

( 75.43 ) Introducción a los sistemas distribuidos

Grupo A10

Trabajo Práctico

Segundo cuatrimestre - 2023

<b>Alumno</b>	<b>Padrón</b>	<b>Mail</b>
Agustín D. Palmeira	90.856	apalmeira@fi.uba.ar
Alejandro Paff	103376	apaff@fi.uba.ar
Alejo Acevedo	99146	aacevedo@fi.uba.ar
Federico Cavazzoli	98533	fcavazzoli@fi.uba.ar

# Índice

<b>Introducción.....</b>	<b>3</b>
<b>Hipótesis y supuestos.....</b>	<b>3</b>
<b>Implementación.....</b>	<b>4</b>
Topología.....	4
Como correr la topologia.....	4
Cómo utilizar mininet.....	4
<b>Dificultades encontradas.....</b>	<b>5</b>
<b>Capturas wireshark.....</b>	<b>5</b>
<b>Preguntas a responder.....</b>	<b>5</b>
1. ¿Cuál es la diferencia entre un Switch y un router? ¿Qué tienen en común?.....	5
2. ¿Cuál es la diferencia entre un Switch convencional y un Switch OpenFlow?.....	5
3. ¿Se pueden reemplazar todos los routers de la Internet por Switches OpenFlow? Piense en el escenario interASes para elaborar su respuesta.....	6
<b>Conclusiones.....</b>	<b>7</b>

# Introducción

El propósito de este informe es proporcionar una descripción detallada del proceso de desarrollo del trabajo práctico, a lo largo de este documento, se presentarán los aspectos clave del proyecto en el contexto del aprendizaje relacionado con sistemas distribuidos, dicho trabajo práctico tiene como objetivo construir una topología con Mininet donde se utilizará Openflow para poder implementar un Firewall a nivel de capa de enlace..

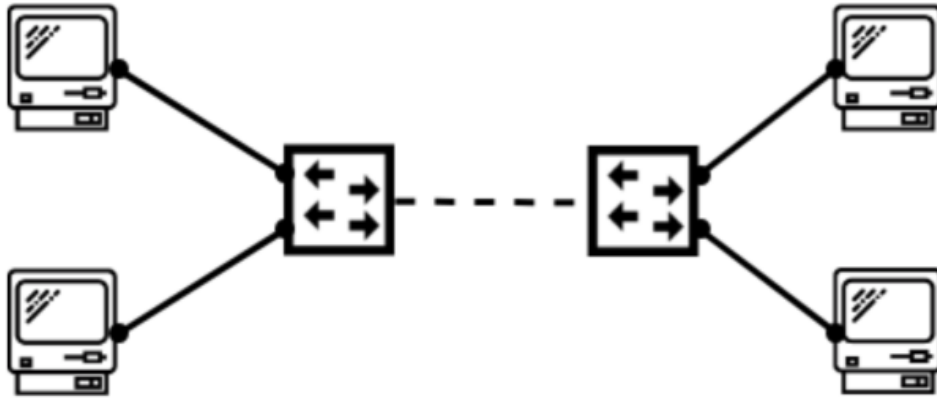
## Hipótesis y supuestos

A continuación se detallan las hipótesis más relevantes asumidas en el trabajo:

- Existe un archivo JSON con la configuración del firewall
- Dicho archivo en caso de no especificar el protocolo en una regla asumirá que se trata de ambos UDP y TCP
- Se utiliza IPv4

# Implementación

En esta sección, se proporcionará una descripción detallada de la implementación general realizada. La topología pedida fué la siguiente



Con el controlador modificado para que funcione como un Firewall

## Topología

La topología tendrá siempre dos hosts en los extremos (no customizable su cantidad). También dispondrá de una cantidad de switches variables, definida por parámetro, formando una cadena entre ellos.

### Prerrequisitos de instalacion en Ubuntu para mininet

```
> sudo apt-get -y install xterm
```

### Instalación de mininet en Ubuntu

```
> sudo apt-get install mininet
```

## Como correr la topologia

Desde la consola ejecutar (por ejemplo con 2 switches):

```
> python3 create_topology_from_scratch.py 2
```

El parámetro es referido a la cantidad de switches, y tiene que ser un número positivo, mayor a 1 y menor o igual a 10. Debe ser mayor a 1 porque como mínimo debe existir un switch entre los hosts extremos.

## Cómo utilizar mininet

Para correr la consola de un host:

```
> h1 xterm &
```

### **Logs creados**

Se creará un archivo con la cantidad de switches definidos (pasados como parámetros al momento de la creación de la topología).

### **Troubleshoot**

En caso de que no corra mininet con un error relacionado a los controllers, intentar con este comando:

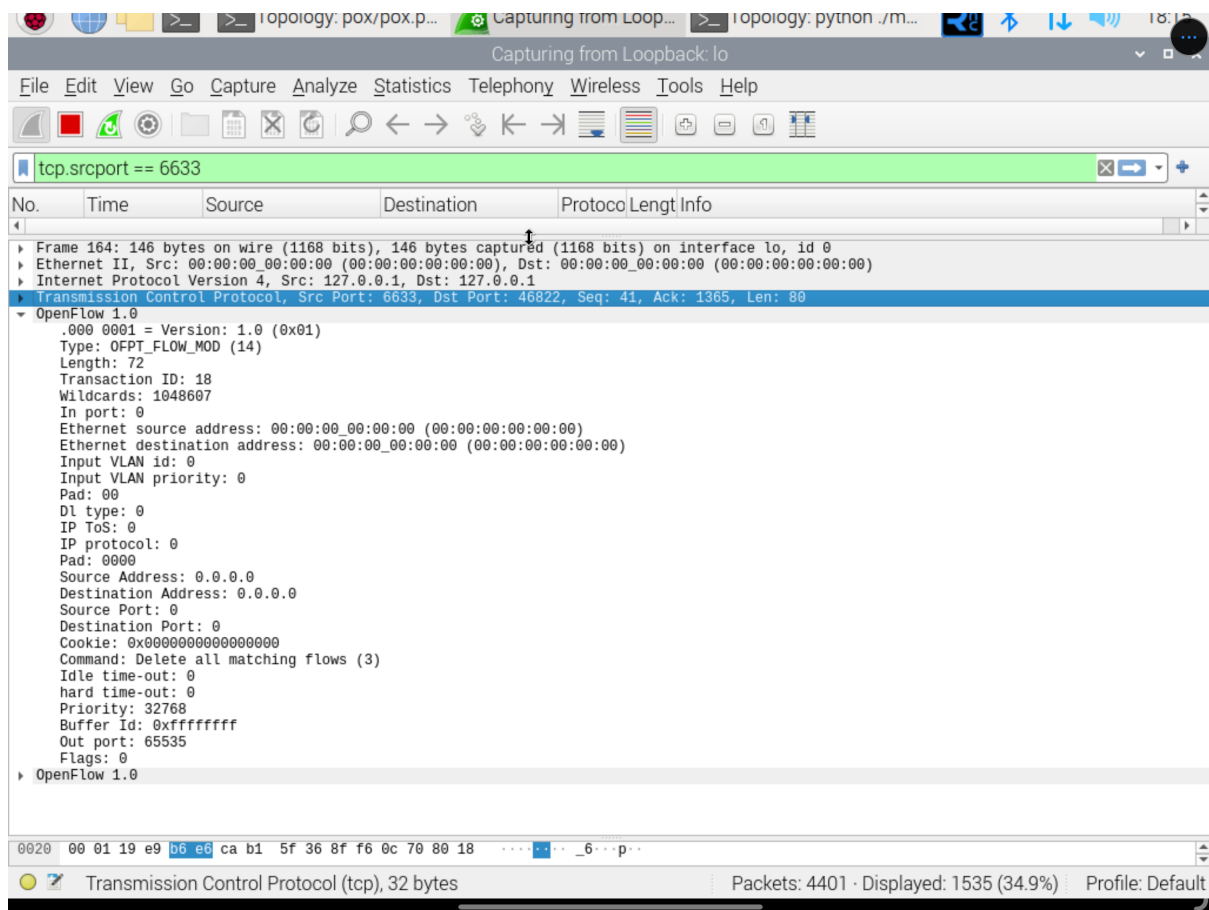
```
> sudo fuser -k 6653/tcp
```

## **Dificultades encontradas**

- Tanto la documentación de Mininet como de Pox no se encuentran bien detalladas y puede ser difícil para el desarrollador implementarlas en su proyecto.

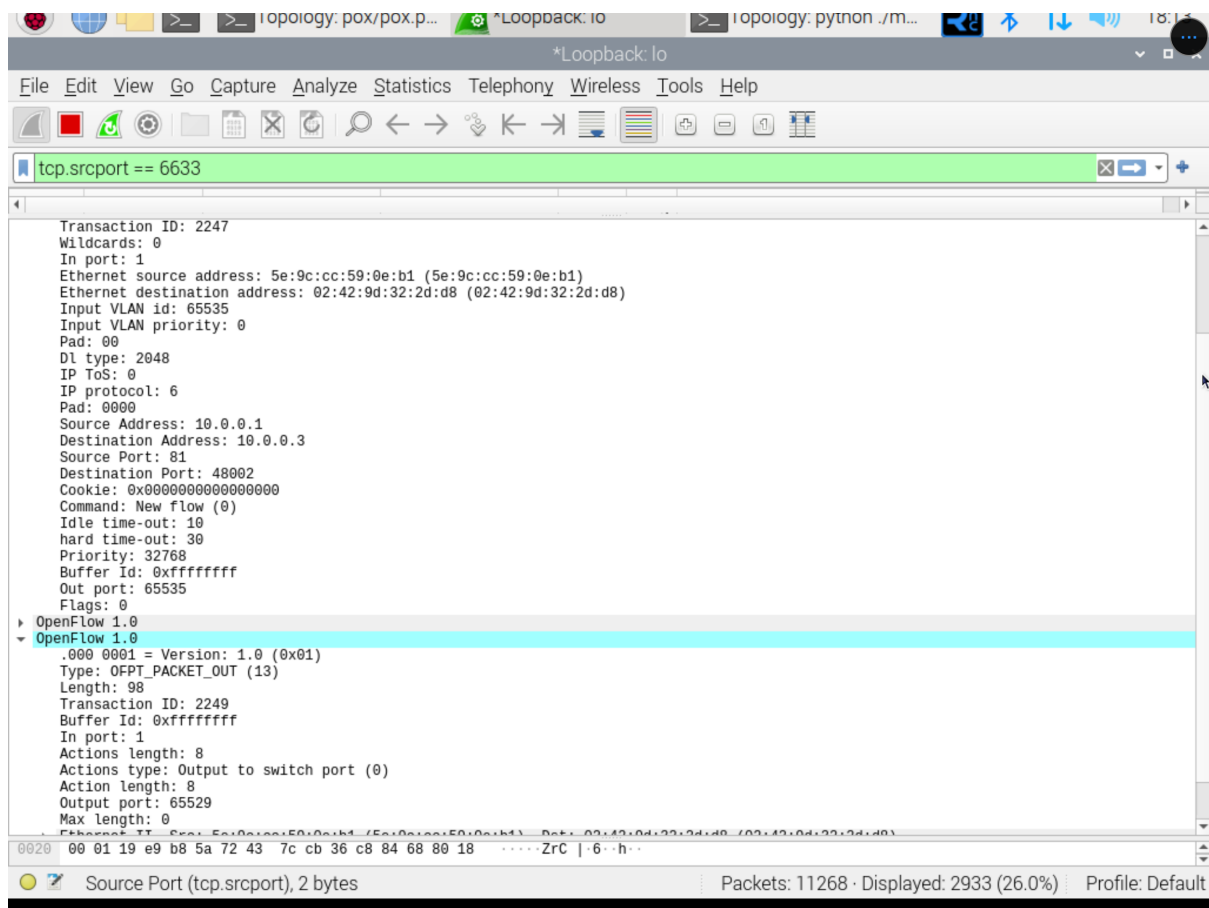
## **Capturas wireshark**

Captura del primer paquete enviado desde el controller al switch para la limpieza de la tabla



En este podemos ver que el comando enviado es delete all matchig flow

Captura de uno de los paquetes enviados desde el controller al switch para enseñarle como rutear los paquetes enviados desde un host al otro



En este paquete podemos ver que se le dice al switch que los paquetes que tengan source address 10.0.0.1 y destination address 10.0.0.3 y puerto de origen 81 y puerto de destino 48092 se user cierto puerto del switch

Captura donde podemos ver como se instalan las reglas desde el controller a uno de los switch

#### OpenFlow 1.0

```
.000 0001 = Version: 1.0 (0x01)
Type: OFPT_FLOW_MOD (14)
Length: 72
Transaction ID: 51
Wildcards: 3678287
In port: 0
Ethernet source address: 00:00:00_00:00:00 (00:00:00:00:00:00)
Ethernet destination address: 00:00:00_00:00:00 (00:00:00:00:00:00)
Input VLAN id: 0
Input VLAN priority: 0
Pad: 00
DI type: 2048
IP ToS: 0
IP protocol: 17
```

Pad: 0000  
Source Address: 0.0.0.0  
Destination Address: 0.0.0.0  
Source Port: 0  
Destination Port: 80  
Cookie: 0x0000000000000000  
Command: New flow (0)  
Idle time-out: 0  
hard time-out: 0  
Priority: 32768  
Buffer Id: 0xffffffff  
Out port: 65535  
Flags: 0

OpenFlow 1.0

.000 0001 = Version: 1.0 (0x01)  
Type: OFPT\_FLOW\_MOD (14)  
Length: 72  
Transaction ID: 52  
Wildcards: 3153999  
In port: 0  
Ethernet source address: 00:00:00\_00:00:00 (00:00:00:00:00:00)  
Ethernet destination address: 00:00:00\_00:00:00 (00:00:00:00:00:00)  
Input VLAN id: 0  
Input VLAN priority: 0  
Pad: 00  
DI type: 2048  
IP ToS: 0  
IP protocol: 6  
Pad: 0000  
Source Address: 0.0.0.0  
Destination Address: 10.0.0.1  
Source Port: 0  
Destination Port: 5001  
Cookie: 0x0000000000000000  
Command: New flow (0)  
Idle time-out: 0  
hard time-out: 0  
Priority: 32768  
Buffer Id: 0xffffffff  
Out port: 65535  
Flags: 0

OpenFlow 1.0

.000 0001 = Version: 1.0 (0x01)  
Type: OFPT\_FLOW\_MOD (14)  
Length: 72  
Transaction ID: 53  
Wildcards: 3153999  
In port: 0



Ethernet source address: 00:00:00\_00:00:00 (00:00:00:00:00:00)  
Ethernet destination address: 00:00:00\_00:00:00 (00:00:00:00:00:00)  
Input VLAN id: 0  
Input VLAN priority: 0  
Pad: 00  
DI type: 2048  
IP ToS: 0  
IP protocol: 17  
Pad: 0000  
Source Address: 0.0.0.0  
Destination Address: 10.0.0.1  
Source Port: 0  
Destination Port: 5001  
Cookie: 0x0000000000000000  
Command: New flow (0)  
Idle time-out: 0  
hard time-out: 0  
Priority: 32768  
Buffer Id: 0xffffffff  
Out port: 65535  
Flags: 0

#### OpenFlow 1.0

.000 0001 = Version: 1.0 (0x01)  
Type: OFPT\_FLOW\_MOD (14)  
Length: 72  
Transaction ID: 54  
Wildcards: 3145935  
In port: 0  
Ethernet source address: 00:00:00\_00:00:00 (00:00:00:00:00:00)  
Ethernet destination address: 00:00:00\_00:00:00 (00:00:00:00:00:00)  
Input VLAN id: 0  
Input VLAN priority: 0  
Pad: 00  
DI type: 2048  
IP ToS: 0  
IP protocol: 6  
Pad: 0000  
Source Address: 10.0.0.1  
Destination Address: 10.0.0.4  
Source Port: 0  
Destination Port: 0  
Cookie: 0x0000000000000000  
Command: New flow (0)  
Idle time-out: 0  
hard time-out: 0  
Priority: 32768  
Buffer Id: 0xffffffff  
Out port: 65535

Flags: 0  
OpenFlow 1.0  
.000 0001 = Version: 1.0 (0x01)  
Type: OFPT\_FLOW\_MOD (14)  
Length: 72  
Transaction ID: 55  
Wildcards: 3145935  
In port: 0  
Ethernet source address: 00:00:00\_00:00:00 (00:00:00:00:00:00)  
Ethernet destination address: 00:00:00\_00:00:00 (00:00:00:00:00:00)  
Input VLAN id: 0  
Input VLAN priority: 0  
Pad: 00  
DI type: 2048  
IP ToS: 0  
IP protocol: 17  
Pad: 0000  
Source Address: 10.0.0.1  
Destination Address: 10.0.0.4  
Source Port: 0  
Destination Port: 0  
Cookie: 0x0000000000000000  
Command: New flow (0)  
Idle time-out: 0  
hard time-out: 0  
Priority: 32768  
Buffer Id: 0xffffffff  
Out port: 65535  
Flags: 0

OpenFlow 1.0  
.000 0001 = Version: 1.0 (0x01)  
Type: OFPT\_FLOW\_MOD (14)  
Length: 72  
Transaction ID: 56  
Wildcards: 3145935  
In port: 0  
Ethernet source address: 00:00:00\_00:00:00 (00:00:00:00:00:00)  
Ethernet destination address: 00:00:00\_00:00:00 (00:00:00:00:00:00)  
Input VLAN id: 0  
Input VLAN priority: 0  
Pad: 00  
DI type: 2048  
IP ToS: 0  
IP protocol: 6  
Pad: 0000  
Source Address: 10.0.0.4  
Destination Address: 10.0.0.1  
Source Port: 0

Destination Port: 0  
Cookie: 0x0000000000000000  
Command: New flow (0)  
Idle time-out: 0  
hard time-out: 0  
Priority: 32768  
Buffer Id: 0xffffffff  
Out port: 65535  
Flags: 0

OpenFlow 1.0

.000 0001 = Version: 1.0 (0x01)  
Type: OFPT\_FLOW\_MOD (14)  
Length: 72  
Transaction ID: 57  
Wildcards: 3145935  
In port: 0  
Ethernet source address: 00:00:00\_00:00:00 (00:00:00:00:00:00)  
Ethernet destination address: 00:00:00\_00:00:00 (00:00:00:00:00:00)  
Input VLAN id: 0  
Input VLAN priority: 0  
Pad: 00  
DI type: 2048  
IP ToS: 0  
IP protocol: 17  
Pad: 0000  
Source Address: 10.0.0.4  
Destination Address: 10.0.0.1  
Source Port: 0  
Destination Port: 0  
Cookie: 0x0000000000000000  
Command: New flow (0)  
Idle time-out: 0  
hard time-out: 0  
Priority: 32768  
Buffer Id: 0xffffffff  
Out port: 65535  
Flags: 0

OpenFlow 1.0

.000 0001 = Version: 1.0 (0x01)  
Type: OFPT\_PACKET\_OUT (13)  
Length: 110  
Transaction ID: 58  
Buffer Id: 0xffffffff  
In port: 1  
Actions length: 8  
Actions type: Output to switch port (0)  
Action length: 8  
Output port: 65531

Max length: 0  
Ethernet II, Src: 2a:e9:45:8d:f7:dd (2a:e9:45:8d:f7:dd), Dst: IPv6mcast\_ff:8d:f7:dd  
(33:33:ff:8d:f7:dd)  
Internet Protocol Version 6, Src: ::, Dst: ff02::1:ff8d:f7dd  
Internet Control Message Protocol v6

Aca podemos ver cada una de las reglas por ejemplo en el primero vemos que especifica el puerto 80 como destino. todos estas reglas podemos ver que se setea que los paquetes sean dropeds

## Preguntas a responder

### 1. ¿Cuál es la diferencia entre un Switch y un router? ¿Qué tienen en común?

Un switch conecta varios dispositivos en una red local, trabaja en la capa de enlace y transmite los paquetes a partir de la dirección MAC, por lo tanto en su tabla necesita una entrada por dispositivo. No requieren ninguna configuración y tienen mejor rendimiento en redes con pocos hosts

En cambio un router conecta diferentes redes, trabaja en la capa de red y necesita una IP para poder funcionar, reconoce los distintos dispositivos en base a su dirección IP la cual sigue una estructura jerárquica. Éste conecta muchos switches con sus respectivas redes creando así una red aún más amplia. Requieren mayor configuración y proveen uso mas inteligente de las rutas entre los hosts

Ambos permiten conectar dispositivos y tienen el trabajo de enviar paquetes e información entre ellos asegurandose (en lo posible) de que lleguen al lugar correcto

### 2. ¿Cuál es la diferencia entre un Switch convencional y un Switch OpenFlow?

En un Switch convencional el plano de datos y el plano de control están un mismo dispositivo. Los switches convencionales no son programables en términos de cómo toman decisiones de reenvío

Un switch OpenFlow separa el plano de datos (por hardware) y de control (por software) y ambos se comunican con el protocolo OpenFlow en base a flujos, ésta metodología es conocida como SDN y permite utilizar información de capas superiores como la de red y de transporte. En lugar de tomar decisiones de reenvío internamente un switch OpenFlow sigue las instrucciones proporcionadas por un controlador SDN centralizado Esto permite una mayor flexibilidad y programabilidad en la gestión del tráfico de red.

### 3. ¿Se pueden reemplazar todos los routers de la Internet por Switches OpenFlow?

Piense en el escenario interASes para elaborar su respuesta

No sería práctico reemplazar todos los routers de internet por switches OpenFlow. Son más adecuados para entornos de red local y no tienen la misma capacidad de enrutamiento complejo que ofrecen los routers.

En el escenario interAS donde múltiples sistemas autónomos cooperan para el enrutamiento de datos a través de BGP (Border Gateway Protocol) se necesitan las funciones específicas de los routers y se necesitaría una cantidad muy grande de entradas BGP almacenadas en el dispositivo. Los switches OpenFlow no están diseñados para gestionar una red tan grande como lo es internet.

# Conclusiones

Minninet es una herramienta muy útil para crear y simular una topología de red en forma virtual. A lo largo de este trabajo se implementaron diversas topologías y se enviaron mensajes de un host a otro y mininet fue capaz de simular correctamente la conexión entre hosts a través del uso de switches.

iperf es una herramienta que permite correr un servidor. Esta vendría a ocupar el rol de la capa de aplicación y fue utilizada para enviar los mensajes para probar el correcto funcionamiento de la topología.

Pox es un controlador que permite configurar switches de manera remota en nuestra red implementada con Mininet. Dicha configuración nos permite modificar la tabla de enrutamiento y de esta forma hacer implementaciones de firewalls. En el caso de este trabajo práctico el firewall tiene la posibilidad de ser configurado a través de un archivo de configuración tipo json.