Buffer Sizing: a Position Paper

Matt Mathis <mattmathis@google.com>, Andrew McGregor <andrewmcgr@gmail.com>

The traditional approach to the buffer sizing question has been to measure traffic properties, and then prescribe buffers necessary to carry that traffic. However, for the fastest devices in the Internet, large buffers are intrinsically extremely expensive or unavailable. For these devices, buffer size is not really a free parameter: vendors strongly bound maximum buffer sizes early in their design process. We might be more effective if we inverted the question: given that switch buffers are often smaller than we would prefer, what could be done elsewhere in the rest of the Internet to improve traffic statistics to make undersized buffers perform better.

Introduction

Moore's law dooms large buffers in the fastest portions of the Internet. The problem is that maintaining constant drain time in the presence of ever increasing interface (link) speeds requires that queue buffer memory evolve faster than Moore's law. Colloquially Moore's law states that the product of speed (clock rate) and complexity (device count) doubles every 18 months. Internet data rates have been doubling roughly every 2 years (slightly slower than Moore's law) so to maintain constant drain time, the speed-complexity product for buffer memory has to double every year. There is no cost effective way to beat Moore's law, and as a consequence over the span of decades the available drain times have been dropping for the fastest devices on the network. Note that just a little bit behind the bleeding edge, the problem becomes tractable (although expensive). Today you can buy huge routers with very large numbers of 10 Gb/s ports, but these are really a factor of 40 behind the fastest components, which would require a factor of 1600 higher speed-complexity product to reach the same scale and drain times.

We need large buffers because the Internet relies on statistical multiplexing of self-clocked protocols, which tend to have very high traffic variance. Self-clocked protocols preserve the time structure from each round trip to the next, such that events that cause packet bursts or packet trains[1] raise the traffic variance over many subsequent round trips. If there is enough buffering at bottlenecks, the bottlenecks themselves can smooth the traffic. In faster parts of the Internet there isn't enough queueing to smooth the traffic and mechanisms that reduce traffic variance are less obvious: there is some diffusion caused by random jitter, and congestion control window reductions reduce a given flow's contribution to the traffic which typically reduces the variance as well.

[1] e.g TCP slowstart, line rate restart after pauses, ACK aggregation or decimation on the return path, ACK compression in time due to channel arbitration or data queued on the return path.

It is only a slight oversimplification to characterize the buffer sizing question as studying the interplay between extremely complicated traffic statistics, average load, buffer sizes and the induced packet losses or ECN marks. This is valuable work that leads to deep insights into the network and traffic it carries.

Considering alternatives

However, with the advent of new technologies that enable Internet servers to pace traffic at industrial scale, we are in a position to suppress the majority of the causes of traffic variance, which would in turn reduce the need for buffering in the network. The pacing technologies, fq_pacing[1] and Carousel[2], together with BBR[3] congestion control, permit servers to transmit data on high precision timers for millions of concurrent flows. The resultant traffic is much smoother than self clocked traffic, and as a consequence needs less network buffer space to attain the same loss rates.

The difference is rather profound: in a non-paced Internet, the short term maximum packet rates are largely governed by sender's interface rates, which are perhaps only an order of magnitude slower than individual strands in wide area aggregated links. In a paced (BBR) Internet, the senders match bottlenecks near the receivers and the packet rates for individual flows are likely to be between 3 and 6 orders of magnitude below individual wide area strands. At some timescales, this traffic is vastly smoother than Markov.

As a community, we should proactively engage and foster this evolution. Rather than prescribing impractical buffer sizes based on current traffic, we should focus on the opposite question: Given the set of strongly bound maximum cost effective buffer sizes: what might be done in Internet hosts and elsewhere in the network to make the Internet function as well as possible with buffers that are smaller than we would prefer?

There are huge opportunities here!

Appendix: Potential Treatments and other Research Questions

Below is a partial list of engineering and research questions that might help to foster the evolution to a paced, short queue Internet.

I. How much does pacing help by itself, with no other changes?
    A. How much of the pain comes from self clocked protocols reflecting bursts and ACK aggregation back into the network?
        1. How can we observe this across the breadth of the Internet?
        2. How much gain from eliminating line rate bursts after stretch/aggregated ACKs?
II. What does "perform better" mean?

A. Packet loss is one obvious metric.
B. Bulk Transport Rate (single stream rate) in large aggregate
C. Maximum Aggregate utilization
    1. How do you control the load?
D. Jitter to non-protected flows (best effort)
    1. We expect less jitter QoE for best effort applications
E. Tail losses and transaction completion times
F. Are there others?

III. Protocol clocking
A. Are today's measures of traffic burstiness really measures of today's networks ability to deliver bursts?
    1. Self Clocked flows become more bursty, until they experience a loss
B. How much does pacing help?
    1. Without BBR (e.g. CUBIC + tcp_fq)?
    2. With BBR?

IV. Application smoothing e.g. Making transactions less abrupt
A. How much gain from eliminating line rate restarts?
B. Any advantage to going smoother than on/off at the known bottleneck rate (or some function of the bottleneck rate)?
C. Any advantage to short timescale rate variances, e.g. sub-rtt bandwidth or loss probing, chirping, etc?

V. Load regulation
A. What is the relationship between burst statistics, buffer size, load and loss rate?
    1. See prior papers on ATM effective bandwidth
    2. This question alone could fill 100% of the agenda
    3. There is or easily can be data on this.
B. What can be done outside the switches to regulate load
    1. e.g. Traffic Engineering, load balancing, admission control, pacing.
    2. What else?
C. BBR dovetails nicely with low threshold ECN in switches
    1. Active area with ECN thresholds that are queue occupancies
        a) DCTCP, L4S, BBRv2, etc
D. If bandwidth is vastly cheaper than buffers, can idle time be managed instead?
    1. e.g. ECN thresholds that are utilizations, instead of queue depth
    2. Conjecture: underfilling links by a few percent is far cheaper than the buffers needed to run well at 100% utilization. Prove or disprove...
    3. Can we use DCTCP style ECN with phantom or virtual queues?
    4. Lots of new research to explore here.....
E. Are we approaching a point where we can define explicit bounds on CC signals?
    1. What should be the maximum loss rate that CC can cause or tolerate?
    2. What should be the maximum ECN marking rate that CC can cause?
    3. What is the right amount of jitter in best effort traffic, and why?

VI. Better ECMP (packets from 1000s of flows are interleaved):

A. Relax the no reordering requirement at small time scales?
B. Can we eliminate flow pinning in the fastest devices?
    1. Does this move the cost/performance point for the entire Internet?

VII. Incentives

A. Can traffic sources with abusive statistics be identified?
    1. Can a sniffer detect sources emitting bursty traffic?
    2. Can a counter based instrument detect sources emitting bursty traffic?
    3. The problem is likely to be very different at different time scales
B. Can we construct a pricing model for buffer space
    1. Could ISPs charge more for unsmoothed self clocked traffic?
C. Can we apply incentives to abusive traffic?
    1. Premium charging for wasting buffer space
        a) E.g. any flow with >1 packet in a queue at overflow

VIII. Although BBR comes across as a monolithic fait accompli, it is a long way from being done. In [4] we argue that since BBR deprecates many of the principles in Jacobson 88, much of the last 30 years of congestion research needs to be revisited. Furthermore many congestion control algorithms that worked well in some environments but never attained significant deployment deserve to be ported to BBR's rate based framework, and reevaluated.

References

[1] Eric Dumazet, Yuchung Cheng. 2013. TSO, fair queuing, pacing: three's a charm. Presentation to IETF/TCPM. Retrieved from https://www.ietf.org/proceedings/88/slides/slides-88-tcpm-9.pdf

[2] A. Saeed, N. Dukkipati, V. Valancius, V. Lam, C. Contavalli and A. Vahdat. 2017, Carousel: Scalable traffic shaping at end hosts. Proc. of ACM SIGCOMM '17. DOI: https://doi.org/10.1145/3098822.3098852

[3] Neal Cardwell, Yuchung Cheng, C. Stephen Gunn, Soheil Hassas Yeganeh, and Van Jacobson. 2016. BBR: Congestion-Based Congestion Control. Queue 14, 5, Pages 50 (October 2016). DOI: https://doi.org/10.1145/3012426.3022184

[4] Matt Mathis, Jamshid Mahdavi. 2019. Deprecating The TCP Macroscopic Model. SIGCOMM Comput. Commun. Rev. 49, 5 (October 2019) https://ccronline.sigcomm.org/2019/ccr-october-2019/deprecating-the-tcp-macroscopic-model/