# **Trabajo final RDSV**

## Curso 2021/22 MUIT

*Grupo 06: Ángela Burgaleta y Raúl Cabria*

# 1. Objetivos

La finalidad de este trabajo es modificar una red residencial virtualizada con OSM, llevada a cabo en la práctica 4. Los cambios efectuados han sido los siguientes:

* Utilizar un Router VyOS virtualizado en lugar de la vcpe
* Añadir soporte de QoS implementado mediante SDN con Ryu
* Añadir conectividad IPv4 desde la red residencial hacia Internet. Con uso de doble NAT en vCPE y en r1
* Sustituir el switch de vclass por un conmutador controlado por OpenFlow
* Añadir una nueva VNF con conexión a Vclass para capturar el tráfico ARP mediante arpwatch
* Gestionar la calidad de servicio en la red de acceso mediante la API REST de Ryu controlando Vclass
* Desplegar dos redes residenciales
* Automatizar el despliegue mediante scripts y OSM

Diagrama

Descripción generada automáticamente

# 2. Desarrollo

## 2.1 Utilizar un Router VyOS como vCPE.

En primer lugar se ha descargado e instalado una imagen de vyos ya creada para utilizarla en el Dockerfile del contenedor. Se ha modificado también el descriptor vcpe.yaml para que utilice la imagen Docker de vnf-vyos.

wget <http://idefix.dit.upm.es/download/vnx/examples/mupi-proxy/vyos-rolling-1.3.tar.gz>

docker image load -i vyos-rolling-1.3.tar.gz

Dockerfile:

FROM vyos/rolling:1.3

RUN mkdir /config

CMD /sbin/init

## 2.2 Automatización del despliegue del escenario con el script init.sh

Texto

Descripción generada automáticamentePara no tener que realizar todos los pasos de forma manual se ha utilizado un script automatizándolos.

## 2.3 Configuración de los túneles VxLAN: brgX-Vclass y Vclass-VyOS

Para el túnel brX-Vclass: Tras haber iniciado el servicio OpenVirtualSwitch en cada VNF, en Vclass se debe agregar un bridge y asociar las interfaces, esto se encuentra automatizado en el script vclass\_start.sh. A este script se le pasa como parámetro (también automatizado con los scripts vcpe-1.sh y vcpe-2.sh) el nombre de la vcpe (en nuestro caso vcpe-1 para la red residencial 1 y vcpe-2 para la red residencial 2) y las direcciones IP para ambas terminaciones del túnel, la que conecta con la red residencial y la que conecta con el Vclass.

Texto

Descripción generada automáticamente

## 

Para el túnel Vclass-Vyos: Se configura en el script vyos\_start.sh las interfaces ethernet y el túnel vxlan con su dirección, la MTU, el identificador y el puerto. A este script se le deben pasar como parámetros el nombre de la vcpe, para saber a qué red residencial se hace referencia, la IP privada para la vcpe y la dirección ip pública que conectará con la ExtNet. En vyos\_start.sh también configuramos la NAT.

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamenteTexto

Descripción generada automáticamenteA continuación, se muestran los scripts comentados anteriormente vcpe-1.sh y vcpe-2.sh con las direcciones IPs. Además accedemos a las máquinas para permitir que se configure la dirección por dhcp.

## 2.4 Asignación de direcciones por DHCP

En el script vyos\_start.sh se realiza la configuración DHCP para los hosts h11, h12, h21 y h22.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

## 2.5 Añadir una nueva VNF para capturar el tráfico mediante arpwatch

Se ha añadido una nueva VNF con un túnel vxlan a Vclass que permita capturar el tráfico arp. Para ello el primer paso ha sido construir una imagen Docker vnf-arpwatch, el Dockerfile que se ha empleado ha sido el siguiente.

Texto

Descripción generada automáticamente

En el descriptor arpwatch.yaml para generar la instancia de la nueva VNF en el OSM la imagen que se especifica es la que se ha creado con el Dockerfile anterior, se muestra en el campo image:vnf-arpwatch. Por otro lado, también es necesario modificar el descriptor de la NS para añadir otro miembro más.

Texto

Descripción generada automáticamenteTexto

Descripción generada automáticamente

Finalmente, se levanta el túnel vxlan3 que conecta la VNF del arpwatch con el Vclass en el script vclass\_start.sh

Texto

Descripción generada automáticamente

## 2.6 Ryu: controlar la calidad de servicio

Para controlar la QOS mediante la VNF Vclass, en primer lugar, ha sido necesario modificar el Dockerfile del vnf-img para añadir los paquetes ryu-bin e iproute2. También se cambia la imagen de ubuntu:xenial a ubuntu:bionic.

Para sustitur el switch por un conmutador OpenFlow que posteriormente se conecte a Ryu se realiza una configuración por comandos ovs-vsctl con el bridge0. En primer lugar, se habilitan los protocolos OpenFlow 10, 12 y 13. Se habilita el modo secure por si tenemos fallos de conectividad. Finalmente se configura el datapath y el controller de br0 con dirección 127.0.0.1 y puerto 6633 (correspondiente a Ryu). El comando ryu-manager que se muestra en la primera línea sirve para conectar el controlador Ryu con el switch.

Texto

Descripción generada automáticamente

Las reglas configuradas han sido las siguientes:

* Red residencial: 12Mbps de bajada
* Para hX1: 8Mbps mínimo de bajada
* Para hX2: 4Mbps máximo de bajada

Se incorporan las siguientes sentencias al script qos.sh:

Captura de pantalla de computadora

Descripción generada automáticamente

# 3. Comandos puesta en marcha del escenario y test

1. Descarga e instalación de la máquina virtual:

/lab/rdsv/bin/rdsv-get-and-install-vnxsdnnfvlab

2. Despliegue del escenario dentro de la carpeta del proyecto:

./init.sh

3. Túneles vxlan:

./vcpe-1.sh

./vcpe-2.sh

4. Calidad de servicio:

./qos.sh vcpe-1

./qos.sh vcpe-2

5. Destruir escenario:

./stop-vnx.sh

./destroy.sh vcpe-1

./destroy.sh vcpe-2

6. COMANDOS ÚTILES

*Ver contenedores Docker activos:*

Docker ps -a

*Acceder al root de un contenedor:*

Docker exec -ti [id] /bin/bash

**ARPWATCH**

Docker exec -ti [id] /bin/bash

Touch /var/lib/arpwatch/arp.dat

Service arpwatch start

*Ping de h11 a broadcast (mandamus a una dirección IP del rango que no se asigne a ningún equipo):*

Ping -c5 192.168.255.53

cat /var/lib/arpwatch/arp.dat

arpwatch -i vxlan3 -d./

**PRUEBAS DE QOS**

*Para el tráfico de bajada:*

*Desde la consola de vyos iperf a h11:*

docker exec -ti [id-vyos] bash -c 'su - vyos'

iperf -c 192.168.255.5 -b 14M -l 1200

*Desde h11 (escucha):*

iperf -s -i 1

*Para el tráfico de subida:*

*Desde s11 (escucha):*

Iperf -s -i 1

*Desde h11 (manda a s11):*

iperf -c 10.2.2.2 -b 14M -l 1200

4. Pruebas de funcionamiento

Una vez ya tenemos creado el escenario podemos probar si se ha realizado según las especificaciones del proyecto. Se han realizado tres pruebas, listadas a continuación:

* Prueba de conexión entre host y servidores
* Funcionamiento de la calidad de servicio mediante iperf
* Funcionamiento de arpwatch

4.1 Prueba de conexión

En las siguientes imágenes vemos como desde h11, con dirección IP 192.168.255.5, de la red residencial 1, se realiza un ping a la dirección IP 10.2.2.2, que corresponde a uno de los servidores, s1 concretamente.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

4.2 Prueba de Qos

Para realizar estas pruebas nos ayudaremos de la herramienta iperf, usada anteriormente en las prácticas de la asignatura.

Si deseamos comprobar la QoS en el sentido de bajada, entonces es necesario generar el tráfico desde el VyOS y escuchar ese tráfico con h11, por ejemplo. Para ello ejecutaremos los siguientes comandos:

root@h11:# iperf -s -i 1

vyos@vyos:# iperf -c 192.168.255.5 -b 14m -l 1200

Con esto generaremos un tráfico de 14Mbps de s1 a h11, y veremos como el límite de 12Mbps que hemos creado para la red residencial no permitirá que el ancho de banda exceda de ese valor.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

En el caso de h12 vemos como ese límite se encuentra en 4Mbps.

3.2.3 Prueba de arpwatch

La prueba de arpwatch consiste en entrar en el contenedor que hemos creado para ello y ejecutar el siguiente comando:

arpwatch -i vxlan3 -d

Mientras tanto, tendremos a un host, por ejemplo, h11, realizando un ping constante a una dirección IP de la red residencia que no exista, por ejemplo, a la 192.168.255.50.

Con el comando anterior lo que veremos es lo que se muestra en la siguiente imagen. Saldrá un mensaje del host que manda mensajes ARP junto con otra información útil. Además, esta información se guarda en el fichero /var/lib/arpwatch/arp.dat.

Texto

Descripción generada automáticamente