Fix the hosts (Position Paper)

Matt Mathis (Google)
Andrew McGregor (Fastly)

Stanford Buffer Sizing Workshop
Dec 2, 2019

Punchline

At the largest scales we can not afford "properly" sized buffers

- They will be perpetually doomed by Moore's law
- It is far more cost effective to fix the end systems
  - Pacing at scale
  - BBR is a good start

My charge to this community: invert the question.

Given buffer sizes are smaller than we would prefer, how can we maximize effective network capacity and efficiency?

Moore's law

Colloquially: Speed-complexity product doubles every 18 Months.

Networks link rates double every 2 years

- Buffer speed has to double every 2 years
- Buffer size has to double every 2 years
- Buffer speed-complexity product needs to **quadruple** every 2 years

But this is economically infeasible in the fastest parts of the Internet

So drain times keep falling

- Sub mS is becoming more common

Why do we want large buffers?

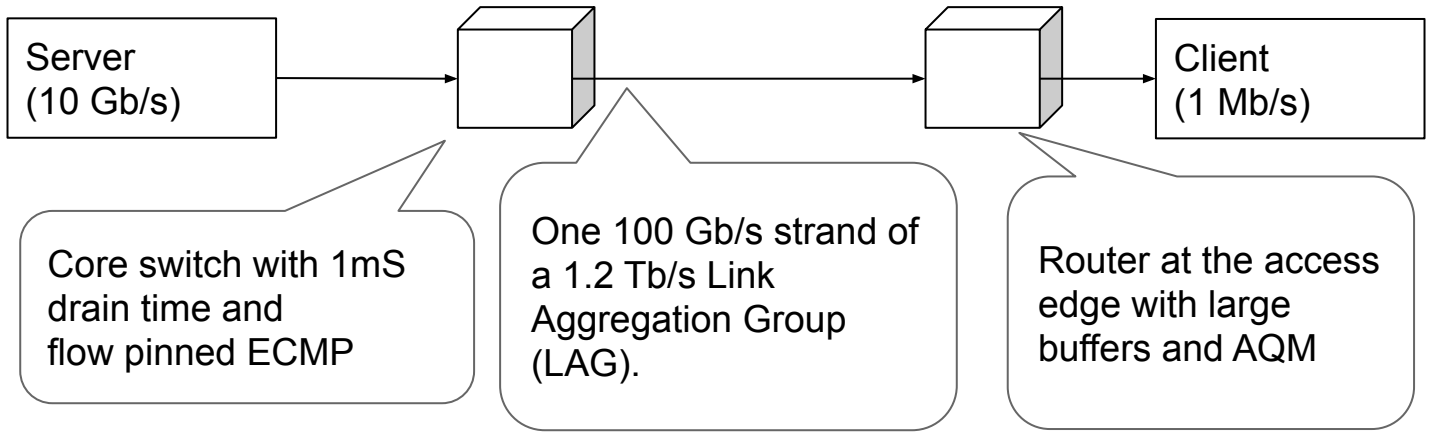
- Many reasons.... but we dwell on one.

- [VJ88] Design principles:
  - Packet conservation and TCP self clock
    - Vast majority of transmissions are triggered by ACKS
  - Explicitly stated: the entire TCP system is clocked by packets flowing through the bottleneck queue
  - This clearly works when buffer size > Bandwidth-Delay-product
  - But does this really work when the buffer size is only 1% of the BDP?
    - The clock source (the bottleneck) does not have enough memory to significantly spread or smooth bursts

BBR: new first principles for Congestion Control

- BBR builds an explicit model of the network
  - Estimate max_BW and min_RTT
- The BBR core algorithm:
  - By default pace at a previously measured Max_BW
  - Dither the pacing rate to measure model parameters
    - Up to observe new max rates
    - Down to observe the min RTT
    - Gather other signals such as ECN
- BBR's "personality" is determined by the heuristics used to dither the rates and perform the measurements
  - These heuristics are completely unspecified in the core algorithm

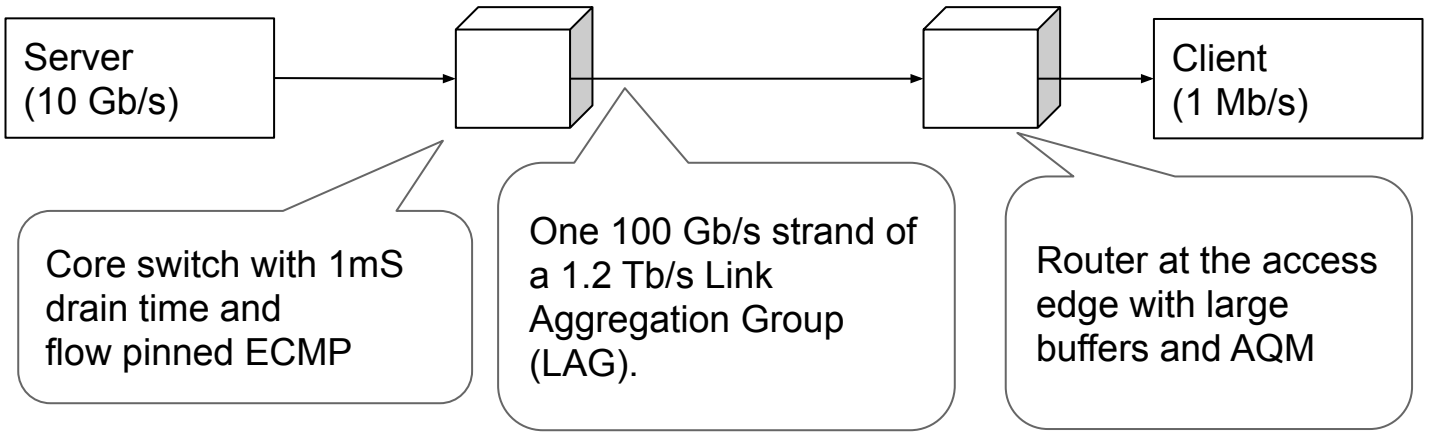
BBR TCP



Assume 50 mS RTT and that the return path batches or thins ACKs.

- TCP estimates max_BW (at far edge) and min_RTT (entire path)
- Servers send at ~1Mb/s per client (dithered to measure bottleneck)
- Traffic is smoother than Markov at some scales
  - Nominally no standing queues in the core
- No loss in the core except true overload or pathological pacing synchronization (extremely unlikely)

Self clock is not good in a short queue Internet



Assume 50 mS RTT and that the return path batches or thins ACKs.

- Server rate bursts are delivered all the way to the far access edge
  - Where the bottleneck clocks the entire system
  - ACK thinning or compression causes persistent server rate bursts
    - e.g. WiFi and LTE channel arbitration
- Concurrent bursts from 11 servers will cause queues in the core
- Pathological ACK synchronization can cause loss at 2% load
- The details of the burst structure come from weakly bound properties
  - Average window size, mechanisms that retime ACKs, etc

Deprecating VJ88 has profound implications

- 30yrs of research on window based CC w/ self clock
  - Some things that we think we "know" are wrong
  - There might be gold in some ideas that were abandoned
  - Pretty much everything needs to be revisited
- Conjectures:
  - BBR framework easily adapts to multiple modeling strategies
  - Most window based CC algorithms have paced equivalents
  - Some CC algorithms fit even a better (e.g. chirping)
  - 20 years of past CC work needs to be ported into BBR

See: Mathis & Mahdavi "Deprecating the TCP Macroscopic Model" [CCR Oct 2019]

Buffer Sizing Research questions

- Ongoing improvements to BBR
- Quantify the impact of bursty traffic on other traffic
  - What does it cost? buffer space or extra headroom (wasted capacity)?
  - Can ISPs incentivize reducing bursty traffic?
- Are there alternatives besides pacing vs self clocked TCP?
- Does application transaction smoothing help?
  - BBR natively restarts at the old max_BW. Should that decay?
- Does ECMP still need flow pinning?
  - Paced packets are less likely to be reordered due to path diversity.
  - How much would it save us to discard flow pinning?

Conclusions

- Moore's law squared dooms large buffers
- Small buffers doom self clocked protocols
- Some form of pacing is inevitable
  - BBR is a good start, but long from done
  - Large content providers already have incentives
    - BBR solves real problems for them