Escuela Superior de Informática

Universidad de Castilla-La Mancha
Escuela Superior de Informática

Planificación e Integración de Sistemas y Servicios.

4º Grado en Ingeniería Informática.

Práctica 4. Gestión de prioridad de tráfico en IP.

Autor:
Alberto Salas Seguín y
Marcos López Sobrino.

Fecha:
23 de diciembre de 2018

1. Entorno de trabajo.

El entorno de trabajo utilizado han sido 3 máquinas virtuales gestionadas mediante Vagrant, con la imagen ***ubuntu/trusty64***. Una de ellas actúa como cliente, otra como servidor y otra como router entre ambas. A continuación, explicamos la configuración de las redes, de las máquinas, y los paquetes instalados.

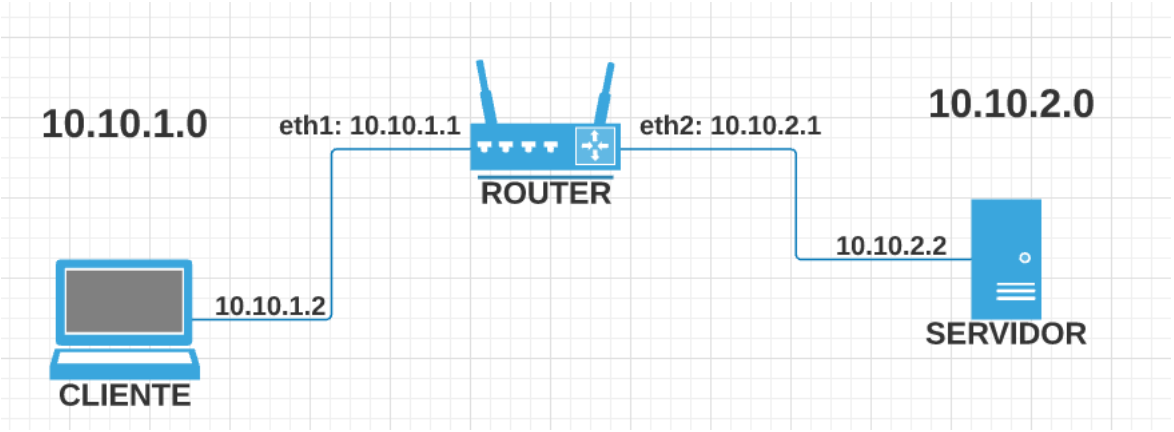

Figura 1: Topología de la red.

1.1. Cliente y servidor.

Como vemos en la figura anterior, la red del cliente, que corresponde a la interfaz **eth1** del router, tiene como IP **10.10.1.0**; mientras que la red del servidor, que corresponde a la interfaz **eth2** del router, tiene como IP **10.10.2.0**.

Para el correcto funcionamiento de las comunicaciones, hay que indicarle a la máquina cuál es la ruta por defecto, para ello, se ejecutan los siguientes comandos:

```
$ sudo ip route del default dev eth0
$ sudo ip route add default via <ip_router> dev eth1
```

donde *ip_router* sería la dirección IP de la interfaz del router dentro de la red correspondiente.

Los paquetes necesarios para el funcionamiento de la práctica en el caso del cliente y el servidor son:

- **iptables**
- **iperf**
- **sip-tester**

1.2. Router.

A la máquina del router se le ha instalado un entorno gráfico para poder usar cómodamente la herramienta **Wireshark**. Además, se le han asignado 2048 MB de RAM. Por otra parte, se ha configurado las interfaces de red con la dirección **.1** de la red correspondiente, tal y como vemos en la figura. Adicionalmente, para que la máquina actúe como router se debe activar el **IP forwarding**, para ello, en el playbook correspondiente del router se ha añadido una regla específica.

Por último, en el router se han instalado los siguientes paquetes:

- Para la interfaz gráfica:
  - **xorg**
  - **gnome-core**
  - **gnome-system-tools**
  - **gnome-app-install**
- **wireshark**
- **iptables**

1.3. Archivos de configuración e inicialización.

Todo lo anterior se recoge en el archivo ***Vagrantfile*** y los distintos ***playbooks***, que se pueden encontrar junto a este documento. Para poner en funcionamiento el entorno de trabajo, ejecutamos:

```
vagrant up --provision
```

2. Configuración de iptables en router, cliente y server.

Dado que tanto en la máquina del router como la del cliente, se van a introducir reglas de iptables para el correcto funcionamiento, para no perder estas reglas se va a utilizar ***iptables-persistent***, el modo para recuperar todas las reglas cada vez que se inicie la máquina es mediante una *task* cuya función es cargar un archivo con todas las reglas necesarias en el directorio ***/etc/iptables/rules.v4***, este directorio se encuentra tanto en la máquina del router como en la máquina del cliente. La regla que hemos utilizado para copiar este archivo en el directorio mencionado es la siguiente:

```
- name: copy iptables
  copy:
    src: <archivo a copiar>
    dest: /etc/iptables/rules.v4
    owner: root
    group: root
    mode: 0644
  notify: "restore iptables"
```

Como vemos en el código anterior, en la última línea se hace mención a ***notify***, lo cual hace necesario la presencia de un manejador, de manera que cada vez que se invoque a la tarea ***copy tables***, se va a hacer el *notify*, cuya función es hacer un **restore** del archivo **/etc/iptables/rules.v4** que hemos copiado. Esto lo podemos ver en el siguiente código:

```
handlers:
    - name: restore iptables
      shell: iptables-restore /etc/iptables/rules.v4
      listen: "restore iptables
```

3. Reglas iptables utilizadas en router, cliente y server.

- **Router.**

  En el caso del router hemos utilizado las reglas iptables que vienen dadas en el enunciado de la práctica. Estas son:

```
iptables -t filter -A INPUT -m dscp --dscp 14 -m limit --
   limit 5/s --limit-burst 5 -j ACCEPT

iptables -t filter -A INPUT -m dscp --dscp 14 -j DROP
```

  Donde los parámetros –limit y –limit-burst significan lo siguiente.

  - **limit.** El módulo limit se utiliza para restringir la tasa de coincidencias, por ejemplo, para suprimir los mensajes de registro. Sólo coincidirá con un número determinado de veces por segundo (por defecto 3 coincidencias por hora, con una ráfaga de 5). Se necesitan dos argumentos opcionales:
    - **-limit.** Seguido de un número y una unidad, especifica el número medio máximo de coincidencias que puede permitirse por segundo. Nos permite configurar el tiempo que tiene que pasar para que esta regla se cumpla entre dos paquetes.
    - –**limit-burst.** Seguido de un número que indica la cantidad máxima de paquetes antes de que se active el límite anterior.

- **Client.** Dado que vamos a marcar paquetes, la tabla que vamos a usar es **mangle**. Por un lado, el tráfico RTP lo vamos a marcar con **Expedited Forwarding**, el **DSCP** recomendado para Expedited Forwarding es **46 (101110)**. Por otro lado, el tráfico SIP lo vamos a marcar como **Assured Forwarding 11**, cuyo valor es 10. De modo que las reglas nos quedan de la siguiente manera:

```
iptables -t mangle -A OUTPUT -p udp --dport 5060 -j DSCP --set-
   dscp 10

iptables -t mangle -A OUTPUT -p udp --dport 10000:20000 -j DSCP
    --set-dscp 46
```

  - La primera regla permite el tráfico udp saliente del cliente que tiene como puerto destino (dport) el puerto 5060, el cual es el puerto dedicado al protocolo SIP. Como hemos dicho antes, este tráfico se ha marcado con AF11 con el valor 10.
  - La segunda regla permite el tráfico udp saliente del cliente que tiene como destino el rango de puertos que corresponden al protocolo RTP (10000:20000) y dicho tráfico se ha marcado como Expedited Forwarding con el valor 46.

- **Server.** Las reglas son las mismas que en el cliente, sólo con una diferencia, que en lugar de utilizar –**dport** (destination port), en el server se va a utilizar –**sport** (source port), lo demás, está configurado igual que en las reglas del cliente.

```
iptables -t mangle -A OUTPUT -p udp --sport 5060 -j DSCP --set-
   dscp 10

iptables -t mangle -A OUTPUT -p udp --sport 10000:20000 -j DSCP
    --set-dscp 46
```

4. Bibliografía

- Configuración reglas Iptables.

  https://netfilter.org/documentation/HOWTO/packet-filtering-HOWTO-7.html#ss7.3

  https://www.cisco.com/c/en/us/support/docs/quality-of-service-qos/qos-packet-marking/10103-dscpvalues.html

  https://community.rti.com/kb/how-set-dscp-flag-your-system