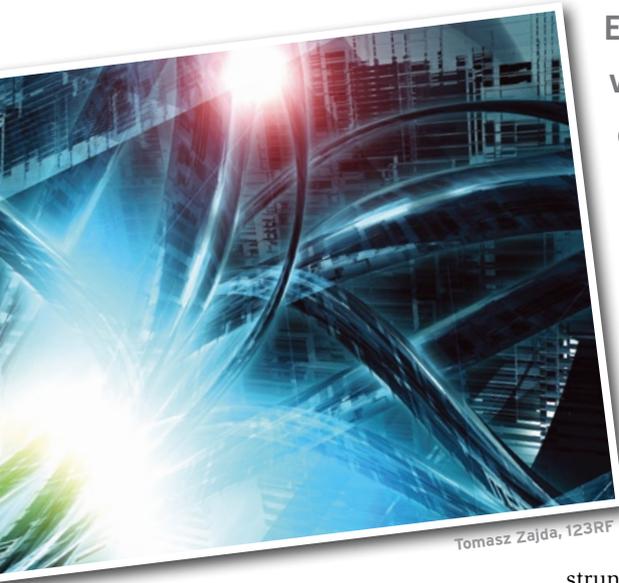


Abstracting the network with SDNs

A New Nexus



Even as the tech world works to figure out just what to do with the potential of cloud computing and big data, along comes a new bit of technology fueled by open source software: software-defined networks. *By Brian Proffitt*

Technology these days is all about abstraction: cloud computing is basically abstracting the concept of servers, so you don't have to worry about them. Big data's non-relational databases and Hadoop clusters perform a similar level of abstraction on database administration, and software-defined networks (SDNs) aim to do the same thing with networking.

How does SDN work?

Think about a traditional network and everything that entails. You have your routers, your switchers, and lots and lots of CAT5 and CAT6 cable strung around: all physical hardware that, when connected in a certain way, defines the flow of data in your organization. Like laying down a network of highways, planning a network takes time; it has to be done right the first time because shuffling things around is expensive.

A network has to do two big things: deliver data and manage the flow of that data. If I am downloading a video from California, the network knows to get it to me here in Indiana. Shunting the data through India and Europe would not be the most efficient method – unless, of course, some big physical failure were to occur between here and the West Coast that required the signal to be sent the long way around the planet.

Inside your company, the same thing happens, but at a smaller scale. Data is passed back and forth, and the traffic is ideally managed in such a way that if Bob in Marketing is uploading the big promo video, he can get it done fast, although perhaps the network slows down email for a bit to let the higher volume and higher priority traffic through.

That's the physical network, and the limitations of its design can show up in a hurry whenever you want to try to do something outside the original plan. Perhaps you want to set up videoconferencing in a new room. The cables and jacks are all there, but to handle the extra capacity, you would need to install the right switchers and routers to deal with the new load. Otherwise, the video calls would be awful and the rest of your network would suffer under the load. "Suffer" actually means crash and burn amidst fiery screams of pain – just so we're clear.

What SDN does is this: Assuming you have the network cable laid out in every room in your company, the SDN layer essentially acts as a virtual software switch or router in place of (or in conjunction with) the physical network device counterparts (Figure 1). This is why you might hear SDNs and virtual networks in the same conversation: two labels for the same concept.

Suddenly, the benefits of the network modifications you would have to make in the videoconferencing scenario become clear for IT. Your network topography is no longer rooted in the physical; rather, it is flexible and adjustable to your systems' needs on the fly. It's easy to see how SDNs can work with front-end IT needs to reconfigure networks for end users, but the real power goes even further. Think about the data center, where hundreds or thousands of servers are interconnected, and apply the concept of SDN to that. Now you can load balance devices across servers and auto-

matically adjust the network architecture to deliver the fastest and most efficient data paths at the same time. Now, imagine a variably shaped network with virtual servers, and you've got a virtual data center on steroids. If those servers have any elasticity in play, then you have an even better cloud operation.

Practical SDN

SDN is more than just some new gimmick for cloud computing; it can touch anything your organization's existing network can touch, which means a fundamental shift in how network administration will be handled across the board. Even home networks can benefit from SDN tools, especially those homes that are watching a lot of multimedia content and need the flexibility of a virtualized networking layer.

With so much potential for improvements to deliver, it's little wonder that a lot of networking companies, like Arista, Cisco, HP, and Juniper are keenly interested in putting their own products on the market. Start-ups like Big Switch and Nicira are making their own mark on the sector, too. Nicira is doing pretty well already; it was acquired by virtualization giant VMware in August 2012.

Big Switch is even pushing the boundaries of SDN past the notion of "virtual switches and routers," which is the layer the networking companies are trying to commercialize now. Big Switch's open source Floodlight platform is just that: a networking platform. Applications, not virtual controllers, will eventually control the network traffic and flow, abstracting the networking layer even further. Applications developed on Floodlight will simply be able to make "I want to do this" calls to the network layer, and the network will conform on the fly, delivering as much networking power as the app needs.

Network-based applications are not a new idea. Cisco was hosting development contests in 2009 for programmers who could build apps that would run right on top of their hardware devices instead of on servers. (I should know, I was a judge for one such contest.) This was the prototype for what Big Switch is now doing on a non-physical level.

Much of this work in SDN is being done around open source technologies. Besides the aforementioned Floodlight

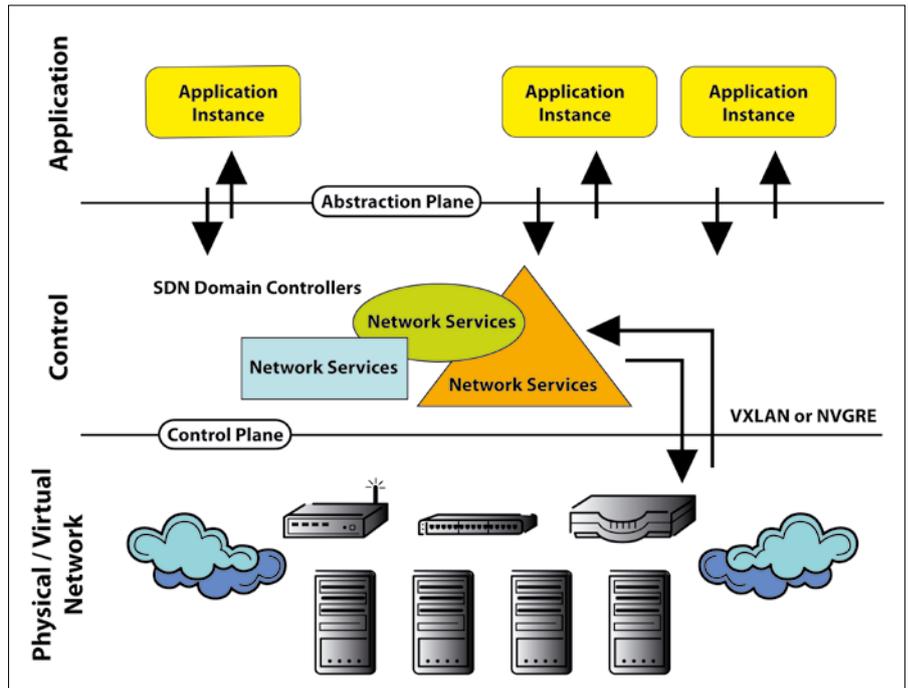


Figure 1: Layers in a software-defined network.

API, a big player in the SDN layer is OpenFlow [1], the Open Network Foundation's specification for a virtual network stack. OpenFlow is based on the VXLAN [2] standard and is being used by Cisco and VMware, who are leading the adoption of this standard. The other standard is NVGRE [3], which Microsoft, Intel, and Dell are pushing.

It is not clear yet which standard is going to come out on top, but the momentum appears to be behind VXLAN, to which the open source OpenFlow adheres. There are other platforms in VXLAN world, too. Nicira's Network Virtualization Platform (NVP) and OpenStack's Quantum networking component are prominent examples.

Right now, there is no "Linux" of SDN: a neatly packaged product that people can download and configure for themselves. Much of the work being done is very bleeding edge and requires a lot of custom configuration (much like OpenStack in the cloud computing space). Given the interest in making SDN work, expect that to change very soon, but although you should be paying attention to this technology, it's best to give it some more time to mature and bake.

With open source software at its heart, it shouldn't be long before we see some serious products in play, and, unlike the early days of Linux, no one will be mocking the coming of SDN. ■■■

INFO

- [1] OpenFlow: <http://www.openflow.org/>
- [2] VXLAN: <http://tools.ietf.org/html/draft-mahalingam-dutt-dcops-vxlan-00>
- [3] NVGRE: <http://tools.ietf.org/html/draft-sridharan-virtualization-nvgre-00>