# Network Virtualization

Nelson L. S. da Fonseca

IEEE ComSoc Summer Scool

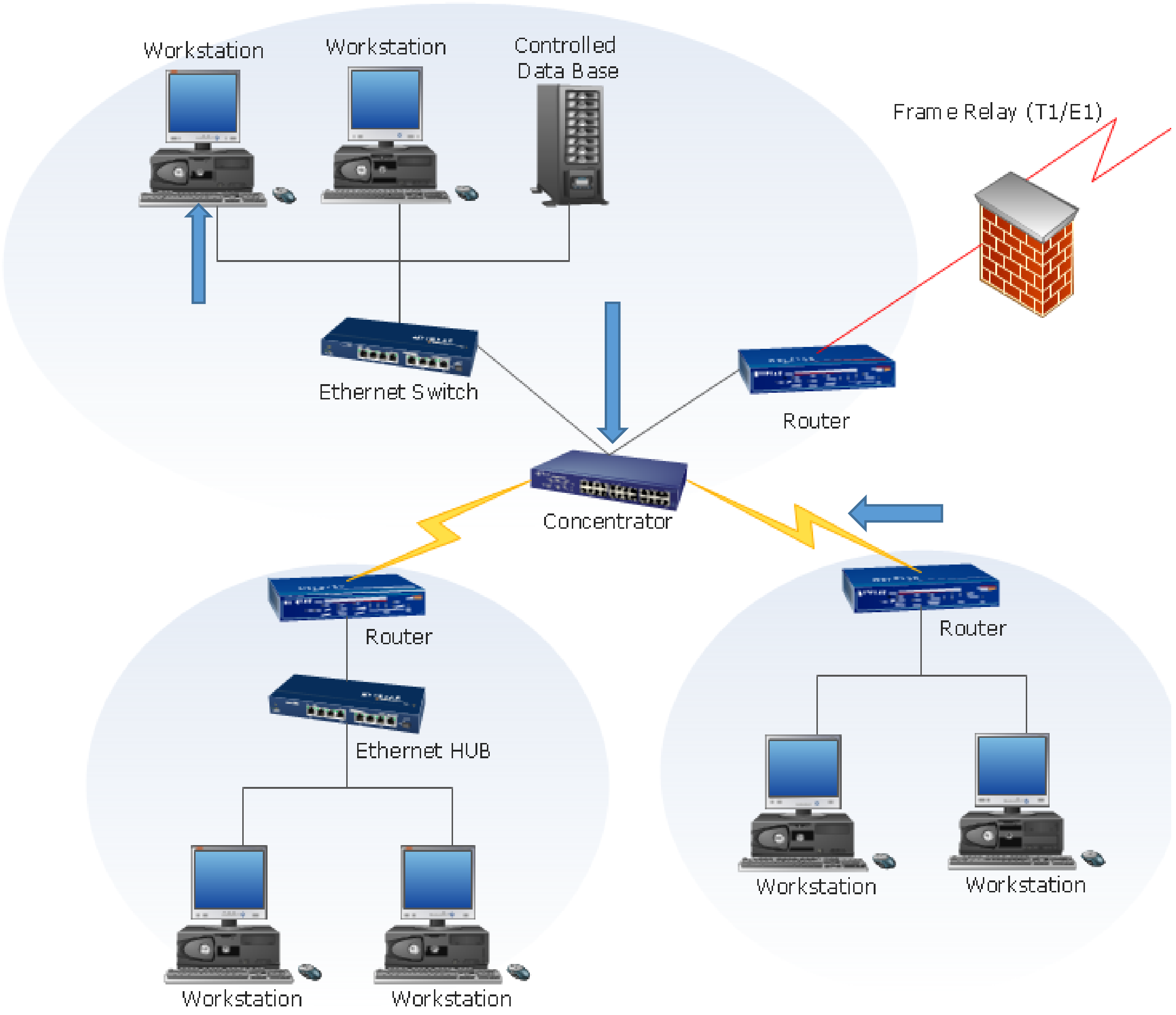
Albuquerque, July 17-21, 2017

# Acknowledgement


- Some slides in this set of slides were kindly provided by:

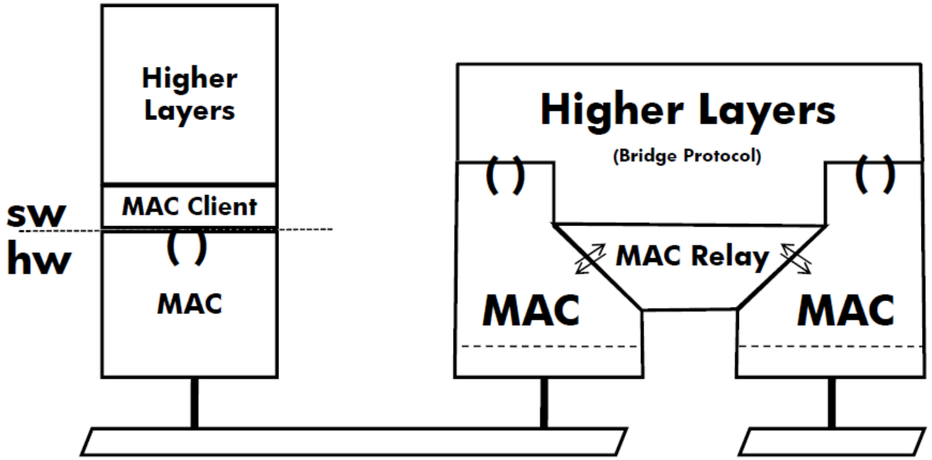
- Raj Jain, Washington University in St. Louis
- Christian Esteve Rothenberg, University of Campinas

# Network Virtualization

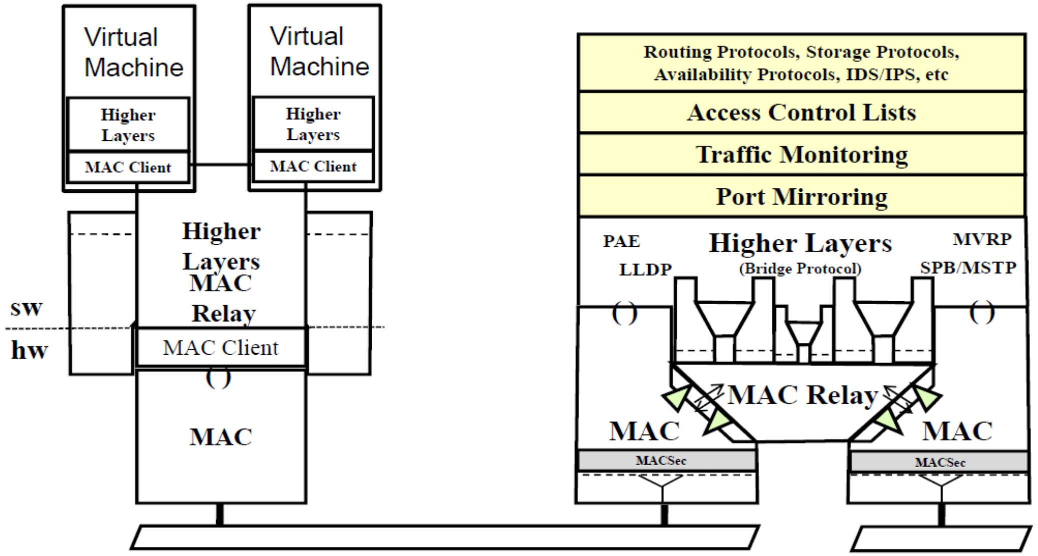


# Networking

## Traditional Networking
### The end-station and bridge




## Modern Networking
### The end-station and bridge




http://www.docstoc.com/docs/88675018/Edge-Virtual-Bridging

# Multitenancy

Multitenancy is the fundamental technology that clouds use to share IT resources cost-efficiently and securely. Just like in an apartment building in which many tenants cost-efficiently share the common infrastructure of the building but have walls and doors that give them privacy from other tenants - a cloud uses multitenancy technology to share IT resources securely among multiple applications and tenants (businesses, organizations) that use the cloud.

http://s3.amazonaws.com/dfc-wiki/en/images/8/8b/Forcedotcom-multitenant-architecture-wp-2012-12.pdf

# Multitenancy

- Network virtualization allows tenant can control:
  - Connectivity layer: Tenant network can be L2 while the provider is L3 and vice versa
  - Addresses: MAC addresses and IP addresses
  - Network Partitions: VLANs and Subnets
  - Node Location: Move nodes freely
- Network virtualization allows providers to serve a large number of tenants without worrying about:
  - Internal addresses used in client networks
  - Number of client nodes
  - Location of individual client nodes
  - Number and values of client partitions (VLANs and Subnets)

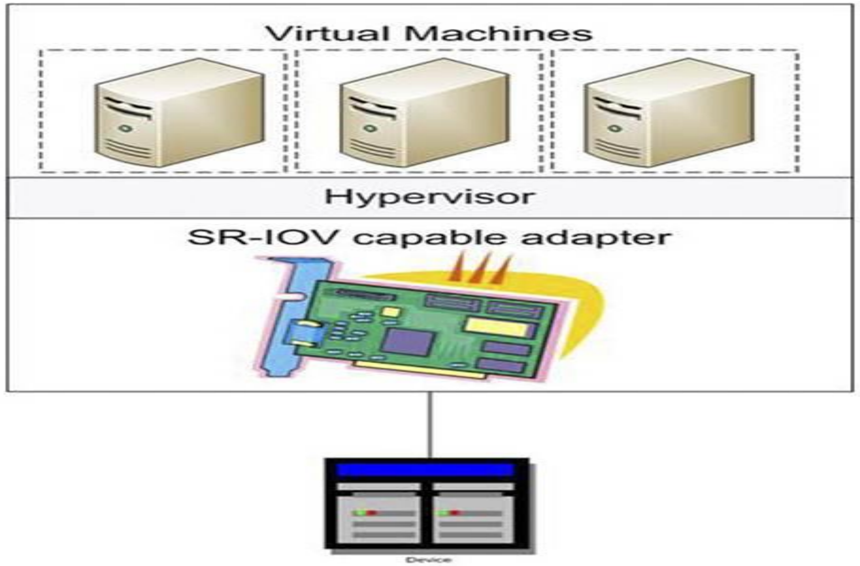
http://www.cse.wustl.edu/~jain/cse570-13/

# Network Virtualization techniques

| | Technique |
|---|---|
| NIC | SR-IOV, MR-IOV |
| Switch | VEB, VEPA, VSS, VBE, DVS, FEX |
| L2 Link | VLAN |
| L2 network using L2 | VLAN |
| L2 network using L3 | NVO3, VXLAN, NVGRE, STT, TRILL, LISP |
| Router | VRF, VRRP |
| L3 network using L3 | MPLS, GRE, IPSec |

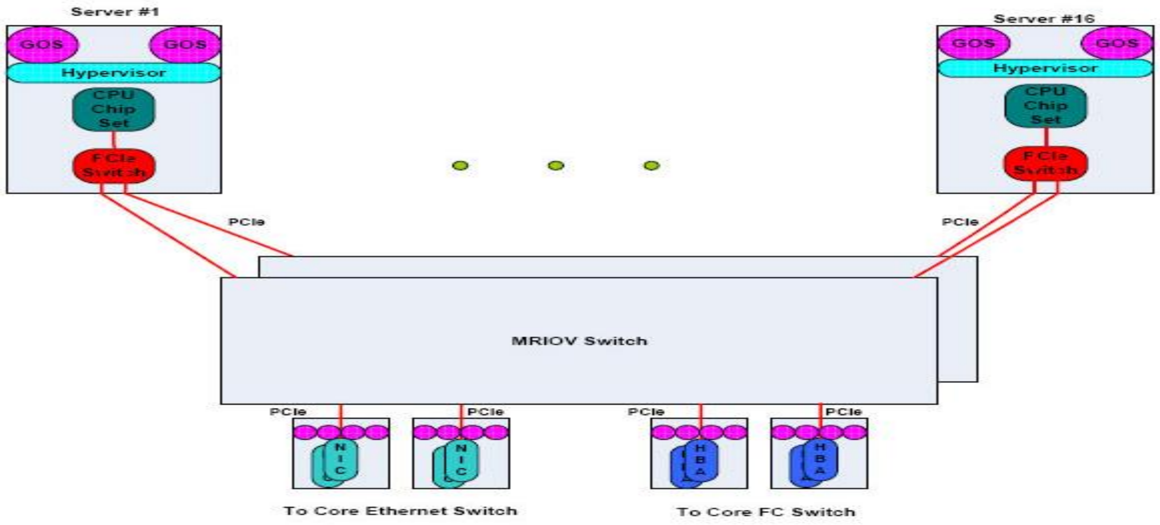
# NIC Virtualization

# SR-IOV




- *Single Root IOV*
- SR-IOV is a specification that allows a PCIe device to appear to be multiple separate physical PCIe devices.
- With SR-IOV, a card that's SR-IOV-capable has the intelligence to manage the virtual connections so the hypervisor doesn't have to, which means you get a few cycles back in your CPU for other things because it's now offloaded to the card.

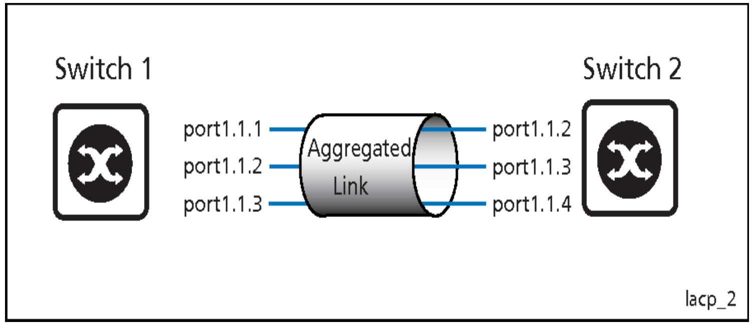
# MR-IoV




- PCI adapter in the switching fabric, not in the adapter
- Can serve several physical adapters

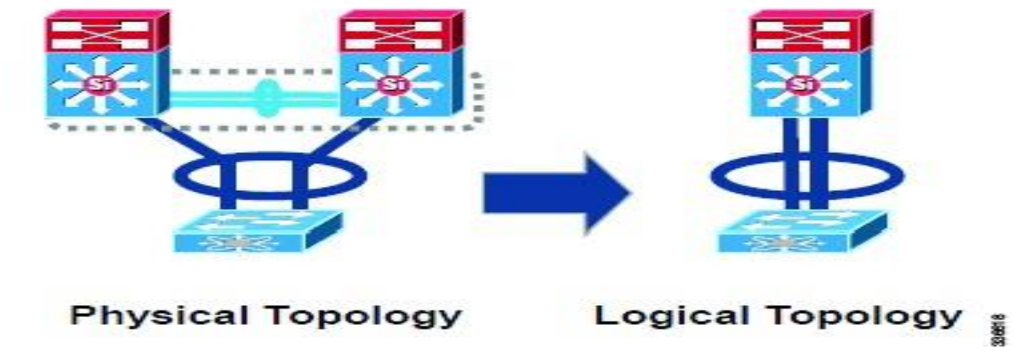
# Link Virtualization

# Link Aggregation Control Protocol



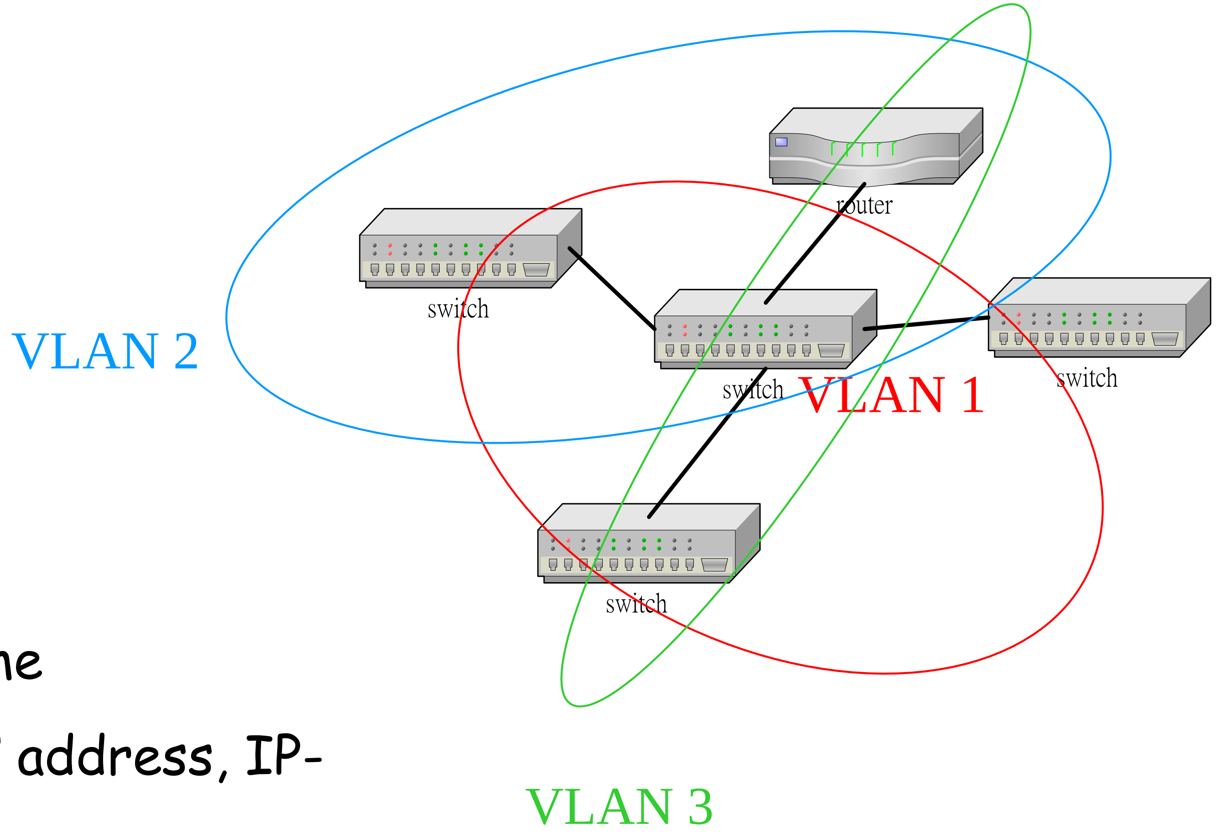

- IEEE 802.3ad
- Link Aggregation Control Protocol (LACP) provides a method to control the bundling of several physical ports together to form a single logical channel. LACP allows a network device to negotiate an automatic bundling of links by sending LACP packets to the peer (directly connected device that also implements LACP)

# Link Aggregation




- A virtual port channel (vPC, Cisco) allows links that are physically connected to two different devices to appear as a single port channel to a third device. The third device can be a switch, server, or any other networking device that supports link aggregation technology.
- Split Multi-link Trunking (SMLT, Nortel) or "Multi-Chassis Link Aggregation (MC-LAG Alcatel-Lucent).

# Virtual Local Area network (VLAN)

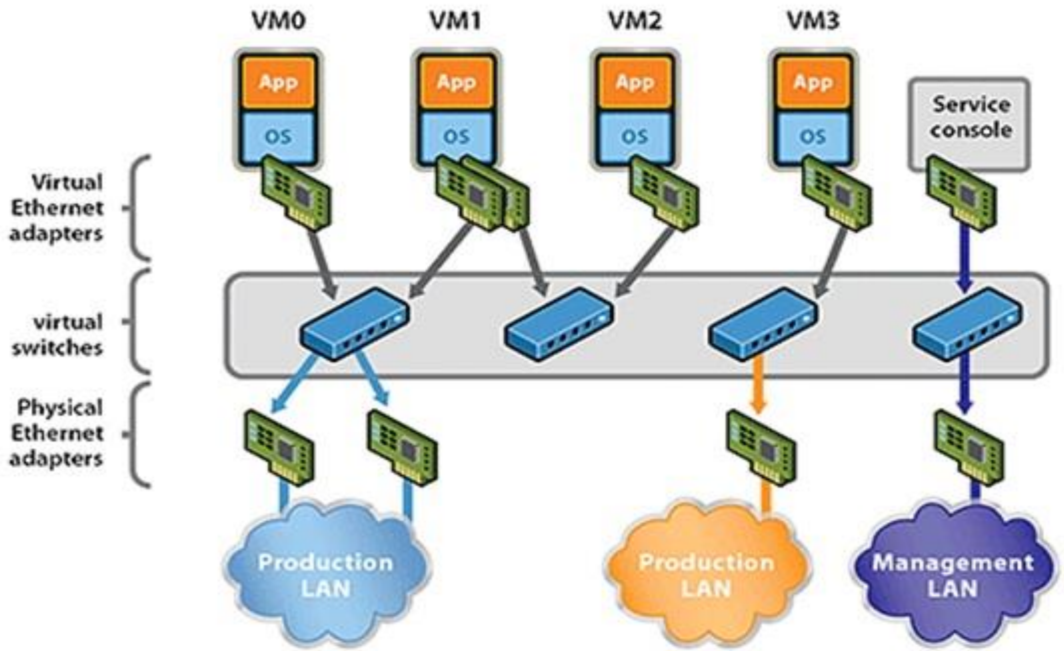



- IEEE 802.1Q
- Logical connection
- tagged frame vs. untagged frame
- Can be associated to port, MAC address, IP-subnet, protocol, application

http://www.ieee802.org/1/pages/802.1Q.html

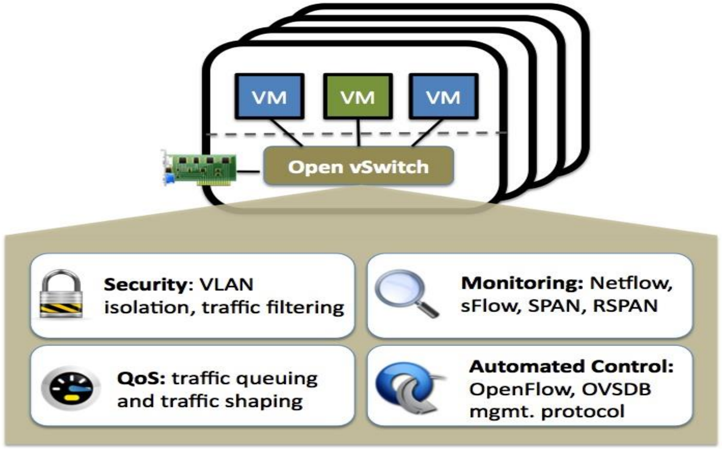
# Switch Virtualization

# vSwitch




- Allows multiple virtual machine to be connected to a physical NIC.
- The vNICs of VMs are connected to a vSwitch
- Hypervisor creates multiplex vNICs, pNIC is controlled by the Hypervisor

# Open vSwitch

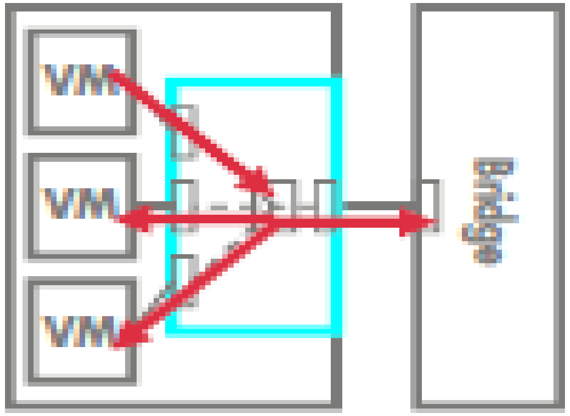



- "Open vSwitch is a production quality, multilayer virtual switch licensed under the open source Apache 2.0 license. It is designed to enable massive network automation through programmatic extension, while still supporting standard management interfaces and protocols (e.g. NetFlow, sFlow, IPFIX, RSPAN, CLI, LACP, 802.1ag). In addition, it is designed to support distribution across multiple physical servers."

http://openvswitch.org/

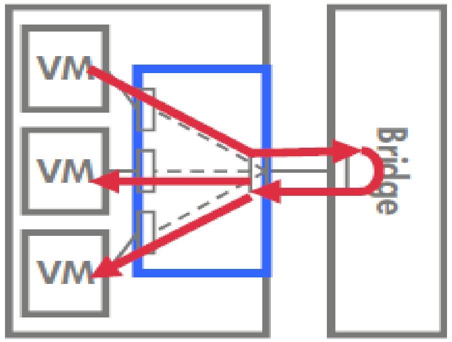
# Virtual Ethernet Bridge (VEB)

- IEEE 802.1Qbg-2012 standard for vSwitch
- Emulates 802.1 bridges,
- switch internally
- Either in hypervisor or NIC
- Works with all bridges
- Limited bridge visibility
- No changes, legacy solution




Virtual Ethernet Bridge (VEB)

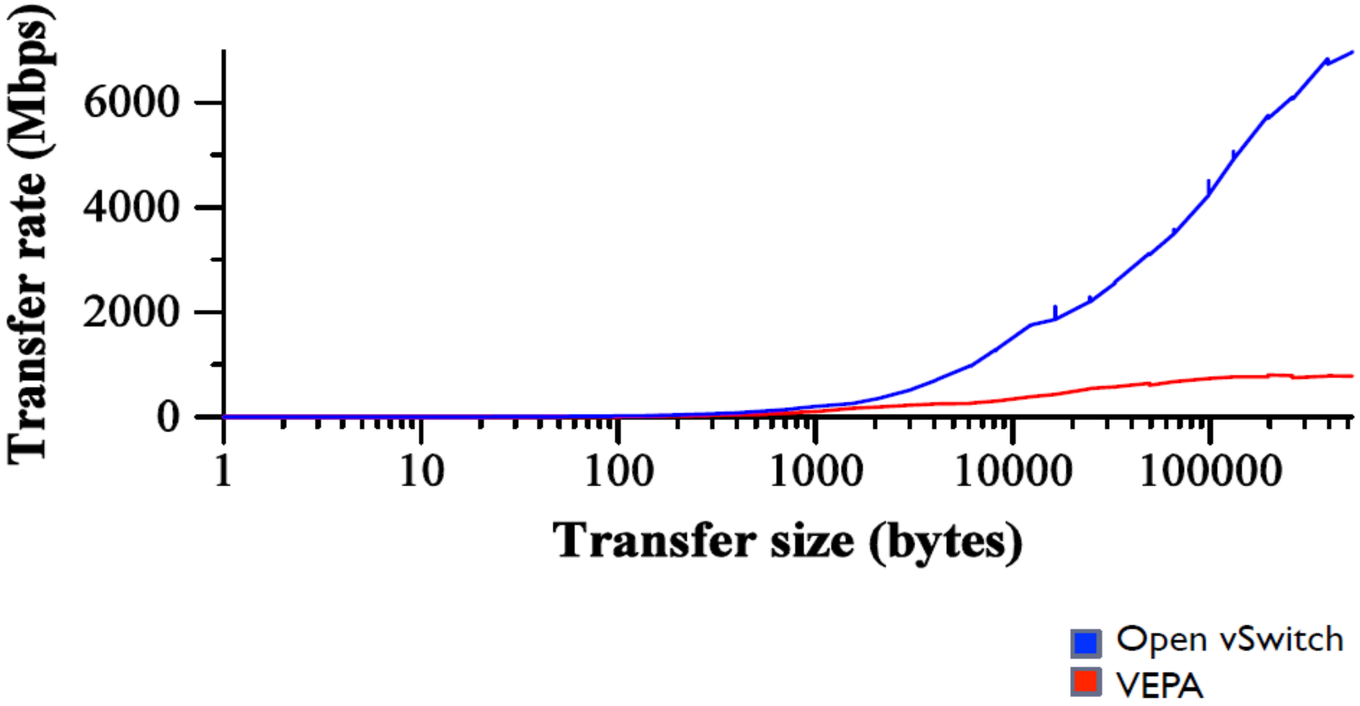
# Virtual Ethernet Port Aggregator (VEPA)




Virtual Ethernet Port Aggregation (VEPA)

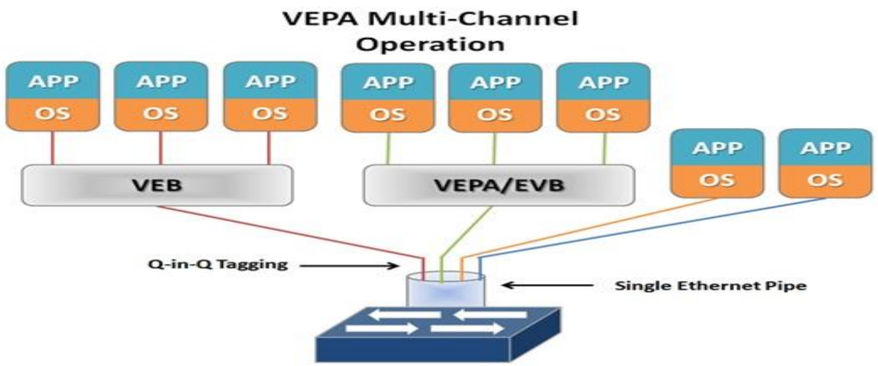
- Relays traffic to external bridge
- Hairpinning Mode – external bridge forwards the traffic, returns traffic to VEPA
- Access to Bridge features (firewalLess load on CPU

# On-box Performance




J. Pettit, J. Gross, B. Pfaff, M. Casado, S. Crosby, "Virtual Switching in an Era of Advanced Edges," 2nd Workshop on Data Center - Converged and Virtual Ethernet Switching (DC-CAVES), ITC 22, Sep. 6, 2010.

# Multichannel

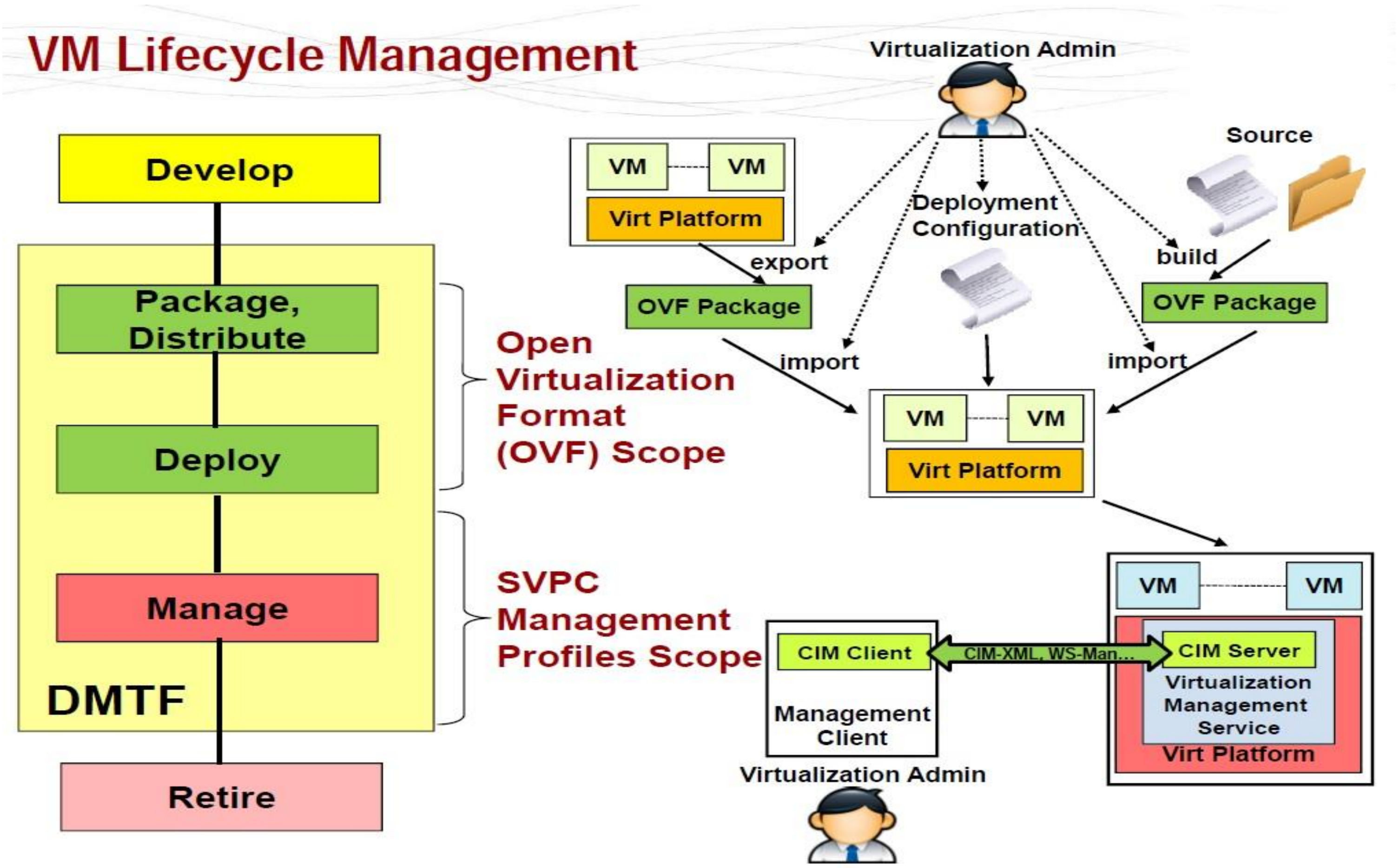



- **S-Channels**: Isolate traffic for multiple vPorts using Service

VLANs (Q-in-Q).

- Multi-Channel VEPA allows a single Ethernet connection (switchport/NIC port) to be divided into multiple independent channels or tunnels. Each channel or tunnel acts as an unique connection to the network. Within the virtual host these channels or tunnels can be assigned to a VM, a VEB, or to a VEB operating with standard VEPA.

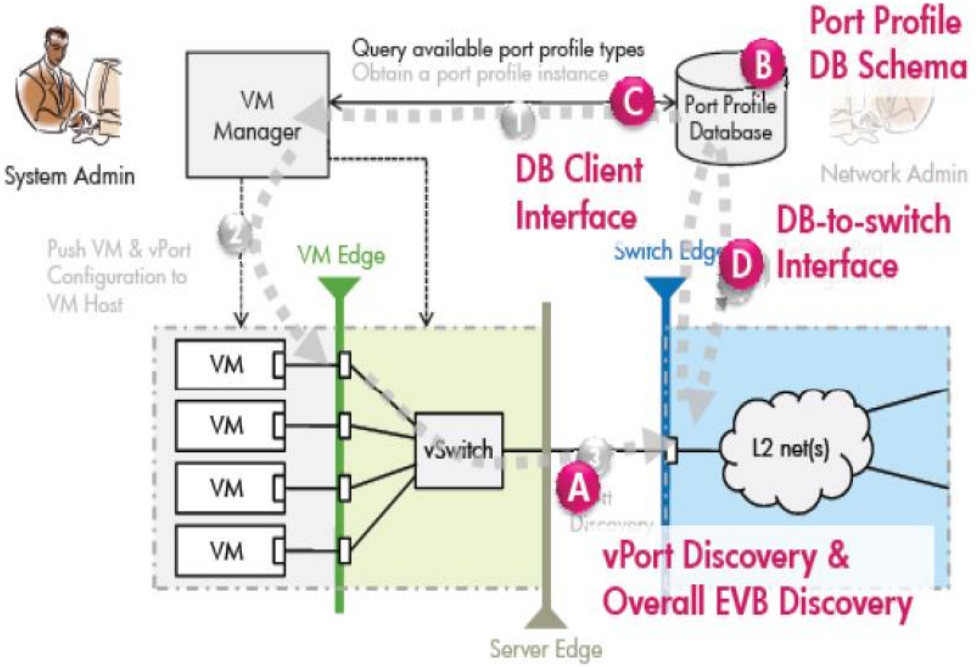
# VM Lifecycle




H. Shah, "Management Standards for Edge Virtual Bridging (EVB) and Network Port Profiles," Nov 2010, http://www.ieee802.org/1/files/public/docs2011/bg-shah-dmtf-evbportprofile-overview-0311.pdf

# Network Port Profile

- Set of atributes that can be applied to one or more virtual machine




```xml
<xs:schema xmlns:ppns="http://schemas.dmtf.org/svpc/portprofile/1"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:cim="http://schemas.dmtf.org/wbem/wscim/1/common"
  xmlns:xml="http://www.w3.org/XML/1998/namespace"
  xmlns:rasd="http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_ResourceAllocationSettingData"
  xmlns:epasd="http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_EthernetPortAllocationSettingData"
  xmlns:ns1="http://schemas.dmtf.org/svpc/portporfile/1"
  targetNamespace="http://schemas.dmtf.org/svpc/portporfile/1" elementFormDefault="qualified" attributeFormDefault="qualified">
  <xs:import namespace="http://www.w3.org/XML/1998/namespace" schemaLocation="http://www.w3.org/2001/xml.xsd"/>
  <xs:import namespace="http://schemas.dmtf.org/wbem/wscim/1/common"
    schemaLocation="http://schemas.dmtf.org/wbem/wscim/1/common.xsd"/>
  <xs:import namespace="http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_ResourceAllocationSettingData"
    schemaLocation="http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2.22.0/CIM_ResourceAllocationSettingData.xsd"/>
  <xs:import namespace="http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_EthernetPortAllocationSettingData"
    schemaLocation="http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2.27.0/CIM_EthernetPortAllocationSettingData.xsd"/>
       .......
  <xs:element name="PortProfile">
    <xs:annotation>
      <xs:documentation>Root element of Port Profile </xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Item" type="epasd:CIM_EthernetPortAllocationSettingData_Type" minOccurs="0" maxOccurs="unbounded"/>
        <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

H. Shah, "Management Standards for Edge Virtual Bridging (EVB) and Network Port Profiles," Nov 2010, http://www.ieee802.org/1/files/public/docs2011/bg-shah-dmtf-evbportprofile-overview-0311.pdf

# Edge Virtual Bridge (EVB) Management

- Network Port Profile: Attributes to be applied to a VM
- Application Open Virtualization Format (OVF) packages may or may not contain network profile

After VM instantiation, generally networking team applies aport profile to VM

- Distributed Management Task Force (DMTF) has extendedOVF format to support port profiles
- Resource allocation profile
- Resource capability profile
- vSwitch profile, etc.

# IEEE 802.1Qbg Protocols for Auto-Discovery and Configuration

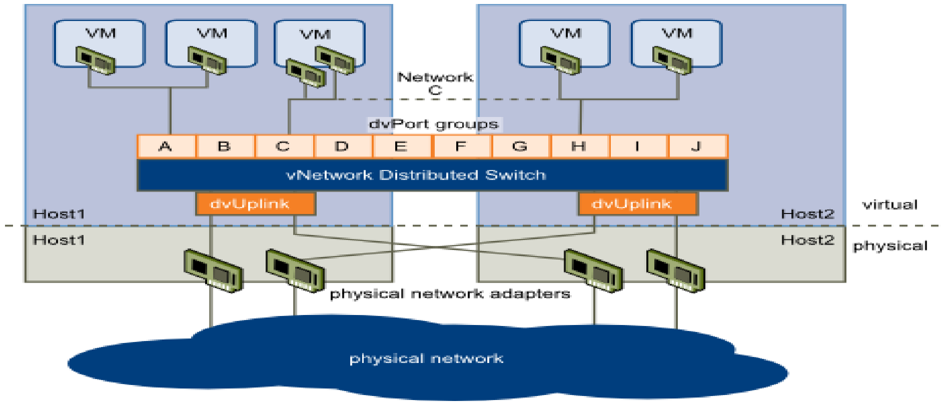

- Edge Discovery and Configuration Protocol (EDCP)
- VSI Discovery and Configuration Protocol (VDP)
- S-Channel Discovery and Configuration Protocol (CDCP)
- Edge Control Protocol (ECP) to provided reliable delivery for VDP

# Switch Aggregation

# Switch Aggregation

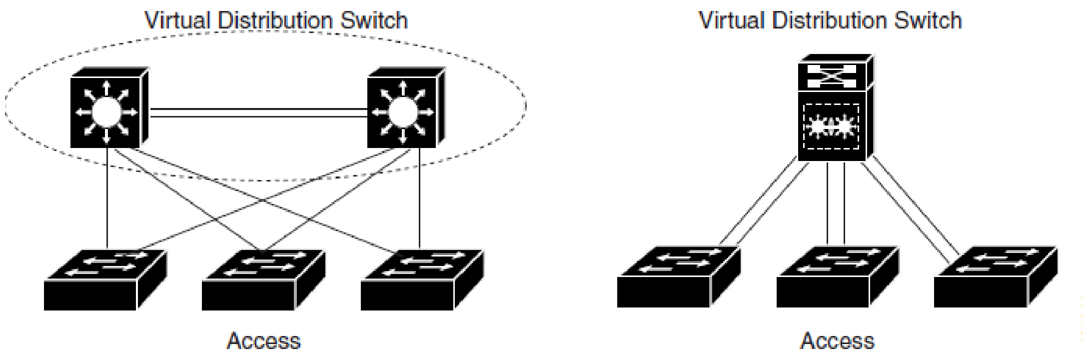
- The large number of virtual machines requires switched with large number of ports
- Different vendor technologies allows the aggregation of virtual switches to make a single switch

# Distributed Virtual Switches




- Vmware Vsphere
- Looks like a distributed virtual switch
- Centralized control plane manages vswitches in different physical machines
- Allows aggregation into groups of ports

# Virtual Switching System



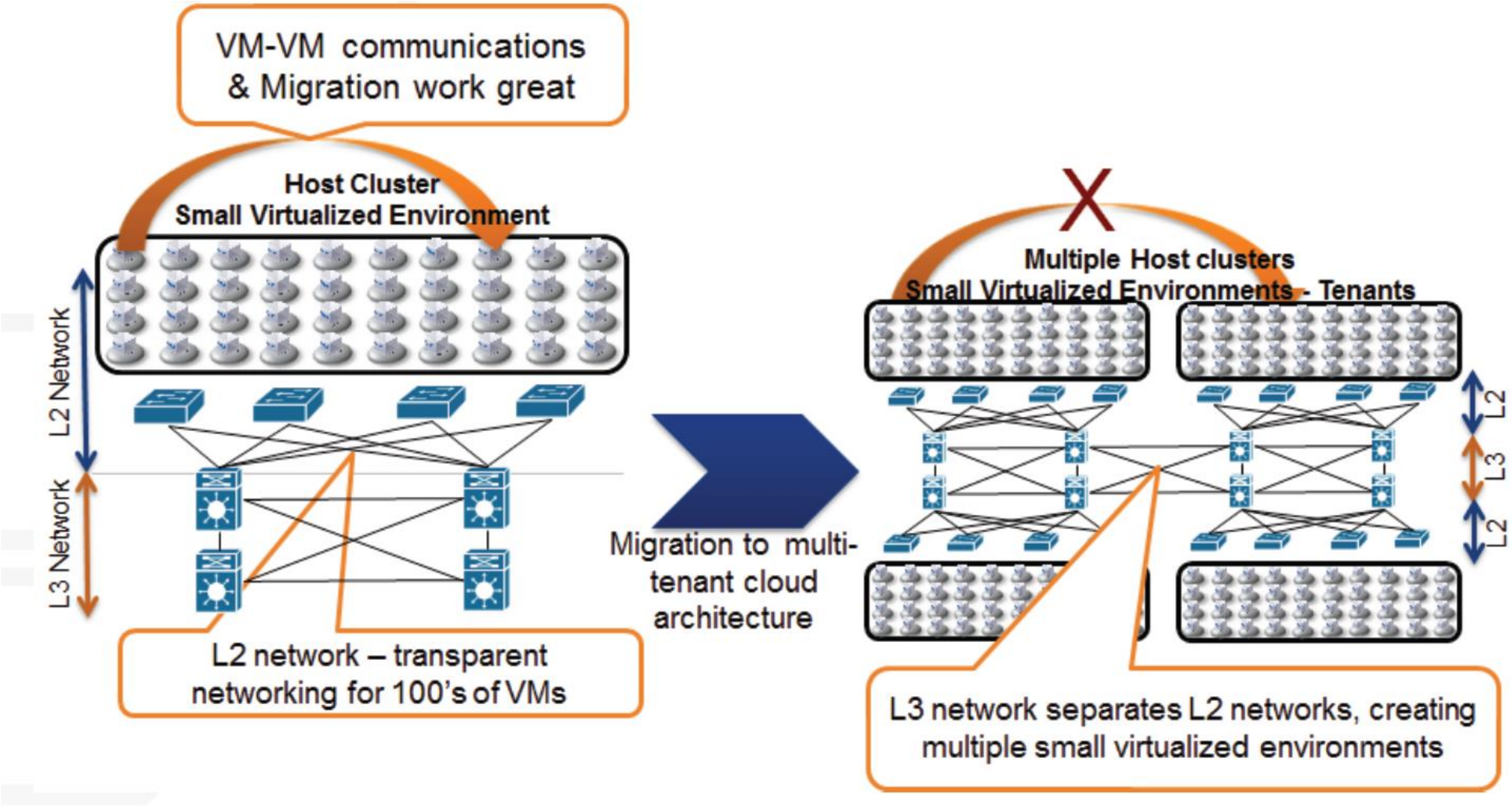

- Cisco
- allows the clustering of two or more physical chassis together into a single, logical entity
- implemented in firmware, only one control plane

# Chassis Virtualization

- “To reduce the management cost of networks comprising large number of bridges through significant reduction in both the number of devices to be managed and the management traffic required.”
- IEEE 802.1BR- standard for fabric extender functions
- Specifies how to form an extended bridge consisting of a controlling bridge and Bridge Port Extenders
- Fabric Extender (Cisco)

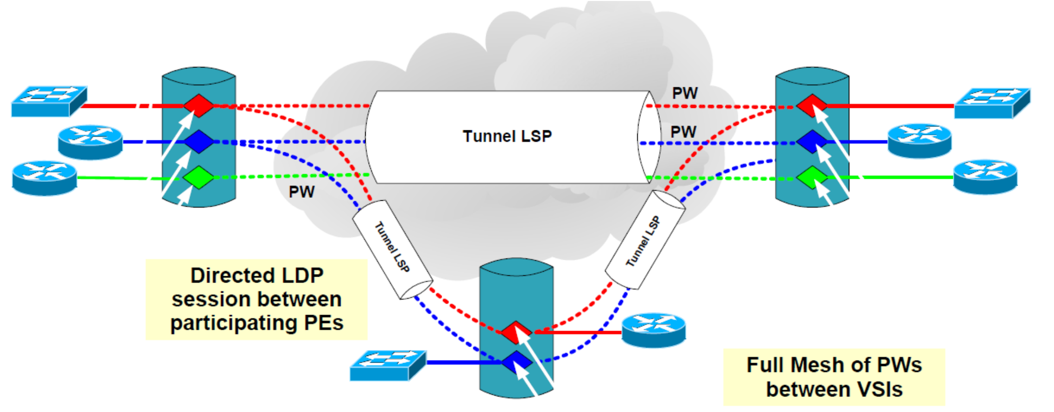
# L2 over L3

# L2 over L3




http://www.emulex.com/artifacts/074d492d-9dfa-42bd-9583-69ca9e264bd3/elx_wp_all_nvgre.pdf

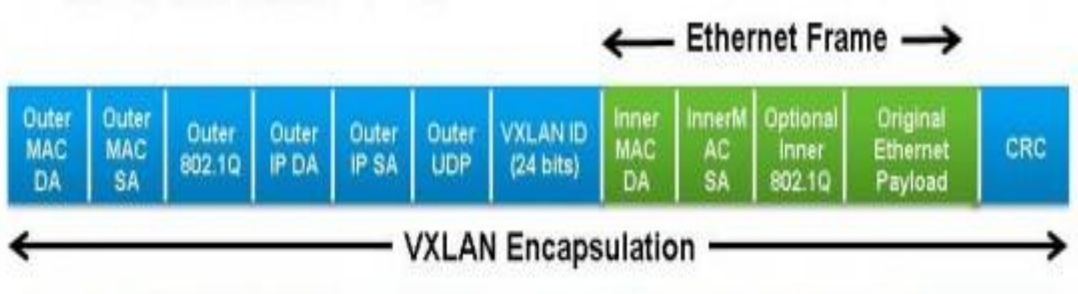
# Virtual Private LAN Service




- **Makes it possible to connect local area networks (LANs) over the Internet, so that they appear to subscribers like a single Ethernet LAN**
- Ethernet-based multipoint to multipoint communication over IP or MPLS networks,

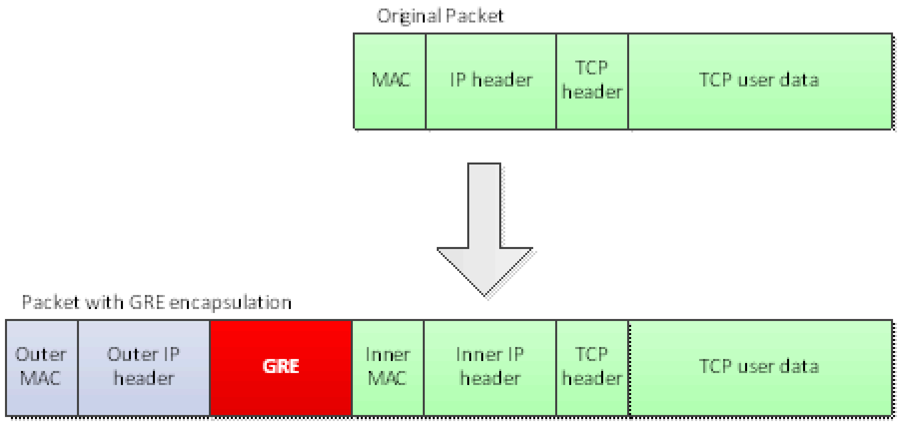
http://www.cisco.com/c/en/us/products/ios-nx-os-software/virtual-private-lan-services-vpls/index.html

# Virtual Extensible LAN (VXLAN)




- Overcomes the limitation of having 4016 VLANS, cloud environment large number of VLANs. VXLAN allows 16 millions logical networks
- STP wastes many links
- Encapsulates L2 in UDP
- VMs are unaware that they are operating on VLAN or VXLAN, vSwitches serve as VTEP (VXLAN Tunnel End Point).
- Tenants can have overlapping MAC addresses, VLANs, and IP addresses – multitenant isolation

# Generic Routing Encapsulation (GRE) L3 over L3




- Encapsulate anything into anything
- GRE header and packet into GRE payload, IP and IPSec are usually the delivery protocol

# GRE-Tunnel

GRE tunnels

GRE tunnels can incapsulate IPv4/IPv6 unicast/multicast traffic, so it is de-facto tunnel standard for dynamic routed networks. You can setup up to 64K tunnels for an unique tunnel endpoints pair. It can work with FreeBSD and cisco IOS. Kernel module is 'ip_gre'. The following example demonstrates configuration of GRE tunnel with two IPv4 routes.

```
# modprobe ip_gre

# lsmod | grep gre
ip_gre                  18244  0
ip_tunnel               23768  1 ip_gre
gre                     13808  1 ip_gre
```

# GRE-Tunnel

Host A

```
# ip tunnel add gretun0 mode gre \
 remote 172.19.20.21 \
 Local 172.16.17.18 \
 ttl 64
# ip link set gretun0 up
# ip addr add 10.0.1.1 dev gretun0
# ip route add 10.0.2.0/24 dev gretun0
```

Host B

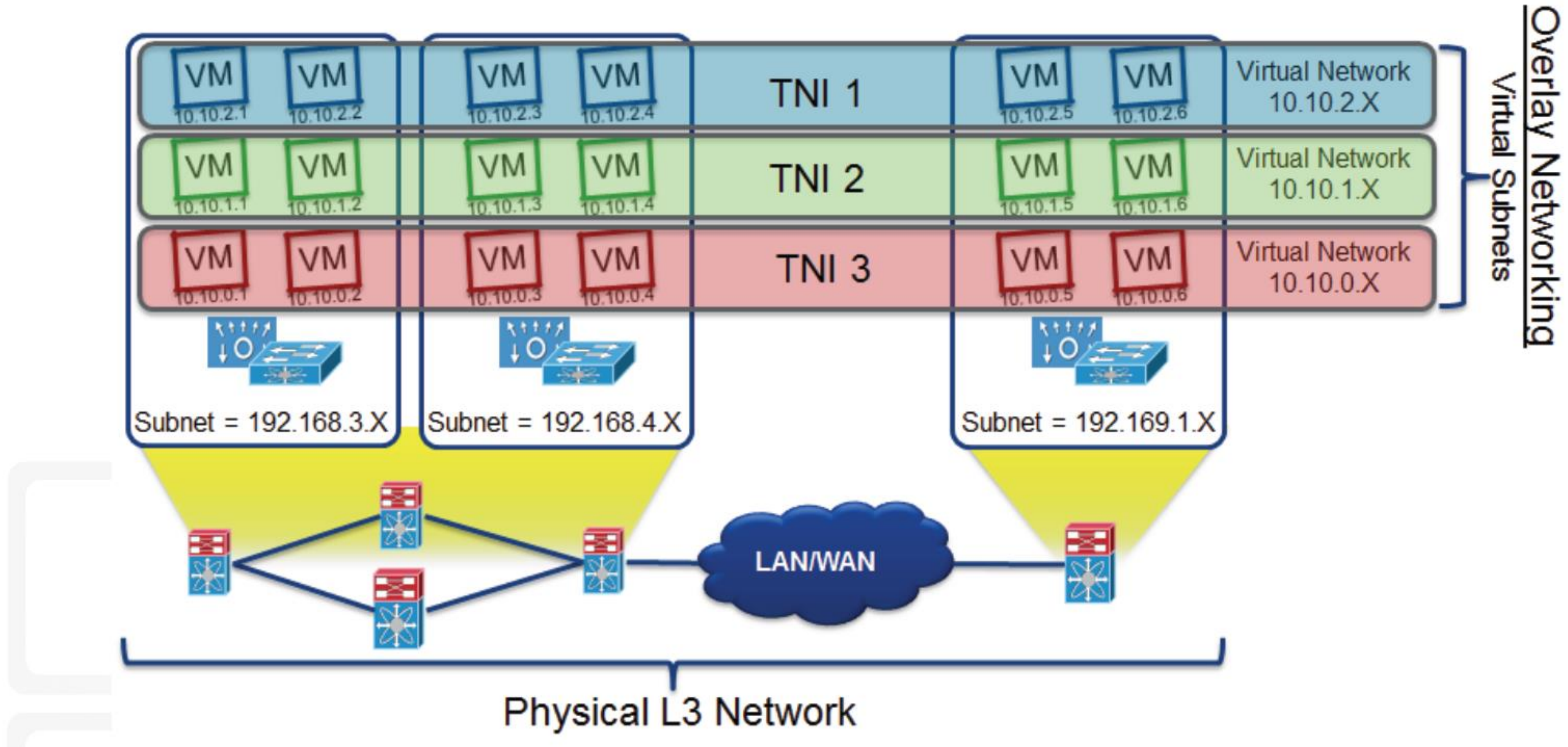
```
# ip tunnel add gretun0 mode gre \
 Remote 172.16.17.18 \
 Local 172.19.20.21 \
 ttl 64
# ip link set gretun0 up
# ip addr add 10.0.2.1 dev gretun0
# ip route add 10.0.1.0/24 dev gretun0
```

# Network Virtualization using Generic Routing Encapsulation (NVGRE)

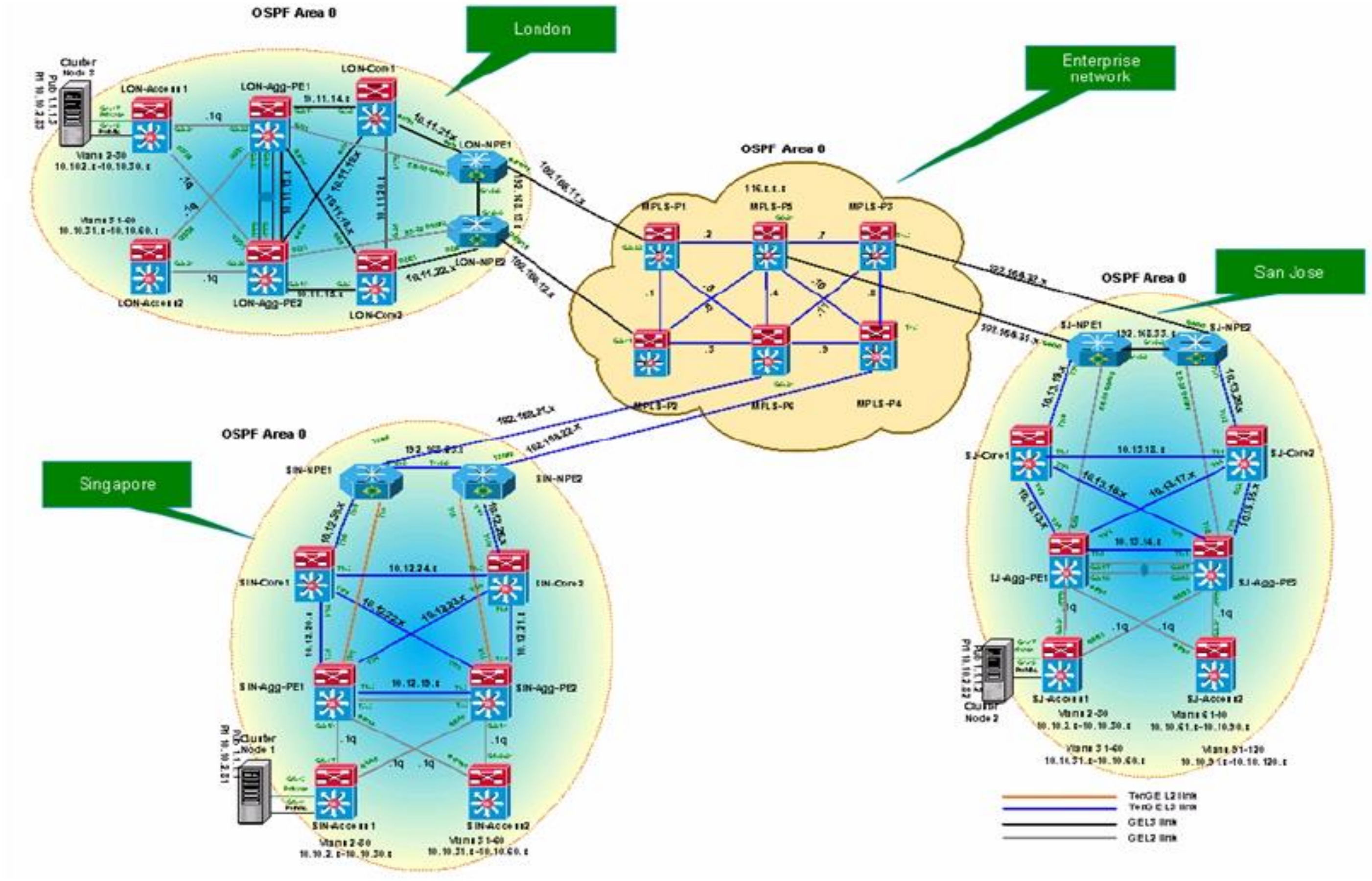
```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|C|R|K|S|s|Recur|  Flags  | Ver |         Protocol Type         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Checksum (optional)      |       Offset (optional)       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Key (optional)                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Sequence Number (optional)                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Routing (optional)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- It uses Generic Routing Encapsulation (GRE) to tunnel layer 2 (Ethernet) packets over layer 3 (IP) networks
- Uses 24 bits of optional key field of GRE header – Virtual Subnet Identifier (VSI)
- VMs in diferente VSI can have the same MAC protocol
- Equal Cost Multipath (ECMP) allowed

# Network Virtualization using Generic Routing Encapsulation (NVGRE)



# Data Center Interconnection



# Data Center Interconnection

# Data Center Interconnection

- Allows distant data centers to be connected in one L2 domain
- Distributed applications
- Disaster recovery
- Maintenance/Migration
- High-Availability
- Consolidation
- Active and standby can share the same virtual IP for switchover.
- Multicast can be used to send state to multiple destinations.

# Data center Interconnection

- Challenges of LAN Extension
- Broadcast storms: Unknown and broadcast frames may create excessive flood
- Loops: Easy to form loops in a large network.
- STP Issues: High spanning tree diameter (leaf-to-leaf) More than 7, Root can become bottleneck and a single point of failure, Multiple paths remain unused
- Tromboning: Dual attached servers and switches generate excessive cross traffic

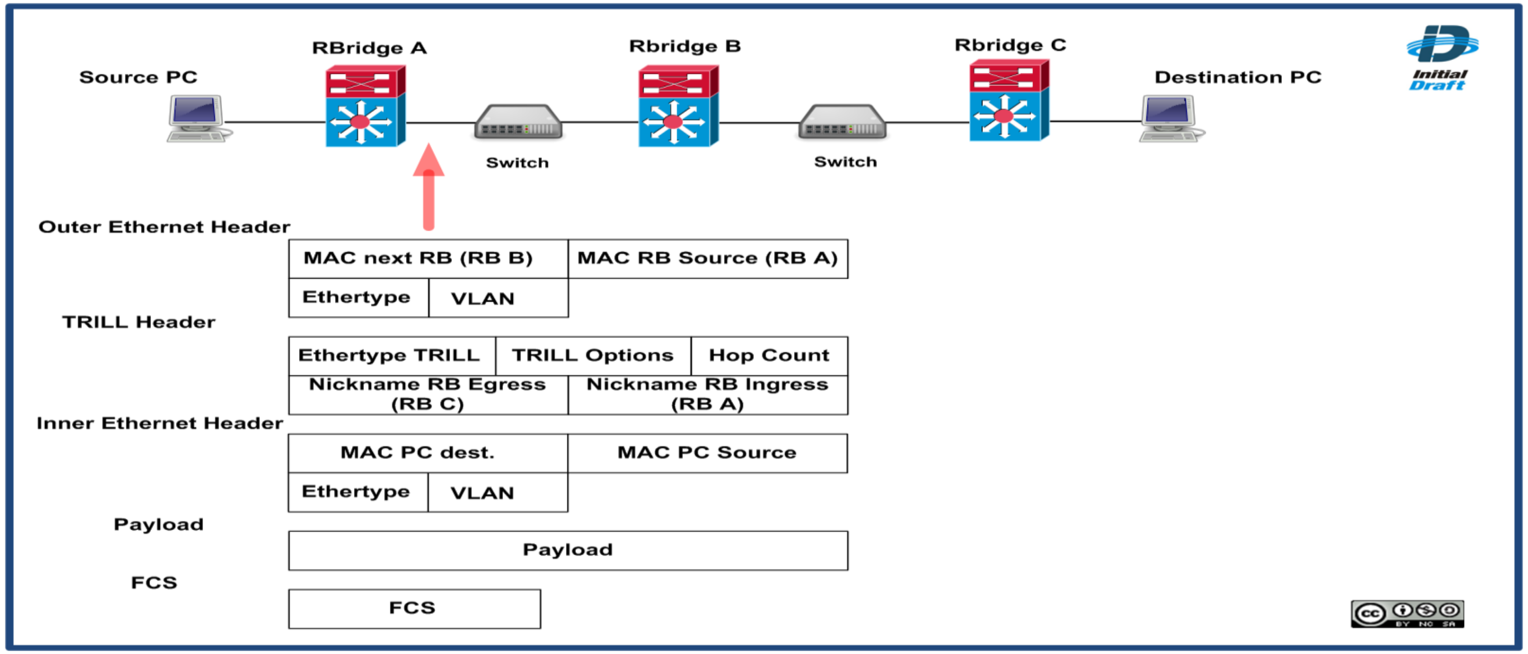
http://www.cse.wustl.edu/~jain/cse570-13/

# TRILL

- Transparent Interconnection of Lots of Links
- Allows a large campus to operate as a single LAN
- Uses MAC addressing and IP routing. TRILL combines techniques from bridging and routing and is the application of link state routing to the VLAN-aware customer-bridging problem
- No Configuration needed: RBridges discover their connectivity and learn MAC addresses automatically
- No loop formation
- Compatible with legacy bridges

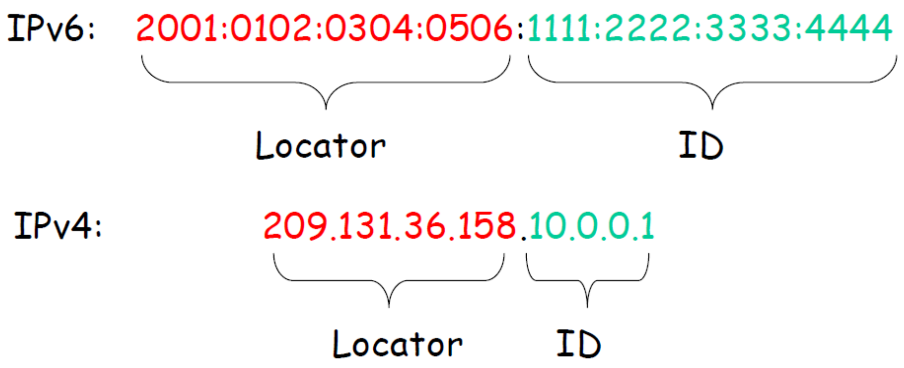
# TRILL

- Encapsulates frame and forward using IS-IS protocol



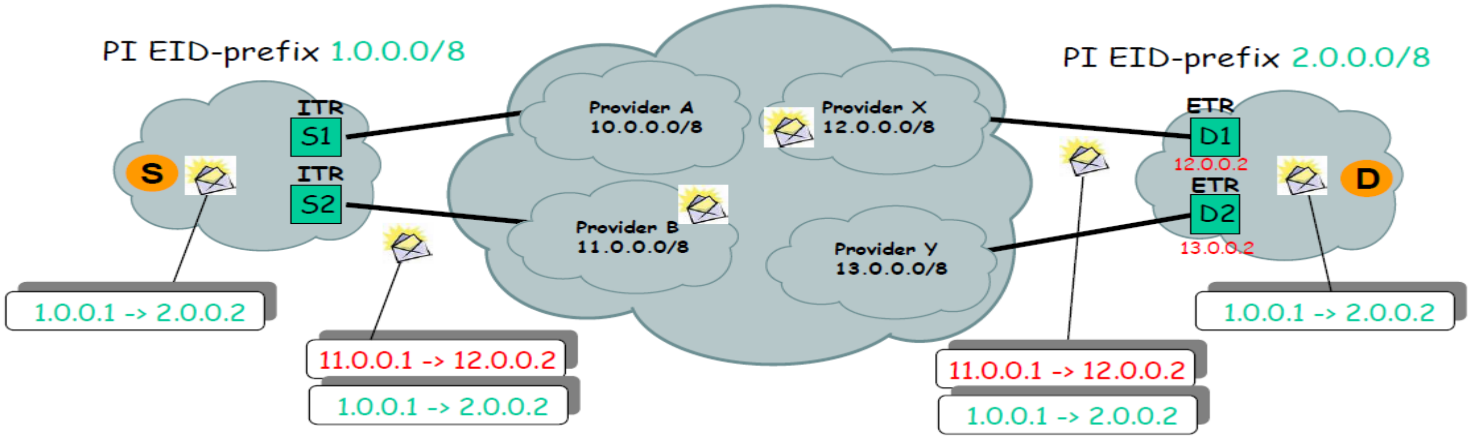

https://blog.initialdraft.com/archives/1412/

# LISP




- Locator/ID Separation Protocol
- The level of indirection allows to keep either ID or Location fixed while changing the other and create separate namespaces which can have different allocation properties
- Inside a site, the routing is based on ID, between sites, the routing is based on locators
- Changes are required only in routers at the edge of the sites.

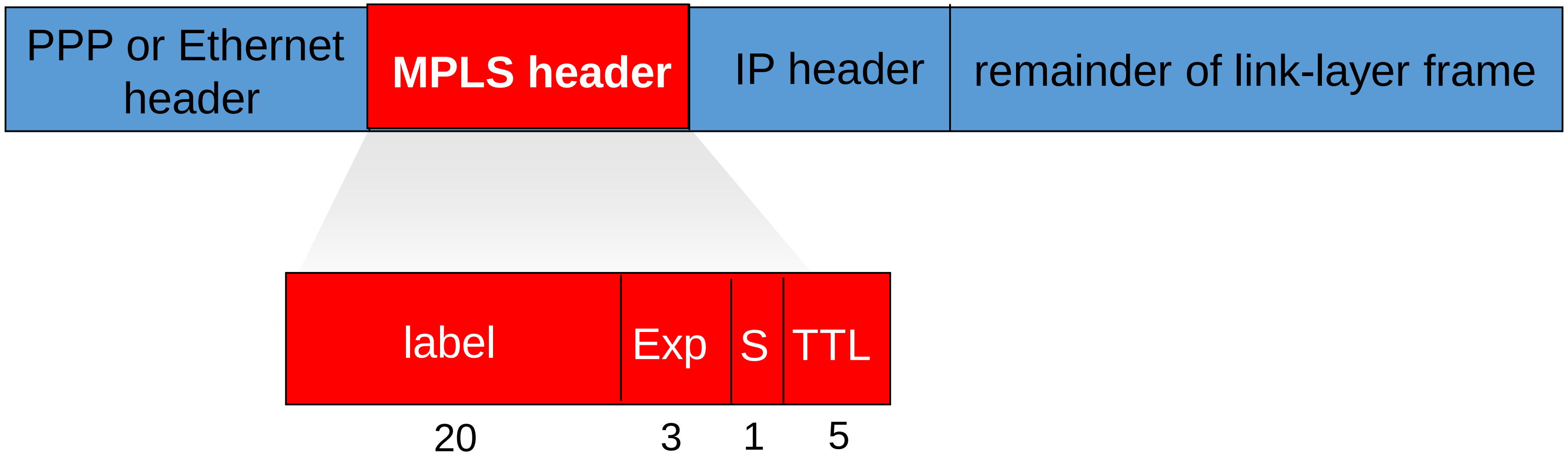
# LISP




- Ingress Tunnel Router (ITR): Encapsulates and transmits
- Egress Tunnel Router (ETR): Receives and decapsulates
- Map-server: ETRs register their EID prefix-to-RLOC mappings
- Map-Resolver: Receives map requests from ITR. Forwards them to mapping system.

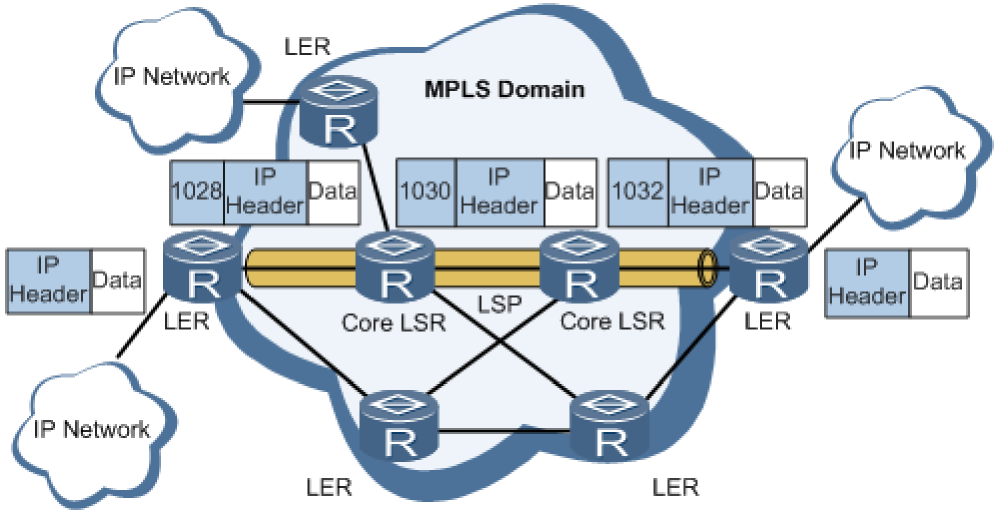
# Multiprotocol label switching (MPLS)

- initial goal: high-speed IP forwarding using fixed length label (instead of IP address)
  - fast lookup using fixed length identifier (rather than shortest prefix matching)
  - borrowing ideas from Virtual Circuit (VC) approach
  - but IP datagram still keeps IP address!



# MPLS

- L3 in L3
- Allow provisioning of QoS – MPLS Diffserv



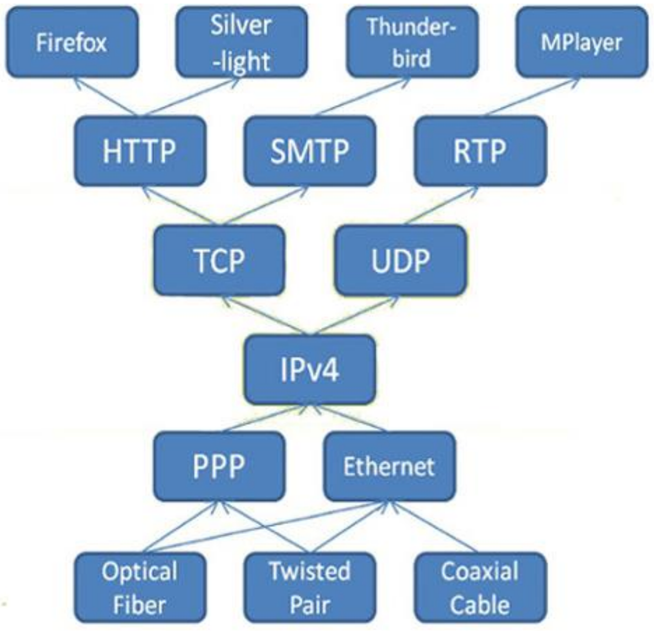
# Research Challenges

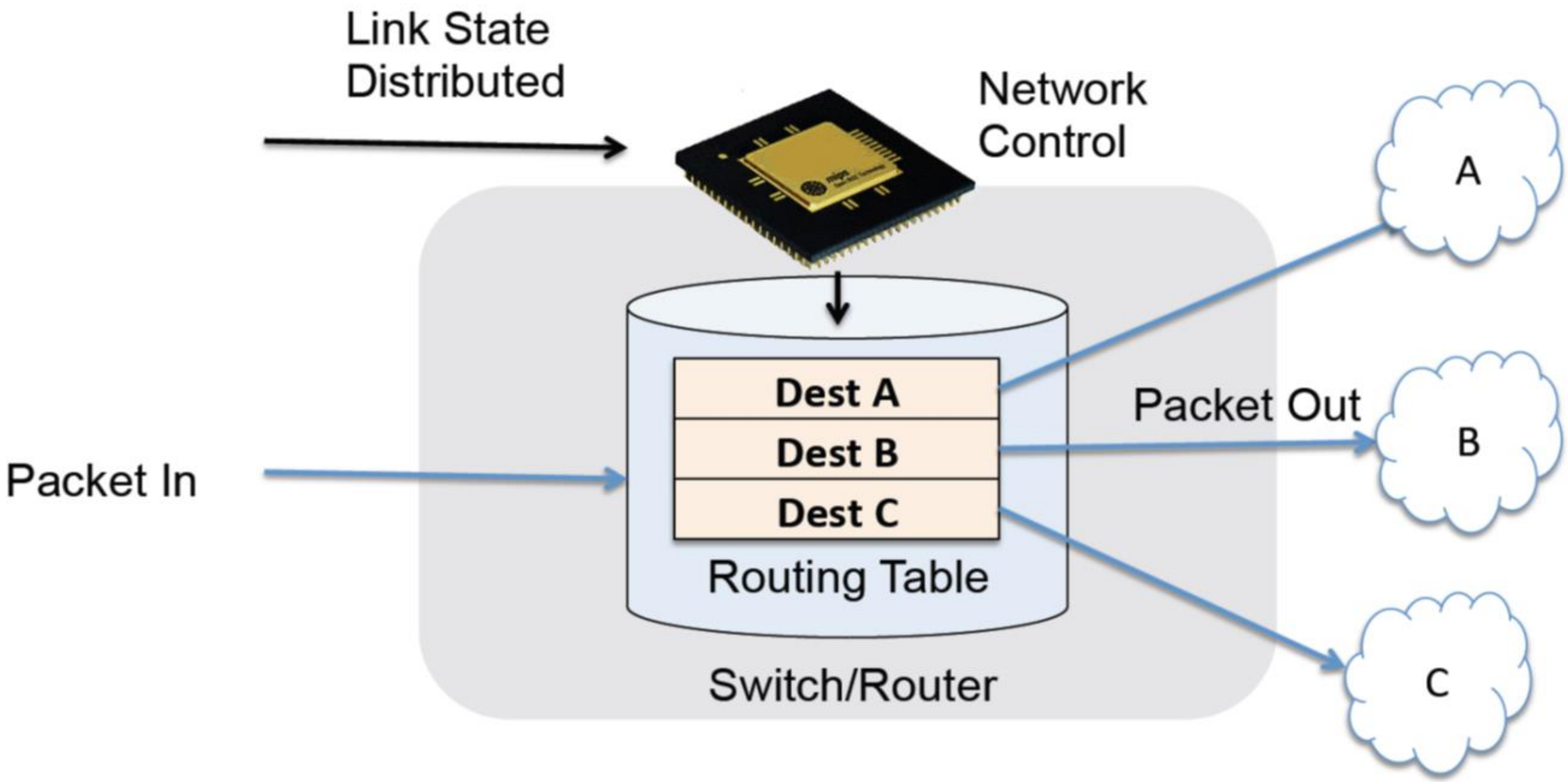
- Emulation:
  - Performance of virtual componente still higher than physical componentes,
  - Performance behaves stochastically, depends on interruption handling, scheduling on the server among others
  - encapsulation-induced overhead
- Complexity:
  - Slather multi-path routing, eventually causing congestion
  - Increase in table size
- Compatibility
  - Device and fabric virtualization challenges performance

# Recent Netwok Virtualization Techniques

# OpenFlow

# Networking as Learned in School (text books)

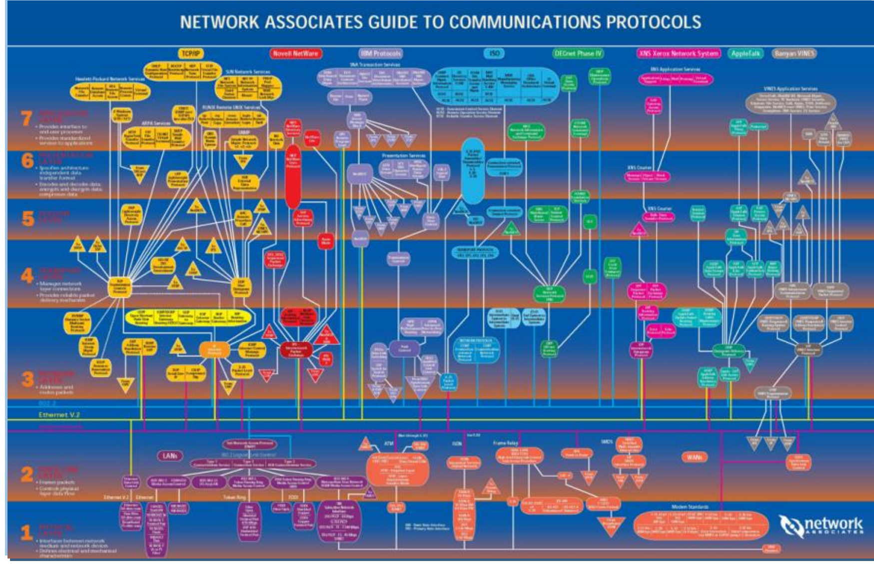


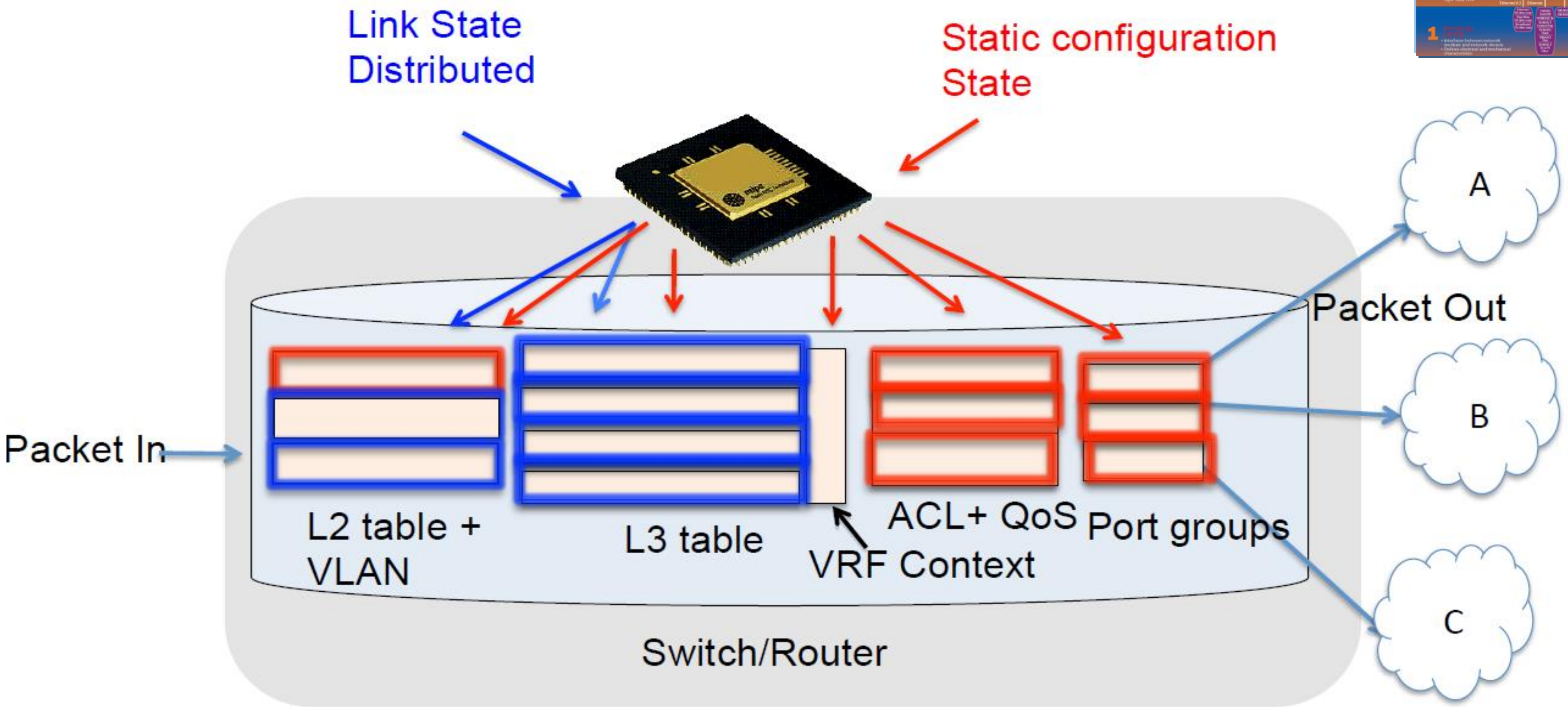




Source: Martin Casado CS244 Spring 2013, Lecture 6, SDN

# Networking in Practice

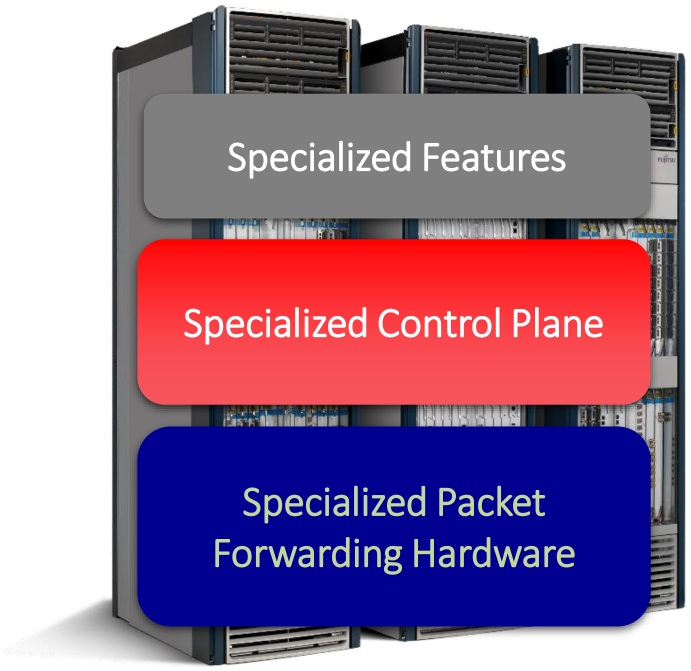
"in theory, theory and practice are the same;
in practice they are not..."




Source: Martin Casado CS244 Spring 2013, Lecture 6, SDN

# Problem with Internet Infrastructure




Hundreds of protocols
6,500 RFCs

Tens of Millions of lines of code
Closed, proprietary, outdated

Billions of gates
Power hungry and bloated

Vertically integrated, complex, closed, proprietary

Not good for network owners and users

**Source: ON.LAB**

# The Four Layers of Networking

- **Data Plane**
  - ✓All activities involving as well as resulting from data packets sent by the end user
  - ✓Forwarding
  - ✓Fragmentation and reassembly
- **Control Plane**
  - ✓All activities that are necessary to perform data plane activities but do not involve end-user data packets
  - ✓Routing tables
  - ✓Setting packet handling policies (e.g., security)
  - ✓Base station beacons announcing availability of services

# The Four Layers of Networking

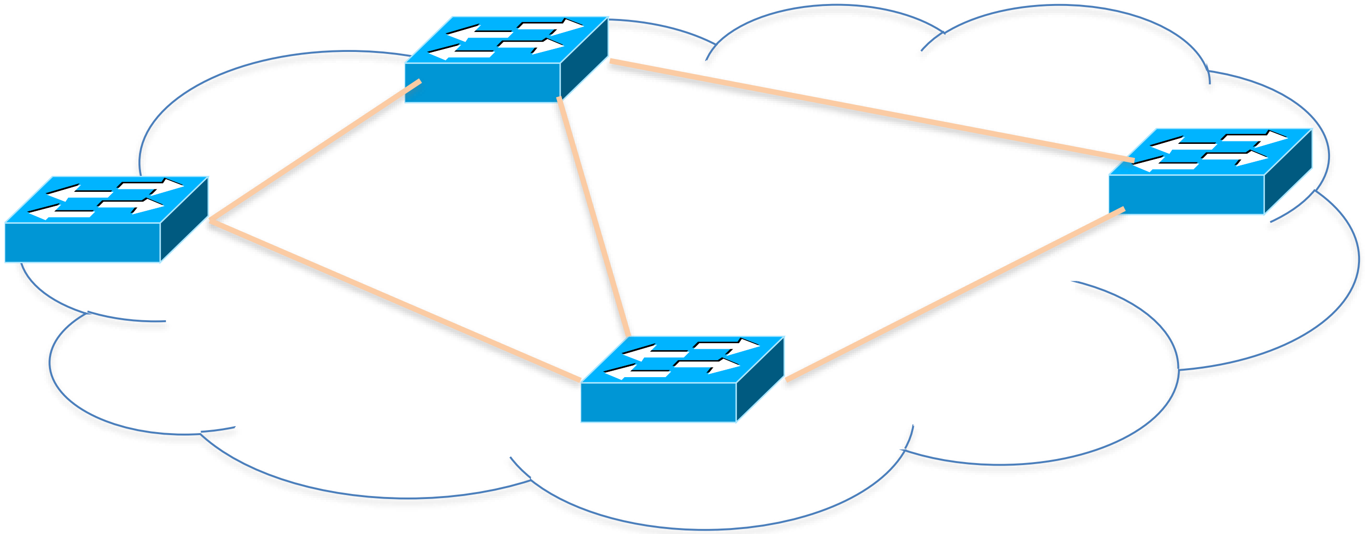
- Services plane
  - ✓Handles special tasks that require much closer scrutiny and processing of the information contained in the packets than is required for the simpler switching/routing tasks that the control plane performs.
  - ✓Firewalls, video streaming, and other such applications are
  - ✓implemented at the services layer.

- Management plane
  - ✓The layer at which the individual network devices are configured with instructions about how to interact with the network.
  - ✓Turning ports on or off
  - ✓Fault, Configuration, Accounting, Performance and Security

http://www.opendatacenteralliance.org/docs/Software_Defined_Networking_Master_Usage_Model_Rev1.0.pdf

# Rethinking the "Division of Labor"
## Traditional Computer Networks

Data plane:
Packet streaming

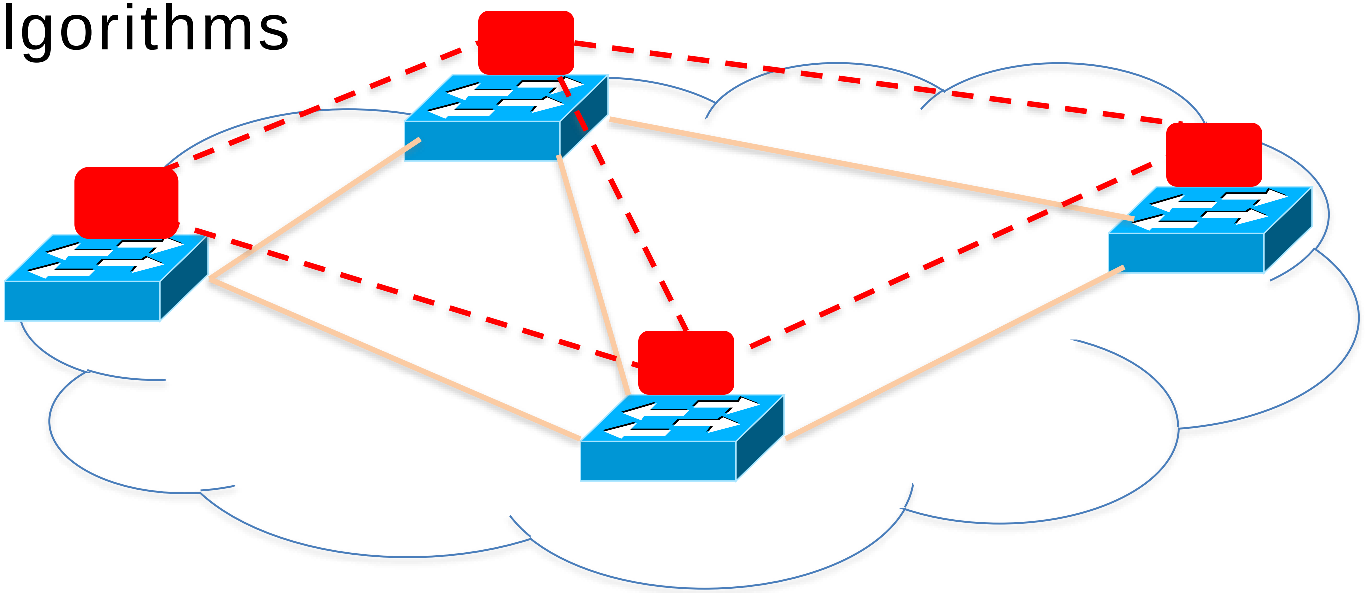


Forward, filter, buffer, mark, rate-limit, and measure packets

**Source: Adapted from J. Rexford**

# Rethinking the "Division of Labor"
## Traditional Computer Networks

Control plane:
Distributed algorithms

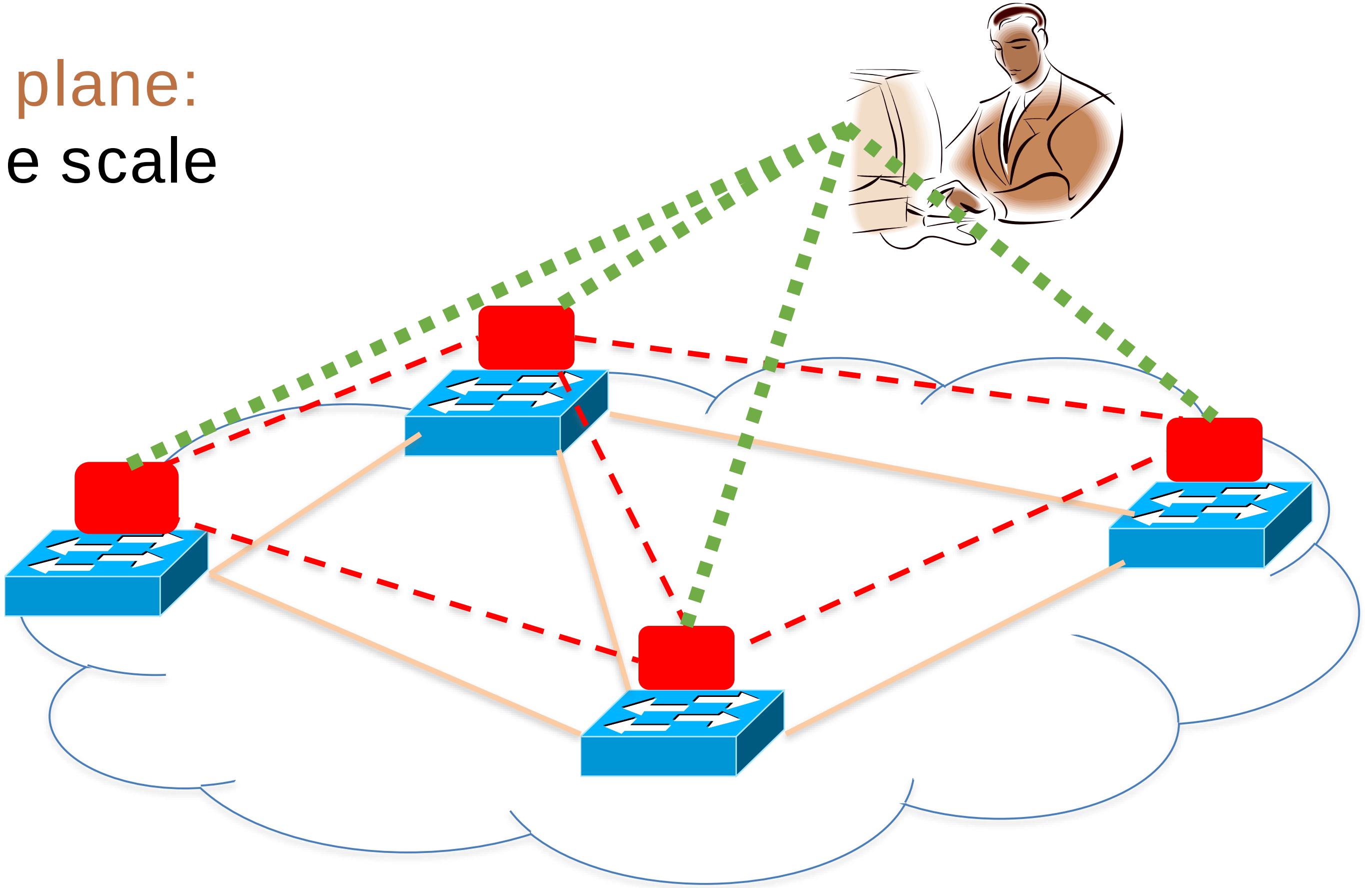


Track topology changes, compute routes, install forwarding rules

Source: Adapted from J. Rexford

# Rethinking the “Division of Labor”
## Traditional Computer Networks

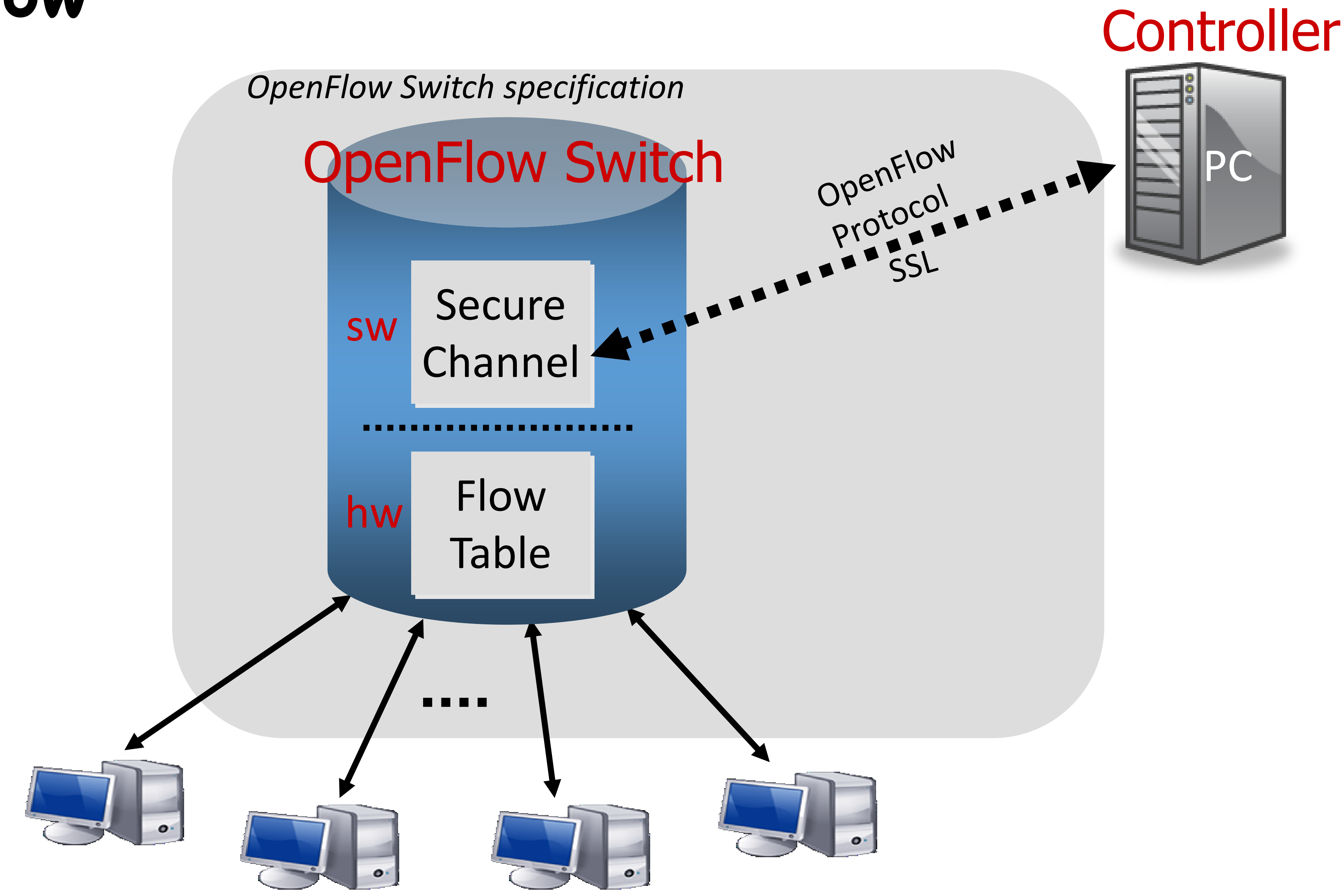
Management plane:
Human time scale



Collect measurements and configure the equipment

Source: Adapted from J. Rexford

# OpenFlow




The Stanford Clean Slate Program

http://cleanslate.stanford.edu

# Open Flow – Main Characteristics

- Separation of control and data planes
- Centralization of control
- Flow based control

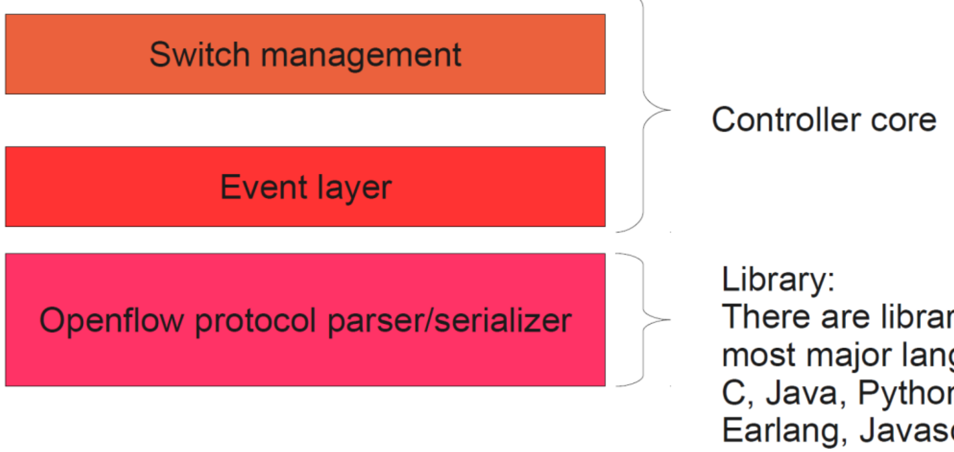
http://www.cse.wustl.edu/~jain/cse570-13/

# OpenFlow Controller

- Manages one or more switch via OpenFlow channels.
- Uses OpenFlow protocol to communicate with a OpenFlow aware switch.
- Acts similar to control plane of traditional switch.
- Provides a network wide abstraction for the applications
- Responsible for programming various tables in the OpenFlow Switch.
- Single switch can be managed by more than one controller for load balancing or redundancy purpose.

Kingston Smiler. S, Introduction to OpenFlow, SDN & NFV

# Top 3 features in most controller

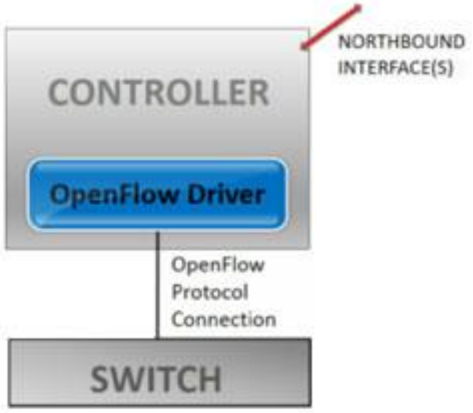



A. Event-driven model
   - Each module registers listeners or call-back functions
   - Example async events include PACKET_IN, PORT_STATUS, FEATURE_REPLY, STATS_REPLY

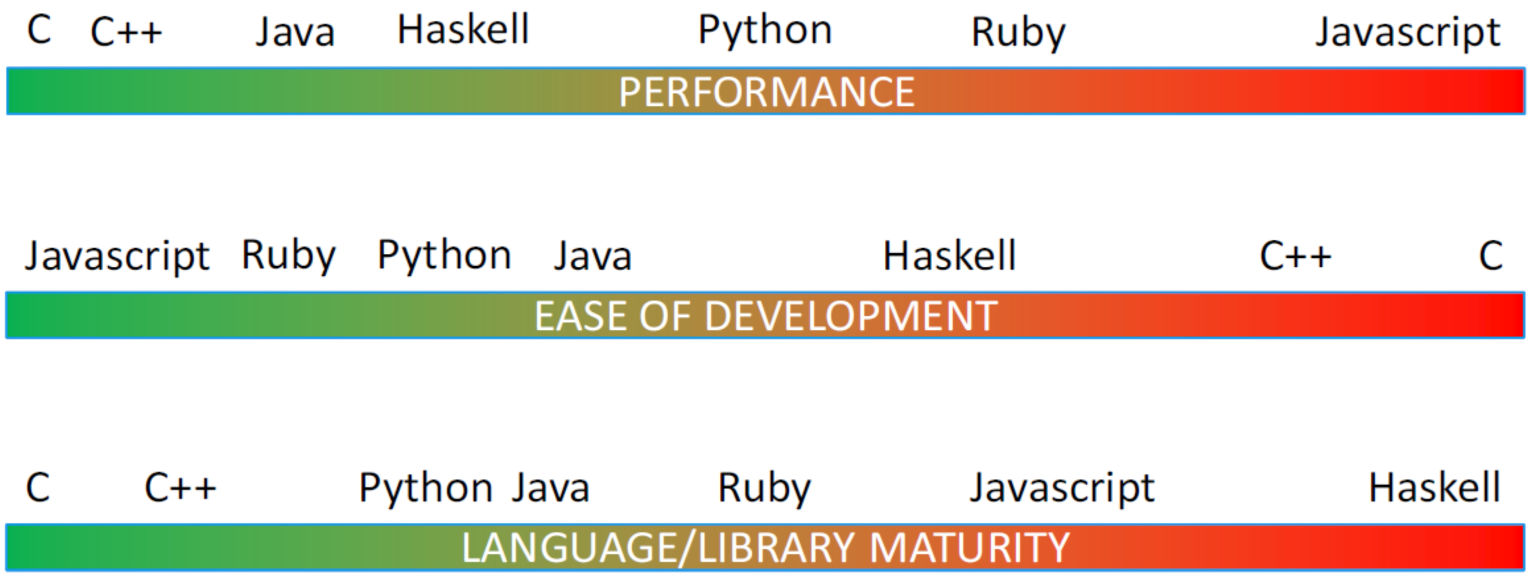
B. Packet parsing capabilities
   - When switch sends an OpenFlow message, module extracts relevant information using standard procedures

C. switch.send(msg), where msg can be
   - PACKET_OUT with buffer_id or fabricated packe
   - FLOW_MOD with match rules and action taken
   - FEATURE_REQUEST, STATS_REQUEST, BARRIER_REQUEST

**Figure 1. OpenFlow Protocol Software Driver Controller/Switch Interaction**



# Choice of Programming Language




| **Language** | **Fast Compilation** | **Managed Memory** | **Cross Platform** | **High Performance** |
|---|---|---|---|---|
| C# | ✓ | ✓ | | ? |
| Java | ✓ | ✓ | ✓ | ? |
| Python | ✓ | ✓ | ✓ | |

# OpenFlow Controller

TABLE VI
CONTROLLERS CLASSIFICATION

| Name | Architecture | Northbound API | Consistency | Faults | License | Prog. language | Version |
|---|---|---|---|---|---|---|---|
| Beacon [186] | centralized multi-threaded | ad-hoc API | no | no | GPLv2 | Java | v1.0 |
| DISCO [185] | distributed | REST | — | yes | — | Java | v1.1 |
| Fleet [199] | distributed | ad-hoc | no | no | — | — | v1.0 |
| Floodlight [189] | centralized multi-threaded | RESTful API | no | no | Apache | Java | v1.1 |
| HP VAN SDN [184] | distributed | RESTful API | weak | yes | — | Java | v1.0 |
| HyperFlow [195] | distributed | — | weak | yes | — | C++ | v1.0 |
| Kandoo [228] | hierarchically distributed | — | no | no | — | C, C++, Python | v1.0 |
| Onix [7] | distributed | NVP NBAPI | weak, strong | yes | commercial | Python, C | v1.0 |
| Maestro [188] | centralized multi-threaded | ad-hoc API | no | no | LGPLv2.1 | Java | v1.0 |
| Meridian [192] | centralized multi-threaded | extensible API layer | no | no | — | Java | v1.0 |
| MobileFlow [222] | — | SDMN API | — | — | — | — | v1.2 |
| MuL [229] | centralized multi-threaded | multi-level interface | no | no | GPLv2 | C | v1.0 |
| NOX [26] | centralized | ad-hoc API | no | no | GPLv3 | C++ | v1.0 |
| NOX-MT [187] | centralized multi-threaded | ad-hoc API | no | no | GPLv3 | C++ | v1.0 |
| NVP Controller [112] | distributed | — | — | — | commercial | — | — |
| OpenContrail [183] | — | REST API | no | no | Apache 2.0 | Python, C++, Java | v1.0 |
| OpenDaylight [13] | distributed | REST, RESTCONF | weak | no | EPL v1.0 | Java | v1.{0,3} |
| ONOS [117] | distributed | RESTful API | weak, strong | yes | — | Java | v1.0 |
| PANE [197] | distributed | PANE API | yes | — | — | — | — |
| POX [230] | centralized | ad-hoc API | no | no | GPLv3 | Python | v1.0 |
| ProgrammableFlow [231] | centralized | — | — | — | — | C | v1.3 |
| Rosemary [194] | centralized | ad-hoc | — | — | — | — | v1.0 |
| Ryu NOS [191] | centralized multi-threaded | ad-hoc API | no | no | Apache 2.0 | Python | v1.{0,2,3} |
| SMaRtLight [198] | distributed | RESTful API | no | no | Apache | Java | v1.0 |
| SNAC [232] | centralized | ad-hoc API | no | no | GPL | C++ | v1.0 |
| Trema [190] | centralized multi-threaded | ad-hoc API | no | no | GPLv2 | C, Ruby | v1.0 |
| *Unified Controller* [171] | — | REST API | — | — | commercial | — | v1.0 |
| *yanc* [196] | distributed | file system | — | — | — | — | — |

Diego Kreutz, Fernando M. V. Ramos, Paulo Verissimo, Christian Esteve Rothenberg, Siamak Azodolmolky, Steve Uhlig. "*Software-Defined Networking: A Comprehensive Survey.*" In Proceedings of the IEEE, Vol. 103, Issue 1, Jan. 2015

# OpenFlow Channel

- Used to exchange OpenFlow message between switch and controller.
- Switch can establish single or multiple connections to same or different controllers
- The SC connection is a TLS/TCP connection. Switch and controller mutually authenticate by exchanging certificates signed by a site-specific private key

Kingston Smiler. S, Introduction to OpenFlow, SDN & NFV

# OpenFlow Switch

- One or more flow tables, group table  and meter table

- Can be managed by one or more controllers.

- The flow tables and group table are used during the lookup or forwarding phase in order to forward the packet to appropriate port.

Kingston Smiler. S, Introduction to OpenFlow, SDN & NFV

# OpenFlow Switch

TABLE IV
OPENFLOW ENABLED HARDWARE AND SOFTWARE DEVICES

| Group | Product | Type | Maker/Developer | Version | Short description |
|---|---|---|---|---|---|
| Hardware | 8200zl and 5400zl [125] | chassis | Hewlett-Packard | v1.0 | Data center class chassis (switch modules). |
| | Arista 7150 Series [126] | switch | Arista Networks | v1.0 | Data centers hybrid Ethernet/OpenFlow switches. |
| | BlackDiamond X8 [127] | switch | Extreme Networks | v1.0 | Cloud-scale hybrid Ethernet/OpenFlow switches. |
| | CX600 Series [128] | router | Huawei | v1.0 | Carrier class MAN routers. |
| | EX9200 Ethernet [129] | chassis | Juniper | v1.0 | Chassis based switches for cloud data centers. |
| | EZchip NP-4 [130] | chip | EZchip Technologies | v1.1 | High performance 100-Gigabit network processors. |
| | MLX Series [131] | router | Brocade | v1.0 | Service providers and enterprise class routers. |
| | NoviSwitch 1248 [124] | switch | NoviFlow | v1.3 | High performance OpenFlow switch. |
| | NetFPGA [48] | card | NetFPGA | v1.0 | 1G and 10G OpenFlow implementations. |
| | RackSwitch G8264 [132] | switch | IBM | v1.0 | Data center switches supporting Virtual Fabric and OpenFlow. |
| | PF5240 and PF5820 [133] | switch | NEC | v1.0 | Enterprise class hybrid Ethernet/OpenFlow switches. |
| | Pica8 3920 [134] | switch | Pica8 | v1.0 | Hybrid Ethernet/OpenFlow switches. |
| | Plexxi Switch 1 [135] | switch | Plexxi | v1.0 | Optical multiplexing interconnect for data centers. |
| | V330 Series [136] | switch | Centec Networks | v1.0 | Hybrid Ethernet/OpenFlow switches. |
| | Z-Series [137] | switch | Cyan | v1.0 | Family of packet-optical transport platforms. |
| Software | contrail-vrouter [138] | vrouter | Juniper Networks | v1.0 | Data-plane function to interface with a VRF. |
| | LINC [139], [140] | switch | FlowForwarding | v1.4 | Erlang-based soft switch with OF-Config 1.1 support. |
| | ofsoftswitch13 [141] | switch | Ericsson, CPqD | v1.3 | OF 1.3 compatible user-space software switch implementation. |
| | Open vSwitch [142], [109] | switch | Open Community | v1.0-1.3 | Switch platform designed for virtualized server environments. |
| | OpenFlow Reference [143] | switch | Stanford | v1.0 | OF Switching capability to a Linux PC with multiple NICs. |
| | OpenFlowClick [144] | vrouter | Yogesh Mundada | v1.0 | OpenFlow switching element for Click software routers. |
| | Switch Light [145] | switch | Big Switch | v1.0 | Thin switching software platform for physical/virtual switches. |
| | Pantou/OpenWRT [146] | switch | Stanford | v1.0 | Turns a wireless router into an OF-enabled switch. |
| | XorPlus [46] | switch | Pica8 | v1.0 | Switching software for high performance ASICs. |

Diego Kreutz, Fernando M. V. Ramos, Paulo Verissimo, Christian Esteve Rothenberg, Siamak Azodolmolky, Steve Uhlig. "*Software-Defined Networking: A Comprehensive Survey.*" In Proceedings of the IEEE, Vol. 103, Issue 1, Jan. 2015

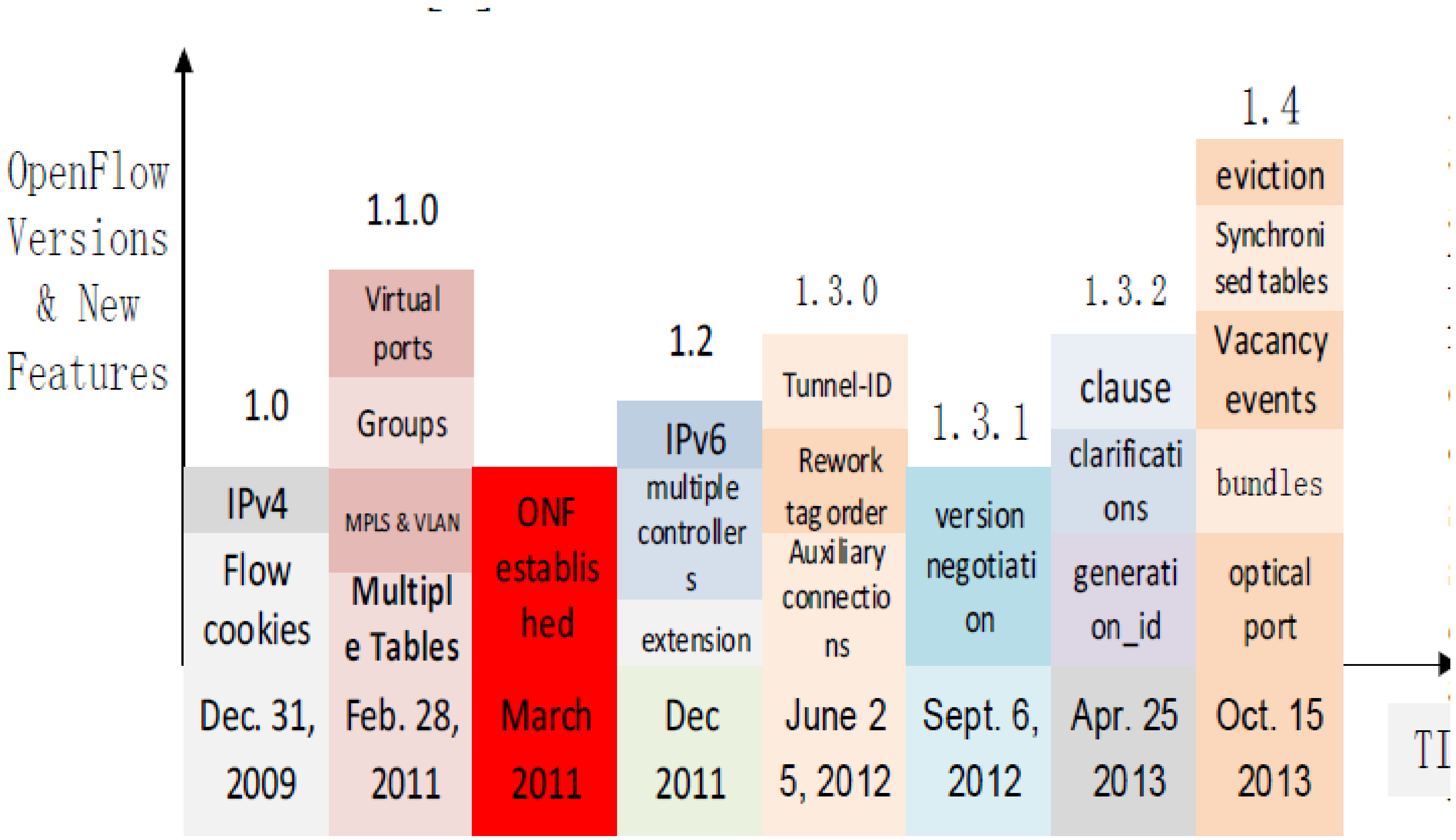



Figure 2: The history of OpenFlow protocol

http://beeyeas.blogspot.com.br/2014/06/openflow-evolution.html

# OpenFlow 1.0 Flow Table & Fields

Header Fields

| Ingress Port | Ethernet | | | VLAN | | IP | | | | TCP/UDP | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | SA | DA | Type | ID | Priority | SA | DA | Proto | TOS | Src | Dst |

**Flow Table**
*OF1.0 style*

| | | |
|---|---|---|
| Classifier | Action | Statistics |
| Classifier | Action | Statistics |
| Classifier | Action | Statistics |
| ⋮ | | |
| Classifier | Action | Statistics |

Actions

| | | |
|---|---|---|
| Forward | Physical Port | |
| | Virtual Port | ALL |
| | | CONTROLLER |
| | | LOCAL |
| | | TABLE |
| | | IN_PORT |
| Drop | | |
| Forward | Virtual Port | NORMAL |
| | | FLOOD |
| Enqueue | | |
| Modify Field | | |

*Mandatory Action*

*Optional Action*

# OpenFlow 1.2 Extensible match support

- Flow match fields described using the OpenFlow Extensible Match (OXM) format - a compact type-length-value (TLV) format

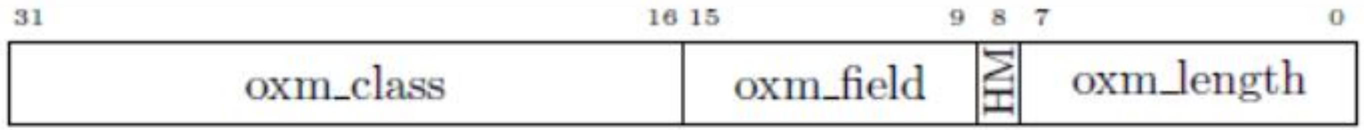
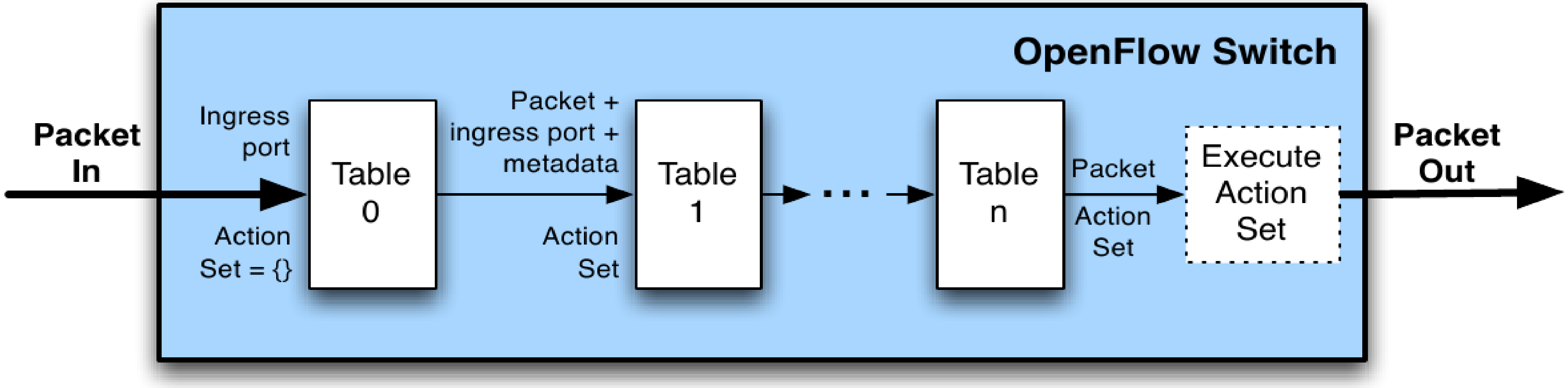



Figure 4: OXM TLV header layout

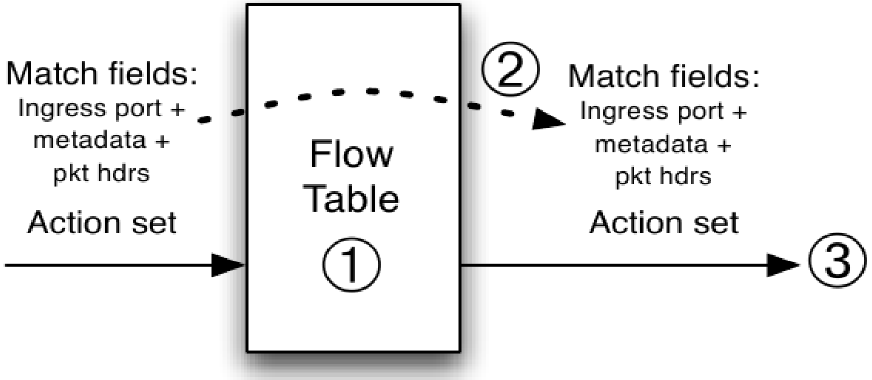
| Name | | Width | Usage |
|---|---|---|---|
| oxm_type | oxm_class | 16 | Match class: member class or reserved class |
| | oxm_field | 7 | Match field within the class |
| | oxm_hasmask | 1 | Set if OXM include a bitmask in payload |
| | oxm_length | 8 | Length of OXM payload |

Table 9: OXM TLV header fields

# OpenFlow 1.3 Pipeline




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




① Find highest-priority matching flow entry

② Apply instructions:

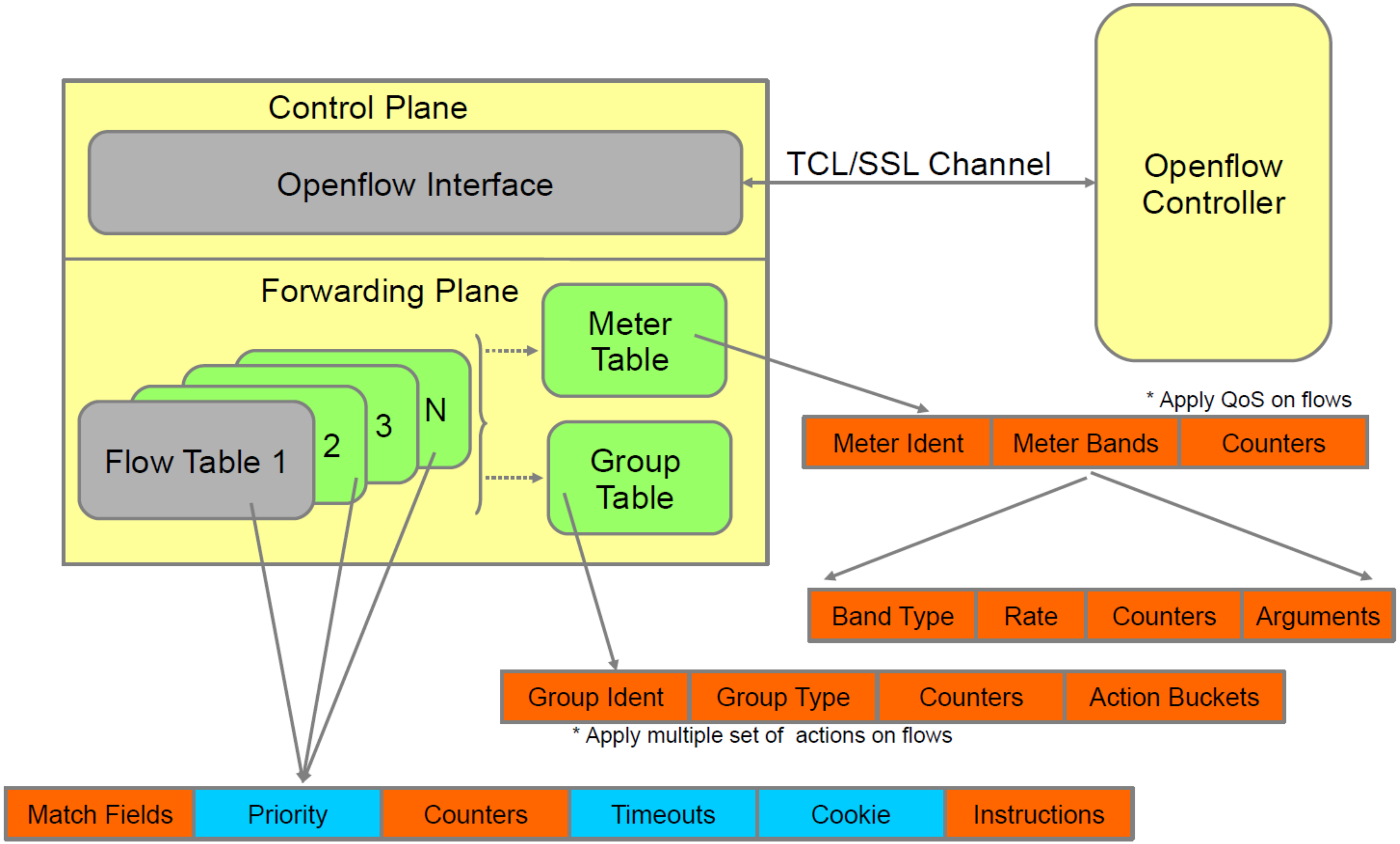
i. Modify packet & update match fields (apply actions instruction)

ii. Update action set (clear actions and/or write actions instructions)

iii. Update metadata

③ Send match data and action set to next table

# OpenFlow 1.3



Control Plane
Openflow Interface
TCL/SSL Channel
Openflow Controller
Forwarding Plane
Meter Table
Group Table
Flow Table 1
2
3
N
* Apply QoS on flows
Meter Ident
Meter Bands
Counters
Band Type
Rate
Counters
Arguments
Group Ident
Group Type
Counters
Action Buckets
* Apply multiple set of actions on flows
Match Fields
Priority
Counters
Timeouts
Cookie
Instructions

OPEN NETWORKING FOUNDATION

# OpenFlow version 1.4.0

OpenFlow Switch Specification

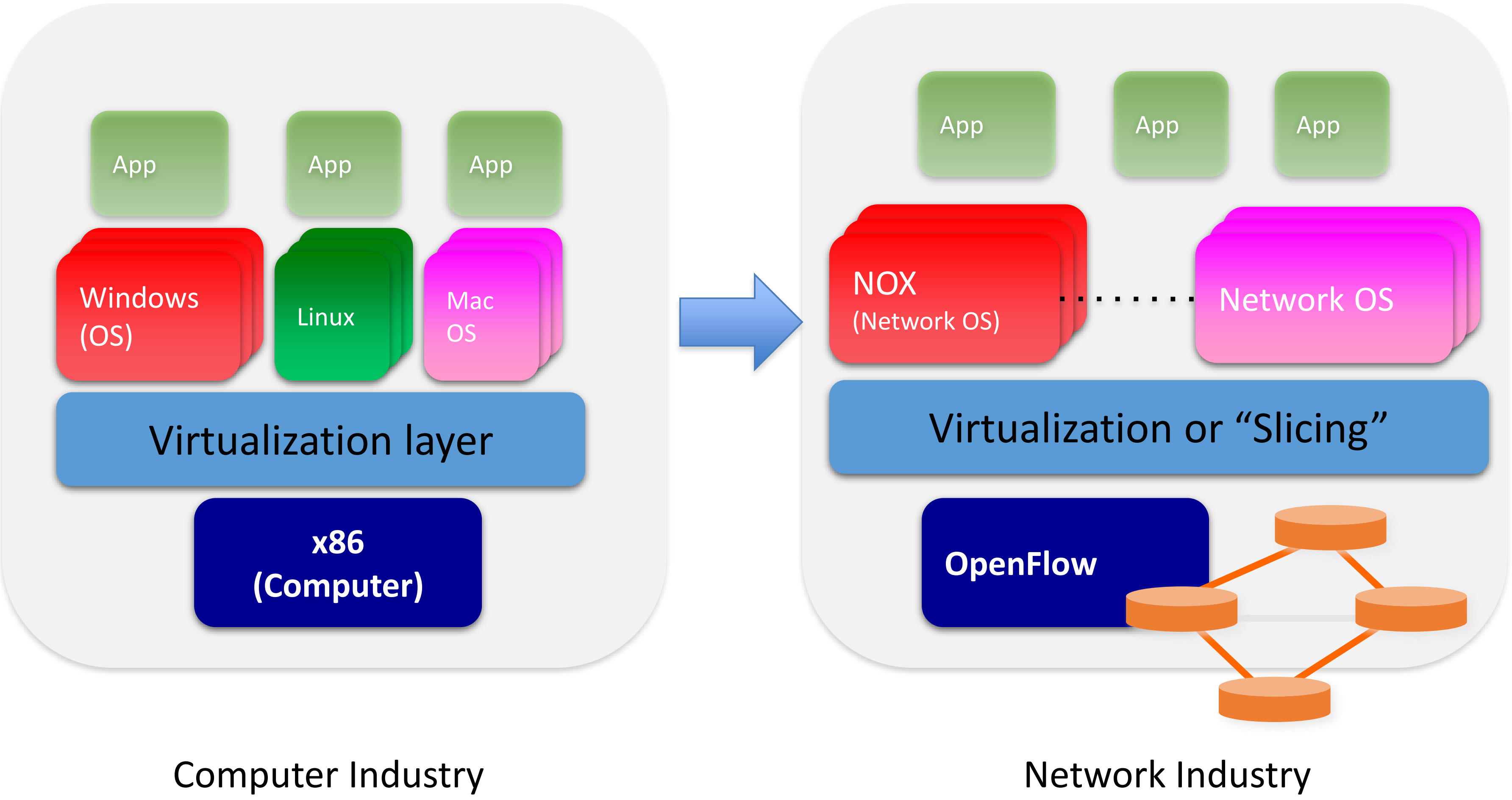
Version 1.4.0 (Wire Protocol 0x05)
August 5, 2013

- Released Aug 2013
- Based on OpenFlow 1.3
- More flexibility :
  - Flexible ports, flexible table-mods, flex set-async
- More features :
  - Bundles (group of OpenFlow requests)
  - Optical port properties
  - Flow entry monitoring and notifications
  - Group and meter change notifications
  - Role status events
  - Flow entry eviction
  - Flow table vacancy events
  - Synchronised tables (ex. learning tables)
  - Other minor features (see changelog)
- Features also available as 1.3.X extensions

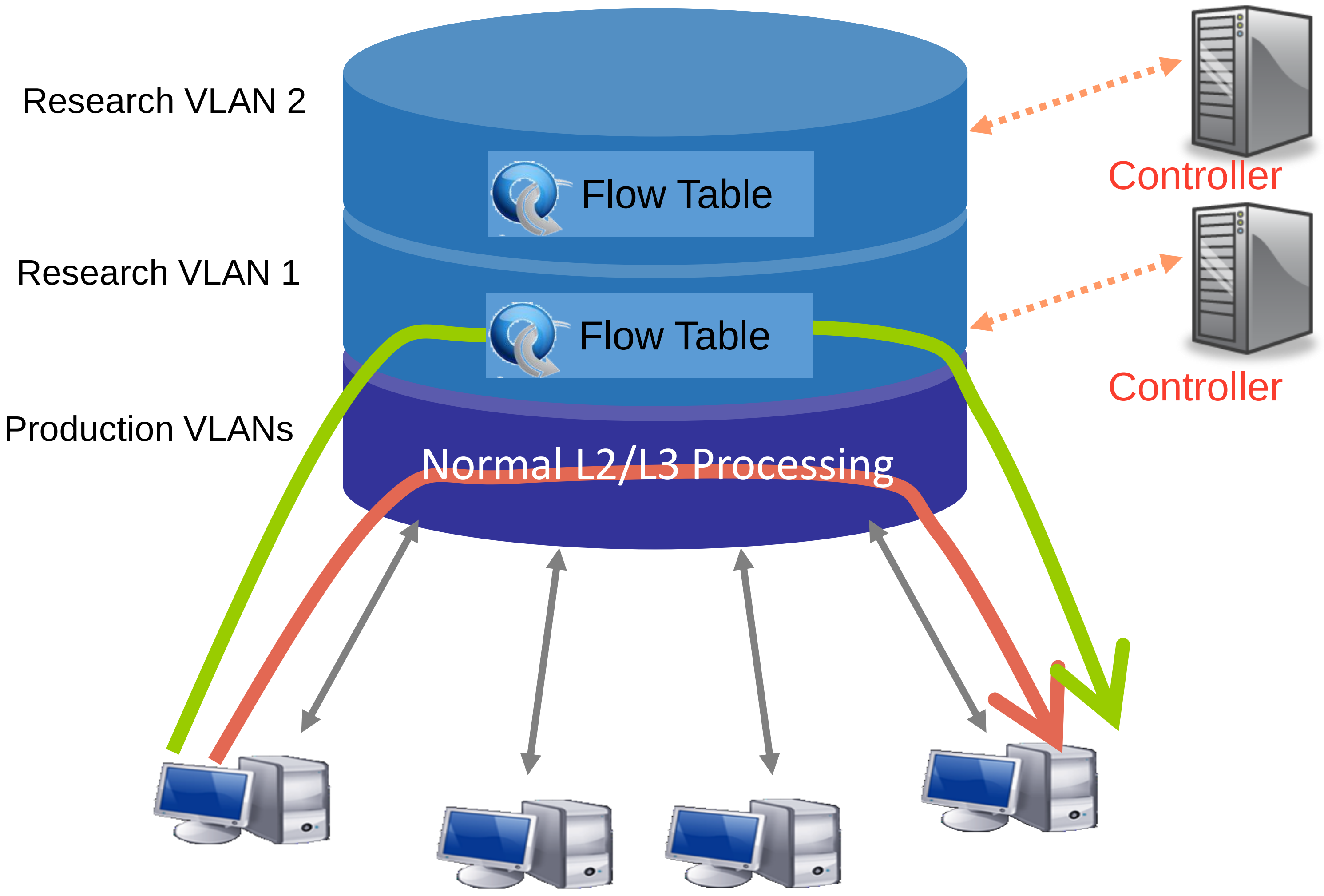
# OpenFlow 1.5.0

1. Egress Tables
2. Packet Type aware pipeline
3. Extensible Flow Entry Statistics
4. Flow Entry Statistics Trigger
5. Copy-Field action to copy between two OXM fields
6. Packet Register pipeline fields
7. TCP flags matching
8. Group command for selective bucket operation
9. Alloc set-field action to set metadata field
10. Allow wildcard to be used in set-field action
11. Scheduled Bundles
12. Controller connection status
13. Meter action
14. Enable setting all pipeline fields in packet-out
15. Port properties for pipeline fields
16. Port property for recirculation
17. Clarify and improve barrier
18. Always generate port status on port config change
19. Make all Experimenter OXM-IDs 64 bits
20. Unified requests for group, port and queue multiparts
21. Rename some type for consistency
22. Specification reorganisation

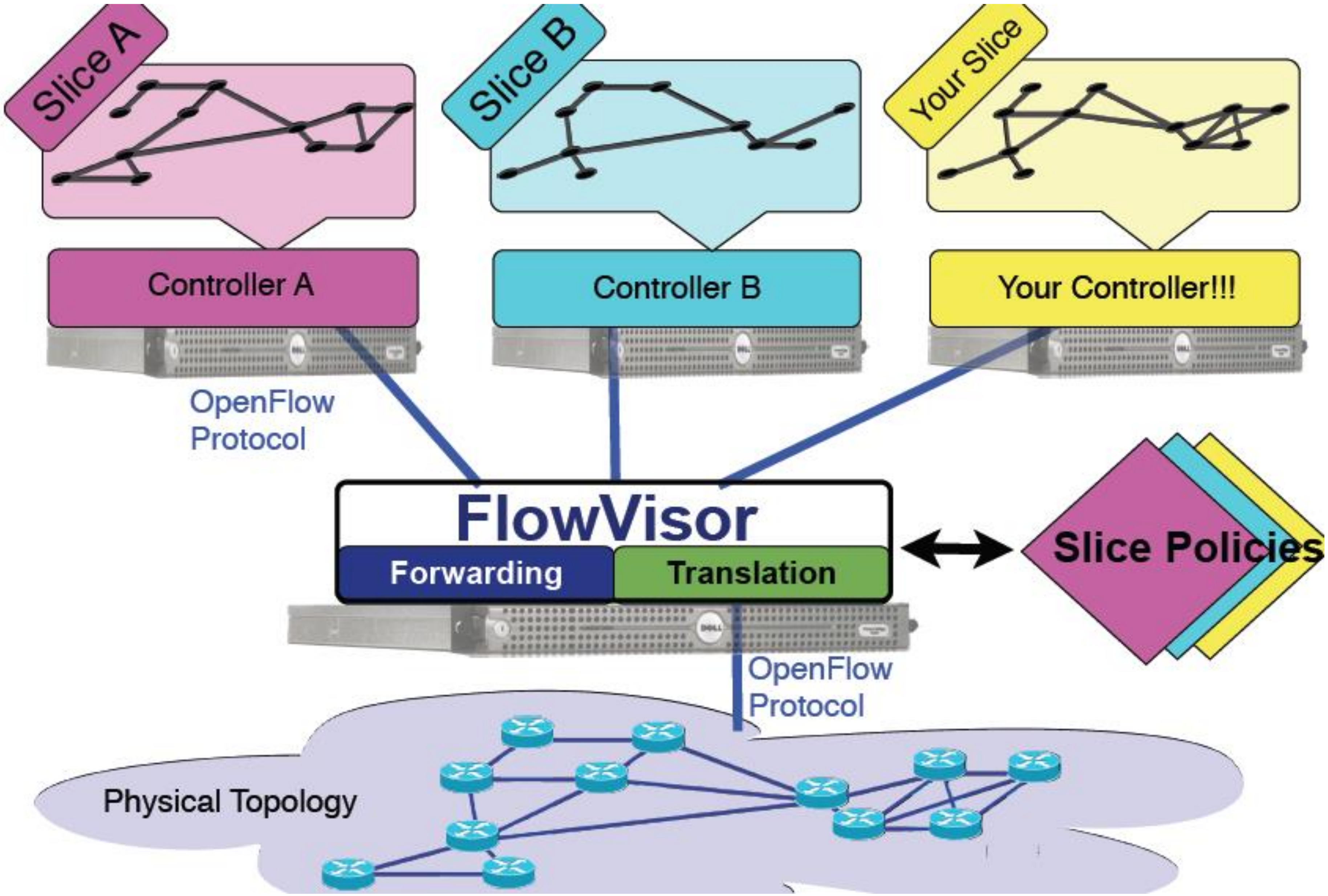
# Virtualization



# Switch Based Virtualization



# Flowvisor Virtualization

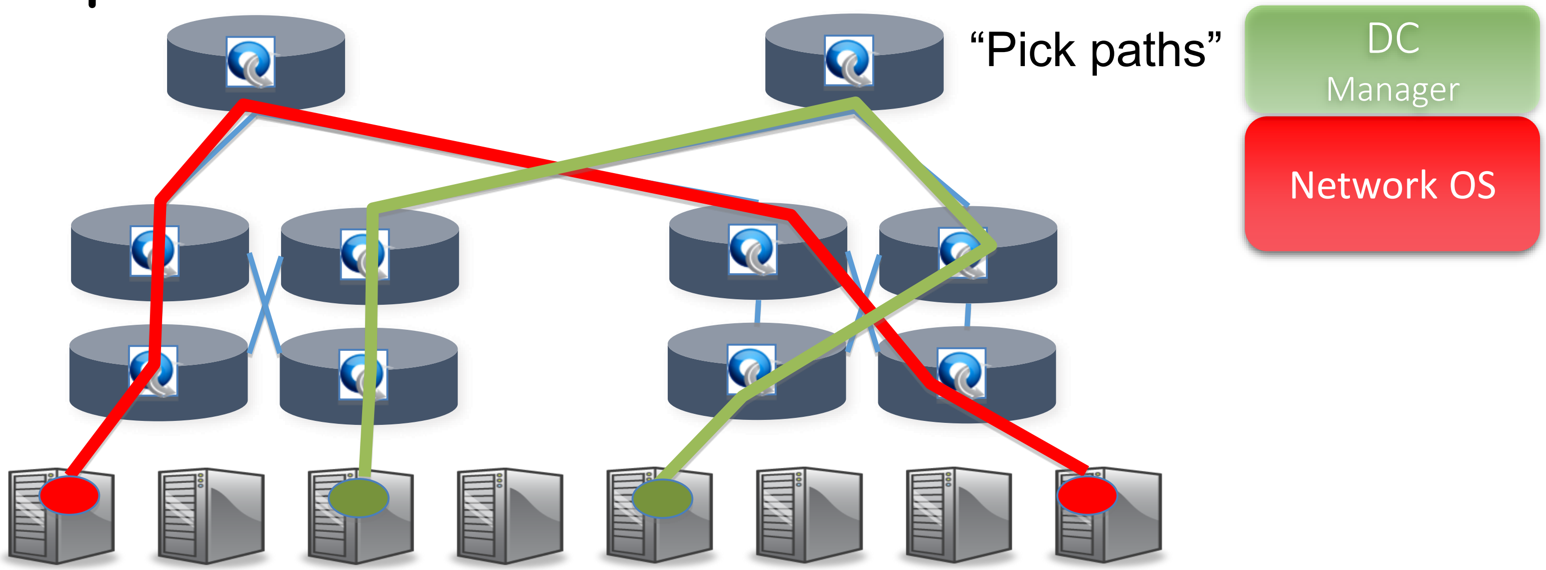


# ElasticTree

**Goal:** Reduce energy usage in data center networks

**Approach:**

1. Reroute traffic
2. Shut off links and switches to reduce power



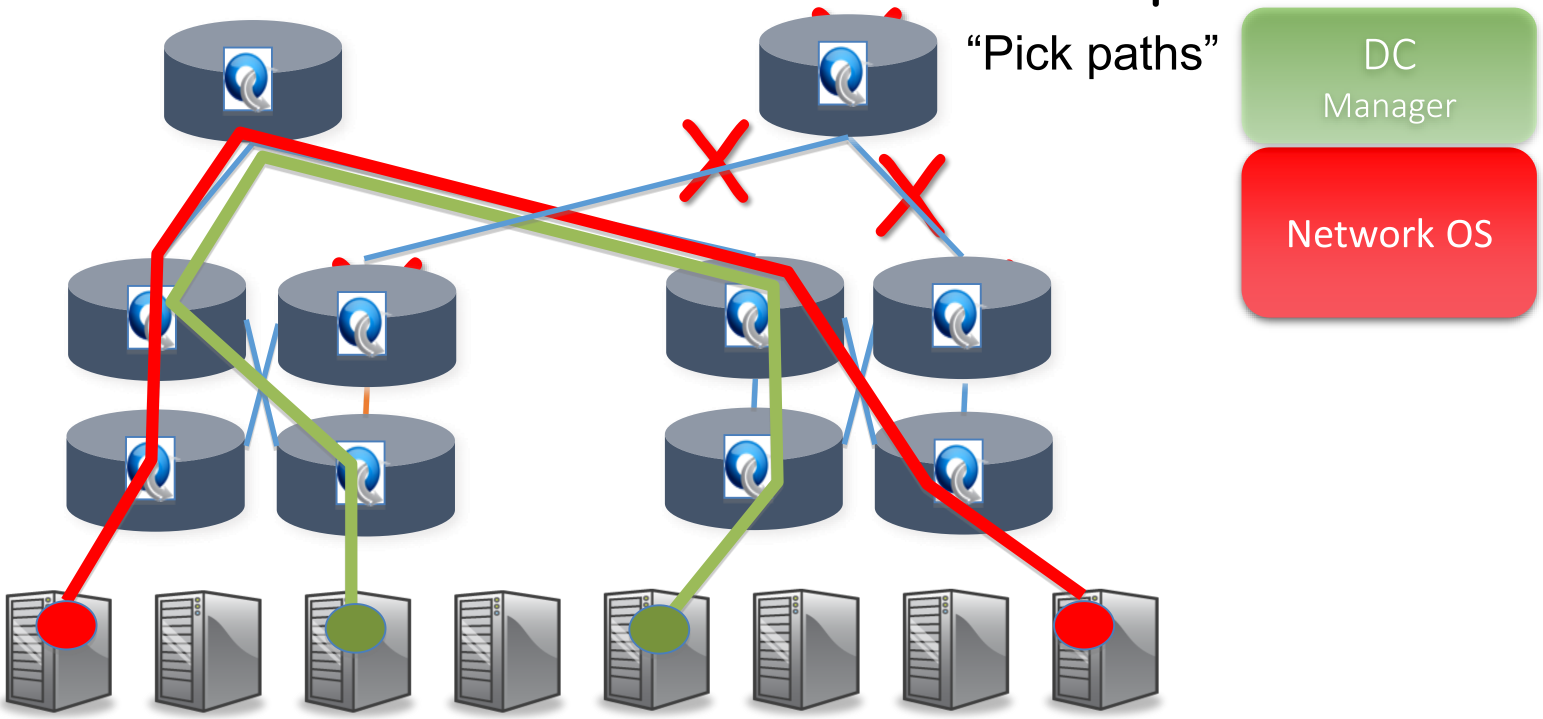

[Brandon Heller, NSDI 2010]

# ElasticTree

**Goal:** Reduce energy usage in data center networks

**Approach:**

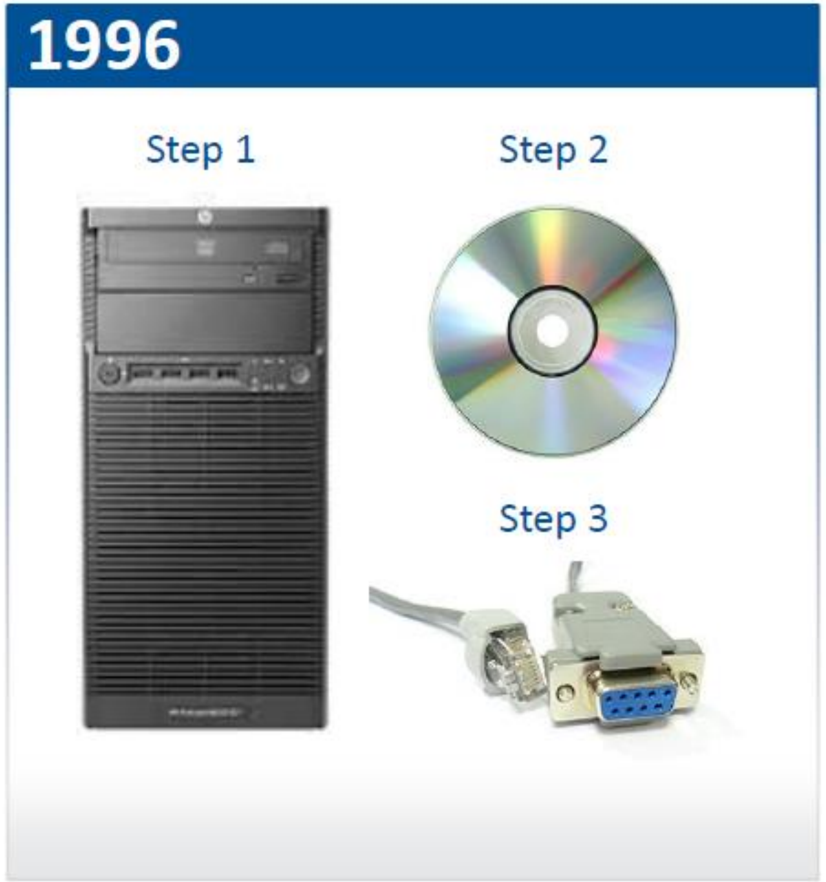
1. Reroute traffic
2. Shut off links and switches to reduce power




[Brandon Heller, NSDI 2010]

# SDN

# Traditional Vs Modern Computing Provisioning Methods





Source: Adopted from Transforming the Network With Open SDN by Big Switch Network

# Traditional Vs Modern Networking Provisioning Methods

**1996**

```
Router> enable
Router# configure terminal
Router(config)# enable secret cisco
Router(config)# ip route 0.0.0.0 0.0.0.0 20.2.2.3
Router(config)# interface ethernet0
Router(config-if)# ip address 10.1.1.1 255.0.0.0
Router(config-if)# no shutdown
Router(config-if)# exit
Router(config)# interface serial0
Router(config-if)# ip address 20.2.2.2 255.0.0.0
Router(config-if)# no shutdown
Router(config-if)# exit
Router(config)# router rip
Router(config-router)# network 10.0.0.0
Router(config-router)# network 20.0.0.0
Router(config-router)# exit
Router(config)# exit
Router# copy running-config startup-config
Router# disable
Router>
```

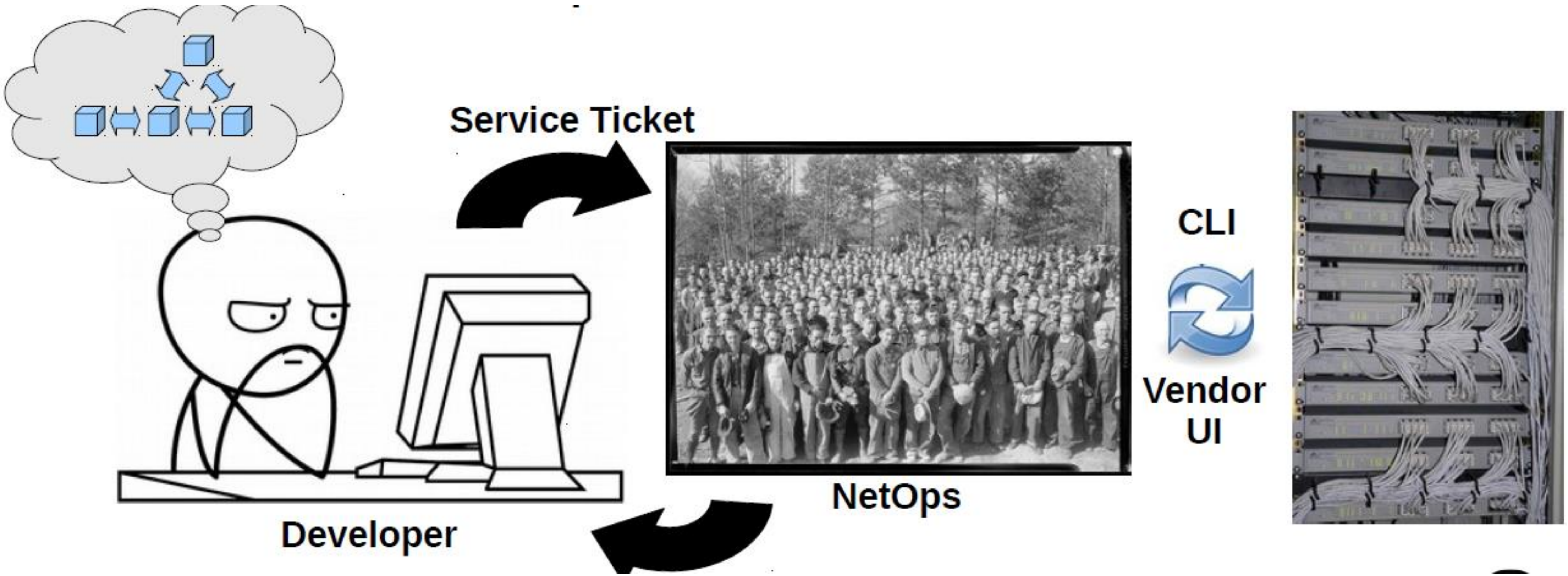
Terminal Protocol: **Telnet**

**2013**

```
Router> enable
Router# configure terminal
Router(config)# enable secret cisco
Router(config)# ip route 0.0.0.0 0.0.0.0 20.2.2.3
Router(config)# interface ethernet0
Router(config-if)# ip address 10.1.1.1 255.0.0.0
Router(config-if)# no shutdown
Router(config-if)# exit
Router(config)# interface serial0
Router(config-if)# ip address 20.2.2.2 255.0.0.0
Router(config-if)# no shutdown
Router(config-if)# exit
Router(config)# router rip
Router(config-router)# network 10.0.0.0
Router(config-router)# network 20.0.0.0
Router(config-router)# exit
Router(config)# exit
Router# copy running-config startup-config
Router# disable
Router>
```

Terminal Protocol: **SSH**

Source: Adopted from Transforming the Network With Open SDN by Big Switch Network


Service Ticket
Developer
NetOps
CLI
Vendor
UI

# Software Defined Networking

In the Software Defined Networking architecture, the control and data planes are decoupled, network intelligence and state are logically centralized, and the underlying network infrastructure is abstracted from the applications.

**Software-Defined Networking:**
**The New Norm for Networks**
**ONF White Paper**
**April 13, 2012**

# What is SDN?

## SDN Definition

**Centralization** of control of the network via the

**Separation** of **control** logic to off-device compute, that

Enables **automation** and **orchestration** of network services via

Open **programmatic** interfaces

## SDN Benefits

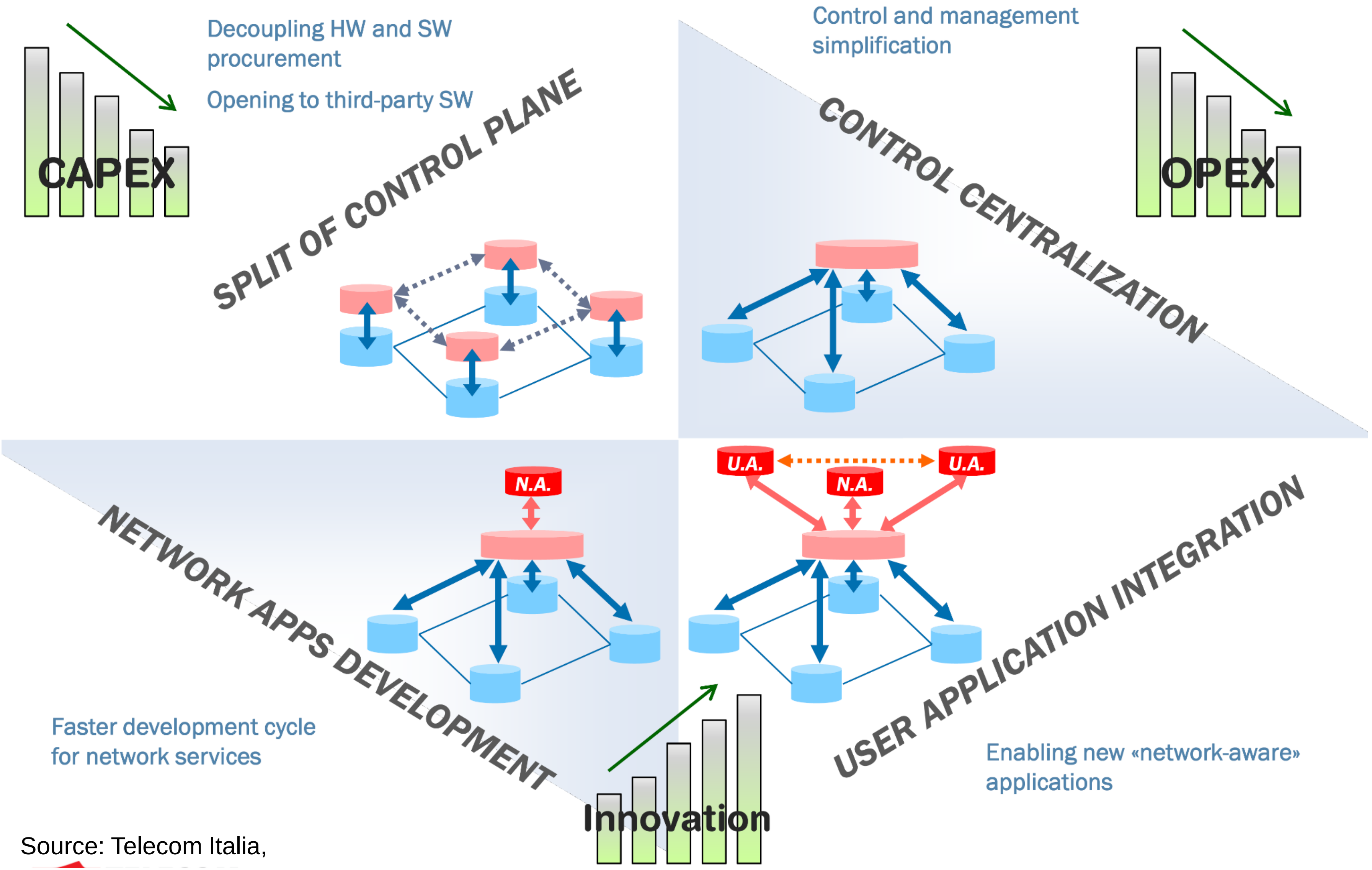
**Efficiency:** optimize existing applications, services, and infrastructure

**Scale:** rapidly grow existing applications and services

**Innovation:** create and deliver new types of applications and services and business models

Source: Adopted from SDN Central (Software-Defined Networking (SDN) Use Cases)

# SDN Drivers



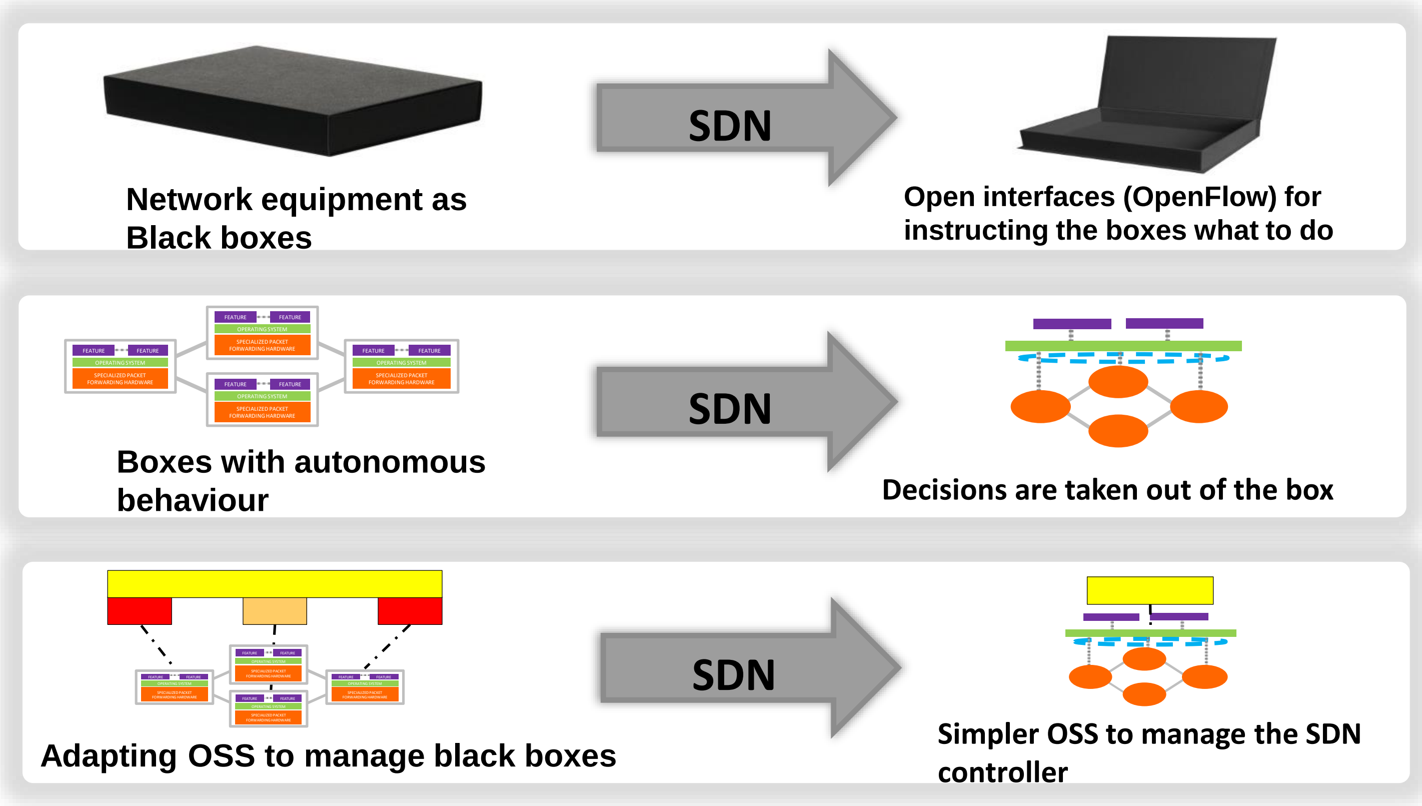

Source: Telecom Italia,

# SDN Approach

| FROM | TO |
| --- | --- |
| Hardware/Appliances | (Open) Software |
| Custom ASICs/FPGAs | Merchant Silicon |
| Distributed Control Plane | (Logically) Centralized Control Plane |
| Protocols | APIs |
| Function-Specific Features | Policy-based Apps and Services |
| Vendor-controlled Releases | Rapid Innovation Cycles |

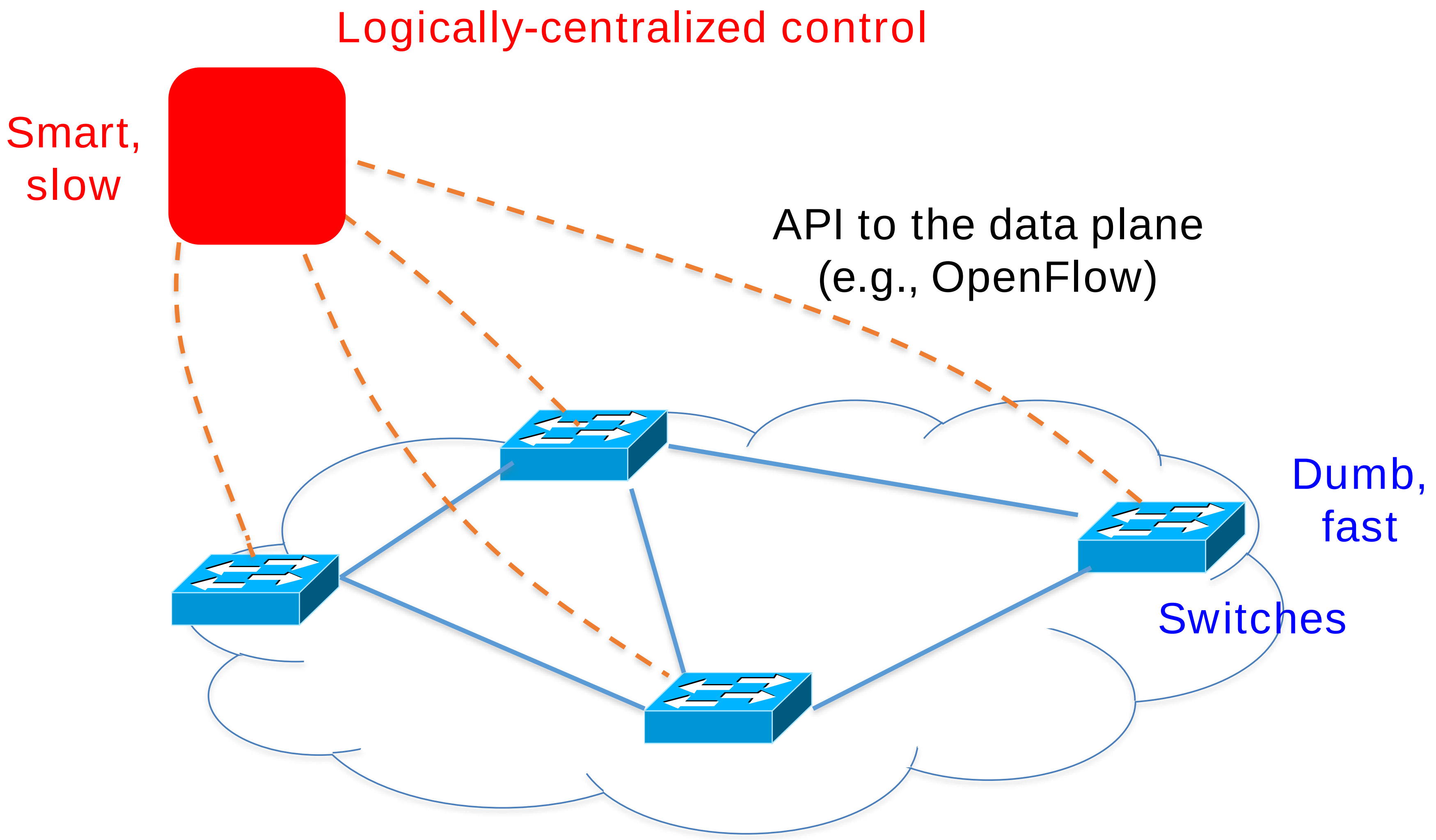
Source: Adapted from ONS12 Presentation by Dan Pitt

# Software Defined Networking (SDN)

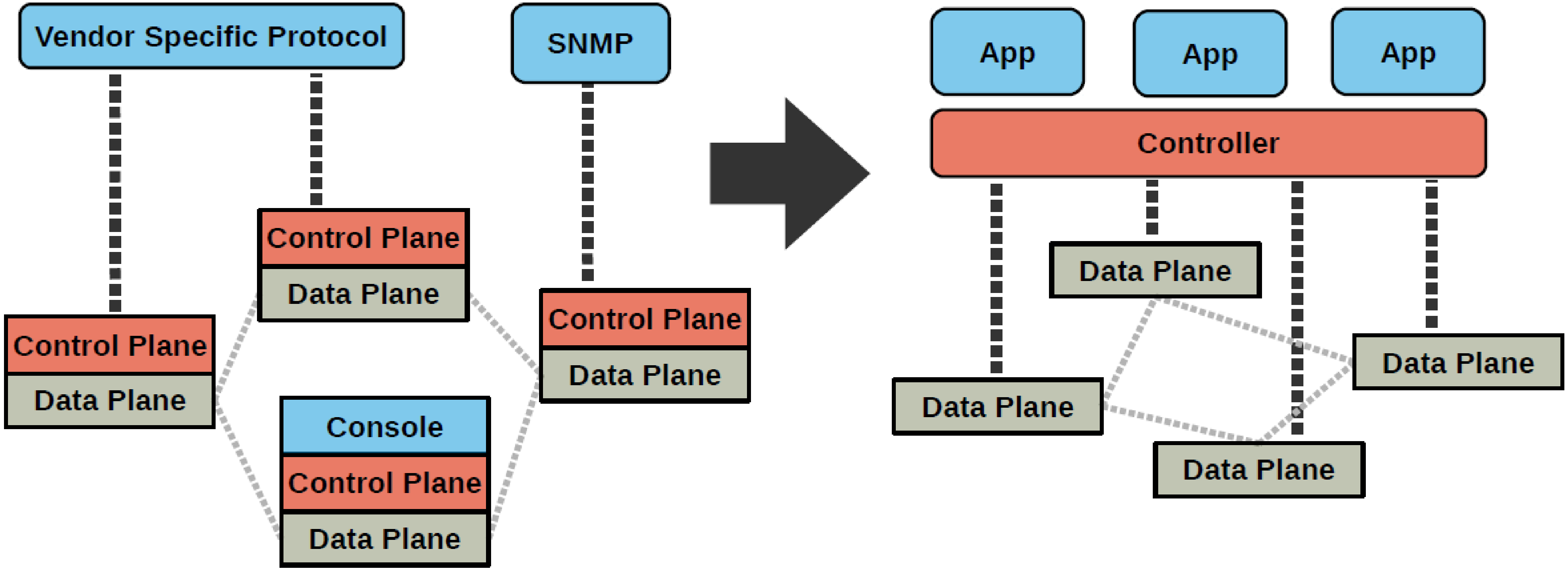



Source: Adapted from D. Lopez Telefonica I+D, NFV

# Software Defined Networking (SDN)

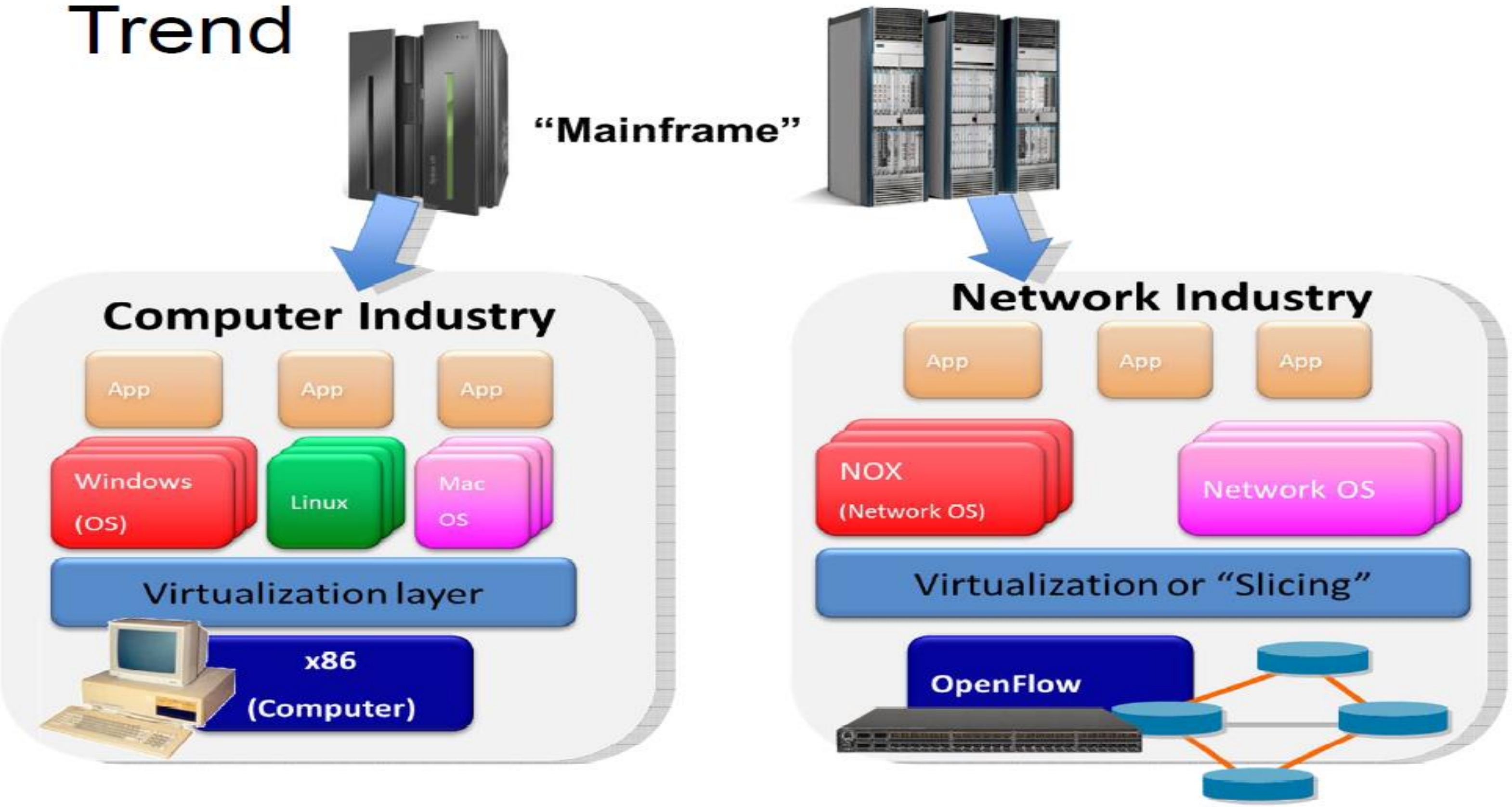



Source: Adapted from J. Rexford



Vendor Specific Protocol
SNMP
Control Plane
Data Plane
Control Plane
Data Plane
Control Plane
Data Plane
Console
Control Plane
Data Plane
App
App
App
Controller
Data Plane
Data Plane
Data Plane
Data Plane

# Trend

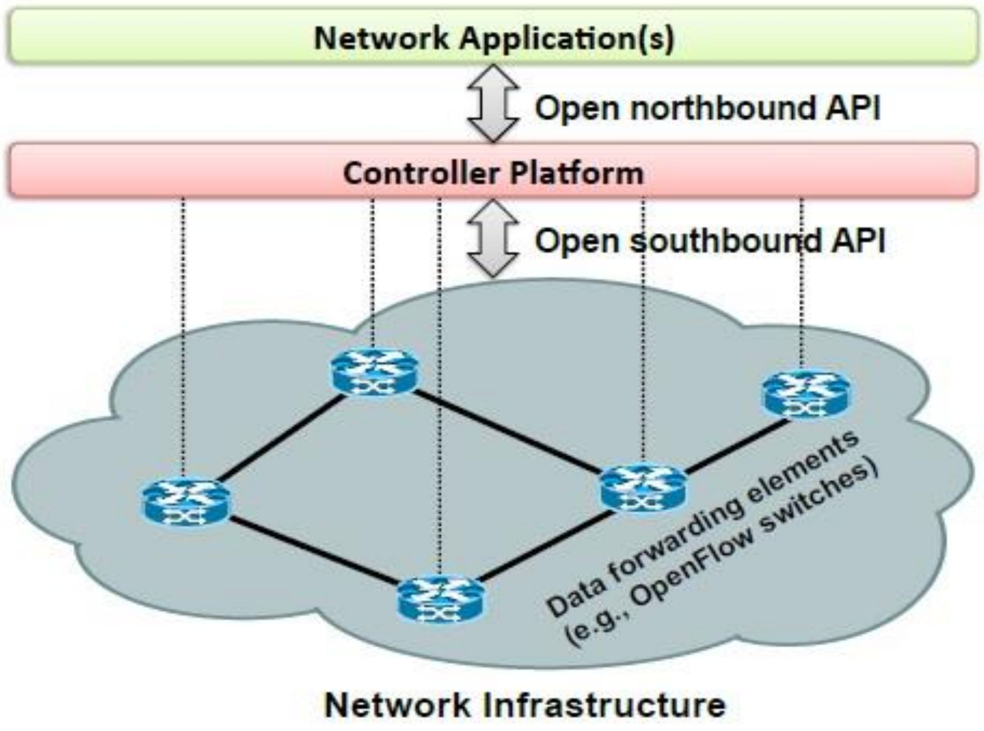


"Mainframe"
Computer Industry
App
App
App
Windows
(OS)
Linux
Mac
OS
Virtualization layer
x86
(Computer)
Network Industry
App
App
App
NOX
(Network OS)
Network OS
Virtualization or "Slicing"
OpenFlow

# SDN: Definitions, Concepts, and Terminology

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 (abstract) data plane

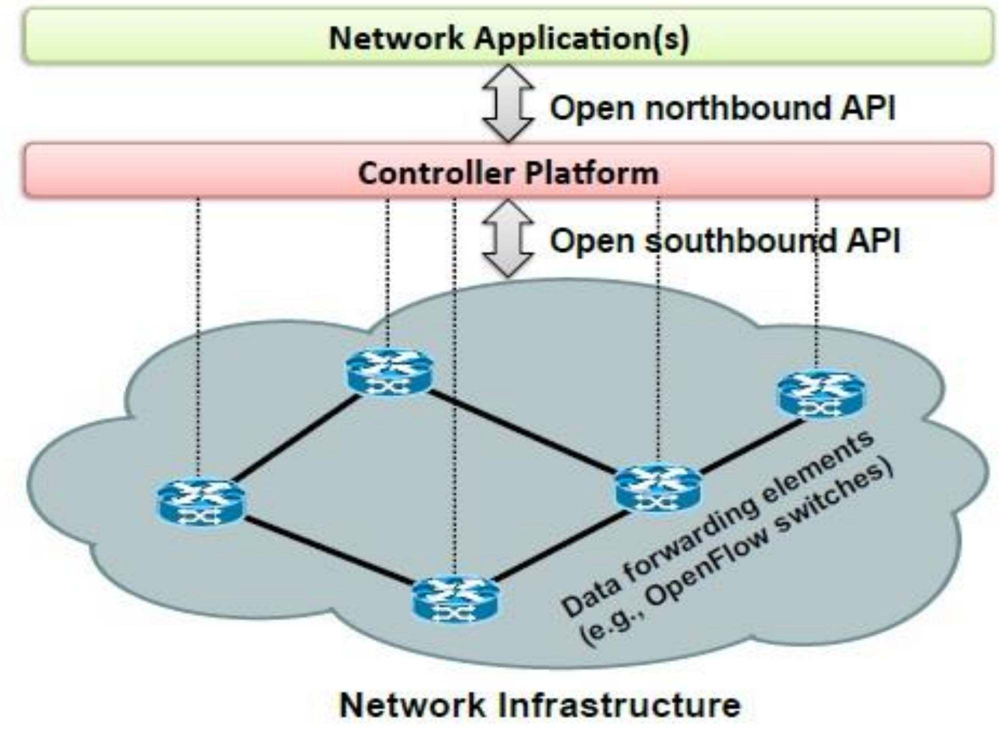



Source:
"Software-Defined Networking: A Comprehensive Survey",
Kreutz et al., In Proceedings of the IEEE, Vol. 103, Issue 1, Jan. 2015..

# SDN: Definitions, Concepts, and Terminology

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

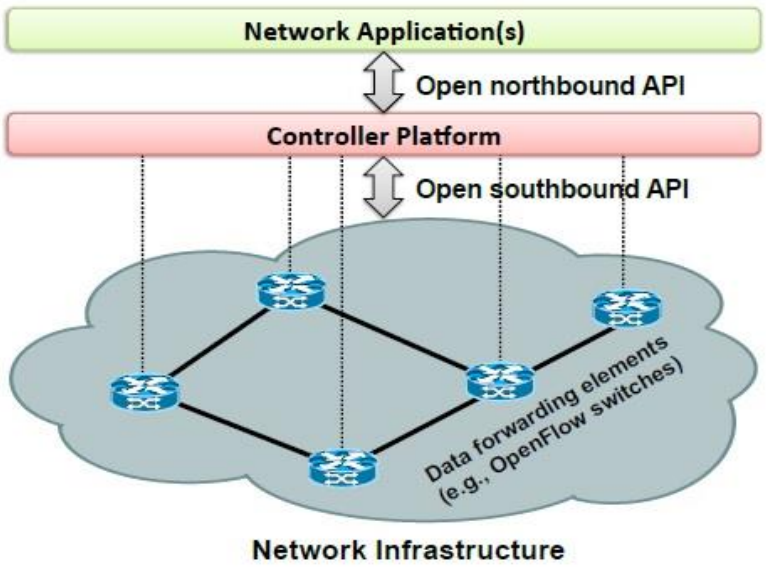
- **Southbound interface:**
  (instruction set to program the data plane
  +
  (protocol btw control- and data planes).
  E.g., OpenFlow, POF, Forces, Netconf




Source:
"Software-Defined Networking: A Comprehensive Survey",
Kreutz et al., In Proceedings of the IEEE, Vol. 103, Issue 1, Jan. 2015..

# SDN: Definitions, Concepts, and Terminology

- **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.



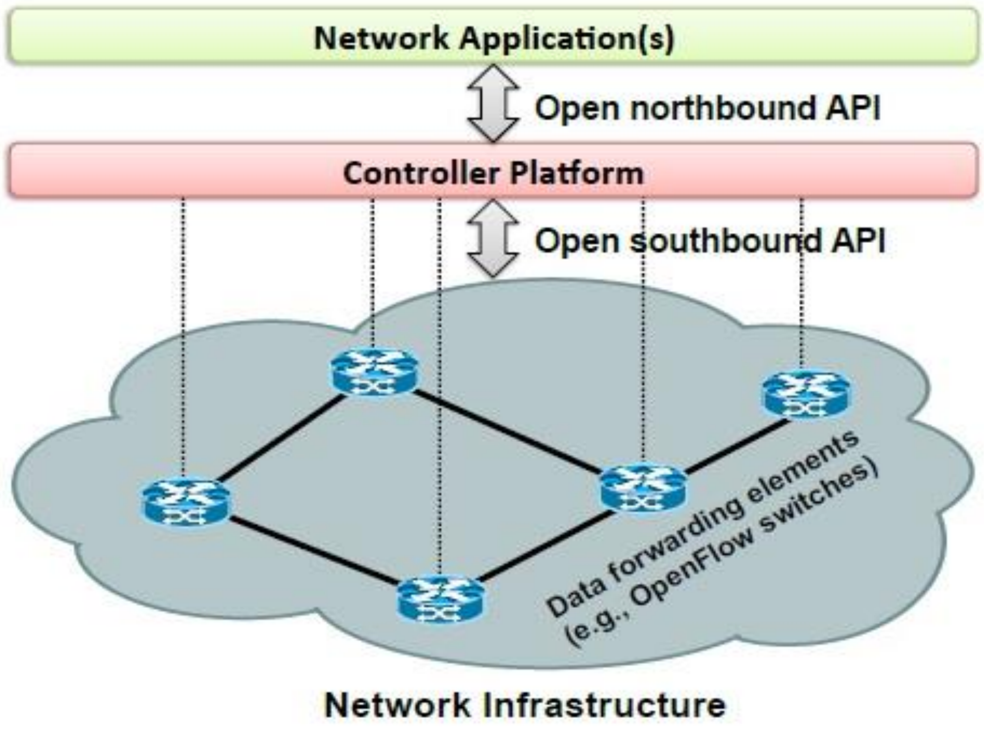

| | Switch port | MAC src | MAC dst | Eth type | VLAN ID | IP Src | IP Prot | TCP sport | TCP dport | Action |
|---|---|---|---|---|---|---|---|---|---|---|
| Switching | * | * | 00:1f :.. | * | * | * | * | * | * | Port6 |
| Flow switching | Port3 | 00:20 .. | 00:1f .. | 0800 | Vlan1 | 1.2.3.4 | 5.6.7.8 | 4 | 17264 | Port6 |
| Firewall | * | * | * | * | * | * | * | * | 22 | Drop |
| Routing | * | * | * | * | * | * | 5.6.7.8 | * | * | Port6 |
| VLAN switching | * | * | 00:1f .. | * | Vlan1 | * | * | * | * | Port6,port7, port8 |

Source:
“Software-Defined Networking: A Comprehensive Survey”,
Kreutz et al., In Proceedings of the IEEE, Vol. 103, Issue 1, Jan. 2015..

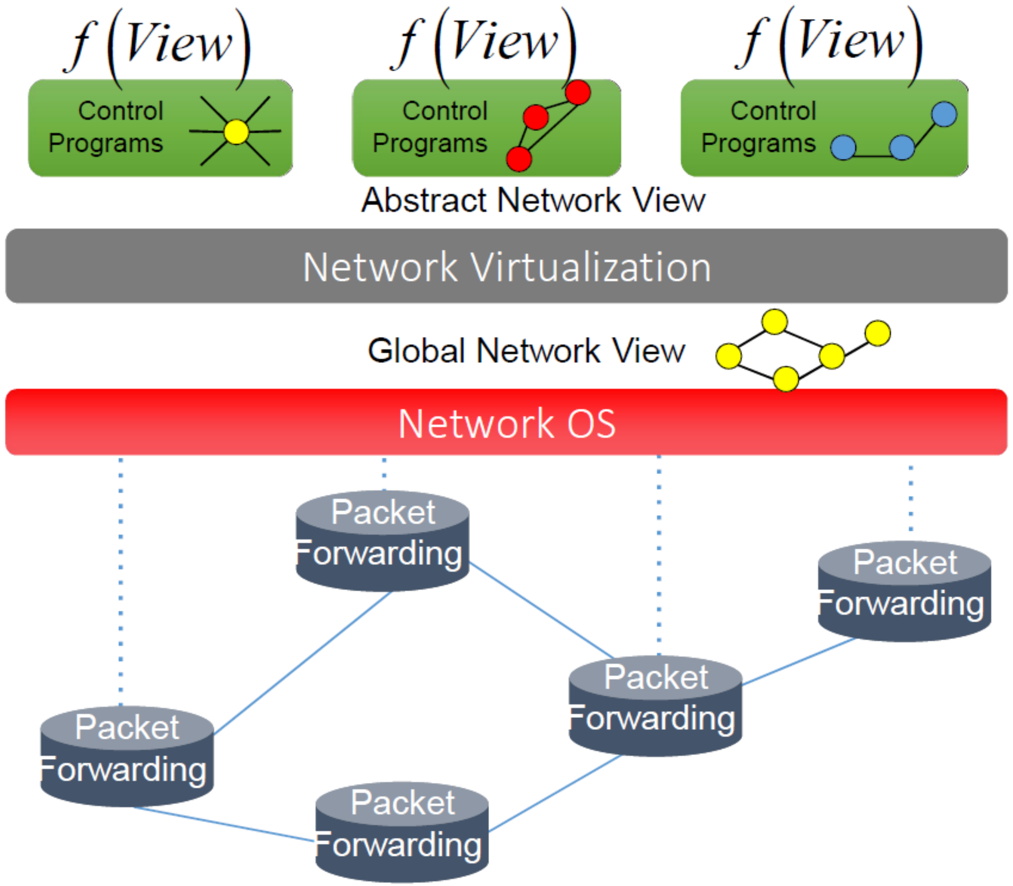
# SDN: Definitions, Concepts, and Terminology

- **Northbound interface:** API offered by control plane to develop network control- and management applications.

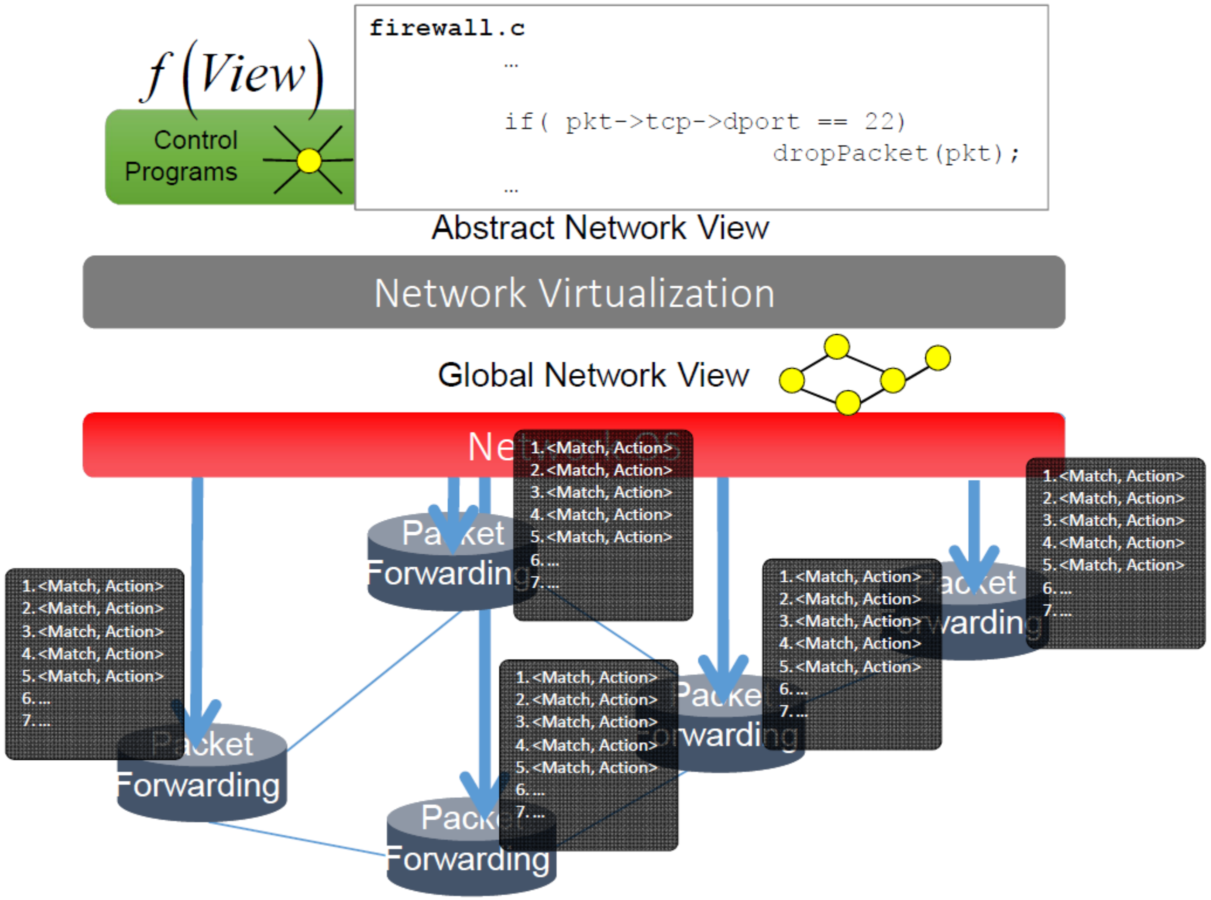
- **Application Layer / Business Applications (Management plane):** functions, e.g., routing, traffic engineering, that use Controller functions / APIs to manage and control network infrastructure.



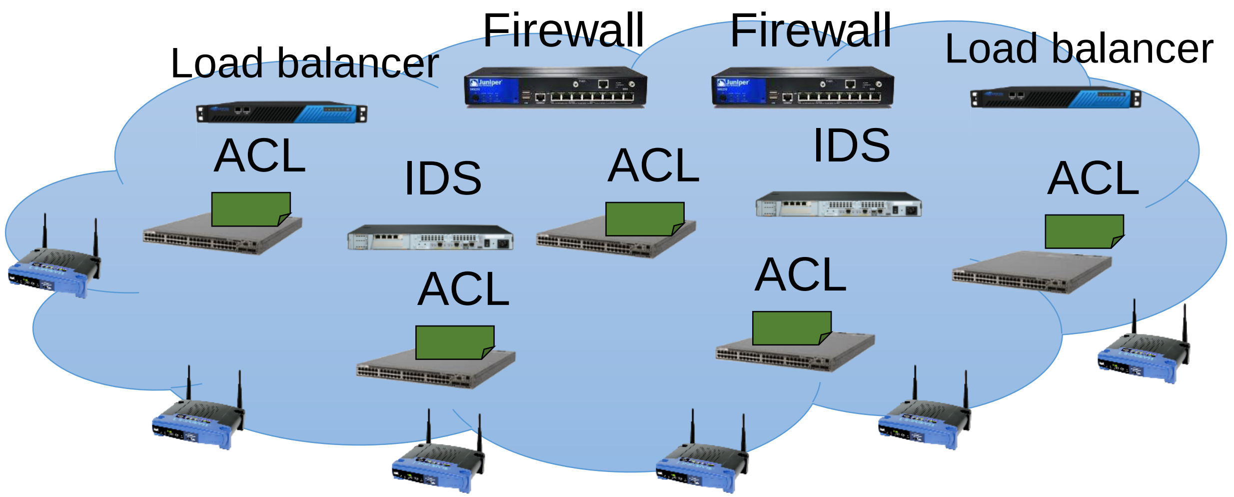

Source:
"Software-Defined Networking: A Comprehensive Survey",
Kreutz et al., In Proceedings of the IEEE, Vol. 103, Issue 1, Jan. 2015..



$f(View)$
Control Programs
$f(View)$
Control Programs
$f(View)$
Control Programs
Abstract Network View
Network Virtualization
Global Network View
Network OS
Packet Forwarding
Packet Forwarding
Packet Forwarding
Packet Forwarding
Packet Forwarding


f(View)
Control Programs
firewall.c
…
if( pkt->tcp->dport == 22)
dropPacket(pkt);
…
Abstract Network View
Network Virtualization
Global Network View
1. <Match, Action>
2. <Match, Action>
3. <Match, Action>
4. <Match, Action>
5. <Match, Action>
6. …
7. …
Forwarding

# Enterprise Network: Current solution



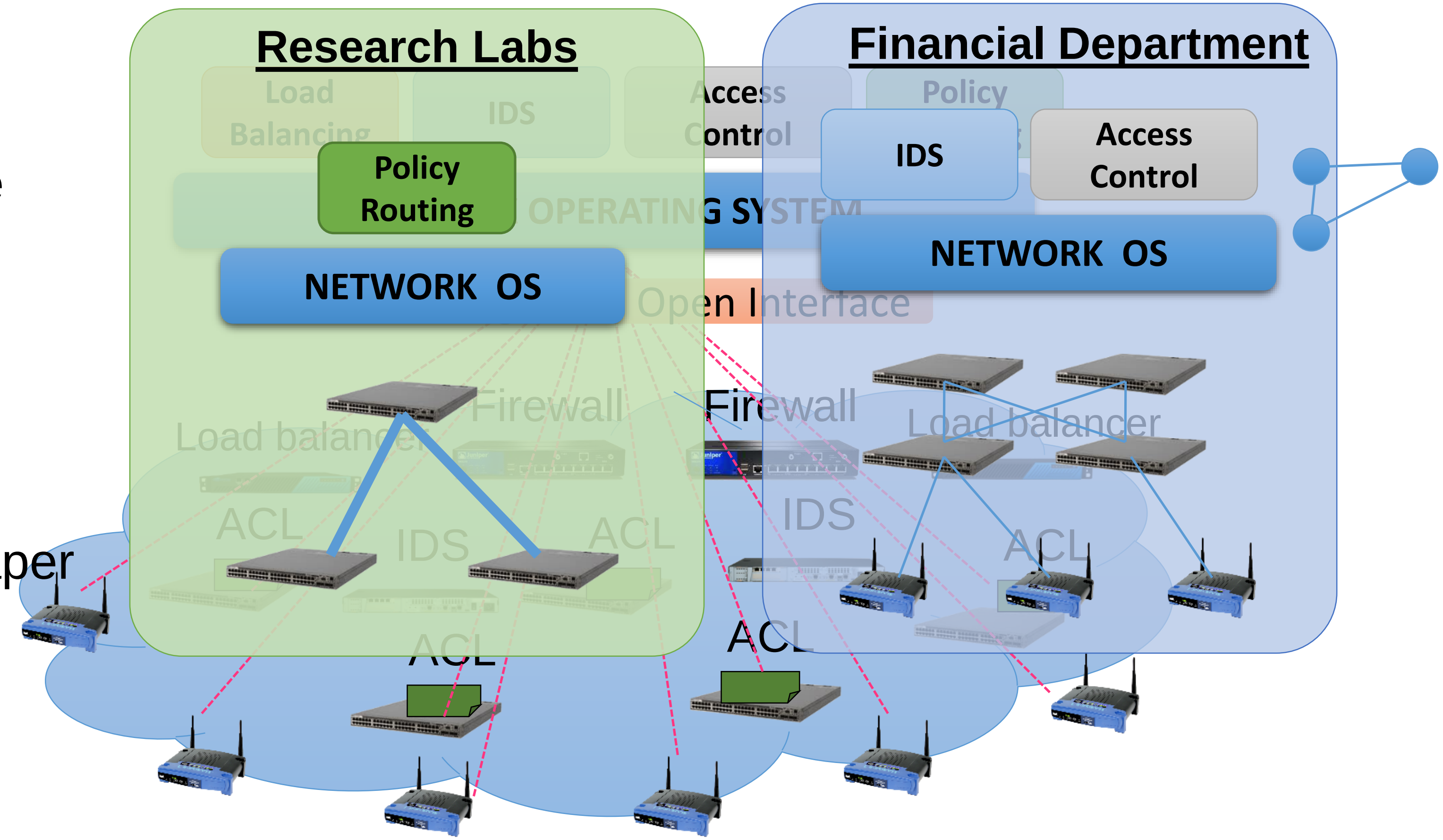

- Proliferation of appliances
- Increased management complexity
  - Device oriented management
  - Each device type has its own management
- High CAPEX, high OPEX
- Too much reliance on vendors

# Enterprise Network with SDN

And you can even delegate control to someone else

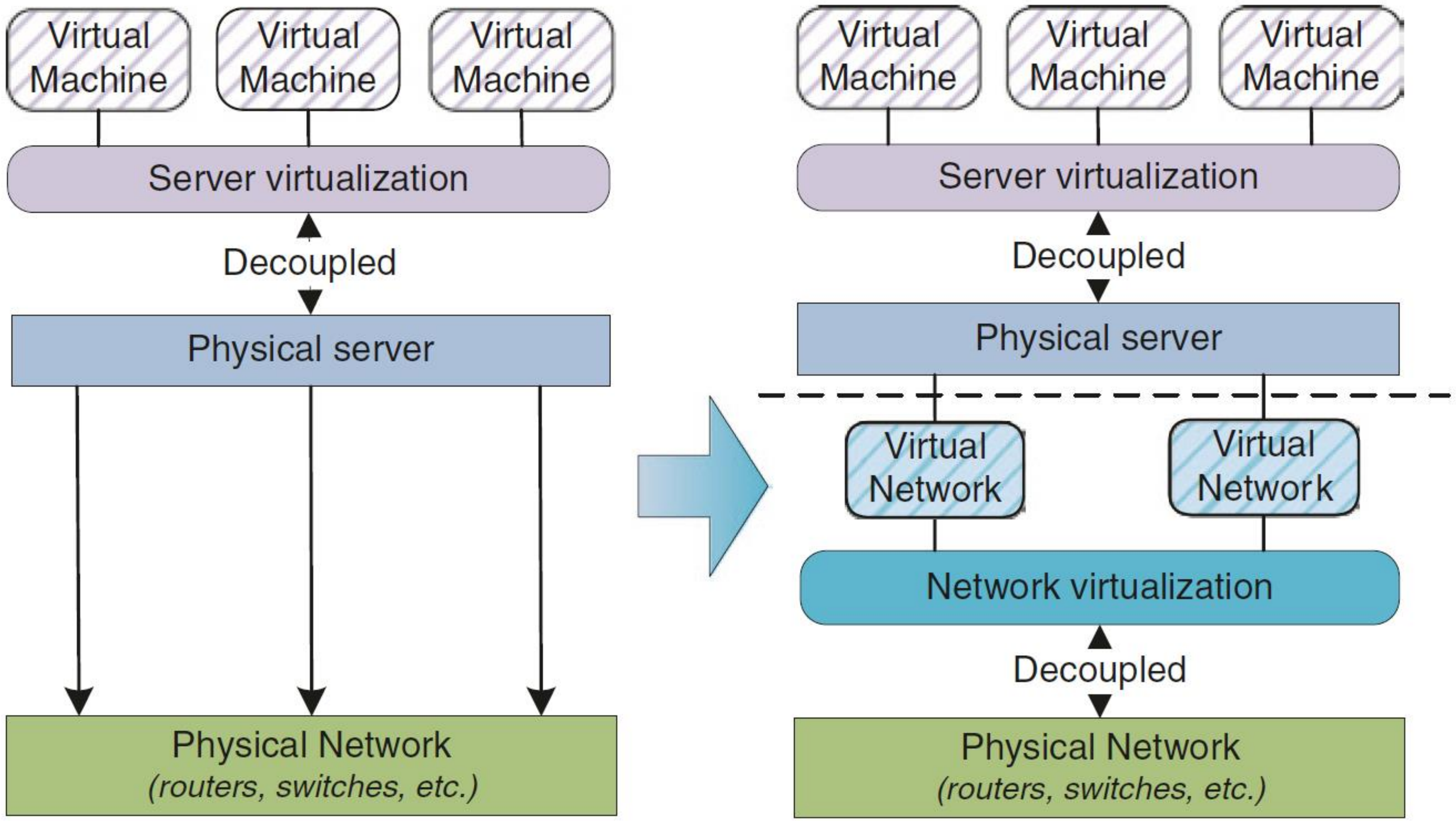
Centralized
Control Plane

Simple, Cheaper
Multi-vendor
Data Plane



# Datacenter Network

Scaling the virtualized datacenter



# Early SDN Deployments

NTT Communications:

- Deployed NEC infrastructure to deliver its Enterprise Cloud Service (as part of its virtualized data center infrastructure)
- Optimized ICT costs while managing global corporate ICT ops.

Google B4 Software Defined WAN (transport SDN foundation)

- Announced at ONS 2012; built custom switches with OF agent
- Filling up the G-scale backbone network pipes for efficiency

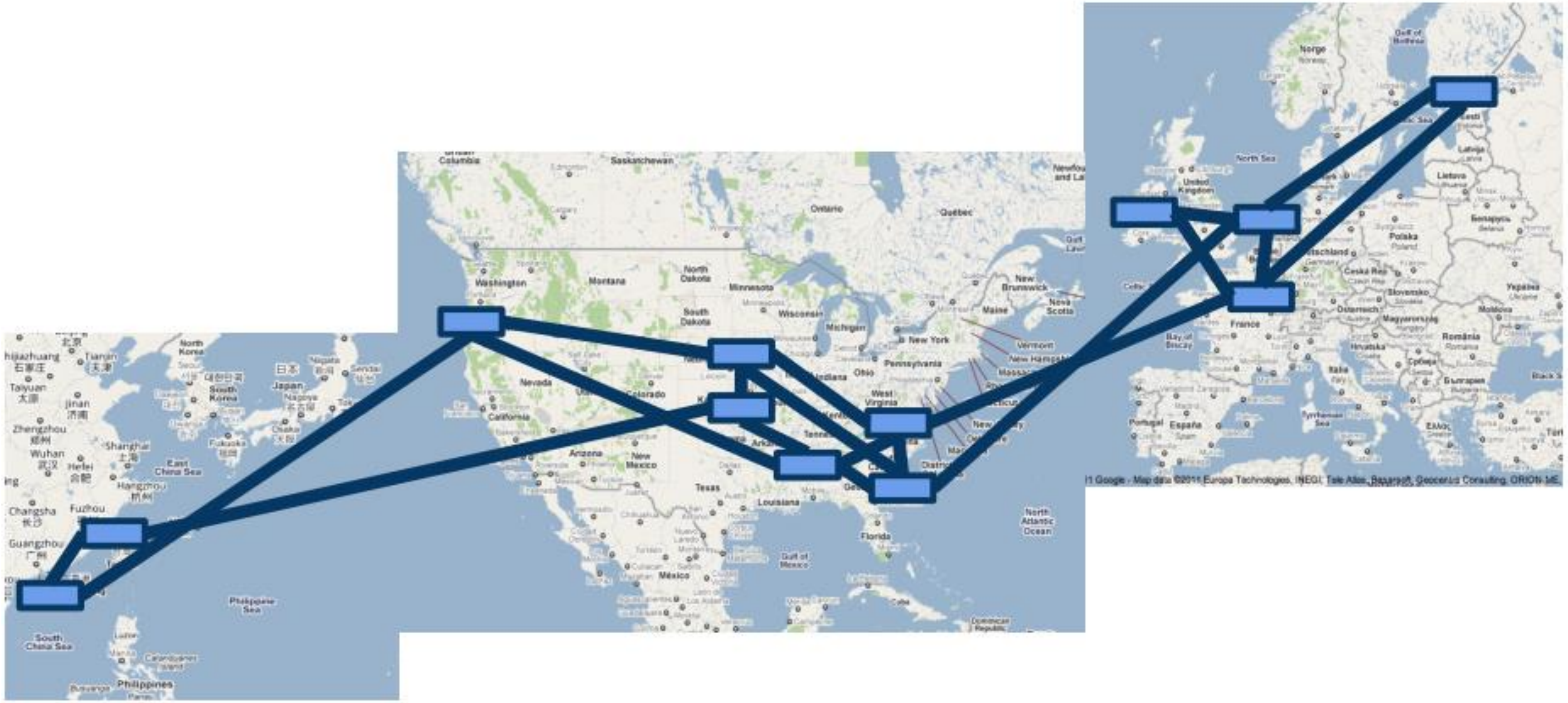
Deutsche Telekom TeraStream project:

- IPv6 network in Croatia for broadband services
- Tail-f NCS controller running Netconf, Yang; Cisco network equipment

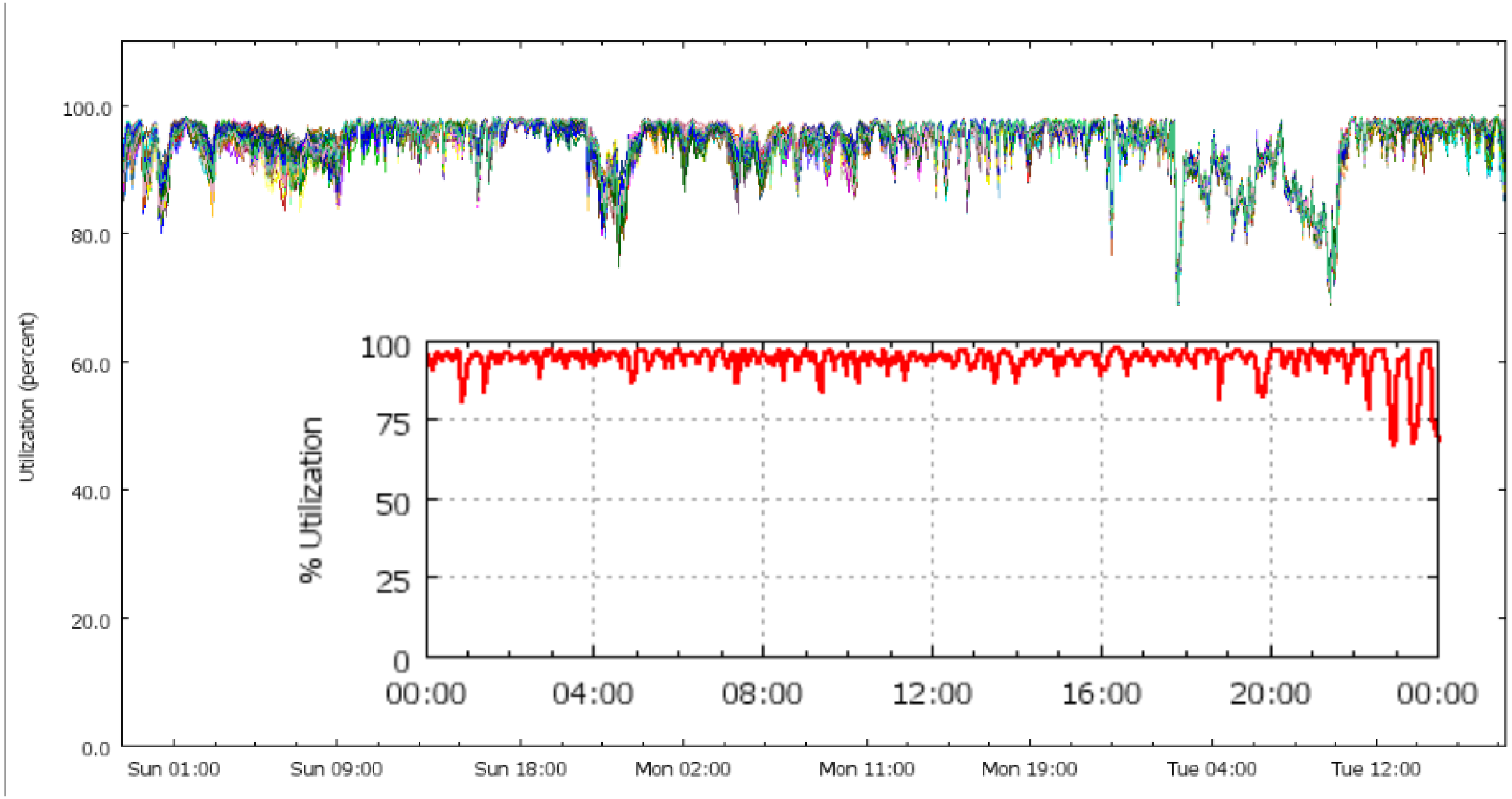
Colt Telecom Carrier Ethernet Service:

- Leverages SDN to offer a multi-vendor carrier Ethernet service using Cyan's:
- Blue Planet software to orchestrate, provision, and ontrol Accedian EtherNIDs
- Z-Series optical platforms to automate service provisioning

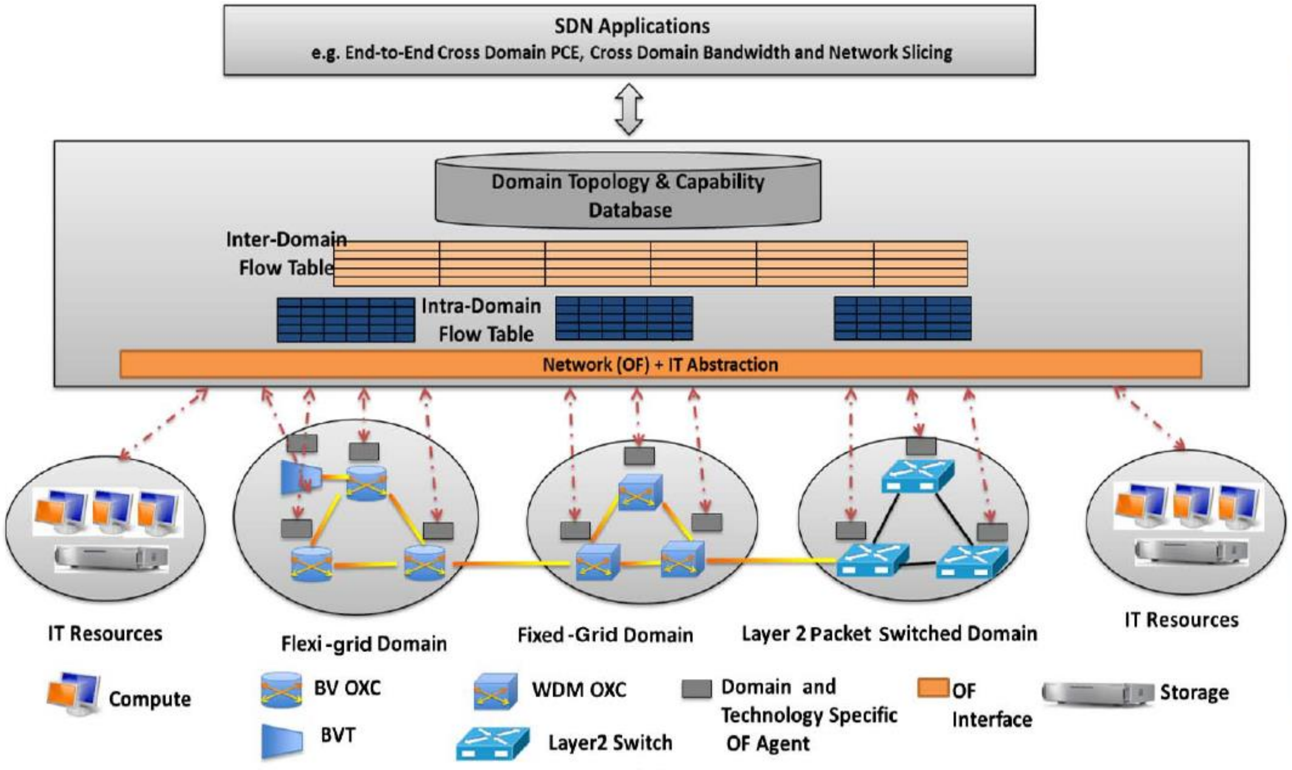
# Google WAN



# Link Utilization



# SDN Optical Network Control Plane




(a)

| Multidomain Scenario | Flow Mapping Rule |
|---|---|
| Flexi Grid ←→ Fixed Grid | Each center frequency and bandwidth must be compatible with WDM Grid |
| Flexi Grid ←→ Packet Switched | One or multiple packet flow identifiers must be mapped to each center frequency + bandwidth |
| Fixed Grid ←→ Packet Switched | One or multiple packet flow identifiers must be mapped to each individual channel |

(b)

Fig. 1. (a) Architecture of multilayer multitechnology control plane. (b) Flow mappings between technologies.

M. Channegowda, R. Nejabati, and D. Simeonidou "Software-Defined Optical Networks Technology and Infrastructure: Enabling Software-Defined Optical Network Operations", IEEE/OSA J. OPT. COMMUN. NETW., VOL. 5, NO. 10, 2013

# SDN Optical Network Control Plane

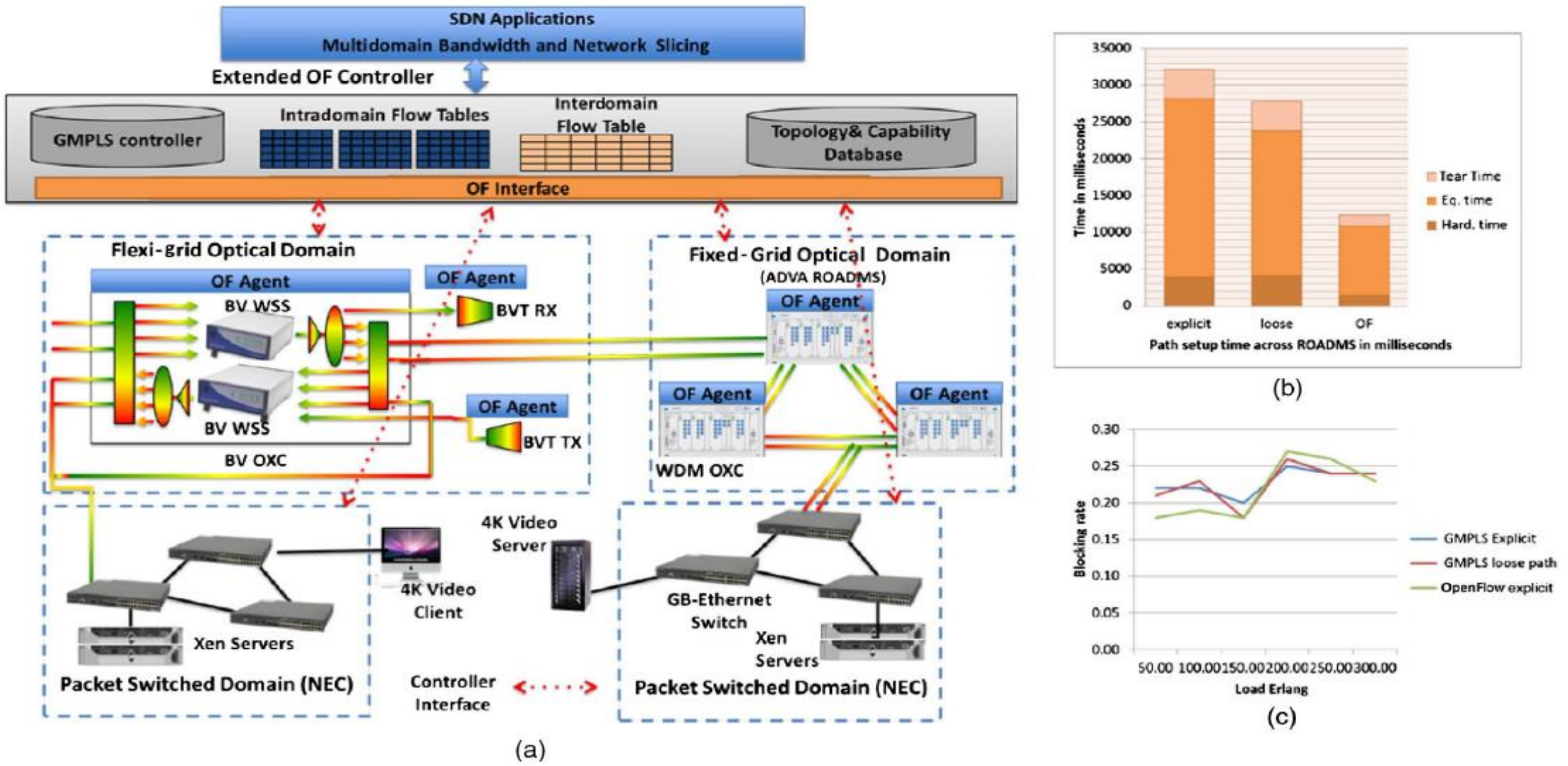



Fig. 4. (a) Demonstration setup: packet-fixed-flexible devices. (b) Path setup times for fixed WDM nodes. (c) Blocking probability versus load for GMPLS–OF and standalone OF approaches.

M. Channegowda, R. Nejabati, and D. Simeonidou "Software-Defined Optical Networks Technology and Infrastructure: Enabling Software-Defined Optical Network Operations", IEEE/OSA J. OPT. COMMUN. NETW., VOL. 5, NO. 10, 2013

# Open Networking Foundation

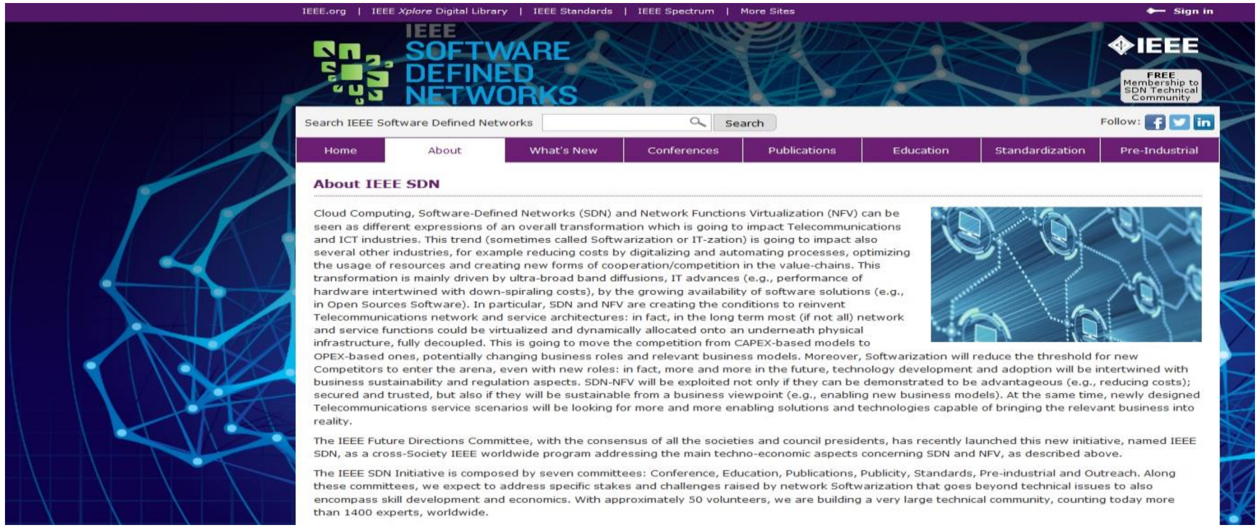
- Open Networking Foundation (ONF) is a user-driven organization dedicated to the promotion and adoption of Software-Defined Networking (SDN) through open standards development.

- https://www.opennetworking.org
  - Technical library, codes, video

# ONF Members

# IEEE SDN

- IEEE Software Defined Networks (Future Direction initiative)
- [http://sdn.ieee.org/about.html](http://sdn.ieee.org/about.html)
- Confernces, publications, standardization


IEEE.org | IEEE Xplore Digital Library | IEEE Standards | IEEE Spectrum | More Sites
Sign in
IEEE SOFTWARE DEFINED NETWORKS
IEEE
FREE Membership to SDN Technical Community
Search IEEE Software Defined Networks
Search
Follow:
Home | About | What's New | Conferences | Publications | Education | Standardization | Pre-Industrial

**About IEEE SDN**

Cloud Computing, Software-Defined Networks (SDN) and Network Functions Virtualization (NFV) can be seen as different expressions of an overall transformation which is going to impact Telecommunications and ICT industries. This trend (sometimes called Softwarization or IT-zation) is going to impact also several other industries, for example reducing costs by digitalizing and automating processes, optimizing the usage of resources and creating new forms of cooperation/competition in the value-chains. This transformation is mainly driven by ultra-broad band diffusions, IT advances (e.g., performance of hardware intertwined with down-spiraling costs), by the growing availability of software solutions (e.g., in Open Sources Software). In particular, SDN and NFV are creating the conditions to reinvent Telecommunications network and service architectures: in fact, in the long term most (if not all) network and service functions could be virtualized and dynamically allocated onto an underneath physical infrastructure, fully decoupled. This is going to move the competition from CAPEX-based models to OPEX-based ones, potentially changing business roles and relevant business models. Moreover, Softwarization will reduce the threshold for new Competitors to enter the arena, even with new roles: in fact, more and more in the future, technology development and adoption will be intertwined with business sustainability and regulation aspects. SDN-NFV will be exploited not only if they can be demonstrated to be advantageous (e.g., reducing costs); secured and trusted, but also if they will be sustainable from a business viewpoint (e.g., enabling new business models). At the same time, newly designed Telecommunications service scenarios will be looking for more and more enabling solutions and technologies capable of bringing the relevant business into reality.

The IEEE Future Directions Committee, with the consensus of all the societies and council presidents, has recently launched this new initiative, named IEEE SDN, as a cross-Society IEEE worldwide program addressing the main techno-economic aspects concerning SDN and NFV, as described above.

The IEEE SDN Initiative is composed by seven committees: Conference, Education, Publications, Publicity, Standards, Pre-industrial and Outreach. Along these committees, we expect to address specific stakes and challenges raised by network Softwarization that goes beyond technical issues to also encompass skill development and economics. With approximately 50 volunteers, we are building a very large technical community, counting today more than 1400 experts, worldwide.

# NFV

# Network Functions Virtualisation (NFV)

## A joint operator initiative and call-for-action to industry

**A joint operator push to the IT and Telecom industry, to provide a new network production environment, based on modern virtualization technology, to lower cost, raise efficiency and to increase agility.**

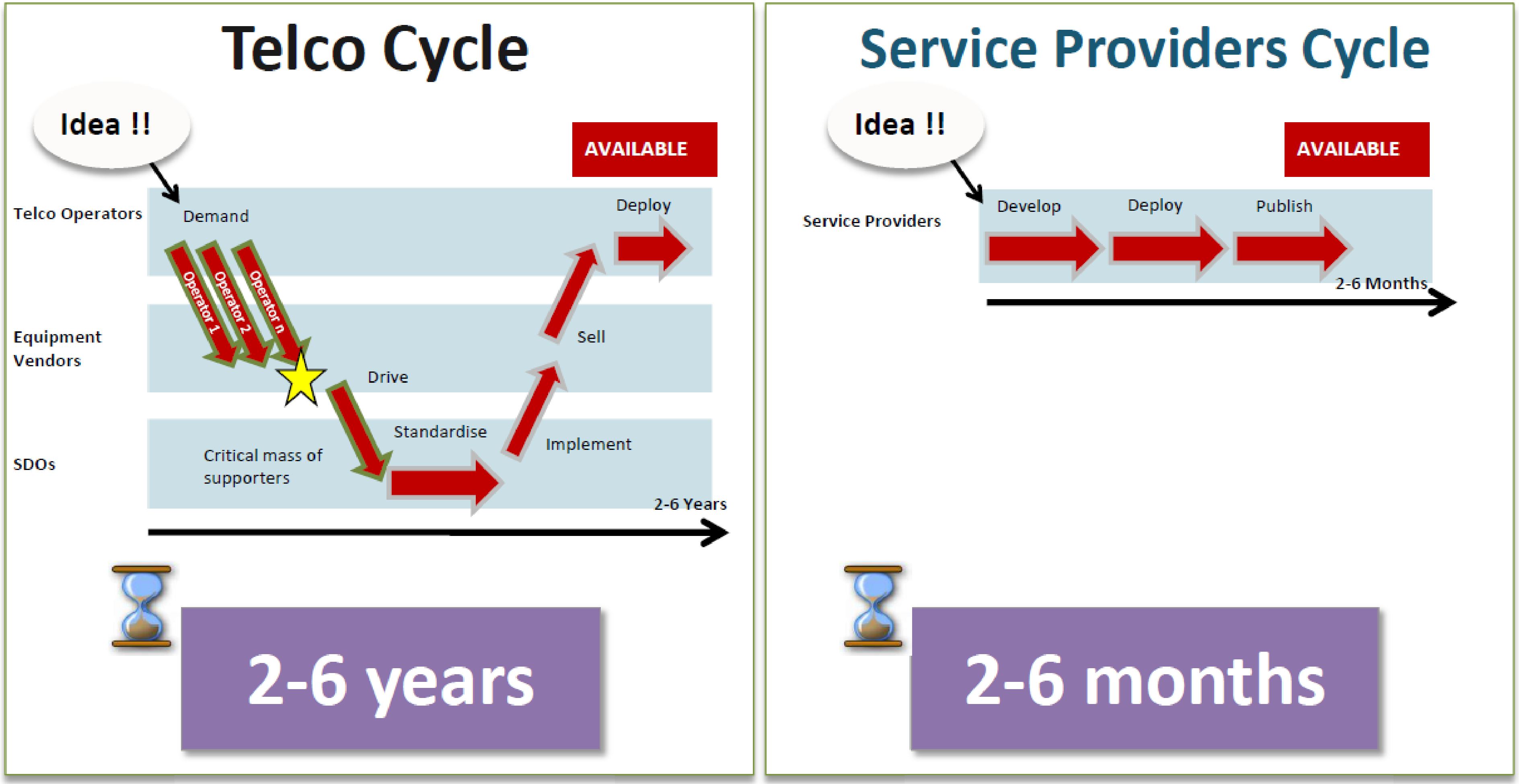
*We believe Network Functions Virtualisation is applicable to any data plane packet processing and control plane function in fixed and mobile network infrastructures (WP)*

at&t BT CenturyLink 中国移动通信 CHINA MOBILE colt smarter / faster / further T KDDI NTT orange TELECOM ITALIA Telefonica verizon

# Motivation
## Problem Statement

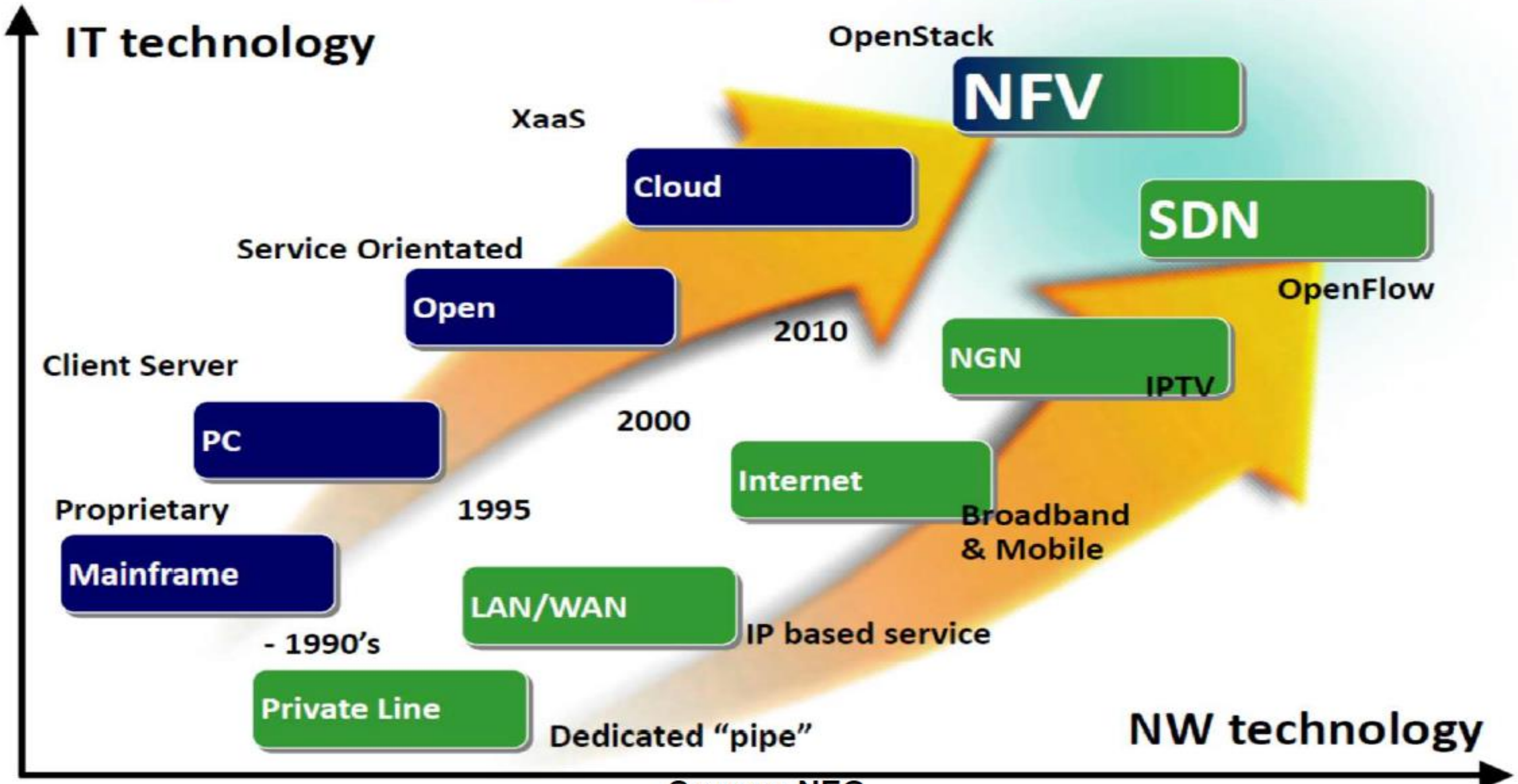
- Complex carrier networks
  - with a large variety of proprietary nodes and hardware appliances.
- Launching new services is difficult and takes too long
  - **Space and power to accommodate**
  - requires just another variety of box, which needs to be integrated.
- Operation is expensive
  - **Rapidly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


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

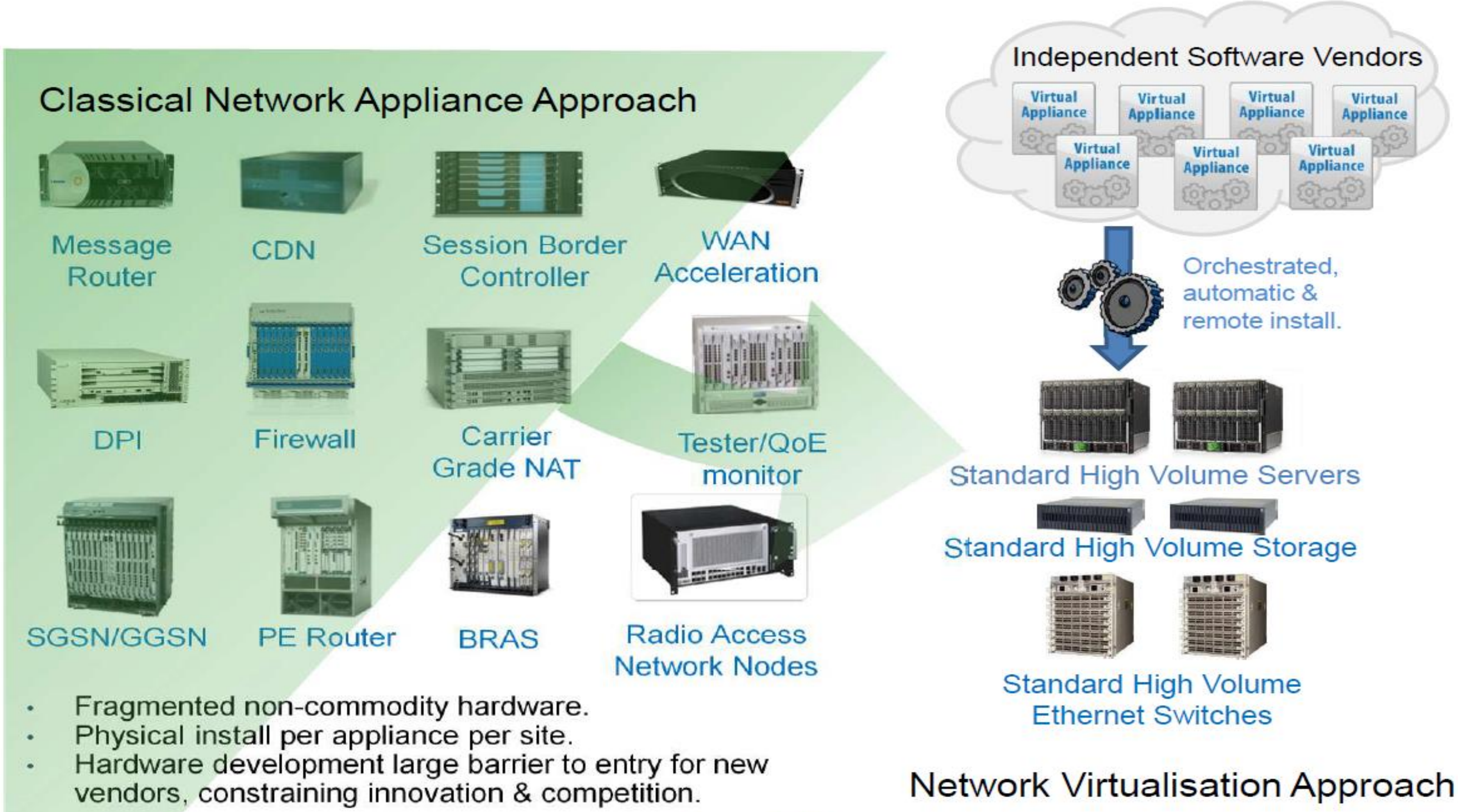



Source: Adapted from D. Lopez Telefonica I+D, NFV

# IT & Networking Growing Together



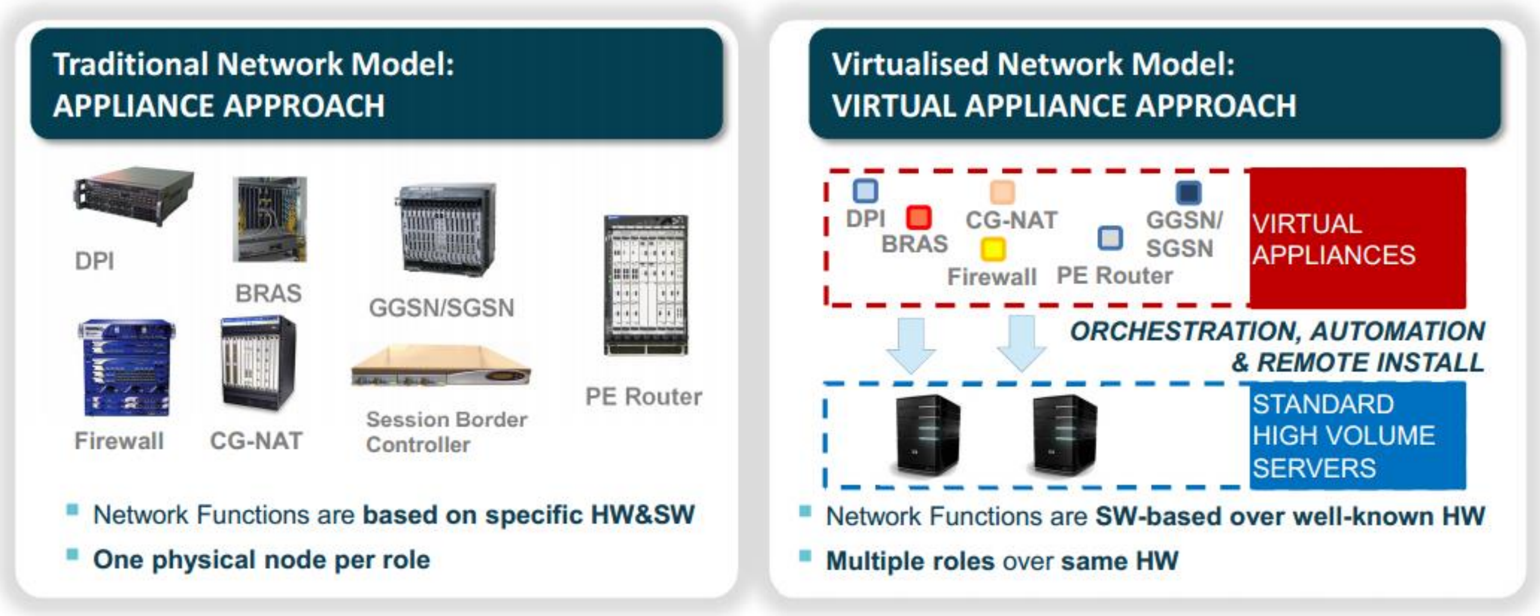

Source: NEC


Classical Network Appliance Approach
Message Router
CDN
Session Border Controller
WAN Acceleration
DPI
Firewall
Carrier Grade NAT
Tester/QoE monitor
SGSN/GGSN
PE Router
BRAS
Radio Access Network Nodes
• Fragmented non-commodity hardware.
• Physical install per appliance per site.
• Hardware development large barrier to entry for new vendors, constraining innovation & competition.
Independent Software Vendors
Virtual Appliance
Orchestrated, automatic & remote install.
Standard High Volume Servers
Standard High Volume Storage
Standard High Volume Ethernet Switches
Network Virtualisation Approach

Source: NFV

# Network Function Virtualization (NFV)

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

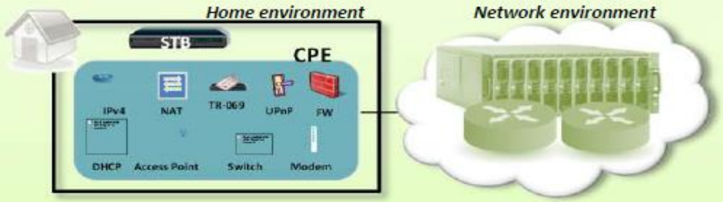



Source: Adapted from D. Lopez Telefonica I+D, NFV

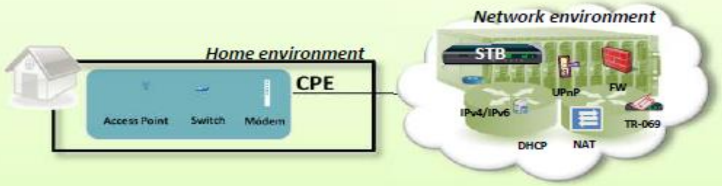
# Some Drivers

Virtual CPE

## Complex home environment




## Home simplification



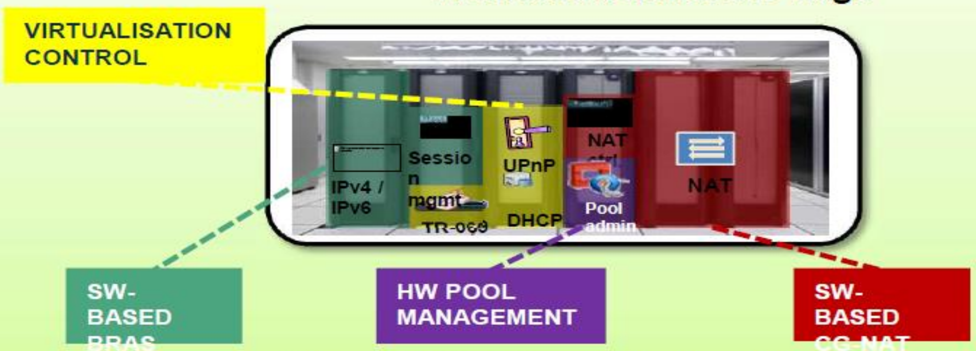

- Simplification or even supression (STB)
- No need for home router replacement as it is updated by configuration
- Fast deployment for new services
- Inexpensive IPv6 migration maintaining legacy home routers

Virtual IP Edge

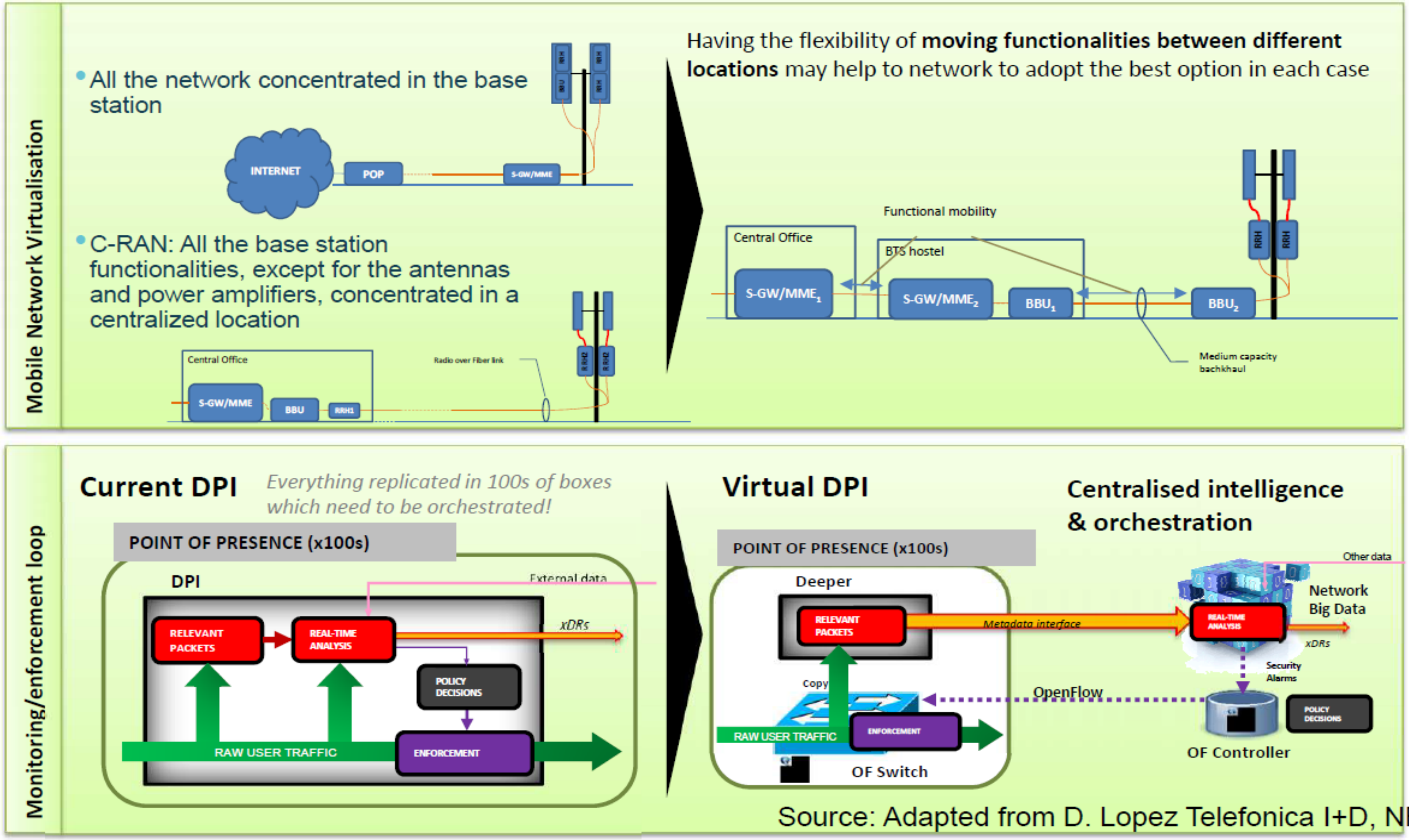
## Multiple IP Edges

- An IP Edge for each service (voice, video content, Internet)
- Scattered and not well integrated control functions (e.g. DPI, BRAS, PCRF)

## A unified software IP Edge

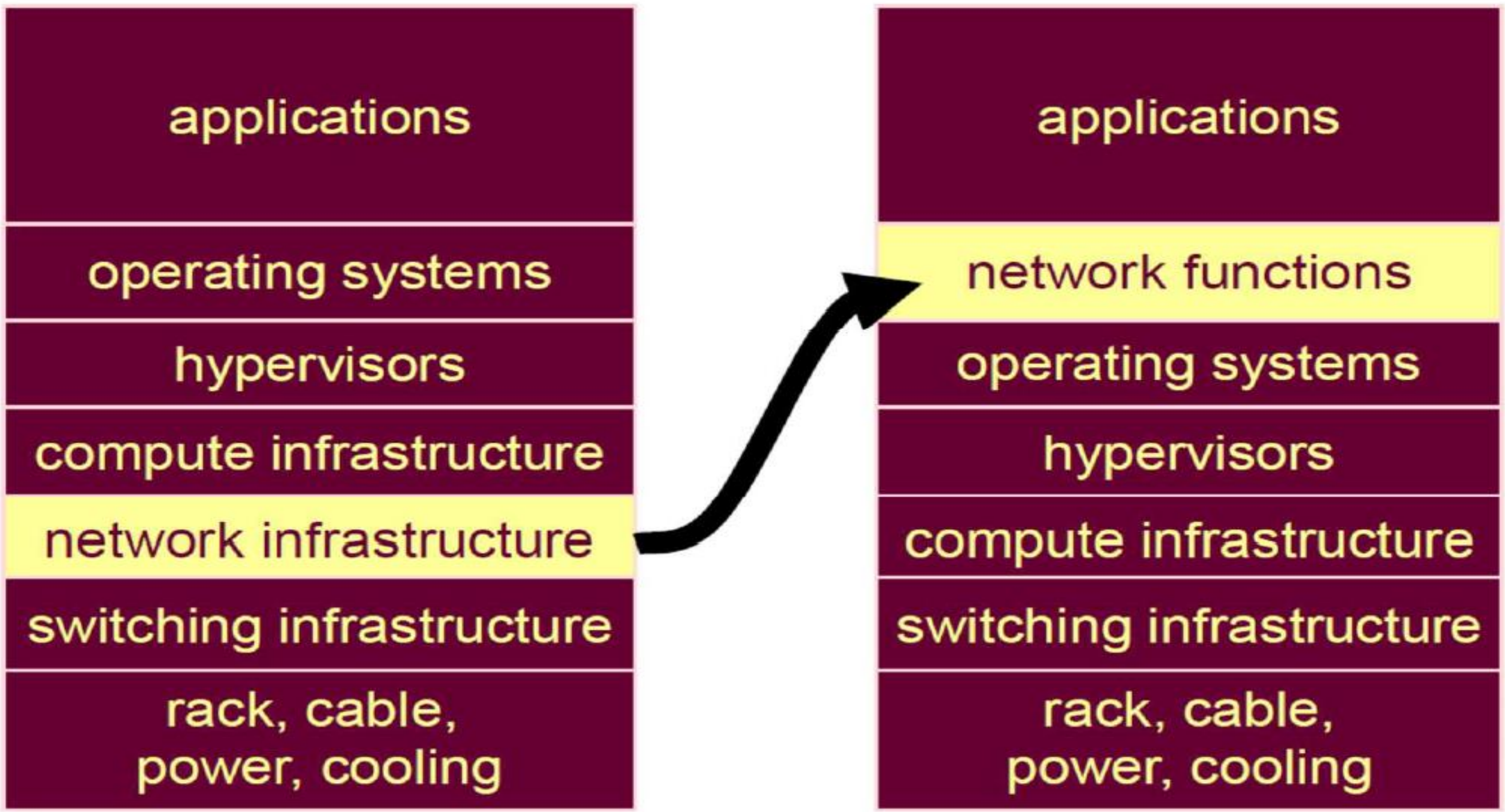



Source: Adapted from D. Lopez Telefonica I+D, NFV




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

# Rethinking relayering

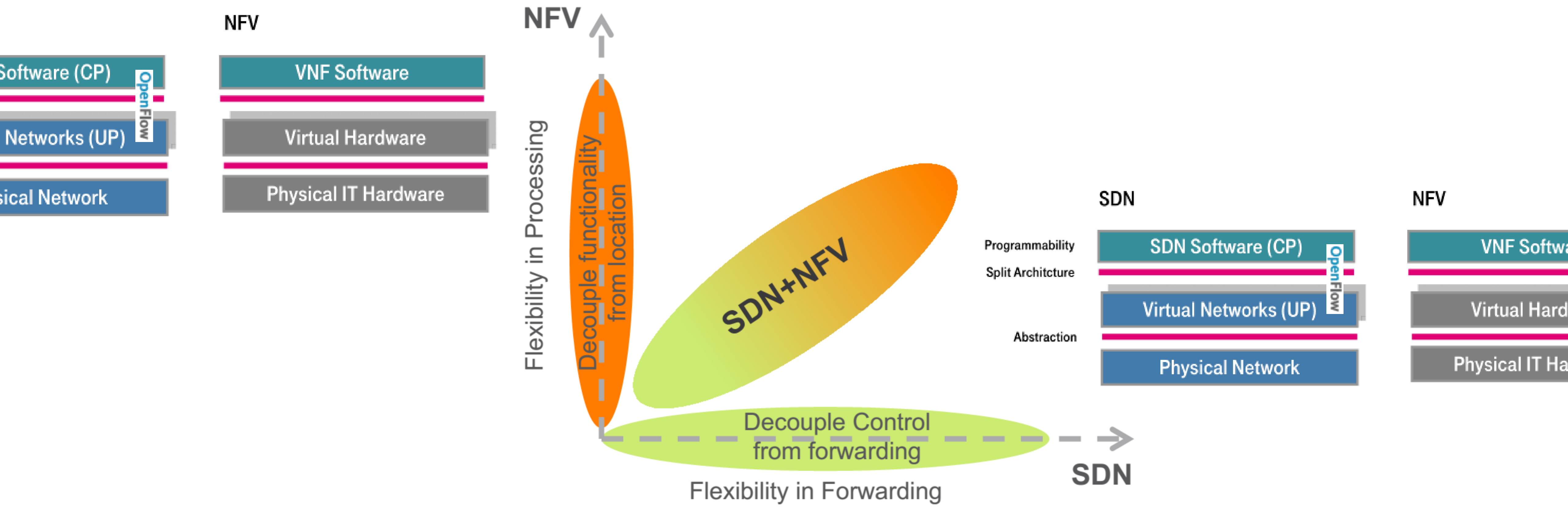


# NFV :: Network Functions Virtualization

- 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
- Enables new ways to implement resilience, service assurance, test and diagnostics and security surveillance
- Provides opportunities for pure software players
- Facilitates innovation towards new network functions and services that are only practical in a pure software network environment
- Applicable to any data plane packet processing and control plane functions, in fixed or mobile networks
- NFV will only scale if management and configuration of functions can be automated
- NFV aims to ultimately transform the way network operators architect and operate their networks, but change can be incremental

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

# Network Softwarization = SDN & NFV
# Network Programmability /Flexibility




Sources: Ahmad Rostami, Ericsson Research (Kista): http://www.itc26.org/fileadmin/ITC26_files/ITC26-Tutorial-Rostami.pdf and Uwe Michel, T-Systems

# NFV vs. SDN

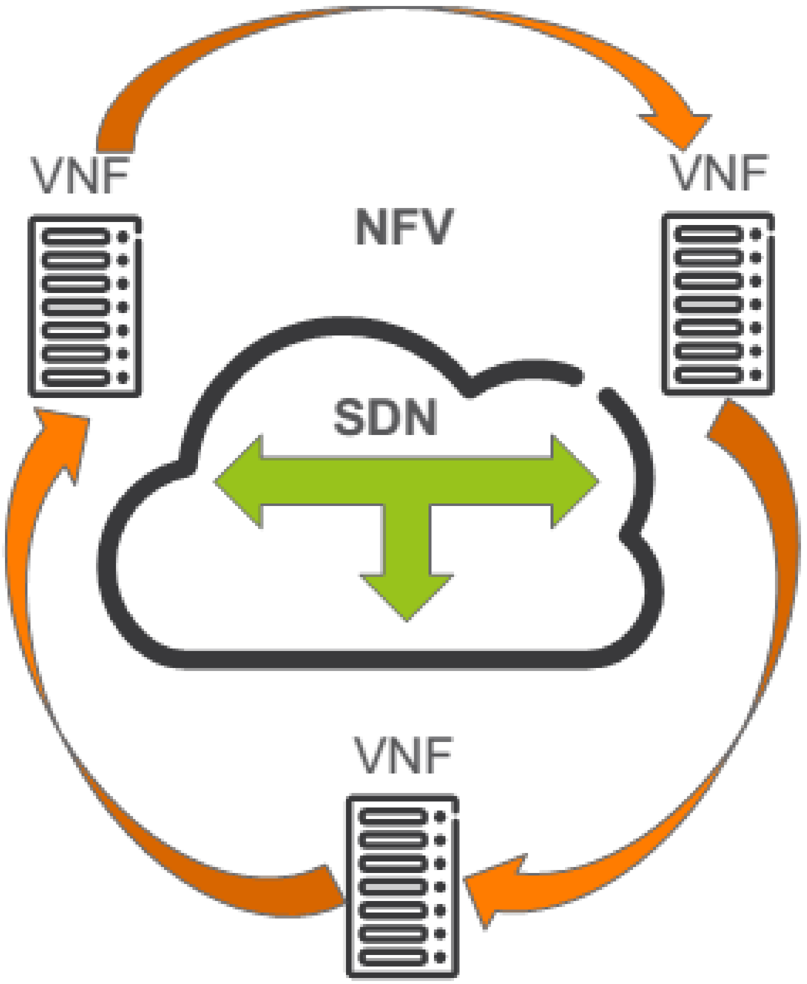
**SDN ›››** flexible forwarding & steering of traffic in a physical or virtual network environment

[Network Re-Architecture]

**NFV ›››** flexible placement of virtualized network functions across the network & cloud

[Appliance Re-Architecture] (initially)

**››› SDN & NFV** are complementary tools for achieving full **network programmability**



# Why 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. Programmability:** 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

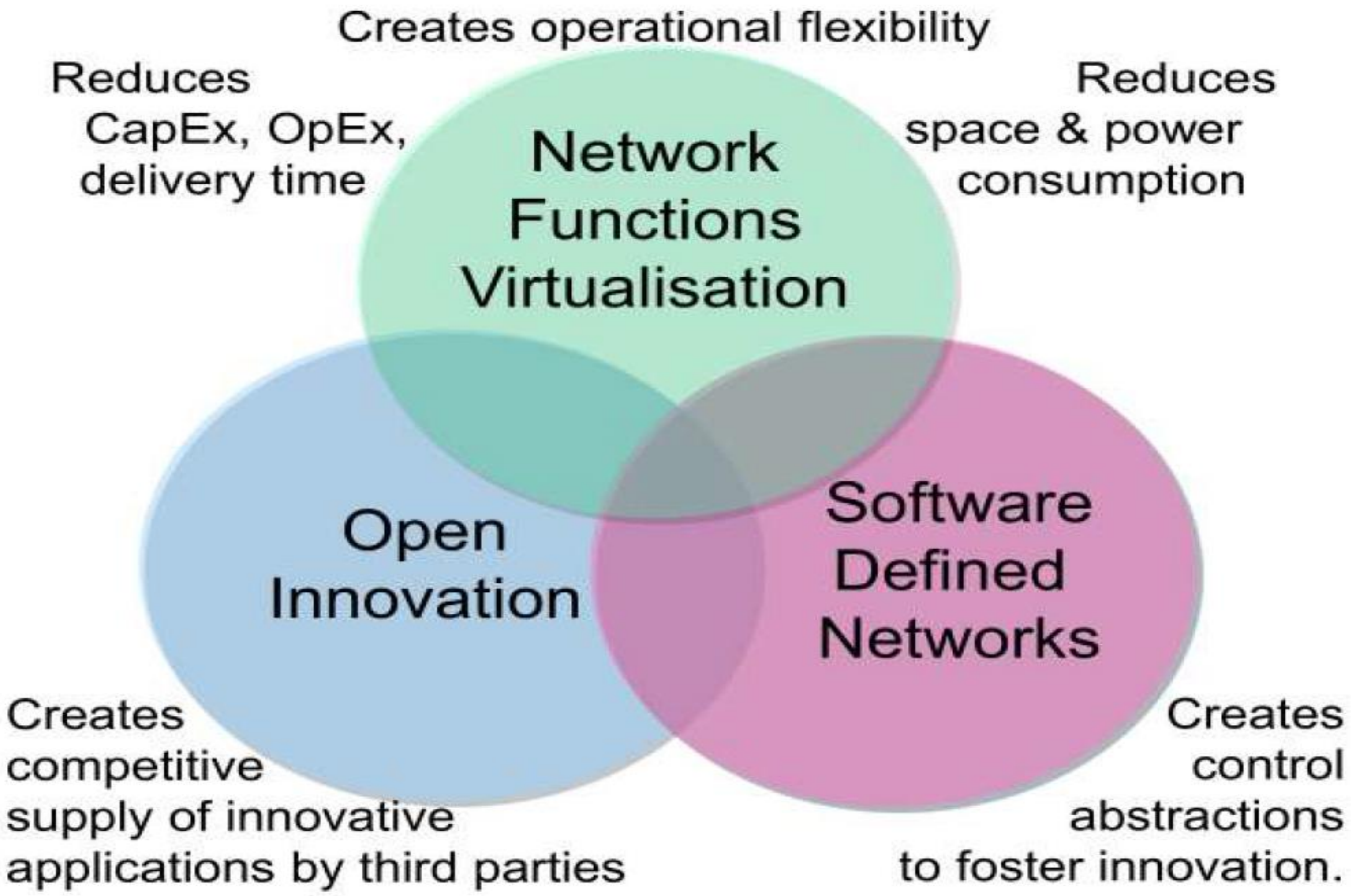
**7. Performance:** Optimize network device utilization

**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

Source: Adapted from Raj Jain




Source: Bob Briscoe, BT

# 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.

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.

Source: Adapted from Raj Jain

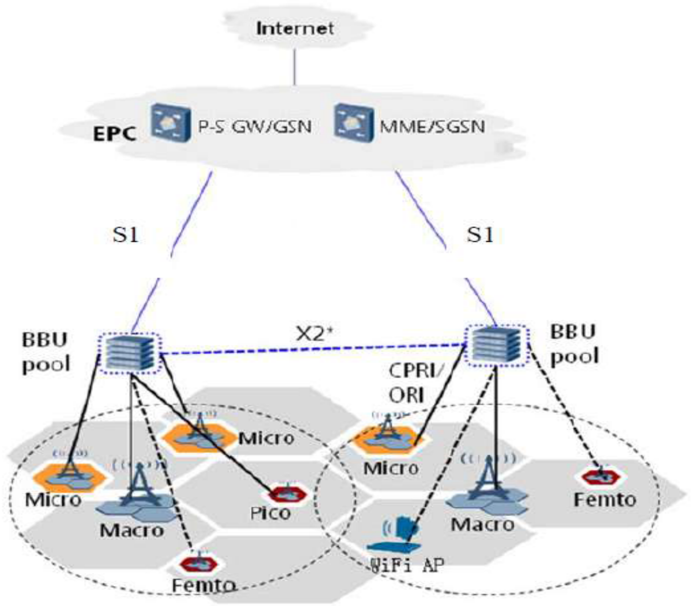
# Virtualization of Mobile Base Station

- **Mobile network traffic is significantly increasing** by the demand generated by application of mobile devices, while the **ARPU (revenue) is difficult to increase**

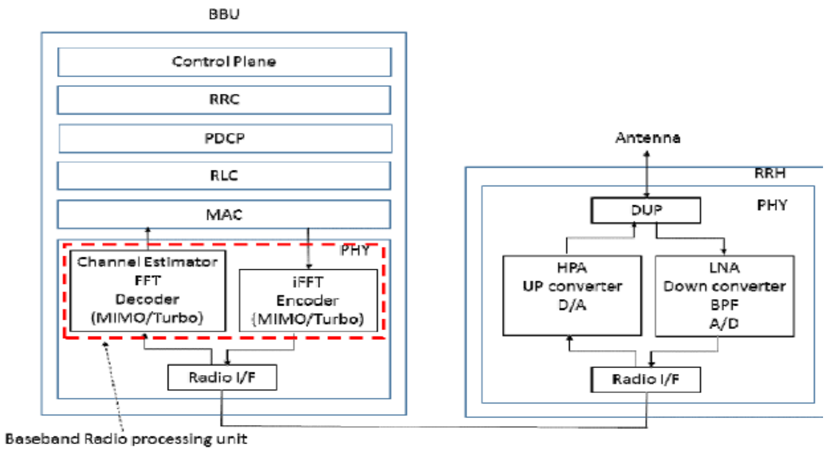
- **LTE is also considered as radio access part of EPS (Evolved Packet System)** which is required to fulfill the requirements of **high spectral efficiency, high peak data rates, short round trip time and frequency flexibility** in radio access network (RAN)

- **Virtualization of mobile base station leverages** IT virtualization technology to realize at least a part of RAN nodes onto **standard IT servers, storages and switches**

# Virtualization of Mobile Base Station




LTE RAN architecture evolution by centralized BBU pool (Telecom Baseband Unit)




Functional blocks in C-RAN

Source: ETSI NFV UC

# NFV Growing ecosystem