

Switch OpenFlow en Mininet

1. Introducción y objetivos

Mininet¹ ofrece un entorno de virtualización de red en Linux que permite crear escenarios con múltiples hosts y conmutadores con sencillez. Es muy empleado para hacer pruebas en escenarios con OpenFlow. En esta práctica se probará este entorno y se creará el código de un controlador muy simple para poner a prueba varios comportamientos diferentes en un conmutador.

2. Inicio en Mininet

Para esta práctica se empleará la imagen de máquina virtual que se ofrece en la web de Mininet y que trae instalado todo lo necesario. Puede descargar dicha imagen directamente de la web² o emplear la que ya se ha descargado en los ordenadores del laboratorio para ahorrar tiempo (consulte a su profesor de prácticas). Importe esa máquina virtual en VirtualBox (no lo haga en el *home* de su usuario pues es bastante grande).

Lea las recomendaciones sobre configuración³.

Para tomar contacto con Mininet siga el *Walkthrough*⁴ que ofrece en su web.⁵

3. Tutorial de Mininet

Diríjase al tutorial de OpenFlow en Mininet⁶. Puede saltar directamente a la sección “Learn Development Tools” para repasar lo visto en el walkthrough y ver algunos comandos más.

Continúe con la sección “Create a Learning Switch”, en concreto la primera parte, sobre el controlador POX. Emplearemos este controlador desarrollado en Python por su sencillez⁷. Puede ignorar las subsecciones sobre otros controladores.

Si no tiene conocimientos de programación en Python puede dirigirse a alguno de sus tutoriales⁸, aunque para esta práctica hará falta muy poco conocimiento y casi se puede resolver con intuición de programación (el código más complejo que escribiremos puede que tenga solo 30 líneas).

4. Control de un conmutador

Se le pide que a continuación desarrolle diferentes formas de controlar el tráfico en el escenario del tutorial de un conmutador, partiendo de su fichero de ejemplo `of_tutorial.py`

1. Modifique `of_tutorial.py` para que el controlador tome todas las decisiones de reenvío. No programará ninguna entrada en la tabla de flujos del conmutador sino que todos los paquetes pasarán al controlador, el cual

¹ <http://mininet.org>

² <http://mininet.org/download/>

³ <http://mininet.org/vm-setup-notes/>

⁴ <http://mininet.org/walkthrough/>

⁵ El puerto empleado por OpenFlow fue cambiado del 6633 al 6653. Tenga cuidado si software o documentación están hablando de puertos diferentes.

⁶ <https://github.com/mininet/openflow-tutorial/wiki>

⁷ Si tiene problemas porque el puerto al que intenta conectarse el switch no coincide con el puerto de OpenFlow que está empleando el controlador edite el código fuente del controlador para que escuche en el puerto apropiado. Busque 6633 en `pox/pox/openflow/of_01.py`

⁸ <https://docs.python.org/2/tutorial/>

llevará una tabla de direcciones MAC por puerto igual que un *learning bridge* e indicará al conmutador que reenvíe el paquete por el puerto correcto.

Punto de control opcional: Muestre el funcionamiento y explique el código

2. Modifique `of_tutorial.py` y en concreto su desarrollo del punto anterior para que ese controlador descarte al azar el 20% de los paquetes.

Punto de control opcional: Muestre el funcionamiento y explique el código

3. Modifique `of_tutorial.py` para implementar un *learning bridge* que programe flujos. Cuando el controlador vea una nueva dirección MAC origen aprenderá dicha dirección asociada al puerto por el cual la ha recibido. En caso de que sepa por qué puerto se llega a la dirección destino de la trama programará dos entradas en la tabla del conmutador, una para cada sentido de las tramas de ese flujo de MAC a MAC. Esas entradas tendrán un tiempo de caducidad por inactividad de 10 segundos y se borrarán en cualquier caso a los 60 segundos. Para cada paquete que reciba el controlador deberá indicar al conmutador por dónde reenviarlo.

Punto de control opcional: Muestre el funcionamiento y explique el código

5. Formato de entrega

La práctica se puede corregir mediante puntos de control o mediante entregable. Si ha pasado los puntos de control no necesita entregar nada. Si no lo ha hecho debe entregar las tres versiones del controlador que ha desarrollado, junto con una explicación detallada sobre el código que ha introducido en `of_tutorial.py`.

Puede crear tres ficheros `of_tutorial.py`, uno para cada versión del controlador, o uno solo con tres funciones diferentes para gestionar los eventos (del mismo estilo que dicho fichero tiene un método para emular un hub y otro en blanco para un conmutador). Entregue en un documento PDF la descripción detallada del funcionamiento de su código. Incluya en el mismo una descripción de los pasos a seguir para probar cada controlador.

Esta entrega se hará por la página web de la asignatura.

6. Comentarios y recomendaciones

- Puede suponer que el conmutador solo recibirá las tramas de un host (es decir, con origen en una dirección MAC en concreto) por un solo puerto (aunque puede recibir por el mismo puerto tramas de múltiples direcciones MAC distintas). Es decir, no va a recibir tramas del mismo host por diferentes puertos.