Is SDN and Network Virtualization the Same Thing?
Due February 1, 2014

Why is there so much confusion about the relationship between SDN and network virtualization? A lot of the discussion in the industry treats these two concepts as separate but related. However, recently I have read a number of white papers that seem to equate SDN with a network virtualization solution, such as the one from VMware/Nicira. So, which way is it? Are SDN and Network virtualization related concepts or are they the same thing?

Part of the challenge with answering the preceding question is that there isn't a universally agreed to definition of SDN. That said, most people buy into SDN being an architecture that decouples the network control and forwarding functions which enables the network control to become directly programmable and the underlying infrastructure to be abstracted for applications and network services.

IT organizations have been implementing network virtualization via overlays for the last few years. However, the initial wave of these solutions didn't feature a controller. Since these solutions typically used flooding as a way to disseminate information about the end systems, these solutions didn't scale well. One of the key distinctions of network virtualization solutions from companies such as VMware/Nicira is that they have centralized the control functionality into a controller. Since these solutions enable virtualization and they also typically enable the chaining of various network services, an argument can be made that these solutions are SDN based on the description of SDN that is in the preceding paragraph. Part of the resistance to equating network virtualization via overlays with SDN is that a number of the proponents of SDN buy into the notion of the SDN controller directly controlling the network elements and these network virtualization overlay solutions can't do that.

However, there are other ways to implement network virtualization. Network virtualization can also be implemented as an application that runs on a controller, leverages the OpenFlow protocol and defines virtual networks based on policies that map flows to the appropriate virtual network using the L1-L4 portions of the header. Yet another way to implement network virtualization is the Virtual Tenant Network (VTN) application that was recently accepted by the OpenDaylight consortium. The VTN solution provides a layer of abstraction between the virtual network and the physical network. With this approach, a virtual network can be designed and deployed using logical network elements such as a vBridge or a vRouter. VTN can be used to implement an overlay network, an OpenFlow network, or a hybrid overlay/OpenFlow network.

So if you want to implement network virtualization what options do you have? Option A: You can implement a network virtualization solution that looks a lot like SDN, but that solution probably has no direct knowledge of, or control over the underlying network elements. Option B: You can implement network virtualization using a solution that most people think of as SDN because it does know about and control the underlying network elements. Option C: You can implement a solution that can function like Option A or Option B or some combination of those options.

So as to not drown in the confusion that results from having too many options, my take is that if network control functionality is centralized into a controller, then that is a software defined network. I look at network virtualization as potentially the killer app for SDN, but acknowledge that it is possible to implement an SDN and not do network virtualization.