Understanding NAT

NAT模式中，虚拟机系统是连接到一个虚拟的网络中的。虚拟机系统被分配到的地址是这个虚拟网络的IP地址，而不占用宿主机网络的IP地址。

Network address translation — or NAT — is a networking option that first appeared in VMware Workstation 3.0.

NAT provides a simple way for virtual machines to use most client applications over almost any type of network connection available to the host. The only requirement is that the network connection must support TCP/IP.

NAT is useful when you have a limited supply of IP addresses or are connected to the network through a non-Ethernet network adapter. NAT works by translating addresses of virtual machines in a private VMnet network to that of the host machine. When a virtual machine sends a request to access a network resource, it appears to the network resource as if the request came from the host machine.

NAT uses the host's own network resources to connect to the external network. Thus, any TCP/IP network resource to which the host has access should be available through the NAT connection.

The chief advantage of NAT is that it provides a transparent, easy to configure way for virtual machines to gain access to network resources.

The NAT device is connected to the VMnet8 virtual switch. Virtual machines connected to the NAT network also use the VMnet8 virtual switch.

The NAT device waits for packets coming from virtual machines on the VMnet8 virtual network. When a packet arrives, the NAT device translates the address of the virtual machine to that of the host before forwarding the packet to the external network. When data arrives from the external network for the virtual machine on the private network, the NAT device receives the data, replaces the network address with that of the virtual machine and forwards the data to the virtual machine on the virtual network. This translation occurs automatically and requires minimal configuration on the guest and the host.

The host computer has a host virtual adapter on the NAT network (identical to the host virtual adapter on the host-only network). This adapter allows the host and the virtual machines to communicate with each other for such purposes as file sharing. The NAT never forwards traffic from the host virtual adapter.

In order to make networking configuration easy, a DHCP server is automatically installed when you install VMware Workstation. Virtual machines running on the network with the NAT device can dynamically obtain their IP addresses by sending out DHCP requests. The DHCP server on the NAT network, which is also used in host-only networking configurations, dynamically allocates IP addresses in the range of <net>.128 through <net>.254, where <net> is the network number assigned to your NAT network. VMware Workstation always uses a Class C address for NAT networks. IP addresses <net>.3 through <net>.127 can be used for static IP addresses. IP address <net>.1 is reserved for the host adapter; <net>.2 is reserved for the NAT device.

In addition to the IP address, the DHCP server on the NAT network also sends out additional configuration information that enables the virtual machine to operate automatically. This information includes the default gateway and the DNS server. In the DHCP response, the NAT device instructs the virtual machine to use the IP address <net>.2 as the default gateway and DNS server. This causes all IP packets destined for the external network and DNS requests to be forwarded to the NAT device.

The NAT device acts as a DNS server for the virtual machines on the NAT network. Actually, the NAT device is a DNS proxy and merely forwards DNS requests from the virtual machines to a DNS server that is known by the host. Responses come back to the NAT device, which then forwards them to the virtual machines.

If they get their configuration information from DHCP, the virtual machines on the NAT network automatically use the NAT device as the DNS server. However, the virtual machines can be statically configured to use another DNS server.

The virtual machines in the private NAT network are not, themselves, accessible via DNS. If you want the virtual machines running on the NAT network to access each other by DNS names, you must set up a private DNS server connected to the NAT network.

In general, any protocol using TCP or UDP can be used automatically by a virtual machine on the NAT network so long as the virtual machine initiates the network connection. This is true for most client applications such as Web browsing, Telnet, passive-mode FTP and downloading streaming video. Additional protocol support has been built into the NAT device to allow FTP and ICMP echo (ping) to work completely transparently through the NAT.

On the external network to which the host is connected, any virtual machine on the NAT network appears to be the host itself, because its network traffic uses the host's IP address. It is able to send and receive data using TCP/IP to any machine that is accessible from the host.

Before any such communication can occur, the NAT device must set up a mapping between the virtual machine's address on the private NAT network and the host's network address on the external network.

When a virtual machine initiates a network connection with another network resource, this mapping is created automatically. The operation is perfectly transparent to the user of the virtual machine on the NAT network. No additional work needs to be done to let the virtual machine access the external network.

The same cannot be said for network connections that are initiated from the external network to a virtual machine on the NAT network.

NAT模式中，虚拟机系统访问宿主网络，和宿主机网络中的主机访问虚拟机是不一样的。默认配置下，宿主机网络中的主机不能访问虚拟机。

When a machine on the external network attempts to initiate a connection with a virtual machine on the NAT network, it cannot reach the virtual machine because the NAT device does not forward the request. Network connections that are initiated from outside the NAT network are not transparent.

However, it is possible to configure port forwarding manually on the NAT device so network traffic destined for a certain port can still be forwarded automatically to a virtual machine on the NAT network. For details, see Advanced NAT Configuration below.

File sharing of the type used by Windows operating systems and Samba is possible among computers on the NAT network — including virtual machines and the host computer. If you are using WINS servers on your network, a virtual machine using NAT networking can access shared files and folders on the host that are known by the WINS server so long as those shared files and folders are in the same workgroup or domain.

Advanced NAT Configuration

Windows host: Configure the NAT device using the Virtual Network Editor (**Edit** > **Virtual Network Settings** > **NAT**).

You can stop and start the virtual NAT device by clicking the appropriate buttons.

To edit NAT settings for a virtual network, choose it from the drop-down menu, then click **Edit**.

Change any NAT settings you wish. Click the appropriate button to set up or change port forwarding or to specify DNS servers the virtual NAT device should use.

Linux host: Use the NAT configuration file on the host to configure the NAT device. This file is `/etc/vmware/vmnet8/nat/nat.conf`.

The configuration file is divided into sections. Each section configures a part of the NAT device. Text surrounded by square brackets — such as `[host]` — marks the beginning of a section. In each section is a configuration parameter that can be set. The configuration parameters take the form `ip = 192.168.27.1/24`.

For an example of a NAT configuration file, see Sample Linux vmnetnat.conf File. The configuration file variables are described below.

The [host] Section

`ip`
The IP address that the NAT device should use. It can optionally be followed by a slash and the number of bits in the subnet.

`netmask`
The subnet mask to use for the NAT. DHCP addresses are allocated from this range of addresses.

`configport`
A port that can be used to access status information about the NAT.

`device`
The VMnet device to use. Windows devices are of the form `VMnet<x>` where `<x>` is the number of the VMnet. Linux devices are of the form `/dev/vmnet<x>`.

`activeFTP`
Flag to indicate if active FTP is to be allowed. Active FTP allows incoming connections to be opened by the remote FTP server. Turning this off means that only passive mode FTP works. Set to `0` to turn it off.

The [udp] Section

`timeout`
Number of minutes to keep the UDP mapping for the NAT.

The [dns] Section

This section is for Windows hosts only. Linux does not use this section.

`policy`
Policy to use for DNS forwarding. Accepted values include `order`, `rotate`, and `burst`.

- `order` — send one DNS request at a time in order of the name servers
- `rotate` — send one DNS request at a time and rotate through the DNS servers
- `burst` — send to three servers and wait for the first one to respond

`timeout`
Time in seconds before retrying a DNS request.

`retries`
Number of retries before the NAT device gives up on a DNS request.

`autodetect`
Flag to indicate if the NAT should automatically detect the DNS servers available to the host.

`nameserver1`
IP address of a DNS server to use.

`nameserver2`
IP address of a DNS server to use.

`nameserver3`
IP address of a DNS server to use.

If autodetect is on and some name servers are specified, the DNS servers specified in `nameserver1`, `nameserver2` and `nameserver3` are added before the list of detected DNS servers.

The [netbios] Section

This section applies to Windows hosts only. Linux does not use this section.

`nbnsTimeout = 2`
Timeout for NBNS queries.

`nbnsRetries = 3`
Number of retries for each NBNS query.

`nbdsTimeout = 3`
Timeout for NBDS queries.

The [incomingtcp] Section

This section is used to configure TCP port forwarding for NAT. In this section, you can assign a port number to an IP address and port number on a virtual machine.

The following line shows the format used in this section.

```
8887 = 192.168.27.128:21
```

This example creates a mapping from port 8887 on the host to the IP address 192.168.27.128 and port 21. When this mapping is set and an external machine connects to the host at port 8887, the network packets are automatically forwarded to port 21 (the standard port for FTP) on the virtual machine with IP address 192.168.27.128.

The [incomingudp] Section

This section is used to configure UDP port forwarding for NAT. In this section, you can assign a port number to an IP address and port number on a virtual machine.

The following line shows the format used in this section. It illustrates a way to forward X server traffic from the host port 6000 to the virtual machine's port 6001.

```
6000 = 192.168.27.128:6001
```

This example creates a mapping from port 6000 on the host to the IP address 192.168.27.128 and port 6001. When this mapping is set and an external machine connects to the host at port 6000, the network packets are automatically forwarded to port 6001 on the virtual machine with IP address 192.168.27.128.

Custom NAT and DHCP Configuration on a Windows Host

If you are an advanced user on a Windows host computer, you may wish to make custom configuration settings by editing the NAT and DHCP configuration files. If your host operating system is installed on the C drive, the configuration files for NAT and DHCP are in the following locations:

- **NAT:** `C:\Documents and Settings\All Users\Application Data\VMware\vmnetnat.conf`
- **DHCP:** `C:\Documents and Settings\All Users\Application Data\VMware\vmnetdhcp.conf`

Note: In VMware Workstation 4, you can change many key NAT and DCHP settings using the Virtual Network Editor (**Edit** > **Virtual Network Settings**). However, if you have made manual changes to the configuration files, some or all of those changes may be lost when you use the Virtual Network Editor. If you have made manual changes, you should make backup copies of the files before changing any settings in the Virtual Network Editor. After making changes in the Virtual Network Editor, you can copy your manual changes back into the appropriate configuration files.

Specifying Connections from Ports Below 1024

When a client machine makes a TCP or UDP connection to a server, the connection comes from a particular port on the client (the source port) and connects to a particular port on the server (the destination port). For security reasons, some servers accept connections only from source ports below 1024. You may see this configuration on machines used as NFS file servers, for example.

If a virtual machine using NAT attempts to connect to a server that requires the client to use a source port below 1024, it is important that the NAT device forward the request from a port below 1024. Beginning in VMware Workstation 4.5, you can specify this behavior in the `vmnetnat.conf` file.

This behavior is controlled by entries in sections headed `[privilegedUDP]` and `{privilegedTCP]`. You may need to add settings to or modify settings in either or both of these sections, depending on the kind of connection you need to make.

You can set two parameters, each of which appears on a separate line.

`autodetect = <n>`
The autodetect setting determines whether the VMware NAT device automatically attempts to map virtual machine source ports below 1024 to NAT source ports below 1024. A setting of `1` means true. A setting of `0` means false. On a Windows host, the default is `1` (true). On a Linux host, the default is `0` (false).

`port = <n>`
The port setting specifies a destination port (where `<n>` is the port on the server that accepts the connection from the client). Whenever a virtual machine connects to the specified port on any server, the NAT device attempts to make the connection from a source port below 1024. You may include one or more port settings in the `[privilegedUDP]` or `[privilegedTCP]` section or in both sections, as required for the connections you need to make. Enter each port setting on a separate line.

Considerations for Using NAT

Because NAT requires that every packet sent and received from virtual machines is in the NAT network, there is an unavoidable performance penalty. Our experiments show that the penalty is minor for dial-up and DSL connections and performance is adequate for most VMware Workstation uses.

NAT is not perfectly transparent. It does not normally allow connections to be initiated from outside the network, although you can set up server connections by manually configuring the NAT device. The practical result is that some TCP and UDP protocols that require a connection be initiated from the server machine — some peer to peer applications, for example — do not work automatically, and some may not work at all.

A standard NAT configuration provides basic-level firewall protection because the NAT device can initiate connections from the private NAT network, but devices on the external network cannot normally initiate connections to the private NAT network.

Using NAT with NetLogon

When using NAT networking in a virtual machine with a Windows guest operating system running on a Windows host, you can use NetLogon to log on to a Windows domain from the virtual machine. You can then access file shares known by the WINS server in the domain.

To use NetLogon, you need to know how WINS servers and Windows domain controllers work. This section explains how to set up the virtual machine to use NetLogon. The setup process is similar to the way you set up a physical computer on one LAN that is using a domain controller on another LAN.

In order to log on to a Windows domain outside the virtual NAT network, the virtual machine needs access to a WINS server for that domain. There are two ways you can connect the virtual machine to a WINS server. You can connect to the WINS server provided by the DHCP server used on the NAT network, provided that the WINS server is already set up on the host. If you want to connect from the virtual machine to a WINS server not set up on the host, you can manually enter the IP address of the WINS server.

Using NAT to Connect to an Existing WINS Server Already Set Up on the Host

In order to use this method, a WINS server in the same workgroup or domain must be set up on the host. These steps use Windows 2000, Windows XP or Windows Server 2003 as a guide. The process is similar for Windows NT, Windows Me and Windows 9x guests.

1. In the virtual machine, right-click on **My Network Places** and select **Properties**.
2. In the Network Connections window, right-click the virtual network adapter and select **Properties**.
3. In the Properties dialog box, select **Internet Protocol (TCP/IP)**, then click **Properties**.
4. In the TCP/IP Properties dialog box, click **Advanced**.
5. Click the **WINS** tab, then under **NetBIOS setting**, select **Use NetBIOS setting from DHCP Server**.
6. Click **OK** twice, then click **Close**.

Manually Entering the IP Address of a WINS Server

Use this method to connect to a WINS server in the same workgroup or domain that is not already set up on the host.

1. In the virtual machine, right-click on **My Network Places** and select **Properties**.
2. In the Network Connections window, right-click the virtual network adapter and select **Properties**.
3. In the Properties dialog box, select **Internet Protocol (TCP/IP)**, then click **Properties**.
4. In the TCP/IP Properties dialog box, click **Advanced**.
5. Click the **WINS** tab, then click **Add**.
6. In the TCP/IP WINS Server dialog box, enter the IP address for the WINS server in the **WINS server** field, then click **OK**. The IP address of the WINS server appears in the **WINS addresses** list on the **WINS** tab.

   Repeat steps 5 and 6 for each WINS server to which you want to connect from this virtual machine.

7. Click **OK** twice, then click **Close**.

Now that the virtual machine has an IP address for a WINS server, you use NetLogon in the virtual machine to log on to a domain and access shares in that domain.

For example, if the WINS server covers a domain with a domain controller it is possible to access that domain controller from the virtual machine and add the virtual machine to the domain. You need to know the user ID and password of the Administrator on the domain controller.

Note: Your access is limited to shares of virtual machines that are on the same NAT network or are bridged on the same domain.

Sample Linux vmnetnat.conf File

```
# Linux NAT configuration file
[host]
# NAT gateway address
ip = 192.168.237.2/24
hostMAC = 00:50:56:C0:00:08
# enable configuration; disabled by default for security reasons
#configport = 33445
# VMnet device if not specified on command line
device = VMnet8
# Allow PORT/EPRT FTP commands (they need incoming TCP stream...)
activeFTP = 1
# Allows the source to have any OUI. Turn this one if you change the OUI
# in the MAC address of your virtual machines.
#allowAnyOUI = 1
[udp]
# Timeout in seconds, 0 = no timeout, default = 60; real value might
# be up to 100% longer
timeout = 30
[dns]
# This section applies only to Windows.
#
# Policy to use for DNS forwarding. Accepted values include order,
# rotate, burst.
#
# order: send one DNS request at a time in order of the name servers
# rotate: send one DNS request at a time, rotate through the DNS servers
# burst: send to three servers and wait for the first one to respond
policy = order;
# Timeout in seconds before retrying DNS request.
timeout = 2
# Retries before giving up on DNS request
retries = 3
# Automatically detect the DNS servers (not supported in Windows NT)
autodetect = 1
```

```
# List of DNS servers to use. Up to three may be specified

#nameserver1 = 208.23.14.2

#nameserver2 = 63.93.12.3

#nameserver3 = 208.23.14.4

[netbios]

# This section applies only to Windows.

# Timeout for NBNS queries.

nbnsTimeout = 2

# Number of retries for each NBNS query.

nbnsRetries = 3

# Timeout for NBDS queries.

nbdsTimeout = 3

[incomingtcp]

# Use these with care - anyone can enter into your virtual machine through

# these...

# FTP (both active and passive FTP is always enabled)

# ftp localhost 8887

#8887 = 192.168.27.128:21

# WEB (make sure that if you are using named webhosting, names point to

# your host, not to guest... And if you are forwarding port other

# than 80 make sure that your server copes with mismatched port

# number in Host: header)

# lynx http://localhost:8888

#8888 = 192.168.27.128:80

# SSH

# ssh -p 8889 root@localhost

#8889 = 192.168.27.128:22

[incomingudp]

# UDP port forwarding example

#6000 = 192.168.27.128:6001
```