BBR Congestion Control Draft

draft-ietf-ccwg-bbr-04

Internet Draft Editors:

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

Speaker: Ian Swett

CCWG at IETF 124: Montreal

Nov 2025

Outline

Overview

- Outline recent BBR Internet Draft updates
- Summarize open issues (6) and open pull requests (4)

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 as 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-04: summary

Changes between draft-ietf-ccwg-bbr-03 and draft-ietf-ccwg-bbr-04 [text diff]:

- Add experimental considerations for ProbeRTT frequency (#86)
- Avoid TCP-specific terms:
  - write_seq/SND.NXT/SND.UNA (#57)
  - Segment -> Packet (#72, #74)
- Editorial improvements:
  - Clarify the recommended units are bytes and seconds (#85)
  - Fixes to variable names, definitions, pseudocode function parameters (#76, #78, #79, #80, #81, #82, #83, #84, #87); metadata (#70)

Changes since draft-ietf-ccwg-bbr-04:

- Fix how pacing should be considered in the decision that the transport is application-limited (#47)
- Fixes to pseudocode function parameters: Fix arguments to IsNewestPacket() (#89)

Open Issue: Risk getting stuck in DRAIN if estimated bandwidth is sufficiently overestimated

Issue #90, PR #6

- The derivation of the drain pacing gain suggests a value of 0.5 but the draft uses 0.35.
- DRAIN could fail to drain the queue after STARTUP if the bandwidth was sufficiently overestimated (and higher pacing gains increase this risk).

Options:

1. Experiment with a drain gain of 0.5
2. Experiment with removing the **Drain** stage and move to **ProbeDown** after **Startup**
   a. Possibly with a lower gain for the first **ProbeDown**

Open Issue: Test cases section in the draft

Issue #71

Interest in a list of test cases for implementers to use for verification

- Qlog traces in an external repository
- Use of a common tool that can test and trace arbitrary binaries with network activity under certain conditions

Open Issue/PR: generalization to non-TCP transports

Open editorial issues

- Section 5.5.9 sounds very TCP-specific #69
- Bandwidth estimation still uses _seq names, which are TCP-centric #26

An open issue with technical implications:

- What is BBR.offload_budget for QUIC? #67
  - The description of the offloat_budget is specific to TCP. What's a good approach to expand it to accommodate QUIC?
  - Open PR for this: Generalize Offload Budget for QUIC (#88)

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

Summary of other open issues

Editorial questions:

- Definition of variables
  - Per-Packet and Connection State variables are spread across multiple Sections #68

Pending Experiments: 2 Open PRs: minor changes

Two open PRs 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

PR: Delivery rate sampling and restarting from idle

An open PR:

- Clarify Unacknowledged rather than inflight for rate sample #56
  - In Sec `4.1.2.2. Transmitting a data packet` of -03
  - **How to check for a connection "restarting from idle", in a transport/implementation-agnostic way?**
    - e.g.: connection is certain that no data packets are in the network
  - TCP-centric version was: `if (SND.NXT == SND.UNA)  /* no packets in flight yet? */`
  - (C.inflight == 0) is not sufficient, for subtle reasons
    - packets spuriously marked lost and later delivered would have incorrect timestamps that could cause significant bandwidth overestimation
  - Open for discussion; ideas are welcome

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

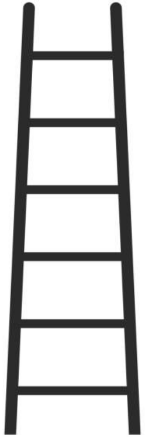
- To see recent changes, you can use several approaches, depending on your preference:
  - From the command line:
    - git clone https://github.com/ietf-wg-ccwg/draft-ietf-ccwg-bbr.git
    - cd draft-ietf-ccwg-bbr/
    - git log -p
  - From github:
    - Commits
    - Merged pull requests

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

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.