BBR Congestion Control Draft

draft-ietf-ccwg-bbr-02

Internet Draft Editors:

Neal Cardwell (Google), Ian Swett (Google), Joseph Beshay (Meta)

Speaker: Ian Swett

Outline

Overview

- Outline recent BBR Internet Draft updates
- Summarize open pull requests and open issues

Goals for this talk:

- Provide a road map for...
  - Readers of the draft
  - Implementers of BBR reading the draft
  - Members of the CCWG/ICCRG community who would like to contribute
- Inviting the community to…
  - Read the draft
  - Contribute to the draft

Overview of draft-ietf-ccwg-bbr

- BBR was adopted a CCWG WG item in October 2024
- Intended status: experimental RFC
- IETF CCWG members are collaborating on github:
  - https://github.com/ietf-wg-ccwg/draft-ietf-ccwg-bbr
- Latest published revision is at:
  - https://datatracker.ietf.org/doc/draft-ietf-ccwg-bbr/
- Latest editor's copy (with hot-off-the-press changes not in published revisions above):
  - https://ietf-wg-ccwg.github.io/draft-ietf-ccwg-bbr/draft-ietf-ccwg-bbr.html
- Draft editors:
  - Neal Cardwell (Google)
  - Ian Swett (Google)
  - Joseph Beshay (Meta)

Changes in draft-ietf-ccwg-bbr-02: summary

- Previous version was: draft-ietf-ccwg-bbr-01, discussed at IETF 121 in Nov, 2024 [slides]
- Diffs: [text diff between draft-ietf-ccwg-bbr-01 and draft-ietf-ccwg-bbr-02]
- Main changes between draft-ietf-ccwg-bbr-01 and draft-ietf-ccwg-bbr-02:
  - Add missing BBRIsProbingBW() pseudocode #10
  - Ensure BBRCheckFullBWReached() pseudocode counts three full rounds #16
  - Remove some TCP and SACK references #15
  - Rename *inflight_lo* and *inflight_hi* to *inflight_shortterm* and *inflight_longterm* #11
  - Define "acknowledged" and "delivered" in a transport-agnostic way #21
  - … a number of other editorial improvements

Goals of evolving the BBR draft text

Goals as we evolve the BBR draft text:

- Clarification
- Simplification
- Better coexistence with Reno/CUBIC
- Better performance
- Avoiding performance regressions in the real world

Proposed bar for publication (keep in mind the target is an experimental RFC):

- Multiple deployments at scale in QUIC and TCP
- Text both TCP and QUIC implementations can follow
- Fair sharing with other BBR flows, coexistence with Reno and Cubic

Thesis: It's better to publish a good draft with deployment experience in a reasonable timeframe than evolve BBR indefinitely without shipping an RFC.

2 Open Pull Requests: minor algorithm changes

2 Open PRs; both are algorithm changes waiting for performance data from experiments:

- #6 Use consistent value for drain pacing gain which matches derivation doc
  - Proposes changing BBR DRAIN gain from 1/2.89 = 0.35 to 1/2 = 0.5
  - To match analytical derivation of DRAIN pacing gain, which derives 1/2 = 0.5
  - Have some old A/B experiment data from Linux TCP YouTube experiments:
    - Unclear if there are statistically significant performance regressions
    - Planning to re-run an experiment to ensure there are no regressions
  - We'd appreciate performance data comparing drain_gain vals from anyone who's able
- #5 Remove BBR.ack_phase from pseudocode
  - A minor algorithm simplification
  - Has one implementation (mvfst QUIC BBR2)
  - We'd like
    - A second implementation
    - Internet performance data to verify there is no performance regression

Issue: generalization to non-TCP transports

Open issue:

- Bandwidth estimation still uses _seq names, which are TCP-centric

The intent is to make the draft as transport agnostic as possible.
We are making progress but not done.

Goal: Ensure implementation of BBR across as many transports as possible

Non-Goal: Create universal approach for mapping any congestion control to any transport

Issue: ProbeRTT every 5 sec or 10 sec?

Open issue:

- Impact of increasing ProbeRTT interval to 10s?

Details:

- BBRv1 used a 10s ProbeRTT interval
  - This 10s value has many "air miles"
- BBRv2/BBRv3 currently use: "ProbeRTTInterval = 5 seconds"
  - Rationale: since BBRv2/BBRv3 ProbeRTT reduce cwnd to 0.5*estimated_BDP instead of 4 packets, they can achieve the same throughput as BBRv1 with 2x more frequent ProbeRTT
- Meta mvfst QUIC BBRv2 found:
  - Increasing ProbeRTT interval to 10s in internet facing connections improves application performance in initial experiments (vs 5s)
    - No increase in rtt estimates
    - No increase in retransmissions
    - Reduces p99 response time for requests by up to 15% for app traffic on busy links

Issue: "inflight" or "in-flight data" or "data in flight"

Open issue:

- "inflight" or "in-flight data" or "data in flight"

- All 3 phrases are used in the draft today
  - RFC 9002 (QUIC) almost always uses ‘in flight’
- We should probably standardize on one
- Purely editorial

Issue: Do we need inflight_shortterm?

Open issue:

- Do we need inflight_shortterm?

Propose resolving this with "Yes"; rationale (here):

- When available bandwidth is reduced (e.g., due to more flows competing at the bottleneck or a reduced bottleneck link rate, e.g. radio issues) this reduces not just the bandwidth available but also the flow's fair share of the path BDP and bottleneck buffer.
- Thus, to avoid excess queuing and loss, the flow needs to reduce its pacing rate so that matches the lower available bandwidth (via bw_shortterm), and reduce its cwnd to match the lower share of the path BDP and bottleneck buffer slots (via inflight_shortterm).

Notes:

- If folks have simulation tests, lab tests, or production data showing that good results are achievable w/o inflight_lo (inflight_shortterm) or equivalent, let's re-open

Issue: decide/document draft's stance on ECN

Open issue:

- Decide and document the BBR draft's stance on ECN

Discussion:

- Do we want to document a recommended response to ECN in the draft?
- No strong opinion on the details as long as it performs well in real-world testing
- Not aware of large-scale real-world test results with BBR w/ ECN over public Internet
- Waiting to implement and test an ECN response for BBR may hold up the draft considerably

Conclusion

- Inviting the community to...
  - Read the draft: draft-ietf-ccwg-bbr
  - Offer contributions/comments/edits, in whatever manner you prefer
- Thanks!

Details/Backups...

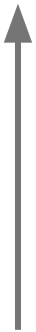
Changes in draft-ietf-ccwg-bbr: how to view

- To see recent changes, you can use several approaches, depending on your preference:
  - From the command line:
    - git clone https://github.com/ietf-wg-ccwg/draft-ietf-ccwg-bbr.git
    - cd draft-ietf-ccwg-bbr/
    - git log -p
  - From github:
    - Commits
    - Merged pull requests

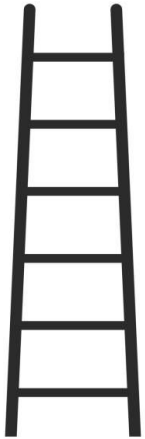
Thoughts about ways to contribute

- Contributions at any "rung of the ladders" below are welcome!
- The higher on the "ladders" (the more concrete/specific/tested the contribution is)...
  - The more useful to the BBR draft effort
  - Given editor time constraints, the more likely the eventual inclusion in the draft
- To finalize significant algorithm changes, we'd like to ultimately reach the top rung of the ladder
- Collaboration encouraged: e.g., idea from person A, implemented by person B, tested by sites B/C

Editorial changes:



- Github pull request with draft text
- Github issue describing the idea
- CCWG email/meeting suggestion



Technical algorithm changes:

- Multiple at-scale Internet deployments
- At-scale Internet deployment data
- Lab/simulation experiment results
- Patch to an open-source BBR code base
- Github pull request with draft text
- Pseudocode
- Github issue describing the idea
- CCWG email/meeting suggestion