

Plataforma de orquestación de servicios basados en NFV

Resumen

En esta práctica, se va a utilizar la plataforma de código abierto Open Source Mano (OSM) para profundizar en las funciones de red virtualizadas y su orquestación. El escenario que se va a utilizar está inspirado en la reconversión de las centrales locales a centros de datos que permiten, entre otras cosas, reemplazar servicios de red ofrecidos mediante hardware específico y propietario por servicios de red definidos por software sobre hardware de propósito general. Las funciones de red que se despliegan en estas centrales se gestionan mediante una plataforma de orquestación como OSM o XOS.

El servicio de red objeto de estudio es el servicio residencial de acceso a Internet. La Figura 1 ilustra las funciones que tradicionalmente realiza el “router residencial” (Customer Premises Equipment – CPE) desplegado en casa del usuario, como switch Ethernet / punto de acceso WiFi, servidor DHCP, traducción de direcciones NAT y reenvío de datagramas IP. El objetivo de la práctica es estudiar como esas funciones pasarán a realizarse en la central local. Como se observa en la Figura 2, el router residencial se sustituye por un equipo que llamaremos “Bridged Residential Gateway (BRG)” que realiza la conmutación de nivel 2 del tráfico de los usuarios entre la red residencial y la central local. El resto de las funciones (DHCP, NAT y router para reenvío IP) se realizan en la central local aplicando técnicas de virtualización de funciones de red (NFV), creando un servicio de CPE virtual (vCPE) gestionado mediante la plataforma de orquestación.

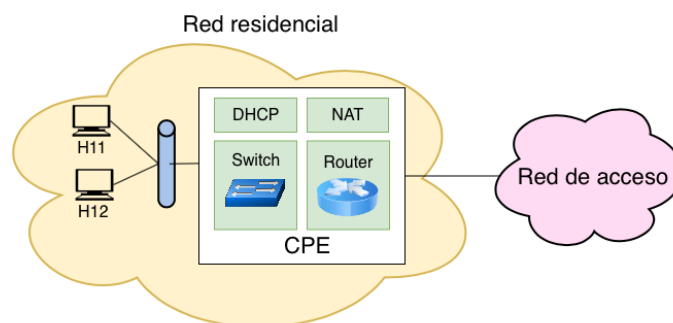


Fig. 1. CPE tradicional

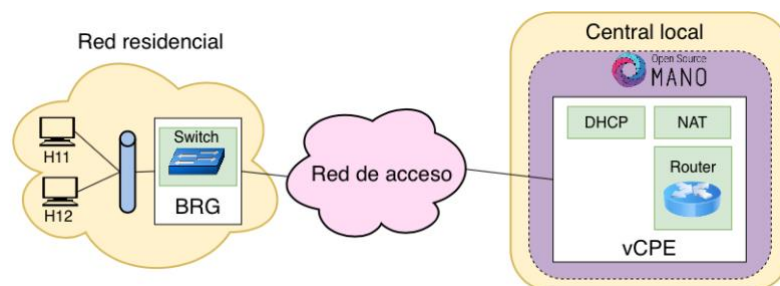


Fig. 2. CPE virtualizado

Escenario

La Figura 3 muestra una visión global del escenario que se va a emular, con dos sistemas finales h11 y h12 en casa del usuario, conectados al brg1 que, a través de la red de acceso AccessNet se conecta a su vez a la central local, donde el **servicio de red vCPE** se va a ofrecer a través de dos VNF encadenadas:

- Una VNF:vclass, que permitiría clasificar el tráfico e implementar políticas de QoS en el acceso del usuario a la red
- Una VNF:vcpe que, tal y como muestra la Figura 2, debe integrar las funciones de servidor DHCP, NAT y reenvío de IP.

El entorno utilizado para gestionar los servicios de red es OSM.

Como se ve en la Figura 3, desde la VNF:vcpe se accederá a la red pública a través del router r1. En el escenario, se desplegará un servidor s1 para emular un servidor en Internet. A su vez, r1 proporcionará acceso a la Internet “real”.

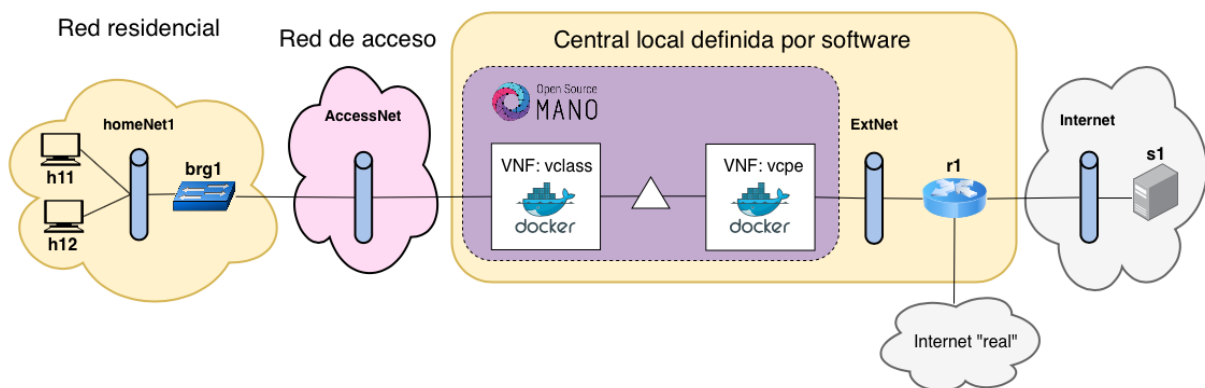


Fig. 3. Visión global del escenario

El escenario de la práctica se va a desplegar en una máquina Linux en VirtualBox, que ya tiene instaladas todas las herramientas necesarias, entre ellas:

- el entorno de **OSM**, al que se accede a través de un navegador
- la infraestructura de NFV (NFVI), controlada por OSM, implementada mediante la plataforma de emulación **vim-emu**¹, que permite la ejecución de las VNFs empaquetadas en forma de contenedores Docker
- la herramienta **VNX**, que se usará para emular los equipos de la red residencial, el router r1 y el servidor s1
- **Open vSwitch (ovs)**, que por un lado es la herramienta que utiliza internamente vim-emu para implementar las interconexiones entre las VNFs, y que por otro lado, se ha utilizado en el escenario de la práctica para emular: el conmutador brg1 de la red residencial, la red de acceso AccessNet y la red externa ExtNet que da salida al router r1.
- Scripts en la carpeta ~/bin para facilitar la gestión del entorno

¹ https://osm.etsi.org/wikipub/index.php/VIM_emulator

El detalle del escenario se puede ver en la Figura 4.

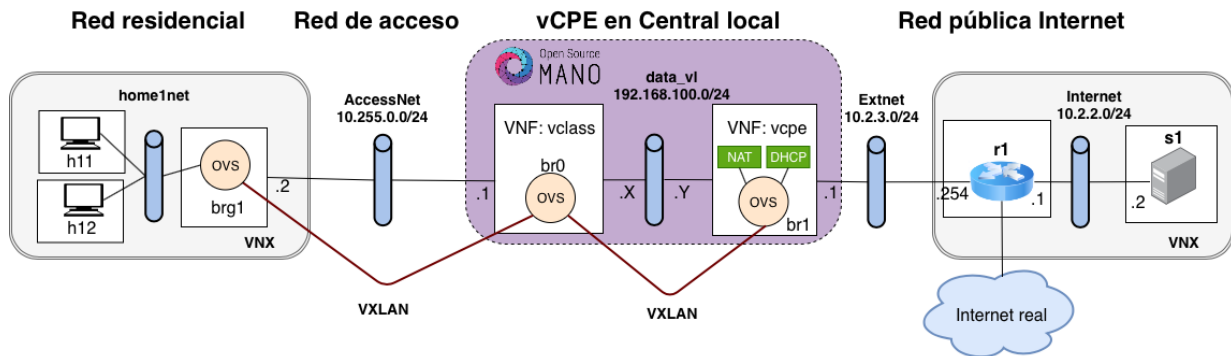


Fig. 4. Escenario detallado

Como ilustra la figura, se utiliza la tecnología **VXLAN** para enviar encapsuladas en datagramas UDP las tramas de nivel 2 que viajan entre brg1, VNF:vclass y VNF:vcpe. Para permitir esta comunicación, tanto brg1 como VNF:vclass tienen interfaces en AccessNet, configuradas con direcciones IP del prefijo 10.255.0.0/24. La asignación de direcciones IP a VNF: vclass y VNF:vcpe en la red que las interconecta (data_vl), está gestionada por OSM, y utiliza direcciones IP del prefijo 192.168.100.0/24.

Desarrollo de la práctica

Para ahorrar tiempo, la ova de la máquina virtual, accesible a través de <https://idefix.dit.upm.es/download/vnx/vnx-vm/VNXSDNNFVLAB2019-v2.ova> (unos 9GB) ha sido copiada e instalada en el laboratorio. En la medida de lo posible, no reorganice el PC del laboratorio, pues la instalación se realiza en un disco temporal y se perdería.

Desde la máquina Linux arrancada con VirtualBox, abra un navegador y descargue el fichero <https://idefix.dit.upm.es/download/rdsv/nfv/NFV-LAB-2019.tgz>. Ábralo y arrastre la carpeta que contiene (NFV-LAB-2019) al Escritorio.

Abra a continuación una ventana de terminal, acceda a la carpeta ~/Desktop/NFV-LAB-2019 y siga los siguientes pasos:

1. Cree los openvswitch AccessNet y ExtNet tecleando el comando:

```
./init.sh
```

Puede comprobar que están creados con el comando

```
sudo ovs-vsctl show
```

2. Acceda mediante el navegador a la interfaz de OSM en localhost. El nombre de usuario es "admin" y la contraseña (también "admin") está preconfigurada.

3. Compruebe en OSM que el VIM está enlazado en el menú VIM Accounts, tal y como muestra la Figura 5. En caso contrario, inicialice vim-emu tecleando el comando

```
osm-restart-vimemu
```

Registered VIM + New VIM

Show entries Search:



Name	Identifier	Type	Operational State	Description	Actions
emu-vim6	a74b6228-62f8-4759-91ea-6f057d4243fb	openstack	ENABLED		 

Fig. 5. Escenario detallado

4. Cree la imagen docker que va a implementar las dos VNFs mediante el fichero Dockerfile que se encuentra en el Directorio "vnf-img":

```
cd vnf-img
sudo docker build -t vnf-img .
```


5. Mientras la imagen docker se crea (puede tardar unos minutos), acceda gráficamente a través del Escritorio a la carpeta NfV-LAB-2019 y observe que hay tres ficheros tar.gz. Dos de ellos se corresponden con los descriptores de las VNFs y el otro con el descriptor del servicio de red. Puede realizar ya el proceso de instalación de los descriptores (onboarding), arrastrando los ficheros .tar.gz de los descriptores a través de los menús Packages->VNF packages y Packages->NS packages.

6. Acceda a la descripción de las VNF y del servicio. Para entregar como resultado de la práctica:

- En la descripción de las VNFs, identifique y copie la información referente a la imagen que se va a utilizar para el contenedor.
- En la descripción del servicio, identifique y copie la información referente a la interconexión de las dos VNFs.

7. Vuelva al terminal donde se estaba creando la imagen vnf-img. Cuando haya terminado, compruebe que la imagen *vnf-img* aparece en docker:

```
sudo docker images
```

8. Desde OSM, despliegue a través del menú NS Packages una instancia del servicio vCPE, pulsando en el símbolo . Deberá elegir el nombre **vcpe-1** para identificar el servicio.

9. Una vez que se haya instanciado el servicio, abra dos nuevas ventanas de terminal para acceder, a través de docker, a los contenedores de las VNF instanciadas. Para ello teclee los comandos:

Ventana para VNF:vcass

```
sudo docker exec -it mn.dc1_vcpe-1-1-ubuntu-1 bash
```

Ventana para VNF:vcpe

```
sudo docker exec -it mn.dc1_vcpe-1-2-ubuntu-1 bash
```

Y a continuación, en cada una de las máquinas, averigüe mediante ifconfig la dirección de la interfaz "eth1-x" (x varía según la instancia de la VNF) que se crea desde OSM para realizar la interconexión de las dos VNFs en la subred 192.168.100.0/24. Compruebe que hay conectividad entre ellas ejecutando ping.

10. Realice una captura de tráfico en VNF:vcpe en la interfaz eth1-x mediante:

Ventana para VNF:vcpe

```
tcpdump -i eth1-x
```

Ventana para VNF:vclass

```
ping -c1 <dir_ip_de_VNF:vcpe>
```

Copie el texto de la captura realizada para entregarlo como resultado de la práctica.

11. Desde el terminal en NFV-LAB-2019 arranque el escenario vnx de las redes residenciales:

```
sudo vnx -f nfsv3_home_lxc_ubuntu64.xml -t
```

El escenario contiene dos redes residenciales, nos centraremos inicialmente en la primera de ellas (sistemas finales h11 y h12). Acceda a los terminales de los hosts h11 y h12 y compruebe que no tienen asignada dirección IP mediante:

```
ifconfig
```

Compruebe también que el cliente DHCP no les permite obtener dirección IP:

```
dhclient  
ifconfig
```

12. Compruebe a través del terminal de brg1 la dirección IP asignada para su comunicación a través de AccessNet.

13. Debido a las limitaciones del escenario emulado con OSM + vim-emu, se necesita realizar una serie de configuraciones del servicio instanciado directamente, accediendo a los contenedores mediante docker. El fichero vcpe_start.sh permite realizar las configuraciones necesarias del servicio vCPE. Ejecútelo para ver cómo se usa, identificando qué parámetros del escenario permite configurar. Puede también acceder al contenido del fichero usando nano.

14. Acceda al contenido del fichero vcpe1.sh, utilizado para configurar la instancia vcpe-1 del servicio:

```
cat vcpe1.sh
```

Identifique, según el contenido de ese fichero, la dirección IP del vCPE en la red privada de casa, y la dirección IP “pública” (en realidad es de un rango privado), que usará el NAT para dar salida al tráfico de la red residencial hacia Internet.

15. Configure el servicio ejecutando

```
./vcpe1.sh
```

16. Vuelva a comprobar la configuración de red de h11 y h12 y, si no han obtenido dirección IP, fuerce el acceso al servidor DHCP mediante el comando:

```
dhclient
```

Indique qué direcciones IP obtienen h11 y h12 en la red residencial “privada”, así como la dirección IP del router residencial virtualizado.

17. Arranque wireshark y póngalo a capturar el tráfico en **brg1-e2**. Desde h11 realice un ping de 5 paquetes a la dirección IP de su router, comprobando que funciona correctamente.

```
ping -c 5 <dir_IP_router_residencial>
```

Detenga wireshark, y guarde la captura con nombre “access1.pcapng”. Analice el tráfico capturado, justificando las cuatro direcciones IP que aparecen en los paquetes capturados, teniendo en cuenta el túnel VXLAN.

18. Arranque el escenario vnx de la red pública y el servidor.

```
sudo vnx -f nfsv3_server_lxc_ubuntu64.xml -t
```

19. Arranque wireshark y póngalo a capturar el tráfico en la red **Internet**. Desde h11 realice un ping de 5 paquetes a la dirección IP de s1 (10.2.2.2), comprobando que funciona correctamente.

```
ping -c 5 10.2.2.2
```

Detenga wireshark, y guarde la captura con nombre “internet1.pcapng”. Analice el tráfico capturado, justificando las direcciones IP que aparecen en los paquetes capturados.

20. Desde h11 compruebe el camino seguido por el tráfico a los sistemas del escenario y a Internet:

```
tracert -d <dir_IP_router_residencial>
```

```
tracert -d 10.2.2.2
```

```
tracert -d 8.8.8.8
```

21. A continuación, realice los pasos necesarios para dar acceso a Internet a la segunda red residencial (h21, h22). Compruebe que funciona correctamente, y que a su vez sigue funcionando la primera red residencial (h11, h12).

22. Repita las capturas wireshark realizadas anteriormente en AccessNet e Internet, ejecutando en este caso los pings desde h21.

Guarde las capturas con nombres access2.pcapng y internet2.pcapng respectivamente.

Entrega de resultados

Suba a través del Moodle un único fichero zip que incluya las cuatro capturas y un fichero pdf con las respuestas a los apartados resaltados en el enunciado.