**Software-Defined Networking (SDN) Definition**

**WHAT IS SDN?**

The physical separation of the network control plane from the forwarding plane, and where a control plane controls several devices.

Software-Defined Networking (SDN) is an emerging architecture that is dynamic, manageable, cost-effective, and adaptable, making it ideal for the high-bandwidth, dynamic nature of today's applications. This architecture decouples the network control and forwarding functions enabling the network control to become directly programmable and the underlying infrastructure to be abstracted for applications and network services. The OpenFlow® protocol is a foundational element for building SDN solutions. The SDN architecture is:

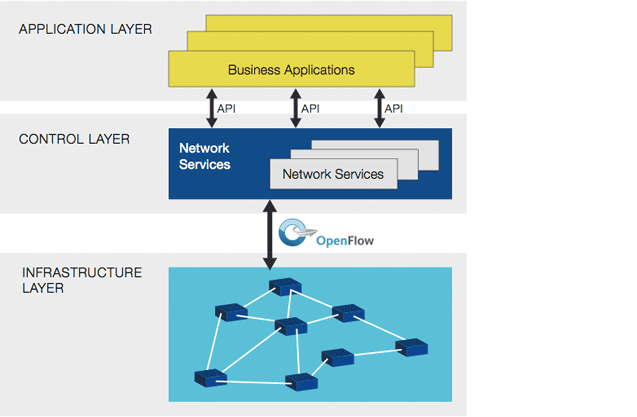
Directly programmable: Network control is directly programmable because it is decoupled from forwarding functions.

Agile: Abstracting control from forwarding lets administrators dynamically adjust network-wide traffic flow to meet changing needs.

Centrally managed: Network intelligence is (logically) centralized in software-based SDN controllers that maintain a global view of the network, which appears to applications and policy engines as a single, logical switch.

Programmatically configured: SDN lets network managers configure, manage, secure, and optimize network resources very quickly via dynamic, automated SDN programs, which they can write themselves because the programs do not depend on proprietary software.

Open standards-based and vendor-neutral: When implemented through open standards, SDN simplifies network design and operation because instructions are provided by SDN controllers instead of multiple, vendor-specific devices and protocols.



**Computing Trends are Driving Network Change**

SDN addresses the fact that the static architecture of conventional networks is ill-suited to the dynamic computing and storage needs of today’s data centers, campuses, and carrier environments. The key computing trends driving the need for a new network paradigm include:

Changing traffic patterns: Applications that commonly access geographically distributed databases and servers through public and private clouds require extremely flexible traffic management and access to bandwidth on demand.

The “consumerization of IT”: The Bring Your Own Device (BYOD) trend requires networks that are both flexible and secure.

The rise of cloud services: Users expect on-demand access to applications, infrastructure, and other IT resources.

“Big data” means more bandwidth: Handling today’s mega datasets requires massive parallel processing that is fueling a constant demand for additional capacity and any-to-any connectivity.

In trying to meet the networking requirements posed by evolving computing trends, network designers find themselves constrained by the limitations of current networks:

Complexity that leads to stasis: Adding or moving devices and implementing network-wide policies are complex, time-consuming, and primarily manual endeavors that risk service disruption, discouraging network changes.

Inability to scale: The time-honored approach of link oversubscription to provision scalability is not effective with the dynamic traffic patterns in virtualized networks—a problem that is even more pronounced in service provider networks with large-scale parallel processing algorithms and associated datasets across an entire computing pool.

Vendor dependence: Lengthy vendor equipment product cycles and a lack of standard, open interfaces limit the ability of network operators to tailor the network to their individual environments.

**Open Networking Foundation: Dedicated to SDN**

Open Networking Foundation (ONF) is a user-driven organization dedicated to the promotion and adoption of SDN, and implementing SDN through open standards where such standards are necessary to move the networking industry forward. As part of its quest to make SDN a commercial reality that meets customer needs, ONF is developing open standards such as the OpenFlow® Standard and the OpenFlow® Configuration and Management Protocol Standard. The OpenFlow® Standard is the first and only vendor-neutral standard communications interface defined between the control and forwarding layers of an SDN architecture. ONF working groups are also paving the way for interoperable solution development by collaborating with the world’s leading experts on SDN and OpenFlow® regarding SDN concepts, frameworks, architecture, and standards.

**What is the difference between SDN and NFV?**

Network function virtualization (NFV) and software-defined networks (SDN) are two closely related technologies that often exist together, but not always. An SDN can be considered a series of network objects (such as switches, routers, firewalls) that deploy in a highly automated manner. The automation may be achieved by using commercial or open source tools customized according to the administrator's requirements. A full SDN may only cover relatively straightforward networking requirements, such as VLAN and interface provisioning.

In many cases, SDN will also be linked to server virtualization, providing the glue that sticks virtual networks together. This may involve NFV, but not necessarily. NFV is the process of moving services, such as load balancing, firewalls and IPS, away from dedicated hardware into a virtualized environment. This is, of course, part of a wider movement toward the virtualization of applications and services.

Functions such as caching and content control can easily be migrated to a virtualized environment but won't necessarily provide any significant reduction in operating costs until some intelligence is introduced. This is because a straight physical to virtual, from an operational perspective, achieves little beyond the initial reduction in power and rack-space consumption. Until some dynamic intelligence is introduced with an SDN technology, NFV inherits many of the same constraints as traditional hardware appliance deployments, such as static, administrator-defined and managed policies.

A good example is virtualized application delivery controllers (ADCs). With careful configuration, it is possible to react to the network state and spin up or down application servers as demands rise and fall. However, traditional hardware deployments have been able to do this for a while, and the configuration is very static; it doesn't cater to the scenario where the ADC itself becomes overloaded or an additional application needs to be brought into production quickly. With SDN features driving NFV, several useful things start to happen. The network can react when things need to change at the micro and macro level. An additional instance can be provisioned in a cluster of virtualized ADCs as the load increases, and production applications can easily be cloned and re-deployed in a development environment. The potential is endless.

So it's perfectly possible to have NFV without the inclusion of a full-blown SDN. The two are often deployed together, and an SDN that drives NFV is a very powerful combination.

Neither NFV nor SDN are turnkey solutions in early 2014 -- a great deal of integration and policy design still need to happen. This can become a reality for many enterprises, but the harness is not entirely in place. That said, the tools are rapidly evolving, and many vendors are bringing technologies to market that support SDN or NFV deployments. Ultimately, the implementation of either or both technologies will be driven by the business needs.