

# Software Define Networks (SDN)

Ing. José Restaino

Ing. Alvaro Sánchez

## Que promete SDN?

- **Agilidad:** a la hora de desarrollar nuevos servicios, aplicaciones e infraestructura en la red
- **Escalamiento:** Capacidad de automatizar y optimizar nuevos servicios en la red
- **Innovación:** Permitir a las organizaciones brindar nuevos tipos de aplicaciones, servicios y modelos de negocio
- **Disminución de costos:** Disminuir los costos de OpEx (“Operational Expenditures” - gastos operativos) ya que se deberían tener más automatismos y menos operación manual y también disminuir los costos de CapEx (Capital Expenditures – gastos en infraestructura) ya que promete utilizar más software no propietario y hardware abierto.

# Que es SDN?

- Hay muchas definiciones, manejaremos la de la ONF (<https://opennetworking.com/sdn-definition/>)
- Separación del plano de control y del plano de Datos
- Control del comportamiento de la red desde una capa superior
- Partes de la infraestructura
  - Plano de datos: switches programables
  - Plano de Control: Controladores, Aplicaciones, ...

# Ventajas de SDN

- Facilita
  - Coordinación entre dispositivos de red
  - Desarrollo de la red (permite utilizar las aproximaciones convencionales a ingeniería de software para resolver los problemas de redes)

# Plano de control vs plano de datos

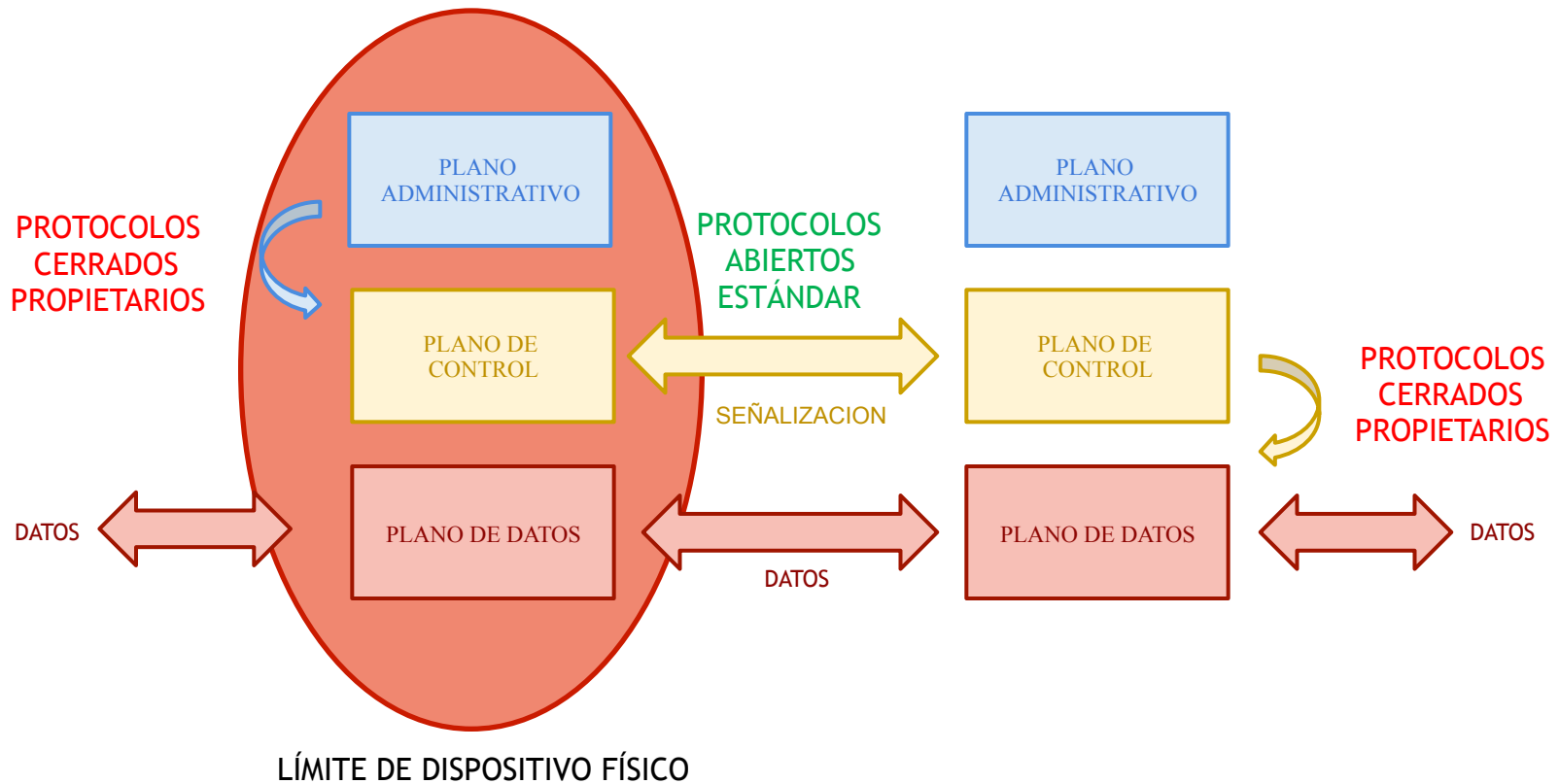
- Plano de control
  - “Cerebro de la red”
  - Puede estar separado de los equipos que reenvían el tráfico
  - Calcula como el tráfico puede ser reenviado
- Plano de datos
  - El hardware tiene que ser programable
  - Debe de aceptar las instrucciones recibidas por el plano de control

# Network Virtualization

- Que es?
  - Representación de una o más topologías en la misma infraestructura
  - Separar la red lógica de la infraestructura

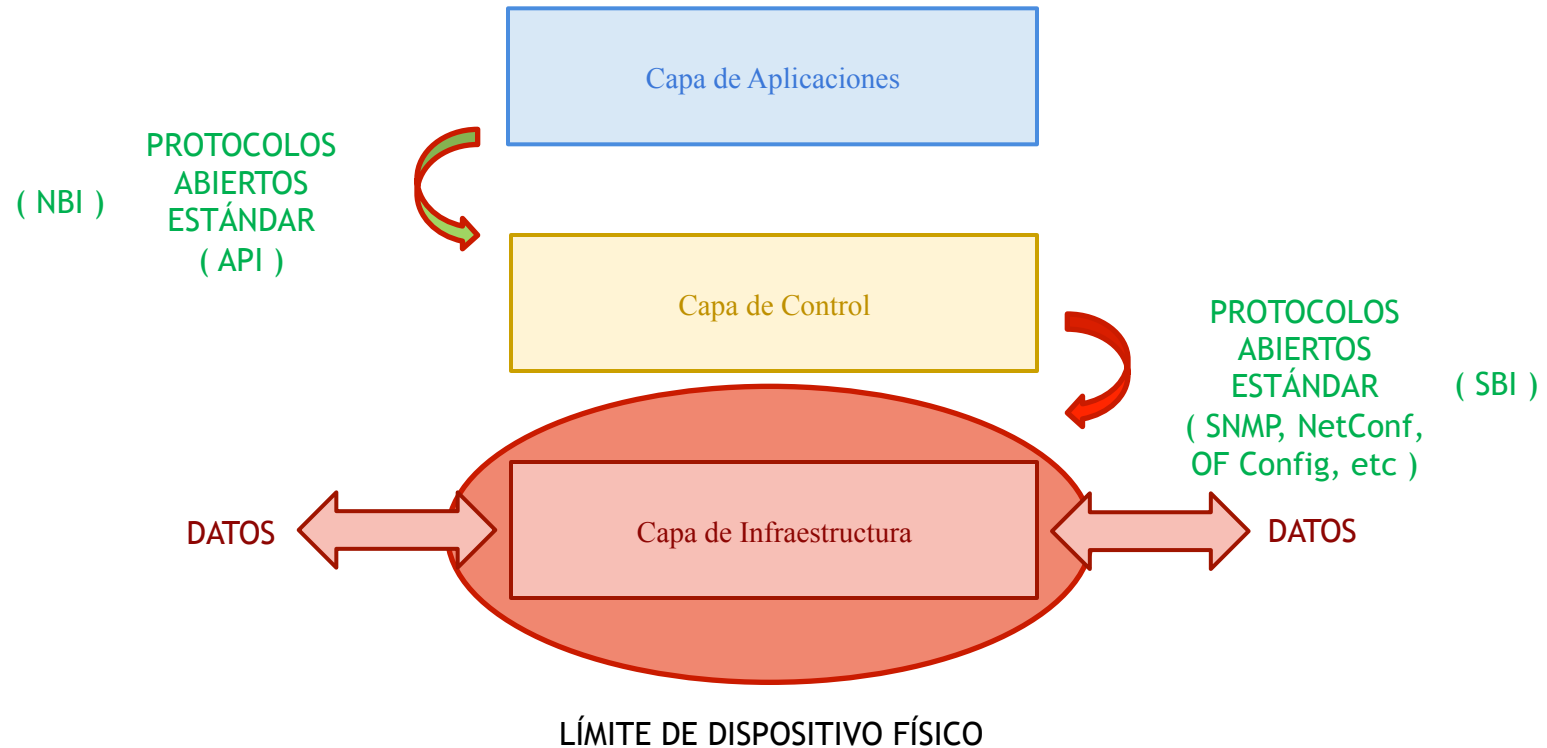
# Arquitectura de SDN

- Sin SDN



# Arquitectura de SDN

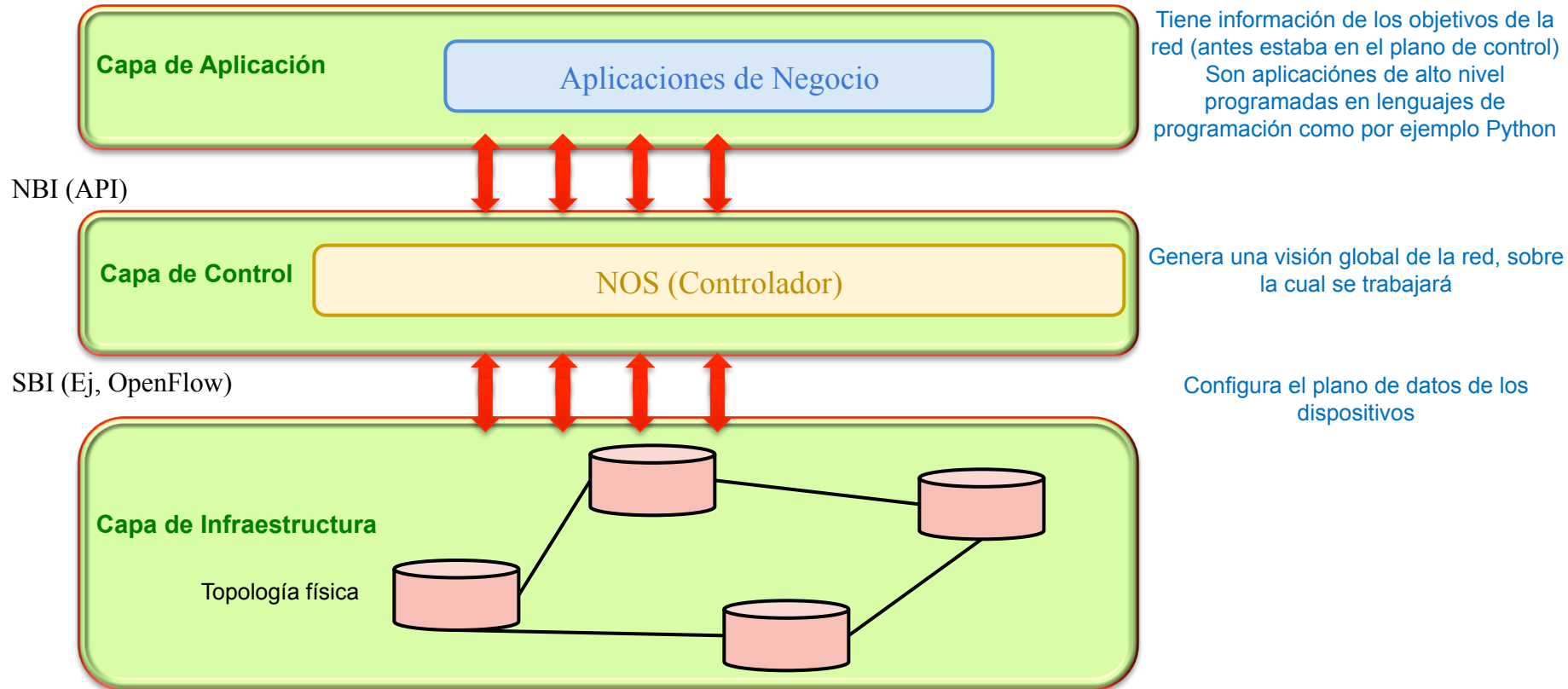
- Con SDN





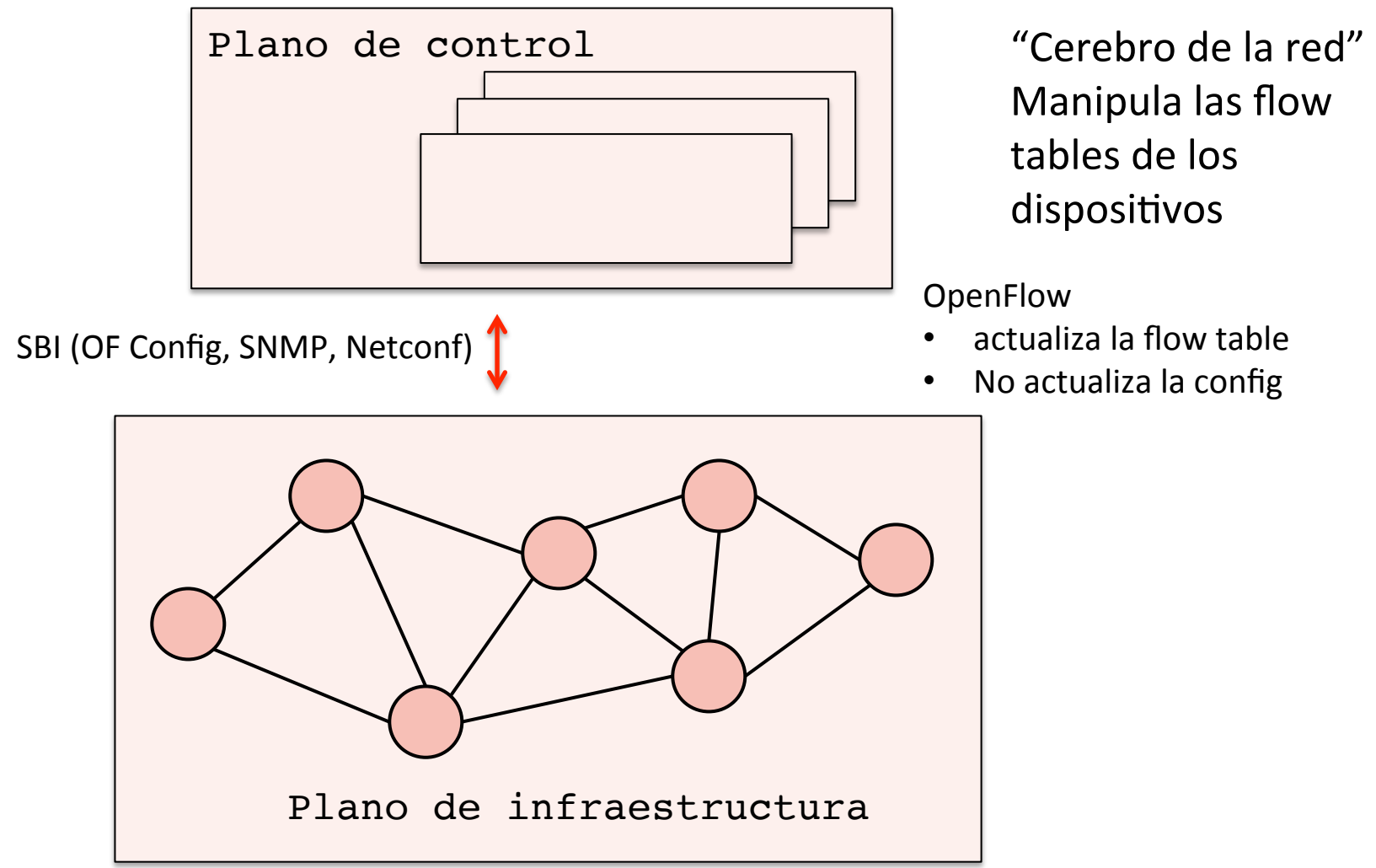
# Arquitectura de SDN

- Que es?
  - Representación de una o más topologías en la misma infraestructura
  - Separar la red lógica de la infraestructura



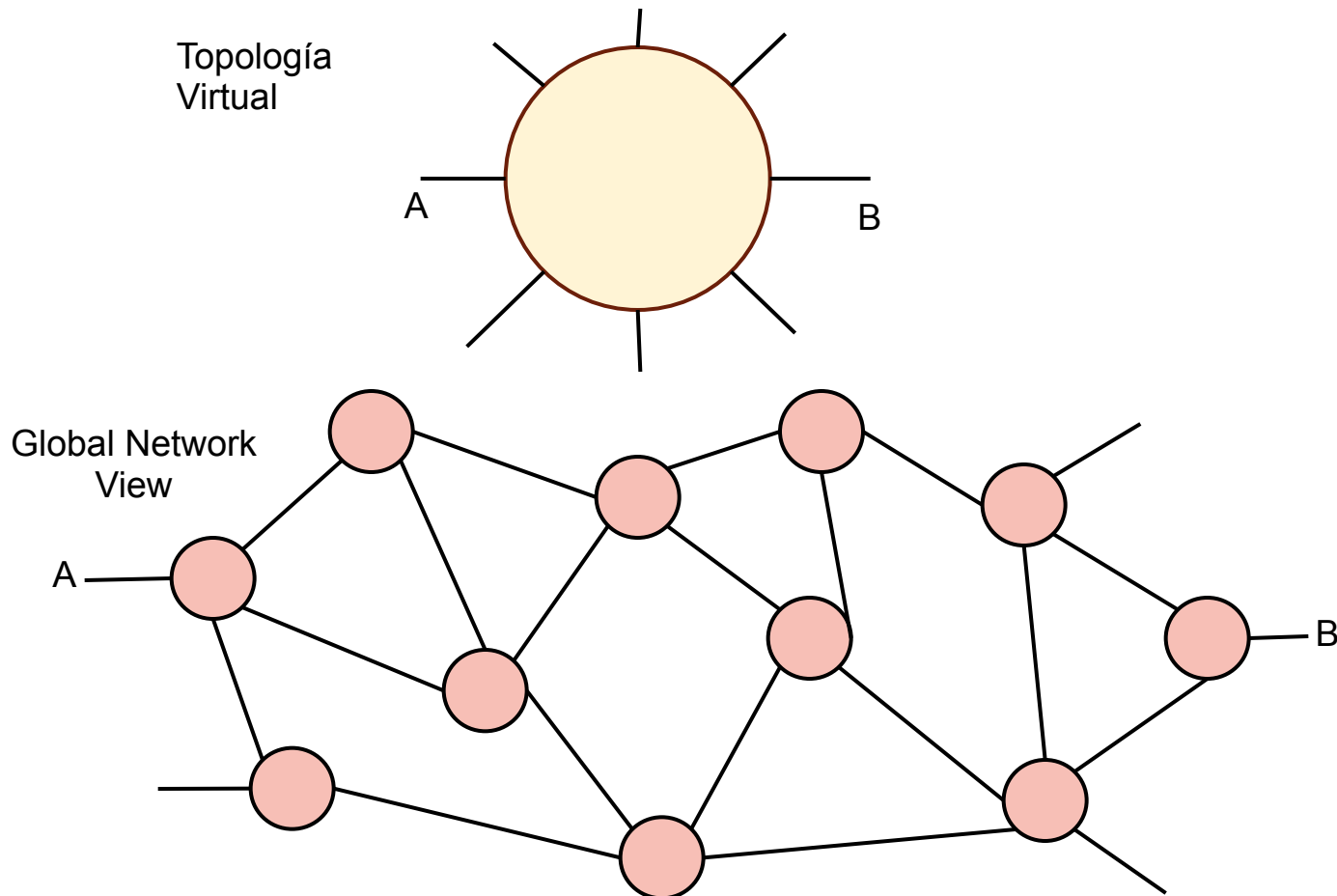
Separación del cerebro vs infraestructura

# Que es SDN?



# Que es SDN?

Dado un requerimiento de ir de por ejemplo poner un filtro para que trafico proveniente de A no llegue a B, lo haremos sobre la topología virtual, independizandonos asi de la topologia



# Algunas definiciones

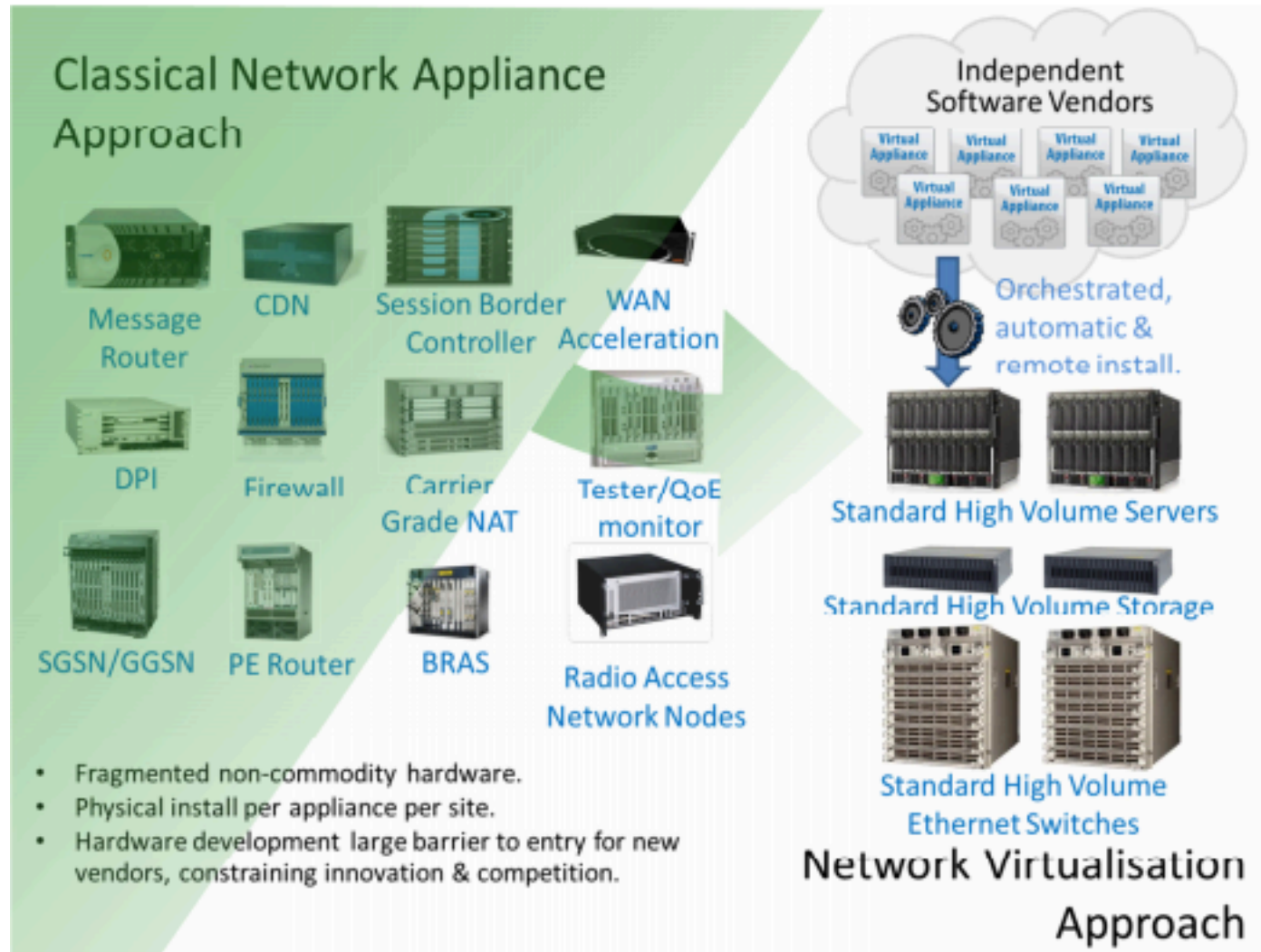
- NBI (North Bound Interface)
  - Interfaces de Alto nivel entre la capa de aplicación y la de control
  - Ejemplo de APIs, REST, NetConf, RestConf, etc
- SBI (South Bound Interface)
  - Interfaces entre la capa de control y la de datos
  - Ejemplo protocolo OpenFlow, SNMP, NetConf, PCEP, BGP-LS, etc

# NFV (Network Function Virtualization)

## Definición

Network Functions Virtualisation aims to transform the way that network operators architect networks by evolving standard IT virtualisation technology to consolidate many network equipment types onto industry standard high volume servers, switches and storage, which could be located in Datacentres, Network Nodes and in the end user premises, as illustrated in Figure 1. It involves the implementation of network functions in software that can run on a range of industry standard server hardware, and that can be moved to, or instantiated in, various locations in the network as required, without the need for installation of new equipment.

# NFV (Network Function Virtualization)



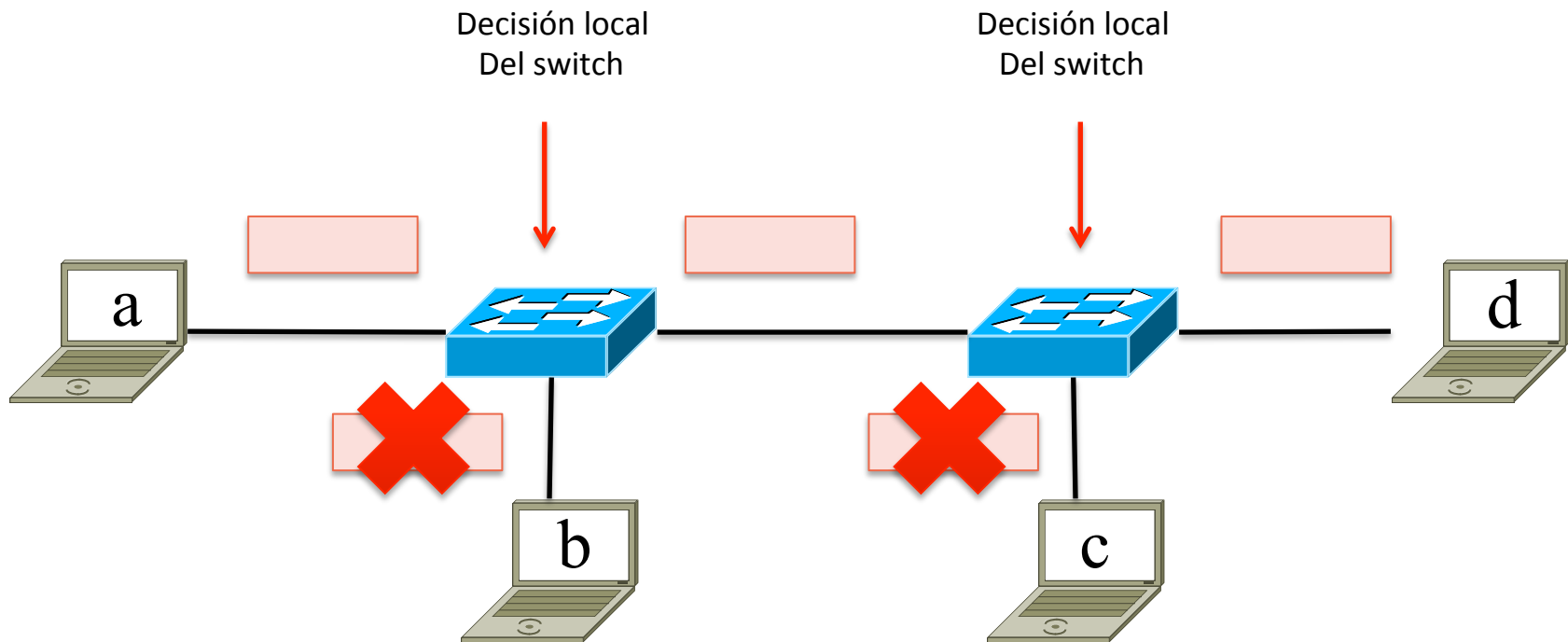
# SD-WAN

## Definición

Pretende utilizar los conceptos de SDN aplicados a los ambitos WAN.

Para interconectar puntos remotos como por ejemplo DataCenters u oficinas, se busca llevar el control de la red a tecnologías en nube utilizando un enfoque de Ingeniería de Software en lugar del enfoque convencional.

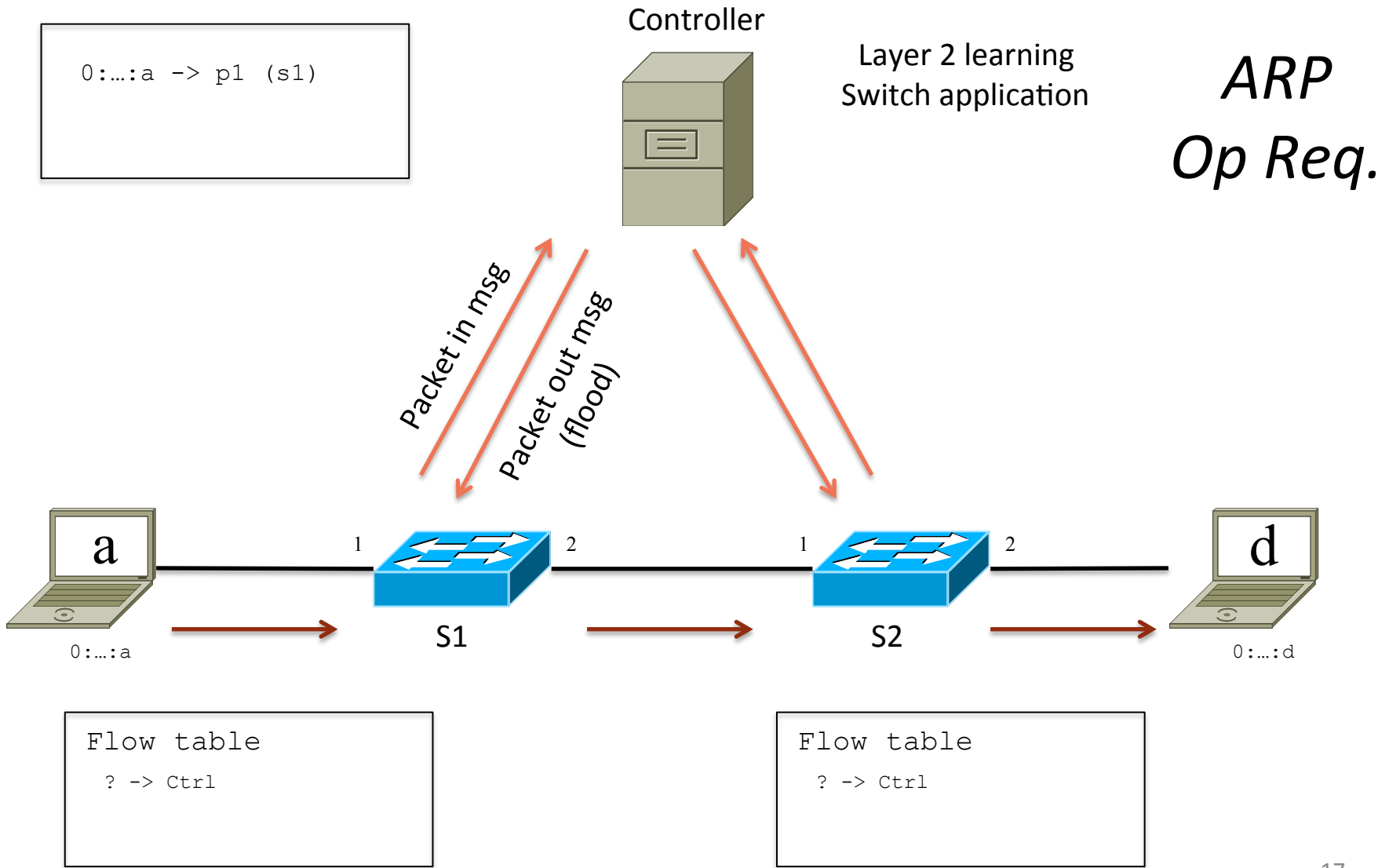
# Switching Tradicional



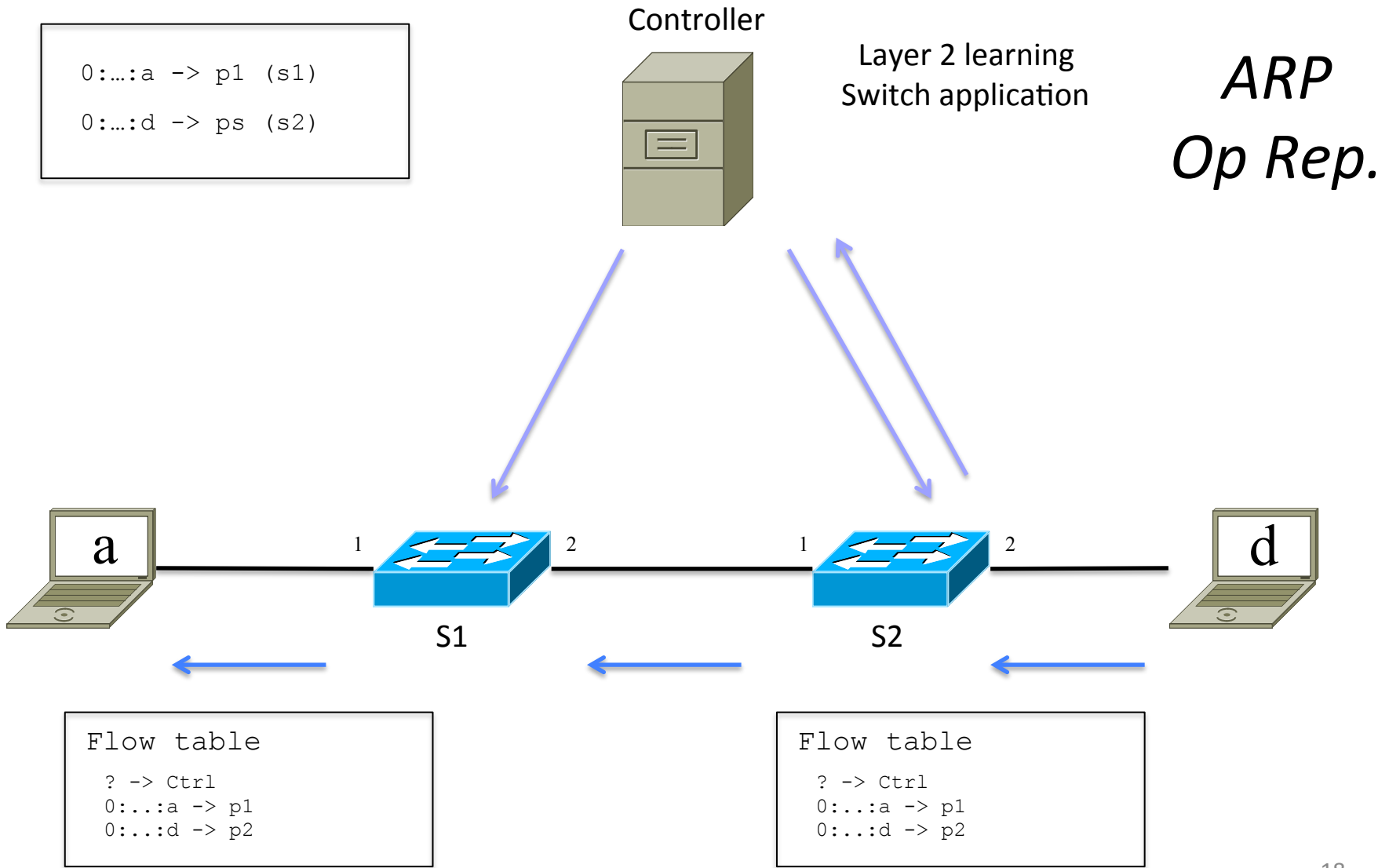
Envío una trama de a a d, los switches toman decisiones locales para decidir el reenvío



# Switching utilizando OpenFlow



# Switching utilizando OpenFlow

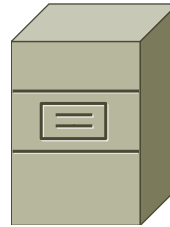


# Switching utilizando OpenFlow

```
0:::a -> p1 (s1)
```

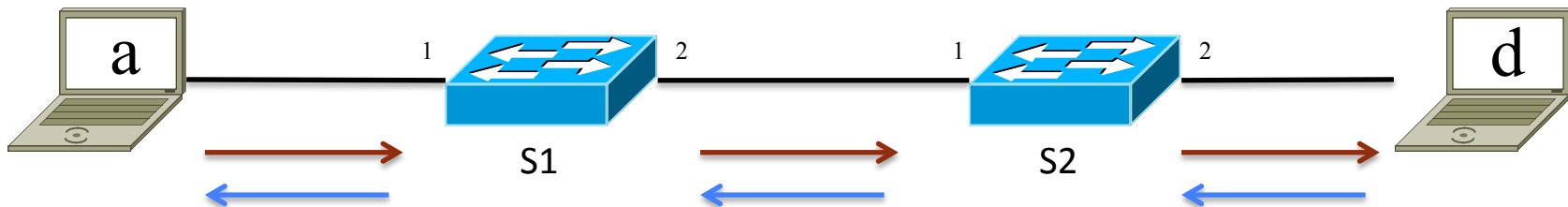
```
0:::d -> ps (s2)
```

Controller



Layer 2 learning  
Switch application

*Envío de tramas,  
Cuando las flow tables  
Tienen información*



Flow table

```
? -> Ctrl
```

```
0:::a -> p1
```

```
0:::d -> p2
```

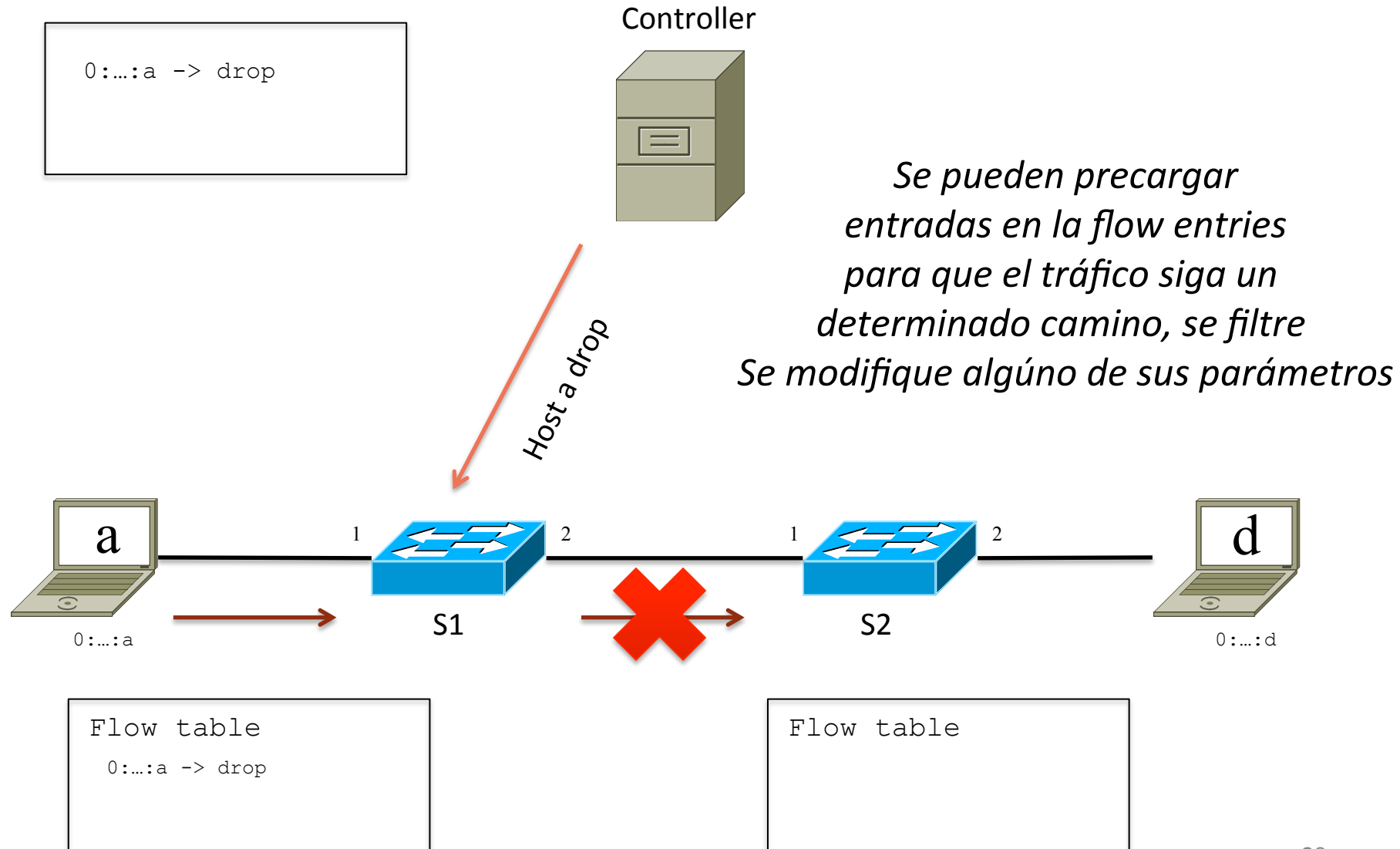
Flow table

```
? -> Ctrl
```

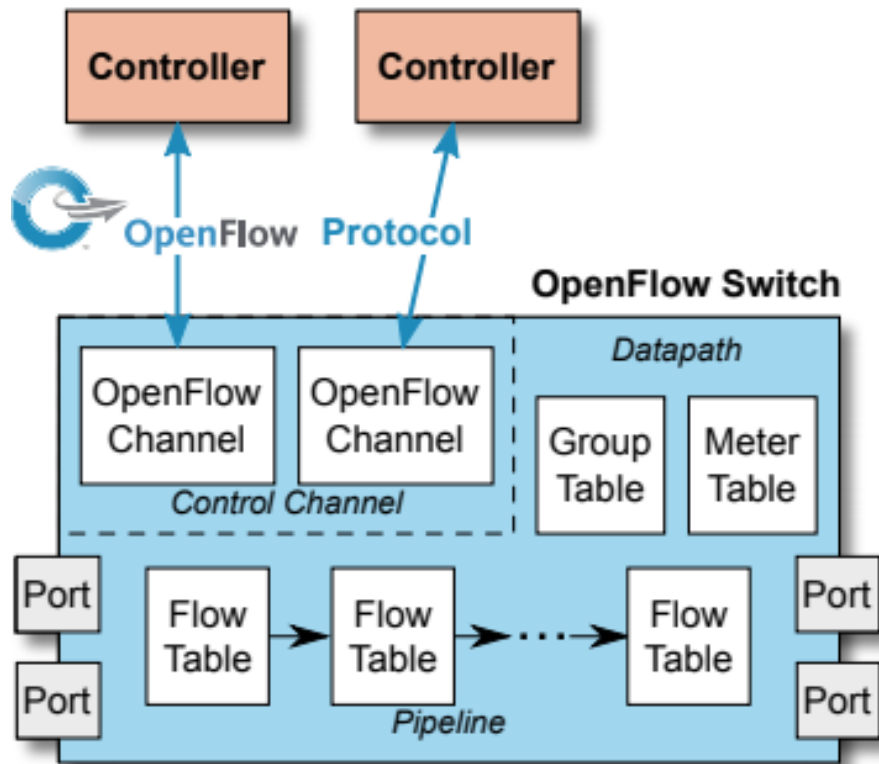
```
0:::a -> p1
```

```
0:::d -> p2
```

# Proactive flow entries



# OpenFlow Switch



**Group Table:** se pueden agrupar puertos en un grupo para implementar multicast y broadcast

**Flow table pipeline:** se permite tener más de una flow table encadenada para brindar más posibilidades de manejo de tráfico

**OpenFlow Channel:** es la interfaz entre el controlador y el OpenFlow Switch

**Meter Table:** En ella, se pueden implementar limitación de BW, QoS para diferentes flujos

# Puertos

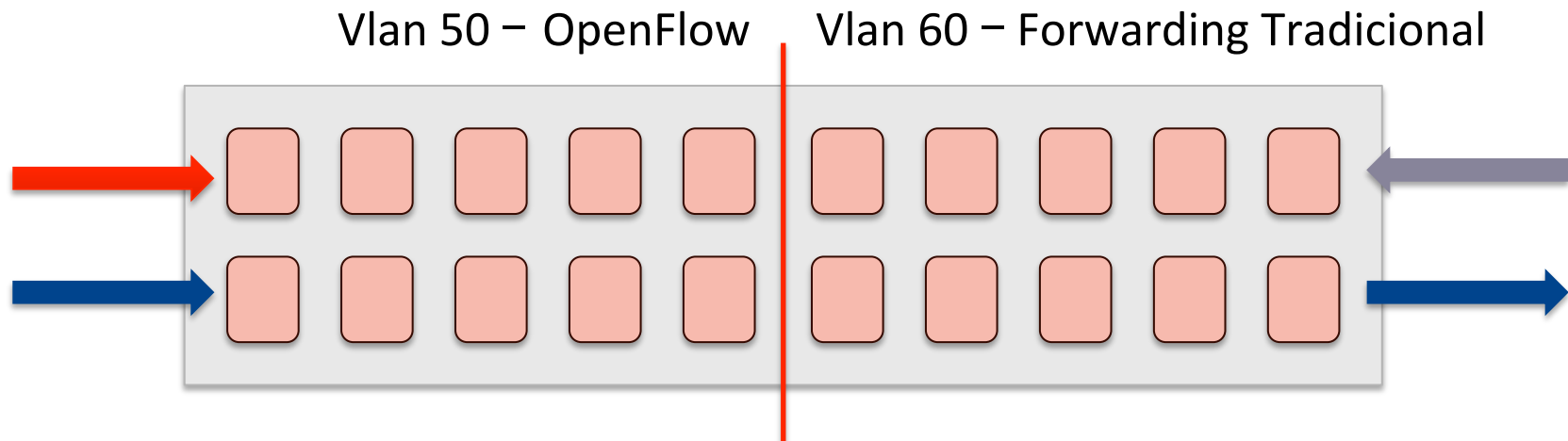
**Puertos:** Son interfaces de red que reenvían paquetes entre un switch OpenFlow y el resto de la red. Los switches OpenFlow se interconectan lógicamente entre ellos utilizando puertos OpenFlow; un paquete puede ser reenviado entre switches OpenFlow si el que envía el paquete lo transmite por un puerto OpenFlow y el receptor lo recibe por un puerto OpenFlow

- **Físicos:** Son puertos que corresponden con los definidos por hardware en el switch
- **Lógicos:** Son puertos que NO se corresponden con los definidos por hardware en el switch, un ejemplo de estos son las interfaces Tunnel o los LAG
- **Reservados:** Los puertos reservados estan predefinidos y cumplen con una función especial; ejemplos:
  - ALL: Representa a todos los puertos, si se utiliza el reenvío se va a realizar por todos los paquetes menos por el que se recibió
  - CONTROLLER: representa el puerto de interconexión con el Controlador
  - NORMAL: Se utiliza para indicar que el reenvío debe realizarse de la forma tradicional

# Switches OpenFlow only Vs OpenFlow Hybrid

- **OpenFlow Only:** Son aquellos que no tienen puertos del tipo NORMAL visto anteriormente, con lo cual no soportan el envío de paquetes utilizando la inteligencia local del switch.
- **OpenFlow Hybrid:** Los mismos tienen la capacidad de utilizar OpenFlow, pero también tienen la capacidad de reenviar paquetes utilizando otros mecanismos. Estos switches pueden dividir en vlans los puertos para determinar cuáles utilizan cada tipo de funcionamiento

# Switches OpenFlow only Vs OpenFlow Hybrid



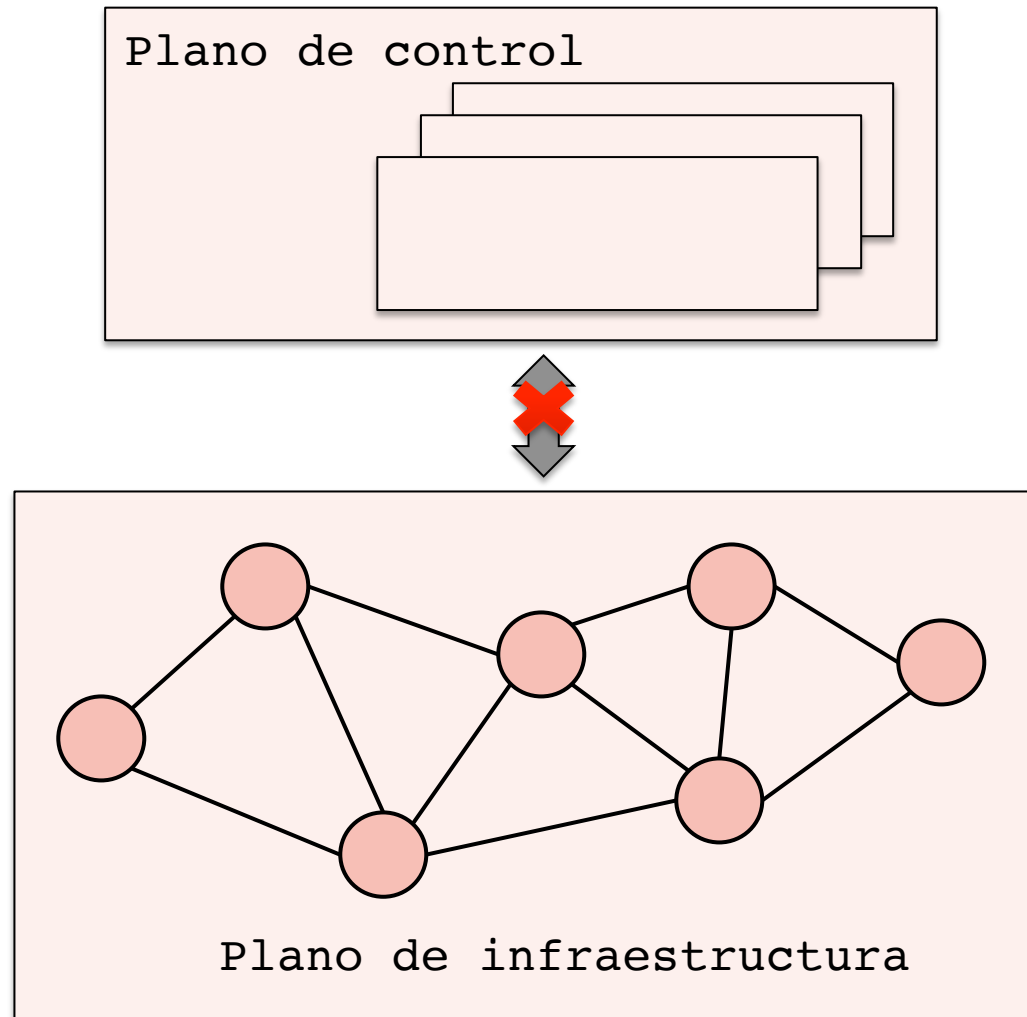
- Si el tráfico ingresa por un puerto de la vlan 50 se reenvía utilizando OF
- Si el tráfico ingresa por un puerto de la vlan 60, se reenvía utilizando el forwarding tradicional
- Es posible enviar tráfico desde un puerto de la vlan 50 a uno de la 60 utilizando el puerto NORMAL

Flow table

0::...:a -> Output Normal Port



# Que ocurre si cae el Controlador



# Que ocurre si cae el Controlador

- ***Fail secure mode***
  - Las entradas en las tablas de flow de los switches van a permanecer hasta que las mismas den timeout.
  - Si se recibe una trama la cual no tiene un match en las tablas de flow, el tráfico será descartado ya que no puede consultar al controlador sobre la manera de reenviar dicho tráfico.
- ***Fail standalone mode***
  - Los switches comienzan a procesar todo hacia el NORMAL port, con lo cual el funcionamiento será igual a el de un switch convencional.

El uso de un modo u otro dependerá de la implementación y la configuración, el más habitual es el Fail Secure mode

# Implementación de OpenFlow (Dudas)

- ***Debemos encender OpenFlow en todos lados?***
  - No, utilizando Hybrid OpenFlow switches, podemos encenderlo solo donde lo queramos y para las vlans que sean necesarias
- ***Puedo utilizar Spanning Tree con OpenFlow***
  - Si se puede utilizar Spanning Tree, el controlador es capaz de tomar la información de SpanningTree utilizando OpenFlow y bloquear los puertos que SPT indique
- ***Puedo tener más de un controlador en mi red?***
  - OpenFlow no lo especifica en su estandar, pero en las implementaciones suele ser verdadero, puedo tener tantos controladores como quiera y los switches pueden tener más de un controlador