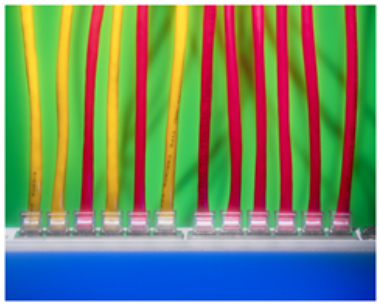
Is That An ACK In Your Packet, Or Are You Just Glad To See Me?

Don MacVittie, 2010-17-12



Every once in a while, I like to step back a bit and write for those who haven't been in the field for a zillion years. For starters, it helps refresh the pool of information out there for people trying to research something they haven't done before. It helps a lot that I enjoy sharing my knowledge, so writing such a blog is like "non-work". Since I'm gearing up for some holiday time, this seemed like a great time to do just such an article, so I cast about and TCP optimizations came to mind.

A lot has been written about TCP optimizations, this take will be for the beginner, and will cover them from the Application Delivery Controller (ADC) perspective. With a background in development, IT management, and storage, I had to learn this stuff the hard way, hopefully this helps some of you skip ahead a few squares in the "IT Learn Something New!" game. My knowledge leans heavily upon F5 gear, specifically BIG-IP LTM, but as usual, I try to stick to features and functionality common to ADCs. At least the big names in ADCs.

One of the very cool bits about an ADC is that most act as a full proxy between the LAN and the WAN. This opens possibilities that would not normally exist in a standard network configuration. The ADC can ack to the server at server speed, while spooling to send to the client at client speed. In many cases, this single possibility helps performance by its mere existence. But there is much more going on in a modern ADC. If you're interested in the deep-delve details along with RFC numbers to research, check out Optimizing WAN and LAN Application Performance With TCP Express on F5.com. It's getting older (well over three years), and is F5-centric, but by including the RFC numbers, the author has left you room to research.

For those who aren't crazy about reading through RFCs, here are some highlights of what you can hope to get out of an ADC. As always, my knowledge is F5 centric, check with your vendor before assuming they've implemented all of these. All of these are turned on or configurable on a BIG-IP.

- Nagle's Algorithm. This pools data until the receiver has ACK'd what has already been sent. By doing so, it sends less packets because it's packing data waiting for ACKs. While this can make it appear that latency has increased, it does generally result in less packets on the wire.
- Dynamic Window Sizing (Including Slow Start). This adjusts the data window size to suit what's on the other end. By doing so, the client can have one window size and the server another, each optimized to the network conditions it is seeing and the way its TCP stack is optimized. Normally the two would have to negotiate this to be the minimum.
- Adaptive Initial Congestion Windows and TCP Slow Start With Congestion Avoidance. These simply change how fast initial Slow Start is handled, so that some connections get to the proper window size quickly.
- Bandwidth Delay Control. An automatic calculation of how much data can be put into a link without overloading it.
- TCP Congestion Avoidance. A set of standards to avoid and recover from lost packets due to link congestion.
- Selective Acknowledgements and Limited and Fast Retransmits. When data is lost, this is a packet-based shorthand for recovery, cutting the time and retransmits required down.
- Connection pooling to servers. We draw an imaginary line in the sand and don't call this a TCP optimization, but it really is – it creates less TCP overhead on your server by putting multiple clients into one connection. Normally the server would open one (or more) for each connection, an ADC, sitting in the middle, can "pool" these connections into one, saving your server from setting aside resources for each individual client.

What does all of this mean? Well first off, these are not all of the possibilities, TCP has had a long history and lots of improvements have been suggested through the RFC system. Our engineers will likely grind their teeth that I distilled all of their hard work down to a few bullet points that don't even cover all the possibilities.

But the point is to help you understand why the simple act of putting an ADC into your network can improve application performance. If your server is communicating with the BIG-IP at its maximum speed, and the client is communicating with the BIG-IP at *its* maximum speed, things seem faster to the end user. Add in the ability to recover quickly on lossy networks, and the more remote the user, the more benefits they'll see. That's pretty cool. And it's free with your ADC. How much of it is free with your ADC, and how well it is implemented is going to be vendor dependent, but much of this stuff has been out there for years, so ask your ADC vendor, I'd be surprised if they told you "yeah, we don't do Nagle's algorithm" or "Congestion Avoidance? Congestion helps your packets get tougher, why would we want to avoid it?"

A modern ADC is a complex system. While implementing TCP and HTTP optimizations is a natural offshoot of what a load balancer does, it is certainly one of the hallmarks of an ADC that this offshoot has been incorporated into the product.

I reiterate that this is simply a starting point. There is lots of good information out there about TCP optimizations (starting with that PDF linked to above), and you can get right to it if you need it. This was just a toe-dip into a very complex world. No doubt I have simplified to the point that some experts will think I've over-simplified. If it piqued your interest though, then I did not oversimplify at all.

The answer to the title? If you have an ADC in your network, the answer is "Both". That IS an ACK in your server's packet, and since its workload is reduced, the server IS glad to see the ADC.

F5 Networks, Inc. | 401 Elliot Avenue West, Seattle, WA 98119 | 888-882-4447 | f5.com

F5 Networks, Inc.
Corporate Headquarters
info@f5.com

F5 Networks
Asia-Pacific
apacinfo@f5.com

F5 Networks Ltd.
Europe/Middle-East/Africa
emeainfo@f5.com

F5 Networks
Japan K.K.
f5j-info@f5.com