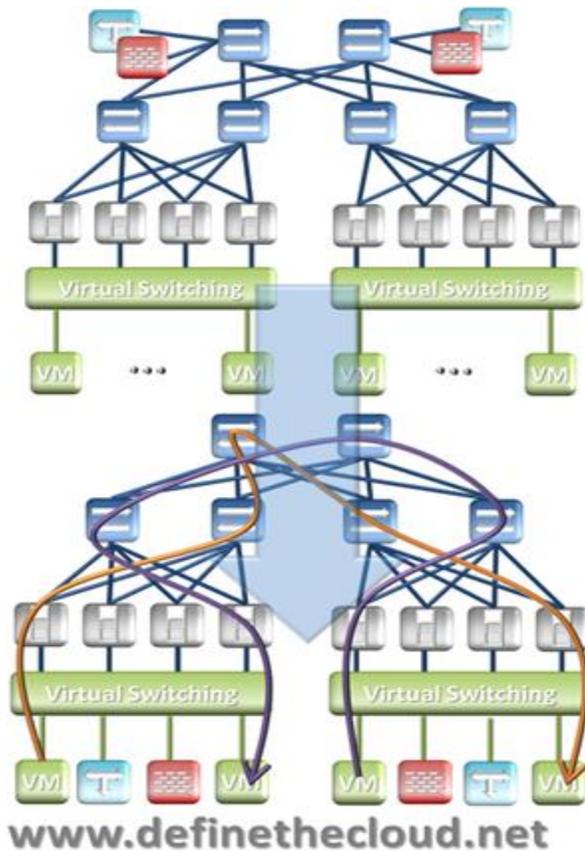


IT'S OUR TIME DOWN HERE— “UNDERLAYS”

This got me thinking about data center network overlays, and the physical networks that actually move the packets some Network Virtualization proponents have dubbed “underlays.” The more I think about it, the more I realize that it truly is our time down here in the “lowly underlay.” I don’t think there’s much argument around the need for change in data center networking, but there is a lot of debate on how. Let’s start with their time up there “Network Virtualization.”

Network Virtualization

Unlike server virtualization, Network Virtualization doesn’t partition out the hardware and separate out resources. Network Virtualization uses server virtualization to virtualize network devices such as: switches, routers, firewalls and load-balancers. From there it creates virtual tunnels across the physical infrastructure using encapsulation techniques such as: VxLAN, NVGRE and STT. The end result is a virtualized instantiation of the current data center network in x86 servers with packets moving in tunnels on physical networking gear which segregate them from other traffic on that gear. The graphic below shows this relationship.



Network Virtualization in this fashion can provide some benefits in the form of: provisioning time and automation. It also induces some new challenges discussed in more detail here: [What Network Virtualization Isn't](#) (be sure to read the comments for alternate view points.) What network virtualization doesn't provide, in any form, is a change to the model we use to deploy networks and support applications. The constructs and deployment methods for designing applications and applying policy are not changed or enhanced. All of the same broken or misused methodologies are carried forward. When working with customers to

begin virtualizing servers I would always recommend against automated physical to virtual server migration, suggesting rebuild in a virtual machine instead.

The reason for that is two fold. First server virtualization was a chance to re-architect based on lessons learned. Second, simply virtualizing existing constructs is like hiring movers to pack your house along with dirt/cobwebs/etc. then move it all to the new place and unpack. The smart way to move a house is donate/yard sale what you won't need, pack the things you do, move into a clean place and arrange optimally for the space. The same applies to server and network virtualization.

Faithful replication of today's networking challenges as virtual machines with encapsulation tunnels doesn't move the bar for deploying applications. At best it speeds up, and automates, bad practices. Server virtualization hit the same challenges. I discuss what's needed from the ground up here: [Network Abstraction and Virtualization: Where to Start?](#). Software only network virtualization approaches are challenged by both restrictions of the hardware that moves their packets and issues with methodology of where the pain points really are. It's their time up there.

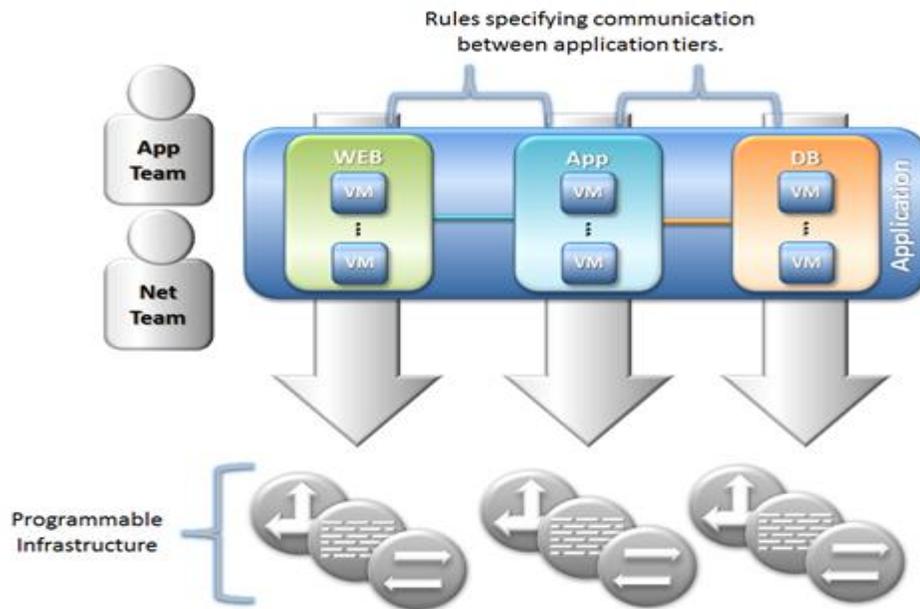
Underlays

The physical transport network which is minimized by some as the “underlay” is actually more important in making a shift to network programmability, automation and flexibility. Even network virtualization vendors will agree, to some extent, on this if you dig deep enough. Once you cut through the marketecture of “the underlay doesn’t matter” you’ll find recommendations for a non-blocking fabric of 10G Access ports and 40G aggregation in one design or another. This is because they have no visibility into congestion and no control of delivery prioritization such as QoS.

Additionally Network Virtualization has no ability to abstract the constructs of VLAN, Subnet, Security, Logging, QoS from one another as described in the link above. To truly move the network forward in a way that provides automation and programmability in a model that’s cohesive with application deployment, you need to include the physical network with the software that will drive it. It’s our time down here.

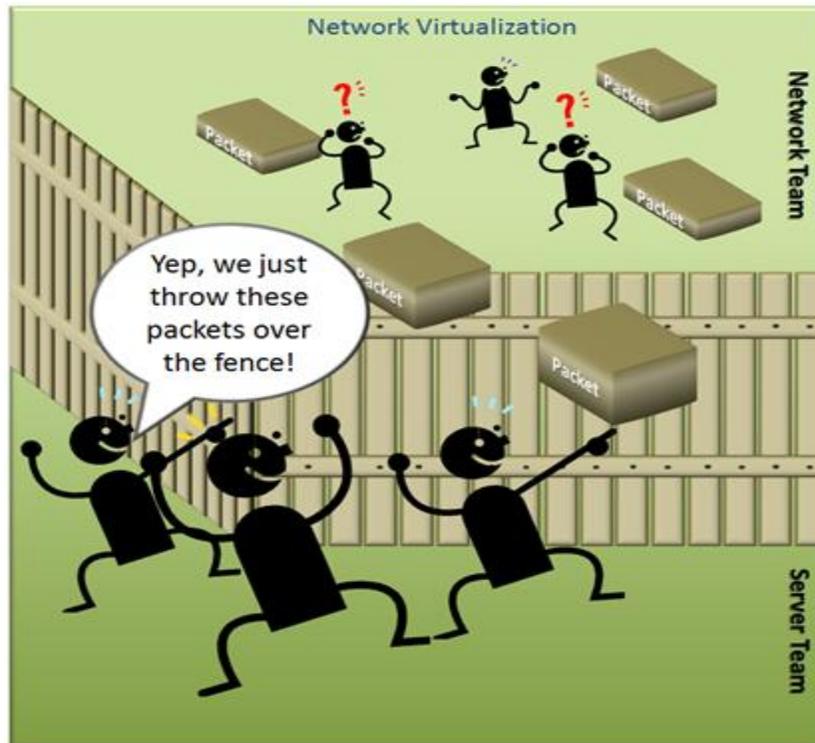
By marrying physical innovations that provide a means for abstraction of constructs at the ground floor with software that can drive those capabilities, you end up with a platform that can be defined by the architecture of the applications that will utilize it. This puts the infrastructure as a whole in a position to be deployed in lock-step with the applications that create differentiation and drive

revenue. This focus on the application is discussed here: Focus on the Ball: The Application. The figure below, from that post, depicts this.



The advantage to this ground up approach is the ability to look at applications as they exist, groups of interconnected services, rather than the application as a VM approach. This holistic view can then be applied down to an infrastructure designed for automation and programmability. Like constructing a building, your structure will only be as sound as the foundation it sits on.

For a little humor (nothing more) here's my comic depiction of Network



Virtualization.

Source: <http://www.definethecloud.net/its-our-time-down-here-underlays/>