Práctica Linux Network Namespace y open vSwitch

Tecnologías y Protocolos de Internet de Nueva Generación

GSyC
Departamento de Teoría de la Señal y Comunicaciones y
Sistemas Telemáticos y Computación

Noviembre de 2017

Para realizar los siguientes ejercicios es necesario que importes en VirtualBox el siguiente fichero que contiene una máquina virtual Linux 16.04: `/var/lib/vms/sdn/sdn.ova`. Asegúrate antes de importar la máquina, en las opciones de virtualbox tienes algo como: `General -> Avanzado -> Carpeta instantáneas = /var/tmp/misuario`

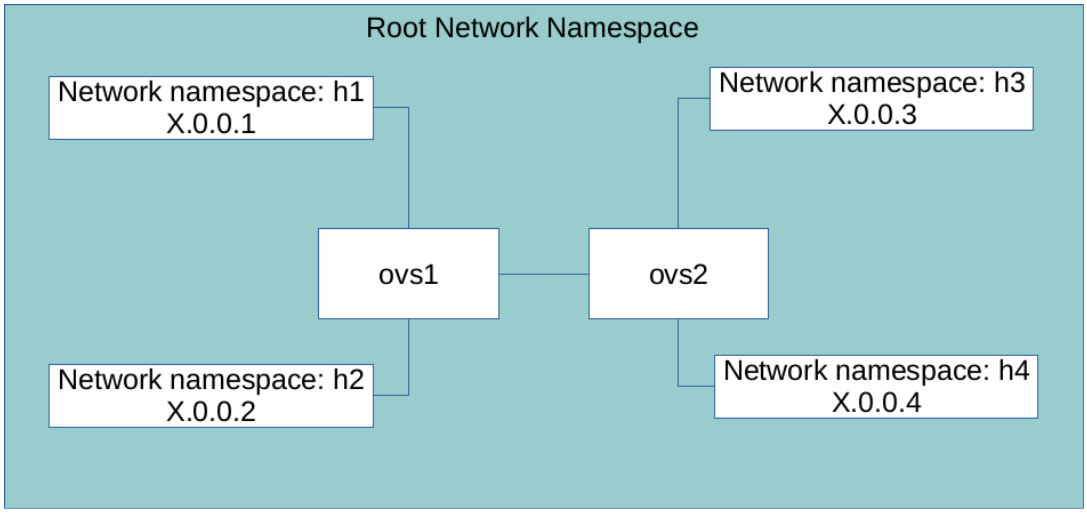
El usuario de la maquina virtual es `tping` contraseña `TPINGTPING`.

Antes de empezar a hacer la práctica, por favor, ejecuta los siguientes comandos para borrar una configuración antigua que ha quedado en la máquina virtual:

```
sudo ovs-vsctl del-br s1
sudo ovs-vsctl del-br s2
sudo ovs-dpctl del-dp ovs-system
```

1. Network namespaces en Linux

Utilizando network namespaces en Linux realiza un script para conseguir la configuración que se muestra en la siguiente figura, usando como valor para X el que te ha sido asignado:



Es conveniente que a la vez que generar el script creando las interfaces `veth`, los *network namespaces* y los switches, generes un script para eliminar esa configuración. De esta forma, si te equivocas en el script de configuración, podrás aplicar el segundo script para eliminar dicha configuración y volver al punto de partida. Al borrar un namespace se eliminarán las interfaces y extremos `veth` de dicho namespace. Recuerda que deberás ejecutar dichos scripts como root para que se puedan ejecutar sus comandos.

Una vez ejecutado el script de configuración:

1. Comprueba la conectividad IP de todos los namespaces.

2. Incluye un dibujo que muestre los network namespaces, los identificadores de las interfaces `veth` (los nombres que has asignado a cada uno de sus extremos y los números que le ha asignado el sistema operativo al crearlos) y las direcciones Ethernet de cada interfaz de red.

3. Con la tabla de direcciones aprendidas en los switches vacía, realiza una captura de tráfico (`ns-01.cap`) en una de las interfaces `veth` que conectan los switches entre sí y ejecuta un `ping` desde `h1` hacia la dirección IP de `h2`. Relaciona las direcciones aprendidas en `ovs1` y `ovs2` con las de las interfaces de red, después de realizar el `ping`. Explica el contenido de la captura.

4. Con la tabla de direcciones aprendidas en los switches vacía, realiza una captura de tráfico (`ns-02.cap`) en una de las interfaces `veth` que conectan los switches entre sí y ejecuta un `ping` desde `h1` hacia la dirección IP de `h4`. Relaciona las direcciones aprendidas en `ovs1` y `ovs2` con las de las interfaces de red, después de realizar el ping. Explica el contenido de la captura.

2. Mininet

Mininet configura automáticamente los *network namespaces* atendiendo a las necesidades de la topología que se desea configurar. Mininet tiene algunas topologías preconfiguradas por defecto, por ejemplo, la configuración de n máquinas conectadas a un único switch: `--topo single,n`. Pero además, mininet permite programar las topologías que se deseen usando un fichero de definición de topología escrito en python.

Estudia el fichero de la topología que se encuentra en `~/mininet/custom/topo-2host-2sw-2host.py` y trata de identificar la topología que se está definiendo en función de su contenido.

Modifica convenientemente el fichero para que las máquinas tengan las mismas direcciones IP que en el apartado anterior, sustituyendo X por el valor que tienes asignado:

```
# Add hosts and switches
leftHost1  = self.addHost( 'h1', ip='X.0.0.1/24' )
leftHost2  = self.addHost( 'h2', ip='X.0.0.2/24' )
rightHost3 = self.addHost( 'h3', ip='X.0.0.3/24' )
rightHost4 = self.addHost( 'h4', ip='X.0.0.4/24' )
```

Arranca mininet con esa topología, por defecto, mininet configurará direcciones IP a cada una de las máquinas. Para ello, entra en la carpeta de la máquina virtual `~/mininet/custom` e inicia mininet de la siguiente forma

```
sudo mn --custom topo-2host-2sw-2host.py --topo=mytopo --controller=none --mac
```

Donde se está arrancado mininet con la topología definida en el fichero `topo-2host-2sw-2host.py` llamada `mytopo`, sin controlador y con asignación .ªmigable"de direcciones mac.

2.1. Network namespaces en mininet

Estudia la configuración de *network namespaces* creada por mininet con los siguientes apartados (para cada uno de ellos muestra el comando que has usado y el resultado obtenido, recuerda que para algunos casos podrás usar diferentes comandos, muestra sólo uno de ellos que te permita conocer el resultado):

1. Indica los switches que se han creado para usarlos desde el espacio de usuario.

2. Indica las interfaces que se han creado en el *root network namespace*. Observarás que además de los switches `s1` y `s2` hay uno que se denomina `ovs-system`. Este switch representa el switch `datapath` en el módulo del kernel.

3. Identifica los extremos de las interfaces `veth` que se encuentran visibles en cada *network namespace* de las máquinas, con los extremos que son visibles en el *root network namespace*. Escribe para cada uno de ellos el identificador numérico (`ifXX`) y el nombre asignado (`hZ-ethW` o `sR-ethP`)

4. Indica los números de puerto asignados en el switch para la conexión de cada uno de los *network namespaces*.

5. Recopila toda la información anterior realizando un dibujo que muestre las direcciones IP, direcciones Ethernet, nombre de las interfaces e identificadores de interfaz que se han creado. Indica también los puertos que tiene cada switch y su número.

6. Esta configuración de switches open vSwitch, en realidad queda implementada dentro del kernel con la definición de un datapath, una abstracción del kernel para construir una caché rápida de flujos que es lo primero que se consulta cuando se recibe un paquete. Observa la información del datapath creado que se denomina `system@ovs-system`.

2.2. Configuración manual de un switch Open vSwitch

2.2.1. Comportamiento "normal" del switch

En la configuración anterior de mininet:

1. Comprueba la conectividad entre todas las máquinas, no deberían comunicarse ninguna ya que hemos arrancado sin controlador y los switches no tienen un comportamiento asociado.
2. Añade el comportamiento normal de un switch a `s1` y a `s2`, `action=normal`.
3. Vacía las tablas de direcciones aprendidas tanto en s1 como en s2.
4. Comprueba que no hay ninguna regla en datapath instalado en el kernel.
5. Comprueba que ahora sí hay conectividad realizando un `ping` desde `h1` a `h2` y observa la tabla de direcciones aprendidas en s1 y s2, identificando a quién pertenece cada una de ellas. Justifica una explicación para ese resultado.
6. Para entender lo que ha ocurrido realmente, realiza los siguientes pasos:
   - Vacía las tablas de direcciones aprendidas tanto en s1 como en s2.
   - Vacía las cachés de ARP de h1 y h2.
   - Comprueba que no hay ninguna regla en datapath instalado en el kernel.
   - Realiza una captura de tráfico en la interfaz eth0 de `h2`.
   - Muestra los flujos instalados en el datapath cada segundo y deja el resultado en un fichero. Para ello escribe un script de shell que haga lo siguiente y permite su ejecución:

     ```
     #!/bin/bash
     while true
     do
       sudo ovs-dpctl dump-flows system@ovs-system >> dp-logfile.txt
       echo "-----------------------------------" >> dp-logfile.txt
       sleep 1
     done
     ```

   - Ejecuta un `ping` para que envíe 10 paquetes desde `h1` a `h2` y espera otros 10 segundos antes de cortar el script. Interrumpe la captura.

   Analiza el resultado obtenido en el fichero `dp-logfile.txt`. Realiza una tabla con la salida que se ha guardado en el fichero de log. Algunas ejecuciones del comando `sudo ovs-dpctl dump-flows system@ovs-system` serán iguales, sólo escribe una tabla para cada uno de los resultados diferentes [1].

in_port	src_eth	dst_eth	eth_type	actions	paquete capturado

7. Comprueba el/los flujo/s instalado/s en los switches e indica en qué tabla se han instalado.
8. Vacía las tablas de direcciones aprendidas tanto en s1 como en s2. Comprueba que ahora sí hay conectividad realizando un `ping` desde `h1` a `h4` y observa la tabla de direcciones aprendidas en s1 y s2.
9. Vuelve a ejecutar el ping desde h1 hacia h4 y déjalo enviando paquetes todo el tiempo. ¿Puedes suponer qué flujos quedarán almacendados en el datapath de forma estable? Observa los flujos que quedan instalados en el datapath del kernel después de unos 20 segundo de haber iniciado el ping y comprueba tus suposiciones.

[1] Dado que la máquina tiene activada la pila IPv6 se están generando paquetes Router Advertisement que afectan a los flujos datapath. No consideres los flujos definidos sobre paquetes IPv6.

2.2.2. Comportamiento específico para un switch

Borra todos los flujos de s1 y s2 y realiza la siguiente configuración de flujos en un fichero para cargarlo posteriormente en el switch s1 y s2:

- **Tabla 0: ACL**
  - No permitir tramas Ethernet con dirección Ethernet origen, broadcast Ethernet.
  - No permitir tramas Ethernet con direcciones Ethernet destino 01:80:c2:00:00:00 y 01:00:0c:cc:cc:cd que pertenecen al protocolo STP
- **Tabla 1: Etiquetar según VLAN**
  - Se desea que las máquinas h1 y h3 estén conectadas a la VLAN 10 y las máquinas h2 y h4 estén conectadas a la VLAN 20. Será necesario etiquetar las tramas Ethernet con las VLANs correspondientes (utiliza la acción `actions=mod_vlan_vid:<VLANID>`, donde `<VLANID>` puede ser 10 o 20 según sea la comunicación).
- **Tabla 2: Reenvío en función de VLAN**
  - Reenviar correctamente por el puerto adecuado en función de la etiqueta VLAN y dirección Ethernet de máquina destinataria, eliminando la etiqueta en caso necesario.

1. Utiliza `ovs-appctl ofproto/trace` para comprobar que las reglas instaladas se aplican a los flujos que se han definido.
2. Prueba la configuración usando `ping` y comprobando que se comunican sólo las máquinas de la misma VLAN.
3. Explica los flujos instalados en el datapath como consecuencia de la comunicación entre h1 y h3.
4. Explica los flujos instalados en el datapath como consecuencia de la comunicación entre h2 y h4.
5. Captura tráfico en el tramo en el que se comunican s1 y s2 para comprobar que los paquetes tienen añadida la etiqueta VLAN, para la comunicación entre h1 y h3 y entre h2 y h4.
6. Captura tráfico en el tramo en el que se comunican s1 y s2 para comprobar que los paquetes tienen añadida la etiqueta VLAN.