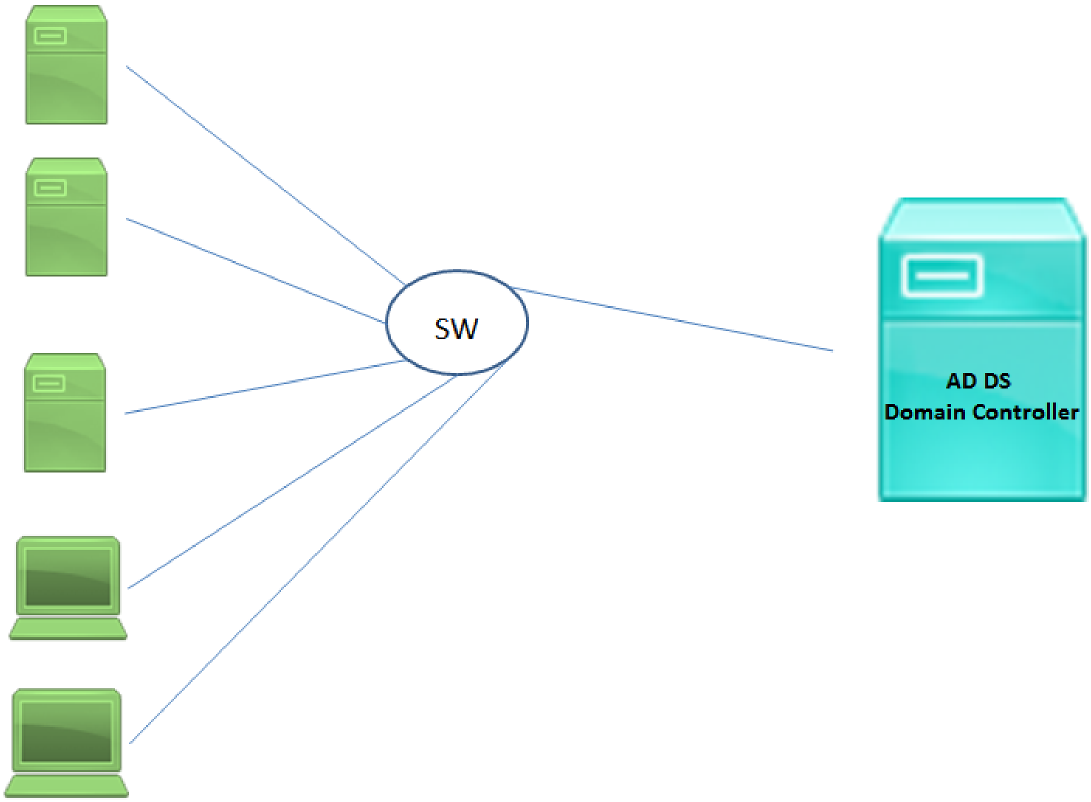
SDN Main Concept

If we want to know what is SDN we should talk a little bit about Directory service such as the one used in Microsoft world.

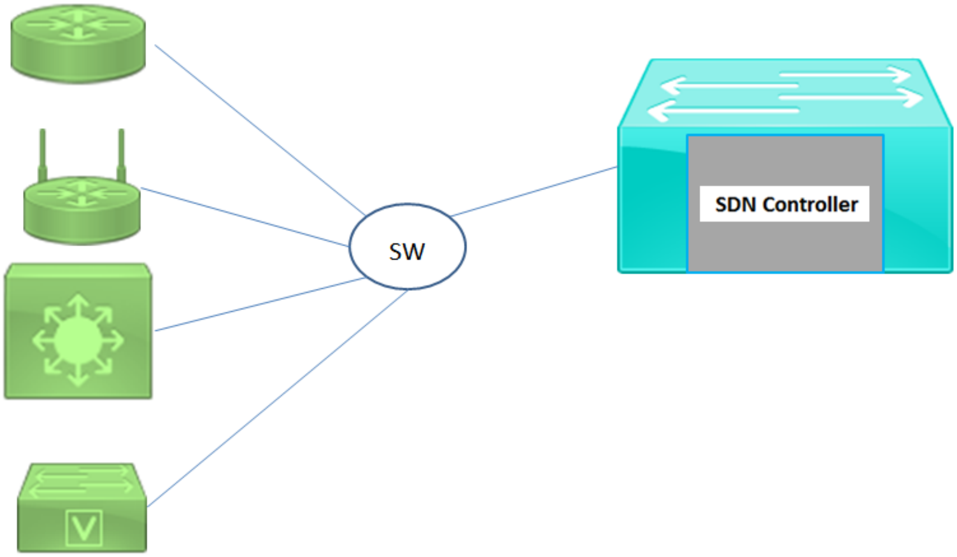
In old days we used to have servers and client machines working individually without any means of centralized administration or automation , later we start using the concept of domain controller which is a server can control all other machines , push our policies to clients once we configure it and even getting statistics & events information from clients for further analysis.



And the question here, can we have a similar concept with network devices such as routers , layer 2 switches and multilayer switches ….etc.

Answer is yes by using SDN (Software Defined Networking) which mean the concept of software (programmatic) control of the network, rather than the more static configuration-controlled networking. The term network programmability itself refers to more focus on software control of the network

And to do so we will use SDN controller, which is a software will let us control how routers and switches work in all possible different situations and this will be done according to our instant needs which make network devices configuration a dynamic & rapid process.



To fully understand how SDN controller will work for us, we need to understand how Network devices such as routers and switches works by divide their jobs to three different layers or planes.

The Data Plane (aka Forwarding Plane)

Refers to the jobs that a networking device does to receive, process, and forward a packets and frames.

Common jobs that a networking device does in the data plane:

- De-encapsulating and re-encapsulating a packet in a data link frame (routers, Layer 3 switches)
- Adding or removing an 802.1Q trunking header (routers and switches)
- Matching the destination MAC address to the MAC address table (Layer 2 switches)
- Matching the destination IP address to the IP routing table (routers, Layer 3 switches)
- Encrypting the data and adding a new IP header (for VPN processing)
- Changing the source or destination IP address (for NAT processing)
- Discarding a message due to a filter (ACLs, port security)

The Control Plane

Without it the Data Plane cannot do anything meant to do since Data Plane take its own jobs decision based on what Control Plane provide, that is why Routing tables, a MAC address table , routing protocols …etc. Will be exists in this plane.

Common control plane protocols:

- Routing protocols (OSPF, EIGRP, RIP, BGP)
- IPv4 ARP
- IPv6 NDP
- Switch MAC learning
- STP

The Management Plane

Not directly impact the data plane. But will include protocols that allow network engineers to manage the devices.

Common management plane protocols:

- Telnet
- SSH
- HTTP/HTTPS

Note: when it comes to switches , LAN switches needed a faster data plane than a generalized CPU could process in software so switches have always had specialized hardware to perform data plane processing.
The switching logic occurs not in the CPU with software, but in an application-specific integrated circuit (ASIC). An ASIC is a chip built for specific purposes, such as for message processing in a networking device.
The ASIC needs to perform table lookup in the MAC address table, so for fast table lookup, the switch uses a specialized type of memory to store the equivalent of the MAC address table: ternary content-addressable memory (TCAM).

So to simply summarize SDN Architecture we can say that the networking devices still exist, and still forward data, but the control plane functions and location can change dramatically.

SDN Controllers & SBI/NBI

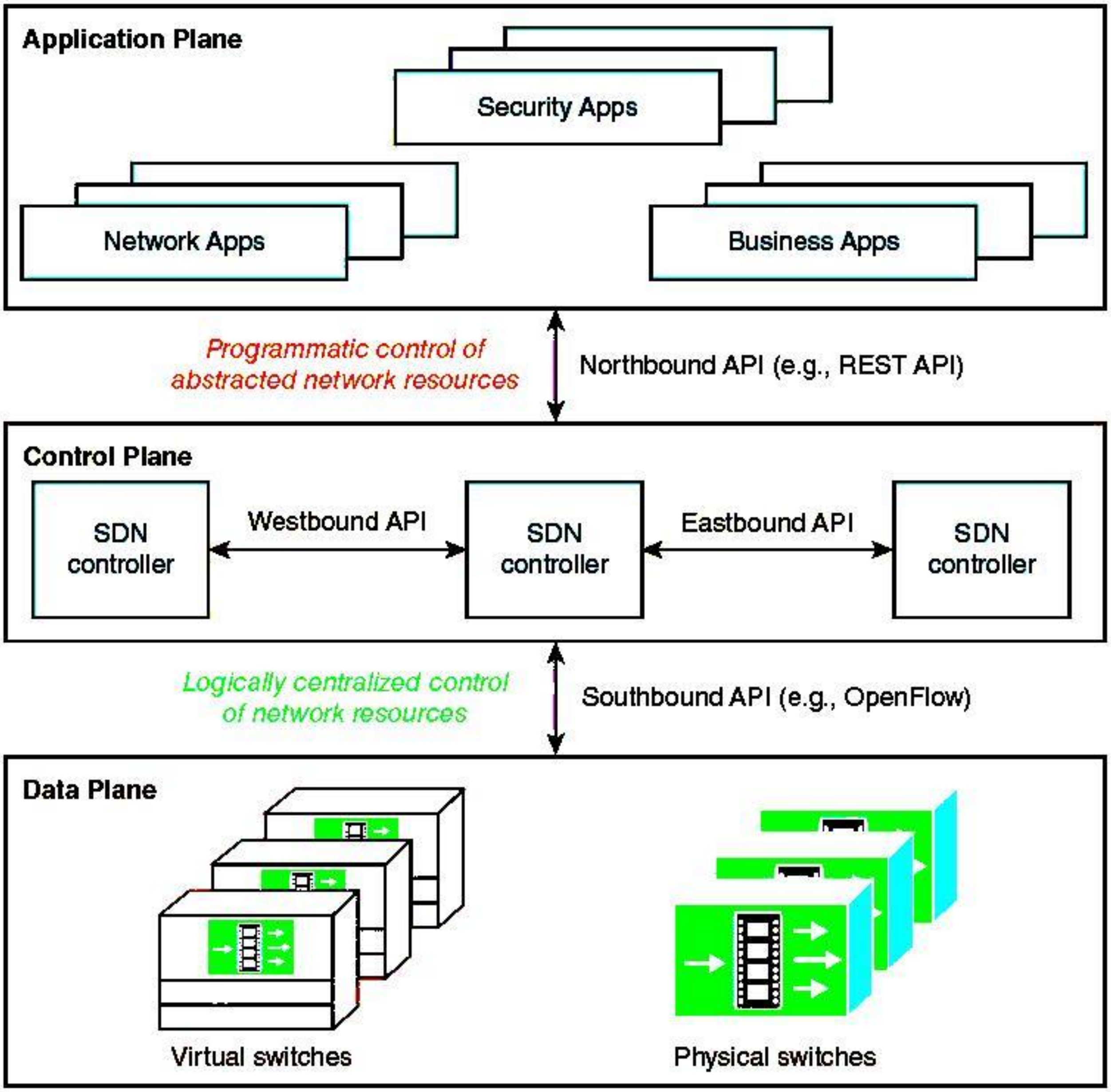
So as I said before SDN controllers will act like domain controllers so instead of using distributed control plane we will use centralized control plane.

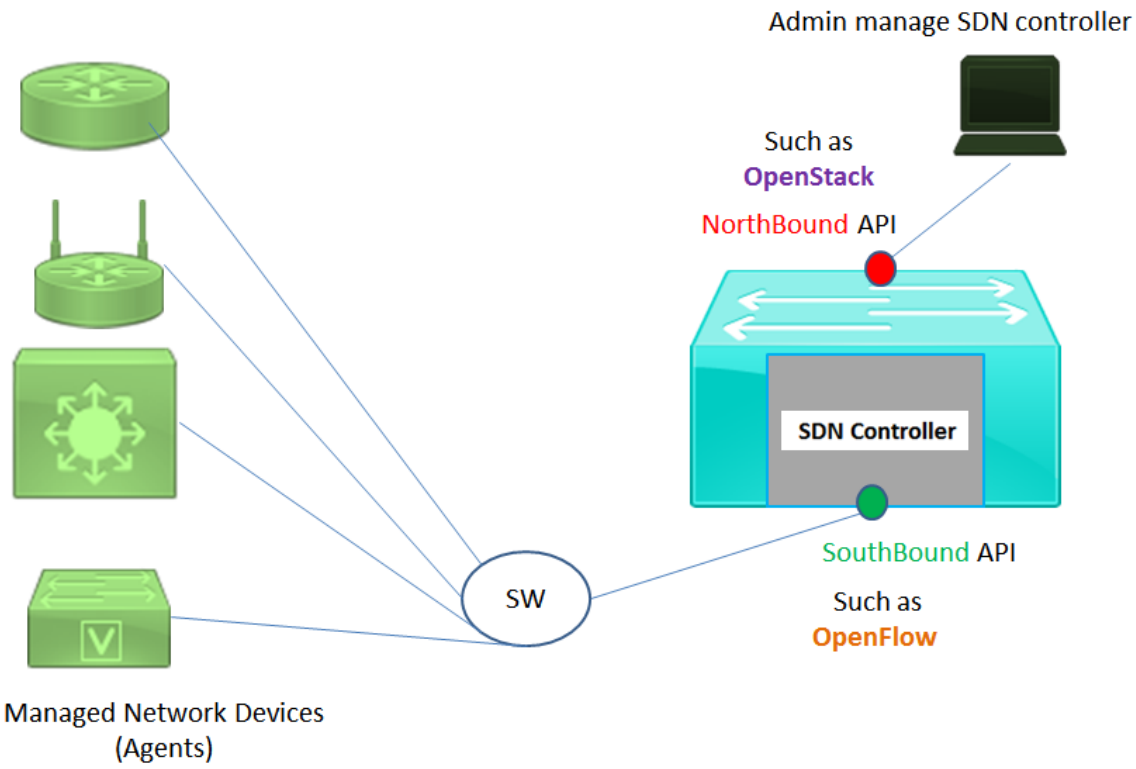
SDN Controller will take all or parts of control plane from network devices and do it by itself in centralized fashion then tell network devices what to do.

A controller does much of the work needed for the control plane in a centralized control model.
So Simply

- SDN controller will make network more smart and can take decisions for them and deploy many network devices automatically
- SDN controller can handle many vendor with standard language
- SDN controller push changes and can rollback if need it to network devices (aka **agents**)
- SDN controller is real time situational awareness , for example can detect congestions
- SDN controller comes in physical or virtual with GUI or CLI or both to access it
- SDN controller main goal is manage the flow of packets on the network

Flow: is communication between device and another device such as router
SDN flow: can send configuration to network device and let it do something else with the flow


Application Plane
Security Apps
Network Apps
Business Apps
Programmatic control of abstracted network resources
Northbound API (e.g., REST API)
Control Plane
SDN controller
Westbound API
SDN controller
Eastbound API
SDN controller
Logically centralized control of network resources
Southbound API (e.g., OpenFlow)
Data Plane
Virtual switches
Physical switches



Remember, SDN is middle man between management interface and network device

SDN controllers have **northbound interface NBI** and **southbound interface SBI**
Northbound up   to management plane (interface connect to admin)
Southbound down to data plane (interface control traffic)

So from **northbound** we manage SDN controller
From **southbound**, SDN controller manage network devices

Most SDN solutions have their own user interface (**northbound** API) such as , **openstack** , VMware vSphere web client .
And **Southbound** API such as **OpenFlow** (standard) or **onePK** API (cisco proprietary)

Remember, the word "interface" when we talking about SBI, NBI, and API is refers to software interfaces

Application programming interface API : is a method for one application (program) to exchange data with another application. Rearranging the words to describe the idea, an API is an interface to an application program.

SBI: It is an interface between a program (the controller) and a program (on the networking device) that lets the two programs communicate.
SBI examples:

- OpenFlow (from the ONF; https://www.opennetworking.org)
- OpFlex (from Cisco; used with ACI)
- CLI (Telnet/SSH) and SNMP (from Cisco; used with APIC-EM)

NBI: opens the controller so its data and functions can be used by other programs, enabling network programmability, with much quicker development. Programs can pull information from the controller, using the controller's APIs. The NBIs also enable programs to use the controller's abilities to program flows into the devices using the controller's SBIs.
NBI examples:

- REST API
- OpenStack
- The API Virtual Private Cloud VPC

REST (Representational State Transfer): describes a type of API that allows applications to sit on different hosts, using HTTP messages to transfer data over the API.

OpenStack: is a cloud operating system that controls large pools of compute, storage, and networking resources throughout a datacenter, all managed through a dashboard that gives administrators control while empowering their users to provision resources through a web interface. **https://www.openstack.org/software/**

onePK: is an element within Cisco's software defined networking (SDN) strategy. onePK is an easy-to-use toolkit for development, automation, rapid service creation, and more.

OpenFlow

Is an open interface for remotely controlling the forwarding tables in network switches, routers, and access points.
When we use OpenFlow we can call the SDN controller a "OpenFlow Controller" and call the switch support it a "OpenFlow Switch".
OpenFlow is implemented on top of TLS which providing s secure OpenFlow Channel.

Why OpenFlow?

Within each proprietary switch is data plane and a control plane and there is huge different between the control planes of switches from different vendors , vendors will never expose their control plane or support anything will expose it , this make it not easy to be programmable .

To solve this issue we used openflow it is open source control protocol that all the vendors support.
API can be placed in vendor switch enabling it to be programmed without exposing the vendor switch code

Open Flow enabled Switches & OpenFlow Controllers

The OpenFlow switches must consult an OpenFlow controller each time a decision must be made.

OpenFlow switches come in two varieties:

1. pure (OpenFlow-only).
2. hybrid (OpenFlow-enabled).

Hybrid switches support OpenFlow in addition to traditional operation and protocols.
Most commercial switches available today are hybrids.

An OpenFlow switch consists of a flow table, which performs packet lookup and forwarding.
Each flow table in the switch holds a set of flow entries that consists of:
1. **Header fields** or match fields, with information found in packet header, ingress port, and metadata, used to match incoming packets.
2. **Counters**, used to collect statistics for the particular flow, such as number of received packets, number of bytes, and duration of the flow.
3. **A set of instructions or actions** to be applied after a match that dictates how to handle matching packets. For instance, the action might be to forward a packet out to a specified port.

Pure OpenFlow switches only support the Required Actions, while hybrid OpenFlow switches may also support the NORMAL action. Either type of switches can also support the FLOOD action.
The Required Actions are:

- Forward
- Drop
- Enqueue
- Modify field

OpenFlow messages use Transport Layer Security (TLS)
The controller's default TCP port is 6633. The switch and controller mutually authenticate by exchanging certificates signed by a site-specific private key.

The OpenFlow protocol defines three message types, each with multiple subtypes:
• Controller-to-switch • Symmetric • Asynchronous

For OpenFlow Controllers we can see few examples in the market such as Floodlight
, NOX and POX
NOX (www.noxrepo.org) was the first OpenFlow controller written in C++ and provides API for Python too but New NOX only supports C++.

POX is Python-only version of NOX. It can be considered as a general, open source OpenFlow controller written in Python, and a platform for rapid development and prototyping of network applications.

Floodlight is a Java-based OpenFlow controller, based on the Beacon implementation, which supports both physical and virtual OpenFlow switches.

Cisco SDN Solutions

There are three different SDN and network programmability solutions available from Cisco.

1. Open SDN Controller and OpenFlow
2. Cisco Application Centric Infrastructure (ACI) and OpFlex
3. Cisco APIC Enterprise Module (APIC-EM)

1-Open SDN and OpenFlow

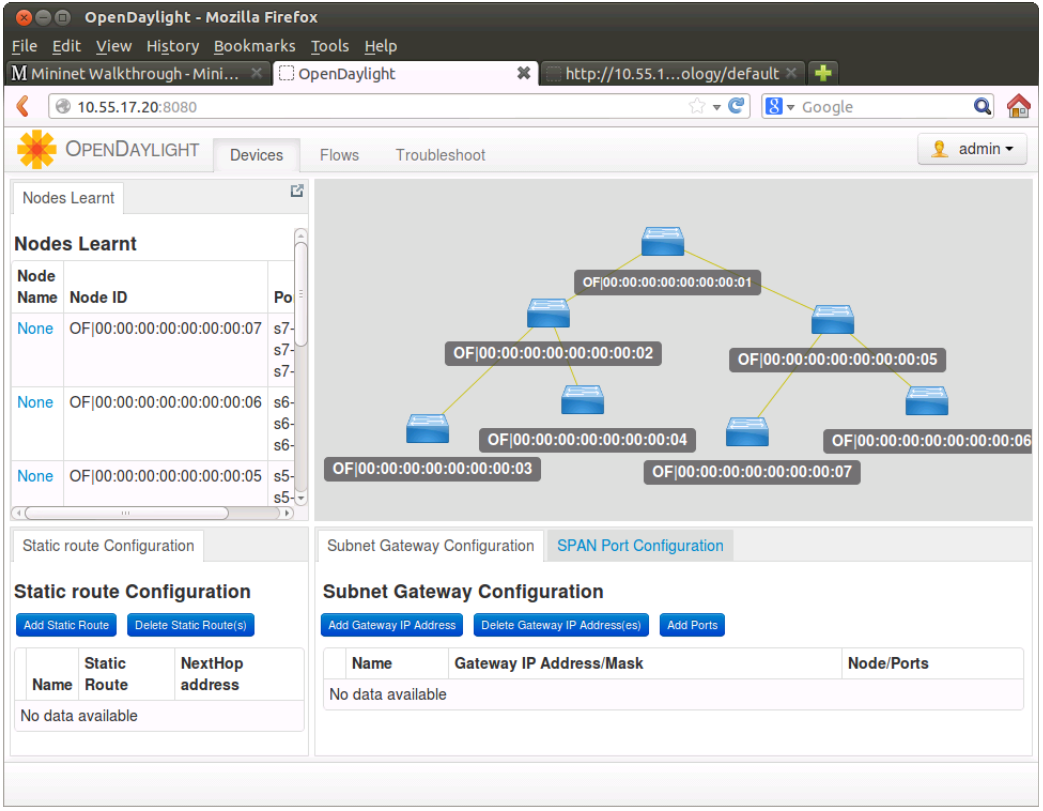
- Comes from the Open Networking Foundation ONF (https://www.opennetworking.org)
- The ONF model of SDN features OpenFlow as the SBI.
- The Open SDN model centralizes most control plane functions, with control of the network done by the controller plus any applications that use the controller's NBIs.

Most common SDN controllers based on ONF are:

- The OpenDaylight controller
- Cisco Open SDN Controller

OpenDaylight SDN controller (https://www.opendaylight.org)

- The OpenDaylight (ODL) project exists as a project of the Linux Foundation. Every one can use its source and create his own modified controller
- It supports variety of SBIs OpenFlow, NetConf, PCEP, BGP-LS, and OVSDB and more .
- Any vendor can take ODL and , add to it, and create a commercial ODL controller.
- The OpenDaylight.org website listing of 15 commercial SDN controllers based on ODL, including the Cisco Open SDN Controller
- ODL is developed using Java and as a JVM it can run on any hardware platform and OS provided it supports Java JVM 1.7 and higher.


OpenDaylight - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Mininet Walkthrough - Mini...
OpenDaylight
http://10.55.1...ology/default
10.55.17.20:8080
Google
OPENDAYLIGHT
Devices
Flows
Troubleshoot
admin
Nodes Learnt
OF|00:00:00:00:00:00:00:01
OF|00:00:00:00:00:00:00:02
OF|00:00:00:00:00:00:00:05
OF|00:00:00:00:00:00:00:04
OF|00:00:00:00:00:00:00:06
OF|00:00:00:00:00:00:00:03
OF|00:00:00:00:00:00:00:07
Static route Configuration
Subnet Gateway Configuration
SPAN Port Configuration
Add Static Route
Delete Static Route(s)
Add Gateway IP Address
Delete Gateway IP Address(es)
Add Ports

Nodes Learnt

Node Name	Node ID	Po
None	OF|00:00:00:00:00:00:00:07	s7- s7- s7-
None	OF|00:00:00:00:00:00:00:06	s6- s6- s6-
None	OF|00:00:00:00:00:00:00:05	s5- s5-

Static route Configuration

Name	Static Route	NextHop address
No data available		

Subnet Gateway Configuration

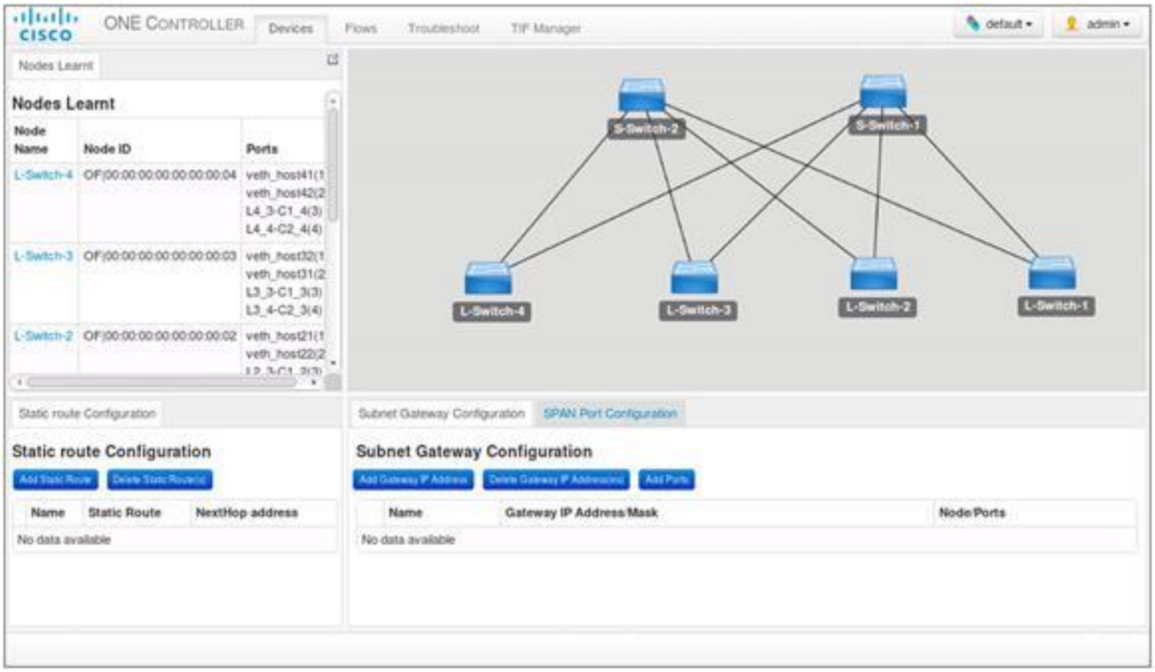
Name	Gateway IP Address/Mask	Node/Ports
No data available		

The web-based GUI of ODL controller

Cisco Open SDN Controller

- SDN controller and it is Cisco's commercial version of ODL.

Note: Cisco does support the OpenFlow and ONF model of OpenFlow , many Cisco products are supporting OpenFlow such as some models of Cisco Nexus switches, plus some Cisco ASR series routers and many other cisco routers and switches supporting OpenFlow.
But Cisco does not appear to be setting about to migrate its entire product line to support OpenFlow



Nodes Learnt

Node Name	Node ID	Ports

L-Switch-4	OF|00:00:00:00:00:00:00:04	veth_host41(1
		veth_host42(2
		L4_3-C1_4(3)
		L4_4-C2_4(4)

L-Switch-3	OF|00:00:00:00:00:00:00:03	veth_host32(1
		veth_host31(2
		L3_3-C1_3(3)
		L3_4-C2_3(4)

L-Switch-2	OF|00:00:00:00:00:00:00:02	veth_host21(1 veth_host22(2 L2_3-C1_2(3)

Static route Configuration

Name	Static Route	NextHop address
No data available		

Subnet Gateway Configuration

Name	Gateway IP Address/Mask	Node/Ports
No data available		

The web-based GUI of Cisco Open SDN controller

<u>2-The Cisco Application Centric Infrastructure ACI</u> **(Important for DC students)**

- The end goal of this solution is about enabling software control of the network and how it operates, so that software can automate and change the network based on current conditions in the network.
- This solution focused on the data center.
- Cisco made the network infrastructure become application centric, hence the name of the Cisco data center SDN solution: Application Centric Infrastructure, or ACI.
- ACI uses a concept of endpoints and policies. The endpoints are the VMs (or even traditional servers with the OS running directly on the hardware). Because several endpoints have the same needs, you group them together into aptly named endpoint groups. Then policies can be defined about which endpoint groups can communicate with whom—for instance, a group of web servers may need to communicate with a group of application servers. The policy also defines other key parameters, like which endpoint groups can access each other (or not), as well as QoS parameters and other services.
- ACI uses a centralized controller called the Application Policy Infrastructure Controller (APIC),It is the controller that creates application policies for the data center infrastructure.

- ACI uses a partially centralized control plane, RESTful and native APIs, and OpFlex as an SBI. The NBIs allow software control from outside the controller. The controller communicates with the switches connected to the endpoints, and asks those switches to then create the correct flows to be added to the switches. Interestingly.

Cisco ACI Main Components

- **Nexus 9000 switches**

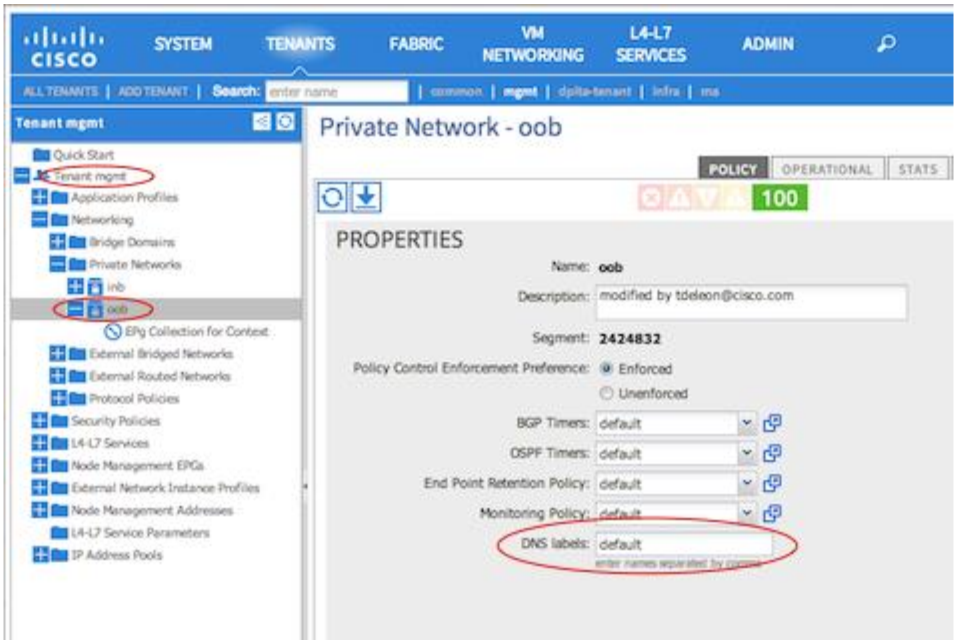
These devices can become part of an ACI fabric through a variant of the NX-OS operating system called ACI Fabric OS.

- **Application Policy Infrastructure Controller (APIC)**

This network controller is responsible for provisioning policies to physical and virtual devices that belong to an ACI fabric.

- **Ecosystem**

APIC handles the interaction with other solutions besides Nexus 9000 switches, which include Cisco Adaptive Security Appliances (ASA) firewalls, Cisco Application Virtual Switch (AVS), VM managers such as VMware vCenter, Microsoft System Center Virtual Machine Manager (SCVMM), application delivery controllers from companies such as F5 and Citrix, and cloud orchestration systems such as OpenStack.



Private Network - oob

PROPERTIES

Name: oob

Description: modified by tdeleon@cisco.com

Segment: 2424832

Policy Control Enforcement Preference: ● Enforced ○ Unenforced

BGP Timers: default

OSPF Timers: default

End Point Retention Policy: default

Monitoring Policy: default

DNS labels: default

enter names separated by comma

The web-based GUI of APIC controller

3-The Cisco APIC Enterprise Module (Important for R&S students)
Now maybe you notice that these two first solutions move significant parts of the control plane functions into the controller and completely depends that switches and other managed devices should support them.

In Open SDN solution we centralize most of the control plane and switches must support OpenFlow.

In Cisco ACI solution we centralizes much but not all of the control plane, leaving some of the control plane in the switches and switches must support ACI (normally only newer models of switches with software that supports ACI).

As you can see Neither the Open SDN model nor the Cisco ACI model uses Traditional switches and routers.

So Cisco made this third SDN solution, APIC Enterprise Module (APIC-EM) which deal with traditional switches and routers

The Cisco Application Policy Infrastructure Controller - Enterprise Module (APIC-EM) is Cisco's SDN Controller for Enterprise Networks (Access, Campus, WAN and Wireless).

To help with network programmability, the solution uses a centralized controller. At the same time, it attempts to support much of the more recent generations of Cisco enterprise routers and switches by using SBIs

- So, SDN controller here called APIC-EM controller.
- Cisco supplies a variety of applications that reside on the controller
- The controller has a RESTful Northbound API
- The control and data planes of the network devices remain unchanged, as part of the effort to support existing devices
- The SBI uses familiar protocols: Telnet, SSH, and SNMP.

- APIC-EM enables easier network automation for customers. To do that, APIC-EM gathers information about the network over the SBI.
- That information includes topology, devices, interfaces, operational status, and configuration. Next, APIC-EM makes that information available through extensive NBI APIs.
- APIC-EM can still change how the devices operate by changing the configuration of the devices.
- APIC-EM can use Telnet and SSH to log in to a device, use the CLI, and issue commands—including reconfiguring the device.

The Cisco APIC-EM platform supports both wired and wireless enterprise networks across the Campus, Branch and WAN infrastructures. It offers the following benefits:

- Creates an intelligent, open, programmable network with open APIs
- Saves time, resources, and costs through advanced automation
- Transforms business intent policies into a dynamic network configuration
- Provides a single point for network wide automation and control

Cisco APIC Enterprise Module Features and Benefits

Feature	Description
Network Information Database (NIDB)	The Cisco APIC-EM periodically scans the network to create a “single source of truth” for IT. This inventory includes all network devices, along with an abstraction for the entire enterprise network.
Network topology visualization	The Cisco APIC-EM automatically discovers and maps network devices to a physical topology with detailed device-level data. You can use this interactive feature to troubleshoot your network.
EasyQoS	The EasyQoS feature enables you to configure quality of service on the devices in your network that have been discovered by the Cisco APIC-EM. Using EasyQoS, you can group devices and then assign classes of service to those devices. The Cisco APIC-EM takes your QoS selections, translates them into the proper device configurations, and deploys the configurations onto those devices.
Cisco Network Plug and Play application	The Cisco Network Plug and Play solution is a converged solution that extends across Cisco's enterprise portfolio. It provides a highly secure, scalable, seamless, and unified zero-touch deployment experience for customers across Cisco routers, switches and wireless access points.
Cisco Intelligent WAN (IWAN) application	The separately licensed IWAN application for APIC-EM simplifies the provisioning of IWAN network profiles with simple business policies. The IWAN application defines business-level preferences by application or groups of applications in terms of the preferred path for hybrid WAN links. Doing so improves the application experience over any connection and saves telecom costs by leveraging cheaper WAN

Feature	Description
	links.
Public Key Infrastructure (PKI) server	The Cisco APIC-EM provides an integrated PKI service that acts as Certificate Authority (CA) to automate X.509 SSL certificate lifecycle management. Applications, such as IWAN and PnP, use the capabilities of the imbedded PKI service for automatic SSL certificate management.
Path Trace application	The path trace application helps to solve network problems by automating the inspection and interrogation of the flow taken by a business application in the network.
High Availability (HA)	HA is provided in N+ 1 redundancy mode with full data persistence for HA and Scale. All the nodes work in Active-Active mode for optimal performance and load sharing.
Back Up and Restore	The Cisco APIC-EM supports complete back up and restore of the entire database from the controller GUI.

APIC-EM Apps

APIC-EM comes with many apps inside it and we can use it to test and manage our network , some of these apps are free to use and called “Basic apps” and some of the apps need a to purchase a license from Cisco so we can use it and called “ Solution apps” .
Also remember APIC-EM itself is free to download from Cisco.

Let’s talk about two important “Basic” apps in APIC-EM

APIC-EM Path Trace app

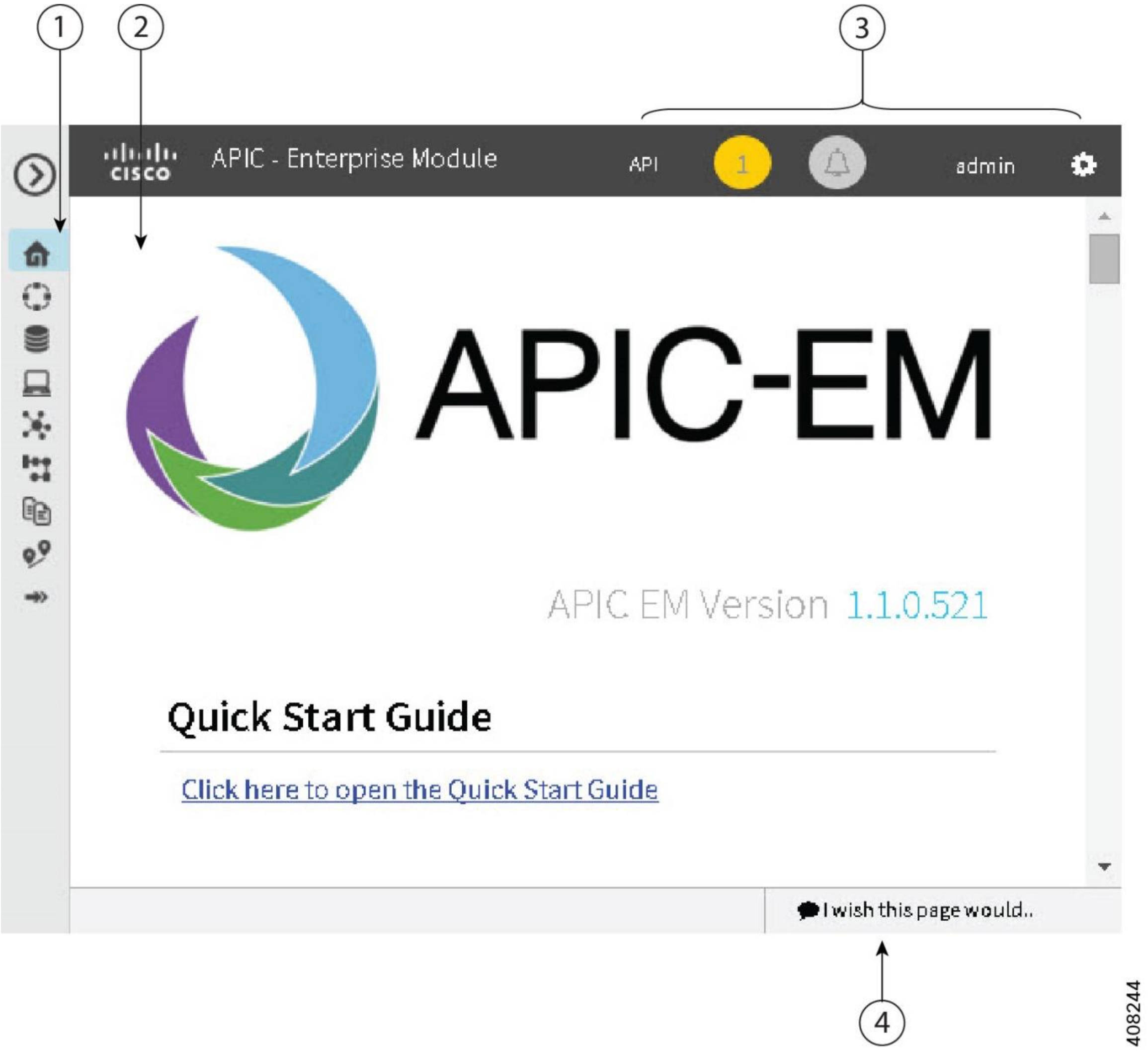
This tool predicts what happens in the data plane of the various devices in the network.
This tool based on another tool called “**APIC-EM Discovery app** “ which first discover your network topology and send all discovered information to your APIC-EM controller.
Then you use Path Trace app by typing for instance source and destination address of a packet.
The Path Trace app examines information pulled by APIC-EM from the devices in the network—the MAC tables, IP routing tables, and other forwarding details in the devices—to analyze where this imaginary packet would flow if sent in the network right now
Finally the Path Trace GUI will display the path with notes and overlaid on map of your network.

APIC-EM Path Trace ACL Analysis Tool app

You might notice that Path Trace app will not put any ACL in his consideration when examine the info and analyze the path…..it just use IP & MAC address.
But when we sue ACL Analysis Tool app with Path Trace app we will let Path Trace not to ignore the fact that maybe there is a ACL used could change the path.

The ACL Analysis app will examines the chosen path as determined by the Path Trace tool (hence the dependency), but it looks for any enabled ACLs.
The ACL Analysis tool analyzes and then characterizes (with notes overlaid on the screen) what packets sent from source to destination would be filtered as it travelled along that path.

Cisco APIC-EM GUI Overview



Callout Number	Name	Description
1	Navigation pane	Provides access to the Cisco APIC-EM features and additional applications, such as IWAN and Network Plug and Play.
2	Window	Area where the feature or application interface is displayed. When you click an option in the Navigation pane, its corresponding window opens.
3	Global toolbar	Area that provides access to tools, such as API documentation, settings, and notifications. For a full explanation of the icons on the Global toolbar, see the Global Toolbar Options table below.
4	Feedback link	Link to a form where you can provide input about your experience using the Cisco APIC-EM features and its GUI and provide suggestions for improvements.

APIC-EM Resources:

APIC-EM Demo: Overview and Path Trace App

https://www.youtube.com/watch?v=DjHa1BoYx70

Cisco Application Policy Infrastructure Controller Enterprise Module Configuration Guide, Release 1.1.x

http://www.cisco.com/c/en/us/td/docs/cloud-systems-management/application-policy-infrastructure-controller-enterprise-module/1-1-x/config-guide/b_apic-em_config_guide_v_1-1-x.html

Fundamentals of Cisco APIC-EM

http://www.cisco.com/c/m/en_us/training-events/events-webinars/apic-em.html

Demo: IWAN Application for APIC-EM

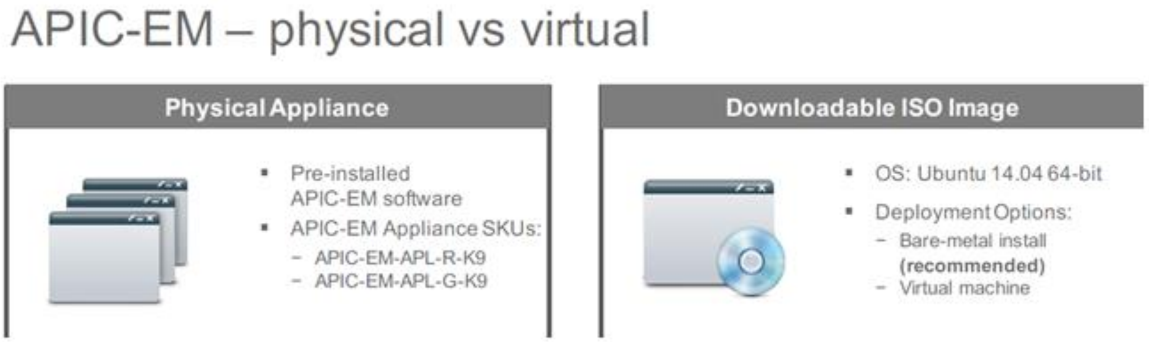
http://www.cisco.com/c/en/us/products/cloud-systems-management/application-policy-infrastructure-controller-enterprise-module/iwan-application-for-apic-em.html

Download APIC EM SDN Controller from the following URL

you will need to install the file as browser extension then reload the page choose where in your computer you want to download the files then click download:

https://developer.cisco.com/site/apic-em/

Remember APIC-EM comes as Physical appliance or virtual image



APIC-EM – physical vs virtual

Physical Appliance

- Pre-installed APIC-EM software
- APIC-EM Appliance SKUs:
  - APIC-EM-APL-R-K9
  - APIC-EM-APL-G-K9

Downloadable ISO Image

- OS: Ubuntu 14.04 64-bit
- Deployment Options:
  - Bare-metal install **(recommended)**
  - Virtual machine

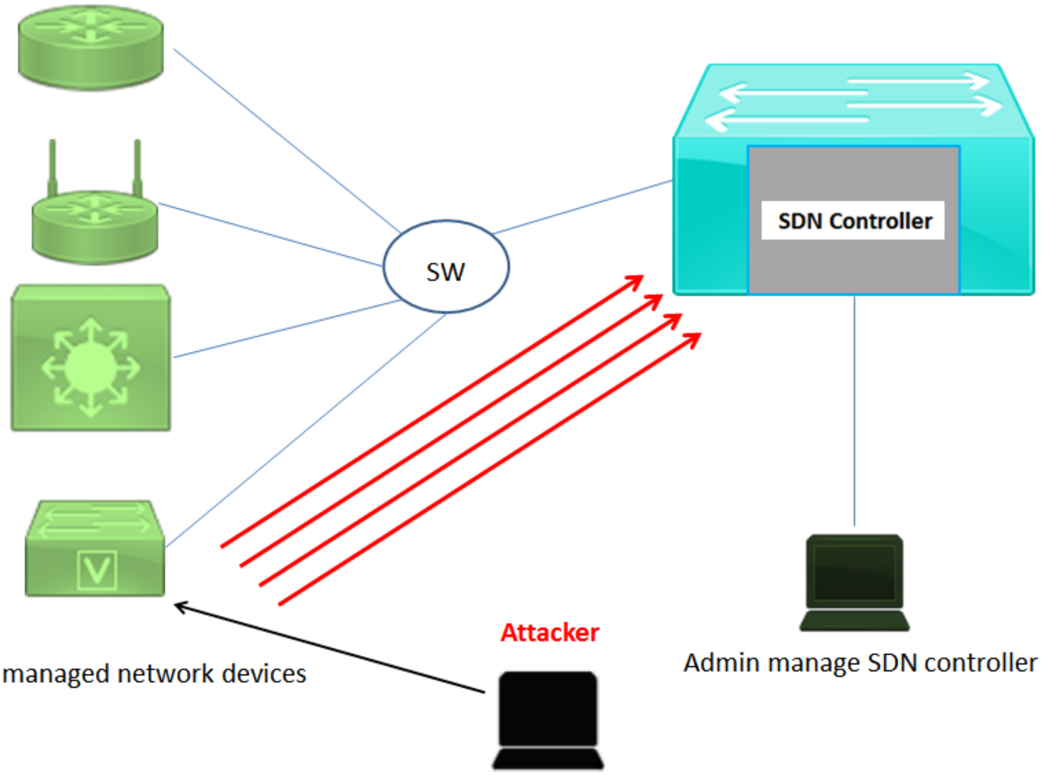
SDN Security Threats & Countermeasures

Now if we want to talk about SDN security , we will need first to imagine some attacks scenarios (SDN is not New Technology but still not used in wide scale , once it is used in wide scale for sure we will see many other scenarios and maybe new attacks terms will added to IT Security dictionary)

I will try to explain four scenarios here as example for what threats could target our SDN environment

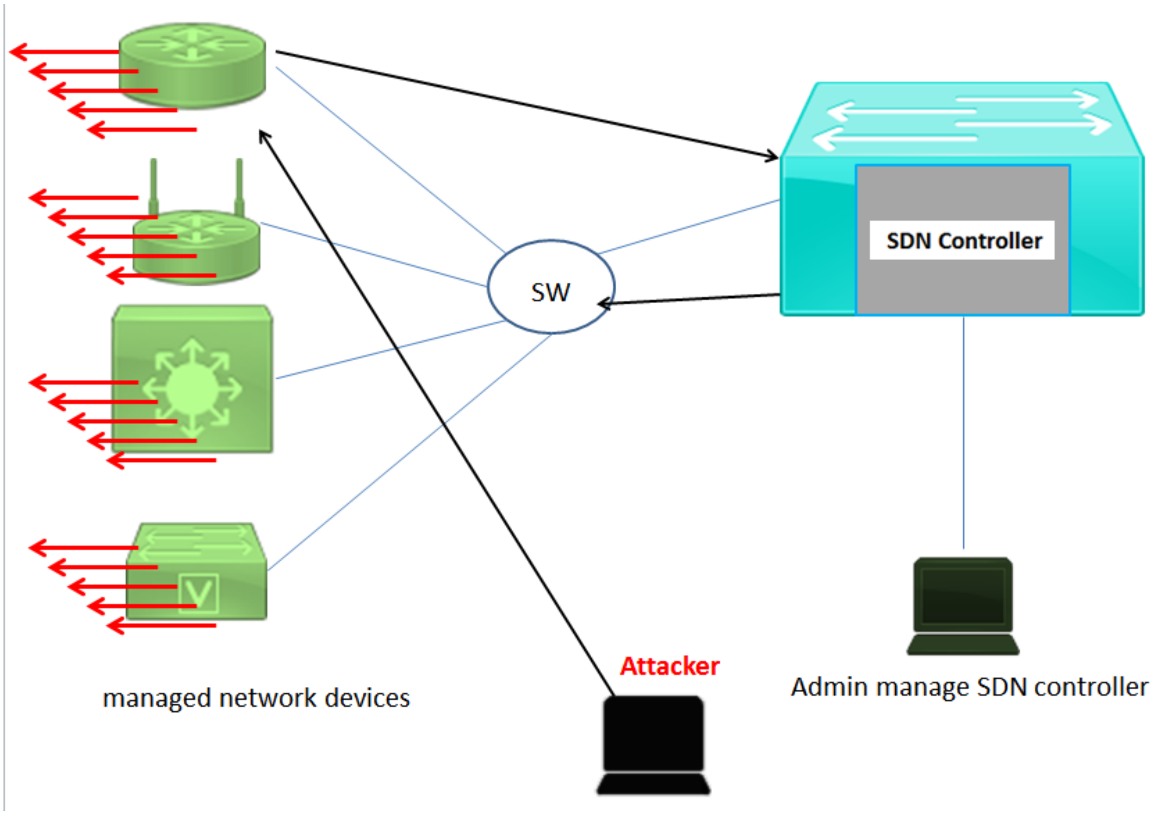
Attack Scenario 1 DoS

- target one device -agent- only and inject **Many** false network flow requests using that device data plane
- controller receive the request and process it
- controller CPU utilization goes high

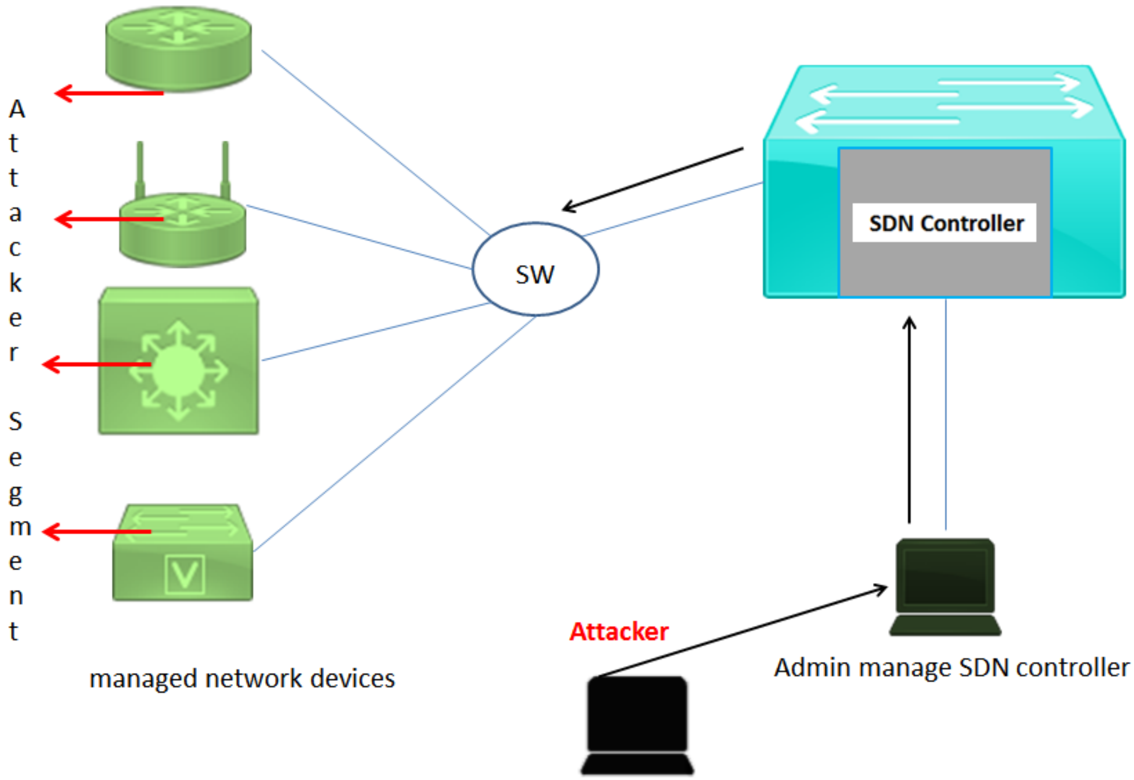


Attack Scenario 2 DoS

- target one device -agent- only and inject false network flow requests using that device data plane
- controller receive the request and process it
- invalid network flows pushed by controller control plane to network devices
- invalid flows installed on all data planes of other devices -agents- and network wide DoS



Attack Scenario 3
Attacker hack the admin PC used to admin SDN Controller , through the NBI he go inside the SDN Controller and let it ask all managed device to use specific path to route traffic which maybe just a null network or even attacker network segment.



Attack Scenario 4
In this scenario lets imagine attacker can exploit any vulnerability found in SDN controller software, SBI/NBI protocols or API's

Countermeasures
To protect our self we should think first about SDN solution components, it is the SDN controllers , the Managed devices , applications & APIs .
Also lets not forget securing access and communications between all of them, in another words we should think about how to secure data when it is in transit, in rest and in process.

I will go through each one of them and mentions the security features and techniques that help you to secure each part of them.

SDN Controller hardening

Secure access: SSH/HTTPS , keep system update and patch , disable unnecessary protocols or services ports . AAA with RBAC , enable host based firewall

Secure operation : keep device OS up-to-date , centralize log collection and monitoring , configuration management)

Managed devices (agents) hardening

Agents are routers, switches, firewalls receive directions from controllers

Secure control plane: CoPP , FHRP sec , CPPr , ICMP redirects , ICMP unreachable , Proxy ARP , securing routing protocols using authentication and route filter

Secure management plane : ssh,scp,https,snmpv3,acl , AAA , MPP , protect console VTY , disable service s, no tftp initial config

Secure data plane: DAI , ip source guard , port security , uRPF , infra ACL , Anti spoofing ACL , disable IP source routing , PVLAN

Secure network services : use QoS , disable unused ports , firewall protection

Secure applications/APIs:

secure coding practice , digital signing of code , code integrity checks , secure development lifecycle , threat modeling , understand and prioritizing risk , preform threat , mitigation test include performance negative test , preform static code analysis such as buffer overflow & resources leaks & null pointers

Secure communications channels between controller & agents:

AAA , logging , audit , SSL , IPsec , TLS, MACsec ,PKI

Management /Provisioning should be used:

RBAC ,Encryption ,Logging ,HA and redundancy ,Change management ,Firewall

SDN common Terms

SDN Software Defined Networking

- Used for automate & control
- Its centralized command & control in the network
- it separates the control plan and data plan

NFV Network Functions Virtualization

Means network devices comes in virtualized version not physical

SDS Software Defined Storage

Same as SDN but for automate & control storage

SDDC Software Defined Data Center

Tie together SDN/NFV/SDS in one data center solution

Q&A

During my writing to this document I asked my Facebook followers to ask questions about SDN and I promised to answer it here, this will make this document more interactive and contain all possible information you concern about.

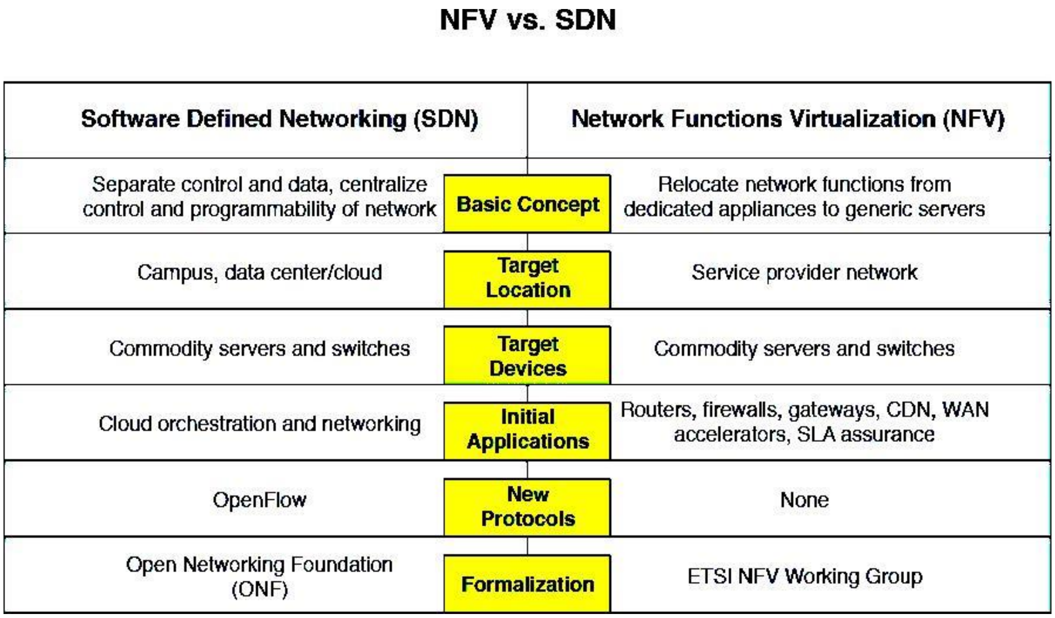
I see NFV keyword always shown beside SDN so What is NFV?

Network Function Virtualization is any Router, Switch, Antivirus , Firewall , IPS …etc. Which run as Virtual Machine VM (Not Physical Machine)

NFV controller can create new switch or routers and will generate them for you as VM's such as vSwitch or vFirewall...etc.
You can controller them using NFV controller or even by SDN controller

Many vendors now provide virtual version of their products, that's why we need NFV.
Nexus 1000v instead of catalyst switch
Virtual f5 load balancer instead of physical appliance
Juniper vRouter
Symantec antivirus
McAfee IPS
Cisco NGIPSv FirePOWER

- NFV controller tell hypervisor manager to create switch or router , then VM's application such as vSphere web client will let us connect to them .
- NVF controller can run Virtual Firewall for us too , or we can create Virtual load balancer/IDS/IPS
- Also we can create antivirus so we will have antivirus appliance VM which can scan all other VM's AV controller aka antivirus controller (in this case its NFV controller itself) will push antivirus definitions updates to the appliance



NFV vs. SDN

Software Defined Networking (SDN)		Network Functions Virtualization (NFV)
Separate control and data, centralize control and programmability of network	**Basic Concept**	Relocate network functions from dedicated appliances to generic servers
Campus, data center/cloud	**Target Location**	Service provider network
Commodity servers and switches	**Target Devices**	Commodity servers and switches
Cloud orchestration and networking	**Initial Applications**	Routers, firewalls, gateways, CDN, WAN accelerators, SLA assurance
OpenFlow	**New Protocols**	None
Open Networking Foundation (ONF)	**Formalization**	ETSI NFV Working Group

When I read about SDN I keep seeing the term "DevOps", what is DevOps?

Actually SDN is one of many reasons lead us to start using this term .
DevOps means developers will work with operations , so both teams share work with new apps then developers team write the code and operations deploy the code
Since Now day's developers and operations should act in same department without any barriers

So it is emphasizes the collaboration and communication of both software developers and other information-technology (IT) professionals

[**Dev**] means developers, all people involved in developing the product
[**Ops**] means system engineer, system admins , operation staff , DBAs , sec professional , net engineer
[**Dev**] means makers
[**Ops**] means people that deal with the creation after it birth

Example:
Developers create SDN solution or just simple Networking application
Operations represented by Network administrators will implement and use this application
Feedback here in both direction and without delay since both working together in same department or Business Unit .

Are there any SDN courses and certifications from Cisco?
Yes, five different Specialist courses with 7 exams :

Cisco Business Application Engineer Specialist
Cisco Network Programmability Developer Specialist
Cisco Network Programmability Design and Implementation Specialist
Cisco Network Programmability Design Specialist
Cisco Network Programmability Engineer Specialist

http://www.cisco.com/c/en/us/training-events/training-certifications/certifications/specialist/network-programmability.html

https://learningnetwork.cisco.com/community/certifications/network-programmability

What is the most related programming language should one know about SDN?
According to Cisco recommended you need to learn one of these programming languages: Python, C or Java

And I can't tell which one is better to study since each one has its own pros & cons , for instance Java applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of computer architecture.
While C language is one of the most powerful programming languages but very hard to learn Same time Python is easy to learn but not using Bytecode.

Let's take an example for how programming languages cannot be predictable in all times , OpenFlow interface has proven to be complex, so researchers are developing network programming languages, such as *Frenetic* and *Pyretic* that will simplify OpenFlow SDN programming, guess what, Pyretic is based on Python.

Can we consider SDN a CIIE killer?
Actually NO, you can see how Cisco nowadays is feeding their different CCIE Tracks with SDN topics.
Think about it, we just discovered new cement material is that means we will not use concrete walls and the gauge of rebar anymore? Sure NO
New cement material here is the SDN
Concrete walls and the gauge of rebar here is the Networks (CCIE tracks)

I would like to add here a comment of one of my FB friends Mr. Terry Vinson and one of the authors of Cisco Press CCIEv5 Study Guide , when he answered a similar question on my FB wall.

"I keep getting asked, "is SDN the CCIE killer?!?" My answer is always the same. This abstracted layer of functionality (Compute, Network & Storage) resides on actual devices. Physical devices that need to be engineered, upgraded, optimized and maintained. OSPF is not going away. BGP is not going to be rendered moot by SDN/NV. In fact if you pop the hood on ACI/9K you will find that every spine is a MP-iBGP Route Reflector where no leaf is more than two hops away from another (in the typical scenario). So NO I do not think SDN is a CCIE killer. It will simply be another tool we have in our tool chest to build the most dynamic, scalable and elastic networks in the industry today. "

Does u see that learning some software programming is a must to be SDN Specialist later on? As some vendors started to have SDN products managed via GUI .
First of all SDN products now days managed via GUI , my answer will be Yes and NO
No if you just want to be an ordinary network engineer, you will master the SDN controller , how using controller GUI and that is it.
Yes if you want to be special and more expert since A new job type is being created by the SDN transition: the network programmer. This is a person who will need to have a wide and deep knowledge of network engineering, as well as a deep knowledge of at least one powerful C-like programming language (C, C++, C#, Java, Objective-C). This role will be responsible for the actual programming of SDN controllers (the interface) and related components.

If you looking for my strong advice, go and learn Programming whatever it is Java, C or python it will help you even to understand other Cisco products for instance when i am teaching Cisco FirePOWER and when it comes to Correlation policy the logic behind how it is working is completely based on IF/THEN programming statement ,Which is the most basic of all the control flow statements.

if you are studying Cisco ISE which is based on policies apply AAA,802.1X ...etc. concepts and can tie all these with directory services such as Microsoft AD or any external LDAP database where we can classify users and hosts (same happens with FirePOWER)
If you play with ISE GUI you will find that the programming IF/THEN aspect shown clearly in ISE GUI where we have Conditions represent the [IF] and could be simple or complex (more than one Condition tied with AND/OR Boolean operations) and when these Conditions happens we specify a Results represent the [THEN] and send these results to routers , switches , wireless LAN controllers or firewalls to implement enforcement using APIs such as REST (HTTPS) API.

SDN is working in same concept but instead of working with identity management such as ISE , SDN work in Routing management and IGP/EGP/QoS control policies and more by using centralizing model and separates the control plan and data plan.

Where we use the programming language in controller or in OpenFlow switch ?
Mostly will be in SDN controllers and when developing controllers itself.
For OpenFlow you will just need to enable in Cisco device (if this device support OpenFlow) with just few simple commands.
Here is a link about how to Configuring OpenFlow in Cisco Nexus 5500 Series NX-OS 7.0
http://www.cisco.com/c/en/us/td/docs/switches/datacenter/nexus5500/sw/system_management/7x/b_5500_System_Mgmt_Config_7x/b_5500_System_Mgmt_Config_7x_chapter_010100.html

Can we use BGP as SBI protocol?
yes we can but when it comes to use SDN solution in DC it will not be preferred by DC experts , they consider it WAN protocol and using it inside DC will not be a good idea , same time some vendors like Juniper trying to push the market to use BGP as SBI protocol but this looks going to nowhere.

Five SDN protocols other than OpenFlow
http://searchsdn.techtarget.com/news/2240227714/Five-SDN-protocols-other-than-OpenFlow
Border Gateway Protocol as a hybrid SDN protocol
http://searchsdn.techtarget.com/feature/Border-Gateway-Protocol-as-a-hybrid-SDN-protocol

What is the main component of OpenFlow?
OpenFlow is a set of protocols and an API.
The OpenFlow protocols are currently divided in two parts:

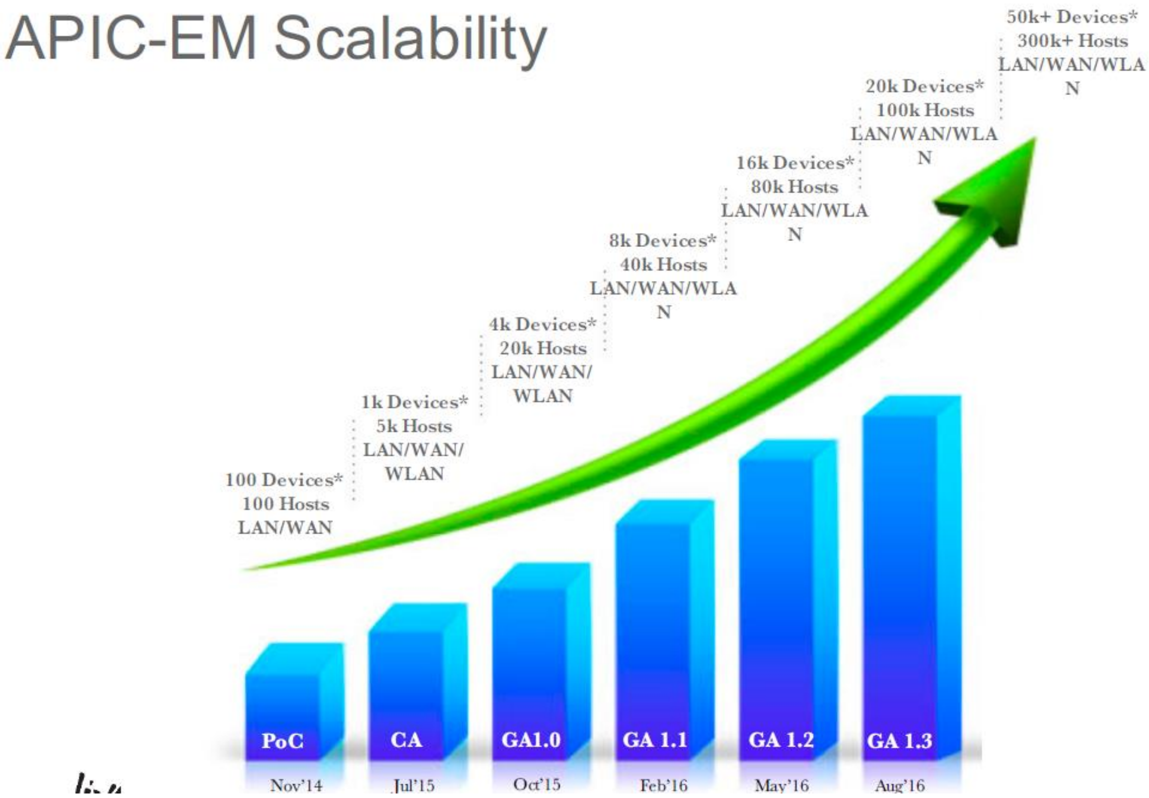
- A wire protocol (currently version 1.3.x) for establishing a control session, defining a message structure for exchanging flow modifications (flowmods) and collecting statistics, and defining the fundamental structure of a switch (ports and tables).
- A configuration and management protocol, of-config (currently version 1.1) based on NETCONF (using Yang data models) to allocate physical switch ports to a particular controller, define high availability (active/standby) and behaviors on controller connection failure.

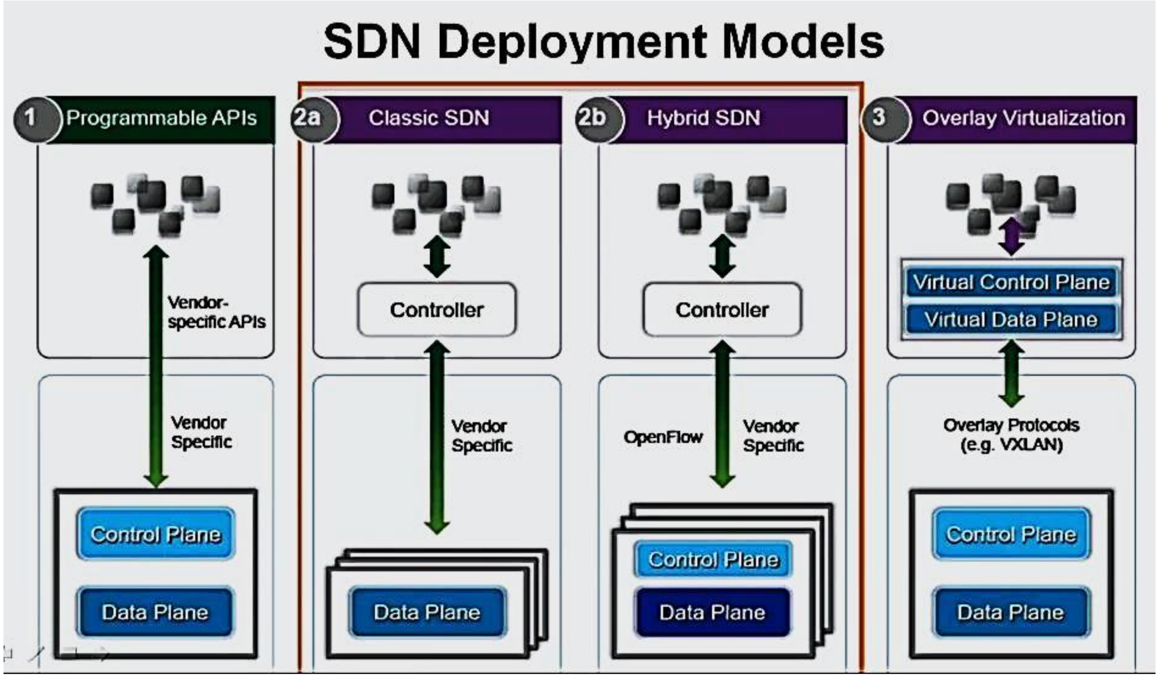
Why, when and at which scale must I go for SDN?
I guess if we go through this document we will see how it is important to centralize the control plane with fully automation support and this one of the biggest reasons for why we go to SDN. Since traditional networks such as IP,MPLS networks are based on distributed control model not a centralized control model like SDN do.

Imagine using SDN controllers inside MPLS SP core networks and how this will decrease the number of control planes and reduce the number of interaction of routing protocols required to create forwarding states

During reading this document you should figure out that SDN has less scale limitation, there are many SDN solutions for Data Centers , for Traditional R&S networks and so on.
Yes we have some limitation with which network devices support OpenFlow but it looks like all vendors now days trying to support it on their new network devices models.

We should also keep in mind that scale limitation is decreased dynamically in SDN world , for instance following diagram shows how scalability limitation is decreased rapidly and in monthly basis with APIC-EM





What SDN and OpenFlow software switches available to use & practice with?

There are currently several OpenFlow software switches available that can be used such as :

- Open vSwitch
- Indigo
- Pantou (OpenWRT)

You can find many Open source projects online:

- For Switches: Open vSwitch, Pantou, Indigo, LINC, XORPlus, OF13SoftSwitch
- For Controllers: Beacon, Floodlight, Maestro, Trema, FlowER, Ryu
- Other Tools: FlowVisor, Avior, RouteFlow, OFlops and Cbench, OSCARS,Twister, FortNOX

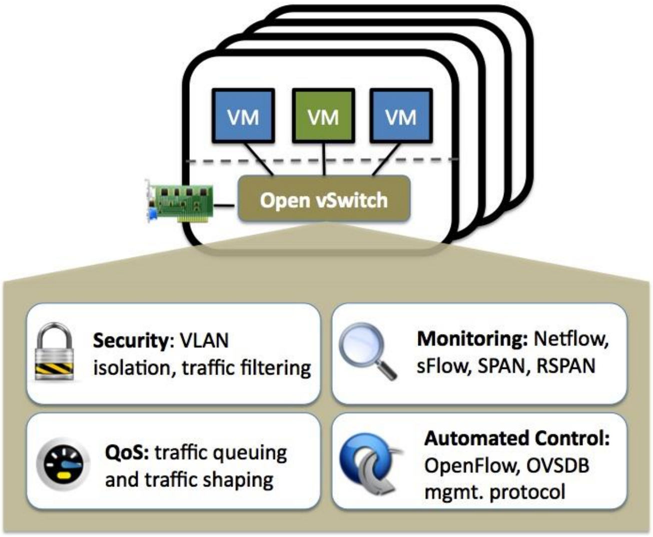
Open vSwitch http://openvswitch.org/

- Will bridge traffic between virtual machines (VMs).
- Open vSwitch is targeted at multi-server virtualization deployments.
- Open vSwitch is a multilayer virtual switch licensed under the Apache license.
- Open vSwitch can operate as both, a soft switch running within the hypervisor, and as the control stack for switching silicon.
- It is the default switch in XenServer 6.0, the Xen Cloud Platform and also supports Xen, KVM, Proxmox VE, and VirtualBox.
- Open vSwitch is included as a part of the Linux 3.3 kernel and packaging for the user space utilities are available on most popular distributions.

We have problem solved by Open vSwitch

Distributed virtual switches (for example, VMware, vDS, and Cisco's Nexus 1000V) use tags to uniquely identify a VM, or to hold some other context that is only relevant in the logical domain.

Much of the problem of building a distributed virtual switch is to efficiently and correctly manage these tags. Open vSwitch includes multiple mechanisms for specifying and maintaining tagging rules, all of which are accessible to a remote process for orchestration.


VM
VM
VM
Open vSwitch
Security: VLAN isolation, traffic filtering
Monitoring: Netflow, sFlow, SPAN, RSPAN
QoS: traffic queuing and traffic shaping
Automated Control: OpenFlow, OVSDB mgmt. protocol

Pantou(OpenWrt)

- Pantou turns a commercial wireless router/access point to an OpenFlow-enabled switch.
- OpenFlow is implemented as an application on top of OpenWrt.
- OpenWrt is an operating system primarily used on embedded devices to route network traffic.

Indigo

- Indigo is an open source OpenFlow implementation that runs on physical switches and uses the hardware features of application specific integrated circuits (ASICs) of Ethernet switches to run OpenFlow at line rates.
- It is based on the OpenFlow Reference Implementation from Stanford and currently implements all required features of the OpenFlow 1.0 standard.

For Controllers

we had POX, and OpenDaylight and OpenFlow controllers.
we also have some others like:

- -Beacon Java-based controller
- -Floodlight Open SDN Controller is an enterprise-class, Apache-licensed, Java-based OpenFlow Controller.

Can I simulate a network with SDN controllers and SDN enabled network devices (agents)?

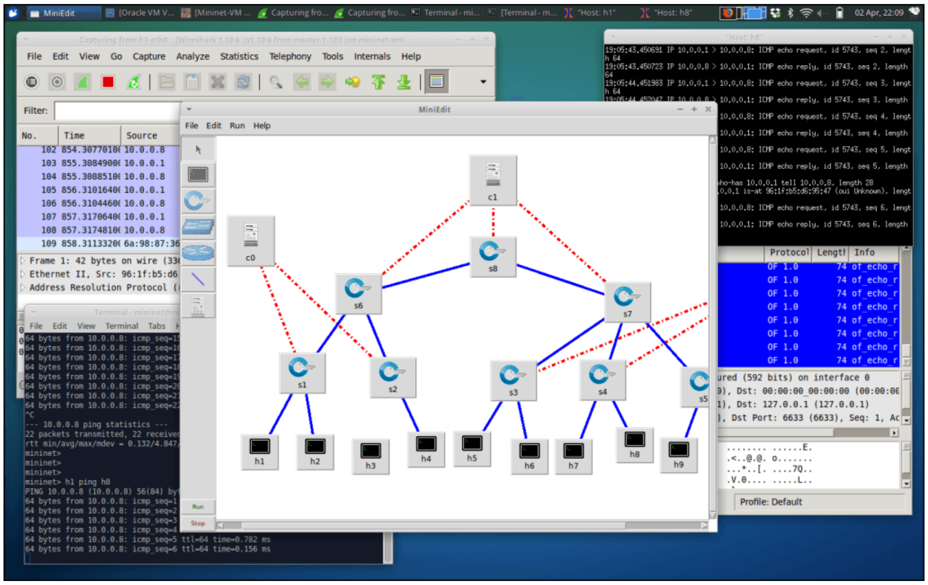
Yes you can by using tools such as Mininet.

•Mininet is a software tool, which allows an entire OpenFlow network to be emulated on a single computer.

• Mininet creates a network of virtual hosts, switches, controllers, and links.

• Mininet hosts run standard Linux network software, and its switches support OpenFlow.

•It is easy to install and is available as a pre-packaged Linux virtual machine (VM) image that runs on VMware or VirtualBox , download for free from : http://mininet.org/



```
No.   Time          Source
  102 854.3077010( 10.0.0.8
  103 855.3084900( 10.0.0.1
  104 855.3088510( 10.0.0.8
  105 856.3101640( 10.0.0.1
  106 856.3104460( 10.0.0.8
  107 857.3170640( 10.0.0.1
  108 857.3174810( 10.0.0.8
  109 858.3113320( 6a:98:87:36
▷ Frame 1: 42 bytes on wire (33(
▷ Ethernet II, Src: 96:1f:b5:d6
▷ Address Resolution Protocol (
```

```
64 bytes from 10.0.0.8: icmp_seq=15
64 bytes from 10.0.0.8: icmp_seq=16
64 bytes from 10.0.0.8: icmp_seq=17
64 bytes from 10.0.0.8: icmp_seq=18
64 bytes from 10.0.0.8: icmp_seq=19
64 bytes from 10.0.0.8: icmp_seq=20
64 bytes from 10.0.0.8: icmp_seq=21
64 bytes from 10.0.0.8: icmp_seq=22
^C
--- 10.0.0.8 ping statistics ---
22 packets transmitted, 22 receive
rtt min/avg/max/mdev = 0.132/4.847/
mininet>
mininet>
mininet>
mininet> h1 ping h8
PING 10.0.0.8 (10.0.0.8) 56(84) by
64 bytes from 10.0.0.8: icmp_seq=1
64 bytes from 10.0.0.8: icmp_seq=2
64 bytes from 10.0.0.8: icmp_seq=3
64 bytes from 10.0.0.8: icmp_seq=4
64 bytes from 10.0.0.8: icmp_seq=5 ttl=64 time=0.782 ms
64 bytes from 10.0.0.8: icmp_seq=6 ttl=64 time=0.156 ms
```

```
19:05:43.450691 IP 10.0.0.1 > 10.0.0.8: ICMP echo request, id 5743, seq 2, lengt
h 64
19:05:43.450723 IP 10.0.0.8 > 10.0.0.1: ICMP echo reply, id 5743, seq 2, length
64
19:05:44.451983 IP 10.0.0.1 > 10.0.0.8: ICMP echo request, id 5743, seq 3, lengt
h 64
19:05:44.452047 IP 10.0.0.8 > 10.0.0.1: ICMP echo reply, id 5743, seq 3, length
                              10.0.0.8: ICMP echo request, id 5743, seq 4, lengt
                              10.0.0.1: ICMP echo reply, id 5743, seq 4, length
                              10.0.0.8: ICMP echo request, id 5743, seq 5, lengt
                              10.0.0.1: ICMP echo reply, id 5743, seq 5, length
                             who-has 10.0.0.1 tell 10.0.0.8, length 28
                             .0.0.1 is-at 96:1f:b5:d6:95:47 (oui Unknown), lengt
                              10.0.0.8: ICMP echo request, id 5743, seq 6, lengt
                              10.0.0.1: ICMP echo reply, id 5743, seq 6, length
```

```
Protocol Length Info
OF 1.0      74 of_echo_r
OF 1.0      74 of_echo_r
OF 1.0      74 of_echo_r
OF 1.0      74 of_echo_r
OF 1.0      74 of_echo_r
OF 1.0      74 of_echo_r
OF 1.0      74 of_echo_r
OF 1.0      74 of_echo_r
ured (592 bits) on interface 0
0), Dst: 00:00:00_00:00:00 (00:00:00
1), Dst: 127.0.0.1 (127.0.0.1)
), Dst Port: 6633 (6633), Seq: 1, Ac
........ ......E.
.<..@.@. o.......
...*..[. ....7Q..
.V.0.... .....L..
```

Mininet GUI

What is the difference between Automation and Orchestration?
Automation can be defined as the use of software to reduce the operational workload of configuration, troubleshooting and asset management. Automation is often seen in the form of Python or Perl scripts that perform a specific or narrow set of functions

Orchestration can be defined as the use of the automation to provide services through the use of applications that drive the network.

As an example, orchestration is application that can take an request from a customer via web portal for new virtual server requiring provisioning. This ideal app will analyses the network configuration and implement the configuration change for the customer and then update the billing system. The network itself might implement in the physical network, in a virtual overlay on hypervisors, across the WAN via encrypted tunnel or one of many other options. The connectivity is far less important than the orchestrated service establishment across many devices and platforms.

What is the difference between network orchestration (OpenStack) and SDN control (OpenDaylight)?

OpenStack can be considered to be not one project but several, consisting of compute, storage and networking. All three platform components are managed by a dashboard Web application. Combined, they can provide a complete cloud network operating system.

Conversely, OpenDaylight is an SDN controller but with a number of northbound APIs that allow interaction with network application and orchestration services, such as OpenStack Neutron, and southbound APIs, such as OpenFlow, NETCONF and BGP. One goal of the project is to extend the services available and provide a de facto set of service APIs.

The projects are compatible, and OpenDaylight can be integrated with OpenStack using an OpenStack via Neutron plugin. This moves the complexity up the stack from OpenStack to OpenDaylight, completing the SDN picture.

Who are the current SDN vendors in the market?

SDN Controllers Vendors:
- Big Switch Networks
- Cisco
- HP
- IBM
- NEC
- Plexxi

NFV Solutions Vendors:
- juniper
- VMware
- CohesiveFT

Virtual Network Device Providers:
- Cisco
- Brocade

- Citrix
- F5
- HP
- Microsoft
- Radware
- VMware

OpenFlow enabled hardware providers:

- Arista
- Brocade
- Centec
- Cisco
- Dell
- Extreme Networks
- HP
- Juniper
- Lenovo ibm
- Noviflow
- Pica8
- Vello systems

Network Management Software Providers (interfaces to manage SDN):

- Anuta
- Dell
- Cyan
- Huawei
- Openstack neutron
- Tail-f systems

This list dynamically increased weekly , keep looking in some URL's like:
https://www.sdxcentral.com/sdn/definitions/sdn-controllers/sdn-controllers-comprehensive-list/

https://www.opennetworking.org/

Beyond Network Infra Devices vendors such as Cisco, can we see SDN in Systems vendors such as Microsoft?
Yes , for sure , Microsoft just Turned their new Windows server 2016 to SDDC solution with many features supporting the concept of Software defined datacenter , for instance they just add server role called "Network Controller" which provides a centralized, programmable point of automation to manage, configure, monitor, and troubleshoot both virtual and physical network infrastructure in your datacenter.
Network Controller is a highly available and scalable server role with SBI & NBI API (for NBI they use REST API).
you can use Network Controller to manage the following physical and virtual network infrastructure:

- Hyper-V VMs and virtual switches
- Physical network switches
- Physical network routers
- Firewall software
- VPN Gateways, including Routing and Remote Access Service (RRAS) Multitenant Gateways
- Load Balancers

Other SDN features provided by windows 2016 are :

- Hyper-V Network Virtualization
- Hyper-V Virtual Switch
- RRAS Multitenant Gateway
- NIC Teaming also known as load balancing and failover (LBFO)

More details:
https://technet.microsoft.com/en-us/library/dn859240(v=ws.11).aspx

Resources:

Cisco Application Centric Infrastructure Fundamentals
http://www.cisco.com/c/en/us/td/docs/switches/datacenter/aci/apic/sw/1-x/aci-fundamentals/b_ACI-Fundamentals.html

Cisco APIC Enterprise Module (APIC-EM)
https://developer.cisco.com/site/apic-em/

SDN Overview
http://www.cisco.com/c/en/us/solutions/software-defined-networking/overview.html

SDN Hub | Software-defined Networking forum
http://sdnhub.org/

OpenFlow Tutorial
http://archive.openflow.org/wk/index.php/OpenFlow_Tutorial

Books:

Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud
https://www.amazon.com/Foundations-Modern-Networking-SDN-Cloud/dp/0134175395

SDN and NFV Simplified: A Visual Guide to Understanding Software Defined Networks and Network Function Virtualization
https://www.amazon.com/SDN-NFV-Simplified-Understanding-Virtualization/dp/0134306406

MPLS in the SDN Era
https://www.amazon.com/MPLS-SDN-Era-Interoperable-Scenarios/dp/149190545X

Navigating Network Complexity: Next-generation Routing with SDN, Service Virtualization, and Service Chaining
https://www.amazon.com/Navigating-Network-Complexity-Next-generation-virtualization/dp/0133989356

SDN: Software Defined Networks
https://www.amazon.com/SDN-Software-Networks-Thomas-Nadeau/dp/1449342302

CCNA Cloud CLDFND 210-451 Official Cert Guide
http://www.ciscopress.com/store/ccna-cloud-cldfnd-210-451-official-cert-guide-9781587147005

Software Defined Networking with OpenFlow
https://www.packtpub.com/networking-and-servers/software-defined-networking-openflow

OpenStack Networking Cookbook
https://www.packtpub.com/virtualization-and-cloud/openstack-networking-cookbook

OpenFlow Cookbook
https://www.amazon.com/OpenFlow-Cookbook-Kingston-Smiler-S/dp/1783987944/ref=sr_1_1?s=books&ie=UTF8&qid=1468503276&sr=1-1&keywords=OpenFlow

Network Programmability and Automation
http://shop.oreilly.com/product/0636920042082.do

Programming and Automating Cisco Networks: A guide to network programmability and automation in the data center, campus, and WAN (will be released in August 2016)
http://www.ciscopress.com/store/programming-and-automating-cisco-networks-a-guide-to-9781587144653

Videos:
Security for Software Defined Networks—Networking Talks
http://www.pearsonitcertification.com/store/security-for-software-defined-networks-networking-talks-9780789753519

An Introduction to Software Defined Networking (SDN) LiveLessons—Networking Talks
http://www.informit.com/store/introduction-to-software-defined-networking-sdn-livelessons-9780789753816

Software Defined Networking (SDN): The Big Picture
https://www.pluralsight.com/courses/sdn-big-picture

SDN Fundamentals (Free video)
https://www.youtube.com/watch?v=Np4p1CDIuzc

SDN and OpenFlow Overview - Open, API and Overlay based SDN (Free video)
https://www.youtube.com/watch?v=l-DcbQhFAQs

Stanford Seminar - Software-Defined Networking at the Crossroads (Free video)
https://www.youtube.com/watch?v=WabdXYzCAOU

Good Luck
CCIE & CCSI: Yasser Auda
https://www.facebook.com/YasserRamzyAuda
https://learningnetwork.cisco.com/people/yasserramzy
https://www.youtube.com/user/yasserramzyauda