

TAKING A GOOD HARD LOOK AT SDN

SDN is sitting at the peak of it's hype cycle (at least I hope it's the peak.) Every vendor has a definition and a plan. Most of those definitions and plans focus around protecting their existing offerings and morphing those into some type of SDN vision. Products and entire companies have changed their branding from whatever they were to SDN and the markets flooded with SDN solutions that solve very different problems. This post will take a deep dive into the concepts around SDN and the considerations of a complete solution. As always with my posts this is focused on the data center network, because I can barely spell WAN, have never spent time on a campus and have no idea what magic it is that service providers do. The first question anyone considering SDN solutions needs to ask is: What problem(s) am I trying to solve. Start with the business drivers for the decision.

There are many that SDN solutions look to solve, a few examples are:

- Faster response to business demands for new tenants, services and applications.
- More intelligent configuration of network services such as load balancers, firewalls etc. The ability to dynamically map application tiers to required services.

- Reductions in cost i.e. CapEx via enabling purchase of lower cost infrastructure and OpEx via reducing administrative overhead of device centric configuration.
- Ability to create new revenue streams via more intelligent network service offerings.
- Reduction in lock-in from proprietary systems.
- Better network integration with cloud management systems and orchestration tools.
- Better network efficiency through closer match of network resources to application demands.

That leaves a lot of areas with room for improvement in order to accomplish those tasks. That's one of the reasons the definition is so loose and applied to such disparate technologies. In order to keep the definition generic enough to encompass a complete solution there are three major characteristics I prefer for defining an SDN architecture:

- Flow Management – The ability to define flows across the network based on characteristics of the flow in a centralized fashion.
- Dynamic Scalability – Providing a network that can scale beyond the capabilities of traditional tools and do so in a fluid fashion.
- Programmability – The ability for the functionality provided by the network to be configured programmatically typically via APIs.

The Complete Picture:

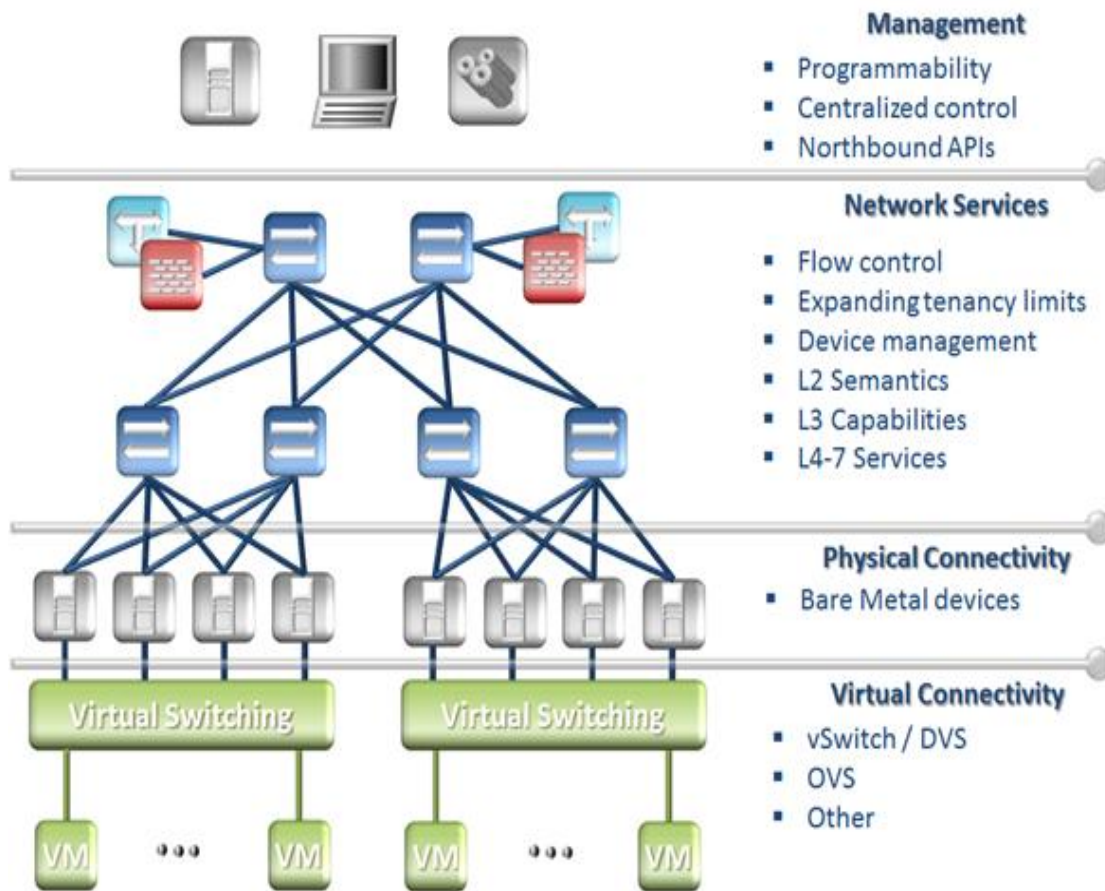
In looking for a complete solution for Software Defined data center network it's important to assess all aspects required to deliver cohesive network services and packet delivery:

- Packet delivery – routing/switching as required. Considerations such as requirements for bridging semantics (flooding, broadcast), bandwidth, multi-pathing etc.
- L4-L7 service integration – The ability to map application tiers to required network services such as load-balancers and firewalls.
- Virtual network integration – Virtual switching support for your chosen hypervisor(s). This will be more complex in multi-hypervisor environments.
- Physical network integration – Integration with bare-metal servers, standalone appliances, network storage and existing infrastructure.
- Physical management – The management of the physical network nodes, required configuration of ports, VLANs, routes, etc.
- Scalability – Ability to scale application or customer tenancy beyond the 4000 VLAN limit.
- Flow management – The ability to program network policy from a global perspective.

Depending on your overall goals you may not have requirements in each of these areas but you'll want to analyze that carefully based on growth expectations.

Don't run your data center like congress kicking the can (problem) down the road.

The graphic below shows the various layers to be considered when looking at SDN solutions.



Current Options:

The current options for SDN typically provide solutions for one or more of these issues but not all. The chart below takes a look at some popular options.

	VLANScale	L4-7	Bare Metal Support	Physical Network Node MGMT	KVM	VMware	Xen	HyperV	L3	Flow MGMT
Nicira/VMware	X	3rd Party	*		X	*	X		3rd Party	X
Overlays	X				X	X	X	X		
OpenFlow			X		X	X	X	X	X	X
Midokura	X	X			X		X		X	

X = Support

*** = Future Support**

This chart is not intended to be all encompassing or to compare all features of equal products (obviously an overlay doesn't compete with a Nicira or Midokura solution, and each of those rely on overlays of some type.) Instead it's intended to show that the various solutions lumped into SDN provide solutions for different areas of the data center network. One or more tools may be necessary to deploy a full SDN architecture and even then there may be gaps in areas like bare metal support, integration of standalone network appliances and provisioning/monitoring/troubleshooting of physical switch nodes (yes that all still matters.)

API Model:

Another model lumped into SDN is northbound APIs for network devices. Several networking vendors are in various stages of support for this model. This model does provide programmability but I would argue against it's scale. Using this model requires top down management systems that understand each device, its capabilities and its API. To scale this type of management system and program network flows this way is not easy and will be error prone. Additionally this model does not provide any additional functionality, visibility or holistic programmability, simply a better way to configure individual devices. That being said managing via APIs is light years ahead of screen scrapes and CLI scripting.

Hardware Matters:

Let me preface with what I'm not saying: I'm not saying that hardware will/won't be commoditized, and I'm not saying that custom silicon or merchant silicon is better or worse.

I am saying that the network hardware you choose will matter. Table sizes, buffer space, TCAM size will all factor in, and depending on your deployment model will be a major factor. The hardware will also need to provide maximum available bandwidth and efficient ECMP load-balancing for network throughput. This load-balancing can be greatly affected by the overlay method chosen based on available header information for hashing algorithms. Additionally your hardware must

support the options of the SDN model you choose. For example in a Nicira/VMware deployment you'll have future support for management of switches running OVS, you may want these to tie in physical servers, etc. The same would apply if you choose OpenFlow. You'll need switch hardware that provides OpenFlow support, additionally it will need to support your deployment model hybrid or pure OpenFlow.

The hardware also matters in configuration, management, and troubleshooting.

While there is a lot of talk of "We just need any IP connectivity" that IP network still has to be configured and managed. Layer 2/3 constructs must be put in place, ports must be configured. This hardware will also have to be monitored, and troubleshot when things fail. This will be more difficult in cases where the overlay is unknown to the L3 infrastructure at which point two separate independent networks will be involved: physical and logical.

Management Model:

There are several management models to choose from and two examples in the choices I compared above. OpenFlow uses a centralized top down approach with the controller pushing flows to all network elements and handling policy for new flows forwarded from those devices. The Nicira/VMware solution uses the same model as OpenFlow. Midokura on the other hand takes a play from distributed systems and pushes intelligence to the edges in that fashion. Each model offers

various pros/cons and will play a major role in the scale and resiliency of your SDN deployment.

Northbound API:

The Northbound API is different than the device APIs mentioned below. This API opens the management of your SDN solution as whole up to higher level systems.

Chances are you're planning to plug your infrastructure into an automation/orchestration solution or cloud platform. In order to do this you'll want a robust northbound API for your infrastructure components, in this case your SDN architecture. If you have these systems in place, or have already picked your horse you'll want to ensure compatibility with the SDN architectures you consider. Not all APIs are created equal, and they are far from standardized so you'll want to know exactly what you're getting from a functionality perspective and ensure the claims match your upper layer systems needs.

Source: <http://www.definethecloud.net/taking-a-good-hard-look-at-sdn/>