

# Software-Defined Networking

¿Sabías que tu red se puede programar?

Alaitz Mendiola

Christian Pinedo

Euskal Encounter #24

22 de Julio del 2016, Barakaldo

# Agenda

- **Introducción a las redes definidas por software**
- **OpenFlow**
- **Aplicaciones**
- **Herramientas necesarias para el taller**
  - Mininet
- **Ejercicios prácticos**
  - Programación de un learning switch
  - Programación de un firewall

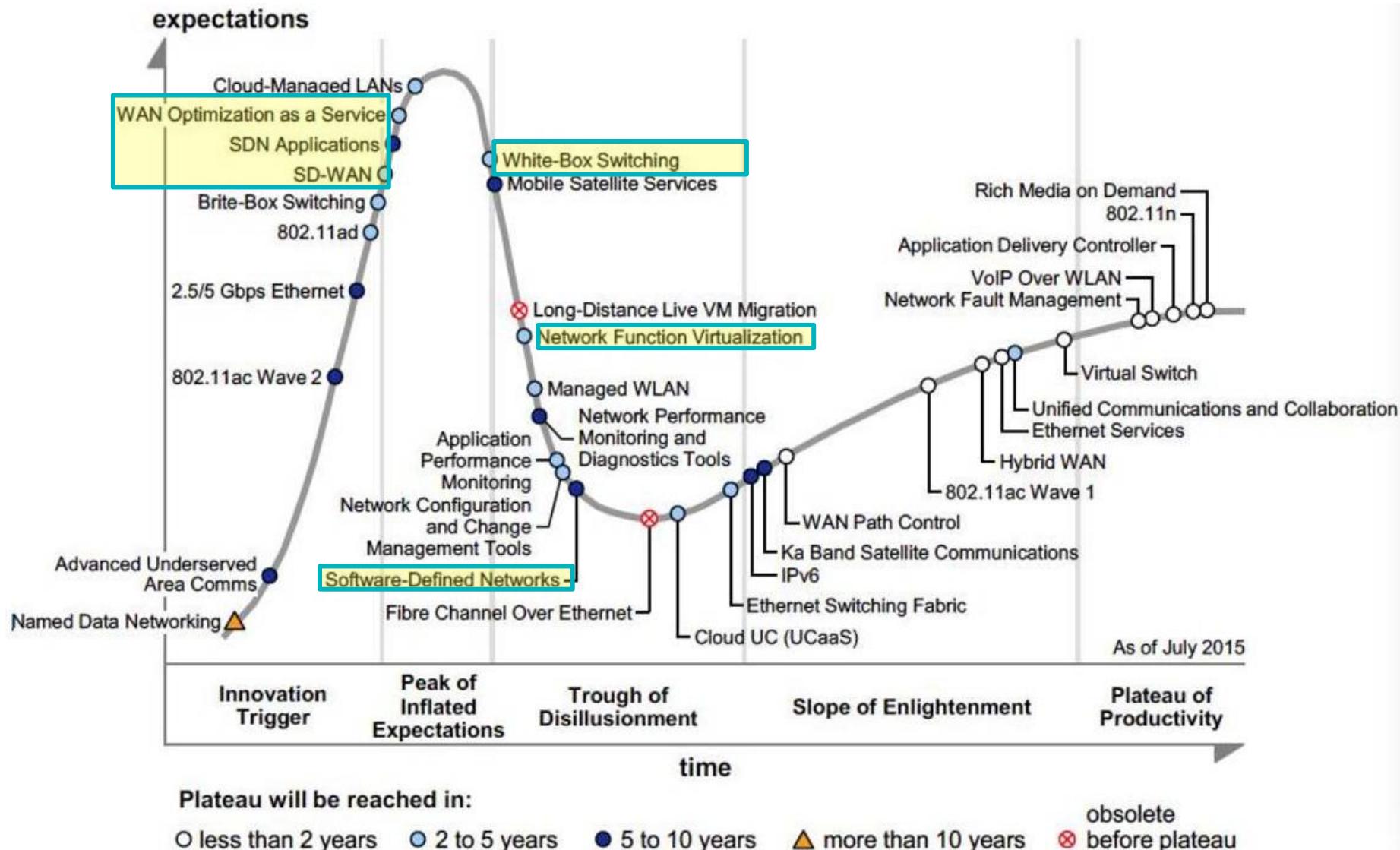
# Introducción a las redes definidas por software

# Introducción

- **Arquitectura de red**
  - Optimización y simplificación de la operación de red.
  - Interacción entre servicios y dispositivos de red.



# Introducción



# Principios fundamentales de las SDNs

*Separación del plano de control y el plano de datos*

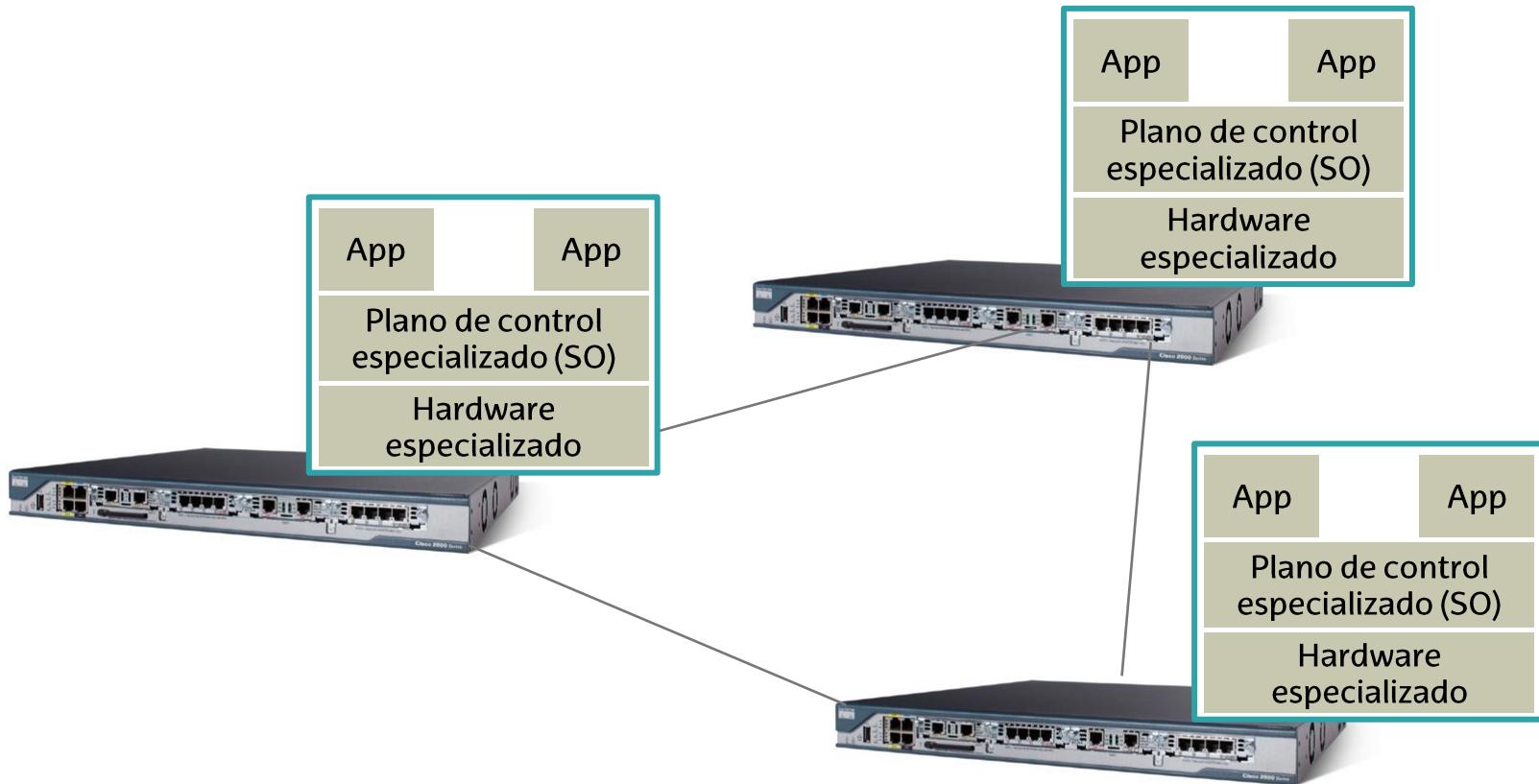
*Plano de control lógicamente centralizado*

*Programación de equipos de red mediante interfaces abiertas*

# Separación del plano de control y el plano de datos

- **Plano de Gestión**
  - Gestión de servicios, monitorización, rendimiento, políticas...
- **Plano de Control**
  - Gestión del estado de la red, establecimiento de servicios, implementación de políticas, resiliencia...
- **Plano de Datos**
  - Procesamiento de paquetes y flujos

# Separación del plano de control y el plano de datos

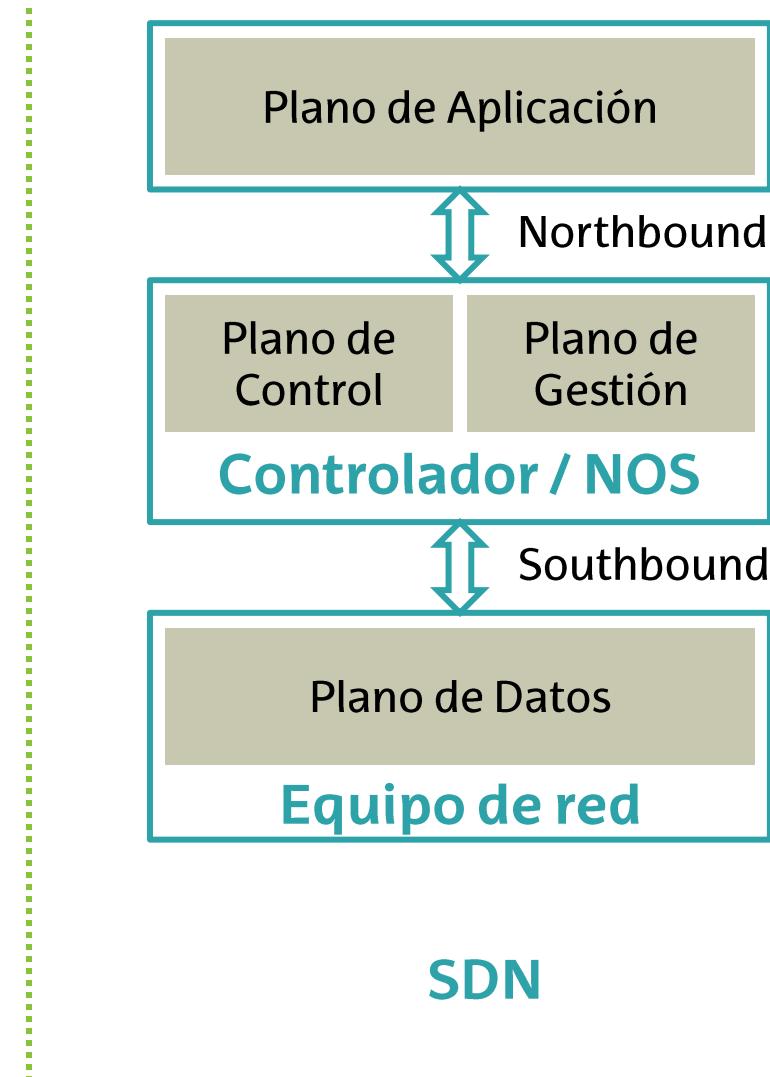


**Networking tradicional**

# Separación del plano de control y el plano de datos

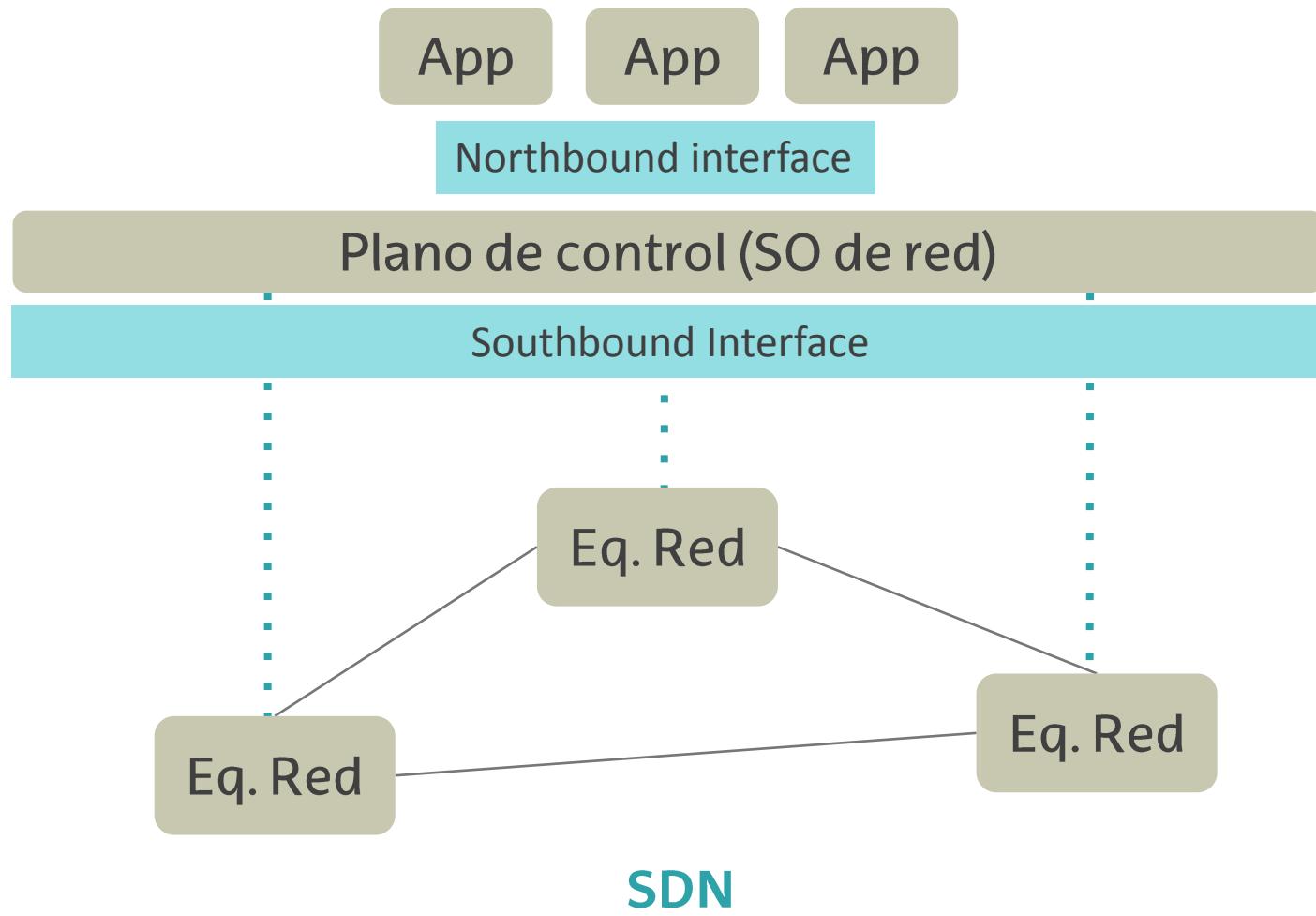


Networking tradicional



SDN

# Separación del plano de control y el plano de datos

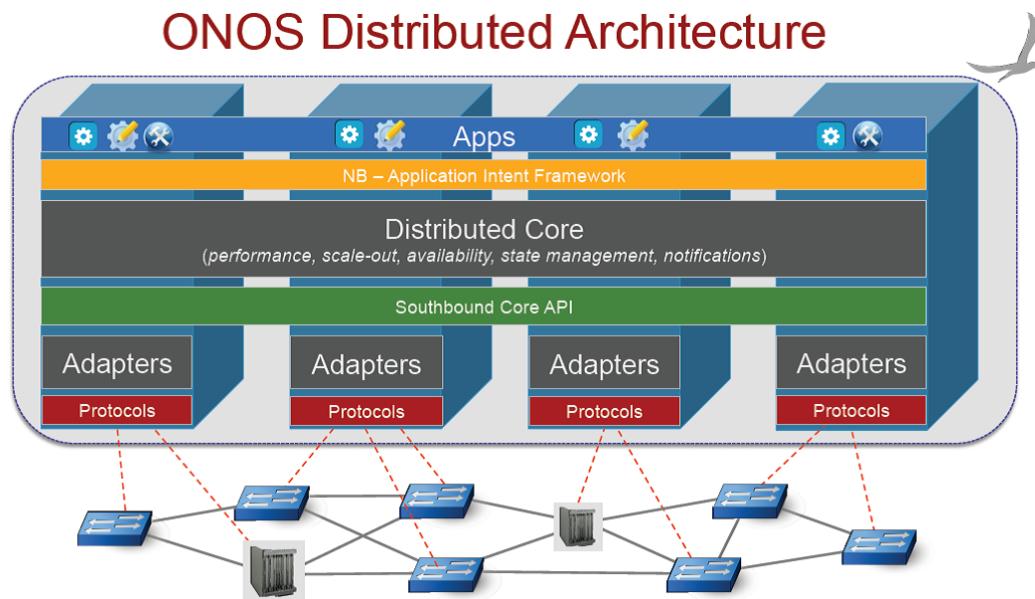


# Plano de control lógicamente centralizado

- Inicialmente se hablaba de un “Plano de control centralizado”
  - Punto de fallo único.
  - Saturación del canal de control.
  - Rendimiento dependiente de la distancia.
  - Poca escalabilidad.
- Actualmente la ONF define el plano de control como **LÓGICAMENTE CENTRALIZADO**.

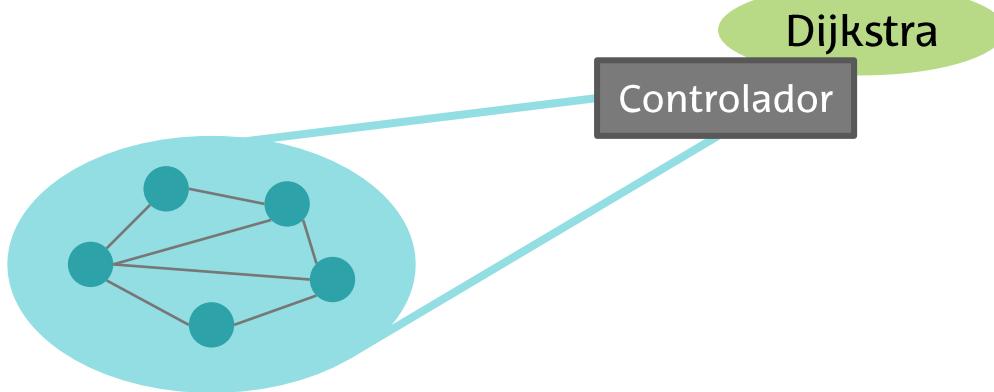
# Plano de control lógicamente centralizado

- **Compuesto por múltiples instancias del controlador.**
  - Cada instancia se encarga de un subconjunto de los equipos de la red.
  - Sincronización.
  - Mayor escalabilidad.



# Plano de control lógicamente centralizado

- **El controlador/NOS dispone de información actualizada del estado de la red.**
  - Topología.
    - Permite aplicar algoritmos de rutado directamente.
    - Facilita la reacción ante fallos en la red.
    - Se reducen los tiempos de convergencia.



# Interfaces abiertas

- **Northbound interface**
  - Facilita la utilización de distintos controladores.  
¿¿REST??
- **East-west interface**
  - Facilita la comunicación entre controladores.  
SDNi
- **Southbound interface**
  - Facilita la utilización de equipos de diferentes fabricantes..

OPENFLOW  
OVSDB

FORCES  
NETCONF

I2RS

BGP-LS/PCEP  
OF-CONFIG

Fuente: [www.himawan.nu](http://www.himawan.nu)

# Ventajas de las redes definidas por software

- **Evolución independiente del software de control.**
- **Mayor flexibilidad.**
- **Conocimiento del estado global de la red.**
- **Despliegue de servicios en menor tiempo.**
- **Mayor programabilidad.**
- **Menor coste operativo.**

# Retos actuales

- **Escalabilidad.**
- **Fiabilidad.**
  - Redundancia de controladores.
- **Consistencia:**
  - Entre múltiples instancias del controlador.

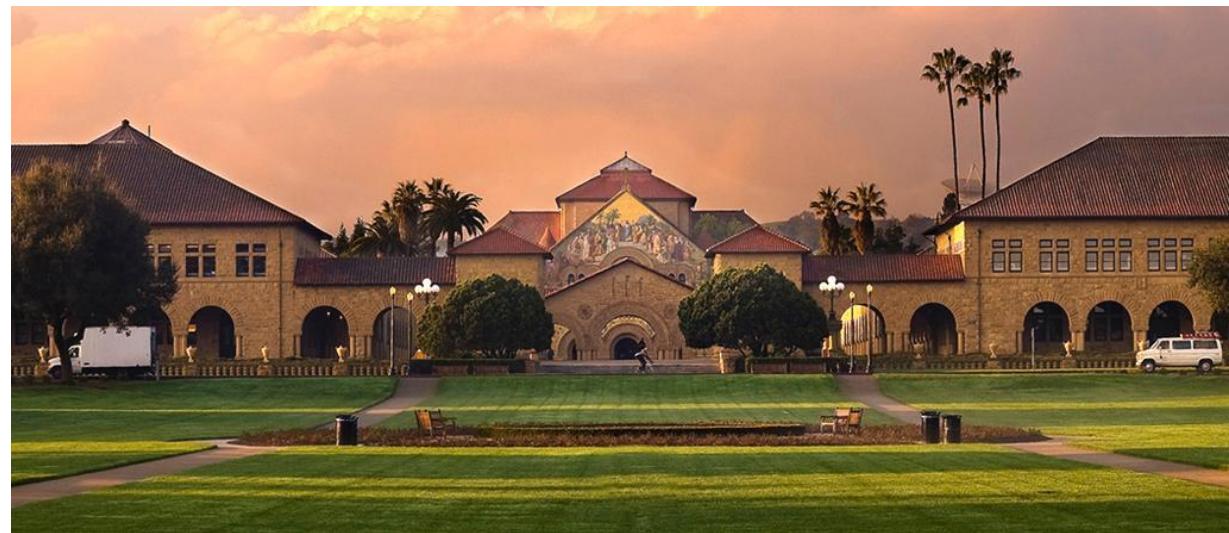
# OpenFlow

# Introducción

- ¿Qué es OpenFlow?
  - Arquitectura SDN
  - Desarrollada en la Universidad de Standford.
    - **Clean Slate program**
  - Estandarizado por la Open Networking Foundation



OPEN NETWORKING  
FOUNDATION

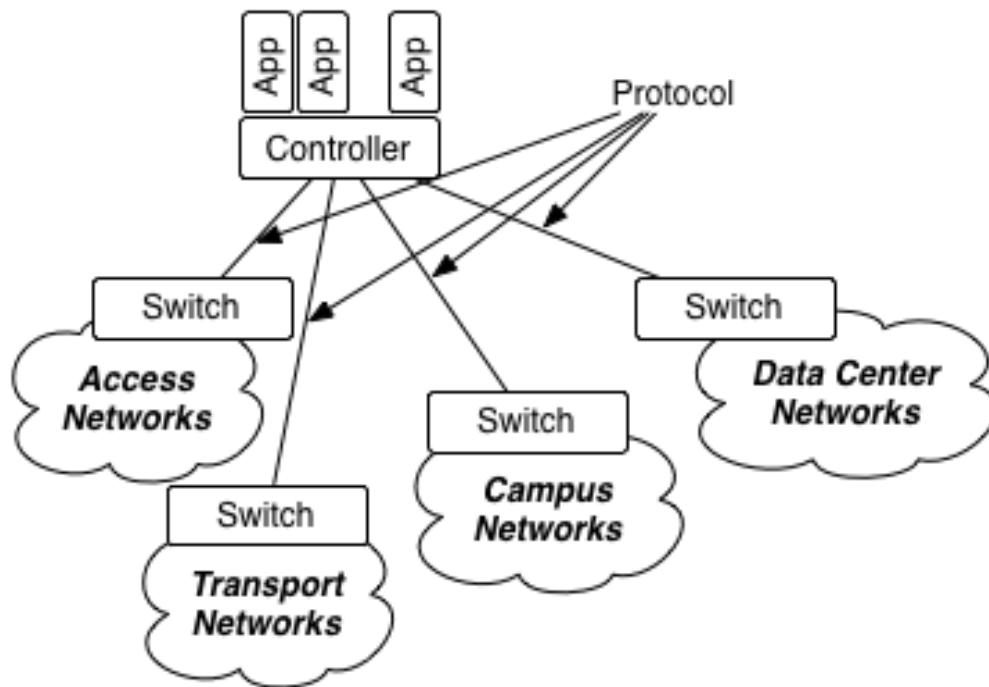


# Introducción

Estándar *de facto* para las redes  
definidas por software

- Agente OpenFlow.
- Redes desplegadas → redes programables.
- Innovación rápida.

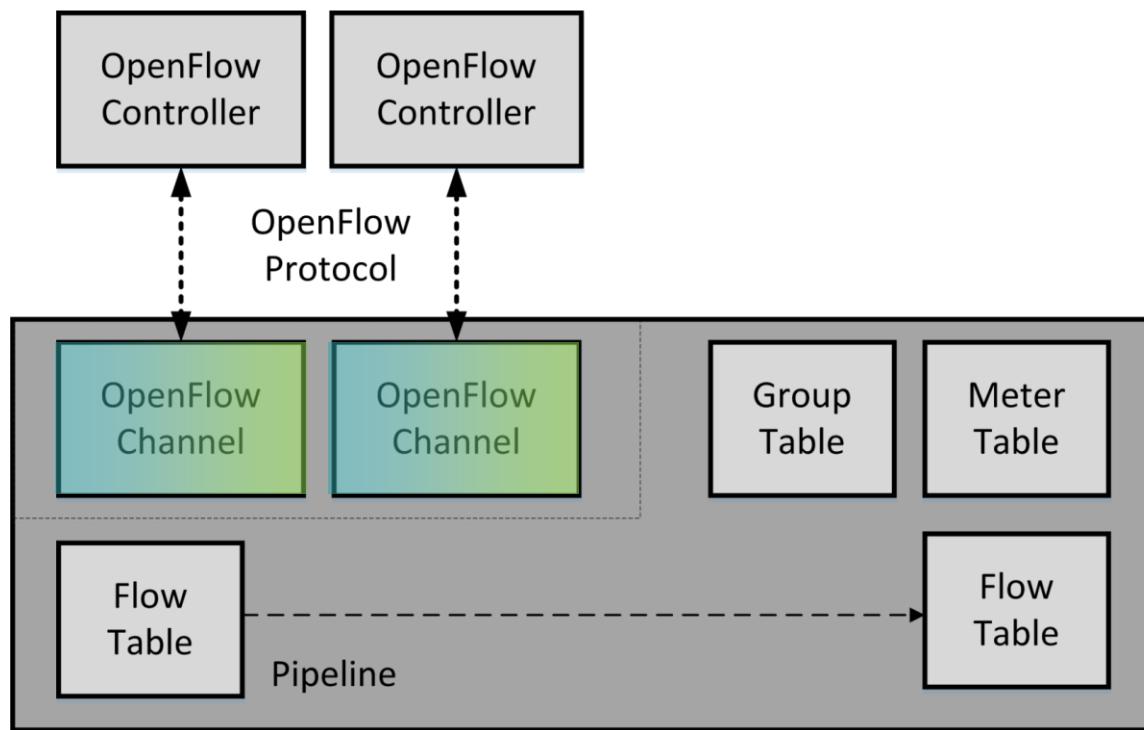
# Arquitectura de un switch OpenFlow



- **Controlador**
  - Cerebro de la red.
  - Plano de control (+ Plano de aplicaciones)
  - Decide cómo se han de procesar los paquetes en el plano de datos.
- **Protocolo OpenFlow**
  - Traslada al switch la lógica del controlador.
- **Switch**
  - Procesamiento de los paquetes.
  - Plano de datos

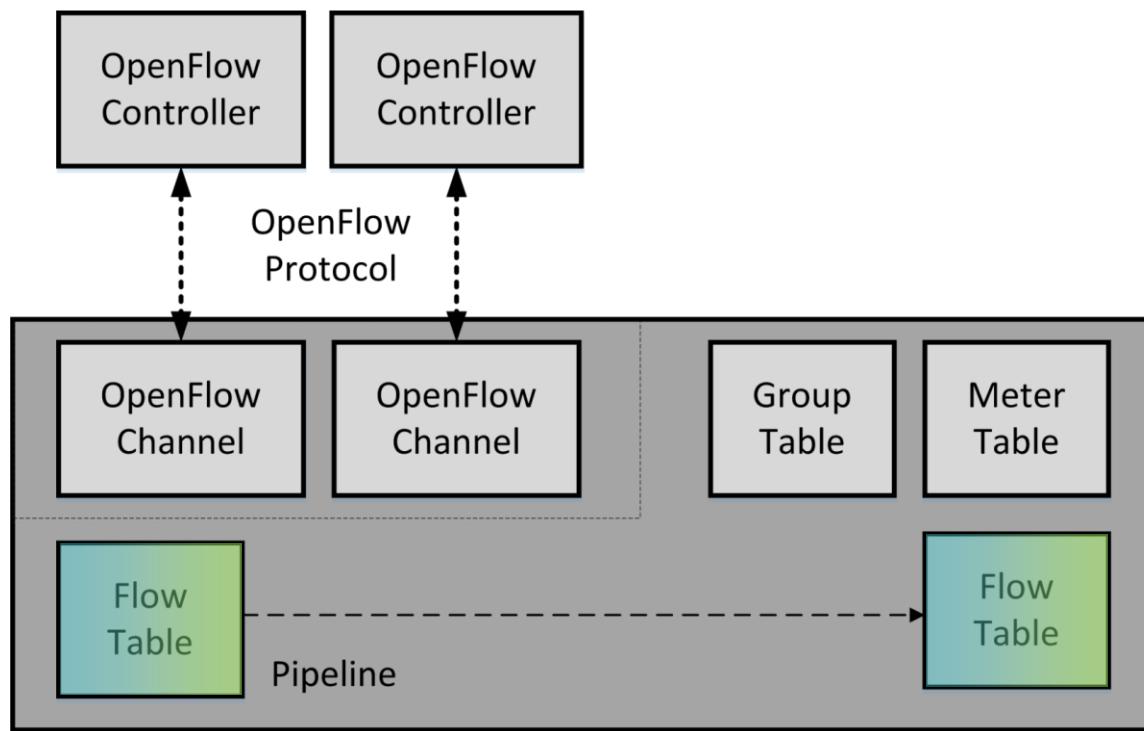
Fuente: <http://flowgrammable.org/sdn/openflow/>

# Arquitectura de un switch OpenFlow



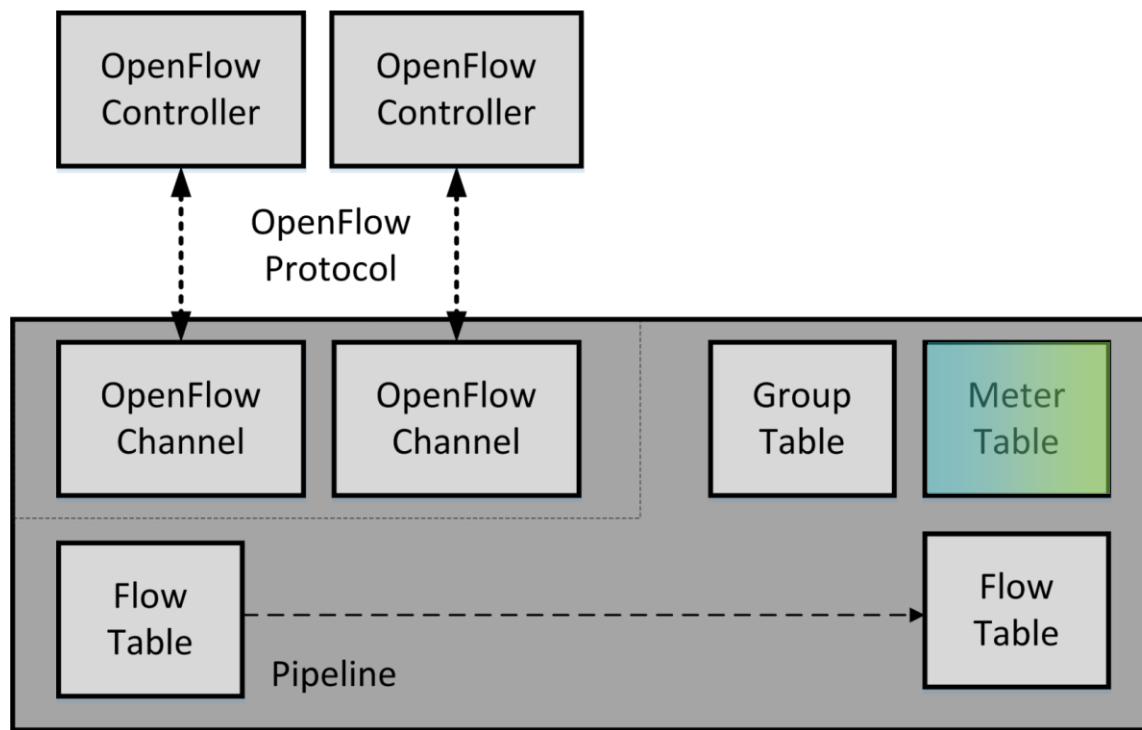
- Permite la comunicación con el controlador externo.

# Arquitectura de un switch OpenFlow



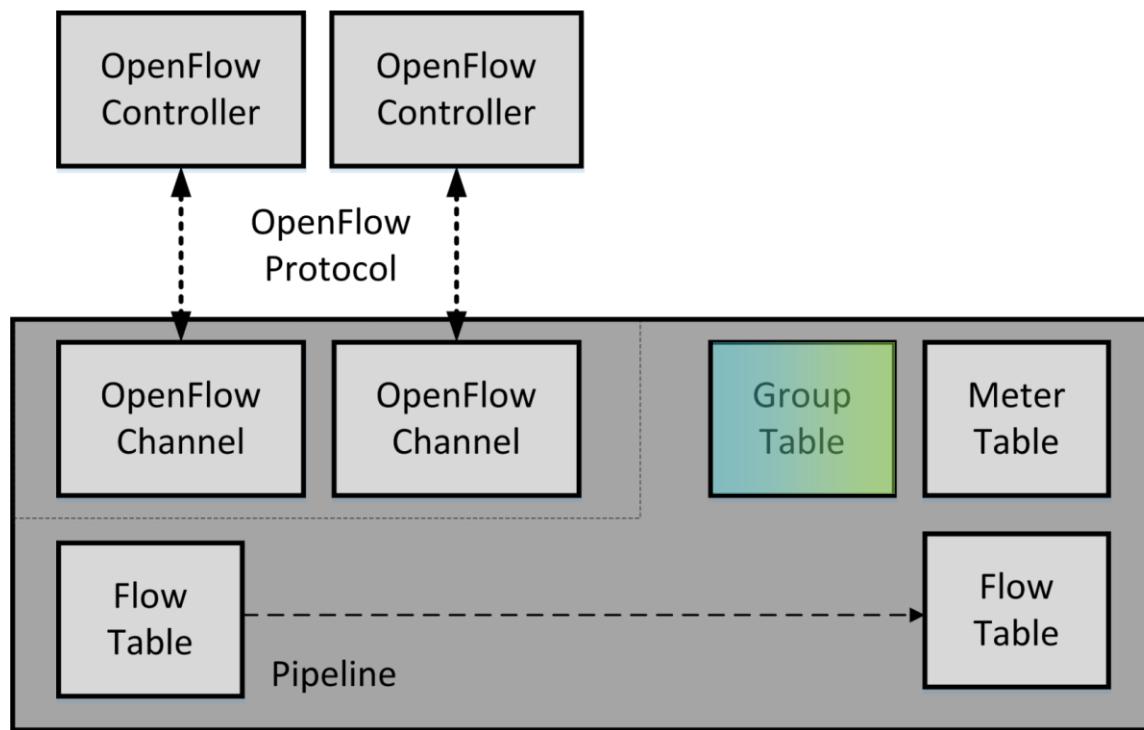
- Contienen las Flow Entries que definen los flujos y el procesamiento que les corresponde.
- Se ejecutan en secuencia.
- A la disposición concreta de flow tables se le conoce como Pipeline.
- Los paquetes entran por los puertos y se **matchean** contra las flow tables,

# Arquitectura de un switch OpenFlow



- Realiza funciones de traffic shaping.
  - Limita el ancho de banda de los flujos.

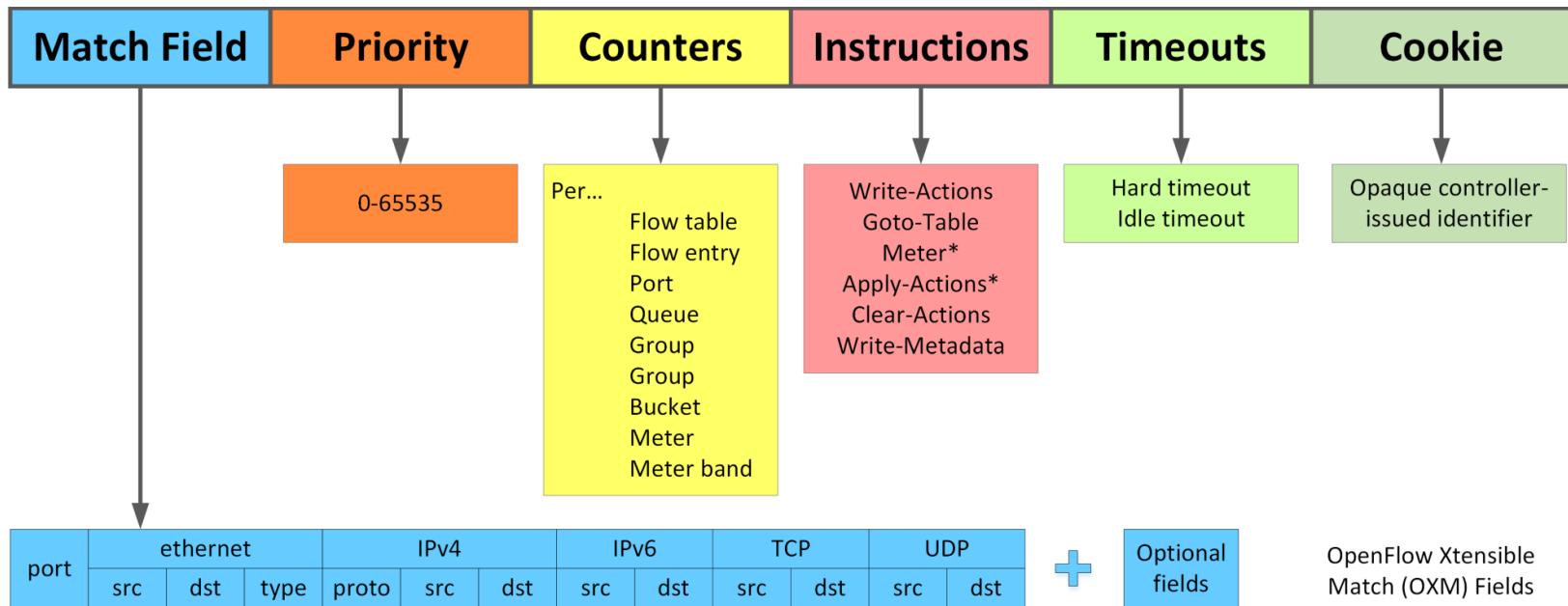
# Arquitectura de un switch OpenFlow



- Se procesa la última.
- Permite aplicar comportamientos de enrutamiento alternativos.
  - Load balancing, flooding, fast-failover.

# Funcionamiento

- Estructura de una *flow entry*.



# Funcionamiento

- **Acciones**
  - Reenviar el paquete:
    - **ALL:** por todos los interfaces, excepto por el que se ha recibido.
    - **CONTROLADOR:** encapsular el paquete y enviar al controlador.
    - IN PORT:** por el puerto por el que se ha recibido.
  - Descartar: no se especifica ninguna .
  - Modificar: Reescritura de campos de la cabecera.
  - Encolar: enviar el paquete a una cola.

# Mensajes OpenFlow 1.3

- **Packet in**
  - Switch → Controlador.
  - Se genera cuando el paquete no pertenece a ningún flujo.
    - Cabeceras del paquete o el paquete entero.
  - Posibilita el modo **reactivo**.
    - Toma de decisiones en el controlador en función del tráfico existente.
- **Packet out**
  - Controlador → Switch
  - El contenido del paquete se genera en el controlador
    - Se envía al switch para su procesamiento.
- **FlowMod**
  - Controlador → Switch
  - Modifica la tabla de flujos.
  - Añadir/Eliminar/Modificar una flow entry.

*Lista completa disponible en <http://flowgrammable.org/sdn/openflow/message-layer/>*

# Comparativa switch legacy vs switch OpenFlow

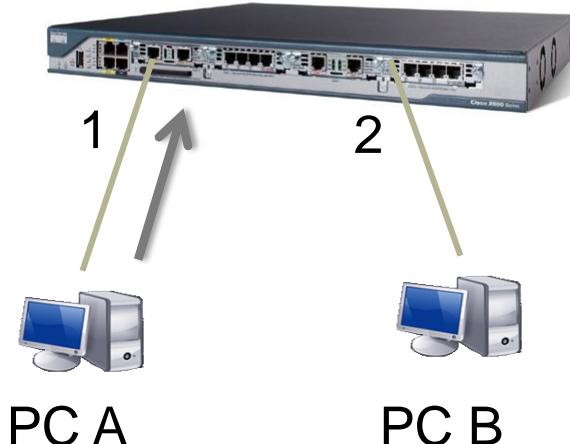
## Switch legacy

MAC Address Table

VLAN	Direcc. MAC	PUERTO	Timer
1	A	1	35
1	B	2	89

Un switch tradicional hace uso de la “MAC Address Table” para **reenviar** las tramas Ethernet

1. PCA envía una trama Ethernet para PCB

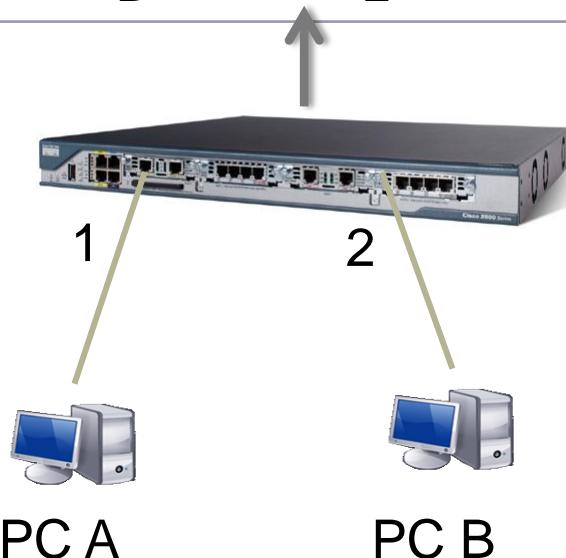


# Comparativa switch legacy vs switch OpenFlow

## Switch legacy

MAC Address Table

VLAN	Direcc. MAC	PUERTO	Timer
1	A	1	35
1	B	2	89



Un switch tradicional hace uso de la “MAC Address Table” para **reenviar** las tramas Ethernet

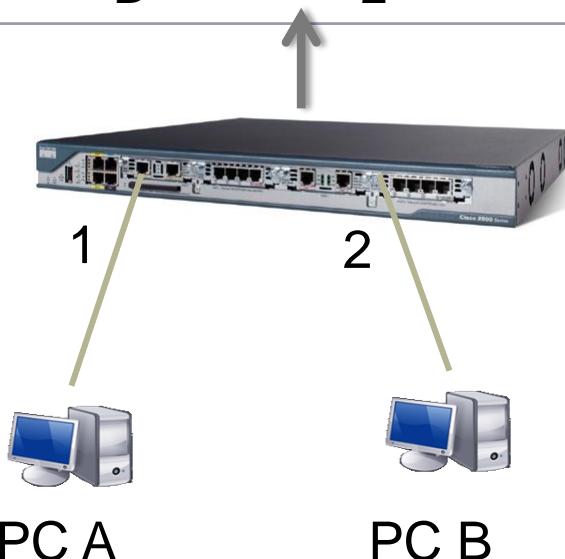
1. PCA envía una trama Ethernet para PCB
2. El switch consulta su “MAC Address Table”

# Comparativa switch legacy vs switch OpenFlow

## Switch legacy

MAC Address Table

VLAN	Direcc. MAC	PUERTO	Timer
1	A	1	0
1	B	2	89

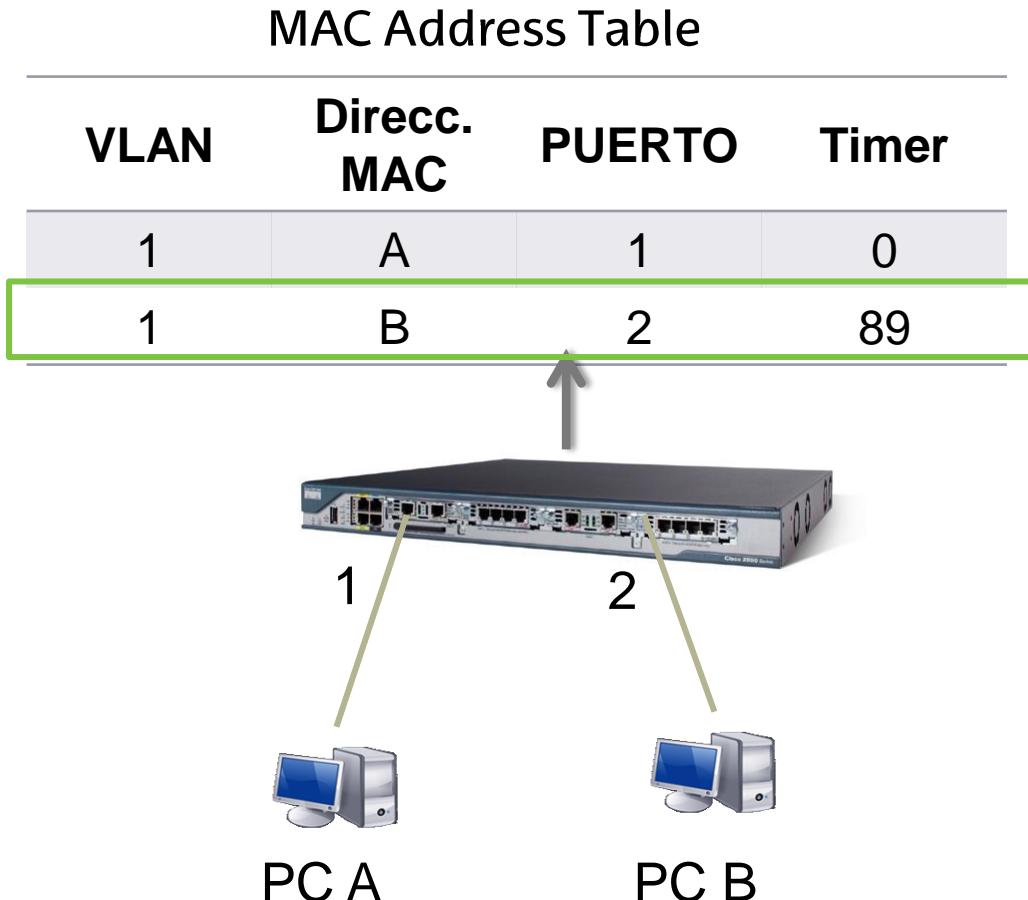


Un switch tradicional hace uso de la “MAC Address Table” para **reenviar** las tramas Ethernet

1. PCA envía una trama Ethernet para PCB
2. El switch consulta su “MAC Address Table”
  1. Aprovecha la “source MAC” address para refreshar la entrada de PCA

# Comparativa switch legacy vs switch OpenFlow

## Switch legacy



Un switch tradicional hace uso de la “MAC Address Table” para reenviar las tramas Ethernet

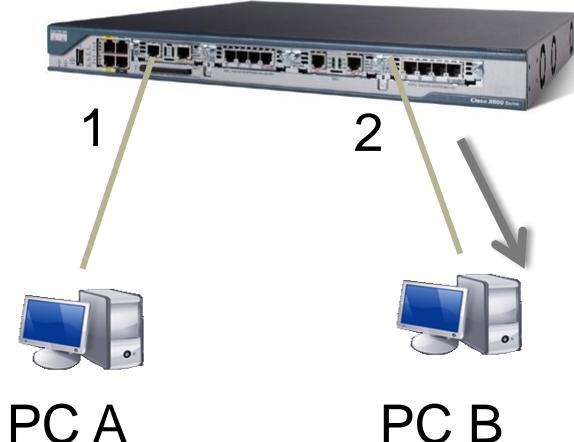
1. PCA envía una trama Ethernet para PCB
2. El switch consulta su “MAC Address Table”
  1. Aprovecha la “source MAC” address para refreshar la entrada de PCA
  2. Busca la MAC del “destination MAC”
    1. Encontrada → unicast forwarding
    2. No-encontrada → unicast flooding

# Comparativa switch legacy vs switch OpenFlow

## Switch legacy

MAC Address Table

VLAN	Direcc. MAC	PUERTO	Timer
1	A	1	0
1	B	2	89



Un switch tradicional hace uso de la “MAC Address Table” para **reenviar** las tramas Ethernet

1. PCA envía una trama Ethernet para PCB
2. El switch consulta su “MAC Address Table”
  1. Aprovecha la “source MAC” address para refreshar la entrada de PCA
  2. Busca la MAC del “destination MAC”
    1. Encontrada → unicast forwarding
    2. No-encontrada → unicast flooding
3. El switch reenvía la trama Ethernet a través del Puerto 2

# Comparativa switch legacy vs switch OpenFlow

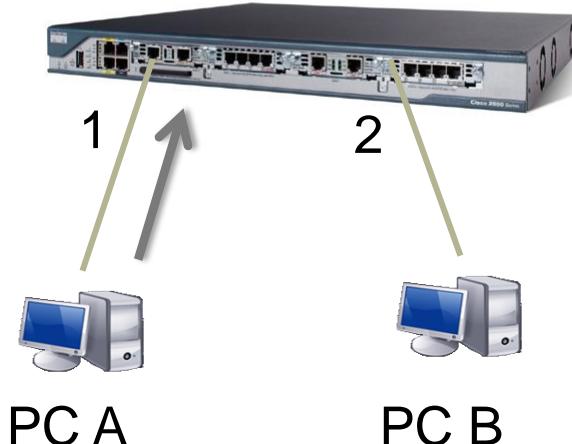
## Switch OpenFlow

Flow Table (simplificada)

Match Field	Instructions
L2,L3,L4 match1	action1
L2,L3,L4 match2	action2

Un switch OpenFlow hace uso de la “Flow Table” para procesar las tramas Ethernet

1. PCA envía una trama Ethernet para PCB



# Comparativa switch legacy vs switch OpenFlow

## Switch OpenFlow

Flow Table (simplificada)

### Match Field

### Instructions

L2,L3,L4 match1

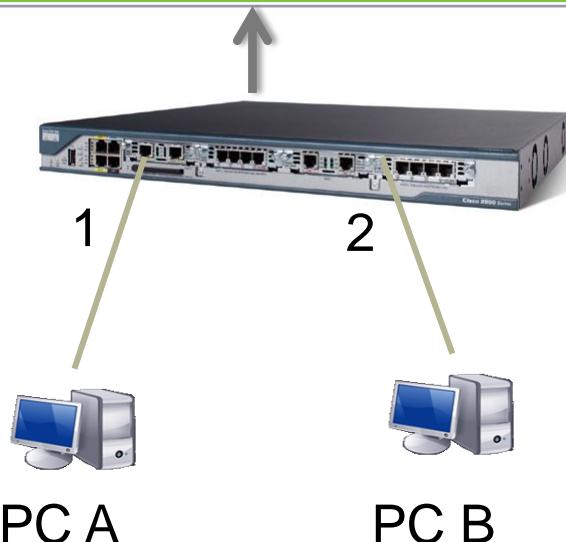
action1

L2,L3,L4 match2

action2

Un switch OpenFlow hace uso de la “Flow Table” para procesar las tramas Ethernet

1. PCA envía una trama Ethernet para PCB
2. El switch consulta la Flow Table para ver si el paquete matchea alguna entrada y realiza la acción correspondiente



# Comparativa switch legacy vs switch OpenFlow

## Switch OpenFlow

Flow Table (simplificada)

### Match Field

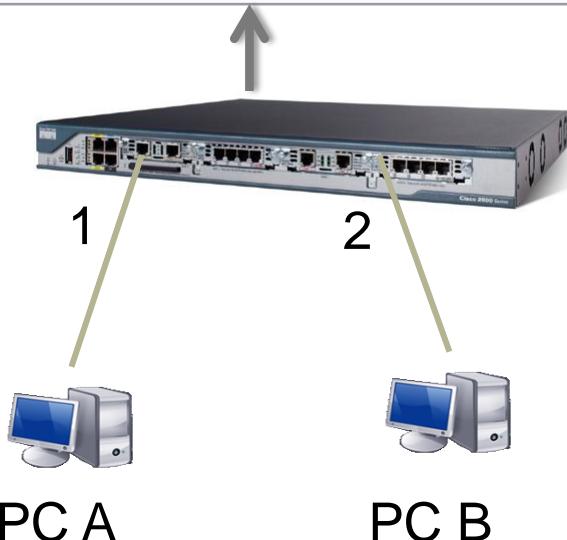
### Instructions

L2,L3,L4 match1

action1

L2,L3,L4 match2

action2



Un switch OpenFlow hace uso de la “Flow Table” para procesar las tramas Ethernet

1. PCA envía una trama Ethernet para PCB
2. El switch consulta la Flow Table para ver si el paquete matchea alguna entrada y realiza la acción correspondiente

Un switch OpenFlow por defecto no se comporta como un switch legacy. ¡Hay que programarlo! ¡Hay que llenar la Flow Table!

# Ejemplos

Dependiendo de cómo definamos los flujos, el mismo equipo se puede comportar como un hub, un switch, un router, etc...

## Switching

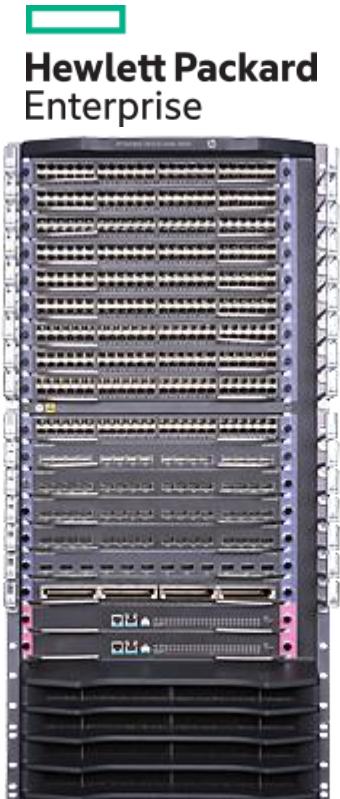
Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	*	00:1f...	*	*	*	*	*	*	*	port6

## 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

# Productos y soluciones disponibles

- **Equipamiento de Red (hybrid)**



HPE FlexFabric 12900



Serie Nexus 5500



QFX5100 Switches



Serie IP8800

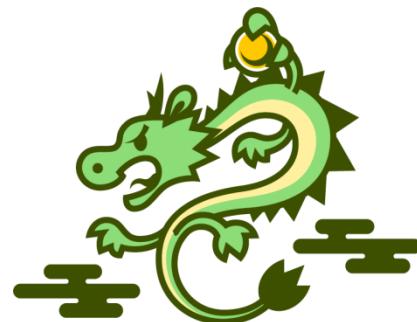
# Productos y soluciones disponibles

- Equipamiento de Red (whiteboxes)



# Productos y soluciones disponibles

- Controladores (open source)



Floodlight

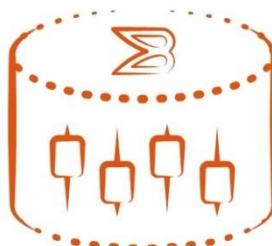


# Productos y soluciones disponibles

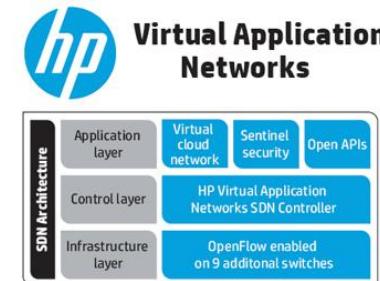
- **Controladores (propietarios)**



Big Cloud Fabric  
(Big Switch Networks)



Brocade SDN controller  
(Brocade Communication Systems)



VAN  
(HP)



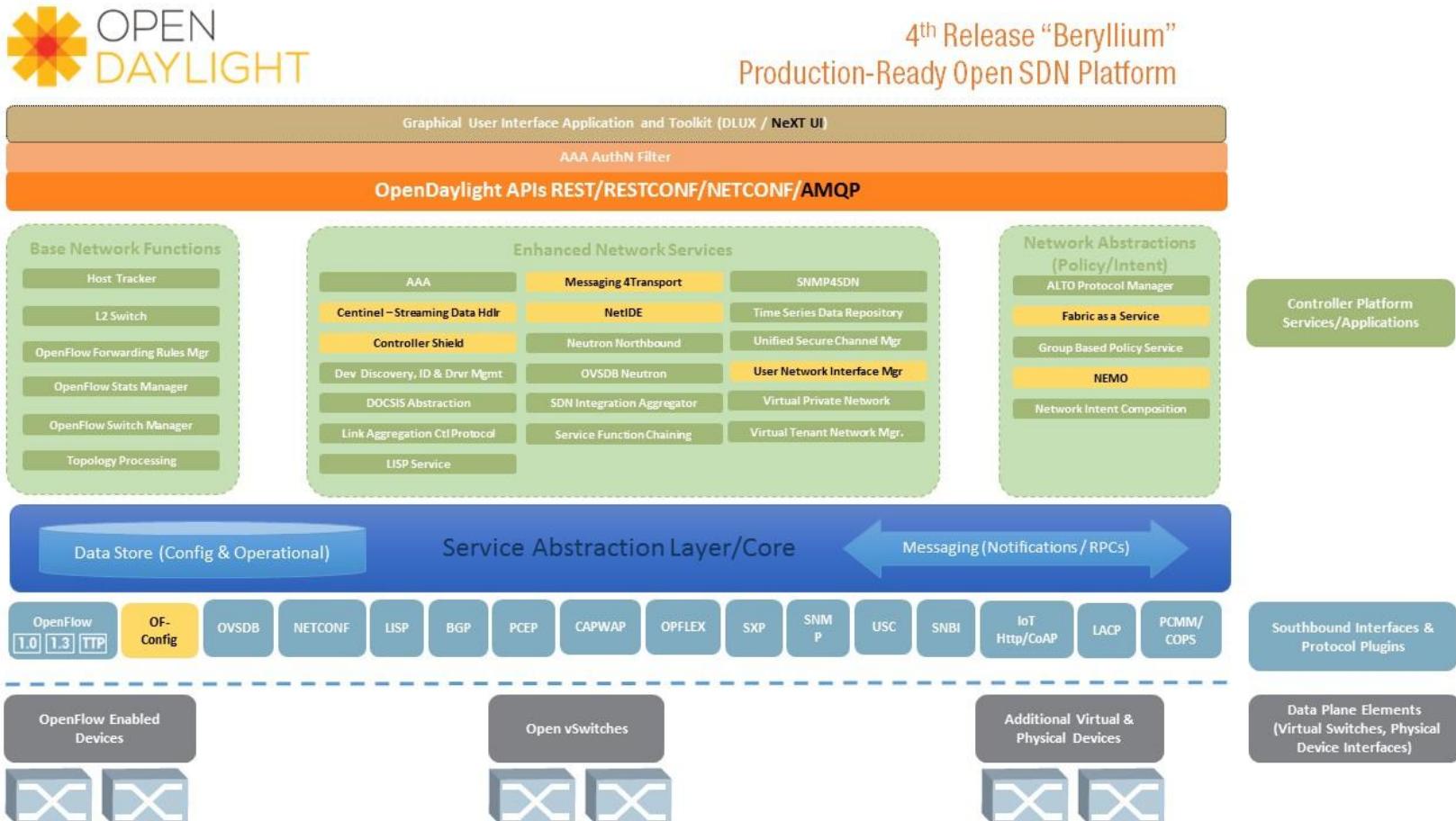
Programmable Network Controller  
(IBM)

## ProgrammableFlow

Programmable Flow  
(NEC)

# Productos y soluciones disponibles

- Sistemas operativos de red (open source)

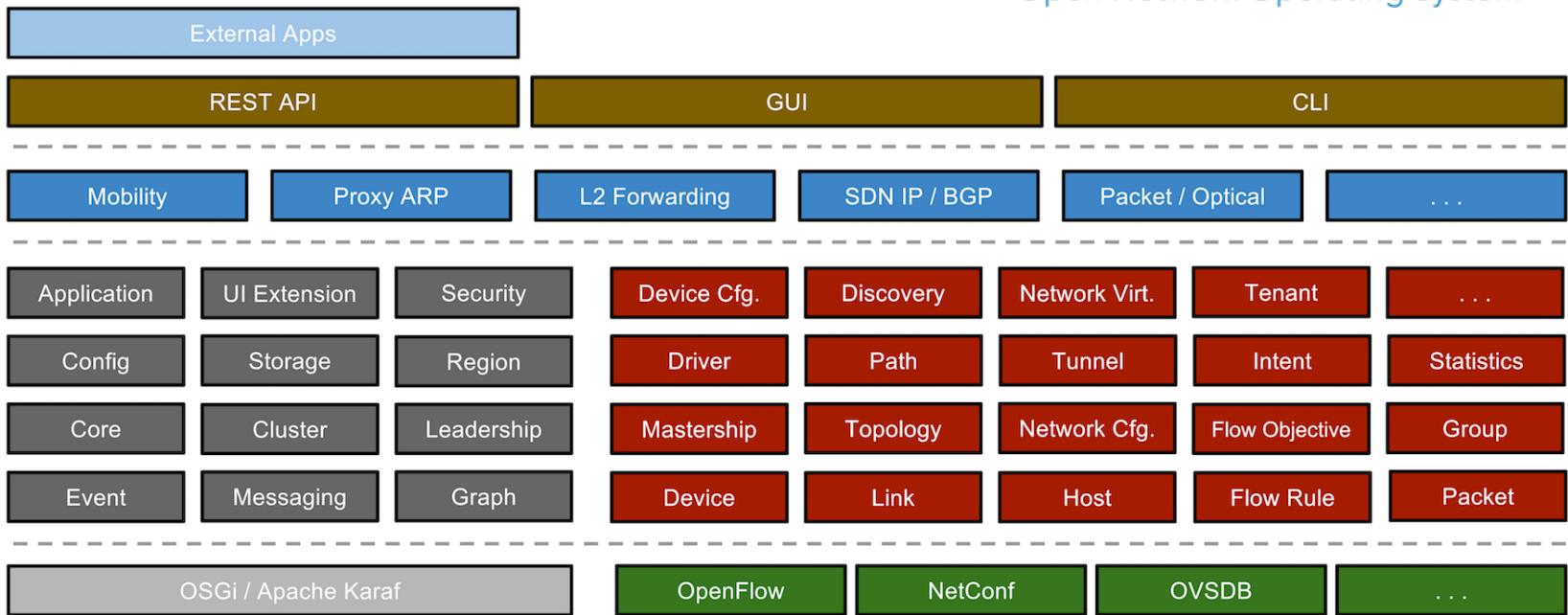


# Productos y soluciones disponibles

- Sistemas operativos de red (open source)



**ONOS**  
Open Network Operating System



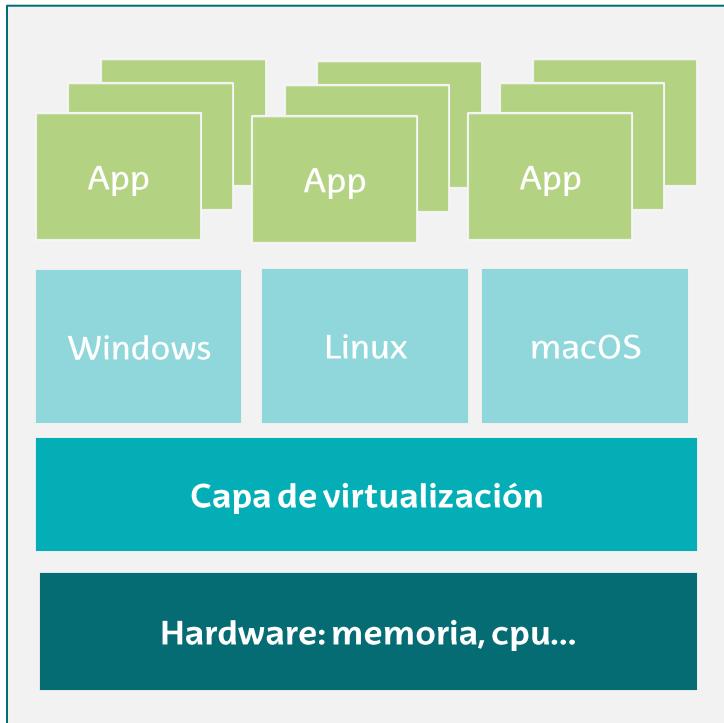
# Aplicaciones

# Virtualización de red

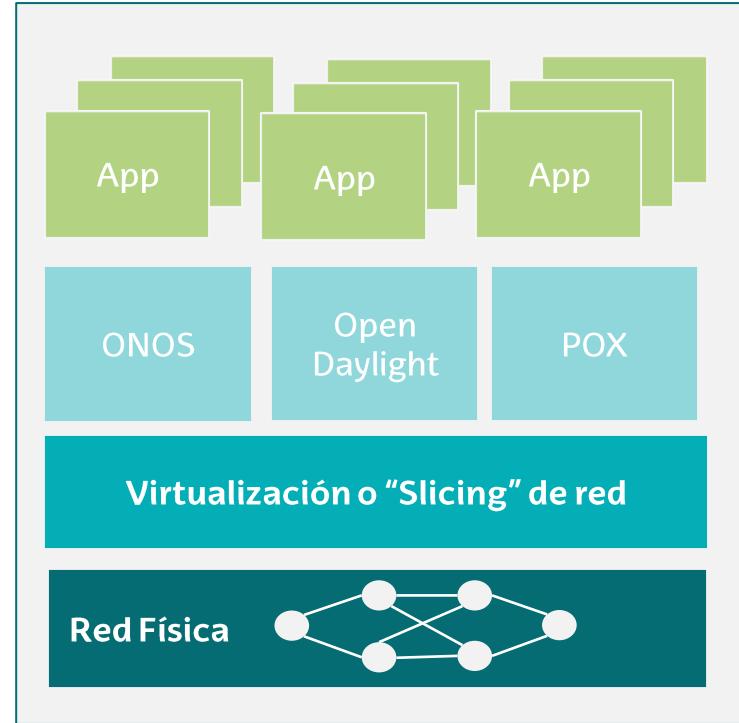
- **Abstracción de la red física:**
  - Soporte para múltiples redes lógicas ejecutándose en un sustrato físico común compartido.
  - Un contenedor de servicios de red.
- **Casos de uso**
  - Centro de datos definido por software
  - Redes de experimentación

# Virtualización de red

- **Equiparación de la infraestructura de red a la infraestructura de computación.**



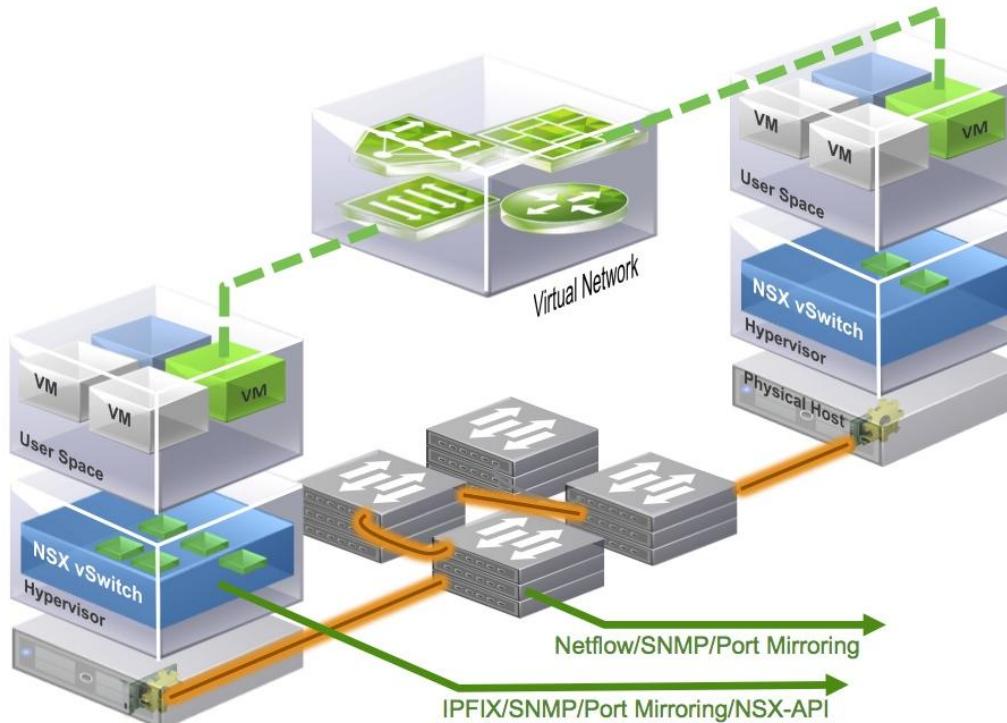
Computación



Red

# Virtualización de red

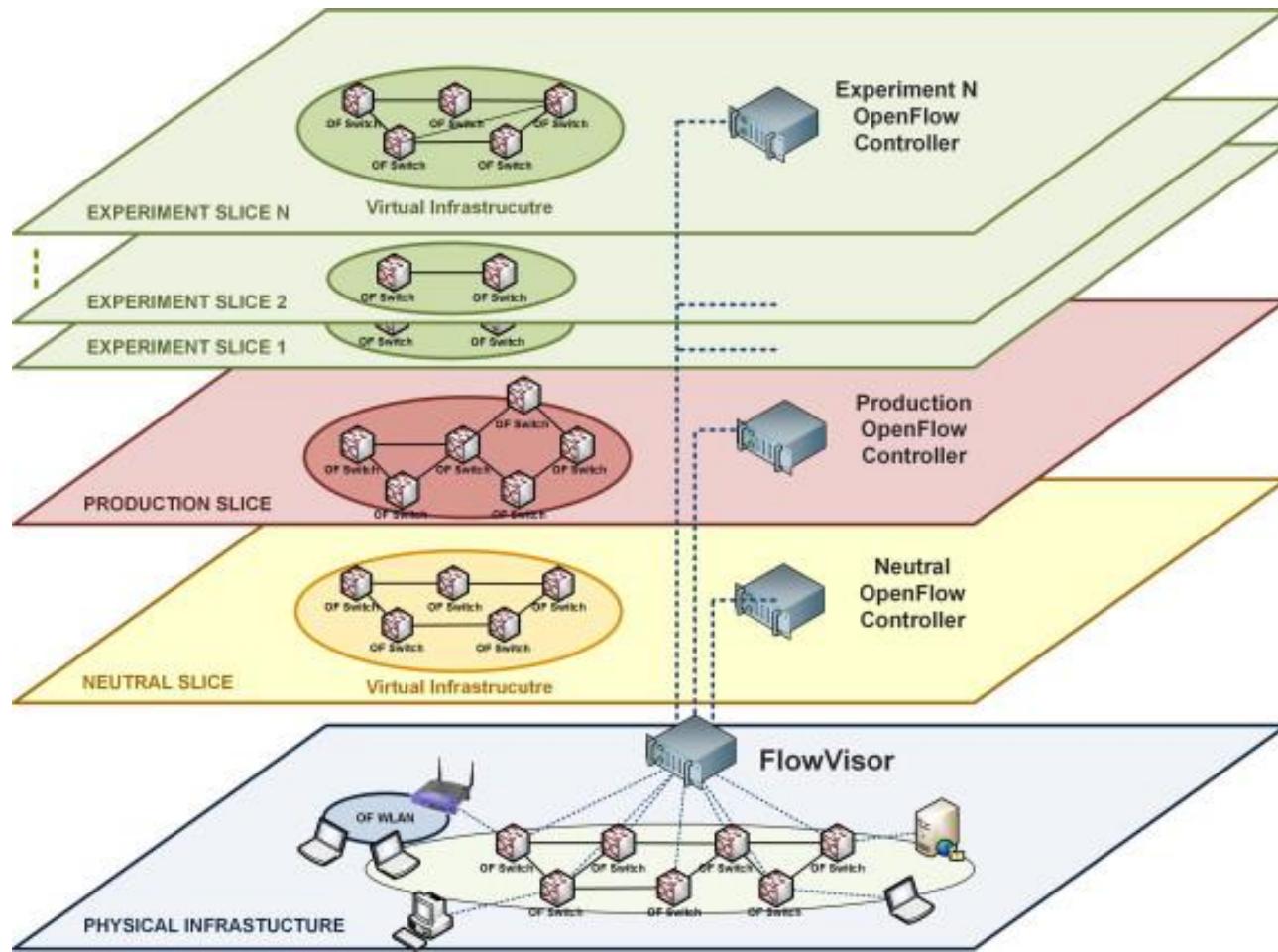
- **Software-Defined Data Center**
  - Extiende la virtualización a todos los recursos presentes en un data center.
  - ITaaS = (Network + Compute + Storage + Security)aaS



Fuente: <http://www.vmware.com/es/products/nsx.html>

# Virtualización de red

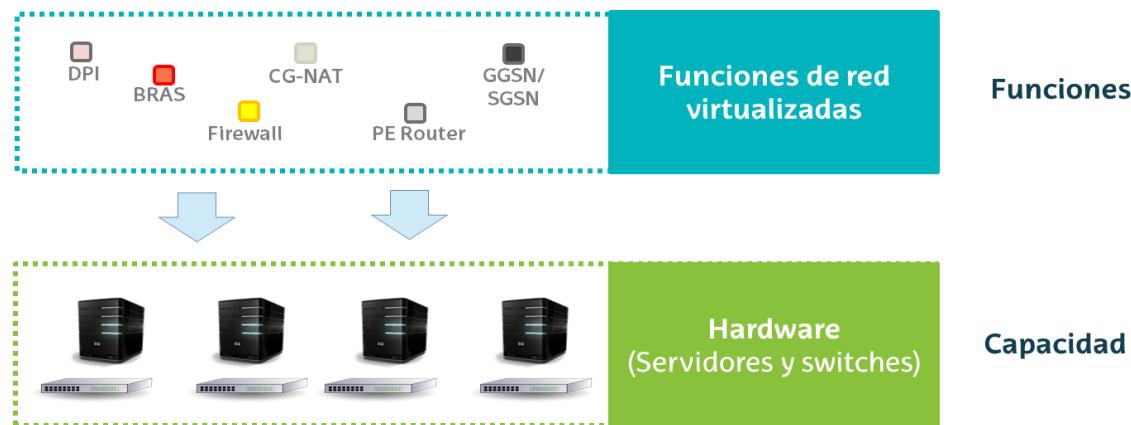
- Experimentación en redes campus



Fuente: <http://i2t.ehu.eus/resources/ehu-oef>

# Network Functions Virtualization

- ¿Qué es?



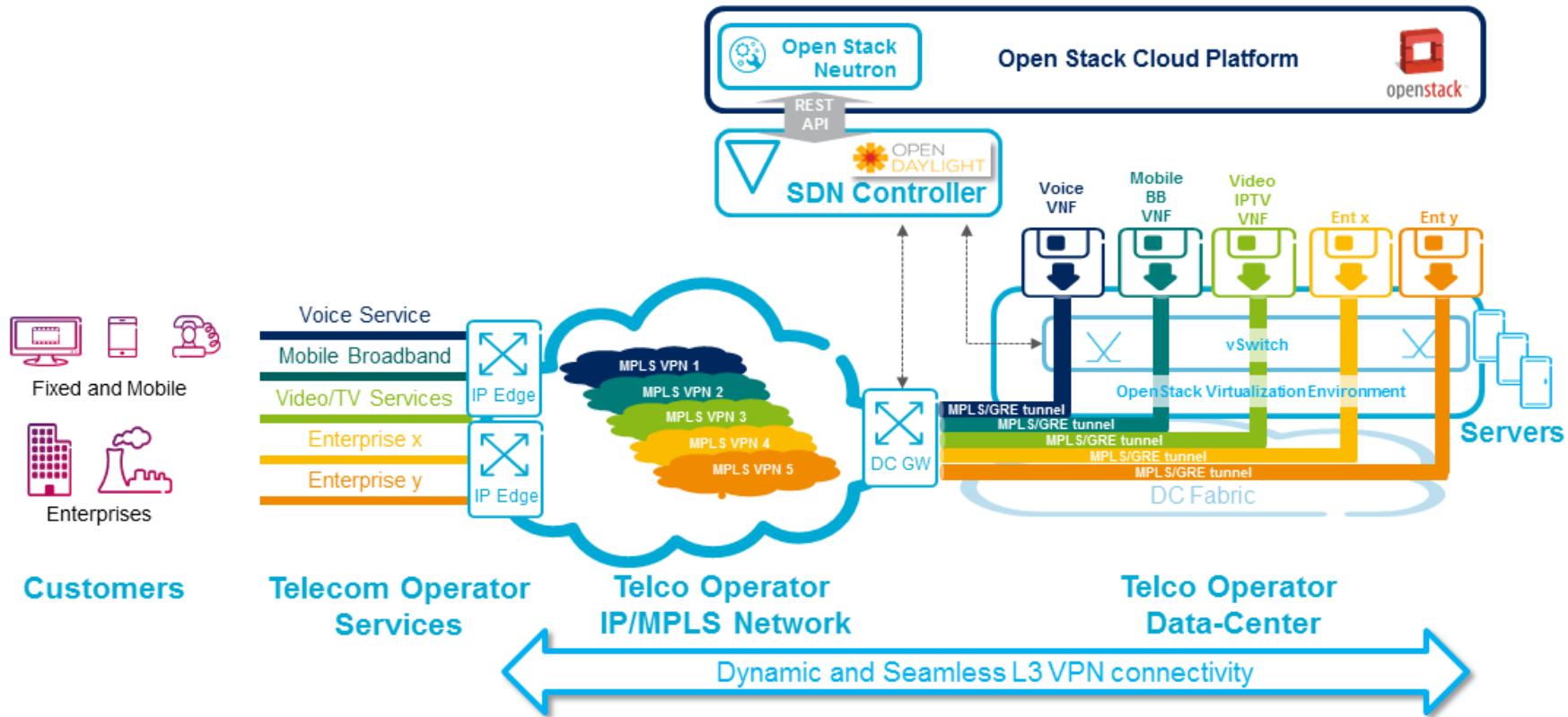
Las funciones de red implementadas hasta ahora en los routers/modems se llevan a un cloud.



Fuente: Telefónica I+D

# Network Functions Virtualization

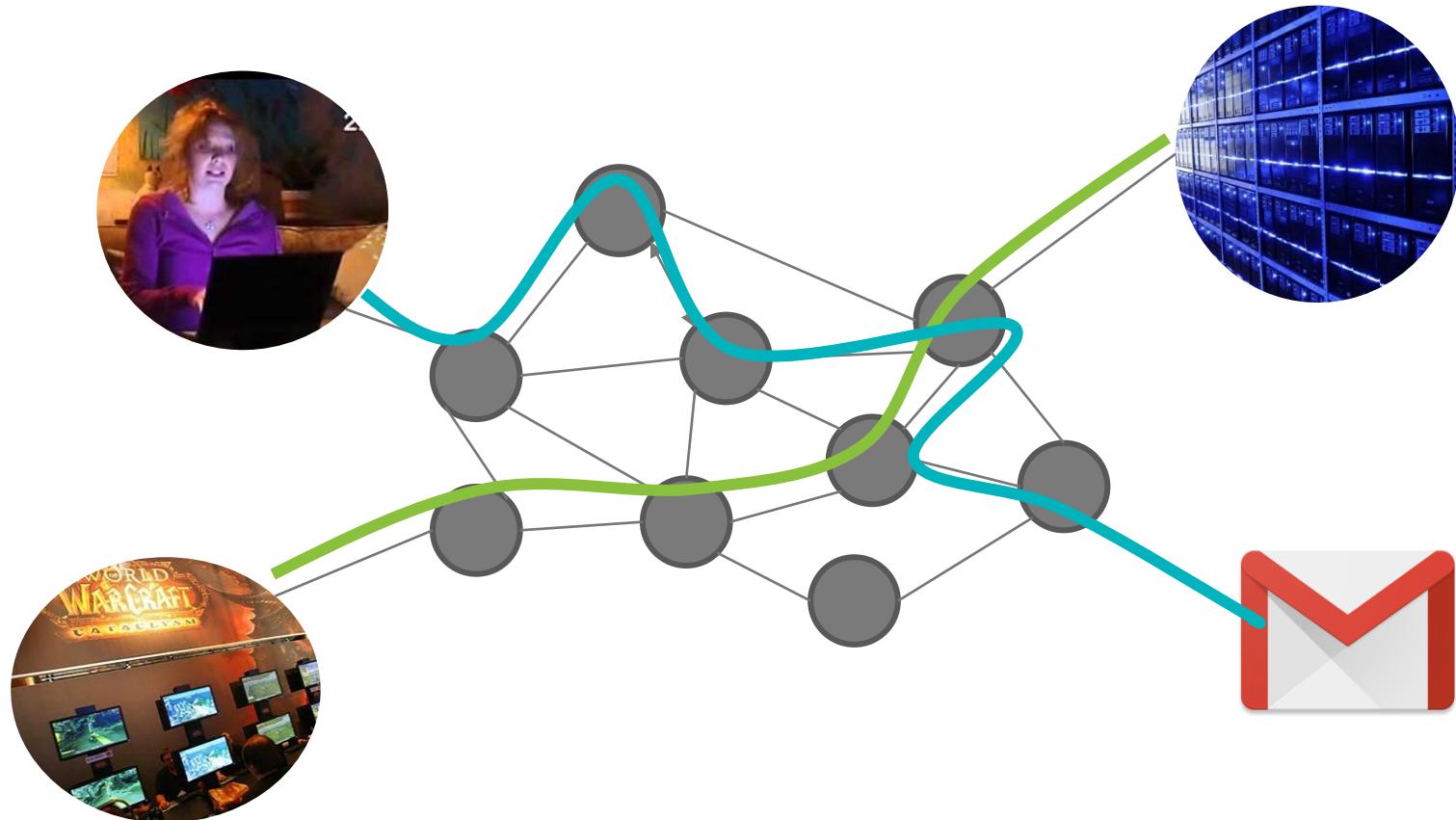
- ¿Cuál es el rol de las SDN?



Fuente: <https://www.ericsson.com/spotlight/cloud/blog/2015/06/15/att-ericsson-presenting-sdn-based-l3vpn-solution-telco-nfv-needs/>

# Ingeniería de tráfico

- Optimizar el rendimiento de la red.
  - Selección de rutas óptimas.



# Ingeniería de tráfico



Servicios de conectividad extremo a extremo con garantías de ancho de banda

# Ingeniería de tráfico

## • SD-WAN

- Cisco Intelligent WAN
  - Capaz de detectar el consumo de ancho de banda por aplicación.
  - Gestiona la QoS en el gateway entre la LAN y la WAN.
  - Selecciona el camino óptimo para cada aplicación en base a sus requerimientos: baja latencia, etc.

Figure 1. Magic Quadrant for WAN Optimization



Source: Gartner (May 2016)

# Las SDN también serán esenciales para...

- **Internet of Things**
- **Industria 4.0**
- **5G**

# MININET

# ¿Qué es Mininet?

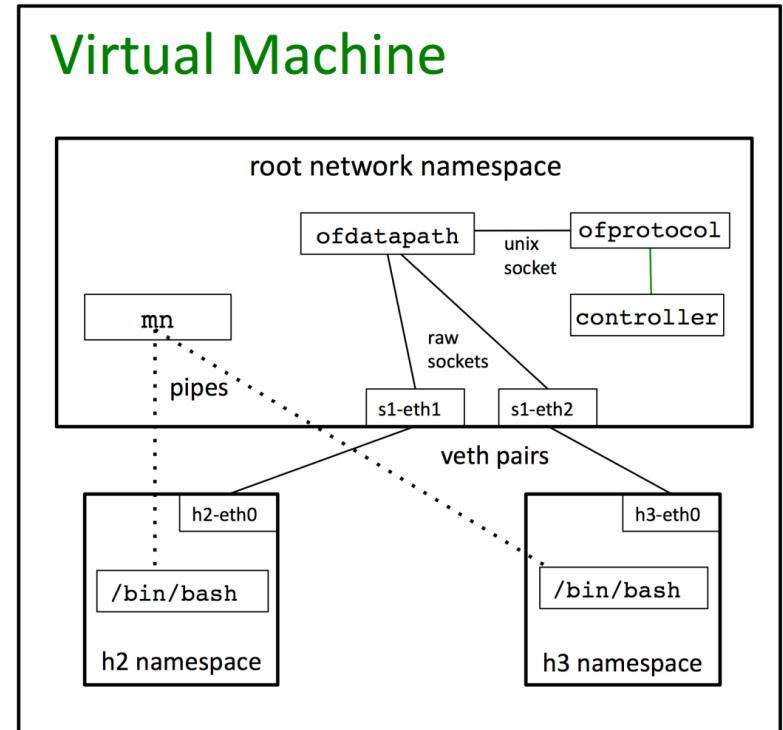
- **Emulador de red**
- **Basado en virtualización de sistema operativo ligera para alcanzar escalabilidad.**
- **Ventajas:**
  - Rapidez.
  - Posibilidad de crear topologías de red customizadas.
  - Posibilidad de ejecutar programas reales (cualquier cosa que pueda ejecutarse en Linux puede ejecutarse en un host Mininet).
  - Open source.

# Alternativas a mininet para trabajar con OpenFlow

- **Sistemas reales:**
  - Difíciles de configurar.
  - Elevado coste económico.
- **Máquinas virtuales en red:**
  - Problemas de escalabilidad.
- **Simulador:**
  - No es el camino al despliegue hardware.

# ¿Cómo son las redes que emula?

- **Switches**
  - Bridges Open vSwitch.
- **Hosts**
  - Proceso bash.
  - Espacio de nombres de red propio.
- **Links**
  - Switch-switch
    - Socket
  - Switch-host
    - Enlaces virtuales (veth)
- **Controlador**



# COMANDOS

```
$ sudo mn [opciones]
```

- **Opciones**

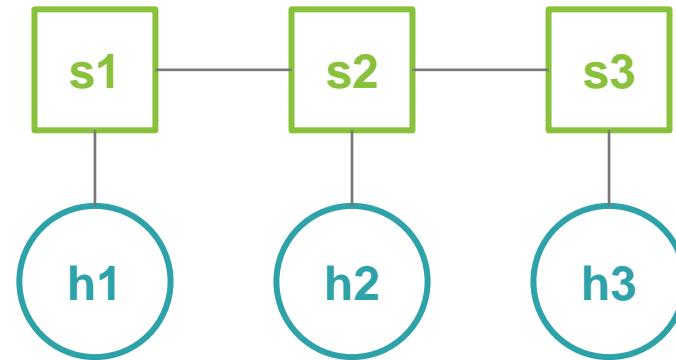
- **--topo:**
  - Define una topología a través de la línea de comandos después del arranque de mininet.
- **--switch:**
  - Define el switch a utilizar. (Por defecto Open vSwitch)
- **--controller:**
  - Define el controlador a utilizar.

*En la manpage tenéis todas las opciones disponibles. ;)*

# DEFINICIÓN DE TOPOLOGÍAS (i)

- **Python**

```
from mininet.net import Mininet  
  
from mininet.topo import LinearTopo  
  
Linear=LinearTopo(k=3)  
  
net=Mininet(topo=Linear)  
  
net.start()  
net.pingAll()  
net.stop()
```



# DEFINICIÓN DE TOPOLOGÍAS (ii)

## Miniedit

- Aplicación gráfica.
- Ejecución

# ENLACES DE INTERÉS

- <http://mininet.org/>
- <https://github.com/mininet/mininet>



# Gracias por vuestra atención

Alaitz Mendiola

[alaitz.mendiola@ehu.eus](mailto:alaitz.mendiola@ehu.eus)

BF-21

Christian Pinedo

[christian.pinedo@ehu.eus](mailto:christian.pinedo@ehu.eus)

BF-19