BBRv3: Algorithm Overview and Google's 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 high-level overview
- BBR deployment status at Google
- BBR code status and open source release plans

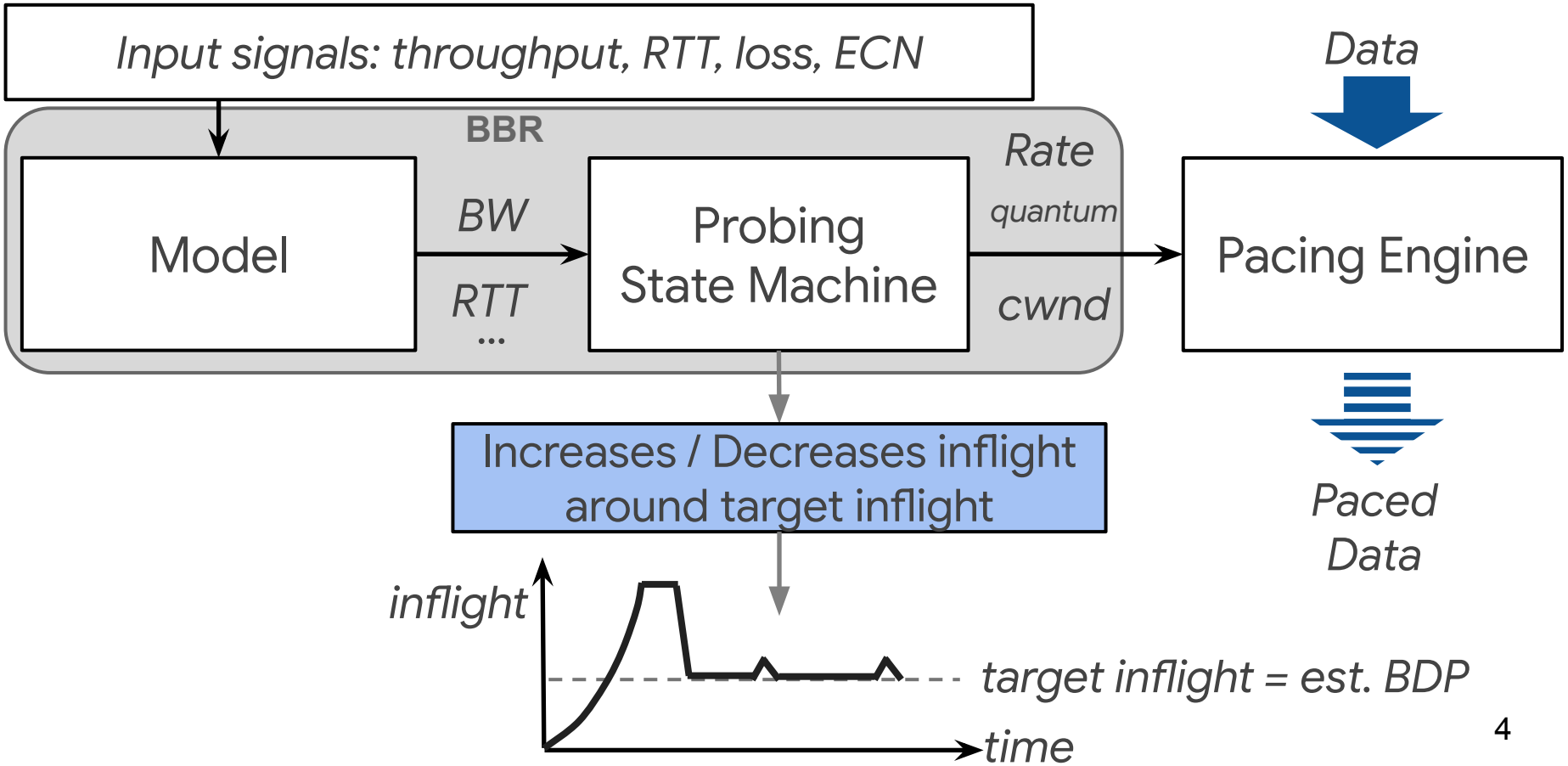
Goals for this talk:

- Responding to requests from CCWG chairs for BBR refresher/overview
- Inviting the community to…
  - Offer feedback on suggested plan, if any, for BBR with respect to IETF drafts/RFCs
  - Read the drafts and offer editorial feedback
  - Share algorithm or code fixes or enhancements
  - Share test results
  - Post bug reports

BBRv3 CC in a nutshell

- Design goals:
  - High throughput with up to a targeted level of random packet loss
  - Bounded queuing delay, even with bloated bottleneck buffers
  - Usable coexistence when sharing Reno and CUBIC congestion control
- Mechanisms:
  - Model-based: dynamically probes and models the network path
    - Models max bandwidth, min RTT, max aggregation, max inflight
  - Signals:
    - Bandwidth, RTT
    - ECN (like DCTCP, L4S)
    - Loss (explicit loss rate cap of 2%)

BBRv3 congestion control: the big picture



BBR versions

- 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
  - **BBRv3:** [discussed at IETF 117: slide link]
    - 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

A quick comparison of CC algorithms

	CUBIC	BBRv1	BBRv3
Model parameters to the state machine	N/A	Max throughput, Min RTT	Max throughput, Min RTT, Max aggregation, Max inflight
Loss	Reduce cwnd by 30% on window with any fast recovery loss	N/A	Explicit loss rate cap (2%)
ECN	RFC3168 (Classic ECN)	N/A	shallow-threshold ECN (like DCTCP/L4S)
Startup	Slow-start until RTT rises (Hystart) or any loss	Slow-start until tput plateaus	Slow-start until tput plateaus or ECN/loss rate > target

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** and [new!] **YouTube** public Internet traffic
  - A/B experiments: BBRv3 vs v1 for small % of users for:
    - QUIC for google.com and YouTube

BBRv3 open source code

- TCP BBRv3 release:
  - Linux TCP BBRv3 is open source (dual GPLv2/BSD), available for review/testing:
    - github.com/google/bbr/blob/v3/README.md
  - Plan to email patches to propose inclusion in mainline Linux TCP ASAP
- 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

- Next:
  - Plan on submitting BBRv3 for inclusion in mainline Linux TCP ASAP
  - Internet Drafts cover BBRv2; plan to update them to cover BBRv3 ASAP:
    - 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
  - Feedback on where to go from here with drafts
- 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.