

Práctica 10. Simuladores de red

Familiarizándose con mininet

Esta práctica se realizará durante la clase de laboratorio.

Objetivos de aprendizaje

- Manejo de la interfaz de comandos del simulador Mininet.
- Manejo de la API de Mininet para crear distintas topologías.

1. Introducción

Redes Definidas por Software (SDN - Software Defined Networks)

SDN es un nuevo paradigma en redes de computadores, que consiste en separar el plano de control del plano de datos (reenvío) la red, lo que permite optimizar cada uno de ellos por separado. En SDN el control de la red se centraliza en uno o más servidores separados de los dispositivos de reenvío (switches). Las decisiones de control se llevan a cabo mediante un programa software que se ejecuta en el controlador. El controlador envía órdenes a los switches para que detecten paquetes con unas ciertas características y lleven a cabo acciones sobre ellos, como reenviar o descartar.

Openflow

Openflow es una interfaz abierta para controlar de forma remota las tablas de reenvío de los conmutadores de red. Insertando la información adecuada en las tablas de reenvío de los conmutadores se pueden crear funciones de red de diversa complejidad, desde un switch convencional hasta un firewall.

En un switch de red clásico, el plano de datos, que reenvía los paquetes, y el plano de control, que toma las decisiones de enrutamiento de alto nivel, están en el mismo dispositivo. OpenFlow separa el plano de datos del plano de control. El plano de datos sigue residiendo en el dispositivo, mientras que el las decisiones de control se toman en un servidor de control. El controlador y el switch se comunican a través de una red de control utilizando el protocolo OpenFlow.

La tabla de flujo de un switch OpenFlow está compuesta por una serie de entradas que consisten en un conjunto de campos del paquete a comparar y una acción (p.e. reenviar por un puerto, modificar un campo o eliminar el paquete). Al recibir un paquete compara los campos de la cabecera con los de la tabla de flujo y ejecuta la acción. Si no existe ninguna entrada compatible en la tabla de flujo, el switch envía el paquete al controlador que deberá tomar una decisión sobre el paquete en un programa software, pudiendo eliminar el paquete o insertar una nueva entrada en la tabla de flujo del switch para que en el futuro el switch pueda tratar paquetes similares directamente.

Mininet

Mininet es un emulador de redes que crea una red virtual definida por software compuesta por host, switches, controladores y enlaces. Los hosts ejecutan software Linux y los switches soportan OpenFlow.

Mininet permite tener una red completa en un PC, lo que permite el prototipado rápido de cualquier infraestructura de red definida por software basada en OpenFlow. Las características de mininet son las siguientes [1]:

- Provee un banco de pruebas para desarrollar aplicaciones basadas en OpenFlow.
- Permite a varios desarrolladores independientes trabajar concurrentemente sobre la misma topología.
- Permite testear topologías complejas sin necesidad de cablear una red física.
- Incluye una interfaz de comandos para depuración y ejecución de pruebas.
- Permite programar topologías adaptadas y también tiene un conjunto básico de topologías parametrizadas que se pueden usar sin necesidad de programar.
- Está provisto de una API en Python para la creación de redes y experimentación.

2. Descarga de la máquina virtual SDN

Aunque mininet puede instalarse en su distribución Linux, es conveniente utilizar alguna de las máquinas virtuales disponibles para no interferir con el sistema base de su ordenador. En este curso utilizaremos la máquina virtual, con la versión 2.2.0 de Mininet, disponible en:

<https://github.com/mininet/mininet/wiki/Mininet-VM-Images>

Existe una versión de 64 bits que funcionará en la mayoría de los ordenadores modernos y otra de 32 bits que se recomienda para los usuarios de Windows que utilicen Virtualbox o Hyper-V.

Una vez descargada y añadida la máquina virtual a Virtualbox, deberá ajustar la configuración de la misma. Lo primero que deberá hacer antes de iniciar la máquina virtual es modificar la configuración de red añadiendo una interfaz de red adicional de tipo **sólo anfitrión**, como se muestra en la figura 1. Esta interfaz de red permitirá comunicar al PC en el que está ejecutando la máquina virtual con la máquina virtual.

Una vez iniciada la máquina virtual, el nombre de usuario y la clave son:

```
username: mininet
password: mininet
```

Tras arrancar la máquina debemos modificar la configuración de la nueva interfaz de red que acabamos de crear. Para ello se edita el fichero `/etc/network/interfaces`, añadiendo las siguientes líneas:

```
auto eth1
iface eth1 inet dhcp
```

A continuación, ejecute la orden `ifup eth1`.

Este paso sólo deberá realizarlo la primera vez que inicie la máquina virtual. A partir de este momento, cada vez que arranque la máquina virtual deberá determinar la dirección IP de la interfaz `eth1`. A continuación, puede conectarse mediante `ssh` desde el host anfitrión a la máquina virtual utilizando el siguiente comando (suponiendo que la dirección IP de la máquina virtual es 192.168.56.101):

```
ssh -X mininet@192.168.56.101
```

El flag `-X` permite que ejecuten aplicaciones gráficas desde la sesión `ssh`.

3. Repaso por los comandos básicos de mininet

En el siguiente enlace podrá encontrar un *walktrought* de los comandos básicos para utilizar mininet:

<http://mininet.org/walkthrough/>

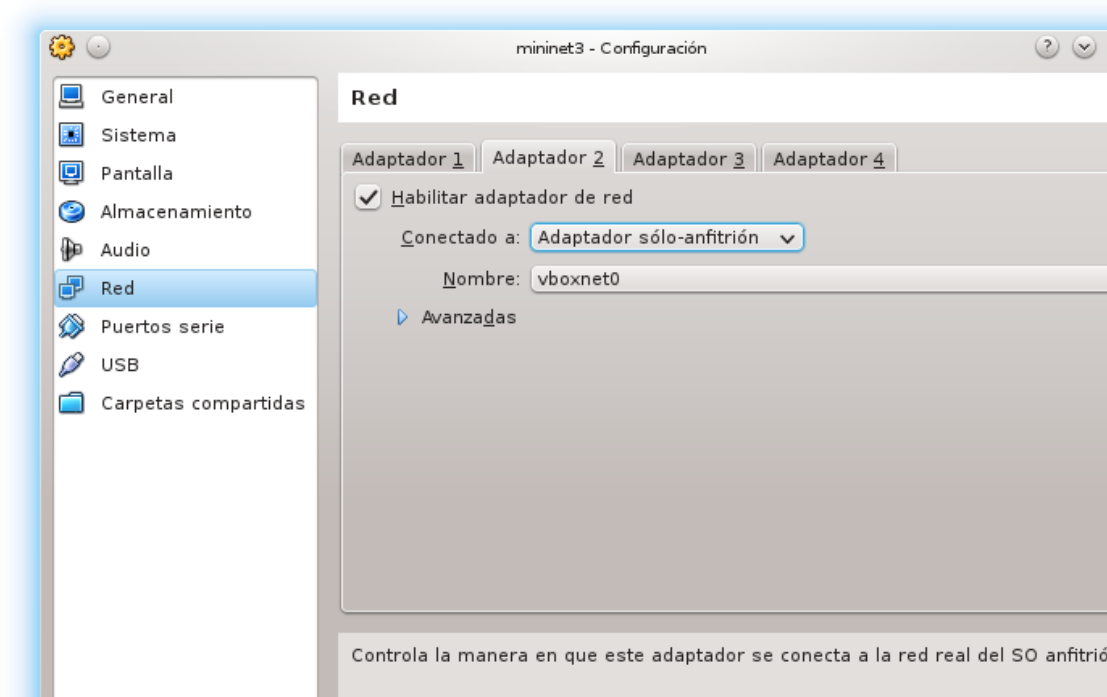


Figura 1: Configuración del adaptador sólo anfitrión.

4. Ejercicios

- Crear una topología lineal de cinco elementos.
 - Comprobar la conectividad entre los hosts.
 - Mostrar la lista de hosts.
 - Mostrar los enlaces.
 - Mostrar la información sobre todos los nodos.
- Crear una topología de árbol binario de 3 niveles.
 - Comprobar la conectividad entre todos los hosts.
 - Cambie el ancho de banda de los enlaces para que éste sea de 1 Mb. Ahora lleve a cabo una prueba de rendimiento utilizando la herramienta `iperf`.
 - Haga ping entre los nodos `h1` y `h2`.
 - Abra un `xterm` en el host `h1` y ejecute `wireshark`. Ahora abra una segunda `xterm` para `h2` y haga ping hacia `h1` desde la `xterm`.
 - Determine si `s1` se comporta como un switch o como un hub.

5. Programación de la API de mininet: Creación de topologías personalizadas

Mininet además de estar provisto de una serie de topologías de red tipo parametrizables que se pueden instanciar desde la línea de órdenes también permite crear topologías personalizadas en el lenguaje de programación Python.

```
1 #!/usr/bin/python
2 #coding=utf-8
3
4 from mininet.topo import Topo
5 from mininet.net import Mininet
6 from mininet.log import setLogLevel
7 from mininet.cli import CLI
8
9 class SingleSwitchTopo(Topo):
10     "Un switch sencillo conectado a n hosts."
11     def build(self, n=2):
12         switch = self.addSwitch('s1')
13         #range(N) generara los numeros 0..N-1
14         for h in range(n):
15             host = self.addHost('h%s' % (h + 1))
16             self.addLink(host, switch)
17
18
19
20 topos = { 'MyTopo4':( lambda: SingleSwitchTopo(4) ), 'MyTopo6':( lambda:
    SingleSwitchTopo(6) ) }
```

A continuación tiene una descripción de las clases, métodos y funciones más importantes del código anterior:

- **Topo**: Clase base para las topologías mininet.
- **build()**: Methodo que se deberá sobrescribir en su clase de topología. Los parámetros de creación (n) se pasarán automáticamente desde **Topo.__init__()**.
- **addSwitch()**: Añade un switch a la topología y devuelve el nombre.
- **addHost()**: Añade un host a la topología y devuelve el nombre.
- **addLink()**: Añade un enlace bidireccional a la topología.

La documentación de las distintas clases, métodos y funciones de mininet se puede encontrar en [2]. Por otra parte, puede encontrar un tutorial de Python en [4]

Para probar esta topología deberá ejecutar mininet:

```
$ sudo mn --custom topo.py --topo MyTopo6
```

donde topo.py es el fichero que contiene el código Python del listado y si utilizamos **--topo MyTopo6** se añaden 6 hosts a la topología de un switch. En cambio, si utilizamos **MyTopo4** se creará la topología con 4 hosts.

6. Crear una topología en árbol

Añada una clase al fichero anterior para crear una topología en árbol. La clase deberá recibir como parámetros el número de niveles del árbol y el fanout (número de hijos de cada nodo). Los niveles intermedios deberán estar compuestos por switches y las hojas deberán ser hosts.

7. Crear una topología Fat-tree

La topología Fat-tree es una arquitectura en árbol que cumple con la siguiente propiedad: Si el fanout es k , la conexión de un nodo con su padre tiene k veces el superior al ancho de banda de ese nodo con sus hijos, si

los tiene. El ancho de banda de la conexión de un host con el switch es el que condiciona el ancho de banda de las conexiones entre switches de los niveles superiores del árbol.

Consulte la documentación de la clase `TCLink` en [2]. También podría ser de ayuda el siguiente ejemplo de llamada al método `addLink`.

```
self.addLink(ps, child_switch,cls=TCLink,bw=5,delay='1ms')
```

8. Asignación de direcciones IP y otras configuraciones a nivel de host

Aunque mininet asigna automáticamente una dirección IP a cada host, también permite asignar direcciones IP desde un script Python. Para ello debemos utilizar. Lea el documento [3], determine cómo asignar direcciones IP a los hosts. A continuación, cree una arquitectura Fat-tree y asigne la dirección IP a cada host del siguiente modo: Las direcciones serán del tipo 192.168.x.y, donde x es el número del switch de acceso al que está conectado e y es el número de host conectado a ese switch.

Referencias

- [1] <http://mininet.org/overview/>
- [2] <http://mininet.org/api/annotated.html>
- [3] <https://github.com/mininet/mininet/wiki/Introduction-to-Mininet>
- [4] <https://docs.python.org/2/tutorial/>