HP OpenFlow and SDN Technical Overview

Technical Solution Guide

Version: 1
September 2013

Table of Contents

Introduction

SDN in a Nutshell

From a 10,000 foot point of view: SDN separate the control plane and data plane of networking devices.

SDN separates the network control plane (the logic that controls the behavior of the network) from the network's data plane (forwarding of network traffic such HTTP, E-mail and FTP)

Rather than using distributed protocols such as OSPF and BGP to manage the control plane, SDN changes the paradigm. In a pure SDN model, OSPF and other protocols are no longer used to manage the control plane.

The separation makes it possible to write a high level program to control the behavior of an entire network. Thus, network operators can create applications to more easily control security, quality of service and other network behavior.

In a pure SDN environment, rather than a network engineer configuring individual network devices via the Command Line Interface (CLI), a central device (SDN controller) controls the data planes (forwarding information base, routing information base) using an open standards protocol called OpenFlow. The control plane is moved to a central controller while network devices retain the data plane locally.

However, in today's networking environment, a hybrid model is used where routing tables (RIB) are still partially controlled via routing protocols such as OSPF, but some traffic is controlled via SDN. A VLAN could be allocated for SDN control and other VLANs remain under the traditional control of OSPF and other protocols.

The actual forwarding of traffic is discussed in the OpenFlow flows chapter.

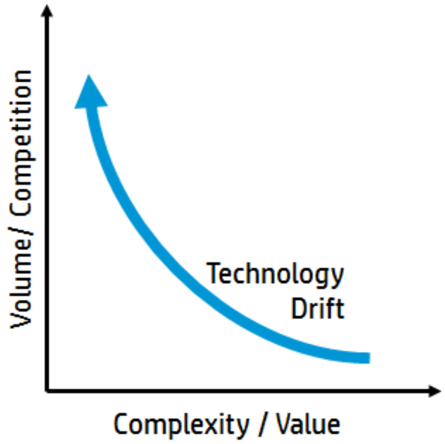
SDN in a nutshell is

"… In the SDN architecture, the control and data planes are decoupled, network intelligence and state are logically centralized, and the underlying network infrastructure is abstracted from the applications …"

Open Networking Foundation

Why SDN?

Figure 1: Network trends and drivers



Network Complexity is increasing:

- Networks are bigger, faster, and applications and services are deployed more dynamically. Both users and applications expect security, resiliency, privacy, traffic separation, end-to-end virtualization and priority treatment.

Hardware Commoditization:

- "Formation of the Open Networking Foundation should be viewed not as the acceptance of OpenFlow by networking vendors, but as the beginning of the real battle to control networking value-add functionality."

  Open Networking Foundation Formed; "The Battle to Commoditize Network Hardware Begins", Gartner, August/2011

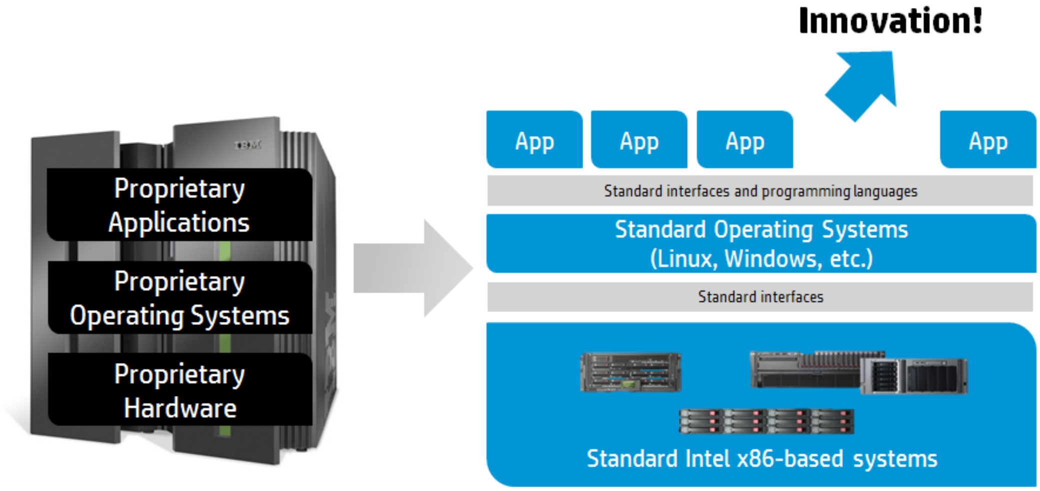
Can networks adapt to the changes taking place?

Traditional networking control mechanisms have been static for many years. There has been very little innovation in the way networks are controlled and managed in the last 10 years.

In contrast, here has been massive development in abstraction in other fields.

Servers are deployed in minutes or seconds. Severs can be moved from one physical host to another using technologies like vMotion. The server move happens at the click of a mouse as directed by an administrator or moved dynamically based on resource utilization. As another example, some HP customers have a 15 minute SLA on a new virtual machine. This means that an end customer or application user can request compute resources in the data centre and expect a 15 minute SLA turn around. It is completely automated. They may be using tools such as V Cloud Director or other tools that VMWare supports. This allows for quick processing and turn-around.

Figure 2: Evolution of server architectures


Proprietary Applications
Proprietary Operating Systems
Proprietary Hardware
Innovation!
App
App
App
App
Standard interfaces and programming languages
Standard Operating Systems (Linux, Windows, etc.)
Standard interfaces
Standard Intel x86-based systems

The same is true for **storage**. Storage can be grown very elastically and quickly. Storage no longer relies on physical disks in each individual server. Disks are removed from physical servers and housed in storage arrays to provide fast and reliable storage for computing and data processing. Storage is abstracted with logical storage and physical storage components. Access to logical storage is provided without regard to physical storage structure. This provides much greater flexibility.

Programming has for many years allowed abstraction. Very few programmers write applications for the physical layer in assembly or some other low-level programming language. Most developers will program using higher level programming languages that allow for rapid development and deployment. The operating systems such as Windows or Mac OS provide abstraction between the application and physical hardware. Object oriented programming allows for reuse of code and the hiding of changes within code using containers. Outside pieces of software do not need to worry about internal changes within a container. Object oriented programming delivers code reuse, scalable solutions, rapid development and quick deployment.

Wireless systems in the past used separate autonomous (fixed) access points. Each access point (AP) was individually configured and managed. Each AP ran its own operating system and had its own security polices and settings. This created an administrative burden when deploying wireless in larger environments. Separate autonomous APs work fine in smaller environments, but when there are hundreds or more, this increases the administrative burden greatly. For example, if a network had 500 APs, that would mean 500 individual configurations, 500 operating systems to keep current, 500 security policies to enforce, 500 QOS polices to control, many hundreds or thousands of SSIDs to setup and so forth.

Things changed in wireless a few years ago with the introduction of controlled (lightweight / managed) APs. A central device would control and configure multiple access points. A team of MSM765zl's could control 800 APs for example. This model provides

- Centralized management of APs

- Automated deployment
- Automated software updates
- Unified policy enforcement
- Unified QOS solutions
- Centralized services such as DHCP

In today's wireless environments, the trend is towards centralized AP management using controllers, rather than distributed autonomous APs.

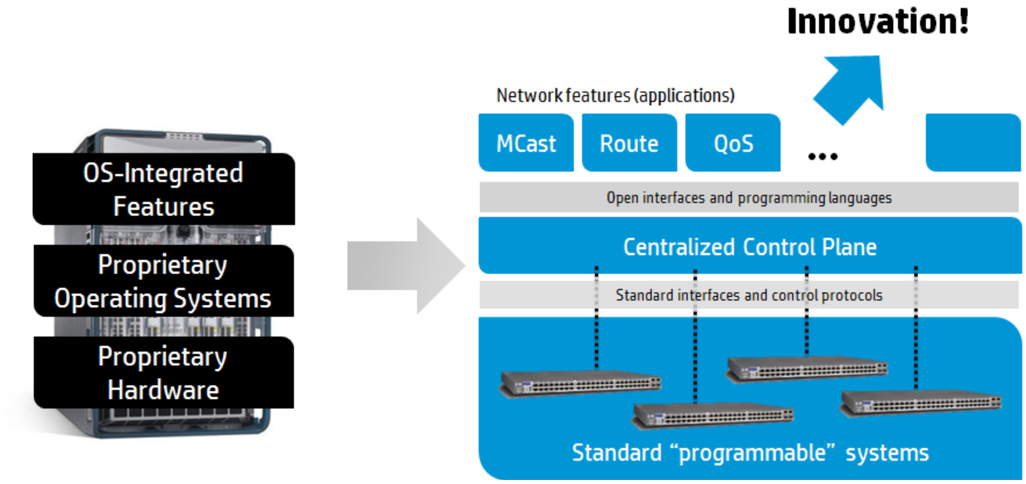
This kind of rapid innovation and quick deployment is still lacking in wired networking today. Network administrators still often configure in a laborious and time consuming manner via the CLI. The CLI is used to configure VLANs and implement how policies are deployed (think about ACLs) for example. Management options such as SNMP have helped with network management, but this is just another method of configuring the localized control plane on each device rather than changing the way networks operate.

Routing protocols such as OSPF still populate the routing table (control plane). OSPF and other routing protocols are local, complex, distributed algorithms. In a network of many devices, each individual device still has its own routing table (control plane) and forwarding information base (FIB / data plane). There is no central control plane replacing protocols such as OSPF or security polices replacing ACLs. Configuration and the control plane are distributed. Each device has its own OS, own configuration, own routing table and so on.

This is exacerbated in a service provider environment because the network is key to generating profits - this applies to both cloud and Internet service providers. Anything that can reduce costs (both CAPEX and OPEX) is viewed in a favorable light.

SDN changes this fundamental setup.

Figure 3: Evolution of network architectures


OS-Integrated Features
Proprietary Operating Systems
Proprietary Hardware
Innovation!
Network features (applications)
MCast
Route
QoS
...
Open interfaces and programming languages
Centralized Control Plane
Standard interfaces and control protocols
Standard "programmable" systems

What does networking look like today? Still the same as it did 10 years ago?

If your network is running Cisco devices or another vendor; and you want to move your core networking, or your core routing to HP, how difficult would it be? If open standard operating systems, tools and processes were being used, it wouldn't be so difficult. But that is often not the case in networking. There are vendor specific integration points in a network that make migrating to another vendor difficult.

HP and others are trying with SDN to follow the same evolution that servers went through. HP wants to move to standards based programmable systems that provide the forwarding data plane locally on devices, but then use a centralized control plan where all of your policy, all of your programming, all your configuration and a lot of the network intelligence is housed.

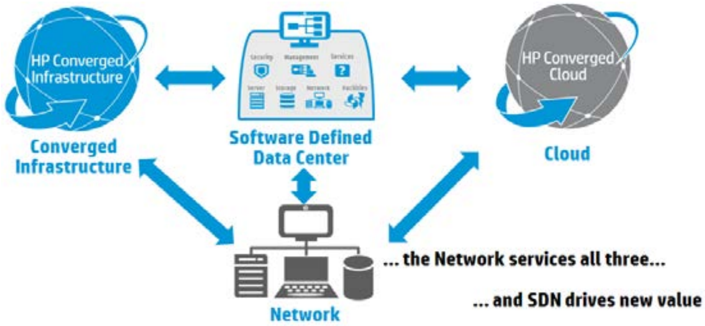
A standard set of open API's are then provided so that applications can be developed for network administration and manipulation. In this case the applications would provide networking features, whether it's multicast, routing, load balancing or firewalling. These services can be provided in a vendor neutral way where customers are not tied to one specific vendor's implementation.

HP's Vision

HP has recognized that there's a transformation occurring in computing. The network is becoming more critical part of that. HP has invested in networking through the acquisition of 3Com as well as the continuous investment in the Provision product line.

HP can now take advantage of these investments by delivering converged infrastructure solutions - the combination of storage, server and networking technologies.

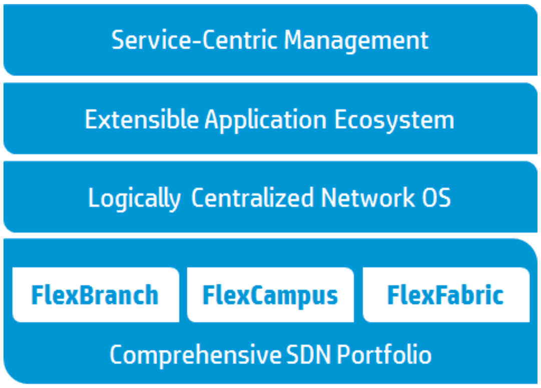
Figure 4: The 3 big infrastructure trends


HP Converged Infrastructure
Converged Infrastructure
Software Defined Data Center
HP Converged Cloud
Cloud
... the Network services all three...
... and SDN drives new value
Network

At the same time there's a need for a high amount of automation in terms of how all the infrastructure gets provisioned. It is impossible to build cloud infrastructure without having all the pieces of the eco system automated.

SDN is the solution that will allow the networking portion of the ecosystem to perform at the same level and with the same agility as storage and servers. SDN allows businesses to move away from managing boxes and infrastructure to focusing more on business development and network enhancement.

Figure 5: HP SDN strategy building blocks


Service-Centric Management

Extensible Application Ecosystem

Logically Centralized Network OS

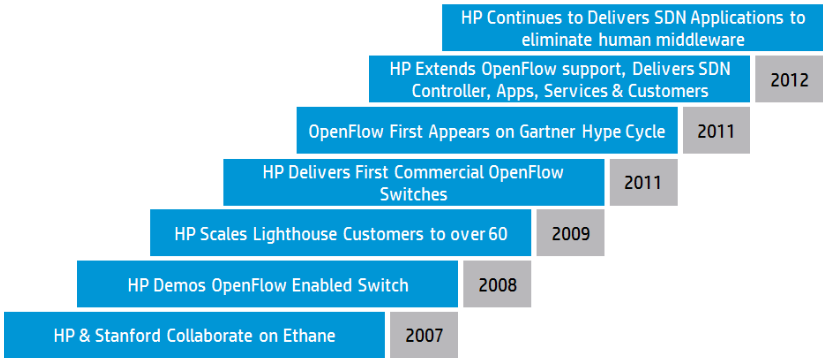
FlexBranch FlexCampus FlexFabric

Comprehensive SDN Portfolio

HP has been involved with SDN development from its beginnings. There has also been rapid industry acceptance of SDN with companies such as Google managing traffic between their data centers using SDN technology (announced in 2012). Thus one of the largest WANs in the world is being managed in production via SDN. Within 3 years of SDN been defined in 2008, the Open Networking Foundation (standards body for SDN) was established in 2011 with companies such as HP, Google, Yahoo and others being members.

Within a few short years, this technology has moved from the lab to large scale production. Why has there been such a rapid acceptance of SDN? New networking technologies such IPv6 haven't been accepted and deployed at this rate?

Figure 6: HP and SDN history


HP Continues to Delivers SDN Applications to eliminate human middleware
HP Extends OpenFlow support, Delivers SDN Controller, Apps, Services & Customers
2012
OpenFlow First Appears on Gartner Hype Cycle
2011
HP Delivers First Commercial OpenFlow Switches
2011
HP Scales Lighthouse Customers to over 60
2009
HP Demos OpenFlow Enabled Switch
2008
HP & Stanford Collaborate on Ethane
2007

Operational Planes

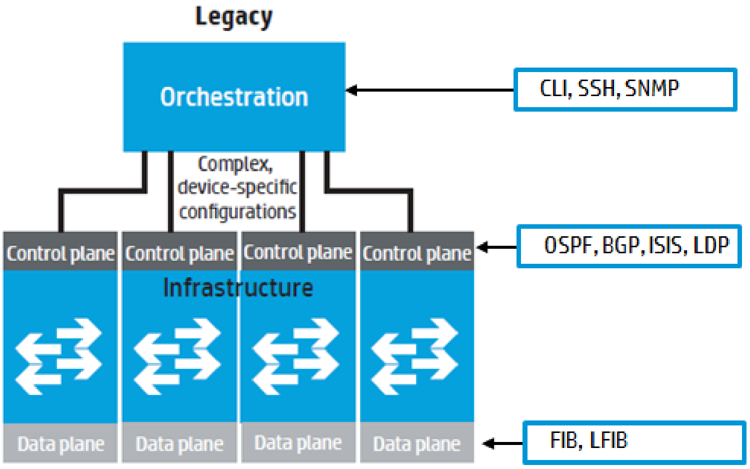
Network devices have three planes of operation:

- Management Plane
- Control Plane
- Forwarding Plane (also known as Data Plane)

Management is still typically done via the CLI. SNMP is also used, but a lot of networks are still managed directly via CLI configuration changes on individual devices.

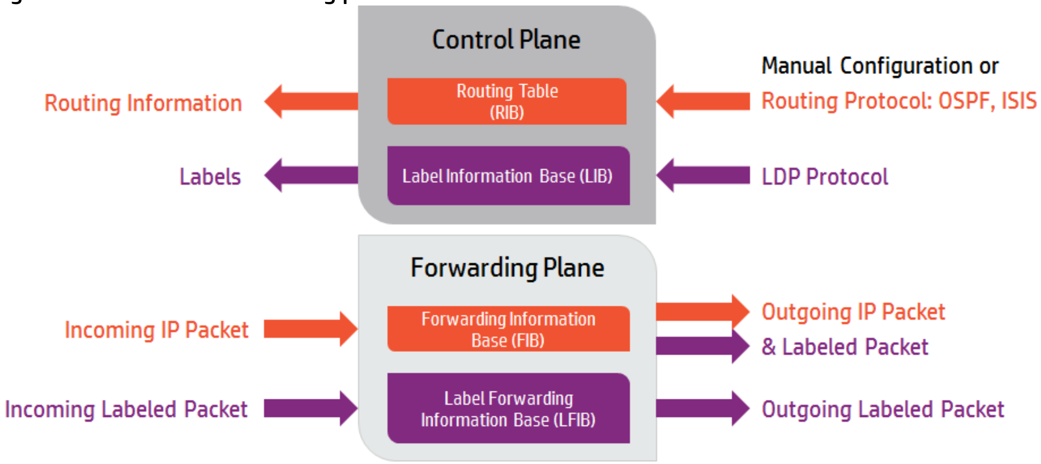
In a traditional router or switch, the forwarding or data plane and the high level routing decisions (control plane) occur on the same device. Examples of control plane protocols include STP, OSPF, BGP, ISIS and LDP. Tables used in the data plane include the Forwarding information base (FIB) or Label Forwarding information base (LFIB).

Figure 7: Operational planes in traditional network device



To emphasize - in a traditional router or switch, both the forwarding or data plane and the high level routing decisions (control plane) occur on the same device.
The control plane is populated either by routing protocols such as OSPF, ISIS, BGP and others or by static routes configured by an administrator. Routes learnt are then advertised to neighboring devices (if valid). This information is in turn sent down to the forwarding table (FIB) also know as the data plane. In an MPLS environment, a protocol such as label distribution protocol (LDP) populates the label information base (LIB) and this is sent down to the Label forwarding information base (LFIB). The control plane thus adds forwarding state to the data plane.
The data plane makes forwarding decisions based on the state information in the FIB (IP), LFIB (MPLS) or MAC-address-table (Layer 2 switching) combined with the information in the frame or header of the packet. The forwarding state + packet header = forwarding decision.

Figure 8: Control and forwading planes functions

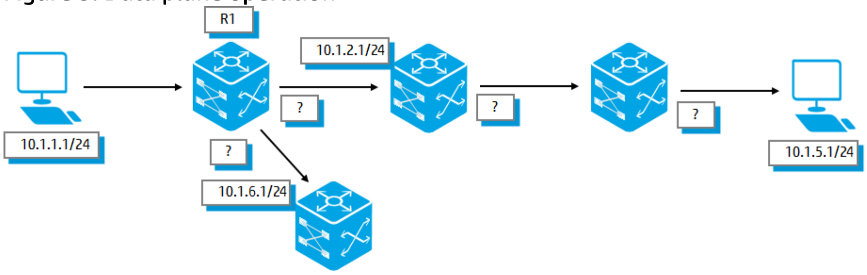


In summary, , when IP packets are received, the FIB is used to forward packets out of a specific interface based on the information contained in the FIB. When an MPLS labeled packet is received, the LFIB is used to process the packet and potentially determine the outgoing interface.

Data Plane Operation

Routers and switches use a forwarding table, called forwarding information base (FIB).

Figure 9: Data plane operation



When a packet arrives at the data plane (HTTP traffic for example) on a router or switch, the data plane processes the packet with local forwarding state. When traffic is sent from 10.1.1.1 to 10.1.5.1, how should the routers process the traffic?

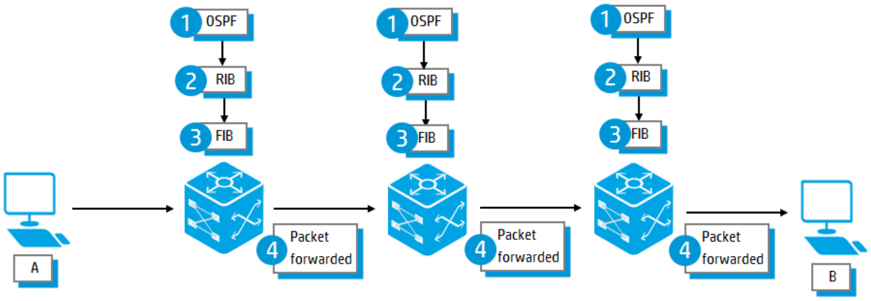
On R1 the FIB on the router has an entry that states that network 10.1.5.0/24 is reachable via next hop 10.1.2.1 out of interface Gigabit Ethernet 0/1. The router will read the destination IP address in the packet (10.1.5.1) and will thus make a forwarding decision based on the forwarding state (FIB table) and packet header.

The data plane decisions need to be made very rapidly and are typically implemented in hardware today via Application Specific Integrated Circuits (ASICs).

Traditional control and data plane interaction

Routing protocols (or static configuration entries) populate routing tables, also called routing information bases (RIB).

Figure 10: Control and data plane interaction



RIBs populate FIBs: packet header + forwarding State = forwarding decision.

There are therefore a number of problems with the current setup of having the data plane, control plane and management plane all vertically integrated into a single platform. As there is a large investment in proprietary code and systems are closed, there is slow innovation.

There is no way to test new functionality or protocols as vendors will not open their code to researchers. SDN calls for open interfaces allowing for rapid innovation.

Distributed configurations can also often be unpredictable and are prone to mistakes. Each network device needs to be individually configured which can lead to an operator error.

Distributed state algorithms such as OSPF are complicated. It is easier to have a central device controlling state information.

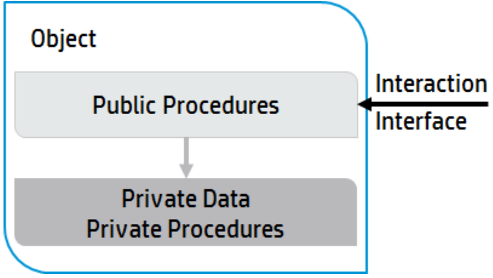
Abstractions

Barbara Liskov received the 2008 Turing Award in March 2009, for her work in the design of programming languages and software methodology that led to the development of object-oriented programming. This was related to programming language and system design, especially related to data abstraction, fault tolerance, and distributed computing.

She sums up that at Computer Science = "*Modularity based on abstraction is the way things get done*".

In simpler terms, abstractions are used to define the relevant interfaces and the end result is a modular scalable system. A modular system allows the reuse of code. The implementation can be changed, but if the interface remains the same, it does not affect other parts of the software system because the parts are separated. Abstractions therefore have great benefits. To build a large-scale software system, modularity based on abstraction is required.

Figure 11: Abstractions and interfaces in object oriented programming



Abstractions in Networking

Data Plan Abstraction

A well known data plane abstraction in use today is the OSI model. The OSI model allows for applications to be developed abstracted from the physical layer or data link layer for example.

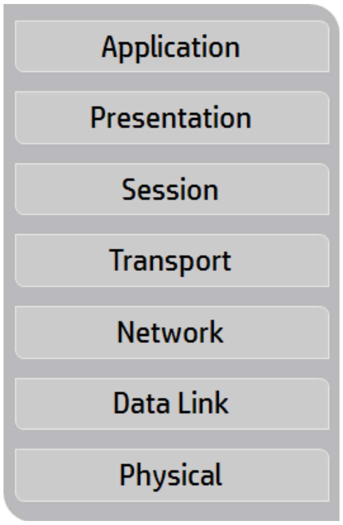
Applications simply rely on

- the transport layer such as TCP or UDP for reliable or unreliable delivery
- which in turn rely on the Network layer for global packet delivery
- which in turn relies on the data link layer for local delivery

- which in turn relies on the physical layer for transfer of actual bits

This is an abstraction model with interfaces that are accessed by higher layer protocols. Each layer is separate and independent of other layers. Higher layer protocols simply access lower layer protocols via interfaces. Changes within one layer do not affect other layers. For example, HTTP traffic could traverse multiple physical layers such as fiber, copper or even air (wireless) without understanding the mechanics of that layer.

Figure 12: Abstractions in the data plane: the OSI model



Innovations in separate layers can take place in parallel and independent of other layers. New developments in 40 Gigabit or 100 Gigabit Ethernet (Physical layer) can take place without waiting for developments in UDP (Transport layer). As long as the interfaces or handles to each layer remain the same, higher layer protocols are not affected.

This has driven major and rapid development of the data plane. Each is a separate problem that can be developed independent of the layer below.

Control Plane Abstraction

What abstractions exist today for the control plane? Answer: There are no real abstractions in the control plane.

There are many mechanisms such as routing, isolation using ACLs and traffic engineering. But, these are not abstractions. There is no modularity with these mechanisms. If a new routing protocol needs to be developed, it will need to be developed from scratch.

The control plane must compute the forwarding state meeting these requirements:

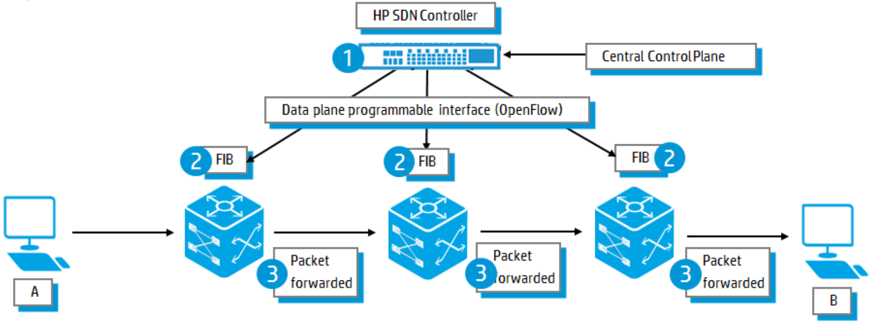
- Problem 1: Compatible with low-level hardware / software devices
- Problem 2: Need to make decisions based on the entire network topology
- Problem 3: Need to configure all switches and routers

SDN Abstractions

In a traditional network environment once again, distributed routing protocols update local control planes on each device. Devices in the network are running a complicated distributed algorithm (think Dijkstra's SPF algorithm for OSPF) for a specific function. This cannot be easily changed. The algorithm would need to be redesigned if changes in function are required.

Routing protocols (or static configuration) populate the RIB. The RIB in turn populates the FIB. Forwarding decisions are made based on entries in the FIB.

Figure 13: Control plane abstraction



In an SDN environment the data path portion still resides on the switch or router, while high-level routing decisions are moved to a separate controller, in this case the HP SDN controller.

The OpenFlow router or switch and controller communicate via the OpenFlow protocol, which defines messages, such as packet-received, send-packet-out, modify-forwarding-table, and get-stats.

Solutions for the Control Plane

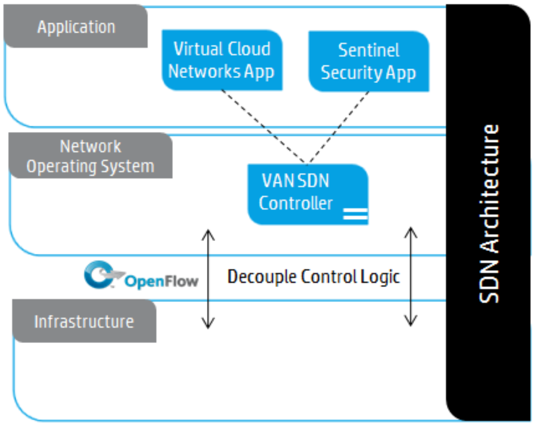
The control plane must compute the forwarding state, meeting these requirements:

- Problem 1: Compatible with low-level hardware / software devices
- Problem 2: Need to make decisions based on the entire network topology
- Problem 3: Need to configure all switches and routers

Solution 1: Abstraction of the forwarding plane

This will hide the complexity of the implementation. Rather than trying to change the operation system code used by network devices, an open interface is used to tell the network devices what to do. This means that there is no longer any concern with regards to specific vendor, or specific device, or specific ASICs used.

Figure 14: SDN architecture



OpenFlow is the interface or protocol used to instruct networking devices what to do with traffic. So, for example, if the packet has a destination of 10.1.1.1, flows within the network device are configured to drop the packet. If a packet has a destination of 10.1.2.1, flows are created within the device to forward the packet out of port GE1/0/1. The configuration is performed based on flow entries <header, action>, rather than routing table entries.

Note

There is often confusion about what SDN and OpenFlow is. Openflow isn't SDN, but is rather a protocol that allows abstraction of hardware. SDN can use the OpenFlow protocol, but doesn't have to. SDN can make use of other protocols instead of using OpenFlow. SDN is a generic term explaining the separation of the control plane and the data plane in networking.

Every network device has one thing in common whether that device is a switch, a router, a firewall, a load balancer, a WAN optimization device or intrusion protection device. Devices receive a packet (packet in), they look at it, they inspect it using the header field or maybe some user data field, and they perform a function. The function can be forwarding in terms of switch or router based on layer-2 and layer-3 information, it can be load balancing, or dropping the packet in the case of a firewall. It could be a multicast feature where you move or forward the packet on multiple ports based on the variety of group settings.

All of these are functions that are performed by a traditional networking device. What SDN does is to separate that function into a central control layer and a distributed forwarding layer.

In the same way as wireless has moved to a central controller (or teamed controllers) and controlled (thin/lightweight) access points, SDN suggests using a central controller and controlled network devices. This allows for reduced complexity and configuration on the network devices and therefore management overhead because these functions are managed from a central location.

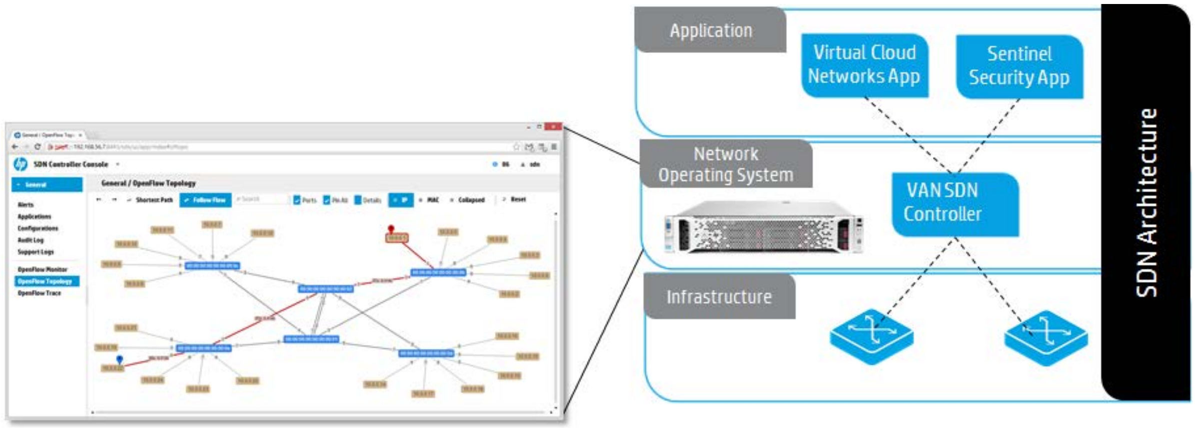
Openflow tries to solve this problem in an open standards way. Unfortunately in the wireless industry even though the ideal solution is a standardized communication protocol between controllers and APs, vendors have gone in different directions so that level of interoperability is still not available. SDN is at the same stage. The industry is trying to use a standards based model to link the controller through a protocol called Openflow which is defined by the ONF. But it remains to be seen if vendors come out with their own implementation of SDN. Will they stay with the standards based protocol Openflow or will they develop their own proprietary protocols?

Solution 2: Abstraction of the network state

The second solution is to move away from complicated distributed algorithms such as OSPF. A network graph can be provided via an API. Network devices can therefore be controlled via this API.

A central logical graph of the network is created on the controller. This can then be accessed via APIs by external applications. The controller runs a network operations system (NOS). The HP SDN controller is the product HP has positioned here. This is either a software product running on Ubuntu Linux or a hardware application. It has full support for the OpenFlow protocol. It has an extensible architecture using open standard protocols such as Java and REST. The controller supports teaming for scalability and redundancy.

Figure 15: SDN controller and the logical graph of the network



The SDN Controller uses OpenFlow to communicate with network devices. Network devices such as switches and routers inform the Controller about themselves as well as inform the controller what neighbors they have. Hosts are discovered when they send traffic. Thus, there is two-way communication between the Controller and the network devices. Routers and switches communicate to the controller with information about the network to form the “view” or topology map. Configurations are sent to the routers and switches to control forwarding. Flow tables are used by network devices rather than routing or MAC address tables.

External applications can manipulate the network via the APIs through the controller using Java or REST APIs. Developers can configure and control the network without having to write software to support multiple proprietary vendor hardware and software

Solution 3: Abstraction of the control plane

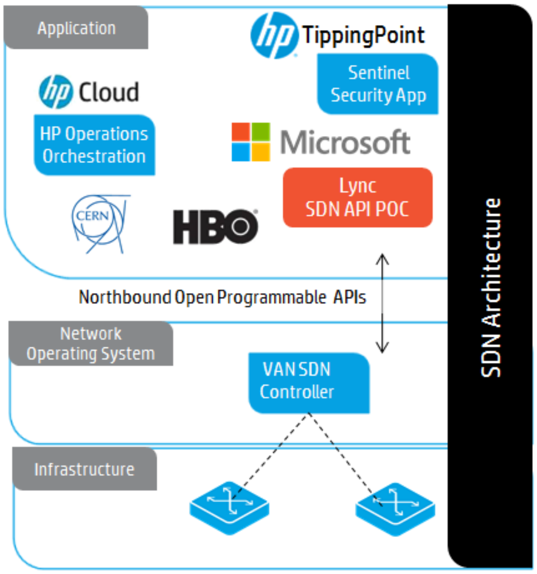
The HP SDN controller provides APIs that can be accessed by applications. External applications can thus manipulate the network via the APIs through the controller using Java or REST. Developers can configure and control the network without having to write software to support multiple proprietary vendor hardware and software. Multiple applications already exist already including HP Virtual Cloud Network for provisioning, HP Sentinel Security App for security, HP UC&C SDN App for Microsoft Lync for QOS and CERN's in house developed app for load balancing.

HP envisions an eco-system of applications that can be made available via an HP app store. These applications could implement policies and forwarding rules. The Controller is simply a container that allows applications to implement state in the network.

Once again, network devices communicate with the Controller using OpenFlow (Southbound). The applications can access the network via Java or REST APIs.

This integration between the Controller and the Applications is known as northbound communication.

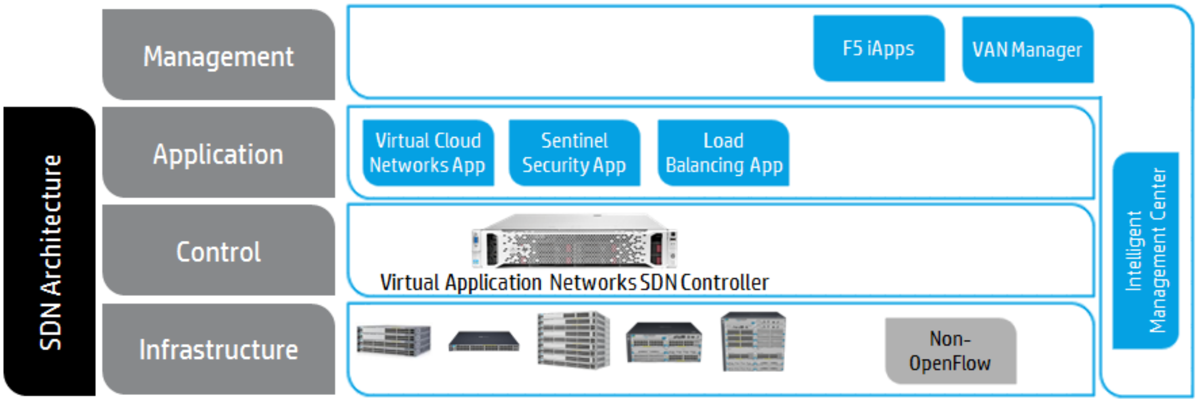
Figure 16: Abstraction of the control plane



SDN Management: IMC

HP has built network operations automation on the Intelligent Management Center which brings automation to non-OpenFlow-enabled devices. HP has also integrated SDN into the Intelligent Management Center (IMC) platform using the Virtual Application Networks Manager (VAN) module for IMC. This enables application deployment in minutes rather than months. Our HP Technology Services Innovation Center in Italy has demonstrated repeatable deployment of a Microsoft Exchange Instance in 5 min. HP have submitted patents on this technology.

Figure 17: HP SDN solution building blocks



Virtual Application Networks includes open RESTful APIs that have enabled integration with F5 iApps for automating deployment of application delivery controllers. Virtual Application Networks is the industry's first policy-based network operations automation from layer 2 -7.

To unlock the full potential of SDN, HP will deliver management for the full SDN architecture integrated into our single pane-of-glass management application – Intelligent Management Center.

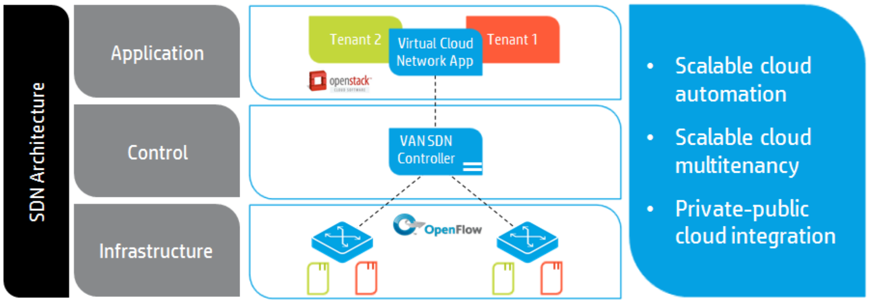
This ensures that management of network services that are automated with SDN work in concert with our VAN automation of the physical network to deliver complete agility.

Application Examples

Virtual Cloud Network Application

HP's Virtual Cloud Network application enables public cloud providers and enterprises to overcome the challenges they face today. Public cloud providers require massive scale and multi-tenancy to deliver competitive cloud services. Yet existing network automation and virtualization solutions have difficulty scaling to the levels a public cloud provider requires.

Figure 18: Virtual cloud network application


SDN Architecture
Application
Control
Infrastructure
Tenant 2
Virtual Cloud Network App
Tenant 1
openstack
VAN SDN Controller
OpenFlow
• Scalable cloud automation
• Scalable cloud multitenancy
• Private-public cloud integration

Enterprises need to interconnect their private environments with their public cloud presences and do so without compromising the integrity of their existing networks. HP's VCN enables the enterprise to securely connect to the cloud and apply its own identity to their cloud environment.

HP's Virtual Cloud Network solution enables the public cloud provider to scale and reduces their risk. First, HP's VCN overlay allows the provider to scale beyond the constraints of current solutions. Second, HP's solution focuses network changes at the edge rather than requiring network changes to occur throughout the entire networking stack, reducing the risk of each change and making automation at scale a reality. Enterprises can now easily interconnect their private environment with the public cloud without compromising the integrity of their networks or making changes.

Finally, continuing HP's commitment to OpenStack, HP's VCN application includes the plug-ins to allow OpenStack's networking service, Neutron, to talk to our SDN VAN controller to request virtual networks, etc

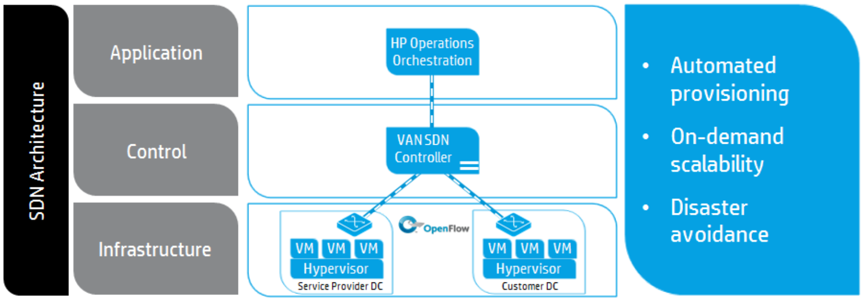
In short, HP's Virtual Cloud Networking fills the critical gaps cloud service providers (public and private) are facing in their networks today, allowing them to move forward and focus on meeting the needs of their businesses.

Dynamic WAN Bandwidth Provisioning Application

HP is working closely with industry leaders such as Verizon and Intel to develop new technologies.

Working with these two companies, a new service offering for service providers is being tested that delivers SDN-based dynamic WAN bandwidth provisioning to support cloud bursting. This technology delivers automated network provisioning, on-demand workload scalability, and disaster avoidance capabilities.

Figure 19: Dynamic WAN bandwitch provisioning application



HP has integrated its Operations Orchestration (OO) software with its SDN controller and the hypervisors located at both customer and service provider data centers.

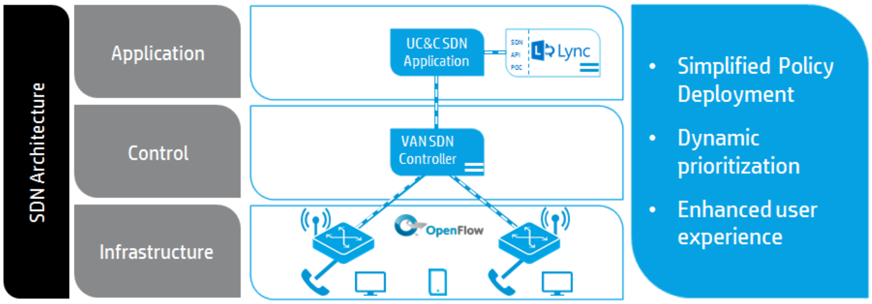
- When a customer data center (Intel) reaches its workload capacity, a cloud burst request is sent to HP Operations Orchestration software.
- HP OO instructs the SDN controller to re-program the OpenFlow-enabled switches at both the customer site and service provider to deliver additional bandwidth, and simultaneously, HP OO initiates new VMs at the service provider's data center to handle the cloud burst.
- When workload levels return to normal, the process is reversed and the VMs are taken down at the service provider and the network is re-provisioned to normal parameters.

UC&C SDN Application for Microsoft Lync

One of the most interesting use cases discussed quite often for SDN is integration of business applications. HP Networking and Microsoft Lync have been collaborating closely to innovate and drive simplification in application policy deployment, from the Cloud all the way to the campus and branch, in a secure and reliable manner.

The Youtube video (http://www.youtube.com/watch?v=byMtYpmh1xQ) demonstrates a technology prototype HP is developing called "Unified Communications & Collaboration SDN Application" for Microsoft Lync to help automate policy deployment across the campus network.

Figure 20: UC&C SDN application for Lync



This demonstration shows a business application interacting directly with the network to simplify policy deployment in Campus and Branch, including dynamic traffic prioritization for real-time multimedia in a secure a trusted manner using OpenFlow.

When a client initiates a call in campus or branch with Lync, whether voice, video or collaboration, the Lync server working with a new Lync SDN API in the data center provides the HP UC&C SDN app the intelligence with call details including source and destination, bandwidth requirements, quality statistics to name a few. The HP UC&C SDN app for Lync uses this information to program the network via the HP Virtual Application Network SDN controller using OpenFlow.

Once the policy is implemented, the call is simultaneously completed to the destination client. The UC&C SDN app uses the intelligence from Lync server along with the robust capabilities of the HP SDN controller to implement best path routing, consistent quality-of-service, and bandwidth guarantees. All of this is done dynamically with the SDN controller providing the central point of control eliminating the need for manual, device-by-device, configuration via the CLI, greatly simplifying policy deployment.

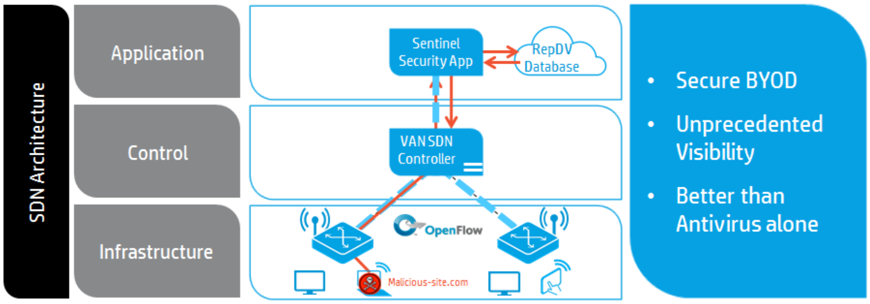
One of the biggest issues facing our customers today is dealing with the increasing number of soft-phones or soft-clients and BYOD. This SDN app ensures the best user experience by dynamically configuring the right policy for latency and bandwidth without ever touching the CLI.

Sentinel Security Application

HP's Sentinel Security Application was announced late last year.

The Sentinel Security application works by instructing the HP Virtual Application Networks SDN controller to program the access layer switches using OpenFlow to intercept malicious traffic by working with HP TippingPoint RepDV Reputation Database. The RepDV database contains information of over 1 Million botnet, malware and spyware sites.

Figure 21: Sentinel Security Application

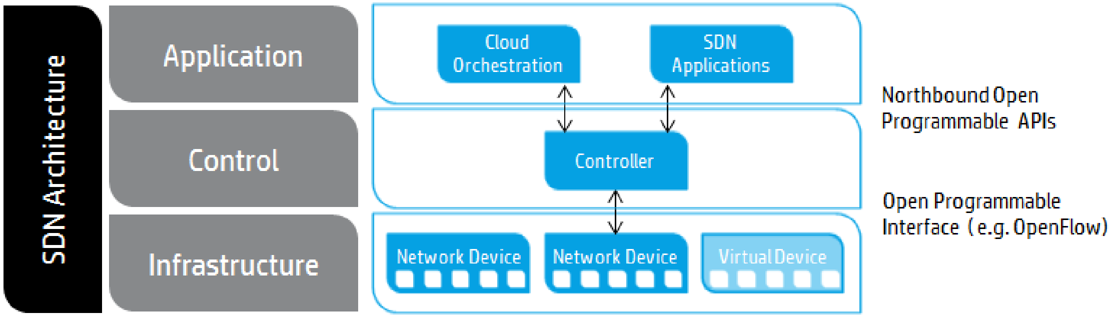


In a BYOD world, the risk of botnet threats and malware exploits are greater than ever. As a user connects to the enterprise network, there is a need to immediately detect this kind of network activity and throttle that behavior before infection spreads.

Summary

SDN abstracts control plane. Network devices support a standard programmable interface with a protocol like OpenFlow – so a heterogeneous network can be programmed in a unified manner. The network devices can be programmed by the SDN controller using the OpenFlow protocol. OpenFlow thus provides an open standards interface to vendor network devices.

Figure 22: SDN Architecture



The API interfaces on the SDN controller integrate with cloud orchestration tools like OpenStack and enable SDN applications that deliver networks services (e.g., network virtualization, intrusion detection, load balancing). Application programmers can ignore the physical network topology and work on abstract topology diagram as provided by the SDN controller. There are other applications such as Microsoft Lync that can take advantage of SDN to implement technologies such as QoS.

Advantages

There are major advantages to using SDN. Rather than having a complicated distributed algorithm that needs to be supported on all network devices, applications can be written to manipulate network devices via the APIs on the SDN controller.

It is easier to coordinate behavior and polices in an SDN environment. So, a network control policy is easier to implement without affecting other network functions such as routing or load balancing using SDN than it is currently where a network administrator need to configure each individual device with a low level vendor specific language.

It is easier to evolve as the control plane is decoupled from network devices. The control plane or software that interfaces with the controller can indenpently evolve without making changes to vendor specific network devices or ASICs.

Resources, contacts, or additional links

HP SDN: http://hp.com/sdn

HP SDN SDK and documentation: http://sdndevcenter.hp.com

Learn more at
hp.com/networking

Sign up for updates

hp.com/go/getupdated

Share with colleagues

Rate this document

October 2013