

SDN, NFV AND THEIR ROLE IN 5G

Presenters

Katia Obraczka	UC Santa Cruz, USA
Christian Rothenberg	UNICAMP, Brazil
Ahmad Rostami	Ericsson Research, Sweden

ACM SIGCOMM Tutorial | 2016-08-22 | Page 1

AGENDA

Part I (9:00-10:30)

Part II (11:00-12:30)

- 5G and the need for programmability
- Software Defined Networking (SDN)
- Network Function Virtualization (NFV)
- Research Efforts
- Use Cases
- Q&A

Coffee Break at 10:30 for 30 Min.

OUTLINE

- > **Introduction to 5G**: we will present a brief overview of 5G including the reference architecture, opportunities, and challenges.
- > SDN & NFV Concepts and their role in 5G: We will present SDN and NFV concepts and how they enable the vision of future 5G networks. Concepts such as network programmability and abstractions as well as network slicing will be introduced.
- > SDN & NFV Enabling Technologies: SDN technologies for the control- and data planes will be presented.. Notable examples of APIs for network abstraction and programmability, hardware acceleration, as well as NFV-enabling features such as service function chaining and service personalization will be covered.
- > Related Projects and Use Cases: We will briefly overview SDN and NFV efforts that target future 5G networks. We will also present use cases that can benefit from the synergy between SDN and NFV including dynamic service chaining, such as virtualized telecommunication services (e.g., vCPE, vIMS, vEPC, MEC) and NFV in radio access networks.
- > Research Challenges and Future Directions: In addition to the challenges in realizing 5G networks, we will further discuss research challenges posed by SDN and NFV including performance monitoring, scalability, management and orchestration, security, portability and heterogeneous network support.

REFERENCES / LITERATURE (1/2)

- > Nunes, B. A., Mendonca, M., Nguyen, X. N., **Obraczka, K.,** & Turletti, T. (2014). A survey of software-defined networking: Past, present, and future of programmable networks. IEEE Communications Surveys & Tutorials.
- > Kreutz, D., Ramos, F. M., Esteves Verissimo, P., **Esteve Rothenberg, C.**, Azodolmolky, S., & Uhlig, S. (2015). Software-defined networking: A comprehensive survey. Proceedings of the IEEE.
- > Rostami, A., "The Evolution of Programmable Networks: from Active Networks to Software Defined Networks (SDN)," Tutorial presented at 26th International Teletraffic Congress (ITC), 2014.
- > Mijumbi, R., Serrat, J., Gorricho, J. L., Latré, S., Charalambides, M., & Lopez, D. (2016). Management and orchestration challenges in network functions virtualization. Communications Magazine, IEEE.
- S. Abdelwahab, B. Hamdaoui, M. Guizani and T. Znati, "Network function virtualization in 5G," in *IEEE Communications Magazine*, vol. 54, no. 4, pp. 84-91, April 2016.
- > Ericsson White paper, "5G system enabling industry and society transformation," 2015. https://www.ericsson.com/res/docs/whitepapers/what-is-a-5g-system.pdf
- > Ericsson Mobility Report 2016. https://www.ericsson.com/res/docs/2016/ericsson-mobility-report-2016.pdf
- > Ericsson White Paper, "Cellular networks for massive IoT," https://www.ericsson.com/res/docs/whitepapers/wp_iot.pdf
- > Ericsson Technology Review, "A vision of the 5G core: flexibility for new business opportunities."
- > Ericsson White Paper, "Cloud RAN the benefits of virtualization, centralization and coordination," https://www.ericsson.com/res/docs/whitepapers/wp-cloud-ran.pdf

REFERENCES / LITERATURE (2/2)

- > I. T. Haque; N. Abu-Ghazaleh. 2016. Wireless Software Defined Networking: a Survey and Taxonomy," in *IEEE Communications Surveys & Tutorials*.
- Akram Hakiri and Pascal Berthou (2015). Leveraging SDN for The 5G Networks: Trends, Prospects and Challenges. http://arxiv.org/abs/1506.02876
- Nachikethas A. Jagadeesan and Bhaskar Krishnamachari. 2014. Software-Defined Networking Paradigms in Wireless Networks: A Survey. ACM Comput. Surv.
- > Akyildiz I.F., Wang P., Lin S.C. 'SoftAir: A software defined networking architecture for 5G wireless systems', Computer Networks 85 (2015) pp.1–18.
- Bradai A., Singh K., Ahmed T., and Rasheed T., 'Cellular Software Defined Networking: A framework'. IEEE
 Communications Magazine Communications Standards Supplement, June 2015, pp. 36-43
- › Guerzoni R., Trivisonno R., Soldani D. 'SDN-Based Architecture and Procedures for 5G Networks'. 5GU 2014, November 26-27, Levi, Finland, DOI 10.4108/icst.5gu.2014.258052
- Mao Yang, Yong Li, Depeng Jin, Lieguang Zeng, Xin Wu, and Athanasios V. Vasilakos. 2015. Software-Defined and Virtualized Future Mobile and Wireless Networks: A Survey. Mob. Netw. Appl. 20, 1
- > Nguyen, V., Do, T. & Kim, Y. SDN and Virtualization-Based LTE Mobile Network Architectures: A Comprehensive Survey. Wireless Pers Commun (2016) 86: 1401. doi:10.1007/s11277-015-2997-7

5G AND THE NEED FOR PROGRAMMABILITY

DRIVING FORCES BEHIND 5G

Networked Society

- Improved User Experience
- Massive Traffic Volumes
- Massive No. of Connected Devices
- Massive No. of Services (e.g., IoT)
- Transformed Industries

Technical Drivers

- Network and Service automation
- Resource & Energy Efficiency
- Virtualization & Clouds
- New HW and SW Technologies (SDN & NFV)

















ARCHITECTURE



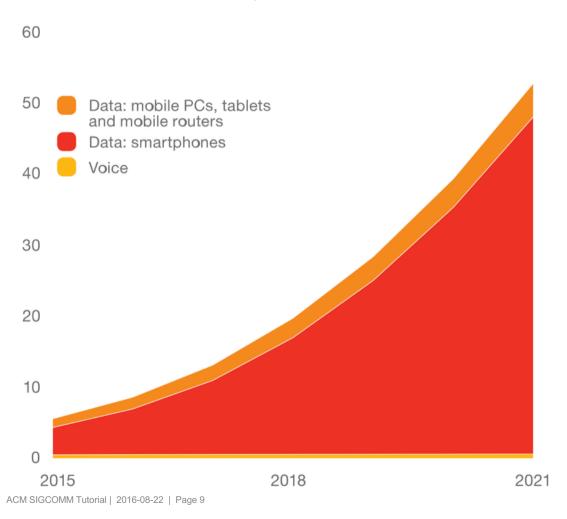
Access/Mobility Network Applications

Cloud/DC Infrastructure

Transport



GLOBAL MOBILE TRAFFIC (MONTHLY EXABYTES)



~45% CAGR

12X Growth in smartphone traffic

Around 90% of mobile traffic will be from smartphones by the end of 2021

source: Ericsson Mobility Report

IOT TO SURPASS MOBILE PHONES IN 2018

Connected devices (billions)



	15 billion	28 billion	CAGR 2015–2021
Cellular IoT	0.4	1.5	27%
Non-cellular IoT	4.2	14.2	22%
PC/laptop/tablet	1.7	1.8	1%
Mobile phones	7.1	8.6	3%
Fixed phones	1.3	1.4	0%
	2015	2021	

source: Ericsson Mobility Report

EVOLUTION TOWARDS 5G

1000x Mobile Data Volumes

10x-100x Connected Devices

5x Lower Latency

3G 4G 5G

-2020

1990

~2000

~2010

10x-100x End-user Data Rates 10x

Battery Life for Low Power Devices

Source: METIS







SMART VEHICLES, TRANSPORT & INFRASTRUCTURE



MEDIA EVERYWHERE



CRITICAL CONTROL
OF REMOTE DEVICES

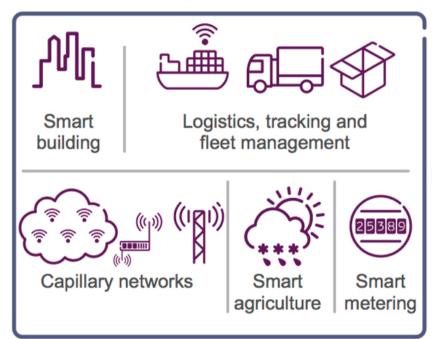


INTERACTION HUMAN-IOT

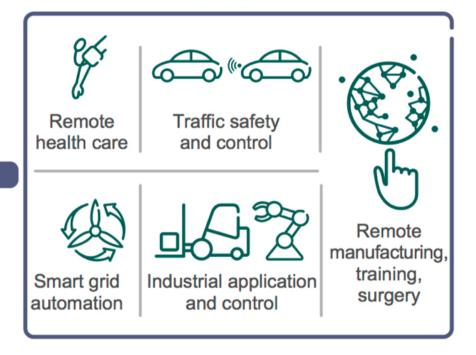
source: Ericsson 5G use cases

DIVERSE REQUIREMENTS OF IOT

Massive IoT



Critical IoT



Low cost, low energy, small data volumes, massive numbers

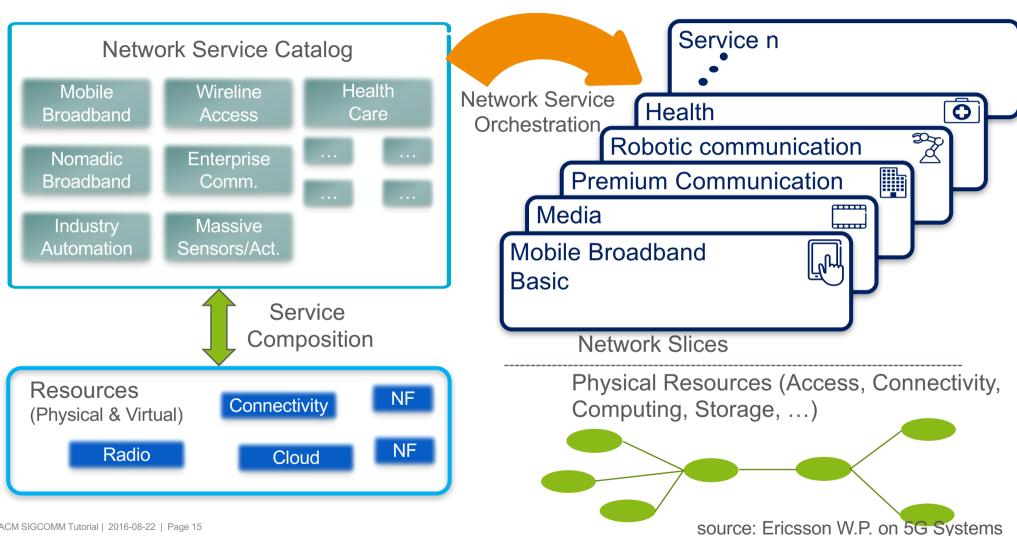
Ultra reliable, very low latency, very high availability

ONE NETWORK - MULTIPLE INDUSTRIES

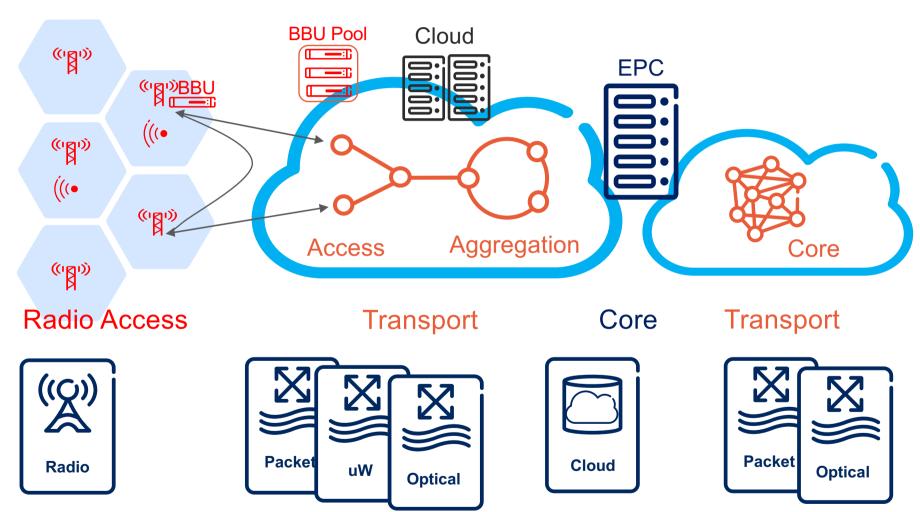


source: Ericsson W.P. on 5G Systems

NFTWORK AS A SERVICE



NETWORK ARCHITECTURE



ACM SIGCOMM Tutorial | 2016-08-22 | Page 16

PROGRAMMABILITY IN 5G NETWORKS

High level of flexibility and programmability in individual domains (mobile core, radio access network and transport network).

Cross-domain programmability and orchestration.

Service Agility

Shorten the time for service creation and service adaptation (e.g., scaling).



Service Diversity

Share a single infrastructure among multiple services with wide range of requirements.



Resource Efficiency

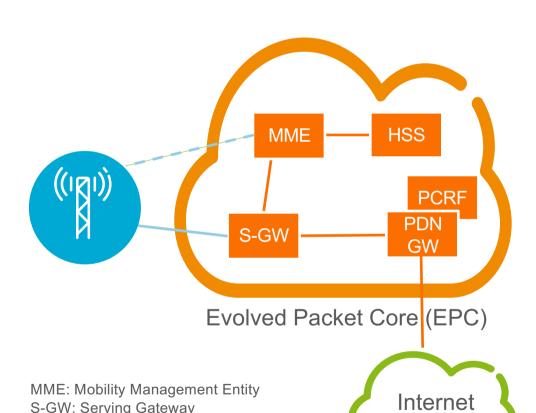
Dynamically allocate the right amount of resources when and where needed.



ACM SIGCOMM Tutorial | 2016-08-22 | Page 17

PROGRAMMABILITY IN MOBILE CORE

CURRENT MOBILE CORE ARCHITECTURE



- A single network architecture for multiple services
- Mix of control and user plane functions
- Appliance-based realization



- Difficult to customize
- Scalability

ACM SIGCOMM Tutorial | 2016-08-22 | Page 19

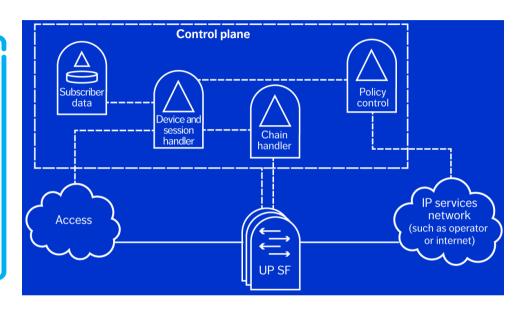
HSS: Home Subscriber Server

PCRF: Policy & Charging Rules Function

S-GW: Serving Gateway PDN: Packet Data Network

FLEXIBLE CORE ARCHITECTURE

- Separation of control and userplane functions
- Decompose core functionality into granular functions
- Virtualize functions





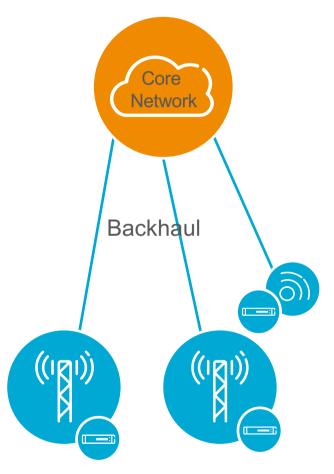
- Customize realization per service/slice
- Centralized control functions

- Selective scaling
- Utilize Cloud Environment
- Flexible placement of functions

Network Function Virtualization (NFV) is an enabler for programmability in mobile core.

PROGRAMMABILITY IN RAN

RAN DEPLOYMENTS - I

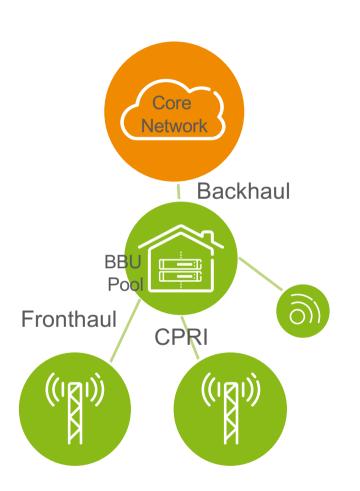


Distributed Baseband

- Flat Architecture
 - Scaling
- IP connectivity between RAN and Core, and among sites



RAN DEPLOYMENTS - II



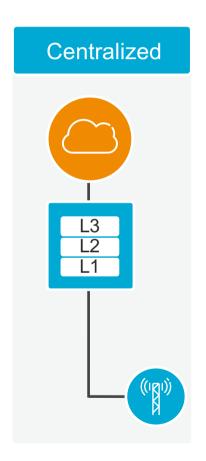
Centralized Baseband (C-RAN)

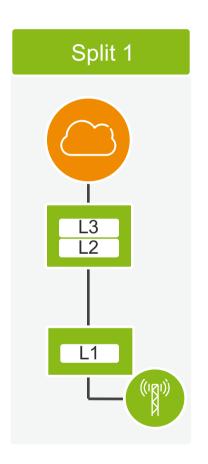
- Pooling gains
- Efficient network management
- Efficient coordination & interference management
- Less network signaling
- Stringent performance requirement on fronthaul (BW, delay and jitter)
- might not be scalable in all 5G scenarios

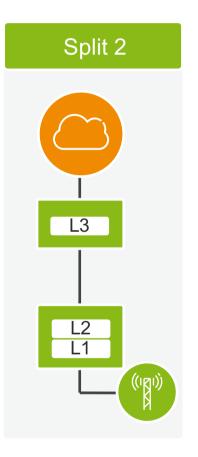
Need for more flexible split of RAN

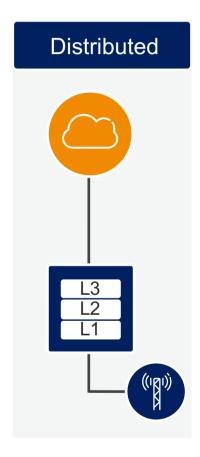
CPRI: Common Public Radio Interface BBU: Baseband processing Unit

CLOUD RAN

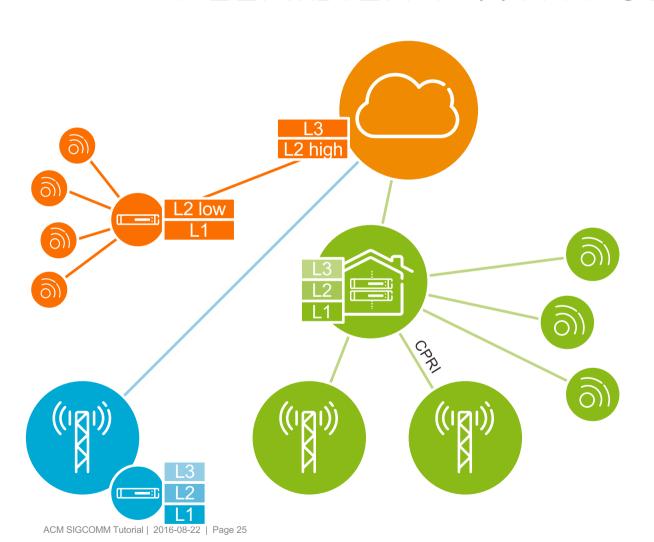








FLEXIBILITY WITH CLOUD RAN



Centralization gains as in C-RAN

- Pooling
- Network Management
- Coordination

Less Transport Requirements

Virtualization gains

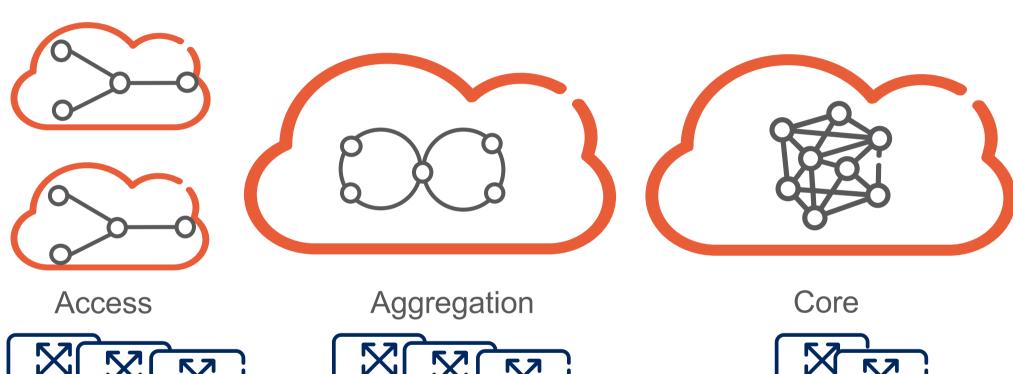
- Selective scaling (E.g. User plane vs Control Plane)
- Cloud-based Realization

Collocation of RAN & Core

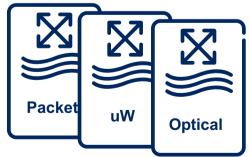
Source of fig.: Ericsson W.P. on Cloud RAN

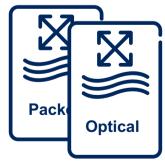
PROGRAMMABILITY IN TRANSPORT

CURRENT TRANSPORT NETWORKS



Packet uW Optical





ACM SIGCOMM Tutorial | 2016-08-22 | Page 27

CURRENT TRANSPORT NETWORK ISSUES

- Monolithic realization of control and forwarding functions
- Proprietary Management
 Interfaces
- Complex control and management
- Several technology domains with independent control

- Lengthy and manual service creation/scaling
- Inefficient Resource Utilization
- Inefficient static sharing
- Difficult cross-layer optimization
- Application Unaware

PROGRAMMABLE TRANSPORT

- Separation of control and forwarding functions
- Define interfaces between control and forwarding
- Open up the control plane for programming
- Develop Efficient sharing mechanisms

- Automation of network and services
- Dynamic creation/update of (virtual) connections/tunnels
- Resource-optimized operation
- Cross-layer optimization (e.g. packet-optical convergence)
- Radio-aware adaptations

Software-Defined Networking (SDN) is an enabler for programmability in transport networks.

SOFTWARE-DEFINED NETWORKING

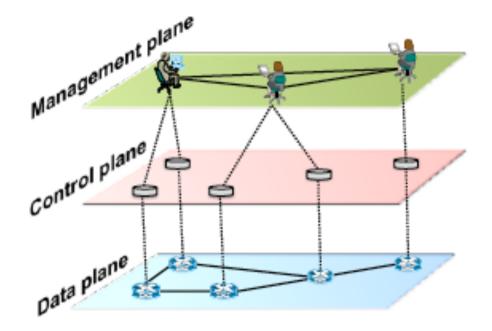
Towards Network Programmability

OUTLINE

- > What's Software-Defined Networking?
- > Why Software-Defined Networking?
- Some History
- > SDN Architectures
- > Definitions, Terminology, Concepts
- > Example Applications

WHAT'S SOFTWARE-DEFINED NETWORKING?

NETWORK'S FUNCTIONAL PLANES



Source: "Software-Defined Networking: A Comprehensive Survey", Kreutz et al., https://arxiv.org/pdf/1406.0440.

WHAT'S SOFTWARE-DEFINED NETWORKING?

> Main principle: data plane decoupled from control plane.

WHY SOFTWARE-DEFINED NETWORKING?

- The Internet has been the victim of its own success!
- > Extremely hard to configure, manage, and evolve.
- "Vertically integrated": tight coupling of control- and data planes embedded/distributed in network devices.
- > Proliferation of "middleboxes".



Vertically integrated Closed, proprietary Slow innovation Small industry

WHY SOFTWARE-DEFINED NETWORKING?

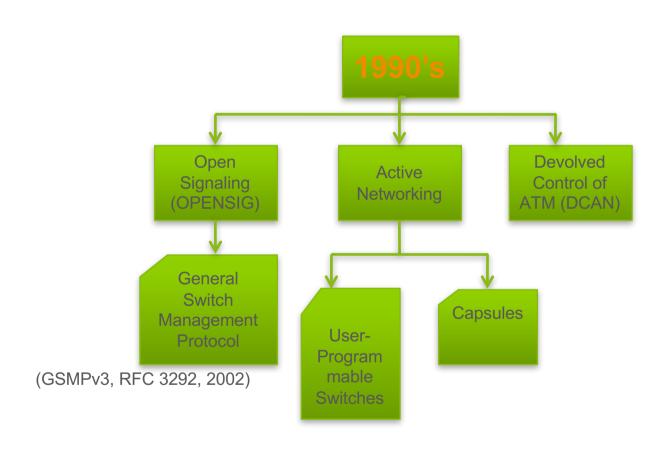
- The Internet has been the victim of its own success!
- Extremely hard to configure, manage, and evolve.
- "Vertically integrated": tight coupling of control- and data planes embedded/distributed in network devices.
- Proliferation of "middleboxes".

Software-Defined Networking to the rescue!

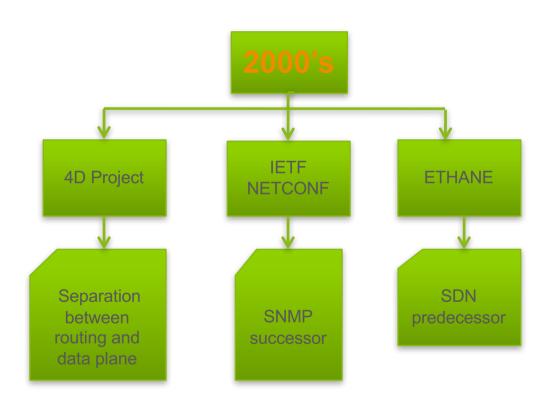
- Separation of control plane from data plane.
 - Network control "logically centralized" in the controller.
 - Forwarding hardware simplified.
- Programmable networks to facilitate management and control and combat "network ossification".
- > Data plane "commoditization".

SOFTWARE-DEFINED NETWORKING: SOME HISTORY

PROGRAMMABLE NETWORKS: SOME HISTORY



PROGRAMMABLE NETWORKS: SOME HISTORY



SOFTWARE-DEFINED NETWORKING ARCHITECTURES

IETF ForCES

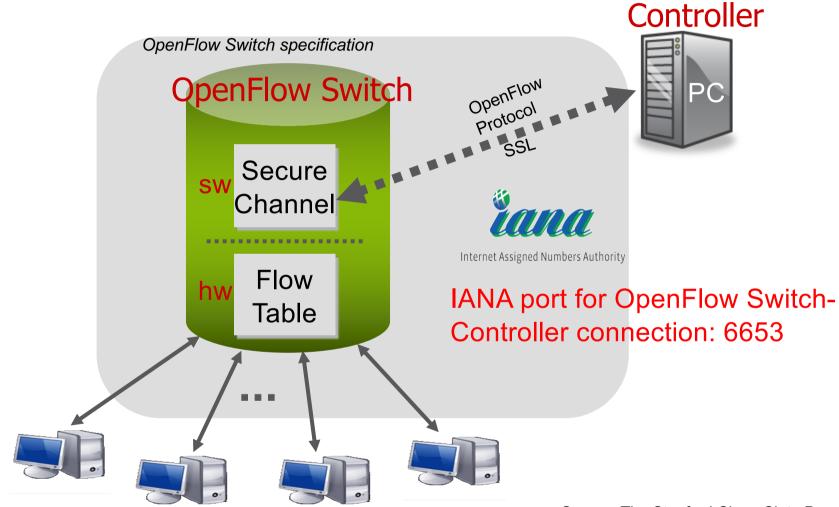
- Forwarding Element (FE) and Control Element (CE)
- > Both reside in the network device
- FE and CE communicate using the ForCES protocol

ONF OpenFlow

- Decoupling between control- and data planes
- Controller and switch communicate using the OpenFlow protocol

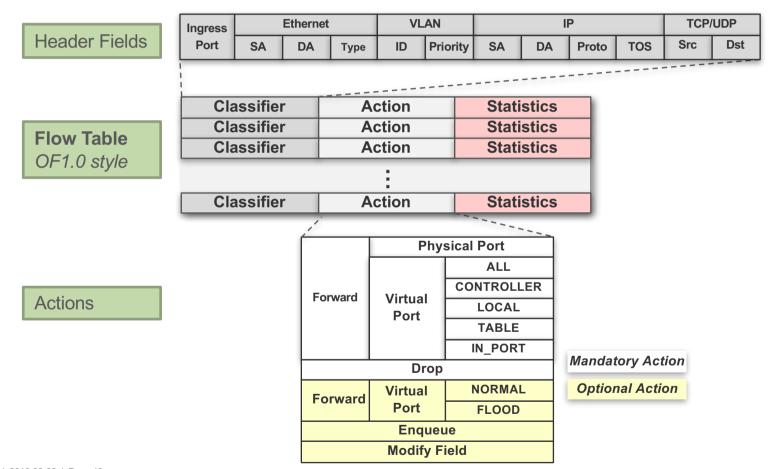
ONF OPENFLOW ARCHITECTURE

ONF OPENFLOW ARCHITECTURE



Source: The Stanford Clean Slate Program, http://cleanslate.stanford.edu

OPENFLOW 1.0 FLOW TABLE & FIELDS



OPENFLOW TABLE ENTRIES

	Switch port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Prot	TCP sport	TCP dport	Action
Switching	*	*	00:1f :	*	*	*	*	*	*	Port6
Flow switching	Port3	00:20	00:1f 	0800	Vlan1	1.2.3.4	5.6.7.8	4	17264	Port6
Firewall	*	*	*	*	*	*	*	*	22	Drop
Routing	*	*	*	*	*	*	5.6.7.8	*	*	Port6
VLAN switching	*	*	00:1f 	*	Vlan1	*	*	*	*	Port6,p ort7, port8

Each Flow Table entry has two timers:

hard_timeout

seconds after which the flow is removed zero mean never times-out

idle_timeout

seconds of no matching packets after which the flow is removed zero means never times-out

If both **idle_timeout** and **hard_timeout** are set, then the flow is removed when the first of the two expires.

OpenFlow Standards

Evolution path:

- OF 1.0 (03/2010): Most widely used version, MAC, IPv4, single table (from Stanford)
- OF 1.1 (02/2011): MPLS tags/tunnels, multiple tables, counters (from Stanford)
- OF 1.2 (12/2011): IPv6, extensible expression
- OF-Config 1.0 (01/2012): Basic configuration: queues, ports, controller assign
- OF 1.3.0 (04/2012): Tunnels, meters, PBB support, more IPv6
- OF-Config 1.1 (04/2012): Topology discovery, error handling
- OF-Test 1.0 (2H2012): Interoperability conformance test processes, suites, labs
- OF 1.3.2 (May 2013), 19 errata, final review
- OF 1.4 (Aug. 2013), 9 changes + 13 extensions, More extensible wire protocol, Flow monitoring, Eviction, Vacancy events, Bundles
- OF 1.5.1 (Dec. 2014), Egress Tables, Packet type aware pipeline, Extensible flow entry statistics

ACM SIGCOMM Tutorial | 2016-08-22 | Page 45

SDN: SOME DEFINITIONS

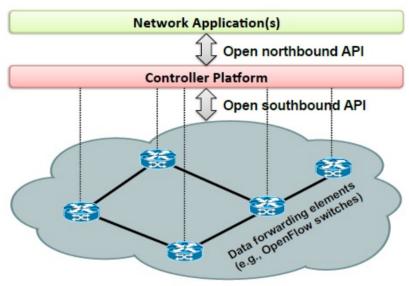
SDN: SOME DEFINITIONS

- "The SDN architecture decouples the network control and forwarding functions enabling the network control to become directly programmable and the underlying infrastructure to be abstracted for applications and network services." Open Networking Foundation (opennetworking.org)
- "Software Defined Networking (SDN) refactors the relationship between network devices and the software that controls them. Opening up the interfaces to programming the network enables more flexible and predictable network control, and makes it easier to extend the network with new functionality." ACM Sigcomm Simposium on Software-DefinedNetworking Research 2016

SDN refers to software-defined networking architectures where:

- Data- and control planes decoupled from one another.
- Data plane at forwarding devices managed and controlled remotely by a "controller".
- Well-defined programming interface between control- and data planes.
- Applications running on controller manage and control underlying data plane

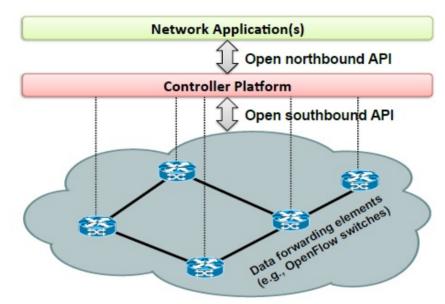
SDN architecture



Network Infrastructure

Source:

- Data plane: network infrastructure consisting of interconnected forwarding devices (a.k.a., forwarding plane).
- > Forwarding devices: data plane hardware- or software devices responsible for data forwarding.
- > Flow: sequence of packets between source-destination pair; flow packets receive identical service at forwarding devices.
- > Flow rules: instruction set that act on incoming packets (e.g., drop, forward to controller, etc)
- > Flow table: resides on switches and contains rules to handle flow packets.



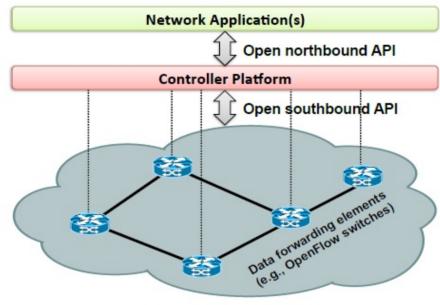
Network Infrastructure

Source:

Control plane: controls the data plane; logically centralized in the "controller" (a.k.a., network operating system).

Southbound interface:

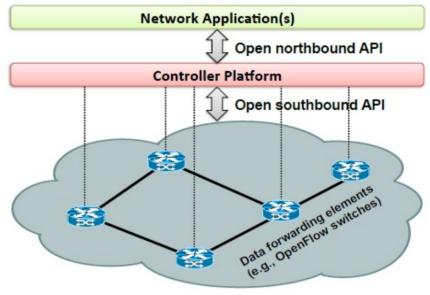
 (instruction set to program the data plane) + (protocol between control- and data planes).



Network Infrastructure

Source:

- Northbound interface: API offered by control plane to develop network control- and management applications.
- Management plane: functions, e.g., routing, traffic engineering, that use control plane functions and API to manage and control network infrastructure.



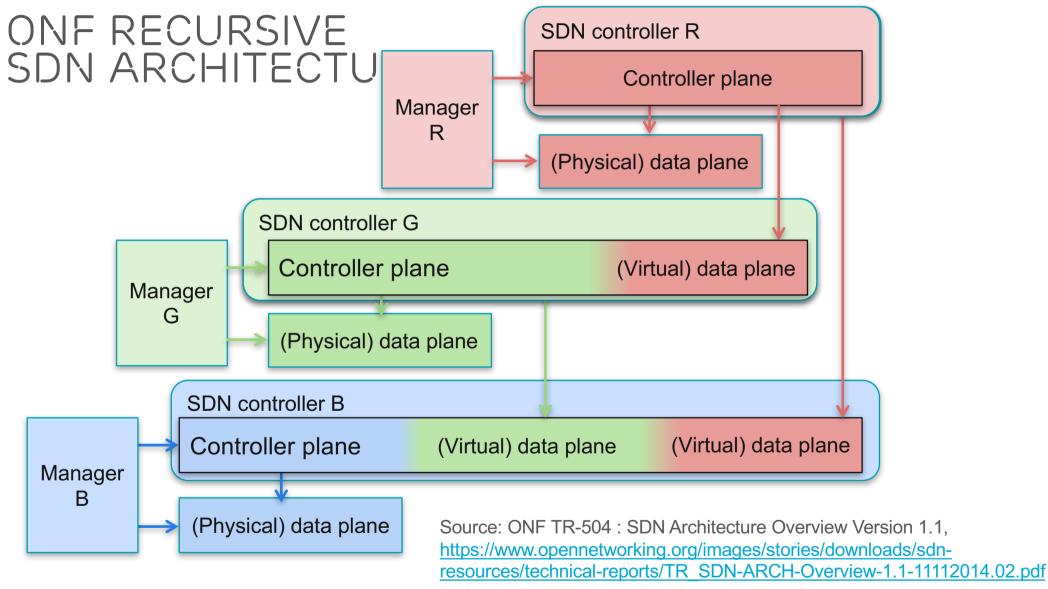
Network Infrastructure

Source:

SDN APPLICATIONS

- > Traffic engineering:
 - -Provide adequate QoS, improve network utilization, reduce power consumption, balance load.
- > Wireless network management/control and mobility support:
 - Seamless handover, load balancing, interoperability between heterogeneous networks, dynamic spectrum usage.

- Measurement and monitoring:
 - Packet sampling, traffic matrix estimation
- > Security:
 - Firewalling, access control, DoS attack detection/ mitigation, traffic anomaly detection.
- > Data-center networking:
 - Data center QOS and traffic engineering, fault detection and resilence, dynamic provisioning, security.



Network Function Virtualization (NFV)

Motivation

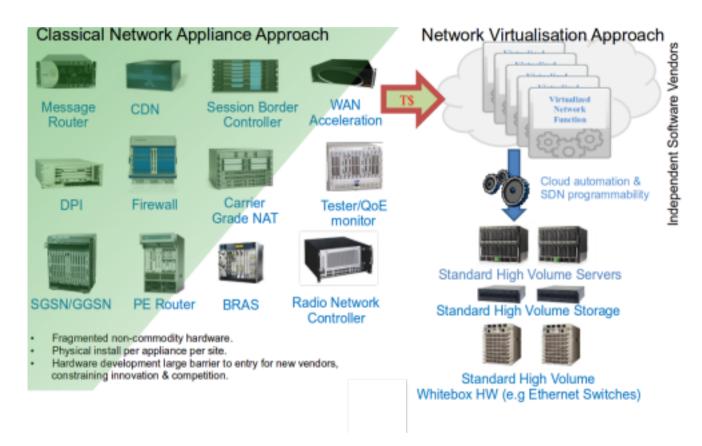


Source: Why Virtualization is Essential for 5G – Francis Chow (5G Summit 2015)

Problem Statement

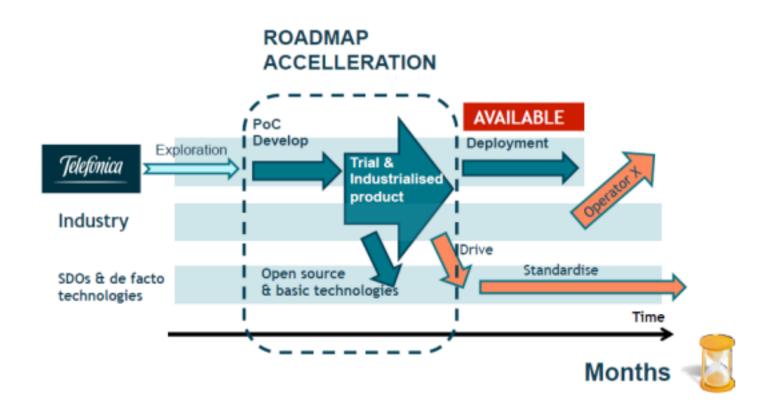
- Complex carrier networks
 - with a large variety of proprietary nodes and hardware appliances
- Launching new services is difficult and takes too long
- Space and power to accommodate
 - requires just another variety of box, which needs to be integrated
- Operation is expensive
 - Rapidly reach end of life due to
 - existing procure-design;
 - integrate-deploy cycle

Some Changes



Source: ETSI NFV ISG – DIRECTION & PRIORITIES – Steven Wright (NFV World Congress 2015)

Transformation



ACM SIGCOMM Tutorial | 2016-08-22 | Page 58 Source: Adapted from D. Lopez Telefonica I+D, NFV

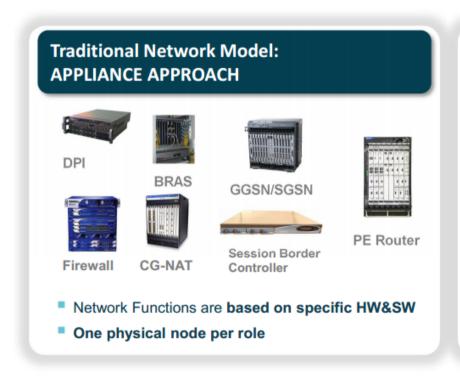
Why NFV?

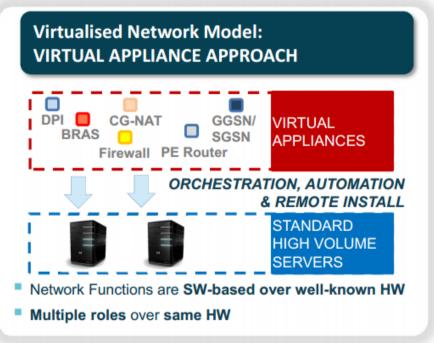
- **1. Virtualization:** Use network resource without worrying about where it is physically located, how much it is, how it is organized, etc.
- 2. Orchestration: Manage thousands of devices
- 3. Programmable: Should be able to change behavior on the fly.
- 4. Dynamic Scaling: Should be able to change size, quantity, as a F(load)
- **5. Automation:** Let machines / software do humans' work
- **6. Visibility:** Monitor resources, connectivity
- 7. Performance: Optimize network device utilization
- **8. Multi-tenancy:** Slice the network for different customers (as-a-Service)
- 9. Service Integration: Let network management play nice with OSS/BSS
- 10. Openness: Full choice of modular plug-ins

Note: These are exactly the same reasons why we need/want SDN.

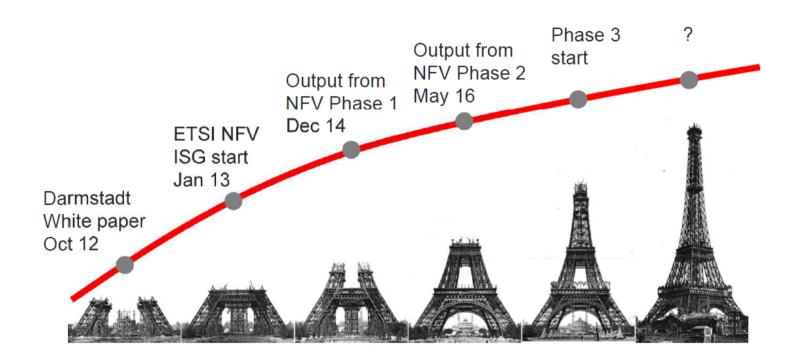
The NFV Concept

A means to make the network more flexible and simple by minimising dependence on HW constraints





The Making of NFV



Souce: NFV Orchestration | Fueling innovation in operator networks - Federico

Benefits & Promises of NFV

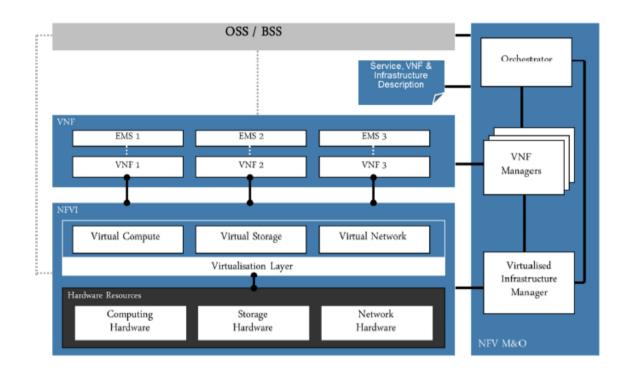
- Reduced equipment costs (CAPEX)
 - through consolidating equipment and economies of scale of IT industry.
- Increased speed of time to market
 - by minimising the typical network operator cycle of innovation.
- Availability of network appliance multi-version and multi-tenancy,
 - allows a single platform for different applications, users and tenants.
- Enables a variety of eco-systems and encourages openness.
- Encouraging innovation to bring new services and generate new revenue streams.

Source: NFV

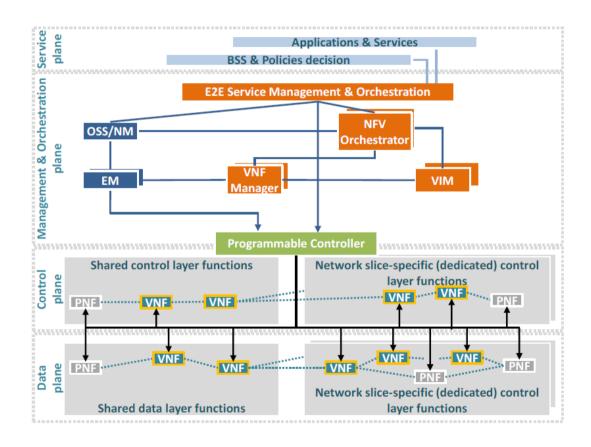
Benefits & Promises of NFV

- Flexibility to easily, rapidly, dynamically provision and instantiate new services in various locations
- Improved operational efficiency
 - by taking advantage of the higher uniformity of the physical network platform and its homogeneity to other support platforms.
- Software-oriented innovation to rapidly prototype and test new services and generate new revenue streams
- More service differentiation & customization
- Reduced (OPEX) operational costs: reduced power, reduced space, improved network monitoring
- IT-oriented skillset and talent

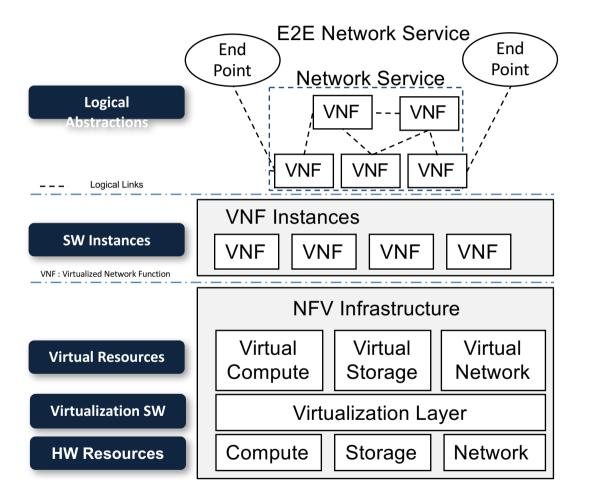
ETSI NFV Architectural Framework



NFV



NFV Layers



Rethinking relayering

applications

operating systems

hypervisors

compute infrastructure

network infrastructure

switching infrastructure

rack, cable, power, cooling

applications

network functions

operating systems

hypervisors

compute infrastructure

switching infrastructure

rack, cable, power, cooling

NFV Concepts

- Network Function (NF): Functional building block with a well defined interfaces and well defined functional behavior
- Virtualized Network Function (VNF): Software implementation of NF that can be deployed in a virtualized infrastructure
- VNF Set: Connectivity between VNFs is not specified, e.g., residential gateways
- VNF Forwarding Graph: Service chain when network connectivity order is important, e.g., firewall, NAT, load balancer
- **NFV Infrastructure (NFVI):** Hardware and software required to deploy, mange and execute VNFs including computation, networking, and storage.
- NFV Orchestrator: Automates the deployment, operation, management, coordination of VNFs and NFVI.

ACM SIGCOMM Tutorial | 2016-08-22 | Page 68

Source: Adapted from Raj Jain

NFV Concepts

- NFVI Point of Presence (PoP): Location of NFVI
- NFVI-PoP Network: Internal network
- Transport Network: Network connecting a PoP to other PoPs or external networks
- **VNF Manager:** VNF lifecycle management e.g., instantiation, update, scaling, query, monitoring, fault diagnosis, healing, termination
- Virtualized Infrastructure Manager: Management of computing, storage, network, software resources
- Network Service: A composition of network functions and defined by its functional and behavioral specification
- NFV Service: A network services using NFs with at least one VNF.

ACM SIGCOMM Tutorial | 2016-08-22 | Page 69

Source: Adapted from Raj Jain

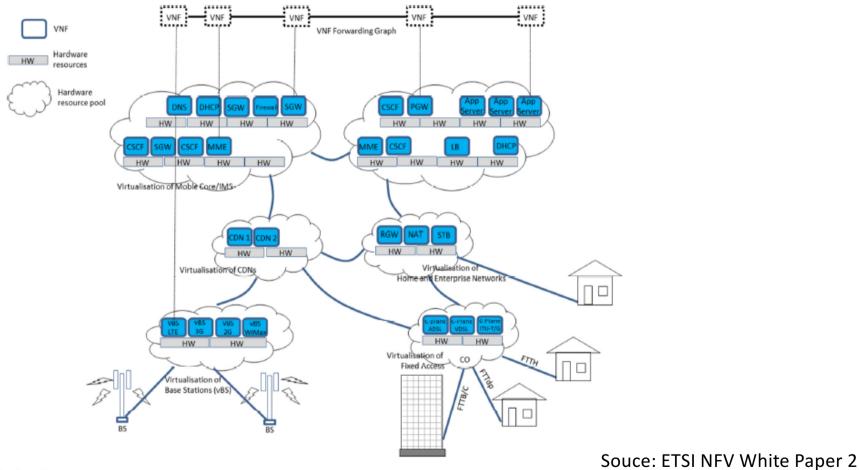
NFV Concepts

- User Service: Services offered to end users/customers/subscribers.
- **Deployment Behavior:** NFVI resources that a VNF requires, e.g., Number of VMs, memory, disk, images, bandwidth, latency
- Operational Behavior: VNF instance topology and lifecycle operations, e.g., start, stop, pause, migration, ...
- VNF Descriptor: Deployment behavior + Operational behavior

ACM SIGCOMM Tutorial | 2016-08-22 | Page 70

Source: Adapted from Raj Jain

Overview of ETSI NFV Use Cases



Architectural Use Cases

- Network Functions & Virtualisation Infrastructure as a Service
 - Network functions go cloudlike
- Virtual Network Function as a Service
 - Ubiquitous, delocalized network functions
- Virtual Network Platform as a Service
 - Applying multi-tenancy at the VNF level
- VNF Forwarding Graphs
 - Building E2E services by composition

Service-Oriented Use Cases

- Mobile core network and IMS
 - Elastic, scalable, more resilient EPC
 - Specially suitable for a phased approach
- Mobile base stations
 - Evolved Cloud-RAN
 - Enabler for SON
- Home environment
 - L2 visibility to the home network
 - Smooth introduction of residential services
- CDNs
 - Better adaptability to traffic surges
 - New collaborative service models
- Fixed access network
 - Offload computational intensive optimization
- Enable on-demand access services

NFV Framework Requirements

- 1. General: Partial or full Virtualization, Predictable performance
- 2. Portability: Decoupled from underlying infrastructure
- 3. Performance: Conforming and proportional to NFs specifications and facilities to monitor
- 4. Elasticity: Scalable to meet SLAs. Movable to other servers.
- 5. Resiliency: Be able to recreate after failure.

Specified packet loss rate, calls drops, time to recover, etc.

- 6. Security: Role-based authorization, authentication
- 7. Service Continuity: Seamless or non-seamless continuity after failures or migration

NFV Framework Requirements

- 8. Service Assurance: Time stamp and forward copies of packets for Fault detection
- 9. Energy Efficiency Requirements: Should be possible to put a subset of VNF in a power conserving sleep state
- 10. Operational and Management Requirements: Incorporate mechanisms for automation of operational and management functions
- 11. Transition: Coexistence with Legacy and Interoperability among multi-vendor implementations
- 12. Service Models: Operators may use NFV infrastructure operated by other operators

• • •

Challenging Path upfront:

Not as simple as cloud applied to telco

The network differs from the computing environment in 2 key factors...

Data plane workloads
(which are huge!)

HIGH PRESSURE ON **PERFORMANCE**

Network requires shape
(+ E2E interconnection)

GLOBAL NETWORK VIEW IS REQUIRED FOR MANAGEMENT

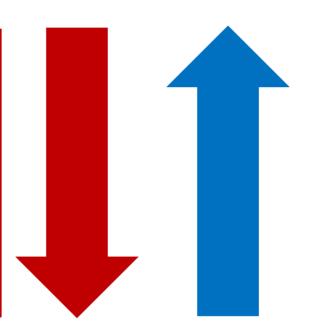
...which are big challenges for vanilla cloud computing.

AN ADAPTED VIRTUALISATION ENVIRONMENT IS NEEDED
TO OBTAIN CARRIER-CLASS BEHAVIOUR

The Road to NFV

Application-driven NFV

- Operator starts with a particular function or domain e.g. IMS
- Increase VNFs over time as technology & opportunity allow
- Faster, less risky; an opportunity to experiment



Platform-driven NFV

- Operator starts to develop a horizontal platform to run VNFs
- Evolve platform to support demanding workloads; add VNFs
- Strategic, disruptive, expensive; long-term

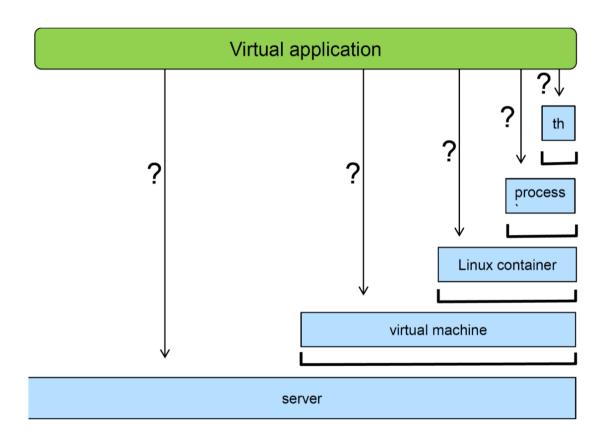
Source: Gabriel Brown, Heavy Reading

Fat vs. fit VNFs

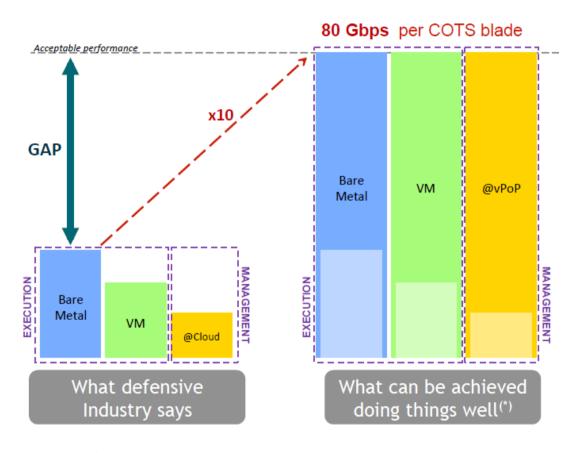
Thick/fat VNFs Thin/fit VNFs (apps) Virtual implementations of Single-service atomic network physical functions functions - Complex multiservice configurations -In-network load balancing, high availability - Integrated load balancing, high -Single interface to AAA availability, AAA IF -Scaling by adding network function - Scale in and out internal to VNF instances Move common Drawbacks Advantages functions to the - Complex to implement, integrate, and -Simple integration and testing infrastructure - Elasticity by simple integration with - Elasticity is difficult to implement orchestration - Slow rollouts, long time to revenue - Fast rollouts, short time to revenue

Source: SDN and NFV Stepping Stones to the Telco Cloud – Prodip Sen (ONS 2016)

Alternative options to virtualize NFV apps

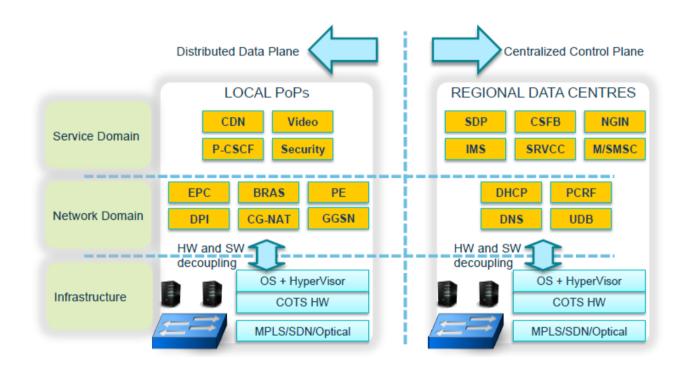


Performance Challenges



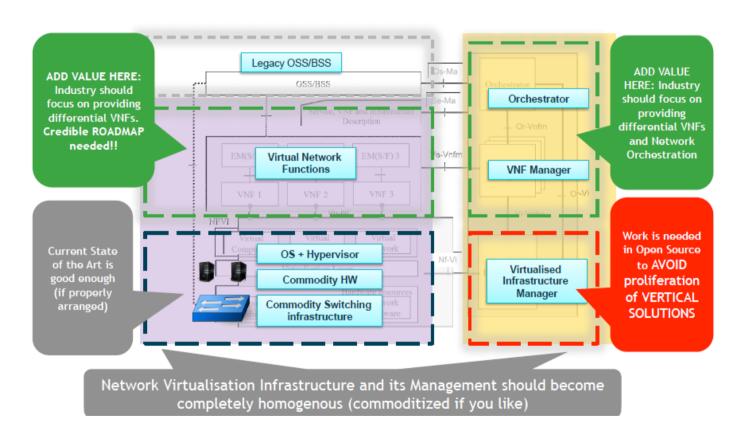
Souce: ETSI NFV White Paper 2

Portability Challenges



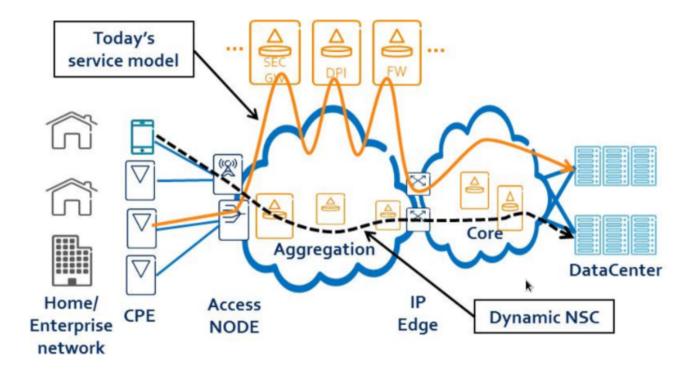
Source: Adapted from D. Lopez Telefonica I+D, NFV

Integration Challenges



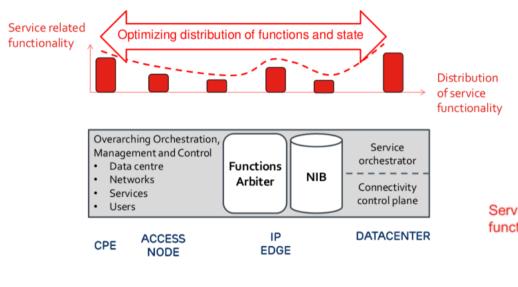
Source: Adapted from D. Lopez Telefonica I+D, NFV

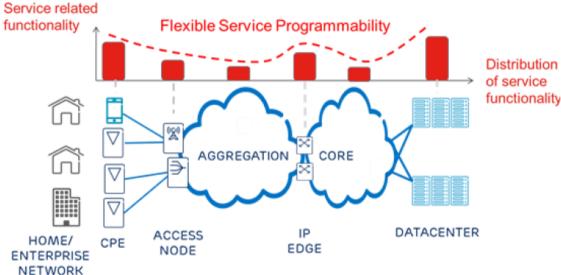
Elasticity Challenges



Source: UNIFY Project Results

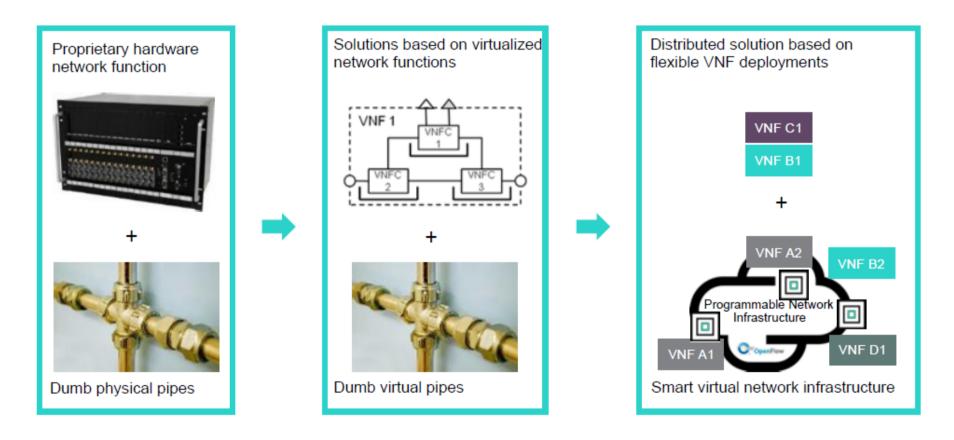
Management & Orchestration Challenges





ACM SIGCOMM Tutorial | 2016-08-22 | Page 84 Source: IEEE PIMRC, 2013

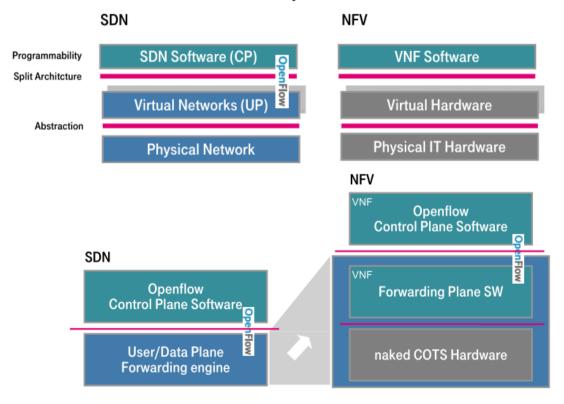
NFV & SDN



Source: SDN and NFV Stepping Stones to the Telco Cloud – Prodip Sen (ONS 2016)

SDN & NFV

SDN and NFV do NOT depend on each other



Source: Uwe Michel, T-Systems

NFV vs SDN

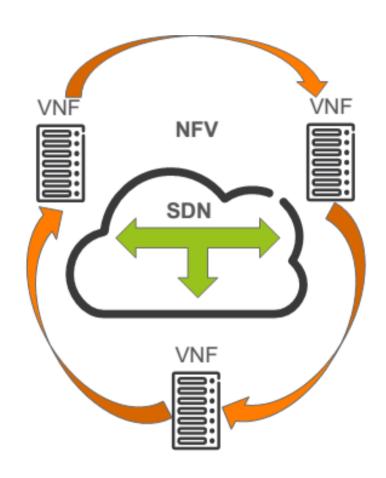
SDN >>> <u>flexible</u> forwarding & steering of traffic in a physical or virtual network environment

[Network Re-Architecture]

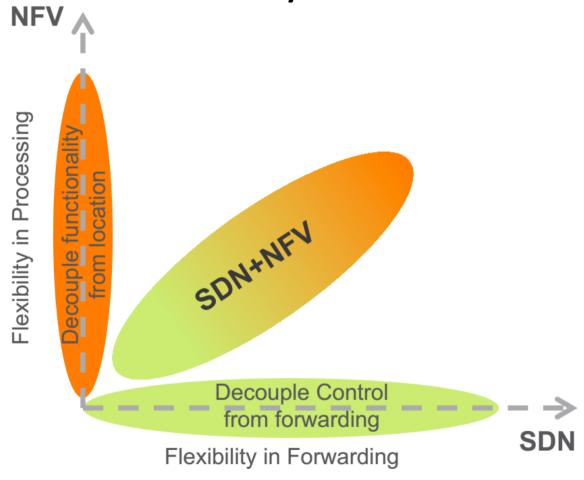
NFV >>> <u>flexible</u> placement of virtualized network functions across the network & cloud

[Appliance Re-Architecture] (initially)

>>> SDN & NFV are complementary tools for achieving full network programmability

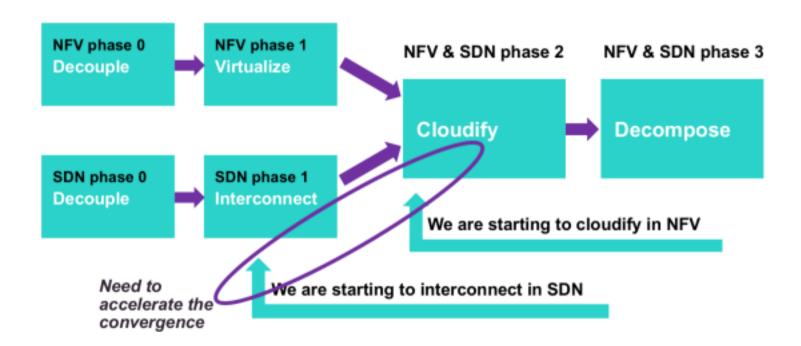


Flexibility with SDN & NFV



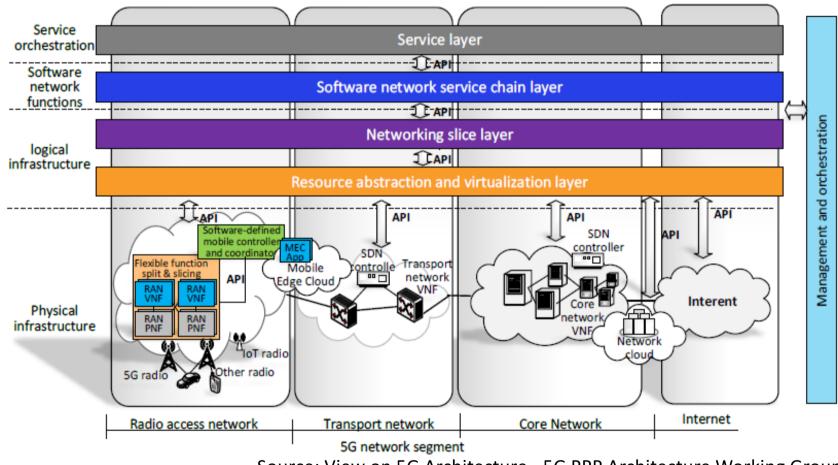
Source: Ahmad Rostami, Ericsson Research (Kista): http://www.itc26.org/fileadmin/ITC26_files/ITC26-Tutorial-Rostami.pdf

SDN & NFV Convergence

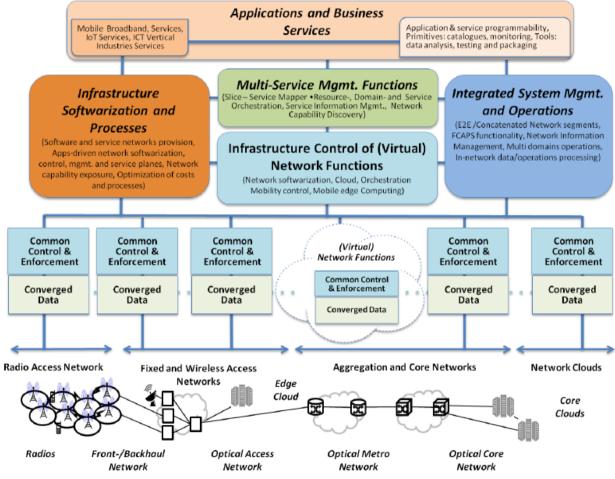


Source: SDN and NFV Stepping Stones to the Telco Cloud – Prodip Sen (ONS 2016)

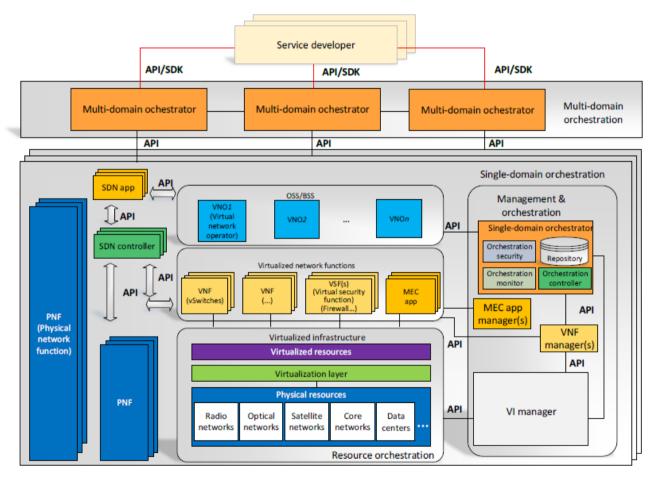
5G + (NFV & SDN)



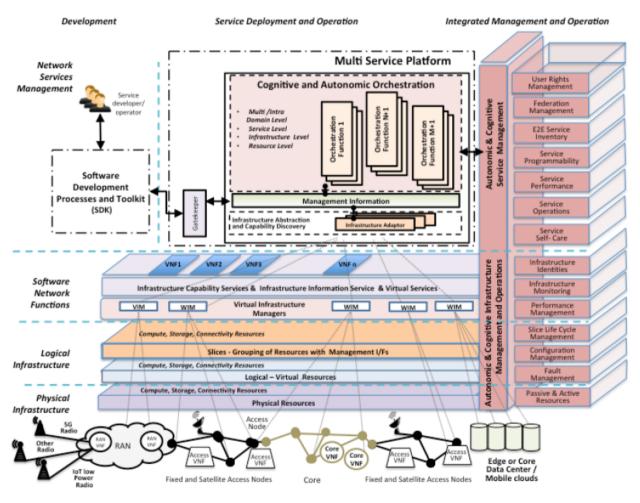
5G Softwarization and Programmability Framework



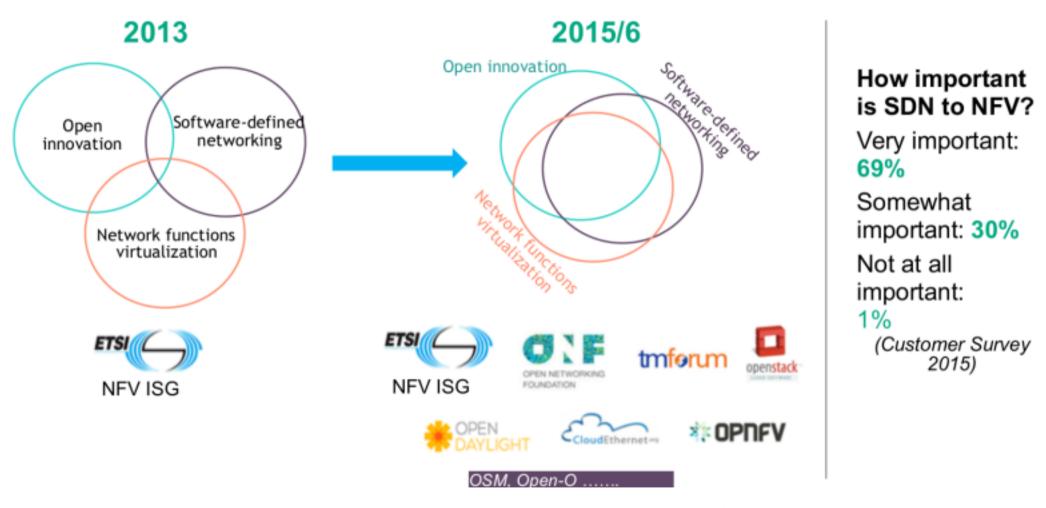
Multi Administrative Domains



The Big Picture



SDN/NFV Open Innovation



Source: SDN and NFV Stepping Stones to the Telco Cloud – Prodip Sen (ONS 2016)

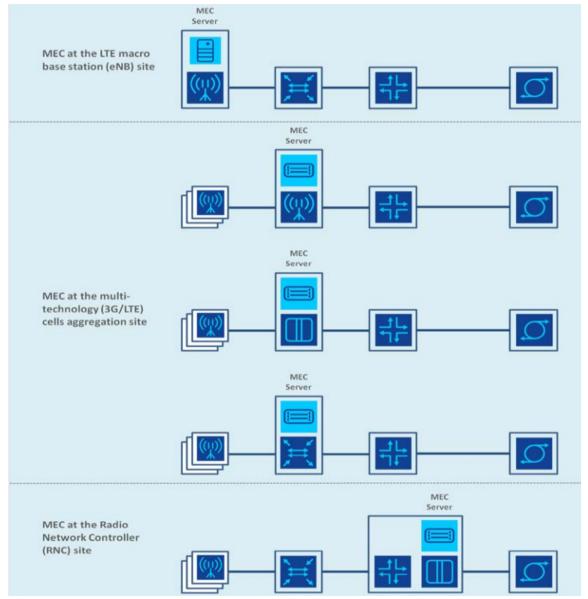
NFV enables MEC: Mobile Edge Computing

- MEC provides IT and cloud-computing capabilities within the RAN in close proximity to mobile subscribers to accelerate content, services and applications so increasing responsiveness from the edge.
- Standardization bodies: ETSI, 3GPP, ITU-T
- RAN edge offers a service environment with ultra-low latency and high bandwidth as well as direct access to real-time radio network information (subscriber location, cell load, etc.) useful for applications and services to offer context-related services
- Operators can open the radio network edge to third-party partners
- Proximity, context, agility and speed can create value and opportunities for mobile operators, service and content providers, Over the Top (OTT) players and Independent Software Vendors (ISVs)
- Source: https://portal.etsi.org/Portals/0/TBpages/MEC/Docs/Mobile-edge-Computing-Introductory Technical White Paper V1%2018-09-14.pdf

MEC:

Mobile Edge Computing

Deployment scenarios of the Mobile-edge Computing server



ACM SIGCOMM Tutorial | 2016-08-22 | Page 96
 Source: https://portal.etsi.org/Portals/0/TBpages/MEC/Docs/Mobile-edge Computing - Introductory Technical White Paper V1%2018-09-14.pdf

MEC Server

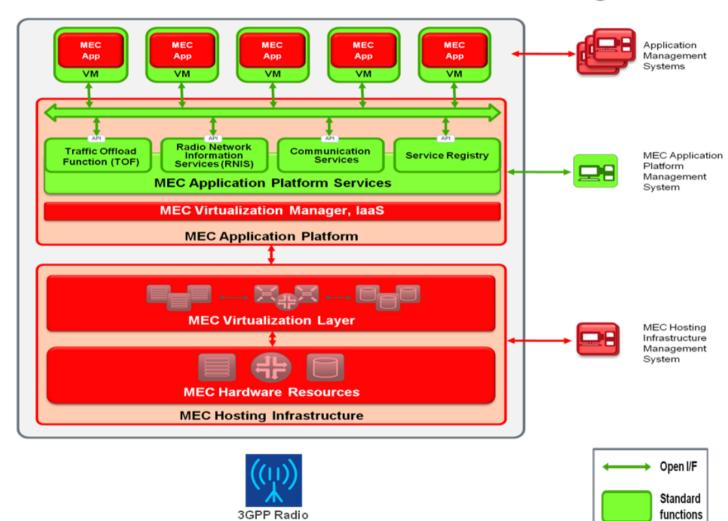
Network Element

Management

MEC:

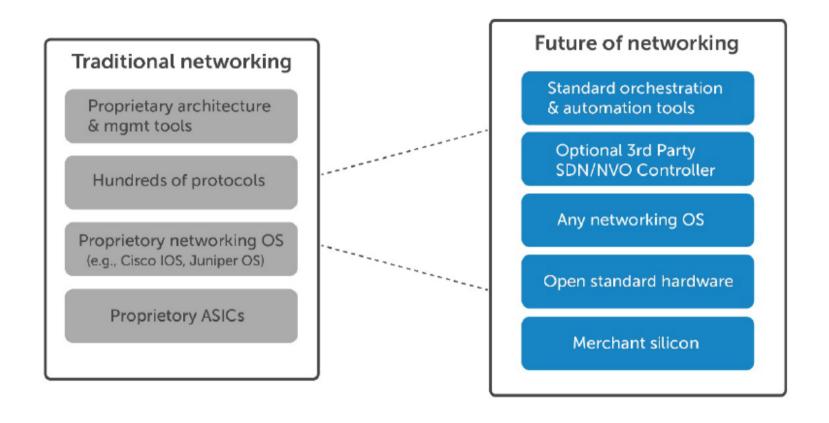
Mobile Edge Computing

MEC server platform overvie



ACM SIGCOMM Tutorial | 2016-08-22 | Page 97 Source: https://portal.etsi.org/Portals/0/TBpages/MEC/Docs/Mobile-edge Computing - Introductory Technical White Paper V1%2018-09-14.pdf

Summing Up



Source: Software-defined networking (SDN): a Dell point of view - A Dell White Paper (2015)

ACM SIGCOMM Tutorial | 2016-08-22 | Page 98

Enabling Technologies & Open Source Efforts

Enabling Technologies

container-based

- Virtualization & Minimalistic OS
 - Docker, ClickOS, Unikernel
- Improving Linux I/O & x86 for packet processing
 - DPDK, Netmap, VALE, Linux NAPI
- Programmable virtual switches / bridges
 - Open vSwitch, P4, Open Networking Linux

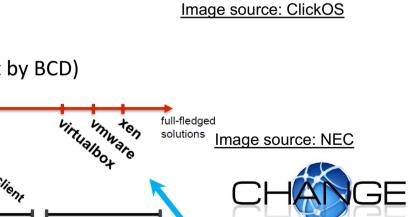
minimalistic OSes

Example start-ups

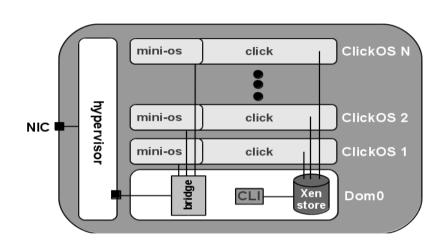
ACM SIGCOMM Tutoria

minimalistic solutions

LineRate Systems, 6WIND, Midonet, Vyatta (bought by BCD)



full virtualization



Enabling Technologies: Open Source

Why Open Source in Networking?

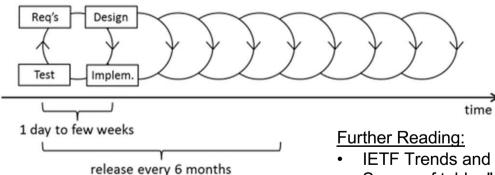
- Higher reliability, more flexibility
- Faster, lower cost, and higher quality development
- Collaborative decisions about new features and roadmaps
- A common environment for users and app developers
- Ability for users to focus resources on differentiating development
- Opportunity to drive open standards

Bottom Line: The open source model significantly accelerates consensus, delivering high performing, peer-reviewed code that forms a basis for an ecosystem of solutions.

Source: Open Source in a Closed Network – Prodip Sen (OPNFV Summit 2015)

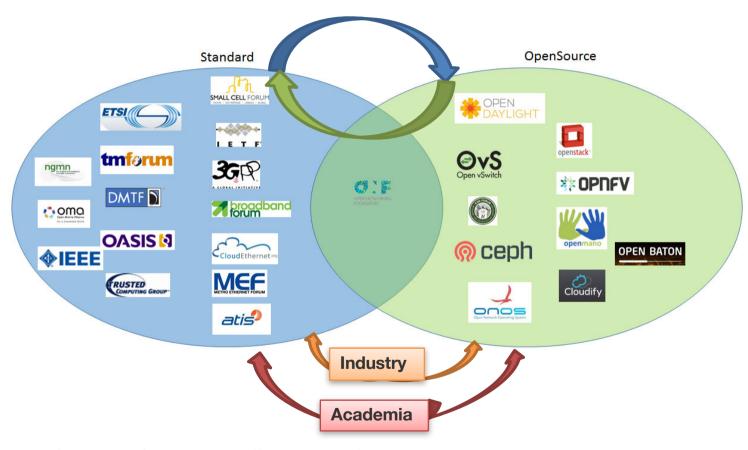
SDN/NFV & Open Source: Evolving and accelerating the path of standardization

	Present with SDN	Past / Traditional						
Drivers	Customer	Vendors						
Goals	Address user / operator needs (customization)	Enable multiple solutions (interoperability)						
Deliverables	Implementations & PoCs	Documents						
Quantity of Standards	Less	More						
Timetable	Few years	Many years						
Validation	PoCs integral to the process	Products and deployments after release						
Point of Control	Contribution to FLOSS codebase.	Seat at standards committee table						
	Ability to understand codebase							
Parties Involved	Anyone with domain expertise and coding ability	Vendors who can afford membership fees.						
		Experts and academics with high standing in their fields						



- IETF Trends and Observations draft-arkko-jetf-trends-and-observations-00
- Source of table: "When Open Source Meets Network Control Planes." In IEEE Computer (Special Issue on Software-Defined Networking), vol.47, no.11, pp.46,54, Nov. 2014.
- Source of figure: A. Manzalini et al., "Towards 5G Software-Defined Ecosystems"

Standard / Open Source Organizations



Source: SDN IEEE Outreach, http://sdn.ieee.org/outreach

Foundations

Target Collaboration

- Neutral and non-competing
- Legal framework for licensing, copyright, intellectual property management







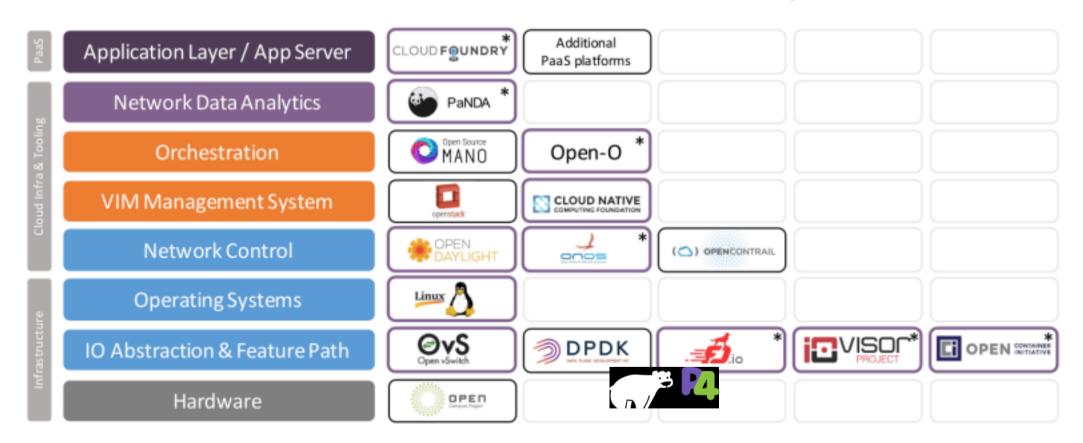






"Companies feel they can collaborate on an open source project through an independent, not-for-profit entity that they trust - this is incredibly important to them," --Allison Randa (Board President of Open Source Initiative)

Open Source Building Blocks 2015 – 2016: Several New Projects



ACM SIGCOMM Tutorial | 2016-08-22 | Page 105 Source: The Open Source NFV Eco-system and OPNFV's Role Therein – Frank Brockners (OPNFV Summit 2016)

Open Sourrce SDN Projects (2014 snapshot)

SDN Realm	Open Source Project Name					
Benchmarking	Cbench (GPLv2, C/Perl/UNIX shell), OFLOPS (GPL, C)					
Debugging / Testing / Simulation	ndb, OFRewind, STS (Apache, Python), OFDissector (BSD, C), liboftrace (BSD, C), OFTest (BSD, Python), Mininet (BSD, Python), fs-sdn (GPL, Python), ns-3 (GPL, C++), TestON (GPL, Python)					
Verification	Hassel (GPL, Python), NetPlumber (GPL, Python), NICE (BSD, Python), FlowChecker, OFTEN					
Control & Management Apps	Topology discovery (GPL, many), HostTracker (GPL, many), Plug-n-Serve (BSD, C++), Aster*x (BSD, C++), FlowScale (Apache, Java), SNAC (Python/C++), Odin (Apache, Python), PANE (BSD, Haskell), FRESCO (GPL, Python/C++), OSCARS (BSD, Java), RouteFlow (Apache, Python/C/C++/Java), Open DayLight (Eclipse, Java)					
Programming Abstractions / Compilers / Isolation	FatTire, Flog, FML, Frenetic (GPL, OCaml), HFT, NetCore, Nettle, Procera, Pyretic (BSD, Python), Maple (Pyhon), FlowN, LibNetVirt (GPL, C/Python), OpenStack Neutron (Apache, Python)					
Controller Platforms	NOX (GPL, C++); POX (GPL, Python); Maestro (LGPL, Java); Beacon (GPL, Java); Floodlight (Apache, Java); Ryu (Apache, Python); Trema (GPL, C/Ruby); FlowER (MIT, Erlang); NodeFlow (GPL, javascript); Mul (GPL, C); OpenDaylight (Eclipse, Java); ONOS (Java); OpenContrail (Apache, C++ / Python)					
Data Plane Virtualization	FlowVisor (BSD, Java), PortVirt (BSD, C), Expedient (Apache, Python), OpenVirteX (Apache, Java)					
OpenFlow Protocol Libraries / Southbound Driver	OFLib-Node (BSD, JavaScript), OpenFlowJ (BSD, Java), OpenFaucet (Apache, Python), Pylib-OpenFlow (BSD, Python/C+freeflow (Apache, C/C++), xDPd/ROFL (MPL, C/C++), libfluid (Apache, C/C++)					
Data Plane Implementations	NetFPGA (BSD, C/Verilog), Open vSwitch (Apache, C), Reference design (BSD, C), ofsoftswitch13 (BSD, C), OpenWRT/Pantou (GPL, C), Switch Light (Eclipse, C), Indigo Virtual Switch (Eclipse, C), LINC-Switch (Apache, Erlang)					

Source: "When Open Source Meets Network Control Planes." In IEEE Computer (Special Issue on Software-Defined Networking). 2014.

NEW: PLEASE SEE AND CONTRIBUTE TO

- https://goo.gl/XCGDGS
- http://bit.do/oss-sdn-nfv



Name	Organization	Main Contribution / Focus (SHORT DESC: 160 char.)	Link-Project	Link-Repo-Code	OpenSource-License	Mailing-list	Link-to-Mailing-List (Subscribe)	Link-to-Mailing-List (Archives)	Status	Link-to-Scientific-Pape	Remarks	Tag 1	Tag 2	Tag 3	Tag 4
OpenvSwitch (OVS)	Linux Foundation	Production quality, multilayer virtual switch designed to enable massive network automation through programmatic extension, while still supporting standard management interfaces and protocols.	http://openvewiich.org/	https://github.com/openvewisch/ove/	Apache 2.0	announce@open/exilch.org	http://mail.openvewiich.org/mailma	http://openvevilch.org/pipernall	Active			SW Dataplane / Switch	vSwitch	Cloud	Network Programmability
OpenSwitch (OSP)	Linux Foundation	OpenSwitch is a network operating system for disaggregated switches that are built around OCP complaint handware and that utilizes the ONE; boot bader to install and uninerall network operating systems.	http://www.apenewitch.net/	https://archive.openswitch.net/artifacia	Apache 2:0	opa-dev@lists.openswitch.net	http://liets.openswitch.net/cgi-bin/e	http://kels.openswitch.netbipern	Active		good project governance	SW Dataplane / Switch	HW Dataplane / Switch		Network Programmability
Indigo	Big Switch Networks	The Indigo agent includes core libraries aimed at enabling support for OpenFlow on physical and hypervisor switches.	http://www.projectfoodlight.org/indigal/https://github.com/foodlight/indiga		Edipse Public License version 1	indigo-emounce@openflowhub.	http://groups.google.com/s/openfic	https://groups.google.com/s/spe	Active			SW Dataplane / Switch	HIV Dataplane / Switch	SDK, APIs, Libraries	SON Dataplane Agent
Ofsofswitch13	CPuD	Research-friendly User-space OpenFlow 1.x software switch forwise from Stanford's original reference switch design used for protetying and experimentation. Imagested into Mininet. Used for open source implementation of new OpenFlow features by ONF members.		https://github.com/CPqD/ofsoftswitch1	RSD Icense	coenfore-decuse Cocenformal	https://mailman.etanford.edu/mailm	https://mailman.stanbert.edulma	Active			SW Dataplane / Switch			
	FlowForwarding	LINC is a pure OpenFlow software switch written in Erlang		https://github.com/FlowForwarding/LIN		inc-dev@flowlorwarding.org	http://groups.google.com/s/flowfor					SW Dataplane / Switch			
Protocal Oblivious Forwarding (POF)	Huzeni	SDN southbound protocol designed for high flexibility.	http://www.poforwarding.org		BSD Icense	palarwarding@huawei.com			Active			SW Dataplane / Switch	HW Dataciane / Switch	SDK, APIs, Libraries	DSL - Domain Specific L
	NTT		https://agopus.github.io/		Apache 2.0		https://intexeuroelorge.net/listellet	Microfes arcelorus, nellmalarchis	Active	https://www.usenic.org/si	desidefault/Nesions2014-		rin Laupan ramen	auri, ro in unames	
Berkeley Extensible Software Switch (BES	Redular Habrands	Modular framework for high-performance software switches allowing to configure custom packet processing datapath by compasing small "modules".	https://github.com/vietSysthess/wiki	https://github.com/NetSyabeas	BSDv3				4-6-			SW Dataplane / Switch	FOW ADD I hander	DSL - Domain Specific L	****
ClickOS	NEC CHICAGO	A minimalistic, tallor-made, virtualized operating system to run Click-based middlebows.	http://cng.neclab.eu/dicioe/		RSDv3	models (States or matter) man de	https://istserv.netlab.nec.de/mailm	Mineral Selection markets may disclose	Action		ystem/lies/conference/re		auri, nris, cutaries		
	MIT, UCLA, and others	The Click modular router: fast modular packet processing and analysis	http://www.nead.ca.uda.edu/click/		MIT	dick@librolat.com		http://ibrelet.com/browser/cick				SW Dataplane / Switch	SOW Altho I bender	DSL - Domain Specific L	200,000
	Snatb		https://enable.co/		Apache 2.0		https://groups.google.com/forum/F					SW Dataplane / Switch		and annual species to	
	GW, UCR	High performance NFV platform for running service chains through Docker NFs	http://edniv.github.ip/on/m/		Apacre 2.0	Smead@ge.edu			Active	hitedlands as on a cit	Limensile and It is	SW Dataplane / Switch			
Openierio	um, uux	regin personnance no v plantern no narreng service chains tribugh Licolar No-E	representative and the second	reprogramme and any opening to	Mac.	manner (Startegy			ACTIVE	THE PERSON AND ADDRESS OF	- emeracopalperario-rea	any Lianapiante / Switch	UPUR.		
Open Network Install Environment (ONIE)	Ones Comunic Project	Open Compute Project open acuros initiative contributed by Cumulus Networks that defines an open finatal environment for bare metal network switches.	http://orie.opencompute.org/	https://github.com/spencompuleprojec	GNILGEL v2	manners de orientidate commo	http://lieta.opencompute.org/mailm	http://ikata.common.com/sta.co	Artin			HW Dataplane / Switch	STW ADIa I bewise		
	Open Compute Project	Linux distribution for "bare metal" switches, that is, network forwarding devices built from commodity components.	http://pennetinus.org/	https://github.com/OpenComputeProje			https://groups.google.com/forum/W					HW Dataplane / Switch			
open manage cases (or re)	Open carry as ringed	Facebook's software stack (user-space applications, libraries, and utilities) for controlling		100	and are to	durant-un-fibridades			74310			rin canpair / anici	aurijio injunis		
Facebook Open Switching System (FBOS	Facebook	and managing network switches.	https://github.com/booksbook/book	https://github.com/bookeok/boss	BSD Scense				Active			HW Dataplane / Switch	SW Dataplane / Switch	SDK, APIs, Libraries	
OpenDaylight (ODL)	Linux Foundation	Production-ready open SDN platform containing features, protocols and plug-ins that can be integrated in a number of ways to deliver a broad set of SDN use cases.	https://www.apendaylight.org	https://github.com/openday/ight	Apache 2.0	controller-users@lists.open.dayli	https://intexpenday6ght.org/mailm	http://lists.apendaylight.org/pipe	malikorésiler-users/			SDN Controller	Virtualization Platform		
ONOS	Linux Foundation	Carrier-grade SDN network operating system designed for high availability, performance, scale-out	http://enoxproject.org/	https://github.com/spennetvorkinglab/	Apache 2:0	anas-discuss@anosproject.org	http://groups.google.com/s/onospr	https://groups.google.com/s/one	Active			SDN Controller	Virtualization Platform		
Floodlight	Big Switch Networks	Java-based OpenFlow 1.0 controller	http://www.projectfoodlight.org/foodl	https://github.com/floodight/floodight	Eclipse Public License version 1	Soodlight-dev@openSowhub.org	http://groups.google.com/a/openfic	https://groups.google.com/s/ope	nfowhub.org/forum/?from	groups#florum/floodlight-d	iev	SDN Controller	Virtualization Platform		
Ryu	NTT	Python-based OpenFlow 1.x controller	https://org.github.iohyu/	https://github.com/berg/ryu	Apache 2.0	ryu-devel@lists.sourceforge.net	https://iets.sourceforge.net/lists/list	http://eource/orge.net/mailarchiv	Active			SDN Controller	Virtualization Platform		
Trema	NEC	Trems is a full-stack framework for developing OpenFlow controllers in Ruby and C.	https://rema.github.io/rema/	https://github.com/trems/trems	GNU GPL v2	trema-dev@googlegroups.com	https://groups.google.com/group/tr	https://groups.google.com/forum	/#forum/trema-dev			SDN Controller	Virtualization Platform		
OpenMul	OpenMUL Foundation	Base SCHIOpenFlow controller platform written almost entirely in C (from scratch) and provides top performance in terms of flow handling (download rate and telency) as well as a very stable application development platform.	http://www.apenmul.org/apenmul-com	t https://github.com/openmul/openmul	GNU GPL v2			http://www.apenmul.org/discuss	ion.htm#/			SON Controller	Virtualization Platform		
POX	Stanford University	Python-based OpenFlow 1.8 controller used for research and experimentation	https://github.com/noxmpalpax	https://github.com/novrepalpax	Apache 2.0	pax-dev@lists.novrepa.org	http://iete.norrepo.org/letinlo.cg/p	http://kets.novrepo.org/pipermail	Closed			SDN Controller	Virtualization Platform		
Design	Stanford University	Java-based OpenFlow 1.0 controller	https://apenflow.stanford.edu/display/	git:/gitosis.stanford.edu/beacon.git	BSD License			https://apenflav.stanfard.edu/fa	Closed			SDN Controller			
SNAC	Stanford University	OpenFlow 1.0 controller with network access control application							Closed			SDN Controller	SDN Application	Security	
NOK	Stanford University	First OpenFlow 1.0 controller implementation	https://github.com/noxrepolnox	https://github.com/novrepoinax	Apache 2.0	max-dev@lists.novrepo.org	http://iete.norrepo.org/letinlo.cgih	http://ists.noinepo.org/pipermail	Closed			SDN Controller			
IRIS	ETRI	The Recursive SDN Openflow Controller by ETRI is an open source version of RES. IRIS is an Openflow-based SDN controller designed to solve acatability and availability lasues of SDN.	http://openiris.etri.re.kef	https://github.com/openinis/IPIS								SDN Controller			
Exalige	Exa Networks	ExaBGP provides a convenient way to implement Software Defined Networking by transforming BGP messages into thends plain text or JSON, which can then be easily handled by simple sortion or your SSO/SS.	hiter British comEvaluationish of	https://github.com/Exa-Networks/eusb	esn	euton-arralloportemora co	https://groups.google.com/forum/W	Mar Samue and comform	Artin			vRouter	BGP	WAN	NO.
		GoBGP is an open source BGP implementation designed from scratch for modern environment and implemented in a modern programming language, the Go Programming													-
GeBGP	NTT	Language.	http://derg.github.io/gobgp/	https://github.com/perg/gobgo	Apache 2.0	gatgp-devel@lists.saurceforge.r	https://idusourcelorge.net/lidulle/	https/wauroefarge.net/mailarchie	Active			vRouter	BGP	WAN	DIP
Bird	CZ.NIC		http://bird.network.co/		GNU GPL	bird-users@network.cz	http://bird.network.co/mailman/listin					vRouter	BGP	WAN	DEP
Quagga	OpenSourceRouting	IP Routing Stack	https://www.apensourcerouting.org/	http://git.eavannah.gnu.org/cg/fiquagg/	GNU GPL	quagga-users@liets.quagga.net	htps://ists.gusgga.ret/mailman/ist	http://ide.quagga.net/pipermail/	Active			vRouter	BGP	WAN	DIP
Celico	Tigera	Highly efficient vRouter in each compute node that leverages the existing Linux kernel forwarding engine without the need for vSwitches	https://www.projectcalico.org/		Apache 2.0		l http://lints.projectcalics.org/mailma					vRouter	BGP	WAN	
XORP	XORP	IP Routing Stack	http://www.xorp.org/		GNU GPL	когр-иметь@когр.огд	http://mailman.icel.berkeley.edu/m					vRouter	BGP	WAN	
	Akanda	Orchestration platform based on Layer 2 agnostic and interfaces with the OpenStack Neuron RS ST APIs featuring applications theopie management to monitor, configure, and manage 3rd party virtualized noutless, load balancers and fewaris.	http://www.do.io/		Agache 2.0	Recenstack-estate channel on h						NEVO	vPoster	CoenStack	

https://docs.google.com/spreadsheets/d/1NHI4MZZWVDpxF_Rs7OOSTUa_aHL2ACUVA_Ov-YQs1DA/edit#gid=0

A growing ecosystem...































5G Related Open Source













Experimentation







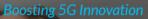




Testbeds

















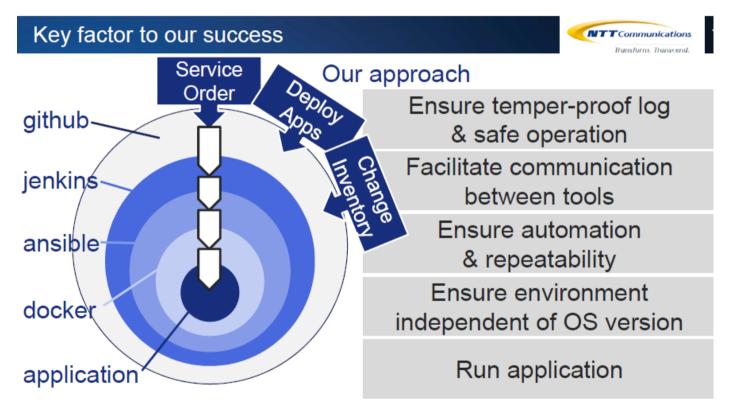
Source: http://openfederatedtestbed.org/

Continuous Integration Environment

- Continuous Integration tool chain speeds development and facilitates collaboration
 - Gerrit code review tool
 - GitHub code repository
 - Jenkins automated build tool
 - Maven code build
 - Ansible/Chef/Puppet code deployment tool
 - Docker/Vagrant deployment to Containers/VMs

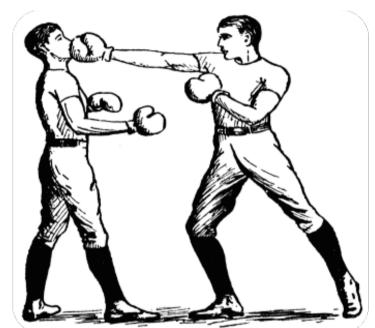
Source: Open Source Carrier Networking – Chris Donley (OPNFV Summit 2016)

Example: NTT (TM FORUM 2016)



Source: A Transformation From Legacy Operation to Agile Operation – Makoto Eguchi (TM ACM SIGCOMM Tutorial | 20TORUM 2016)

The Frontier of Networking



Existing

- CLIs
- Closed Source
- Vendor Lead
- Classic Network **Appliances**

New

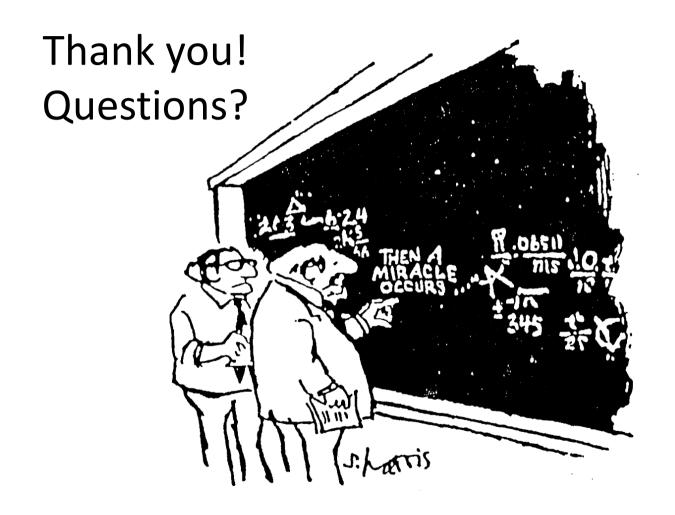
- **APIs**
- Open Source
- **Customer Lead**
- **Network Function** Virtualization (NFV)

Adapted from: Kyle Mestery, Next Generation Network Developer Skills

Some Takeaways

- Open source speeds up development and promotes interoperability
- Rapid prototypes are a great way to show SDN/NFV value, and now 5G too!
- Developer ramp-up time is challenging,
 - but once integrated, we can make rapid progress
- A common CI environment facilitates sharing between projects
- Cadence, short iteration cycles, fast feedback
- Some people are saying:
 - "open source is the new standardization"
 - "code is the coin of the realm"
- Stay tuned:
 - http://www.sdn-os-toolkits.org/
 - http://sdn.ieee.org/outreach





BACKUP

Perspective

Question for Operators Please give your opinion on which statements most accurately reflect your understanding of your company's approach to deploying Open Source software 100% 90% 80% 70% 60% 50% 40% 30% 20% 10% Source software in our networks Our network is hosted in an Open Source enabled Open Source Open Source opportunity to officantly decrease Operating costs software creates the vendors to mitigate with Open Source We do not use Open products We deploy Open oftware will play an represents an Open Source and we avoid using it We rely on our any risks associated Source in our mportant role in our and we allow it in our code and we do not. Disagree

Source: Survey Results: Bridging the Gap Between Open Standards and Open Source - Elizabeth Rose (OPNFV Summit 2016)

Challenges: Closed vs. Open

Carrier Network Software	Open Source Softwa	
Proprietary with custom features for specific users	Open, generic enough to be used in wide range of applications	
Features : Roadmaps agreed between vendors & users before releases	Features emerge based on community engagement. Release feature list is after the fact.	
Upgrades: Releases are well planned but slow to implement	Upgrades: Frequent upgrades are expected requiring a Continuous Integration environment	
Solutions: Typically vertically integrated	Solutions: Flexible with many possibilities for horizontal and vertical integration	
Guarantees: System / solution vendor responsible.	Guarantees: User or separate system integrator responsible.	
Carrier Mindset: No room for failure	Open Source: Fail fast, and fix faster	
Carrier Mindset: One throat to choke	Open Source: No throat to choke	

Source: Open Source in a Closed Network – Prodip Sen (OPNFV Summit 2015)

Open Source Road



Source: The NFV Revolution Must Be Open – Dave Neary (OPNFV Summit 2016)

Approach for Contributions

- Open Source Projects need more than code!
 - Documentation
 - -QA
 - Infrastructure
 - Blogs
 - IRC

- Evaluate how your project idea can fit with the existing project
 - Does it overlap?
 - Does it provide extra value?
 - Can something be abstracted?
 - Is the community interested?
 - Does it fit the communities goals?
 - Do you have a plan for testing, documentation and support?

Project Evolution: Examples

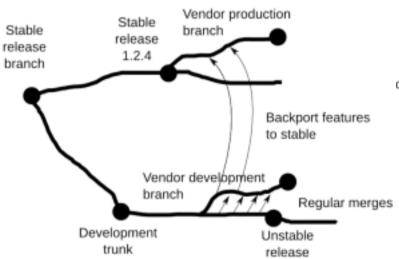
Project	Started	Evolved To	Foundation
Open vSwitch	Small core group of contributors	Taking on a new project (OVN), growing slowly but surely	No foundation*
openstack.	Rough consensus of many disparate groups into one codebase	Focused group of repositories	OpenStack Foundation
** OPEN DAYLIGHT	Many separate groups under the same large umbrella	More and more projects and repositories	Linux Foundation

Source: Upstream Open Source Networking Development: The Good, The Bad, and the Ugly – Kyle Mestery, ACM SIGCOMM Tutorial Justin, Pettit, Russell Bryant (ONS 2016)

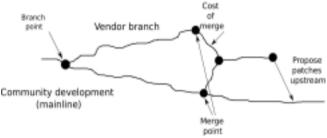
Building on Open Source Projects

Branching strategy

Ideal situation



This is what **upstream** does



Cost: Vendor work + cost of merge + "community overhead"

"Do NOT fall into the trap of adding more and more stuff to an out-of-tree project. It just makes it harder and harder to get it merged. There are many examples of this." -Andrew Morton

Source: Swimming Upstream – Dave Neary (OPNFV Summit 2016)

Open Ended Questions

- What are the inhibitors to adoption of Open Source Implementations?
 - Issues surrounding licensing (esp. GPL)
 - Industry understanding of Open Source licensing
 - Competitive issues and fragmentation
 - Security
 - Quality and robustness
 - Maintenance and support

Source: Survey Results: Bridging the Gap Between Open Standards and Open Source - Elizabeth Rose (OPNFV Summit 2016)