Bufferbloat: Dark Buffers in the Internet

Jim Gettys • *Bell Labs*

We have conflated "speed" with "bandwidth."

As Stuart Chesire wrote in "It's the Latency, Stupid" (http://rescomp.stanford.edu/~cheshire/rants/Latency.html), "Making more bandwidth is easy. Once you have bad latency, you're stuck with it."

Bufferbloat is the existence of excessively large (bloated) buffers in systems, particularly network communication systems. Bufferbloat is now (almost?) everywhere.

Today's routers, switches, gateways, broadband gear, and so on have bloated buffer sizes to where we often measure latency in seconds, rather than microseconds or milliseconds.

Telephone standards for maximum desirable latencies are in the 150–200 ms range, and human perception for some latency is as low as 10 ms. You can never get that time back. Any *unnecessary* latency beyond the minimum imposed by the speed of light is too much.

Although some buffering is required to smooth bursts in communications systems, we've lost sight of fundamentals: packet loss is (currently the only) way to signal congestion in the network, and congestion-avoiding protocols such as TCP rely on *timely* congestion notification to regulate their transmission speeds.

What happens when we put large or truly bloated buffers into our systems, in a misguided attempt to avoid all packet loss, or when we aim to eke out almost unmeasurable increases in performance on an artificial benchmark, or just because the buffer memory doesn't cost us anything and happens to be there?

Really bad things happen (see http://gettys.wordpress.com/2010/12/06/whose-house-is-of-glasse-must-not-throw-stones-at-another/), as John Nagle's cogent explanation, RFC 970 (from 1985!), describes:

> A datagram network with infinite storage, first-in-first-out queuing, and a finite packet lifetime will, under overload, drop all packets.

Some of the buffers we now observe in the Internet are effectively infinite in size. More is not necessarily better. More is often worse.

Not all packet loss is evil: some packet loss can be essential for correct operation. But once your bloated buffers fill, there's no *timely* congestion notification by packet loss or explicit congestion notification (ECN), and eliding notification has destroyed the congestion avoidance servo loop in transport protocols. Only the buffers on either side of the bottleneck (lowest bandwidth) link fill, and if those buffers are not managed, they can and do fill completely, inducing much higher packet loss than that you attempted to avoid. Other buffers in the path, remaining nearly empty, remain dark and undetectable.

Bufferbloat induces painful latencies for you and all others who share your network path. Any application that saturates a link with bloated buffers can induce bufferbloat pain: uploading videos to YouTube, emailing messages with large images attached, backing up large files or file systems, downloading large files, such as ISO images, a Linux distribution image, or a movie via Bittorrent, watching Netflix, and even visiting certain kinds of webpages can all fill these buffers.

Any semblance of interactivity of your network is gone; any hope for good teleconferencing or voice over IP, or fragging your opponent before they frag you is lost. Even Web browsing becomes painful, and applications often fail entirely.

cont. on p. 3

 1089-7801/11/$26.00 © 2011 IEEE

cont. from p. 2

Wonder no more why your network connections are so poor. This is why the "Internet is slow today" refrain is so common.

With modern TCP stacks (almost everything except Windows XP), even a single TCP transfer can induce bufferbloat suffering. The problem is not limited to TCP; UDP-based protocols are equally capable of filling bloated buffers. But you never see these buffers until they start to fill, and you can observe them only indirectly; they are "dark buffers" — like dark matter in the universe.

Last year the ICSI Netalyzr group proved that our broadband Internet technologies — cable, DSL, and FIOS alike — suffer badly from bufferbloat. And bufferbloat isn't confined to these technologies, but has also infected our home routers, 3G networks, and even our operating systems (see http://conferences.sigcomm.org/imc/2010/papers/p246.pdf).

For example, Linux typically has at least two major contributors to bufferbloat — the network stack's transmit queue and the ring buffers in the device driver — but several more places exist where buffers can hide. These buffers are often hundreds of packets in size on modern hardware, and we can find such large transmit rings on similar hardware on other operating systems. They might lurk in line cards in network gear, in modems, or elsewhere — the hiding places are endless.

Bufferbloat also infects many of our corporate and ISP networks. I believed we had solved congestion problems for Internet routers with the development of active queue management (AQM) algorithms such as random early detection (RED; see www.icir.org/floyd/papers/early.pdf), but the cottage industry of more than 100 papers on tuning RED proves this belief incorrect. Classic RED 93 can't solve our wireless problems. Although failing to use AQM when possible might be misguided, all those RED tuning papers help us understand why some network operators (both corporate and public) do not trust RED and are reluctant to enable it.

Won't adding more bandwidth help? Usually not. Buffering has been growing, frequently faster than bandwidth, over generations of often upward compatible technologies. Plug a current device into a previous generation link, and your buffers become insanely large, even in the rare case those buffers were static sized "correctly." They will now be sized for maximum theoretical bandwidth over maximum latency paths, often much larger than you will ever experience. Yet the actual bandwidth available varies, often by orders of magnitude. This demonstrates that a single static answer seldom exists regarding the correct buffer size in any system.

Adding bandwidth can even make your suffering worse: for example, if you have more broadband bandwidth than 802.11 bandwidth, the bottleneck shifts to that hop, where your laptop and home router bufferbloat is often even worse than in the broadband link. Now those "dark buffers" cause your pain.

We must systematically stamp out bufferbloat wherever it occurs in our systems by managing buffers at all times wherever they. We can mitigate the worst bufferbloat by eliminating the grossly mis-sized static buffers, but actual solutions require serious work, further research, and by the use of some form of AQM, in its most general sense.

We're also rapidly destroying TCP slow start, with independent changes by both Web server and Web browser. I even fear for the Internet's stability.

We have a large mess on our hands that spans hardware, software, firmware, operating systems, home routers, broadband, 3G, 802.11, and just about everywhere I have looked. We're all in this *bloat* we built together, and had better work together to row to shore quickly. Dark storm clouds surround us. Only together will we shine a light on all the dark buffers hidden in the Internet. I've been drawing together experts across all these problems at bufferbloat.net — please help out.

Jim Gettys is a member of technical staff at Bell Labs. He cannot possibly acquire as many grey hairs as Vint Cerf, having torn most of them out while chasing bufferbloat. Thanks go to Vint for the opportunity to write this issue's column. Contact Jim at jg@freedesktop.org.