

BBRv2 is a model-based congestion control algorithm that aims for low queues, low loss, and (bounded) Reno/CUBIC coexistence. To maintain a model of the network path, it uses measurements of bandwidth and RTT, as well as (if they occur) packet loss and/or DCTCP/L4S-style ECN signals. Note that although it can use ECN or loss signals explicitly, it does not require either; it can bound its in-flight data based on its estimate of the BDP.

BBRv2는 모델 기반의 정체 제어 알고리즘으로, 낮은 대기열, 낮은 손실, (경계) Reno/CUB 공존을 목표로 한다. 네트워크 경로 모델을 유지하기 위해 대역폭과 RTT의 측정뿐만 아니라 (발생하는 경우) 패킷 손실 및/또는 DCTCP/L4S 스타일의 ECN 신호도 사용한다. ECN 또는 손실 신호를 명시적으로 사용할 수 있지만 둘 중 하나를 요구하지는 않는다. BDP의 추정치에 근거하여 비행 중 데이터를 바인딩할 수 있다.

The model has both higher and lower bounds for the operating range:

lo: bw\_lo, inflight\_lo: conservative short-term lower bound

hi: bw\_hi, inflight\_hi: robust long-term upper bound

The bandwidth-probing time scale is (a) extended dynamically based on estimated BDP to improve coexistence with Reno/CUBIC; (b) bounded by an interactive wall-clock time-scale to be more scalable and responsive than Reno and CUBIC.

이 모델은 운용 범위의 상한과 하한 모두를 가지고 있다:

lo: bw\_lo, inflight\_lo: 보수적인 단기 하한

hi: bw\_hi, inflight\_hi: 강력한 장기 상한

대역폭 확장 시간 척도는 (a) Reno/CUBIC와의 공존을 개선하기 위해 추정된 BDP를 기반으로 동적으로 확장되며, (b) Reno 및 CUBIC보다 확장성과 응답성이 더 높은 대화형 벽시계 시간 척도로 제한된다.

**Elapsed real time**, **real time**, **wall-clock time**, or **wall time** is the actual time taken from the start of a [computer program](https://en.wikipedia.org/wiki/Computer_program) to the end.

**BBR v2 takes packet loss into its control logic**

We bound the BBR-native inter-bw-probe wall clock time to be:

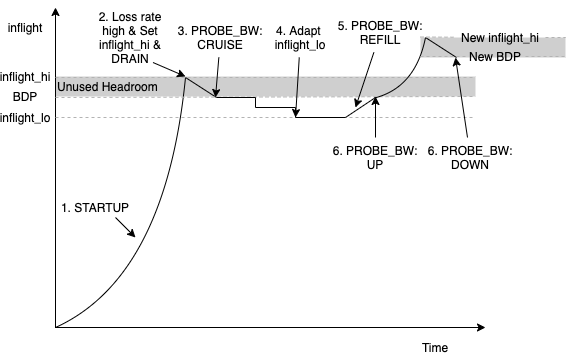
(a) higher than 2 sec: to try to avoid causing loss for a long enough time to allow Reno at 30ms to get 4K video bw, the inter-bw-probe time must be at least: 25Mbps .030sec / (1514bytes) 0.030sec = 1.9secs

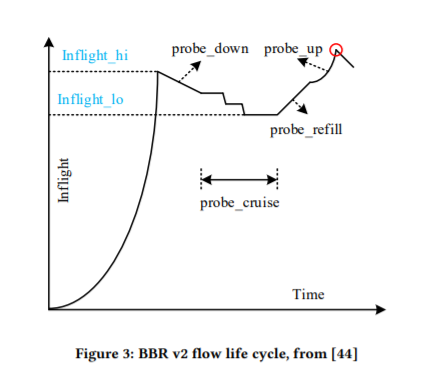
(b) lower than 3 sec: to ensure flows can start probing in a reasonable amount of time to discover unutilized bw on human-scale interactive time-scales (e.g. perhaps traffic from a web page download that we were competing with is now complete).

BBR-native inter-bw-probe wall clock time to be:

(a) 2초 이상: 30ms에 Reno가 4K 비디오 bw를 얻을 수 있을 정도로 장시간 손실을 방지하려면 bw-probe 간 시간이 최소 25Mbps .030sec/(1514bytes) 0.030s = 1.9s여야 함

(b) 3초 미만: 흐름이 인간 규모의 대화형 시간 추적에서 미활용 bw를 발견하기 위한 합리적인 시간 내에 탐색을 시작할 수 있도록 보장하기 위해(예: 우리가 경쟁하고 있던 웹 페이지 다운로드의 트래픽이 현재 완료되었을 수 있음)



  
BBR flow will send packets at the full estimated bandwidth and packet loss is not exploited to indicate link congestion. If the queue length at the bottleneck is smaller than 1.5 ∗ *BDP* [44], multiple BBR flows will cause high packet loss.

Reno/CUBIC flows gain low throughput when share bottleneck with BBR flows. BBR v2 [44] is proposed to solve these issues in BBR v1. BBR v2 is claimed that it can make better coexistence with Reno/CUBIC flows and achieves low queue delay.

BBR 흐름은 완전한 예상 대역폭에서 패킷을 전송하며 패킷 손실은 링크 정체를 나타내기 위해 이용되지 않는다. 병목현상의 대기열 길이가 1.5 BDP[44]보다 작을 경우, 복수의 BBR 흐름이 높은 패킷 손실을 유발한다. Reno/CUBIC 흐름은 BBR 흐름의 공유 병목 현상으로 인해 처리량이 낮아진다. BBR v2[44]는 BBR v1에서 이러한 문제를 해결하기 위해 제안되었다. BBR v2는 Reno/CUBIC 흐름과 더 잘 공존하고 낮은 대기열 지연을 달성할 수 있다고 주장되었다.

BBR v2 takes **packet loss** into its control logic. The life cycle of BBR v2 flow is shown in Figure 3. When the estimated bandwidth does not exceed the target for 3 times, the sender assumes reaching the full bandwidth in BBR v1. Besides that, condition on packet loss is added to exist from StartUp to Drain: **the number of loss packet in a round exceeds 8 and packet loss rate exceeds *loss threshold* (0.02)**. Such condition is applied to avoid excessive packet loss.

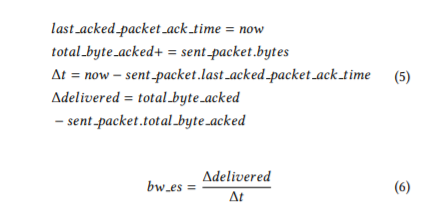
BBR v2는 패킷 손실을 제어 논리에 반영한다. BBR v2 흐름의 수명주기는 그림 3과 같다. 예상 대역폭이 3회 동안 목표를 초과하지 않을 때, 송신자는 BBR v1의 전체 대역폭에 도달한다고 가정한다. 그 외에도, 패킷 손실의 조건은 StartUp에서 Drain까지 존재하도록 추가된다: 라운드에서 손실 패킷의 수는 8을 초과하고 패킷 손실률은 손실 임계값(0.02)을 초과한다. 그러한 조건은 과도한 패킷 손실을 피하기 위해 적용된다.

**INFLIGHT HI**

**INFLIGHT LO**

**If the condition on packet loss holds true, the calculated BDP is assigned to inflight hi**. When a new acknowledge packet arrives, **inflight lo is updated as Equation (8)**. Here, ∆delivered is 4 calculated in the same way in Equation (5) and kBeta is 0.3

패킷 손실에 대한 조건이 참이면 계산된 BDP는 inflight hi에 할당된다. 새로운 승인 패킷이 도착하면, inflight lo는 방정식 (8)로 업데이트된다. 여기서 ∆delivered는 방정식 (5)에서 같은 방법으로 계산하고 kBeta는 0.3이다.





In ProbeBW state, the working mechanism of BBR v2 is quite different from BBR v1. The phases (probe up, probe down, probe cruise) switching is no longer depended on the time interval *RTTmin*. *cwnd* is also not set as 2 ∗ *BDP* and is related with *infliдht lo* and *infliдht hi*. *infliдht hi* is updated if the inflight packets are too high (inflight too high), in which the loss packet rate exceeds *loss threshold* in last round. In the probe down phase, the*pacinд дain* is 0.75, the phase will be switched to probe cruise if the inflight packets are drained to *BDP* or the condition*infliдht too hiдh* holds true. The *cwnd* in probe cruise is calculated by Equation (9). *kHeadRoom* is 0.15. *infliдht hi* indicates the channel is in dangerous area. To leave headroom to *cwnd* is to alleviate link congestion to some extent. The *interval* for probe cruise is randomized from 2 seconds to 3 seconds. If duration in probe cruise phase exceeds the *interval*, a probe refill phase is applied as shown in Figure 3. *cwnd* is set as *infliдht hi* to increase the inflight packets in a round. The probe refill is to make preparation for probe up.