BBRv3: Algorithm Bug Fixes and Public Internet Deployment

Google TCP BBR team: Neal Cardwell, Yuchung Cheng, Kevin Yang, David Morley

Soheil Hassas Yeganeh, Priyaranjan Jha, Yousuk Seung

Van Jacobson

Google QUIC BBR team: Ian Swett, Bin Wu, Victor Vasiliev

https://groups.google.com/d/forum/bbr-dev

Outline

- BBR algorithm updates
- BBR deployment status at Google
- BBR code status and open source release plans

Target for this talk:

- Responding to requests from other transport stack maintainers implementing BBR
- Documenting the BBR algorithm
- Announcing a new open source release of Linux TCP BBR
- Inviting the community to…
  - Read the drafts and offer editorial feedback
  - Share algorithm or code fixes or enhancements
  - Share test results
  - Post bug reports

BBR Versions

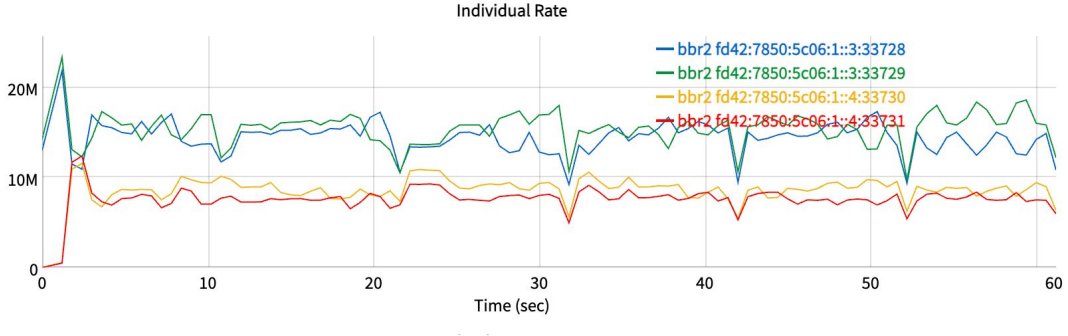
- New: a version increment for BBR, from v2 to v3
  - Recent bug fixes change the bandwidth/fairness convergence properties of BBR
  - So most test results for BBRv2 will not apply to BBRv3
  - So we are incrementing the version number for BBR from v2 to v3
- A summary of BBR versions:
  - **BBRv1: [obsolete/deprecated]**
    - Bandwidth, RTT as signals primary signals; loss used over short time scales
  - **BBRv2: [obsolete/deprecated]**
    - BBRv1 + using ECN, loss as signals (with bugs mentioned previously at IETF**)**
  - **BBRv3**:
    - BBRv2 + bug fixes and performance tuning
  - **BBR.Swift**: [discussed at IETF 109: slide link]
    - BBRv3 + using network_RTT (excluding receiver delay) as primary CC signal

BBRv3 bug fix 1: fix bw convergence *with* loss/ECN

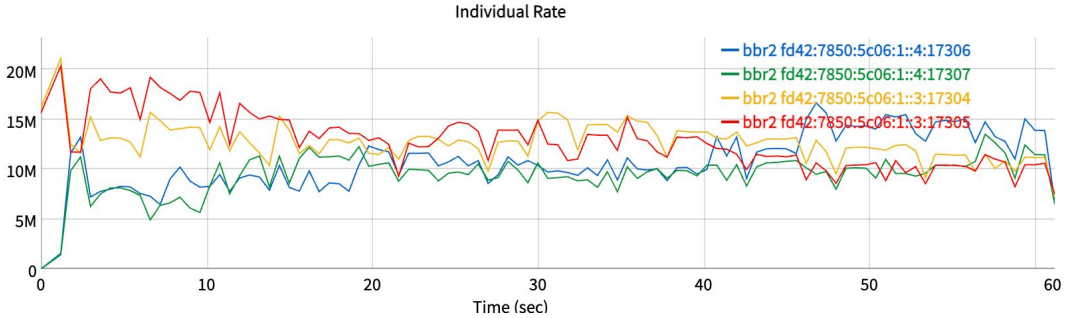
- Bug in BBRv2:
  - Symptoms: after loss/ECN set inflight_hi, later bandwidth probing stopped early
    - Before the flow caused loss/ECN signals again or reached fair share
  - Root cause: circular dependence between max bandwidth, max in-flight data
  - Impact: Caused BBRv2 flows to...
    - Not reach fair share competing vs BBR or Reno/CUBIC
    - Take a long time to reach full utilization when congestion subsides
- Fix:
  - Keep probing for bw until either:
    - Loss rate or ECN mark rate exceed tolerance thresholds, OR
    - Inflight_hi has not limited sending recently and bandwidth saturates
      - Bandwidth saturation estimator is same as in STARTUP mode
      - Estimate bw is saturated if >= 3 round trips w/o bw increasing by 25%
  - Reduces CUBIC/Reno share when competing with >1 BBR flow

BBRv3 bug fix 1: fix bw convergence *with* loss/ECN

Before bug fix 1:



After bug fix 1:



Example test results from:

transperf bulk TCP transfer test with 4 TCP BBRv3 flows with

bottleneck_bw=50Mbps, min_rtt=40ms, **buffer=1*BDP**

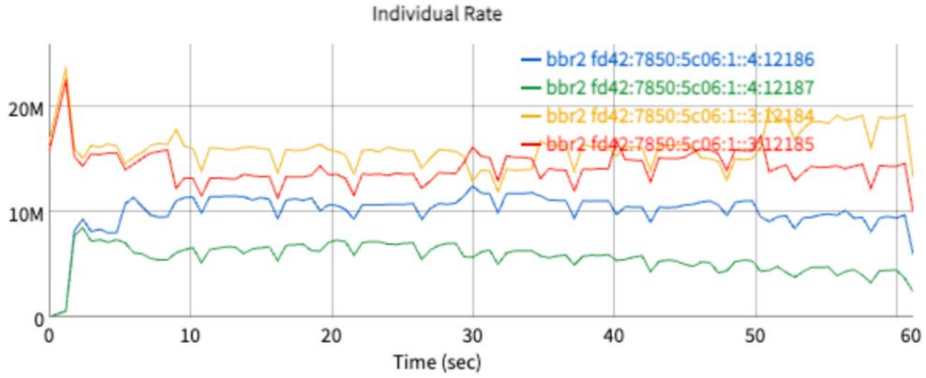
(at t=0s flows 0, 1 start; at t=1s flows 2, 3 start)

BBRv3 bug fix 2: fix bw convergence *without* loss/ECN

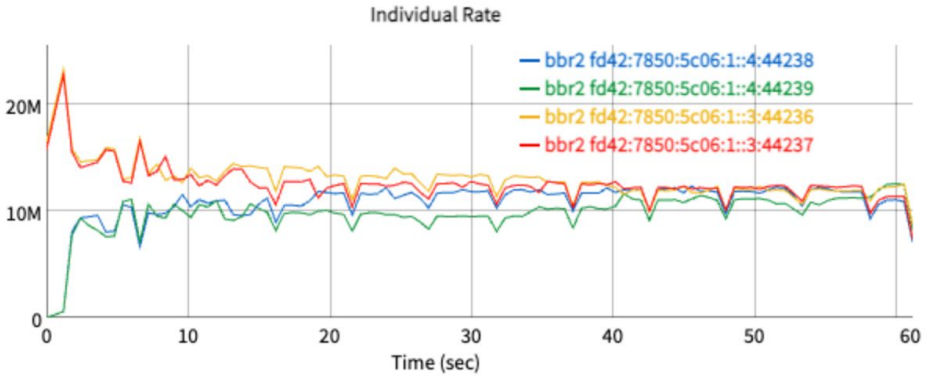
- Bug in BBRv2:
  - Symptoms: in buffer >1.5*BDP, BBRv2 flows often did not converge to fair share
  - Root causes:
    - 1: Fixed cwnd gain could prevent slow flows from raising their sending rate
    - 2: Slow flows holding 0.75*estimated_bw yielded too much bw to fast flows
  - Impact: lack of intra-protocol fairness for BBRv2 w/ buf>1.5*BDP, w/o ECN or loss
- Fixes:
  - 1: Increase cwnd gain from 2.0 to 2.25 when probing for bandwidth (ProbeBW_UP)
    - To ensure sending rate can increase when probing for bandwidth
  - 2: Change pacing gain of 0.75x to 0.9x (ProbeBW_DOWN)
    - 0.9x is derived from on the ProbeBW_UP pacing gain of 1.25x...
    - ...as the minimum pacing gain value that allows slow flows to consistently utilize enough bw to cause fast flows to yield bw for fairness convergence

BBRv3 bug fix 2: fix bw convergence *without* loss/ECN

Before bug fix 2:



After bug fix 2:



Example test results from:

transperf bulk TCP transfer test with 4 TCP BBRv3 flows with

bottleneck_bw=50Mbps, min_rtt=40ms, **buffer=100*BDP**

(at t=0s flows 0, 1 start; at t=1s flows 2, 3 start)

BBRv3 performance tuning

- Performance tuning changes:
  - STARTUP cwnd gain: 2.89 => 2.0 [analytic derivation]
  - STARTUP pacing gain: 2.89 => 2.77 [analytic derivation]
  - When exiting STARTUP, set inflight_hi based on:
    - max(estimated BDP, max number of packets delivered in last round trip)
  - To trigger exit of STARTUP based on packet loss...
    - Require fewer loss events in a single round trip (6 rather than 8)
- Primary impact of these changes:
  - Lower queuing delays and packet loss rates during and shortly after STARTUP

BBR deployment status at Google

- Google-internal traffic:
  - **BBRv3** is TCP congestion control for all internal **WAN traffic**
  - **BBR.Swift** is TCP congestion control used **within a datacenter**
- Google-external traffic:
  - **BBRv3** is TCP CC for all Google.com public Internet traffic
  - A/B experiments: BBRv3 vs v1 for small % of users for:
    - TCP for YouTube
    - QUIC for google.com and YouTube

BBRv3 performance impact for public Internet traffic

- Impact of BBRv3 vs BBRv1 on Google.com and YouTube TCP public Internet traffic:
  - Lower retransmit rate (12% reduction)
  - Slight latency improvement (0.2% reduction) for:
    - Google.com web search
    - Starting YouTube video playback
  - Latency wins seem to be from lower loss rate (less/faster loss recovery)

BBR Open Source Code

- TCP BBRv3 release:
  - Linux TCP (dual GPLv2/BSD): github.com/google/bbr/blob/v3/README.md
  - Main updates: the bug fixes described in this presentation
  - TCP BBR v3 release is open source (dual GPL/BSD), available for review/testing
  - Plan to email patches to propose inclusion in mainline Linux TCP in August
- BBRv1 code in Linux TCP "bbr" module will be upgraded to BBRv3
- Why upgrade BBRv1->BBRv3 in place rather than a separate module? BBRv3 has...
  - Better coexistence with Reno/CUBIC, vs v1
  - Lower loss rates, vs v1
  - Lower latency for short web requests (from google.com, YouTube data), vs v1
  - Throughput similar to v1 (within 1% of v1 on YouTube)

Conclusion

- Summary:
  - Open sourced BBRv3 on github with significant bug fixes vs BBRv2
  - BBRv3 used for all TCP for Google.com public Internet and internal WAN traffic
  - BBRv3 under A/B testing for YouTube TCP, YouTube and Google.com QUIC
- Next:
  - Plan on submitting BBRv3 for inclusion in mainline Linux TCP in August
  - Will update BBR Internet Drafts to cover BBRv3:
    - Delivery rate estimation: draft-cheng-iccrg-delivery-rate-estimation
    - BBR Congestion control: draft-cardwell-iccrg-bbr-congestion-control
- We invite the community to share…
  - Feedback on the algorithm, code, or drafts
  - Test results, issues, patches, or ideas
- Thanks!

Q & A

https://groups.google.com/d/forum/bbr-dev

Internet Drafts, paper, code, mailing list, talks, etc.

Special thanks to Eric Dumazet, Nandita Dukkipati, Matt Mathis, Luke Hsiao, C. Stephen Gunn, Jana Iyengar, Pawel Jurczyk, Biren Roy, David Wetherall, Amin Vahdat, Leonidas Kontothanassis, and {YouTube, google.com, SRE, BWE} teams.