Software-Defined Networking / Network function virtualization

Grupo de Trabajo de SDN/NFV de CUDI

“La mejor forma de predecir el futuro es implementarlo.”

David Heinemeier Hansson

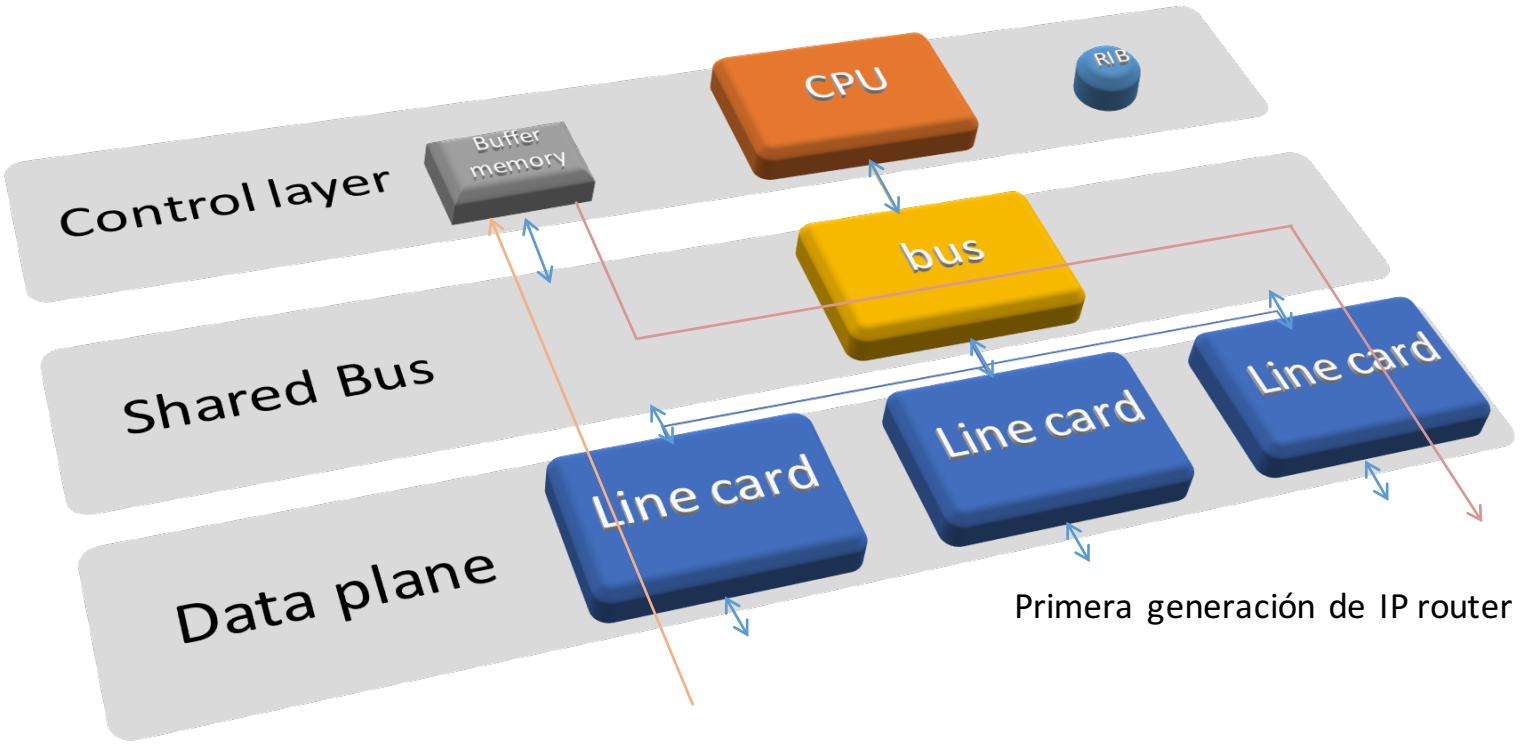
Resumen

- Objetivos del curso
  - Comprender la historia e importancia de los términos asociados con SDN.
  - Identificar varias controladoras SDN y sus objetivos de despliegue.
  - Articular los componentes y las vías de comunicación de la topología.
  - Aprender las especificaciones básicas, funciones y mensajes de OpenFlow.
  - Familiarizarse con la interfaz norte de un controlador SDN.
  - Examinar las metodologías básicas de solución de problemas (troubleshooting).
  - Obtener información sobre cómo la UDG está implementando OpenFlow en la red universitaria.
- Meta
  - A la finalización de esta clase el alumno tendrá un conocimiento práctico de OpenFlow y su puesta en práctica.

Evolución de las Redes e Historia de SDN / OpenFlow

Evolución de los IP Router

En el comienzo

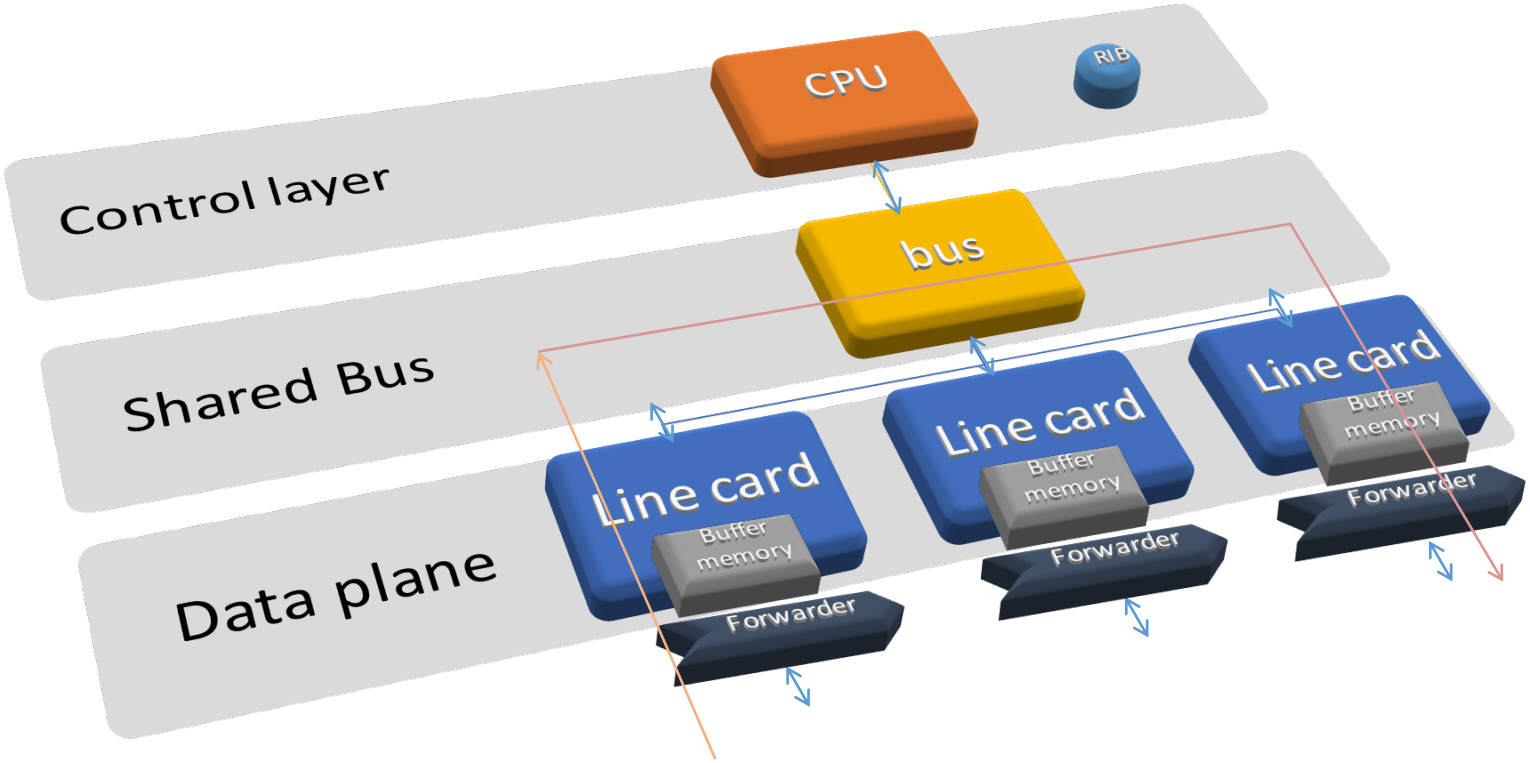


Primera generación de IP router

András Császár, http://www.ericsson.com/res/thecompany/docs/journal_conference_papers/packet_technologies/evolution_of_router_architectures_and_ip_networks.pdf

Evolución de los IP Router

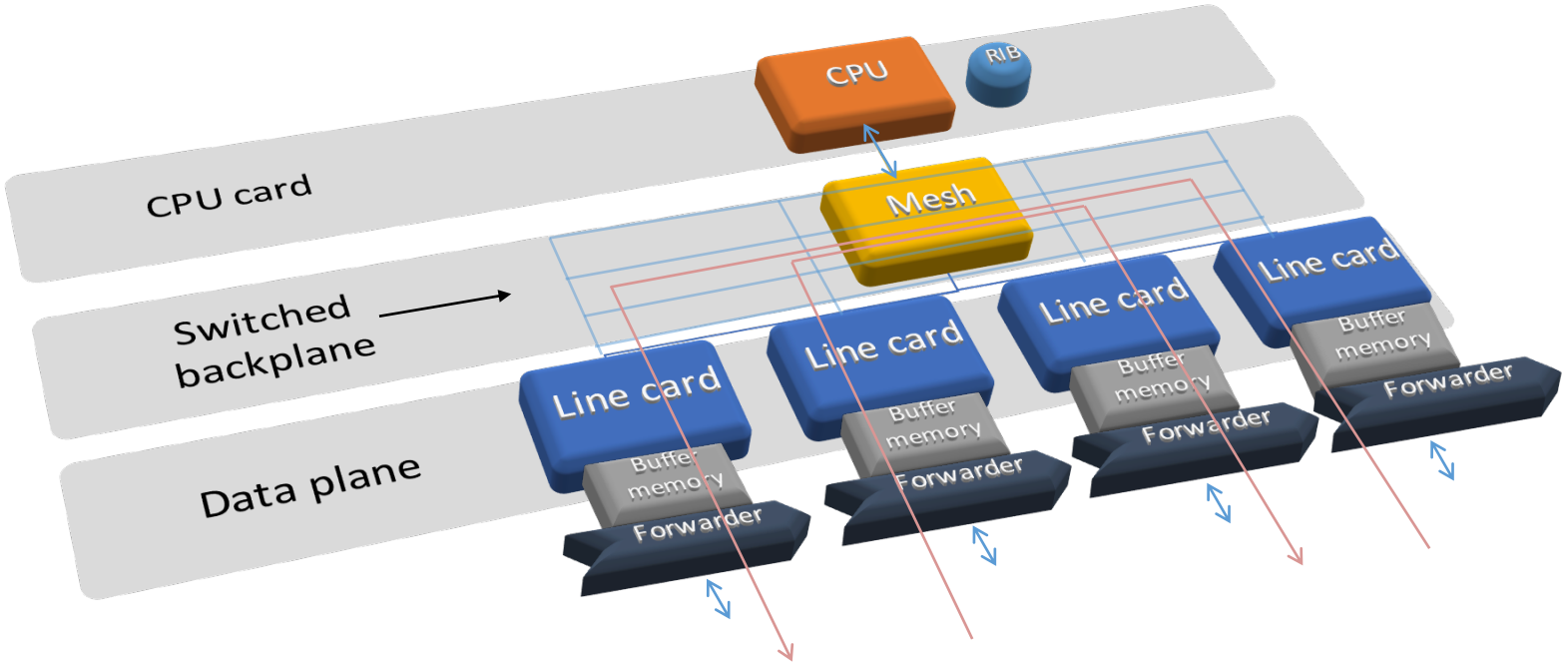
Segunda generación de IP router



András Császár, http://www.ericsson.com/res/thecompany/docs/journal_conference_papers/packet_technologies/evolution_of_router_architectures_and_ip_networks.pdf

Evolución de los IP Router

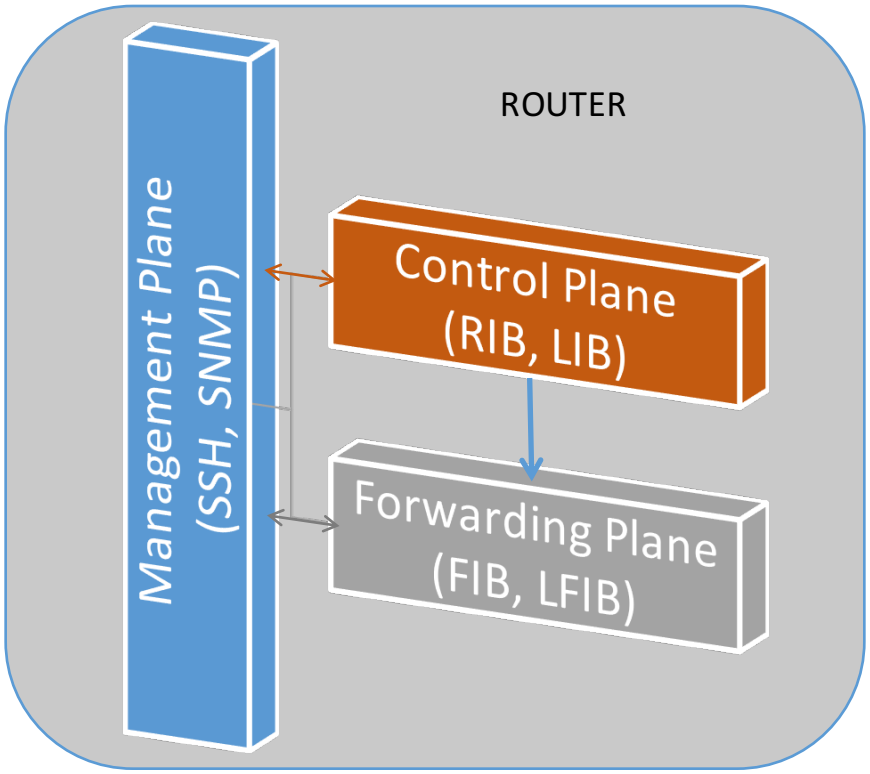
Tercera generación de IP router



András Császár, http://www.ericsson.com/res/thecompany/docs/journal_conference_papers/packet_technologies/evolution_of_router_architectures_and_ip_networks.pdf

Evolución de los IP Routers

Separación de planos

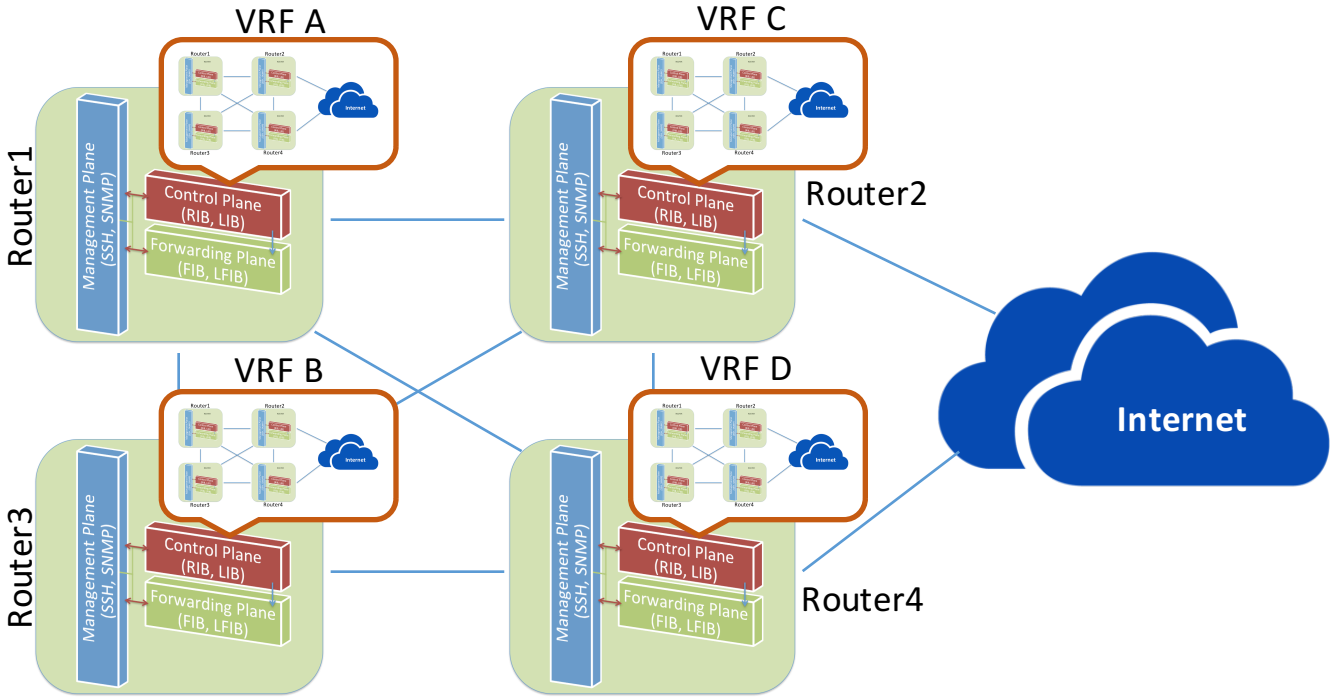


Planos en los Routers

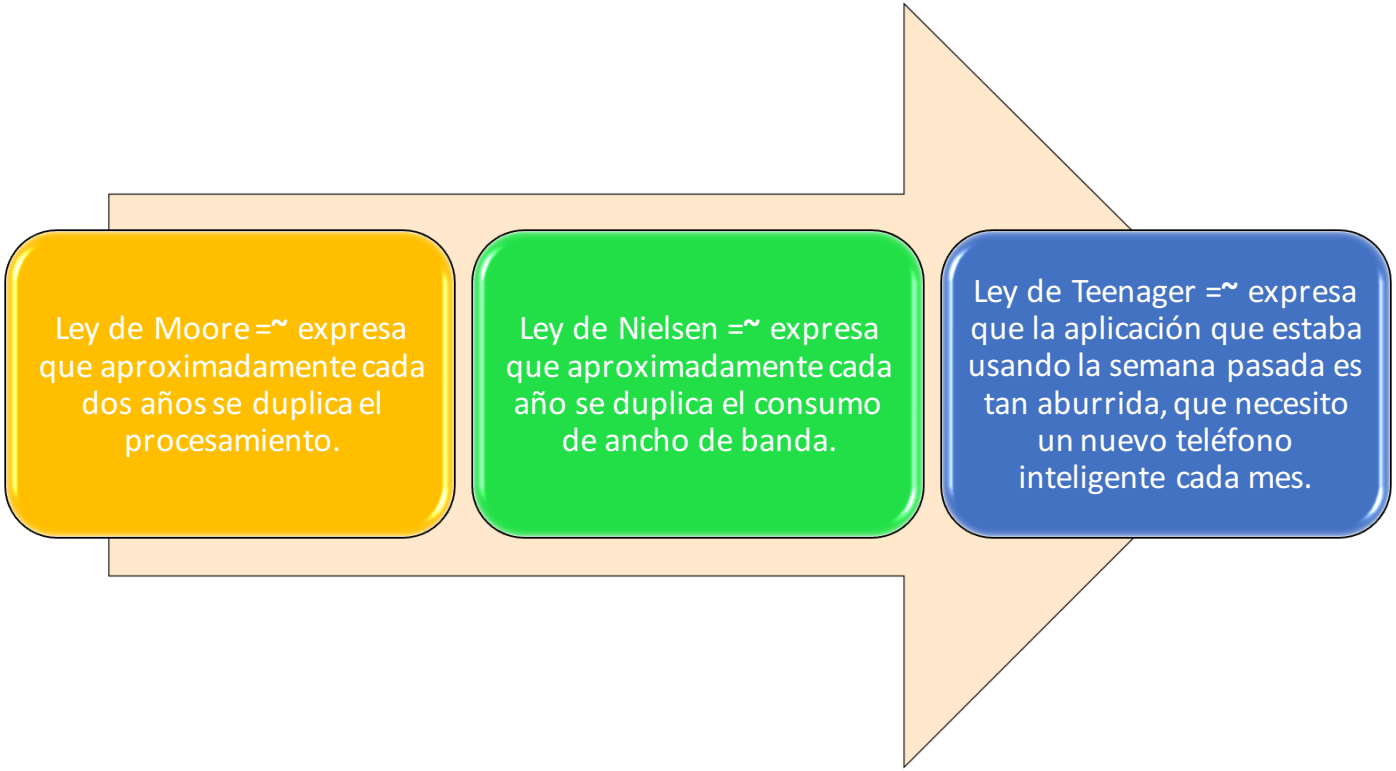
- Los routers tienen tres planos bien identificados:
  - Forwarding
  - Control
  - Management
- El desarrollo arquitectónico había tenido lugar casi exclusivamente para el plano de *datos* o *forwarding*.
- El plano de *control* se había mantenido prácticamente igual.

Ejemplo de Backbone

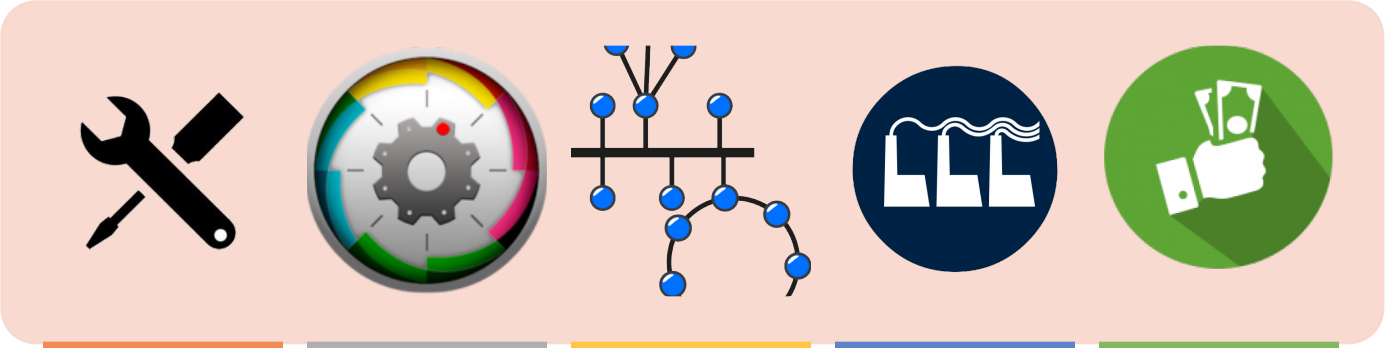
IP/MPLS



Leyes


Ley de Moore =~ expresa que aproximadamente cada dos años se duplica el procesamiento.
Ley de Nielsen =~ expresa que aproximadamente cada año se duplica el consumo de ancho de banda.
Ley de Teenager =~ expresa que la aplicación que estaba usando la semana pasada es tan aburrida, que necesito un nuevo teléfono inteligente cada mes.

Desafío de las redes tradicionales



Configuración manual de la red.

A falta de procesos automatizado, implica más recursos para el control y mantenimien-to.

El proceso de creación o modificación de la topologías de red puede demorar horas o días.

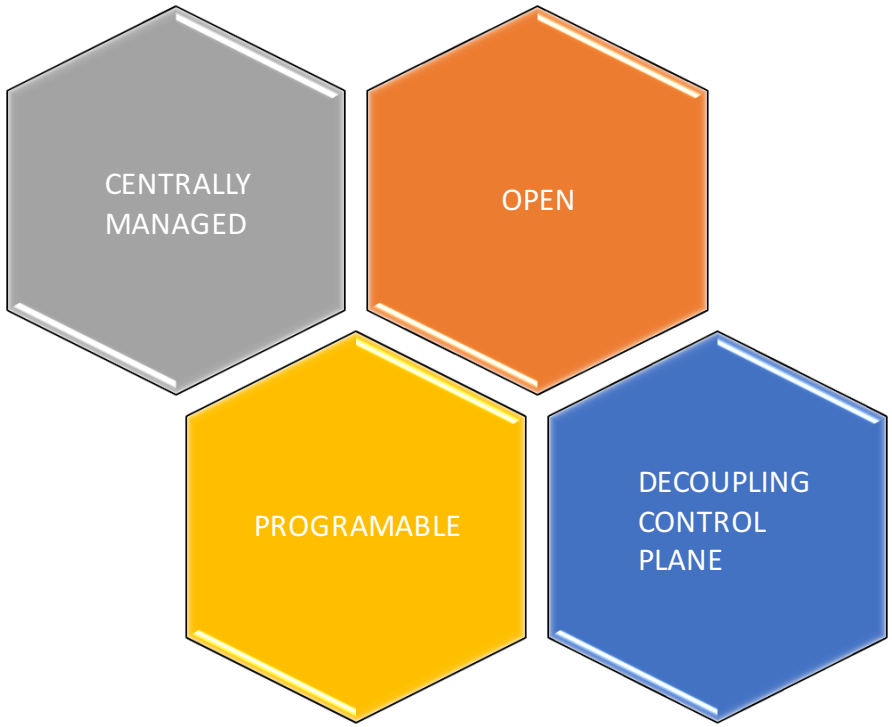
Las habilidades necesarias para realizar todos estos procesos son complejas, y difieren de la solución de un fabricante a otro.

Costosa adquisición de infraestructu-ra de seguridad para control abusos de la red y negaciones de servicio.

Nuevos requisitos de la red

- Eficiencia
  - Simplificar la red y las operaciones.
- Velocidad
  - Implementación rápida de nuevos servicios.
- Innovación
  - Poner en práctica nuevos modelos de negocio.

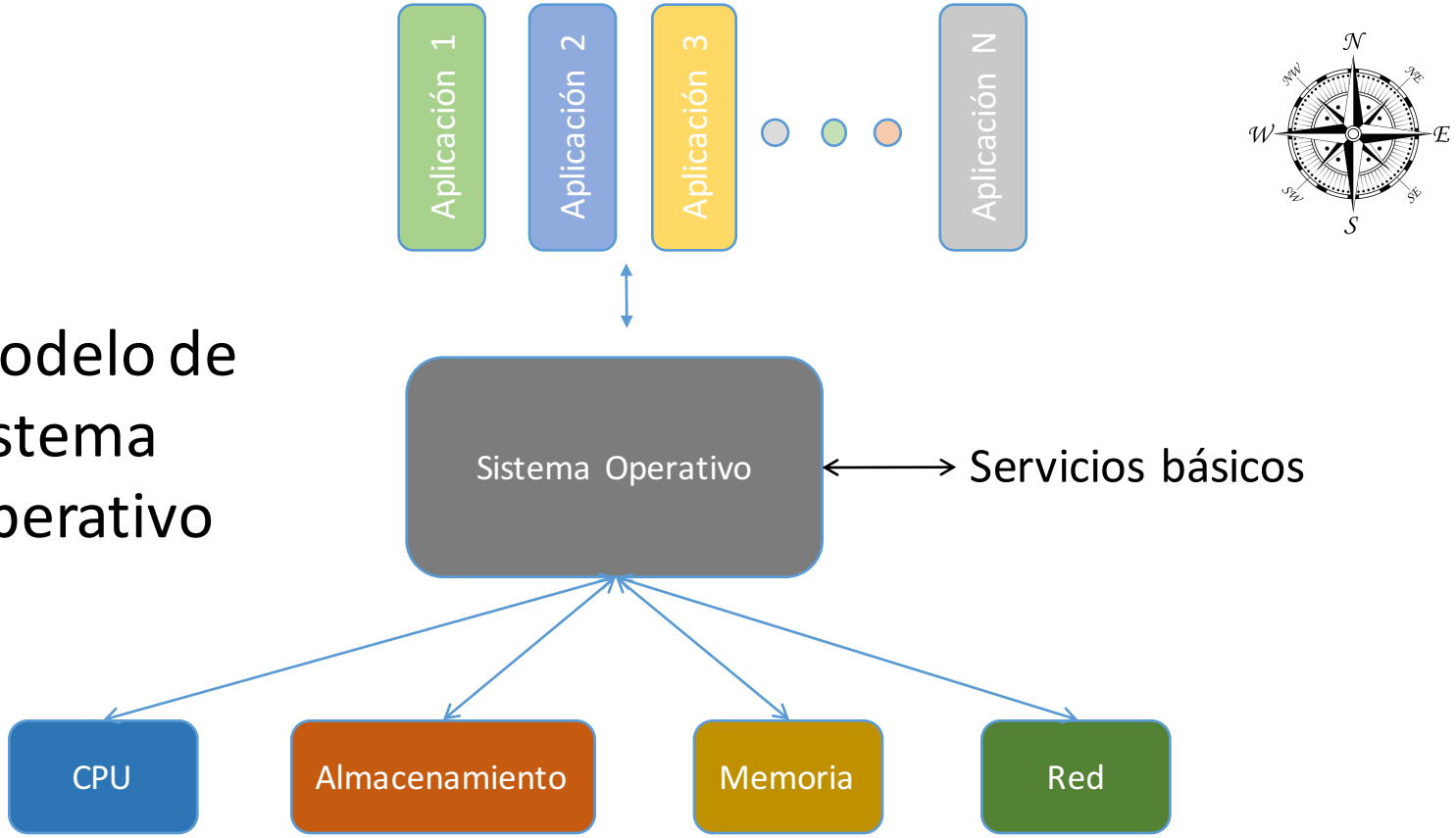
Big Picture


CENTRALLY MANAGED
OPEN
PROGRAMABLE
DECOUPLING CONTROL PLANE

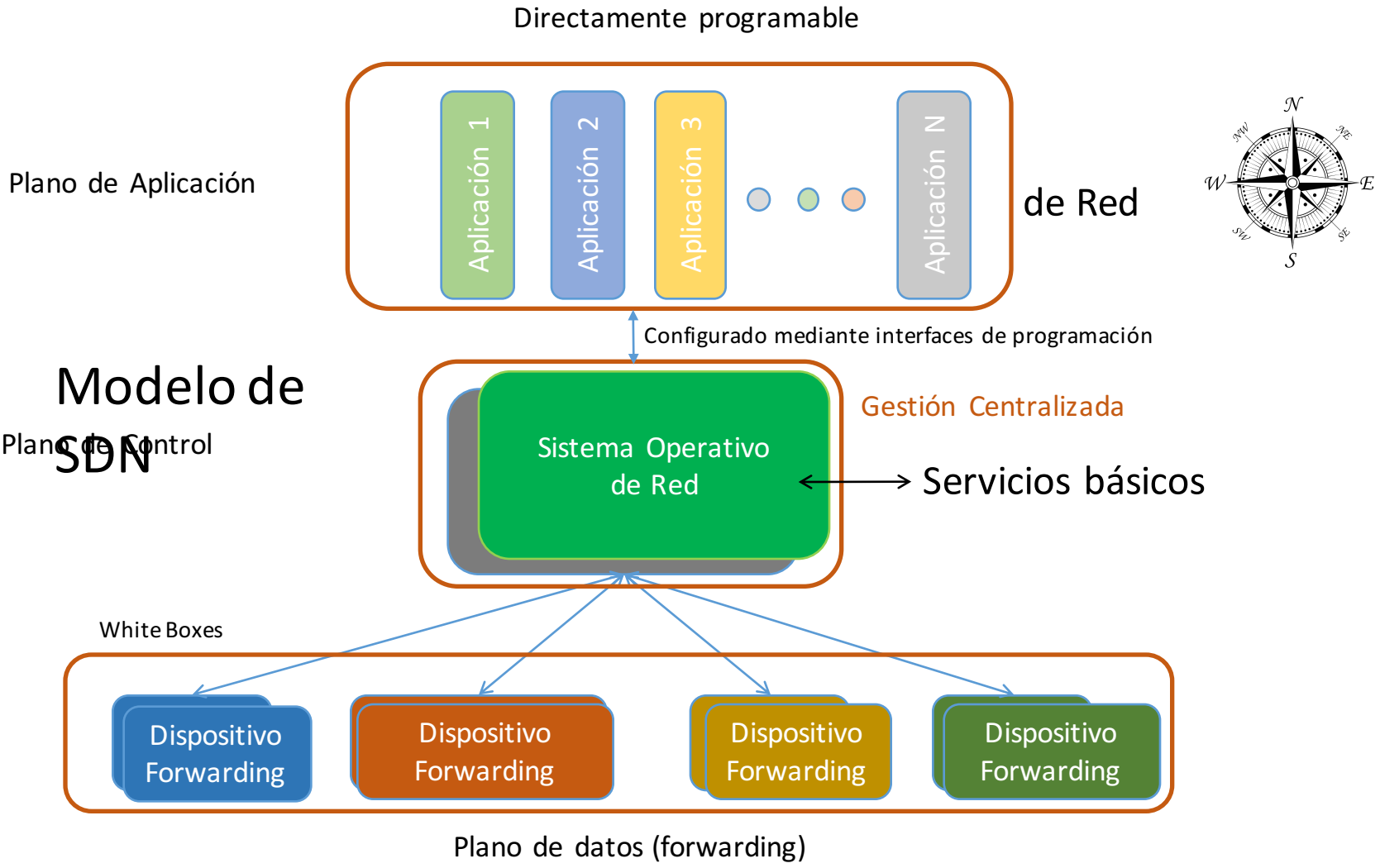
Arquitectura

Modelo de Sistema Operativo

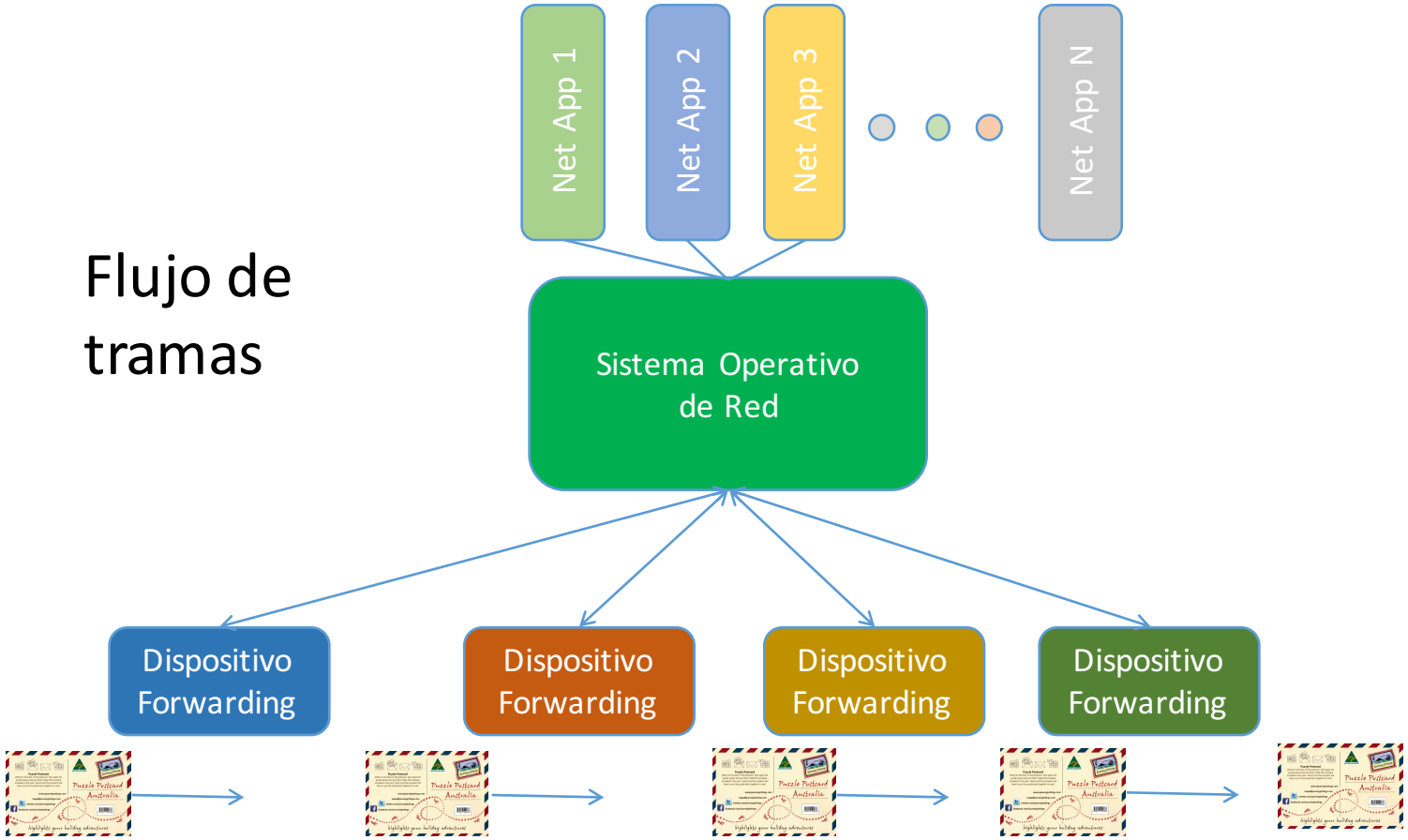
Modelo de Sistema Operativo



Modelo de SDN

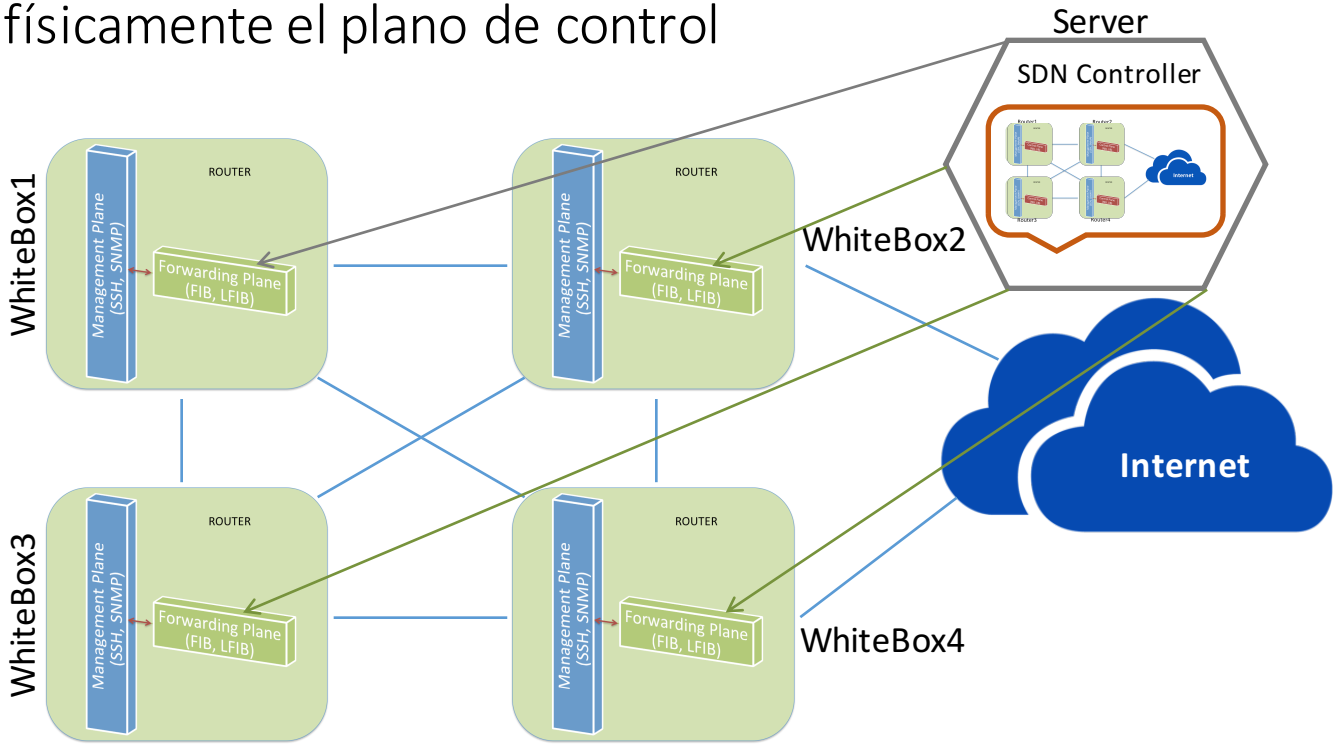


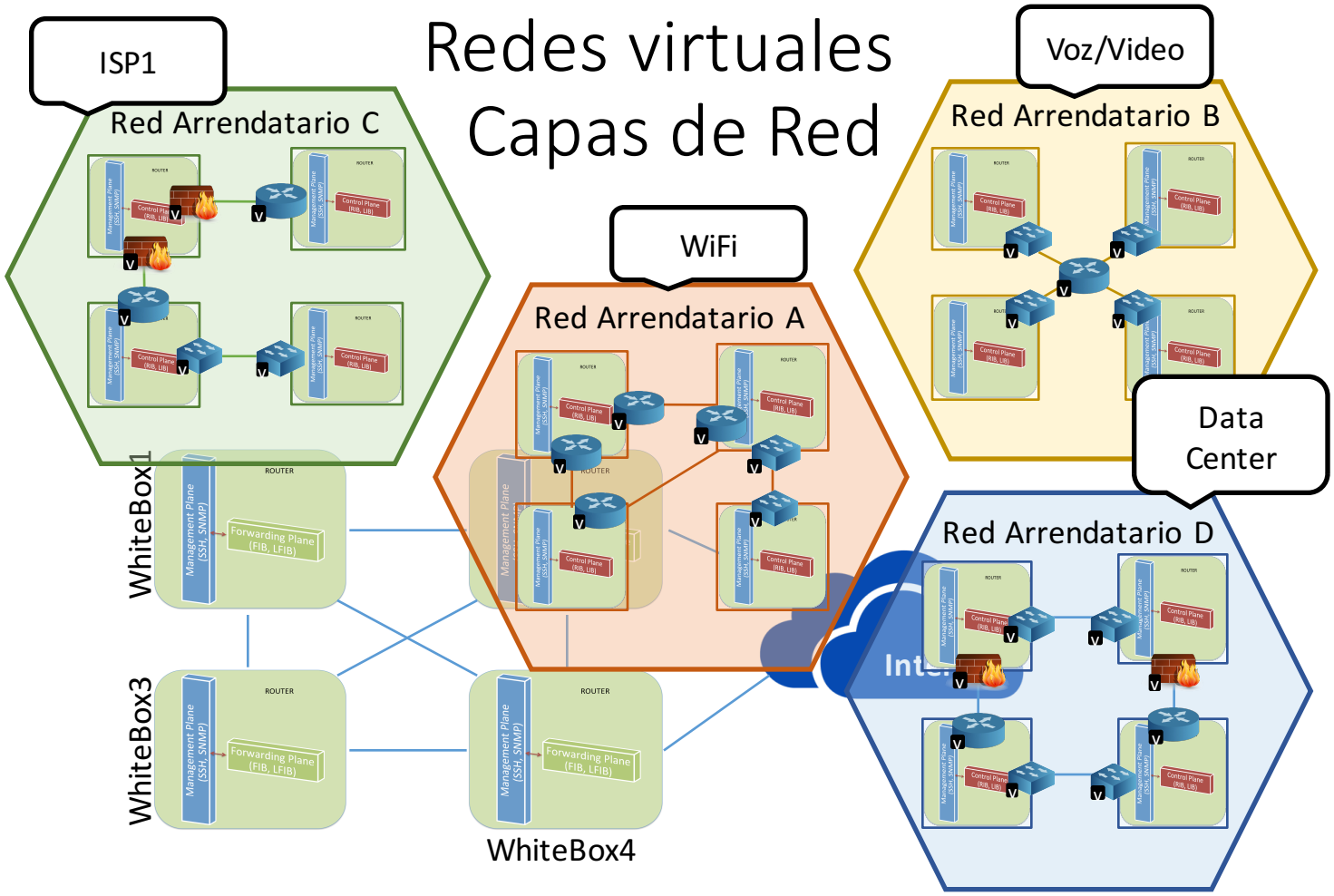
Flujo de tramas



Eficiencia

Desacoplar físicamente el plano de control

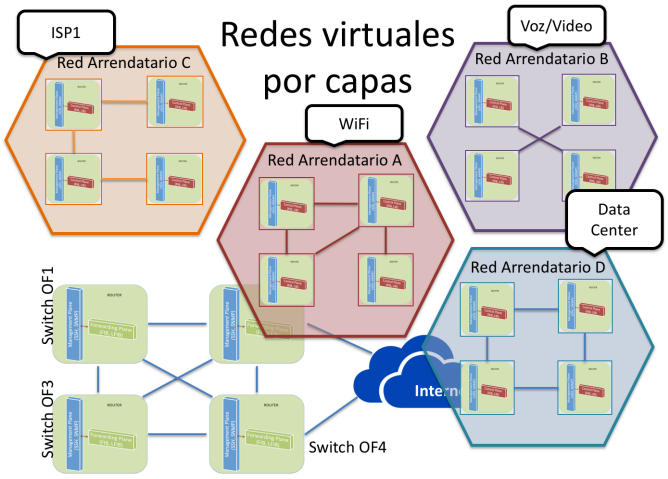




Fuente: http://www.ericsson.com/spotlight/cloud/blog/2015/02/17/5g-network-slices/

VTN: Red de Arrendamiento Virtual

- Un minuto ¡Esto es MPLS!



NO,

- es una administración centralizada
- es más ágil
- es configurado mediante programación

Esto es Software-Defined Networking

Fuente: http://www.ericsson.com/spotlight/cloud/blog/2015/02/17/5g-network-slices/

Visión historica de SDN

Los despliegues de SDN/OpenFlow fueron inicialmente de investigación en las redes de las universidades (Stanford).

Antes de la creación de la ONF en 2011, la especificación OpenFlow fue gestionada por un grupo de individuos reunidos físicamente en la Universidad de Stanford.

OpenFlow es sólo una instancia de los principios SDN.

SDN es una herramienta para permitir un mayor grado de control sobre los dispositivos de red.

¿Qué es OpenFlow?

- Es un estándar administrado por la Open Networking Foundation (ONF).
- Se originó en la Universidad de Stanford entre 2005-2012

Dr. Martín Casado

OpenFlow Switch Specification

Version 1.3.0 (Wire Protocol 0x04)
June 25, 2012

ONF TS-006

Fuente: https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.3.0.pdf

¿Qué es OpenFlow?

- OpenFlow es simplemente un protocolo de gestión de tabla de forwarding.
- Un Switch OpenFlow consiste de:
  - Una tabla de flujos,
  - que realiza las operaciones de búsqueda y expedición de tramas, y
  - un canal seguro a un controlador externo.

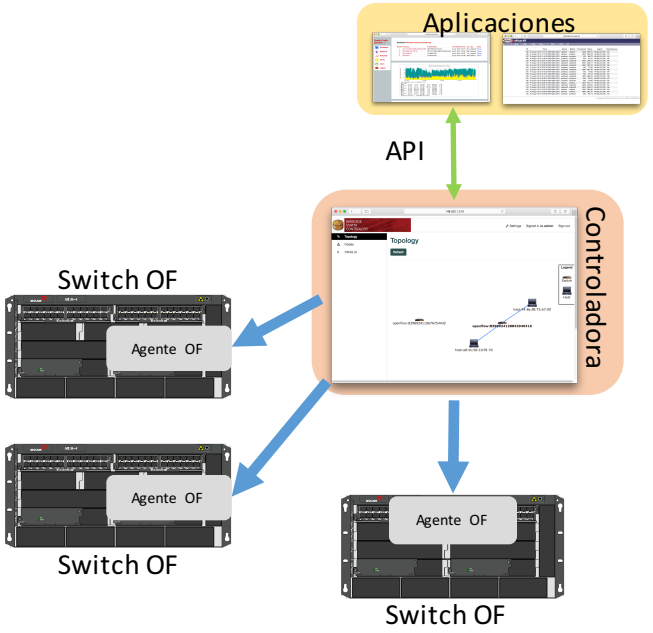
Tabla de flujos

- Entradas de flujo (valores de cabecera que coincidan contra las tramas).
- Contadores de actividad.
- Un conjunto de acciones que se aplican a las tramas que concuerden.
- Si no hay acciones que estén presentes, la trama es descartada.

OpenFlow Switch Specification – version 1.3.0
https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.3.0.pdf

Elementos básicos: Controladora y agentes

- Algunas funcionalidades están mejor implementadas desde la coordinación centralizada de todos los dispositivos en un dominio de red.



OpenFlow tiene 2 componentes

OpenFlow Controller

- Controla uno o más switches.
- Formula flujos.
- Programa switches.
- Las aplicaciones pueden ejecutarse en la mismo espacio de direcciones como la controladora.
- Las directivas pueden provenir de aplicaciones externas a través de REST API.

OpenFlow Switch

- El agente OpenFlow se ejecuta en el switch para comunicarse con la controladora.
- Las instrucciones provienen de la controladora para poblar la tabla de fujo del switch.

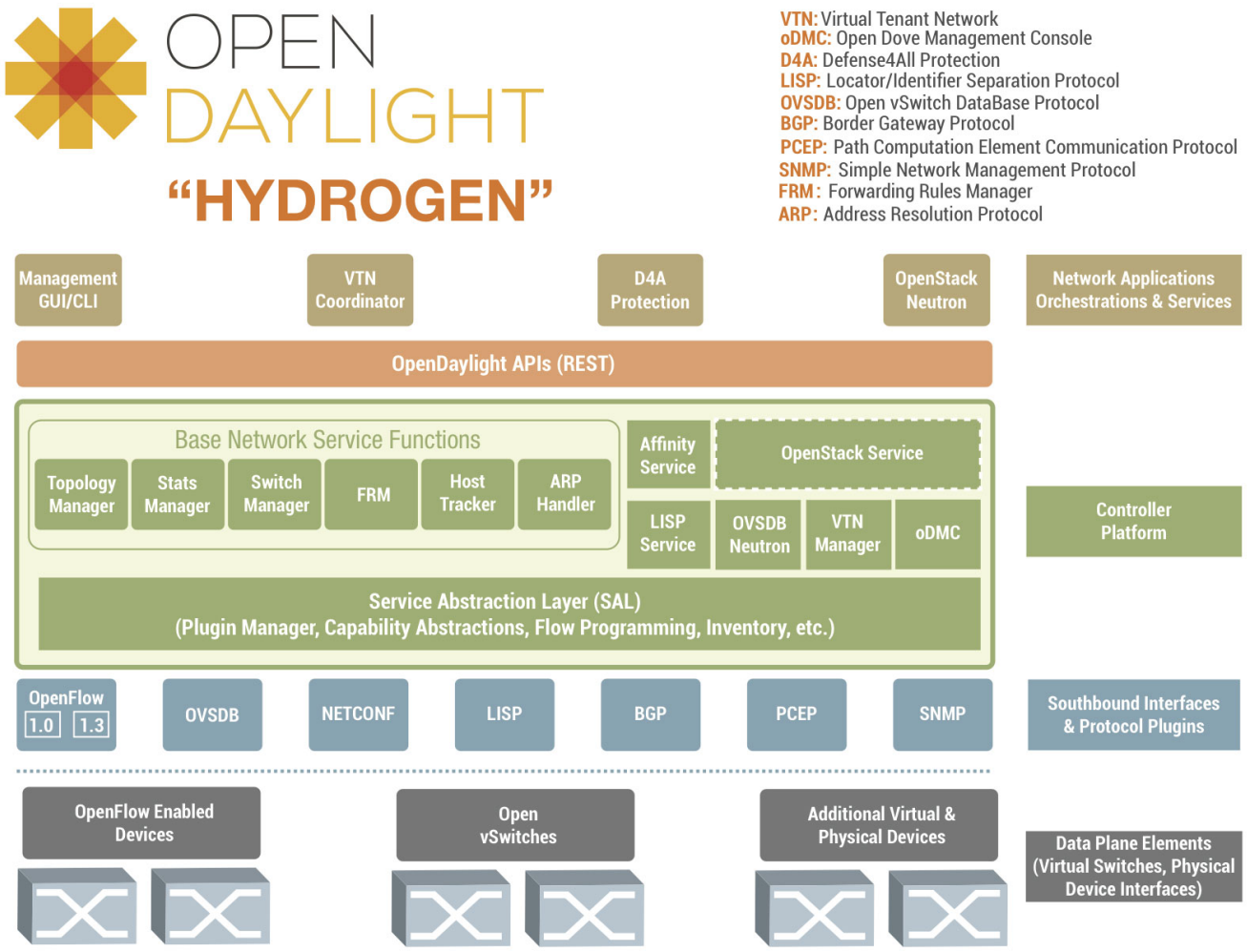
Controladoras SDN

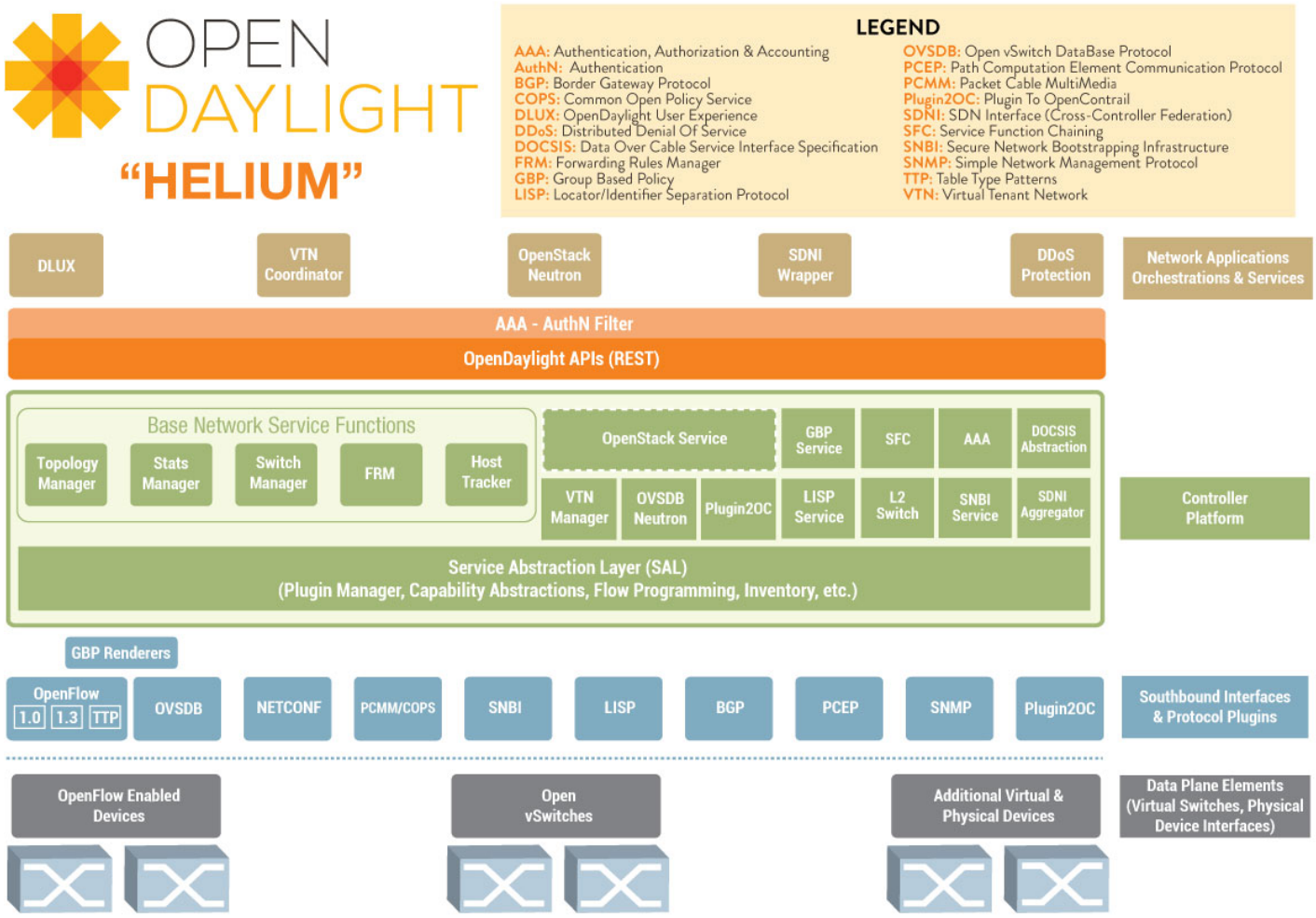
Controladoras SDN – OpenFlow

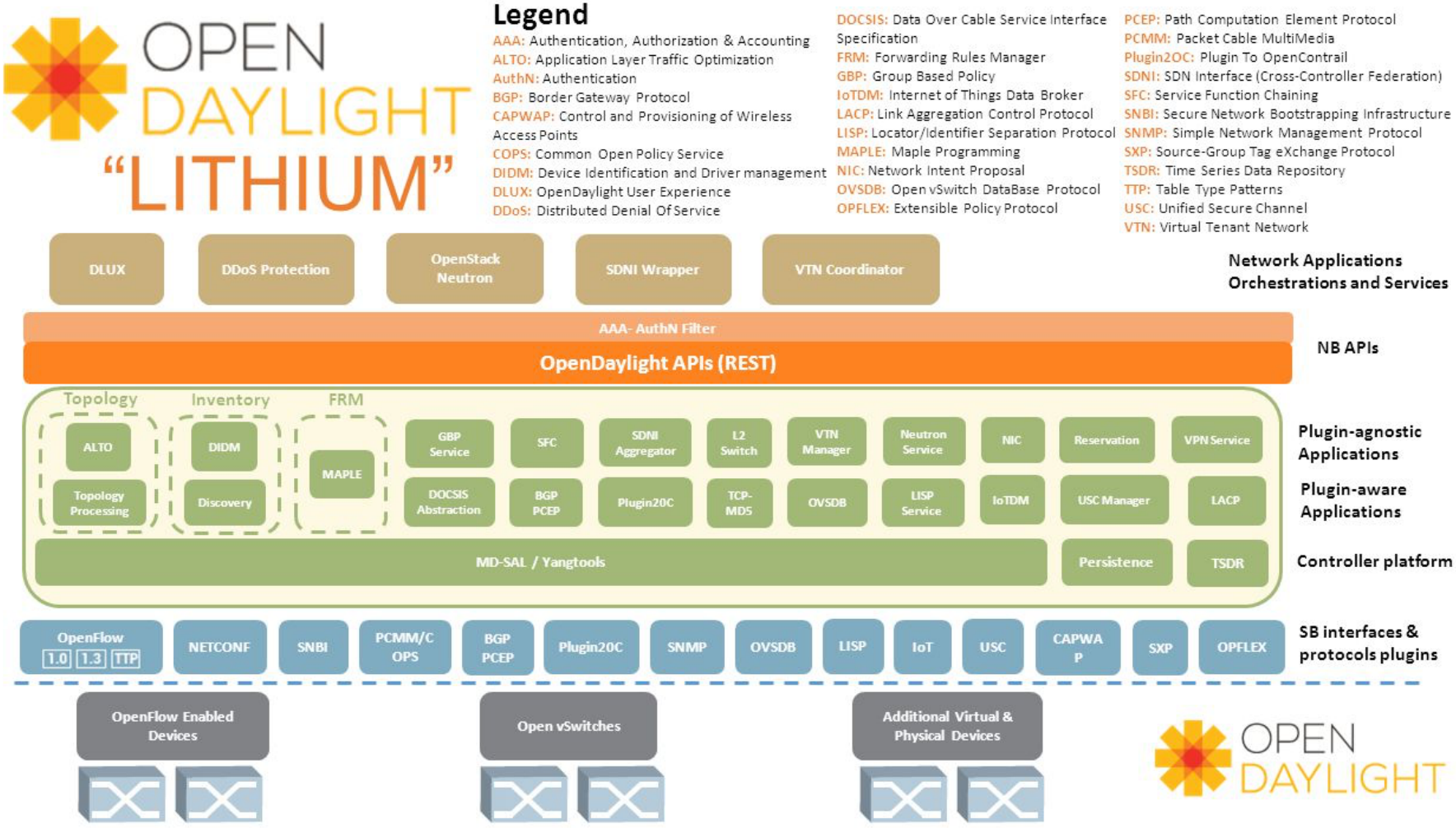
Inicitivas Open y comunidades que lo impulsan:	Iniciativas especificas de fabricantes:	Consideraciones:
• OpenDayLight • ONOS • NOX/POX • Project FloodLight • OpenContrail • Ryu • RouteFlow • Beacon • Etcétera.	• Brocade Vyatta Controller • Cisco Open SDN Controller • Juniper Contrail • Alcatel-Lucent Nuage • Etcétera.	• Lenguaje de programación • Rendimiento • Curva de aprendizaje • Base de usuarios y soporte • Enfoque: producción, educación e investigación.

- NOX
  - Fue la primera controladora OpenFlow.
  - Ha sido la base de muchos y diversos proyectos de investigación, a comienzos de la exploración de las redes definidas por software.
- POX
  - Es una versión de NOX en Python que proporciona funcionalidad dentro de un ambiente de prototipo rápido.

- Proyecto Open Source
  - Miembros Platinum como: Brocade, Cisco, Citrix, Dell, Ericsson, HP, Intel, Red Hat, entre otros
  - Incluye soporte de multiples protocolos (southbound)
    - OpenFlow, OVSDB, NetConf, BGP-LS, PCE, etcétera.
  - Northbound API: RestConf con YANG
  - Servicio central es un YANG.


OPEN
DAYLIGHT
"HYDROGEN"
VTN: Virtual Tenant Network
oDMC: Open Dove Management Console
D4A: Defense4All Protection
LISP: Locator/Identifier Separation Protocol
OVSDB: Open vSwitch DataBase Protocol
BGP: Border Gateway Protocol
PCEP: Path Computation Element Communication Protocol
SNMP: Simple Network Management Protocol
FRM: Forwarding Rules Manager
ARP: Address Resolution Protocol
Management GUI/CLI
VTN Coordinator
D4A Protection
OpenStack Neutron
Network Applications Orchestrations & Services
OpenDaylight APIs (REST)
Base Network Service Functions
Topology Manager
Stats Manager
Switch Manager
FRM
Host Tracker
ARP Handler
Affinity Service
OpenStack Service
LISP Service
OVSDB Neutron
VTN Manager
oDMC
Controller Platform
Service Abstraction Layer (SAL)
(Plugin Manager, Capability Abstractions, Flow Programming, Inventory, etc.)
OpenFlow 1.0 1.3
OVSDB
NETCONF
LISP
BGP
PCEP
SNMP
Southbound Interfaces & Protocol Plugins
OpenFlow Enabled Devices
Open vSwitches
Additional Virtual & Physical Devices
Data Plane Elements (Virtual Switches, Physical Device Interfaces)


OPEN DAYLIGHT
"HELIUM"
LEGEND
AAA: Authentication, Authorization & Accounting
AuthN: Authentication
BGP: Border Gateway Protocol
COPS: Common Open Policy Service
DLUX: OpenDaylight User Experience
DDoS: Distributed Denial Of Service
DOCSIS: Data Over Cable Service Interface Specification
FRM: Forwarding Rules Manager
GBP: Group Based Policy
LISP: Locator/Identifier Separation Protocol
OVSDB: Open vSwitch DataBase Protocol
PCEP: Path Computation Element Communication Protocol
PCMM: Packet Cable MultiMedia
Plugin2OC: Plugin To OpenContrail
SDNI: SDN Interface (Cross-Controller Federation)
SFC: Service Function Chaining
SNBI: Secure Network Bootstrapping Infrastructure
SNMP: Simple Network Management Protocol
TTP: Table Type Patterns
VTN: Virtual Tenant Network
DLUX
VTN Coordinator
OpenStack Neutron
SDNI Wrapper
DDoS Protection
Network Applications Orchestrations & Services
AAA - AuthN Filter
OpenDaylight APIs (REST)
Base Network Service Functions
Topology Manager
Stats Manager
Switch Manager
FRM
Host Tracker
OpenStack Service
GBP Service
SFC
AAA
DOCSIS Abstraction
VTN Manager
OVSDB Neutron
Plugin2OC
LISP Service
L2 Switch
SNBI Service
SDNI Aggregator
Service Abstraction Layer (SAL)
(Plugin Manager, Capability Abstractions, Flow Programming, Inventory, etc.)
Controller Platform
GBP Renderers
OpenFlow 1.0 1.3 TTP
OVSDB
NETCONF
PCMM/COPS
SNBI
LISP
BGP
PCEP
SNMP
Plugin2OC
Southbound Interfaces & Protocol Plugins
OpenFlow Enabled Devices
Open vSwitches
Additional Virtual & Physical Devices
Data Plane Elements (Virtual Switches, Physical Device Interfaces)


OPEN DAYLIGHT
"LITHIUM"
Legend
AAA: Authentication, Authorization & Accounting
ALTO: Application Layer Traffic Optimization
AuthN: Authentication
BGP: Border Gateway Protocol
CAPWAP: Control and Provisioning of Wireless Access Points
COPS: Common Open Policy Service
DIDM: Device Identification and Driver management
DLUX: OpenDaylight User Experience
DDoS: Distributed Denial Of Service
DOCSIS: Data Over Cable Service Interface Specification
FRM: Forwarding Rules Manager
GBP: Group Based Policy
IoTDM: Internet of Things Data Broker
LACP: Link Aggregation Control Protocol
LISP: Locator/Identifier Separation Protocol
MAPLE: Maple Programming
NIC: Network Intent Proposal
OVSDB: Open vSwitch DataBase Protocol
OPFLEX: Extensible Policy Protocol
PCEP: Path Computation Element Protocol
PCMM: Packet Cable MultiMedia
Plugin2OC: Plugin To OpenContrail
SDNI: SDN Interface (Cross-Controller Federation)
SFC: Service Function Chaining
SNBI: Secure Network Bootstrapping Infrastructure
SNMP: Simple Network Management Protocol
SXP: Source-Group Tag eXchange Protocol
TSDR: Time Series Data Repository
TTP: Table Type Patterns
USC: Unified Secure Channel
VTN: Virtual Tenant Network
DLUX
DDoS Protection
OpenStack Neutron
SDNI Wrapper
VTN Coordinator
Network Applications Orchestrations and Services
AAA- AuthN Filter
OpenDaylight APIs (REST)
NB APIs
Topology
Inventory
FRM
ALTO
Topology Processing
DIDM
Discovery
MAPLE
GBP Service
SFC
SDNI Aggregator
L2 Switch
VTN Manager
Neutron Service
NIC
Reservation
VPN Service
DOCSIS Abstraction
BGP PCEP
Plugin20C
TCP-MD5
OVSDB
LISP Service
IoTDM
USC Manager
LACP
Plugin-agnostic Applications
Plugin-aware Applications
MD-SAL / Yangtools
Persistence
TSDR
Controller platform
OpenFlow 1.0 1.3 TTP
NETCONF
SNBI
PCMM/COPS
BGP PCEP
Plugin20C
SNMP
OVSDB
LISP
IoT
USC
CAPWAP
SXP
OPFLEX
SB interfaces & protocols plugins
OpenFlow Enabled Devices
Open vSwitches
Additional Virtual & Physical Devices
OPEN DAYLIGHT

- El proyecto ONOS esta asociado con el Linux Foundation.
  - Tiene como misión construir la mejor comunidad de código abierto para acelarar la adopción de SDN+NFV.
  - Incluye conectores (plug-in) hacia diversos dispositivos, como:
    - OpenFlow, NetConf, Southbound Interfaces.
  - En la parte norte cuenta con módulos de control, administración y configuración de aplicaciones.
    - A través de una interfaz de programación, intuitiva, flexible.

Resumen las controladoras

	NOX	POX	ODL	Floodlight	Ryu
Lenguaje	C++	Python	JAVA	JAVA	Python
Rendimiento	Rápida	Lenta	Rápida	Rápida	Lenta
Versión de OpenFlow	1.0 / 1.3	1.0	1.0 / 1.3	1.0	1.0 a 1.4
Distribuida	no	no	sí	sí	sí
Curva de Aprendizaje	moderada	fácil	compleja	compleja	moderada
Nota		Investigación, experimentación y demostración	Soporte de aplicaciones de fabricantes		Controladora Python Open source

Switches OpenFlow

Ambiente del laboratorio 1

- Descargar los archivos de forma individual.
- Los archivos incluyen software de virtualización, una terminal SSH, un X server, y la imagen VM.
- La imagen del tutorial se distribuye como una imagen de VirtualBox comprimido (o VDI).
- **VirtualBox** permite ejecutar una máquina virtual dentro de una máquina física, es gratuito y está disponible para Windows, Mac y Linux. Nota: se puede exportar la imagen VirtualBox a formato vmdk y utilizarlo con VMWare.
- Las siguientes instrucciones asumen el uso de VirtualBox, pero las instrucciones pueden aplicarse independientemente del software de virtualización.

Requerimientos

- Descargar los archivos correspondiente al sistema operativo, más el tutorial de la maquina virtual (VM):
  - Virtual Machine Image (OVF, 64bits, **Mininet 2.2.0**)
  - Virtual Machine Image (OVF, 32bits, **Mininet 2.2.0**) recomendado para hardware y Windows anteriores.
- Para la VM el usuario es “mininet” y contraseña “mininet”.

https://github.com/mininet/mininet/wiki/Mininet-VM-Images

Requerimientos

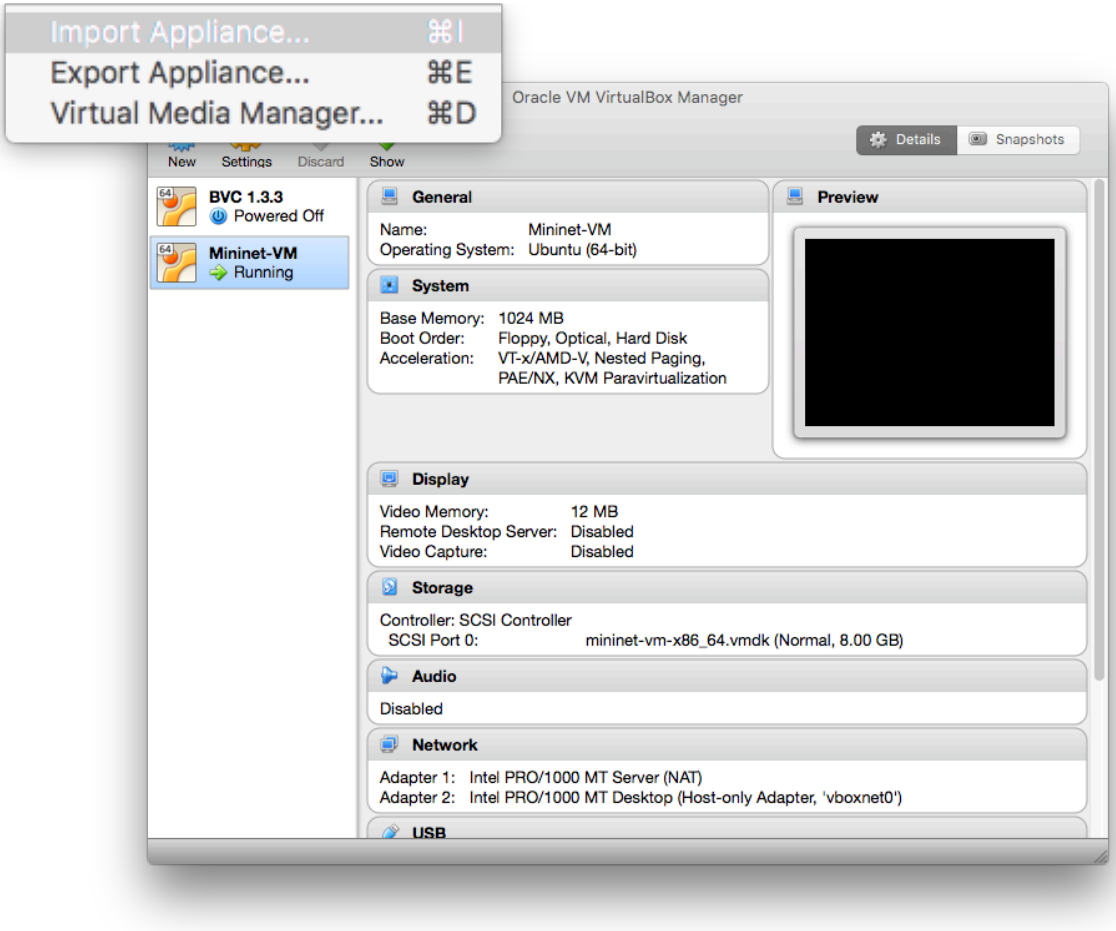
- Es necesario el software de virtualización, un X server y una terminal SSH:

Tipo de S.O.	Versión de S.O.	Software de virtualización	X Server	Terminal
Windows	7 o superior	VirtualBox	Xming	PuTTy
Windows	XP	VirtualBox	Xming	PuTTY
Mac	OS X 10.7 o superior	VirtualBox	Descarga e instala XQuartz	Terminal.app (lo incluye el S.O.)
Linux	Ubuntu 10.04 o superior	VirtualBox	El X server ya esta instalado por defecto	Gnome terminal + SSH

Topología de laboratorio 1

Importar Máquina Virtual

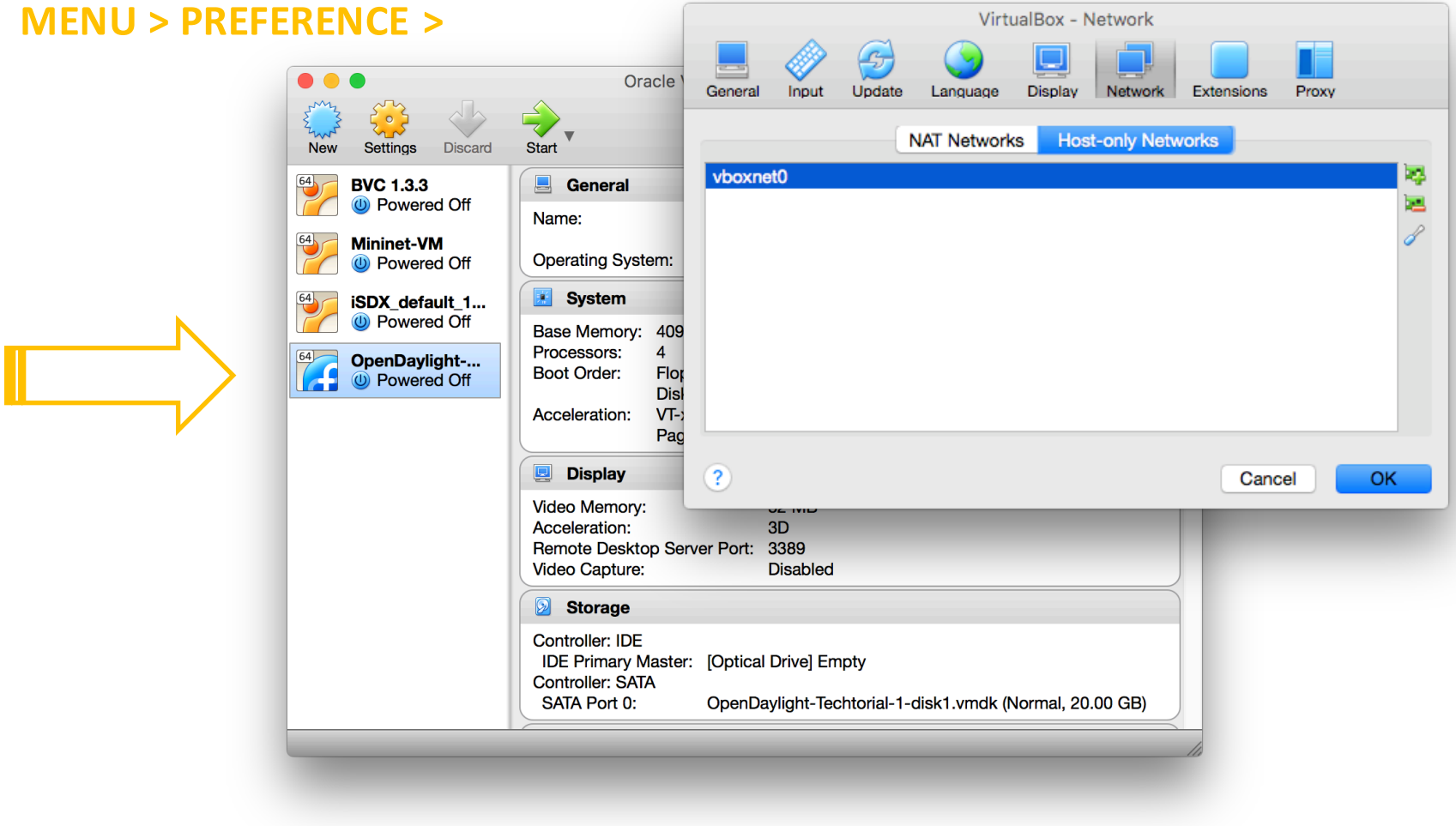
MENU > FILE >



- **Import**
- **Finish VM setup**

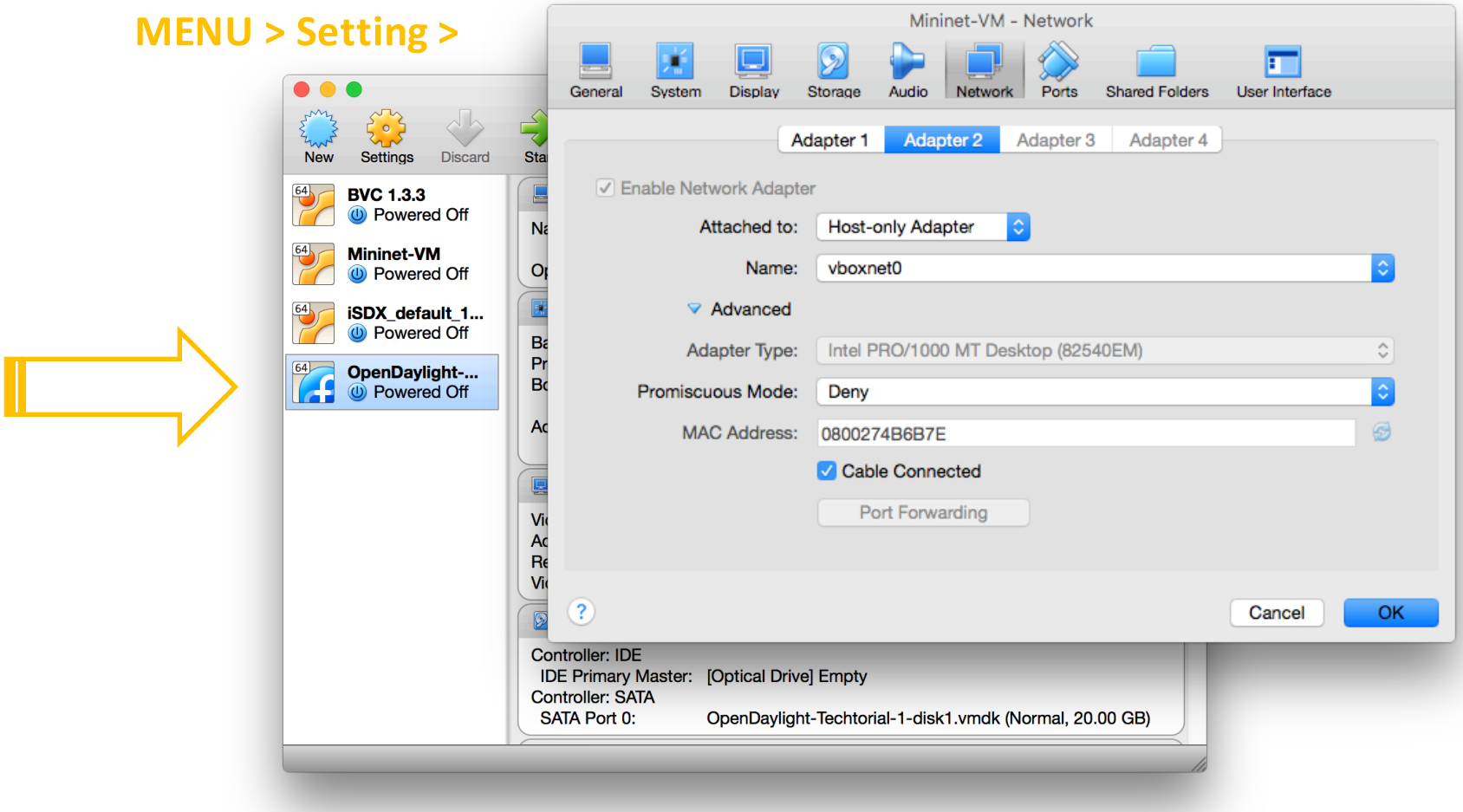
Finalizar Máquina Virtual

MENU > PREFERENCE >



Finalizar Máquina Virtual

MENU > Setting >



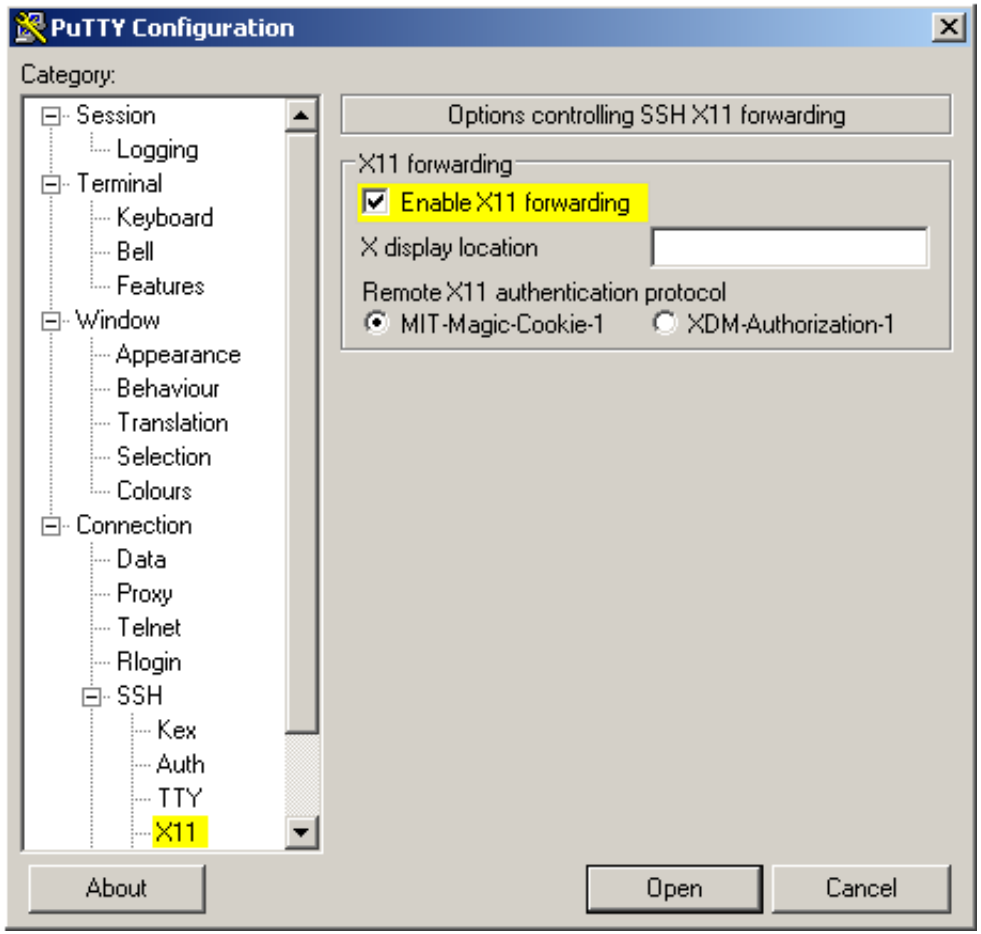

Desde MAC OSX, Windows, Linux
olmos — -bash — 80×24
bridge0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
options=63<RXCSUM,TXCSUM,TSO4,TSO6>
ether ae:87:a3:11:4a:00
Configuration:
id 0:0:0:0:0:0 priority 0 hellotime 0 fwddelay 0
maxage 0 holdcnt 0 proto stp maxaddr 100 timeout 1200
root id 0:0:0:0:0:0 priority 0 ifcost 0 port 0
ipfilter disabled flags 0x2
member: en1 flags=3<LEARNING,DISCOVER>
ifmaxaddr 0 port 6 priority 0 path cost 0
member: en2 flags=3<LEARNING,DISCOVER>
ifmaxaddr 0 port 7 priority 0 path cost 0
nd6 options=1<PERFORMNUD>
media: <unknown type>
status: inactive
vboxnet0: flags=8943<UP,BROADCAST,RUNNING,PROMISC,SIMPLEX,MULTICAST> mtu 1500
ether 0a:00:27:00:00:00
inet 192.168.56.1 netmask 0xffffff00 broadcast 192.168.56.255
vboxnet1: flags=8842<BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
ether 0a:00:27:00:00:01
utun0: flags=8051<UP,POINTOPOINT,RUNNING,MULTICAST> mtu 1280
inet 148.202.221.181 --> 148.202.221.181 netmask 0xffffff00
nd6 options=1<PERFORMNUD>
Jaimes-MacBook-Pro:~ olmos$
File Edit View
[mininet@opendaylight ~]\>
mininet : bash
ifconfig -a
ipconfig /all

Mac OS X - Linux

- Abrir una terminal (Terminal.app en Mac, Gnome terminal en Ubuntu), dentro de la terminal, ejecutar el siguiente comando:
  - $ ssh –X [usuario]@[dirección IP]
  - Remplace el [usuario] con el correcto nombre de la VM: mininet
  - Remplace la [dirección IP] con la otorgado por DHCP a la VM. Si SSH no conecta, asegure que conteste vía ping.
- Ingresar el password: mininet. Siguiente, arrancar una X terminal usando:
  - $ xterm

Windows

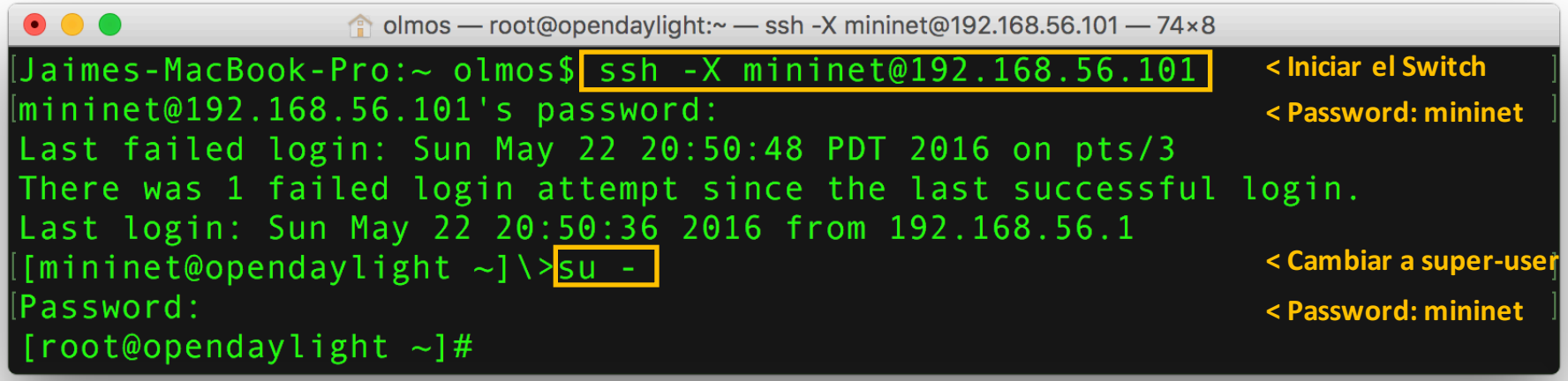
- Para hacer uso de aplicaciones X11 como xterm y wireshark, debe ejecutar el servidor Xming (X server) y una conexión SSH (PuTTY) habilitada con X11 forwarding:



Windows

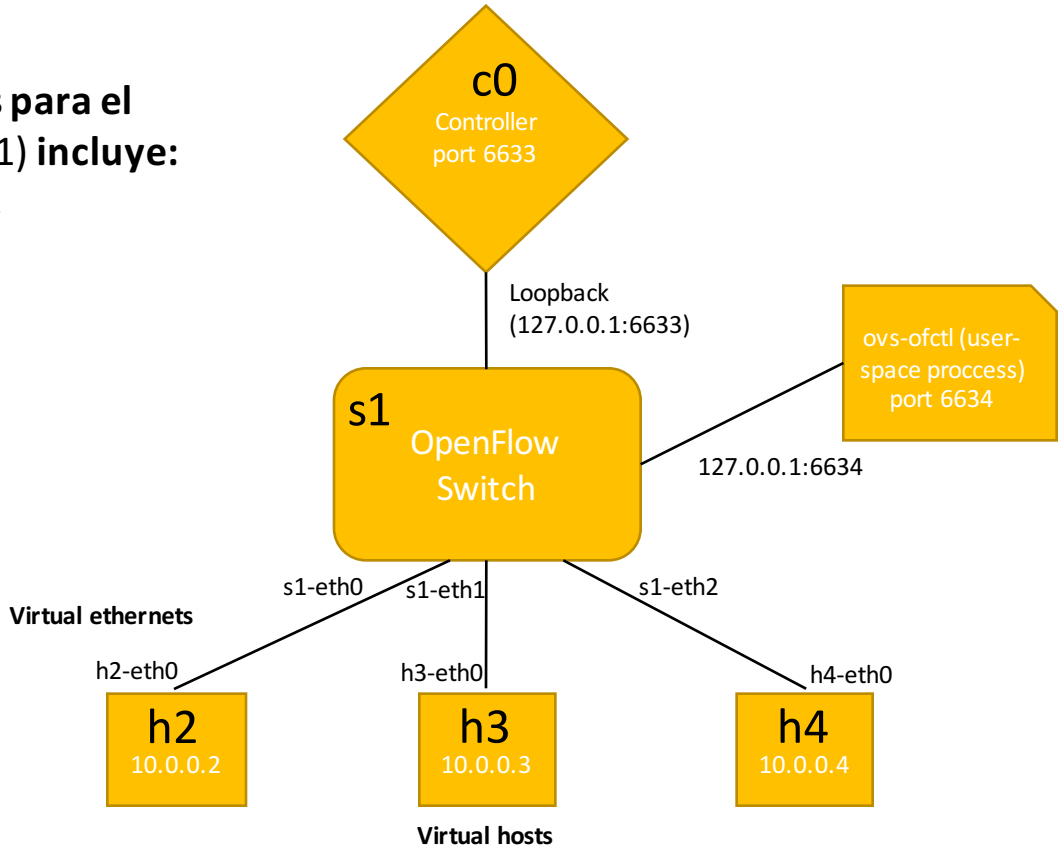
- Ejecurtar una terminal (PuTTy), ejecutar los siguientes comandos:
  - C:\ cd <directorio de PuTTy>
  - C:\ putty.exe –X [usuario]@[dirección IP]
  - Remplace el [usuario] con el correcto nombre de la VM: mininet
  - Remplace la [dirección IP] con la otorgado por DHCP a la VM. Si SSH no conecta, asegure que conteste vía ping.
- Ingresar el contraseña: mininet. Siguiente, arrancar una X terminal usando:
  - $ xterm –sb 500
  - El –sb 500 es opcional y sirve para permitir 500 lineas en la ventana.

Entrar a MININET



Topología Virtual

La red que usaras para el primer ejercicio (1) **incluye:**
3 hosts, 1 switch y 1 controladora

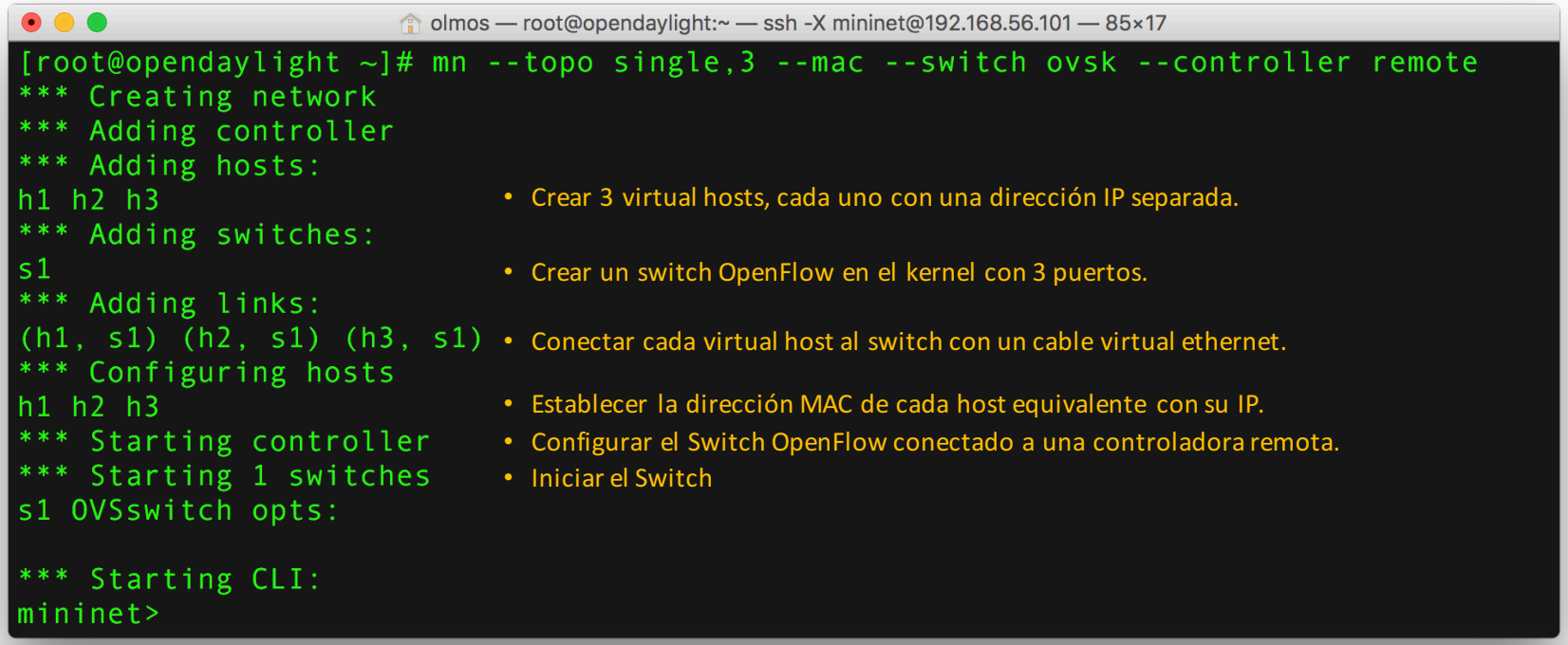


$ sudo mn --topo single,3 --mac --switch ovsk --controller remote

Crear una red en MININET

- **`$ sudo mn --topo single,3 --mac --switch ovsk --controller remote`**
- Este comando indica a MiniNet levantar 3 hosts, un switch (basado en OpenVirtualSwitch) de topología simple, configurar una dirección MAC e IPv4 para cada hosts y una Controlador OpenFlow por defecto como localhost.

¿Qué hizo Mininet?



- Listar los nodos disponibles (hosts) desde la consola de MiniNet, comando:
  mininet> nodes
- Listar los comandos disponibles en MiniNet: help

olmos — root@opendaylight:~ — ssh -X mininet@192.168.56.101 — 80×29

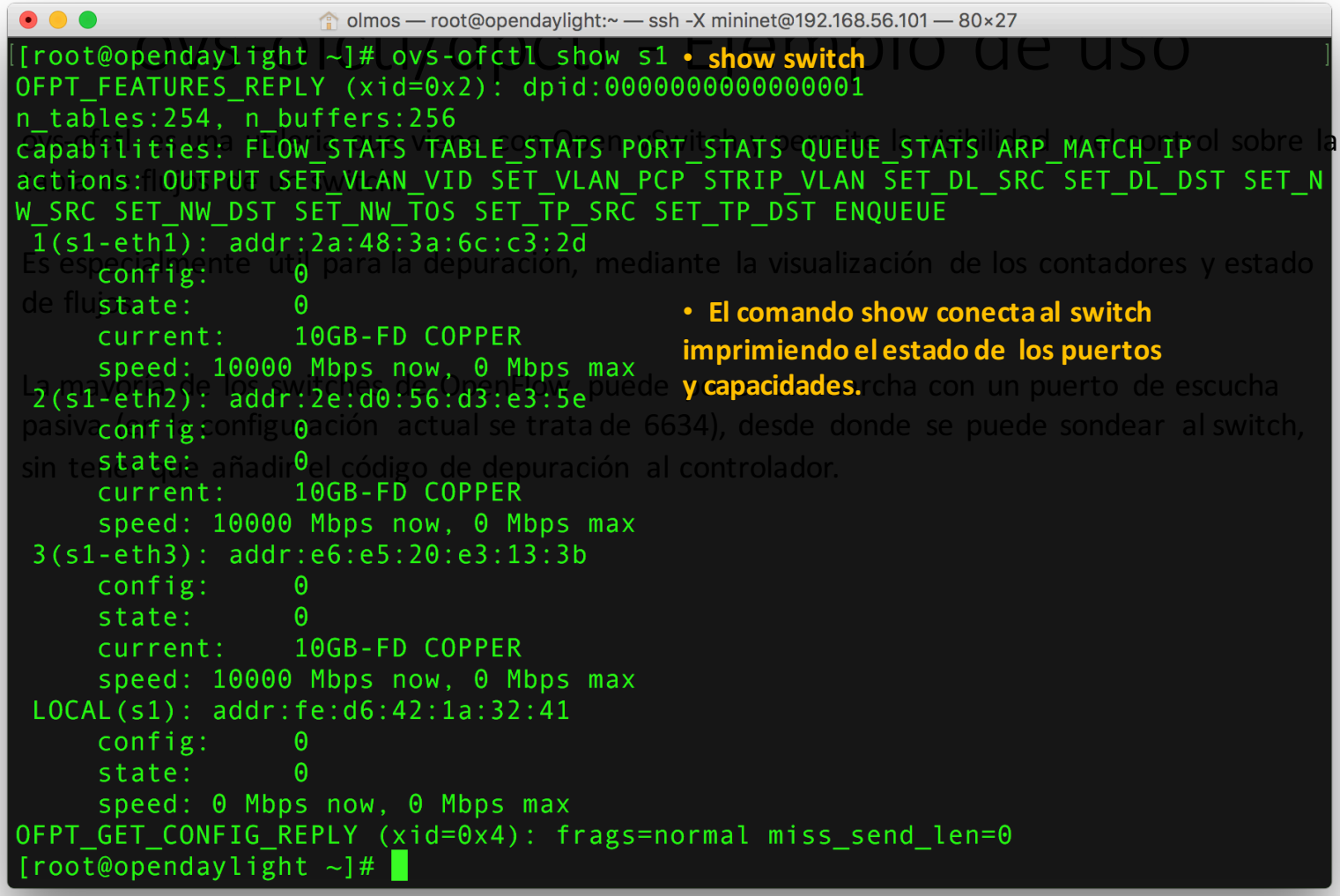
```
mininet> nodes
available nodes are:
c0 h1 h2 h3 s1
mininet> help
Documented commands (type help <topic>):
========================================
EOF    exit   intfs     link   noecho       pingpair      py    source  xterm
dpctl  gterm  iperf     net    pingall      pingpairfull  quit  time
dump   help   iperfudp  nodes  pingallfull  px            sh    x

You may also send a command to a node using:
  <node> command {args}
For example:
  mininet> h1 ifconfig
The interpreter automatically substitutes IP addresses
for node names when a node is the first arg, so commands
like
  mininet> h2 ping h3
should work.

Some character-oriented interactive commands require
noecho:
  mininet> noecho h2 vi foo.py
However, starting up an xterm/gterm is generally better:
  mininet> xterm h2

mininet>
```

Listar la configuración de red del host virtual

Ejecutar un ping entre hosts virtuales

Editar archivos en host virtual

Emulador de terminal para el sistema de ventanas X (host virtual)


olmos — root@opendaylight:~ — ssh -X mininet@192.168.56.101 — 80×27
[root@opendaylight ~]# ovs-ofctl show s1
OFPT_FEATURES_REPLY (xid=0x2): dpid:0000000000000001
n_tables:254, n_buffers:256
capabilities: FLOW_STATS TABLE_STATS PORT_STATS QUEUE_STATS ARP_MATCH_IP
actions: OUTPUT SET_VLAN_VID SET_VLAN_PCP STRIP_VLAN SET_DL_SRC SET_DL_DST SET_NW_SRC SET_NW_DST SET_NW_TOS SET_TP_SRC SET_TP_DST ENQUEUE
1(s1-eth1): addr:2a:48:3a:6c:c3:2d
config: 0
state: 0
current: 10GB-FD COPPER
speed: 10000 Mbps now, 0 Mbps max
2(s1-eth2): addr:2e:d0:56:d3:e3:5e
config: 0
state: 0
current: 10GB-FD COPPER
speed: 10000 Mbps now, 0 Mbps max
3(s1-eth3): addr:e6:e5:20:e3:13:3b
config: 0
state: 0
current: 10GB-FD COPPER
speed: 10000 Mbps now, 0 Mbps max
LOCAL(s1): addr:fe:d6:42:1a:32:41
config: 0
state: 0
speed: 0 Mbps now, 0 Mbps max
OFPT_GET_CONFIG_REPLY (xid=0x4): frags=normal miss_send_len=0
[root@opendaylight ~]#
• show switch
• El comando show conecta al switch imprimiendo el estado de los puertos y capacidades.

Prueba de ping

- Ahora, vuelve a la consola MiniNet y tratar de ping desde h1 a h2. En la consola MiniNet:

olmos — root@opendaylight:~ — ssh -X mininet@192.168.56.101 — 77×12

```
mininet> h1 ping -c5 h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
From 10.0.0.1 icmp_seq=1 Destination Host Unreachable
From 10.0.0.1 icmp_seq=2 Destination Host Unreachable
From 10.0.0.1 icmp_seq=3 Destination Host Unreachable
From 10.0.0.1 icmp_seq=4 Destination Host Unreachable
From 10.0.0.1 icmp_seq=5 Destination Host Unreachable

--- 10.0.0.2 ping statistics ---
5 packets transmitted, 0 received, +5 errors, 100% packet loss, time 4001ms
pipe 4
mininet>
```

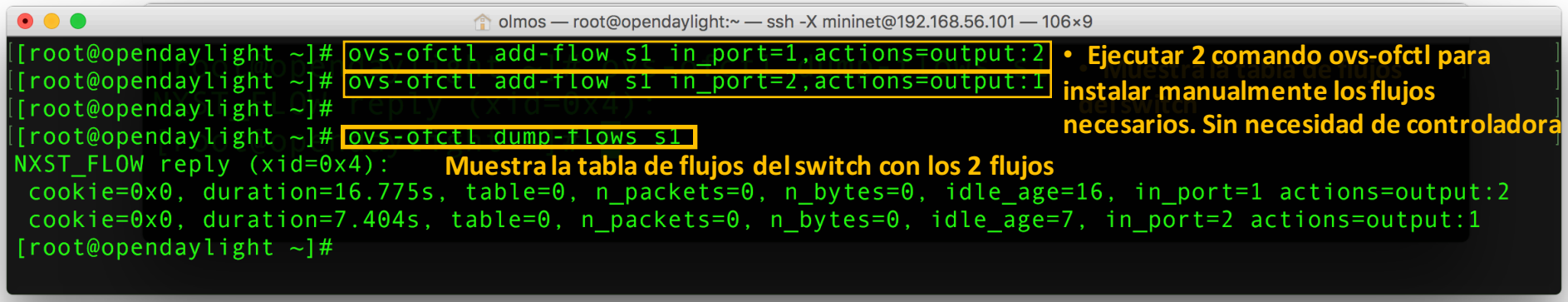
- **Ping desde h1 a h2**

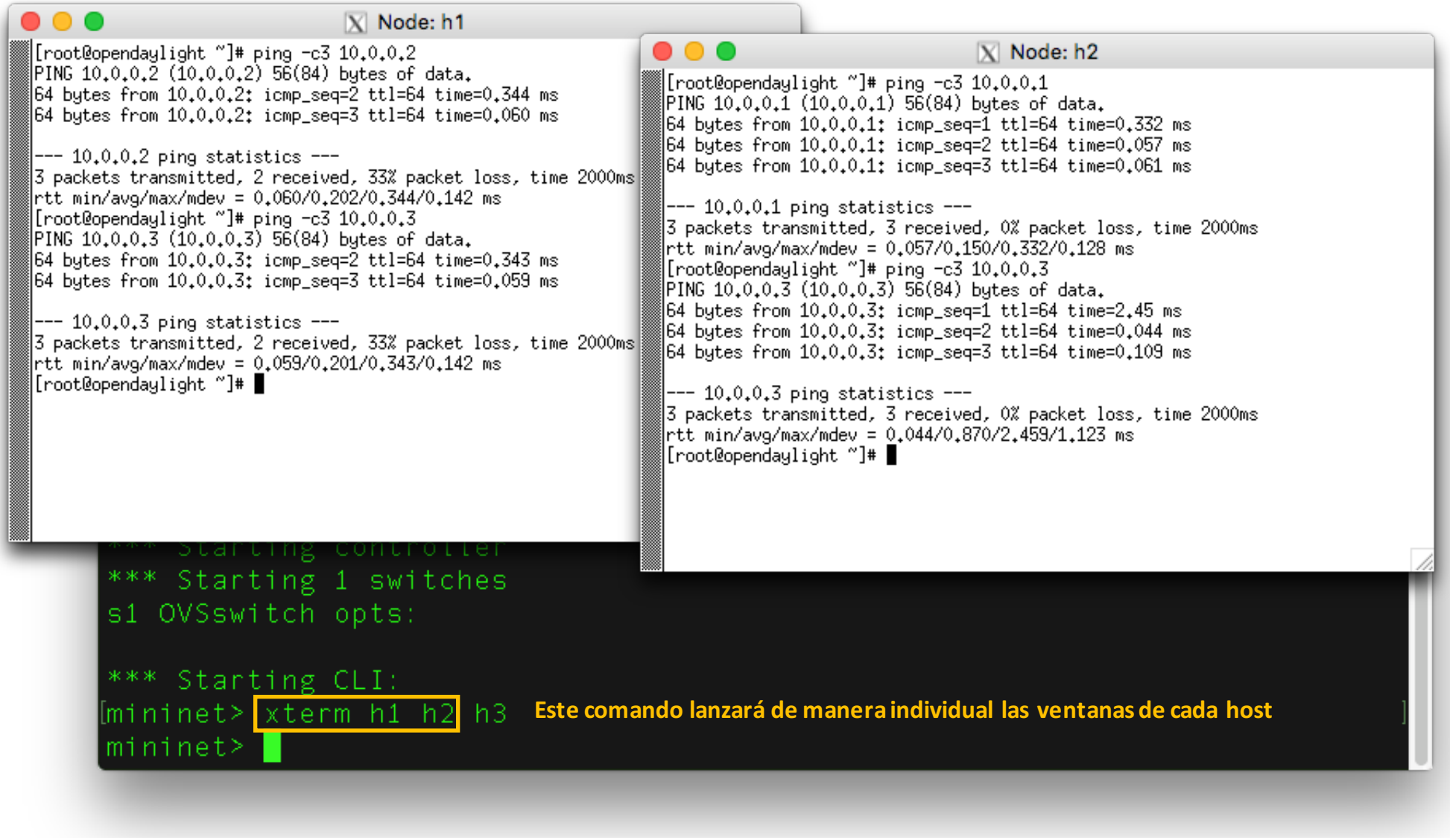
No hay respuesta.

¿Por qué no obtiene alguna respuesta? ¿Qué falta?

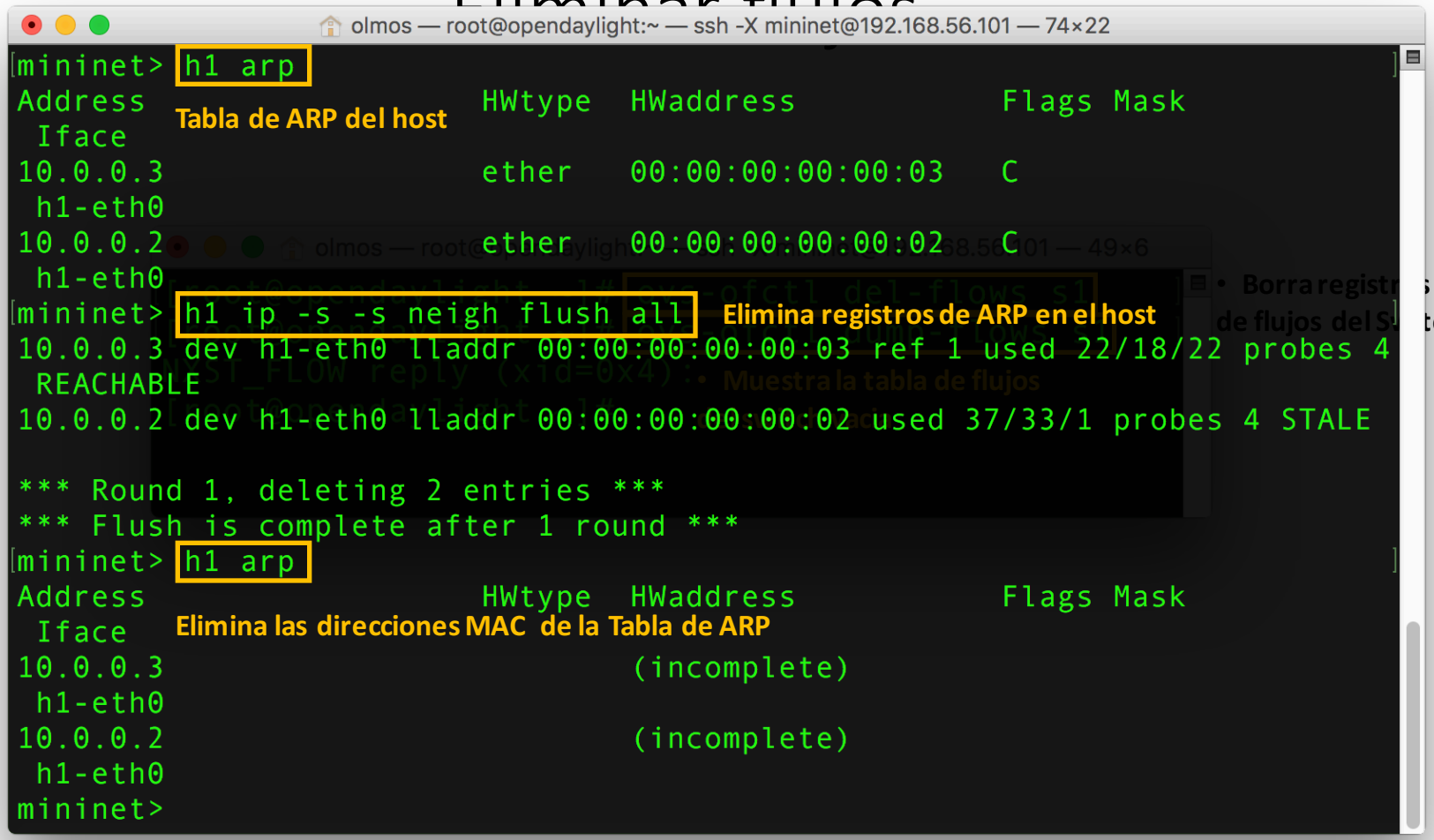
Prueba de ping

- Puesto que no hemos empezado con algún controlador aún, la tabla de flujos debe estar vacía.




Node: h1
[root@opendaylight ~]# ping -c3 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.344 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.060 ms
--- 10.0.0.2 ping statistics ---
3 packets transmitted, 2 received, 33% packet loss, time 2000ms
rtt min/avg/max/mdev = 0.060/0.202/0.344/0.142 ms
[root@opendaylight ~]# ping -c3 10.0.0.3
PING 10.0.0.3 (10.0.0.3) 56(84) bytes of data.
64 bytes from 10.0.0.3: icmp_seq=2 ttl=64 time=0.343 ms
64 bytes from 10.0.0.3: icmp_seq=3 ttl=64 time=0.059 ms
--- 10.0.0.3 ping statistics ---
3 packets transmitted, 2 received, 33% packet loss, time 2000ms
rtt min/avg/max/mdev = 0.059/0.201/0.343/0.142 ms
[root@opendaylight ~]#
Node: h2
[root@opendaylight ~]# ping -c3 10.0.0.1
PING 10.0.0.1 (10.0.0.1) 56(84) bytes of data.
64 bytes from 10.0.0.1: icmp_seq=1 ttl=64 time=0.332 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=0.057 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=0.061 ms
--- 10.0.0.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2000ms
rtt min/avg/max/mdev = 0.057/0.150/0.332/0.128 ms
[root@opendaylight ~]# ping -c3 10.0.0.3
PING 10.0.0.3 (10.0.0.3) 56(84) bytes of data.
64 bytes from 10.0.0.3: icmp_seq=1 ttl=64 time=2.45 ms
64 bytes from 10.0.0.3: icmp_seq=2 ttl=64 time=0.044 ms
64 bytes from 10.0.0.3: icmp_seq=3 ttl=64 time=0.109 ms
--- 10.0.0.3 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2000ms
rtt min/avg/max/mdev = 0.044/0.870/2.459/1.123 ms
[root@opendaylight ~]#
*** Starting 1 switches
s1 OVSswitch opts:
*** Starting CLI:
mininet> xterm h1 h2 h3
mininet>
Este comando lanzará de manera individual las ventanas de cada host

Eliminar flujos



en la tabla
tch

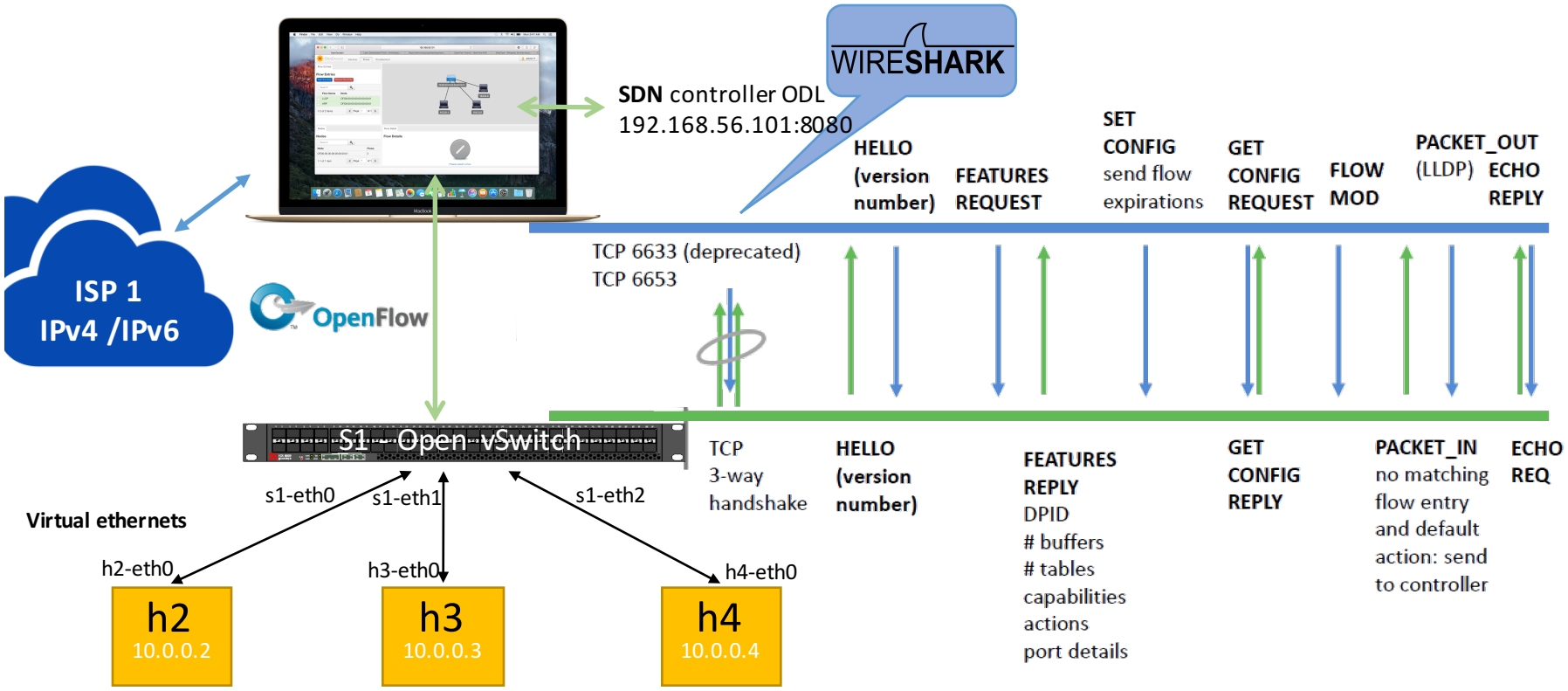
Ambiente del laboratorio 2

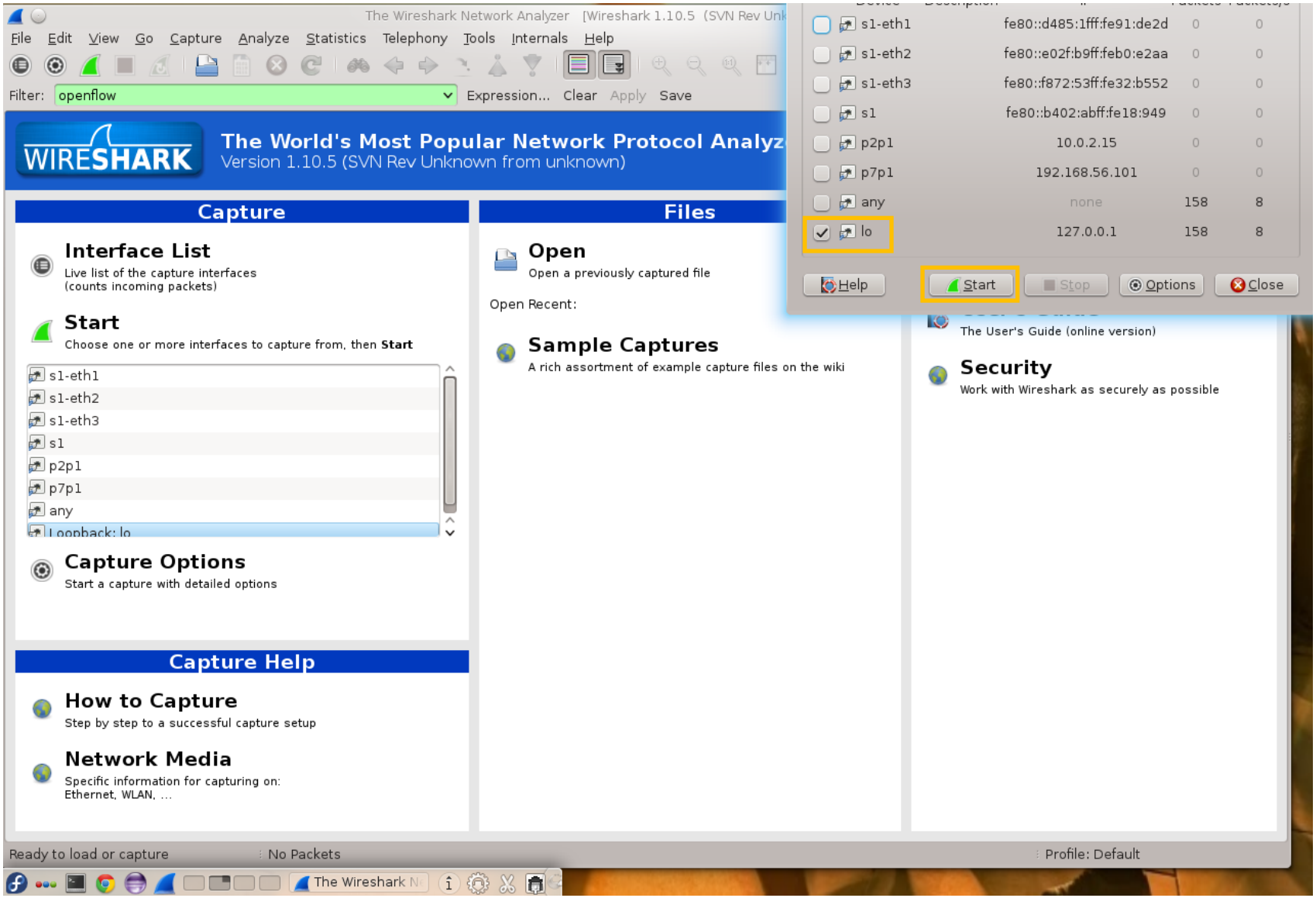
- Software de virtualización, una terminal SSH, un X server, y la imagen VM.
- Sniffer Wireshark.
- **OpenDaylight** Hydrogen Base 1.0.

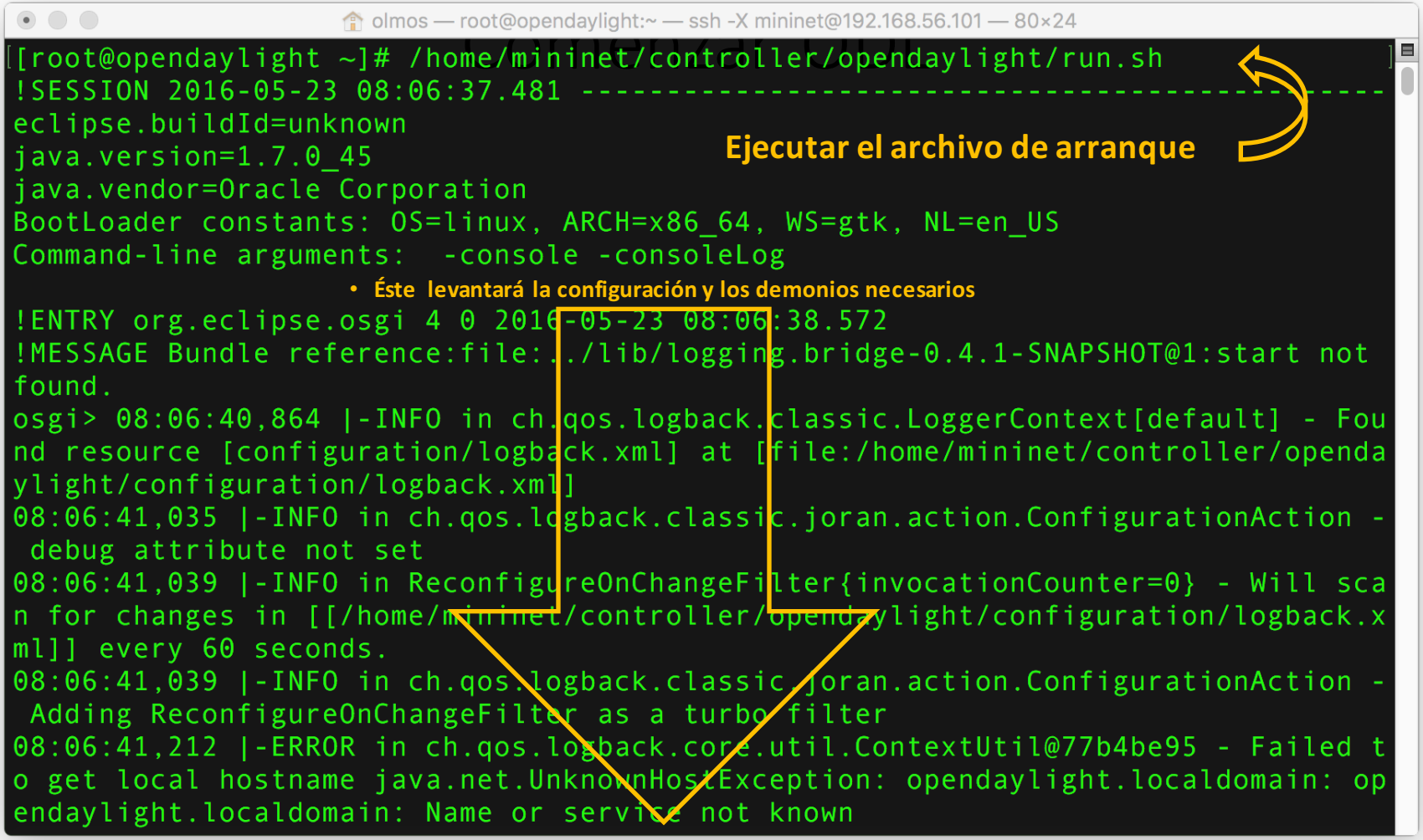
Protocolo OpenFlow

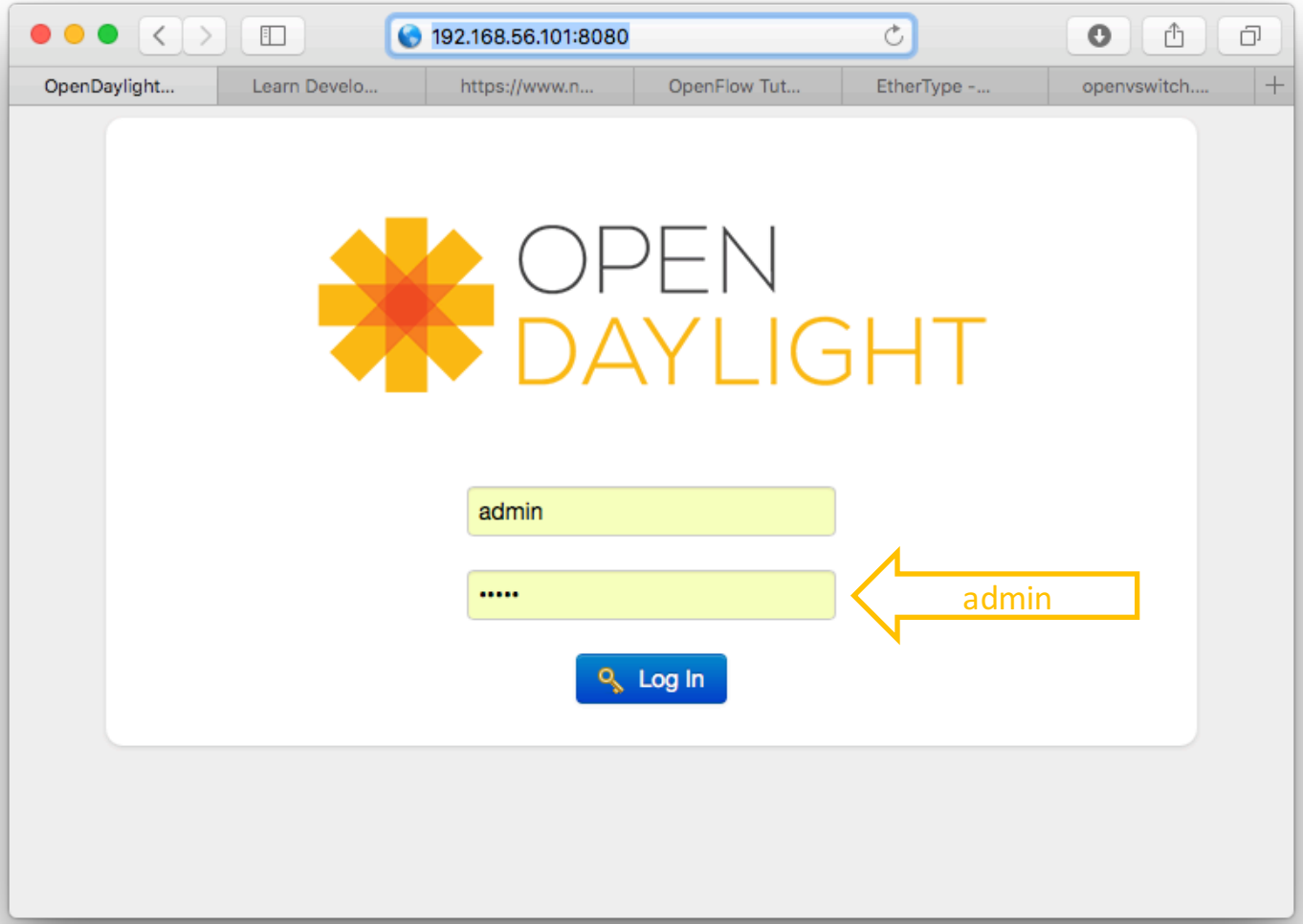
- El protocolo OpenFlow soporta tres tipos de mensajes:
  - Controladora-a-Switch - iniciado por la controladora y puede no requerir una respuesta.
  - Asíncrono – el switch pueden envíar a la controladora sin que la controladora lo haya solicitando.
  - Simétrico - enviado por cualquiera, las controladoras o el switch sin ser solicitada.
- **Controlador-a-Switch**
  - Solicitud de característica
  - Configuración
  - Modificar el estado – utilizado para agregar / eliminar / modificar los flujos en el switch
- **Asíncrono**
  - Trama-entrada
  - Retira flujo
- **Simétrico**
  - Hello
  - Echo

Conexión Switch OF

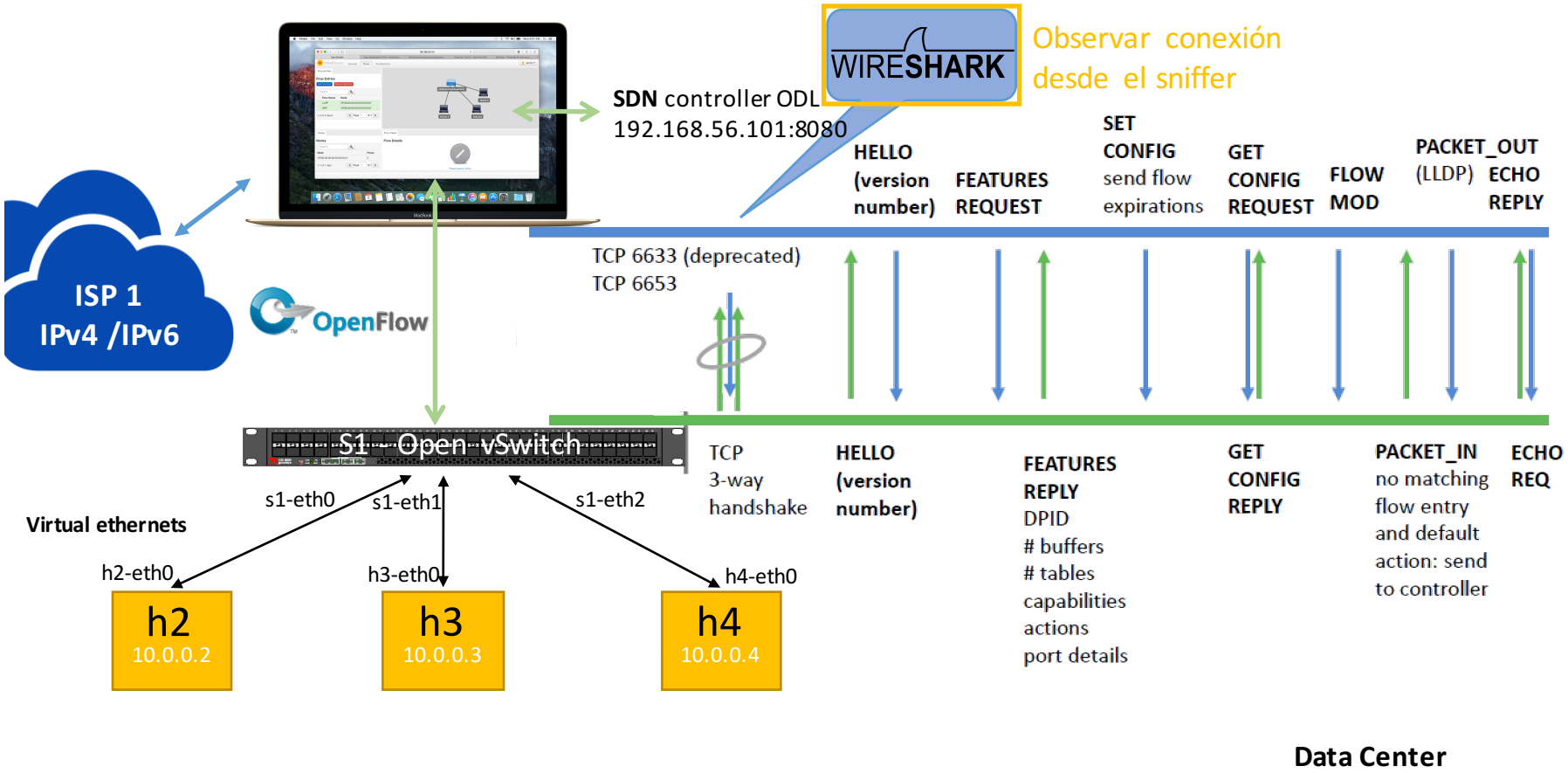



The Wireshark Network Analyzer [Wireshark 1.10.5 (SVN Rev Unk
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter: openflow
Expression... Clear Apply Save
WIRESHARK
The World's Most Popular Network Protocol Analyz
Version 1.10.5 (SVN Rev Unknown from unknown)
Capture
Interface List
Live list of the capture interfaces
(counts incoming packets)
Start
Choose one or more interfaces to capture from, then Start
s1-eth1
s1-eth2
s1-eth3
s1
p2p1
p7p1
any
Loopback: lo
Capture Options
Start a capture with detailed options
Capture Help
How to Capture
Step by step to a successful capture setup
Network Media
Specific information for capturing on:
Ethernet, WLAN, ...
Files
Open
Open a previously captured file
Open Recent:
Sample Captures
A rich assortment of example capture files on the wiki
The User's Guide (online version)
Security
Work with Wireshark as securely as possible
s1-eth1 fe80::d485:1fff:fe91:de2d 0 0
s1-eth2 fe80::e02f:b9ff:feb0:e2aa 0 0
s1-eth3 fe80::f872:53ff:fe32:b552 0 0
s1 fe80::b402:abff:fe18:949 0 0
p2p1 10.0.2.15 0 0
p7p1 192.168.56.101 0 0
any none 158 8
lo 127.0.0.1 158 8
Help
Start
Stop
Options
Close
Ready to load or capture
No Packets
Profile: Default
The Wireshark N


olmos — root@opendaylight:~ — ssh -X mininet@192.168.56.101 — 80×24
[root@opendaylight ~]# /home/mininet/controller/opendaylight/run.sh
!SESSION 2016-05-23 08:06:37.481 ----------
eclipse.buildId=unknown
java.version=1.7.0_45
java.vendor=Oracle Corporation
BootLoader constants: OS=linux, ARCH=x86_64, WS=gtk, NL=en_US
Command-line arguments: -console -consoleLog
!ENTRY org.eclipse.osgi 4 0 2016-05-23 08:06:38.572
!MESSAGE Bundle reference:file:../lib/logging.bridge-0.4.1-SNAPSHOT@1:start not found.
osgi> 08:06:40,864 |-INFO in ch.qos.logback.classic.LoggerContext[default] - Found resource [configuration/logback.xml] at [file:/home/mininet/controller/opendaylight/configuration/logback.xml]
08:06:41,035 |-INFO in ch.qos.logback.classic.joran.action.ConfigurationAction - debug attribute not set
08:06:41,039 |-INFO in ReconfigureOnChangeFilter{invocationCounter=0} - Will scan for changes in [[/home/mininet/controller/opendaylight/configuration/logback.xml]] every 60 seconds.
08:06:41,039 |-INFO in ch.qos.logback.classic.joran.action.ConfigurationAction - Adding ReconfigureOnChangeFilter as a turbo filter
08:06:41,212 |-ERROR in ch.qos.logback.core.util.ContextUtil@77b4be95 - Failed to get local hostname java.net.UnknownHostException: opendaylight.localdomain: opendaylight.localdomain: Name or service not known
Ejecutar el archivo de arranque
• Éste levantará la configuración y los demonios necesarios


192.168.56.101:8080
OpenDaylight...
Learn Develo...
https://www.n...
OpenFlow Tut...
EtherType -...
openvswitch....
OPEN
DAYLIGHT
admin
admin
Log In

Conexión Switch OF



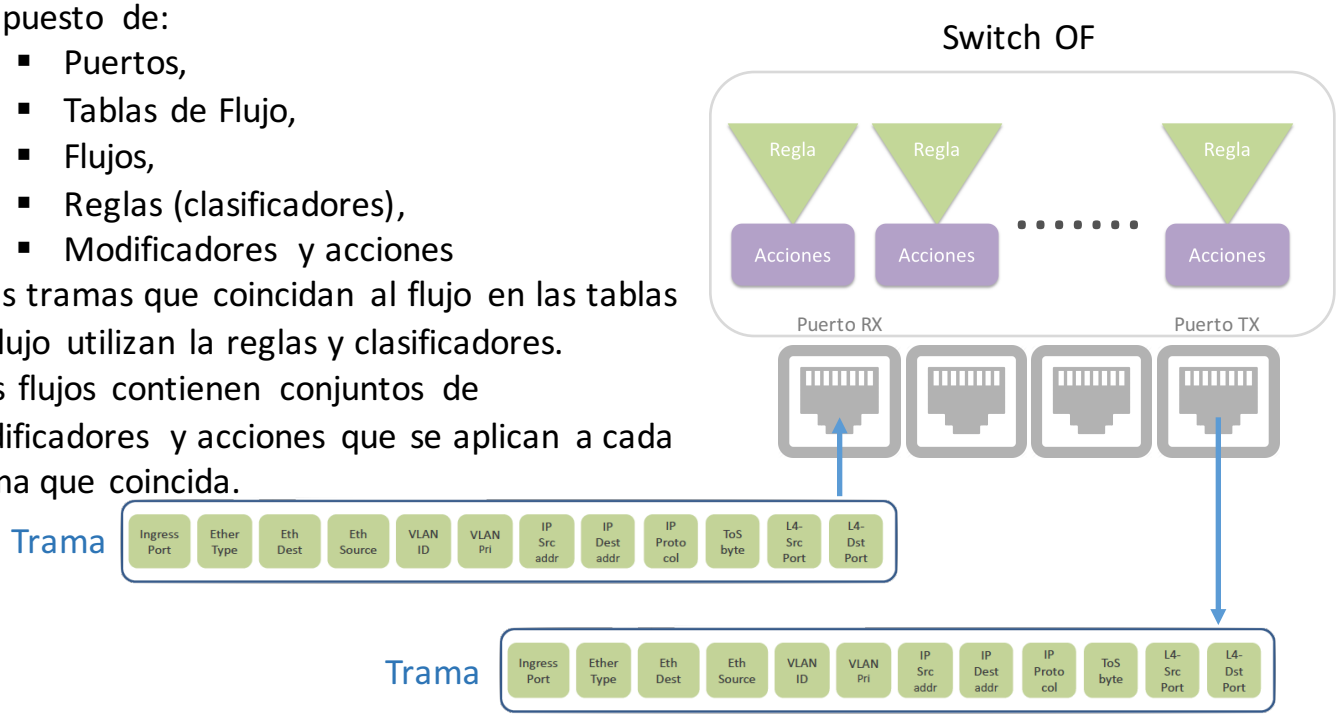
Plano de datos de un Switch OF

El plano de datos de un switch OpenFlow esta compuesto de:

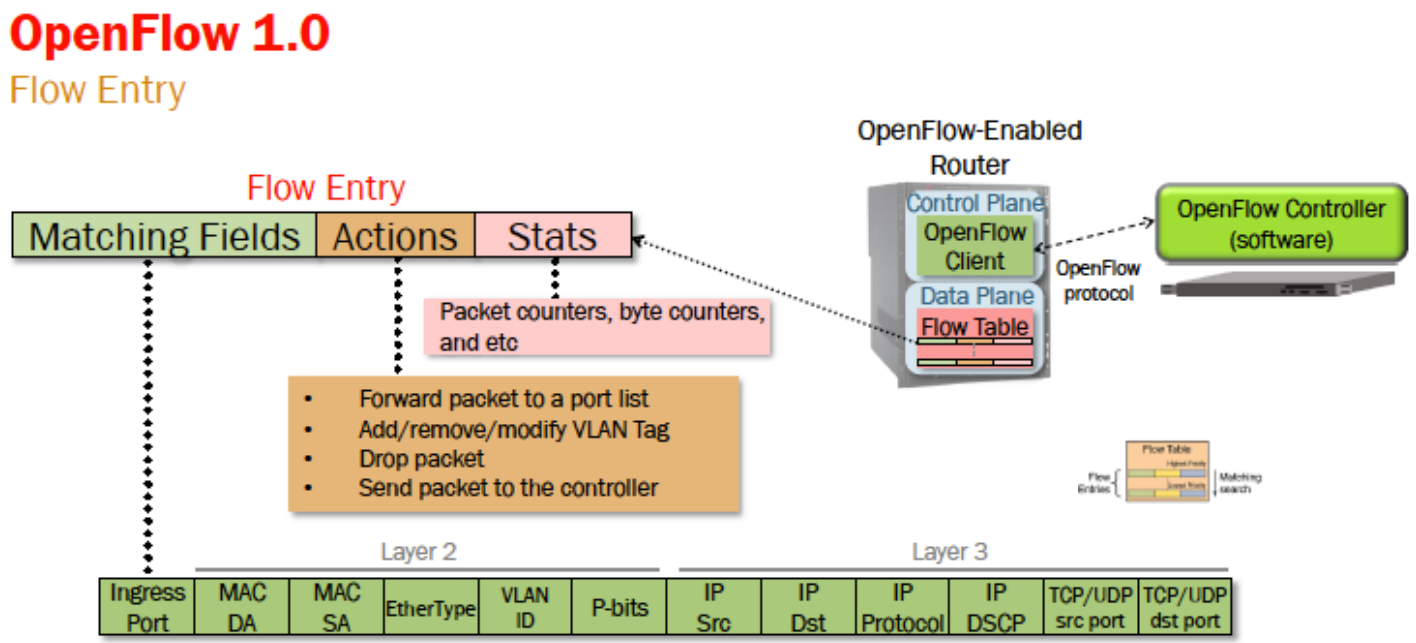
- Puertos,
- Tablas de Flujo,
- Flujos,
- Reglas (clasificadores),
- Modificadores y acciones

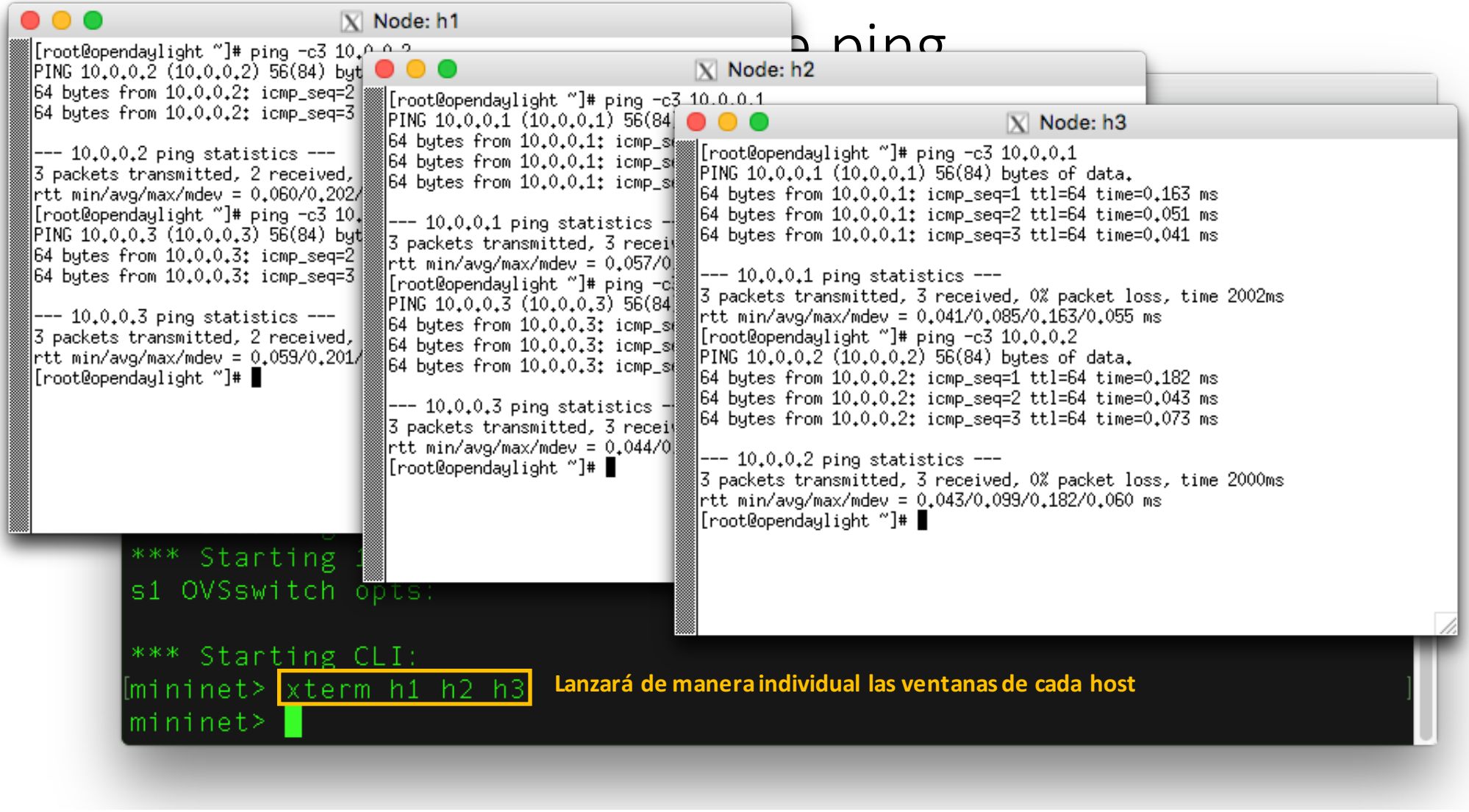
• Las tramas que coincidan al flujo en las tablas de flujo utilizan la reglas y clasificadores.

•Los flujos contienen conjuntos de modificadores y acciones que se aplican a cada trama que coincida.



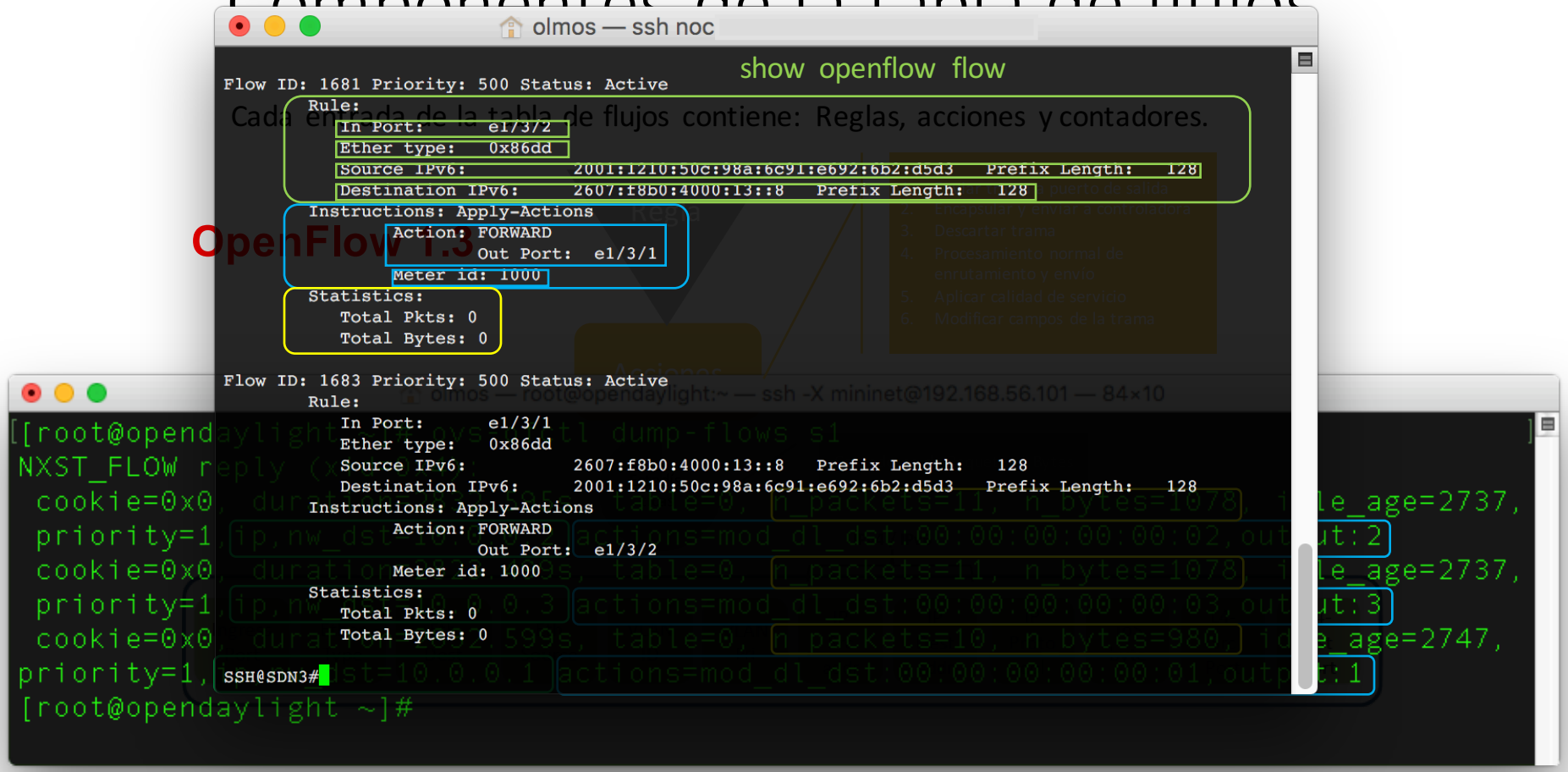
Componentes de la tabla de flujos




Node: h1
[root@opendaylight ~]# ping -c3 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) byt
64 bytes from 10.0.0.2: icmp_seq=2
64 bytes from 10.0.0.2: icmp_seq=3
--- 10.0.0.2 ping statistics ---
3 packets transmitted, 2 received,
rtt min/avg/max/mdev = 0.060/0.202/
[root@opendaylight ~]# ping -c3 10.
PING 10.0.0.3 (10.0.0.3) 56(84) byt
64 bytes from 10.0.0.3: icmp_seq=2
64 bytes from 10.0.0.3: icmp_seq=3
--- 10.0.0.3 ping statistics ---
3 packets transmitted, 2 received,
rtt min/avg/max/mdev = 0.059/0.201/
[root@opendaylight ~]#
Node: h2
[root@opendaylight ~]# ping -c3 10.0.0.1
PING 10.0.0.1 (10.0.0.1) 56(84
64 bytes from 10.0.0.1: icmp_s
64 bytes from 10.0.0.1: icmp_s
64 bytes from 10.0.0.1: icmp_s
--- 10.0.0.1 ping statistics --
3 packets transmitted, 3 recei
rtt min/avg/max/mdev = 0.057/0
[root@opendaylight ~]# ping -c
PING 10.0.0.3 (10.0.0.3) 56(84
64 bytes from 10.0.0.3: icmp_s
64 bytes from 10.0.0.3: icmp_s
64 bytes from 10.0.0.3: icmp_s
--- 10.0.0.3 ping statistics --
3 packets transmitted, 3 recei
rtt min/avg/max/mdev = 0.044/0
[root@opendaylight ~]#
Node: h3
[root@opendaylight ~]# ping -c3 10.0.0.1
PING 10.0.0.1 (10.0.0.1) 56(84) bytes of data.
64 bytes from 10.0.0.1: icmp_seq=1 ttl=64 time=0.163 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=0.051 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=0.041 ms
--- 10.0.0.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 0.041/0.085/0.163/0.055 ms
[root@opendaylight ~]# ping -c3 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=0.182 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.043 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.073 ms
--- 10.0.0.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2000ms
rtt min/avg/max/mdev = 0.043/0.099/0.182/0.060 ms
[root@opendaylight ~]#
*** Starting
s1 OVSswitch opts:
*** Starting CLI:
[mininet> xterm h1 h2 h3
mininet>
Lanzará de manera individual las ventanas de cada host

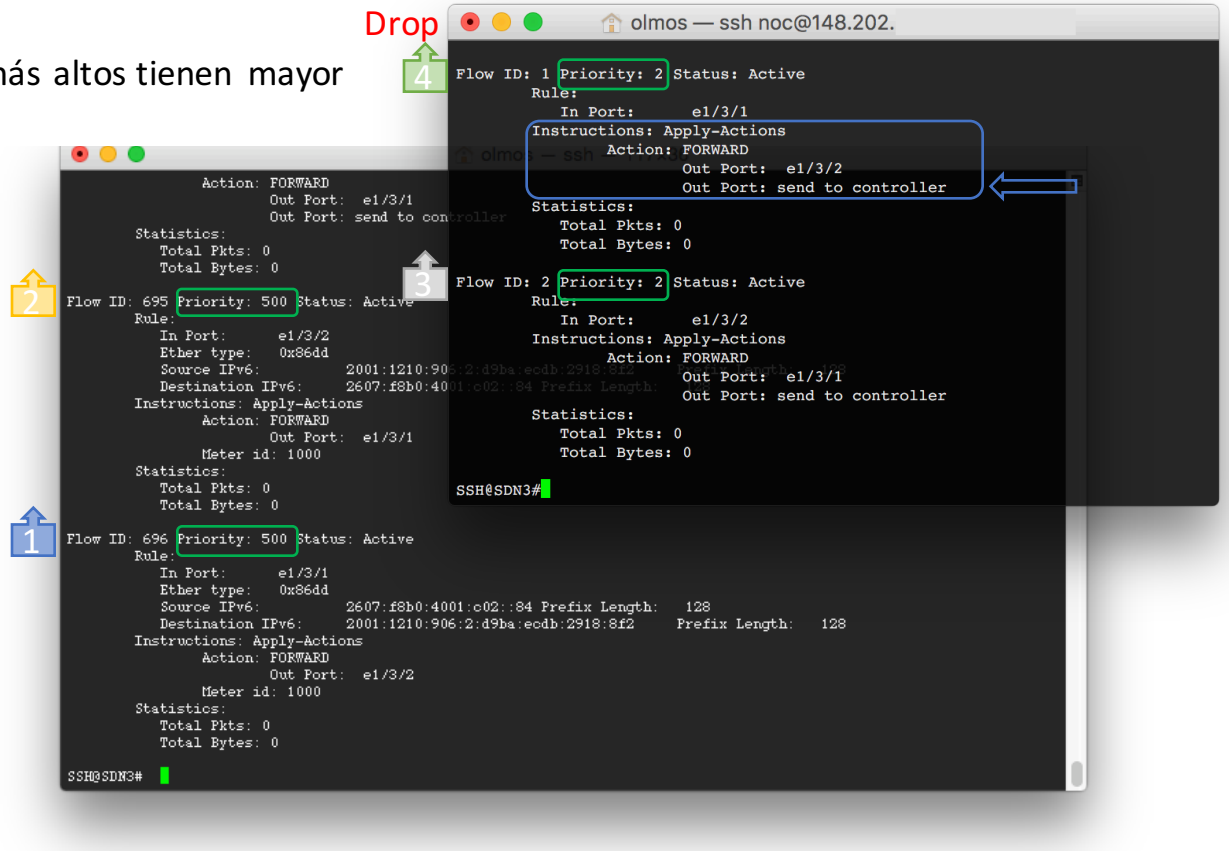
Componentes de la tabla de flujos

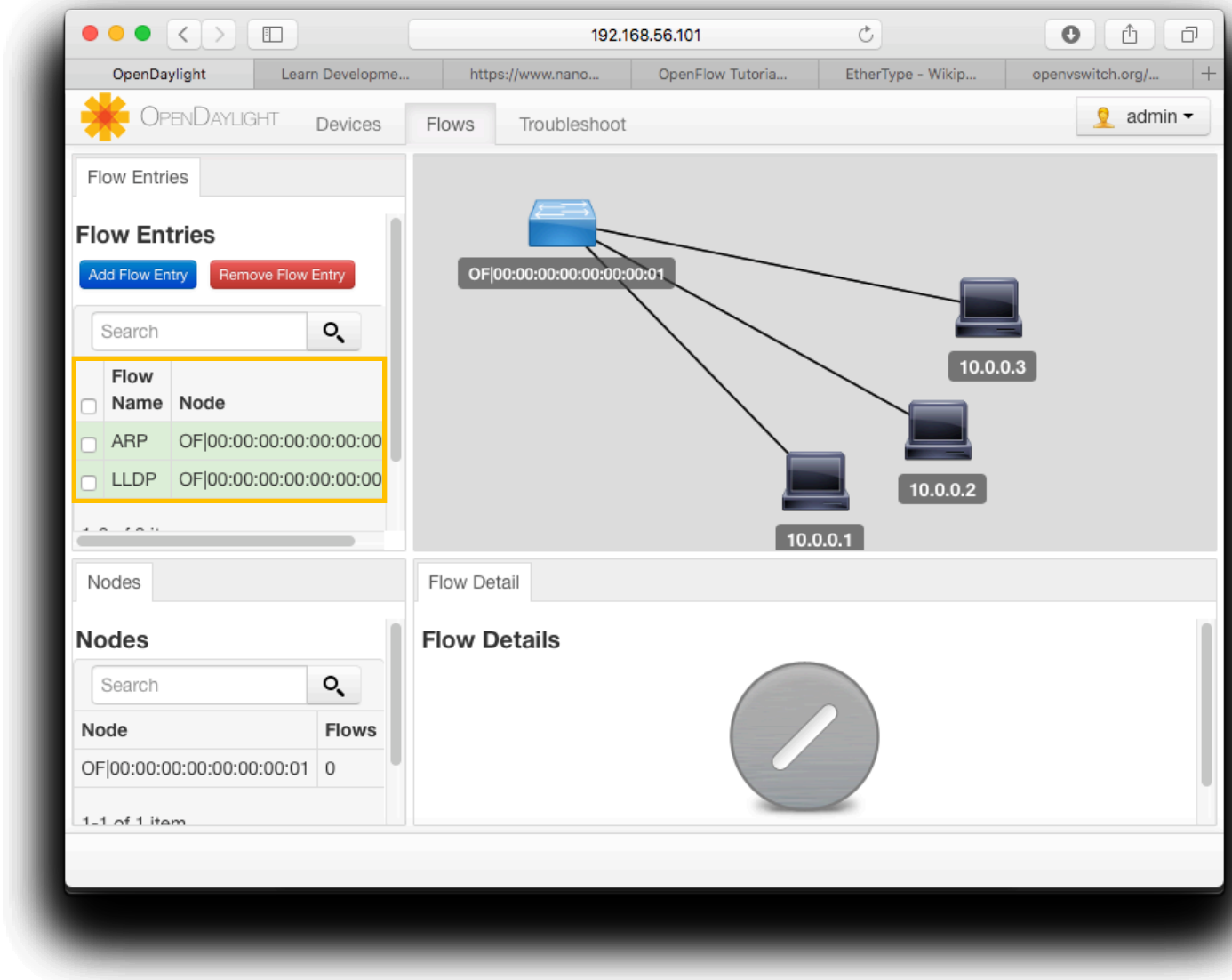
Cada entrada de la tabla de flujos contiene: Reglas, acciones y contadores.

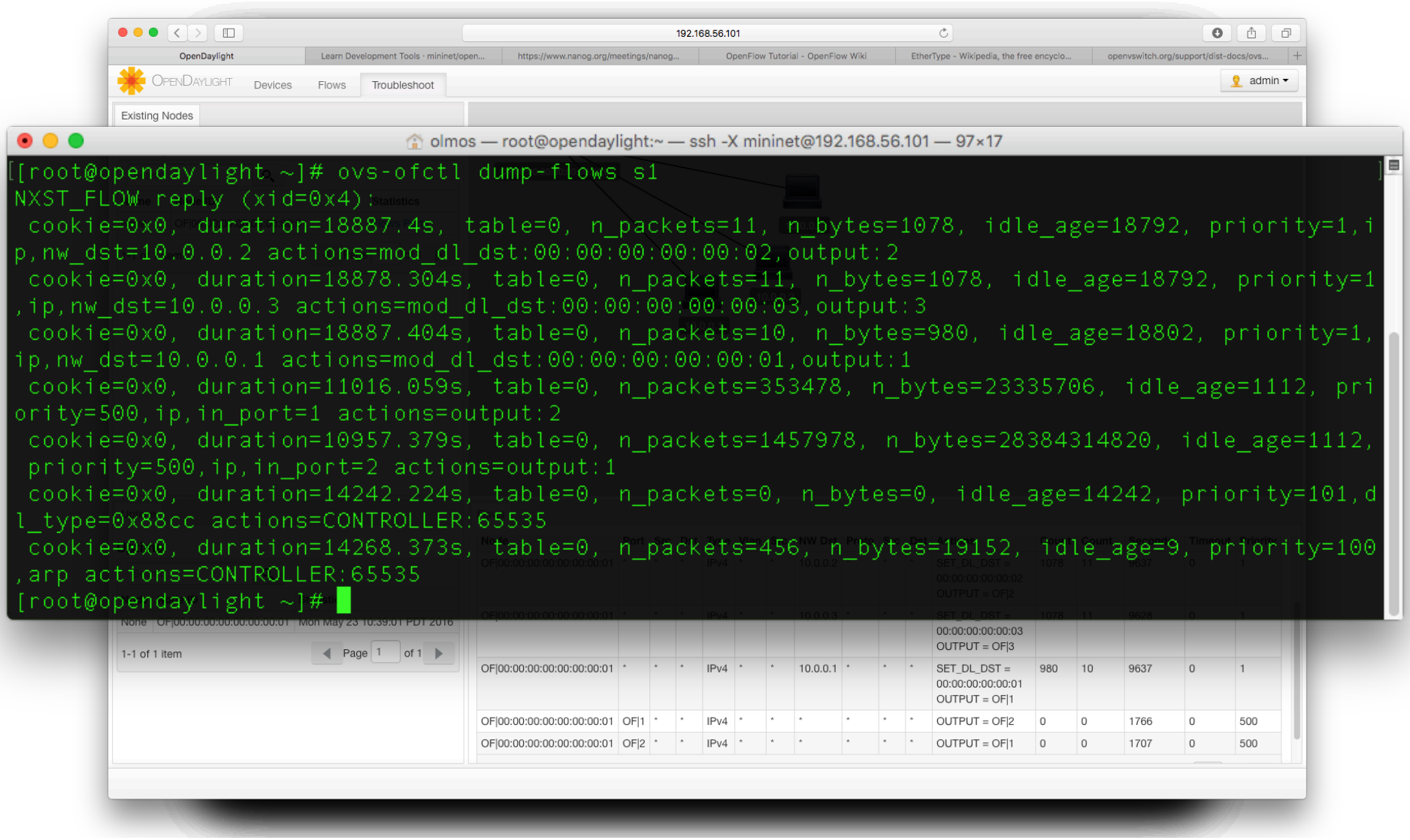


Prioridades en los Flujos

- Las tramas se comparan con las entradas de flujo basado en prioridades.
- La primer entrada con una coincidencia exacta aplicará y no continuará avanzando.
- Los números más altos tienen mayor prioridad.




192.168.56.101
OpenDaylight
Learn Developme...
https://www.nano...
OpenFlow Tutoria...
EtherType - Wikip...
openvswitch.org/...
OPENDAYLIGHT
Devices
Flows
Troubleshoot
admin
Flow Entries
Flow Entries
Add Flow Entry
Remove Flow Entry
Search
Flow Name
Node
ARP
OF|00:00:00:00:00:00:00:00
LLDP
OF|00:00:00:00:00:00:00:00
OF|00:00:00:00:00:00:00:01
10.0.0.3
10.0.0.2
10.0.0.1
Nodes
Nodes
Search
Node
Flows
OF|00:00:00:00:00:00:00:01
0
1-1 of 1 item
Flow Detail
Flow Details


192.168.56.101
OpenDaylight
Learn Development Tools · mininet/open...
https://www.nanog.org/meetings/nanog...
OpenFlow Tutorial - OpenFlow Wiki
EtherType - Wikipedia, the free encyclo...
openvswitch.org/support/dist-docs/ovs...
OpenDaylight
Devices
Flows
Troubleshoot
admin
Existing Nodes
olmos — root@opendaylight:~ — ssh -X mininet@192.168.56.101 — 97×17
[[root@opendaylight ~]# ovs-ofctl dump-flows s1
NXST_FLOW reply (xid=0x4):
cookie=0x0, duration=18887.4s, table=0, n_packets=11, n_bytes=1078, idle_age=18792, priority=1,ip,nw_dst=10.0.0.2 actions=mod_dl_dst:00:00:00:00:00:02,output:2
cookie=0x0, duration=18878.304s, table=0, n_packets=11, n_bytes=1078, idle_age=18792, priority=1,ip,nw_dst=10.0.0.3 actions=mod_dl_dst:00:00:00:00:00:03,output:3
cookie=0x0, duration=18887.404s, table=0, n_packets=10, n_bytes=980, idle_age=18802, priority=1,ip,nw_dst=10.0.0.1 actions=mod_dl_dst:00:00:00:00:00:01,output:1
cookie=0x0, duration=11016.059s, table=0, n_packets=353478, n_bytes=23335706, idle_age=1112, priority=500,ip,in_port=1 actions=output:2
cookie=0x0, duration=10957.379s, table=0, n_packets=1457978, n_bytes=28384314820, idle_age=1112, priority=500,ip,in_port=2 actions=output:1
cookie=0x0, duration=14242.224s, table=0, n_packets=0, n_bytes=0, idle_age=14242, priority=101,dl_type=0x88cc actions=CONTROLLER:65535
cookie=0x0, duration=14268.373s, table=0, n_packets=456, n_bytes=19152, idle_age=9, priority=100,arp actions=CONTROLLER:65535
[root@opendaylight ~]#
1-1 of 1 item
Page 1 of 1
OF|00:00:00:00:00:00:00:01 * * * IPv4 * * 10.0.0.1 * * * SET_DL_DST = 00:00:00:00:00:01 OUTPUT = OF|1 980 10 9637 0 1
OF|00:00:00:00:00:00:00:01 OF|1 * * IPv4 * * * * * * OUTPUT = OF|2 0 0 1766 0 500
OF|00:00:00:00:00:00:00:01 OF|2 * * IPv4 * * * * * * OUTPUT = OF|1 0 0 1707 0 500

Ejemplo de rendimiento por UDP entre dos nodos

En modo server:

```
mininet> h1 iperf -s -u -p 12345
------------------------------------------------------------
Server listening on UDP port 12345
Receiving 1470 byte datagrams
UDP buffer size: 8.00 KByte (default)
------------------------------------------------------------
[  3] local 10.0.0.1 port 12345 connected with 10.12.240.33 port 46248
```

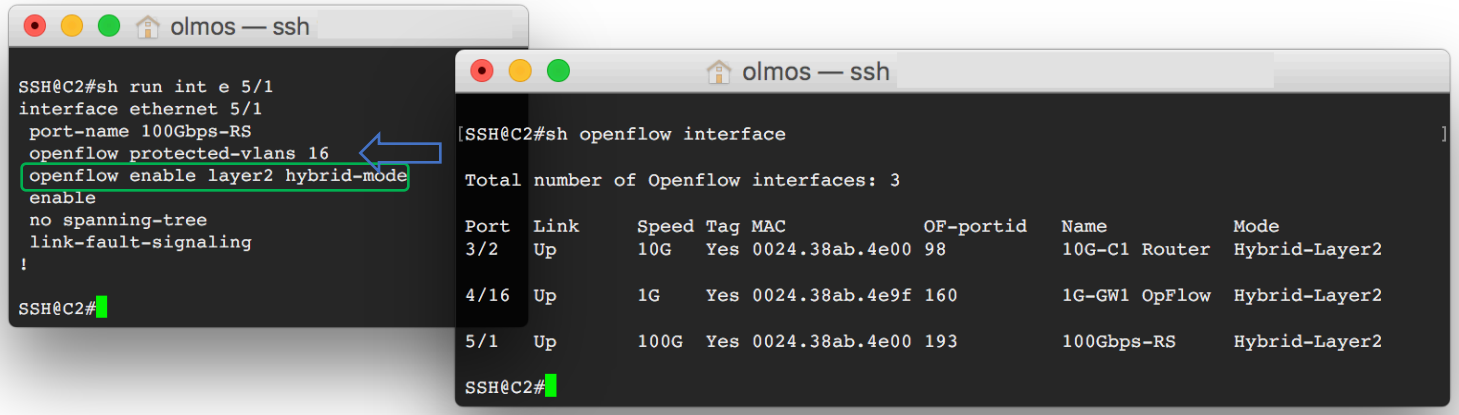
En modo cliente:

```
mininet> h2 iperf -c 10.0.0.1 -u -p 12345 -t 30 -b 250M
------------------------------------------------------------
Client connecting to 10.0.0.1, UDP port 12345
Sending 1470 byte datagrams
UDP buffer size: 8.00 KByte (default)
------------------------------------------------------------
[  3] local 10.0.0.2 port 46248 connected with 10.12.240.32 port 12345
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-30.0 sec    298 MBytes  83.4 Mbits/sec
```

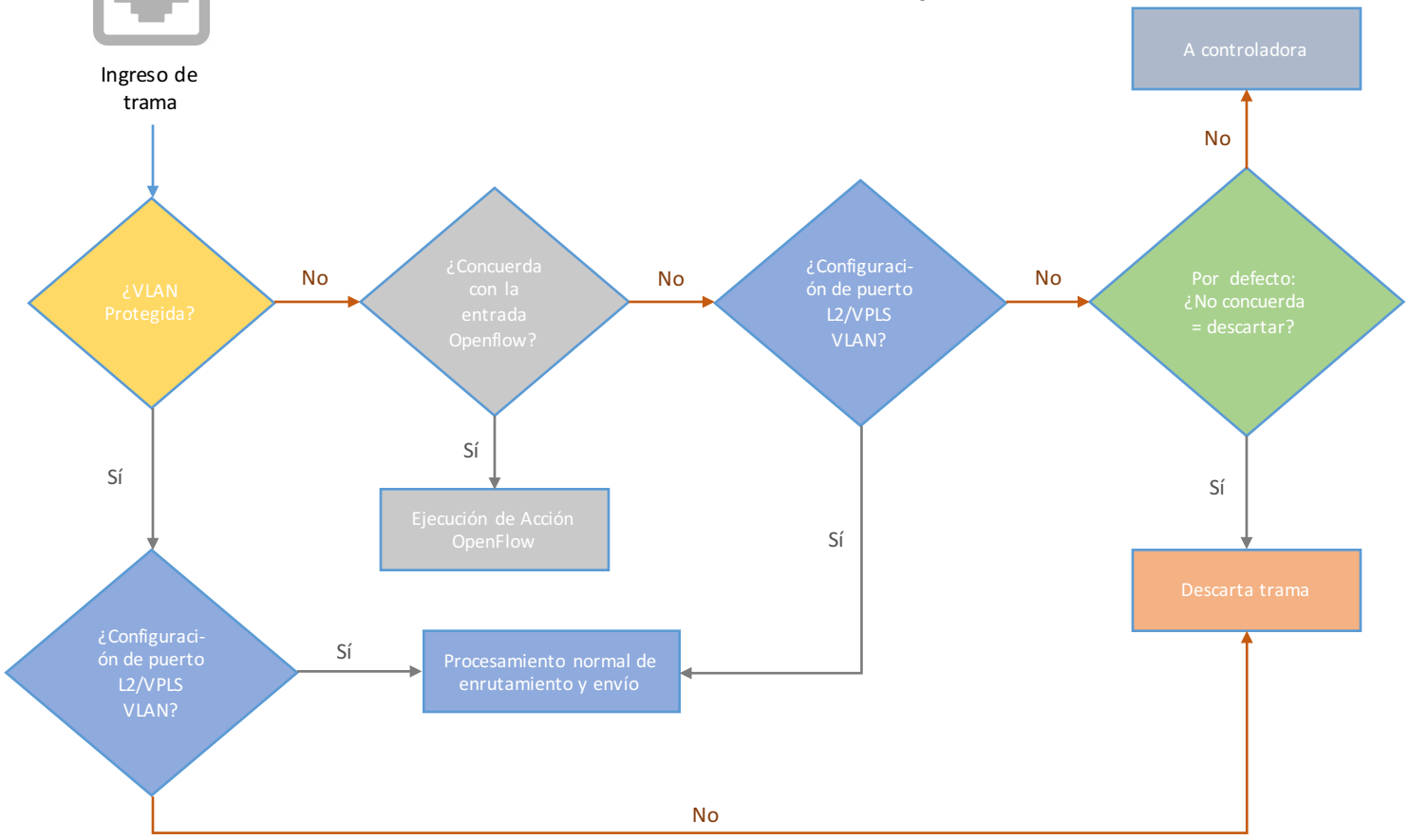
Después de la conexión del cliente, iperf prueba la red y provee estadísticas de throughput de la red.

Red híbrida (Switch)

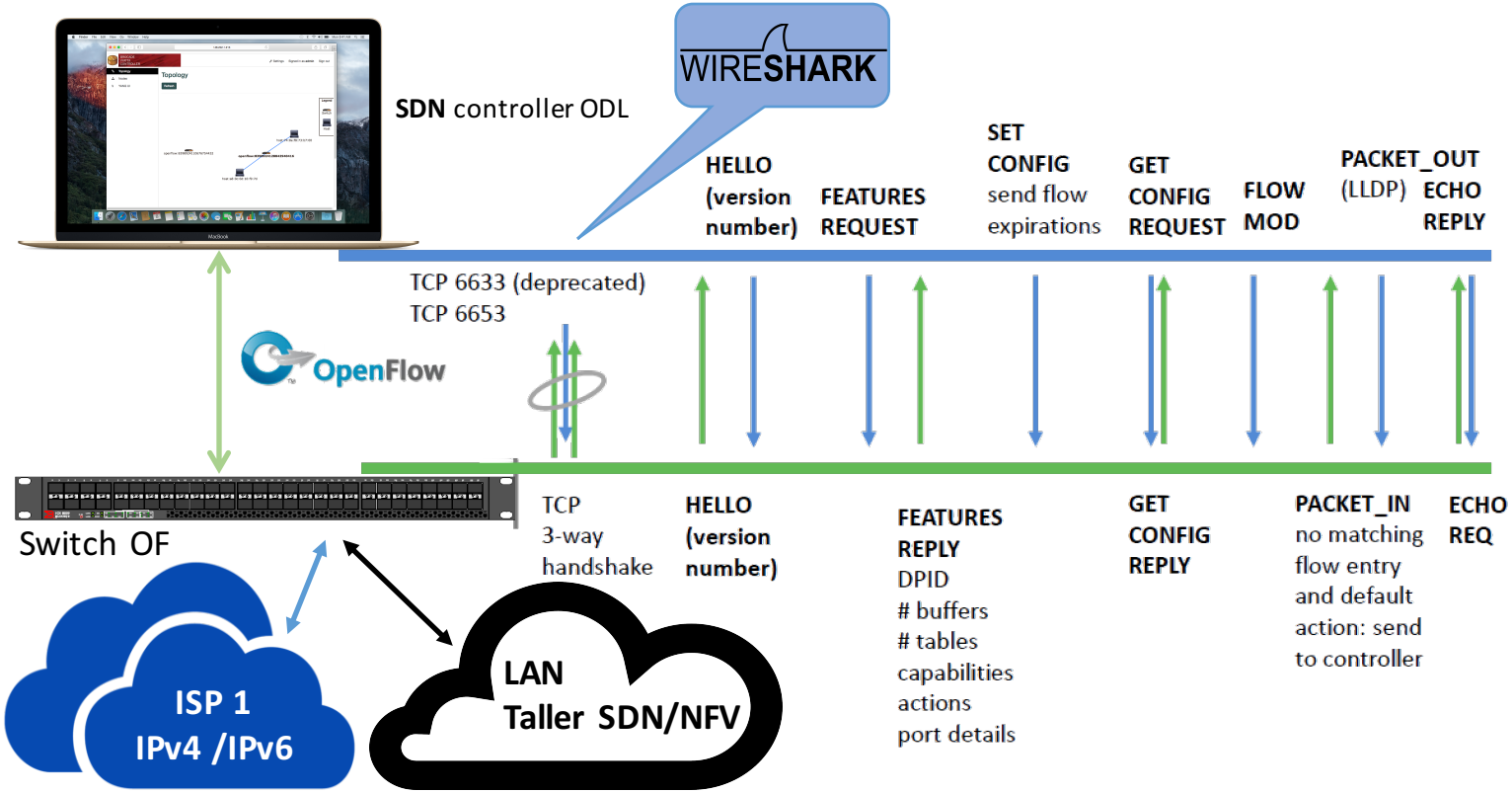
- Una red híbrida tiene ambas modalidades de **switch tradicional** y **SDN switch** operando en la misma topología.
- Los puertos pueden ser designados como puertos sólo OpenFlow, el resto de los puertos de operar bajo el control del sistema operativo del switch o de **modo híbrido**.
  - Protegiendo una o varias VLAN



Puertos híbridos OpenFlow



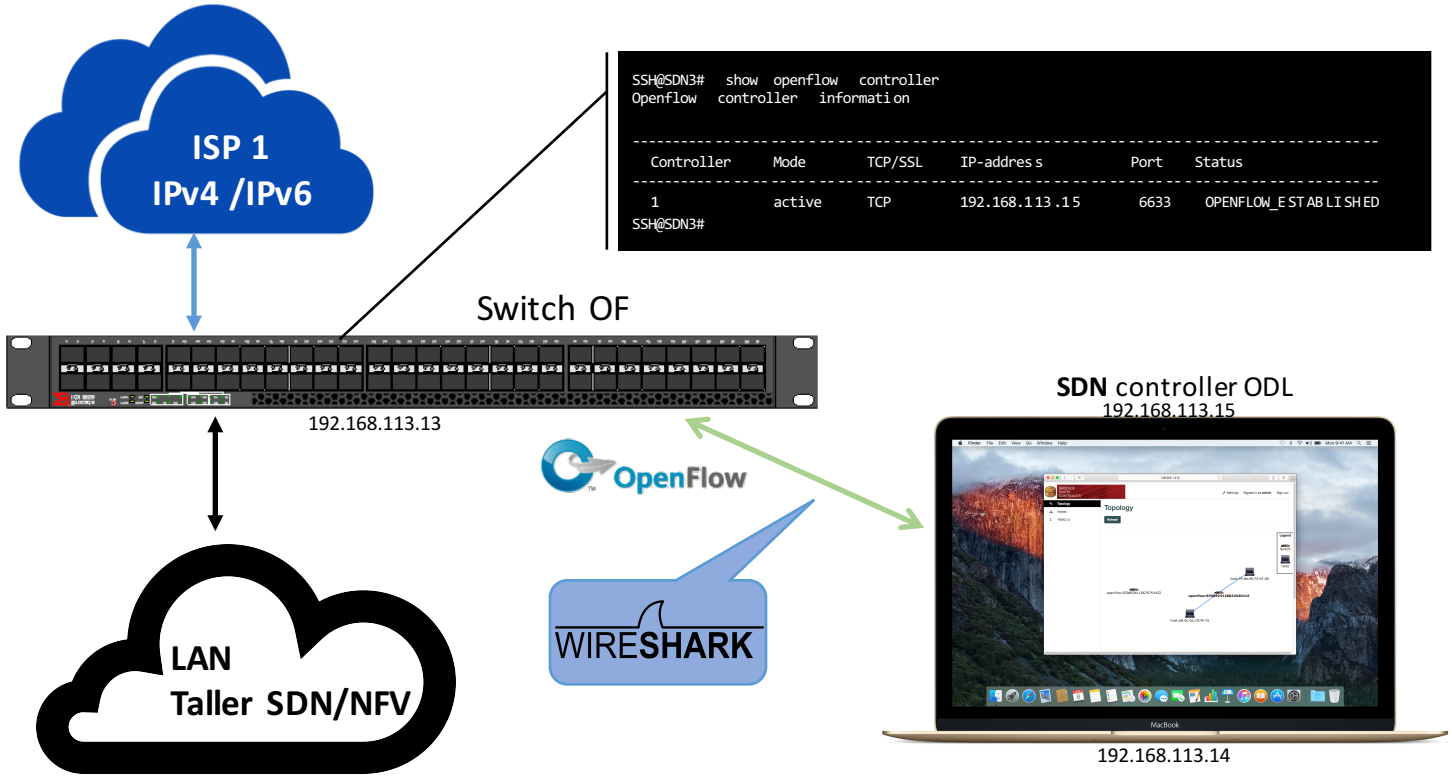
Conexión Switch OF



Ambiente del laboratorio 3

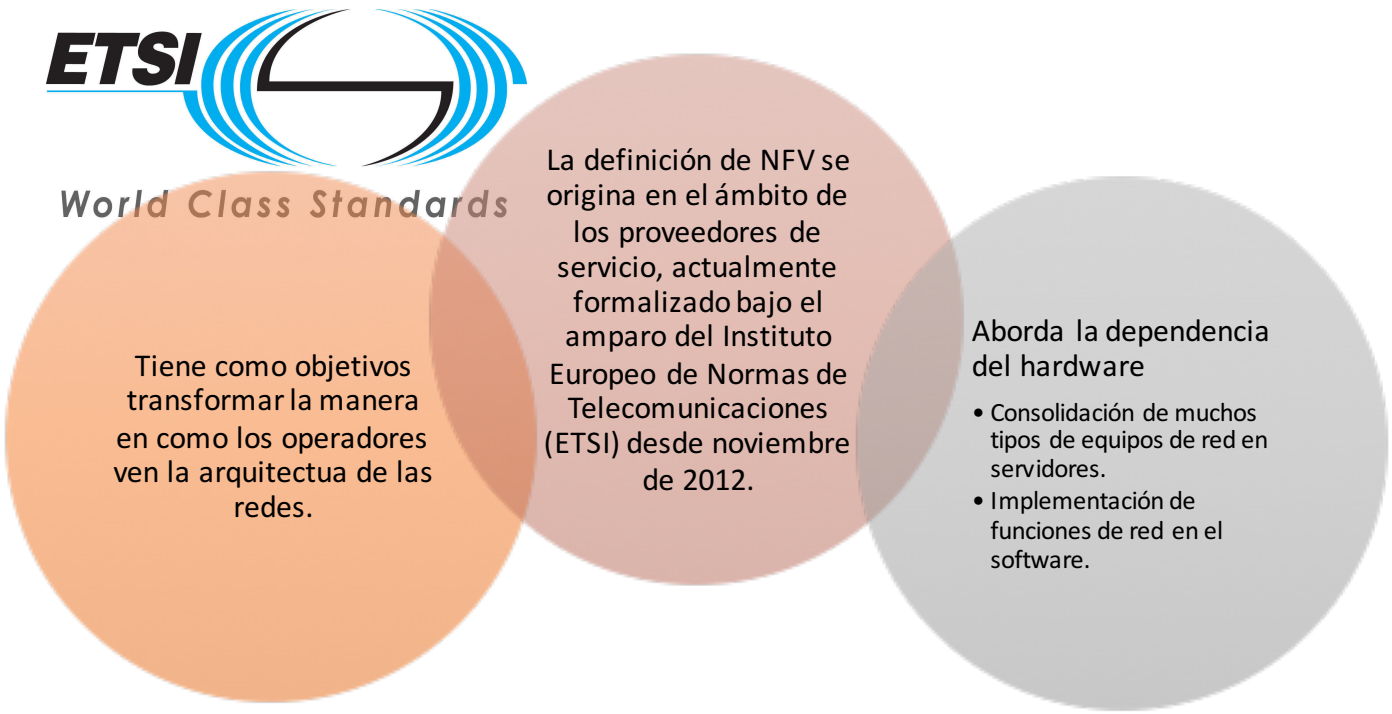
- Los siguientes ejemplos de protocolo OpenFlow se ilustran con:
  - Una controladora SDN OpenDayLight (versión: Hydrogen, Helium o Lithium) ejecuta en una máquina virtual
  - Oracle VM VirtualBox: producto de virtualización
  - ICX6610-48: OpenFlow switch híbrido capa 2-3
  - Wireshark: analizador tráfico de protocolos para Unix y Windows.

Topología de laboratorio



NFV Network Function Virtualization

¿Qué es NFV?



Fuente: http://www.etsi.org/technologies-clusters/technologies/nfv

Beneficios de las NFV

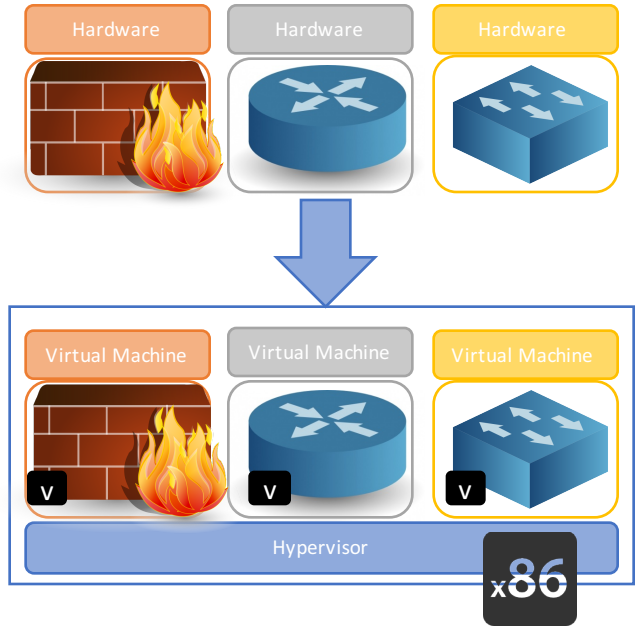
La adopción de servidores convencionales (x86) para la consolidación y la simplificación de la infraestructura.

Reduce los costos de equipos y menor consumo de energía.

El aumento de la velocidad en el aprovisionamiento, la creación de plantillas e inclusive la migración dinámica (en caliente) de máquinas virtuales para lograr alta disponibilidad y balanceo de cargas.

El aumento de velocidad del tiempo de salida a mercado (TTM).

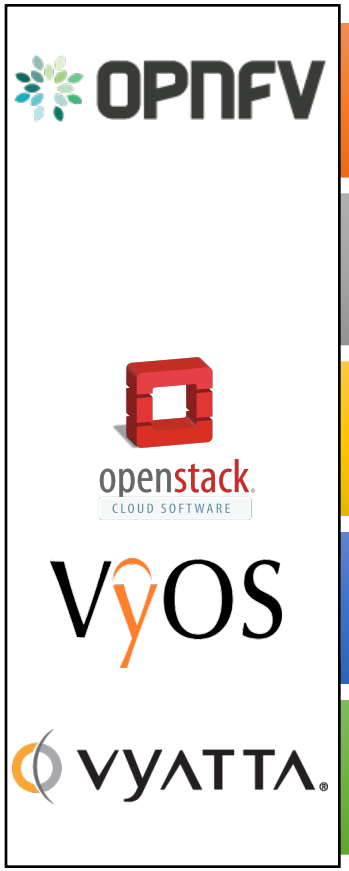
Optimiza la configuración de la red y/o topología en tiempo casi real.



Fuente: http://www.etsi.org/technologies-clusters/technologies/nfv

NFV Network Function Virtualization

Iniciativas NFV



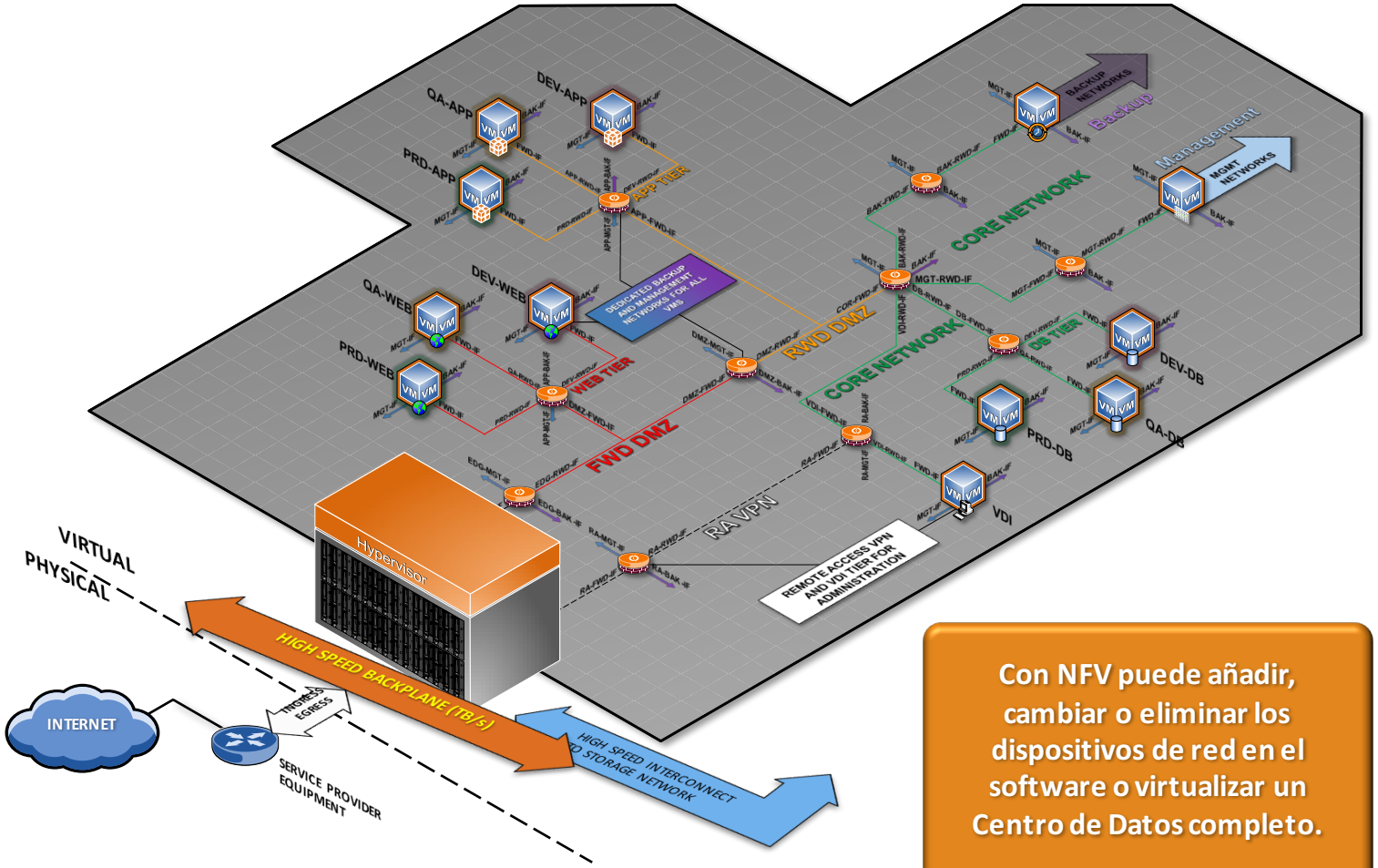
OPNFV (OpenPlataform NFV – Linux Foundation) https://www.opnfv.org

OpenNFV (HP) http://www8.hp.com/us/en/cloud/nfv-architecture.html

NFV on OpenStack (Mirantis)

VyOS Linux-based network operating system that provides software-based network routing, firewall, and VPN functionality http://vyos.net

Vyatta (vRouter) http://www.**vyatta**.com

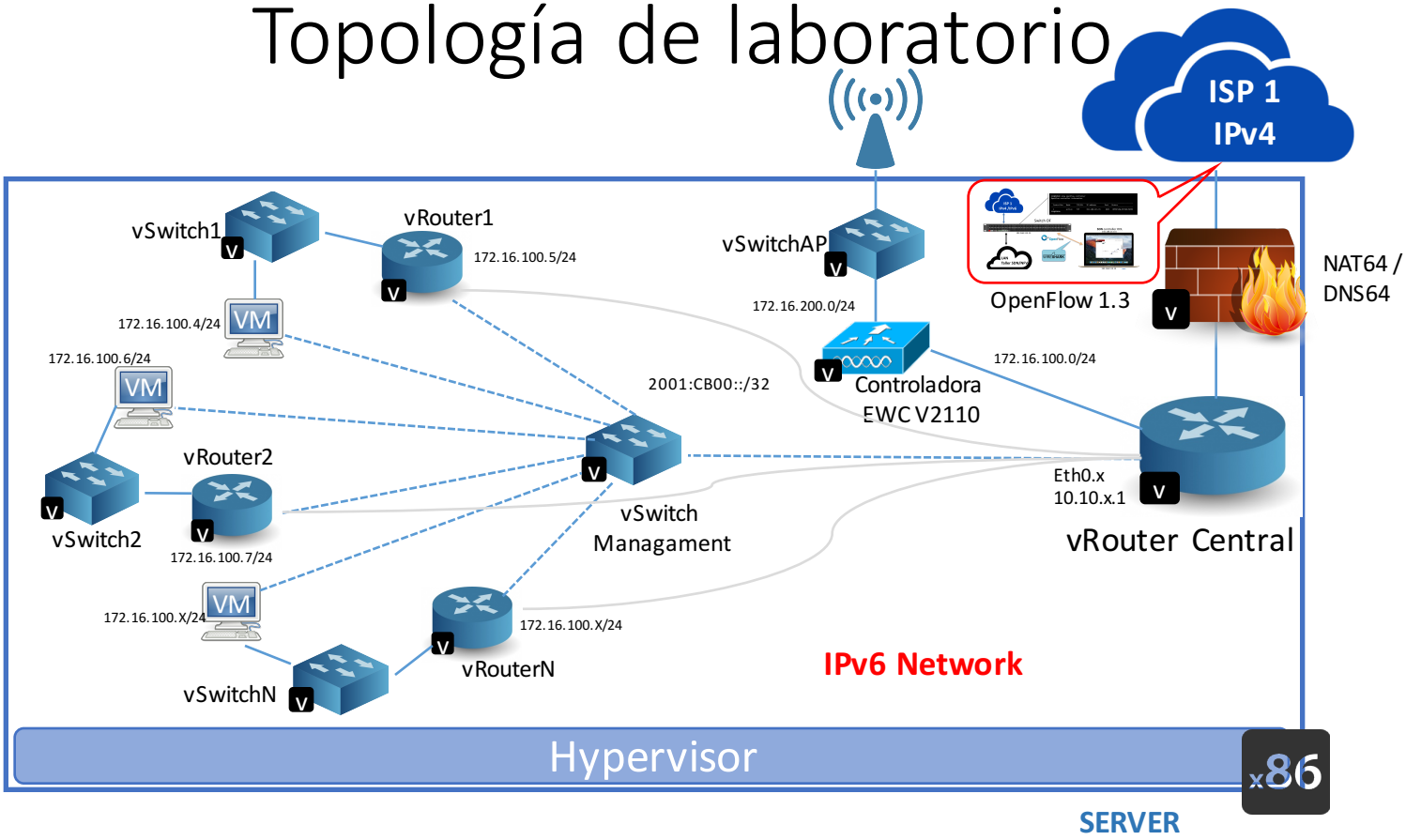


Con NFV puede añadir, cambiar o eliminar los dispositivos de red en el software o virtualizar un Centro de Datos completo.

Ambiente del laboratorio

- Los siguientes ejemplos de NFV se ilustran con:
  - Servidor en rack RH1288 V2
  - ESXi 5.5 Hypervisor : producto de virtualización.
  - Vyatta 5600 vRouter.
  - Máquinas virtuale Ubuntu 14 desktop.

Topología de laboratorio



Routers	eth0.1xx	eth1	eth 2 - Mgmt	Hosts	eth0 - Mgmt
Vyatta 2	2001:CB00:1000:102::/64	2001:CB00:1000::102:0/126	172.16.100.5	Host 2	172.16.100.4
Vyatta 3	2001:CB00:1000:103::/64	2001:CB00:1000::103:0/126	172.16.100.7	Host 3	172.16.100.6
Vyatta 4	2001:CB00:1000:104::/64	2001:CB00:1000::104:0/126	172.16.100.9	Host 4	172.16.100.8
Vyatta 5	2001:CB00:1000:105::/64	2001:CB00:1000::105:0/126	172.16.100.11	Host 5	172.16.100.10
Vyatta 6	2001:CB00:1000:106::/64	2001:CB00:1000::106:0/126	172.16.100.13	Host 6	172.16.100.12
Vyatta 7	2001:CB00:1000:107::/64	2001:CB00:1000::107:0/126	172.16.100.15	Host 7	172.16.100.14
Vyatta 8	2001:CB00:1000:108::/64	2001:CB00:1000::108:0/126	172.16.100.17	Host 8	172.16.100.16
Vyatta 9	2001:CB00:1000:109::/64	2001:CB00:1000::109:0/126	172.16.100.19	Host 9	172.16.100.18
Vyatta 10	2001:CB00:1000:1010::/64	2001:CB00:1000::1010:0/126	172.16.100.21	Host 10	172.16.100.20
Vyatta 11	2001:CB00:1000:1011::/64	2001:CB00:1000::1011:0/126	172.16.100.23	Host 11	172.16.100.22
Vyatta 12	2001:CB00:1000:1012::/64	2001:CB00:1000::1012:0/126	172.16.100.25	Host 12	172.16.100.24
Vyatta 13	2001:CB00:1000:1013::/64	2001:CB00:1000::1013:0/126	172.16.100.27	Host 13	172.16.100.26
Vyatta 14	2001:CB00:1000:1014::/64	2001:CB00:1000::1014:0/126	172.16.100.29	Host 14	172.16.100.28
Vyatta 15	2001:CB00:1000:1015::/64	2001:CB00:1000::1015:0/126	172.16.100.31	Host 15	172.16.100.30
Vyatta 16	2001:CB00:1000:1016::/64	2001:CB00:1000::1016:0/126	172.16.100.33	Host 16	172.16.100.32
Vyatta 17	2001:CB00:1000:1017::/64	2001:CB00:1000::1017:0/126	172.16.100.35	Host 17	172.16.100.34
Vyatta 18	2001:CB00:1000:1018::/64	2001:CB00:1000::1018:0/126	172.16.100.37	Host 18	172.16.100.36
Vyatta 19	2001:CB00:1000:1019::/64	2001:CB00:1000::1019:0/126	172.16.100.39	Host 19	172.16.100.38
Vyatta 20	2001:CB00:1000:1020::/64	2001:CB00:1000::1020:0/126	172.16.100.41	Host 20	172.16.100.40
Vyatta 21	2001:CB00:1000:1021::/64	2001:CB00:1000::1021:0/126	172.16.100.43	Host 21	172.16.100.42
Vyatta 22	2001:CB00:1000:1022::/64	2001:CB00:1000::1022:0/126	172.16.100.45	Host 22	172.16.100.44
Vyatta 23	2001:CB00:1000:1023::/64	2001:CB00:1000::1023:0/126	172.16.100.47	Host 23	172.16.100.46
Vyatta 24	2001:CB00:1000:1024::/64	2001:CB00:1000::1024:0/126	172.16.100.49	Host 24	172.16.100.48
Vyatta 25	2001:CB00:1000:1025::/64	2001:CB00:1000::1025:0/126	172.16.100.51	Host 25	172.16.100.50

```
interfaces {
    ethernet eth0 {
        vif 101 {
            address 2001:CB00:1000::10X:2/126
        }
    }
    ethernet eth1 {
        address 2001:cb00:1000:10X::A/64
        ipv6 {
            address {
                eui64 2001:cb00:1000:10X::/64
            }
            router-advert {
                other-config-flag true
                prefix 2001:cb00:1000:10X::/64 {
                    autonomous-flag true
                    on-link-flag true
                }
            }
        }
}
```

```
protocols {
    static {
route6 ::/0 {
            next-hop 2001:cb00:1000::101:X {
            }
        }
    }
}
```

X =			
	102	1011	1020
	103	1012	1021
	104	1013	1022
	105	1014	1023
	106	1015	1024
	107	1016	1025
	108	1017	
	109	1018	
	1010	1019	

```
host@Host:~$ sudo vi /etc/resolv.conf
# Dynamic resolv.conf(5) file for glibc resolver(3) generated by resolvconf(8)
#     DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WILL BE OVERWRITTEN
nameserver 2001:cb00::1
```

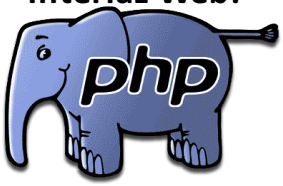
```
host@Host:~$ sudo vi /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet static
address 172.16.100.2
netmask 255.255.255.0
network 172.16.100.0
broadcast 172.16.100.255

auto eth1
iface eth1 inet dhcp
```

Aplicación SDN - Optimización de Tráfico

Aplicación SDN - Optimización de Tráfico

- **A**plicación abierta que optimiza del tráfico de red a través del monitoreo proactivo y el establecimiento de políticas de flujo para mejorar la utilización de recursos, mitigar ataques de red y reducir la congestión de la red de forma automatizada.
- **E**scrita en **Python**.
- **C**omponentes de sistema:
  - Colector
  - Base de datos
  - Controladora
  - Protocolo

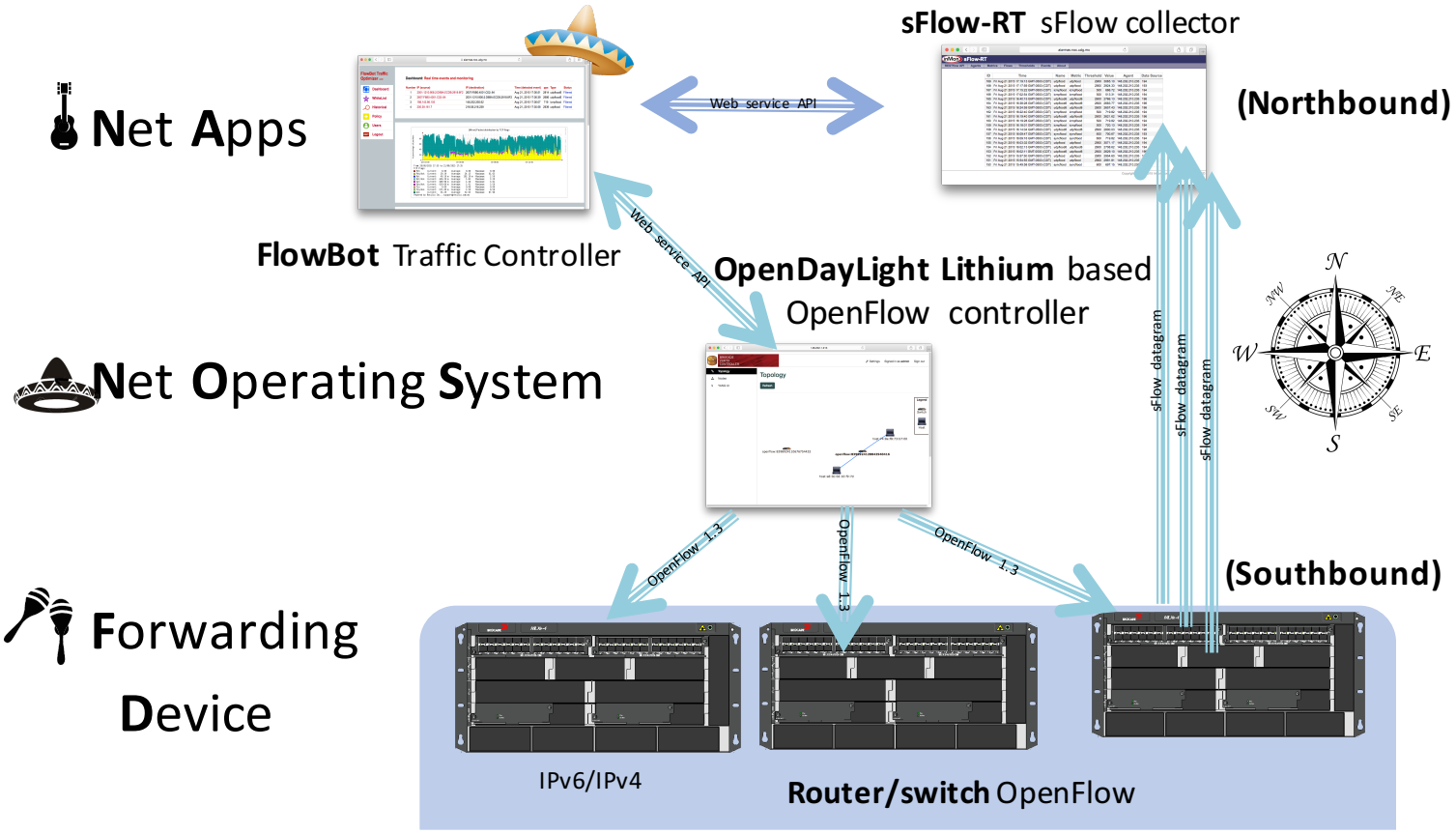
Interfaz Web:



Aplicaciones de red

- **D**esarrollada por el **NOC-UDG** con el apoyo de la iniciativa privada, INITEL-RESILIO.
- **E**n el pasado el **NOC-UDG** desarrolló diversas aplicaciones para monitoreo y optimización de tráfico de red.
  - MRTG, Perl, RRDTool, remote shell (SSH, Telnet)

Network Operation Center
universidad de guadalajara

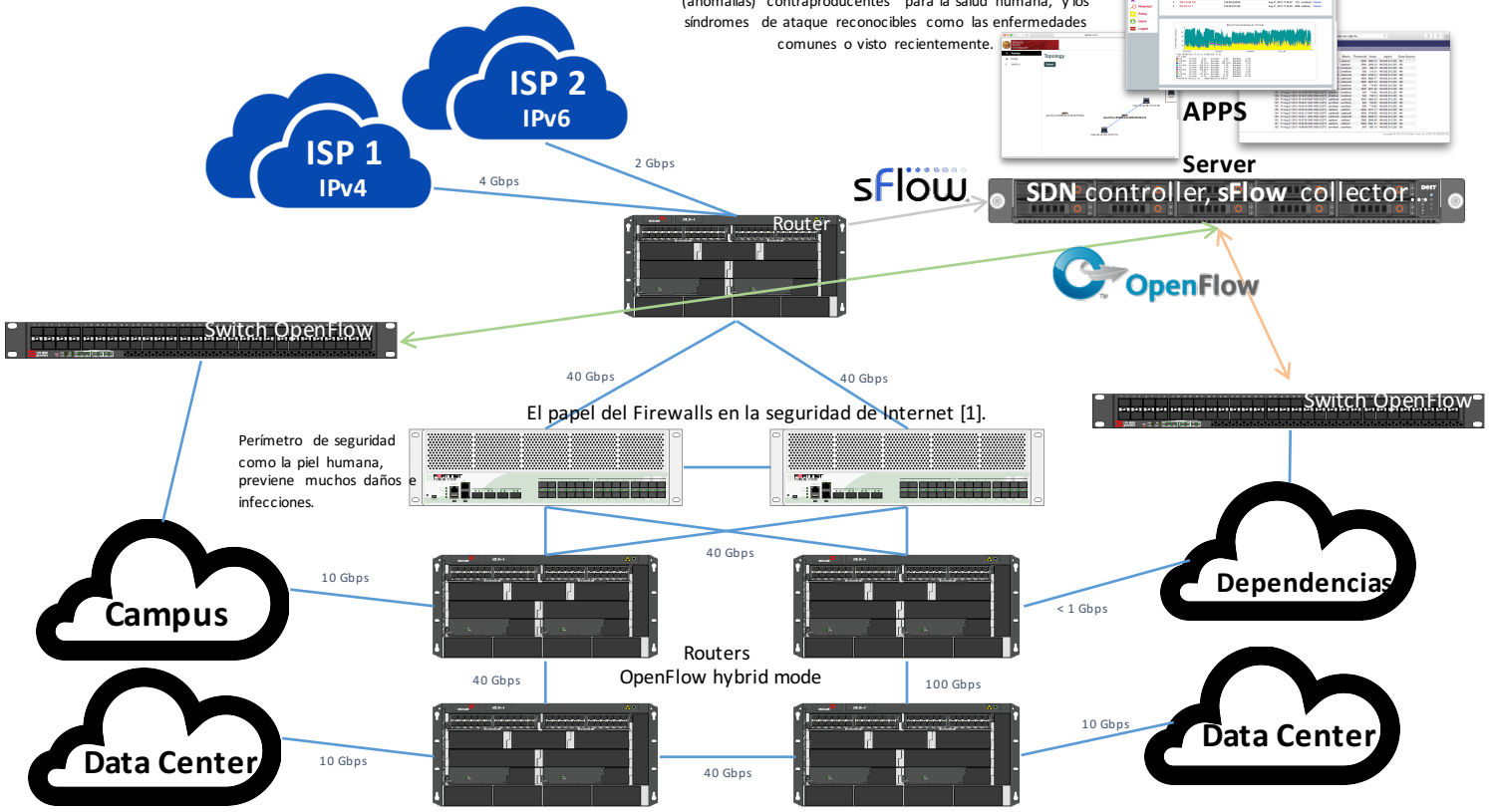

Net Apps
Net Operating System
Forwarding Device
FlowBot Traffic Controller
Web service API
sFlow-RT sFlow collector
(Northbound)
OpenDayLight Lithium based OpenFlow controller
sFlow datagram
sFlow datagram
sFlow datagram
OpenFlow 1.3
OpenFlow 1.3
OpenFlow 1.3
(Southbound)
IPv6/IPv4
Router/switch OpenFlow

Características de la Aplicación SDN

- **M**onitoreo y detección de patrones de comportamiento enla red (congestión, abusos, DoS, etc.).
  - A través de ports mirror, sflow, syslog, etc.
- **S**oporte de los protocolos IPv6 e IPv4.
- **R**egistro de lista de blanca de direcciones IP (VIP).
- **A**cciones:
  - **R**edireccionamiento de tráfico.
  - **B**loqueo de tráfico.
  - **C**alidad de servicio (administración de ancho de banda).
  - **B**alanceo de cargas de tráfico.
- **T**iempos de aplicación de acciones.

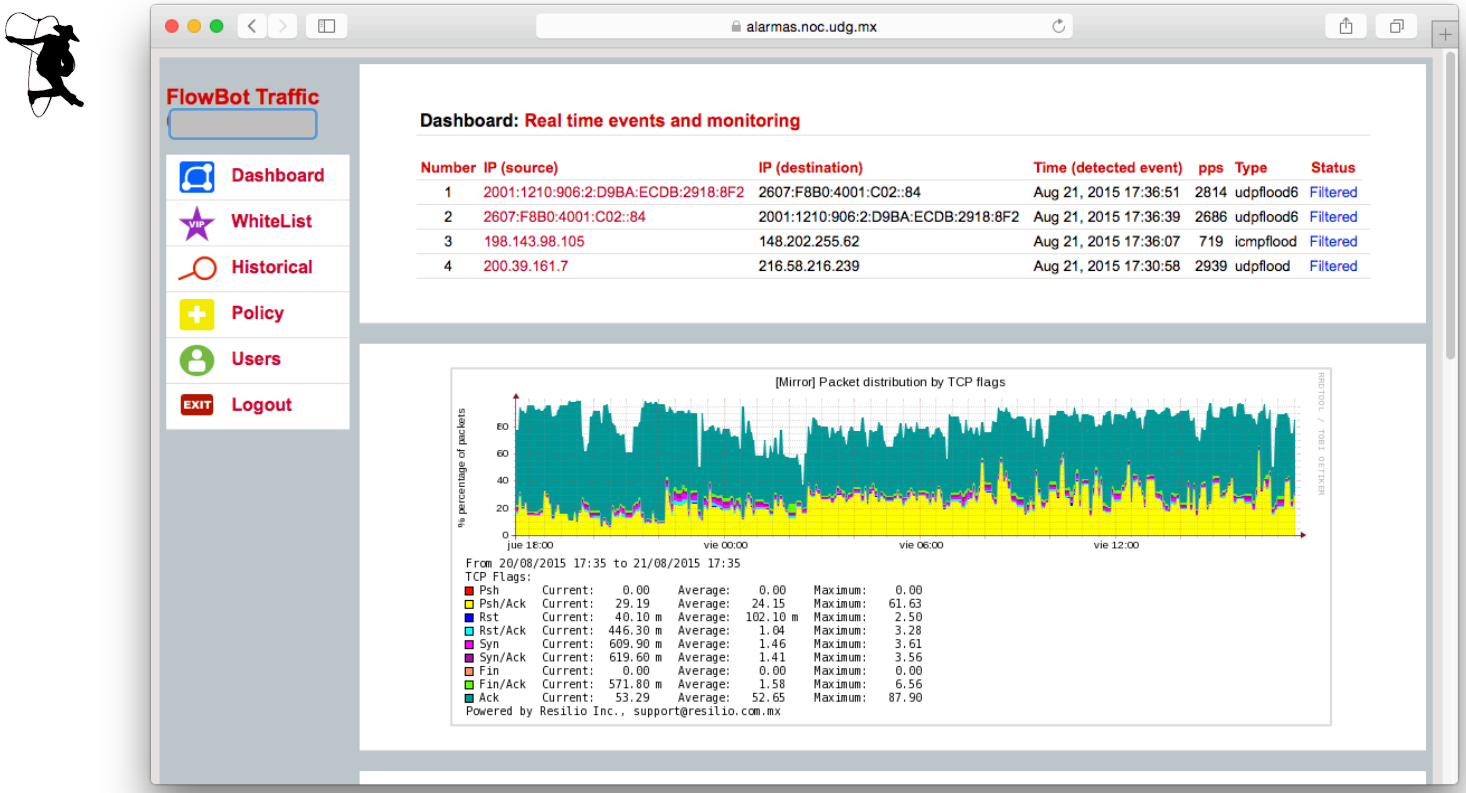
Topología de la UDG

En el cuerpo, luego que se presume que el perímetro de seguridad ha sido violado; defensas reales contra los ataques en el cuerpo incluyen potentes sistemas que detectan cambios (anomalías) contraproducentes para la salud humana, y los síndromes de ataque reconocibles como las enfermedades comunes o visto recientemente.



[1] On Firewalls in Internet Security http://www.ietf.org/id/draft-gont-opsawg-firewalls-analysis-00.txt

Dashboard de la Aplicación SDN



Mascota oficial: León Negro

Tabla de flujos IPv4 – Switch OF

UNIVERSIDAD DE GUADALAJARA
Administración General

olmos — ssh — 118×36

```
          Destination IP:       200.39.161.7   Subnet IP:       255.255.255.255
       Instructions: Apply-Actions
              Action: FORWARD
                      Out Port:  e1/3/2
              Meter id: 1000
       Statistics:
          Total Pkts: 30764
          Total Bytes: 2608873

Flow ID: 1212 Priority: 500 Status: Active
       Rule:
          In Port:      e1/3/2
          Ether type:   0x800
          Source IP:            200.39.161.7   Subnet IP:       255.255.255.255
          Destination IP:       216.58.216.239 Subnet IP:       255.255.255.255
       Instructions: Apply-Actions
              Action: FORWARD
                      Out Port:  e1/3/1
              Meter id: 1000
       Statistics:
          Total Pkts: 57152
          Total Bytes: 65261600

Flow ID: 1214 Priority: 500 Status: Active
       Rule:
          In Port:      e1/3/1
          Ether type:   0x800
          Source IP:            198.143.98.105 Subnet IP:       255.255.255.255
          Destination IP:       148.202.255.62 Subnet IP:       255.255.255.255
       Instructions: Apply-Actions
              Action: DROP
       Statistics:
          Total Pkts: 0
          Total Bytes: 0

SSH@SDN1#
```

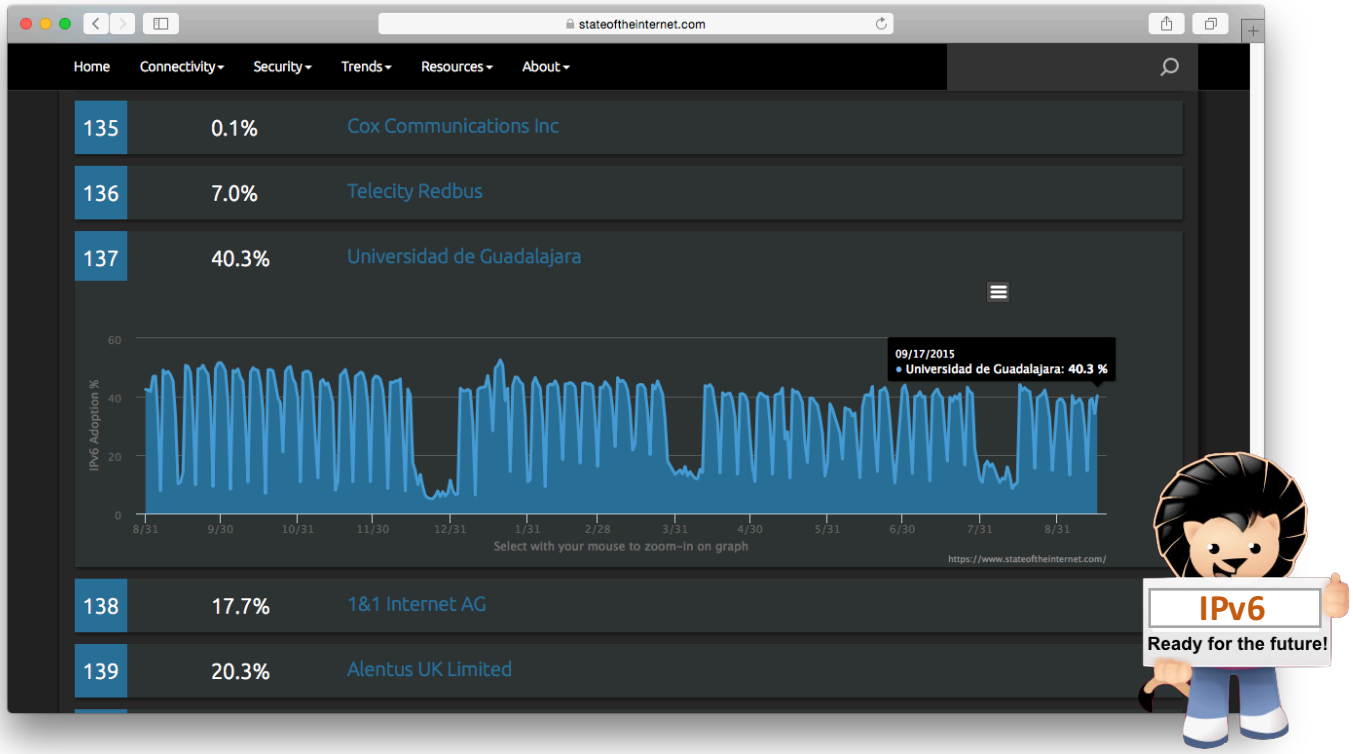
Tabla de flujos IPv6 – Switch OF

SDN/NFV - IPv6

Representaciones lógicas de:
Funciones de capa 2: **vBridge**
Funciones de capa 3: **vRouter**
Funciones de seguridad: **vFirewall**
De túneles: **vTunnel**
Virtual interface
De conectividad capa 1: Virtual Link (**vLink**)

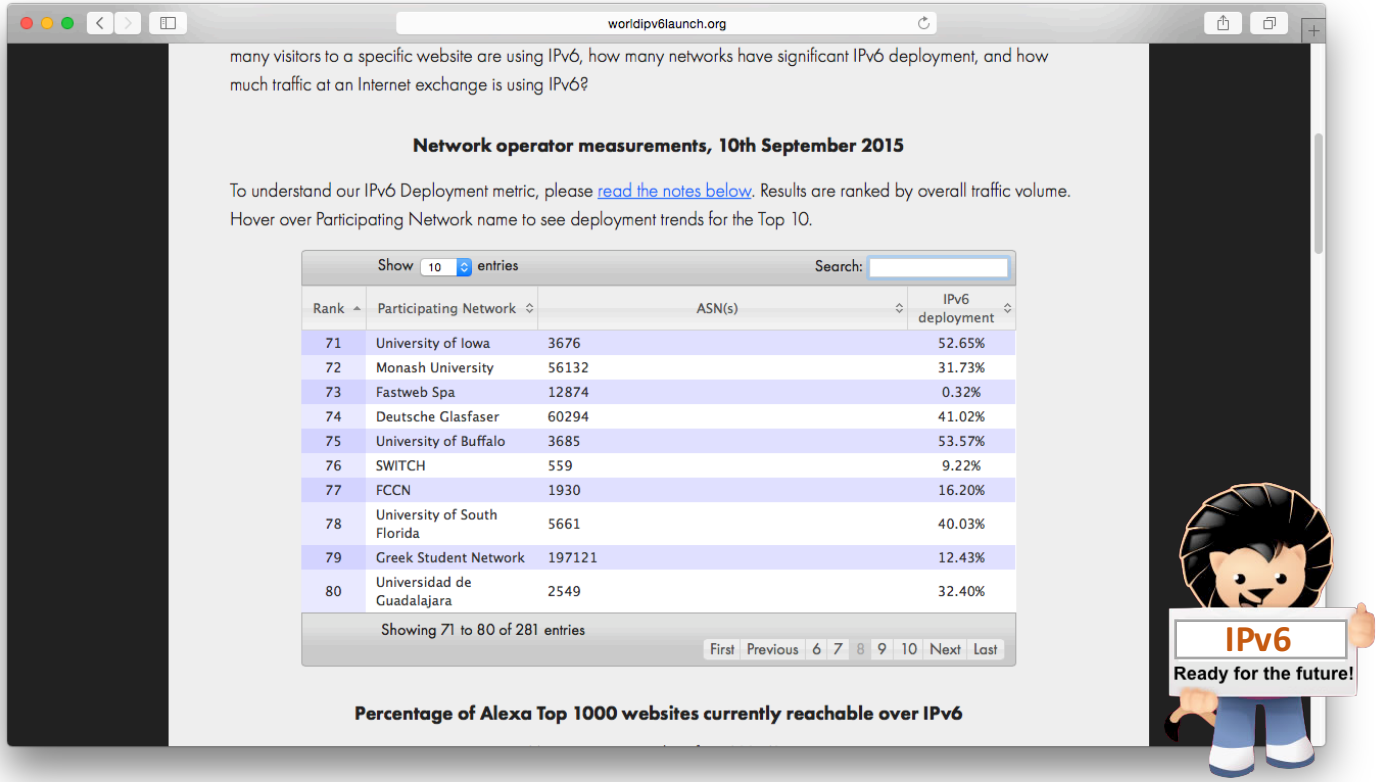
Dual-stack network UDG

Trends visualizations IPv6 adoption:
https://www.stateoftheinternet.com/



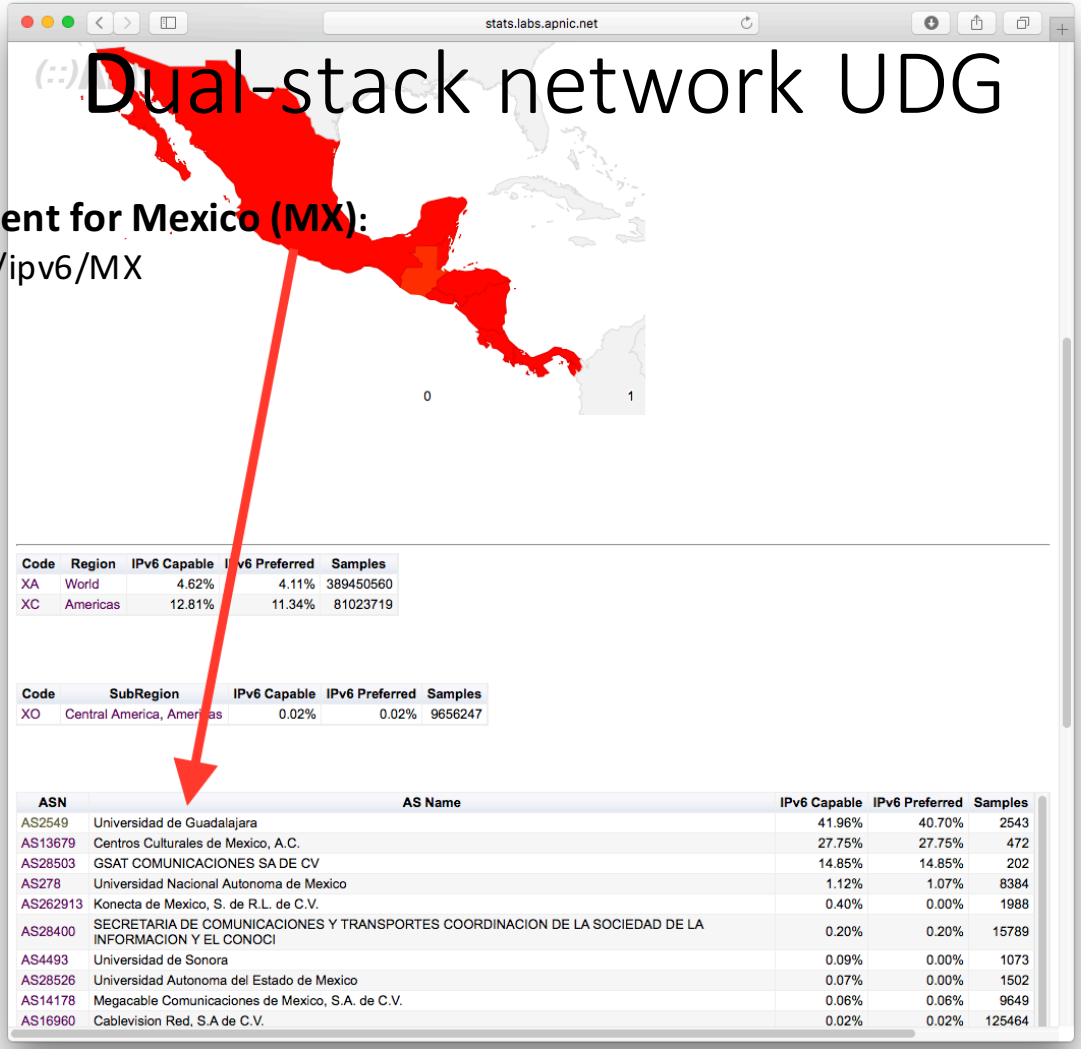
Dual-stack network UDG

World IPv6 Launch:
http://www.worldipv6launch.org/



Dual-stack network UDG

IPv6 Country Deployment for Mexico (MX):
http://stats.labs.apnic.net/ipv6/MX



- **J**aime Olmos de la Cruz
  @olmosv6
  jaime@noc.udg.mx
  http://www.ipv6.udg.mx

- **T**eobaldo Leal Arriaga
  tleal@initel.com.mx
  http://www.initel.com.mx