White Paper

Software-Defined Reality: 7 Things You Should Know about SDN

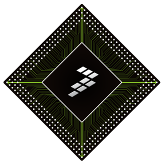
Exploring the reality behind the hype of software-defined networking

Software-defined networking (SDN) represents an entirely new way to manage connectivity: One that is defined not by the equipment makers, but by those who use the network. It's a network intelligent and flexible enough to route the most important traffic first, to direct network resources to where they are needed most, and to adapt and change over time to meet new user needs.

But just like any new technology, there is both hype and roadblocks to overcome. SDN offers significant benefits to operators, but will also require many changes in the way we think about and use the network.

This white paper will explore 7 realities behind the hype of SDN, shedding light on its true nature and how it can reach its full potential.

Table of Contents



freescale.com

1. SDN Has Unlimited Applications

When mentioning software-defined networking (SDN), you would probably think of large data centers. This makes sense because there is a strong association between SDN and cloud data centers, given that the SDN technology was initially adopted and subsequently evolved from these data centers. SDN was the secret sauce that allowed data centers to be more profitable and agile. Therefore, SDN wasn't something that these data centers wanted to broadly share with the rest of the world. This hindered development of SDN and made it difficult for other types of companies to see the value of SDN.

It has only been within the last two years that SDN technologies have been let out of the solitary confinement of data centers. Open communities such as ETSI-NFV and OPNFV are starting to apply this basic idea of centralized programmability to WAN, Metro and CPE equipment and alternate transports.

As carriers began to take notice, they realized that by using SDN, operators were able to efficiently deliver high quality services to a large end user community without ripping up their hardware on a regular basis. By implementing a virtualized platform, carriers were able to increase network speed and agility, hence enhancing service delivery. This infrastructure update also enabled carriers to address latency constraints and increased ability to solve problems in real-time. These improvements have made their way into the basic fabric of SDN.

Further, carrier networks have extended beyond their own walls to encompass end users. They have extended these SDN/NFV concepts from the WAN to customer on-premise devices. For example, Google rolled out its own next-generation data centers, incorporating SDN and NFV into them. But they also spread these concepts to the general public, providing new services, faster speeds and better user experience.

SDN has expanded outside the data centers to end users and has offered countless opportunities for other markets. However there has been some hesitation from enterprises, since adopting SDN means a new area to manage, new responsibilities and a whole new set of programming rules to learn. As a respite, new applications are beginning to emerge that will make the value proposition much stronger for the enterprises. For example, hybrid data centers allow enterprises to farm out services from an enterprise's private data center to a public one and back. With SDN implemented, it becomes very easy to move that data and workloads around to the place best-suited to execute them.

But why stop at just enterprise adoption? How about applying SDN techniques to industrial, commercial and the various markets served by distributors? Due to its flexibility, SDN offers countless opportunities for all these markets. Once its reins come off and the industry becomes increasingly aware of its advantages, the possibilities are endless!

2. Any Switch Can Become an SDN Switch with Software

Equipment vendors have been attempting to introduce Software-Defined Networking (SDN) equipment to the market over the past two years, which implies there are special hardware features required to create an SDN switch. But still, in reality, some of these new switches are not quite as open, flexible and programmable as they seem on the surface.

Put simply, a switch is either SDN or traditional. The main difference between these two categories is the software called an OpenFlow Agent (OF-Agent), which takes SDN commands coming from the controller and translates them into a language the switch can understand. But many of the “SDN switches” in the market today do not allow the network operators an access to the OF-Agent, making it next to impossible for the operators to change anything about the switch. This can make it unfeasible for companies to use equipment from various vendors, as the implementations of the OF-Agent can vary wildly from vendor to vendor. This reduces interoperability and keeps customers locked in with a specific vendor and its capabilities.

However, for sure, there are also true bare metal switches, where the software and hardware are sold separately, allowing the end-user to load software of their choice, instead of relying on a vendor. Further to that, there are also ways to get around the attempted vendor lock in of the so-called SDN switch. The Open Network Install Environment (ONIE), is an Open Compute Project open source initiative, which defines an open “install environment” for bare metal network switches where end users have a choice among different network OS. By taking over ownership of the software, companies are no longer forced to rely on a specific vendor for software needs and updates. This allows companies to actually control those so-called open SDN switches through software. More importantly, it allows legacy, traditional network systems to migrate to a SDN system without replacing existing hardware and thus offering lower-cost and a more flexible switching alternative.

Open vSwitch(OVS), a production quality open-source software implementation of a distributed virtual multilayer switch, aids in SDN adoption and effective network automation. OVS allows for overlay networking, which is a computer network built on the top of another network. This allows the underlying network to behave in a traditional way and two virtual machines on two different servers to be connected to each other directly by a piece of software.

The future of networking relies more and more on software. Per the November 2014 IHS Infonetics Carrier SDN and NFV Hardware and Software market size and forecast report, revenue from SDN software (orchestration and controller, and network apps) will grow from $46 million in 2014 to $1.7 billion in 2018, with a 2013–2018 CAGR of 185%. Also in 2018, SDN+NFV software is 77% and hardware is 23% of total service provider SDN/NFV revenue. SDN and NFV (Network Function Virtualization) represent the shift of focus from hardware to software. Software is the key that unlocks the capabilities and benefits of SDN for any switch on any network. You just have to know where to look and apply.

3. Not all SDN is Created Equal

Oftentimes, new trends and technologies get away from their roots. Just look at the telephone and the way millennials and younger generations use it to talk to each other without ever opening their mouths to speak. The communication is still there, but the talking – the thing the telephone was invented to do – has begun to fade.

Similarly, SDN and NFV are evolving away from their roots. When NFV was first introduced by 18 carriers at Mobile World Congress in 2012, it was positioned as a technology that would benefit from, but not depend on SDN. Today, the two technologies are practically married through a strong relationship between the Open Networking Foundation (ONF) and the European Telecommunications Standards Institute (ETSI). ONF is working on the first ONF-sponsored NFV proof of concept on service chaining using OpenSDN technology. With a strong liaison relationship in place, OpenSDN will now be driven by NFV use cases that will take OpenFlow beyond the data center into carrier networks and into the customer premise.

But that's not the only way SDN is changing and not all of it is for the better. Increasingly, more and more OEMs talk about SDN in a way that sounds great, but is really just a perpetuation of the closed network. They're creating a network that is more software-driven, but it still operates like a black box, keeping operators out of the networking equipment. This is appealing because it seems simpler to manage than a full SDN network, but it also reduces the benefits that SDN can deliver. Even OpenDaylight, which has a strong community behind it, might be limited on this front.

But all hope is not lost. ONF has increasingly been talking about OpenSDN, a standard that all the different OEMs can adhere to. This creates interoperability between the hardware vendors, but maintains the decoupling between data path and control path so that the programmability and flexibility that SDN promises is still available. ONF's new reference designs for OpenSDN will emphasize both OpenDaylight and ONOS. This prevents one single entity from driving the architecture and the technology in a certain direction, hence helping maintain the community's openness. This will also allow SDN to develop in a way that allows applications to tell the network what it needs (its intent), rather than direct the network what to do, and then let the control plane of the network determine how it meets the applications needs.

ONF has also created an open source community for code development, allowing anyone to participate. This will allow network operators to become increasingly involved and have some sense of ownership over the code, putting in the support and the man effort needed to develop it further. As the code continues to develop in this community, I expect that within 3-5 years, we'll start seeing a healthy software community to manage the hardware without the need for an embedded solution OEM to dominate the market. This will also allow for network operators to have a heavy hand in deciding how the network behaves.

After all, SDN started in universities, where it was a very open environment. In a way, this is SDN getting back to its roots. If an open source community can do 80 percent of the work, software developers can focus 100 percent of their time on the remaining 20 percent of the code, focusing exclusively on innovation and differentiation. By sharing, we'll move the networking field forward faster than if we create our own silos.

4. SDN Can Increase Network Security

Open source SDN could be scary. Especially after Heartbleed, Shellshock and the recent FREAK scare, people are understandably leery of the security of open source. There is a bit of a mystique that opening up the software programmable interface to anyone that wants to come in and code makes the code vulnerable and open to manipulations.

But in reality, an open programming model, like the one being embraced by the Open Networking Foundation (ONF), is actually more secure, particularly in the context of SDN. SDN provides one the ability to recognize and pin point a problem quickly and then quarantine and apply measures to address it. In a threat environment that's constantly changing, one needs a network that can evolve as fixes are made and also stay ahead of conceivable future threats.

Increased visibility into the code base also makes it easier to address issues. While it will be ignorant to say that there isn't a potential for problems like those associated with FREAK and other vulnerabilities, it's difficult for me to believe that these were deliberately architected into the software without anyone noticing. We all know that one of the key advantages of open source is that someone is always watching the code and keeping everyone honest. Instead, I believe that these vulnerabilities were simply an oversight that the bad guys found and exploited.

On the contrary, when a hole is exploited in a fixed system, it's a lot harder to address the problem. It's hard to update the embedded software, so if a problem arises, there really isn't a quick way to stop it. Because SDN infrastructure is not fixed in time, there's increased flexibility to fight attacks. As an example, it wasn't too long ago that the U.S. and Chinese government were trading accusations about whether one had built deliberate backdoors into the networking equipment being bought by the other. While the truth of these accusations is a discussion for another time, the bare facts are that in a traditional network, this sort of deviousness is very possible.

In a flexible and programmable SDN network, on the other hand, one can break the linkage between the software and hardware vendors. One can run whatever software one wants, choosing the amount of visibility one wants to offer into the software and how much one trusts that software to not have built-in backdoors. Or one could even go into the software itself to close any vulnerabilities one might find.

SDN holds a huge potential to increase network security. Due to its open, flexible, programmable nature, it may not seem secure on its surface, but beneath that façade is the ability to find and close security vulnerabilities quickly and easily, particularly when compared to traditional, fixed networks.

5. SDN is About Service Agility, Not Hardware Commoditization

Software-defined networking holds a lot of potential. By decoupling the data plane from the control plane, operators gain flexibility on various fronts – in terms of the hardware they choose, the software to go with it and the way their network operates. One of the biggest advantages of SDN is that it opens up the hardware market to increased competition. Currently, large OEMs charge premium prices for their equipment, driving propriety standards that lock customers into purchasing all their hardware from one vendor.

But with the increase in bare metal equipment that will come with SDN, operators no longer need to be locked in and will be able to pick and choose equipment from different vendors. It will result in OEMs to compete against each other on a variety of different features and differentiating points, including price, which will inevitably drive down equipment costs. Many look at this as one of the chief benefits of SDN. Demand for more bandwidth is rising exponentially and carriers are struggling to keep up, while keeping costs down. Reduced equipment costs due to increased programmability in the network can only help.

However, an important point to note is that equipment costs are unlikely to go down enough to make a significant dent in operators' balance sheets. What they really need is a way to drive increased revenue. While the focus on reducing costs is good, those who are looking at SDN to aid their balance sheet need to be focused more on the increased service agility that SDN provides rather than commoditization.

By decoupling hardware from software, carriers are freed from the OEM product cycle churn. In a traditional network, if you wanted to roll out a new feature or service, you would first need to go talk to your OEM vendor, who would in turn have discussions with its business channels, put your request on the roadmap, and then introduce it to the market. There holds two problems. First, the new feature or service is finally supported on the product long after the initial request was made, making it difficult for carriers to respond to immediate market needs. Second, it provides that feature or service to not only that single customer that made the initial request, but also to other customers, who are likely competitors.

But with SDN, new features and services can be introduced in software - quickly, easily and selectively. Carriers can test a new feature in a key market to see how their customers respond. It can then be rolled out more broadly if it's a hit, driving new revenue streams and even bringing in new customers in a very short duration.

Per Infonetics "2014 SDN and NFV (Network Function Virtualization) Service Provider Research" — the two primary factors driving service providers to SDN/NFV are service agility and the global view of the network across multiple vendors. This increased service agility allows carriers to not only differentiate themselves, but also increase their revenue per end user and respond to market needs quickly. This is a much more effective way for carriers to accommodate the costs of increasing need for bandwidth than decreasing equipment costs. SDN has many benefits and by focusing on the ways it increases revenue, rather than just the ways that it cuts costs, it becomes an easier decision to begin implementing it.

6. SDN is Turning the Network into an Applications Platform

While service agility and increased revenue are important benefits of software-defined networking (SDN), the advantages of programming interfaces cannot be overlooked. These programming interfaces will allow SDN to turn the network into an applications platform. They will be one of SDN's biggest selling points, as they will enable the creation of any application imaginable to support networking needs.

Simply, SDN will offer an increased level of competitiveness to hardware vendors. Companies will compete based on how their product features are differentiated in the market to execute or support various SDN protocols such as OpenFlow etc., until one such standard emerges as the dominant one. This standardization will allow a large software community to emerge in order to write networking apps that run on any SDN equipment.

This is similar to what we saw in the PC industry, where operating systems made PCs easier to manage and support application delivery. More recently, we saw this trend in smartphones and tablets, which developed this concept even further by creating app development platforms and an apps store to simplify delivery.

SDN applications are programs that directly and programmatically communicate the network requirements and desired network behavior to the SDN controller. As SDN transitions the network from behaving autonomously using protocol-based rules into a fully programmable apps platform, SDN functions can be delivered as a service on a networking infrastructure (NaaS). Eventually, this will become a programmable platform where operators can develop and deploy more applications using the network. When the programmable interface becomes generalized and expressive, vendors could develop new, innovative ways of networking. This programmability offers an ecosystem to experiment, where software developers can perceive what the market wants next and introduce the idea to the network, gauging success by how many users accept it. The network is quickly becoming a more customizable platform that would be an asset to the businesses rather than a burden or requirement.

Think about your smartphone for a minute. It doesn't matter what OS you're running, you've probably got a few apps on there. Almost everyone has an app for voice calls, text messages, email and browsing the internet. Most of us have a camera app and a maps app. But if you look at your spouse's phone or your sibling's phone or your coworker's phone, it's a pretty good bet that many of your apps are different from theirs. And just like the apps you keep on your smartphone are different than those that your friends keep, the applications you keep on your network will differ as well, though there could be many similarities. Almost everyone will have apps for firewalls and package acceleration. But you might also develop a network app specific to your business, aimed at helping you save money, get more customers and earn more revenue by increasing efficiency.

As the network slowly evolves into an applications platform, we'll see increased innovation applications that change the way that users interact with the network and the way operators leverage the network to benefit their business.

7. Humans are more of a Hindrance to SDN Advancement than the Technology

In today's world filled with cell phones, tablets and other smart devices, operators are on the hot seat to manage network services quickly in order to avoid a slow and overloaded network. Software-defined networking (SDN) is generating a lot of attention, mainly as a tool to provide the flexibility and programmability needed to do just that. While the hype around its possibilities is here, is SDN ready for primetime? Technically, yes. But there is problem, which isn't related to the technology, but rather with the people who must manage this change.

The enterprise network has been dominated by trusted OEMs that provided full support for their equipment with its complicated embedded software. SDN helps the network break free of this dominating vendor lock-in by offering bare metal switches, which have the programmability to control and change various parts of the network in order to offer increased flexibility to the network operators. But this also shifts the ownership of network solutions from the OEMs that designed the software and hardware of the traditional network to the operators, presenting operators with a new area to manage, new responsibilities and a whole new set of programming rules to learn.

This puts added pressure on the operators and carries a lot of risk for them, making the value proposition for SDN less clear. This is particularly true for the enterprise, although many telecom carriers and cloud data centers are already making adjustments to change their business model accordingly. Many enterprises are seeking ways to reduce this risk by looking for software that will manage their SDN for them. But this creates the same vendor lock-in that SDN is seeking to eliminate. Instead, the industry should work together to create a way for operators to fully control the network within an interface that they are already familiar with. Then, while an SDN expert might be required to deploy an SDN system, an easy-to-use interface could be used to control the network's behavior. This places operators firmly in control without requiring them to learn an entirely new programming skillset or to hire a fleet of SDN experts.

Solving these SDN adoption issues likely involves restructuring IT boundaries and an emergence of trusted software solutions in order to enable operators to regain control of network configuration in light of their newfound hardware independence. Doing so will be well worth the effort as the network's programmability and flexibility allows companies to deploy new services, eliminate slow traffic and create a better user experience for both employees and customers.

8. Conclusion: Making SDN a Reality

Unlike many emerging technology applications, the technical requirements for SDN already exist. Network hardware has become truly open and the software and coding to support that hardware are well into development. The business case is clear. While a few technical obstacles remain, there is no doubt that they will be overcome in the coming years.

The last barrier SDN faces is the fear of the unknown and the sheer amount of organizational change that it will require. However, if those changes are approached with sensitivity to what came before, this is one more obstacle that can be easily overcome, allowing businesses to capitalize on the numerous opportunities SDN represents.

Learn more at freescale.com