DOI: 10.1145/3690840

BY DAVID COLLIER-BROWN

You Don't Know Jack about Bandwidth

If you are an ISP and your customers hate you, take heart: This is now a solvable problem.

Imagine you are a company with a lot of remote employees, and they all hate the local Internet service providers (ISPs). Videoconferences are the worst: People cannot hear each other, they randomly start sounding like Darth Vader, and they occasionally just disappear from the conversation.

Or you are a small ISP, and your *customers* say they hate you.

When you talk to your ISP or supplier, they say, "Buy more bandwidth." When your customers complain to their ISP, they are told the same thing. But when you measure how much bandwidth you are using at the busiest part of the day, it is 30% or 50%.

In a sense, the suppliers are right: Bandwidth really boils down to the "diameter of the pipe" between you and your employees or customers. If it is too small your data, connections, and more will really get bogged down.

Once you are not filling the pipe, though, when you are only 50% busy, you need to look at the software and see what is keeping your packets from arriving in a reasonable amount of time. For example, I once measured the time to send a "ping" to downtown Toronto from my home office in the suburbs. It took 0.13s to get downtown and back. That is the normal ping time for Istanbul, Turkey, roughly 8,000 km away.[a]

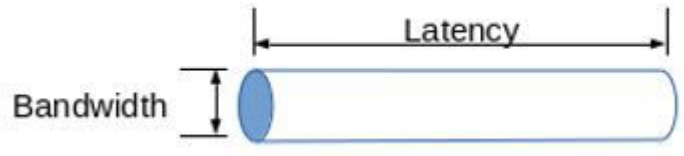
That is a pure software problem, and it is called *high latency*.

Bandwidth Is How Much, Latency Is How Slow

If you think of the Internet as a series of pipes, then bandwidth conveys how much data you can send down the pipe. It is the diameter, in Figure 1.

If you buy a narrow, little pipe, you absolutely will have bad performance: You will not be able to get all your data through it in any reasonable amount of time. You will have starved yourself. If you

Figure 1. Bandwidth versus latency.



[a] See https://bit.ly/3XBlvde

buy a pipe big enough to contain your data, you are fine. The data will all flow straight down the pipe instead of having to sit around waiting for the previous bytes to squeeze their way through.

However, buying a pipe with a larger diameter than you need will not help. Your data will still flow into it without delay, but it will not go any faster. It will not exceed the speed of light. There isn't a "warp speed" available for the Internet.

Why Is It Slow?

Latency is how quickly (or slowly) data shows up. The least latency you can have equals the length of the pipe divided by the speed of light. Consider an outgoing Zoom call, comprising a bunch of short snippets of picture and sound, nicely compressed, with a period of time between each, as shown in Figure 2.

Each little slice of audio and video shoots through the pipe by itself, leaving room for other slices. For a Zoom call to fail, you need a lot of other traffic at the same time, enough to fill up the pipe. If you are working from home, this could mean someone in your family is streaming a movie or uploading photos from a cell phone.

That causes *contention*, shown in Figure 3, where the photos (the big, long slice at the top) elbow their way into the pipe ahead of the smaller Zoom slices.

If you have bad router software, the smaller Zoom slices will have to sit around in a queue (buffering) until the photos finish. Because the Zoom slices are delayed, other people on the Zoom call may see you freeze or stutter. Sometimes you will sound like you are shouting from the bottom of a well. On really bad days, you will just drop out.

Bad Router Software

In the home, the problem is in the software on the home router, which connects to the wider Internet. The bug is called *bufferbloat,*[1] as described by Phillipa Harrison in the *Communications* research article, "Buffer-Bloated Router? How to Prevent It and Improve Performance."[2] In this bug, the sender buffers up the photos and as soon as the network is available, sends them all. While doing that, it cannot send the videoconferencing slices, so it delays them and sends them only after the photos complete.

Figure 2. Latency of Zoom call.

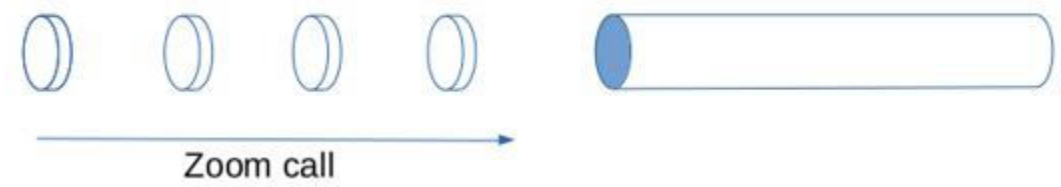
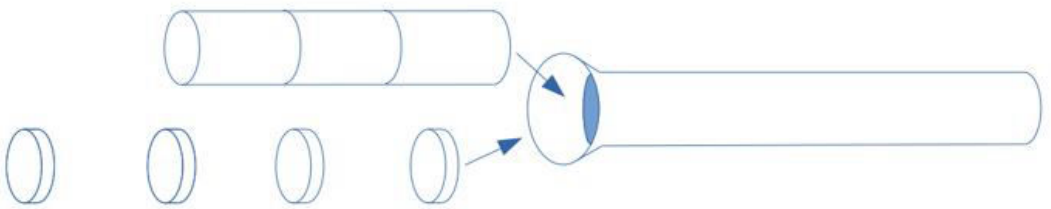


Figure 3. Contention.



That might be fine for computers, but humans discussing something need their words to come in the right order, without delays or interruptions. The photos should share the network fairly with the videoconference and not steal all the bandwidth.

Good Router Software

In the best of all possible worlds, every router on the Internet would be running software that has the latest changes to minimize latency, usually fq_codel or CAKE. Unfortunately, not everyone can update as often as they would like, and router manufacturers are often years behind on rewriting their proprietary operating systems to include the newest fixes. Some are even back at PIE, a much older predecessor to CAKE.

Home modem/router/Wi-Fi boxes are, if anything, worse. They get built, used, and thrown away when they die. They never get updated. The net result is a lot of old, buggy software on the Internet, and a lot of packets sitting in queues twiddling their thumbs as they wait for a chance to be sent.

Individuals took control of the problem by adding a smarter router between their home and their ISP's modem/router/Wi-Fi box. The smart router keeps data flowing as fast as the pipe will allow, sending "I'm having contention" signals that the older software can understand when any of the machines in your home network send too much.

Fixing the ISP

Adding devices to the home does not scale. If you are a company, you cannot afford to buy a new home router for each of your employees. If you are an ISP, you cannot just tear out painfully expensive core routers to install newer ones.

The answer is not to tear out routers but to adapt the techniques used in the home. If your customers hate you, put a device with modern software in the path to your customers, downstream of your routers. Such a device is dedicated only to dealing with buffering, bloat, and contention. That also keeps the price down, because an older single-socket Xeon with eight cores can handle 10Gbps of traffic.

The name of the software for the ISP device is LibreQoS (https://libreqos.io/), where *QoS* means *quality of service*. It came out from the same team that pioneered the fq_codel and CAKE software that addressed the problem at the home-router end.

They viewed the problem from the Internet end and set out to help ISPs and other companies that could not trivially buy more bandwidth, but instead had to make better use of what they had.

Their solution is to apply a complete suite of fixes to IP networking, which includes:

- Fair queuing
- Bandwidth shaping
- Active queue management
- Diffserv handling
- Ack filtering
- Network topology awareness
- Parallel processing
- Bypassing Linux bridge overheads
- Built-in round-trip time measurement to show the improvement

All these fixes work together to find the highest rate that does not overload the slowest part of the system. That means packets do not have to sit in a queue at the ISP, waiting for a chance to be sent, like the videoconferencing packets waiting at home for an image to finish uploading.

The first fixes are fair queuing and control of delay, from fq_codel.

CAKE then added Active Queue Management (AQM), which performs the same kind of bandwidth probing that vanilla TCP does but adds in-band congestion signaling to detect congestion as soon as possible. The transmission rate is slowly raised until a congestion signal is received, then reduced. Once reduced, it starts increasing again, continuously searching for the highest rate it can use without causing a queue to build up. The algorithm used is an extension of fq_codel.

The fairness component prevents particular flows from being starved of resources by other streams, like our photo example, or by hosts starting large numbers of streams to try stealing as much of the network bandwidth as they can.

Diffserv is a mechanism for labeling packets as interactive (high priority) through bulk (low priority), so bulk file transfer can be recognized and kept from delaying videoconferences.

Related to this, but not part of diffserv itself, is an algorithm to detect and properly prioritize sparse flows, so they will not be starved.

Ack filtering entirely removes packets that are merely carrying an acknowledgment of received data, so long as there are enough non-empty packets flowing the same way to take over that task.

Next, we come to LibreQoS itself, which adds components specific to running at the ISP end of the network. The first component is a network topology tree, because an ISP will have many routes to its customers, each with its own maximum rate that must be discovered and processed by its own instance of the flow-shaping code.

On top of this sits a hierarchical token-bucket classifier that sorts the incoming packets into the correct CPU core and the correct "shaper" rule. The standard Linux shaper that CAKE uses is single-threaded and tends to run out of capacity above 6Gbps. The LibreQoS shaper is limited only by the number and speed of the CPU cores and Ethernet cards.

LibreQoS 1.4 introduced a new fast Ethernet bridge, based on eXpress Data Path (XDP) and extended Berkeley Packet Filter (eBPF), to avoid bottlenecking on the `ksoftirqd` thread in the kernel. When they were finished with the performance work, the LibreQoS crew added software to measure and manage the improvement: a new low-overhead *round-trip time sensor*, a packet-capture system, and a suite of real-time programs to measure and visualize the performance of large numbers of routes and customers.

This is a mature technology and runs on standard equipment, such as an ISP might already have. As a concrete example, an ISP delivering 1Gbps service plans to its customers and with up to 10Gbps total throughput would need a CPU such as an Intel Xeon E-2388G with eight cores, 16GB of memory, and a dual-port 10G Ethernet card—a very ordinary 1u-sized machine from 2021.

Now a company with bad performance can ask its ISP to fix it and point at the software and people who have already used it. If the ISP already knows it has a performance complaint, it can get ahead of the problem by proactively implementing LibreQoS.

Proof

If you have already solved the problem with LibreQoS, you can see the fix happening in real time. In Figure 4, a customer is downloading Diablo 2 and Diablo 4 at the same time over a 100Mbps link. The link shows as completely saturated in the throughput graph, but the delay is tiny: 1.7ms with peaks to 3.1. Fair queuing keeps the delay below levels that humans or Zoom can detect.

It can also be measured from a home office if you are one of the happy people who have a debloated router. Using the Waveform Internet speed test,[b] my untreated connection to a local ISP scored a "D" (Figure 5) on its report card: not suitable for video gaming, videoconferencing, or even audio-only VoIP calls. And ***badly*** is exactly how it behaved until I put a debloated home router in front of the ISP's device.

[b] See https://bit.ly/3XBlvde

Conversely, that same 150/15Mbps connection with a debloated router earned an "A+" grade, a less-bloated speed report, and in practice has been delivering excellent videoconference performance for several years.

Conclusion

Bandwidth probably is not the problem when your employees or customers say they have terrible Internet performance. Once they have something in the range of 50 to 100 Mbps, the problem is latency—how long it takes for the ISP's routers to process their traffic.

If you are an ISP and all your customers hate you, take heart. This is now a solvable problem, thanks to a dedicated band of individuals who hunted it down, killed it, and then proved out their solution in home routers.

Figure 4. Download metrics.

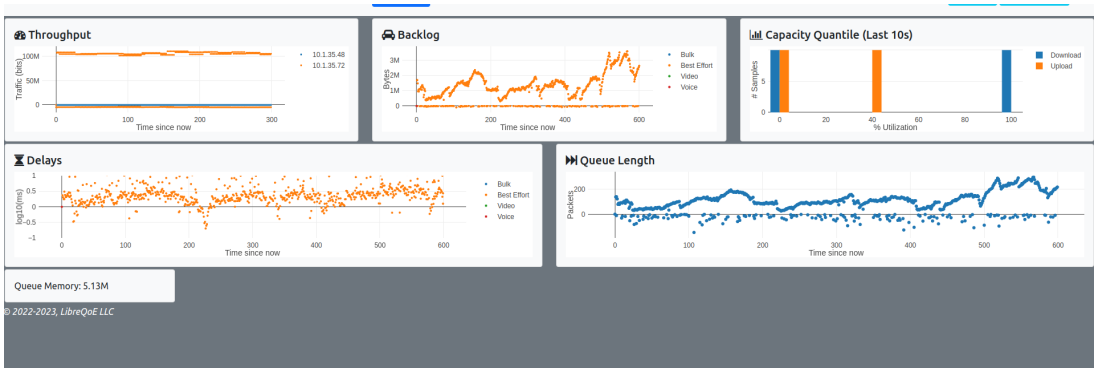
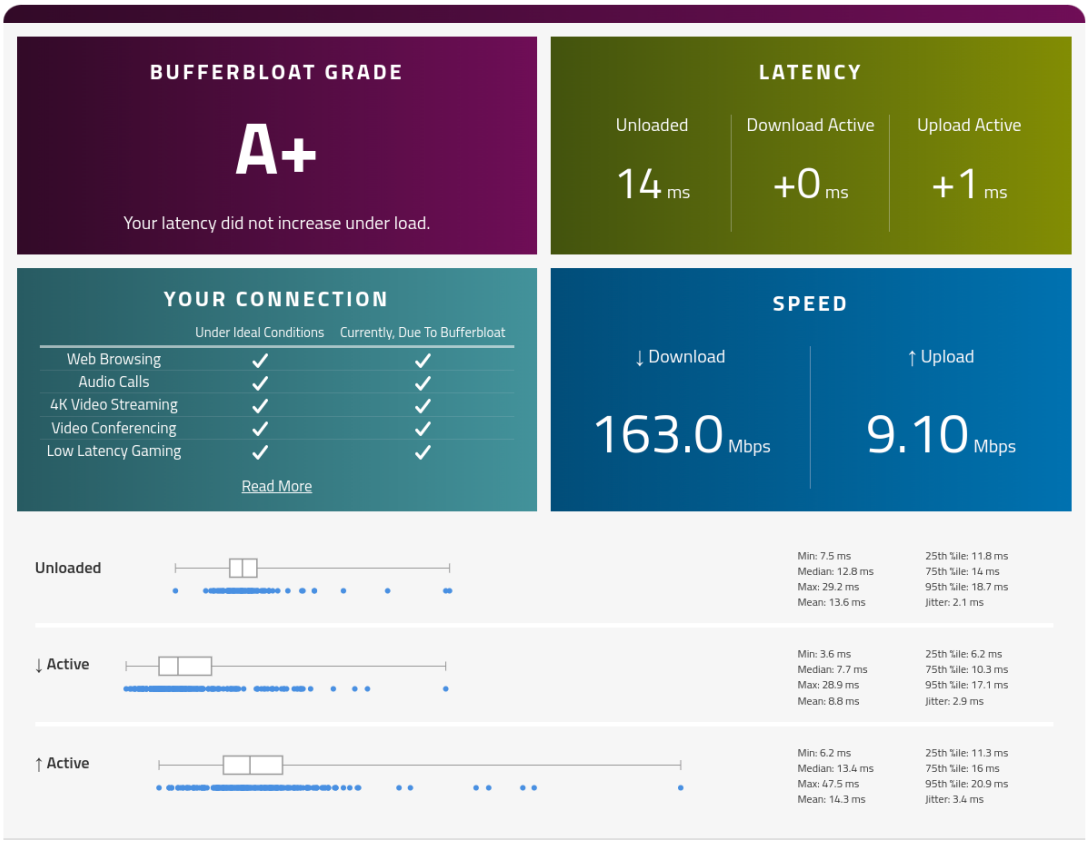


Figure 5. Waveform Internet speed test.



References

1. Gettys, J. and Nichols, K. Bufferbloat: Dark buffers in the Internet. *ACM Queue 9*, 11 (2011); https://bit.ly/3MSIWK0.
2. Harrison, P. Buffer-bloated router? How to prevent it and improve performance. *Communications 66*, 6 (2023), 73–77; https://bit.ly/4gwfxTB.
3. Høiland-Jørgensen, T. Bufferbloat and beyond: Removing performance barriers in real-world networks. *Doctoral thesis*. Karlstad University Studies (2018); https://bit.ly/3Xr9PK9.

David Collier-Brown is an author and systems programmer, formerly with Sun Microsystems, who mostly does performance and capacity work from his home base in Toronto, Canada.