BBR Congestion Control

Neal Cardwell, Yuchung Cheng,

C. Stephen Gunn, Soheil Hassas Yeganeh,

Van Jacobson

googlegroups.com/d/forum/bbr-dev

Google Networking Research Summit - February 8, 2017

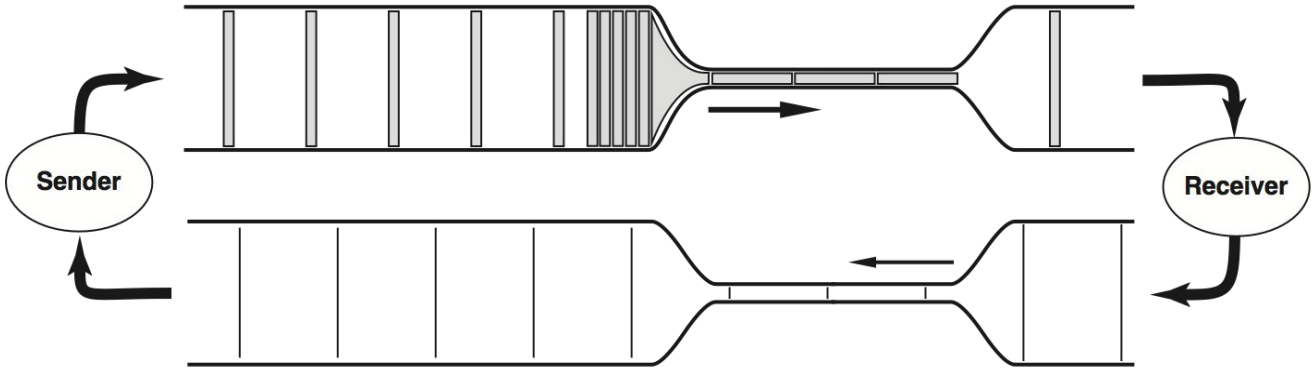
BBR Congestion Control: Overview

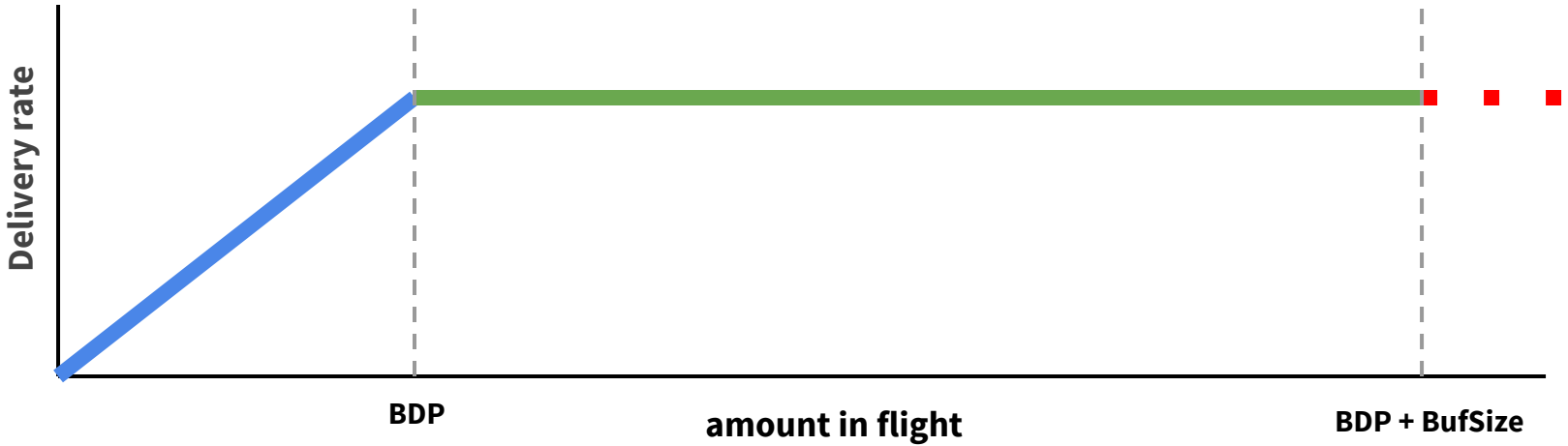
- **BBR** = **B**ottleneck **B**andwidth and **R**ound-trip propagation time
- **BBR seeks high throughput with a small queue by probing BW and RTT**
- Ground-up redesign of congestion control
  - *Not* loss-based, delay-based, ECN-based, AIMD-based
- **Models** the network path: probes and estimates max BW and min RTT
- Result:
  - High throughput even with shallow buffers and moderate loss rates
  - Low delay even with deep buffers ("bufferbloat")
- Used at Google: internal WAN networks; rolling out on google.com, YouTube
- Open source (Linux v4.9 TCP; QUIC; WIP: FreeBSD TCP)
- Incrementally deployable: sender-only upgrade

Motivation

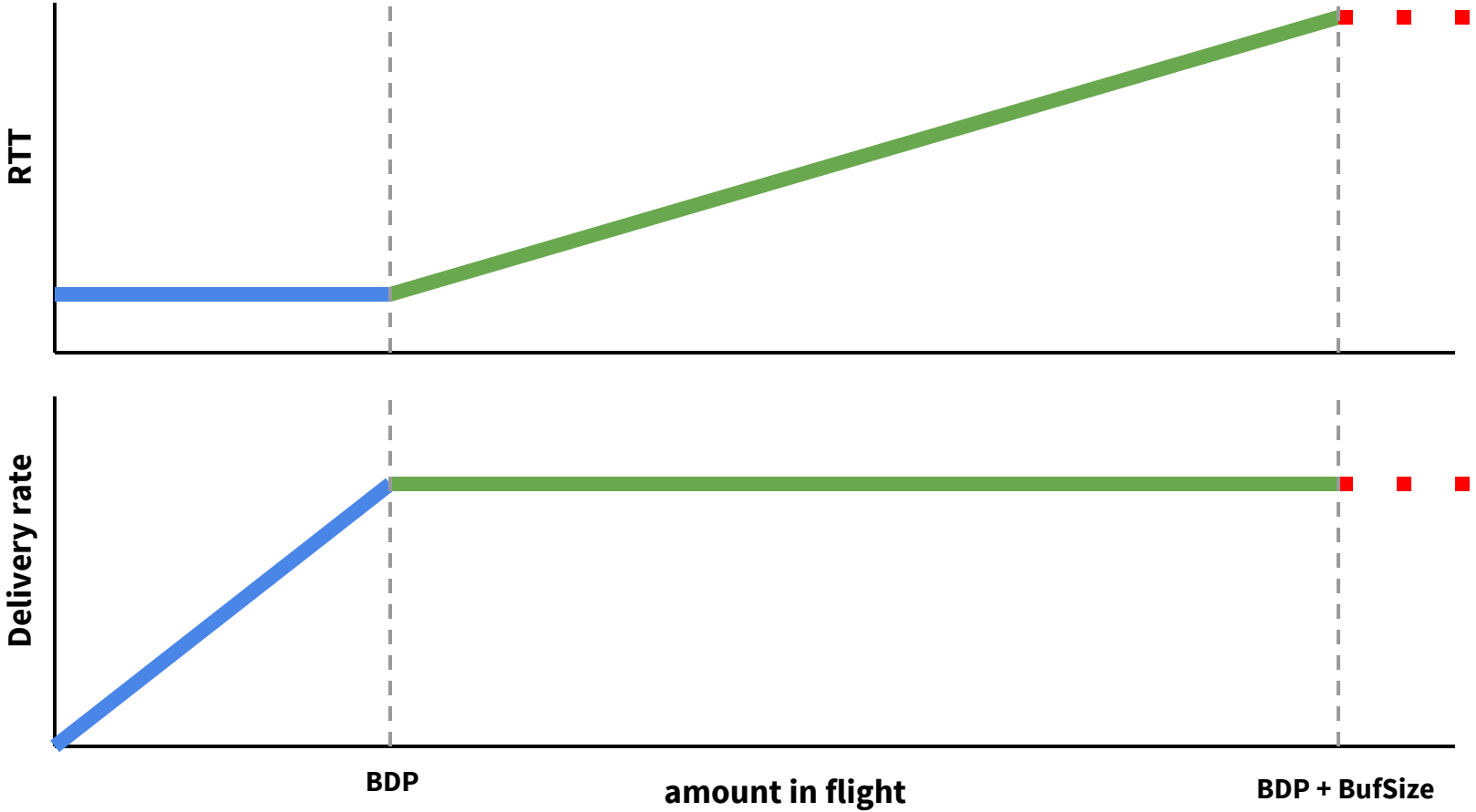
- The problem: Internet performance is often not realizing its potential:
  - Last-mile networks with seconds of latency ("bufferbloat")
  - Gigabit wide area networks need infeasible loss rates to use bandwidth
    - 10Gbit/sec, 100ms RTT needs 1/30M loss (if 1% loss, get .003 Gbit/sec)
- The culprit: loss-based congestion control
  - *"Congestion Avoidance and Control"*, Jacobson & Karels, SIGCOMM 1988
  - Internet standard
- Google's unique vantage point for attacking the problem:
  - Representative traffic to every corner of the Internet
  - Control network stack on the senders

Network congestion and bottlenecks: bandwidth

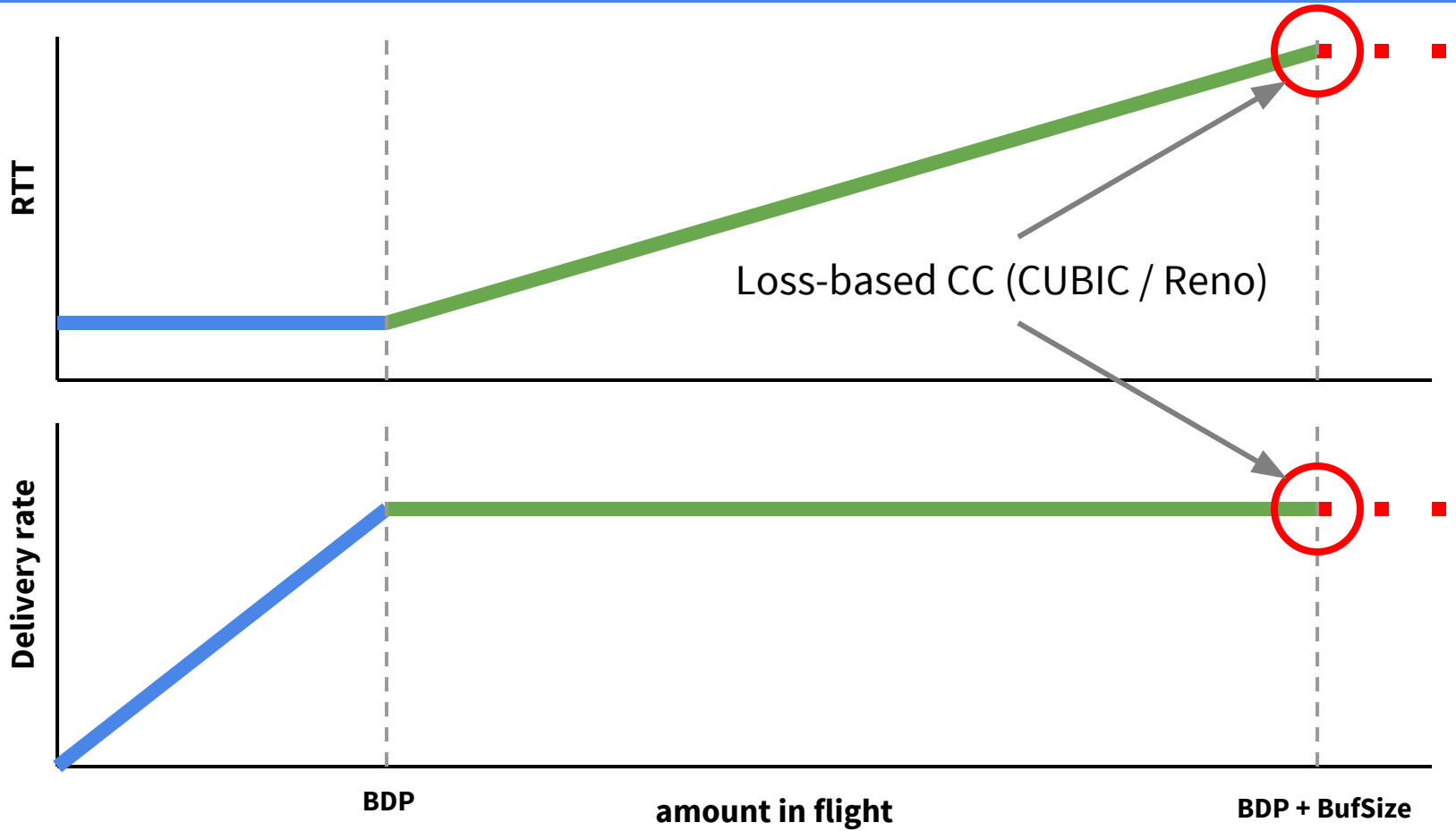




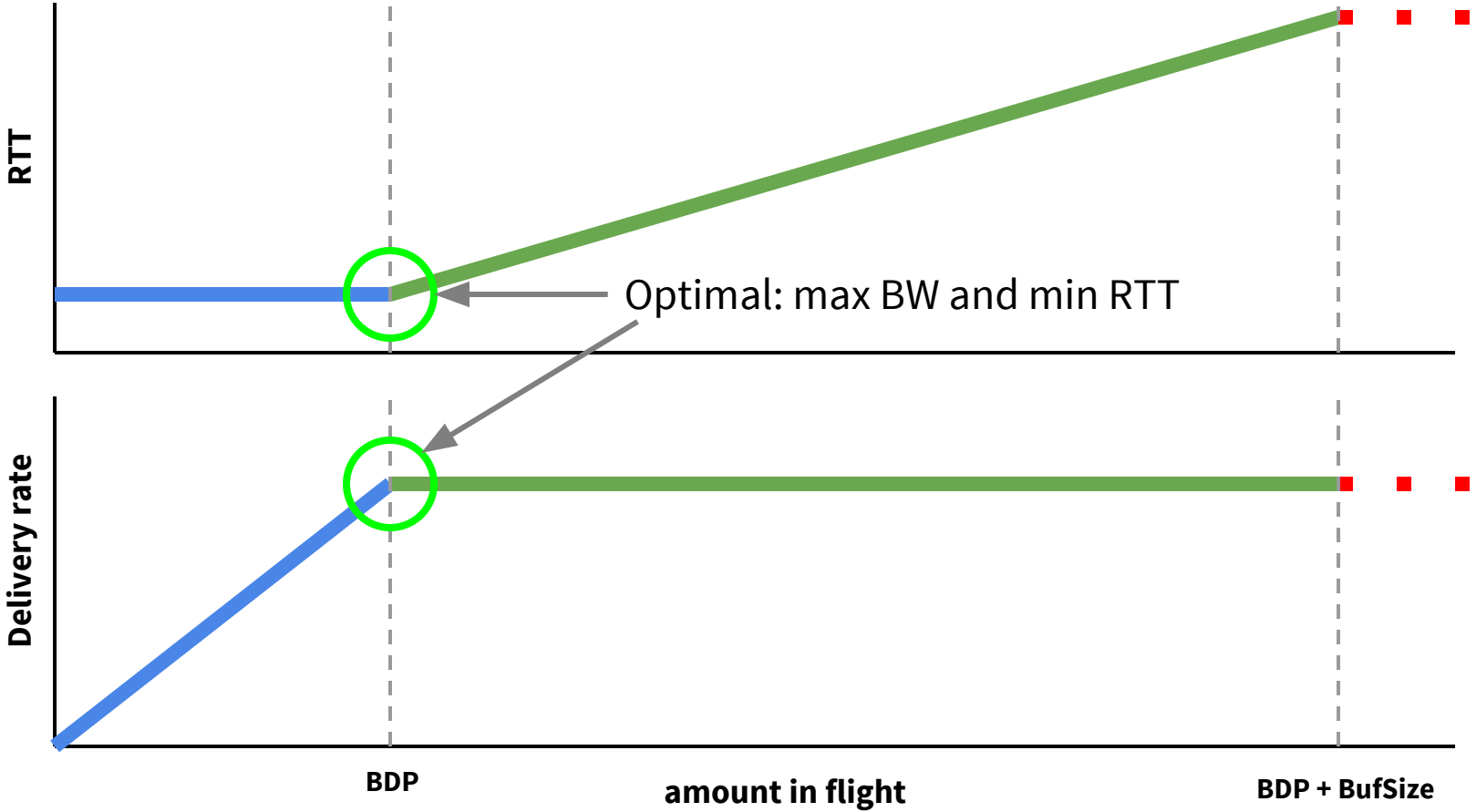
Network congestion and bottlenecks: delay



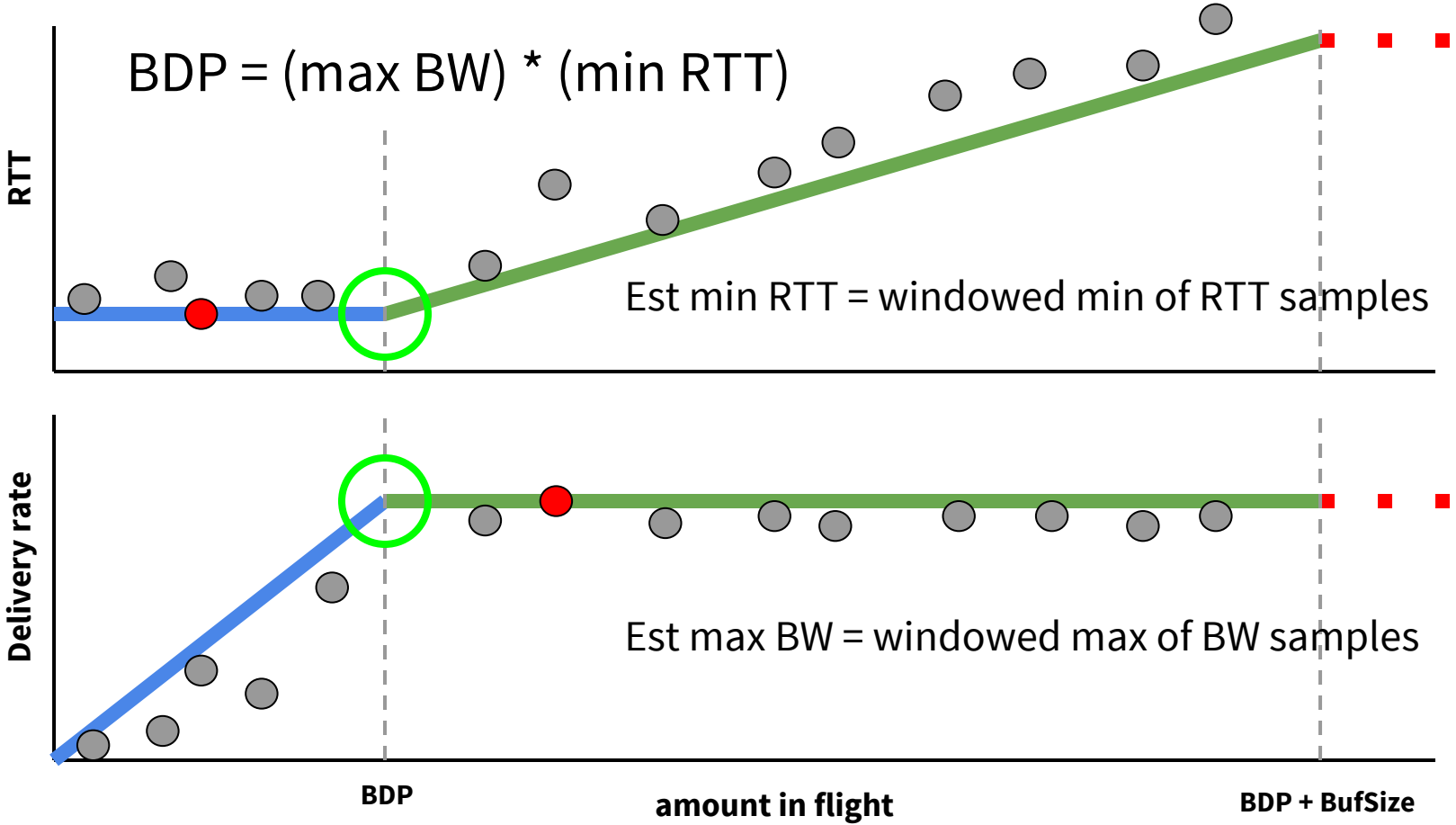
Loss-based Congestion Control



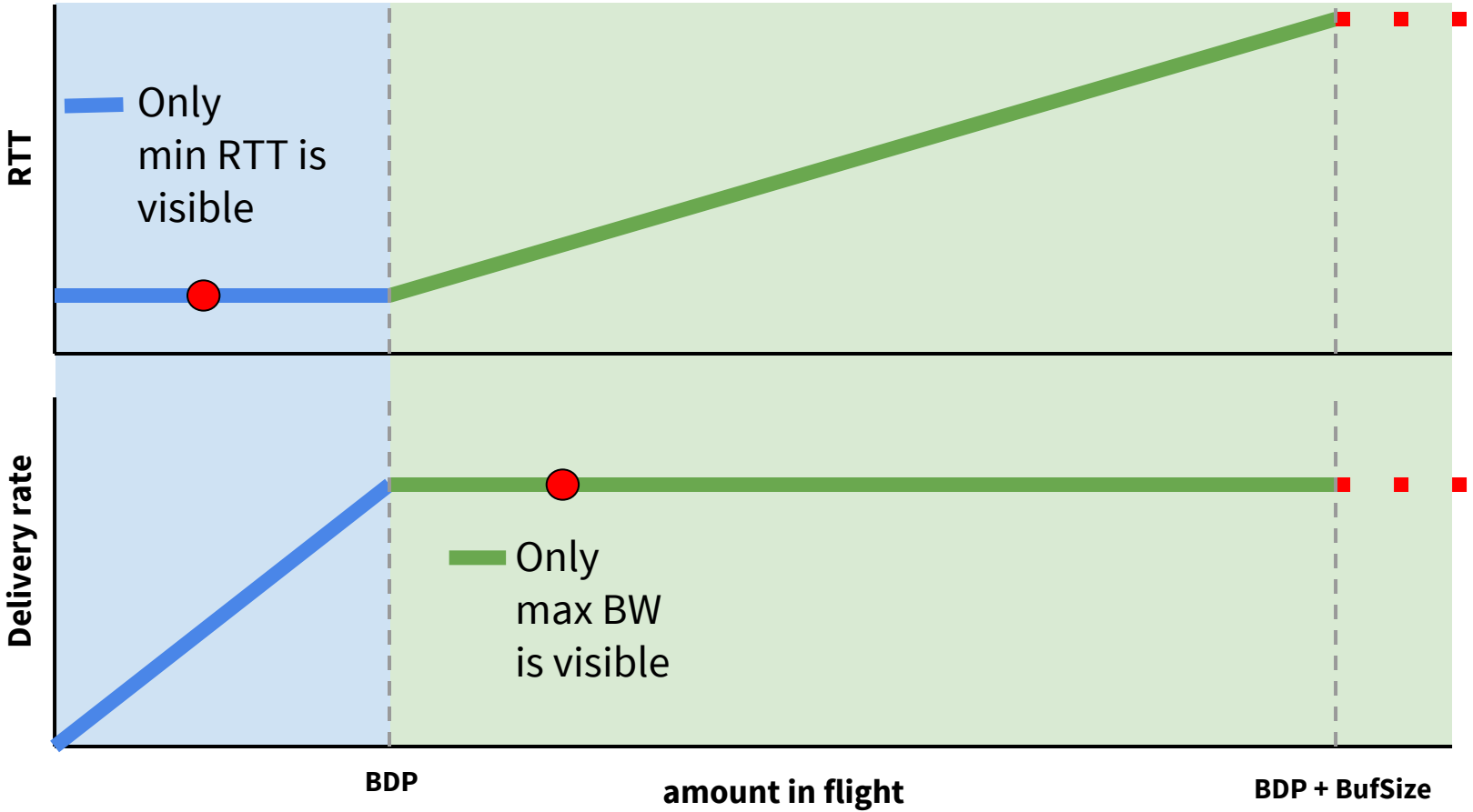
Optimal Operating Point



Estimating optimal point (max BW, min RTT)



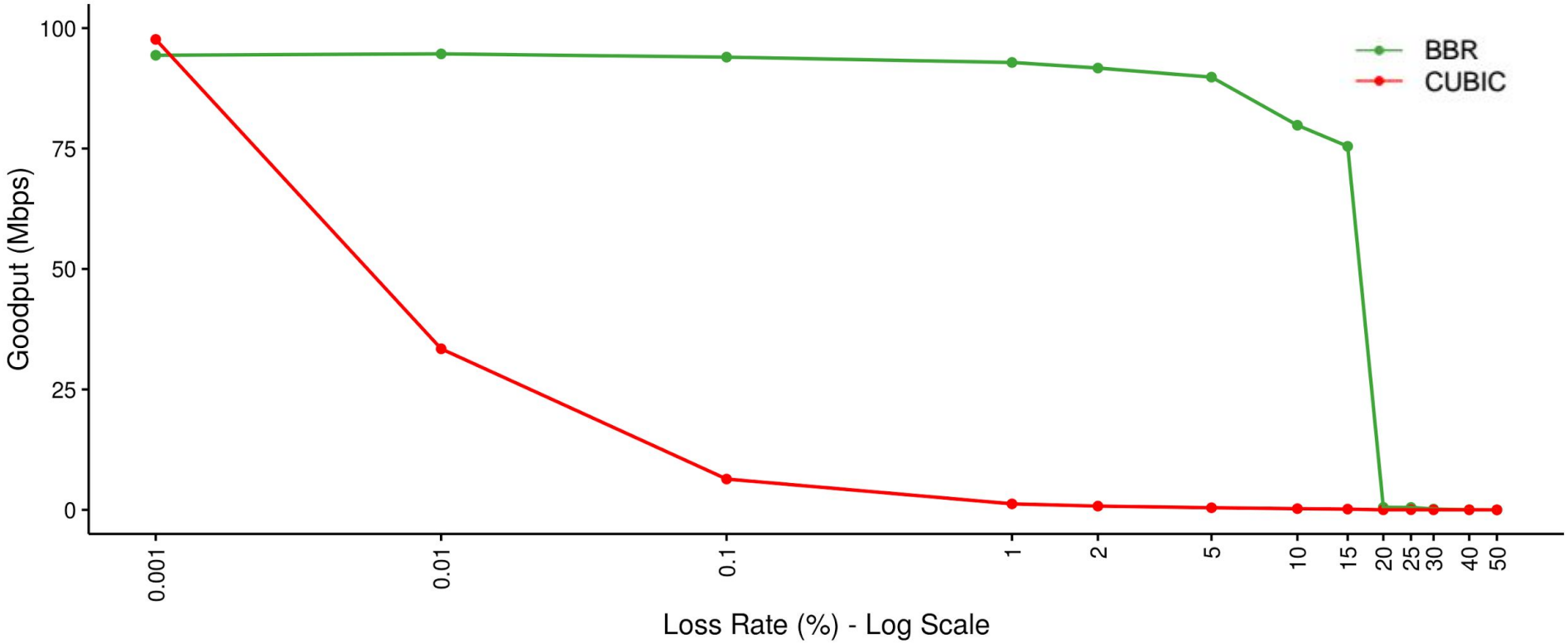
To see max BW, min RTT: probe both sides of BDP



BBR: Core Design

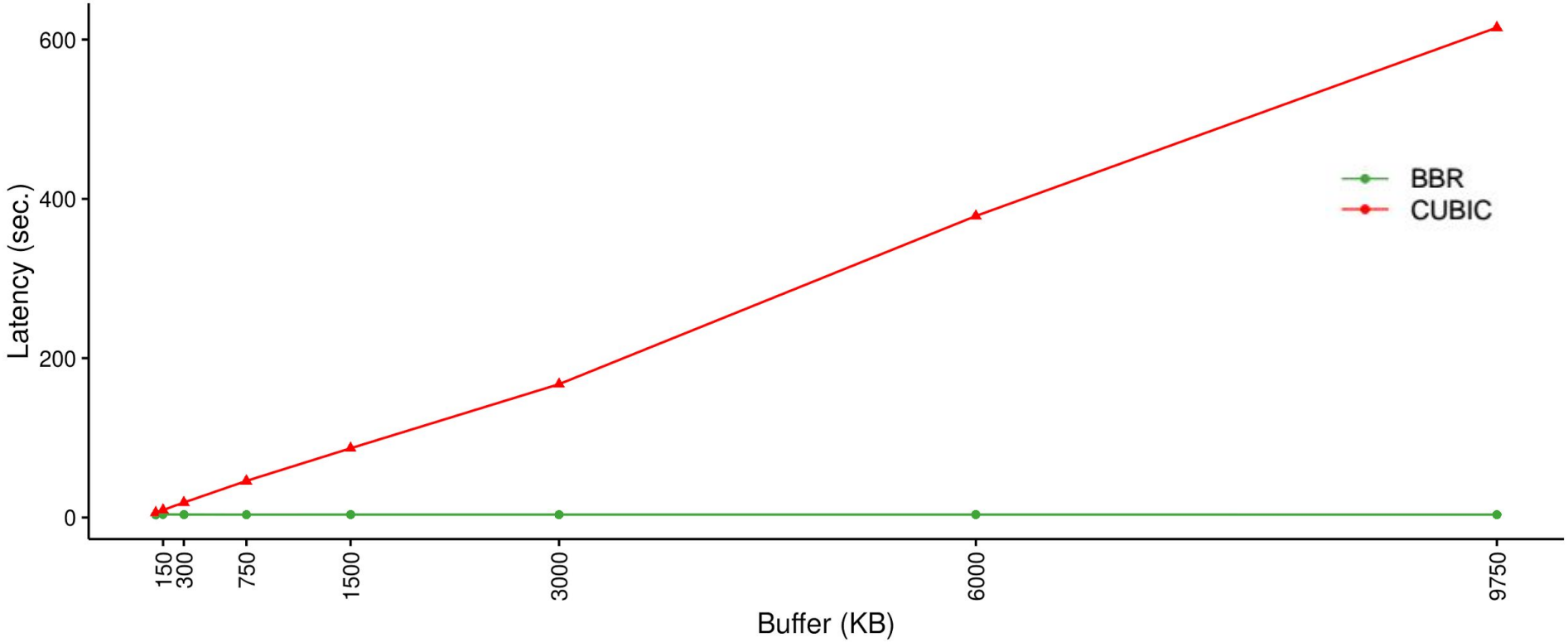
- **Model** network path
  - Update estimates of max BW and min RTT on each ACK
- **Control** sending based on the model, to...
  - Sequentially probe max BW and min RTT, to feed the model samples
  - Pace near estimated BW, to reduce queues and loss
  - Vary pacing rate to keep inflight near BDP (for full pipe but small queue)

BBR: fully use bandwidth, despite high packet loss



BBR vs CUBIC: synthetic bulk TCP test with 1 flow, bottleneck_bw 100Mbps, RTT 100ms

BBR: low queue delay, despite bloated buffers



BBR vs CUBIC: synthetic bulk TCP test with 8 flows, bottleneck_bw=128kbps, RTT=40ms

BBR: Deployment Experience

- Deployed on Google internal WANs; rolling out on google.com, YouTube
- On Google B4 WAN between datacenters (BBR used for vast majority of TCP)
  - RPCs 2-20x faster than CUBIC (8MB @ default QoS)
  - Bulk throughput up to 130x faster than CUBIC (@ default QoS)
- On Google.com
  - Faster web page downloads (particularly in developing world)
- On YouTube
  - Higher bandwidth
  - Less rebuffering
  - Lower delay: Cuts median RTT by 53% (by 80% in developing world)

BBR: Conclusion

- BBR: model-based congestion control
  - Goal: maximize bandwidth, then minimize queue
  - Result:
    - 100x bandwidth of CUBIC w/ big BDP, moderate loss (0.1% - 15%)
    - lower queuing latency with bufferbloated last mile links
- Next (WIP): reducing loss rates, improving fairness vs. loss-based CC
- For more information:
  - *"BBR: Congestion-Based Congestion Control"*, Neal Cardwell, Yuchung Cheng, C. Stephen Gunn, Soheil Hassas Yeganeh, Van Jacobson. ACM Queue, Sep/Oct 2016 and CACM, Feb 2017.
  - Mailing list for discussion: https://googlegroups.com/d/forum/bbr-dev
  - Reference code in Linux v4.9 TCP