Buffer Bloat

The term "Bufferbloat" was coined by Jim Gettys, when spending much effort on attempts to understand why his home network was performing so poorly (1.2 seconds latency plus horrible jitter).
Using tools like Smoke Ping and ICSI Netalyzr he found that all these major problems were caused by excessive buffering in the network.

It has long been known that overly large buffers can lead to problems; an early description of some of them is John Nagle's RFC 970: *On Packet Switches With Infinite Storage* from 1985. Jim Gettys' hypothesis is that some recent developments have sneakily conspired to make this a common occurrence in practice, and that it is time to do something about this.

Nowadays due to cheaper and cheaper memory, buffering is overdone in many network devices. While some buffering is needed in order to reduce loss, big buffers lead to higher delay and reduced throughput. Considering that TCP's congestion-avoidance mechanisms rely on packet drops to detect congestion, a big buffer will be flooded with packets before any drops actually happen. And the more packets are in the queue, the higher is the latency given that once the buffer is filled it takes time to drain and let in other packets. As a result, services that require low latency, such as network gaming, VoIP, or chat programs, can slow to the point of becoming unusable.

To solve this problem, there is no single right solution. Buffer size cannot be configured on most routers and switches. Moreover, the excessive buffer might not be a problem inside the network, while it certainly is at the edge of the network including operating systems, home routers and broadband gear. A small buffer can't be the right solution because when the queue fills an increased loss rate is generated, which shows up in timeout-driven retransmissions. A better solution would be to have a deep buffer (especially needed for bursts and temporary surges) that is kept shallow using AQM techniques.

Achievements

One can look at Bufferbloat as a performance issue, but it is also useful to look at Bufferbloat as a *movement* that was started by Jim Gettys and a few allies to make progress against this issue. Within the space of a few years, this movement has been relatively successful in raising awareness of the (potential) issue, and catalyzing technical work both on the measurement side and on the (OS) implementation side.

Public Awareness

Maybe most importantly, "Bufferbloat" has quickly become a well-known concept in the network engineering community. It also finding its way into course curricula.

Demonstrations of the Bufferbloat problem and mitigation mechanisms such as CoDel/fq_codel have been shown at events such as the *IETF-86 "Bits-n-Bytes" tutorials* (video).

Advances in Active Queue Management

CoDel and `fq_codel`

Based on the classic RED queue-management approach, Kathleen Nichols and Van Jacobson proposed a new AQM technique called CoDel, which moves the drop decision to dequeue time so that it can use packets' dwelling time in the queue as a drop criterion. This makes the algorithm robust to variations in service rate (output speed); something that has become very important because modern link technologies, especially in the wireless world, can have widely varying service rates depending on quality and occupancy of the wireless medium. CoDel has recently been implemented in the Linux kernel and thus is, or will become, available to many home router vendors who use that as the basis of their OSes.

Along with basic CoDel, `fq_codel` was added to the Linux kernel. `fq_codel` combines CoDel with a variant of fair queueing. The intent of `fq_codel` is that it can be applied to a bottleneck interface without any further configuration, and keep queueing delays low, especially for real-time traffic and other "sparse" flows, while still providing high throughput for bulk transfers. Empirical studies suggest that it does so successfully for a wide range of interface speeds, with low computational overhead.

PIE

A group of authors from Cisco Systems recently proposed PIE (Proportional Integral Controller Enhanced) as a more implementation-friendly alternative to CoDel.

CeroWRT

An important crystallization point for Bufferbloat activists is CeroWRT, a derivation of the OpenWRT Linux-based embedded system that is very popular for network devices such as wireless home routers. Bufferbloat-related improvements to the Linux kernel are included very quickly, and the configuration interface was modified to allow tuning for lower latency. Some of the improvements have been picked up by the "mainstream" OpenWRT system.

Other Technical Improvements

Network Transmit Queue Limits and in particular *Byte Queue Limits* were introduced in the Linux kernel in 2011.

TCP Small Queues were added to the Linux 3.6 kernel.

Recent revisions of the important DOCSIS (Data Over Cable Service Interface Specification) 3.0 standard include a feature called "Upstream Buffer Control".

BBR TCP is a congestion control mechanism for TCP that strives to avoid bufferbloat by using *pacing* based on estimates of bottleneck bandwidth and RTT.

Useful Links and Reading

- BufferBloat project
- Jim Gettys' blog - Whose house is of glasse must not throw stones at another, December 2010
- Jim Getty's blog - The criminal mastermind: bufferbloat!, December 2010
- acmqueue article - Bufferbloat: What's Wrong with the Internet?, a discussion with Vint Cerf, Van Jacobson, Nick Weaver, and Jim Gettys, ACM Queue, December 2011
- acmqueue article - Bufferbloat: dark buffers in the internet, Jim Gettys and Kathleen Nichols, ACM Queue, January 2012. There's a Google Tech Talk with the same title from June 2011, which includes a good introduction by Vint Cerf.
- On Packet Switches With Infinite Storage, RFC 970, John Nagle, 1985
- Bufferbloat and Other Internet Challenges, Vint Cerf, IEEE Internet Computing, Vol. 18 Issue 5, September/October 2014
- Bufferbloat and Beyond—Removing Performance Barriers in Real-World Networks, Toke Høiland-Jørgensen, Ph.D. Thesis, November 2018

– Alessandra Scicchitano - 2012-06-12
– SimonLeinen - 2013-03-11—2018-11-23