Chapter 6. Virtual Networking

Table of Contents

As mentioned in Section 3.9, “Network Settings”, Oracle VM VirtualBox provides up to eight virtual PCI Ethernet cards for each virtual machine. For each such card, you can individually select the following:

- The hardware that will be virtualized.
- The virtualization mode that the virtual card operates in, with respect to your physical networking hardware on the host.

Four of the network cards can be configured in the **Network** section of the **Settings** window in VirtualBox Manager. You can configure all eight network cards on the command line using **VBoxManage modifyvm**. See Section 8.10, “VBoxManage modifyvm”.

This chapter explains the various networking settings in more detail.

6.1. Virtual Networking Hardware

For each card, you can individually select what kind of *hardware* will be presented to the virtual machine. Oracle VM VirtualBox can virtualize the following types of networking hardware:

- AMD PCNet PCI II (Am79C970A)
- AMD PCNet FAST III (Am79C973), the default setting
- Intel PRO/1000 MT Desktop (82540EM)
- Intel PRO/1000 T Server (82543GC)
- Intel PRO/1000 MT Server (82545EM)
- Paravirtualized network adapter (virtio-net)

The PCNet FAST III is the default because it is supported by nearly all operating systems, as well as by the GNU GRUB boot manager. As an exception, the Intel PRO/1000 family adapters are chosen for some guest operating system types that no longer ship with drivers for the PCNet card, such as Windows Vista.

The Intel PRO/1000 MT Desktop type works with Windows Vista and later versions. The T Server variant of the Intel PRO/1000 card is recognized by Windows XP guests without additional driver installation. The MT Server variant facilitates OVF imports from other platforms.

The Paravirtualized network adapter (virtio-net) is special. If you select this adapter, then Oracle VM VirtualBox does *not* virtualize common networking hardware that is supported by common guest operating systems. Instead, Oracle VM VirtualBox expects a special software interface for virtualized environments to be provided by the guest, thus avoiding the complexity of emulating networking hardware and improving network performance. Oracle VM VirtualBox provides support for the industry-standard *virtio* networking drivers, which are part of the open source KVM project.

The virtio networking drivers are available for the following guest operating systems:

- Linux kernels version 2.6.25 or later can be configured to provide virtio support. Some distributions have also back-ported virtio to older kernels.
- For Windows 2000, XP, and Vista, virtio drivers can be downloaded and installed from the KVM project web page:

  http://www.linux-kvm.org/page/WindowsGuestDrivers.

Oracle VM VirtualBox also has limited support for *jumbo frames*. These are networking packets with more than 1500 bytes of data, provided that you use the Intel card virtualization and bridged networking. Jumbo frames are not supported with the AMD networking devices. In those cases, jumbo packets will silently be dropped for both the transmit and the receive direction. Guest operating systems trying to use this feature will observe this as a packet loss, which may lead to unexpected application behavior in the guest. This does not cause problems with guest operating systems in their default configuration, as jumbo frames need to be explicitly enabled.

6.2. Introduction to Networking Modes

Each of the networking adapters can be separately configured to operate in one of the following modes:

- **Not attached.** In this mode, Oracle VM VirtualBox reports to the guest that a network card is present, but that there is no connection. This is as if no Ethernet cable was plugged into the card. Using this mode, it is possible to *pull* the virtual Ethernet cable and disrupt the connection, which can be useful to inform a guest operating system that no network connection is available and enforce a reconfiguration.
- **Network Address Translation (NAT).** If all you want is to browse the Web, download files, and view email inside the guest, then this default mode should be sufficient for you, and you can skip the rest of this section. Please note that there are certain limitations when using Windows file sharing. See Section 6.3.3, “NAT Limitations”.
- **NAT Network.** A NAT network is a type of internal network that allows outbound connections. See Section 6.4, “Network Address Translation Service”.
- **Bridged networking.** This is for more advanced networking needs, such as network simulations and running servers in a guest. When enabled, Oracle VM VirtualBox connects to one of your installed network cards and exchanges network packets directly, circumventing your host operating system's network stack.
- **Internal networking.** This can be used to create a different kind of software-based network which is visible to selected virtual machines, but not to applications running on the host or to the outside world.

- **Host-only networking.** This can be used to create a network containing the host and a set of virtual machines, without the need for the host's physical network interface. Instead, a virtual network interface, similar to a loopback interface, is created on the host, providing connectivity among virtual machines and the host.
- **Cloud networking.** This can be used to connect a local VM to a subnet on a remote cloud service.
- **Generic networking.** Rarely used modes which share the same generic network interface, by allowing the user to select a driver which can be included with Oracle VM VirtualBox or be distributed in an extension pack.

  The following sub-modes are available:

  - **UDP Tunnel:** Used to interconnect virtual machines running on different hosts directly, easily, and transparently, over an existing network infrastructure.
  - **VDE (Virtual Distributed Ethernet) networking:** Used to connect to a Virtual Distributed Ethernet switch on a Linux or a FreeBSD host. At the moment this option requires compilation of Oracle VM VirtualBox from sources, as the Oracle packages do not include it.

The following table provides an overview of the most important networking modes.

Table 6.1. Overview of Networking Modes

Mode	VM→Host	VM←Host	VM1↔VM2	VM→Net/LAN	VM←Net/LAN
Host-only	+	+	+	–	–
Internal	–	–	+	–	–
Bridged	+	+	+	+	+
NAT	+	Port forward	–	+	Port forward
NATservice	+	Port forward	+	+	Port forward

The following sections describe the available network modes in more detail.

6.3. Network Address Translation (NAT)

Network Address Translation (NAT) is the simplest way of accessing an external network from a virtual machine. Usually, it does not require any configuration on the host network and guest system. For this reason, it is the default networking mode in Oracle VM VirtualBox.

A virtual machine with NAT enabled acts much like a real computer that connects to the Internet through a router. The router, in this case, is the Oracle VM VirtualBox networking engine, which maps traffic from and to the virtual machine transparently. In Oracle VM VirtualBox this router is placed between each virtual machine and the host. This separation maximizes security since by default virtual machines cannot talk to each other.

The disadvantage of NAT mode is that, much like a private network behind a router, the virtual machine is invisible and unreachable from the outside internet. You cannot run a server this way unless you set up port forwarding. See Section 6.3.1, “Configuring Port Forwarding with NAT”.

The network frames sent out by the guest operating system are received by Oracle VM VirtualBox's NAT engine, which extracts the TCP/IP data and resends it using the host operating system. To an application on the host, or to another computer on the same network as the host, it looks like the data was sent by the Oracle VM VirtualBox application on the host, using an IP address belonging to the host. Oracle VM VirtualBox listens for replies to the packages sent, and repacks and resends them to the guest machine on its private network.

> **Note**
>
> Even though the NAT engine separates the VM from the host, the VM has access to the host's loopback interface and the network services running on it. The host's loopback interface is accessible as IP address 10.0.2.2. This access to the host's loopback interface can be extremely useful in some cases, for example when running a web application under development in the VM and the database server on the loopback interface on the host.

The virtual machine receives its network address and configuration on the private network from a DHCP server integrated into Oracle VM VirtualBox. The IP address thus assigned to the virtual machine is usually on a completely different network than the host. As more than one card of a virtual machine can be set up to use NAT, the first card is connected to the private network 10.0.2.0, the second card to the network 10.0.3.0 and so on. If you need to change the guest-assigned IP range, see Section 9.8, “Fine Tuning the Oracle VM VirtualBox NAT Engine”.

6.3.1. Configuring Port Forwarding with NAT

As the virtual machine is connected to a private network internal to Oracle VM VirtualBox and invisible to the host, network services on the guest are not accessible to the host machine or to other computers on the same network. However, like a physical router, Oracle VM VirtualBox can make selected services available to the world outside the guest through *port forwarding*. This means that Oracle VM VirtualBox listens to certain ports on the host and resends all packets which arrive there to the guest, on the same or a different port.

To an application on the host or other physical or virtual machines on the network, it looks as though the service being proxied is actually running on the host. This also means that you cannot run the same service on the same ports on the host. However, you still gain the advantages of running the service in a virtual machine. For example, services on the host machine or on other virtual machines cannot be compromised or crashed by a vulnerability or a bug in the service, and the service can run in a different operating system than the host system.

To configure port forwarding you can use the graphical **Port Forwarding** editor which can be found in the **Network** settings dialog for network adaptors configured to use NAT. Here, you can map host ports to guest ports to allow network traffic to be routed to a specific port in the guest.

Alternatively, the command line tool **VBoxManage** can be used. See Section 8.10, “VBoxManage modifyvm”.

You will need to know which ports on the guest the service uses and to decide which ports to use on the host. You may want to use the same ports on the guest and on the host. You can use any ports on the host which are not already in use by a service. For example, to set up incoming NAT connections to an **ssh** server in the guest, use the following command:

```
VBoxManage modifyvm "VM name" --nat-pf1 "guestssh,tcp,,2222,,22"
```

In the above example, all TCP traffic arriving on port 2222 on any host interface will be forwarded to port 22 in the guest. The protocol name `tcp` is a mandatory attribute defining which protocol should be used for forwarding, `udp` could also be used. The name `guestssh` is purely descriptive and will be auto-generated if omitted. The number after `--nat-pf` denotes the network card, as with other **VBoxManage** commands.

To remove this forwarding rule, use the following command:

```
VBoxManage modifyvm "VM name" --natpf1 delete "guestssh"
```

If for some reason the guest uses a static assigned IP address not leased from the built-in DHCP server, it is required to specify the guest IP when registering the forwarding rule, as follows:

```
VBoxManage modifyvm "VM name" --natpf1 "guestssh,tcp,,2222,10.0.2.19,22"
```

This example is identical to the previous one, except that the NAT engine is being told that the guest can be found at the 10.0.2.19 address.

To forward *all* incoming traffic from a specific host interface to the guest, specify the IP of that host interface as follows:

```
VBoxManage modifyvm "VM name" --natpf1 "guestssh,tcp,127.0.0.1,2222,,22"
```

This example forwards all TCP traffic arriving on the localhost interface at 127.0.0.1 through port 2222 to port 22 in the guest.

It is possible to configure incoming NAT connections while the VM is running, see Section 8.20, “VBoxManage controlvm”.

6.3.2. PXE Booting with NAT

PXE booting is now supported in NAT mode. The NAT DHCP server provides a boot file name of the form `vmname.pxe` if the directory `TFTP` exists in the directory where the user's `VirtualBox.xml` file is kept. It is the responsibility of the user to provide `vmname.pxe`.

6.3.3. NAT Limitations

There are some limitations of NAT mode which users should be aware of, as follows:

- **ICMP protocol limitations.** Some frequently used network debugging tools, such as **ping** or **traceroute**, rely on the ICMP protocol for sending and receiving messages. Oracle VM VirtualBox ICMP support has some limitations, meaning **ping** should work but some other tools may not work reliably.
- **Receiving of UDP broadcasts.** The guest does not reliably receive UDP broadcasts. In order to save resources, it only listens for a certain amount of time after the guest has sent UDP data on a particular port. As a consequence, NetBios name resolution based on broadcasts does not always work, but WINS always works. As a workaround, you can use the numeric IP of the desired server in the `\\server\share` notation.
- **Some protocols are not supported.** Protocols other than TCP and UDP are not supported. GRE is not supported. This means some VPN products, such as PPTP from Microsoft, cannot be used. There are other VPN products which use only TCP and UDP.
- **Forwarding host ports below 1024.** On UNIX-based hosts, such as Linux, Oracle Solaris, and macOS, it is not possible to bind to ports below 1024 from applications that are not run by `root`. As a result, if you try to configure such a port forwarding, the VM will refuse to start.

These limitations normally do not affect standard network use. But the presence of NAT has also subtle effects that may interfere with protocols that are normally working. One example is NFS, where the server is often configured to refuse connections from non-privileged ports, which are those ports not below 1024.

6.4. Network Address Translation Service

The Network Address Translation (NAT) service works in a similar way to a home router, grouping the systems using it into a network and preventing systems outside of this network from directly accessing systems inside it, but letting systems inside communicate with each other and with systems outside using TCP and UDP over IPv4 and IPv6.

A NAT service is attached to an internal network. Virtual machines which are to make use of it should be attached to that internal network. The name of internal network is chosen when the NAT service is created and the internal network will be created if it does not already exist. The following is an example command to create a

NAT network:

```
VBoxManage natnetwork add --netname natnet1 --network "192.168.15.0/24" --enable
```

Here, natnet1 is the name of the internal network to be used and 192.168.15.0/24 is the network address and mask of the NAT service interface. By default in this static configuration the gateway will be assigned the address 192.168.15.1, the address following the interface address, though this is subject to change. To attach a DHCP server to the internal network, modify the example command as follows:

```
VBoxManage natnetwork add --netname natnet1 --network "192.168.15.0/24" --enable --dhcp on
```

To add a DHCP server to an existing network, use the following command:

```
VBoxManage natnetwork modify --netname natnet1 --dhcp on
```

To disable the DHCP server, use the following command:

```
VBoxManage natnetwork modify --netname natnet1 --dhcp off
```

A DHCP server provides a list of registered nameservers, but does not map servers from the 127/8 network.

To start the NAT service, use the following command:

```
VBoxManage natnetwork start --netname natnet1
```

If the network has a DHCP server attached then it will start together with the NAT network service.

To stop the NAT network service, together with any DHCP server:

```
VBoxManage natnetwork stop --netname natnet1
```

To delete the NAT network service:

```
VBoxManage natnetwork remove --netname natnet1
```

This command does not remove the DHCP server if one is enabled on the internal network.

Port-forwarding is supported, using the `--port-forward-4` switch for IPv4 and `--port-forward-6` for IPv6. For example:

```
VBoxManage natnetwork modify \
  --netname natnet1 --port-forward-4 "ssh:tcp:[]:1022:[192.168.15.5]:22"
```

This adds a port-forwarding rule from the host's TCP 1022 port to the port 22 on the guest with IP address 192.168.15.5. Host port, guest port and guest IP are mandatory. To delete the rule, use the following command:

```
VBoxManage natnetwork modify --netname natnet1 --port-forward-4 delete ssh
```

It is possible to bind a NAT service to specified interface. For example:

```
VBoxManage setextradata global "NAT/win-nat-test-0/SourceIp4" 192.168.1.185
```

To see the list of registered NAT networks, use the following command:

```
VBoxManage list natnetworks
```

NAT networks can also be created, deleted, and configured using the Network Manager tool in VirtualBox Manager. Click **File**, **Tools**, **Network Manager**. See Section 6.11, “Network Manager”.

> **Note**
>
> Even though the NAT service separates the VM from the host, the VM has access to the host's loopback interface and the network services running on it. The host's loopback interface is

accessible as IP address 10.0.2.2 (assuming the default configuration, in other configurations it's the respective address in the configured IPv4 or IPv6 network range). This access to the host's loopback interface can be extremely useful in some cases, for example when running a web application under development in the VM and the database server on the loopback interface on the host.

6.5. Bridged Networking

With bridged networking, Oracle VM VirtualBox uses a device driver on your *host* system that filters data from your physical network adapter. This driver is therefore called a *net filter* driver. This enables Oracle VM VirtualBox to intercept data from the physical network and inject data into it, effectively creating a new network interface in software. When a guest is using such a new software interface, it looks to the host system as though the guest were physically connected to the interface using a network cable. The host can send data to the guest through that interface and receive data from it. This means that you can set up routing or bridging between the guest and the rest of your network.

Note

Even though TAP interfaces are no longer necessary on Linux for bridged networking, you *can* still use TAP interfaces for certain advanced setups, since you can connect a VM to any host interface.

To enable bridged networking, open the **Settings** dialog of a virtual machine, go to the **Network** page and select **Bridged Network** in the drop-down list for the **Attached To** field. Select a host interface from the list at the bottom of the page, which contains the physical network interfaces of your systems. On a typical MacBook, for example, this will allow you to select between en1: AirPort, which is the wireless interface, and en0: Ethernet, which represents the interface with a network cable.

Note

Bridging to a wireless interface is done differently from bridging to a wired interface, because most wireless adapters do not support promiscuous mode. All traffic has to use the MAC address of the host's wireless adapter, and therefore Oracle VM VirtualBox needs to replace the source MAC address in the Ethernet header of an outgoing packet to make sure the reply will be sent to the host interface. When Oracle VM VirtualBox sees an incoming packet with a destination IP address that belongs to one of the virtual machine adapters it replaces the destination MAC address in the Ethernet header with the VM adapter's MAC address and passes it on. Oracle VM VirtualBox examines ARP and DHCP packets in order to learn the IP addresses of virtual machines.

Depending on your host operating system, the following limitations apply:

- **macOS hosts.** Functionality is limited when using AirPort, the Mac's wireless networking system, for bridged networking. Currently, Oracle VM VirtualBox supports only IPv4 and IPv6 over AirPort. For other protocols, such as IPX, you must choose a wired interface.

- **Linux hosts.** Functionality is limited when using wireless interfaces for bridged networking. Currently, Oracle VM VirtualBox supports only IPv4 and IPv6 over wireless. For other protocols, such as IPX, you must choose a wired interface.

  Also, setting the MTU to less than 1500 bytes on wired interfaces provided by the sky2 driver on the Marvell Yukon II EC Ultra Ethernet NIC is known to cause packet losses under certain conditions.

  Some adapters strip VLAN tags in hardware. This does not allow you to use VLAN trunking between VM and the external network with pre-2.6.27 Linux kernels, or with host operating systems other than Linux.

- **Oracle Solaris hosts.** There is no support for using wireless interfaces. Filtering guest traffic using IPFilter is also not completely supported due to technical restrictions of the Oracle Solaris networking subsystem. These issues may be addressed in later releases of Oracle Solaris 11.

  On Oracle Solaris 11 hosts build 159 and above, it is possible to use Oracle Solaris Crossbow Virtual Network Interfaces (VNICs) directly with Oracle VM VirtualBox without any additional configuration other than each VNIC must be exclusive for every guest network interface.

  When using VLAN interfaces with Oracle VM VirtualBox, they must be named according to the PPA-hack naming scheme, such as e1000g513001. Otherwise, the guest may receive packets in an unexpected format.

6.6. Internal Networking

Internal Networking is similar to bridged networking in that the VM can directly communicate with the outside world. However, the outside world is limited to other VMs on the same host which connect to the same internal network.

Even though technically, everything that can be done using internal networking can also be done using bridged networking, there are security advantages with internal networking. In bridged networking mode, all traffic goes through a physical interface of the host system. It is therefore possible to attach a packet sniffer such as Wireshark to the host interface and log all traffic that goes over it. If, for any reason, you prefer two or more VMs on the same machine to communicate privately, hiding their data from both the host system and the user, bridged networking therefore is not an option.

Internal networks are created automatically as needed. There is no central configuration. Every internal network is identified simply by its name. Once there is more than one active virtual network card with the same internal network ID, the Oracle VM VirtualBox support driver will automatically *wire* the cards and act as a network switch. The Oracle VM VirtualBox support driver implements a complete Ethernet switch and supports both broadcast/multicast frames and promiscuous mode.

In order to attach a VM's network card to an internal network, set its networking mode to Internal Networking. There are two ways to accomplish this:

- Use the VM's **Settings** window in VirtualBox Manager. In the **Network** category of the Settings window, select **Internal Network** from the drop-down list of networking modes. Select the name of an existing internal network from the drop-down list below, or enter a new name into the **Name** field.
- Use the command line, for example:

  ```
  VBoxManage modifyvm "VM name" --nic<x> intnet
  ```

  Optionally, you can specify a network name with the command:

  ```
  VBoxManage modifyvm "VM name" --intnet<x> "network name"
  ```

  If you do not specify a network name, the network card will be attached to the network `intnet` by default.

Unless you configure the virtual network cards in the guest operating systems that are participating in the internal network to use static IP addresses, you may want to use the DHCP server that is built into Oracle VM VirtualBox to manage IP addresses for the internal network. See Section 8.50, “VBoxManage dhcpserver”.

As a security measure, by default, the Linux implementation of internal networking only allows VMs running under the same user ID to establish an internal network. However, it is possible to create a shared internal networking interface, accessible by users with different user IDs.

6.7. Host-Only Networking

Host-only networking can be thought of as a hybrid between the bridged and internal networking modes. As with bridged networking, the virtual machines can talk to each other and the host as if they were connected through a physical Ethernet switch. As with internal networking, a physical networking interface need not be present, and the virtual machines cannot talk to the world outside the host since they are not connected to a physical networking interface.

When host-only networking is used, Oracle VM VirtualBox creates a new software interface on the host which then appears next to your existing network interfaces. In other words, whereas with bridged networking an existing physical interface is used to attach virtual machines to, with host-only networking a new *loopback* interface is created on the host. And whereas with internal networking, the traffic between the virtual machines cannot be seen, the traffic on the loopback interface on the host can be intercepted.

> **Note**
>
> Hosts running recent macOS versions do not support host-only adapters. These adapters are replaced by host-only networks, which define a network mask and an IP address range, where the host network interface receives the lowest address in the range.
> The host network interface gets added and removed dynamically by the operating system, whenever a host-only network is used by virtual machines.
> On macOS hosts, choose the **Host-Only Network** option when configuring a network adapter. The **Host-Only Adapter** option is provided for legacy support.

Host-only networking is particularly useful for preconfigured virtual appliances, where multiple virtual machines are shipped together and designed to cooperate. For example, one virtual machine may contain a web server and a second one a database, and since they are intended to talk to each other, the appliance can instruct Oracle VM VirtualBox to set up a host-only network for the two. A second, bridged, network would then connect the web server to the outside world to serve data to, but the outside world cannot connect to the database.

To enable a host-only network interface for a virtual machine, do either of the following:

- Go to the **Network** page in the virtual machine's **Settings** dialog and select an **Adapter** tab. Ensure that the **Enable Network Adapter** check box is selected and choose **Host-Only Adapter** for the **Attached To** field.
- On the command line, use **VBoxManage modifyvm** *vmname* **--nic***x* **hostonly**. See Section 8.10, “VBoxManage modifyvm”.

For host-only networking, as with internal networking, you may find the DHCP server useful that is built into Oracle VM VirtualBox. This is enabled by default and manages the IP addresses in the host-only network. Without the DHCP server you would need to configure all IP addresses statically.

- In VirtualBox Manager you can configure the DHCP server by choosing **File**, **Tools**, **Network Manager**. The Network Manager window lists all host-only networks which are presently in use. Select the network name and then use the **DHCP Server** tab to configure DHCP server settings. See Section 6.11, “Network Manager”.
- Alternatively, you can use the **VBoxManage dhcpserver** command. See Section 8.50, “VBoxManage dhcpserver”.

> **Note**
>
> On Linux and macOS hosts the number of host-only interfaces is limited to 128. There is no such limit for Oracle Solaris and Windows hosts.

On Linux, macOS and Solaris Oracle VM VirtualBox will only allow IP addresses in 192.168.56.0/21 range to be assigned to host-only adapters. For IPv6 only link-local addresses are allowed. If other ranges are desired, they can be enabled by creating `/etc/vbox/networks.conf` and specifying allowed ranges there. For example, to allow 10.0.0.0/8 and 192.168.0.0/16 IPv4 ranges as well as 2001::/64 range put the following lines into `/etc/vbox/networks.conf`:

```
* 10.0.0.0/8 192.168.0.0/16
* 2001::/64
```

Lines starting with the hash **#** are ignored. The following example allows any addresses, effectively disabling range control:

```
* 0.0.0.0/0 ::/0
```

If the file exists, but no ranges are specified in it, no addresses will be assigned to host-only adapters. The following example effectively disables all ranges:

```
# No addresses are allowed for host-only adapters
```

6.8. UDP Tunnel Networking

This networking mode enables you to interconnect virtual machines running on different hosts.

Technically this is done by encapsulating Ethernet frames sent or received by the guest network card into UDP/IP datagrams, and sending them over any network available to the host.

UDP Tunnel mode has the following parameters:

- **Source UDP port:** The port on which the host listens. Datagrams arriving on this port from any source address will be forwarded to the receiving part of the guest network card.
- **Destination address:** IP address of the target host of the transmitted data.
- **Destination UDP port:** Port number to which the transmitted data is sent.

When interconnecting two virtual machines on two different hosts, their IP addresses must be swapped. On a single host, source and destination UDP ports must be swapped.

In the following example, host 1 uses the IP address 10.0.0.1 and host 2 uses IP address 10.0.0.2. To configure using the command-line:

```
VBoxManage modifyvm "VM 01 on host 1" --nic<x> generic
VBoxManage modifyvm "VM 01 on host 1" --nic-generic-drv<x> UDPTunnel
VBoxManage modifyvm "VM 01 on host 1" --nic-property<x> dest=10.0.0.2
VBoxManage modifyvm "VM 01 on host 1" --nic-property<x> sport=10001
VBoxManage modifyvm "VM 01 on host 1" --nic-property<x> dport=10002

VBoxManage modifyvm "VM 02 on host 2" --nic<y> generic
VBoxManage modifyvm "VM 02 on host 2" --nic-generic-drv<y> UDPTunnel
VBoxManage modifyvm "VM 02 on host 2" --nic-property<y> dest=10.0.0.1
VBoxManage modifyvm "VM 02 on host 2" --nic-property<y> sport=10002
VBoxManage modifyvm "VM 02 on host 2" --nic-property<y> dport=10001
```

Of course, you can always interconnect two virtual machines on the same host, by setting the destination address parameter to 127.0.0.1 on both. It will act similarly to an internal network in this case. However, the host can see the network traffic which it could not in the normal internal network case.

> **Note**
>
> On UNIX-based hosts, such as Linux, Oracle Solaris, and Mac OS X, it is not possible to bind to ports below 1024 from applications that are not run by `root`. As a result, if you try to configure such a source UDP port, the VM will refuse to start.

6.9. VDE Networking

Virtual Distributed Ethernet (VDE) is a flexible, virtual network infrastructure system, spanning across multiple hosts in a secure way. It enables L2/L3 switching, including spanning-tree protocol, VLANs, and WAN emulation. It is an optional part of Oracle VM VirtualBox which is only included in the source code.

VDE is a project developed by Renzo Davoli, Associate Professor at the University of Bologna, Italy.

The basic building blocks of the infrastructure are VDE switches, VDE plugs, and VDE wires which interconnect the switches.

The Oracle VM VirtualBox VDE driver has a single parameter: VDE network. This is the name of the VDE network switch socket to which the VM will be connected.

The following basic example shows how to connect a virtual machine to a VDE switch.

1. Create a VDE switch:

   ```
   vde_switch -s /tmp/switch1
   ```

2. Configure VMs using the command-line:

   ```
   VBoxManage modifyvm "VM name" --nic<x> generic

   VBoxManage modifyvm "VM name" --nic-generic-drv<x> VDE
   ```

   To connect to an automatically allocated switch port:

   ```
   VBoxManage modifyvm "VM name" --nic-property<x> network=/tmp/switch1
   ```

   To connect to a specific switch port *n*:

   ```
   VBoxManage modifyvm "VM name" --nic-property<x> network=/tmp/switch1[<n>]
   ```

   This command can be useful for VLANs.

3. (Optional) Map between a VDE switch port and a VLAN.

   Using the switch command line:

   ```
   vde$ vlan/create <VLAN>

   vde$ port/setvlan <port> <VLAN>
   ```

VDE is available on Linux and FreeBSD hosts only. It is only available if the VDE software and the VDE plugin library from the VirtualSquare project are installed on the host system.

> **Note**
>
> For Linux hosts, the shared library libvdeplug.so must be available in the search path for shared libraries.

For more information on setting up VDE networks, please see the documentation accompanying the software. See also http://wiki.virtualsquare.org.

6.10. Cloud Networks

Cloud networks can be used for connections from a local VM to a subnet on a remote Oracle Cloud Infrastructure instance. See Section 6.11.3, “Cloud Networks Tab” for details of how to create and configure a cloud network using the Network Manager tool in VirtualBox Manager.

To enable a cloud network interface for a virtual machine, do either of the following:

- Go to the **Network** page in the virtual machine's **Settings** dialog and select an **Adapter** tab. Ensure that the **Enable Network Adapter** check box is selected and choose **Cloud Network** for the **Attached To** field.
- On the command line, use **VBoxManage modifyvm** *vmname* **--nic***x* **cloud**. See Section 8.10, “VBoxManage modifyvm”.

6.11. Network Manager

The **Network Manager** tool in VirtualBox Manager enables you to create, delete, and configure the following types of networks used by Oracle VM VirtualBox:

- Host-only networks. See Section 6.11.1, “Host-Only Networks Tab”.
- NAT networks. See Section 6.11.2, “NAT Networks Tab”.
- Cloud networks. See Section 6.11.3, “Cloud Networks Tab”.

To display the Network Manager, go to the global **Tools** menu and click **Network**.

6.11.1. Host-Only Networks Tab

The Host-Only Networks tab in Network Manager lists all host-only networks that are currently in use.

- Click **Create** to add a new host-only network to the list.
- Click **Remove** to remove a host-only network from the list.
- Click **Properties** to show or hide settings for the selected host-only network.

To configure a host-only network, select the network name in the **Name** field and do the following:

- Use the **Adapter** tab to configure the network adapter for the host-only network.
- Use the **DHCP Server** tab to configure settings for the DHCP server used by the host-only network. The DHCP server is built into Oracle VM VirtualBox and manages IP addresses for the network automatically.

6.11.2. NAT Networks Tab

The NAT Networks tab in Network Manager lists all NAT networks that are currently in use.

- Click **Create** to add a new NAT network to the list.
- Click **Remove** to remove a NAT network from the list.
- Click **Properties** to show or hide settings for the selected NAT network.

To configure a NAT network, select the network name in the **Name** field and do the following:

- Use the **General Options** tab to configure the network settings used by the NAT network. For example, the network address and mask of the NAT service interface.

- Use the **Port Forwarding** tab to configure port forwarding rules used by the NAT network.

6.11.3. Cloud Networks Tab

The Cloud Networks tab in Network Manager lists all cloud networks that are currently in use.

- Click **Create** to add a new cloud network to the list.
- Click **Remove** to remove a cloud network from the list.
- Click **Properties** to show or hide settings for the selected cloud network.

To configure a cloud network, select the network name in the **Name** field and specify the following:

- **Name:** The name used for the cloud network.
- **Provider:** The cloud service provider, such as Oracle Cloud Infrastructure.
- **Profile:** The cloud profile used to connect to the cloud network.
- **ID:** The OCID for the cloud tunneling network. Click the **Network** icon to view the subnets on Oracle Cloud Infrastructure that are available for tunneling traffic.

  See Section 1.16.10, "Using a Cloud Network" for details of how you can use the **VBoxManage cloud** command to create and configure a virtual cloud network (VCN) on Oracle Cloud Infrastructure.

6.12. Limiting Bandwidth for Network Input/Output

Oracle VM VirtualBox supports limiting of the maximum bandwidth used for network transmission. Several network adapters of one VM may share limits through bandwidth groups. It is possible to have more than one such limit.

> **Note**
>
> Oracle VM VirtualBox shapes VM traffic only in the transmit direction, delaying the packets being sent by virtual machines. It does not limit the traffic being received by virtual machines.

Limits are configured through **VBoxManage**. The following example creates a bandwidth group named Limit, sets the limit to 20 Mbps and assigns the group to the first and second adapters of the VM:

```
VBoxManage bandwidthctl "VM name" add Limit --type network --limit 20m
VBoxManage modifyvm "VM name" --nicbandwidthgroup1 Limit
VBoxManage modifyvm "VM name" --nicbandwidthgroup2 Limit
```

All adapters in a group share the bandwidth limit, meaning that in the example above the bandwidth of both adapters combined can never exceed 20 Mbps. However, if one adapter does not require bandwidth the other can use the remaining bandwidth of its group.

The limits for each group can be changed while the VM is running, with changes being picked up immediately. The following example changes the limit for the group created in the previous example to 100 Kbps:

```
VBoxManage bandwidthctl "VM name" set Limit --limit 100k
```

To completely disable shaping for the first adapter of VM use the following command:

```
VBoxManage modifyvm "VM name" --nicbandwidthgroup1 none
```

It is also possible to disable shaping for all adapters assigned to a bandwidth group while VM is running, by specifying the zero limit for the group. For example, for the bandwidth group named Limit:

```
VBoxManage bandwidthctl "VM name" set Limit --limit 0
```

6.13. Improving Network Performance

Oracle VM VirtualBox provides a variety of virtual network adapters that can be attached to the host's network in a number of ways. Depending on which types of adapters and attachments are used the network performance will be different. Performance-wise the virtio network adapter is preferable over Intel PRO/1000 emulated adapters, which are preferred over the PCNet family of adapters. Both virtio and Intel PRO/1000 adapters enjoy the benefit of segmentation and checksum offloading. Segmentation offloading is essential for high performance as it allows for less context switches, dramatically increasing the sizes of packets that cross the VM/host boundary.

> **Note**
>
> Neither virtio nor Intel PRO/1000 drivers for Windows XP support segmentation offloading. Therefore Windows XP guests never reach the same transmission rates as other guest types. Refer to MS Knowledge base article 842264 for additional information.

Three attachment types: Internal, Bridged, and Host-Only, have nearly identical performance. The Internal type is a little bit faster and uses less CPU cycles as the packets never reach the host's network stack. The NAT attachment type is the slowest and most secure of all attachment types, as it provides network address translation. The generic driver attachment is special and cannot be considered as an alternative to other attachment types.

The number of CPUs assigned to VM does not improve network performance and in some cases may hurt it due to increased concurrency in the guest.

Here is a short summary of things to check in order to improve network performance:

- Whenever possible use the virtio network adapter. Otherwise, use one of the Intel PRO/1000 adapters.
- Use a Bridged attachment instead of NAT.
- Make sure segmentation offloading is enabled in the guest OS. Usually it will be enabled by default. You can check and modify offloading settings using the **ethtool** command on Linux guests.
- Perform a full detailed analysis of network traffic on the VM's network adaptor using a third party tool such as Wireshark. To do this, a promiscuous mode policy needs to be used on the VM's network adaptor. Use of this mode is only possible on the following network types: NAT Network, Bridged Adapter, Internal Network, and Host-Only Adapter.

  To setup a promiscuous mode policy, either select from the drop down list located in the **Network Settings** dialog for the network adaptor or use the command line tool **VBoxManage**. See Section 8.10, “VBoxManage modifyvm”.

  Promiscuous mode policies are as follows:

  - `deny`, which hides any traffic not intended for the VM's network adaptor. This is the default setting.
  - `allow-vms`, which hides all host traffic from the VM's network adaptor, but allows it to see traffic from and to other VMs.
  - `allow-all`, which removes all restrictions. The VM's network adaptor sees all traffic.