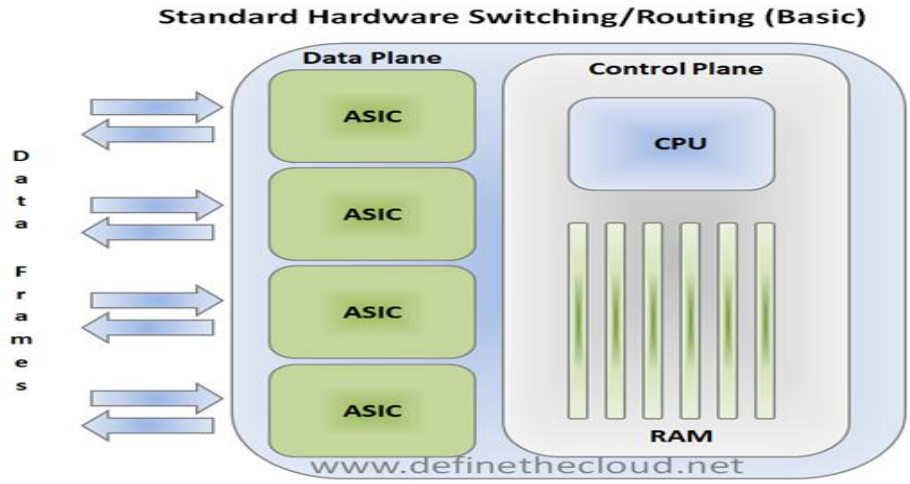
SDN – CENTRALIZED NETWORK COMMAND AND CONTROL

Software Defined Networking (SDN) is a hot topic in the data center and cloud community. The geniuses <sarcasm> over at IDC predict a $2 billion market by 2016 (expect this number to change often between now and then, and look closely at what they count in the cost.) The concept has the potential to shake up the networking business as a whole (http://www.networkcomputing.com/next-gen-network-tech-center/240001372) and has both commercial and open source products being developed and shipping, but what is it, and why?

Let's start with the why by taking a look at how traditional networking occurs.

Traditional Network Architecture:



The most important thing to notice in the graphic above is the separate control and data planes. Each plane has separate tasks that provide the overall

switching/routing functionality. The control plane is responsible for configuration of the device and programming the paths that will be used for data flows. When you are managing a switch you are interacting with the control plane. Things like route tables and Spanning-Tree Protocol (STP) are calculated in the control plane. This is done by accepting information frames such as BPDUs or Hello messages and processing them to determine available paths. Once these paths have been determined they are pushed down to the data plane and typically stored in hardware. The data lane then typically makes path decisions in hardware based on the latest information provided by the control plane. This has traditionally been a very effective method. The hardware decision making process is very fast, reducing overall latency while the control plane itself can handle the heavier processing and configuration requirements.

This method is not without problems, the one we will focus on is scalability. In order to demonstrate the scalability issue I find it easiest to use Quality of Service (QoS) as an example. QoS allows forwarding priority to be given to specific frames for scheduling purposes based on characteristics in those frames. This allows network traffic to receive appropriate treatment in times of congestion. For instance latency sensitive voice and video traffic is typically engineered for high priority to ensure the best user experience. Traffic prioritization is typically based on tags in the frame known as Class of Service (CoS) and or Differentiated

Services Code Point (DSCP.) These tags must be marked consistently for frames entering the network and rules must then be applied consistently for their treatment on the network. This becomes cumbersome in a traditional multi-switch network because the configuration must be duplicated in some fashion on each individual switching device.

An easier example of the current administrative challenges consider each port in the network a management point, meaning each port must be individually configured. This is both time consuming and cumbersome.

Additional challenges exist in properly classifying data and routing traffic. A fantastic example of this would be two different traffic types, iSCSI and voice. iSCSI is storage traffic and typically a full size packet or even jumbo frame while voice data is typically transmitted in a very small packet. Additionally they have different requirements, voice is very latency sensitive in order to maintain call quality, while iSCSI is less latency sensitive but will benefit from more bandwidth. Traditional networks have few if any tools to differentiate these traffic types and send them down separate paths which are beneficial to both types.

These types of issues are what SDN looks to solve.

The Three Key Elements of SDN:

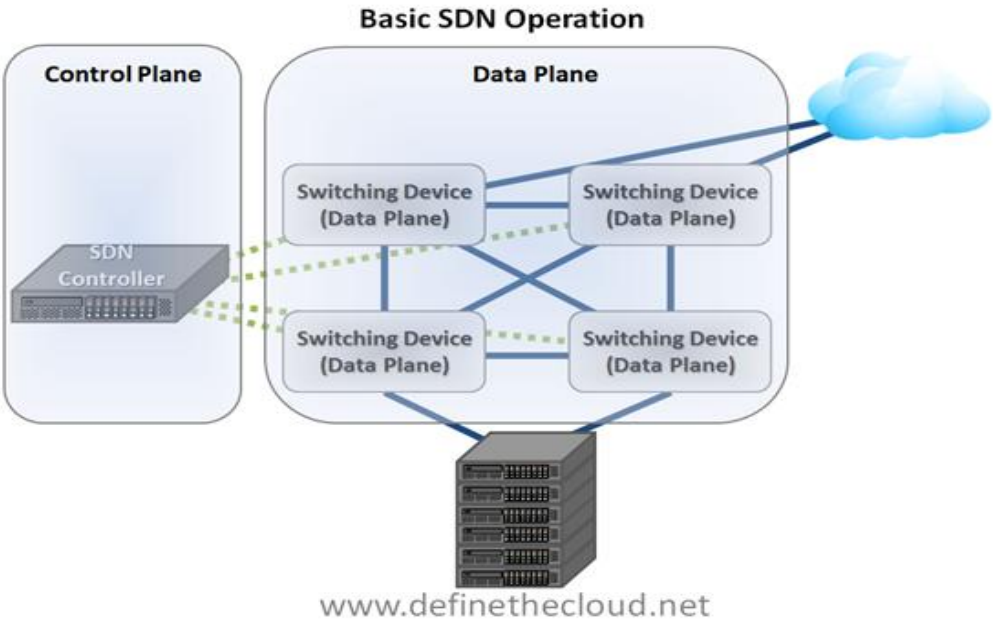
- Ability to manage the forwarding of frames/packets and apply policy

- Ability to perform this at scale in a dynamic fashion
- Ability to be programmed

Note: In order to qualify as SDN an architecture does not have to be Open, standard, interoperable, etc. A proprietary architecture can meet the definition and provide the same benefits. This blog does not argue for or against either open or proprietary architectures.

An SDN architecture must be able to manipulate frame and packet flows through the network at large scale, and do so in a programmable fashion. The hardware plumbing of an SDN will typically be designed as a converged (capable of carrying all data types including desired forms of storage traffic) mesh of large lower latency pipes commonly called a fabric. The SDN architecture itself will in turn provide a network wide view and the ability to manage the network and network flows centrally.

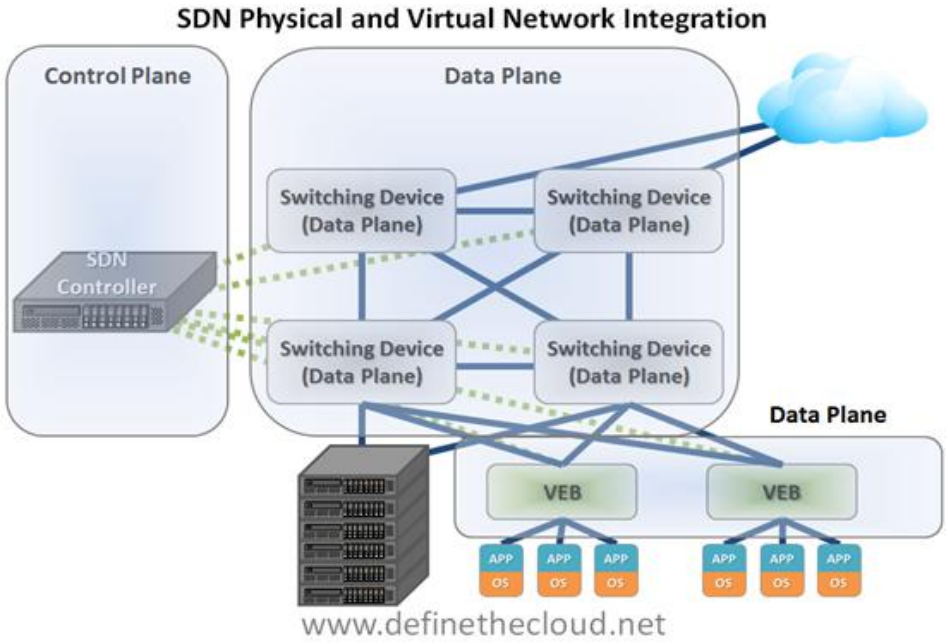
This architecture is accomplished by separating the control plane from the data plane devices and providing a programmable interface for that separated control plane. The data plane devices receive forwarding rules from the separated control plane and apply those rules in hardware ASICs. These ASICs can be either commodity switching ASICs or customized silicone depending on the functionality and performance aspects required. The diagram below depicts this relationship:



In this model the SDN controller provides the control plane and the data plane is comprised of hardware switching devices. These devices can either be new hardware devices or existing hardware devices with specialized firmware. This will depend on vendor, and deployment model. One major advantage that is clearly shown in this example is the visibility provided to the control plane. Rather than each individual data plane device relying on advertisements from other devices to build it's view of the network topology a single control plane device has a view of the entire network. This provides a platform from which advanced routing, security, and quality decisions can be made, hence the need for programmability. Another major capability that can be drawn from this centralized control is visibility. With a centralized controller device it is much easier to gain

usable data about real time flows on the network, and make decisions (automated or manual) based on that data.

This diagram only shows a portion of the picture as it is focused on physical infrastructure and serves. Another major benefit is the integration of virtual server environments into SDN networks. This allows centralized management of consistent policies for both virtual and physical resources. Integrating a virtual network is done by having a Virtual Ethernet Bridge (VEB) in the hypervisor that can be controlled by an SDN controller. The diagram below depicts this:



This diagram more clearly depicts the integration between virtual networking systems and physical networking systems in order to have cohesive consistent control of the network. This plays a more important role as virtual workloads migrate. Because both the virtual and physical data planes are managed centrally

by the control plane when a VM migration happens it's network configuration can move with it regardless of destination in the fabric. This is a key benefit for policy enforcement in virtualized environments because more granular controls can be placed on the VM itself as an individual port and those controls stick with the VM throughout the environment.

Note: These diagrams are a generalized depiction of an SDN architecture. Methods other than a single separated controller could be used, but this is the more common concept.

With the system in place to have centralized command and control of the network through SDN and a programmable interface more intelligent processes can now be added to handle complex systems. Real time decisions can be made for the purposes of traffic optimization, security, outage, or maintenance. Separate traffic types can be run side by side while receiving different paths and forwarding that can respond dynamically to network changes.

Source: http://www.definethecloud.net/sdn-centralized-network-command-and-control/