

Protocolos y Tecnologías para Servicios Móviles y Multimedia

E.T.S. Telecomunicación

Máster en Ingeniería de Telecomunicación

Curso 2022/2023

Práctica 1a: Evaluación de aplicaciones SDN con controlador POX

El objetivo de esta práctica es evaluar de forma práctica el efecto de aplicaciones SDN básicas. Para ello se empleará la máquina virtual Linux SDN Hub, en la que se ejecutarán varias aplicaciones sobre el controlador POX sobre el entorno de virtualización mininet.

Para realizar la práctica se recomienda abrir, al menos, tres terminales.

Terminal 1:

Ejecutar **mininet** como:

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

Ejecutar **ping** u otra aplicación que genere tráfico (iPerf)

```
mininet> h1 ping -c2 h2
```

Terminal 2:

Ejecutar el controlador **POX** junto con la aplicación de red (*app* representa la aplicación a ejecutar):

```
$/home/ubuntu/pox/pox.py app
```

Terminal 3:

Ejecutar **Wireshark** para monitorizar todos los interfaces (se recomienda ejecutar uno para cada enlace a monitorizar):

```
$sudo wireshark &
```

Se propone analizar (como mínimo) estas aplicaciones para la práctica 1a:

/home/ubuntu/pox/pox/forwarding/**forwarding.tutorial_l2_hub**

/home/ubuntu/pox/pox/forwarding/**forwarding.l2_learning**

Se pide como resultado:

Un documento con capturas de pantallas y explicaciones de lo que ocurre en la red.

1. Antes de arrancar la aplicación con el controlador Pox
 - ¿Hay conectividad entre hosts? ¿Por qué ocurre esto?
2. Ejecutar la aplicación **forwarding.tutorial_l2_hub**
 - ¿Qué nuevos mensajes aparecen en la red?
 - Ejecutar un ping entre 2 hosts e indicar que nuevo intercambio de mensajes ocurre en la red y entre que entidades se produce dicho intercambio.
3. Ejecutar la aplicación **forwarding.l2_learning**
 - Ejecutar un ping entre 2 hosts, ¿cuál es la principal diferencia que existe entre esta aplicación y la anteriormente ejecutada? Explicar esta diferencia basándose en los mensajes intercambiados actualmente en la red.
 - ¿Qué ocurre si el ping se ejecuta durante un periodo de tiempo lo suficientemente grande (>30seg)? ¿Está reflejado de alguna forma en la captura de Wireshark?

Práctica 1b: Análisis de la aplicación para balanceo de carga en servidores web

El ejemplo que se va a utilizar es *tutorial_stateless_lb*, que implementa una política de asignación de peticiones HTTP a diferentes servidores mediante una única dirección de acceso virtual (10.0.0.5).

1. Crear la topología con mininet:

```
$ sudo mn --arp --topo single,4 --mac --switch ovsk --controller remote
```

2. Activar la captura Wireshark en todos los interfaces para analizar el tráfico:

```
$ sudo wireshark
```

3. Iniciar el controlador y la aplicación:

```
$/home/ubuntu/pox/.pox.py forwarding.tutorial_stateless_lb
```

4. Actualizar la tabla arp de todos los hosts para crear un host virtual 10.0.0.5:

```
mininet> h1 arp -s 10.0.0.5 00:00:00:00:00:05  
mininet> h2 arp -s 10.0.0.5 00:00:00:00:00:05  
mininet> h3 arp -s 10.0.0.5 00:00:00:00:00:05  
mininet> h4 arp -s 10.0.0.5 00:00:00:00:00:05
```

5. Crear servidores web en los hosts h2, h3 y h4:

```
mininet> xterm hX
```

```
hX$ python -m SimpleHTTPServer
```

6. Realizar diversas descargas en h1 mediante la ejecución continua de:

```
mininet> h1 curl 10.0.0.5:8000
```

Se pide como resultado:

1. Analizar la salida en los terminales de los hosts (h2, h3 y h4) y estudiar el tráfico con Wireshark para explicar cómo se procesan los paquetes de petición HTTP. Se debe analizar especialmente la secuencia de mensajes Openflow y se deben adjuntar las capturas de pantalla necesarias.

Material de ayuda

Herramientas útiles:

1. Teclado en español : `sudo setxkbmap -layout 'es,es' -model pc105`
2. Wireshark: debe activarse después de crear mininet para disponer de todos los interfaces y utilizar el filtro `openflow_v1`
3. Grabación de sesión sin arrancar wireshark: usar `openflow.debug` como añadido en la línea de `pox` para crear `.pcap`

Explicaciones de códigos sobre POX

<https://noxrepo.github.io/pox-doc/html/>

Mensajes Openflow más relevantes:

HELLO, ECHO: confirma inicio y funcionamiento de conexión

FEATURES: Solicitud y envío de características del switch

STATS: Cuantificación del tráfico en el switch

SET_CONFIG: Solicitud y confirmación de que se han configurado parámetros

BARRIER: Solicitud y confirmación de que se han ejecutado las acciones pendientes

PACKET_IN: Transferencia de paquete completo al controlador

PACKET_OUT: Envío de un paquete a un puerto concreto del switch

FLOW_MOD: Crea reglas en el switch

Referencia a mensajes OF:

http://archive.openflow.org/wk/index.php/OpenFlow_Tutorial#ofp_flow_mod_OpenFlow_message

Ver reglas activas en el switch como resultado de FLOW_MOD

```
sudo ovs-ofctl -O OpenFlow10 dump-tables s1 | more
```