UE PRIP

Principes des réseaux informatiques par la pratique

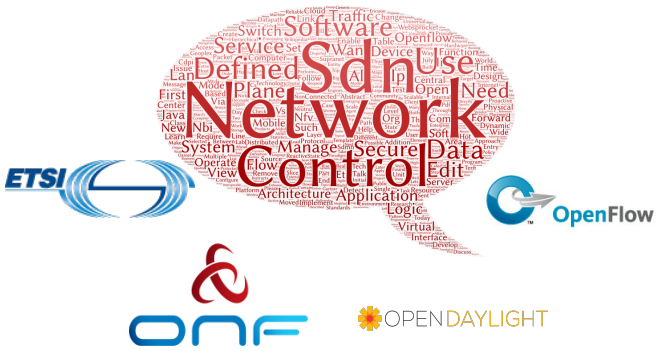
Introduction to new networking paradigms: SDN and NFV

Isabel Amigo

2022

Notes:

The topics covered today...



Notes:

Lecture's Objectives

1. Introduce future networking paradigms and concepts
2. At the end of the lecture you should be able to respond to the following questions regarding these concepts:
   - Why? What? How? Who?
   - Are SDN and folks disruptive technologies?

Notes:

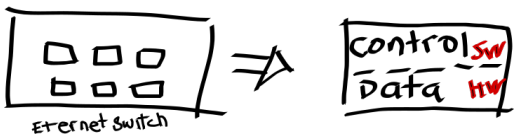
- SDN: Software-defined networking

Structure

- SDN drivers and overview
- SDN
- OpenFlow
- A word on NETCONF/YANG
- NFV
- Current State of SDN
- Conclusion

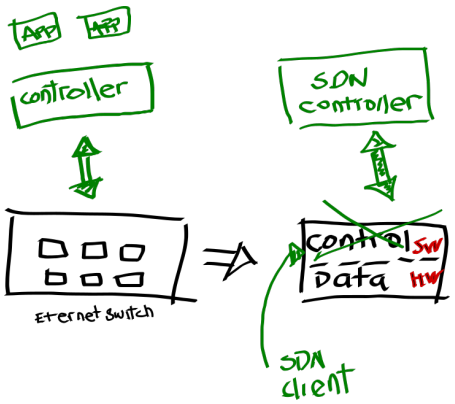
Notes:

Traditional Node



Notes:

SDN Node



Notes:

Recall

Management Plane

- Configuration
- Monitoring

Control Plane

- Establishing the state in routers
- Determines how and where packets are forwarded
- Slow time-scale (per control event)

Data Plane

- Processing and forwarding packets
- Based on state in routers and endpoints
- Per-packet timescale (fast!)

Notes:

- Management example: CLI
- Control plane example: Routing, traffic engineering, firewall state, ...
- Data plane example: IP, TCP, Ethernet, etc.

SDN DRIVERS

Why do we need new paradigms?

Notes:

Why have networks remain almost the same for years?

- Difficult to innovate
- Expensive equipments
- Closed systems

Notes:

- E.g. OSPF (IPv4 1998, IPv6 2008), BGP (proposed in 1995 RFC last updated in 2006)
- Proprietary HW and SW
- Vendor lock-in Problems
- Vendor-dependent administration

Driver 1: Datacenters (DC) and Virtualization

- The trend is to virtualize, why?
- Virtualization acceleration on DCs, with the underutilization of servers
- Multi-tenant DCs $\Rightarrow$ VMs
- Need of isolated networks and to reconfigure if VM migrates $\rightarrow$ VLANs, VXLANs

Can't we have something more agile to programmatically construct and dynamically change the logical network infrastructure?

Notes:

- Virtualization is claimed to reduce costs, energy consumption
  - Generic and programmable hardware
  - Specified Software
  - OpenSource initiatives
- Servers underutilization lead to the idea of sharing the same server along many customers, what we call the "tenants".
- Each tenant wants this to be transparent for him, which lead to the use of VMs. From the cloud provider point of view VMs also allow to have dynamic reallocation of resources.
- Of course each tenant also needs a network which is isolated from the other tenants, and that remains consistent along VMs migrations
- This led to solutions as VLAN and later as VXLANs
- This is one of SDN's use cases highly appealing for the industry

Driver 2: The need of abstractions

Abstraction $\Rightarrow$ Interfaces $\Rightarrow$ Modularity

Notes:

- To deal with complex systems we need to make them modular
- For that we need to define abstractions and interfaces
- This is the approach of other disciplines...

Imagine...

- A computer for which you can only develop programs using vendor-dependent commands?
- Having to configure every element of the system (memory, disk etc.)

We need abstractions in networks, decouple the problem, make it modular

Notes:

- Evolution in programming
  - Machine language
  - OS
  - High-level programming language

Current networking abstractions

Layers! ☺
but they only deal with the data plane ☹

Notes:

- Layers are the key of the success of the Internet, but provide abstractions only regarding the data plane

What abstractions do networks have related to control plane?

None!

Notes:

- Initially networks were much more simpler: Ethernet, IP, TCP....
- New control requirements led to great complexity
  - Isolation (e.g. VLANs, ACLs)
  - Traffic engineering (e.g. MPLS, ECMP, weights)
  - Packet processing middle-boxes (e.g. firewalls, NATs)
  - Payload analysis (e.g. DPI)
- All these mechanisms were designed and deployed independently rendering a complicated control plane design

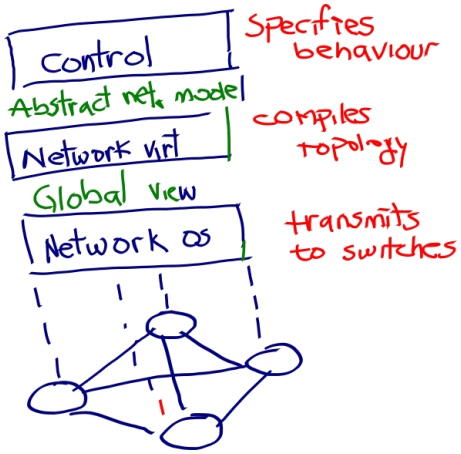
Abstractions in the genesis of SDN

1. Global network view
   - Instead of distributed protocols, configuration is Function(view)
2. Simplified network view
   - Control specifies goals, doesn't configure physical devices
3. Forwarding
   - Communication of control program with forwarding circuits (HW)

Notes:

- Abstractions proposed by Scott Shenker, CS Professor at UC Berkeley. One complete presentation: `https://www.slideshare.net/martin_casado/sdn-abstractions`
- A network is intrinsically distributed, asking a control plane to work from a distributed state, where there are no guarantees of communication renders the problem quite complex
- Abstraction 1: Shield control mechanisms from state distribution, While allowing access to this state
  - Implemented through a "Network Operating System" (NOS)
  - No longer designing distributed control protocols:
  - Design one distributed system (NOS), Use for all control functions
- Abstraction 3: OpenFlow, or other protocol

Initial SDN Architecture proposal



Notes:

- Source: Scott Shenker SDN Lectures (`http://inst.eecs.berkeley.edu/~ee122/`)

Driver 3: An existent tendency

Tendency to separate control and data planes already existed:

- MPLS edge routers computing the path
- Control and Data plane in routers already separated... but within the same chassis
- Always been like that in telephony networks

Notes:

SDN Genesis

- ...
- 2007 Ethane (Martin Casado *et al.* [2])
  - Part of Clean Slate program: *What would the Internet look like in 15 years if we restarted from a clean slate?*
  - Main contributions: Centralized controller and OpenFlow
- 2011 ONF is created, to standardize OpenFlow
- May 2011 Marvell and Larch Networks announced the availability of an OpenFlow-enabled switch
- ...

Notes:

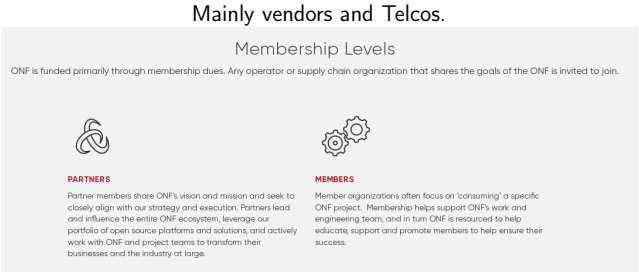
ONF: Open Network Foundation [5]

- *Open Networking Foundation (ONF) is a user-driven organization dedicated to the promotion and adoption of Software-Defined Networking (SDN) through open standards development.*
- Founding members Deutsche Telekom, Facebook, Google, Microsoft, Verizon, and Yahoo!

Notes:

ONF members

Mainly vendors and Telcos.



x9 among which: Intel, Google, AT&T,Deutsche Telekom

x110 among which Cisco, Dell, Ericsson, Nokia, Microsoft

+ several collaborators

As for March 2022

Notes:

So what is SDN?

Notes:

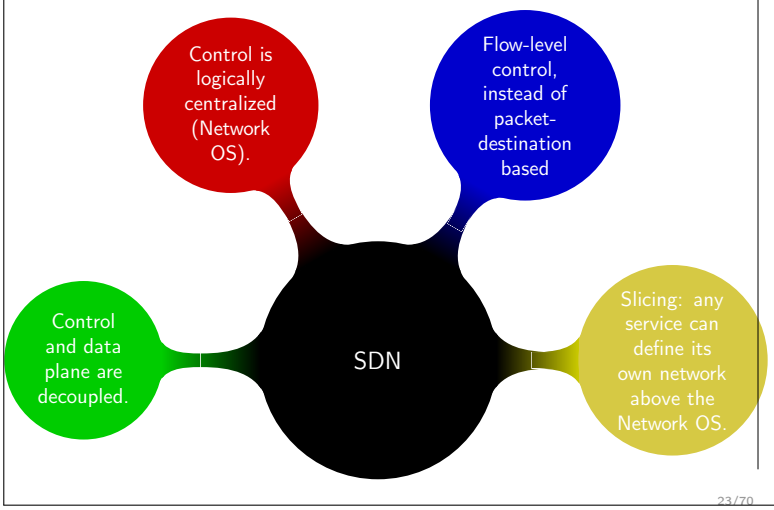
SDN principles

- Separate control and data plane
- Control plane executable in commodity HW
- Programmable data plane

Notes:

- If.. then... else in the control plane
- Agility
- Diminish human-introduced errors

The four key components of SDN



Notes:

- Separate control and data plane ⇒ define an open interface (vs private interfaces in traditional networks)
- The control plane defines forwarding rules applicable to flows, this allows to take into account the needs of different services
- The SDN controller can thus program the network infrastructure in order to allow for different services. This is similar to the role of an operating system in computers, the controller is thus also referred as « Network OS ».
- Even different services can share the same infrastructure and at the same time be isolated from each other, this concept is referred as « Network Slicing ».

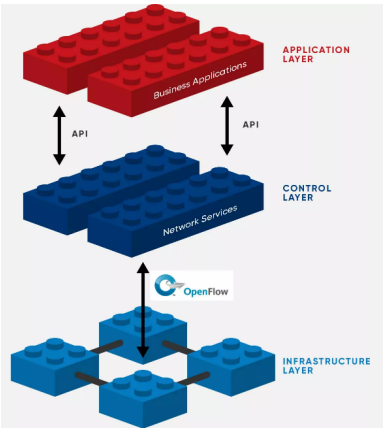
SDN Wish list

- Facilitate innovation
- Allow experiments and research without the need of expensive equipments (reduce CapEx)
- Flexibility
- Fast upgrades
- Increase speed to market
- Reduce OpEx

Notes:

- How? Abstractions, Open Interfaces, Programmability
- SW cycles faster than HW cycles
- No longer lead with different CLIs from different vendors or even different OSs and versions of same vendor

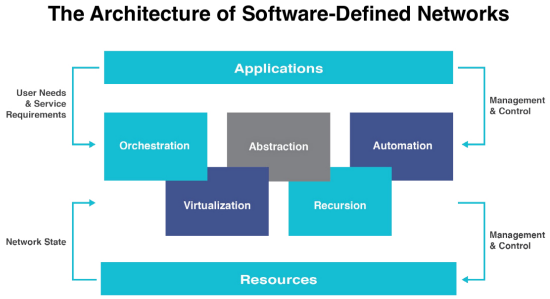
ONF Proposed Architecture



Source: https://opennetworking.org/sdn-definition/

Notes:

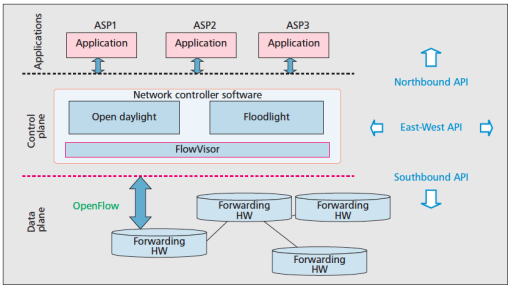
ONF Proposed Architecture



Source: SDN Architecture - A Primer, White Paper, ONF, 09/2016

Notes:

Commonly used reference architecture



Notes:

- Applications examples:
  - Network virtualisation (use network resources regardless of their localisation)
  - Orchestration (Cloud)
  - Dynamic modification of a service functioning
  - Scale out (Cloud)
  - Automation of manual tasks (Troubleshooting, provisioning)
  - Monitoring
  - Performance optimization (Traffic engineering, load balancing)
  - Multi-tenancy (controlling addresses, topology, routing , security for each client)
  - Service Integration (Load balancers, firewalls, Intrusion Detection Systems (IDS))
  - ...

Software-Defined Networking, What is it? Several attempts of definitions

Def. 1 (Wikipedia)

SDN is an approach to computer networking that allows network administrators to programmatically initialize, control, change, and manage network behavior dynamically via open interfaces and abstraction of lower-level functionality.

Notes:

Software-Defined Networking, What is it? Several attempts of definitions

Def 2 (ONF)

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

Notes:

Software-Defined Networking, What is it? Several attempts of definitions

The OpenDayLight presentation

The modern software-defined networking (SDN) movement grew out of a simple question: why shouldn't networking devices be programmable just as other computing platforms are?
The benefits of such an approach were obvious: no more arcane protocols to learn. No more waiting and hoping for networking vendors to develop specialized features you need. And if you could develop your own features, you could then optimize your device selection for price and performance independently of feature-richness.

Notes:

Standardization Bodies for SDN-related protocols

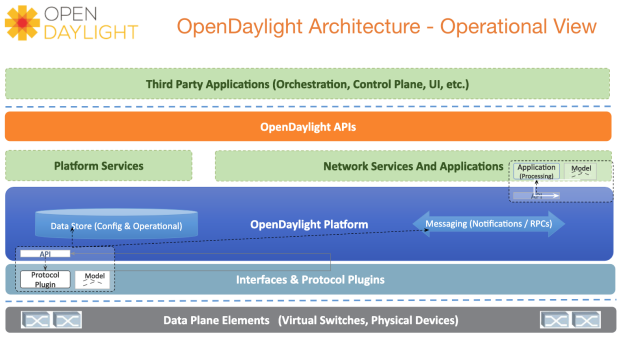
- ONF: Open Network Foundation
- IETF
- NFV/ETSI
- ...

Notes:

- IETF
  - Not the main standardization actor for SDN
  - Has workgroups such as SDN Research group, Interface to Routing systems,, Application layer traffic optimisation
- We will see NFV/ETSI within some slides

OpenDayLight

In OpenDayLight words: the most widely deployed open source SDN controller platform(in 6 years, 10 releases, 1000+ authors/submitters, 100K+ commits, powers networks of 1B+ global subscribers).



Notes:

- OpenDayLight (ODL)
  - Founded in 2013 is a collaborative open source project hosted by the Linux Foundation (now by LF Networking)
  - Arose out of the SDN movement, with a clear focus on network programmability.
  - Rather than producing standards the goal is to produce working code for SDN/NFV environments
  - Has heavy industry involvement and backing
  - ODL code is integrated or embedded in more than 35 vendor solutions and apps,

What is OpenFlow? Is it an essential building block of SDN?

Notes:

OpenFlow

- *First standard communications interface defined between the control and forwarding layers of an SDN architecture.*
- Standardised by ONF

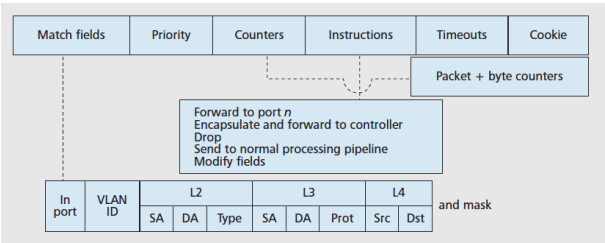
Notes:

OpenFlow?

- Instructions available for programming network devices
- Allows communication between control and data plane
- Forwarding: determined by flow tables
- Flow tables are filled-in by the controller

Notes:

OpenFlow Forwarding- One entry on one table



Notes:

- Match fields: specify conditions under which a packet is matched. As for version 1.3.4 there are around 40 parameters that can be used for matching, coming from L2,L3,L4 headers as well as input port and metadata.
- Priority: sets the precedence of the flow entry, used along with the matched field
- Counters: statistics on matching packets
- Instructions: Where actions are specified
- Timeouts: idle time before the entry will expire
- Cookie: added in version 1.3 to filter flow entries, not used in packet processing
- Flags: added in version 1.3.3
- A packet might be matched with several entries on one table, the one with most matched fields is selected

Some simplified examples

Switching

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	*	00:1f:..	*	*	*	*	*	*	*	port6

Flow Switching

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
port3	00:20..	00:1f..	0800	vlan1	1.2.3.4	5.6.7.8	4	17264	80	port6

Firewall

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	*	*	*	*	*	*	*	*	22	drop

Notes:

- Source: OpenFlowTutorial by Brandon Heller
  `https://github.com/mininet/openflow-tutorial/wiki`

Some simplified examples

Routing

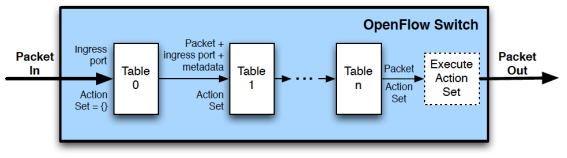
Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	*	*	*	*	*	5.6.7.8	*	*	*	port6

VLAN Switching

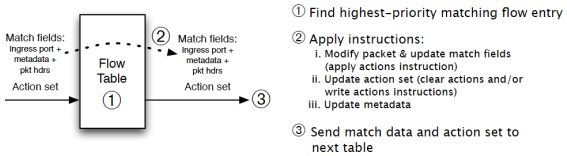
Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	*	00:1f..	*	vlan1	*	*	*	*	*	port6, port7, port9

Notes:

OpenFlow Forwarding-Pipeline



(a) Packets are matched against multiple tables in the pipeline



(b) Per-table packet processing

Notes:

- Version 1.0.0 only included one table, but this doesn't scale well
- Version 1.1.0 included multiple tables and the so-called pipeline processing mechanism
- Pipeline processing allows to create hierarchical processing options with transitions "if-then-go-to" logic
- All packets go through table 0, and can then be processed by another table, or not
- An action set is associated to the packet, at every matching either the actions indicated in the entry are executed or added to the action set. One of this actions can be "go to table x" where x is always a higher number table. Otherwise the packet is passed to the next table.

OpenFlow Actions

Output	identifies the output port (i.e.: interface) for the packet
Set-Queue	identifies a queue within the output port
Group	actions defined for that group must be applied
Push-Tag/Pop-Tag	modifies VLAN id or MPLS tags
Set-Field	modifies values in header (e.g. DSCP, dst IP,..)
Change-TTL	modifies TTL value (IPv4/MPLS) or hop limit (IPv6)
Drop	if no action provided, then the action is drop

Notes:

- Recall! current forwarding algorithm: longest-prefix match based on the packet's destination address
- As we have seen, one entry in a table flow has an instructions field. This is actually an *Instructions set* which is composed of a list of instructions. Each instruction can be accompanied with an action or not
- An Instruction can be: Write actions (write actions to action set), Apply actions (apply actions immediately), Clear actions (clear actions from action set), Goto table, write metadata (as for version 1.4.0)

Why multiple tables?

Example:

- table 0 identifies the output interface, according to dst. address
- table 1 identifies the scheduling policy according to the value of DSCP

Defining sets of actions across multiple tables provides a more modular configuration

Notes:

- What would we want multiple tables for?
  - performing independent actions based on matching different fields in a packet, which effectively requires a separate look up
  - having a two stage model, where packets are first tagged (or meta-tagged) based on some packet characteristic, and then more matching and processing occurs.
- both cases can, in theory, be handled in a single table, but the handling is generally awkward and the need to combine matching fields forces an explosion of the number flow entries

OpenFlow Protocol - Messages

- Messages Header; version (8 bytes), Type(8 bytes) Length (16 bytes)
- 30 type of messages, among which:
  - Hello
  - Packet-In
  - Packet-out
  - Set-config
- Messages can belong to one of three categories:
  - Controller-to switch
  - Asynchronous
  - Symmetric

Notes:

- Controller-to switch
  - Used by the controller to : add, modify, delete flow table entries and query statistics and features
- Asynchronous
  - Sent from the switch to the controller without any solicitation of the controller
- Symmetric
  - E.g. hello message

So OpenFlow...

- Communication protocol controller ↔ OpenFlow switch
- Runs over TCP
- Security: TLS (non mandatory, but encouraged!)
- Switching based on flows and tables
- Current specification version 1.5.1 (May 2015), 1.6 (September 2016)
- Probably being to be replaced with a more flexible alternative

Notes:

- Latest extensions: SPTN OpenFlow Protocol Extensions Juin 2017), Optical Transport Protocol Extensions (May 2017)
- Some considerations from [6]:
  - Is the original interface supporting disaggregation, it was hugely instrumental in launching the SDN journey, but it proved to be only a small part of what defines SDN today.
  - Equating SDN with OpenFlow significantly under-values SDN, but it is an important milestone because it introduced Flow Rules as a simple-but-powerful way to specify the forwarding behavior.
  - Today, work is underway to replace OpenFlow with a more flexible (i.e., programmable) alternative.

OpenFlow is not the only possible southbound interface

Other southbound interfaces:

- NETCONF
- XMPP
- Open vSwitch Data Base Management Protocol (OVSDB)
- Cisco's onePK

Notes:

- In particular you will see in the lab. OVSDB and OpenFlow

What is NETCONF/YANG? Is it SDN-oriented?

Notes:

NETCONF/YANG

NETCONF: (Network Configuration Protocol)

- provides mechanisms to install, manipulate, and delete the configuration of network devices
- standardised by IETF
- can be seen as another southbound SDN interface
- usually used along with YANG

YANG (Yet Another Next Generation)

- a data **modelling language**
- provides a **standardized way** to model the operational and configuration data of a network device

NETCONF/YANG provides a standardized way to programmatically update and modify the configuration of a network device

Notes:

NETCONF vs OpenFlow?

OpenFlow was the first SDN southbound standard, but no longer the only one/preferred one.

Are there in contradiction? Not really...

- NETCONF is a protocol that allows to modify networking device's configuration
- OpenFlow is a protocol that allows to modify its forwarding table

Is NETCONF/YANG SDN-friendly?

- NETCONF does not really separate control from data plane, why?
- but solves some of existing problems we've already seen, which ones?

Notes:

What is NFV?

Is it the same as SDN? Are they concurrents?

Notes:

- NFV: Network Function Virtualisation

NFV: Network Function Virtualization

Def. (wikipedia)

is a network architecture concept that uses the technologies of IT virtualization to virtualize entire classes of network node functions into building blocks that may connect, or chain together, to create communication services.

Notes:

- Application of the concept of VMs to the telecommunications world
- Instead of using proprietary closed systems, use VMs on servers with adequate software to fulfill network functions

NFV

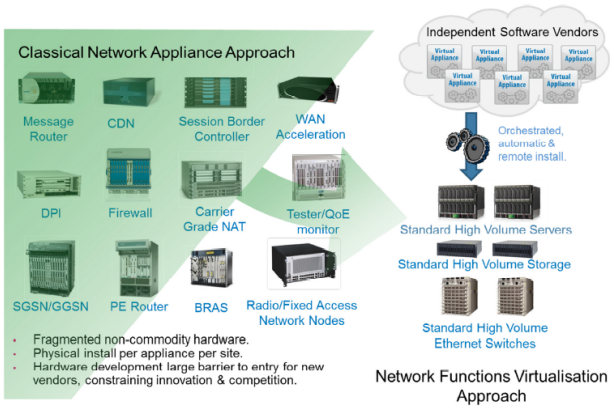
Def. (ETSI [3])

NFV transforms network architectures through the implementation of network functions in software that can run on a range of industry standard server hardware...

Notes:

- ETSI: European Telecommunications Standards Institute. An independent, not-for-profit, standardization organization in the telecommunications industry (equipment makers and network operators) in Europe
- In November 2012 seven of the world's leading telecoms network operators selected ETSI to be the home of the Industry Specification Group for NFV

NFV: from specialized closed systems to network functions implemented in white boxes



Notes:

- Source: `https://portal.etsi.org/NFV/NFV_White_Paper2.pdf`

In NFV, Network functions are:

- Written in software
- Deployed within data centers or in "white boxes" programmed as network nodes
- Objectives: agility, easy modification, service oriented networking

we have thus virtual network functions (VNFs)

Notes:

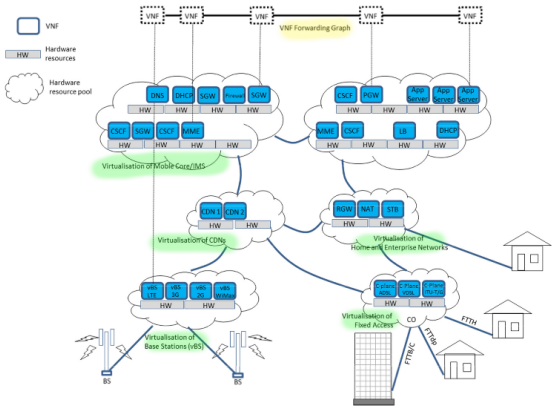
VNFs

- Analogy with cloud computing
- Virtualization of Network Functions
- Use cases: Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS), Network as a Service (NaaS)

Notes:

- For example, a VNF owner doesn't necessarily own the NFV Infrastructure needed for the proper functioning and operation of the VNF

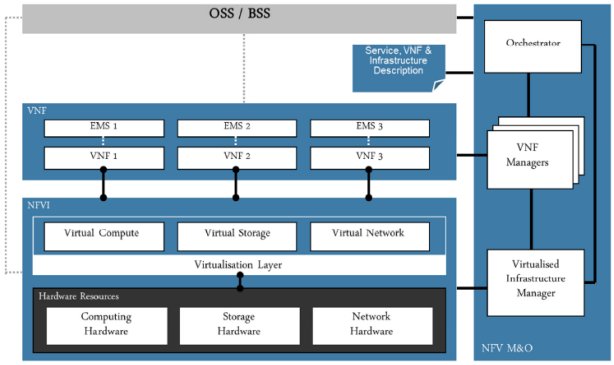
VNFs examples



Notes:

- Lets take one example: Content Delivery Networks (CDN) cache
  - CDNs commonly deploy content caches near the edge of a network to improve customers' quality of experience.
  - Today, caches use dedicated hardware on a per-CDN provider, per-operator basis.
  - These HW remain under-utilised for most of their lifetime (dimensioned for peak load)
  - With virtualised caches, the underlying HW resources could be shared among multiple providers' CDN caches and potentially other VNFs, thus improving resources usage.
- Other candidates for virtualisation non shown in the figure: middle boxes such as NAT, load balancer, firewall, etc.
- Forwarding graph or service chain: organization of all the needed VNFs for a given networking service
- Source: `https://portal.etsi.org/NFV/NFV_White_Paper2.pdf`

NFV architecture proposed by ETSI



Source: https://portal.etsi.org/NFV/NFV_White_Paper2.pdf

Notes:

- NFVI (Network Functions Virtualisation Infrastructure) provides the virtual resources required to support the execution of the VNFs. It includes Commercial-Off-The-Shelf (COTS) hardware, accelerator components where necessary, and a software layer which virtualises and abstracts the underlying hardware.
- VNF (Virtualised Network Function) software implementation of a network function, capable of running over the NFVI. It can be accompanied by an Element Management System (EMS), which understands and manages an individual VNF and its peculiarities.
- VNF entity corresponding to today's network nodes, are expected to be delivered as pure software free from hardware dependency.
- NFV M&O (Management and Orchestration) covers orchestration and lifecycle management of physical and/or software resources that support the infrastructure virtualisation, and the lifecycle management of VNFs.
- NFV M&O focuses on the virtualisation-specific management tasks necessary in the NFV framework.
- NFV M&O interacts with (NFV external) OSS/BSS landscape, which allows NFV to be integrated into an already existing network-wide management landscape.

NFV and SDN can be independent but are complementary

- NFV alone: VNFs on VMs running on commodity servers interconnected by standard networking
- SDN alone: Network Functions on dedicated hardware interconnected through an SDN controller
- NFV + SDN: VNFs on VMs running on commodity servers interconnected through an SDN controller

Notes:

- SDN eases the provisioning of network services by chaining VNFs
- The SDN controller can be implemented as a VNF

SDN Open questions/issues

- Scalability?
- Redundancy?
- Controller placement?
- Controller-controller interface? not standardized yet

Notes:

Are there real products and deployments?

Notes:

SDN Controllers

- There are several software controllers
- Some of them: OpenDayLight, ONOS, floodlight ...
- Others more academic POX, Ryu, ...
- Vendors such as Cisco have also their controller solutions

Notes:

- In particular, you will use POX controller in the lab. POX provides a python API to OpenFlow messages

OpenFlow Switches

- There are commercial switches supporting OpenFlow
- Vendors including Cisco, Juniper, Big Switch Networks, Brocade, Arista, Extreme Networks, IBM, Dell, NoviFlow, HP, NEC, among others
- But vendors also sell their own SDN solutions based on their own abstractions and interfaces

Notes:

Some ONF active projects

- CORD (Central Office Re-architected as a Datacenter)
- ONOS (Open Network Operating System) is an SDN controller
- Mininet
- SD-RAN SDN for 5G Radio Access Network

Notes:

- A project re-thinking the central office as a datacenter, with everything-as-a-service
- CORD platform leverages SDN, NFV and Cloud technologies to build agile datacenters for the network edge (central office). Integrating multiple open source projects, CORD delivers a cloud-native, open, programmable, agile platform for network operators to create innovative services.

Google's SDN deployment: A well-know success story [4]

- Private WAN connecting Google's data centers across the planet
- SDN architecture using OpenFlow switches
- Before: long convergence time for setting up MPLS tunnels
- After: centralized traffic engineering service drives links to near 100% utilization
- Splitting application flows among multiple paths to balance capacity against application priority/demands

Notes:

SDN Adoption

SDN has seen wide adoption:

- across data centers (64%),
- WANs (58%),
- and access networks (40%).

As for the "2020 Global Networking Trends report" [1].

Notes:

So SDN and folks: Disruptive or Accommodating Technologies?

Notes:

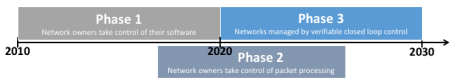
Conclusion

- New networking paradigms, key words: Programmability, agility, virtualisation, abstraction
- Great interest of the industry (some adoptions + members of standardization groups)
- Earliest adoptions in the Datacenters
- Two killing use-cases
  - SD-WAN
  - 5G network slicing
- Further adoption?
- will SDN controller become a new vendor lock-in product?

Notes:

- SD-WAN (software-defined networking in a wide area network (WAN)).
- SDN for the management and operation of a WAN
- It allows companies to build high-performance WANs using lower-cost and commercially available Internet access than todays VPN's and MPLS's based solutions.

Future of SDN



Projecting into the future, with Phase 3 of SDN focusing on verifiable, top-down control of network behavior [6].

Notes:

Conclusion

Whats next in networking?

- Intent-based networking
- Self-driven networks
- 'From Automated to Autonomous Networks' (as by BikasKoley, Google)
- The Rise of Network as a Service (NaaS) (as by '2022 Global Networking Trends Report', Cisco)

Notes:

References I

[1] Cisco 2020 Global Networking Trends.
Technical report, 2020.

[2] Martin Casado, Michael J. Freedman, Justin Pettit, Jianying Luo, Nick McKeown, and Scott Shenker.
Ethane: Taking Control of the Enterprise.
SIGCOMM Comput. Commun. Rev., 37(4):1–12, August 2007.

[3] ETSI; European Telecommunications Standards Institute ISG NFV.
http://www.etsi.org/technologies-clusters/technologies/nfv.

[4] Sushant Jain, Alok Kumar, Subhasree Mandal, Joon Ong, Leon Poutievski, Arjun Singh, Subbaiah Venkata, Jim Wanderer, Junlan Zhou, Min Zhu, Jon Zolla, Urs Hölzle, Stephen Stuart, and Amin Vahdat.
B4: Experience with a Globally-deployed Software Defined Wan.

Notes:

References II

SIGCOMM Comput. Commun. Rev., 43(4):3–14, August 2013.

[5] ONF: Open Networking Foundation.
https://www.opennetworking.org.

[6] Larry Peterson, Carmelo Cascone, Brian O'Connor, Thomas Vachuska, and Bruce Davie.
Software-Defined Networks: A Systems Approach.

Notes:

Acronyms

SDN	Software-defined networking
NFV	Network Function Virtualization
VNF	Virtual Network Function
ONF	Open Networking Foundation
NOS	Network Operating System
ODL	OpenDaylight
XMPP	Extensible Messaging and Presence Protocol
NETCONF	Network Configuration Protocol
VXLAN	Virtual eXtensible LAN
CDN	Content Delivery Network

Notes: