AUBURN
UNIVERSITY

BBR: Congestion-Based Congestion Control

Promise Owei & Ticao Zhang

Electrical and Computer Engineering

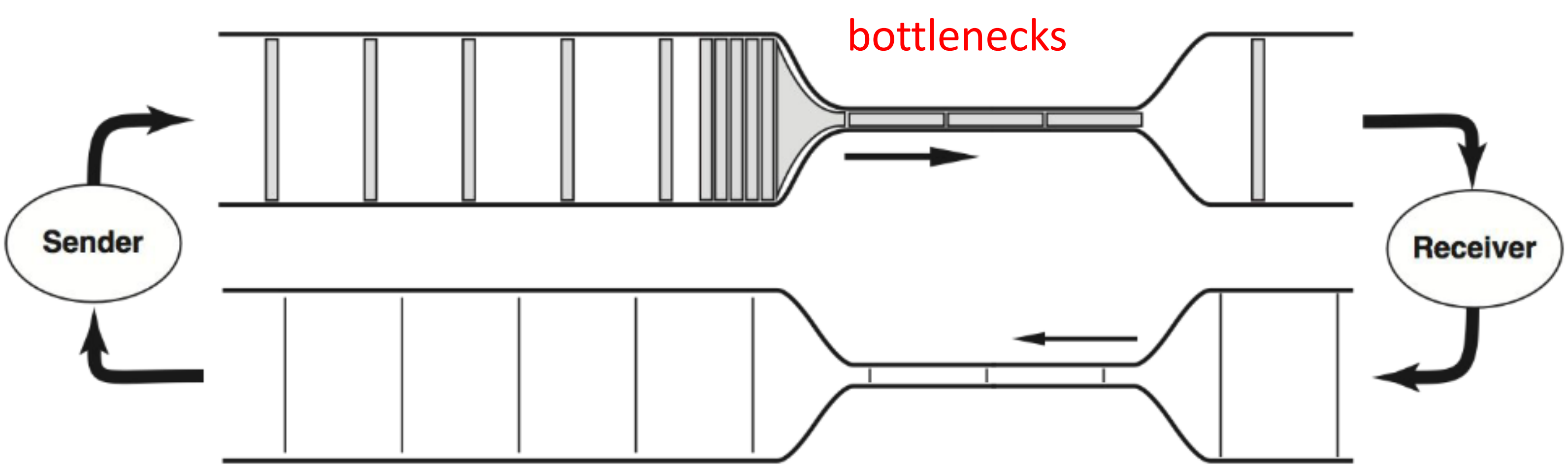
Dec 5th, 2017

Outline

- Introduction and motivation
  - Congestion control
- How to implement?
  - CUBIC
  - BBR
  - Performance comparison
- Performance
- Summary

Introduction and Motivation

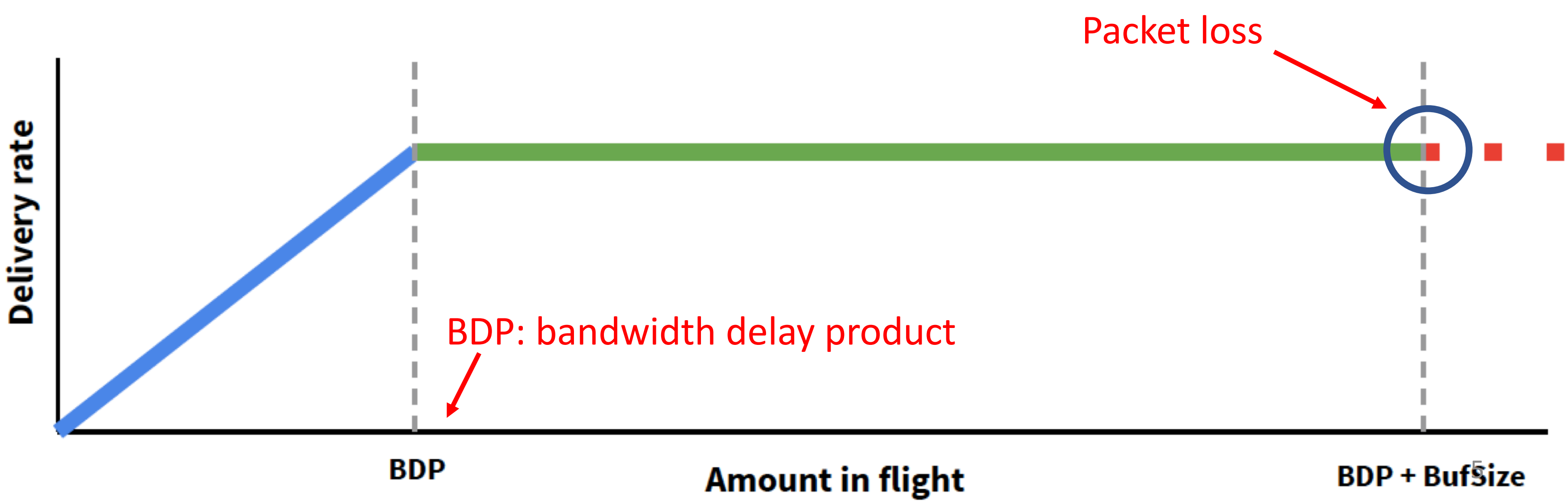
Congestion and bottlenecks



- Congestion: The load in a network is greater than the capacity of the network!
- Congestion control: Mechanisms that can either prevent congestion, before it happens, or remove congestion after it has happened

Congestion control algorithms

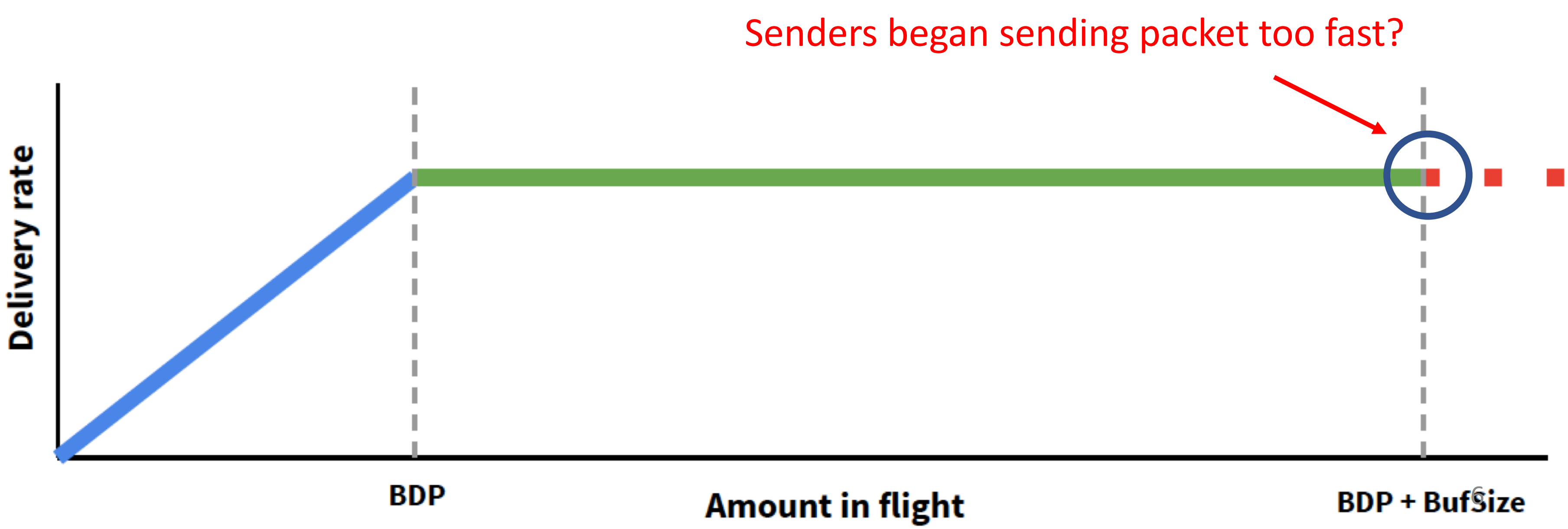
- Loss-based congestion control
  - Reno
  - Cubic (an extension to current TCP standards)
- When switches' and routers' queues well-matched with the bandwidth of the network, loss based method works well



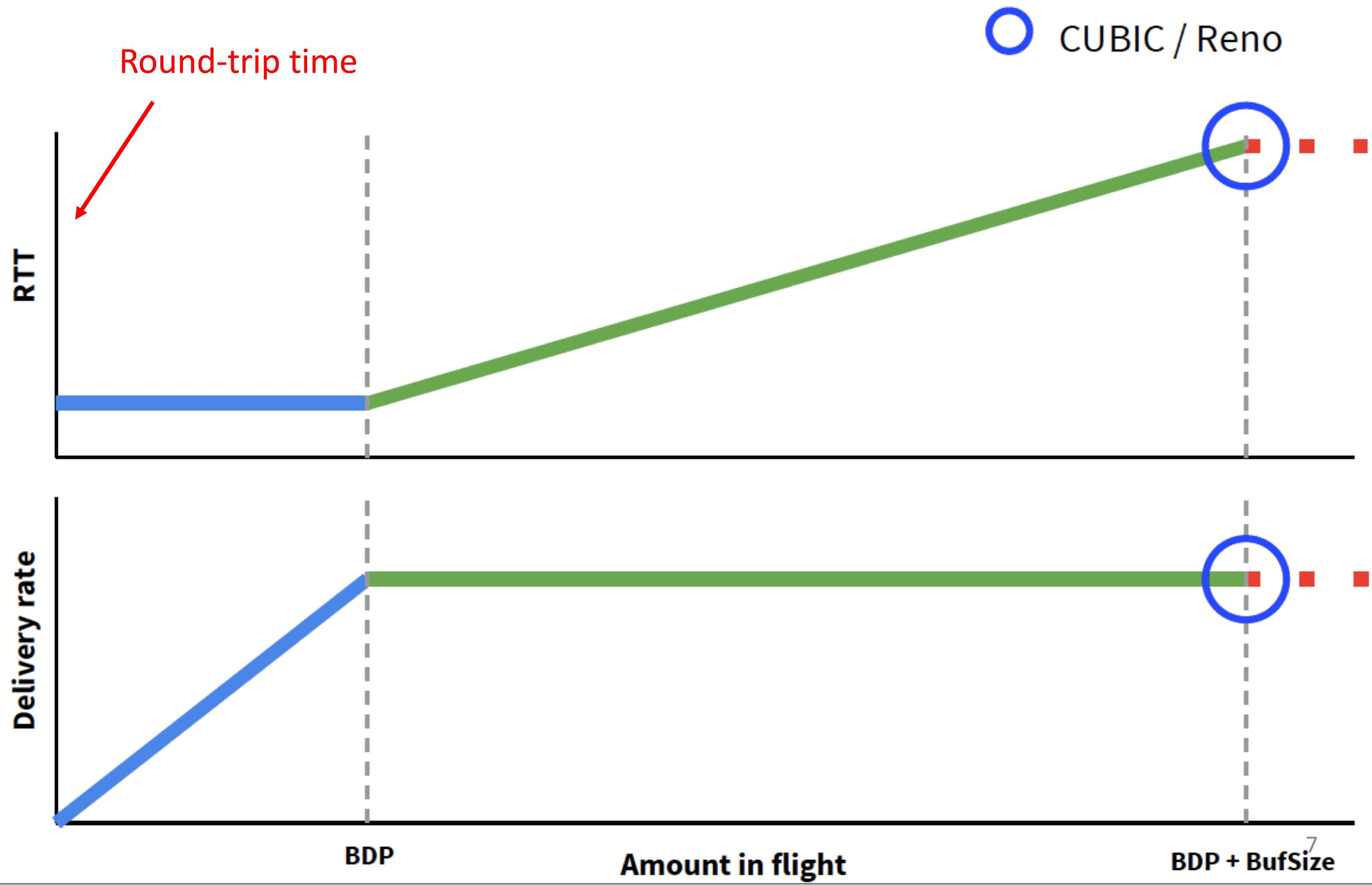
Problem

packet loss ≠ congestion

- "bufferbloat" problem: congestion happens before packet loss due to deep buffer.
- "Low ultilization" problem: congestion happens after packet loss due to shallow buffer.
- Large RTT problem

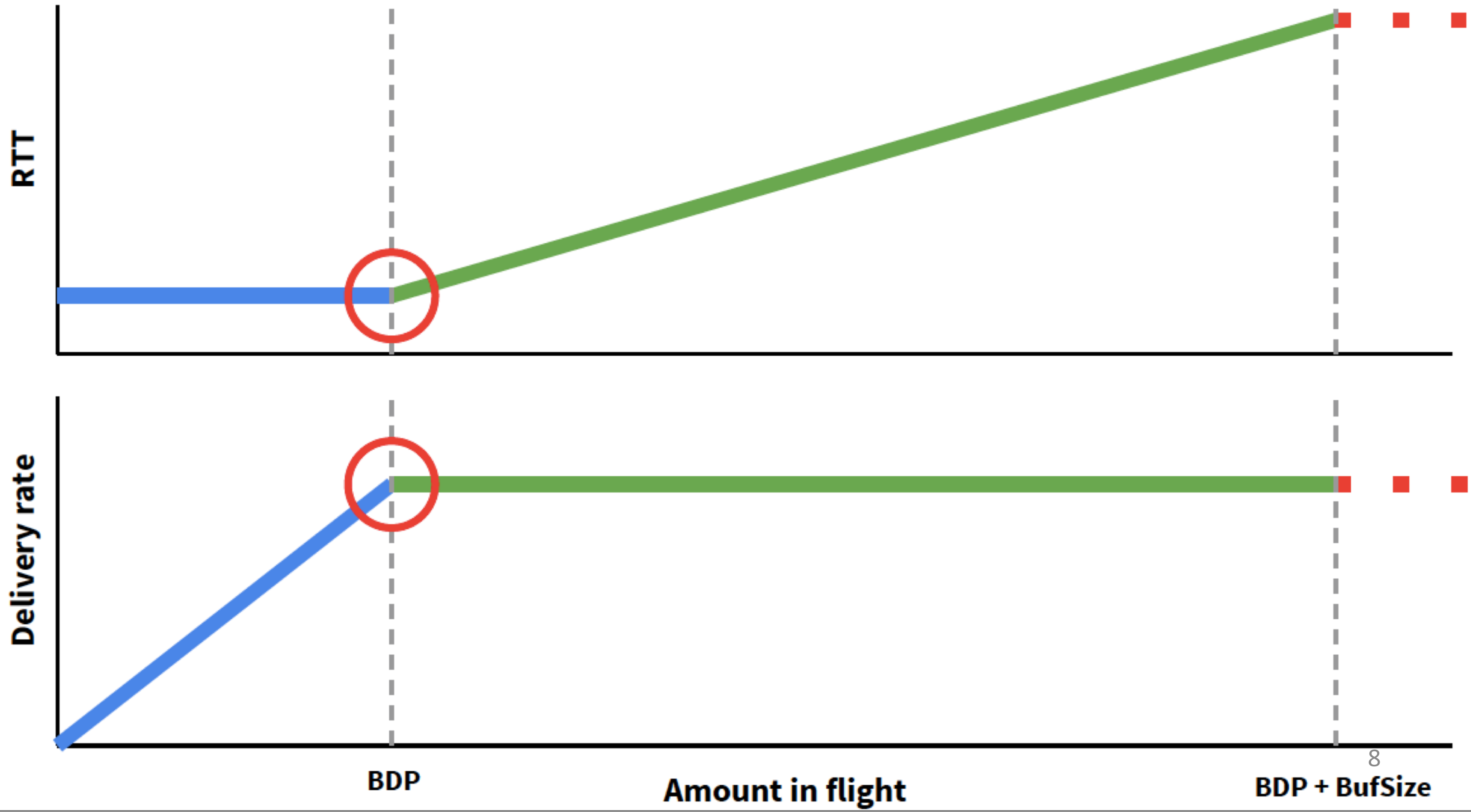


Problem


CUBIC / Reno
Round-trip time
RTT
Delivery rate
BDP
Amount in flight
BDP + BufSize

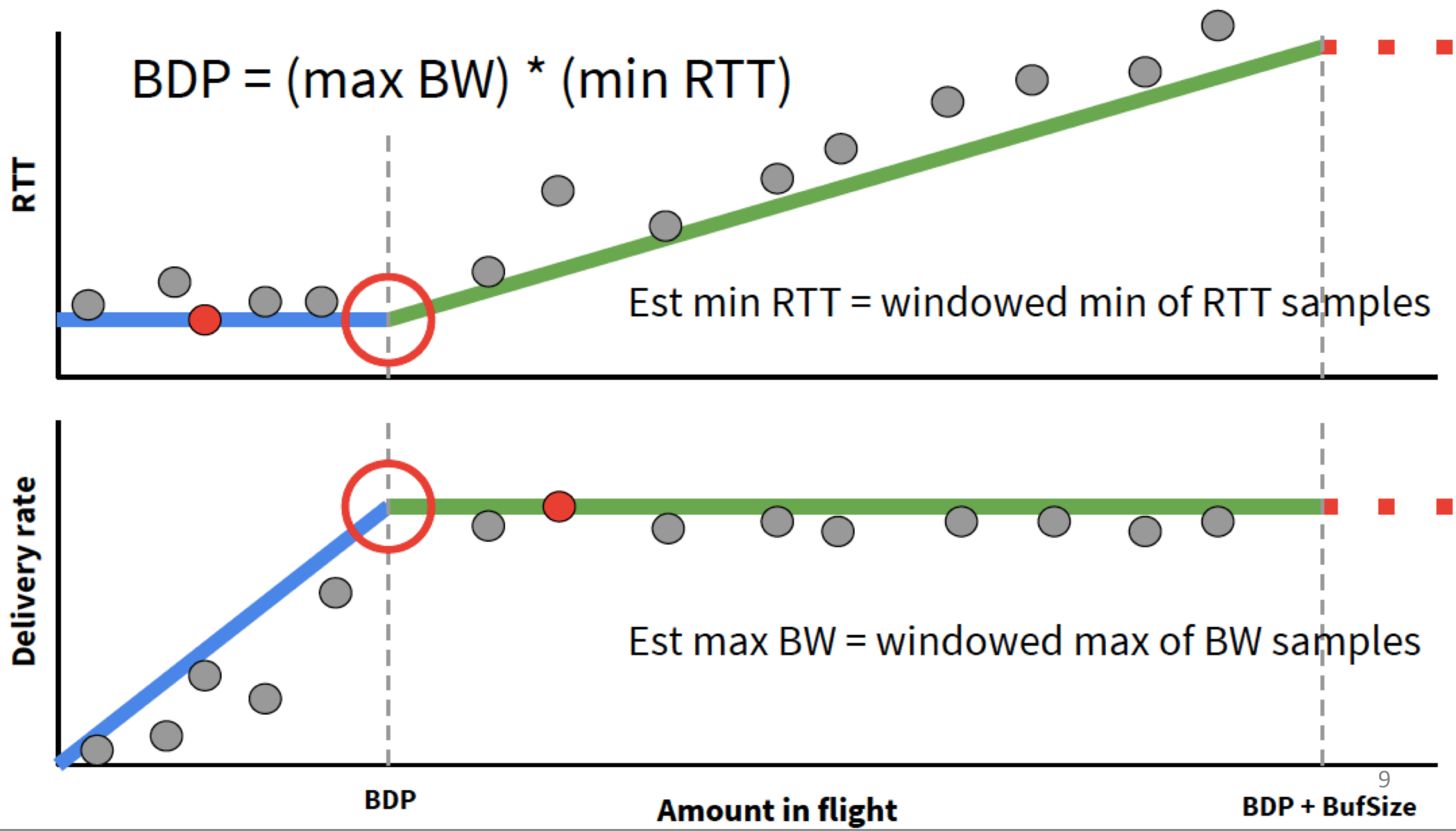
Optimal point

Optimal: max BW and min RTT (Gail & Kleinrock. 1981)



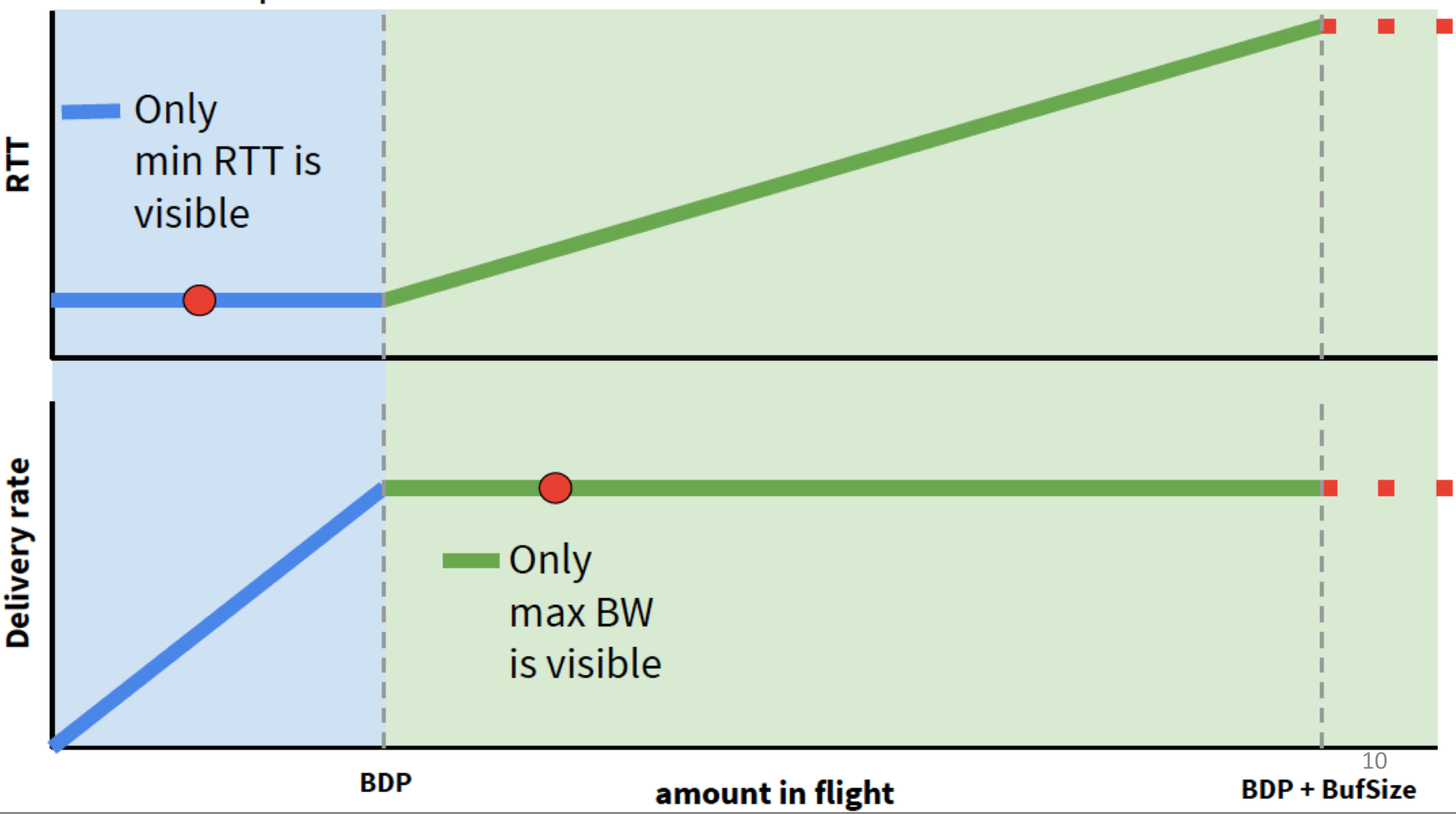
Solution

Estimating optimal point (max BW, min RTT)



Solution

But to see both max BW and min RTT,
must probe on both sides of BDP...



Solution

- BBR helps to stay near (max BW, min RTT) point

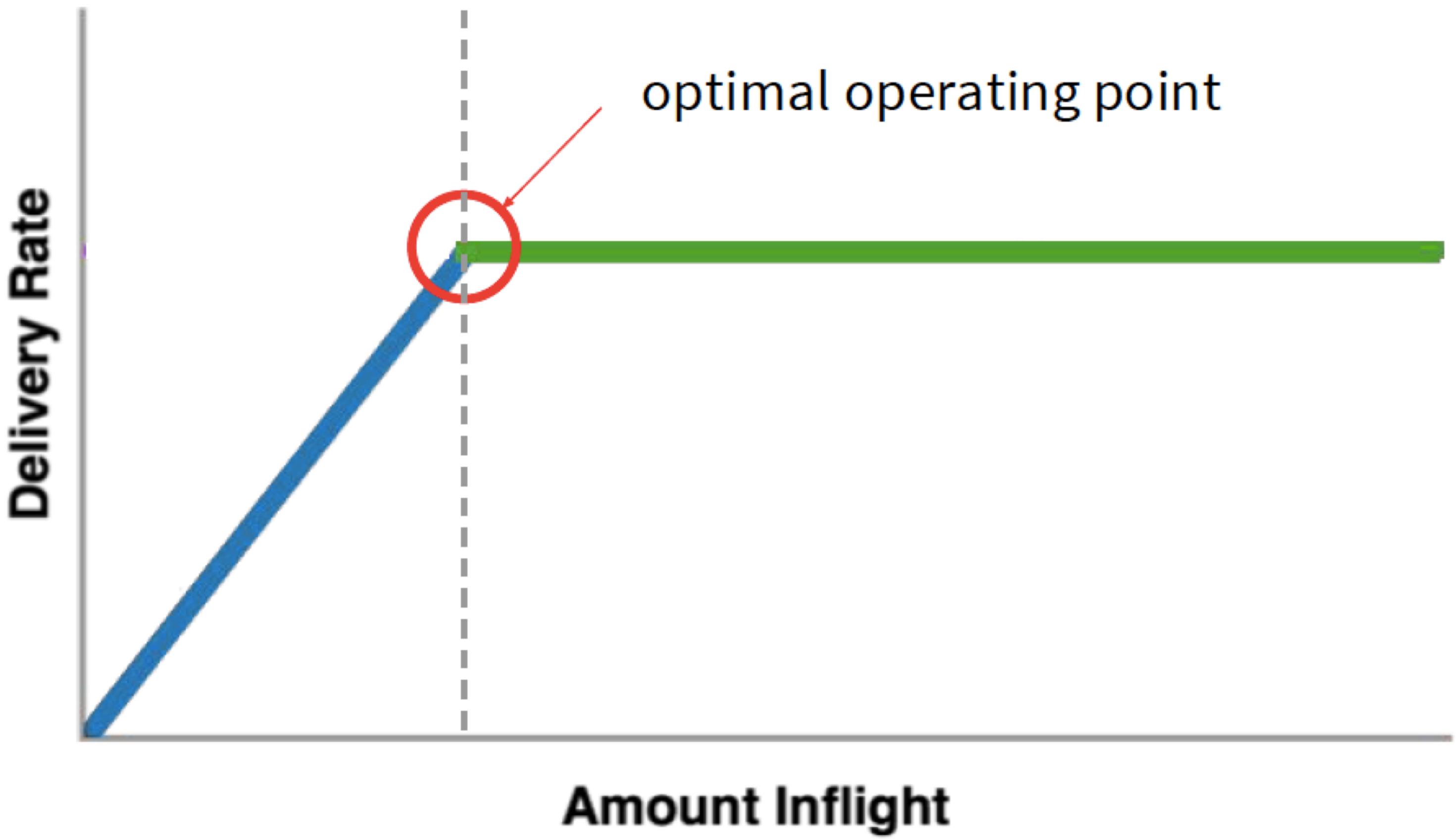
- BBR = Bottleneck Bandwidth and Round-trip propagation time
  - **Builds** a model of the network path
  - **Updates** windowed max BW and min RTT estimate on each ACK
  - **Controls** sending based on this model

- BBR seeks high throughput with small queue by probing BW and RTT sequentially.

Implementation

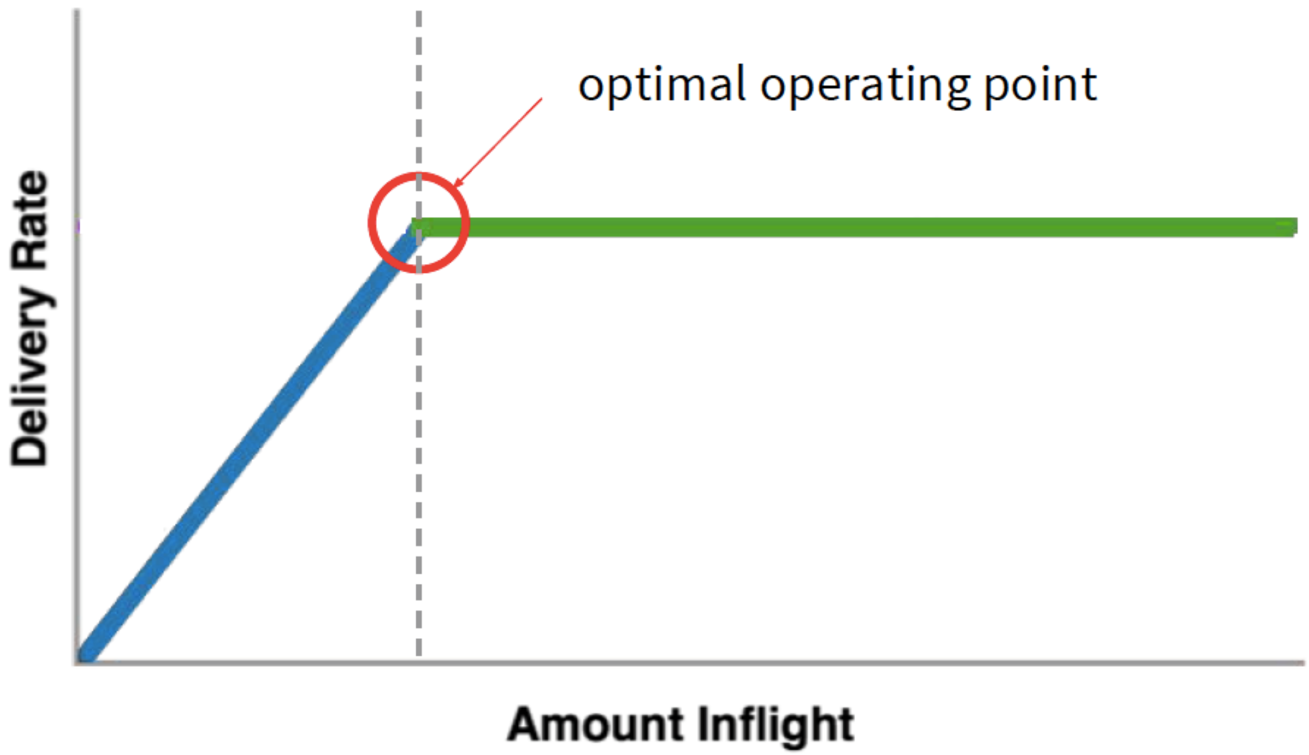
BBR Target

- walk toward (max BW, min RTT)



BBR model

- Model the network path
  - BBR.BtlBw: the estimated bottleneck bandwidth
  - BBR.Rtprop: the estimated two-way round-trip propagation delay of the path

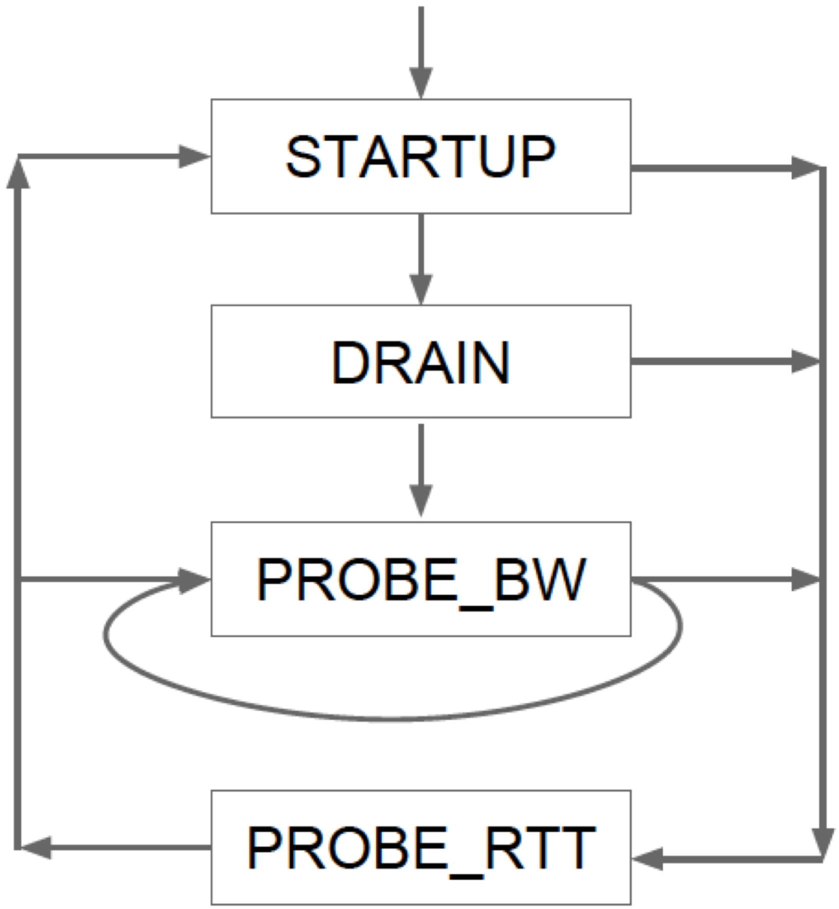


BBR control parameters

- BBR use the path model to control the sending's behavior
- Cubic
  - cwnd: the maximum volume of data allowed in flight
- BBR
  - pacing rate: the rate at which BBR sends data
  - cwnd
  - send quantum:

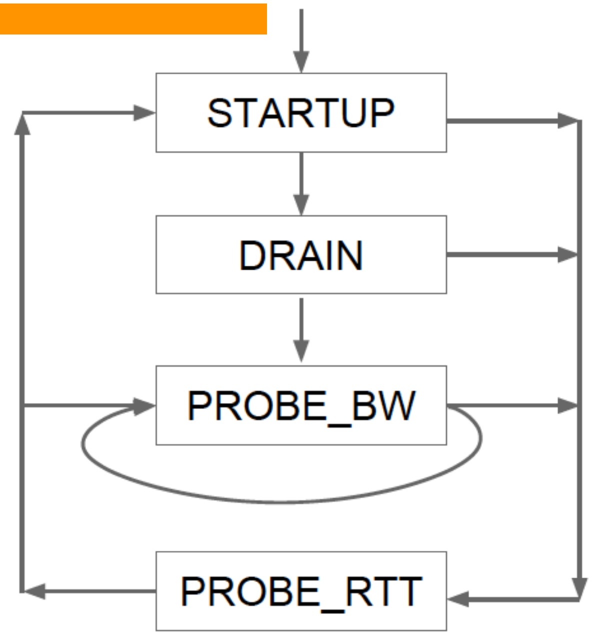
BBR state machine

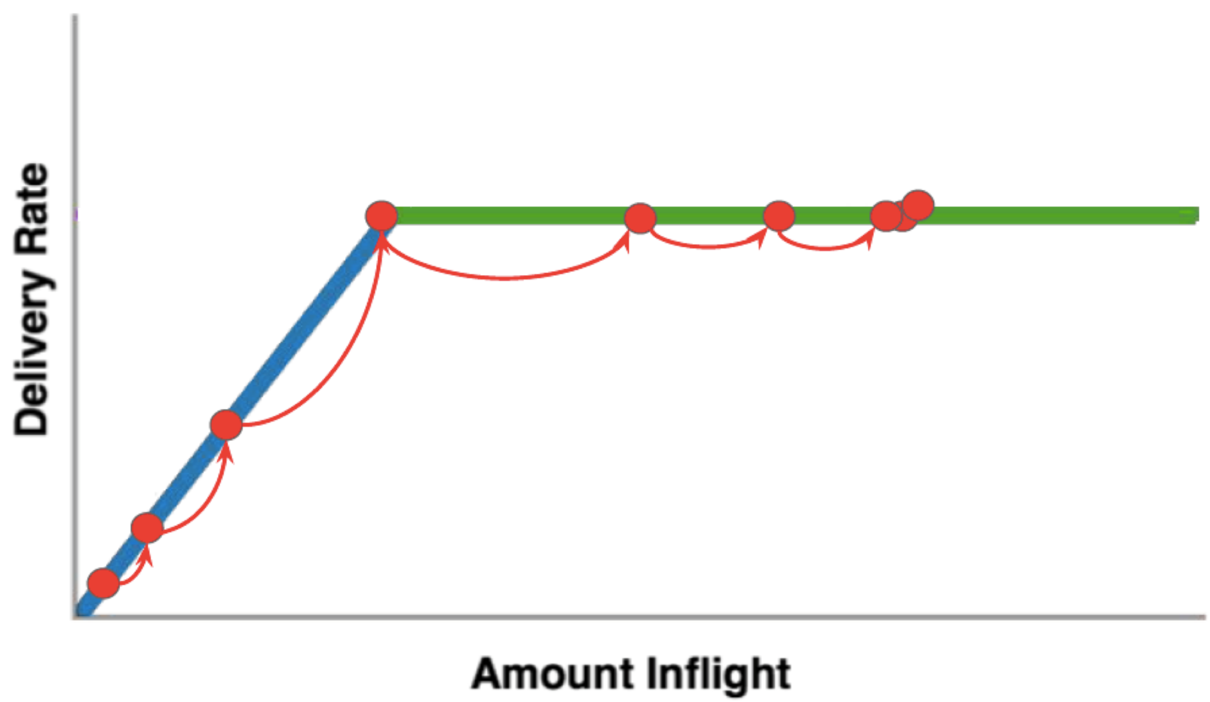
- BBR varies its three control parameters with a simple state machine



BBR state machine

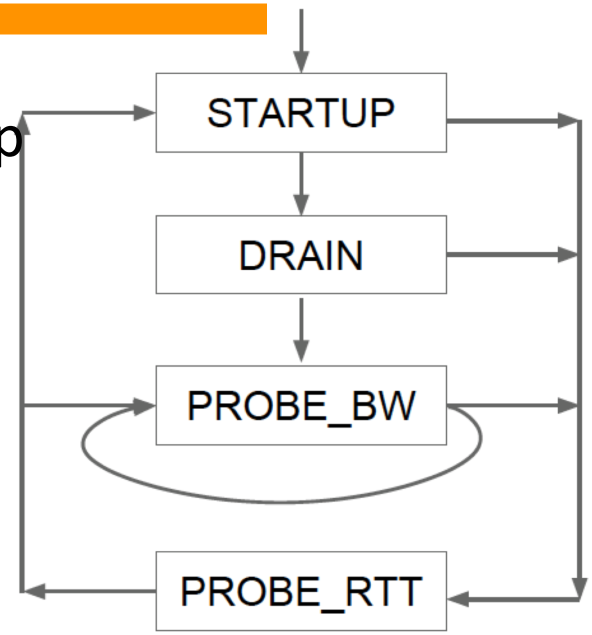
- Startup: exponential BW search

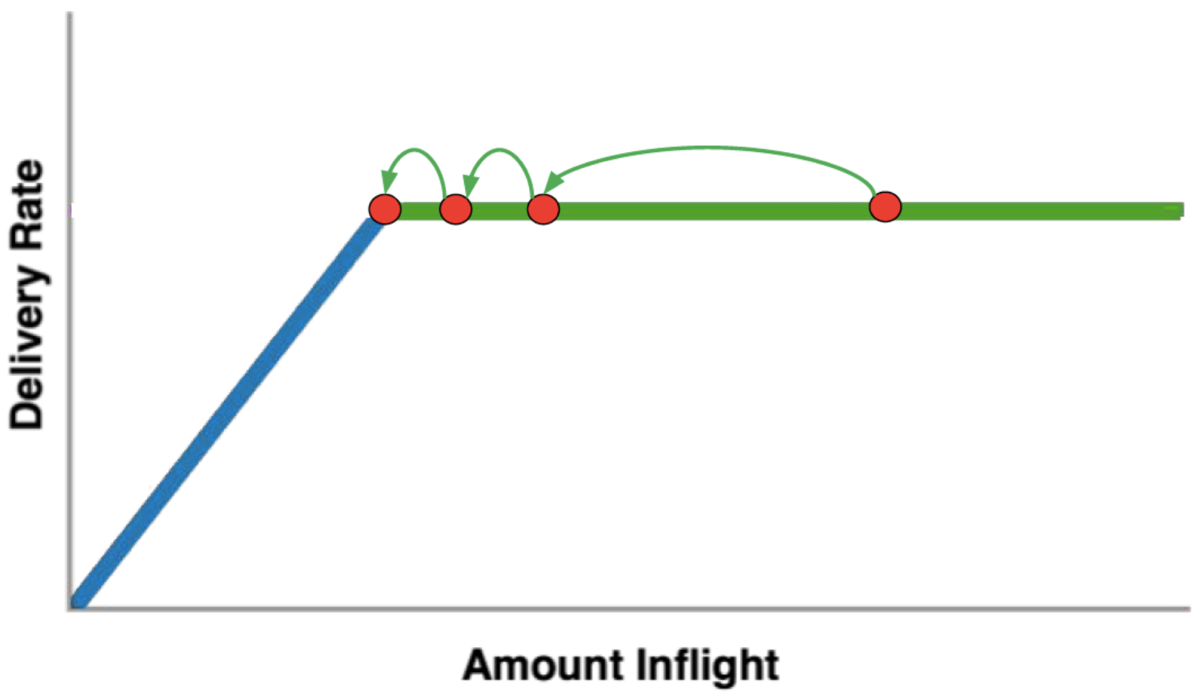




BBR state machine

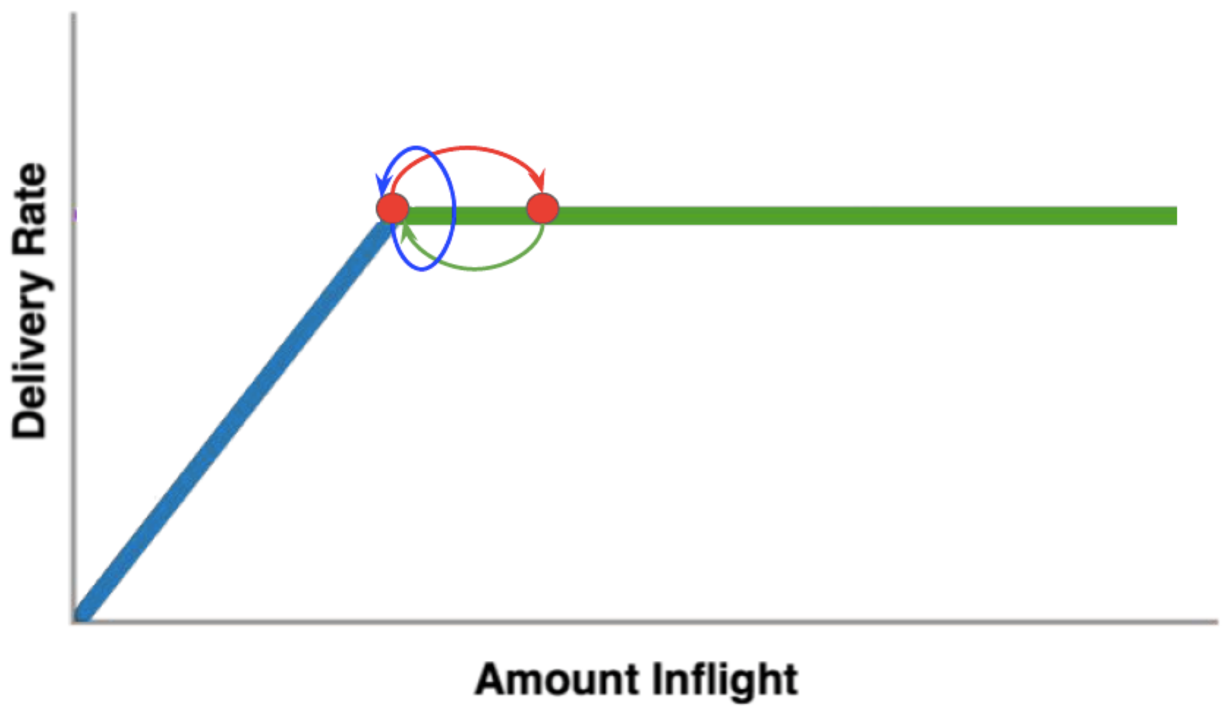
- Drain: drain the queue created during startup

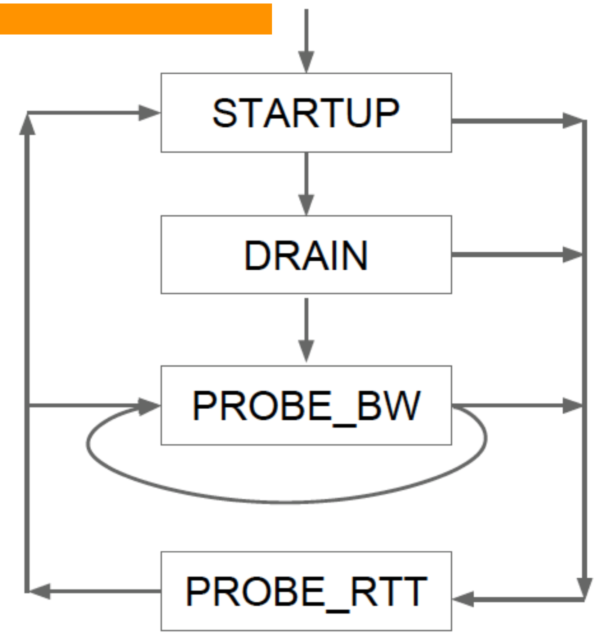




BBR state machine

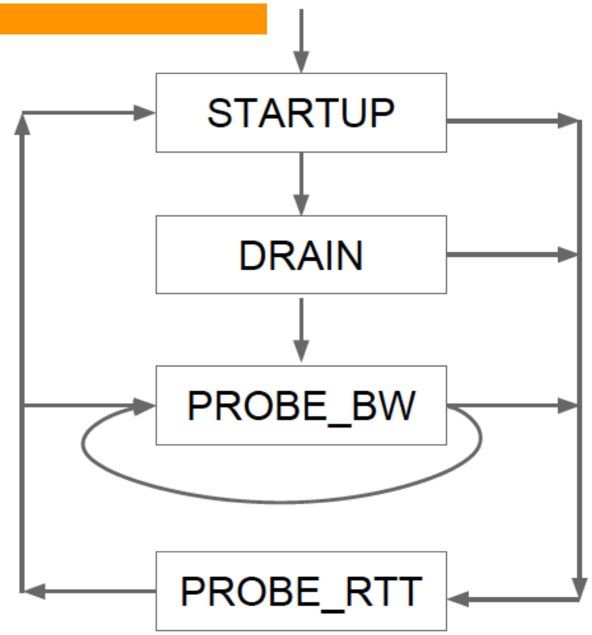
- Probe_BW: explore max BW, drain queue

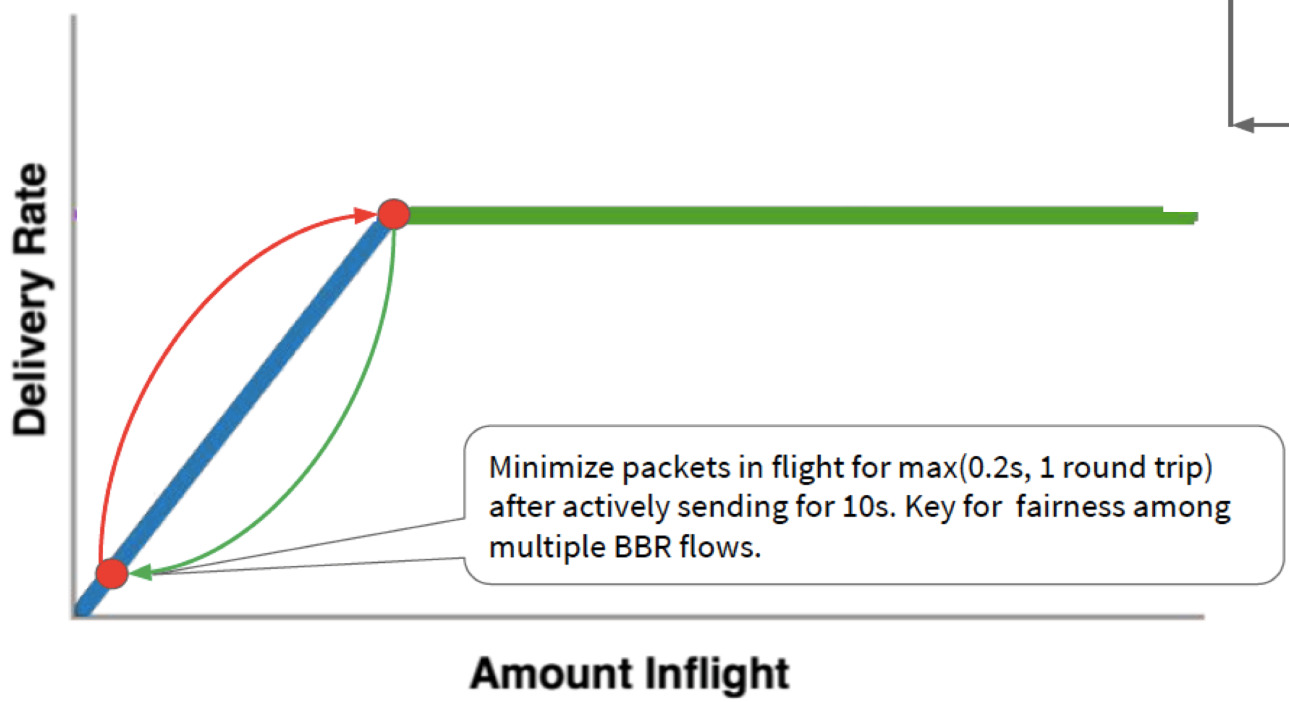




BBR state machine

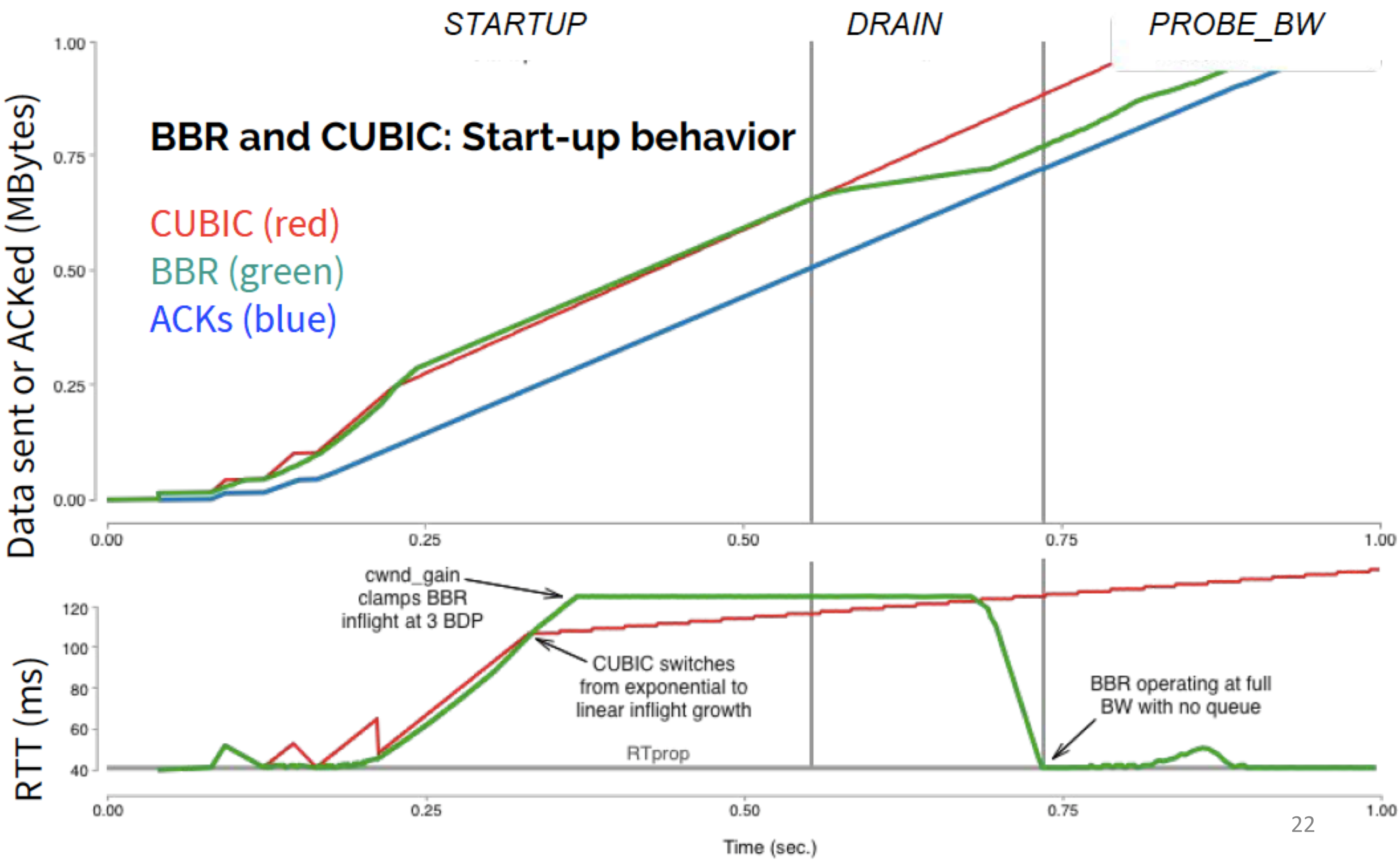
- Probe_RTT: drain queues to refresh min_RTT





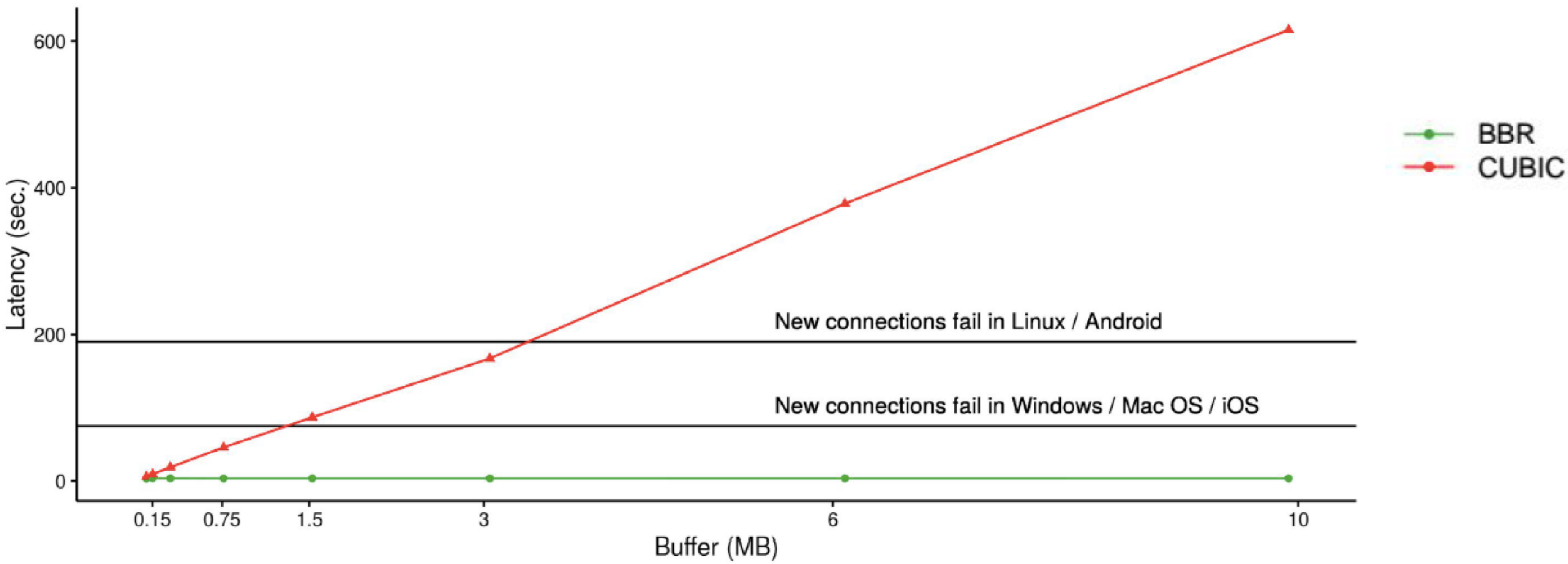
Performance

Startup behavior



Latency

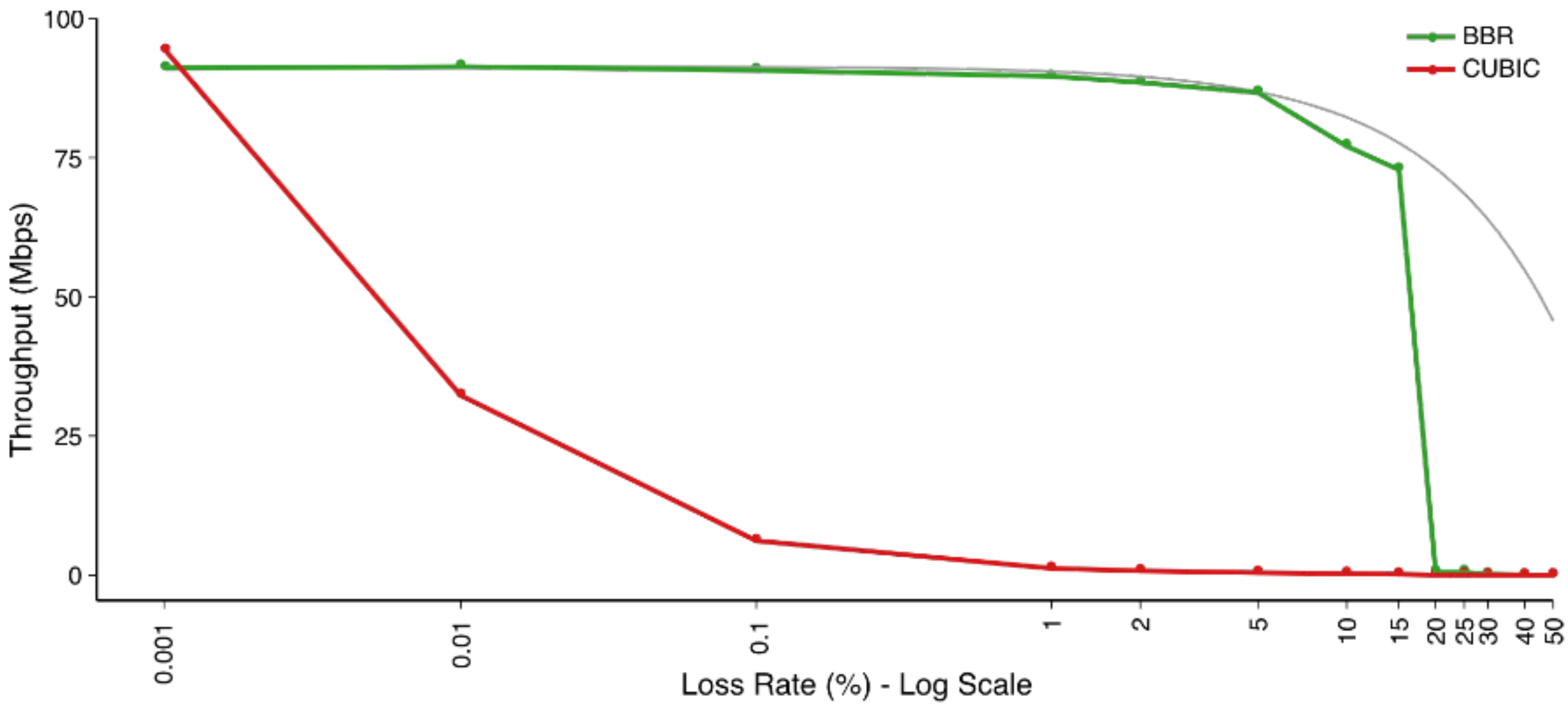
Low queue delay, despite bloated buffers



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

Throughput

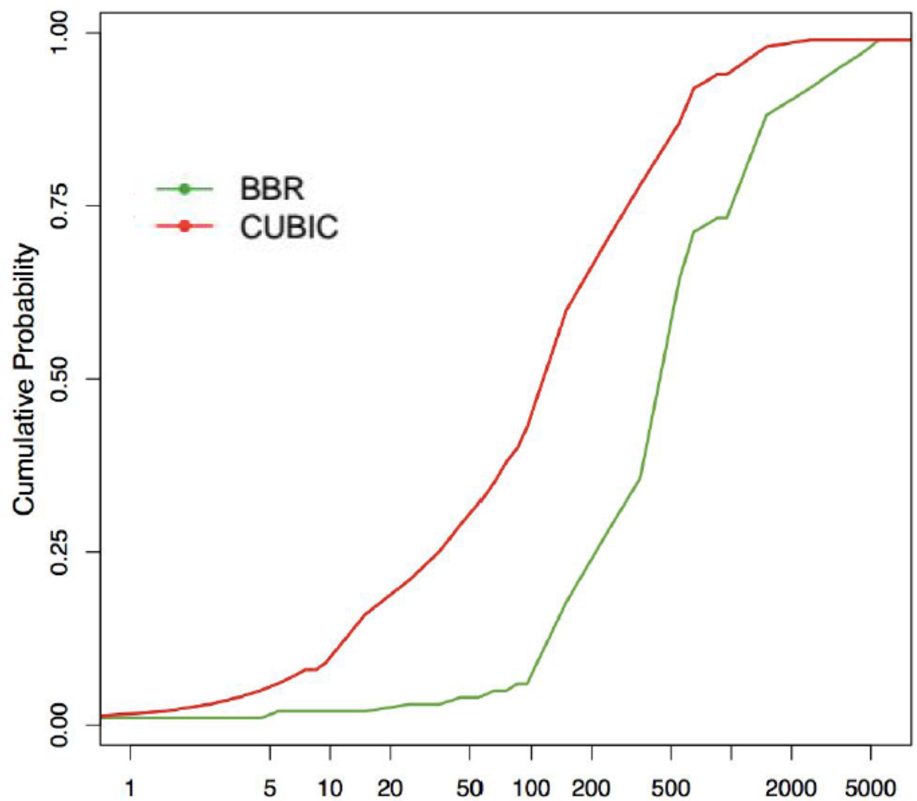
Fully use bandwidth, despite high loss



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

BBR improves Google Cloud Platform performance

BBR is 2-20x faster on Google WAN



- BBR used for all TCP on Google B4
- Most BBR flows so far rwin-limited
  - max RWIN here was 8MB (tcp_rmem[2])
  - 10 Gbps x 100ms = 125MB BDP
- after lifting rwin limit:
  - BBR 133x faster than CUBIC

Conclusion

- BBR: model-based congestion control
  - Goal to maximize bandwidth then minimize queue
  - Orders of magnitude higher bandwidth and lower latency
  - Runs purely on the sender and no changes to the protocol, receiver
  - Only depends on RTT and packet-delivery acknowledgement

BBR will continue to evolve …

"BBR: Congestion-based Congestion Control", ACM Queue, Oct 2016

Neal Cardwell, Yuchung Cheng, C. Stephen Gunn, Soheil Hassas Yeganeh, Van Jacobson

Q&A

Thanks