New Communications = Multiplexification

Don MacVittie, 2012-02-05

I wrote a good while back about the need to translate all the various storage protocols into one that could take root and simplify the lives of IT. None of the ones currently being hawked seem to be making huge inroads in the datacenter, all have some uses, none is unifying. Those peddling the latest, greatest thing of course want to sell you on their protocol because they hope to be The One, but it's not about selling, it's about useful. At the time FCoE was the new thing. I don't get much chance to follow storage like I used to, but I haven't heard of anything new since the furor over FCoE started to calm down, so presume the market is still sitting there, with NAS split between two, and block storage split between many.

There is a similar fragmentation trend going on in networking at the moment too. There have always been a zillion transport standards, and as long as the upper layers can be uniform, working out how to fit your cool new satellite link into Ethernet is a simple problem from the IT perspective. Either the vendor solves the issue or they fail due to lack of usefulness. But higher layers are starting to see fragmentation. In the form of SPDY, Speed + mobility, etc. In both of these cases, HTTP is being supplanted by something that requires configuration differences and is not universally supported by clients. And yet the benefits are such that IT is paying attention. IPv6 is causing similar issues at the lower layers, and it is worth mentioning here for a reason.

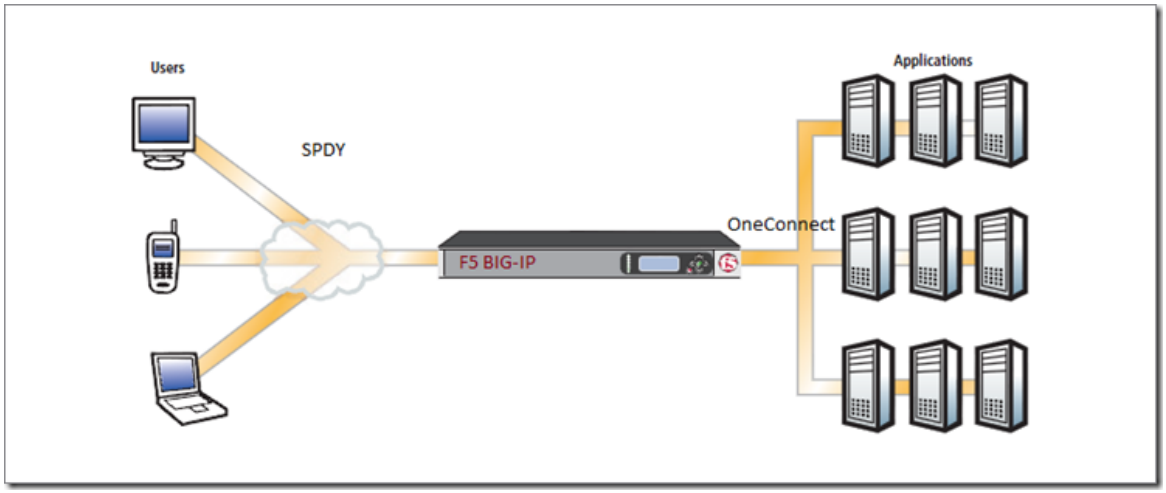
The key, as Lori and I have both written, is that IT cannot afford to rework everything at once to support these new standards, but feels an imperative (for IP address space from IPv6, for web app performance for the http layer changes) to implement them whenever possible. The best solution to these problems – where upgrading has its costs and failing to upgrade has other costs – is to implement a gateway. F5s IPv6 Gateway is one solution (other vendors have them too - I'll talk about the one I know here, but assume it applies to the others and verify that with your vendor) that's easy to talk about because it is being utilized in IT shops to do just that. With the gateway implemented, sitting in front of your DC, it translates from IPv6 to IPv4, meaning that the datacenter can be converted at a sane pace, and support for IPv4 is not a separate stack that must be maintained while client adoption catches up. If a connection comes in to the gateway, if it is IPv4 and the server speaks IPv4, the connection is passed through. The same occurs if both client and server support IPv6. If the client and server have a mismatch, the gateway translates between them. That means you get support the day a gateway is deployed, and over time can transfer your systems while maintaining support for all clients.

This type of solution works handily for protocols like SPDY too – offering the ability to say a server supports SPDY when in fact it doesn't, the gateway does and translates between SPDY and HTTP. Deploying a SPDY gateway gives instant SPDY support to web (and application) servers behind the gateway, buying IT time to reconfigure those web servers to actually support SPDY. SPDY accelerates everything on the client side, and http is only used on the faster server side where the network is dedicated. Faster has an asterisk by it though. What if the app or web server is at a remote site? You're going right back out onto the Internet and using HTTP unoptimized.

In those cases – and other cases where network response time is slow - something is needed on the backend to keep those performance gains without finding the next bottleneck as soon as the SPDY gateway is deployed. F5 uses several technologies to improve backend communications performance, and other vendors have similar solutions (though ours are better – biased though I may be). For F5's part, secure tunnels, WAN optimization, and a very relevant feature of BIG-IP LTM called OneConnect all work together to minimize backend traffic.

OneConnect is a cool little feature that minimizes the connections from the BIG-IP to the backend server by pooling and reusing them. This process does several things, but importantly, it takes setup and teardown time for connections out of the picture. So if a (non-SPDY) client makes four connections to get its data, the BIG-IP merges them with other requests to the same server and essentially multiplexes them. Funny thing is, this is one of the features of SPDY on the other side, with the primary difference that SPDY is client focused (merges connections from the client), and OneConnect is server focused (merges connections to the server). The client side is "all connections from this client", while the server side is "all connections to this server (regardless of client)", but otherwise they are very similar.

This enters interesting territory, because now we're essentially multi-multi-plexing. But we're not. Here's a simple diagram utilizing only a couple of clients and generic server/application farm to try and show the sequence of events:



1. SPDY comes into a gateway as a single stream from the client

2. The gateway translates into HTTP's multiple streams

3. BIG-IP identifies the server the request is for

4. If a connection exists to the server, BIG-IP passes the request through the existing connection

5. When responses are sent, this process is handled in reverse. Responses come in over OneConnect and go out SPDY encoded.

There is only a brief period of time where native HTTP is being communicated, and presumably the SPDY gateway and the BIG-IP are in *very* close proximity.

The result is application communications that are optimized end-to-end, but the only changes to your application architecture are configuring the SPDY Gateway and OneConnect. Not too bad for a problem that normally requires modification of each web and application servers that will support SPDY. As alluded to above, if the application servers are remote from the SPDY Gateway, the benefits are even more pronounced, just due to latency on the back end.

All the benefits of both SPDY and OneConnect, and you will be done before lunch. Far better than loading modules into every webserver or upgrading every app server.

Alternatively, you could continue to support only HTTP, but watching the list of clients that transparently support SPDY, the net result of doing so is very likely to be that customers gravitate to your competitors whose websites seem to be faster.

The Four V's of Big Data

The "All of the Above" Approach to Improving Application Performance

Google SPDY Accelerates Mobile Web

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