

# Particionado de Red Basado en Redes SDN

Autor: Ángel Guzmán Martínez

Tutor: Jorge Navarro Ortiz

**Departamento de Teoría de la Señal, Telemática y  
Comunicaciones**

**Universidad de Granada**



- |   |                            |   |                |
|---|----------------------------|---|----------------|
| 1 | Objetivos y Motivación     | 5 | Implementación |
| 2 | Introducción a SDN         | 6 | Conclusiones   |
| 3 | Descripción de<br>OpenFlow |   |                |
| 4 | Hipervisor OpenFlow        |   |                |

- 1 Objetivos y Motivación
- 2 Introducción a SDN
- 3 Descripción de OpenFlow
- 4 Hipervisor OpenFlow
- 5 Implementación
- 6 Conclusiones

## Objetivos

- Entender las limitaciones de las redes tradicionales.
- Familiarizaros con el concepto de SDN y *network slicing*.
- Particionar una red SDN y comprobar su funcionamiento.

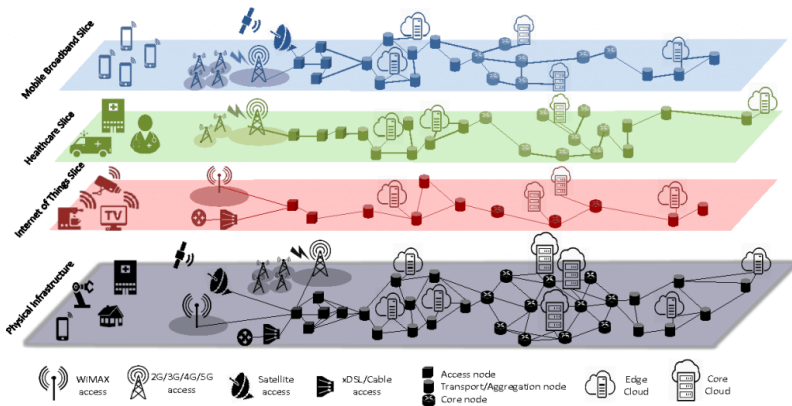
## Objetivos

- Entender las limitaciones de las redes tradicionales.
- Familiarizaros con el concepto de SDN y *network slicing*.
- Particionar una red SDN y comprobar su funcionamiento.

## ¿Por qué particionar una red (network slicing)?

- Tráfico heterogéneo. IoT, eMBB, URLLC...
- Redes Tradicionales. Limitaciones en *Quality of Service* (QoS).
- Implantación de *Software Defined Network* (SDN).

# Ejemplo de Network Slicing



Fuente: <https://www.onug.net/blog/5g-network-slicing-and-enterprise-networking/>

- 1 Objetivos y Motivación
- 2 Introducción a SDN**
- 3 Descripción de OpenFlow
- 4 Hipervisor OpenFlow
- 5 Implementación
- 6 Conclusiones

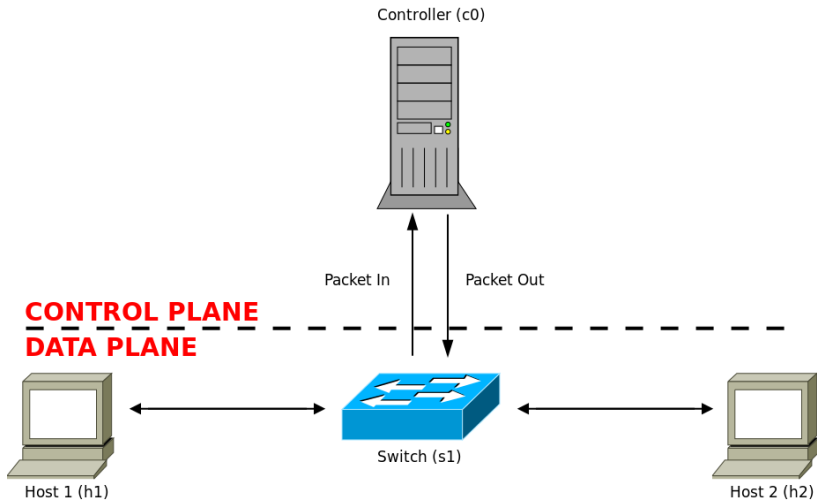
## ¿Qué es una red SDN?

En esencia, el concepto de SDN consiste en la separación entre el plano de control y el plano de datos.

Analogía con *Cloud Computing*.



# Introducción a SDN



Ventajas de una red SDN frente a una tradicional.

- Flexibilidad y adaptabilidad.
- Visión general de la red.
- Monitorización.
- *Hardware* homogéneo.
- Mantenimiento y configuración.

- 1 Objetivos y Motivación
- 2 Introducción a SDN
- 3 Descripción de OpenFlow
- 4 Hipervisor OpenFlow
- 5 Implementación
- 6 Conclusiones

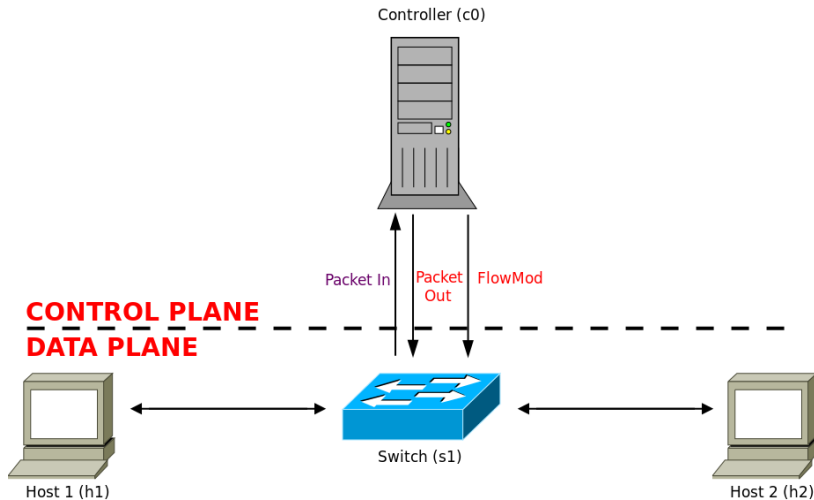
OpenFlow es el protocolo de comunicación *southbound* más utilizado en redes SDN.

## *Comunicación southbound*

Es la comunicación entre los *forwarding devices* y el controlador o controladores.

Los tipos de paquetes más importantes son:

- *Packet\_In*.
- *Packet\_Out*.
- *FlowMod*.



Estructura de una regla de flujo (*flow*) OpenFlow.

## **Matching Fields**

Dirección IP destino

Dirección IP origen

Protocolo de Transporte

Puerto TCP destino

Puerto TCP origen

## **Action**

Enviar por puerto X

Descartar

Packet\_In

Inundar

Estructura de una regla de flujo (*flow*) OpenFlow.

## Matching Fields

Dirección IP destino

Dirección IP origen

Protocolo de Transporte

Puerto TCP destino

Puerto TCP origen

## Action

Enviar por puerto X

Descartar

Packet\_In

Inundar

```
sudo ovs-ofctl add-flow s1 nw_dst=10.0.0.1,actions=output:2
```

Estructura de una regla de flujo (*flow*) OpenFlow.

## Matching Fields

Dirección IP destino

Dirección IP origen

Protocolo de Transporte

Puerto TCP destino

Puerto TCP origen

## Action

Enviar por puerto X

Descartar

Packet\_In

Inundar

```
sudo ovs-ofctl add-flow s1 nw_dst=10.0.0.1,actions=output:2
```



Estructura de una regla de flujo (*flow*) OpenFlow.

## Matching Fields

Dirección IP destino

Dirección IP origen

Protocolo de Transporte

Puerto TCP destino

Puerto TCP origen

## Action

Enviar por puerto X

Descartar

Packet\_In

Inundar

```
sudo ovs-ofctl add-flow s1 nw_dst=10.0.0.1,actions=output:2
```

Estructura de una regla de flujo (*flow*) OpenFlow.

## Matching Fields

Dirección IP destino

Dirección IP origen

Protocolo de Transporte

Puerto TCP destino

Puerto TCP origen

## Action

Enviar por puerto X

Descartar

Packet\_In

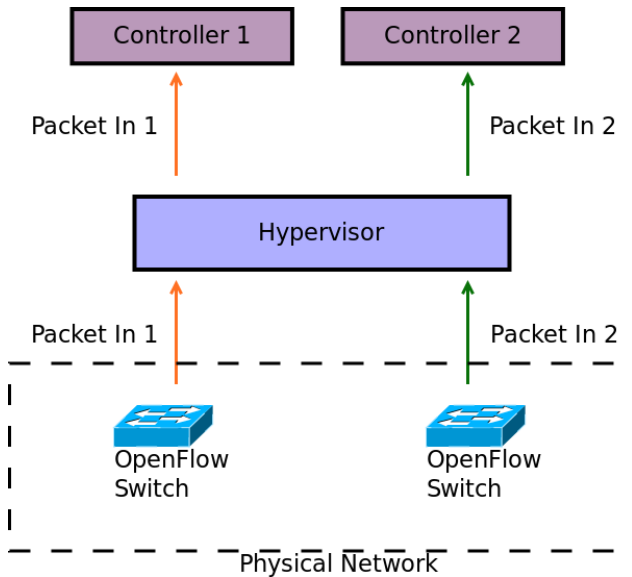
Inundar

```
sudo ovs-ofctl add-flow s1 nw_dst=10.0.0.1,actions=output:2
```

- 1 Objetivos y Motivación
- 2 Introducción a SDN
- 3 Descripción de OpenFlow
- 4 **Hipervisor OpenFlow**
  - **Introducción**
  - Sintaxis
  - Ventajas
- 5 Implementación
- 6 Conclusiones

- Tipo especial de controlador.
- Situado entre los *switches* OpenFlow y los controladores.
- Intercepta mensajes OpenFlow y los distribuye acorde a su configuración.
- Clave para *network slicing*. *Slices*. Múltiples controladores.
- Configuración análoga a reglas OpenFlow.
- Hipervisor *open source*: **FlowVisor**.

# Hipervisor OpenFlow



- 1 Objetivos y Motivación
- 2 Introducción a SDN
- 3 Descripción de OpenFlow
- 4 Hipervisor OpenFlow**
  - Introducción
  - **Sintaxis**
  - Ventajas
- 5 Implementación
- 6 Conclusiones

Regla ejemplo con la sintaxis usada por FlowVisor. Análogo a las reglas OpenFlow.

```
fvctl -n add-flowspace example 1 200 in_port=2 LoRa=6
```

Regla ejemplo con la sintaxis usada por FlowVisor. Análogo a las reglas OpenFlow.

```
fvctl -n add-flowspace example 1 200 in_port=2 LoRa=6
```

**Comando para añadir un flowspace**



Regla ejemplo con la sintaxis usada por FlowVisor. Análogo a las reglas OpenFlow.

```
fvctl -n add-flowspace example 1 200 in_port=2 LoRa=6
```

**Nombre del flowspace**

Regla ejemplo con la sintaxis usada por FlowVisor. Análogo a las reglas OpenFlow.

```
fvctl -n add-flowspace example 1 200 in_port=2 LoRa=6
```

**Identificador del switch (DPID)**

Regla ejemplo con la sintaxis usada por FlowVisor. Análogo a las reglas OpenFlow.

```
fvctl -n add-flowspace example 1 200 in_port=2 LoRa=6
```

**Prioridad de la regla**

Regla ejemplo con la sintaxis usada por FlowVisor. Análogo a las reglas OpenFlow.

```
fvctl -n add-flowspace example 1 200 in_port=2 LoRa=6
```

**Campo que analizar el paquete**

Regla ejemplo con la sintaxis usada por FlowVisor. Análogo a las reglas OpenFlow.

```
fvctl -n add-flowspace example 1 200 in_port=2 LoRa=6
```

**Slice al que asignar el paquete**

Regla ejemplo con la sintaxis usada por FlowVisor. Análogo a las reglas OpenFlow.

```
fvctl -n add-flowspace example 1 200 in_port=2 LoRa=6
```

**Permisos del slice**

- 1 Objetivos y Motivación
- 2 Introducción a SDN
- 3 Descripción de OpenFlow
- 4 Hipervisor OpenFlow**
  - Introducción
  - Sintaxis
  - Ventajas**
- 5 Implementación
- 6 Conclusiones

## Ventajas

- Simplificación de la lógica de los controladores.
- Mantenimiento.
- Adaptabilidad.
- Aislamiento entre *slices*.

## Aplicaciones

- Ingeniería de tráfico.
- Multiplexación de servicios en la misma red física.
- Uso de una red de producción como escenario de pruebas.



- 1 Objetivos y Motivación
- 2 Introducción a SDN
- 3 Descripción de OpenFlow
- 4 Hipervisor OpenFlow
- 5 **Implementación**
  - **Virtualización**
    - TCP Port Slicing
    - Demo TCP Port Slicing
    - IP Address Slicing
    - Demo IP Address Slicing
- 6 Conclusiones

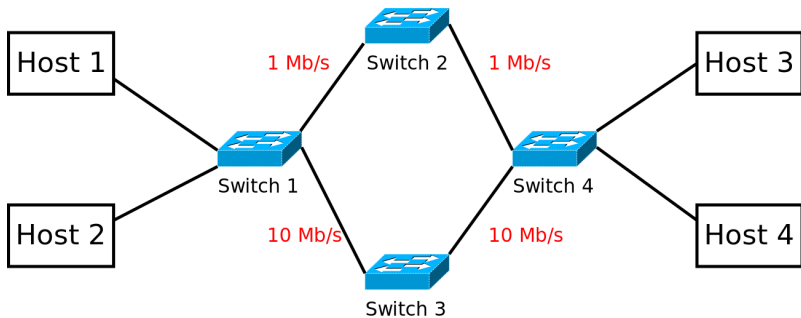
No disponemos de *switches* OpenFlow para montar una red SDN.  
Tenemos que recurrir a emulación.

- Open vSwitch.
- Mininet.
- Iperf.

No disponemos de *switches* OpenFlow para montar una red SDN.  
Tenemos que recurrir a emulación.

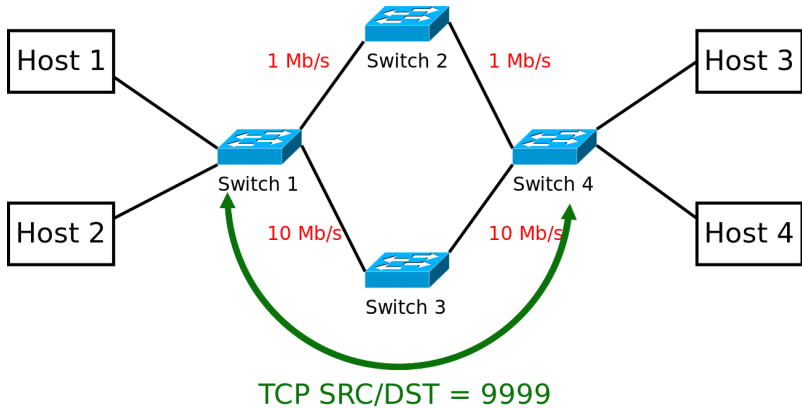
- Open vSwitch.
- Mininet.
- Iperf.

Como controlador OpenFlow usaremos **POX**.



- 1 Objetivos y Motivación
- 2 Introducción a SDN
- 3 Descripción de OpenFlow
- 4 Hipervisor OpenFlow
- 5 **Implementación**
  - Virtualización
  - **TCP Port Slicing**
  - Demo TCP Port Slicing
  - IP Address Slicing
  - Demo IP Address Slicing
- 6 Conclusiones

# TCP Port Slicing



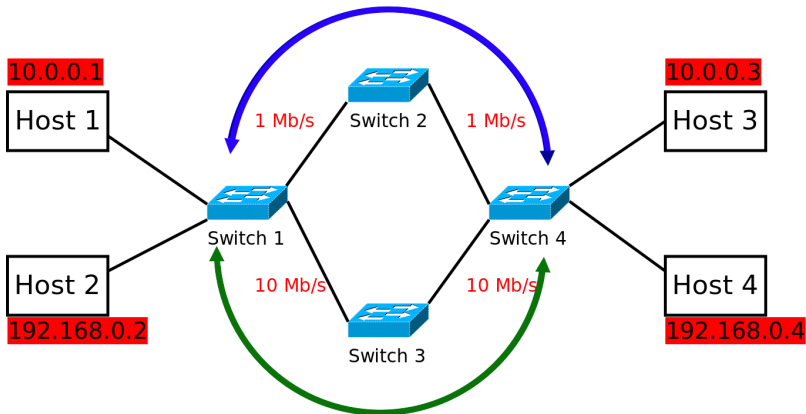
- 1 Objetivos y Motivación
- 2 Introducción a SDN
- 3 Descripción de OpenFlow
- 4 Hipervisor OpenFlow
- 5 **Implementación**
  - Virtualización
  - TCP Port Slicing
  - **Demo TCP Port Slicing**
  - IP Address Slicing
  - Demo IP Address Slicing
- 6 Conclusiones

- 1 Objetivos y Motivación
- 2 Introducción a SDN
- 3 Descripción de OpenFlow
- 4 Hipervisor OpenFlow
- 5 Implementación**
  - Virtualización
  - TCP Port Slicing
  - Demo TCP Port Slicing
  - IP Address Slicing**
  - Demo IP Address Slicing
- 6 Conclusiones



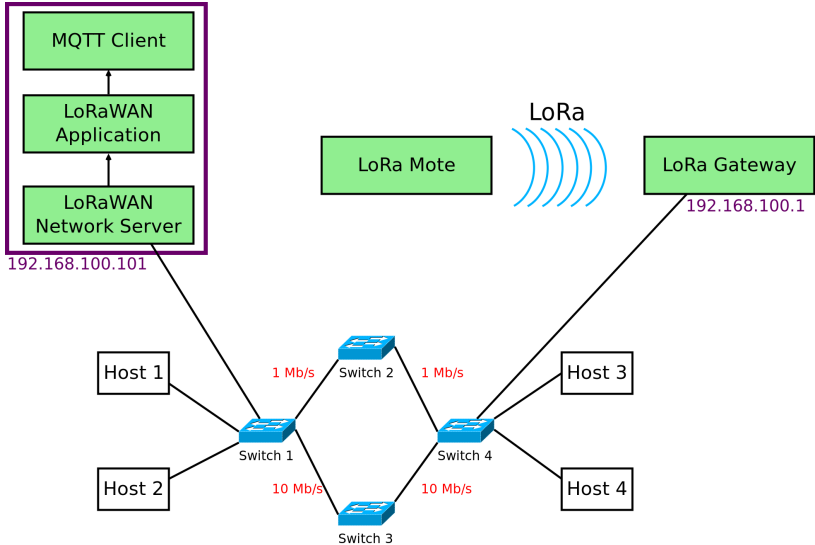
# IP Address Slicing

SRC/DST IP ADDRESS = 192.168.0.0/16



SRC/DST IP ADDRESS = 10.0.0.0/16

# IP Address Slicing



# IP Address Slicing



- 1 Objetivos y Motivación
- 2 Introducción a SDN
- 3 Descripción de OpenFlow
- 4 Hipervisor OpenFlow
- 5 **Implementación**
  - Virtualización
  - TCP Port Slicing
  - Demo TCP Port Slicing
  - IP Address Slicing
  - **Demo IP Address Slicing**
- 6 Conclusiones

- 1 Objetivos y Motivación
- 2 Introducción a SDN
- 3 Descripción de OpenFlow
- 4 Hipervisor OpenFlow
- 5 Implementación
- 6 Conclusiones

## ¿Hemos cumplido con los objetivos?

- Hemos entendido las limitaciones de redes tradicionales.
- Familiarización con redes SDN y *network slicing*.
- Particionado de red de dos formas diferentes.

## ¿Hemos cumplido con los objetivos?

- Hemos entendido las limitaciones de redes tradicionales.
- Familiarización con redes SDN y *network slicing*.
- Particionado de red de dos formas diferentes.

## Respecto al *network slicing*.

- Solución al tráfico heterogéneo.
- Alternativa a *QoS*.
- Simplifica la configuración del controlador o controladores.
- Aislamiento entre secciones de la misma red.