

# Configuraciones de red en sistemas virtuales

Alberto Molina Coballes <alberto.molina@hispalinux.es>  
José Domingo Muñoz Rodríguez <josedom24@gmail.com>  
IES Gonzalo Nazareno. Dos Hermanas (Sevilla)

16 de septiembre de 2006

## Resumen

En este documento se describe las diferentes configuraciones de red elementales que pueden utilizarse en Xen 3.0 y VMware Server 1.0. Esta documentación se elaboró para el curso *Máquinas virtuales para la puesta en marcha de un portal educativo* organizado por el CEP de Sevilla en Septiembre de 2006.

# 1. Introducción

En este artículo se explican los aspectos principales de las configuraciones de red que pueden tener lugar entre las máquinas virtuales huésped y el anfitrión. Hay bastantes aspectos comunes en el tratamiento que hacen al respecto Xen y VMware Server, aunque es conveniente exponerlos de forma separada.

## 2. Xen 3.0

Lo explicado en esta sección puede encontrarse de forma más extensa en la parte relativa a redes del wiki de Xen [1].

Existen tres configuraciones elementales de red en Xen:

- Puente (*bridge*)
- Encaminador (*router*)
- NAT

Pueden construirse configuraciones mucho más complejas a partir de éstas, pero en principio hay que decidirse por utilizar una de ellas, modificando de forma adecuada el fichero `/etc/xen/xen-config.sxp`

### 2.1. Puente

Para utilizar un puente debemos tener desmomentadas las siguientes líneas en el fichero de configuración de Xen:

```
(network-script network-bridge)
(vif-script vif-bridge)
```

Los pasos que en realidad se realizan son:

Cuando se lanza `xend`, se ejecuta el script `network-bridge`, que:

- Crea un nuevo puente llamado `xenbr0`.
- La interfaz *real* `eth0` se echa abajo.
- Se copian la dirección IP y MAC de `eth0` a la interfaz virtual `veth0`.
- Se renombra la interfaz real como `peth0`
- Se renombra la interfaz virtual como `eth0`
- `peth0` y `vif0.0` se conectan al puente `xenbr0`.
- Se levantan `xenbr0`, `peth0`, `eth0` y `vif0.0`.

Cuando se inicia un `domU`, `xend` (que está corriendo en `dom0`) ejecuta el script `vif-bridge`, que:

- Conecta `vif<id#>.0` to `xenbr0`
- Levanta `vif<id#>.0`

Donde `id#` es el número de identificación del domU.

En la figura 1 puede verse de forma esquemática como quedan conectados los dominios tras configurar un puente en Xen con un domU.

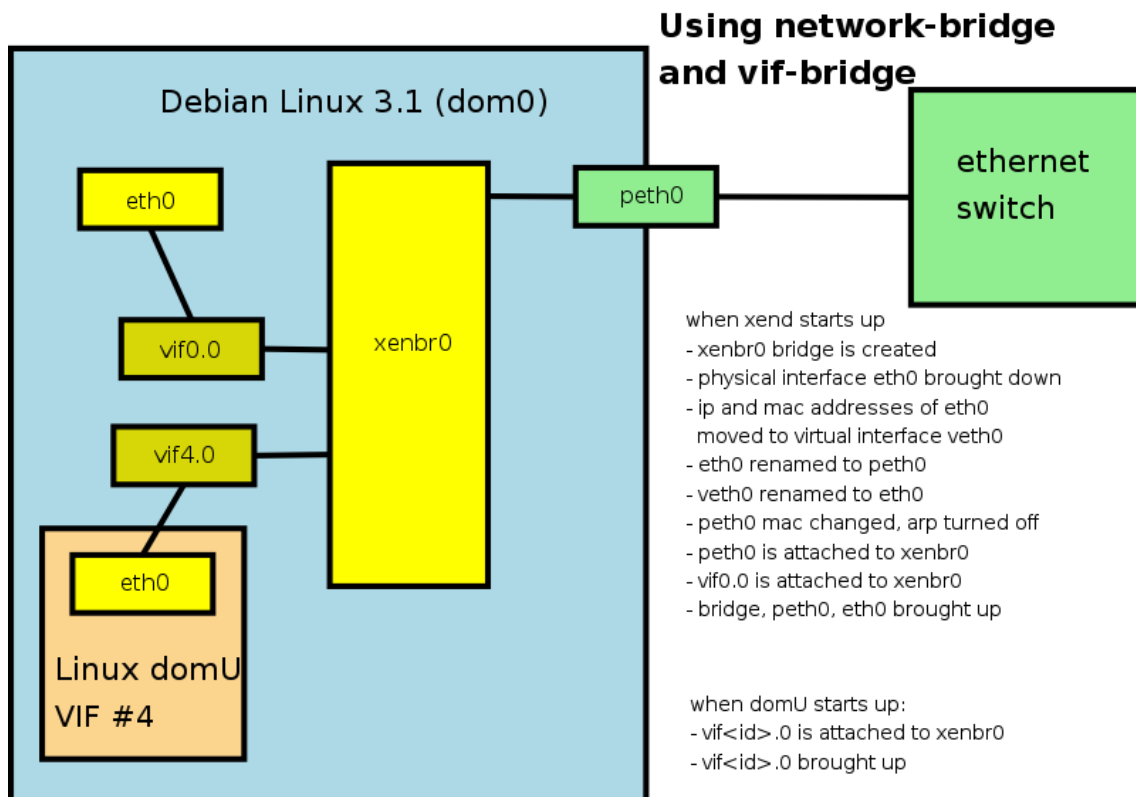


Figura 1: A través del puente se conecta un domU a la misma red que el dom0, utilizando su interfaz de red

La salida de `ifconfig` en este caso es:

```

eth0      Link encap:Ethernet  HWaddr 00:40:D0:DD:B2:84
          inet addr:192.168.0.68 Bcast:192.168.0.255 Mask:255.255.255.0
          inet6 addr: fe80::240:d0ff:fe65:b284/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:13 errors:0 dropped:0 overruns:0 frame:0
          TX packets:7 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:1248 (1.2 KiB)  TX bytes:810 (810.0 b)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:8 errors:0 dropped:0 overruns:0 frame:0
          TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:560 (560.0 b)  TX bytes:560 (560.0 b)

peth0     Link encap:Ethernet  HWaddr FE:FF:FF:FF:FF:FF
  
```

```

inet6 addr: fe80::fcff:ffff:feff:ffff/64 Scope:Link
UP BROADCAST RUNNING NOARP MULTICAST MTU:1500 Metric:1
RX packets:12 errors:0 dropped:0 overruns:0 frame:0
TX packets:21 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:1674 (1.6 KiB) TX bytes:3450 (3.3 KiB)
Interrupt:3 Base address:0xc000

vif0.0 Link encap:Ethernet HWaddr FE:FF:FF:FF:FF:FF
inet6 addr: fe80::fcff:ffff:feff:ffff/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:7 errors:0 dropped:0 overruns:0 frame:0
TX packets:14 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:810 (810.0 b) TX bytes:1338 (1.3 KiB)

xenbr0 Link encap:Ethernet HWaddr FE:FF:FF:FF:FF:FF
inet6 addr: fe80::200:ff:fe00:0/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:14 errors:0 dropped:0 overruns:0 frame:0
TX packets:6 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:1142 (1.1 KiB) TX bytes:468 (468.0 b)

```

Donde podemos observar que la interfaz de red real `peth0` no tiene una dirección MAC real, la tiene `eth0`. Cuando se le añade un nuevo dominio, la interfaz `eth0` del nuevo dominio tendrá una dirección MAC real (con Xen como fabricante). Es decir, el comportamiento conjunto de todo es como varias máquinas reales conectadas a un switch, con `peth0` equivalente a la conexión de salida del switch.

Esta configuración es útil por ejemplo cuando la máquina se encuentra en una red local y podemos utilizar todas las direcciones IP que queramos dentro del mismo rango.

## 2.2. Encaminador

Para que Xen se comporte como un encaminador, hay que descomentar las líneas:

```

(network-script network-route)
(vif-script vif-route)

```

Cuando se lanza `xend`, se ejecuta el script `network-route`, que:

- Activa el bit de forward en `dom0`<sup>1</sup>

Cuando se inicia un `domU`, `xend` ejecuta (dentro de `dom0`) `vif-route`, que:

- Copia la dirección IP de `eth0` a `vif<id#>.0`
- Levanta `vif<id#>.0`

---

<sup>1</sup>/proc/sys/net/ipv4/ip\_forward

- Añade una ruta estática en el domU, apuntando a la interfaz `vif<id>.0`

En la figura 2 puede verse de forma esquemática como quedan conectados los dominios tras configurar el dom0 como un encaminador. La salida de `ifconfig`

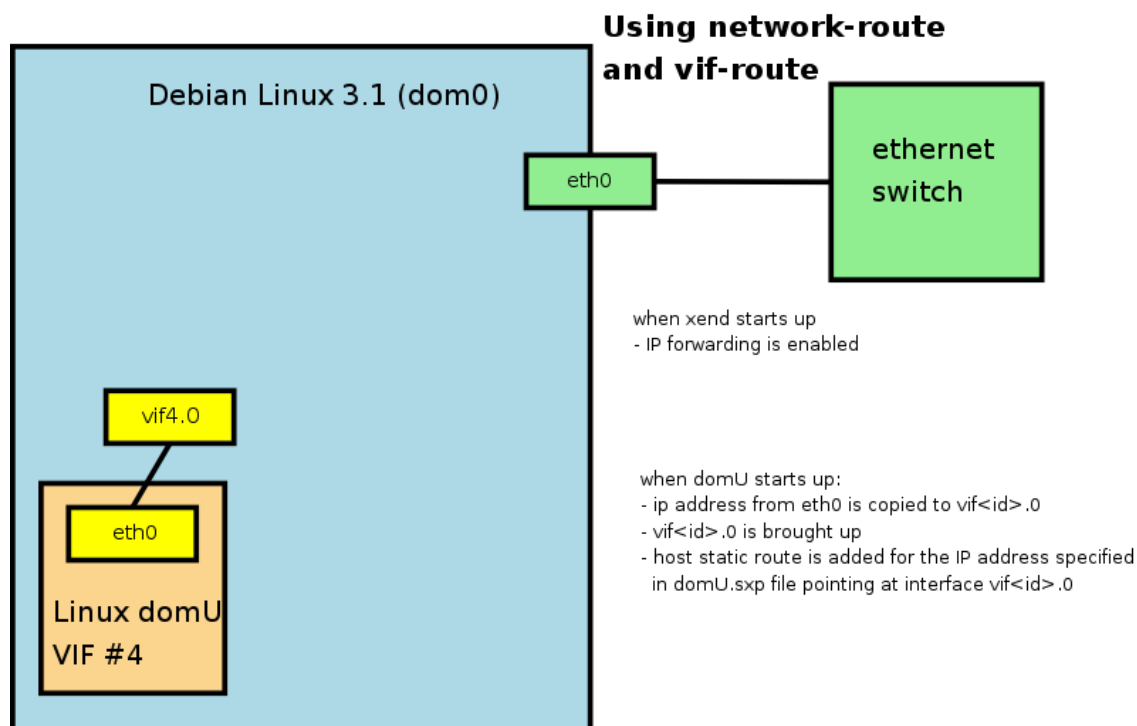


Figura 2: A través del puente se conecta un domU a la misma red que el dom0, utilizando su interfaz de red

en el dom0 es:

```
eth0  Link encap:Ethernet  HWaddr 00:40:D0:65:B2:84
      inet addr:192.168.0.64 Bcast:192.168.0.255 Mask:255.255.255.0
      inet6 addr: fe80::240:d0ff:fe65:b284/64 Scope:Link
      UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
      RX packets:73 errors:0 dropped:0 overruns:0 frame:0
      TX packets:47 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:1000
      RX bytes:6681 (6.5 KiB)  TX bytes:4766 (4.6 KiB)
      Interrupt:3 Base address:0xc000

lo    Link encap:Local Loopback
      inet addr:127.0.0.1  Mask:255.0.0.0
      inet6 addr: ::1/128 Scope:Host
      UP LOOPBACK RUNNING  MTU:16436  Metric:1
      RX packets:8 errors:0 dropped:0 overruns:0 frame:0
      TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:0
      RX bytes:560 (560.0 b)  TX bytes:560 (560.0 b)

vif1.0 Link encap:Ethernet  HWaddr FE:FF:FF:FF:FF:FF
      inet addr:192.168.0.64 Bcast:192.168.0.255 Mask:255.255.255.255
```

```

inet6 addr: fe80::fcff:ffff:feff:ffff/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:1719 errors:0 dropped:0 overruns:0 frame:0
TX packets:1726 errors:0 dropped:19 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:93686 (91.4 KiB) TX bytes:131506 (128.4 KiB)

```

y la de route -n:

```

Kernel IP routing table
Destination      Gateway          Genmask         Flags   Iface
192.168.0.100    0.0.0.0         255.255.255.255 UH      vif1.0
192.168.0.0      0.0.0.0         255.255.255.0   U       eth0
0.0.0.0          192.168.0.1    0.0.0.0         UG      eth0

```

Donde 192.168.0.100 es la dirección IP asignada al domU.

En el domU hay que poner como gateway el mismo que tenga el dom0, en este caso 192.168.0.1

Cuando las tramas salen por eth0, tienen la misma dirección MAC, para todas las direcciones IP. Dependiendo de la configuración del equipo esto puede ser un problema, que puede solucionarse incluyendo en el script network-route la línea:

```
echo 1 >/proc/sys/net/ipv4/conf/eth0/proxy_arp
```

Desde un equipo externo podremos comprobar que aparecen dos direcciones IP asociadas a la misma dirección MAC:

```
ranec:~# cat /proc/net/arp
```

IP address	HW address	Mask	Device
192.168.0.64	00:40:D0:65:B2:84	*	eth1
85.136.XX.X	00:05:00:E3:4D:DB	*	eth0
192.168.0.100	00:40:D0:65:B2:84	*	eth1

Es decir, cuando disponemos de varias direcciones IP dentro del mismo rango, tendremos que optar por utilizar una configuración de puente o una de encaminador, dependiendo de si el equipo al que estamos conectando nos acepta o no diferentes direcciones MAC, para las diversas direcciones IP. En el primer caso trabajaremos en el nivel de enlace de la pila TCP/IP y en el segundo caso en el nivel de red.

## 2.3. NAT

No existe documentación oficial sobre la configuración de redes con NAT en Xen, así que lo que se explica a continuación es fruto simplemente de la experiencia al configurarlo. Deben existir por tanto gran cantidad de errores o inexactitudes.

Para que Xen utilice NAT, hay que descomentar las líneas:

```
(network-script network-nat)
(vif-script vif-nat)
```

En este caso, el dom0 estará conectado a dos subredes diferentes, típicamente a una con direcciones públicas en eth0 y otra con direcciones privadas en vif<#>.0. A través del mecanismo de traducción de direcciones NAT, irá cambiando las direcciones origen/destino de cada paquete de forma adecuada.

Cuando se lanza xend, se ejecuta el script network-nat, que:

- Activa el bit de forward en dom0

Cuando se inicia un domU, xend ejecuta (dentro de dom0) vif-nat, que:

- Pone una dirección IP en vif<id#>.0, correspondiente al rango de la subred que se defina en domU<sup>2</sup>
- Levanta vif<id#>.0
- Añade las reglas de iptables correspondientes para hacer NAT.

El esquema es exactamente igual al de la figura 2, aunque ahora las direcciones IP de los domU no están en el mismo rango que la dirección IP “externa” del dom0.

La salida de ifconfig en el dom0 es:

```
eth0  Link encap:Ethernet  HWaddr 00:40:D0:65:B2:84
      inet addr:192.168.0.64 Bcast:192.168.0.255 Mask:255.255.255.0
      inet6 addr: fe80::240:d0ff:fe65:b284/64 Scope:Link
      UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
      RX packets:3946 errors:0 dropped:0 overruns:0 frame:0
      TX packets:2915 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:1000
      RX bytes:5385321 (5.1 MiB)  TX bytes:219429 (214.2 KiB)
      Interrupt:3 Base address:0xc000

lo    Link encap:Local Loopback
      inet addr:127.0.0.1  Mask:255.0.0.0
      inet6 addr: ::1/128 Scope:Host
      UP LOOPBACK RUNNING  MTU:16436  Metric:1
      RX packets:8 errors:0 dropped:0 overruns:0 frame:0
      TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:0
      RX bytes:560 (560.0 b)  TX bytes:560 (560.0 b)

vif1.0 Link encap:Ethernet  HWaddr FE:FF:FF:FF:FF:FF
      inet addr:10.0.0.227 Bcast:0.0.0.0 Mask:255.255.255.255
      inet6 addr: fe80::fcff:ffff:feff:ffff/64 Scope:Link
      UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
      RX packets:1 errors:0 dropped:0 overruns:0 frame:0
```

---

<sup>2</sup>Debería poner la dirección que se defina como gateway en el fichero de configuración del domU correspondiente, pero no es así: Se pone una dirección de forma aleatoria.

```
TX packets:3 errors:0 dropped:3 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:42 (42.0 b) TX bytes:230 (230.0 b)
```

y la de route -n:

```
Kernel IP routing table
Destination      Gateway          Genmask          Flags   Iface
10.0.0.100       0.0.0.0          255.255.255.255 UH      vif1.0
192.168.0.0      0.0.0.0          255.255.255.0   U       eth0
0.0.0.0          192.168.0.1     0.0.0.0         UG      eth0
```

donde podemos ver que la ruta a través de `vif1.0` es hacia una máquina (etiqueta “H”) con IP 10.0.0.100 y no un segmento de red. Si añadimos un nuevo dominio dentro del mismo rango, se creará una nueva interfaz virtual y se añadirá una nueva ruta. Si añadimos muchos dominios dentro del mismo rango puede ser muy farragoso después tener que ir añadiendo reglas de iptables para cada interfaz virtual.

En el domU, es indiferente el gateway que se ponga, siempre que esté dentro del rango y sea una dirección IP libre, ya que está conectado con al dom0 a través de un cable cruzado virtual.

Una comprobación de lo anterior es la siguiente:

```
brun:~# route -n
Kernel IP routing table
Destination      Gateway          Genmask          Flags   Iface
10.0.0.0         0.0.0.0          255.255.255.0   U       eth0
0.0.0.0         10.0.0.254       0.0.0.0         UG      eth0
brun:~# ping -c 3 192.168.0.1
PING 192.168.0.1 (192.168.0.1) 56(84) bytes of data.
64 bytes from 192.168.0.1: icmp_seq=1 ttl=63 time=511 ms
64 bytes from 192.168.0.1: icmp_seq=2 ttl=63 time=1.54 ms
64 bytes from 192.168.0.1: icmp_seq=3 ttl=63 time=0.288 ms
```

donde `vif1.0` tiene dirección IP 10.0.0.227 en lugar de 10.0.0.254 y sin embargo domU tiene salida al exterior a través del NAT del dom0.

### 3. VMware Server 1.0

Se pueden establecer tres tipos de configuraciones de red básicas:

- Red con un puente (*bridged networking*)
- NAT (*Network Address Translation*)
- “Host only”

y a partir de ellas configuraciones mucho más complejas. Cuando se configura VMware Server en el servidor a través del script `vmware-config.pl`, se va preguntando si queremos establecer o no los distintos tipos de configuraciones y asignándoles un dispositivo. Por defecto estos dispositivos se denominan:



- /dev/vmnet0 Puente
- /dev/vmnet1 Host only
- /dev/vmnet8 NAT

El esquema típico que podríamos tener en un anfitrión en el que hubiésemos habilitado los tres tipos de redes básicas con sus dispositivos por defecto sería:

```
ayla:~# ifconfig
eth0  Link encap:Ethernet  HWaddr 00:40:D0:65:B2:84
      inet addr:192.168.0.68 Bcast:192.168.0.255 Mask:255.255.255.0
      inet6 addr: fe80::240:d0ff:fe65:b284/64 Scope:Link
      UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
      RX packets:3063 errors:0 dropped:0 overruns:0 frame:0
      TX packets:3098 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:1000
      RX bytes:2107273 (2.0 MiB)  TX bytes:751399 (733.7 KiB)
      Interrupt:3 Base address:0xc000

lo    Link encap:Local Loopback
      inet addr:127.0.0.1  Mask:255.0.0.0
      inet6 addr: ::1/128 Scope:Host
      UP LOOPBACK RUNNING  MTU:16436  Metric:1
      RX packets:10 errors:0 dropped:0 overruns:0 frame:0
      TX packets:10 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:0
      RX bytes:660 (660.0 b)  TX bytes:660 (660.0 b)

vmnet1 Link encap:Ethernet  HWaddr 00:50:56:C0:00:01
      inet addr:192.168.127.1 Bcast:192.168.127.255 Mask:255.255.255.0
      inet6 addr: fe80::250:56ff:fec0:1/64 Scope:Link
      UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
      RX packets:0 errors:0 dropped:0 overruns:0 frame:0
      TX packets:6 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:1000
      RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)

vmnet8 Link encap:Ethernet  HWaddr 00:50:56:C0:00:08
      inet addr:172.16.62.1 Bcast:172.16.62.255 Mask:255.255.255.0
      inet6 addr: fe80::250:56ff:fec0:8/64 Scope:Link
      UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
      RX packets:0 errors:0 dropped:0 overruns:0 frame:0
      TX packets:6 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:1000
      RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
```

Donde el único dispositivo que no aparece es el puente asociado al dispositivo /dev/vmnet0

Las direcciones MAC de los dispositivos virtuales comienzan con 00:50:56, que corresponde con una interfaz virtual de VMware.

Cuando se inicia el demonio vmware, se obtiene la siguiente salida en el fichero /var/log/syslog (las líneas más significativas):

```
kernel: /dev/vmnet: open called by PID 5428 (vmnet-bridge)
kernel: /dev/vmnet: hub 0 does not exist, allocating memory.
kernel: /dev/vmnet: port on hub 0 successfully opened
kernel: bridge-eth0: enabling the bridge
kernel: bridge-eth0: up
kernel: bridge-eth0: already up
kernel: bridge-eth0: attached
```

Se levanta el puente sobre la interfaz física eth0.

```
kernel: /dev/vmnet: open called by PID 5442 (vmnet-natd)
kernel: /dev/vmnet: hub 8 does not exist, allocating memory.
kernel: /dev/vmnet: port on hub 8 successfully opened
kernel: /dev/vmnet: open called by PID 5527 (vmware-vmx)
kernel: eth0: Promiscuous mode enabled.
kernel: device eth0 entered promiscuous mode
kernel: bridge-eth0: enabled promiscuous mode
kernel: /dev/vmnet: port on hub 0 successfully opened
kernel: /dev/vmnet: open called by PID 5547 (vmnet-netifup)
kernel: /dev/vmnet: hub 1 does not exist, allocating memory.
kernel: /dev/vmnet: port on hub 1 successfully opened
kernel: /dev/vmnet: port on hub 8 successfully opened
```

Se levantan la interfaces virtuales vmnet1 y vmnet8

```
vmnet-dhcpd: Internet Software Consortium DHCP Server 2.0
vmnet-dhcpd: Configured subnet: 172.16.62.0
vmnet-dhcpd: Setting vmnet-dhcp IP address: 172.16.62.254
kernel: /dev/vmnet: open called by PID 5585 (vmnet-dhcpd)
kernel: /dev/vmnet: port on hub 8 successfully opened
vmnet-dhcpd: Recving on      VNet/vmnet8/172.16.62.0
vmnet-dhcpd: Sending on     VNet/vmnet8/172.16.62.0
```

Se levanta el servidor DHCP en la interfaz vmnet8

```
vmnet-dhcpd: Configured subnet: 192.168.127.0
vmnet-dhcpd: Setting vmnet-dhcp IP address: 192.168.127.254
kernel: /dev/vmnet: open called by PID 5586 (vmnet-dhcpd)
kernel: /dev/vmnet: port on hub 1 successfully opened
vmnet-dhcpd: Recving on      VNet/vmnet1/192.168.127.0
vmnet-dhcpd: Sending on     VNet/vmnet1/192.168.127.0
```

Se levanta el servidor DHCP en la interfaz vmnet1.

### 3.1. Puente

Un puente (*bridge*) es un dispositivo que trabaja en el nivel de enlace de la pila TCP/IP. Un puente se utiliza en VMware Server para unir máquinas virtuales que estén en el mismo segmento de red.

En la figura 3 aparece un esquema simple de un puente sobre el dispositivo `vmnet0` utilizado por VMware Server.

El caso típico es cuando tenemos una interfaz de red física en el sistema anfitrión dentro de una RAL y queremos que los sistemas huésped pertenezcan a la

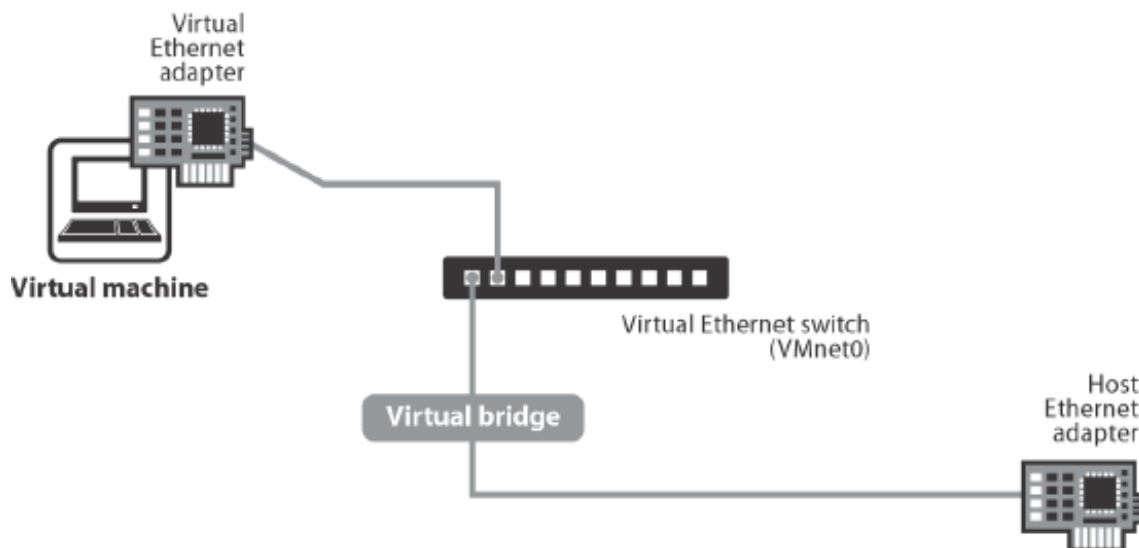


Figura 3: A través del puente se conecta una máquina virtual a la misma red que el anfitrión, utilizando su interfaz de red

misma red. Por ejemplo, en la salida de `ifconfig` anterior, el dispositivo `eth0` del sistema anfitrión tenía la dirección IP `192.168.0.68` y la salida de `ifconfig eth0` de un sistema huésped que accede a través del puente `/dev/vmnet0` es:

```
eth0  Link encap:Ethernet  HWaddr 00:0C:29:73:A4:09
      inet addr:192.168.0.64  Bcast:192.168.0.255  Mask:255.255.255.0
      inet6 addr: fe80::20c:29ff:fe73:a409/64  Scope:Link
      UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
      RX packets:367 errors:0 dropped:0 overruns:0 frame:0
      TX packets:282 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:1000
      RX bytes:41072 (40.1 KiB)  TX bytes:37972 (37.0 KiB)
      Interrupt:169 Base address:0x1400
```

Es decir, los dos equipos están dentro de la misma subred.

La dirección MAC de la interfaz virtual de un sistema huésped comienza por `00:0C:29`, que se corresponde con la empresa AMD.

### 3.2. "Host-only"

Este modo se utiliza cuando queremos establecer un segmento de red que conecte al sistema anfitrión con el sistema huésped a través de una red privada. De forma automática se instala un servidor DHCP que escucha peticiones de direcciones IP a través de `vmnet1`.

En principio el sistema huésped no estará conectado a otra red, salvo que el sistema anfitrión se modifique adecuadamente, a través de un proxy por ejemplo.

Un esquema de esta configuración de red puede verse en la figura 4

Si modificamos la configuración de red del mismo sistema huésped del apartado anterior a "host-only". Obtendríamos la salida siguiente en el fichero de registros `/var/log/syslog` del sistema anfitrión:

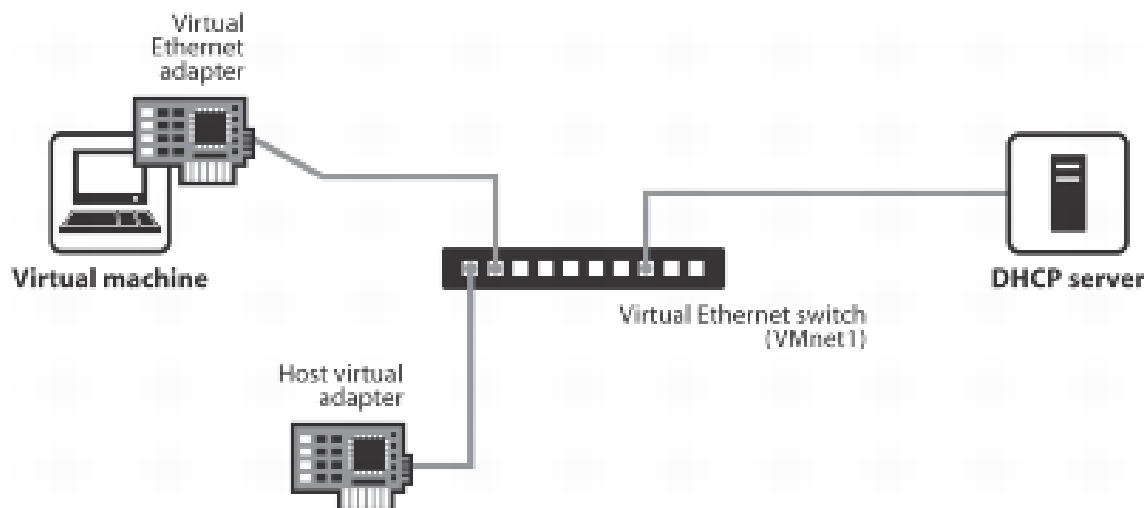


Figura 4: Host-only networking crea una red que está completamente contenida dentro del equipo anfitrión

```

vmnet-dhcpd: DHCPDISCOVER from 00:0c:29:73:a4:09 via vmnet1
vmnet-dhcpd: DHCPOFFER on 192.168.127.128 to 00:0c:29:73:a4:09 via vmnet1
vmnet-dhcpd: DHCPREQUEST for 192.168.127.128 from 00:0c:29:73:a4:09 via vmnet1
vmnet-dhcpd: DHCPACK on 192.168.127.128 to 00:0c:29:73:a4:09 via vmnet1

```

de lo que podemos deducir que el equipo con dirección MAC `00:0c:29:73:a4:09` ha solicitado una dirección IP al sistema anfitrión y ha obtenido la `192.168.127.128`.

La tabla de encaminamiento del sistema huésped queda:

```

alberto@creb:~$ /sbin/route -n
Kernel IP routing table
Destination      Gateway          Genmask         Iface
192.168.127.0    0.0.0.0         255.255.255.0  eth0

```

Es decir, no tiene pasarela para conectarse con otras redes, sólo con el sistema anfitrión (IP `192.168.0.1`) o con otros sistemas huésped del mismo segmento.

### 3.3. NAT

Este modo se utiliza cuando queremos conectar un sistema huésped a una red, compartiendo la dirección del sistema anfitrión. En este caso también se configura de forma automática un servidor DHCP.

El caso más típico de utilización de NAT es cuando la dirección IP de la interfaz física del sistema anfitrión es la única dirección pública de acceso a Internet y debe compartirse con todos los sistemas huésped, para lo que deberá existir un mecanismo de traducción de las direcciones privadas a la pública.

Un esquema de esta configuración de red puede verse en la figura 5.

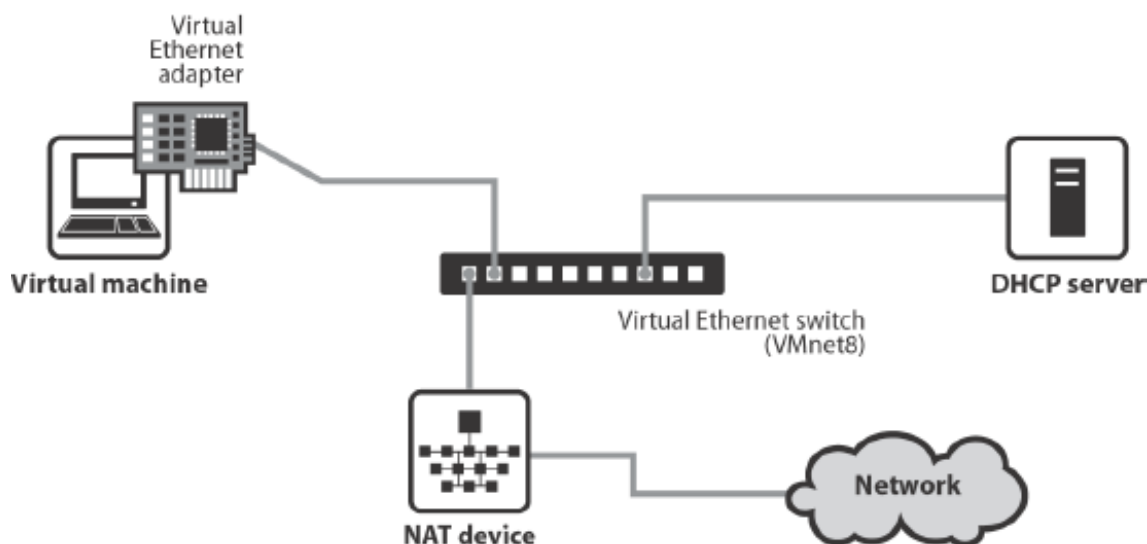


Figura 5: Mediante NAT se modifica la dirección IP del huésped por la dirección IP “externa” del anfitrión.

Si modificamos la configuración de red del mismo sistema huésped del apartado anterior a NAT, obtendríamos la salida siguiente en el fichero de registros `/var/log/syslog` del sistema anfitrión:

```
vmnet-dhcpd: DHCPDISCOVER from 00:0c:29:73:a4:09 via vmnet8
vmnet-dhcpd: DHCPOFFER on 172.16.62.128 to 00:0c:29:73:a4:09 via vmnet8
vmnet-dhcpd: DHCPREQUEST for 172.16.62.128 from 00:0c:29:73:a4:09 via vmnet8
vmnet-dhcpd: DHCPACK on 172.16.62.128 to 00:0c:29:73:a4:09 via vmnet8
```

de lo que podemos deducir que el equipo con dirección MAC `00:0c:29:73:a4:09` ha solicitado una dirección IP al sistema anfitrión y ha obtenido la `172.16.62.128`.

La tabla de encaminamiento del sistema huésped queda:

```
alberto@creb:~$ /sbin/route -n
Kernel IP routing table
Destination      Gateway          Genmask         Iface
172.16.62.0     0.0.0.0         255.255.255.0  eth0
0.0.0.0         172.16.62.2    0.0.0.0        eth0
```

Es decir, tiene pasarela para conectarse con otras redes (`172.16.62.2`), sólo con el sistema anfitrión (IP `192.168.0.1`) o con otros sistemas huésped del mismo segmento.

## Referencias

- [1] Redes en Xen 3.0  
<http://wiki.xensource.com/xenwiki/XenNetworking>

- [2] Lista de correo de Xen. Mensaje 30 de Febrero de 2006  
<http://lists.xensource.com/archives/html/xen-users/2006-02/msg00030.html>
- [3] Lista de correo de Xen. Mensaje 949 de Marzo de 2006  
<http://lists.xensource.com/archives/html/xen-users/2006-03/msg00949.html>
- [4] Manual de Máquinas virtuales de VMware Server  
[http://www.vmware.com/pdf/server\\_vm\\_manual.pdf](http://www.vmware.com/pdf/server_vm_manual.pdf)