Next Generation Networking with SDN and NFV

Salvatore Costanzo
Department of Informatics and Telecommunications, University of Athens
{scostanzo@di.uoa.gr}

Outline

- **Networking Technology Trends:** Network Virtualization and Cloud Orchestration
  - Why virtualize the network?
  - What are the benefits for network operators and their customers?
- **The Need for Elastic Network Architectures**
  - Enabler solutions for elastic networks:
    - Management of network services through abstraction of lower level functionality using Software Defined Networking (SDN)
    - Virtualization of network node functions (NFV)
- **SDN/NFV Use Case Scenarios:**
  - LTE Mobile Network Virtualization: benefits, challenges and solutions
  - Cloud mobile Radio Access Network

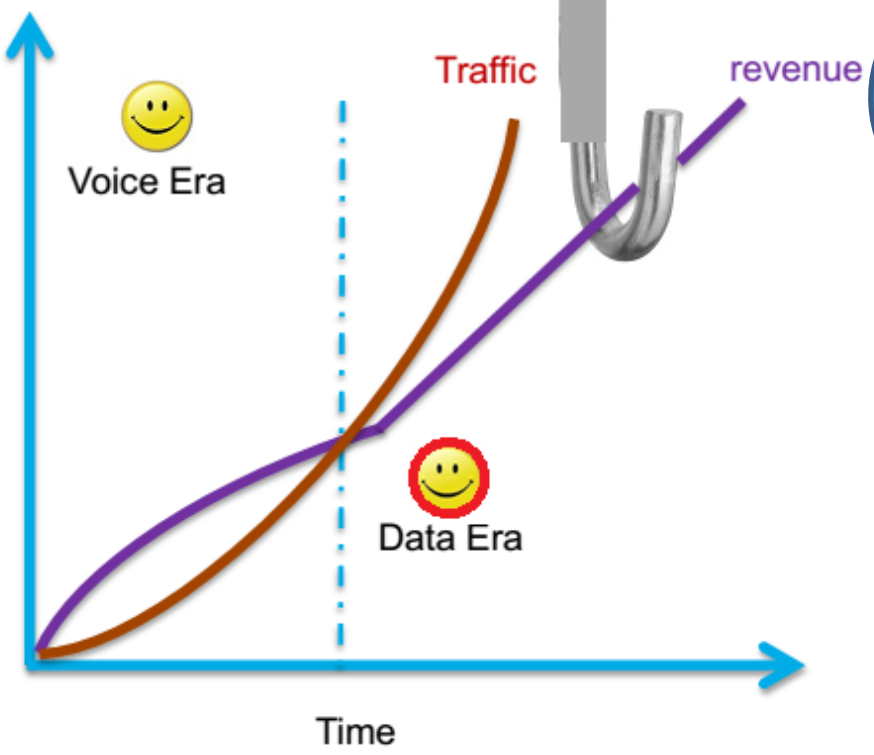
Challenges in networking

- Explosion of devices and traffic: huge capital investment
- Network operators face an increasing disparity between costs and revenues
- Complexity: large and increasing variety of proprietary network hardware appliances
- Lack of flexibility and agility: launching new services is difficult and takes too long

The need of elastic networks

Enablers?

- Network Virtualization
- Software Defined Networking
- Network Functions Virtualization
- Cloud orchestration



Network Virtualization

What is Virtualization?

vir·tu·al *adjective* \'vər-chə-wəl, -chəl; 'vərch-wəl\

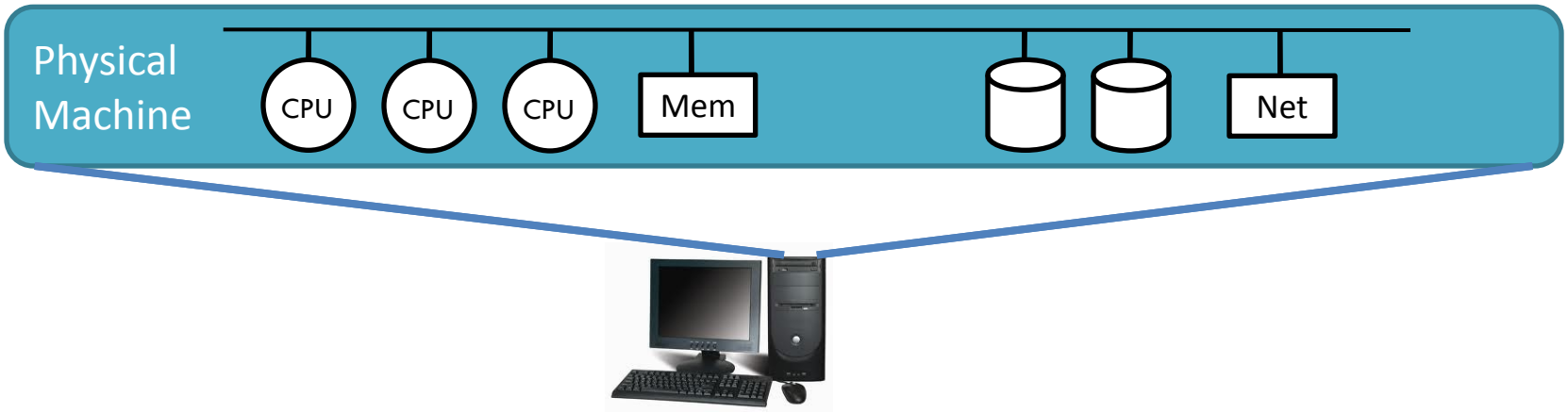
: very close to being something without actually being it

: existing or occurring on computers or on the Internet

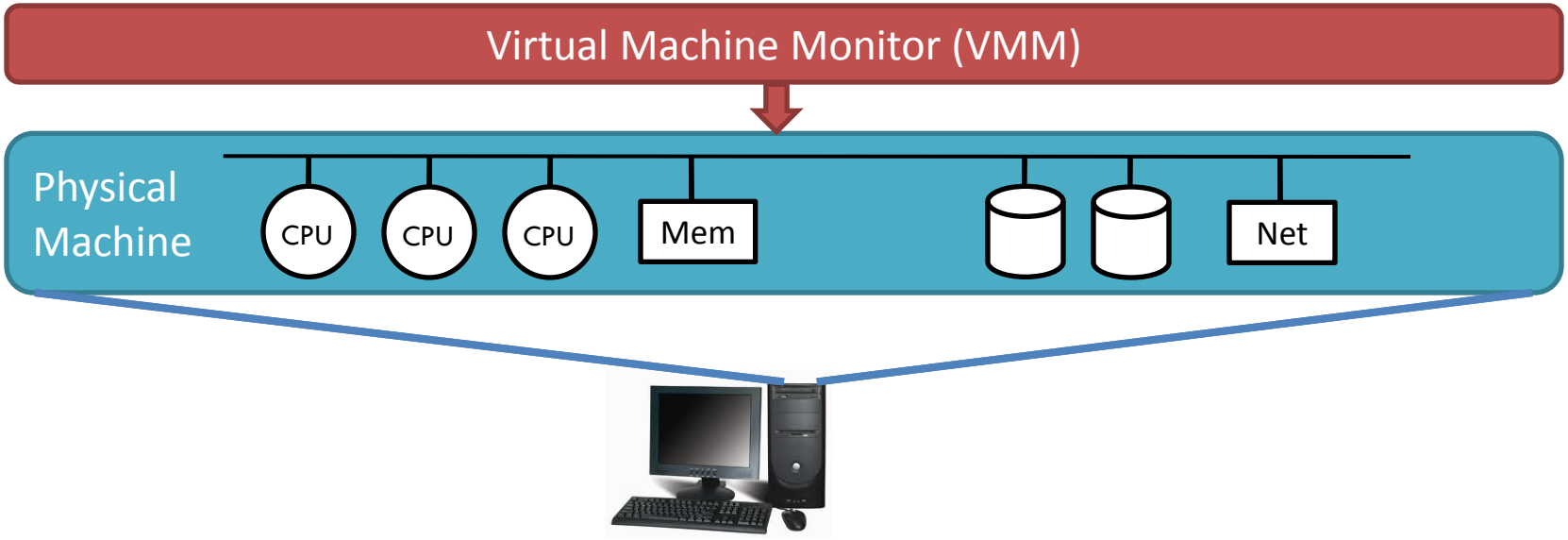
- Virtualization is the process of creating virtual versions of physical resources that emulate the same physical characteristics.

- Trend of Virtualized everything:
  - Virtual machines: VMware, Xen
  - Data-center virtualization
  - Network Virtualization

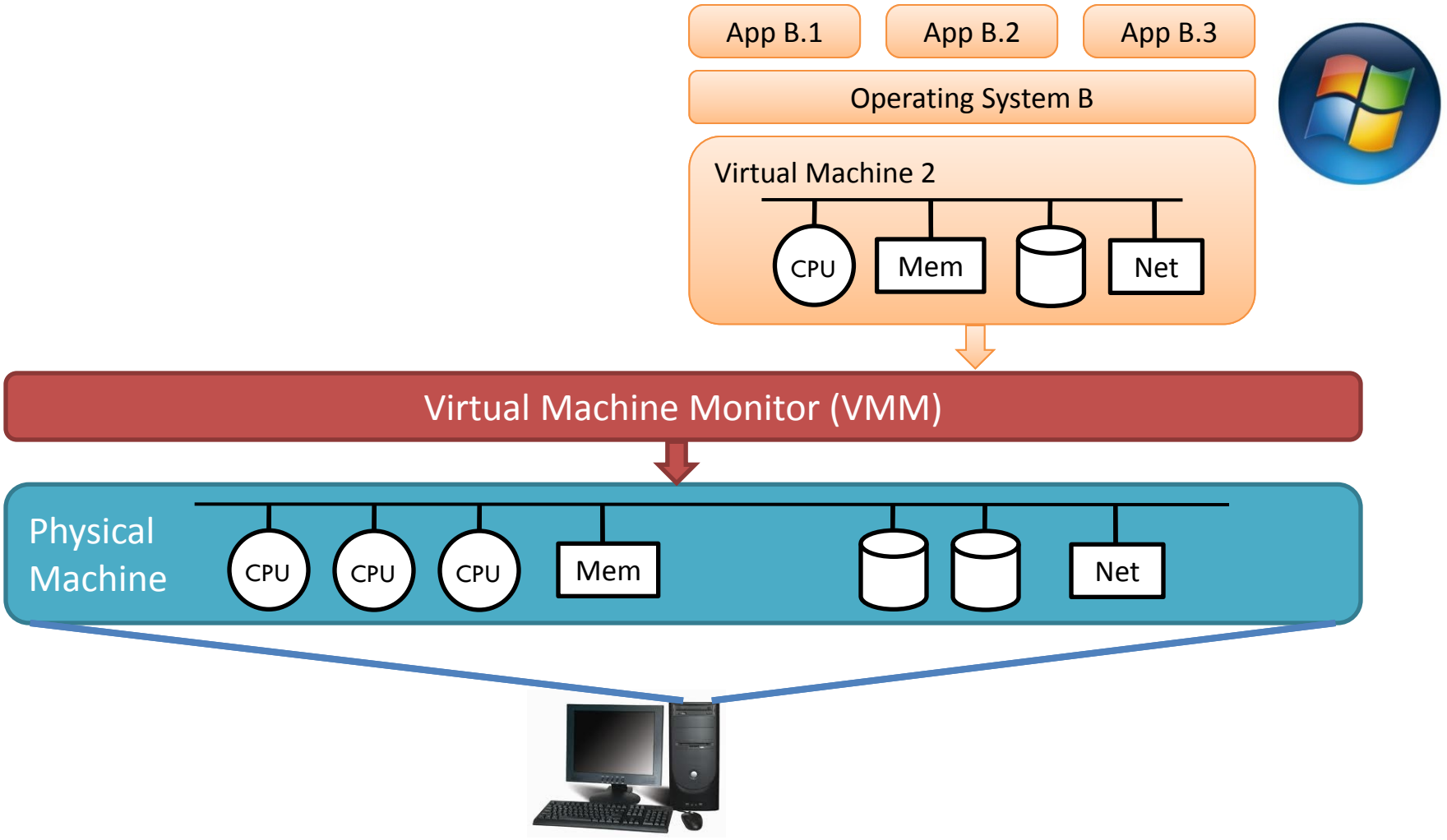
Example: Virtualization in computers: Virtual Machine



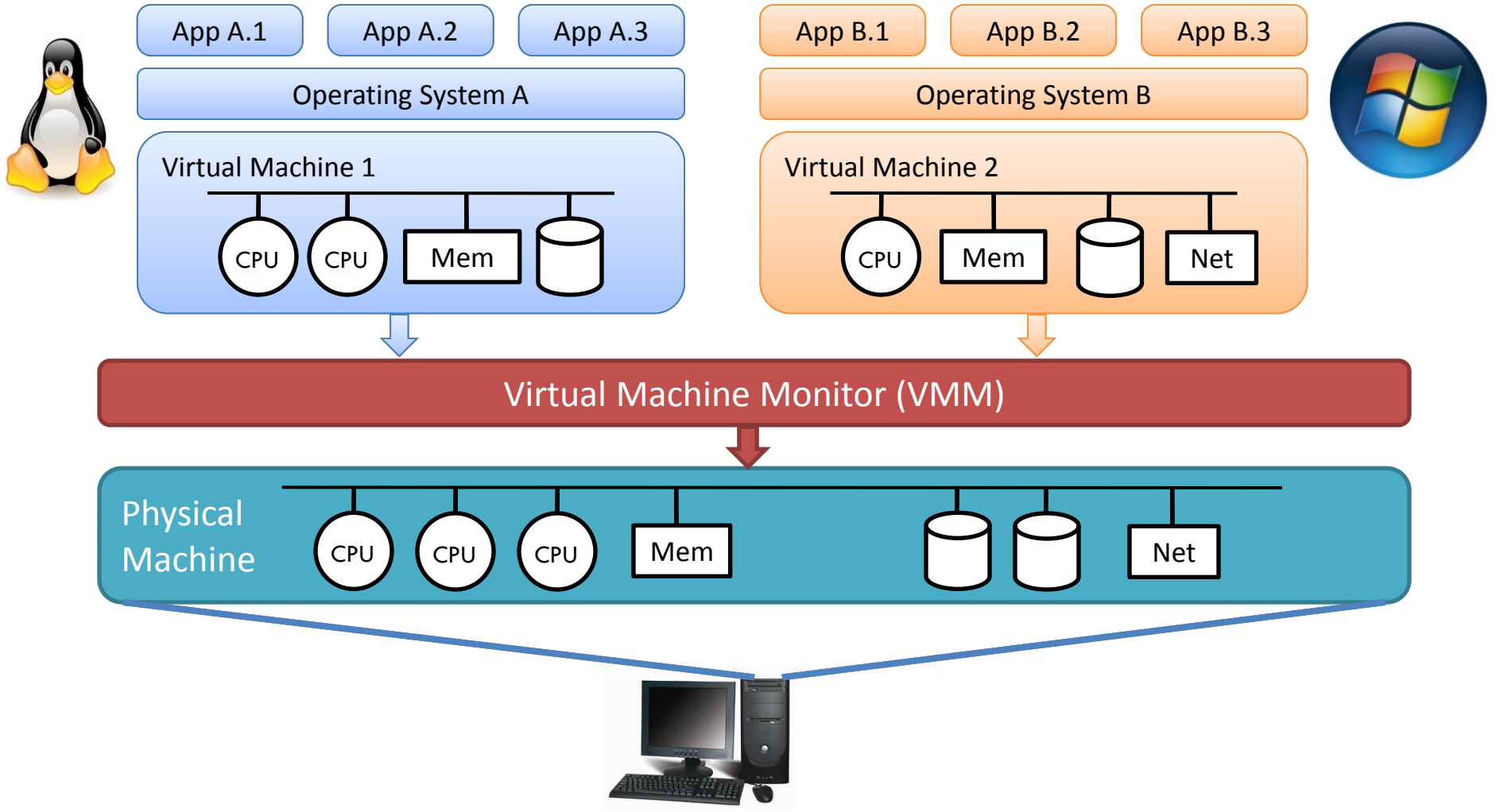
Example: Virtualization in computers: Virtual Machine



Example: Virtualization in computers: Virtual Machine



Example: Virtualization in computers: Virtual Machine

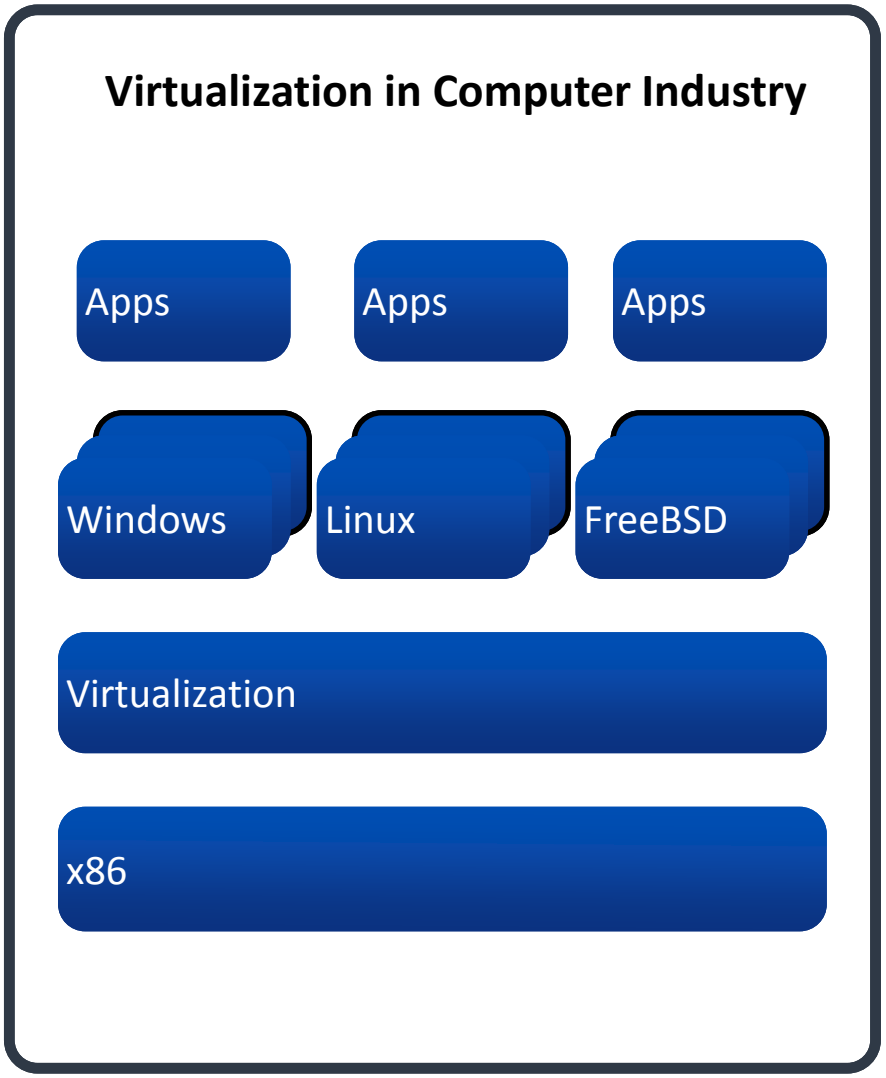


Why Virtualize?

- Virtualization adds flexibility, allows heterogeneity, and improves manageability of the computing infrastructure

- Virtualization allows resource sharing :
  - Reduced number of equipment devices
  - Higher availability
  - Reduced time needed for deployments using virtualized infrastructure
  - Lower cost of ownership
  - More resilient and simpler to manage

Computer and Network virtualization



Computer and Network virtualization

Virtualization in Computer Industry

Apps | Apps | Apps

Windows | Linux | FreeBSD

Virtualization

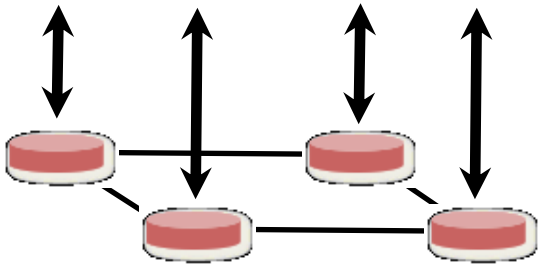
x86

Virtualization in Network Industry

Apps | Apps | Apps

Network OS | NOX | Beacon

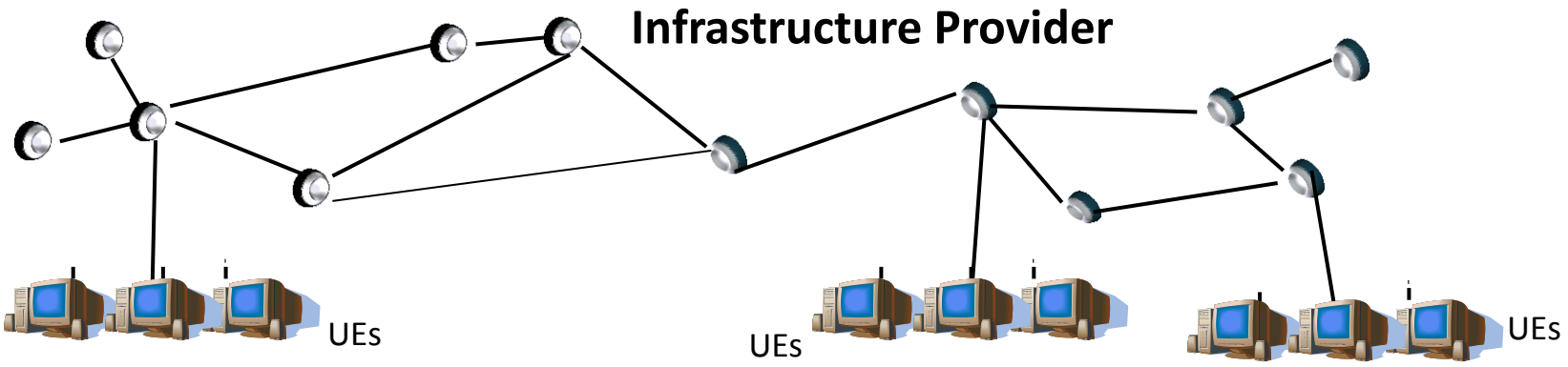
Virtualization



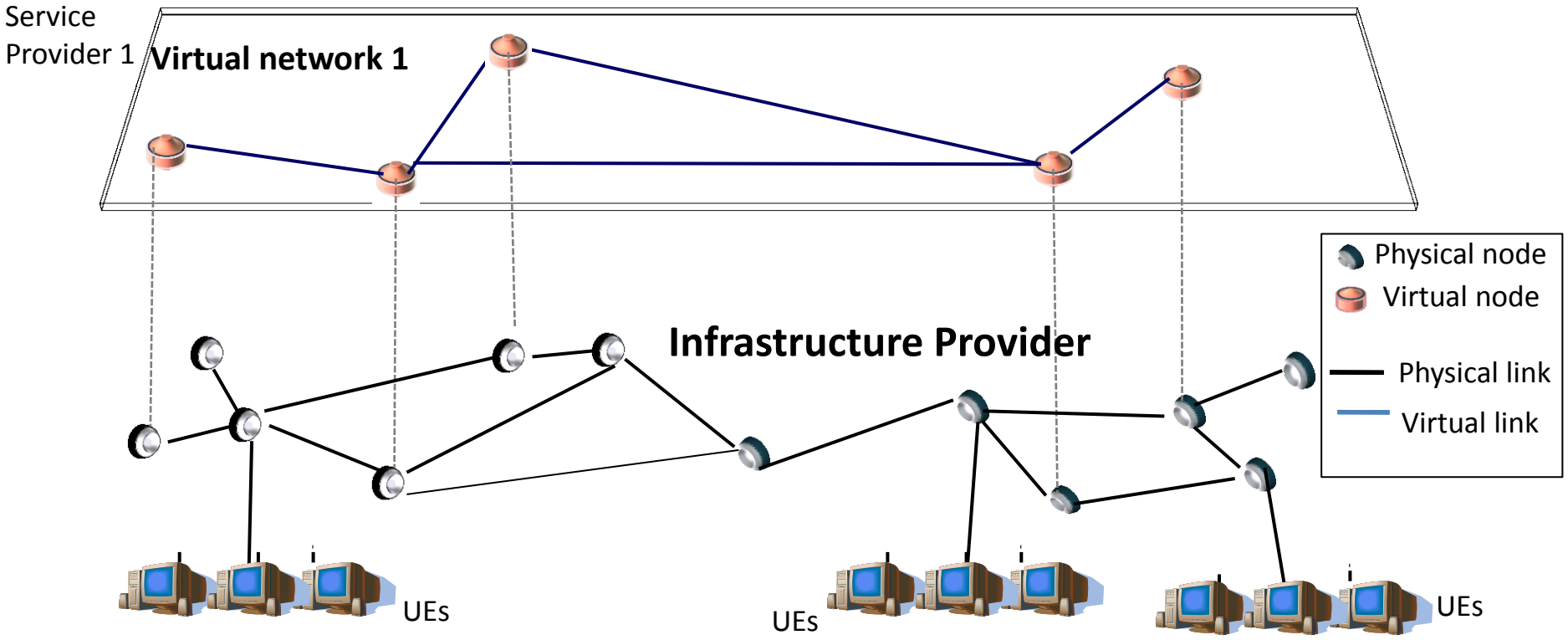
Virtual Networks: applications

- Overlay Networks
  - An overlay network is a computer network which is built on the top of another network. Nodes in the overlay can be thought of as being connected by virtual or logical links, each of which corresponds to a path, perhaps through many physical links, in the underlying network

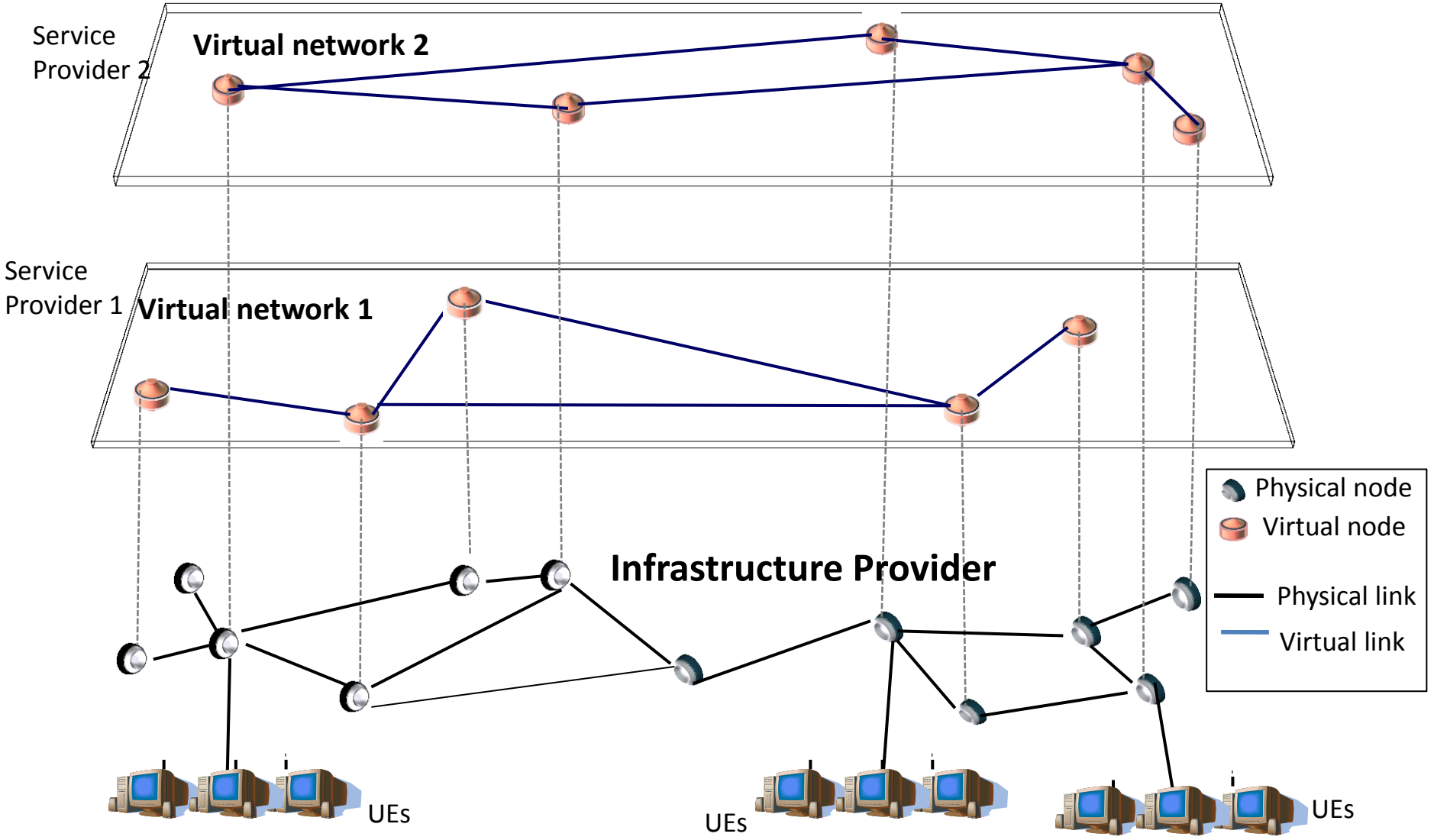
Network Virtualization Scenario



Network Virtualization Scenario

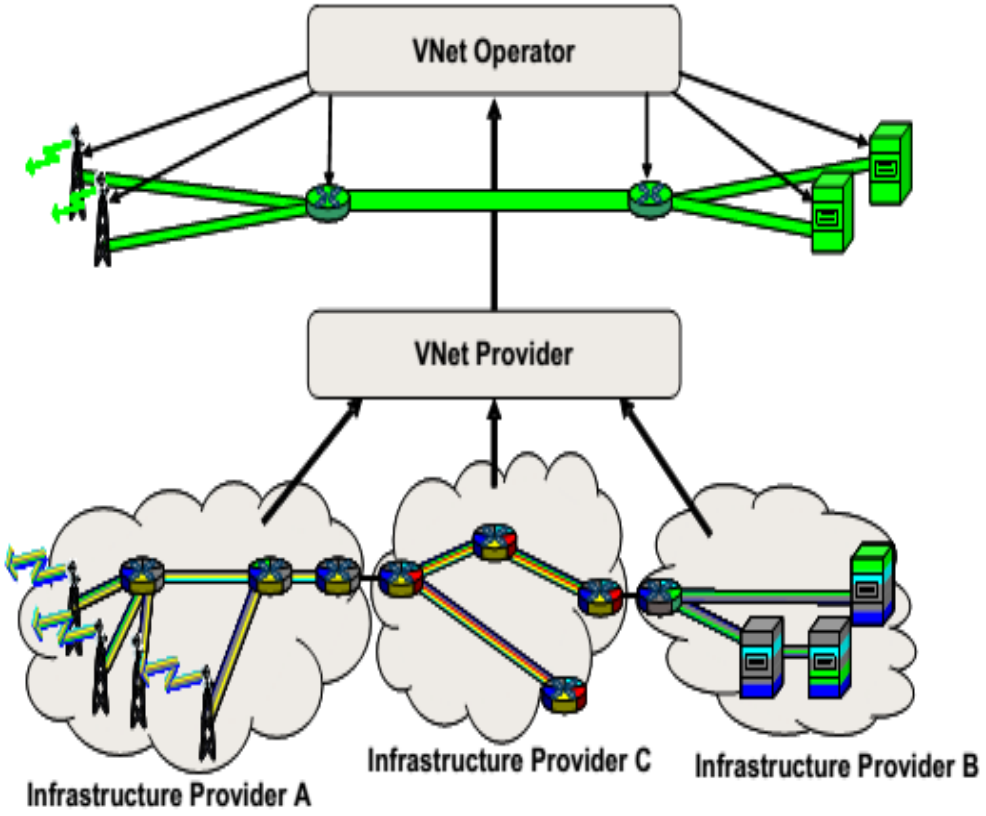


Network Virtualization Scenario



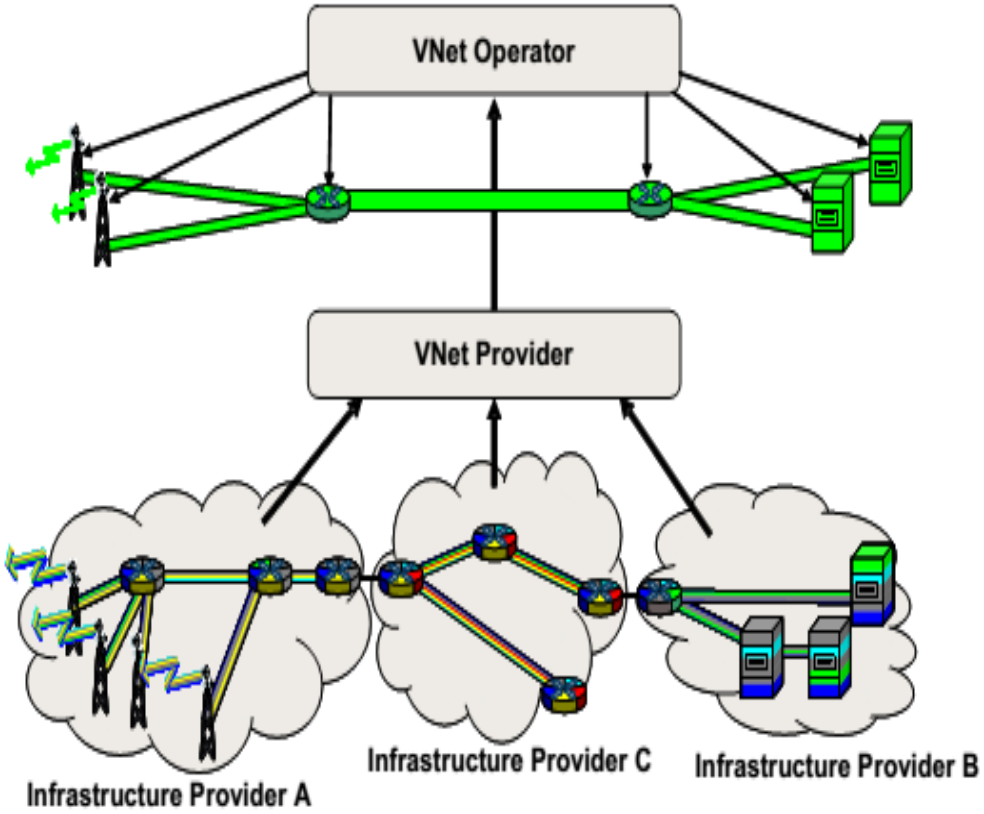
Network Virtualization: Business roles

- Network virtualization refers to the creation of a set of overlay architectures built on top of one or more existing physical infrastructures.



Network Virtualization: Business roles

- Network virtualization refers to the creation of a set of overlay architectures built on top of one or more existing physical infrastructures.



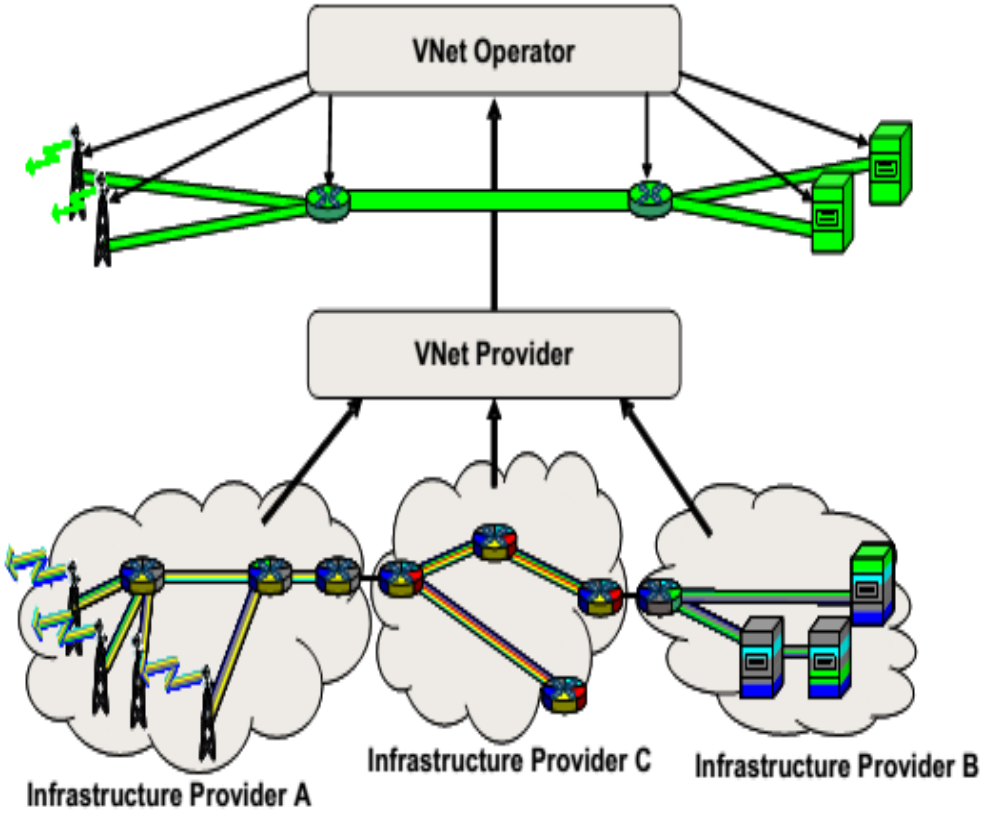
Infrastructure providers:
They own and manage the physical network devices and virtualize the physical resources

Network Virtualization: Business roles

- Network virtualization refers to the creation of a set of overlay architectures built on top of one or more existing physical infrastructures.

Virtual Network Providers :
who combine the virtual resources
in order to form Virtual Networks

Infrastructure providers:
They own and manage the
physical network devices
and virtualize the physical resources



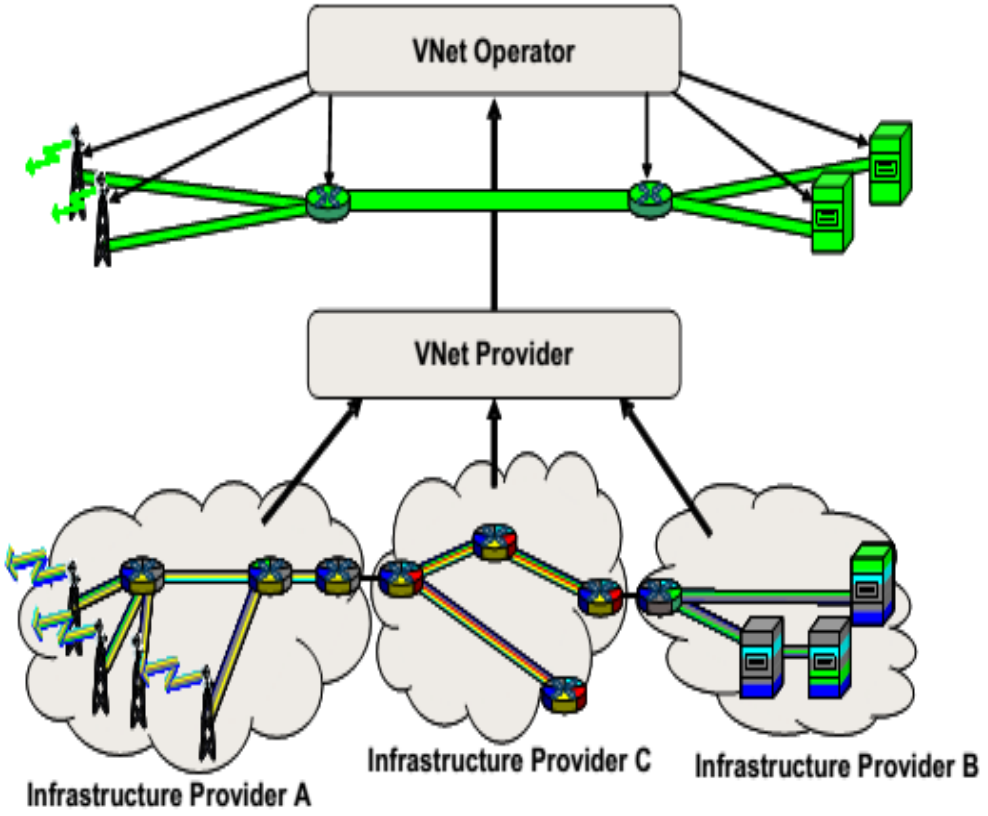
Network Virtualization: Business roles

- Network virtualization refers to the creation of a set of overlay architectures built on top of one or more existing physical infrastructures.

Virtual Network Operators :
who run the Virtual Network and
provide their services to the end-users.

Virtual Network Providers :
who combine the virtual resources
in order to form Virtual Networks

Infrastructure providers:
They own and manage the
physical network devices
and virtualize the physical resources



Network Virtualization for Mobile networks?

Motivations

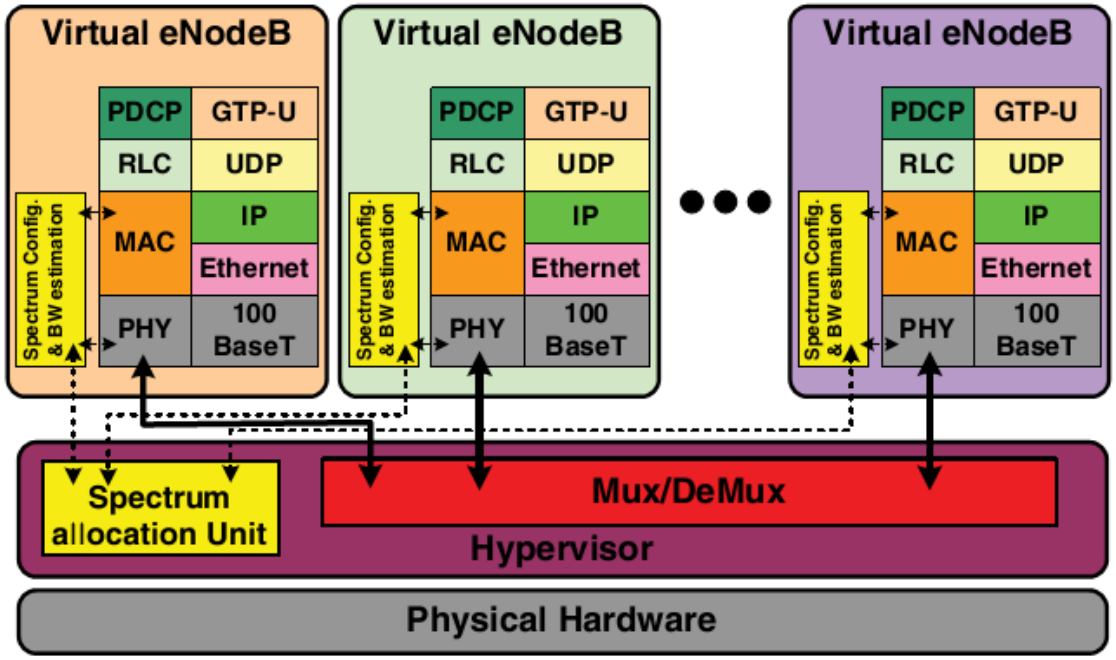
- The wireless resources of mobile networks are expensive and scarce. Network Virtualization will bring a more efficient utilization of the scarce wireless resources.

- Network Virtualization is a good solution for:
  - reducing the number of base stations (reduce energy usage)
  - reducing the overall investment capital required by mobile operators to setup their own infrastructure.
  - allowing smaller players to come into the market and provide new services to their customers using a virtual network without the need of built a network infrastructure

Use Case for LTE: Virtualization of base stations

Two layers of scheduling:

- one layer for splitting the spectrum between the different virtual operators
- and one layer for splitting the allocated spectrum among the users belonging to the same operator



The scheduling can be based on different criteria such as: bandwidth, data rate, power, interference, pre-defined contract, channel condition, traffic load or a combination of these

*As appears in Yasir Zaki,Liang Zhao,Carmelita Goerg ,Andreas Timm-Giel,"LTE mobile network virtualization Exploiting multiplexing and multi-user diversity gain", in Vehicular Technology Conference (VTC Spring), 2012 IEEE 75th

Enable rapid innovation in networking with Network Virtualization

Ossification in networking

- The growth and extension of the current Internet is marked by ossification .
- Network virtualization is viewed as one of the enablers to overcome this ossification since make it easier to introduce innovation in networking

Ossification in networking

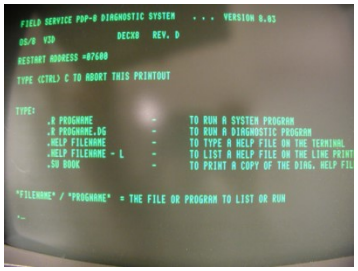
- The growth and extension of the current Internet is marked by ossification .
- Network virtualization is viewed as one of the enablers to overcome this ossification since make it easier to introduce innovation in networking

Let’s take a look to the evolution of networking…

Everything is evolving: machines, O.S., games.

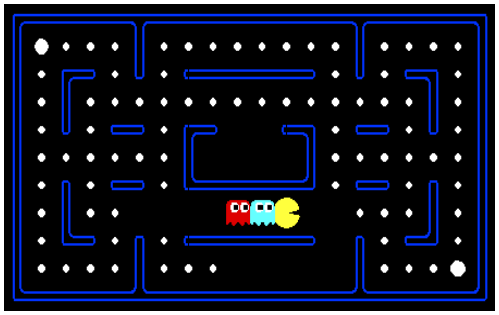
Fist laptop IBM 5100 (1973)

iPad



OS-8 (1971)

Macosx

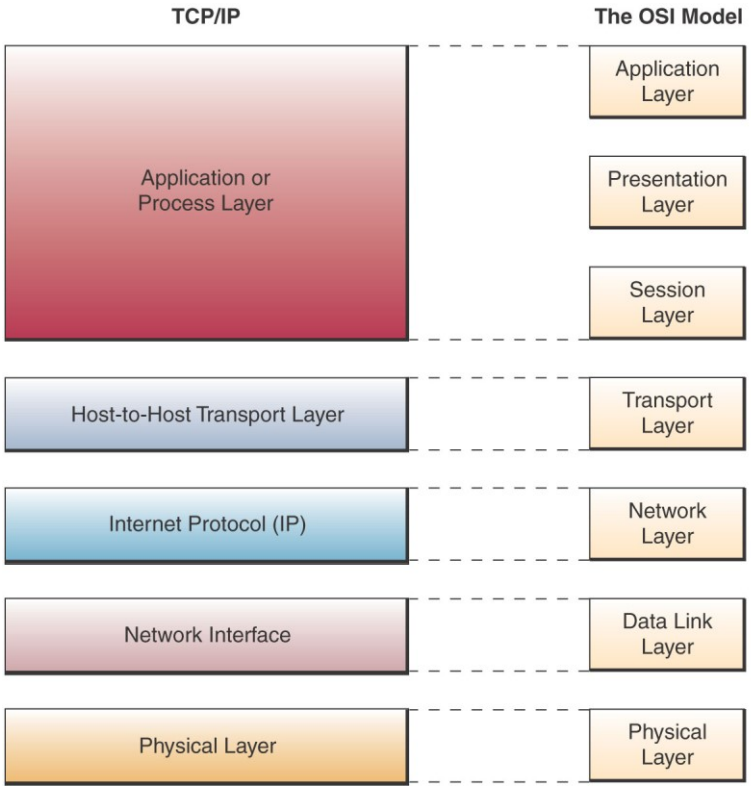


Pacman (1980)

Nintendo wii

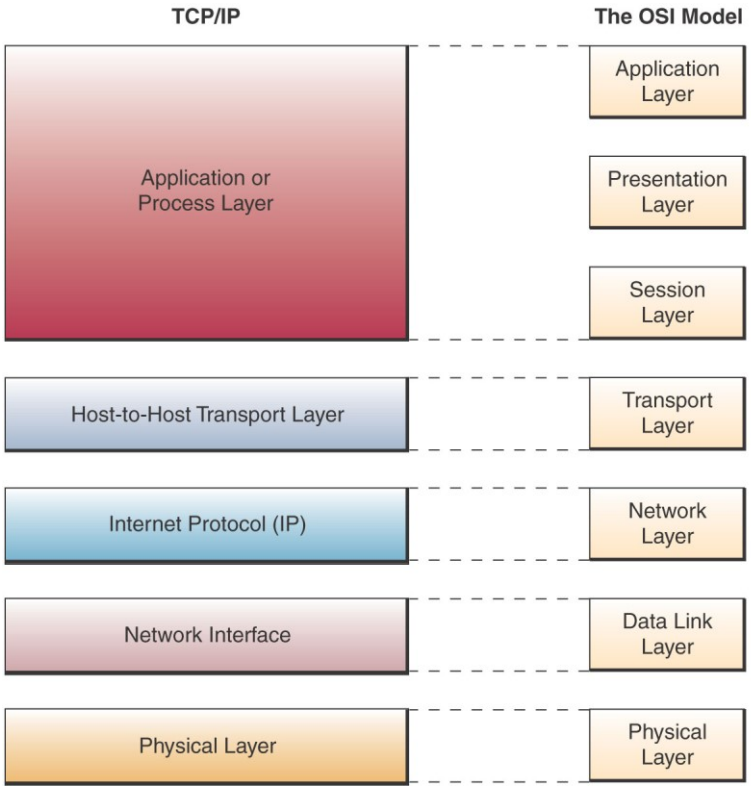
Everthing is evolving except the Network…

TCP/IP (70's)

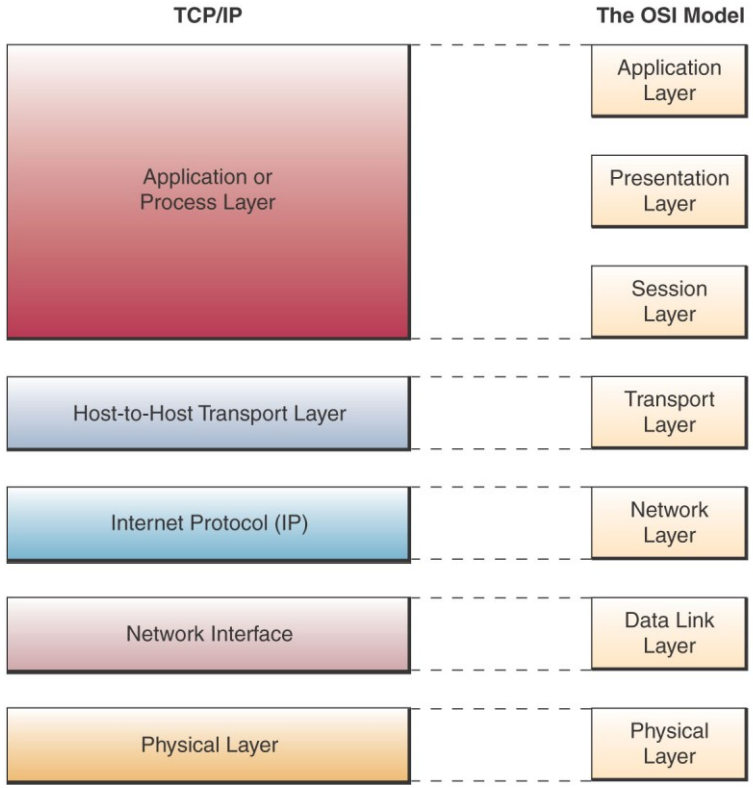


Everthing is evolving except the Network…

TCP/IP (70's)



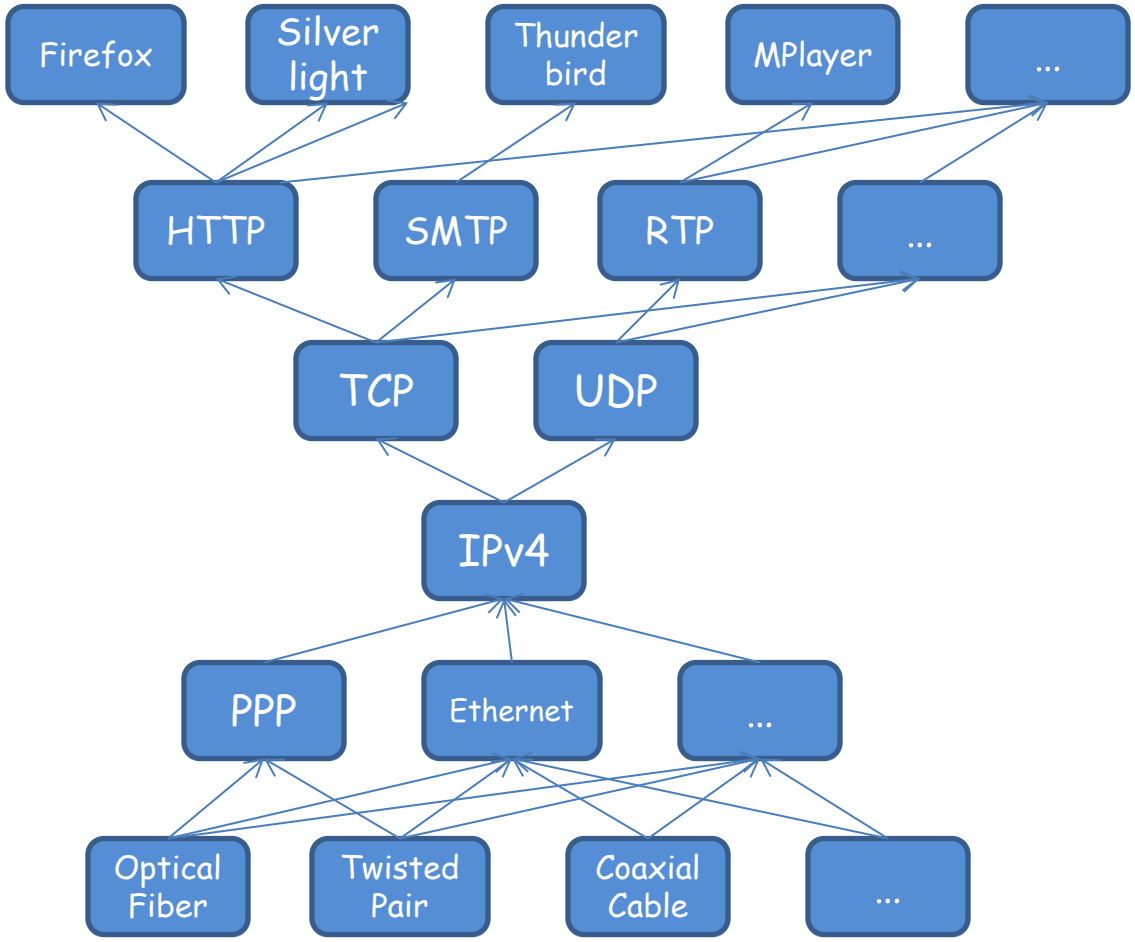
Today? Still .. TCP/IP



We are still using the same network architecture of 1970!

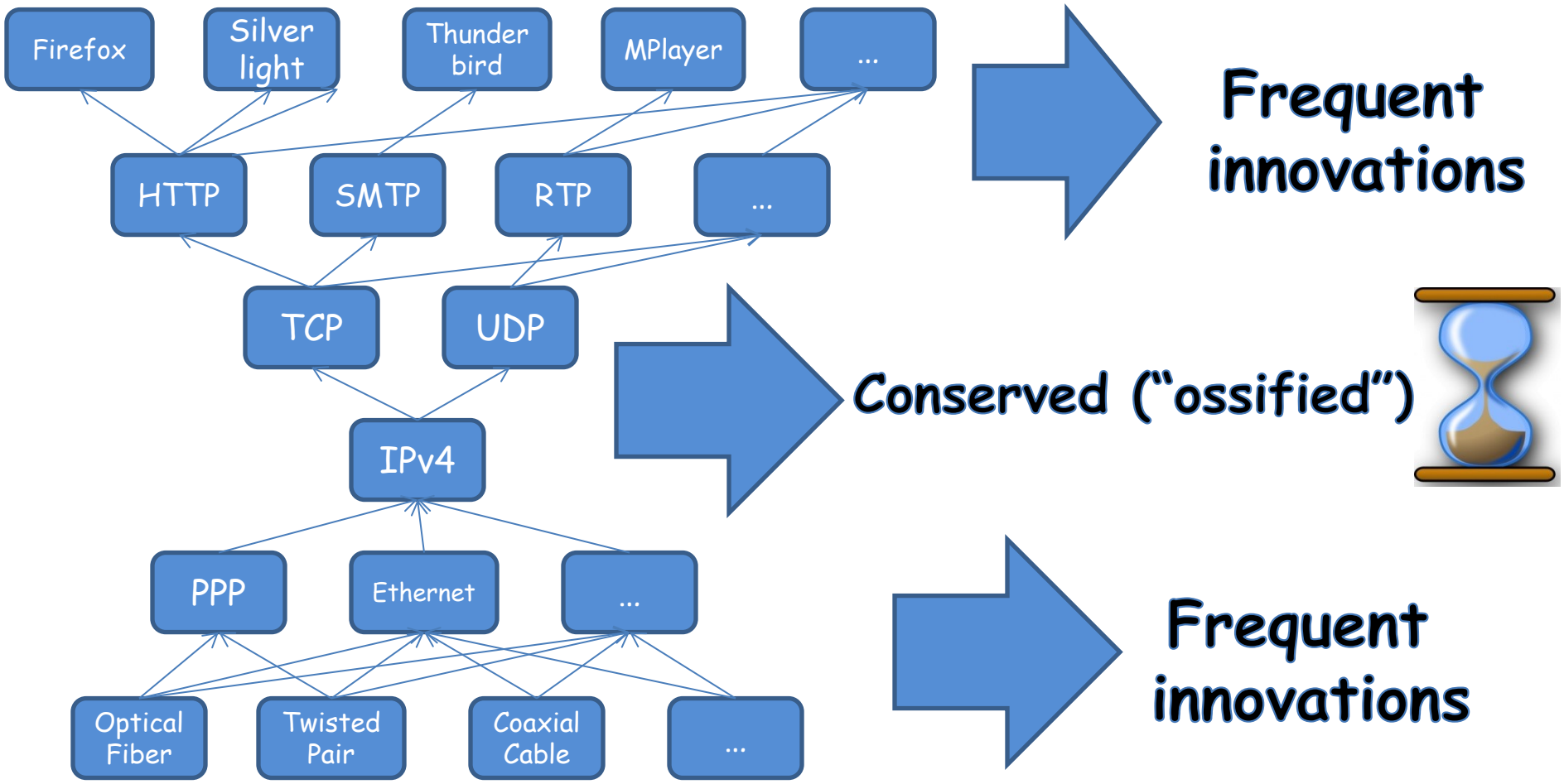
Why networks don't evolve?

Internet Hourglass problem…



Why networks don't evolve?

Internet Hourglass problem…



Limitations of Current Networking Technologies

- Complexity that leads to Static Nature
- Inability to Scale
- Vendor Dependence
- Hardware based networks, not easy to evolve

Limitations of Current Networking Technologies

- Complexity that leads to Static Nature
- Inability to Scale
- Vendor Dependence
- Hardware based networks, not easy to evolve

We need a software based approach…

Why a modern smartphone is better than Nokia 3310?

HW-oriented world

MyCode

Nokia 3310 **only**

Because of a software oriented paradigm!

SW-oriented world

MyCode

Abstract Android device
(API interface)

interface

Every physical device compliant with Android interface

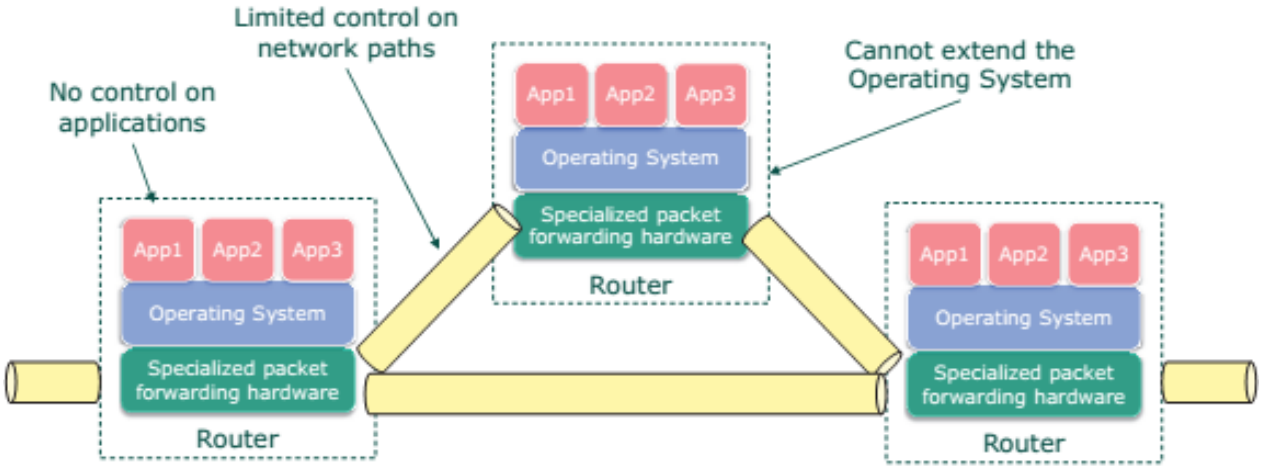
Benefits of a Software-oriented paradigm

- Abstract models hide technical, complicated, useless details
- Applications on abstract models can run on many different devices
- Applications can be replaced in any time without the need to change the hardware

Are there benefits of applying a software paradigm in a network architecture?

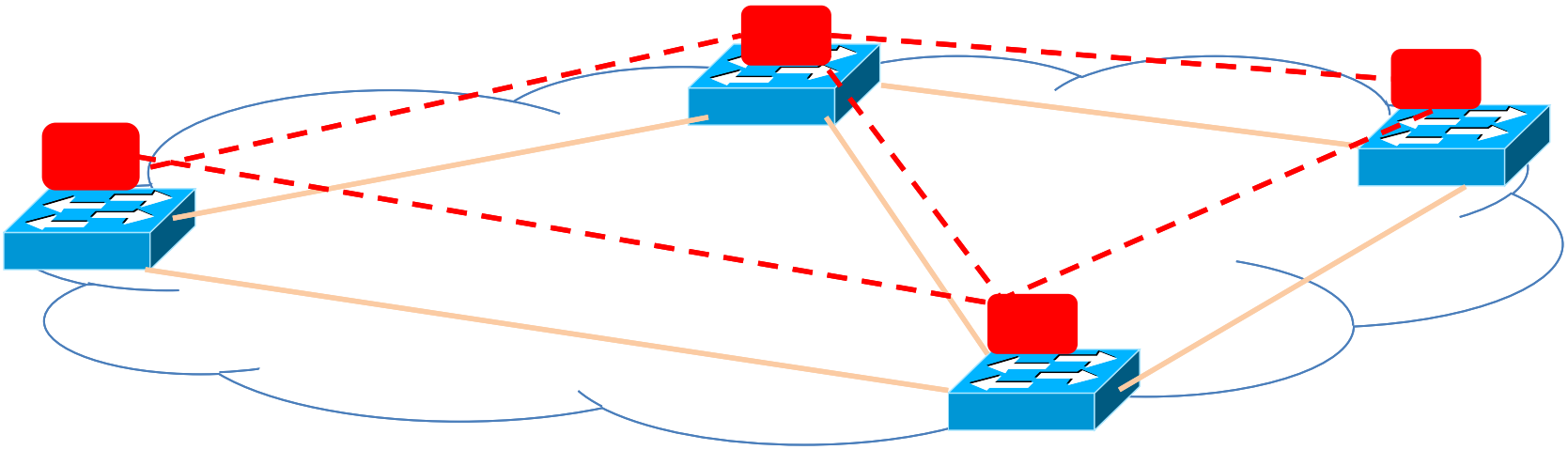
Internet, after 30+ years

- Almost the same protocols, same philosophy
- Reason: static and monolithic network apparatus:
  - Black box policy: no one can modify them, except the manufacturer



Closed to innovations in the infrastructure

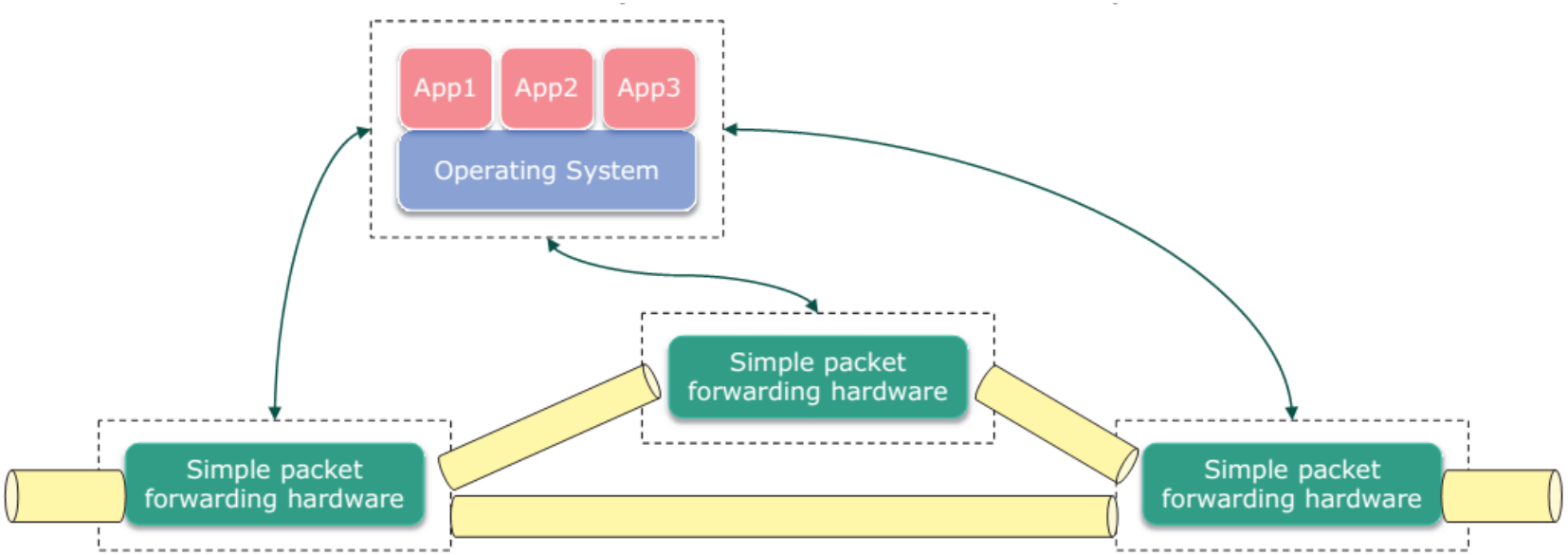
Planes of Networking



- **Data Forwarding plane:**
  - Packet forwarding
  - Fragmentation and reassembly
- **Control plane:**
  - Making routing tables
  - Setting packet handling policies

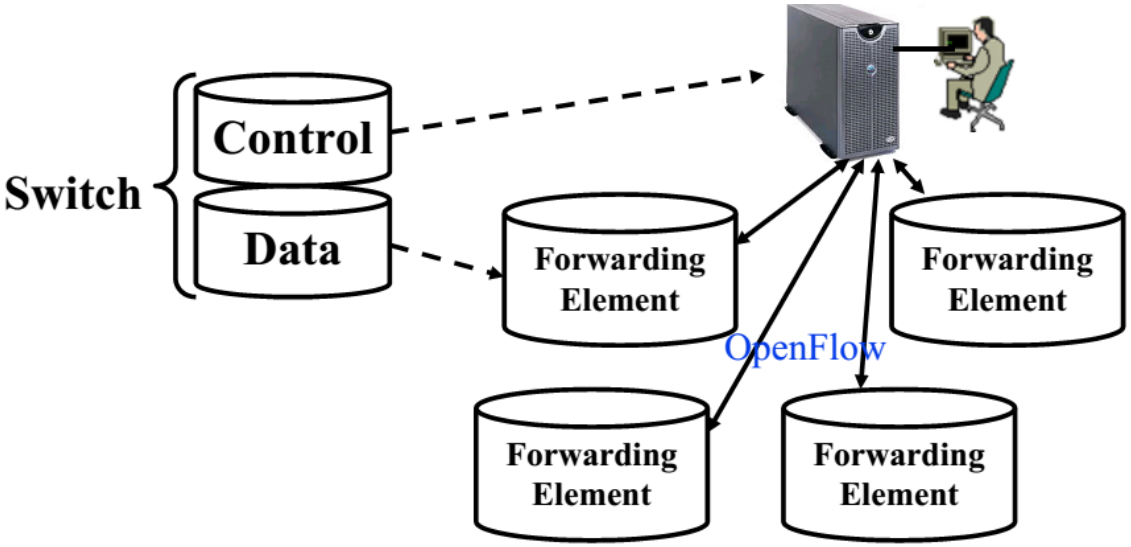
Software Defined Network

- Paradigm that introduces the possibility to program the network
- Based on three pillars
  - Separation of control and forwarding functions
  - Centralization of control
  - Well-defined interfaces (northbound and southbound)



SDN approach

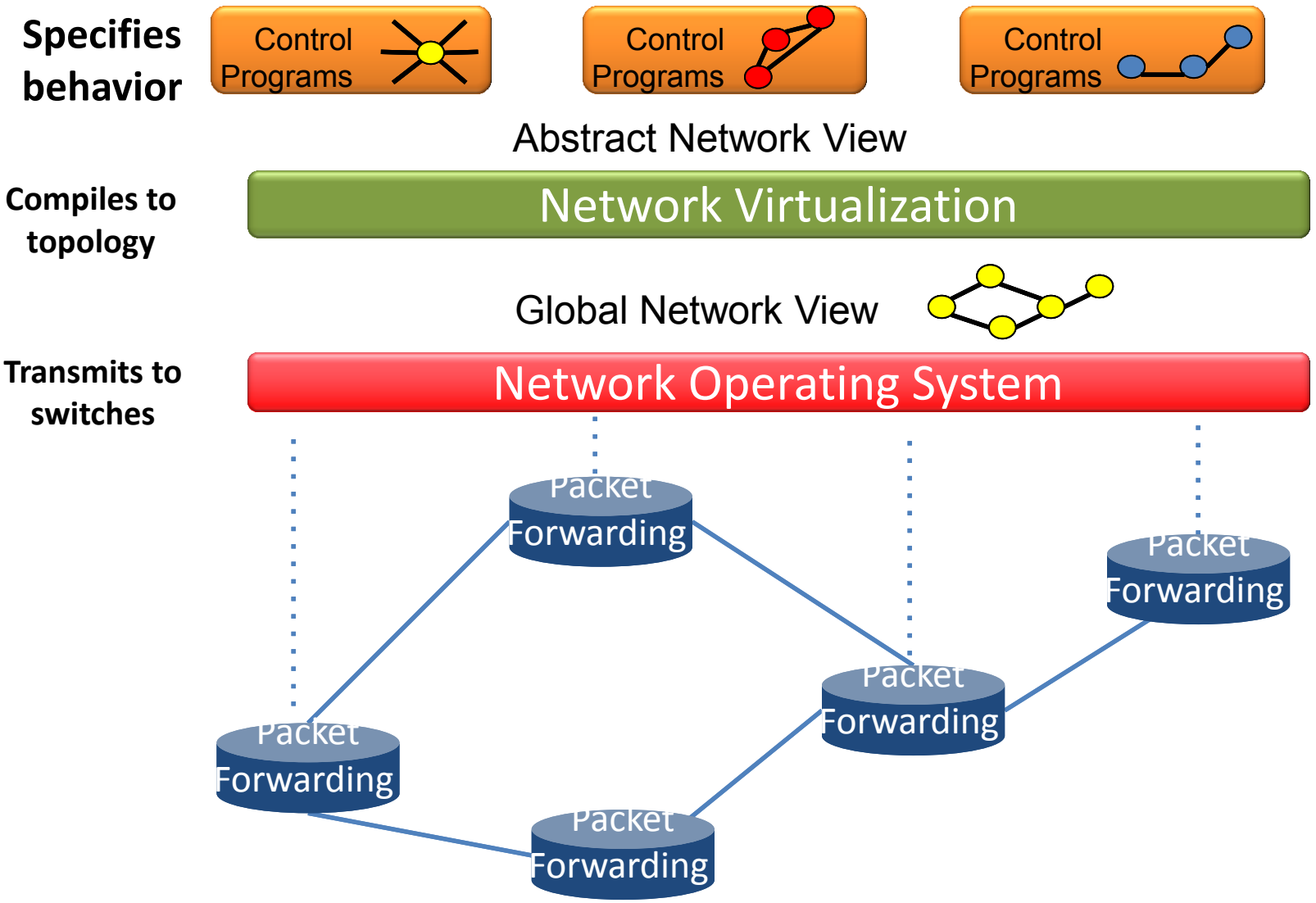
- Control logic is moved to a central controller
  - Switches only have forwarding elements
  - One expensive controller with a lot of cheap switches



By programming the controller, we can quickly change the entire network behavior

Network Management is not more platform dependent!

SDN architecture



Why SDN ? Abstraction

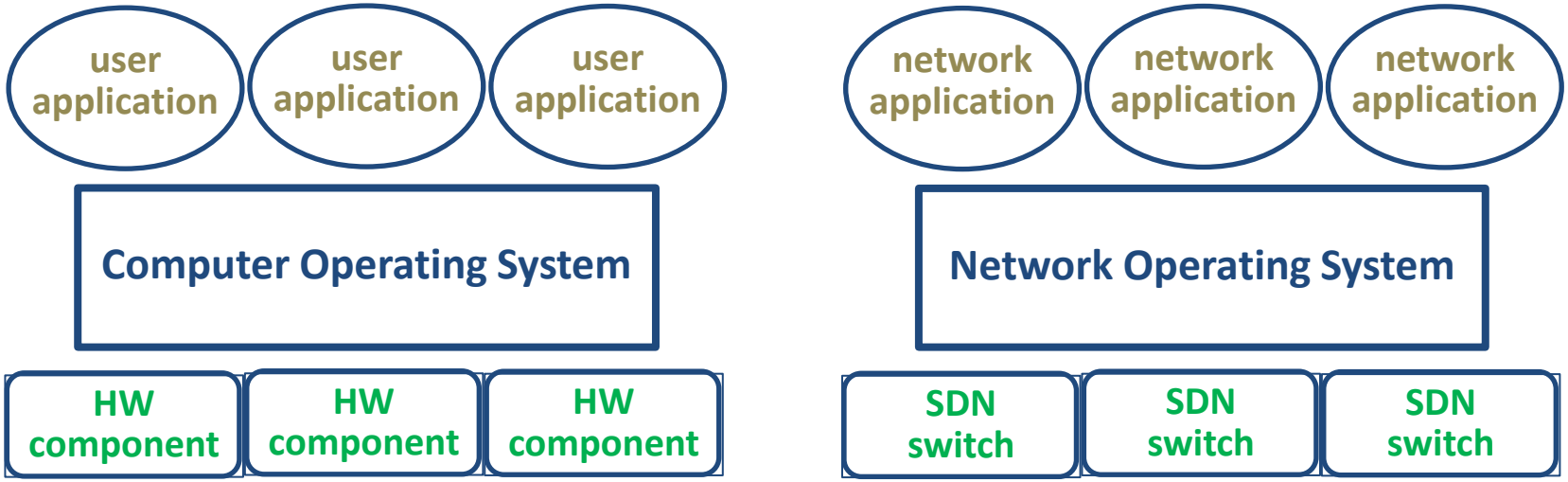
- In Computers an *abstraction* is a representation that reveals semantics needed *at a given level* while hiding implementation details:
  - a programmer to focus on necessary concepts without getting bogged down in unnecessary details
- **Much of the progress in CS resulted from finding new abstractions:**
  - It is very slow to code directly in assembly language (with 1 abstraction : mnemonics for opcodes)
  - It is much faster coding in high-level imperative language like Python
  - It is much faster yet coding in a declarative language
  - It is fastest coding in a domain-specific language (only contains the needed abstractions)

Control plane abstractions

SDN principle 1: *APIs instead of protocols*

Replace control plane protocols with well-defined APIs to network applications

SDN works as a Network Operating System
Network Apps can be added without changing the NOS



Packet forwarding abstraction

SDN principle 2: *Packet forwarding as a computational problem*

The function of any network node is to

1. receive a packet
2. observe packet fields
3. apply algorithm (classification, decision logic)
4. optionally edit packet
5. forward or discard packet

- SDN switch treats an incoming packet as a simple sequence of bytes.
- **SDN switch neither knows nor cares to which network layer the bytes it examines belong**

Flows

SDN principle 3:

Packets are handled solely based on the flow to which they belong

Flows are thus just like **F**orwarding **E**quivalence **C**lasses

A a flow may be determined by

- an IP prefix in an IP network
- A label in an MPLS network
- VLANs in VLAN cross-connect networks

The granularity of a flow depends on the application

Network state and graph algorithms

SDN principle 4: *Eliminate distributed protocols*

- **Replace distributed routing protocols with graph algorithms performed at a central location**

The forwarding algorithm is not performed at the network node,

So, how does it know how to forward packets ?

- In order to perform the forwarding algorithm on a packet belonging to a flow it need to know something of the network state (complete knowledge or very limited local knowledge)
- The decision algorithms is performed in a central location by an entity with full knowledge: **CONTROLLER**

Configuration

SDN principle 5:

SDN switches are **stupid** and flows are *configured* by an SDN controller

The network state to be stored at the individual NE is just a *flow table describing* how to forward a packet belonging to a flow

Conventional network elements have two parts:

1. smart but slow CPUs that calculate the forwarding policy
2. simple switch that just forwards packets according to the policy at step 1

Since the algorithms to build the forwarding policy are performed elsewhere

SDN brings the added bonus that we don't need the CPU

- Such a simplified network element is called **SDN switch**
- The flow table is configured by the **SDN controller**

Is SDN better than routing ?

OK, SDN switches may be cheaper…

but is that the only advantage of SDN ?

Distributed routing protocols are limited to

- finding simple connectivity
- minimizing number of hops

but can not perform more sophisticated operations, such as

- optimizing paths under constraints (e.g., security)
- setting up non-overlapping backup paths
- integrating networking functionalities (e.g., NAT, firewall) into paths

An SDN controller is omniscient (“God box”)

- can perform arbitrary optimization calculations on the network graph
- directly configures the forwarding actions of the SDN switches

But this advantage comes at a price

- the controller is a single point of failure
- additional (overhead) bandwidth is required
- additional set-up delay may be incurred

Standardization of SDN solutions

SDN Standard Developing Organizations

OPEN NETWORKING FOUNDATION

The SDO for OpenFlow standardization
Perceived as leader for SDN standardization.

I E T F

ETSI

Leading Telco operators established ISG
"Network Function Virtualization (NFV)"
Pre-standardization work for Carrier Networks.
Will probably **use SDN** to orchestrate NFV.

Orthogonal topics compared to ONF SDN
Focus on extending existing protocols for SDN without OpenFlow
Real work starts now! (I2RS)

ITU

Joint Coordination Activity on Software-Defined Networking

Question 21, a group for Future Networks
It is an established SDN group

broadband forum
member company

Some SDN-related work started

MEF

3GPP

tmforum

GSMA

OIF OPTICAL INTERNETWORKING FORUM

Some SDN-related discussions started

Open Networking Foundation (ONF)

- A non-profit industry consortium
  - Founded: March 2011, >130 member organizations
  - Telecom operators, network providers, service providers
  - Equipment vendors, networking and virtualization software suppliers, and chip technology providers
- Vision: Make Software-Defined Networking the new norm for networks
  - ONF wants to transform networking industry to software industry through open SDN standards
- Mission: commercialize and promote SDN and its underlying technologies for their user benefits
- ONF attempts to create the most relevant SDN standards
- Aims at the dissemination and practical implementation of SDN through open standard development such as OpenFlow

ONF structure

Management Structure

- Board of Directors (no vendors allowed)
- Executive Director (presently Dan Pitt, employee, reports to board)
- Technical Advisory Group (makes recommendations not decisions, reports to board)
- Working Groups (chartered by board, chair appointed by board)
- Council of Chairs (chaired by executive director, forwards draft standards to board)

ONF Board members

- **Dan Pitt** Executive Director
- **Nick McKeown** Stanford University
- **Scott Shenker** UC Berkeley and ICSI
- Deutsche Telecom AG
- Facebook
- Goldman Sachs
- Google
- Microsoft
- NTT Communications
- Verizon
- Yahoo

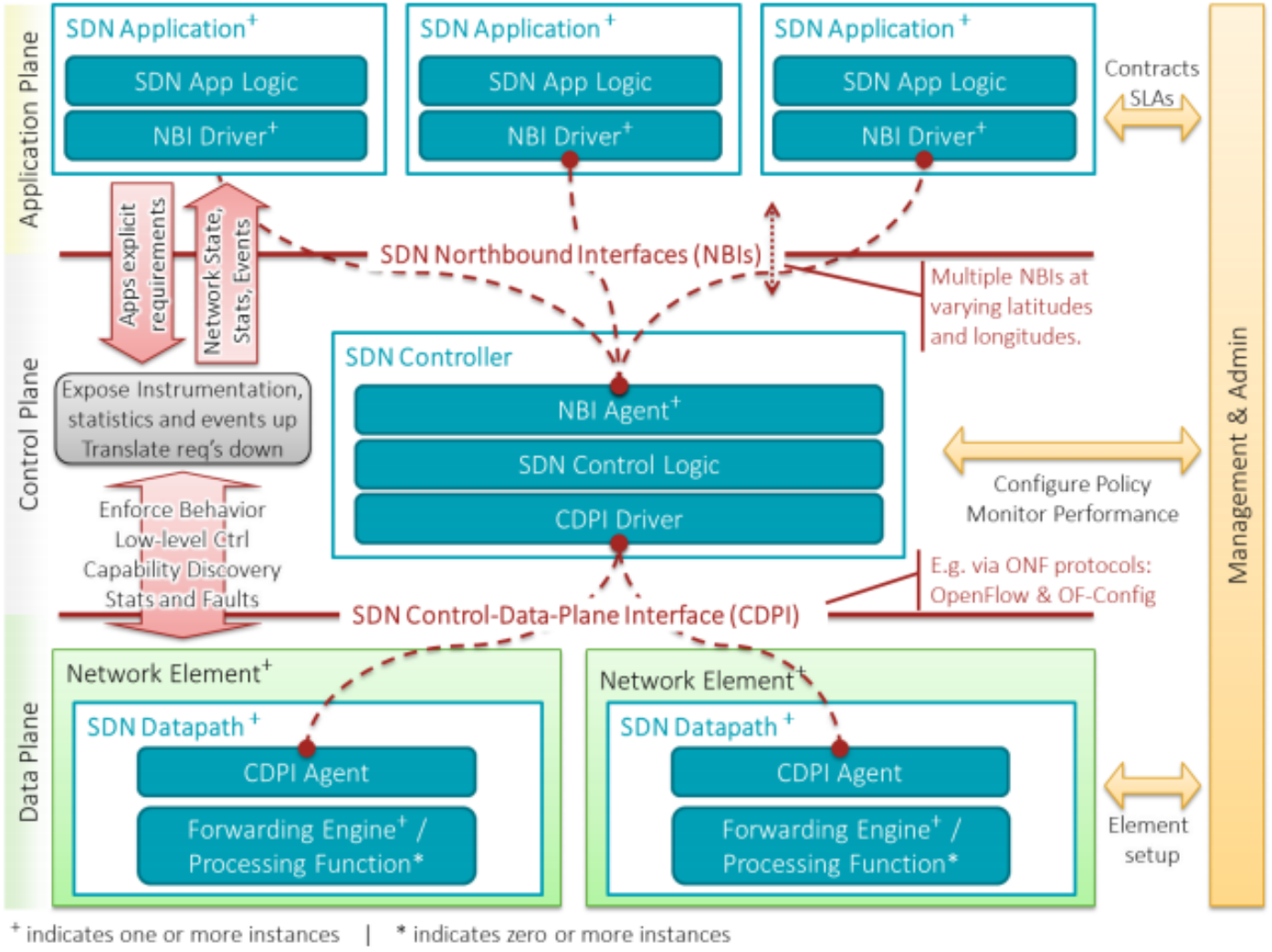
Working Groups

- Architecture and Framework
- Forwarding Abstraction
- Northbound Interface (new)
- Optical Transport (new)
- Wireless and Mobile (new)
- Configuration and Management
- Testing and Interoperability
- Extensibility
- Migration
- Market Education
- Hybrid - closed

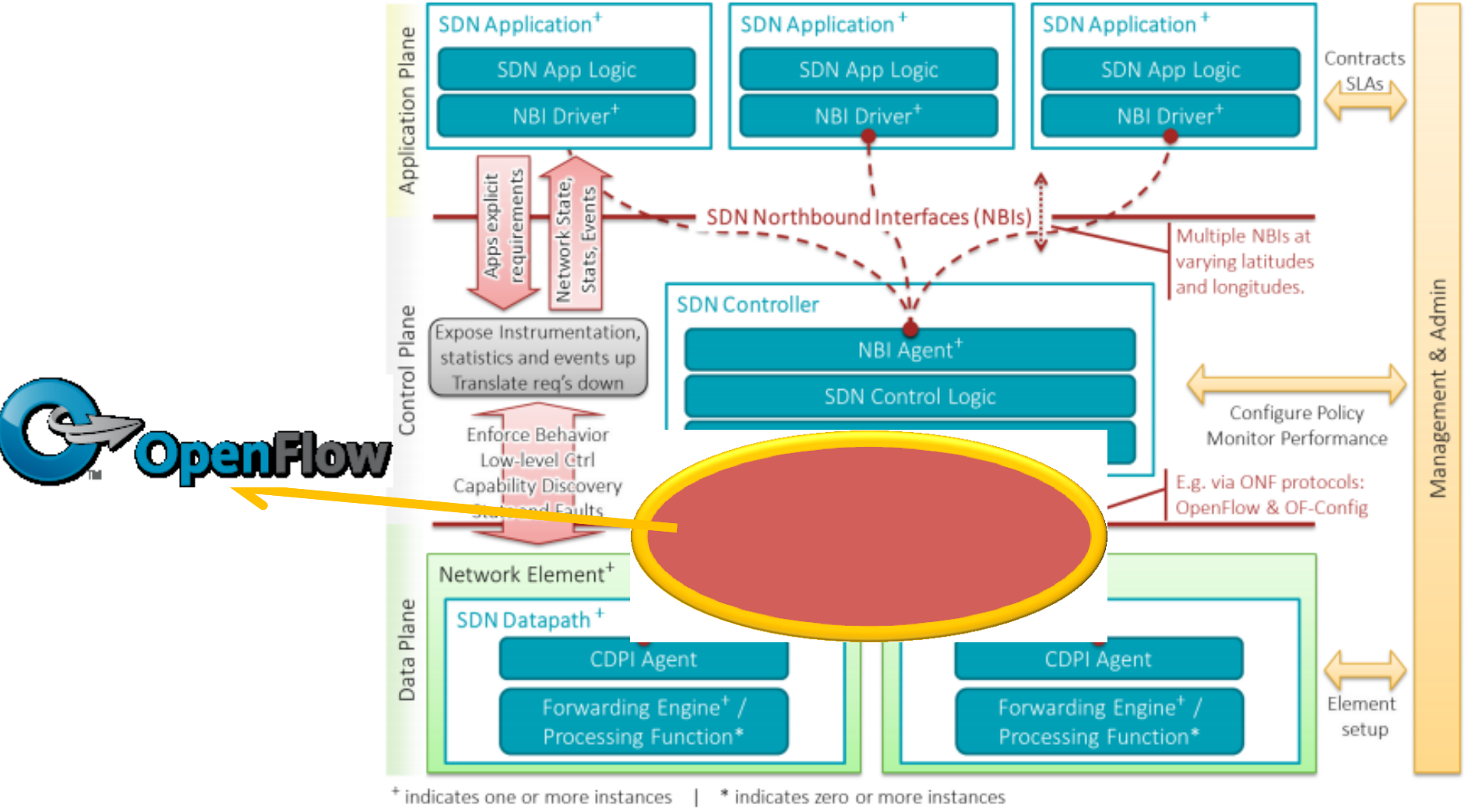
ONF SDN Architecture Overview

Key features

- Applications can be network aware
- Control plane is logically centralized
- Control plane is decoupled from the data plane
- SDN Controller has complete control of the SDN Datapaths



ONF SDN Architecture Overview



- Define mechanisms for configuration of OpenFlow capable network devices to realize the vision of Software Defined Networking

The OpenFlow specifications describe
- the southbound protocol between OF controller and OF switches
- the operation of the OF switch

The OpenFlow specifications do not define
- the northbound interface from OF controller to applications
- how to boot the network
- how an E2E path is set up by touching multiple OF switches
- how to configure or maintain an OF switch (see of-config)

The **OF-CONFIG** specification defines
a configuration and management protocol between
OF configuration point and *OF capable switch*
- configures which OpenFlow controller(s) to use
- configures queues and ports
- remotely changes port status (e.g., up/down)

OpenFlow matching

- The basic entity in OpenFlow is the ***flow***
  - A flow is a sequence of packets that are forwarded through the network in the same way
- Packets are classified as belonging to flows
  - based on **match fields** (switch ingress port, packet headers, metadata)
  - detailed in a **flow table** (list of match criteria)
- Only a finite set of match fields is presently defined
- The matching operation is *exact match* with certain fields allowing *bit-masking*
- Since OF 1.1 the matching proceeds in a **pipeline**

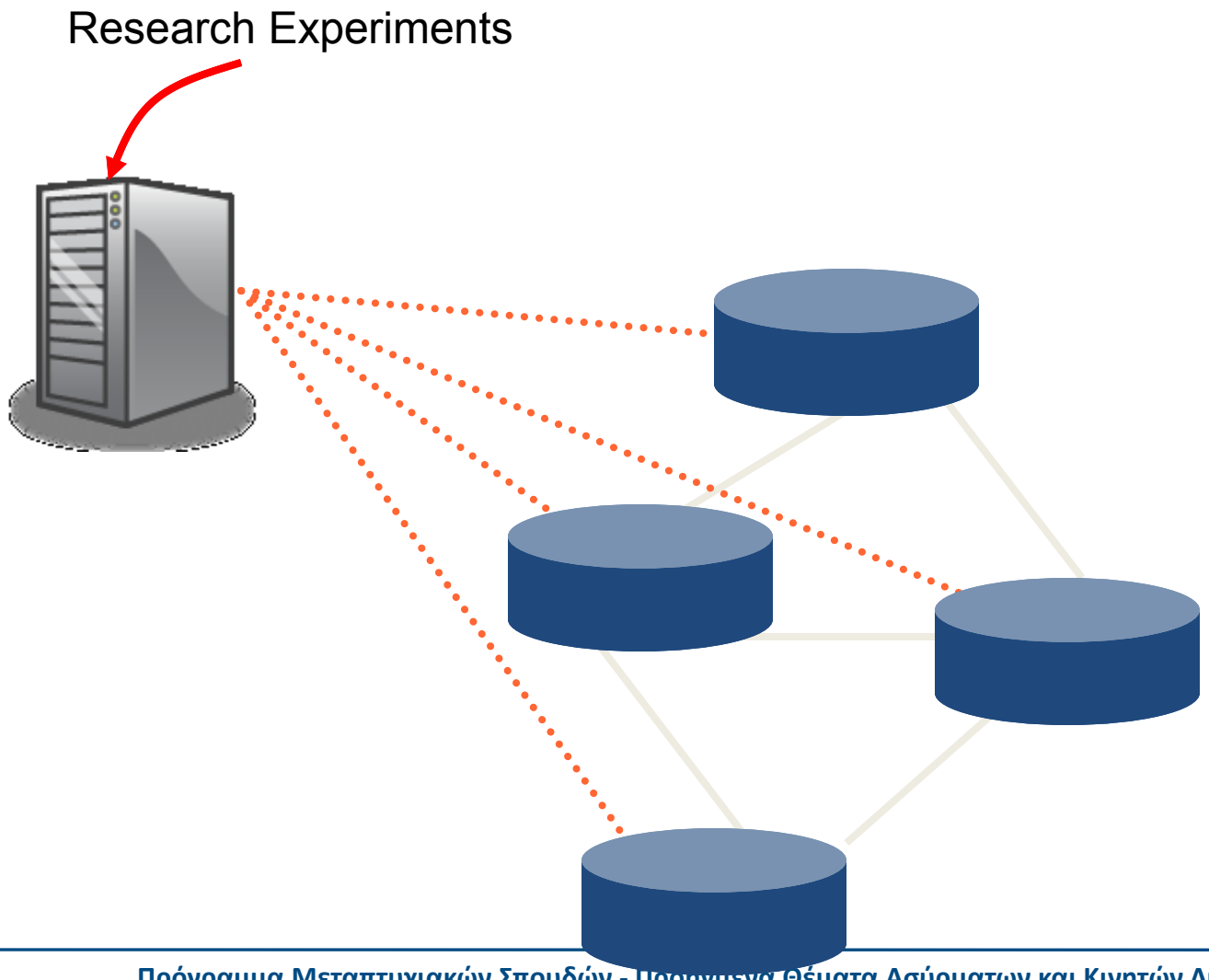
OpenFlow: flow table

	match fields	actions	counters
flow entry →	match fields	actions	counters
	match fields	actions	counters
flow miss entry →		actions	counters

- The flow table is populated only by the controller
- The incoming packet is matched by comparing to match fields
- For simplicity, matching is exact match to a static set of fields
- If matched, actions are performed and counters are updated
- Actions include editing, metering, and forwarding

Step 1:

Separate Control from Datapath



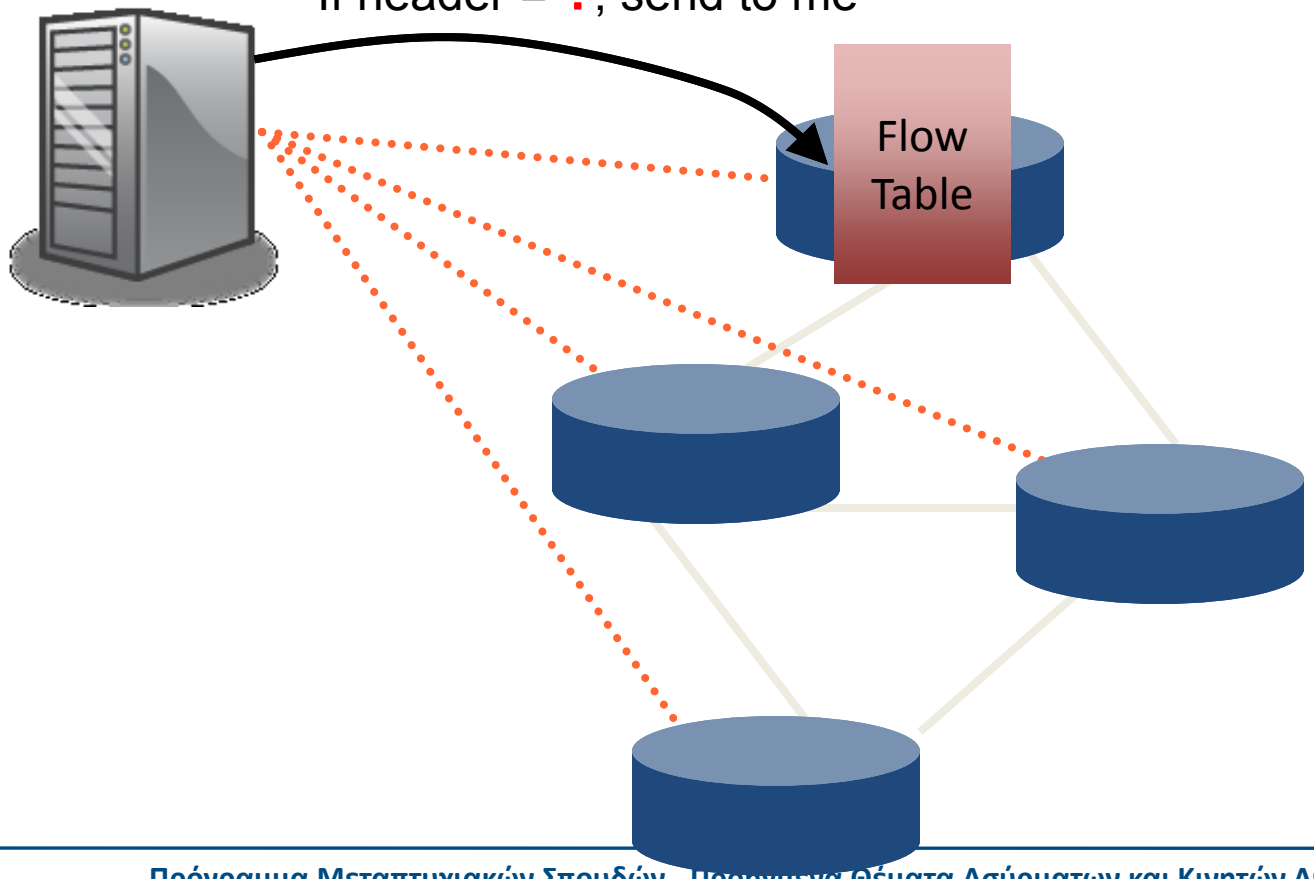
Step 2:

Cache flow decisions in datapath

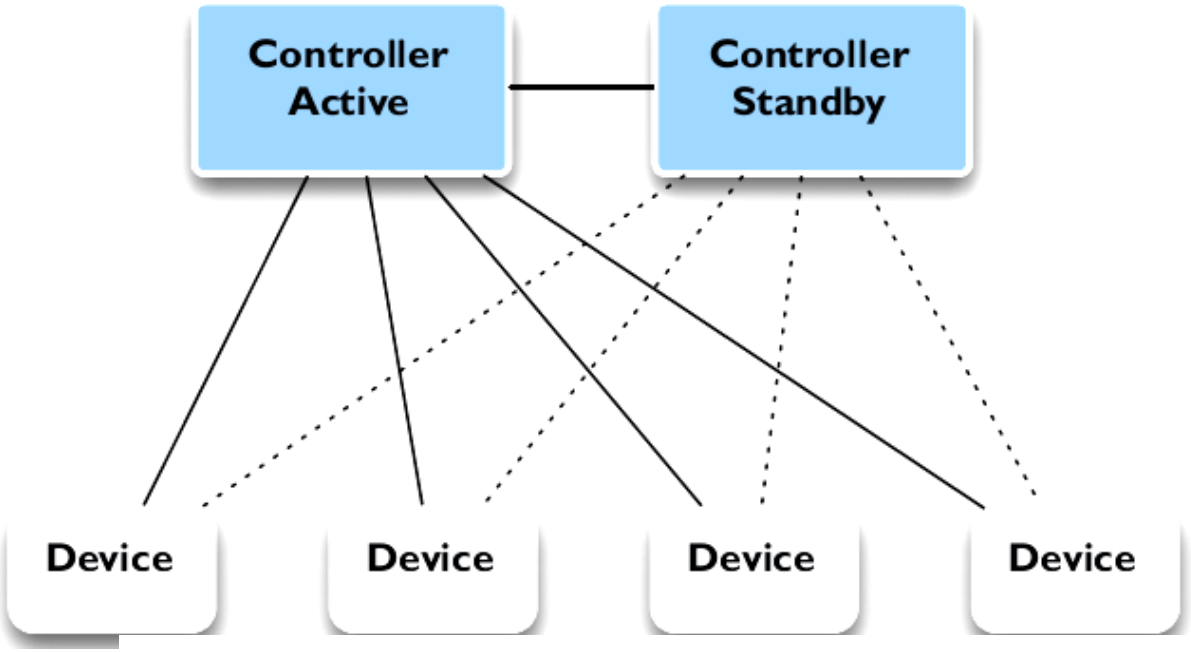
"If header = ***x***, send to port 4"

"If header = ***y***, overwrite header with ***z***, send to ports 5,6"

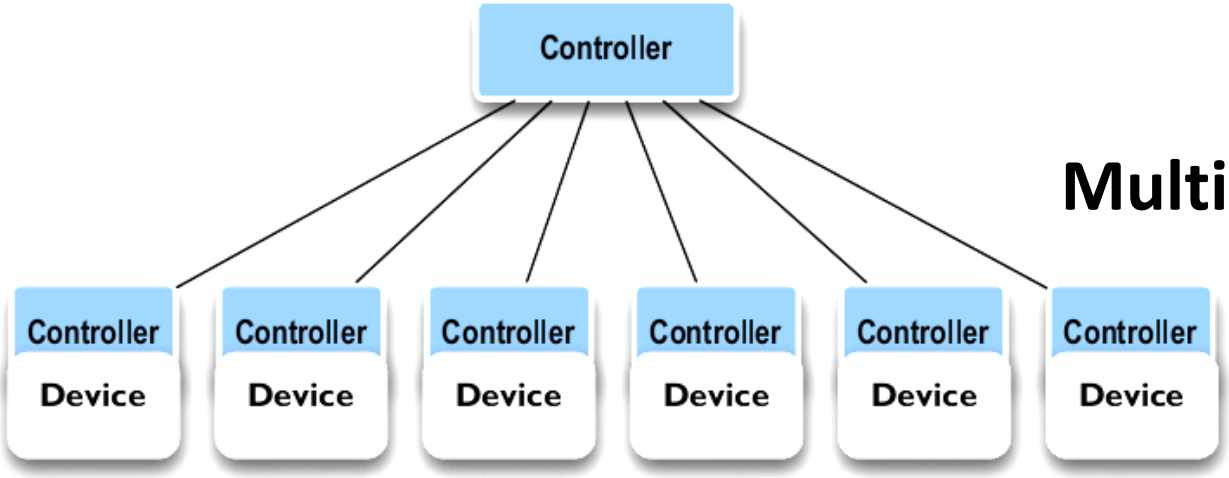
"If header = **?**, send to me"



CONTROLLER ARCHITECTURES

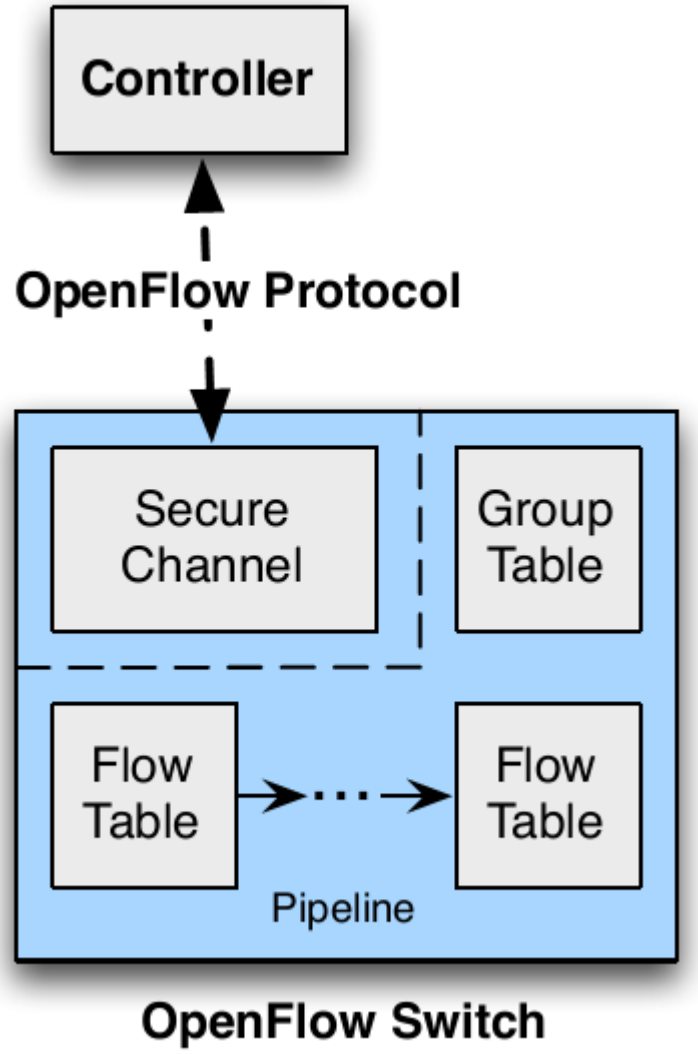


Centralized
Or
Distribuited



Multilayer

OpenFlow Switch



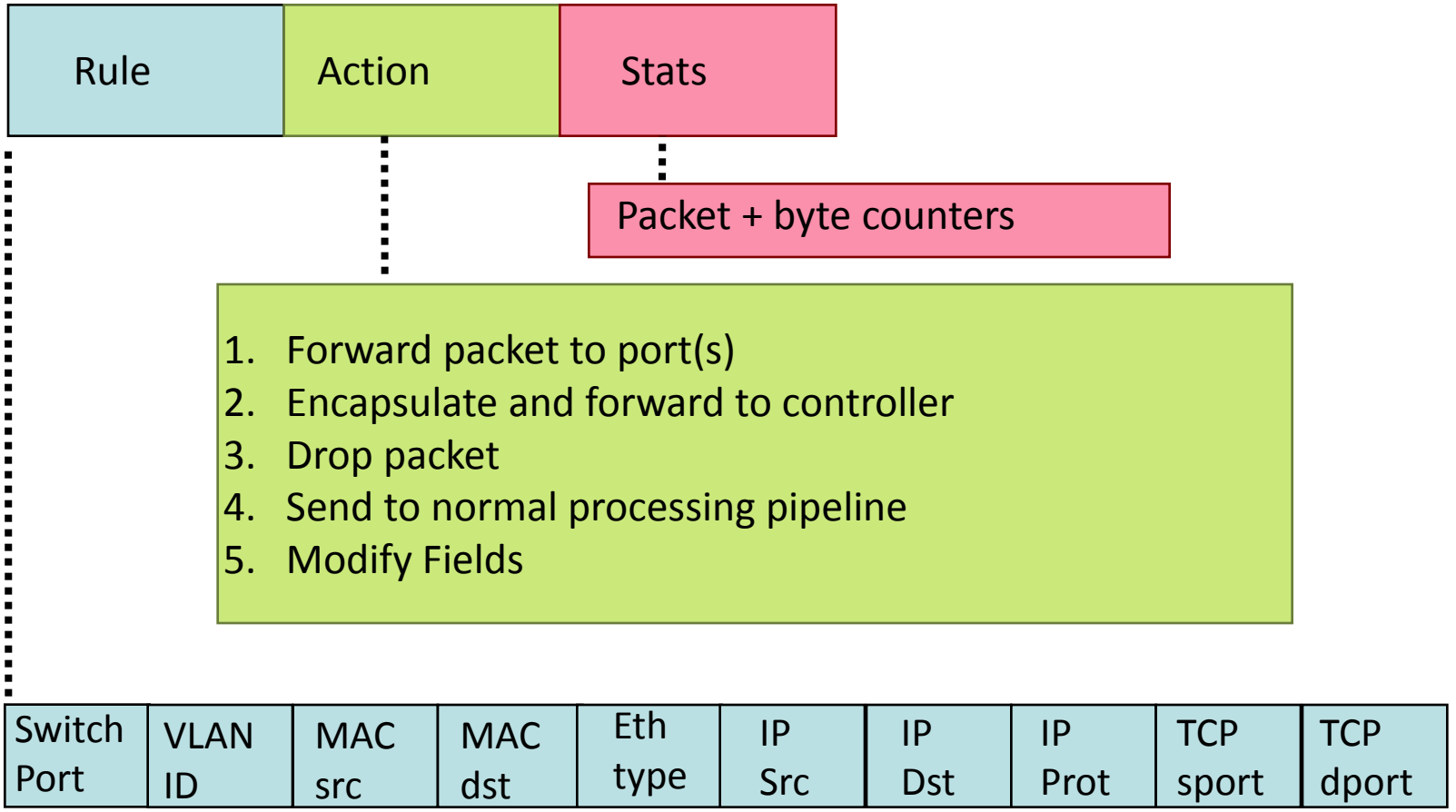
An OpenFlow Switch consists of :

- one or more flow tables and a group table, which perform packet lookups and forwarding
- an OpenFlow channel to an external controller

The controller manages the switch via the OpenFlow protocol. Using this protocol, the controller can add, update, and delete flow entries, both reactively (in response to packets) and proactively.

OpenFlow Basics

Flow Table Entries



+ mask what fields to match

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	Forward
*	*	*	*	*	*	*	*	*	22	drop

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

OF 1.1+ flow table operation

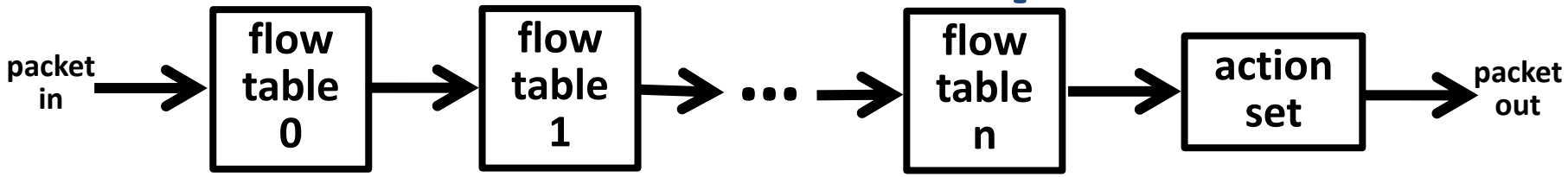


Table matching

- each flow table is ordered by priority
- highest priority match is used (match can be made "negative" using drop action)
- matching is exact match with certain fields allowing bit masking
- table may specify ANY to wildcard the field
- fields matched may have been modified in a previous step

Although the pipeline was introduced for scalability
it gives more expressibility to OF matching syntax (although no additional semantics)
In addition to the verbose

```
if (field1=value1) AND (field2=value2) then …
if (field1=value3) AND (field2=value4) then …
```

it is now possible to accommodate

```
if (field1=value1) then if (field2=value2)  then …
                        else  if (field2=value4)  then …
```

Unmatched packets

What happens when no match is found in the flow table ?

- **A flow table *may* contain a flow miss entry to catch unmatched packets**
- **The flow miss entry must be inserted by the controller just like any other entry it is defined as wildcard on all fields, and lowest priority**
- The flow miss entry may be configured to :
  - discard packet
  - forward to subsequent table
  - forward (OF-encapsulated) packet to controller
  - use “normal” (conventional) forwarding (for OF-hybrid switches)
- If there is no flow miss entry the packet is by default discarded but this behavior may be changed via of-configuration operations

OF switch ports

The ports of an OpenFlow switch can be physical or logical

The following ports are defined :

- physical ports (connected to switch hardware interface)
- logical ports connected to tunnels (tunnel ID and physical port are reported to controller)
- ALL output port (packet sent to all ports except input and blocked ports)
- CONTROLLER packet from or to controller
- TABLE represents start of pipeline
- IN_PORT output port which represents the packet's input port
- NORMAL optional port sends packet for conventional processing (hybrid switches only)
- FLOOD output port sends packet for conventional flooding

OpenFlow statistics

OF switches maintain **counters** for every

- flow table
- flow entry
- port
- queue
- group
- group bucket
- meter
- meter band

Counters are unsigned and wrap around without overflow indication

Counters may count received/transmitted packets, bytes, or durations

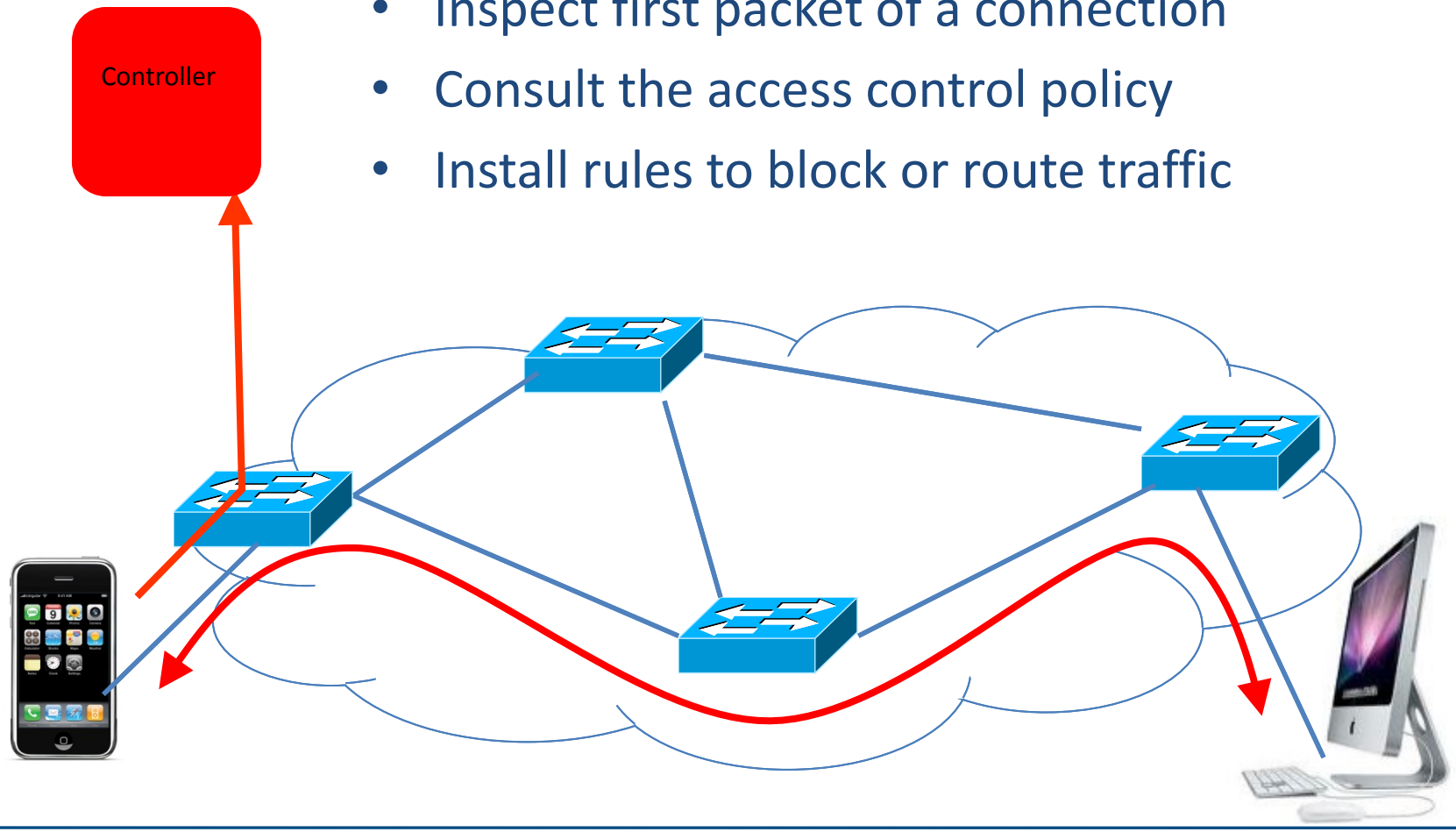
Flow removal and expiry

- Flows may be explicitly deleted by the controller at any time
- Flows may be configured with finite lifetimes and are automatically removed upon expiry
- Each flow entry has two timeouts
  - hard_timeout : if non-zero, the flow times out after X seconds
  - idle_timeout : if non-zero, the flow times out after not receiving a packet for X seconds

- When a flow is removed for any reason, there is flag which requires the switch to inform the controller:
  - that the flow has been removed
  - the reason for its removal (expiry/delete)
  - the lifetime of the flow
  - statistics of the flow

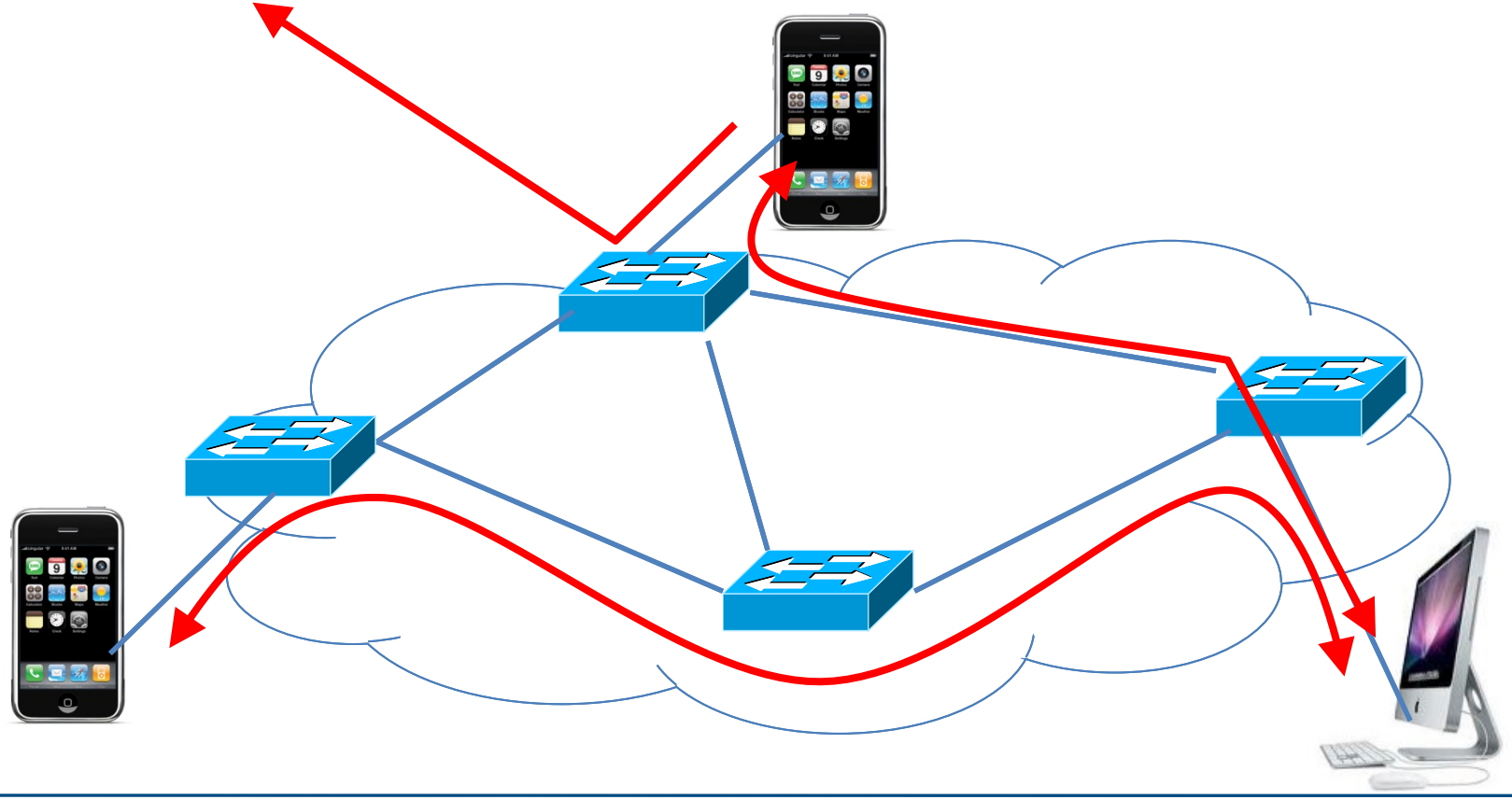
OpenFlow Applications:
Dynamic Access Control

- Inspect first packet of a connection
- Consult the access control policy
- Install rules to block or route traffic



Seamless Mobility/Migration

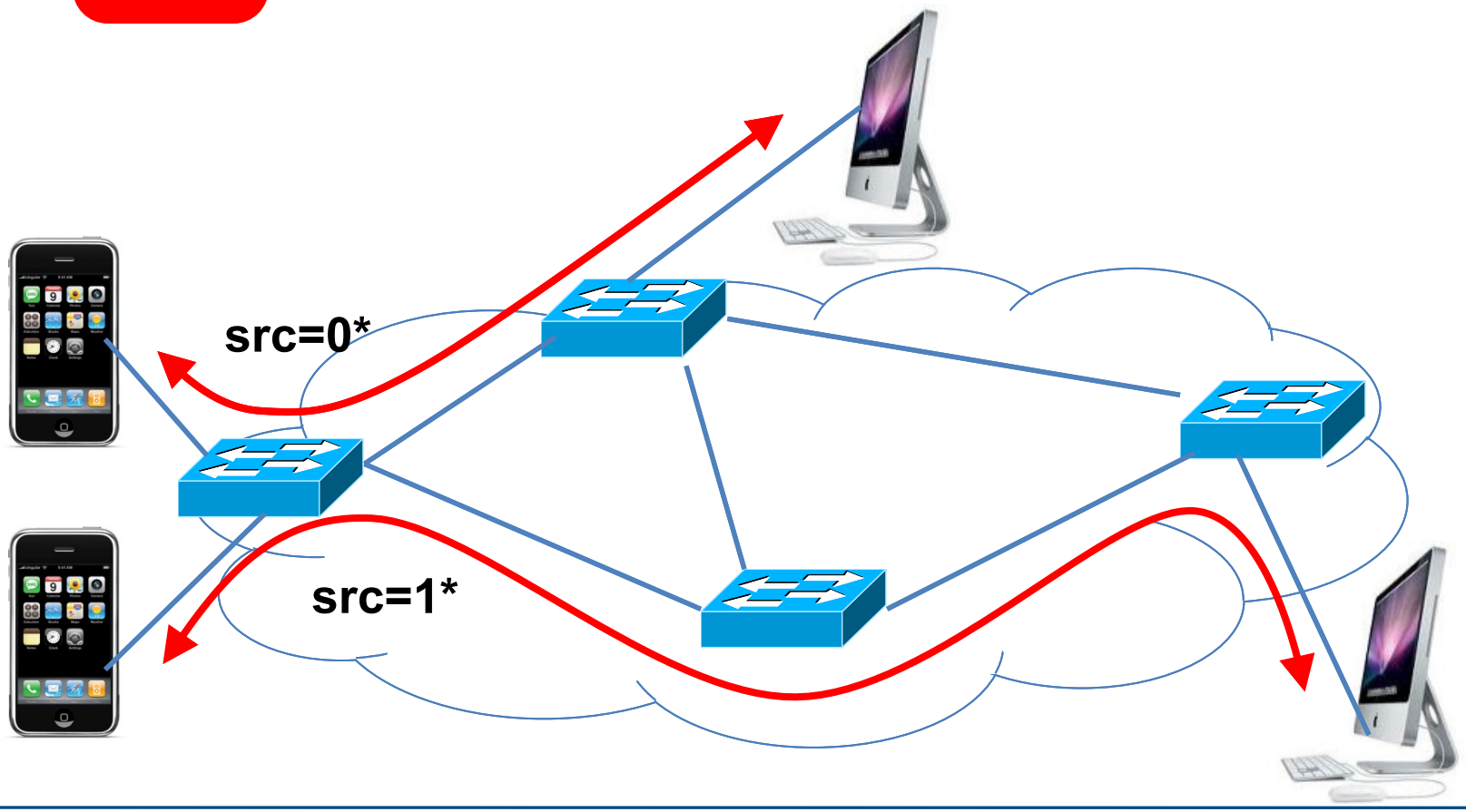
- See host send traffic at new location
- Modify rules to reroute the traffic



Server Load Balancing

Controller

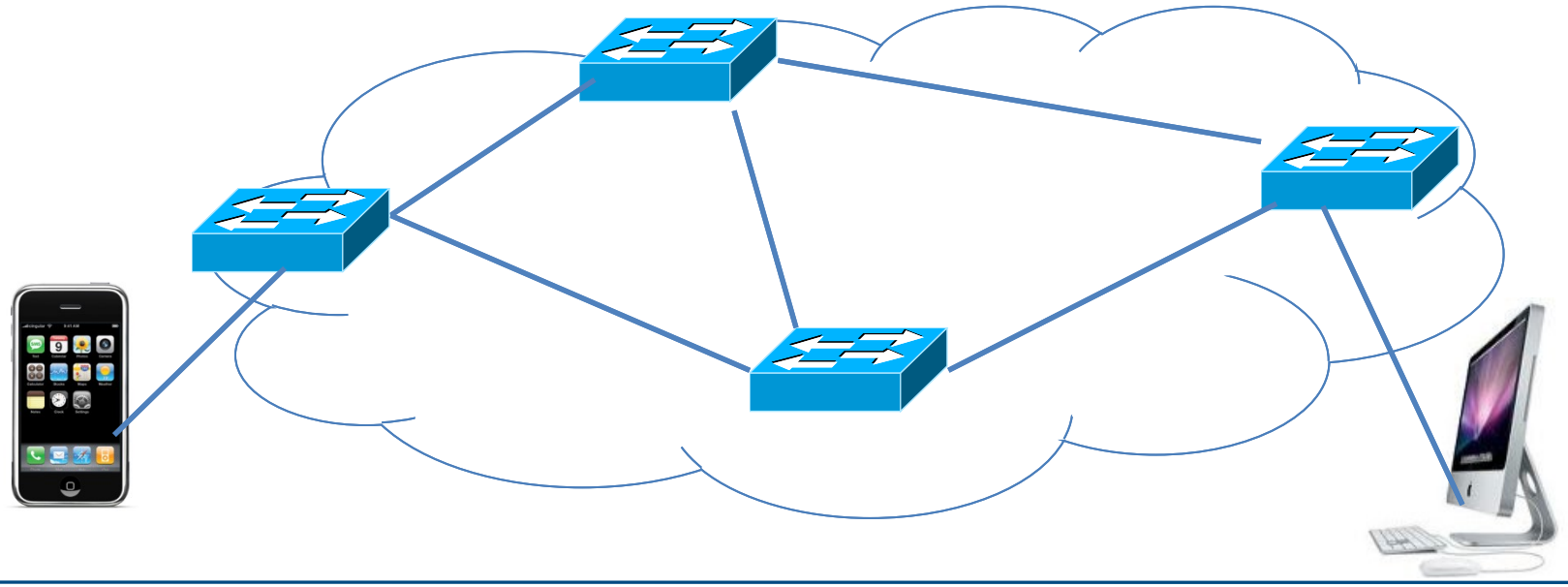
- Pre-install load-balancing policy
- Split traffic based on source IP



Network Slicing

Controller #1 **Controller #2** **Controller #3**

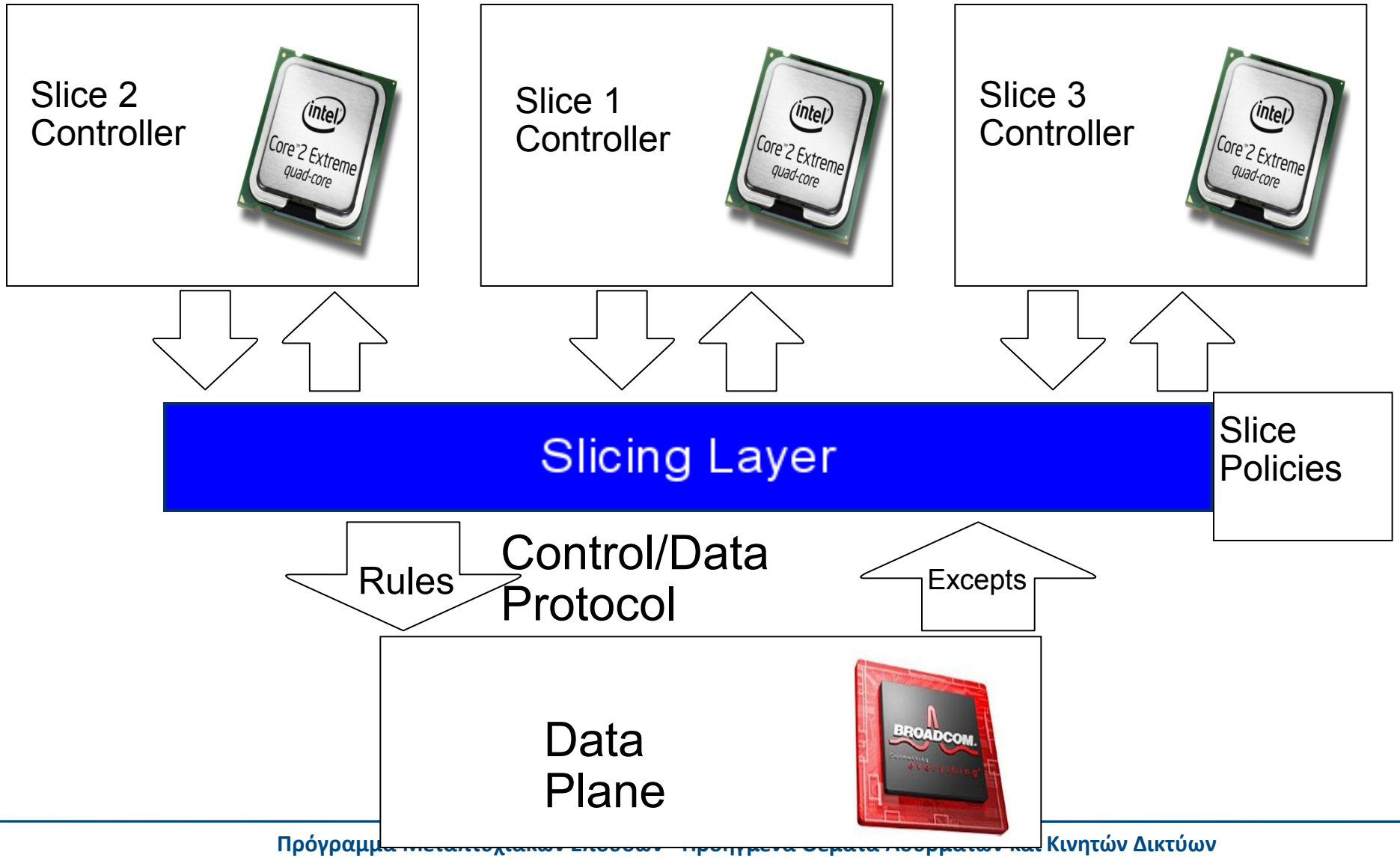
Partition the space of packet headers



Network Slicing

- Divide the production network into logical *slices*
  - each slice/service controls its own packet forwarding
  - users pick which slice controls their traffic: opt-in
  - existing production services run in their own slice
    - e.g., Spanning tree, OSPF/BGP
- Enforce strong isolation between slices
  - actions in one slice do not affect another
- Allows the (logical) testbed to mirror the production network
  - real hardware, performance, topologies, scale, users
- Prototype implementation: FlowVisor

Add a Slicing Layer Between Planes



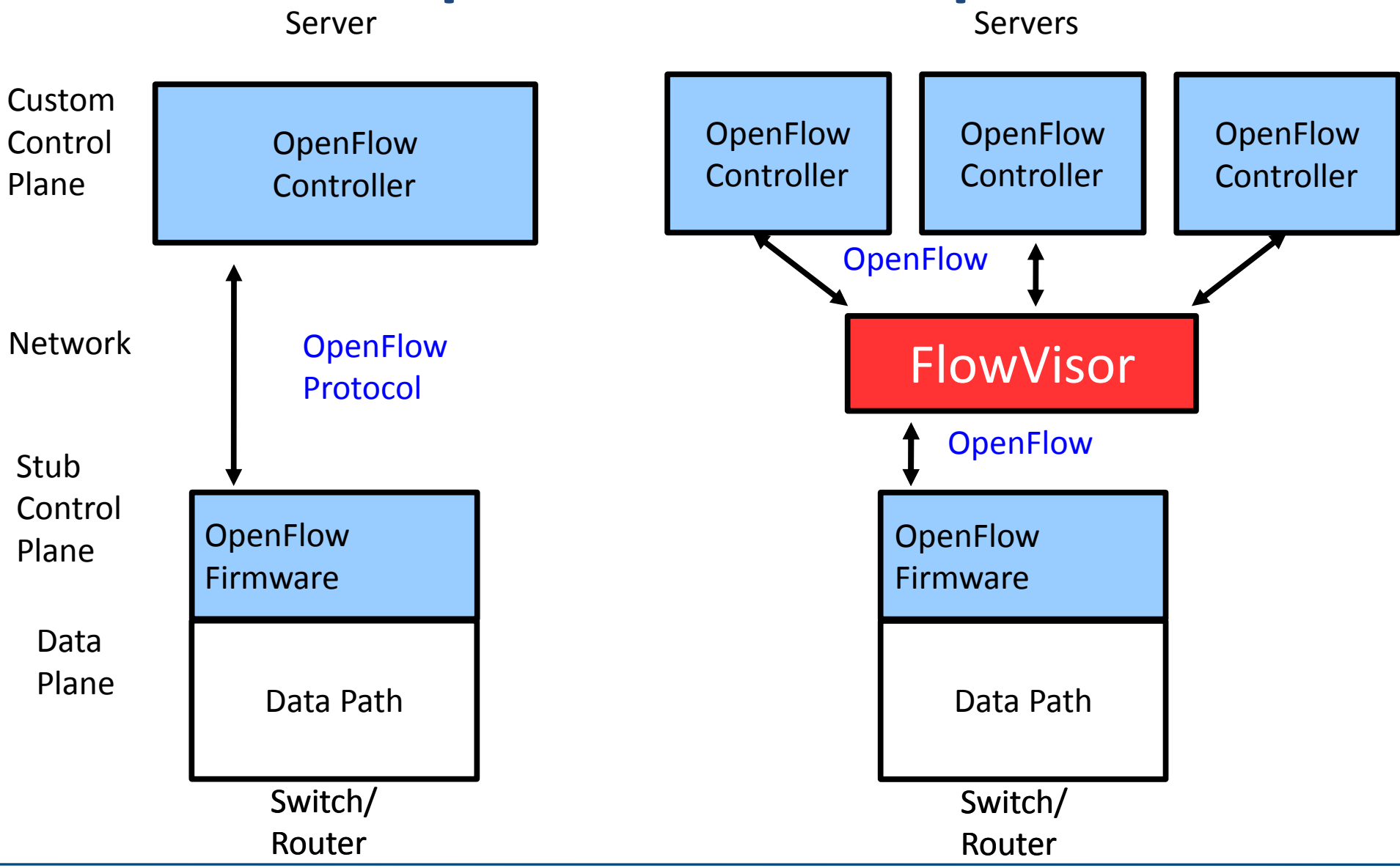
Network Slicing Architecture

- A network slice is a collection of sliced switches/routers
  - Data plane is unmodified
  - Packets forwarded with no performance penalty
  - Slicing with existing ASIC
- Transparent slicing layer
  - each slice believes it owns the data path
  - enforces isolation between slices
    - i.e., rewrites, drops rules to adhere to slice police
  - forwards exceptions to correct slice(s)

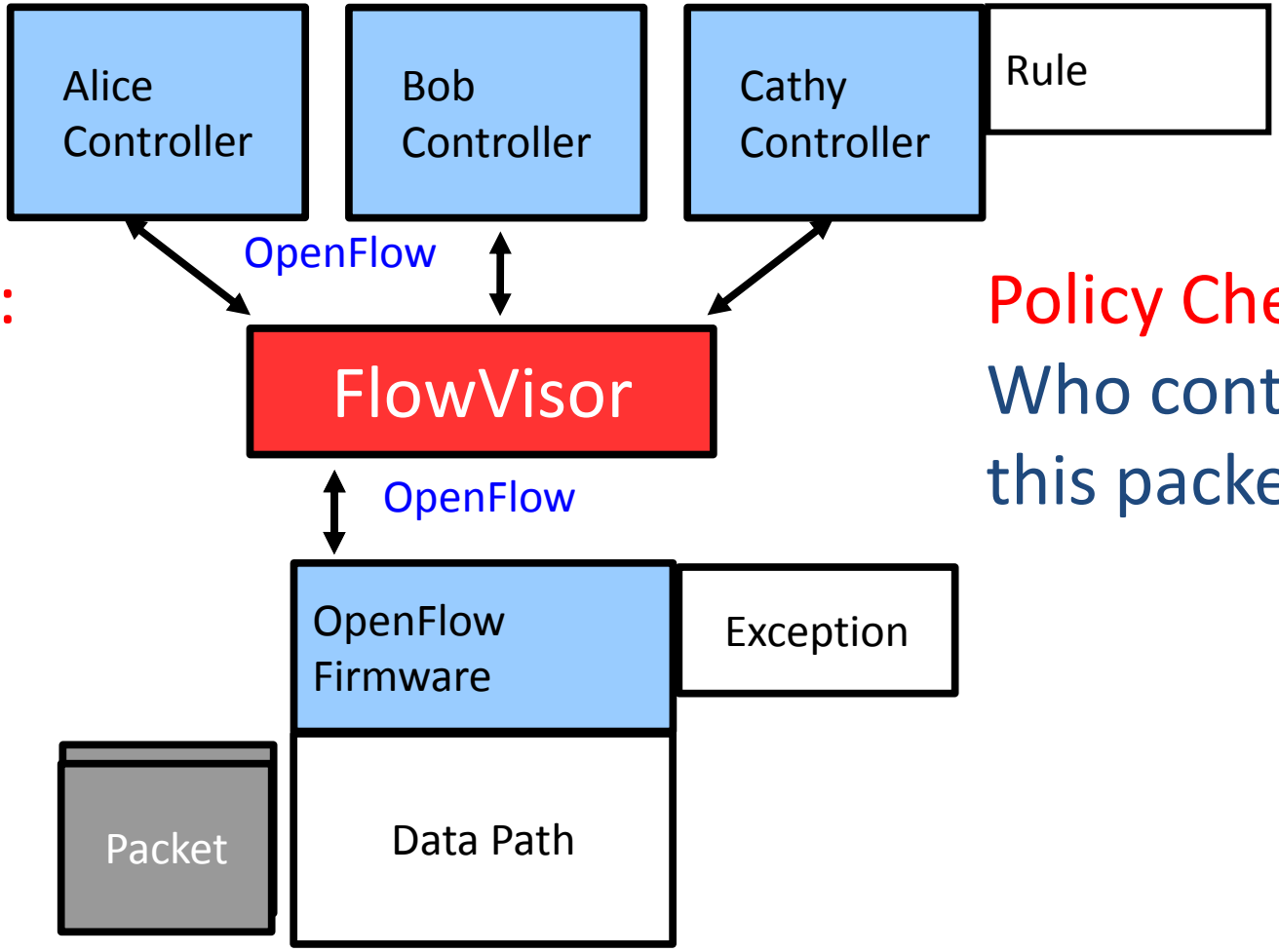
Slicing Policies

- The policy specifies resource limits for each slice:
  - Link bandwidth
  - Maximum number of forwarding rules
  - Topology
  - Fraction of switch/router CPU
  - *FlowSpace: which packets does the slice control?*

FlowVisor Implemented on OpenFlow



FlowVisor Message Handling



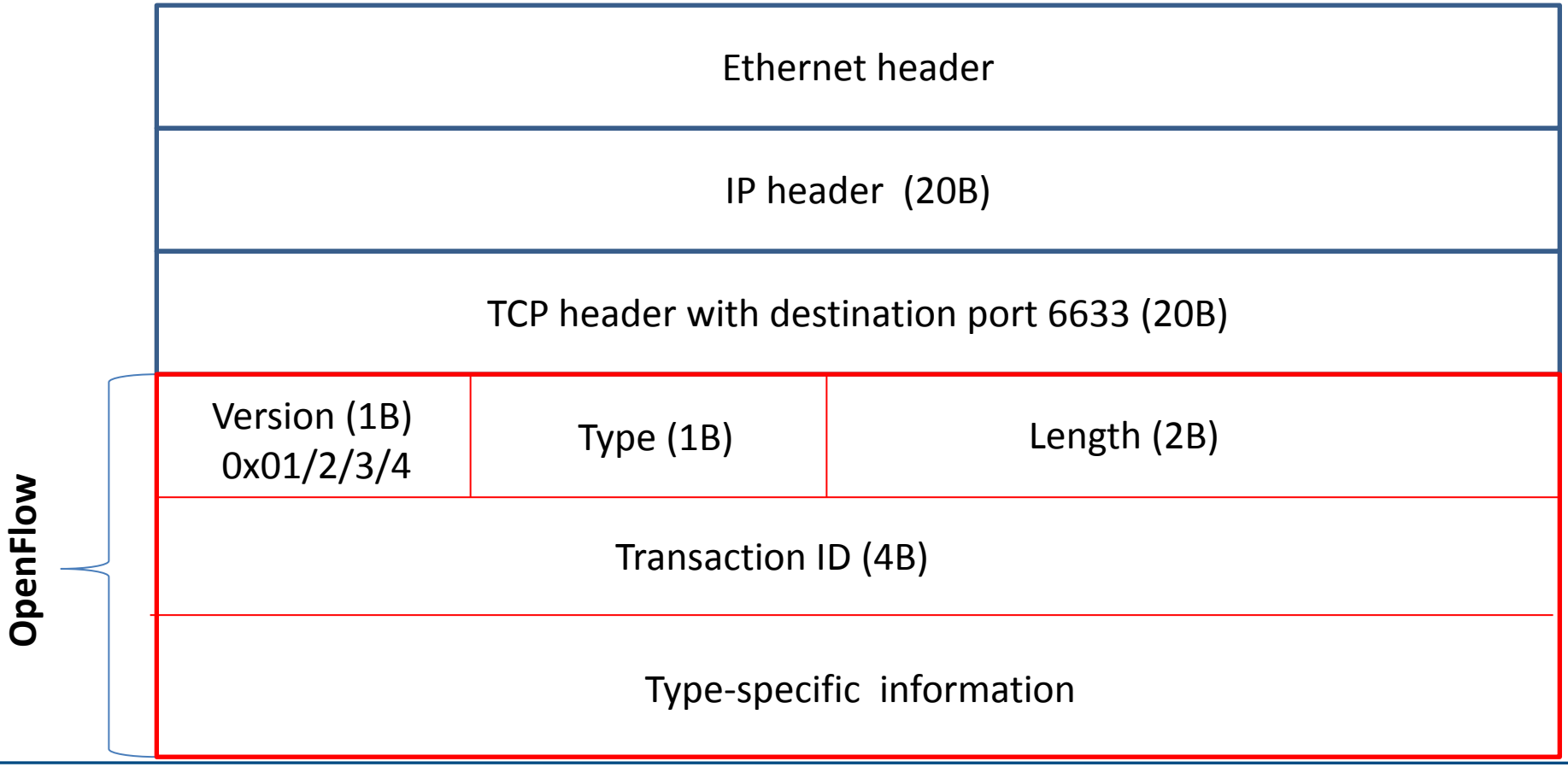
Policy Check:
Is this rule allowed?

Policy Check:
Who controls this packet?

OpenFlow protocol packet format

OF runs over TCP (optionally SSL for secure operation) using port 6633 and is specified by C **struct**s

OF is a very low-level specification (assembly-language-like)



OpenFlow messages

The OF protocol was built to be *minimal* and *powerful* (like x86 instruction set ☺)
and indeed it is low-level assembly language-like

There are 3 types of OpenFlow messages :

OF controller to switch

- populates flow tables which SDN switch uses to forward
- request statistics

OF switch to controller (asynchronous messages)

- packet/byte counters for defined flows
- sends packets not matching a defined flow

Symmetric messages

- hellos (startup)
- echoes (heartbeats, measure control path latency)
- experimental messages for extensions

OpenFlow message types

Symmetric messages
- **0** HELLO
- **1** ERROR
- **2** ECHO_REQUEST
- **3** ECHO_REPLY
- **4** EXPERIMENTER

Switch configuration
- **5** FEATURES_REQUEST
- **6** FEATURES_REPLY
- **7** GET_CONFIG_REQUEST
- **8** GET_CONFIG_REPLY
- **9** SET_CONFIG

Asynchronous messages
- **10** PACKET_IN = 10
- **11** FLOW_REMOVED = 11
- **12** PORT_STATUS = 12

Controller command messages
- **13** PACKET_OUT
- **14** FLOW_MOD
- **15** GROUP_MOD
- **16** PORT_MOD
- **17** TABLE_MOD

Multipart messages
- **18** MULTIPART_REQUEST
- **19** MULTIPART_REPLY

Barrier messages
- **20** BARRIER_REQUEST
- **21** BARRIER_REPLY

Queue Configuration messages
- **22** QUEUE_GET_CONFIG_REQUEST
- **23** QUEUE_GET_CONFIG_REPLY

Controller role change request messages
- **24** ROLE_REQUEST
- **25** ROLE_REPLY

Asynchronous message configuration
- **26** GET_ASYNC_REQUEST
- **27** GET_ASYNC_REPLY
- **28** SET_ASYNC

Meters and rate limiters configuration
- **29** METER_MOD

Session setup and maintenance

- An OF switch may contain default flow entries to use before connecting with a controller
- The switch will boot into a special failure mode
- An OF switch is usually pre-configured with the IP address of a controller
- OF is best run over a secure connection (TLS/SSL), but can be run over unprotected TCP
- **Hello** messages are exchanged between switch and controller upon startup
- **Echo_Request** and **Echo_reply** are used to verify connection liveliness and optionally to measure its latency or bandwidth
  - If a session is interrupted by connection failure
    - the OF switch continues operation with the current configuration
    - Upon re-establishing connection the controller may delete all flow entries

Can OpenFlow architecture scale to large networks ?

- **Switch flows**
  - Existing OF switch table can handle 1000s of flows
  - With multiple tables, or use of switch fabric and memory for basic matching, this grows to 100s of thousands of flows per switch
- **Controller based on commercial server can handle**
  - a single server processor can handle 100Gbps = 150 Mpps (enough to control many 1000s of switches)
- **A single server can handle 1000s to 10,000s of TCP connections**:
  - limitation of about 10K switches per controller
    - solution is slicing to use multiple controllers, (still an open issue)

SDN success story- Google

Google operates two large backbone networks:

- Internet-facing backbone (user traffic): **I-scale**
- Datacenter backbone (internal traffic): **G-scale**

Managing large backbones is hard

- OpenFlow has helped Google to improve backbone performance and reduce backbone complexity and cost
- The two backbones have very different requirements and traffic characteristics
  - I-scale has smooth diurnal pattern
  - G-scale is bursty with wild demand swings , requires complex TE

Since early 2012 G-scale is managed using OpenFlow

Since no suitable OF device was available Google built its own switches from merchant silicon and open source stacks

SDN success story – Google (cont.)

Why did Google re-engineer G-scale ?

The new network has centralized traffic engineering that leads to network utilization is close to 95% !

This is done by continuously collecting real-time metrics

- global topology data
- bandwidth demand from applications/services
- fiber utilization

Path computation simplified due to global visibility and computation can be concentrated in latest generation of servers

The system computes optimal path assignments for traffic flows and then programs the paths into the switches using OpenFlow.

As demand changes or network failures occur the service re-computes path assignments and reprograms the switches

Network can respond quickly and be hitlessly upgraded

OK, so we can virtualize a basic switch…

What else may be useful ?

Network Functions Virtualization

NFV

- Approximately two years after its creation, the concept of Network Functions Virtualization (NFV) has established its pre-eminence in the telecoms operators' scenario:
  - Has already brought together around 150 members in an Industry Specification Group (ISG) within the European Telecommunication Standards Institute (ETSI).

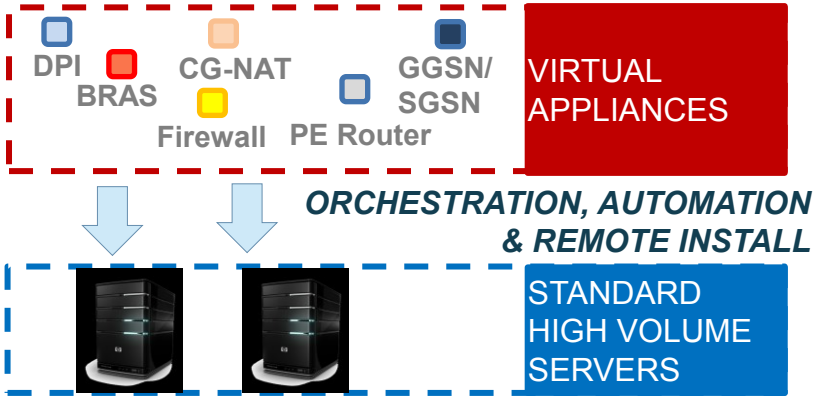
The NFV Concept

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

Traditional Network Model: APPLIANCE APPROACH

- Network Functions are **based on specific HW&SW**
- **One physical node per role**

Virtualised Network Model: VIRTUAL APPLIANCE APPROACH

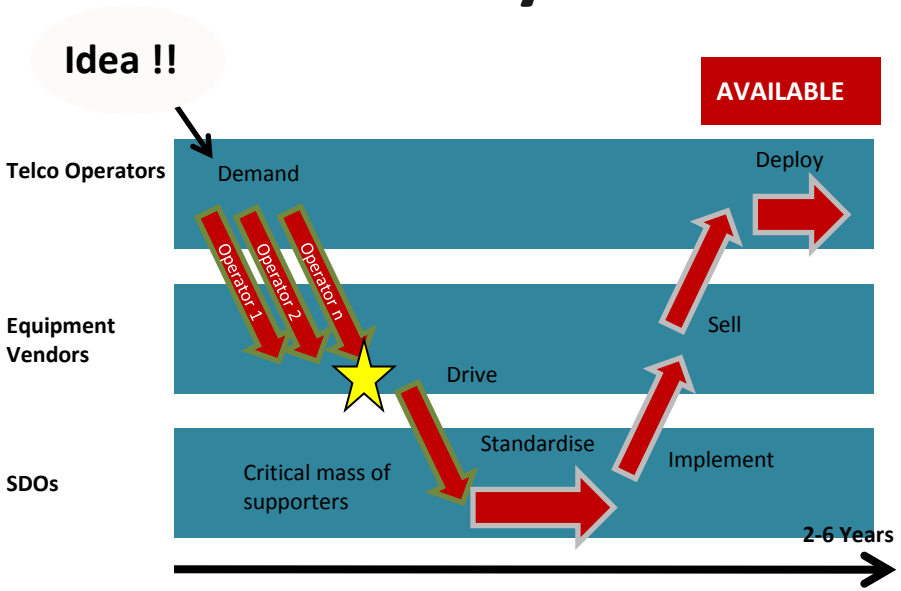


- Network Functions are **SW-based over well-known HW**
- **Multiple roles** over **same HW**

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

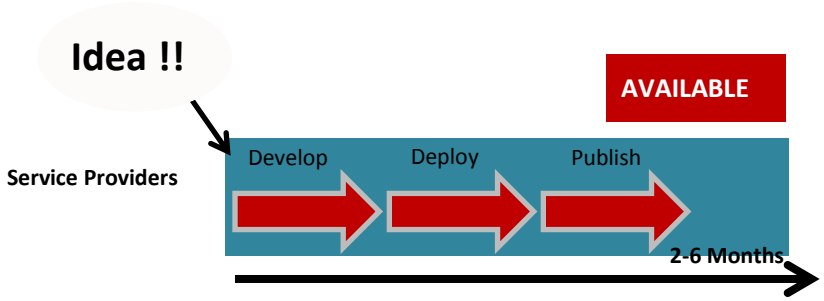
Reduce cycle of innovation

Telco Cycle



2-6 years

Service Providers Cycle



2-6 months

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

Benefits & Promises of NFV

- Network Functions Virtualization is about implementing **network functions in software** - that today run on proprietary hardware - leveraging (high volume) standard servers and IT virtualization
- Supports **multi-versioning and multi-tenancy of network functions**, which allows use of a single physical platform for different applications, users and tenants
- **Flexibility** to easily, rapidly, dynamically provision and instantiate new services in various locations
- **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

So, why we need/want NFV(/SDN)?

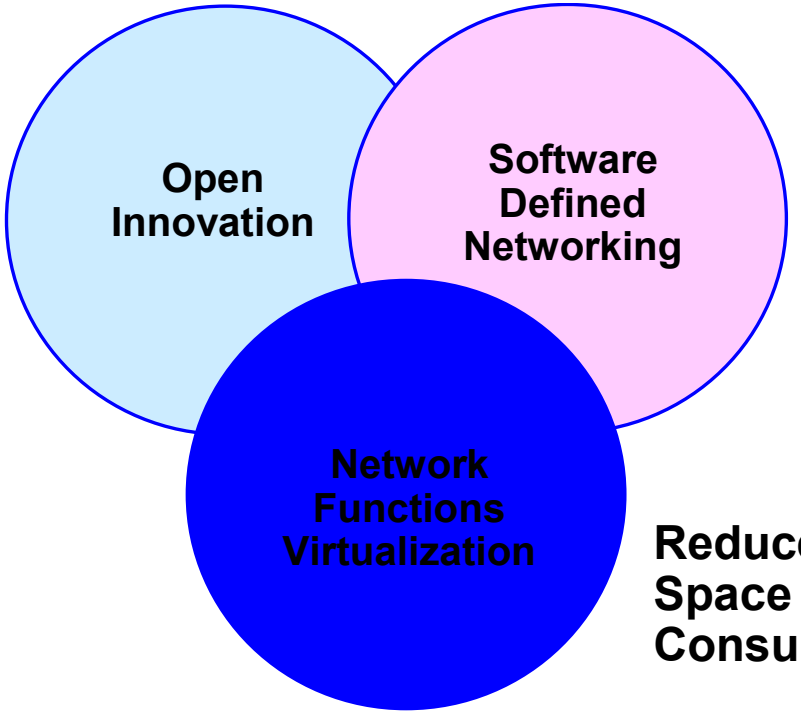
1. **Virtualization:** Use network resource without worrying about where it is physically located, how much it is, how it is organized, etc.
2. **Orchestration:** Manage thousands of devices
3. **Programmable:** Should be able to change behavior on the fly.
4. **Dynamic Scaling:** Should be able to change size, quantity
5. **Automation**
6. **Visibility:** Monitor resources, connectivity
7. **Performance:** Optimize network device utilization
8. **Multi-tenancy**
9. **Service Integration**
10. **Openness:** Full choice of modular plug-ins

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

NFV and SDN

- NFV and SDN are highly complementary
- Both topics are mutually beneficial but not dependent on each other

Creates competitive supply of innovative applications by third parties



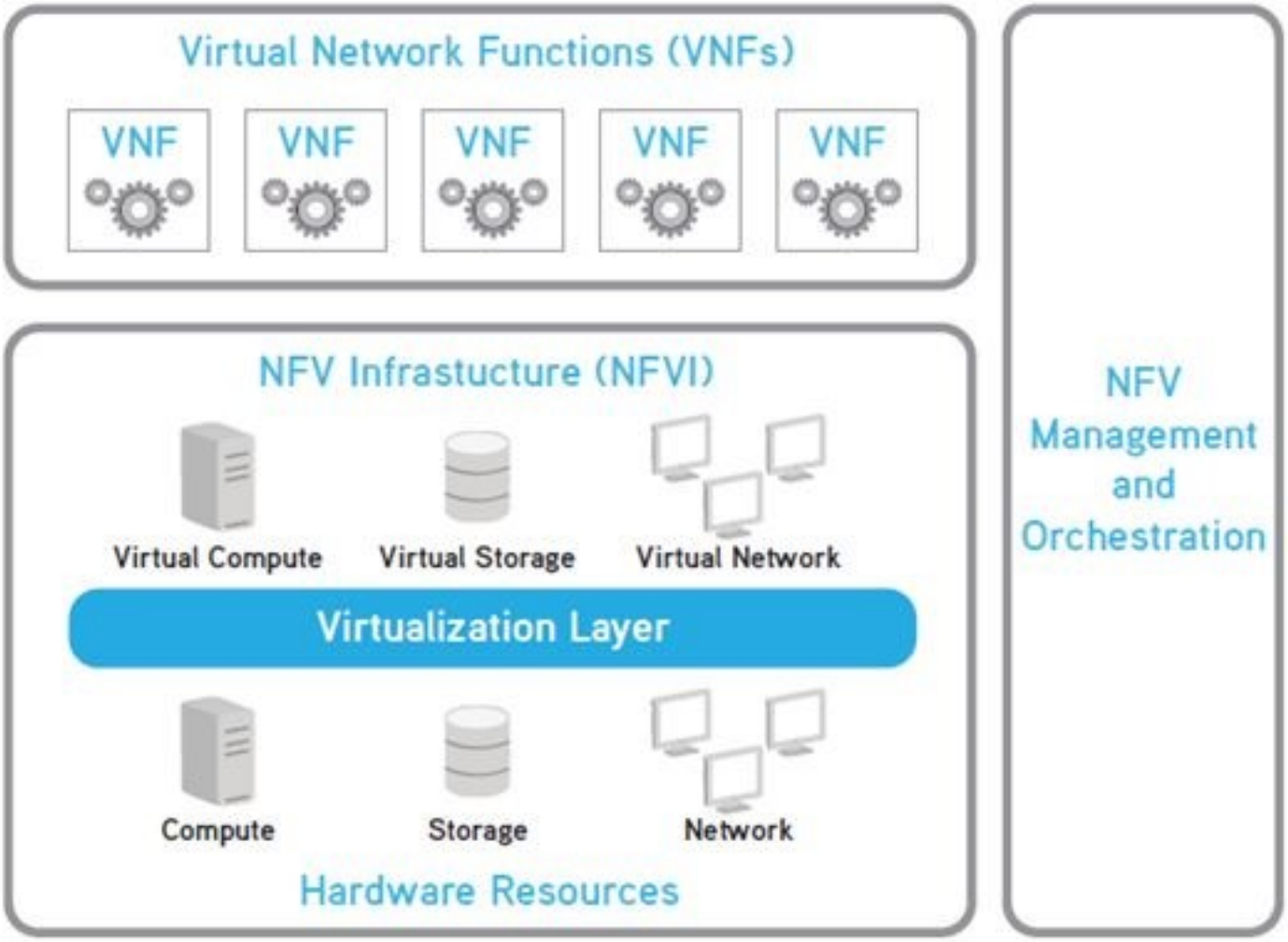
Creates network abstractions to enable faster innovation

Reduces CAPEX, OPEX, Space & Power Consumption

NFV vs SDN

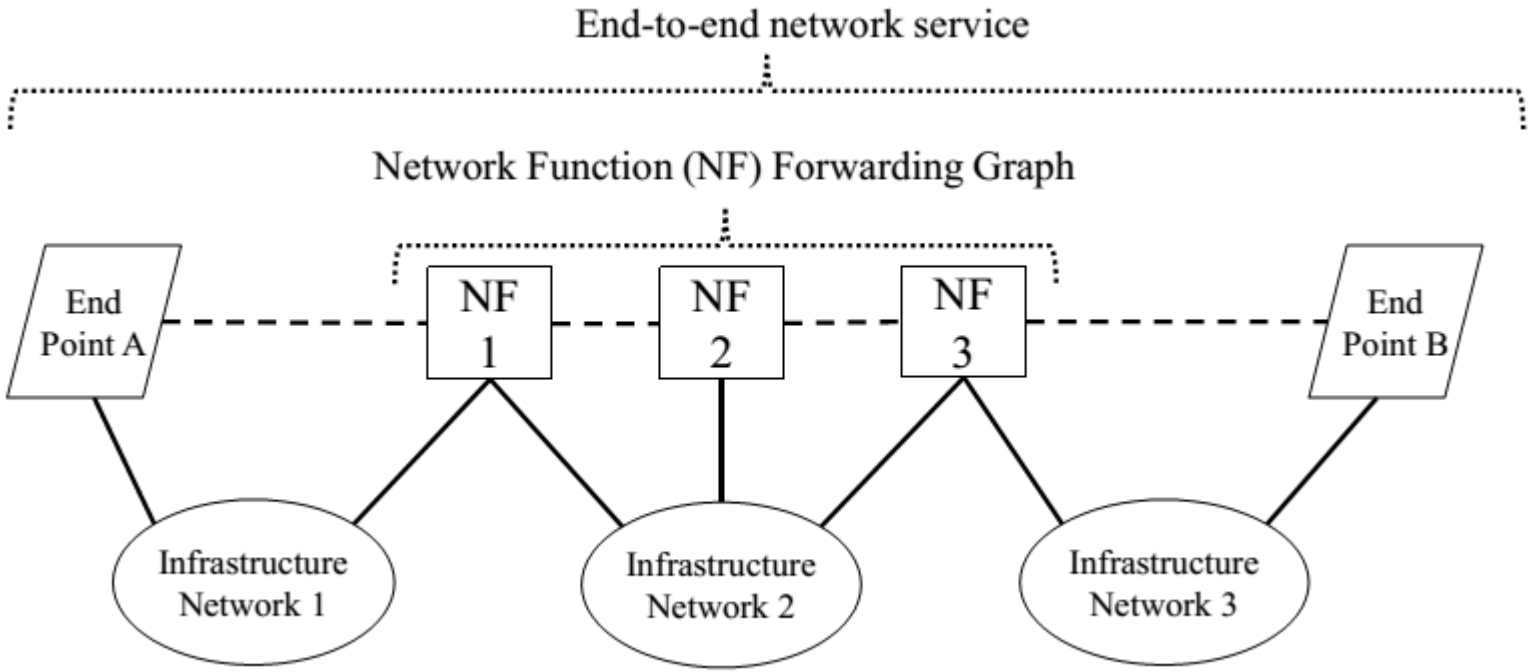
- **NFV: re-definition of network equipment architecture**
  - NFV was born to meet Service Provider (SP) needs:
    - Lower CAPEX by reducing/eliminating proprietary hardware
    - Consolidate multiple network functions onto industry standard platforms
- **SDN: re-definition of network architecture**
  - SDN comes from the IT world:
    - Separate the data and control layers, while centralizing the control
    - Deliver the ability to program network behavior using well-defined interfaces
- **Both have similar goals but approaches are very different**
  - SDN needs new interfaces, control modules, applications.
  - NFV requires moving network applications from dedicated hardware to virtual containers

High level NFV framework

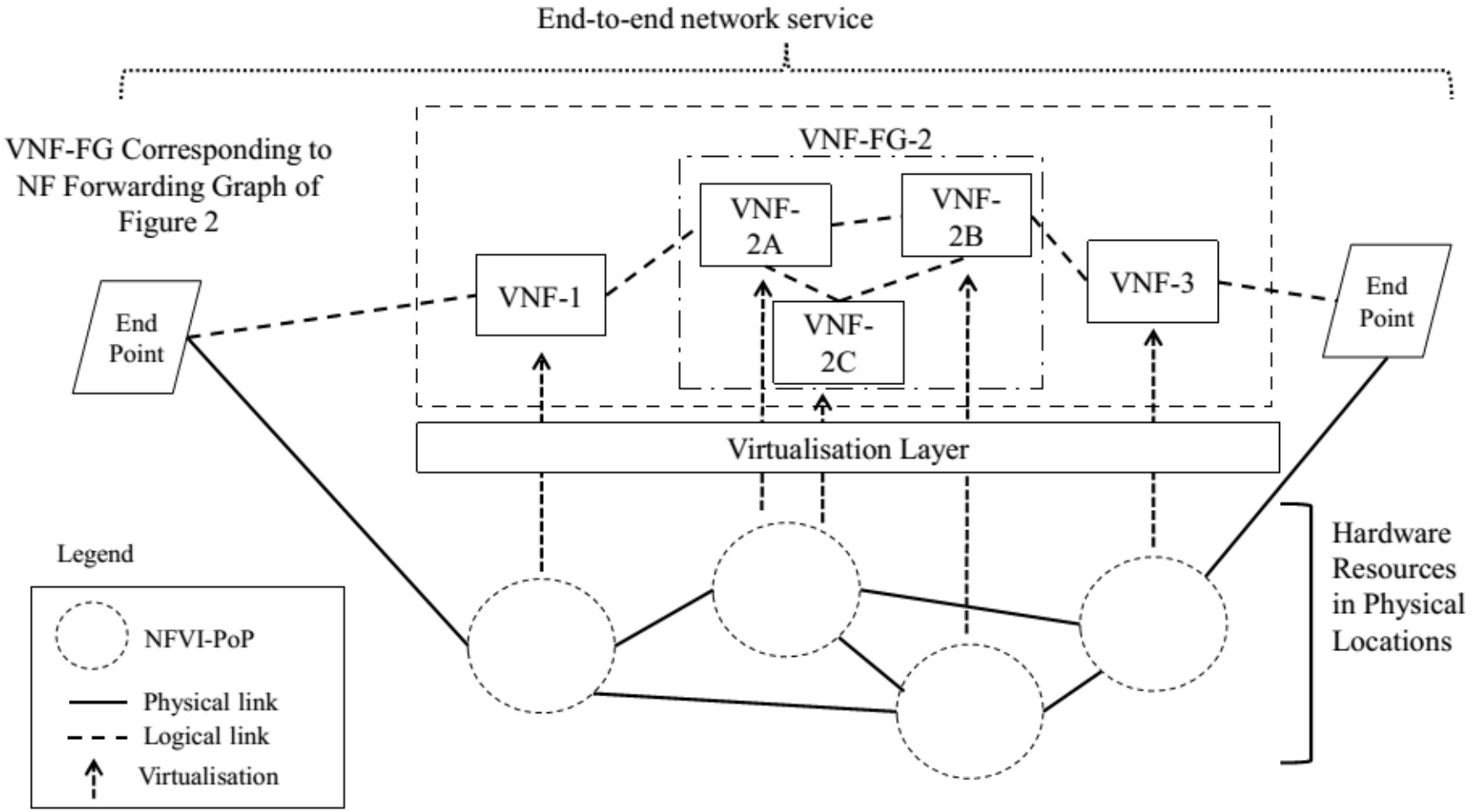


Network Services in NFV

An end-to-end network service (e.g. mobile voice/data, Internet access, a virtual private network) can be described by an NF Forwarding Graph of interconnected Network Functions (NFs) and end points.



End to End Network Service



Example with VNFs and nested forwarding graphs

NFV Challenges

- **Achieving high performance virtualised network appliances**
  - portable between different HW vendors, and with different hypervisors
- **Management and orchestration of virtual network appliances**
  - ensuring security from attack and misconfiguration
  - appropriate level of resilience to HW and SW failure
- **Integrating multiple virtual appliances from different vendors**
  - Network operators need to be able to “mix & match” HW,
  - hypervisors and virtual appliances from different vendors,
  - without incurring significant integration costs and avoiding lock-in.

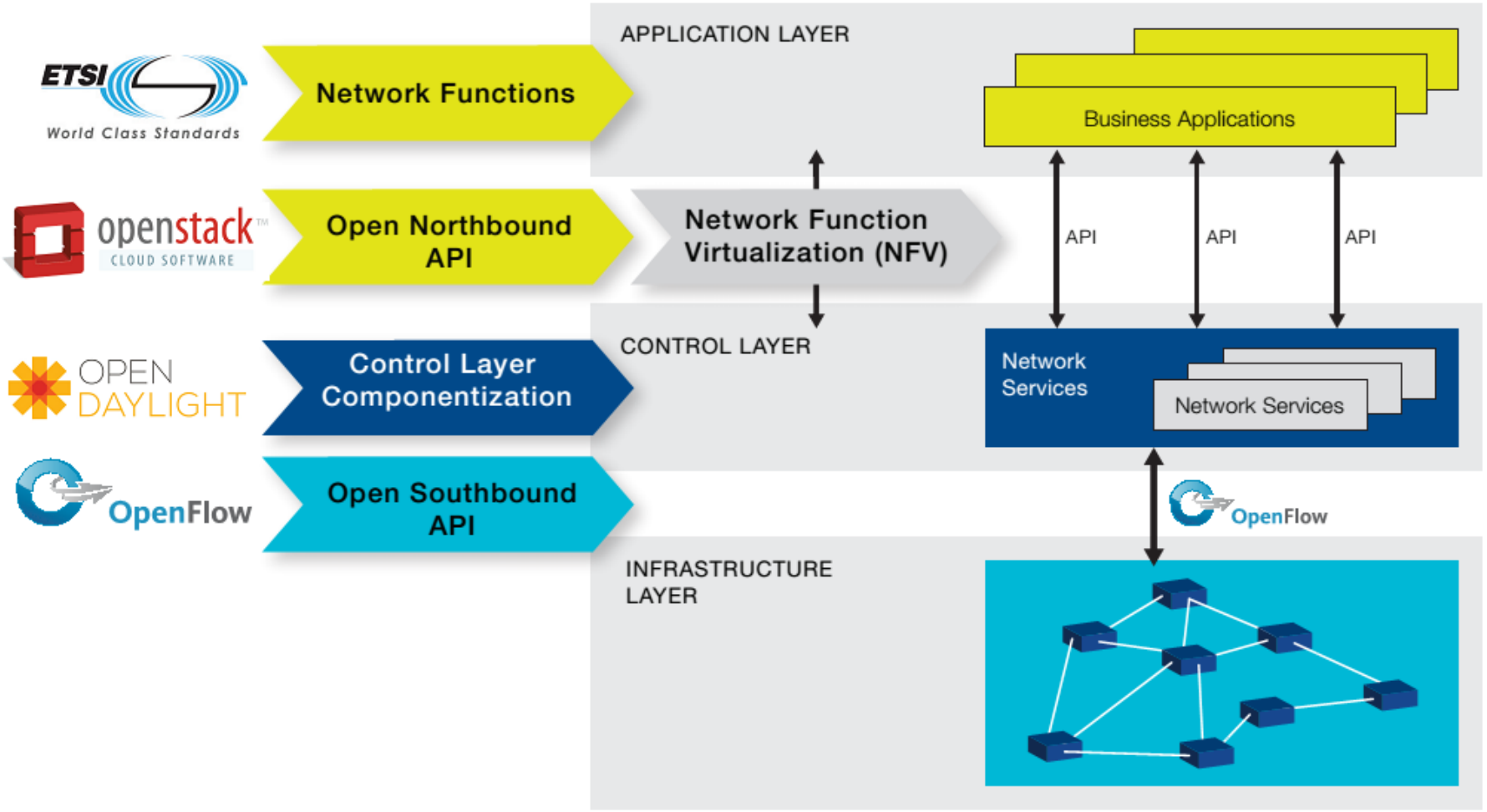
OpenFlow-enabled SDN and Network Functions Virtualization

Source: ONF Solution Brief February 17, 2014

NFV and SDN

- NFV is intended to optimize the deployment of network functions (such as firewalls, DNS, load balancers, etc.),
- OpenFlow-based SDN is focused on optimizing the underlying networks.
- ONF is undertaking SDN standardization. The ETSI NFV ISG is not a standards body, but rather will produce requirements that network operators can adapt for their individual environments.
- Both bodies are driven by a strong end-user culture.

NFV and SDN


ETSI
World Class Standards
Network Functions
openstack
CLOUD SOFTWARE
Open Northbound API
OPEN DAYLIGHT
Control Layer Componentization
OpenFlow
Open Southbound API
APPLICATION LAYER
Business Applications
Network Function Virtualization (NFV)
API
API
API
CONTROL LAYER
Network Services
Network Services
OpenFlow
INFRASTRUCTURE LAYER

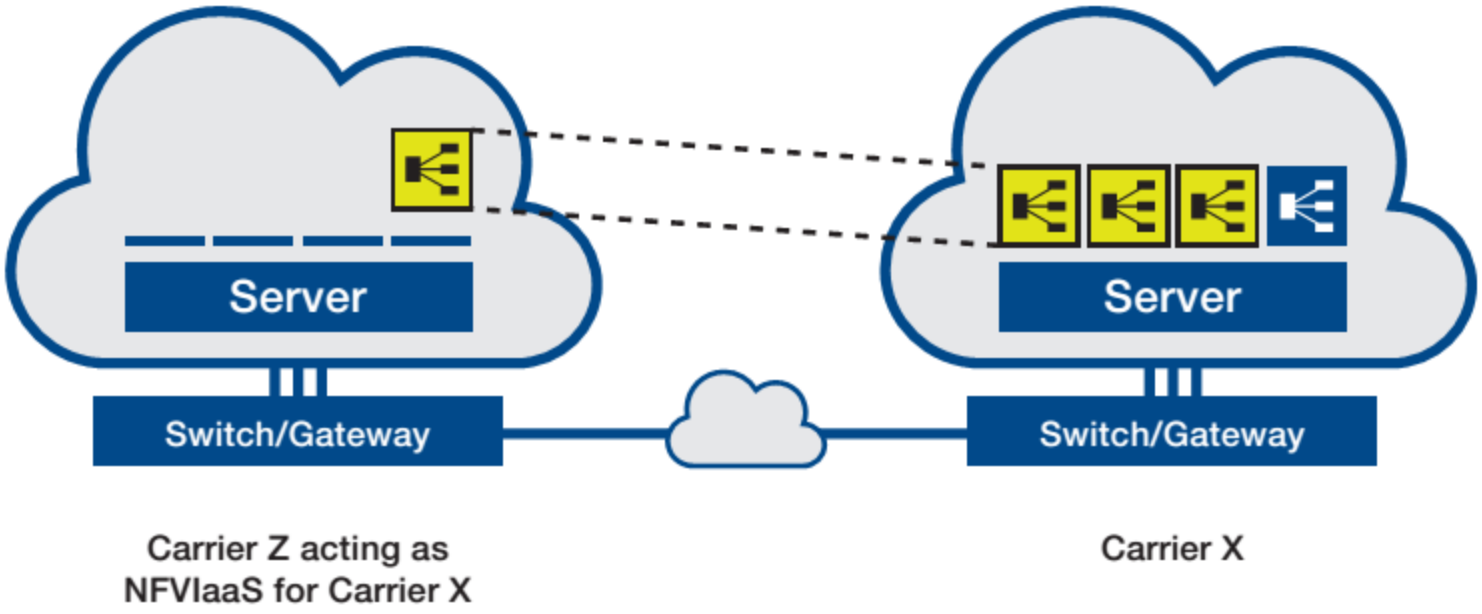
NFV
use cases

NFV INFRASTRUCTURE AS A SERVICE

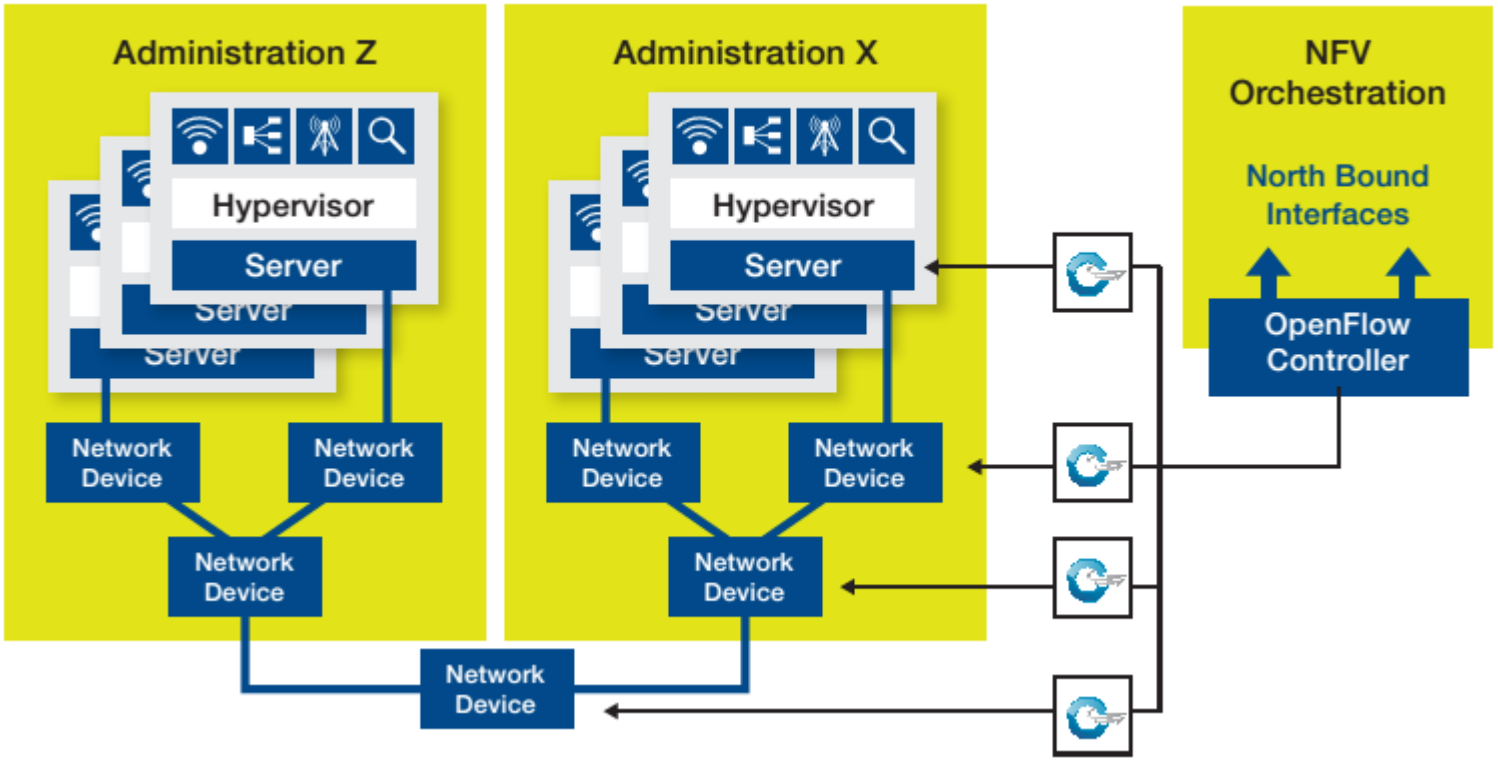
- For the delivery of cloud services:
  - One service provider can offer services using the NFV infrastructure (NFVI) of another service provider.
  - Example: service provider X offers a virtualized load balancing service. Some of carrier X's customers need load balancing services at locations where that company doesn't maintain NFVI, but where service provider Z does.

NFV INFRASTRUCTURE AS A SERVICE

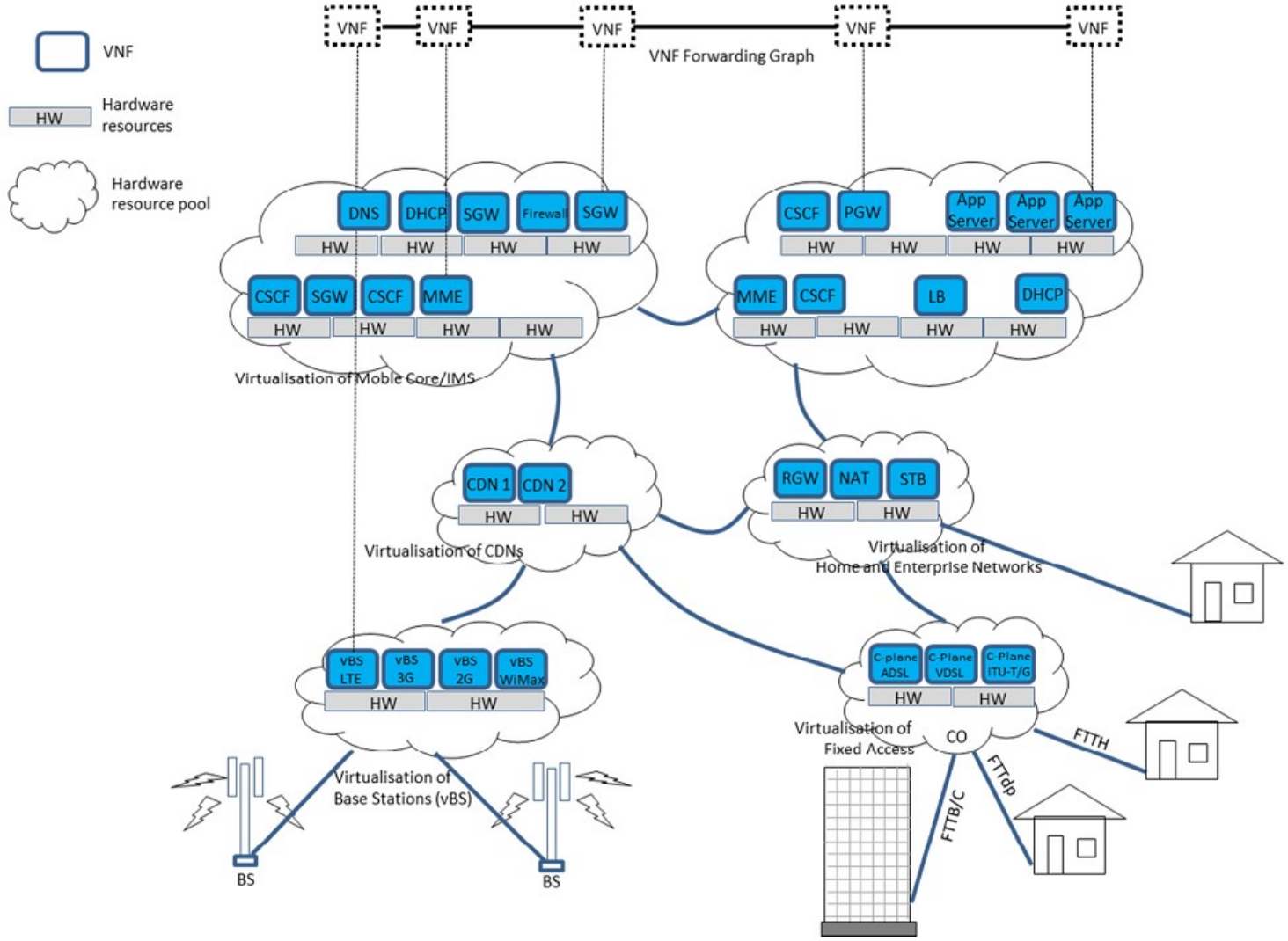
- NFVIaaS offers a means for carrier Z to lease NFV infrastructure (compute, network, hypervisors, etc.) to service provider X, which gives the latter access to infrastructure that would otherwise be prohibitively expensive to obtain.
- Through leasing, such capacity is available on demand, and can be scaled as needed.



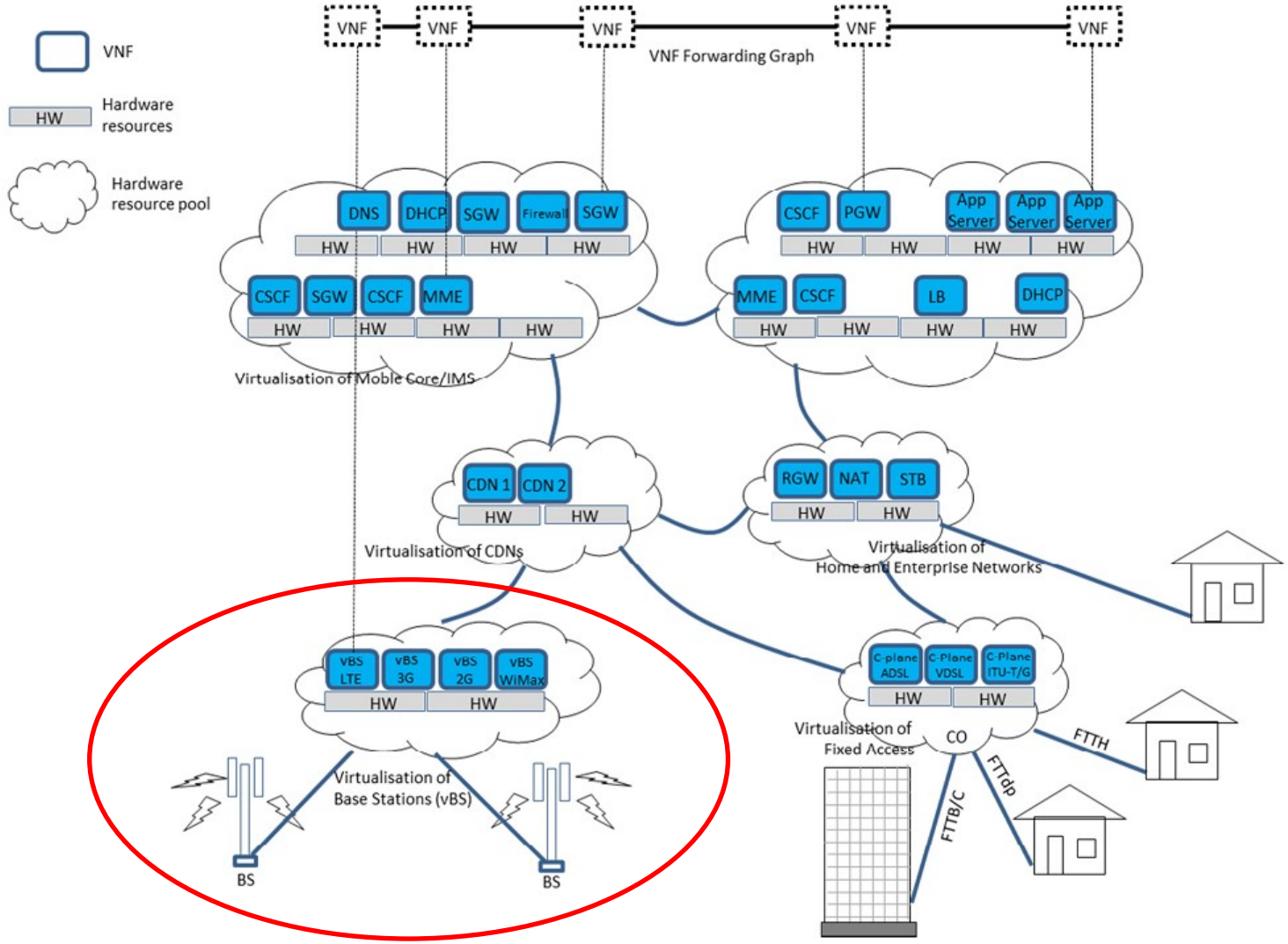
OpenFlow-enabled SDN: A Flexible NFV Networking Solution



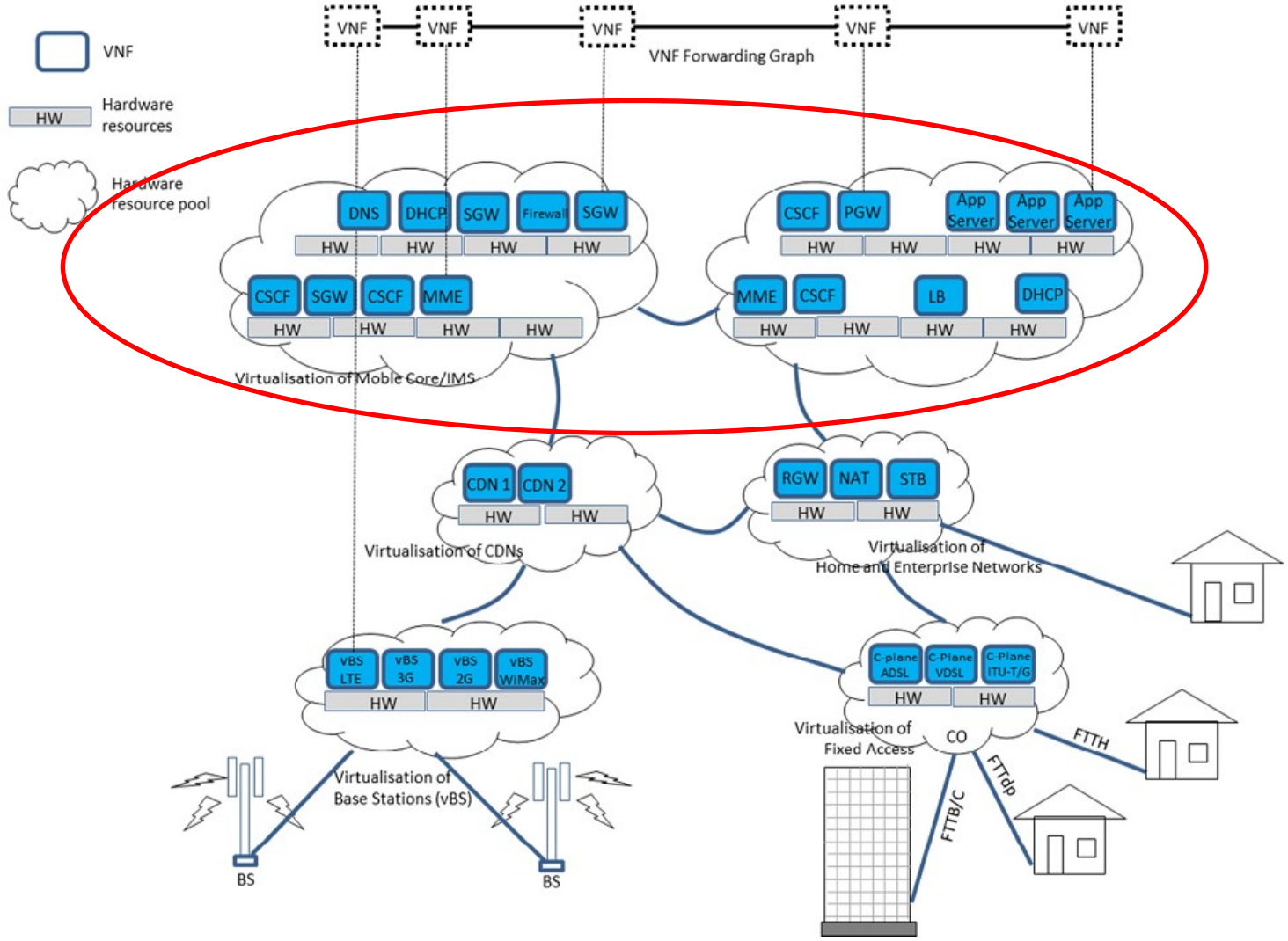
NFV: Use Cases



NFV: Use Cases



NFV: Use Cases



Virtualization of Mobile Core Network

Possible advantages:

- **Reduced TCO** (Total Cost of Ownership).
- **Improved network usage efficiency** due to flexible allocation of different Network Functions on such hardware resource pool.
- **Higher service availability** and resiliency provided to end users/customers by dynamic network reconfiguration inherent to virtualization technology.
- **Elasticity**: Capacity dedicated to each Network function can be dynamically modified according to actual load on the network, thus increasing scalability.
- **Topology reconfiguration**: Network topology can be dynamically reconfigured to optimize performances.

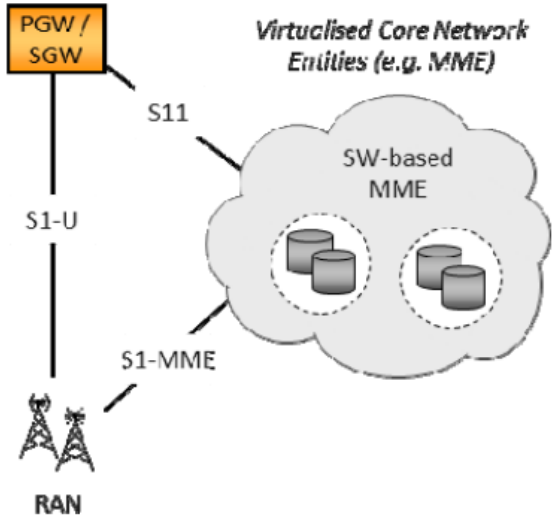
Virtualization of Mobile Core Network

Virtualization of Mobile Core Network Functions:

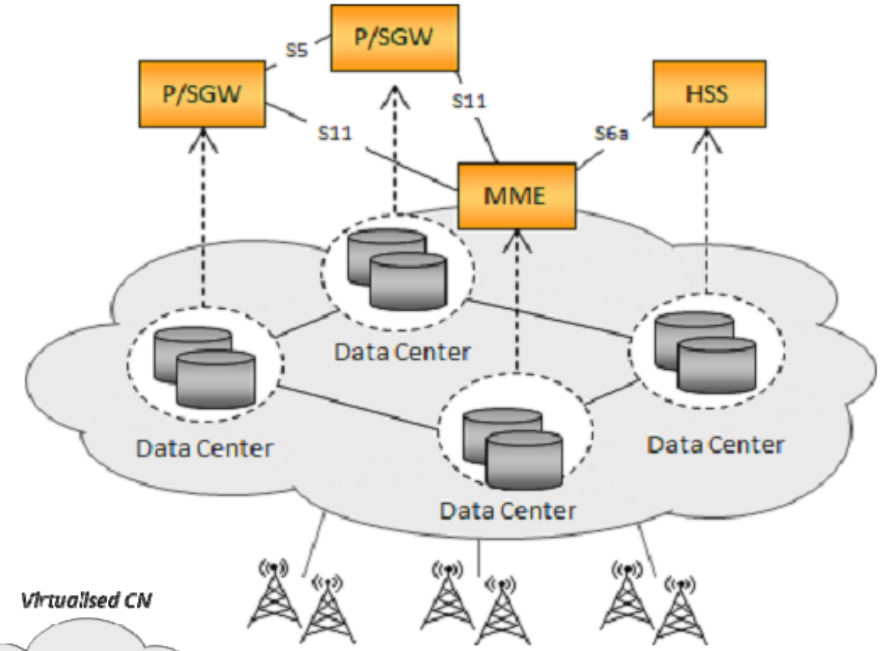
- EPC Core & Adjunct Network Functions e.g. MME, S/P-GW, PCRF, etc.
- 3G/EPC Interworking Network Functions e.g. SGSN, GGSN, etc.

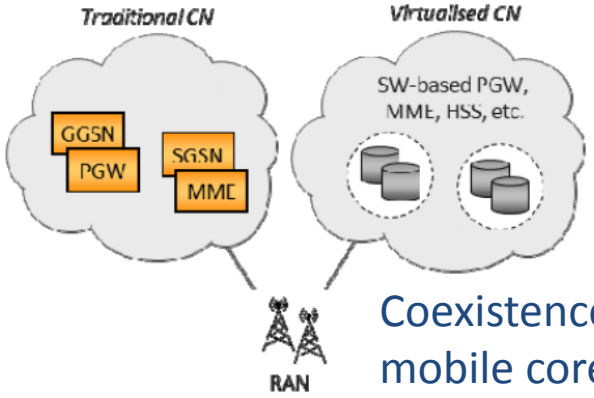
Virtualization of EPC

- Different scenarios may be enabled where, for example, the entire EPC is Virtualised in a single NFVI-PoP or only some NFs are Virtualised



Partial virtualization of the core network





Coexistence of Virtualized and non Virtualized mobile core network:

Virtualization of base stations

- **A RAN node utilization is usually lower** than its MAX capacity because the system is designed to cover the peak load
- Base Station virtualization can **achieve sharing of resources** among multiple logical RAN nodes from different systems, dynamically allocating the resource
- **Centralized-RAN** (C-RAN) technology with virtualization can leverage more efficient resource utilization among different physical BSs

Cloud-RAN

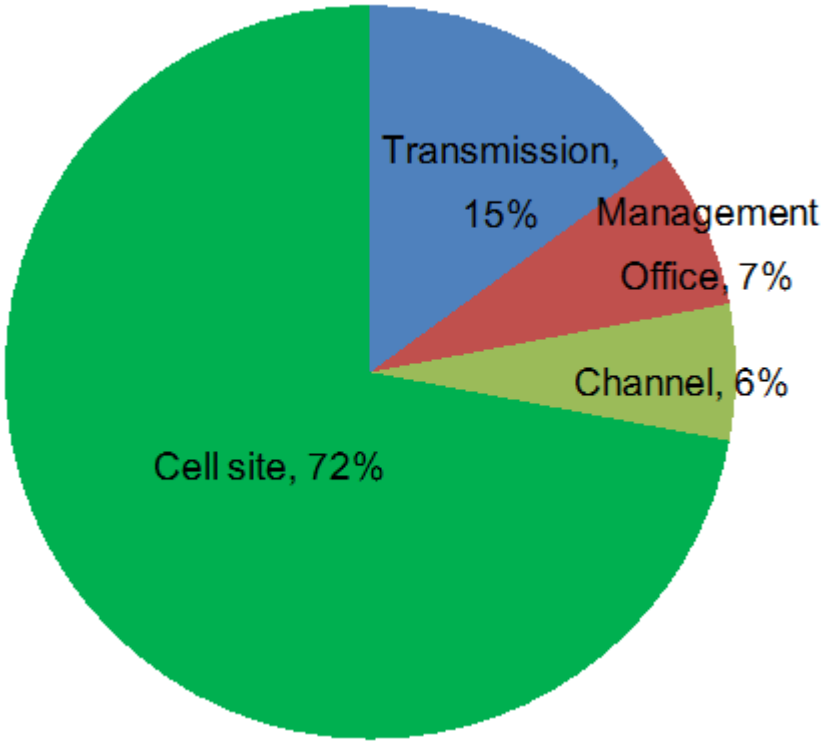
- Cloud-RAN is a new cellular network architecture for the future mobile network infrastructure.
- It was first introduced by China Mobile Research Institute in April 2010 in Beijing, China.
  - C-RAN is a centralized, cloud computing based new radio access network (commonly known as cellular network) architecture that can support 2G, 3G, 4G system and future wireless communication standards.

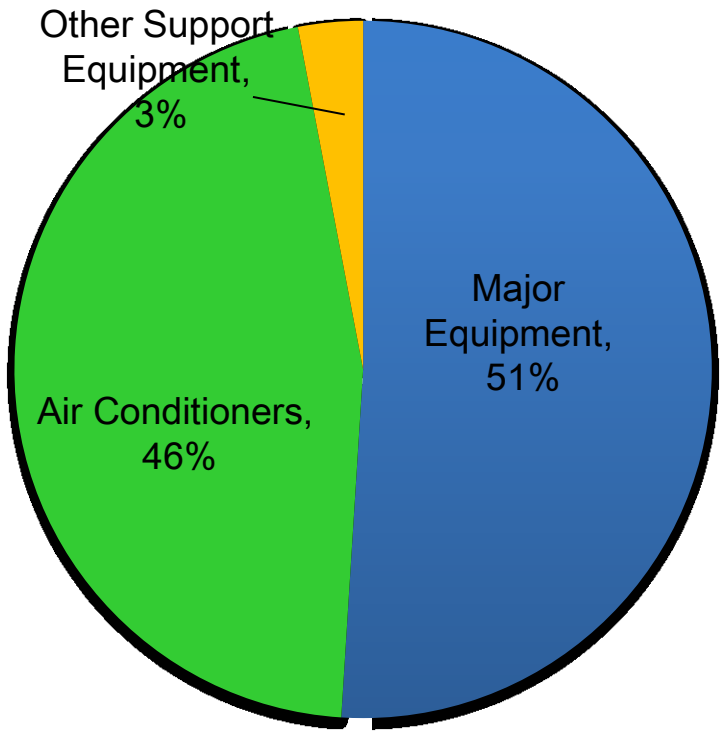
C-RAN Motivations

- Limitations of traditional cellular architectures:
  - **each BTS is costly to build and operate.**
  - when more BTS are added to the system to improve capacity:
    - Increasing of interference among BTS
  - because mobile users are moving from one place to another, the traffic of each BTS is fluctuating very much from time to time, **(tide effect):**
    - **the average utilization rate of individual BTS is pretty low.**
    - these processing resources cannot be shared with other BTS.
      - Thus all the BTS must be designed to handle the maximum traffic expected no matter the size of the average traffic.

A lot of waste of processing resources and waste of powers at idle times

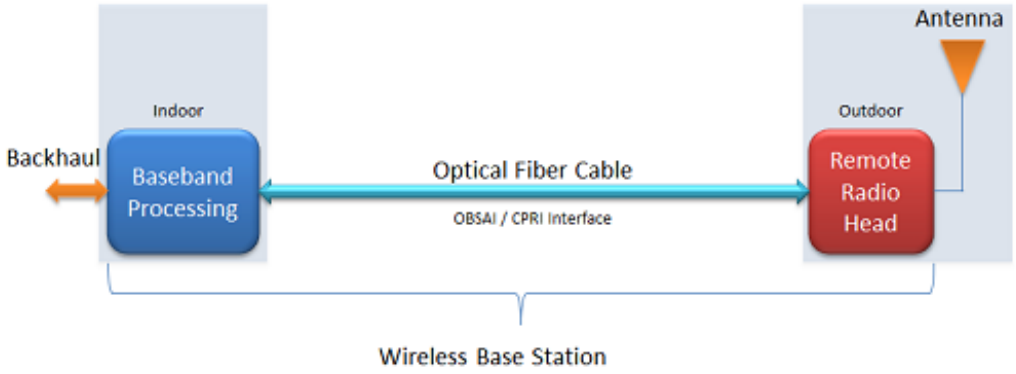
BS cell site is the major source of power consumption.





*Source: Base on China Mobile survey on commercial networks

Base Station Architecture Model

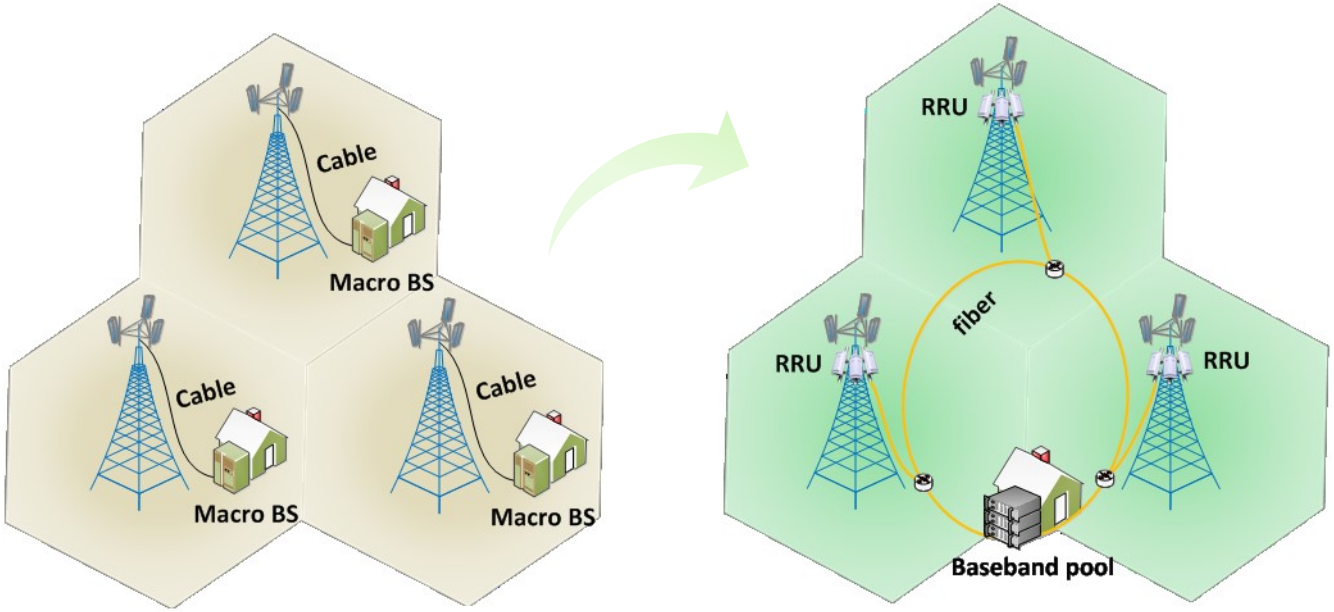


- the radio function unit, remote radio head (RRH), is separated from the digital function unit, the baseband unit (BBU) by fiber.
  - The RRH can be installed on the top of tower
  - The fibre link between RRH and BBU allows much more flexibility in network planning and deployment as they can be placed a few hundreds meters or a few kilometres away now.

C-RAN approach

- It makes use of the latest CPRI standard, low cost CWDM/DWDM technology, or mmWave to allow transmission of baseband signal over long distance thus achieving large scale centralized base station deployment
- It applies recent Data Centre Network technology to allow a low cost, high reliability, low latency and high bandwidth interconnect network in the BBU pool
- It utilizes open platform and real-time virtualization technology rooted in cloud computing to achieve dynamic shared resource allocation in BBU pool and support of multi-vendor, multi-technology environment.

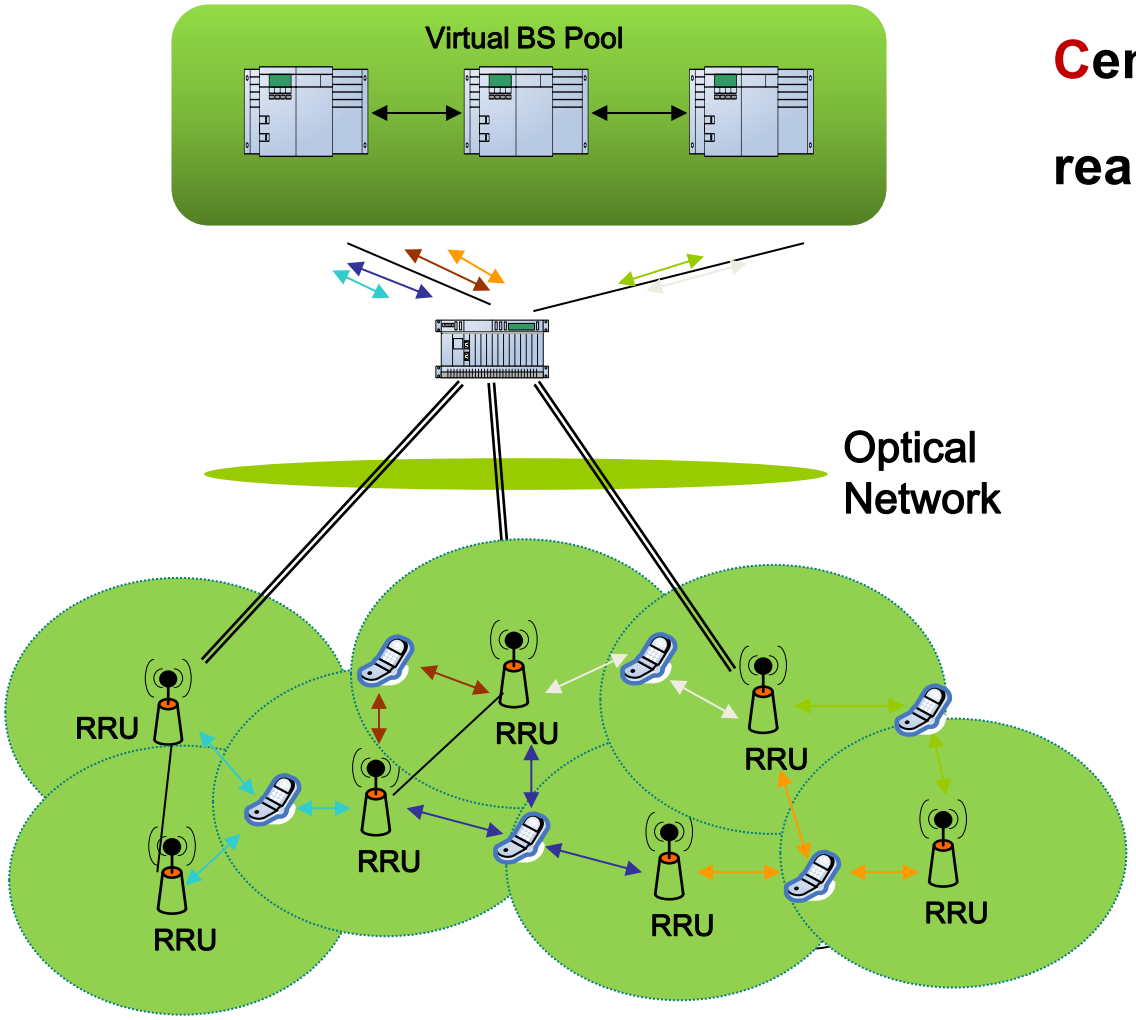
Centralized Baseband Pool



Traditional macro base-stations

C-RAN Centralized baseband pool

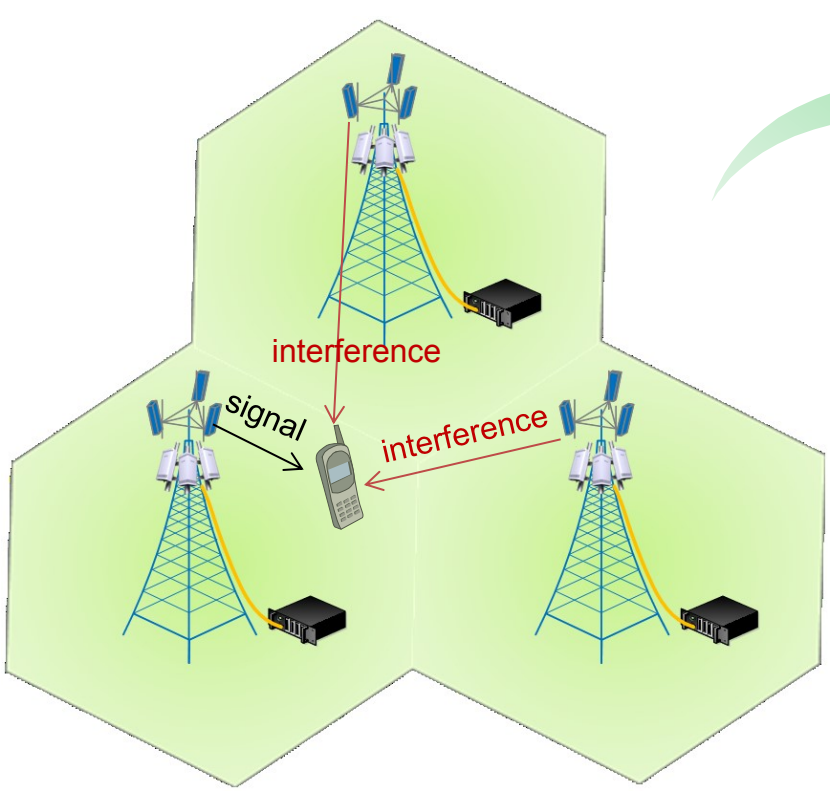
C-RAN



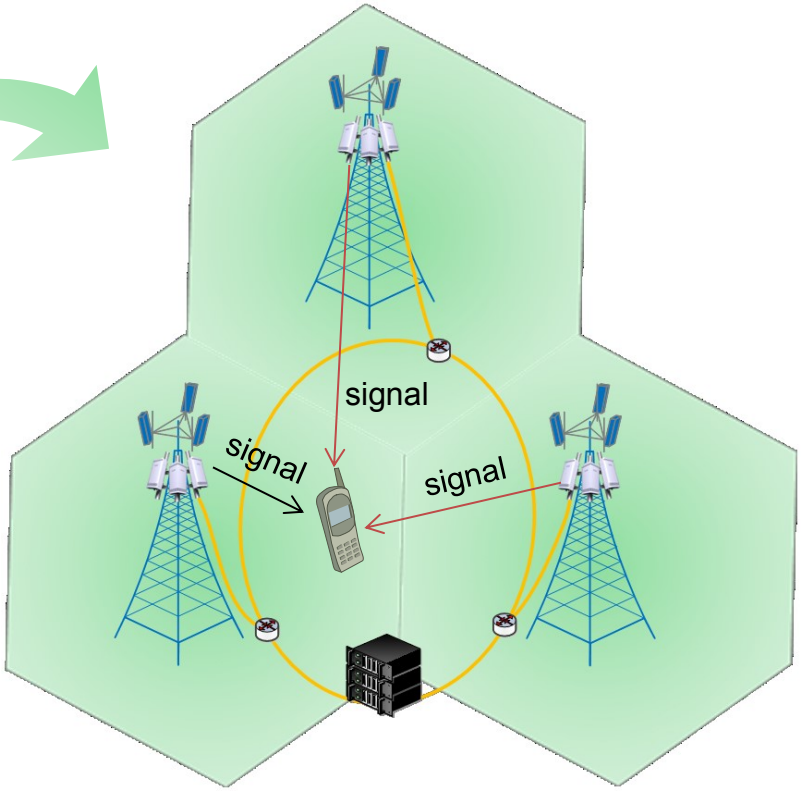
Centralized baseband pool

real time Cloud computing

Collaborative Radio

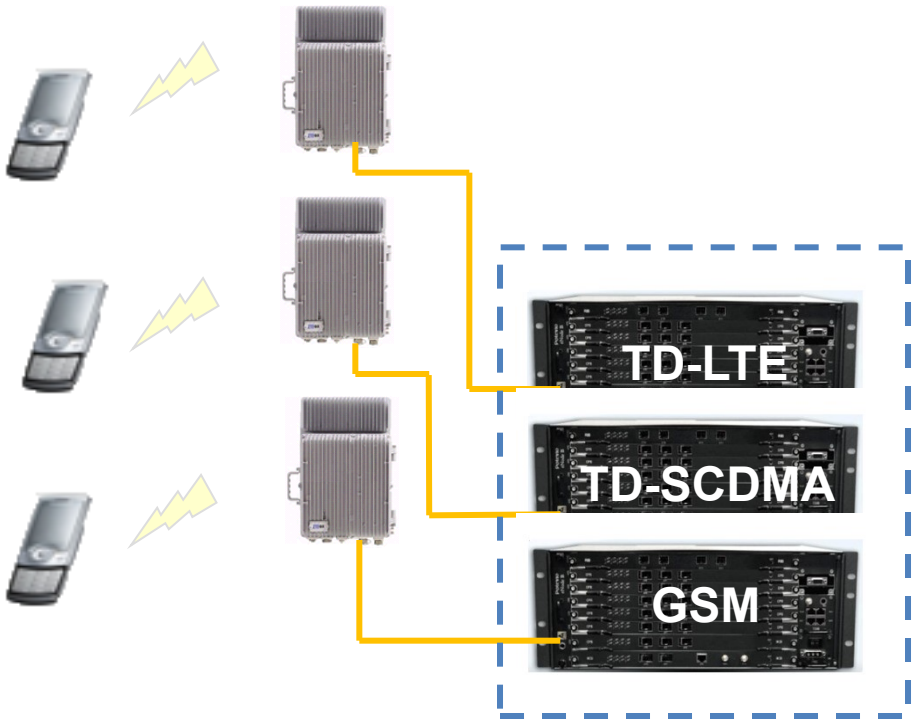


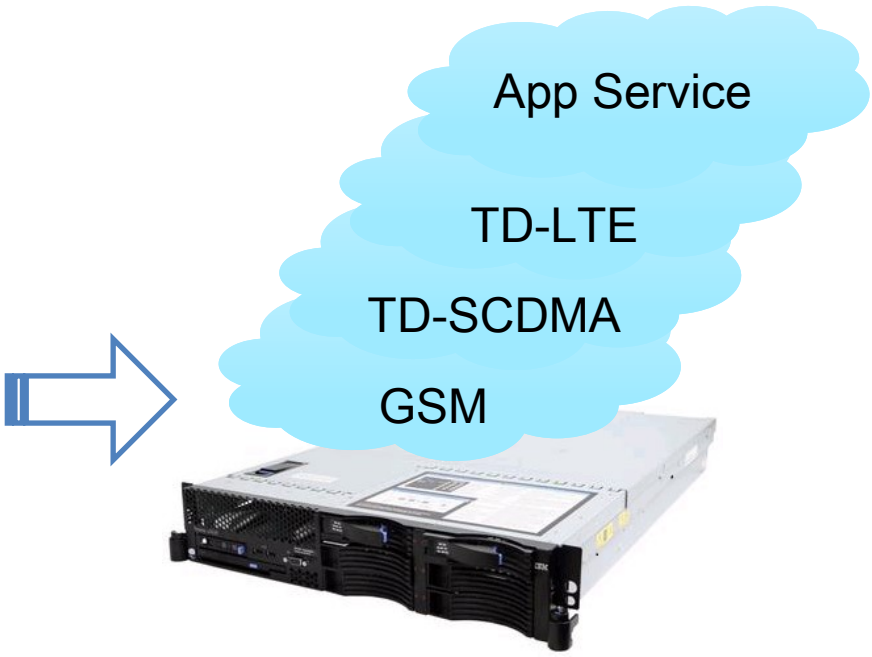
Separately processing



Joint-processing

C-RAN



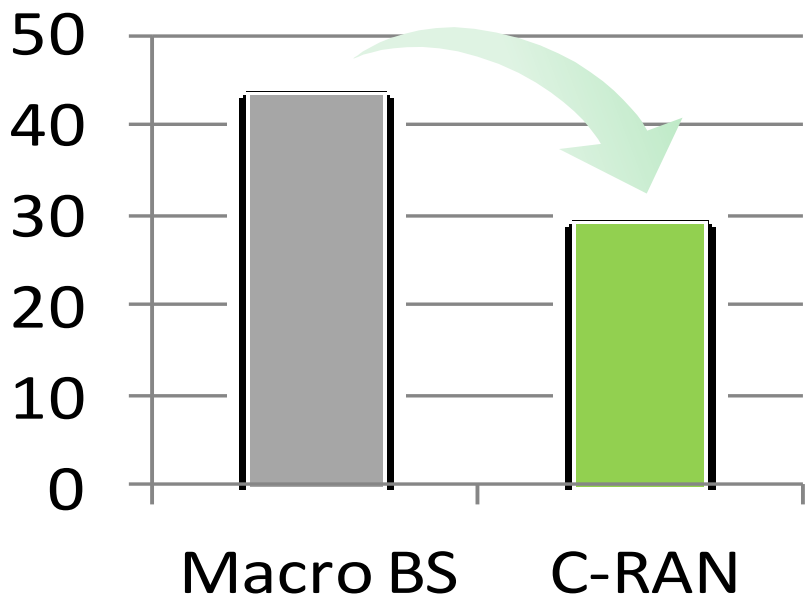


Traditional baseband equipments

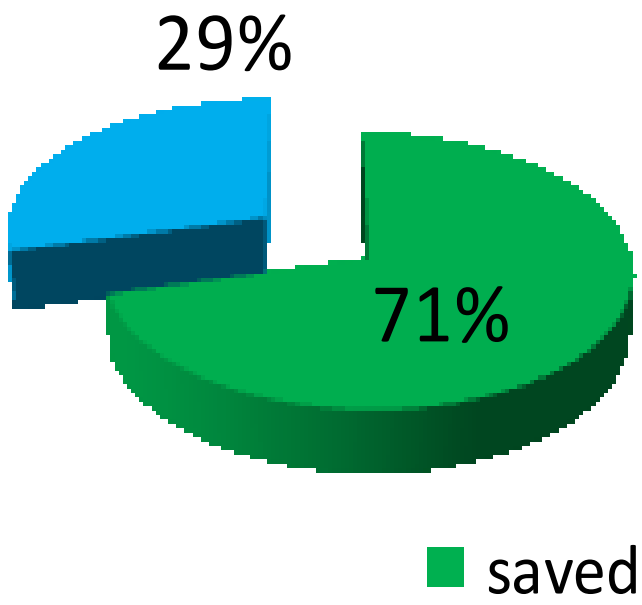
Software Defined Radio
based on
Virtualized Cloud Platform

Clean System Target

Construction cost per site reduced by 1/3



Power consumption reduced by at most 71%



*Source: Base on China Mobile field trial surveys

Cloud-RAN

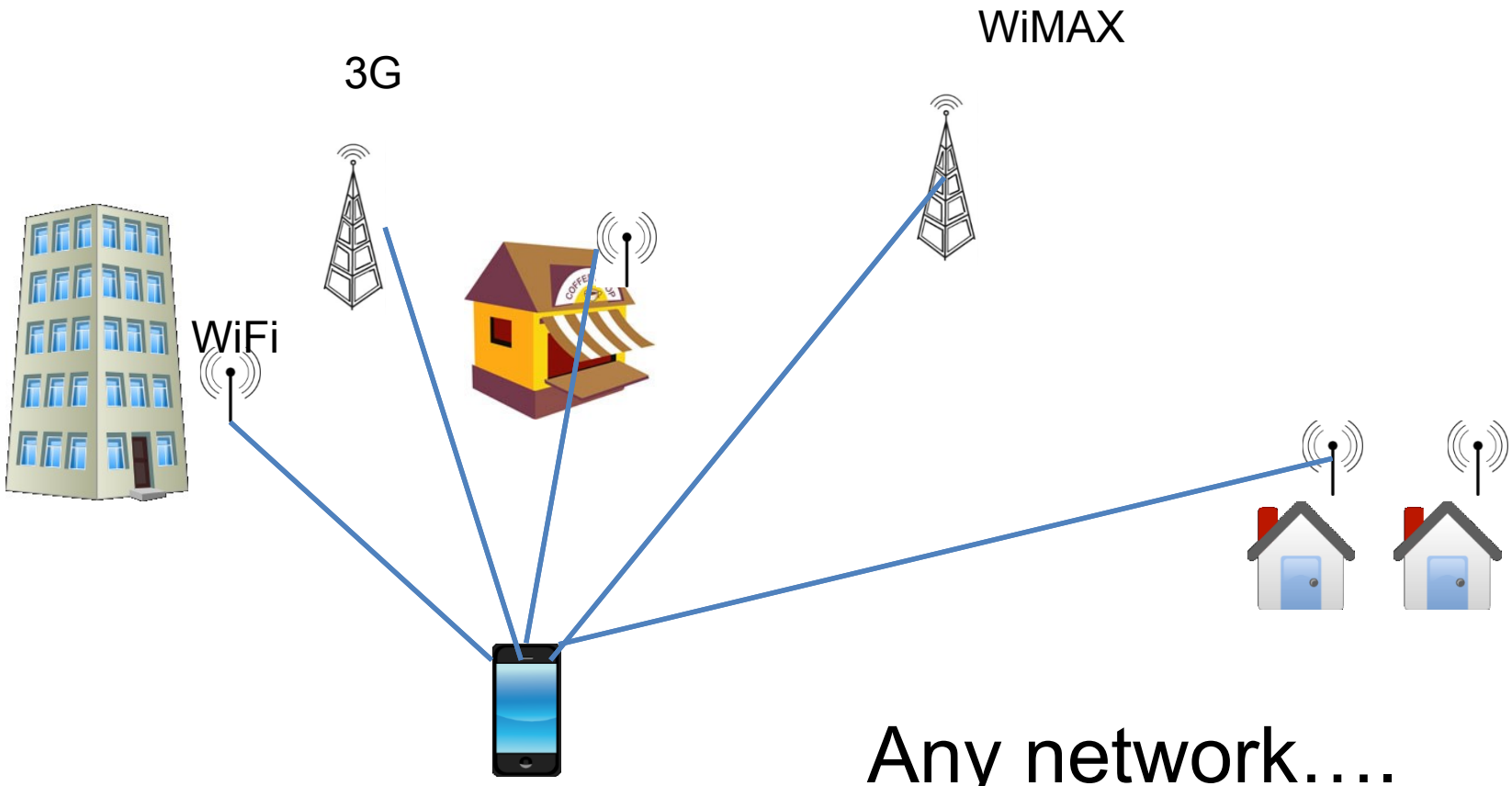
- Separates the computing intensive baseband processing from the remote radio deployments
- Baseband processing is pooled at a semi-centralized location
- C-RAN Enables
  - Use of commodity HW to run baseband processing tasks
  - More fluid resource allocation
  - Enables new feature implementation like CoMP and eICIC
  - Helps ease capacity crunch by placing radios closer to the user

C-RAN concept

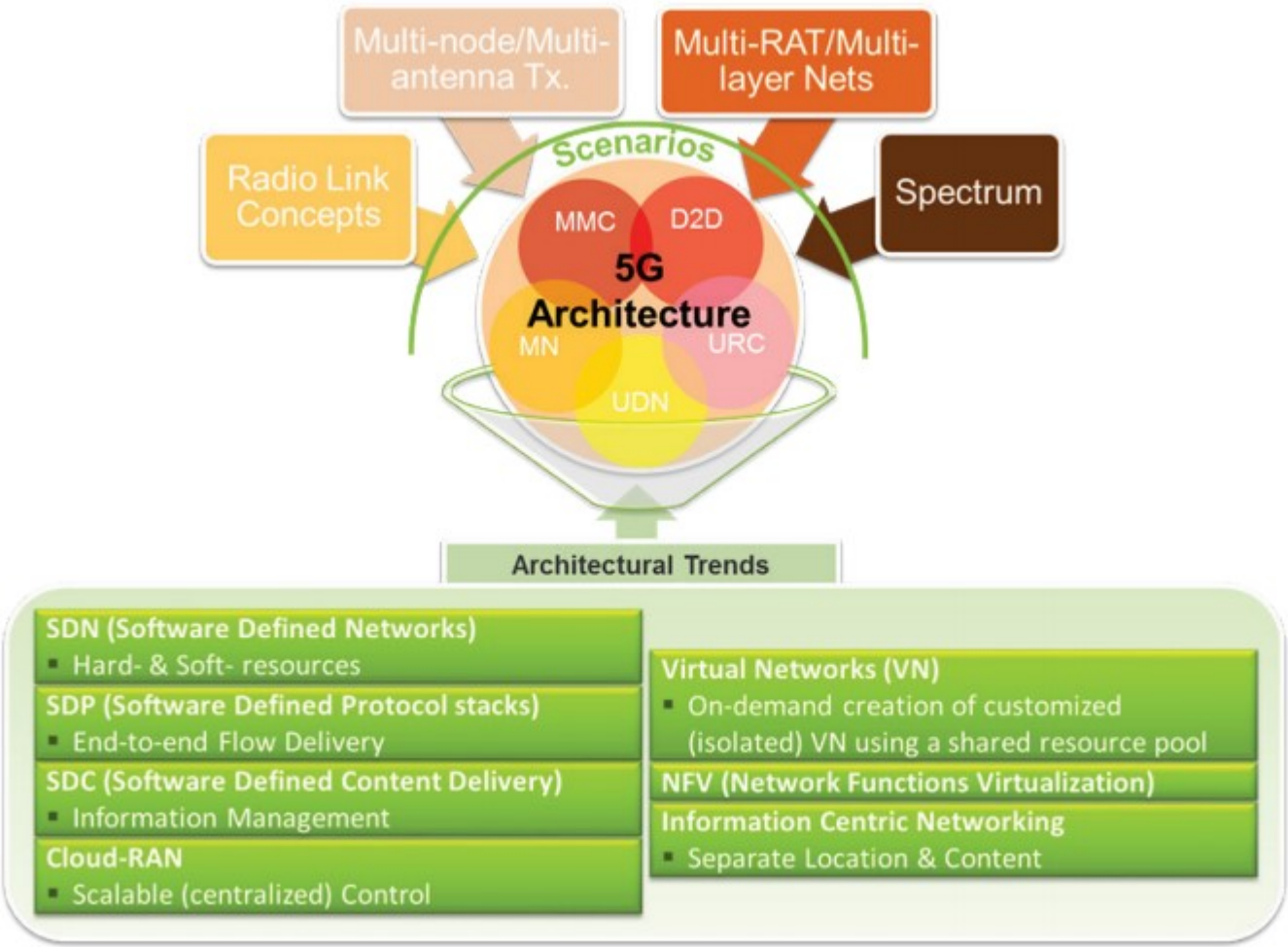
- Services are provided through optimum access technology
- Resources and coverage of a geographical area can be changed dynamically
- SON can be used to get information for providing the necessary configuration
  - Resources are aggregated and dynamically allocated
  - Reconfigurable BSs and controllers to support multiple Radio Access technologies

Radio Access Network Virtualization

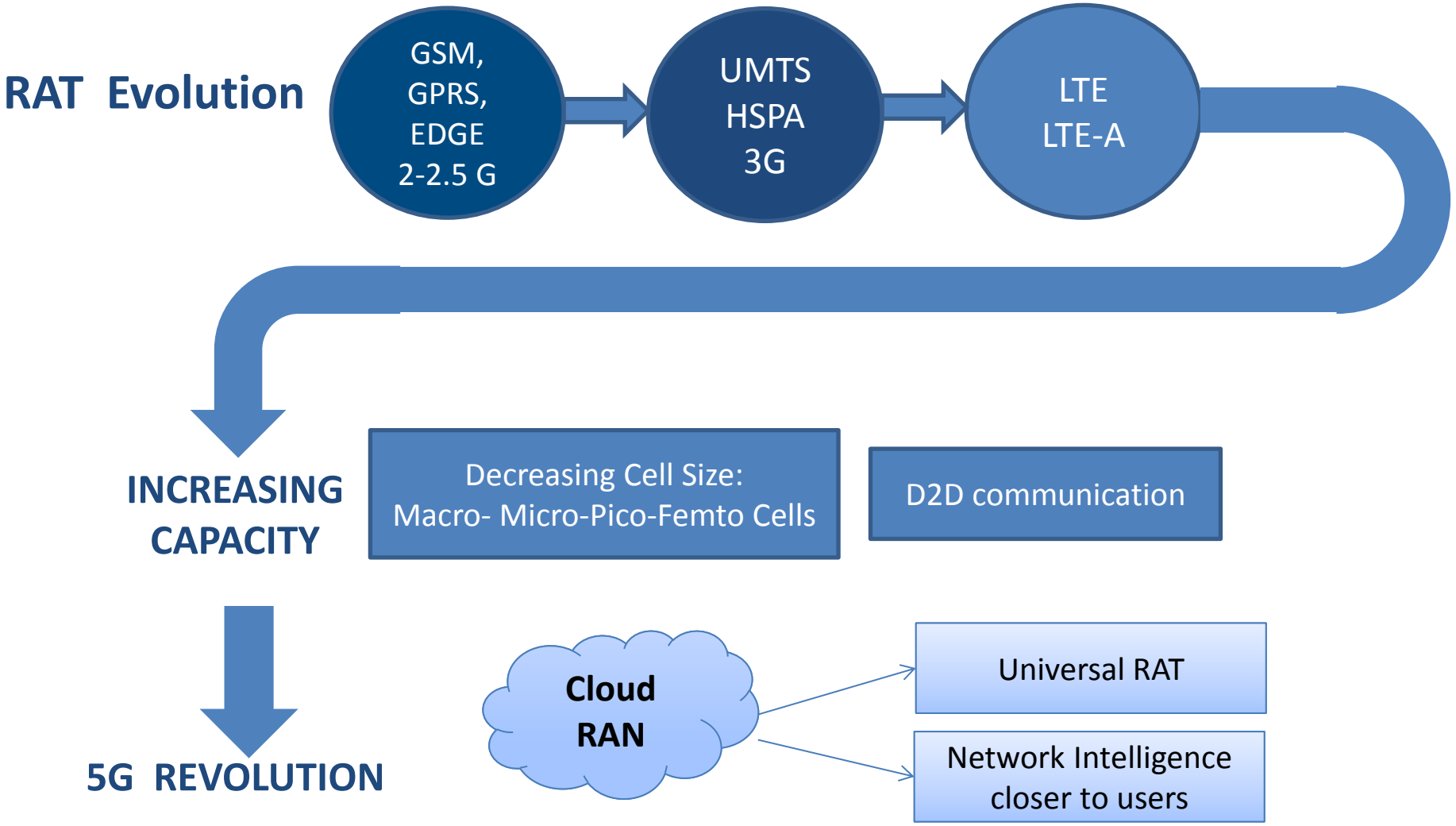
Open the wireless infrastructure so users can choose any free spectrum, any network, or many networks, any time



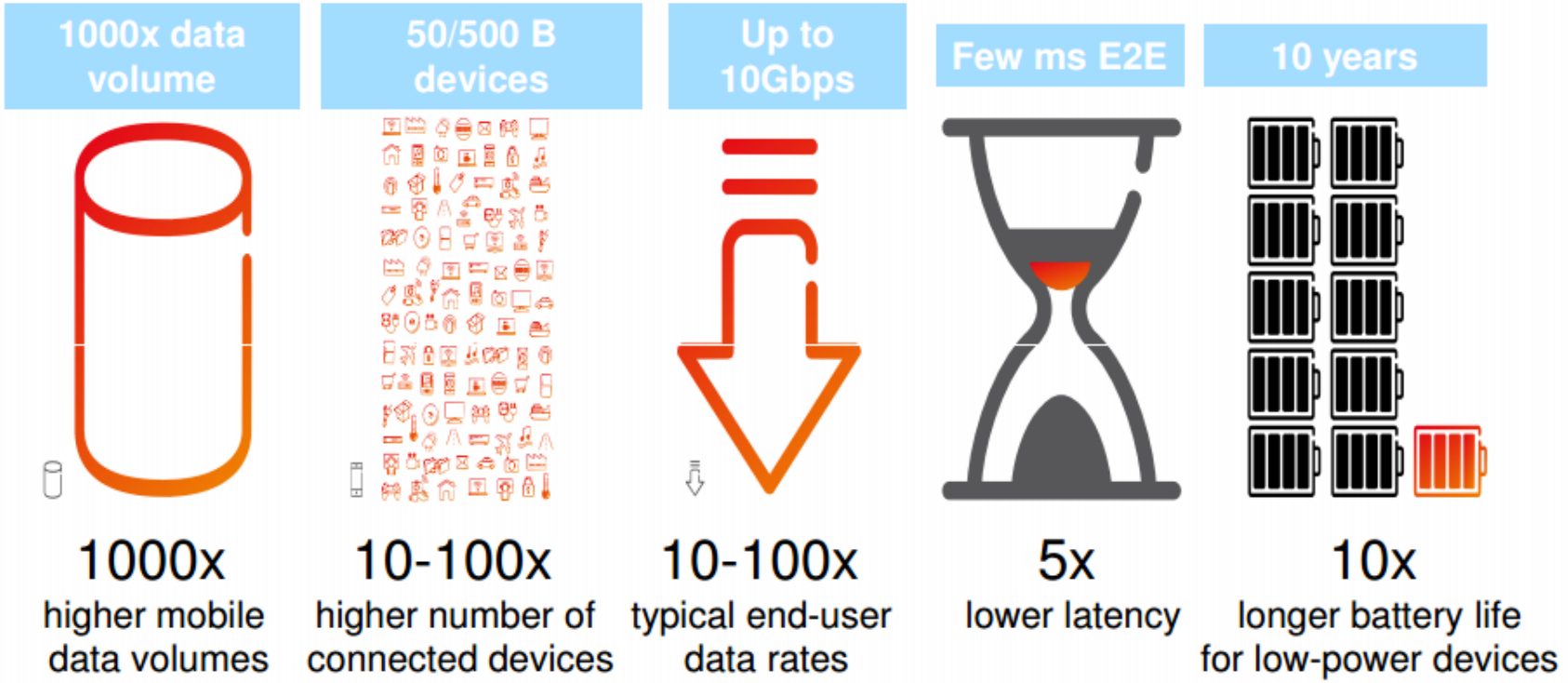
The road towards 5G…



(R)Evolution towards 5G



Are we ready for the 5G objectives?



Summary

- We looked at the concept of virtualization and architectures and paradigms for enabling elasticity in the future networks.
- Network Virtualization offers:
  - transparent abstraction of networking platform and resources
  - multiple logical interpretations of the physical characteristics
  - resource partitioning
  - resource sharing
- Network Functions Virtualization offers the potential to transform carrier/network operator operations while achieving significant agility and cost reduction.
- SDN is emerging as the key enabler for NFV, offering the dynamic behavior, automation, and openness required for carrier networks in the future.

References

1. Chowdhury, N.M.M.K.; Boutaba, R., "Network virtualization: state of the art and research challenges," Communications Magazine, IEEE , vol.47, no.7, pp.20,26, July 2009
2. J. Carapinha and J. Jimenez, “Network virtualization a view from the bottom," ACM SIGCOMM workshop on Virtualized Infrastructure Systems and Architectures (visa), Barcelona, Spain, August 2009.
3. O. M. E. Committee. “Software-defined Networking: The New Norm for Networks”. Open Networking Foundation, 2012.
4. Open Networking Foundation: http://www.opennetworking.org/
5. N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner. “OpenFlow: Enabling Innovation in Campus Networks”. White paper. March 2008.
6. Y.Stein, “SDN-NFV and all that”, mini course ,available at http://www.dspcsp.com/
7. Network Sharing in LTE Opportunity & Solutions. “Network Sharing in LTE,” Technology White Paper, Alcatel Lucent, 2010.
8. Liang Zhao; Ming Li; Zaki, Y.; Timm-Giel, A.; Gorg, C., "LTE virtualization: From theoretical gain to practical solution," Teletraffic Congress (ITC), 2011 23rd International, vol., no., pp.71,78, 6-9 Sept. 2011
9. Costa-Perez, X.; Swetina, J.; Tao Guo; Mahindra, R.; Rangarajan, S., “Radio Access Network Virtualization for Future Mobile Carrier Networks,” Communications Magazine, IEEE, vol.51, no.7, pp.27, 35, July 2013.
10. ETSI GS NFV 001: "Network Functions Virtualisation (NFV); Use Cases
11. “ONF Solution Brief.”, February 17, 2014, available at http://www.opennetworking.org/
12. China Mobile Research institute "C-RAN the road towards green RAN" , white paper, version 3 December 2013
13. Felita, C.; Suryanegara, M., "5G key technologies: Identifying innovation opportunity," QiR (Quality in Research), 2013 International Conference on , vol., no., pp.235,238, 25-28 June 2013

Thank you