서는 Google의 BBRv3의 효용성을 판단하기 위해 Mininet 에뮬레이션 테스트베드를 가상머신에 구성하고 RTT, 패킷 손실률 변화에 따른 BBRv3의 처리율을 측정하고 현재 많이 활용되는 손실기반 CCA인 Cubic의 처리율과 비교하였다. 실험결과 BBRv3의 전송 성능은 Cubic에 비해 여러 범위의 RTT 및 패킷 손실률에 대해 안정적인 결과를 산출하였다. 이러한 결과는 과학응용을 위한 고성능 네트워크 환경을 제공하는 분야에서도 활용가능한 선택지가 될 수 있음을 보이지만 BBRv3의 공격적인 네트워크 대역폭 이용을 위한 접근법은 기존 Cubic과 같은 손실기반 CCA들과의 공평성 문제를 여전히 야기할 수 있으므로, production network 환경에서의 적용을 위한 추가적인 연구가 필요하다.

참고문헌

[1] BBRv3 [Internet]. Available : <https://github.com/google/bbr/blob/v3/README.md>
 [2] N. Cardwell et al., "BBRv2: A model-based congestion control," Proceedings of IETF 105th

meeting, 2019.

[3] N. Cardwell et al., "BBRv3: Algorithm Bug Fixes and Public Internet Deployment," Proceedings of IETF 117th meeting, 2023.

[4] E. Dart et al., "The Science DMZ: A Network Design Pattern for Data-Intensive Science", Proceedings of IEEE/ACM Annual SuperComputing Conference (SC13), Denver, USA, Nov. 2013.