Manual de usuario Sun xVM VirtualBox (Versión 2.1.0)*

Indice

* La presente no es una traducción completa del manual, se enfoca sólo a las cuestiones relacionadas a redes. Tampoco está realizada de manera exhaustiva (no soy, como verán, un traductor profesional, pero creo que alcanza para ayudar a entender un poco mejor el funcionamiento de esta estupenda aplicación que es Virtual Box). Los textos subrayados (excepto los hiperenlaces) indican que no estuve seguro acerca de cuál era la traducción exacta. Espero que les sea de utilidad. Ángel <mykantele.angel@gmail.com>

[*Lo siguiente es de la versión 1.6.3, descartados en la actual versión 2.1.0. N. del Trad.*]

(...)

3.7.5 Configuraciones de red

La sección "Red" en una ventana de configuraciones de una máquina virtual (VM) te permite ajustar la forma en que Virtual Box (VB) presenta las tarjetas de red virtuales a tu VM, y cómo operarlas.

Cuando creas una VM, VB por defecto habilita una de esas cuatro tarjetas y selecciona para esto el modo NAT. De esta forma el huésped (guest) puede conectarse al mundo exterior usando la red del anfitrión (host) y el mundo exterior puede conectarse a los servicios en el huésped que quieras que sean visibles al exterior de la VM.

> Nota: si estás instalando Windows Vista en una VM, probablemente no tengas red al principio. Mira el capítulo 4.2.4, *Redes en Windows Vista*, para instrucciones sobre cómo solucionar este problema.

En la mayoría de los casos, el ajuste "NAT" funcionará bien para vos. Sin embargo, VB es extremadamente flexible en la forma en que puede virtualizar redes. Soporta hasta ocho tarjetas de red virtuales por VM, de los cuales los cuatro primeros pueden ser configurados detalladamente en la GUI.[1] Todas las tarjetas (8) pueden ser configuradas en la línea de comandos con *VBoxManage.* A causa de esto, hemos dedicado un capítulo entero de este manual para discutir la configuración de redes; por favor mira el capítulo 6, *Redes virtuales*.

(...)

4.2.4 Redes en Windows Vista

Windows Vista ya no trae un driver para tarjetas AMD PCnet Ethernet pero VB lo proporciona por defecto al huésped. En consecuencia, después de instalar Vista en una VM, inicialmente no habrá red. Por comodidad, VB trae un driver para la tarjeta AMD PCnet, que viene con el "Windows Guest Additions" (Aplicaciones del huésped). Si lo instalas en un huésped Vista, el driver también sera automáticamente instalado.

Si por alguna razón te gustaría instalar el driver manualmente, lo puedes encontrar en Guest Additions ISO (VboxGuestAdditions.iso). Para instalarlo, monta la imagen (como se describió arriba, selecciona "Instalar Guest Additions" desde el menú "Dispositivos"). Luego, inicia el Asistente para agregar hardware de Windows y direcciónalo al CD de Guest Additions donde puede ser encontrado un driver para la tarjeta PCnet en la carpeta *AMD_PCnet.*[2]

---

1 *Nota del traductor:* Graphical User Interface - interfaz gráfica de usuario.

2 *N. de Trad.:*Podría pasar que Windows no te permita instalar el driver. En ese caso cuando elijas, en la solapa "Red" de la ventana de configuración de la VM, el modo "Interfaz anfitrión" debajo en la ventana te aparecerá la advertencia "¡Configuración inválida detectada!" y arriba de este mensaje te informa que "en la página se ha detectado una Interface de Red Anfitrión incorrecta". Además, en la sección "Interfaces Anfitrión (H)" cuando quieras crear una nueva haciendo click en "Agregar" te puede aparecer el mensaje de que el software que se está instalando "no ha superado la prueba del logotipo de Windows..." y te rechazará la instalación del driver. Lo mismo puede pasar con el driver para

Alternativamente, cambia los ajustes de la VM del huésped vista para usar una tarjeta de red Intel en vez de la AMD PCnet; mira el capítulo 3.7.5 *Configuraciones de red*.

(...)

6 Redes virtuales

Como se mencionó brevemente en el capítulo 3.7.5, configuraciones de red, VB proporciona hasta cuatro tarjetas virtuales PCI Ethernet por cada VM. Por cada tarjeta puedes seleccionar individualmente:

1. el hardware que será mejor virtualizado;
2. el modo de virtualización que la tarjeta virtual manejará con respecto a tu hardware físico de red en el anfitrión.

Cuatro de las tarjetas de red pueden ser configuradas en la sección "Red" del diálogo de configuraciones de la GUI de VB. Puedes configurar las ocho tarjetas de red en la línea de comandos vía *VBoxManage modifyvm*. Mira el capítulo 8.5.

6.1 Hardware de red virtual

Para cada tarjeta de red, puedes seleccionar individualmente qué clase de hardware será presentado a la VM. VB puede virtualizar los siguientes cuatro tipos de hardware de red:

- ✓ AMD PCNet PCI II;
- ✓ AMD PCNet FAST III (valor por defecto);
- ✓ Intel PRO/1000 MT Desktop (PC de escritorio);
- ✓ Intel PRO/1000 T Server (servidor).

PCNet FAST III es el valor por defecto por que es soportado por casi todos los sistemas operativos (SO) fuera de la caja, además del administrador de arranque GNU GRUB. No obstante, con VB 1.6 fue añadido soporte para el tipo Intel PRO/1000 MT porque Microsoft dejó el soporte para las tarjetas AMD PCNet con Windows Vista, y por lo tanto los huéspedes Vista no tienen red sino es con la instalación manual del driver. Ver capítulo 4.2.4, *redes en Windows Vista*, página 55 por detalles. La variante servidor (*server*) de la tarjeta Intel PRO/1000 fue añadida con VB 1.6.2 por que es reconocida por huéspedes Windows XP sin la instalación de drivers adicionales.

VB ha limitado el soporte para los llamados datagramas gigantes – jumbo frames -, es decir, paquetes con más de 1500 bytes de datos, con la condición de que utilices la tarjeta de virtualización Intel y la *interfaz anfitrión.* En otras palabras, los jumbo datagramas no son soportados en modo NAT o en dispositivos de red AMD; en esos casos, serán bajados de la transmisión y de la dirección de recepción. Los SO huéspedes tratando de usar esta

---

soporte de dispositivos USB. Esto se soluciona yendo a "Panel de control", "Sistema", solapa "Hardware", sección "Controladores" y click en "Firma de controladores". Allí te da la opción por tres acciones cuando intentas instalar un hardware y su controlador: 1. "Ninguna: instalar el software sin pedir mi aprobación", 2. "Advertir: preguntarme siempre que tenga que elegir una acción" y 3. "Bloquear: no instalar controladores de software sin firma" (ajustada como predeterminada). Yo elegí la segunda.

característica observarán que es un paquete perdido, que puede llevar a comportamientos inesperados en la aplicación en el huésped. Esto no causa problemas con SO huéspedes en su configuración por defecto, ya que los jumbo datagramas deben ser explícitamente habilitados.

6.2 Introducción a los modos de conexión

Cada uno de los adaptadores de red pueden ser configurados separadamente para operar en uno de los siguientes cuatro modos:

- ✓ no conectado
- ✓ traducción de dirección de red (NAT)
- ✓ interfaz anfitrión
- ✓ red interna

Por defecto, las tarjetas de red virtuales son puestas para usar NAT, las cuales son bastante aptas para las necesidades de una conexión estándar (accediendo internet desde programas que corren en el huésped y proporcionando servicios de red para máquinas en una intranet local). En concreto, si todo lo que quieres es navegar por la web, descargar archivos y mirar el correo dentro del huésped, entonces la configuración por defecto de la conexión NAT debería ser suficiente para vos y pasar por alto con seguridad el resto de esta sección. Por favor observá que la utilidad *ping* no funciona en NAT y que hay ciertas limitaciones cuando se use la función *compartir archivos* de Windows.

Para necesidades de conexiones avanzadas tales como simulación de red, la *interfaz anfitrión* puede ser usada para poner un software adicional basado en interfaz de red en el anfitrión al cual es conectada la VM. Por último, el modo *red interna* puede ser usado para crear una red virtual que es visible para las VM seleccionadas pero no para las aplicaciones funcionando en el anfitrión o para el mundo exterior. Las siguientes secciones describen los modos de red disponibles en mayor detalle.

6.3 Modo "no conectado"

Cuando el modo de una tarjeta de red virtual es puesta a "no conectado", VB informa al huésped que una tarjeta de red está presente pero que no hay conexión (como si ningún cable Ethernet fuese enchufado a la tarjeta). De esta forma es posible "tirar" del cable Ethernet virtual e interrumpir la conexión, lo cual puede ser útil para informar al SO huésped que ninguna conexión de red está disponible y forzar una reconfiguración.

6.4 Traducción de dirección de red (NAT)

Ésta es la forma más simple de acceder a una red externa desde una VM. Por lo general, ésta no requiere ninguna configuración en la red anfitrión ni en el sistema huésped. Por eso es el modo de conexión por defecto en VB.

Una VM con NAT habilitada funciona más como una computadora real que se conecta a internet a través de un router. El "router", en este caso, es el motor de red de VB, el cual muestra ("mapea") el tráfico desde y hacia la VM transparentemente. La desventaja del modo NAT es que, más que como una red privada detrás de un router, la VM es invisible e inaccesible desde internet;

no puedes correr un servidor de esta forma a menos que configures *redireccionamiento de puertos* -port forwarding- (descripto debajo).

La VM recibe sus direcciones de red y configuración en la red privada desde un servidor DHCP que esta integrado en VB. La dirección IP que la VM recibe está por lo general en una red completamente diferente a la del anfitrión.

Como más de una tarjeta de una VM puede ser puesta para usar NAT, la primer tarjeta es conectada a la red privada 10.0.2.0, la segunda a la red 10.0.3.0 y así sucesivamente. Si necesitas cambiar por alguna razón el rango de IP asignado al huésped, consulta entonces el capítulo 9.11, *configurando las direcciones de una interfaz de red NAT,* pg. 123.

Los datagramas -frames- de red enviados por el SO huésped son recibidos por el motor NAT de VB, que obtiene los datos TCP/IP y los reenvía usando el SO anfitrión. Para un programa en el anfitrión, o para otra computadora en la misma red del anfitrión, pareciera que los datos fueron enviados por VB en el anfitrión, usando una dirección IP perteneciente al anfitrión. VB está a la escucha de respuestas a los paquetes enviados, y los reempaqueta y reenvía a la máquina huésped en su red privada.

6.4.1 Configurando redireccionamiento de puertos con NAT

Como la VM está conectada a una red privada interna a VB e invisible al anfitrión, los servicios de red en el huésped no son accesibles para la máquina anfitrión o para otras computadoras en la misma red. Sin embargo, VB puede proporcionar servicios disponibles fuera del huésped mediante el uso de *redireccionamiento de puertos*. Esto significa que VB escucha ciertos puertos en el anfitrión y reenvía todos los paquetes que les llegue hacia el huésped a través de los puertos usados por los servicios que están siendo redireccionados. Para una aplicación en el anfitrión u otra máquina física (o virtual), pareciera como si el servicio que está siendo intermediado (proxied) estuviera realmente funcionando sobre el anfitrión (nota que esto también significa que no puedes correr el mismo servicio en el mismo puerto en el anfitrión). Sin embargo, aún puedes obtener la ventaja de correr el servicio en una VM (por ejemplo, los servicios en la máquina anfitrión o en otra máquina virtual no pueden ser comprometidos o bloqueados por una vulnerabilidad o un bug en el servicio, y el servicio puede correr en un SO distinto al del sistema anfitrión).

Puedes poner el servicio huésped que desees intermediar (to proxy) usando la herramienta de línea de comandos *VBoxManage*. Necesitarás conocer en el huésped cuáles puertos usa el servicio y decidir cuáles puertos usar en el anfitrión (a menudo querrás, pero no siempre podrás, usar los mismos puertos en el huésped y en el anfitrión). Puedes usar cualquiera de los puertos en el anfitrión que no estén ya siendo usados por un servicio. Un ejemplo de cómo configurar conexiones entrantes NAT a un servidor *ssh* en el huésped precisa de los siguientes tres comandos:

```
VBoxManage setextradata "Linux Guest"
            "VBoxInternal/Devices/pcnet/0/LUN#0/Config/guestssh/Protocol"
TCP
VBoxManage setextradata "Linux Guest"
```

"VBoxInternal/Devices/pcnet/0/LUN#0/Config/guestssh/GuestPort" 22
VBoxManage setextradata "Linux Guest" "VBoxInternal/Devices/pcnet/0/LUN#0/Config/guestssh/HostPort" 2222

Este ejemplo supone una tarjeta de red virtual PCNet; si has configurado el huésped para usar la Intel PRO/1000, remplaza "pcnet" con "e1000" en el comando de arriba. Igualmente, si quieres configurar una instancia de interfaz distinta remplaza el /0/ con el índice apropiado. pcnet y e1000 son contados separadamente en este sentido, y el conteo comienza en 0 para ambos tipos.

El nombre *guestssh* es arbitrario y elegido para esta configuración de redirección (forwarding) en concreto. Con esa configuración en su lugar, todas las conexiones TCP al puerto 2222 en el anfitrión serán redireccionados (forwarded) al puerto 22 en el huésped. *Protocol* puede ser TCP o UDP (no son *case sensitive*, es decir, se pueden escribir con mayúsculas o con minúsculas). Para quitar el "mapeo" (mapping), usa los mismos comandos pero excluyendo los valores (en este caso TCP, 22 y 2222).

No es posible configurar conexiones entrantes NAT mientras la VM está funcionando. No obstante, puedes cambiar los ajustes a una VM que actualmente está guardada (o apagada con una instantánea -snapshot).

6.4.2 Arranque PXE con NAT

El arranque PXE ahora está soportado en el modo NAT. El servidor dhcp de NAT proporciona un archivo de arranque con nombre de tipo *vmname.pxe* si el directorio *TFTP* está en el mismo directorio en el que está guardado el archivo VirtualBox.xml del usuario. Es responsabilidad del usuario proporcionar el archivo vmname.pxe.

6.4.3 Limitaciones de NAT

Hay cuatro limitaciones del modo NAT que los usuarios deberían conocer:

El protocolo ICMP es muy limitado: algunas herramientas de depuración frecuentemente usados en redes (v.g. *Ping* o *tracerouting*) dependen del protocolo ICMP para el envío/recepción de mensajes. Aunque el soporte ICMP ha sido mejorado con VB 2.1 (*ping* ahora debería funcionar), algunas otras herramientas podrían funcionar con fallos.

La recepción de transmisiones UDP no es confiable: el huésped no hace fiable la recepción de transmisiones desde que, a fin de salvar recursos, sólo escucha por una cierta cantidad de tiempo después de enviar datos UDP a un puerto en particular. Como consecuencia, la resolución de nombres basada en NetBios no siempre funciona (pero WINS siempre lo hace). Para esto, puedes usar el número IP del servidor deseado en la dirección \\server\share.

Protocolos tales como GRE no son soportados: los protocolos que no sean TCP y UDP no son soportados. Esto significa que algunos productos VPN -Red Privada Virtual- (v.g. PPTP de Microsoft) no pueden ser usados. Hay otros productos VPN que usan simplemente TCP y UDP.

Imposibles los envíos a hosts con puertos menores a 1024: en hosts basados en Unix (v.g. Linux, Solaris, MacOS X) no es posible unir puertos por debajo de 1024 desde aplicaciones que no corren como *root*. Por lo tanto si

intentas configurarlo como puerto de redirección, entonces la VM se resistirá arrancar.

Éstas limitaciones normalmente no afectan el uso de redes estándar. Pero la presencia de NAT también tiene efectos sutiles que pueden interferir con protocolos que trabajan normalmente. Un ejemplo es NFS, donde el servidor frecuentemente está configurado para rechazar conexiones desde puertos no privilegiados (v.g. Puertos iguales o mayores a 1024).

6.5 Interfaz anfitrión

Con la interfaz anfitrión, VB usa un controlador de dispositivo en tu sistema *anfitrión* que filtra los datos de tu adaptador de red físico. Por lo tanto, este driver es denominado filtro "net". Esto permite a VB interceptar datos desde la red física e inyectarlos en él, creando efectivamente una nueva interfaz de red por software. Cuando un huésped usa dicha interfaz, al sistema anfitrión le parecería como si el huésped estuviera físicamente conectado a la interfaz usando un cable de red: el anfitrión puede enviar datos al huésped a través de la interfaz y recibir datos de él. Esto significa que puedes rutear (routing) o puentear (bridging) entre el huésped y el resto de tu red.

Para que esto funcione, VB necesita un driver en tu sistema anfitrión. La forma en que trabaja interfaz anfitrión ha sido completamente reescrita con VB 2.0 y 2.1, dependiendo del SO del anfitrión. Desde la perspectiva del usuario, la principal diferencia es que la configuración complicada ya no es necesaria en ninguno los SO anfitriones soportados.[3]

Nota: Incluso aunque TAP ya no es necesario con la nueva interfaz anfitrión de VB 2.1, todavía *puedes* usar interfaces TAP para ciertas configuraciones avanzadas, ya que puedes conectar una VM a cualquier interfaz anfitrión – que también podría ser una interfaz TAP.

Con el nuevo mecanismo, para habilitar Interfaz Anfitrión, todo lo que necesitas hacer es abrir el diálogo Configuración de una VM, ir a la página "Red" y seleccionar "Interfaz Anfitrión" en la lista desplegable del campo "Conectar a". finalmente, elige la interfaz anfitrión deseada desde la lista al pie de la página, que contiene las interfaces de red físicas de tu sistema. En una típica MacBook, por ejemplo, esto te permitirá elegir entre "en1: AirPort" (que es la interfaz inalámbrica) y "en0: Ethernet", que representa la interfaz con un cable de red.

Dependiendo de tu SO anfitrión, las siguientes limitaciones deberían tenerse en cuenta:

- ✓ En anfitriones Macintosh, la funcionalidad es limitada cuando usas AirPort (la red inalámbrica de MAC) para interfaz anfitrión. Actualmente, VB soporta sólo IPv4 en AirPort. Para otros protocolos tales como Ipv6 e IPX, debes elegir

3 Para anfitriones MAC OS X y Solaris, los drivers filtros net fueron agregados ya en VB 2.0 (como soporte inicial para interfaz anfitrión en estas plataformas). Con VB 2.1, los drivers filtros también fueron añadidos a anfitriones Windows y Linux, remplazando el mecanismo presente previamente para esas plataformas; especialmente en Linux, el método anterior requerido para crear interfaces TAP y puentes, que fue complejo y variado desde una distribución a la siguiente. Ya nada de esto es cecesario.

una interfaz cableada.

✓ En anfitriones Linux, la funcionalidad es limitada cuando usas interfaces inalámbricas para interfaz anfitrión. Actualmente, VB soporta sólo IPv4 en inalámbrico. Para otros protocolos tales como Ipv6 e IPX, debes elegir una interfaz cableada.

Además, poner MTU a menos de 1500 bytes en interfaces cableadas proporcionados por el driver sky2 en la Marvell Yukon II EC Ultra Ethernet NIC es conocida por ocasionar paquetes perdidos bajo ciertas condiciones.

✓ En anfitriones Solaris, no hay soporte para el uso de interfaces inalámbricas. El filtrado de tráfico de huésped usando IPFilter tampoco es completamente soportado debido a restricciones técnicas del subsistema de red de Solaris. Estas cuestiones deberían abordadas en una futura publicación de OpenSolaris.

Con VB 2.0.4 y superior, es posible usar Interfaces de Red Virtual de Ballesta -Crossbow Virtual Network Interfaces (VNICs)- con interfaz anfitrión, pero con las siguientes advertencias:

- Una VNIC no puede ser compartida entre múltiples interfaces de red huéspeds, es decir, cada interfaz de red huésped debe tener su propia, exclusica VNIC.

- La VNIC y la interfaz de red huésped que usa la VNIC deben tener asignadas idénticas direcciones MAC.

6.6 Red interna

La red interna es similar a la interfaz anfitrión en el que la VM puede comunicarse directamente con el mundo exterior. Sin embargo, el "mundo exterior" está limitado a otras VM que se conectan a la misma red interna.

Aunque técnicamente todo lo que puede ser hecho usando red interna también puede ser hecho usando interfaz anfitrión, hay dos buenas razones por las que este otro modo fue implementado:

1. **Seguridad.** En el modo interfaz anfitrión todo el tráfico va a través de una interfaz del sistema anfitrión. Por lo tanto es posible conectar un sniffer (como Ethereal) al interfaz anfitrión y registrar todo el tráfico que va sobre determinada interfaz. Si por alguna razón prefieres dos mas VMs en la misma máquina para comunicarte privadamente, escondiendo sus datos al anfitrión y al usuario, entonces la interfaz anfitrión no es una opción.

2. **Velocidad.** La red interna es más eficiente que la interfaz anfitrión, ya que VB puede transmitir directamente los datos sin tener que enviarlos a través de la red del SO anfitrión.

La red interna es creada automáticamente cuando es necesitada, esto es, no hay configuración central. Cada red interna es identificada simplemente por su nombre. Una vez que hay más de una tarjeta de red virtual activa con la misma ID de red interna, el driver de soporte de VB "conecta" las tarjetas y actúa como un switch. Éste driver implementa un swith Ethernet completo y soporta datagramas -frames- broadcast/multicast y modo promiscuo.

Para conectar una tarjeta de red de una VM, coloca su modo de red a "Red

interna". Hay dos formas de realizar esto:

✓ Puedes usar el diálogo "Configuración" en la GUI de VB. En la categoría "Red" selecciona "Red interna" desde la lista desplegable de modos de conexión ["Conectar a"]. Ahora selecciona el nombre de una red interna existente de la lista de abajo ["Nombre de la interfaz"] o ingresa un nuevo nombre en el campo de entrada.

✓ Puedes usar VBoxManage modifyvm <VM name> -nic<x> intnet. Opcionalmente, puedes especificar el nombre de una red con el comando VBoxManage modifyvm <VM name> intnet<x> <network name>. Si no especificas un nombre, la tarjeta de red será conectada a la red *intnet* por defecto. Ver capítulo 8.5, *VBoxManage modifyvm.*

En cualquier caso, tendrás que configurar las tarjetas de red (virtual) en el SO huésped que está participando de la red interna para usar direcciones IP estáticas (por que las redes internas provistas por VB no soportan DHCP, como lo haría el motor NAT de VB). Estas direcciones IP deberían usar direcciones IP en la misma subred (v.g. 192.168.2.1 y 192.168.2.2). Es posible que tengas que desactivar los firewalls de los huéspedes para permitir que se comuniquen entre sí.

Como medida de seguridad, la implementación de una red interna Linux sólo permite correr VMs bajo la misma ID de usuario para establecer una red interna.

7 Interfaces -front-ends- alternativas; VMs remotas

7.1 Introducción

Como se mencionó brevemente en el capítulo 1.2, VB tiene un diseño interno muy flexible que te permite utilizar diferentes interfaces para controlar las mismas VMs. Para ilustrarte, puedes, por ejemplo, arrancar una VM con la fácil de usar GUI de VB y entonces detenerla desde la línea de comandos. Con el soporte de VB para el Protocolo de Escritorio Remoto (VRDP), incluso puedes correr remotamente VMs en un servidor acéfalo/ciego (?) -headless- y tener toda la salida gráfica redireccionada sobre la red.

En detalle, las siguientes interfaces están incluidas en el paquete estándar VB:

1. *VirtualBox* es nuestra GUI, a cuya descripción esta dedicada la mayoría de este manual de usuario, especialmente en el capítulo 3, *Arrancando con VB*. Aunque es la más fácil de usar de nuestras interfaces, (todavía) no cubre las características que VB proporciona. De todas formas, es la mejor forma de tener que conocer inicialmente VB.

2. *VBxManage* es nuestra interfaz de línea de comandos y es descripta en la siguiente sección.

3. *VBoxSDL* es una alternativa, interfaz gráfica simple con una serie de rasgos intencionalmente limpia, diseñada para mostrar sólo VMs que son controladas en detalle con *VBoxManage*. Esto es interesante para ambientes de negocio donde mostrar todos los accesorios de la GUI completa no es factible. *VBoxSDL* es descripto en el capítulo 7.3.

4. Finalmente, *VboxHeadless* es además otra interfaz que no produce salida visible en el anfitrión en absoluto, pero solamente actúa como un servidor VRDP. Ahora, aunque las otras interfaces gráficas (VirtualBox y VBoxSDL) también tienen soporte VRDP incorporado y pueden funcionar como un servidor VRDP, esta interfaz particular no requiere soporte gráfico. Es útil, por ejemplo, si quieres hospedar tus VMs en un servidor Linux acéfalo que no tiene instalado X Window. Para detalles mira el capítulo 7.4.1.

Si la interfaz de arriba no satisface tus necesidades particulares, es relativamente fácil crear todavía otra interfaz para el motor complejo de virtualización que es el núcleo de VB, ya que el núcleo de VB expone claramente todas sus características en una limpia API COM/XPCOM. Esta API es descripta en el capítulo 10.

(...)

7.4 VMs remotas (soporte VRDP)

La GUI de VB tiene incorporada un servidor para VRDP. Esto te permite mirar la salida de una ventana de una VM remotamente en cualquier otra computadora y controlarla desde aquí, como si estuviera funcionando en la máquina remota.

VRDP es una extensión compatible hacia atrás para el Protocolo de Escritorio Remoto de Windows (RDP). Generalmente las actualizaciones gráficas y el audio son enviados desde la máquina remota al cliente, mientras las acciones del mouse y el teclado son devueltas.

Puedes usar cualquier visor RDP estándar, como el que viene con Windows (generalmente encontrado en "Accesorios" => "Comunicaciones" => "Conexión a escritorio remoto"), o en Linux el programa estándar open-source *rdesktop*, para conectar a la VM remotamente.

Deberías usar la dirección IP del sistema anfitrión como el servidor de dirección. El servidor VRDP usa por defecto el puerto estándar RDP TCP 3389. El puerto puede ser cambiado en las configuraciones de la GUI de la VM o con la opción VBoxManage modifyvm command -vrdpport. Observa que sólo una máquina puede usar un puerto dado a la vez. También en anfitriones Windows el puerto por defecto RDP (3389) podría estar ya usado por el servidor RDP de Windows, en este caso deberías elegir otro puerto para tu VM.

En la GUI de VB el servidor VRDP está deshabilitado por defecto, puede ser fácilmente habilitado en una VM con la GUI o con VBoxManage:

VBoxManage modifyvm <vmname> -vrdp on

Si usas *VBoxHeadless* (descripto debajo), el soporte VRDP será habilitado automáticamente.

Ajustes adicionales para *modifyvm* son *-vrdpport* y *-vrdpauthtype*; mira el capítulo 8.5 por detalles.

7.4.1 VBoxHeadless, el servidor sólo VRDP

Aunque que el servidor VRDP incorporado en la GUI es perfectamente capaz de correr remotamente VMs, esto no es conveniente para tener que correr VB si nunca quieres tener VMs visibles localmente en primer lugar. En concreto, si

estás corriendo servidores cuyo único propósito es hospedar VMs y se supone que todas tus VMs corren remotamente sobre VRDP, entonces es inútil en absoluto tener una GUI en el servidor -especialmente desde que, en hosts Linux o Solaris, VB viene con las librerías de dependencias Qt y SDL, lo que es inconveniente si prefieres no tener en absoluto X Window en tu servidor.

Por lo tanto VB viente con otra interfaz que no produce salida visible en el anfitrión en absoluto, pero en lugar de ello sólo entrega datos VRDP. Con VB 1.6, este "servidor acéfalo" ahora es acertadamente denominado *VBoxHeadless*. (En versiones previas fue llamada *VBoxVRDP*. Para garantizar la compatibilidad hacia atrás, la instalación de VB todavía instala también un ejecutable con aquel nombre).

Para arrancar una VM con VBoxHeadless tienes dos opciones:

✓ puedes usar VBoxManage startvm <vmname> -type vrdp. La opción extra -type hace que el núcleo de VB use *VBoxHeadless* como la interfaz del motor interno de virtualización.

✓ La forma recomendada, sin embargo, es usar VBoxHeadless directamente, como sigue:

VBoxHeadless -startvm <uuid|name>

Esto por que cuando arranca la interfaz headless a través de VBoxManage, no serás capaz de ver o registrar los mensajes que VBoxHeadless puede sacar en la consola. Especialmente en errores de arranque, cuya salida puede ser conveniente para diagnóstico de problemas.

Nota que cuando usas VBoxHeadless para arrancar una VM, el servidor VRDP incorporado siempre estará habilitado, sin tener en cuenta si lo has activado en las configuraciones de la VM. Si esto es inconveniente (por ejemplo por que quieres acceder a la VM solamente vía *ssh*), arranca así:

VBoxHeadless -startvm <uuid|name> -vrdp=off

Para que el servidor VRDP use los ajustes desde la configuración de la VM, como deberían tener las otras interfaces, usa esto:

VBoxHeadless -startvm <uuid|name> -vrdp=config

7.4.2 Paso a paso: creando una VM en un servidor acéfalo

Las siguientes instrucciones te pueden dar una idea de cómo crear una VM en un servidor acéfalo en una conexión de red. Crearás una VM, establecer una conexión VRDP e instalar un SO huésped -todo sin tener que tocar el servidor acéfalo-. Todo lo que necesitas es lo siguiente:

1. VB en una máquina servidor con un SO soportado; para el siguiente ejemplo, adoptamos un servidor Linux;

2. Un archivo ISO en el servidor, conteniendo los datos de instalación para instalar el SO huésped (en el ejemplo asumimos Windows XP);

3. Una conexión de terminal al host en el que puedas acceder a una línea de comando (v.g., vía telnet o ssh);

4. Un visor RDP en el cliente remoto; en un cliente Linux podrías usar

redesktop para conectar; desde una máquina Windows, podrías el visor RDP que viene con él (“Accesorios” -> “Comunicaciones” -> “Conexión a escritorio remoto”).

Observa una vez mas que en la máquina servidor, ya que sólo usaremos el servidor acéfalo, ni Qt ni SDL ni X Window serán necesitados.

1. En el servidor acéfalo crea una VM:

VBoxManage createvm -name "Windows XP" -register

Observa que si no especificas *-register*, tendrás que usar después manualmente el comando *registervm*.

2. Asegúrate que los ajustes para esta VM son apropiados para el SO huésped que instalaremos. Por ejemplo:

```
VBoxManage modifyvm "Windows XP" -memory "256MB" \
            -acpi on -boot1 dvd -nic1 nat
```

3. Crea un disco duro virtual (en este caso, 10GB de tamaño) y regístralo con VB:

VBoxManage createhd -filename "WinXP.vdi" -size 10000 -register

4. Coloca este archivo VDI recién creado como el primer disco duro virtual de la nueva VM:

VBoxManage modifyvm "Windows XP" -hda "WinXP.vdi"

5. Registra el archivo ISO que contiene la instalación del SO que quieres instalar posteriormente:

VBoxManage openmedium dvd /ruta/completa/a/iso.iso

6. Conecta este ISO a la VM, así pueda arrancar de él:

VBoxManage modifyvm "Windows XP" -dvd /ruta/completa/ao/iso.iso

(Alternativamente, puedes usar directamente *VBoxManage controlvm dvdattach*, sin tener que registrar primero la imagen; mira por detalles el capítulo 8.7).

7. Arranca la VM usando *VBoxHeadless:*

VBoxHeadless -startvm "Windows XP"

Si todo funcionó, deberías ver un aviso de protegido por derechos de autor. Si, en vez de eso, te devuelve a la línea de comandos, entonces algo salió mal.

8. En la máquina cliente, lanza el visor RDP y trata de conectarlo al servidor. Suponiendo un cliente Linux, trata lo siguiente:

rdesktop -a 16 my.host.address

(Con *rdesktop*, la opción *-a 16* solicita una profundidad de color de 16 bits por píxel, que recomendamos. También, después de la instalación, deberías poner la profundidad de color de SO huésped al mismo valor).

7.4.3 USB remoto

Como una característica especial encima del soporte VRDP, VB soporta

también dispositivos remotos sobre la conexión. Esto es, el huésped VB que corre en una computadora puede acceder al dispositivo USB de la computadora remota, en el que los datos RDP están siendo mostrados de la misma forma en que los dispositivos USB que están conectados al anfitrión real. Esto permite correr VM en un anfitrión VB que funciona también como un servidor, donde un cliente puede conectar desde otro sitio y necesita sólo un adaptador de red y un display capaz de correr un visor RDP. Cuando un dispositivo USB es enchufado en el cliente, el servidor VB remoto puede accederlo.

Para esos dispositivos USB remotos, se aplica la misma regla en el filtro que para los otros dispositivos, como se describe en el capítulo 3.7.7.1, *Ajustes USB.* Todo lo que tienes que hacer es especificar "Remoto" (o "Cualquiera") cuando se establecen esas reglas.

El acceso a dispositivos USB remotos solo es posible si el cliente RDP soporta esta extensión. VB incluye un cliente RDP apropiado para Linux, *rdesktop-vrdp*. Más clientes RDP funcionando en otras plataformas serán proporcionados en versiones futuras de VB.

7.4.4 Autenticación RDP

Por cada VM que es accesible remotamente vía RDP, puedes determinar individualmente cuándo y cómo son autenticadas las conexiones RDP.

Para esto usa el comando VBoxManage modifyvm con la opción -vrdpauthtype. Para una introducción general, mira el capítulo 8.5. Tres métodos de autenticación están disponibles:

✓ El método "nulo" significa que no hay autenticación en absoluto; cualquier cliente puede conectarse al servidor VRDP y así a la máquina virtual. Esto es, por supuesto, muy inseguro y sólo para ser recomendado a redes privadas.

✓ El método "externo" provee autenticación externa a través de una librería especial de autenticación.

VB viene con dos librerías por defecto para autenticación externa:

– En anfitriones Linux, *VRDPAuth.so*, autentica los usuarios contra el sistema PAM del anfitrión.
– En anfitriones Windows, *VRDAuth.dll*, autentica los usuarios contra el sistema WinLogon.

En otras palabras el método "externo" por defecto lleva a cabo la autenticación con las cuentas de usuario que existen en el sistema anfitrión.

Sin embargo, puedes remplazar este módulo de autenticación con cualquier otro módulo. Para esto, VB provee una interfaz bien definida que te permite escribir tu propio módulo de autenticación (capítulo 9.3).

✓ Finalmente, el método de autenticación "huésped" realiza la autenticación con con un componente especial que viene con el Guest Additions; por consiguiente, la autenticación no es llevada a cabo con los usuarios huéspedes, sino con las cuentas del usuario anfitrión. Este método en la actualidad aún esta en pruebas y todavía no está soportado.

7.4.5 Encriptación RDP

La característica de RDP encriptación de flujo de datos (data stream encryption), que está basada en el cifrado simétrico RC4 (con claves hasta 128 bit). Las claves RC4 están siendo remplazadas en intervalos regulares (cada 4096 paquetes).

RDP proporciona tres métodos diferentes de autenticación:

1. Históricamente, la autenticación RDP4 fue usada con clientes que no realizan ningún control para verificar la identidad del servidor al cual se conectan. Desde que las credenciales de usuarios pueden ser obtenidos usando un ataque *man in the middle* (MITM), esta autenticación es insegura y generalmente no debería ser usada.

2. La autenticación RDP5.1 emplea un certificado de servidor para que el cliente tenga la clave pública. De esta forma está garantizado que el servidor posea la correspondiente clave privada. Sin embargo, como esta clave privada fuertemente codificada se hizo pública hace muchos años, esta autenticación también es insegura y no puede ser recomendada.

3. La autenticación RDP5.2 está basada en TLS 1.0 con certificados proporcionados al cliente -customer-supplied certificates-. El servidor facilita un certificado al cliente que debe ser firmado por una autoridad certificadora (CA) en la que confía el cliente (para clientes RDP Microsoft, la CA tiene que ser añadida a la base de datos de Autoridades de Certificados Raíz de Confianza de Microsoft -Windows Trusted Root Certificate Authorities-. VB te permite suministrar tu propia CA y tu certificado de servidor y usar OpenSSL para encriptación.

Aunque VB soporta todo lo de arriba, sólo la autenticación RDP5.2 debería ser usada en ambientes donde la seguridad es una preocupación. Ya que el cliente que se conecta al servidor determina el tipo de encriptación que será usado, con *rdesktop*, el visor RDP de Linux, usa las opciones -4 o -5.

7.4.6 Conexiones VRDP múltiples

El servidor VRDP incorporado de VB soporta conexiones simultáneas a la misma VM desde diferentes clientes. Todos los clientes conectados ven la misma pantalla de salida y comparten un puntero de mouse y el foco del teclado. Esto es igual a varias personas usando la misma computadora al mismo tiempo, turnándose en el teclado. El siguiente comando habilita el modo múltiple conexión:

```
VBoxManage modifyvm VMNAME -vrdpmulticon on
```

Si el anfitrión usa múltiples monitores entonces múltiples modos de conexión deben estar activos para usarlos al mismo tiempo (capítulo 9.6).

(...)

9 Temas avanzados

(...)

9.11 Configurando las direcciones de una interfaz NAT

En modo NAT, la interfaz de red del huésped es asignada por defecto al

rango de IPv4 10.0.x.0/24, donde x corresponde a la instancia de la interfaz NAT +2 de la VM. Entonces x es 2 si hay sólo una instancia NAT activa a la vez. En el caso en el que el huésped es asignado a la dirección 10.0.2.15, la puerta de enlace es puesta a 10.0.2.2 y el nombre del servidor puede ser establecido a 10.0.2.3.

Si, por cualquier razón, la red NAT necesita ser cambiada, esto puede lograrse con el siguiente comando:

VBoxManage modifyvm "My VM" -natnet1 "192.168/16"

Este comando reservará las direcciones 192.168.0.0 ...192.168.254.254 para el primer ejemplo de red NAT de "My VM". La IP del huésped debería ser asignada a 192.168.0.15 y la puerta de enlace podría ser establecida a 192.168.0.2.

(...)

11 Detección de problemas

(...)

11.4 Anfitriones Windows

(...)

11.4.3 Respuesta lenta al usar el Cliente RDP de Microsoft

Al conectar a una VM por el cliente RDP de Microsoft (llamado Conexión a Escritorio Remoto), pueden haber extensos retrasos entre la entrada (mover el mouse en un menú es la situación más clara) y la salida. Esto es por que el cliente RDP acumula la entrada por cierto tiempo antes de enviarlo al servidor VRDP incorporado en VB.

El intervalo puede ser disminuido mediante un ajuste a la clave del registro de Windows a un valor más que el de defecto, de 100. la clave no existe inicialmente y debe ser de tipo DWORD. La unidad para estos valores es milisegundos. Valores alrededor de 20 son adecuados para conexiones de escasa banda ancha entre el cliente RDP y el servidor. Valores alrededor de 4 pueden ser usados para una conexión Ethernet de un gigabit. Generalmente valores debajo de 10 logran un desempeño muy próximo al del dispositivo de entrada local y la pantalla del anfitrión en el que está funcionando la VM.

Dependiendo de si los ajustes deberían ser cambiados para un usuario individual o para el sistema,

HKEY_CURRENT_USER\Software\Microsoft\Terminal Server Client\Min Send Interval

o

HKEY_LOCAL_MACHINE\Software\Microsoft\Terminal Server Client\Min Send Interval

pueden ajustarse apropiadamente.

(...)

Glosario

API Interfaz de Programación de Aplicación (Application Programming Interface).

COM. Modelo de Objeto Componente de Microsoft (Microsoft Component Objet Model). Una infraestructura de programación para software modular. COM permite proporcionar a las aplicaciones interfaces de programación que pueden ser accedidas desde varios otros lenguajes y aplicaciones. VB hace uso de COM interna y externamente para proveer API comprensible a desarrolladores de terceros.

DHCP. Protocolo de Configuración Dinámica de Host (Dynamic Host Configuration Protocol): permite que un dispositivo de red adquiera automáticamente su dirección IP (y otros detalles de red), para evitar tener que configurar todos los dispositivos de una red con direcciones IP fijas. VB tiene un servidor DHCP incorporado que entrega una dirección IP a una VM cuando la red está configurada a NAT; ver capítulo 6, *Red virtual.*

MAC. Control de Acceso de Medios (Media Access Control): una parte de una tarjeta de red Ethernet. Una dirección MAC es un número de 6 byte que identifica una tarjeta de red. Generalmente escrita en sistema hexadecimal donde los bytes son separados por dos puntos, v.g., 00:17:3A:5E:CB:08

NAT. Traducción de Direcciones de Red (Network Address Translation): una técnica para compartir interfaces de red mediante la cual una interfaz modifica el código y/o el objetivo de direcciones IP de paquetes de red de acuerdo a reglas específicas. Comúnmente empleada por routers y firewalls para proteger una red interna de internet, VB puede usar NAT para compartir fácilmente un hardware de red físico de un host con su VM. Ver el capítulos 6.4, *Traducción de dirección de red (NAT).*

PXE. Entorno de Ejecución de Prearranque (Preboot Execution Environment): un estándar industrial para el arranque de computadoras desde redes remotas. Incluye DHCP para la configuración y TFTP para la transferencia de archivos. Mediante el uso de UNDI, una pila controladores de hardware independiente está disponible desde código bootstrap para el acceso de tarjeta de red.

RDP. Protocolo de Escritorio Remoto (Remote Desktop Protocol). Un protocolo desarrollado por Microsoft como una extensión al protocolo de video conferencia ITU T.128 y T.124. Con RDP, una PC puede ser controlada desde un lugar remoto usando una conexión de red sobre la cual los datos son transferidos en ambas direcciones. Generalmente las actualizaciones gráficas y el audio son enviados desde la máquina remota y las acciones de entrada del mouse y el teclado son enviados desde el cliente. VB contiene una implementación mejorada de los estándares pertinentes llamada "VirtualBox RDP" (VRDP). Que es compatible en gran medida con la implementación RDP de Microsoft

Nota del traductor: los siguientes capítulos y secciones pertenecen a la versión 1.6.2 del manual y fueron descartados en la versión 2.1.0. Creo que hay partes que pueden ser tenidas en cuenta (especialmente, por que ilustran acerca de configuraciones avanzadas)

6.5 Introducción a interfaz anfitrión

Puedes crear varias interfaces anfitrión en el sistema anfitrión (mira la siguiente subsección para instrucciones sobre cómo hacerlo), pero cada una de ellas sólo puede ser conectada a una sola tarjeta de red virtual en un solo huésped a la vez. En otras palabras, por cada tarjeta de red virtual que usase *interfaz anfitrión*, necesitarás crear una nueva interfaz en el anfitrión.

Advertencia: la configuración de interfaz anfitrión requiere cambios en tu configuración de la red del anfitrión. No cambies ajustes de conexión en sistemas remotos o de producción a menos que sepas lo que estás haciendo. En especial, probablemente no querrás configurar interfaz anfitrión a una máquina remota que has conectado vía *ssh.*

Hay pocos límites en la cantidad de sistemas que pueden ser creados usando interfaz anfitrión. Por su sencillez, sólo describiremos una configuración simple usando puente de red para los diferentes SO anfitriones que VB soporta. Para necesidades de conexión más avanzadas, recomendamos que consultes la documentación general acerca de redes en tu SO anfitrión.

El puente de red es uno de las formas más sencillas de usar interfaz de anfitrión. El puente permite conectar varios dispositivos de red juntos por software, de modo que los datos enviados a uno de los dispositivos serán enviados a todos ellos. Para nuestros propósitos, esto significa que las VM pueden enviar paquetes a través de la tarjeta de red del anfitrión, usando su propio hardware de dirección de red, y recibir paquetes que se le envíen. Otras computadoras en tu red verán a tu huésped como si estuviera físicamente conectada a la red. Necesitarás hardware de cableado de red (Ethernet) en el anfitrión, por esto es que los más comunes dispositivos de red inalámbricos no soportan puenteo.

En algunos entornos de red (con frecuencia compañías de red), son tomadas medidas preventivas: a varias direcciones MAC, que están siendo usadas en una sola interfaz de red, se les bloquea temporalmente la comunicación con la interfaz. Esto para prevenir ciertos ataques de red, pero también para proteger configuraciones de puentes que funcionan correctamente.

6.6 Interfaz anfitrión y puentes en anfitriones Windows

Cuando instalas VB en un anfitrión Windows, se instala un controlador de red especial en tu sistema. Éste controlador, NDIS de VB, puede ser usado para crear interfaces anfitrión adicionales. Éstos deben ser creados explícitamente antes de que puedan ser conectados a una VM.

Usa la herramienta VBoxManage para crear nuevas interfaces anfitrión en tu sistema Windows:

VBoxManage createhostif "VM1 external"

Alternativamente puedes usar la configuración de red en la GUI de VB para crear y borrar interfaces de anfitrión. Cada nueva interfaz anfitrión así creada aparece como una tarjeta de red adicional en las propiedades de "Conexiones de red" de Windows estándar. Después de haberla creado de esta forma, puedes seleccionar "interfaz anfitrión" como el modo de conexión en la ventana de configuración de una VM y seleccionar la nueva interfaz en la lista desplegable "Nombre de la interfaz". Con el ejemplo de arriba, esta lista debería contener "VM1 external".

Si tu anfitrión está corriendo Windows XP o más reciente, puedes también usar la característica de puenteo incorporada para conectar tu interfaz anfitrión a tu tarjeta de red física. Después de crear las interfaces anfitrión deseadas, selecciona tu adaptador de red físico en la carpeta ""Conexiones de red" y los adaptadores de interfaces anfitrión deseados y selecciona "Conexiones de puente" en el menú contextual. Observa que tienes que transferir tus configuraciones de red desde tu adaptador de red físico a la conexión de puente como se mencionó arriba, porque tu adaptador físico sólo funcionará como un medio de transporte en tu configuración de puente. Cuando más de una conexión está activa en un puente, Windows pondrá automáticamente tu adaptador físico Ethernet en modo promiscuo, así que recibirá datos de red de todas las conexiones puenteadas.

6.7 Interfaz anfitrión y puentes en anfitriones Linux

Antes de continuar, por favor lee el capítulo 6.5, *Introducción a interfaz anfitrión*.

Nota: hay algunos cambios a la forma en que la configuración de la interfaz anfitrión dinámico es hecha en VB 1.4.0, debido a los cambios en las versiones 2.6.18 y posteriores del kernel Linux. Además, esta sección entera del manual fue reescrita para la versión 1.4.0 de VB. Por favor relee estas secciones si has usado interfaces dinámicas en versiones anteriores.

Desde que el kernel de Linux ha incorporado soporte para dispositivos de redes virtuales (también llamados interfaces TAP), VB hace uso de esto en Linux en vez de proporcionar drivers personalizados para red anfitrión. Los interfaces TAP se comportan como interfaces de red físicos en tu anfitrión y trabajarán con cualquier herramienta de red instalada en tu sistema anfitrión. Desde el punto de vista del anfitrión, pareciera que la tarjeta de red del huésped está conectada a la interfaz TAP con un cable de red. A fin de usar interfaz anfitrión en VB, debes tener acceso al dispositivo /dev/net/tun. Comprueba a qué grupo pertenece este dispositivo y asegúrate que cualquier usuario que necesite a la red anfitrión de VB sea miembro de este grupo. En la mayoría de los casos, este dispositivo pertenecerá al grupo *vboxusers*.

En anfitriones Linux, tienes la opción de crear interfaces de red *permanentes* a los que los huéspedes pueden conectarse cuando son creados o de crear una interfaz *dinámica* para un huésped cuando es iniciado y removerlo cuando es detenido. Las interfaces permanentes son mas adecuadas para anfitriones con un conjunto de huéspedes que no realiza cambios con frecuencia (tales como algunos sistemas de servidores), y son más fáciles de configurar. Teniendo VB

creadas las interfaces dinámicamente proporciona más flexibilidad, pero normalmente requerirá que ingreses una contraseña de administrador cada vez que una interfaz es creada o eliminada.

6.7.1 Interfaces anfitrión permanentes y puentes

En anfitriones Linux, la configuración de una interfaz anfitrión permanente por lo general consiste de tres pasos:

1. debes crear un puente en el anfitrión y agregarle una de las interfaces de red físicas del anfitrión, generalmente *eth0.* Esto te dejará conectar esa interfaz a las interfaces virtuales usadas por las VM.

Ten en cuenta que el puenteo es un concepto Ethernet no un concepto TCP/IP. En redes físicas el puente es normalmente usado para conectar dos redes Ethernet, dejando que las computadoras en una red se comuniquen con las computadoras en la otra a través de un solo punto de contacto sin tener que fusionar las redes en una sola.[4]

2. Por cada tarjeta de red huésped que use interfaz anfitrión, debes crear una nueva interfaz anfitrión "virtual" (generalmente denominada *vbox0* o similar) y agregarla al puente.
3. Finalmente, especificar el nombre de la nueva interfaz anfitrión en los ajustes de la tarjeta de red de la VM.

Desafortunadamente, las distros Linux difieren sustancialmente acerca de cómo configurar una red. Por eso no podemos proporcionar instrucciones para todas las distribuciones Linux. Nos hemos limitado a describir cómo configurar un puente en Debian, Ubuntu, Fedora/Red Hat y OpenSUSE; además, ofrecemos algunas instrucciones genéricas para usuarios avanzados.

VB viene con dos utilidades, *VBoxAddIF* y *VBoxDeleteIF*, que trabajan en todas las distros. Estas herramientas te permiten crear y eliminar interfaces anfitrión permanentes (que serán recreadas automáticamente cava vez que enciendas tu computadora anfitrión) y opcionalmente agregarlas a un puente existente. También suministramos una utilidad llamada *VBoxTunct1* que puedes usar para crear una interfaz temporal. Estas herramientas son descriptas en el capítulo 6.7.1.5, *Utilidades para interfaz anfitrión en Linux*. Incluso si planeas usar interfaces anfitrión para crear otros sistemas de red, que es lo que describimos aquí, te recomendamos que leas las siguientes instrucciones a fin de entender cómo trabajan las interfaces. Algunas distros (como Debian y Ubuntu) tienen incorporadas herramientas para crear interfaces anfitrión; puedes también usar esas herramientas en esas distros.

6.7.1.1 Anfitriones Debian y Ubuntu

Para configurar un interfaz anfitrión en un moderno anfitrión Debian o Ubuntu, sigue estos pasos:

1. instala el paquete de utilidades para puentes (*bridge*-utils). Puedes

4 Una introducción útil a los puentes puede encontrarse aquí: http://gentoo-wiki.com/HOWTO_setup_a_gentoo_bridge. Aunque está dirigida a un sistema Gentoo, contiene información genérica útil.

hacerlo desde la línea de comandos como sigue:

sudo apt-get install bridge-utils

2. Luego debes agregar una línea al archivo /etc/network/interfaces para describir el puente. La siguiente muestra de entrada crea un puente llamado *br0,* le agrega la interfaz anfitrión *eth0* y le indica obtener una dirección IP usando DHCP así que al anfitrión le resta ser capaz de acceder a la red.

```
auto br0
iface br0 inet dhcp
bridge_ports eth0
```

Probablemente querrás cambiar esto para ajustarlo a las necesidades de tu red. En concreto, quizá quieras asignar una dirección IP estática al puente. Encontrarás mas información en los archivos:

a) /usr/share/doc/bridge-utilities/README.Debian.gz and
b) /usr/share/doc/ifupdown/examples/network-interfaces.gz.

3. Reinicia la red en el anfitrión:

sudo /etc/init.d/networking restart

Después de esto el puente será recreado cada vez que enciendas el sistema anfitrión.

4. Ahora, para crear una interfaz anfitrión permanente llamada *vbox0* (todas las interfaces anfitrión creadas de esta forma deben ser denominados *vbox* seguido de un número) y agregarla al puente de red creado arriba, usa el siguiente comando (mira el capítulo 6.7.1.5, *Utilidades para interfaz anfitrión en Linux,* para mas detalles):

sudo VBoxAddIF vbox0 <user> br0

Remplaza <*user*> con el nombre del usuario que se supone es capaz de usar la nueva interfaz.

Para indicarle a VB que use la interfaz, elige la VM que se usará en la ventana principal de VB, configura uno de sus adaptadores de red para usar *interfaz anfitrión* (usando "Configuración", "Red", "Conectar a") e ingresa *vbox0* en el campo "Nombre de la interfaz". Sólo puedes usar una determinada interfaz (*vbox0, vbox1* y así sucesivamente) con un solo adaptador de red virtual.

Alternativamente, puedes usar la herramienta de línea de comandos VBoxManage (en este ejemplo estamos conectando la interfaz a la primera tarjeta de red de la VM "My VM"):

VBoxManage modifyvm "My VM" -hostifdev1 vbox0

Para configurar una interfaz anfitrión usando los métodos nativos de Debian y Ubuntu, haz lo siguiente en vez de los cuatro pasos de arriba:

1. Instala el paquete Utilidades de modo usuario Linux (*uml-utilities)*, que contiene herramientas para crear interfaces TAP. Puedes hacerlo desde la línea de comandos como sigue:

sudo apt-get install uml-utilities

A fin de que VB sea capaz de acceder la interfaz, el usuario que correrá la VM debe ser agregado al grupo *uml-net*, por ejemplo con el siguiente comando (remplaza <*user*> con su nombre de usuario):

sudo gpasswd -a <user> uml-net

Tendrás que loguearte otra vez para que los cambios sean efectivos.

2. Para describir la interfaz TAP a tu sistema Debian o Ubuntu, agrega una entrada al archivo /etc/network/interfaces. Ésta nombra la interfaz y también debe especificar el usuario que correrá la VM usando la interfaz.

La siguiente muestra de entrada crea la interfaz *tap0* para el usuario <*user*> (otra vez, reemplaza con tu nombre de usuario):

```
auto tap0
iface tap0 inet manual
up ifconfig $IFACE 0.0.0.0 up
down ifconfig $IFACE down
tunctl_user <user>
```

Probablemente querrás cambiar la entrada basado en tus necesidades de red. El archivo /usr/share/doc/uml-utilities/README.Debian en tu computadora anfitrión tendrá documentación adicional.

3. Para agregar la interfaz TAP al puente, remplaza la linea

bridge_ports eth0

en la sección puente en /etc/network/interfaces con la línea

bridge_ports eth0 tap0

4. Reinicia la red en el anfitrión:

sudo /etc/init.d/networking restart

6.7.1.2 Puentes en anfitriones OpenSUSE

Las siguientes instrucciones aclaran cómo crear un puente en openSUSE. Observa que el puenteo en anfitriones openSUSE pueden no trabajar apropiadamente si estás usando NetworkManager para administrar tus conexiones de red. Para crear un puente en un anfitrión openSUSE recién creado, primero debes instalar el paquete de utilidades de puentes (*bridge-utils)*. Si trabajas desde la línea de comandos esto puede ser hecho como sigue:

sudo /sbin/yast -i bridge-utils

Entonces debes crear un archivo de texto describiendo el puente que será creado. El nombre del archivo debe concordar con el nombre del puente que deseas crear. Para crear el puente *br0*, deberías llamar el archivo /etc/sysconfig/network/ifcfg-br0. Abajo tienes un ejemplo de un archivo que crea un puente incluyendo el dispositivo de red *eth0,* obtiene una dirección IP mediante DHCP (a través del dispositivo de red) y es iniciado automáticamente cuando arranca openSUSE. Probablemente querrás ajustar esto para que

coincida con tus requerimientos de red.

```
BOOTPROTO='dhcp'
NETMASK='255.255.255.0'
STARTMODE='auto'
USERCONTROL='no'
DHCLIENT_TIMEOUT=30
BRIDGE='yes'
BRIDGE_PORTS='eth0'
```

Para que funcione este ejemplo, también necesitarás cambiar la configuración de la interfaz de red *eth0* a una dirección IP estática de 0.0.0.0, ya que openSUSE no lo hace automáticamente cuando la interfaz es agregada al puente. Puedes hacerlo usando la interfaz gráfica o cambiando los siguientes ajustes en el archivo /etc/sysconfig/network/ifcfg-eth-xx:xx:xx:xx:xx:xx, donde la última parte debería ser remplazada con las direcciones de hardware de la tarjeta de red.

```
BOOTPROTO='static'
IPADDR='0.0.0.0'
```

Puedes activar el puente inmediatamente después de crearlo con el comando:

```
sudo /sbin/ifdown eth0
sudo /sbin/ifup br0
```

El puente deberá ser activado automáticamente desde ya o cuando el anfitrión es reiniciado.

Ahora, para crear una interfaz anfitrión permanente llamada *vbox0* (todas las interfaces anfitrión creadas de esta forma deben ser nombradas *vbox* seguido de un número) y agregarlo al puente de red creado arriba, usa el siguiente comando (mira el capítulo 6.7.1.5, *Utilidades para interfaces anfitrión Linux*, para mas detalles).

```
sudo VBoxAddIF vbox0 <user> br0
```

Remplaza <*user*> con el nombre del usuario que se supone es capaz de usar la nueva interfaz.

Para indicarle a VB que use esta interfaz (*vbox0*) para una VM, selecciónala en la ventana principal, ajusta uno de sus adaptadores de red para que use *interfaz anfitrión* (usando "Configuración", "Red", "Conectar a") e ingresa "vbox0" en el campo "Nombre de la interface". Sólo puedes usar una interfaz determinada (*vbox0, vbox1* y así sucesivamente) con una sola VM.

Alternativamente, puedes usar herramienta de línea de comandos VBoxManage (en este ejemplo estamos conectando la interfaz a la primera tarjeta de red de la VM "My VM":

```
VBoxManage modifyvm "My VM" -hostifdev1 vbox0
```

6.7.1.3 Puentes en anfitriones RedHat y Fedora

Primero debes instalar el paquete utilidades de puentes (*bridge-utils*). Luego debes crear un archivo de configuración describiendo el puente que quieres

crear. Lo siguiente es el contenido de un ejemplo de archivo de configuración /etc/sysconfig/network-scripts/ifcfg-br0, que pone el puente *br0* a obtener sus direcciones IP usando DHCP y arrancar automáticamente cuando el sistema es iniciado. Seguramente querrás ajustarlo para que coincida con tus requerimientos de red.

```
DEVICE=br0
TYPE=Bridge
BOOTPROTO=dhcp
ONBOOT=yes
```

Para agregar la tarjeta de red *eth0* al puente, añade la siguiente línea al final del archivo /etc/sysconfig/network-scripts/ifcfg-eth0:

```
BRIDGE=br0
```

Puedes activarlo inmediatamente después de crearlo con el comando:

```
sudo /sbin/service network restart
```

El puente será activado automáticamente desde ahora o cuando el anfitrión es arrancado.

Ahora, para crear una interfaz anfitrión permanente denominada *vbox0* (todas las que son creadas de esta forma deben ser nombradas *vbox* seguido de un número) y agregarla al puente de red creado arriba, usa el siguiente comando:

```
sudo VBoxAddIF vbox0 <user> br0
```

Remplaza <*user*> con el nombre del usuario que se supone puede usar la nueva interfaz.

Para indicarle a VB que use esta interfaz (*vbox0*) para una VM, selecciona la VM en la ventana principal, configura uno de sus adaptadores de red que use *interfaz anfitrión* (usando "Configuración", "Red", "Conectar a") e ingresa "vbox0" en el campo "Nombre de la interface". Sólo puedes usar una interfaz determinada (*vbox0, vbox1* y así sucesivamente) con una sola VM.

Alternativamente, puedes usar herramienta de línea de comandos VBoxManage (en este ejemplo estamos conectando la interfaz a la primera tarjeta de red de la VM "My VM":

```
VBoxManage modifyvm "My VM" -hostifdev1 vbox0
```

6.7.1.4 Puentes con otras distribuciones

Las distros Linux más modernas proporcionan su propia forma de configurar puentes ethernet. Recomendamos que sigas las instrucciones dadas por tu distro para hacerlo. Para las distros que no proporcionen un método, abajo facilitamos instrucciones genéricas. Por favor asegúrate que entiendes perfectamente el modo en que trabajan los scripts de red de tu distro, ya que implica hacer cambios a la configuración de red de tu anfitrión en formas normalmente sólo hechas por los scripts de red, y así pueden interferir con tu configuración de red.

Antes que nada, necesitarás instalar los bridge utilities (generalmente

llamados *bridge-utils o similar*). Una vez instalados, como *root*, sigue estas instrucciones para crear y configurar un puente:

1. crea un nuevo puente con este comando

brctl addbr br0

2. si no estás usando DHCP, corre *ifconfig* y apunta la configuración de red de tu interfaz de red existente (v.g. *eth0*), que necesitaremos para copiar al puente en un minuto.

3. elimina la configuración de la dirección IP del dispositivo de red existente (v.g. *eth0*) con:

ifconfig eth0 0.0.0.0

Advertencia: perderás conectividad de red en *eth0* en este momento.

4. agrega tu adaptador de red al puente:

brctl addif br0 eth0

5. transfiere la configuración de red previamente usada por tu adaptador ethernet físico al nuevo puente. Si estás usando DHCP, esto debería funcionar:

dhclient br0

De lo contrario, corre ifconfig br0 x.x.x.x netmask x.x.x.x y usa los valores que anotaste previamente.

6. Para crear una interfaz anfitrión permanente llamada *vbox0* (todas las que sean creadas de esta forma deben ser llamadas *vbox* seguido de un número) y agrégala al puente de red creada arriba, usa el siguiente comando:

VBoxAddIF vbox0 <user> br0

Remplaza <*user*> con el nombre del usuario capaz de usar la nueva interfaz.

6.7.1.5 Utilidades de interfaz anfitrión para Linux

Aunque Linux viene con soporte incorporado para interfaces de red virtuales, no hay muchos programas disponibles para administrarlas. VB facilita tres herramientas para este propósito: *VboxAddIF,VBoxDeleteIF y VBoxTunct1.* Ésta última es de hecho la utilidad *tunct1* del proyecto User Mode Linux. En esta sección describiremos cómo usarlas.

VboxAddIF crea una interfaz TAP permanente que no desaparecerá cuando el sistema anfitrión es reiniciado. Esta interfaz debería ser llamada *vbox0, vbox1* o similar. El siguiente comando crea la interfaz *vbox0* para el usuario <*user*> y la agrega al puente *br0*. Si no deseas agregar la interfaz a un puente, puedes omitir el nombre del puente.

sudo VBoxAddIF vbox0 <user> br0

Para crear una interfaz TAP temporal que desaparecerá cuando el anfitrión es reiniciado, usa el comando VBoxTunct1. El siguiente ejemplo crea la interfaz *vbox0* para el usuario <*user*>:

sudo VBoxTunctl -t vbox0 -u <user>

Si tienes instaladas las utilidades para puentes (mira las secciones anteriores), puedes añadir esta interfaz temporal a un puente Ethernet usando el comando

```
sudo brctl addif br0 vbox0
```

Remplaza *br0* con el nombre del puente y *vbox0* con el nombre de la interfaz. Antes de que puedas usarla, aún necesitarás hacerla activa (o "subirla" en la terminología de red, generalmente usando la utilidad estándar Linux *ifconfig*) y configurarla con una dirección IP e información relacionada. Para eliminar una interfaz temporal, realiza lo siguiente, remplazando *vbox0* con el nombre de la interfaz a suprimir:

```
sudo VBoxTunctl -d vbox0
```

6.7.2 Creando interfaces dinámicamente cuando una VM arranca

Como una alternativa a las interfaces permanentes previamente descriptas, puedes indicarle a VB que ejecute comandos (generalmente scripts) para poner en marcha tu red dinámicamente, cada vez que una VM arranca o se detiene. Normalmente esto es hecho para crear las interfaces TAP en el momento en que la VM se pone en marcha, aunque también puedes usar esta característica para configurar interfaces existentes. Si no estás usando interfaces permanentes entonces el comando de arranque debería escribir el nombre de la interfaz que ha creado, generalmente algo como *tap0* o *tap2*, a su salida estándar (el comando *VBoxTunct1 -b* hace exactamente esto) y cuando la máquina se detiene el comando ejecutado debería eliminar la interfaz otra vez.

Los comando y scripts usados en la configuración de red dependerán de lo que quieras colocar. Ambos comandos son dados a un archivo que describe el dispositivo TAP Linux como su primer argumento (ésto es válido si la VM está usando previamente interfaces creadas) y el nombre de la interfaz, si es conocida, como segundo argumento. En la mayoría de las circunstancias, sólo querrás usar el segundo argumento.

Aquí hay un ejemplo de una configuración de un script que crea una interfaz TAP y lo agrega al puente de red *br0*.

```
#!/bin/bash

# Crea una nueva interfaz TAP para el usuario 'vbox' y recuerda su nombre.
interface=`VBoxTunctl -b -u vbox`

# Si por alguna razón la interfaz no pudiera ser creada, devuelve 1
# indica esto a VirtualBox.
if [ -z "$interface" ]; then
exit 1

fi

# Escribe el nombre de la interfaz a la salida estándar.
echo $interface

# Sube la interfaz.
/sbin/ifconfig $interface up
```

```
# Y agrégala al puente.
/sbin/brctl addif br0 $interface
```

Si este script es guardado como /home/vbox/setuptap.sh y hecho ejecutable, puede ser usado para crear una interfaz TAP cuando una VM es arrancada, mediante la configuración de uno de los adaptadores de red de la máquina para que use *interfaz anfitrión* (sin especificar un dispositivo en el campo "Nombre de la interface") e ingresando gksudo /home/vbox/setuptap.sh en el campo "Aplicación de arranque" -setup application- (remplaza *gksudo* por *kdesu*, o por lo que sea apropiado a tu sistema). Alternativamente puedes usar la herramienta de línea de comandos VBoxManage (en el siguiente ejemplo para una máquina llamada "Linux VM"):

```
VBoxManage modifyvm "Linux VM" -tapsetup1 "gksudo /home/vbox/setuptap.sh"
```

Un ejemplo de un script correspondiente para eliminar la interfaz del puente y apagarla debería ser:

```
#!/bin/bash

# Eliminar la interfaz del puente. El segundo parámetro del script es
# el nombre de la interfaz.
/sbin/brctl delif br0 $2

# Y usa VBoxTunctl para eliminar la interfaz.
VBoxTunctl -d $2
```

Si es guardado como /home/vbox/cleanuptap.sh y hecho ejecutable, la VM puede ser informada para ejecutarlo cuando se apaga ingresando gksudo /home/vbox/cleanuptap.sh, en el campo "Aplicación de finalización" -termination application- en los ajustes de configuración de red de VB, o mediante VBoxManage:

```
VBoxManage modifyvm "Linux VM" -tapterminate1
"gksudo /home/vbox/cleanuptap.sh"
```

Nota: la interfaz -front end- VBoxSDL para VB (mira el capítulo 7.3, *VboxSDL, el visualizador simplificado de VM*) permite una forma adicional de configurar interfaces TAP si es arrancado desde un proceso padre personalizado. Este proceso padre puede asignar la interfaz TAP requerida y permitir que VB herede el control del archivo. Para que funcione, el archivo descriptor tiene que ser pasado a VBoxSDL usando la opción -tapfd<N> <fd>. En este caso, los scripts de arranque y terminación no serán llamados.