Práctica 1 - Virtualización con VirtualBox

1. Introducción y objetivos

El objetivo de esta práctica es familiarizarse con los entornos de virtualización, en concreto con algunas funcionalidades de VirtualBox similares (aunque de forma simplificada) a lo que se puede emplear en un entorno de servidor en el centro de datos. Practicaremos con la posibilidad de crear redes internas donde algunos hosts virtuales puedan actuar por ejemplo como routers o servidores (o balanceadores, o firewalls, o proxies, o IDS...), así como con las diferentes formas de dar acceso al exterior del host a las VMs.

Esta práctica no pretende ser un tutorial paso-a-paso. Se plantean los objetivos de cada apartado pero puede haber múltiples formas de alcanzarlos, incluso se puede llevar a cabo la práctica en plataformas diferentes (Linux del laboratorio, macOS, Windows). Lea todo el guión y preste atención a los comentarios y recomendaciones. Aproveche la práctica para interiorizar el comportamiento de diferentes modalidades de networking con VMs.

2. Networking virtual

VirtualBox permite diferentes formas de networking entre las máquinas virtuales y con el exterior. Consulte primero al capítulo de su manual al respecto para más detalle[1].

En este apartado se le pide que cree un escenario como el que se muestra en la Figura 1. En ella se muestran 3 subredes interconectadas por 2 routers y unos PCs de ejemplo en cada subred.

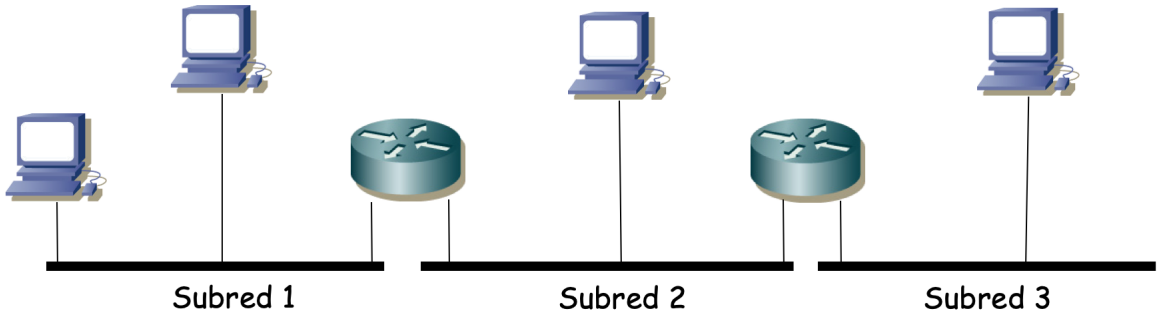

Figura 1 - Topología capa 3

Las subredes se formarán en capa 2 con conmutadores Ethernet (Figura 2).

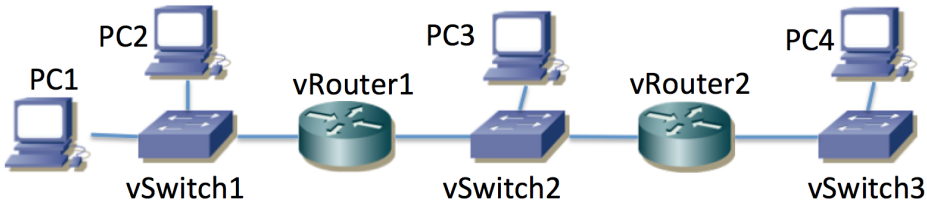

Figura 2 - Topología pseudo-física de red

Sin embargo, no vamos a llevar a cabo esta topología con equipos físicos sino que los equipos vRouter1 y vRouter2 serán máquinas virtuales Linux con dos interfaces Ethernet

[1] https://www.virtualbox.org/manual/ch06.html

actuando como routers, los equipos PC1, PC2, PC3 y PC4 serán máquinas virtuales con un interfaz de red actuando como PCs normales; y todos ellos serán guests en el mismo host empleando VirtualBox.

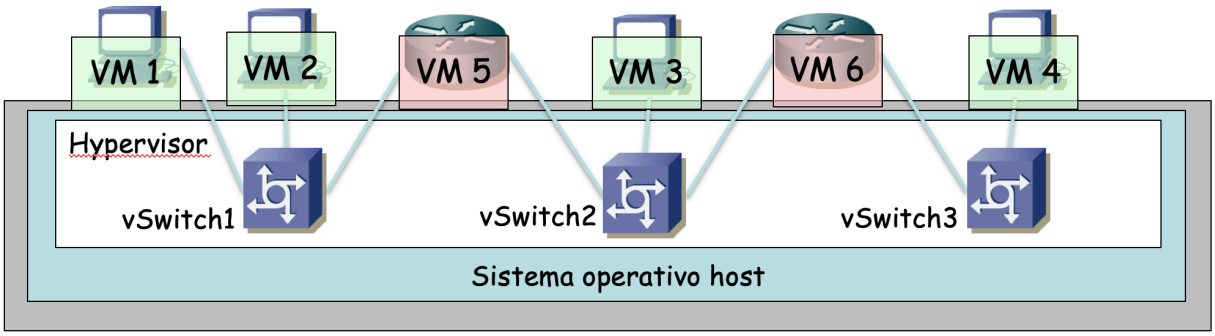

Figura 3 - Elementos virtualizados

Los equipos vSwitch1, vSwitch2 y vSwitch3 podrían ser equipos Linux actuando como puentes, sin embargo para este ejercicio se recomienda emplear la capacidad de crear redes internas (*Internal Network*) independientes en VirtualBox[2]. Cree tres redes internas para actuar como cada una de esas LANs. Estas redes internas, para un uso "normal" se comportan casi como un conmutador, aunque tenga en cuenta que no tienen exactamente el mismo comportamiento y en ciertos escenarios complejos pueden tener un comportamiento inesperado.

Consulte la documentación de VirtualBox sobre las redes internas. No necesita crearlas explícitamente. Dos interfaces de VMs estarán en la misma internal network si le ha puesto el mismo nombre al asignarla a los interfaces (Figura 4).

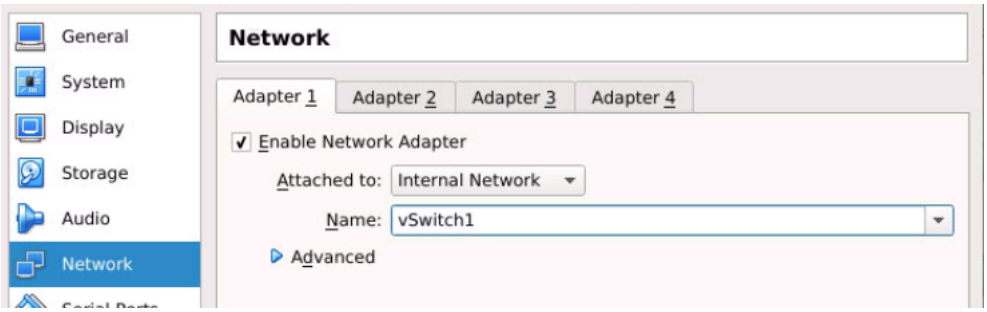

Figura 4 - Configuración de vNIC en una internal network

Configure direcciones IP y rutas en todos los PCs y routers y compruebe la conectividad y caminos con traceroute. Si la instalación de Linux que ha empleado para los PCs incluye servidor y cliente de ssh puede probar también a acceder de una máquina virtual a la otra.

Punto de control 1 (70%): Muestre el escenario completo en funcionamiento al profesor. Por ejemplo pruebe a hacer traceroute entre cualquier par de PCs del escenario virtual.

Recomendaciones, pistas, consejos y comentarios

- No hace falta que ninguno de los PCs incluya un interfaz gráfico aunque puede resultarle más cómodo para emplear algún analizador de tráfico (wireshark, tcpdump).
- Vaya creando la topología y probándola poco a poco. Por ejemplo cree primero las VMs para PC1 y PC2 conectadas a la misma Internal Network, configúrelas y comprueba la conectividad. A continuación añada una tercera máquina (vRouter1) que tenga dos

[2] https://www.virtualbox.org/manual/ch06.html#network_internal

interfaces de red, uno en la Internal Network vSwitch1 y el segundo en la Internal Network vSwitch2. Continúe paso a paso. Recuerde configurar rutas (por ejemplo rutas por defecto) en los PCs (en PC3 puede elegir como siguiente salto en la ruta por defecto a vRouter1 o a vRouter2). En los routers puede configurar ruta por defecto hacia el otro o ruta en concreto a la subred remota.

- Para los guest puede emplear una instalación pequeña de Linux, por ejemplo una *Slitaz*[3]. Si escoge una distribución pensada para escritorio tenga en cuenta que probablemente necesite activar el reenvío de paquetes con el comando *sysctl*[4].
- Algunos Linux pequeños montan el sistema de ficheros en un RAM disk, así que los cambios a ficheros no permanecen tras un reinicio del mismo (pero esto probablemente no resulte trascendente en esta práctica). Un *Linux Core* o una *Slitaz* puede correrla como un LiveCD o hacer una instalación local. Consulte en la web de la asignatura si hay disponible alguna instalación ya hecha, aunque siempre es un buen ejercicio hacer una propia (pero se le recomienda hacerlo fuera del horario de prácticas para aprovechar el tiempo del mismo).
- Para ahorrar espacio en disco puede crear una máquina virtual y clonarla varias veces (clon enlazado o "*linked clone*") en VirtualBox[5]. Cree una VM donde instale un sistema operativo para hacer de uno de los PCs (o incluso de router) y las demás VMs puede hacerlas clones de la anterior. Un clon enlazado no copia todo el disco de la VM sino que en el clon va apuntando los cambios que se van haciendo en disco (*copy-on-write*). Tenga cuidado con los clones pues son idénticos. Eso quiere decir que los interfaces de red tienen la misma dirección MAC. Esto será un problema si emplea dos clones en la misma internal network (son dos PCs con la misma dirección MAC en la misma LAN). En el wizard de clonado tiene opciones para que al clonar se cambie la dirección MAC de los interfaces en el clon (Figura 5). Puede borrar los clones pero si borra la VM de la que son clones (si son enlazados) evidentemente tendrá problemas con los clones. En algunas distribuciones de Linux (por ejemplo Slitaz) y VirtualBox hemos detectado que en la clonación cambia el nombre del interfaz de red (de eth0 a eth1); en tal caso puede que necesite editar algún script de arranque (en la Slitaz vale con editar /etc/network y cambiar INTERFACE="eth0" por eth1).

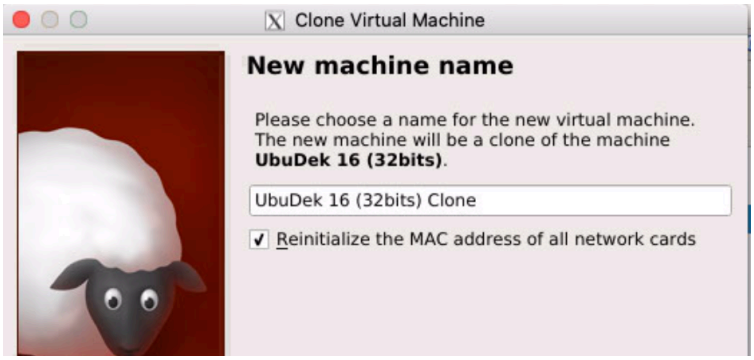

Figura 5 - Clonación cambiando direcciones MAC

- Use las snapshots de VirtualBox[6] (no confunda esto con capturas de pantalla, lo que se está guardando es todo el estado de la VM). Esto le permitirá por ejemplo volver a un estado anterior de la VM si ha tenido algún problema con ella. Por ejemplo, una vez que haga la instalación del sistema operativo en la VM haga una snapshot. Cuando haga

---

[3] http://www.slitaz.org

[4] `sysctl net.ipv4.ip_forward=1`

[5] https://www.virtualbox.org/manual/ch01.html#fig-clone-wizard

[6] https://www.virtualbox.org/manual/ch01.html#snapshots

cambios en la VM y vea que son estables haga otra snapshot (Figura 6). Si en algún momento tiene un problema con la VM podrá volver al estado del snapshot que desee. Puede incluso crear clones a partir del estado de un snapshot. Ahorrará mucho tiempo si tiene problemas.

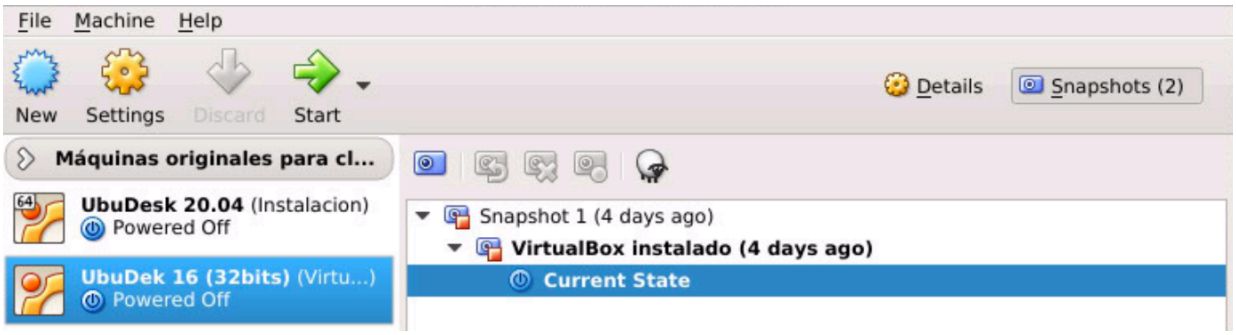

Figura 6 - Snapshots

- Si emplea un guest con un sistema operativo de escritorio como puede ser un *Ubuntu Desktop* tenga en cuenta que normalmente hay servicios del sistema encargados de configurar automáticamente los interfaces de red. Puede que usted vaya a configurar un interfaz de red en línea de comandos (comandos *ifconfig* o *ip*) y se encuentre al rato con que el sistema le ha cambiado la configuración. Normalmente se puede desactivar ese comportamiento con el panel de control correspondiente a la configuración de red en ese sistema operativo de escritorio (también puede ahorrarse problemas de este tipo y ceder ante la tentación de configurar el interfaz de red con dicho panel de control).
- Las cuentas de alumnos en los ordenadores del laboratorio guardan su directorio *HOME* en un servidor, montado mediante NFS. Cada cuenta de usuario tiene asignado un máximo espacio que puede ocupar en ese directorio. Puede ver lo que está ocupando con el comando *quota*

```
$ quota
Disk quotas for user rng40 (uid 23640):
     Filesystem  blocks   quota   limit   grace   files   quota   limit   grace
10.1.1.199:/opt3
                    152  500000  500000              34  100000  100000
```

(En ejemplos de línea de comandos empleo '$' para representar el prompt de la Shell, no es parte del comando).

Tenga cuidado con actividades como por ejemplo la descarga de grandes ficheros a su directorio de usuario pues consumen de ese espacio y tendrá problemas si alcanza el máximo. Puede crear ficheros más grandes en /VB/rng/rngXY donde rngXY sea su cuenta de usuario. Tenga en cuenta que esos ficheros no estarán disponibles en otras máquinas del laboratorio pues se encuentran en el disco local. Tenga cuidado de no llenar el disco local.

- Se ha dejado configurado VirtualBox para que las VMs se guarden en el directorio VirtualBoxVM en el directorio de cada usuario, el cual es un *link* a un directorio dentro del disco local de la máquina (en /VB/rng). Esto quiere decir que una máquina que cree en un ordenador del laboratorio (si no cambia nada) no podrá lanzarla en otro pues no tiene acceso a estos ficheros, aunque la vea en el listado de VMs (pues esto está en un fichero de configuración de VirtualBox que sí está en la parte compartida de su HOME).

Su directorio HOME es accesible desde cualquier máquina del laboratorio, sin embargo, tenga en cuenta que el directorio local no está compartido con otras máquinas, así que lo que deje ahí estará solo en el ordenador donde lo guardó. También vigile su uso del disco pues si lo llena empezará a tener problemas de uso del ordenador.

3. VBoxManage

El GUI de VirtualBox es cómodo pero no ofrece todas las funcionalidades que tiene la herramienta ni es cómodo de automatizar. Para ello se dispone de la utilidad de línea de comandos `VBoxManage`. Tiene su documentación en el manual en línea (`man VBoxManage`) o un resumen ejecutando el programa sin opciones.

Por ejemplo, pruebe a obtener un listado de las VMs disponibles con:

```
$ VBoxManage list vms
```

El listado contiene tanto el nombre de las VMs como un identificador alfanumérico. Puede emplear cualquiera de los dos para identificar a una VM cuando quiera actuar sobre ella con este comando. Por ejemplo, puede lanzar la VM con:

```
$ VBoxManage startvm <VMid>
```

(Sustituya <VMid> por el identificador o el nombre de la VM).

Como puede ver, en general tras VBoxManage se especifica un comando y sus opciones. El comando `modifyvm` le permitirá cambiar la configuración de la VM mientras que el comando `controlvm` permite todo tipo de acciones sobre ella (detenerla, apagarla, etc).

Inspeccione opciones del comando.

4. Reenvío por puerto (*Port Forwarding*)

Seguramente esté acostumbrado a emplear interfaces en las VMs que están conectados a un NAT. Esta es la configuración por defecto de las VMs creadas en VirtualBox para que puedan acceder a Internet a través del interfaz del host. VirtualBox actúa como NAT para la VM. Cada VM que tenga el interfaz "Attached to: NAT" está aislada del resto de VMs. El NAT incluye un servidor de DHCP y ofrece a todas las VMs conectadas a NAT la misma dirección IP.

Igual que en un típico NAT en hardware, se puede configurar el reenvío en base al puerto de transporte destino de forma que máquinas externas al host puedan contactar con servicios en una VM. La configuración de *Port Forwarding* se hace en la parte de configuración de red de la VM en el GUI de VirtualBox, bajo el desplegable de opciones avanzadas (Figura 7).

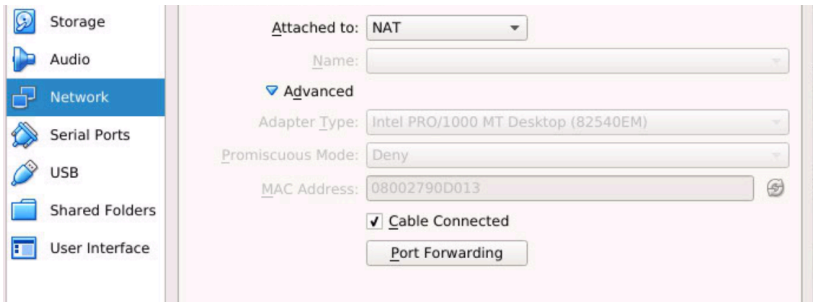

Figura 7 - Opciones de configuración de red

Prepare una VM con un servidor (por ejemplo un servidor ssh o un servidor web), exponga ese servidor en el host mediante una regla de *port forwarding* y pruebe a contactar con ese servidor desde otro host del laboratorio. Puede hacer la configuración también desde línea de comandos empleando VBoxManage[7].

[7] https://www.virtualbox.org/manual/UserManual.html#natforward

Punto de control 2 (15%): Muestre que puede acceder a un servicio de una VM desde un host en la LAN.

Podría también desde una VM en un host acceder a una VM en otro host mediante *Port forwarding* en la segunda. Estando ambas VMs tras un NAT de su hypervisor examine cómo cambian los paquetes IP entre que los envía una VM, pasan por la LAN física (los reenvía el host) y llegan a la otra VM.

5. Bridged network

Pruebe a continuación a colocar un par de VMs puenteadas con el interfaz físico del host (*Bridged Network,* Figura 8). Si está haciendo esto en el laboratorio necesitará direcciones IP para configurar manualmente en esas VMs pues el servidor de DHCP del laboratorio no les va a entregar dirección IP al no reconocer sus direcciones MAC. No elija una dirección IP cualquiera, pregunte al profesor de prácticas pues hay direcciones de la subred reservadas para este propósito. Compruebe que el tráfico que envían las VMs llega a otras máquinas del laboratorio siendo puenteadas en todo el camino (no habrá cambios en la cabecera Ethernet en el host del laboratorio respecto a como la construye la VM).

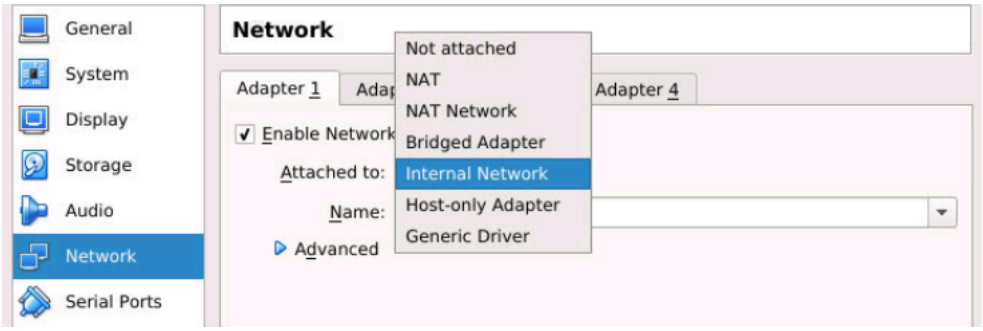

Figura 8 - Opciones de networking en el GUI de VirtualBox

Punto de control 3 (10%): Muestre una prueba de que la VM está puenteada a la LAN comprobando en otro PC del laboratorio (de la misma LAN) que ve las tramas que le llegan de la VM con la dirección MAC origen de la vNIC de la misma.

6. Puentes con VMs

A continuación añadiremos una máquina virtual que actúe como un bridge en lugar de como un router. Dicha máquina virtual será también un Linux. Cree dos VMs que actúen como PCs y una tercera VM con dos interfaces que será el puente. Interconecte cada PC con la VM-puente, para ello conecte el interfaz del primer PC con el primer interfaz de la VM-puente mediante una *internal network* y el segundo PC con el segundo interfaz de la VM-puente mediante otra *internal network* diferente. En los interfaces de la VM que actuará como puente configure en las opciones avanzadas que permitan modo promiscuo.

En la VM-puente emplee las *bridge-utils* para crear el puente y configurarlo. El comando principal es `brctl` y puede crear el puente con la opción `addbr` y añadir al puente creado interfaces con la opción `addif`.

Punto de control 4 (5%): Muestre una prueba de que la VM-puente está puenteando entre las otras VMs comprobando en una de ellas que ve las tramas que le llegan de la otra con la dirección MAC origen de la vNIC de la misma.