


NETWORKS

How TCP's congestion control saved the internet

We guess it's OK it did

 [Bruce Davie](#)

Sun 24 Sep 2023 // 14:15 UTC

SYSTEMS APPROACH With the annual SIGCOMM conference taking place this month, we observed that congestion control still gets an hour in the program, 35 years after the first paper on TCP congestion control was published. So it seems like a good time to appreciate just how much the success of the internet has depended on its approach to managing congestion.

Following my recent [talk](#) and [article](#) on 60 years of networking, which focused almost entirely on the internet and ARPANET, I received quite a few comments about various networking technologies that were competing for ascendancy at the same time.

These included the OSI stack (anyone remember CLNP and TP4?), the Coloured Book protocols (including the Cambridge Ring), and of course ATM (Asynchronous Transfer Mode) which was actually the first networking protocol on which I worked in depth. It's hard to fathom now, but in the 1980s I was one of many people who thought that ATM might be the packet switching technology to take over the world.

ATM proponents used to refer to existing technologies such as Ethernet and TCP/IP as “legacy” protocols that could, if necessary, be carried over the global ATM network once it was established. One of my fond memories from those days is of Steve Deering (a pioneer of IP networking) boldly (and correctly) stating that ATM would never be successful enough to even be a legacy protocol.

One reason I skipped over these other protocols when earlier retelling the history of networking was simply to save space – it's a little-known fact that my Systems Approach colleague Larry Peterson and I aim for brevity, especially since receiving a [one-star review](#) on Amazon that called our book “a wall of text.” But I was also focused on how we got to the internet of today, where TCP/IP has effectively out-competed other protocol suites to achieve global (or near-global) penetration.

There are many theories about why TCP/IP was more successful than its contemporaries, and they are not readily testable. Most likely, there were many factors that played into the success of the internet protocols. But I rate congestion control as one of the key factors that enabled the internet to progress from moderate to global scale.

It is also an interesting study in how the particular architectural choices made in the 1970s proved themselves over the subsequent decades.

Distributed resource management

In David Clark's paper [\[PDF\]](#) "The Design Philosophy of the DARPA Internet Protocols," a stated design goal is: "The internet architecture must permit distributed management of its resources." There are many different implications of that goal, but the way that Jacobson and Karels [\[PDF\]](#) first implemented congestion control in TCP is a good example of taking that principle to heart.

Their approach also embraces another design goal of the internet: accommodate many different types of networks. Taken together, these principles pretty much rule out the possibility of any sort of network-based admission control, a sharp contrast to networks such as ATM, which assumed that a request for resources would be made from an end-system to the network before data could flow.

Part of the "accommodate many types of networks" philosophy is that you can't assume that all networks have admission control. Couple that with distributed management of resources and you end up with congestion control being something that end-systems have to handle, which is exactly what Jacobson and Karels did with their initial changes to TCP.

The history of TCP congestion control is long enough to fill a book ([and we did](#)) but the work done at Berkeley from 1986 to 1998 casts a long shadow, with Jacobson's [1988 SIGCOMM paper](#) ranking among the most cited networking papers of all time.

Slow-start, AIMD (additive increase, multiplicative decrease), RTT estimation, and the use of packet loss as a congestion signal were all in that paper, laying the groundwork for the following decades of congestion control research. One reason for that paper's influence, I

believe, is that the foundation it laid was solid, while it left plenty of room for future improvements—as we see in the continued efforts to improve congestion control today.

And the problem is fundamentally hard: we're trying to get millions of end-systems that have no direct contact with each other to cooperatively share the bandwidth of bottleneck links in some moderately fair way using only the information that can be gleaned by sending packets into the network and observing when and whether they reach their destination.



As fans of decentralization, we've moved our social media home to [Mastodon](#). If you need any further reason to get off X/Twitter, [this](#) may help. And you can [read](#) how Mastodon is "rewinding the clock on social media — in a good way."

Meanwhile, Emily Bender has a new paper [[PDF](#)] on the role of LLMs in information retrieval, making a compelling case that generative AI may not be what you want in a search engine.

And given our interest in network verification, we were pleased to see this paper on experience with Batfish at [SIGCOMM](#) this month.



Arguably one of the biggest leaps forward after 1988 was the realization by Brakmo and Peterson (yes, that guy) that packet loss wasn't the only signal of congestion: so too was increasing delay. This was the basis for the 1994 [TCP Vegas paper](#), and the idea of using delay rather than loss alone was quite controversial at the time.

However, Vegas kicked off a new trend in congestion control research, inspiring many other efforts to take delay into account as an early indicator of congestion before loss occurs. Data center TCP ([DCTCP](#)) and Google's [BBR](#) are two examples.

One reason that I give credit to congestion control algorithms in explaining the success of the internet is that the path to failure of the internet was clearly on display in 1986. Jacobson describes some of the early congestion collapse episodes, which saw throughput fall by three orders of magnitude.

When I joined Cisco in 1995 we were still hearing customer stories about catastrophic congestion episodes. The same year Bob Metcalfe, inventor of Ethernet and [recent Turing Award winner](#), famously predicted that the internet would collapse as consumer internet access and the rise of the web drove rapid growth in traffic. It didn't.

Congestion control has continued to evolve, with the QUIC protocol, for example, offering both better mechanisms for detecting congestion and the option of experimenting with multiple

congestion control algorithms. And some congestion control has moved into the application layer, eg: Dynamic Adaptive Streaming over HTTP ([DASH](#)).

An interesting side effect of the congestion episodes of the 1980s and 1990s was that we observed that small buffers were sometimes the cause of congestion collapse. An [influential paper](#) by Villamizar and Song showed that TCP performance dropped when the amount of buffering was less than the average delay \times bandwidth product of the flows.

Unfortunately, the result only held for very small numbers of flows (as was acknowledged in the paper) but it was widely interpreted as an inviolable rule that influenced the next several years of router design.

This was finally debunked by the buffer sizing work of [Appenzeller *et al*](#) in 2004, but not before the unfortunate phenomenon of [Bufferbloat](#) – truly excessive buffer sizes leading to massive queuing delays – had made it into millions of low-end routers. The [self-test](#) for Bufferbloat in your home network is worth a look.

So, while we don't get to go back and run controlled experiments to see exactly how the internet came to succeed while other protocol suites fell by the wayside, we can at least see that the internet avoided potential failure because of the timely way congestion control was added.

It was relatively easy in 1986 to experiment with new ideas by tweaking the code in a couple of end-systems, and then push the effective solution out to a wide set of systems. Nothing inside the network had to change. It almost certainly helped that the set of operating systems that needed to be changed and the community of people who could make those changes was small enough to see widespread deployment of the initial BSD-based algorithms of Jacobson and Karels.

It seems clear that there is no such thing as the perfect congestion control approach, which is why we continue to see new papers on the topic 35 years after Jacobson's. But the internet's architecture has fostered the environment in which effective solutions can be tested and deployed to achieve distributed management of shared resources.

In my view that's a great testament to the quality of that architecture. ®



Larry Peterson and Bruce Davie are the authors of [Computer Networks: A Systems Approach](#) and the related [Systems Approach](#) series of books. All their content is open

source and available on [GitHub](#). You can find them on [Mastodon](#), their writings on [Substack](#), and past *The Register* columns [here](#).

