

The State of SDN – Have We Reached the Tipping Point?

That we're on the edge of widespread SDN (software-defined networking) adoption has been rumored for years, now. It seems there's always a new report on when SDN will catch on and how big of a success it will be. While these reports are usually over optimistic, they have me thinking about when the tipping point of SDN will be reached – when the momentum and notoriety the SDN movement is gaining will finally pass the threshold separating isolated experimentation from widespread adoption.

There have been innumerable reports predicting that SDN adoption is inevitable and will be inextricably integrated with the future of networking. IDC published a study of the SDN market earlier this year and predicted a 53.9% CAGR from 2014 through 2020, by which point the market will be valued at \$12.5b. To add to the excitement: the Technology Trends 2016 report ranked SDN as the best technology investment for 2016.

It's easy to get caught up in the excitement, but it's important to be suspicious of these numbers, as well. Just because the SDN market will be worth \$12.5b in 2020 doesn't mean it's worth much in 2016. In fact, if you interpolate from the numbers given and the CAGR, you'll find that the current valuation of the SDN market is roughly \$0.61b. \$610m is nothing to scoff at, but it suggests that SDN deployments are still fairly uncommon. That's why it's the best technology investment for 2016 – it'll grow.

SDN is still largely in the testing phase, being used predominantly in proof-of-concept projects. While that means it hasn't found much of a home in commercial projects, it also means that widespread adoption can't be too far behind. A lot depends on the success of the PoC trials, but it's worth noting that there are myriad other obstacles that SDN will have to overcome en route to its proliferation throughout enterprise networks.

Obstructions to SDN Adoption

Maybe the expectations for SDN deployments were too high early on. Hype trains lead to rampant optimism and misunderstanding, in this case leading people to believe that SDN was some kind of network elixir. I don't believe there's anything inherently wrong with optimism, but when optimism leads to blindness or unreasonable expectations, it's a problem.

SDN won't achieve anything by itself, but intelligently deploying it on top of the right infrastructure will give the adopter all of the benefits they expect. SDN deployments aren't easy – they require the rigorous testing of the software's interoperability on top of various network architectures. As you can imagine, that testing represents a significant barrier to entry. However, once SDN is used in a majority of networks as the result of this rigorous testing, the benefits for all SDN users will increase exponentially. SDN is struggling to wriggle out from the shadow its own hype was casting, but once expectations are tempered a bit, SDN should impress with its own merit.

Because SDN is largely still in the testing phase, there aren't many use cases to point to. This makes it difficult to convince potential adopters to integrate it with their own networks. Everyone knows that SDN works, but it's the lack of a concrete, fully-deployed use case that's impeding its proliferation.

The lack of a pure use case makes the benefits of SDN hard to prove. SDN primarily benefits network efficiency, while profits are mostly made from applications. It's obvious that the system handling the applications is just as important as the applications themselves, but the importance is less easy to measure than technology that directly benefits applications. Also hard to measure are the cultural obstructions to SDN adoption.

Most current network engineers are hardware operators and SDN requires software developers. That kind of disconnect is causing strife in the industry – engineers have to learn new skills or be displaced as SDN continues to gain acceptance. Network incumbents don't want to have to re-train all of their engineers, nor do they want to give up their market share. Major network vendors deal in both hardware and software, and switching out one piece of their vendor lock-in puzzle destroys the whole thing. It's in their interest to hold on to their proprietary stacks as long as they can, but even they are being pushed into developing for SDN's inevitable future.

Unfortunately, these same vendors are adding software to the SDN ecosystem that could be described as devoutly anti-standardization and pro-differentiation. We can talk about the various obstructions to SDN all we want, but the biggest issue is the lack of a single protocol or standard that will push people past arguing, experimenting, and differentiating, and into widespread deployment. The more SDN proliferates, the more standards will present themselves, but we could all save ourselves a lot of time if we could agree on an interoperable protocol early on.

How to Accelerate SDN Adoption

There's no question that, at some point, SDN will reach widespread adoption. It'll be deployed to reduce costs and increase revenue once the testing proves that it can. Any testing environment that proves either of these goals will go a long way toward accelerating the adoption process.

Service-oriented architecture can assist SDN's implementation by framing the debate around application delivery instead of the network itself. As I established earlier, since applications are the money makers, network improvements frequently come as an afterthought in the enterprise. By implicating the network whenever we discuss application delivery, we should be able to re-frame the discussion with an understanding of the network's importance.

Applications are increasingly programmable, meaning they can automatically communicate what they need from the network. If an app needs to be redirected, or if it needs sudden bandwidth or storage provisioning, the network should be able to react to that demand. SDN is capable of orchestrating this dynamic response. Enterprise network operators need to understand that, since applications are the money makers, they need the best network they can get if the business is to continue its success. And if SDN is really going to be a factor in the enterprise, there has to be some kind of coherent protocol that everyone can understand and work toward improving.

SDN needs a protocol like OpenFlow to manage the flow of traffic through the network, but nobody can seem to decide on a single protocol, or an interoperable protocol, to work with.

Eventually the protocols will standardize, but at the moment it's a sandstorm. OpenFlow, while it had widespread utilization at one point, has fallen out of fashion. The lack of interoperability from iterations of the protocol forced developers to create their own. This led to a dwindling OpenFlow user base and the current diffusion of various protocols throughout the industry.

Open source is about collaboration – using the best modules to stitch together something more than the sum of its parts, and SDN has to come from the same perspective. Without a unified, interoperable standard, the work behind SDN is a waste of time. SDN exists to improve the flow of data across disparate networks and to dynamically (de-)provision bandwidth and reroute traffic based on the needs of the moment, but if traffic will be funneled into a proprietary bottleneck, what's the point? It's like buying a Ferrari and aiming it at wall.