



网络应用及管理

信息工程学院

IP 地址

- IP分为 *.*.*.*
网络号.主机号
- 物理网线没问题的情况下，同网络下的不同**IP**就能互通，即同网络号下的不同主机号间的主机间能互通。
- 公网是要申请付费的，公网地址是唯一的。
- 私网是免费使用的，地址可以重复的。

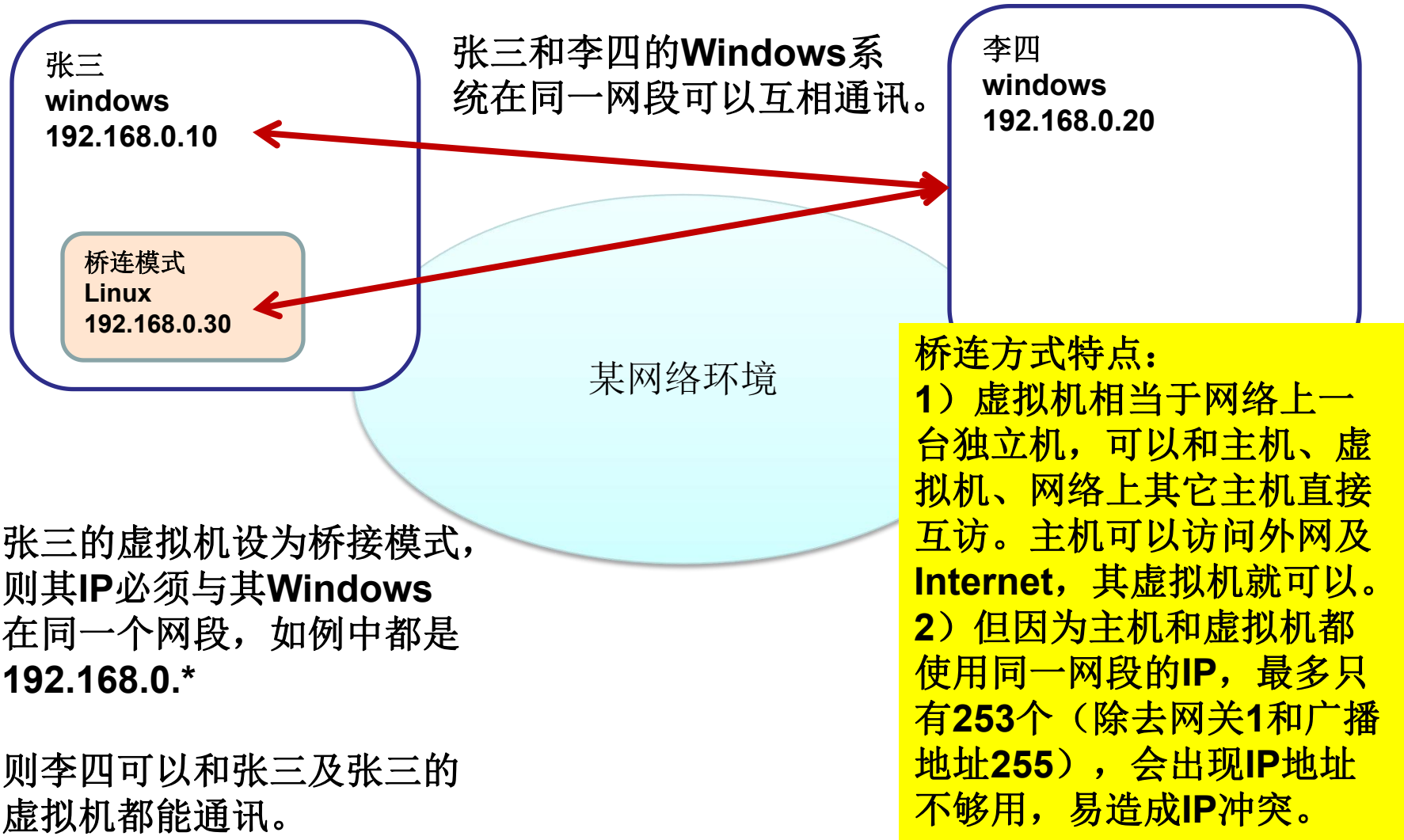
常用私网**IP**为： 10.*.*.*

192.168.*.*

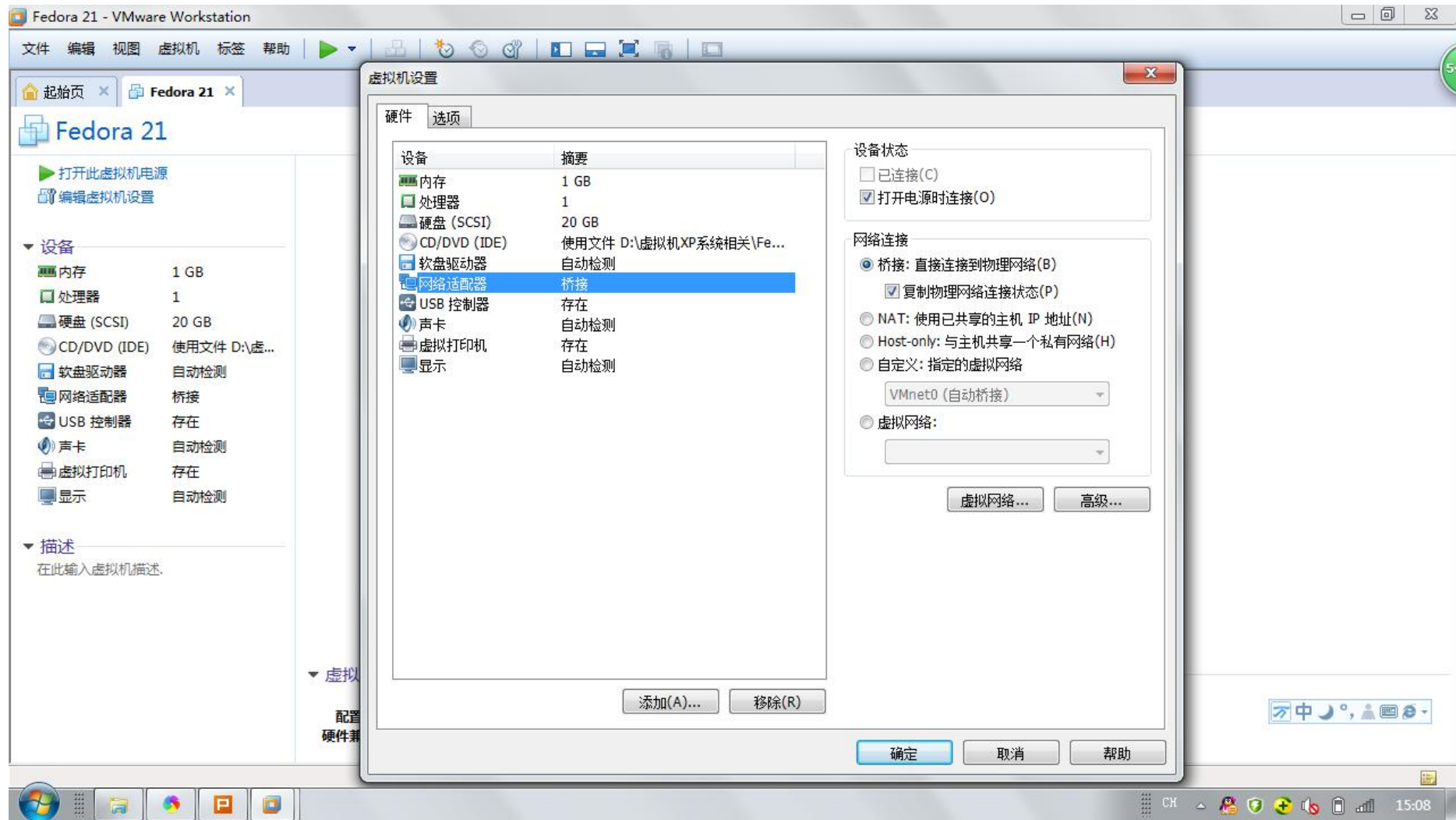
172.16.*.*~172.31.*.*

VMWare的三种网络连接模式

- **bridged(桥接模式): 默认使用VMnet0**



配置虚拟机设置中的网络适配器-选择桥接 勾选了复制物理连接



勾选复制物理连接即可访问Internet。

如果用主机的无线网卡桥接

WLAN

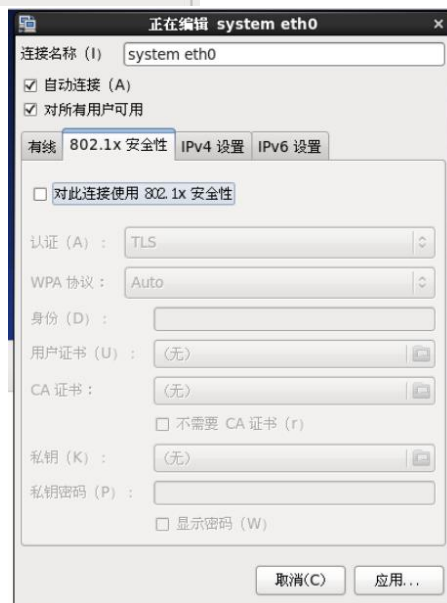
SSID: Xiaomi_00D5
协议: Wi-Fi 4 (802.11n)
安全类型: WPA2-个人
网络频带: 2.4 GHz
网络通道: 6
本地链接 IPv6 地址: fe80::2152:25e2:
IPv4 地址: 192.168.31.210
IPv4 DNS 服务器: 192.168.31.1
制造商: Qualcomm Con
描述: Qualcomm QC/
驱动程序版本: 12.0.0.916
物理地址(MAC): E8-6F-38-47-66



不要勾选复制

WINDOWS中无线网卡的信息如图

在Linux中要将网络连接的添加有线的system eth0，
将其配置为与主机无线网卡在同一网段



主机的无线网卡是：192.168.31.210
eth0则设为：192.168.31.211
DNS服务器同主机的。

```
root@centos6:/etc/sysconfig/network-scripts
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
[root@centos6 network-scripts]# pwd
/etc/sysconfig/network-scripts
[root@centos6 network-scripts]# cat ifcfg-system_eth0
TYPE=Ethernet
BOOTPROTO=none
IPADDR=192.168.31.211
PREFIX=24
GATEWAY=192.168.31.1
DNS1=192.168.31.1
DEFROUTE=yes
IPV4_FAILURE_FATAL=yes
IPV6INIT=no
NAME="system eth0"
UUID=31358344-702b-4b50-992a-524cd3d57a34
ONBOOT=yes
[root@centos6 network-scripts]#

[root@centos6 ~]# ping baidu.com
PING baidu.com (39.156.69.79) 56(84) bytes of data:
64 bytes from 39.156.69.79: icmp_seq=1 ttl=50 time=39.3 ms
64 bytes from 39.156.69.79: icmp_seq=2 ttl=50 time=40.6 ms
64 bytes from 39.156.69.79: icmp_seq=3 ttl=50 time=38.7 ms
64 bytes from 39.156.69.79: icmp_seq=4 ttl=50 time=37.3 ms
^C
--- baidu.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3236ms
rtt min/avg/max/mdev = 37.327/39.006/40.652/1.201 ms
[root@centos6 ~]#
```

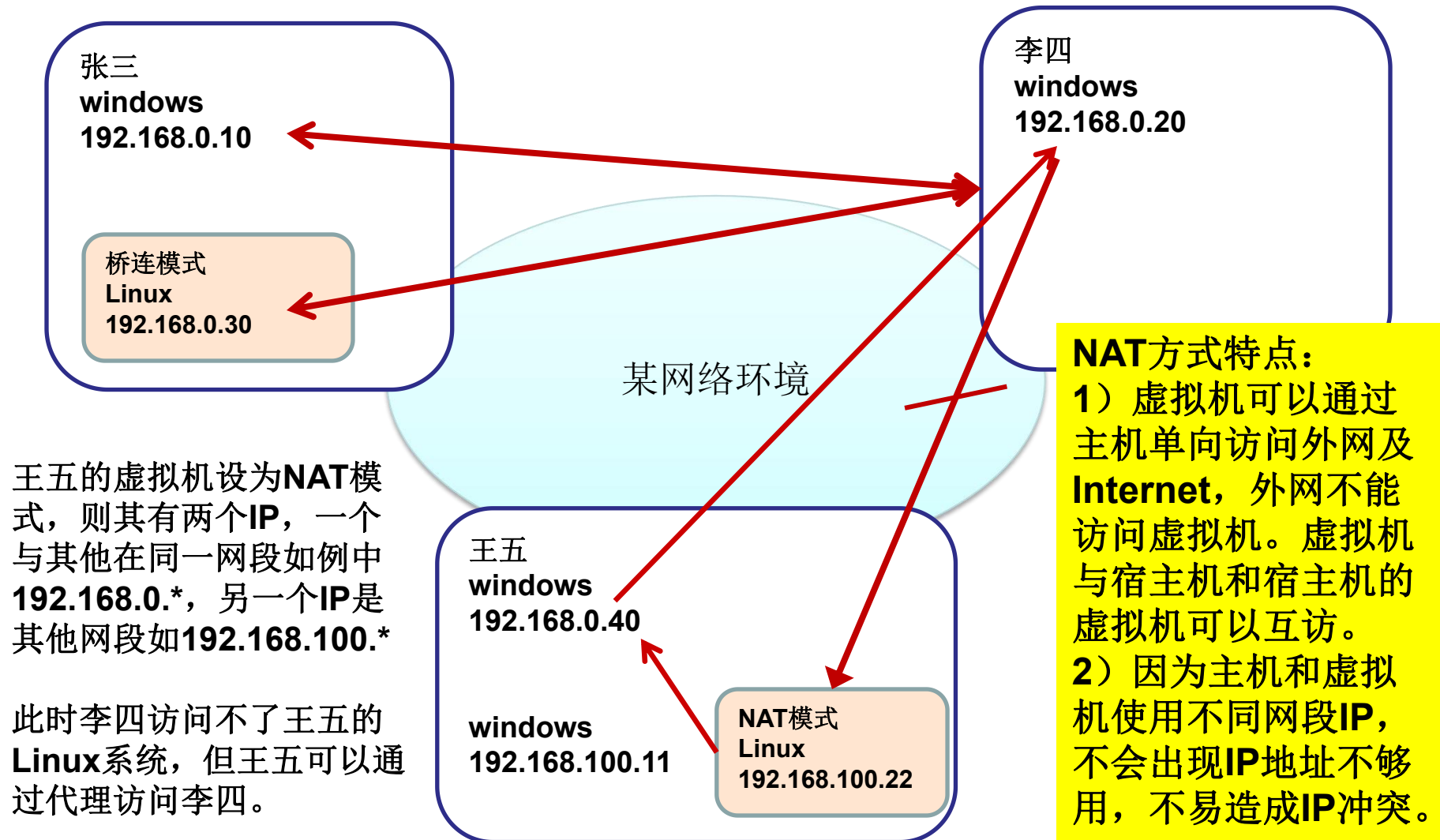
```
root@centos6:~
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
[root@centos6 ~]# ifconfig
eth0      Link encap: Ethernet  HWaddr 00:0C:29:76:7F:D2
          inet addr: 192.168.31.211  Bcast: 192.168.31.255  Mask: 255.255.255.0
          inet6 addr: fe80::20c:29ff:fe76:7fd2/64 Scope: Link
          UP BROADCAST RUNNING MULTICAST  MTU: 1500  Metric: 1
          RX packets: 2866415 errors: 0 dropped: 0 overruns: 0 frame: 0
          TX packets: 220 errors: 0 dropped: 0 overruns: 0 carrier: 0
          collisions: 0 txqueuelen: 1000
          RX bytes: 917121942 (874.6 MiB)  TX bytes: 34021 (33.2 KiB)

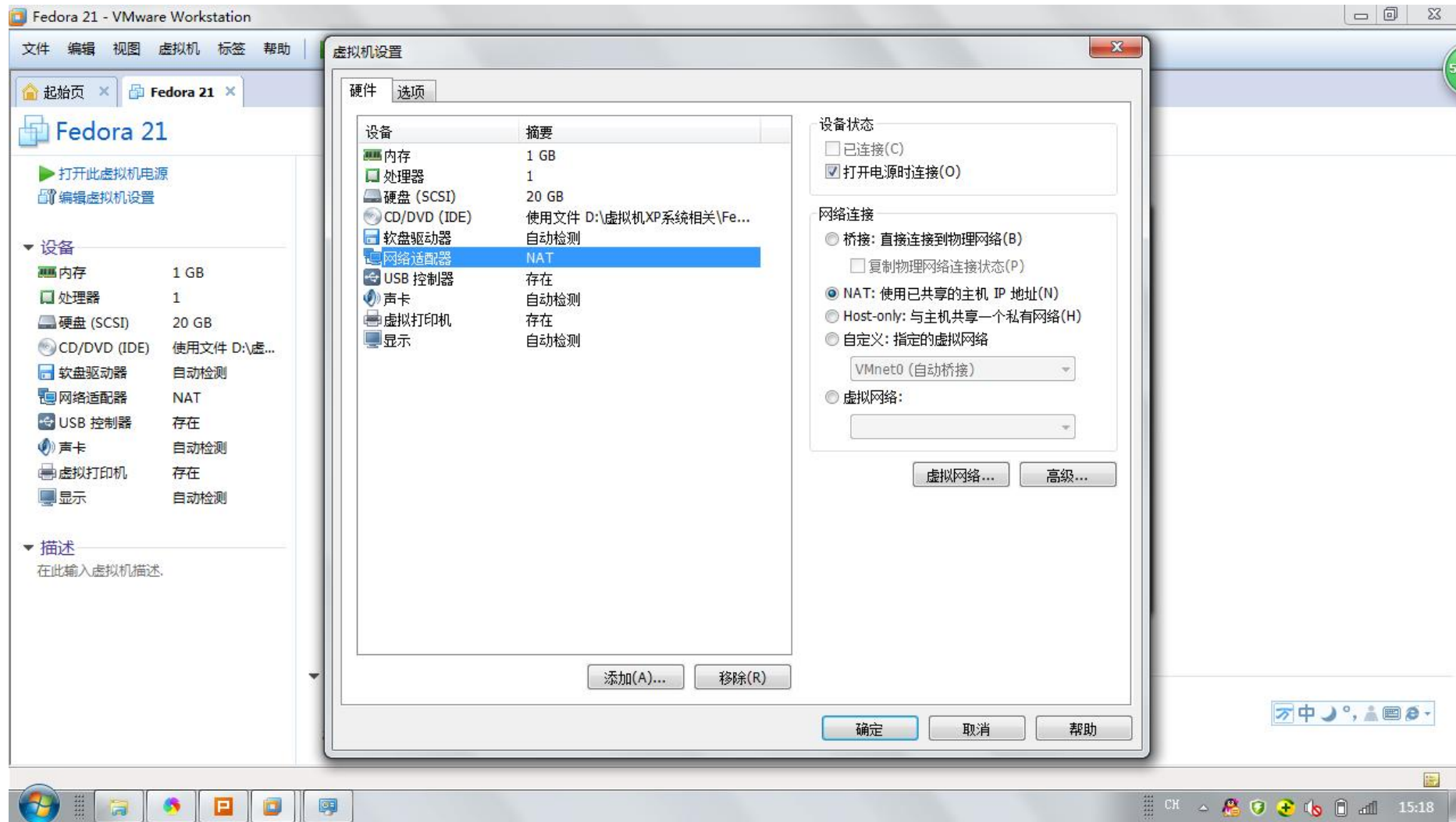
lo        Link encap: Local Loopback
          inet addr: 127.0.0.1  Mask: 255.0.0.0
          inet6 addr: ::1/128 Scope: Host
          UP LOOPBACK RUNNING  MTU: 65536  Metric: 1
          RX packets: 10 errors: 0 dropped: 0 overruns: 0 frame: 0
          TX packets: 10 errors: 0 dropped: 0 overruns: 0 carrier: 0
          collisions: 0 txqueuelen: 0
          RX bytes: 584 (584.0 b)  TX bytes: 584 (584.0 b)

[root@centos6 ~]#
```

VMWare的三种网络连接模式

- **NAT(网络地址转换模式): 默认使用VMnet8**

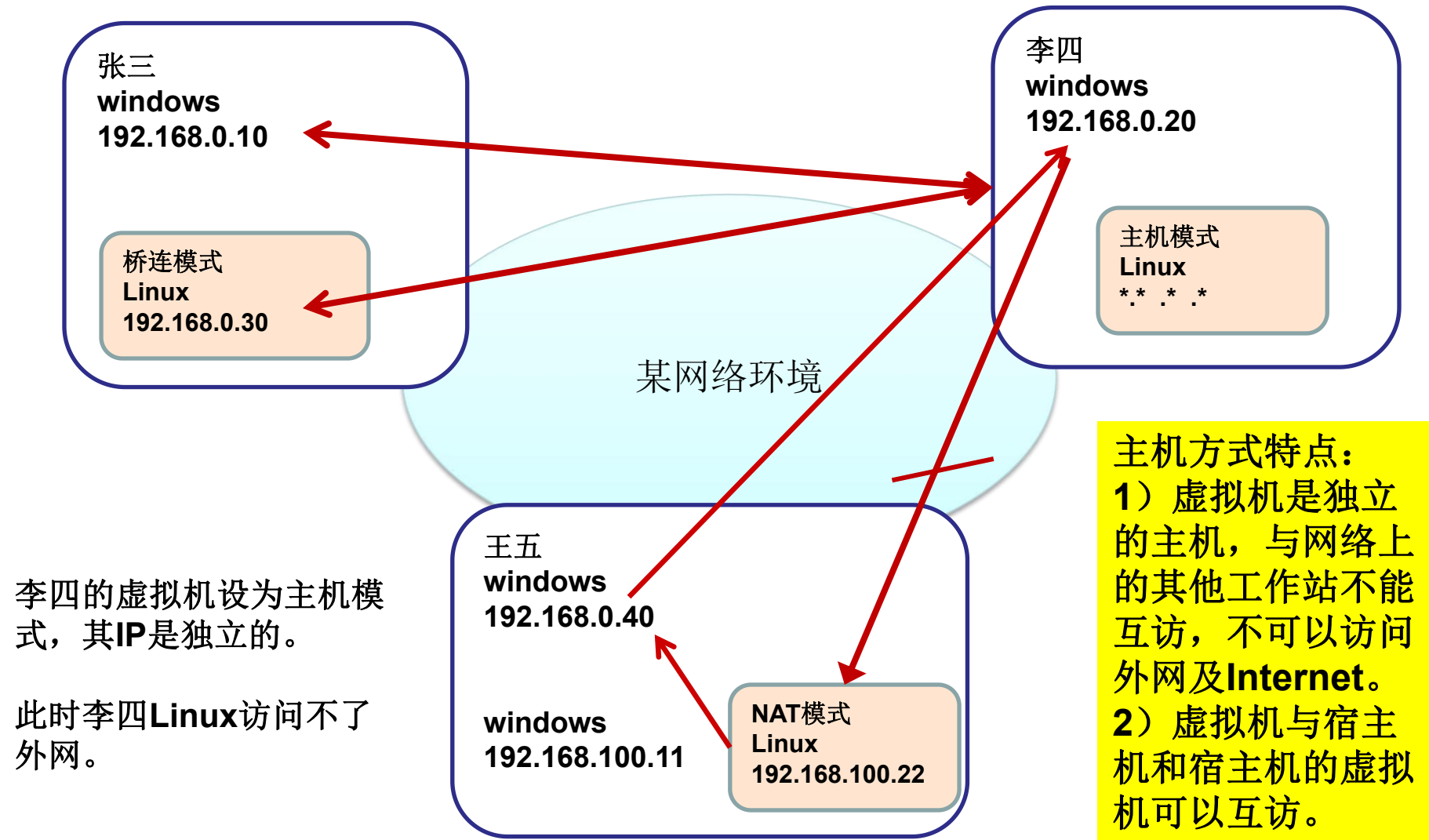




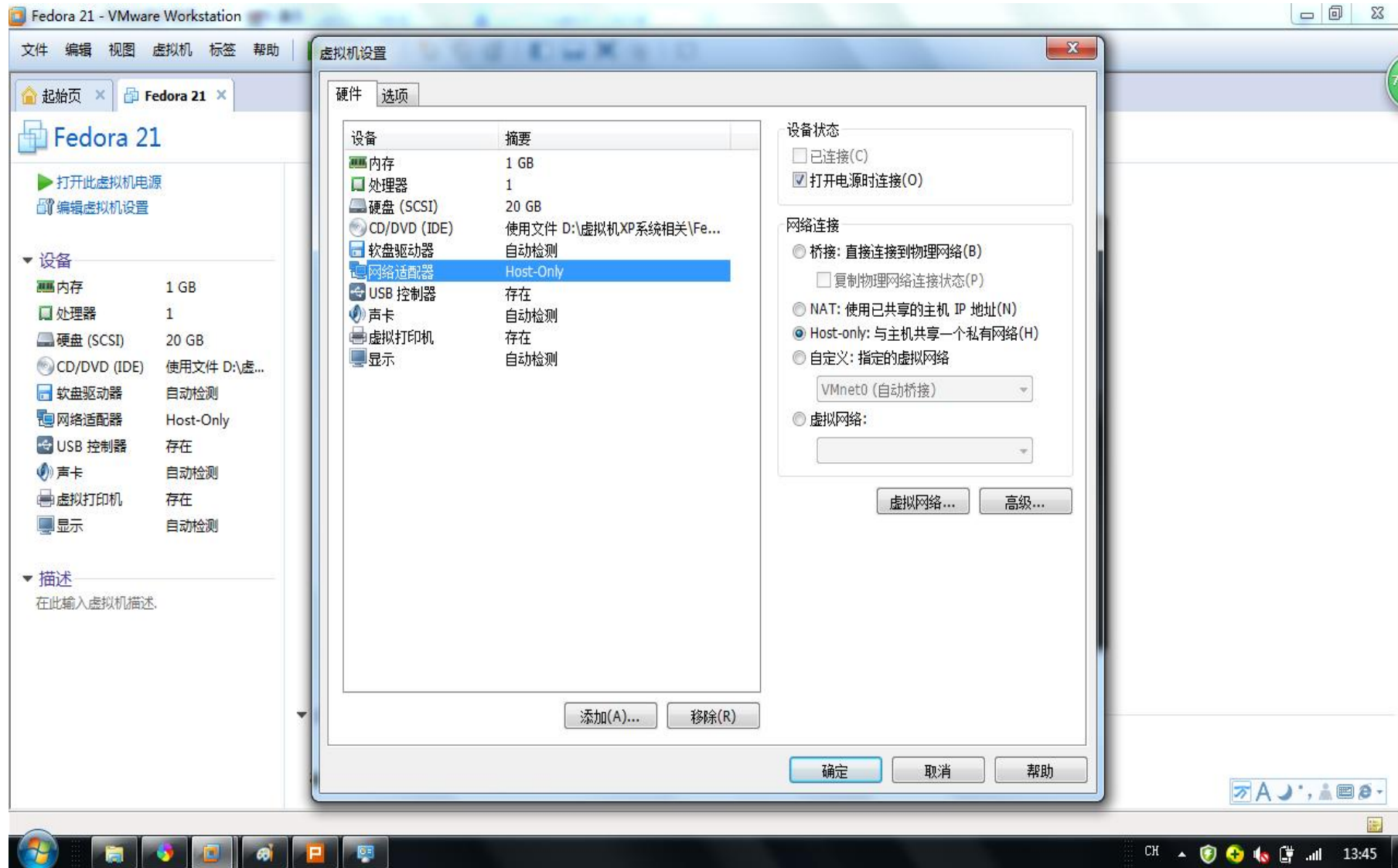
NAT方式在虚拟系统中不用进行任何手工配置就能直接访问Internet，较常用。

VMWare的三种网络连接模式

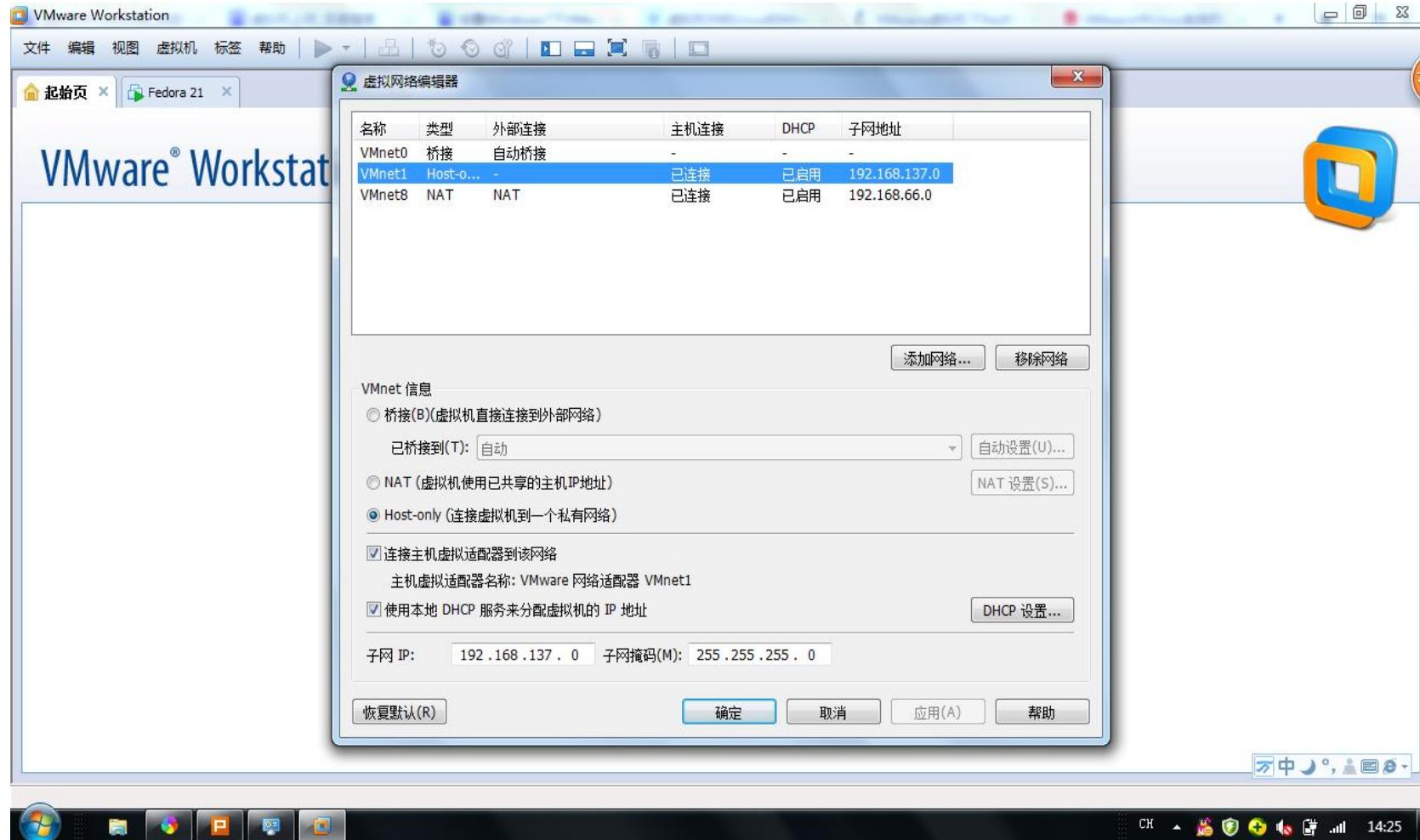
- **host-only(主机模式): 默认使用VMnet1**



配置虚拟机设置中的网络适配器--勾选Host-only

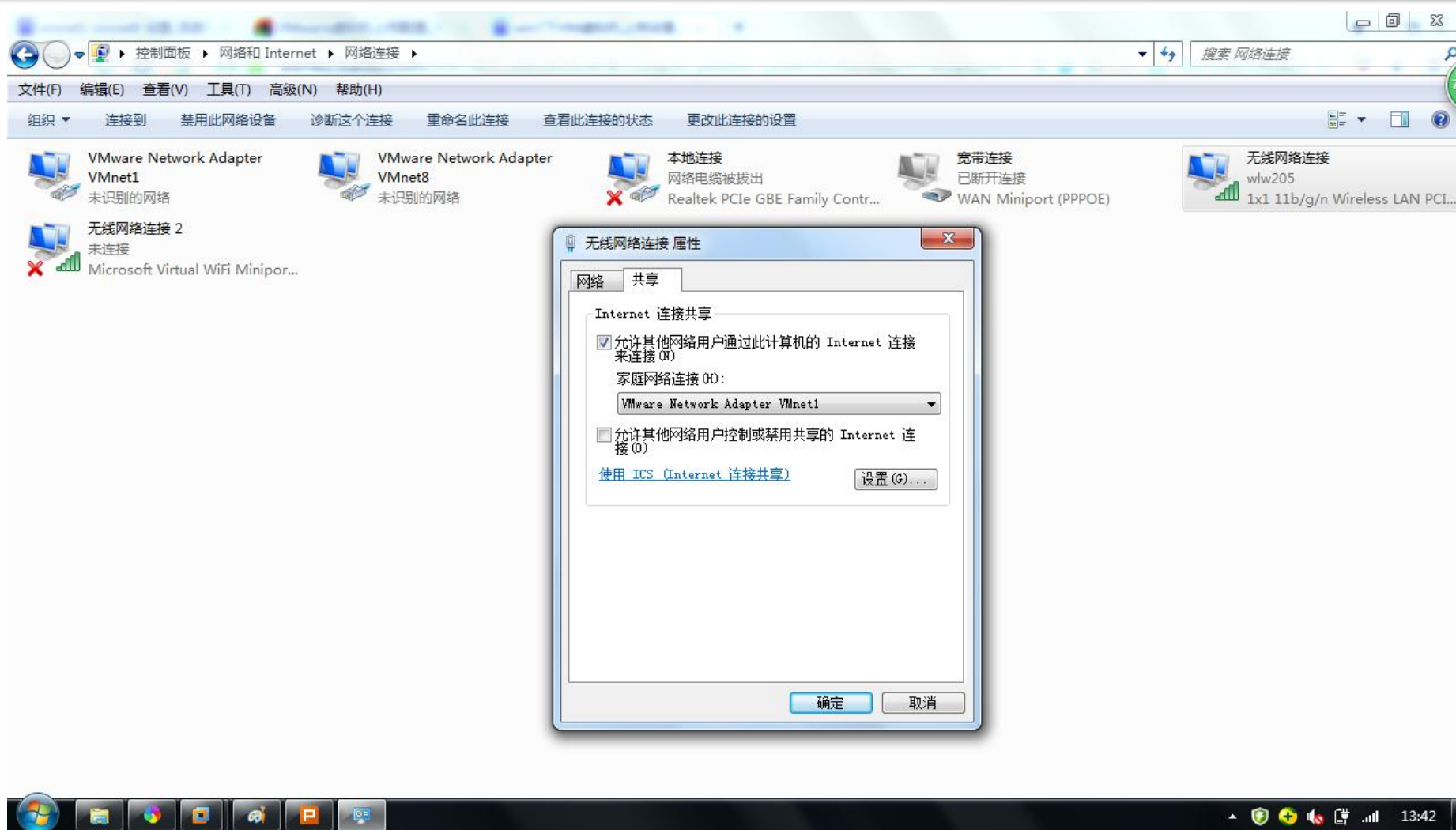


查看虚拟网络编辑器设置



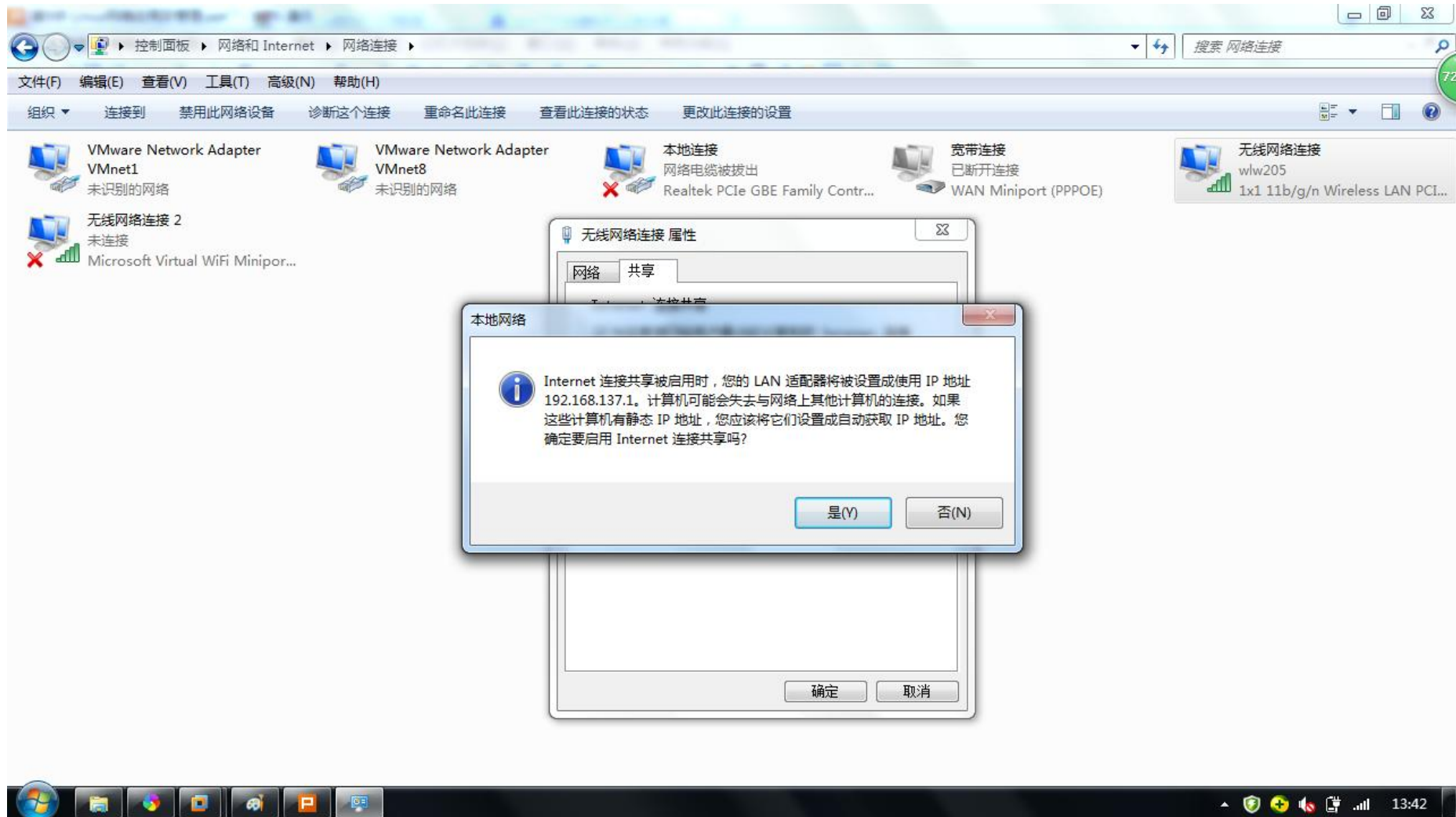
如果想利用VMWare创建一个与网内其他机器相隔离的虚拟系统，进行某些特殊的网络调试工作，可以选择host-only模式。

设置网络共享连接

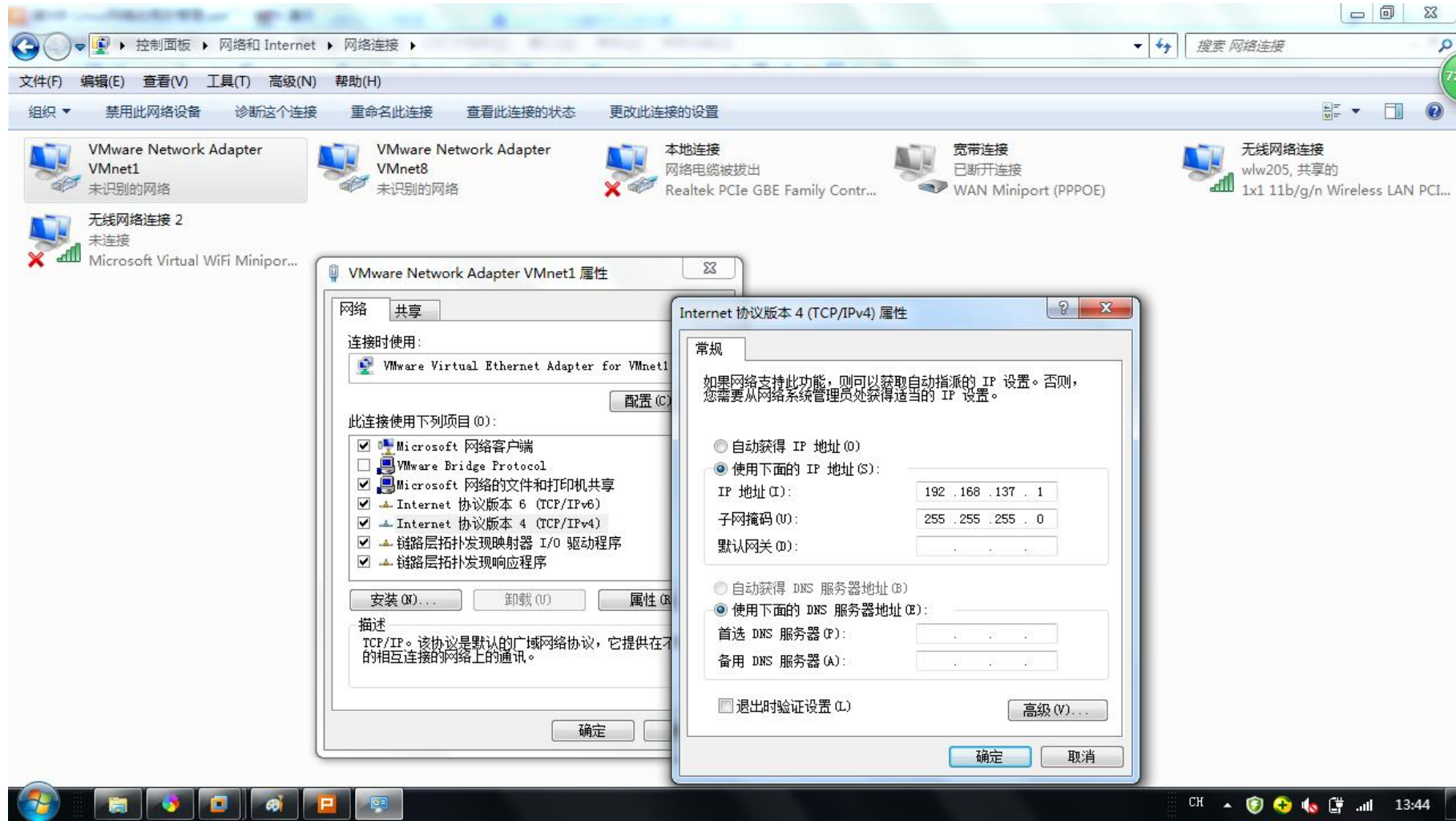


主机模式要做共享连接才能访问Internet。

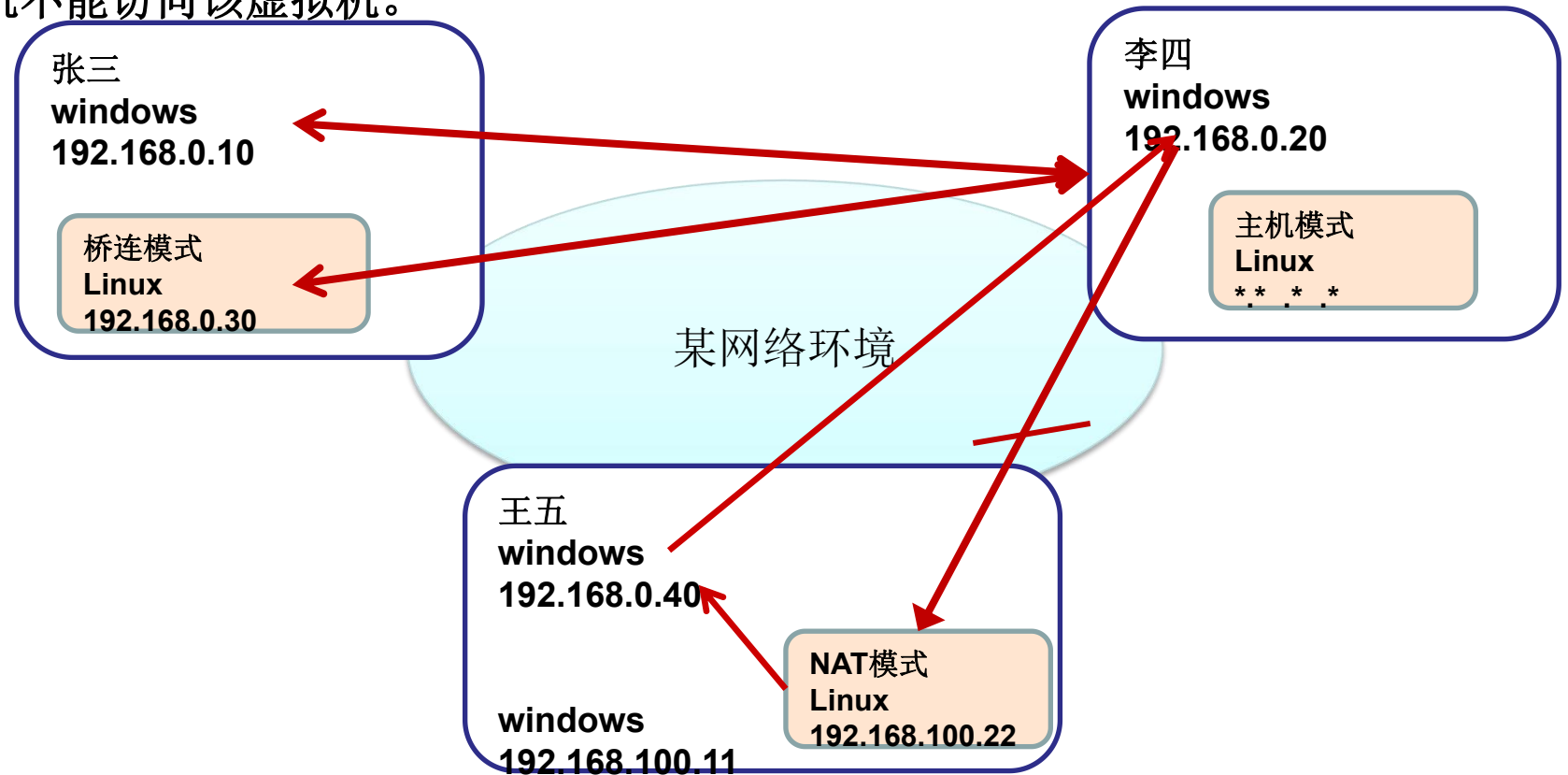
确认共享连接



查看VMnet1的IP



- bridged**模式下**VMnet0**虚拟网络：1)虚拟机做复制配置即可同主机一样访问Internet；2)主机与该虚拟机、网络中其他主机、虚拟机都可互访。
- NAT**模式下**VMnet8**虚拟网络：1)虚拟机不做手动配置就可访问Internet；2)主机和该虚拟机可互访，但网络上其他主机不能访问该虚拟机。
- host-only**模式下**VMnet1**虚拟网络：1)虚拟机要手动配置网络共享才可访问Internet；2)主机和该虚拟机能互访，但网络上其他主机不能访问该虚拟机。



常用的网络服务命令

◆ ifconfig

- # ifconfig [接口名] //显示网络接口的配置信息
- # ifconfig <接口名> <up/down> //激活/禁用某个网络接口
- # ifconfig <接口名> ip地址 netmask 子网掩码 //配置网络接口IP地址

```
wlw@CM00:~  
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)  
[wlw@CM00 ~]$ ifconfig  
eth0      Link encap: Ethernet  HWaddr 00:0C:29:76:7F:D2  
          inet addr: 192.168.26.3  Bcast: 192.168.26.255  Mask: 255.255.255.0  
          inet6 addr: fe80::20c:29ff:fe76:7fd2/64 Scope: Link  
          UP BROADCAST RUNNING MULTICAST  MTU: 1500  Metric: 1  
          RX packets: 247  errors: 0  dropped: 0  overruns: 0  frame: 0  
          TX packets: 44  errors: 0  dropped: 0  overruns: 0  carrier: 0  
          collisions: 0  txqueuelen: 1000  
          RX bytes: 17637 (17.2 KiB)  TX bytes: 3108 (3.0 KiB)  
  
lo        Link encap: Local Loopback  
          inet addr: 127.0.0.1  Mask: 255.0.0.0  
          inet6 addr: ::1/128 Scope: Host  
          UP LOOPBACK RUNNING  MTU: 65536  Metric: 1  
          RX packets: 8  errors: 0  dropped: 0  overruns: 0  frame: 0  
          TX packets: 8  errors: 0  dropped: 0  overruns: 0  carrier: 0  
          collisions: 0  txqueuelen: 0  
          RX bytes: 480 (480.0 b)  TX bytes: 480 (480.0 b)
```

常用的网络服务命令

◆ ping

- ping IP地址/主机名 //检测IP连通性
- ping -c n IP地址/主机名 //指定得到n个应答后中断操作

```
[wlv@CM00 ~]$ ping www.baidu.com
PING www.a.shifen.com (14.215.177.38) 56(84) bytes of data.
64 bytes from 14.215.177.38: icmp_seq=1 ttl=128 time=26.4 ms
64 bytes from 14.215.177.38: icmp_seq=2 ttl=128 time=37.8 ms
64 bytes from 14.215.177.38: icmp_seq=3 ttl=128 time=25.1 ms
64 bytes from 14.215.177.38: icmp_seq=4 ttl=128 time=26.8 ms
64 bytes from 14.215.177.38: icmp_seq=5 ttl=128 time=52.3 ms
^C
--- www.a.shifen.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4781ms
rtt min/avg/max/mdev = 25.125/33.742/52.378/10.376 ms
```

```
[wlv@CM00 ~]$ ping 192.168.26.2
PING 192.168.26.2 (192.168.26.2) 56(84) bytes of data.
64 bytes from 192.168.26.2: icmp_seq=1 ttl=128 time=0.428 ms
64 bytes from 192.168.26.2: icmp_seq=2 ttl=128 time=0.804 ms
64 bytes from 192.168.26.2: icmp_seq=3 ttl=128 time=0.299 ms
^C
--- 192.168.26.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2786ms
rtt min/avg/max/mdev = 0.299/0.510/0.804/0.215 ms
[wlv@CM00 ~]$ ping 192.168.26.4
PING 192.168.26.4 (192.168.26.4) 56(84) bytes of data.
From 192.168.26.3 icmp_seq=2 Destination Host Unreachable
From 192.168.26.3 icmp_seq=3 Destination Host Unreachable
From 192.168.26.3 icmp_seq=4 Destination Host Unreachable
^C
--- 192.168.26.4 ping statistics ---
5 packets transmitted, 0 received, +3 errors, 100% packet loss, time 4769ms
pipe 3
```

常用的网络服务命令

◆ route

- `#route` **//显示路由表**
- `#route add -net 网络地址 netmask 子网掩码 dev 网卡设备名` **//添加路由**
- `#route del -net 网络地址 netmask 子网掩码` **//删除路由**
- `#route add default gw 网关IP地址 dev 网卡设备名` **//添加默认网关**
- `#route del default gw 网关IP地址 dev 网卡设备名` **//删除默认网关**

例:

- `#route`
- `#route add -net 192.168.0.0 netmask 255.255.255.0 dev eth0`
- `#route del -net 192.168.0.0 netmask 255.255.255.0`
- `#route add default gw 192.168.0.1 dev eth0`

例：显示当前系统的路由表信息

```
[wlw@CM00 ~]$ route
Kernel IP routing table
Destination      Gateway          Genmask         Flags Metric Ref    Use Iface
192.168.26.0     *                255.255.255.0   U        1      0      0 eth0
default          localhost        0.0.0.0          UG       0      0      0 eth0
```

NAT模式（网络环境）

老师

Linux
用ifconfig查看
eth0的IP为：
192.168.26.3

网络

Windows
用ipconfig查看
vmnet8虚拟网卡IP，
为：192.168.26.1

真实无线网卡IP，
为：192.168.3.7

Internet

网关

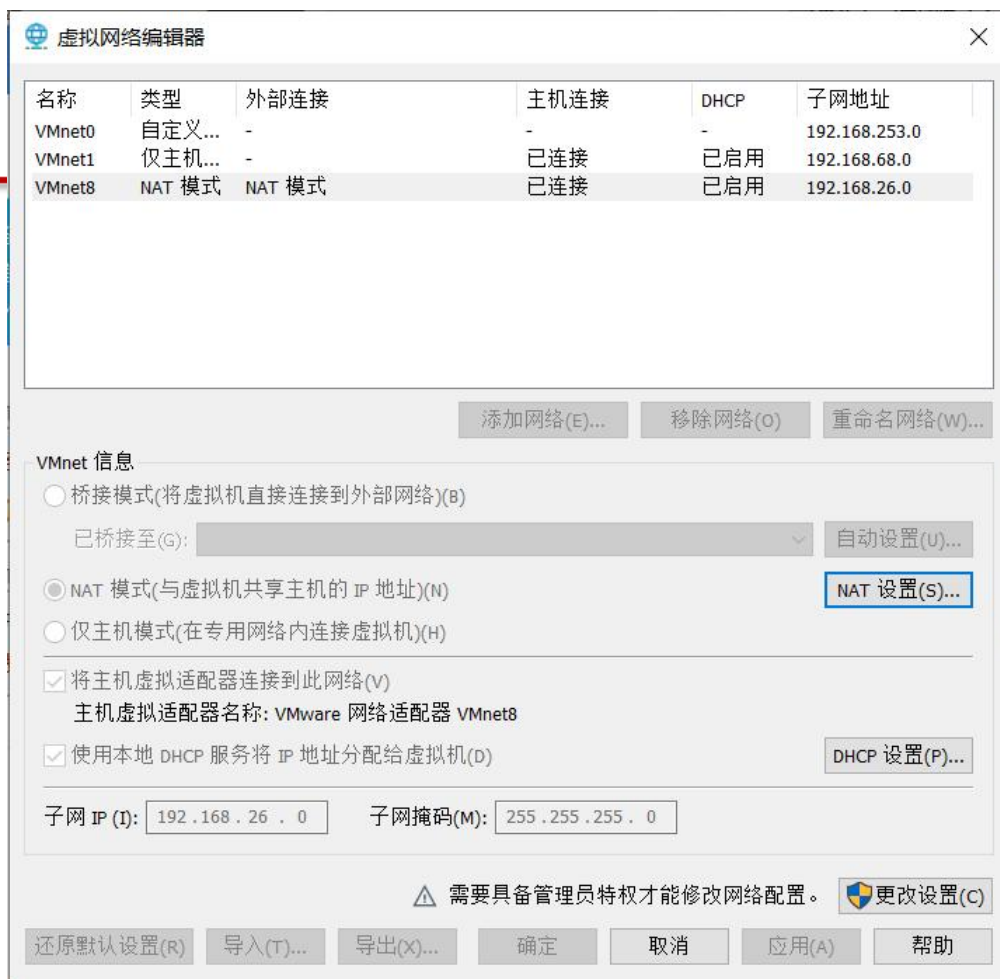
192.168.26.2

教室网络

小王的IP，为：
192.168.3.111

```
wlw@CM00:~  
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)  
wlw@CM00 ~]$ ifconfig  
eth0  Link encap:Ethernet  HWaddr 00:0C:29:76:7F:D2  
       inet addr:192.168.26.3  Bcast:192.168.26.255  Mask:255.255.255.0  
       inet6 addr: fe80::20c:29ff:fe76:7fd2/64 Scope:Link  
       UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1  
       RX packets:247 errors:0 dropped:0 overruns:0 frame:0  
       TX packets:44 errors:0 dropped:0 overruns:0 carrier:0  
       collisions:0 txqueuelen:1000  
       RX bytes:17637 (17.2 KiB)  TX bytes:3108 (3.0 KiB)  
  
lo    Link encap:Local Loopback  
       inet addr:127.0.0.1  Mask:255.0.0.0  
       inet6 addr: ::1/128 Scope:Host  
       UP LOOPBACK RUNNING  MTU:65536  Metric:1  
       RX packets:8 errors:0 dropped:0 overruns:0 frame:0  
       TX packets:8 errors:0 dropped:0 overruns:0 carrier:0  
       collisions:0 txqueuelen:0  
       RX bytes:480 (480.0 b)  TX bytes:480 (480.0 b)
```

```
选择命令提示符  
以太网适配器 VMware Network Adapter VMnet1:  
连接特定的 DNS 后缀 . . . . . :  
本地链接 IPv6 地址. . . . . : fe80::4cd8:e00f:ba4a:4da1%15  
IPv4 地址. . . . . : 192.168.68.1  
子网掩码. . . . . : 255.255.255.0  
默认网关. . . . . :  
  
以太网适配器 VMware Network Adapter VMnet8:  
连接特定的 DNS 后缀 . . . . . :  
本地链接 IPv6 地址. . . . . : fe80::f9c4:8a4e:993e:1e14%7  
IPv4 地址. . . . . : 192.168.26.1  
子网掩码. . . . . : 255.255.255.0  
默认网关. . . . . :  
  
以太网适配器 以太网 2:  
媒体状态 . . . . . : 媒体已断开连接  
连接特定的 DNS 后缀 . . . . . :  
  
无线局域网适配器 WLAN:  
连接特定的 DNS 后缀 . . . . . :  
本地链接 IPv6 地址. . . . . : fe80::9024:cecf:ef:bfa%8  
IPv4 地址. . . . . : 192.168.3.7  
子网掩码. . . . . : 255.255.255.0  
默认网关. . . . . : 192.168.3.1  
  
以太网适配器 蓝牙网络连接 2:  
媒体状态 . . . . . : 媒体已断开连接  
连接特定的 DNS 后缀 . . . . . :  
  
C:\Users\may>
```

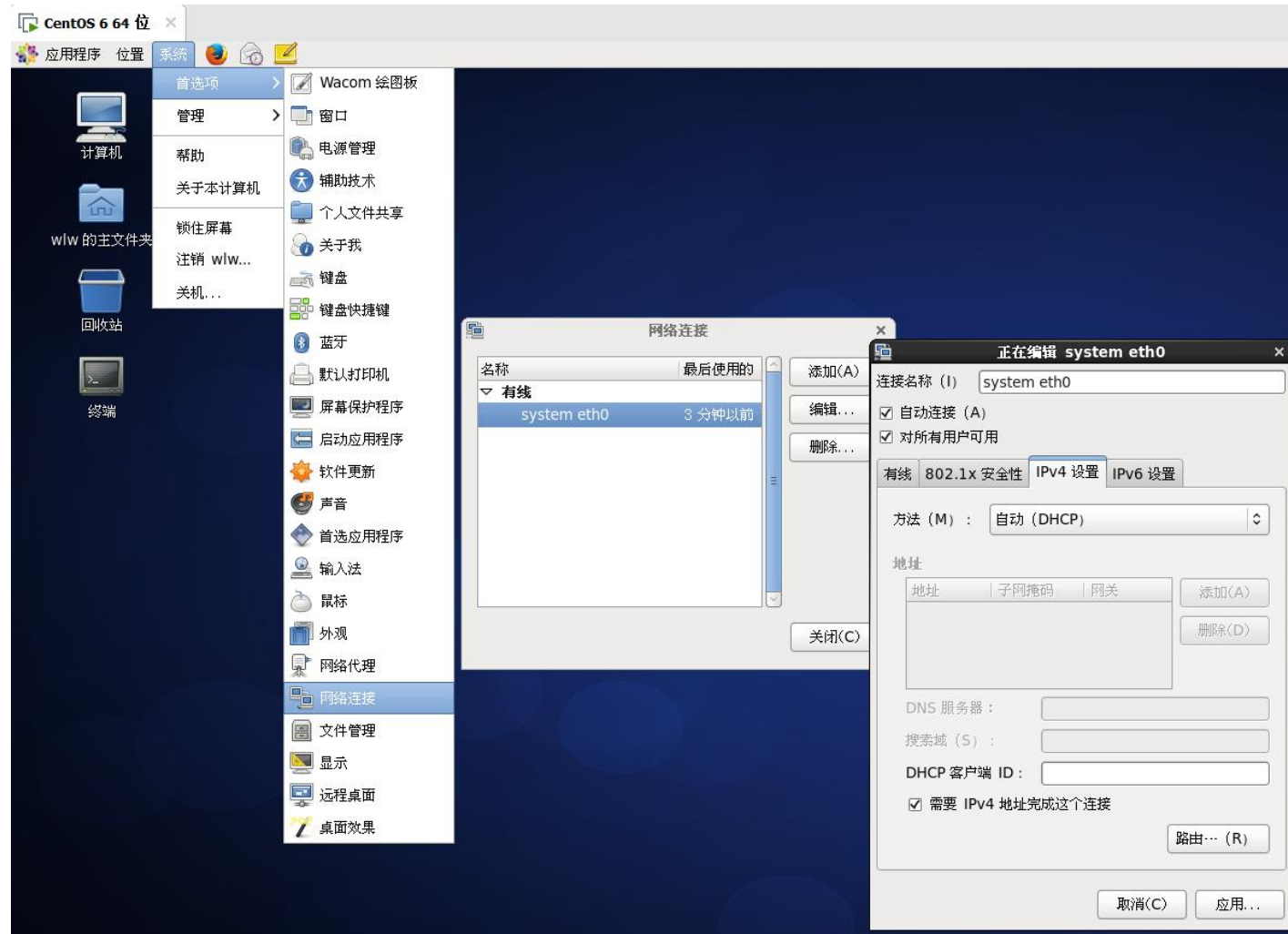


虚拟网络编辑中查看网关



虚拟机的虚拟网络编辑中的子网配置即
可以修改vmnet8的IP，即子网IP。
在虚拟网络编辑选中NAT模式后，点开设
置按钮可以查到网关IP，为
192.168.26.0

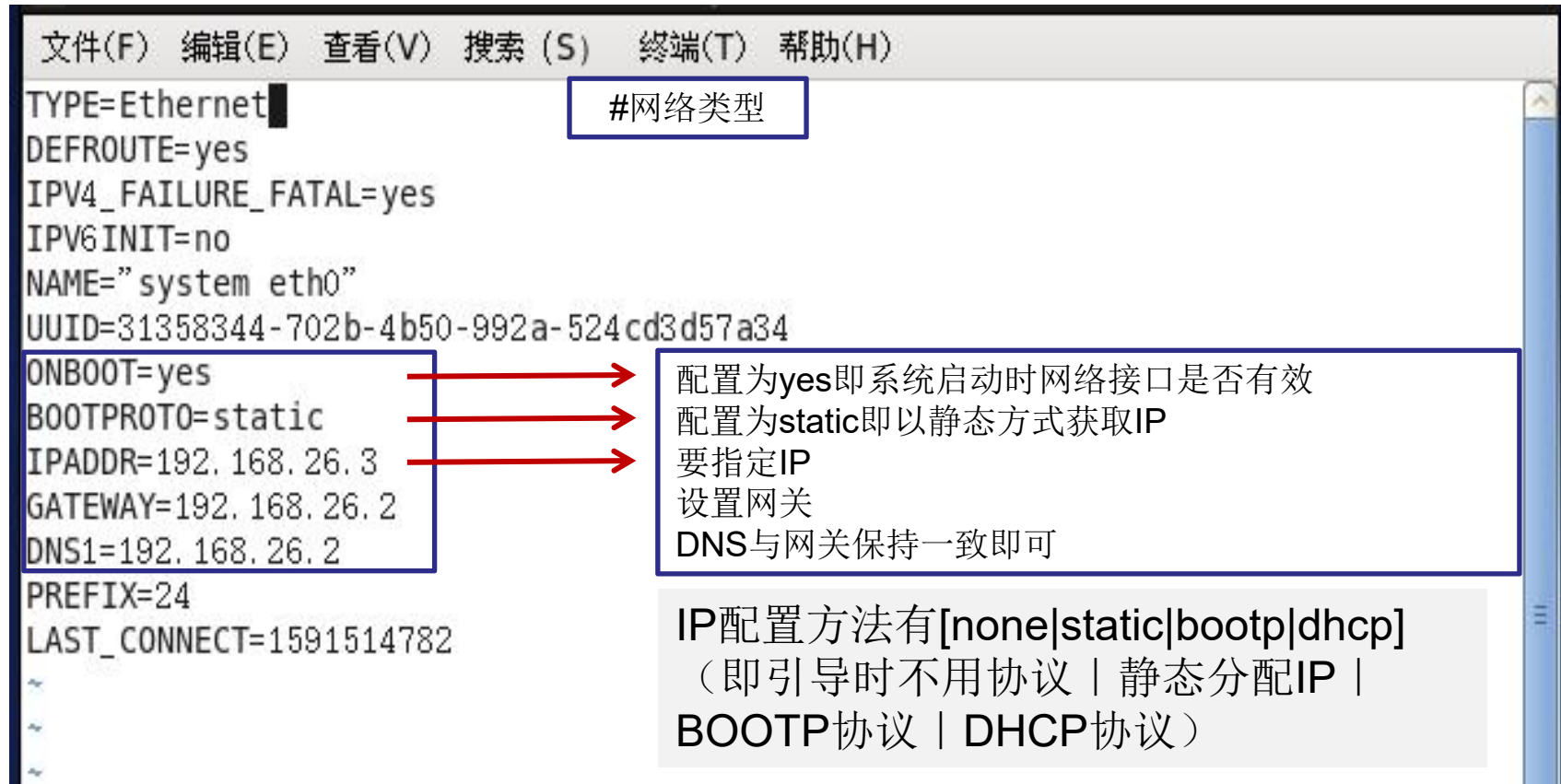
自动获取IP



说明：每次启动会自动连接，但因为是自动获取IP，每次不同，不适合用做服务器。

指定固定IP

直接修改eth0网卡的配置文件 **/etc/sysconfig/network-scripts/ifcfg-system_eth0** 来指定IP。



```
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
TYPE=Ethernet
DEFROUTE=yes
IPV4_FAILURE_FATAL=yes
IPV6INIT=no
NAME="system eth0"
UUID=31358344-702b-4b50-992a-524cd3d57a34
ONBOOT=yes
BOOTPROTO=static
IPADDR=192.168.26.3
GATEWAY=192.168.26.2
DNS1=192.168.26.2
PREFIX=24
LAST_CONNECT=1591514782
~
~
~
```

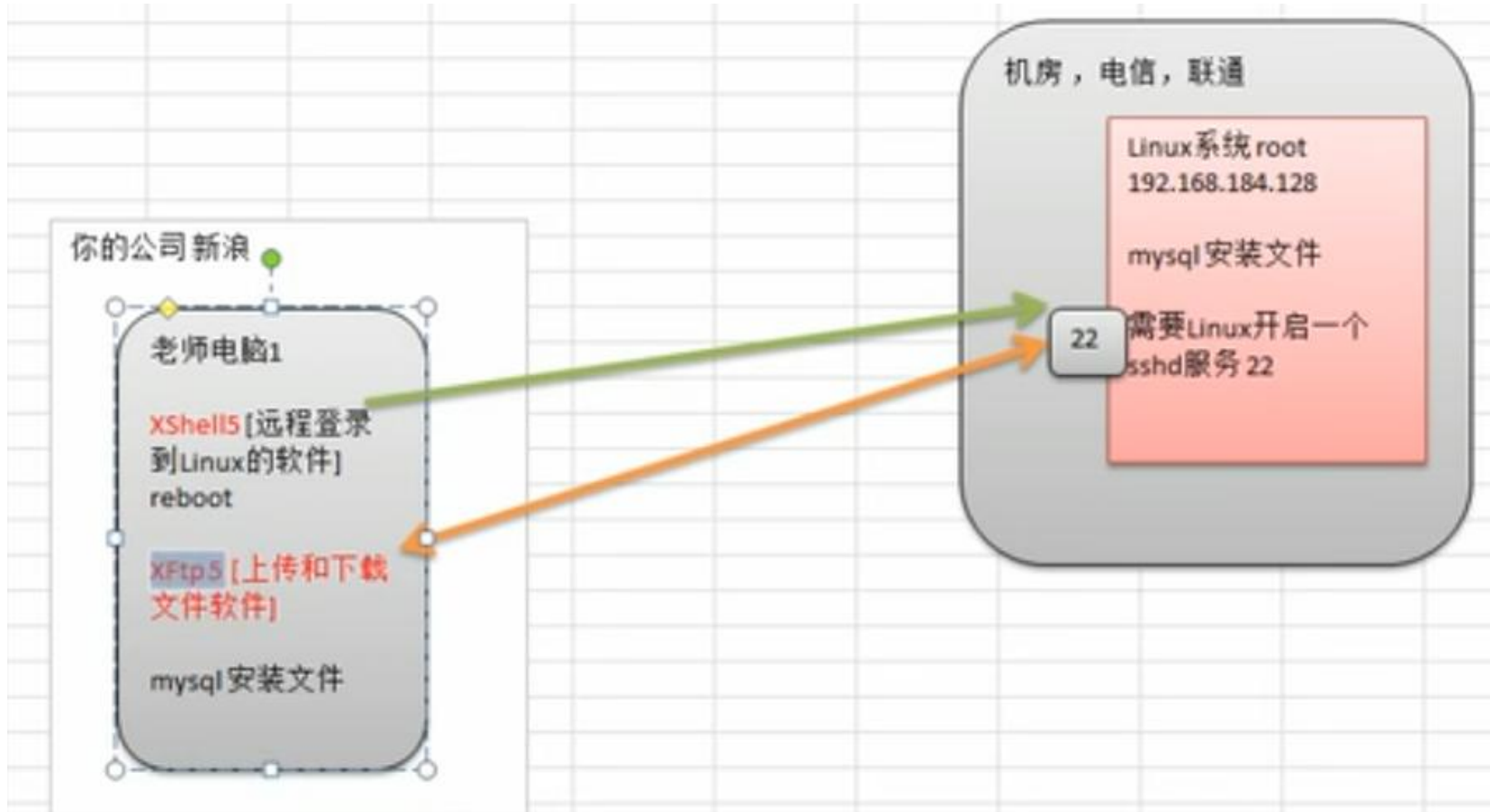
#网络类型

配置为yes即系统启动时网络接口是否有效
配置为static即以静态方式获取IP
要指定IP
设置网关
DNS与网关保持一致即可

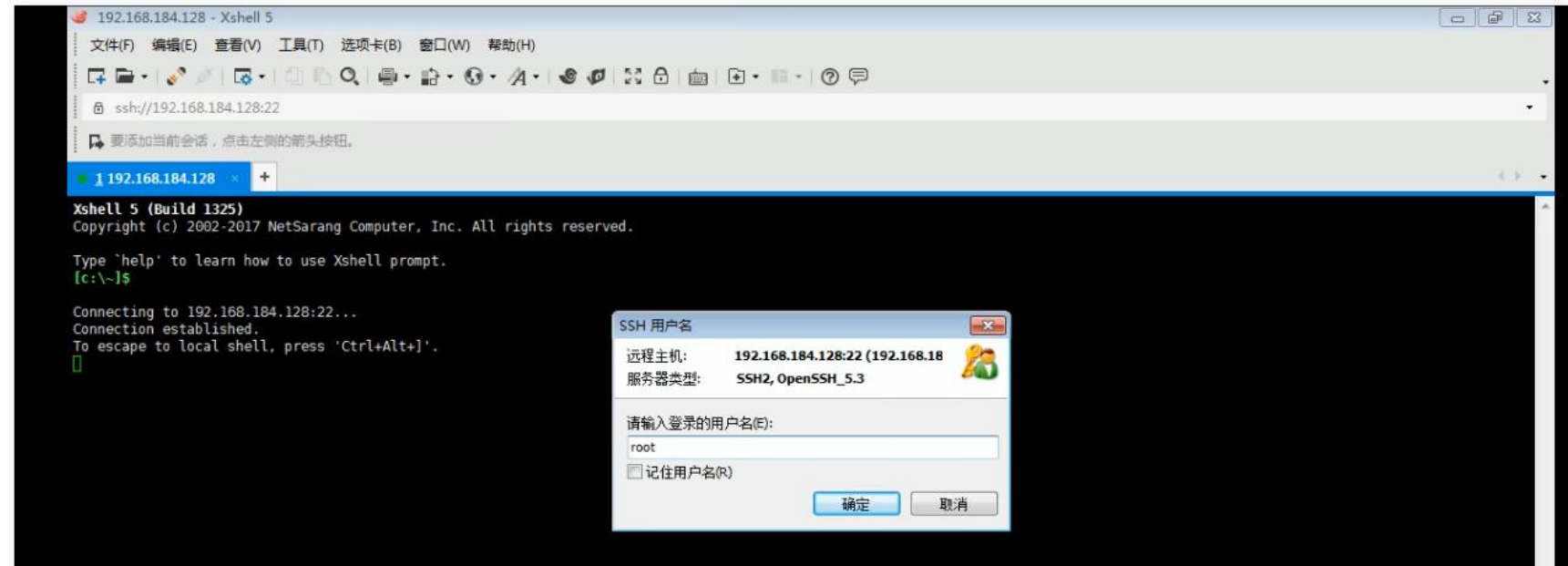
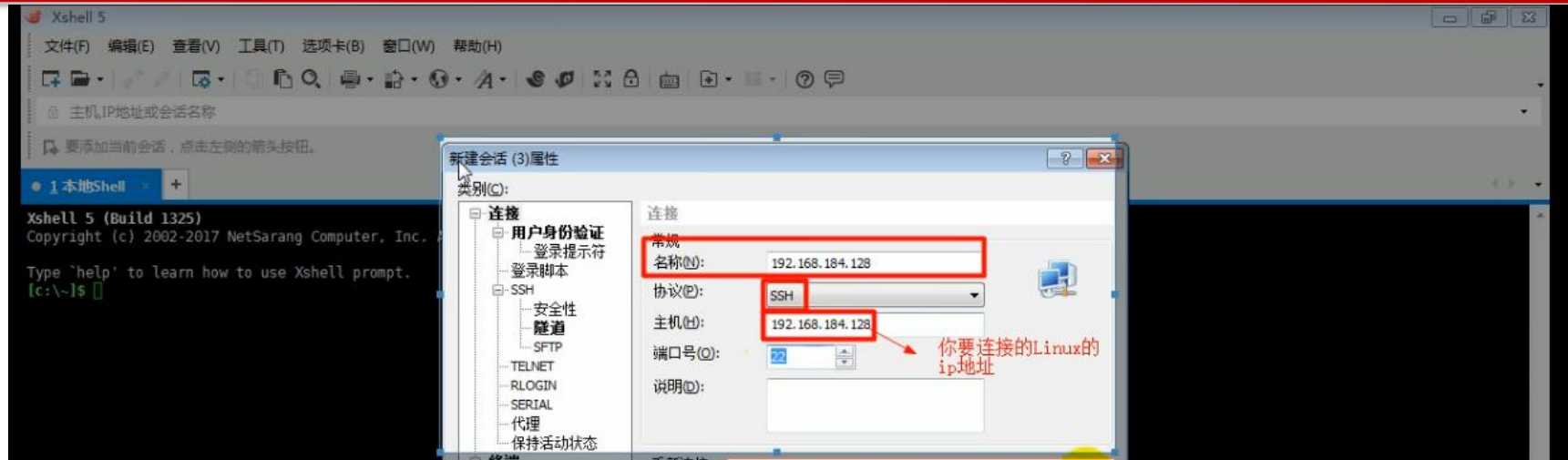
IP配置方法有[none|static|bootp|dhcp]
(即引导时不用协议 | 静态分配IP |
BOOTP协议 | DHCP协议)

修改后要重启网络服务或重启系统生效: **service network restart/reboot** 再ping 外网, 再测试Linux与Windows是否能ping通。

- 如果Linux作服务器使用，则ip配置完成后，可以通过远程连接工具进行访问和传输文件。常用的远程工具有：**putty**、**secureCRT**、**Xshell**、**Xftp**等。



XSHELL配置



```
1 CentOS-may x +
Xshell 7 (Build 0099)
Copyright (c) 2020 NetSarang Computer, Inc. All rights reserved.

Type 'help' to learn how to use Xshell prompt.
[C:\~]$

Connecting to 192.168.26.3:22...
Could not connect to '192.168.26.3' (port 22): Connection failed.

Type 'help' to learn how to use Xshell prompt.
[C:\~]$
```

```
1 root@CM00:~ x +
Xshell 7 (Build 0099)
Copyright (c) 2020 NetSarang Computer, Inc. All rights reserved.

Type 'help' to learn how to use Xshell prompt.
[C:\~]$

Connecting to 192.168.26.3:22...
Could not connect to '192.168.26.3' (port 22): Connection failed.

Type 'help' to learn how to use Xshell prompt.
[C:\~]$

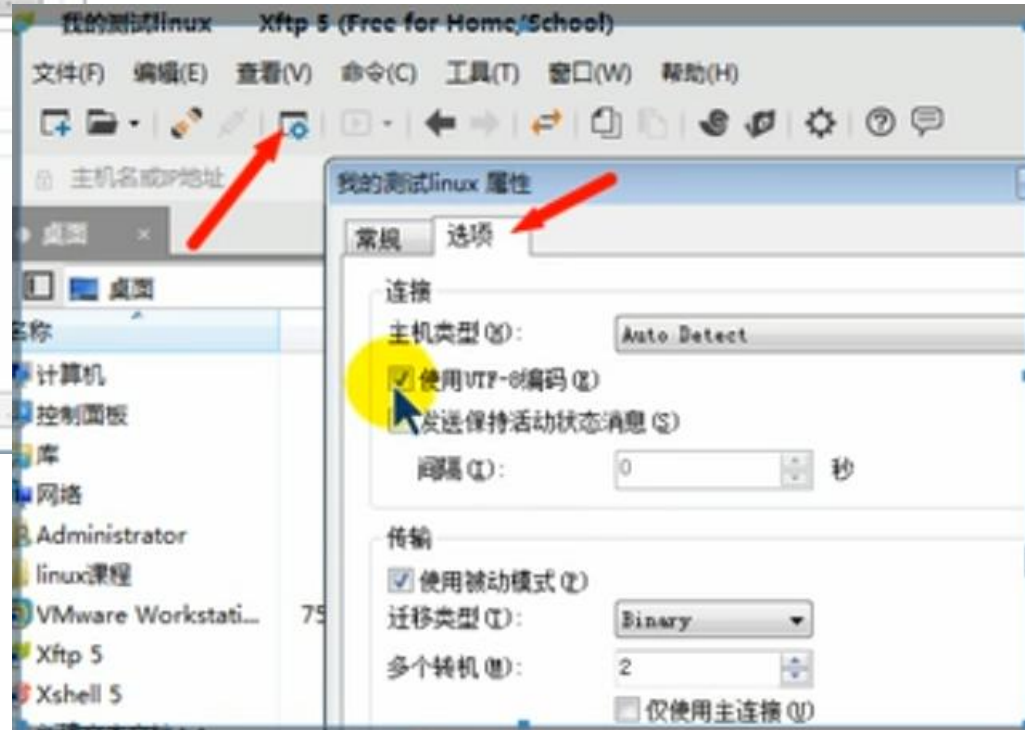
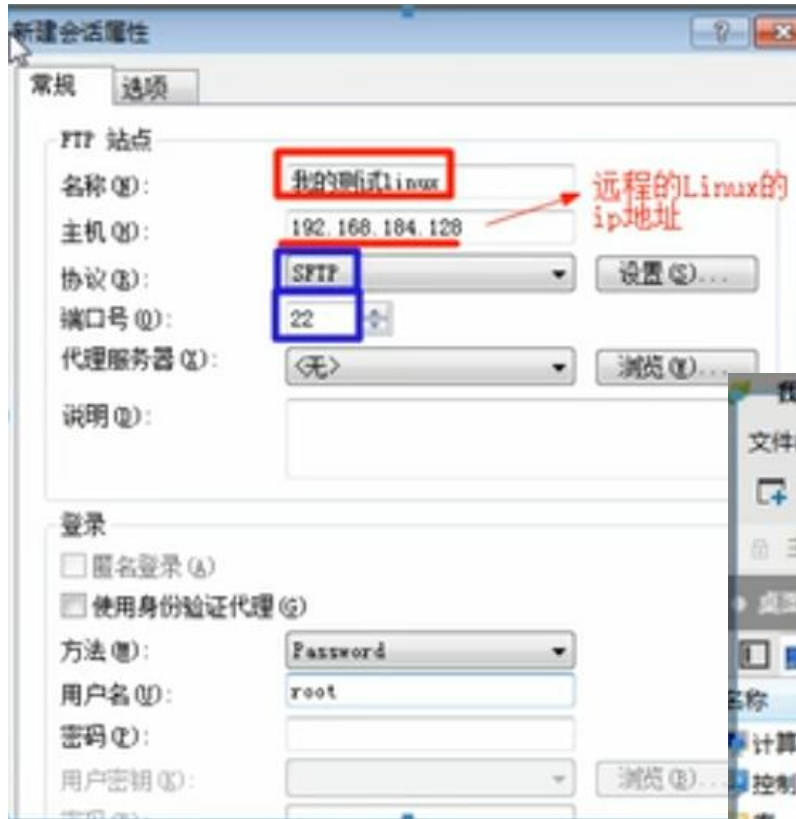
Connecting to 192.168.26.3:22...
Canceled.

Type 'help' to learn how to use Xshell prompt.
[C:\~]$

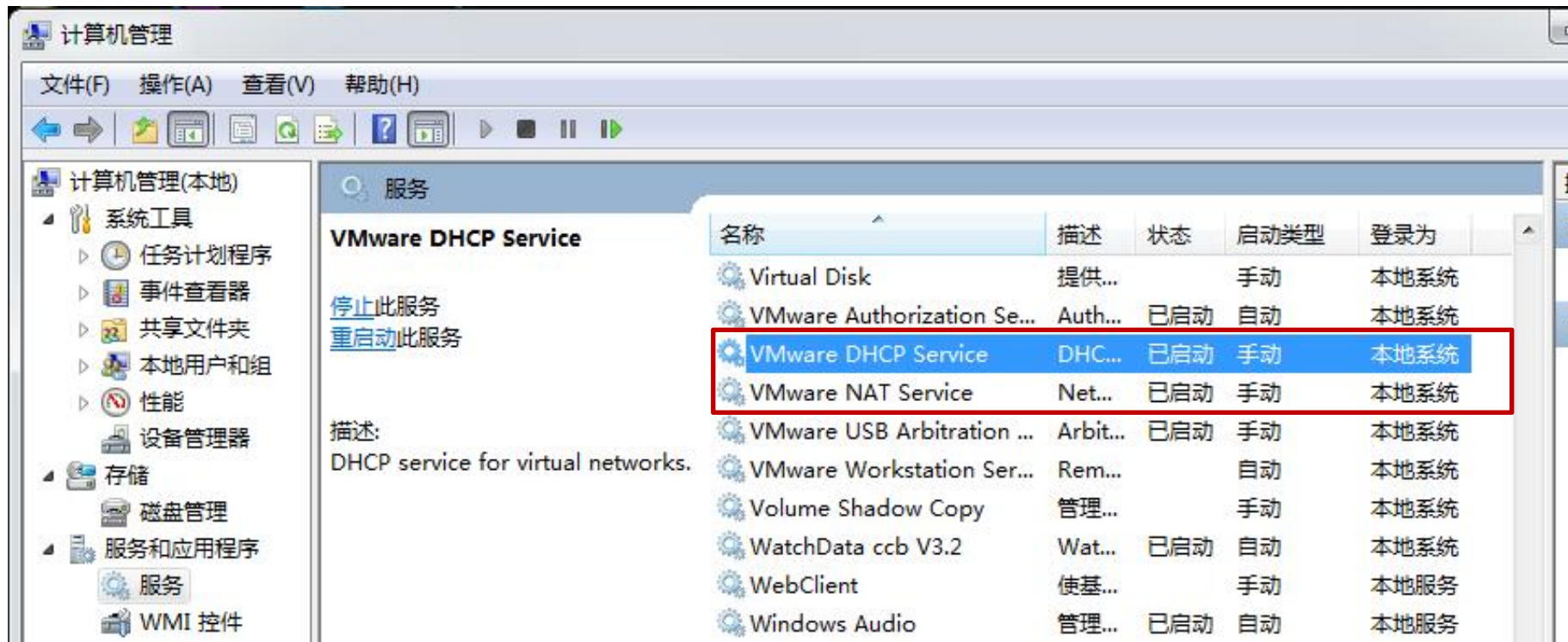
Connecting to 192.168.26.3:22...
Connection established.
To escape to local shell, press 'Ctrl+Alt+J'.

Last login: Wed Apr 28 08:47:38 2021 from 192.168.26.1
[root@CM00 ~]#
```

XFTP的配置



- 设置好网络后每次启动虚拟机之前，都要启动两项服务，就不用每次进入**LINUX**后启动网络服务。
- 点击桌面的"计算机"-“管理”-“服务和应用程序”-“服务”，然后找到**"VMware DHCP Service"**和**"VMware NAT Service"**并启动。





SSH网络服务



传统的网络服务程序，如：**ftp**、**pop**和**telnet**在本质上都是不安全的，因为它们在网上用明文传送口令和数据，很容易受到“中间人”（**man-in-the-middle**）这种方式的攻击。

所谓“中间人”的攻击方式，就是“中间人”冒充真正的服务器接收客户传给服务器的数据，然后再冒充客户把数据传给真正的服务器。服务器和客户之间的数据传送被“中间人”一转手做了手脚之后，就会出现很严重的问题。

通过使用**SSH**（**Secure SHell protocol**，安全外壳协议），客户可以把所有传输的数据进行加密，这样“中间人”这种攻击方式就不可能实现了，而且也能够防止**DNS**和**IP**欺骗。**OpenSSH**是**SSH**的替代软件且免费。

SSH 组成

ssh协议使用**tcp 22**号端口，是**C/S**架构，分为服务器端与客户端。

- 服务器端的程序有 **sshd**，所需要的软件包：
 - **openssh-server** 实现**SSH**服务器的功能
 - **openssh-clients** 包含**SSH**服务的客户端程序
 - **openssh-askpass**和**openssh-askpass-gnome** 只有在Linux的图形界面下使用**SSH**服务时才需要.

- 客户端的程序：
 - 在Linux下有**ssh**
 - 在Windows下有**putty**，**SecureCRT**，**SSH Secure Shell Client ...**

- 安装openssh包和sshd
 - 输入命令 `yum install openssh/sshd`
- 查看是否安装了openssh和sshd
 - 输入命令 `rpm -qa | grep openssh或sshd`

```
[root@localhost ~]# rpm -qa | grep openssh
openssh-askpass-6.6.1p1-16.fc21.i686
openssh-clients-6.6.1p1-16.fc21.i686
openssh-6.6.1p1-16.fc21.i686
openssh-server-6.6.1p1-16.fc21.i686
[root@localhost ~]# rpm -qa | grep sshd
[root@localhost ~]# _
```

- 查看服务是否开启
 - 输入命令 `ps -le | grep openssh/sshd`

```
[root@localhost ~]# ps -le | grep openssh
[root@localhost ~]# ps -le | grep sshd
4 S      0  1153      1  0  80      0 - 2847 poll_s ?          00:00:00 sshd
```

- 操作系统即使是最小安装**ssh**默认是已经安装的。
- 默认**ssh**服务也是开启的，可以通过命令**service sshd status**来查看**sshd**服务是否开启；

```
[root@localhost ~]# service sshd status
Redirecting to /bin/systemctl status sshd.service
■ sshd.service - OpenSSH server daemon
   Loaded: loaded (/usr/lib/systemd/system/sshd.service; disabled)
   Active: active (running) since Thu 2015-12-17 12:47:02 MST; 20min ago
     Docs: man:sshd(8)
           man:sshd_config(5)
  Main PID: 1153 (sshd)
    CGroup: /system.slice/sshd.service
            └─1153 /usr/sbin/sshd -D
[root@localhost ~]# _
```

- 如果希望开机自动启动**SSHD**服务，可输入
\$ sudo systemctl enable sshd.service
sshd服务程序的启动脚本 **/etc/init.d/ssh**



如果SSHD服务没有开启，则用# **service sshd start**命令开启服务。



```
[root@localhost /]# service sshd status
Redirecting to /bin/systemctl status sshd.service
■ sshd.service - OpenSSH server daemon
   Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled)
   Active: inactive (dead)
     Docs: man:sshd(8)
           man:sshd_config(5)
[root@localhost /]# service sshd start
Redirecting to /bin/systemctl start sshd.service
[root@localhost /]# /bin/systemctl start sshd.service
[root@localhost /]# service sshd status
Redirecting to /bin/systemctl status sshd.service
■ sshd.service - OpenSSH server daemon
   Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled)
   Active: active (running) since Mon 2016-01-04 20:44:58 MST; 17s ago
     Docs: man:sshd(8)
           man:sshd_config(5)
  Main PID: 1258 (sshd)
    CGroup: /system.slice/sshd.service
            └─1258 /usr/sbin/sshd -D

Jan 04 20:44:58 localhost.localdomain sshd[1258]: Server listening on 0.0.0.0 port 22.
Jan 04 20:44:58 localhost.localdomain sshd[1258]: Server listening on :: port 22.
[root@localhost /]#
```

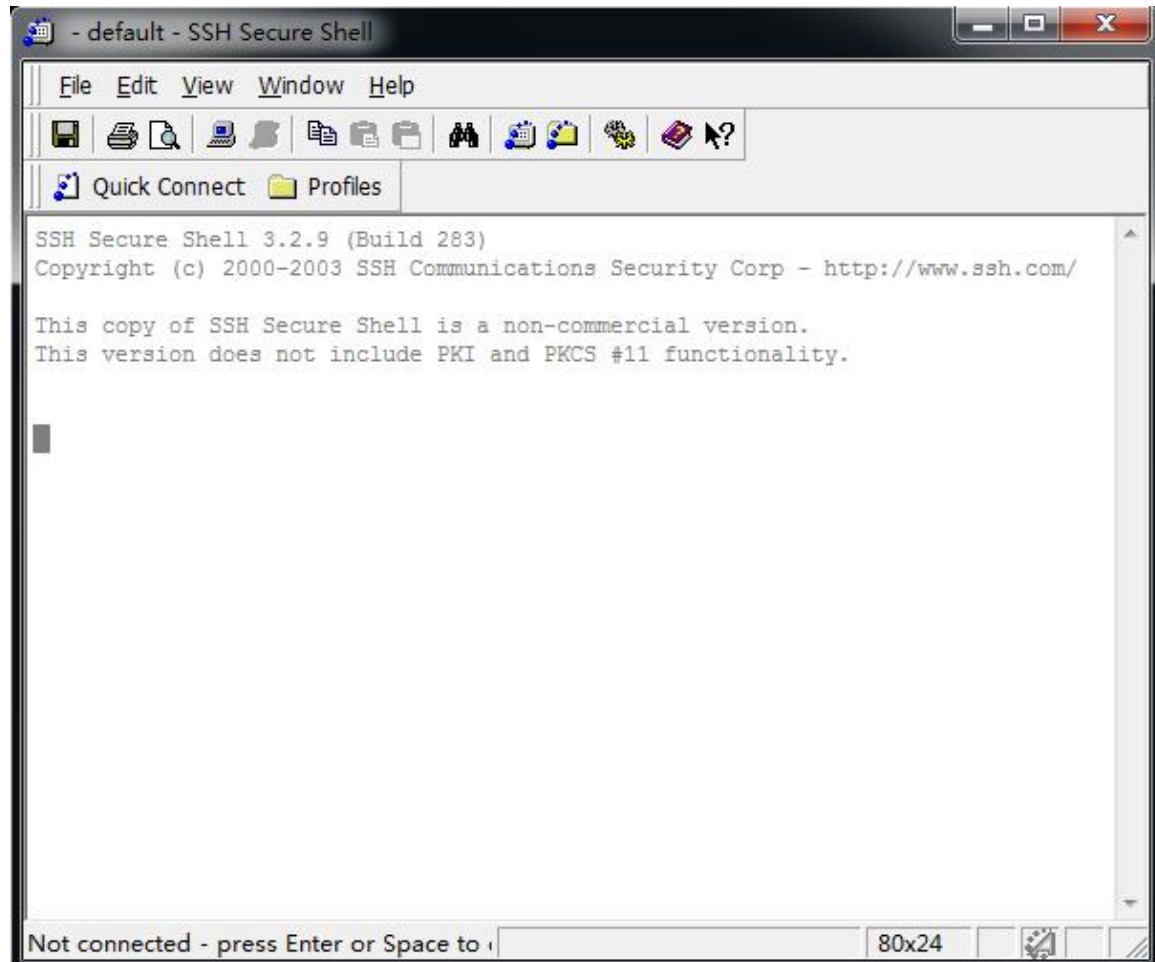
虚拟机中用SSH 实现Linux与Windows之间的文件传输

工作环境:

- 虚拟机Linux: Fedora
- WINDOWS下文件传输工具:

SSHSecureShellClient-3.2.9

- 1. 在Windows中安装文件传输工具
SSHSecureShellClient-3.2.9，主界面如下



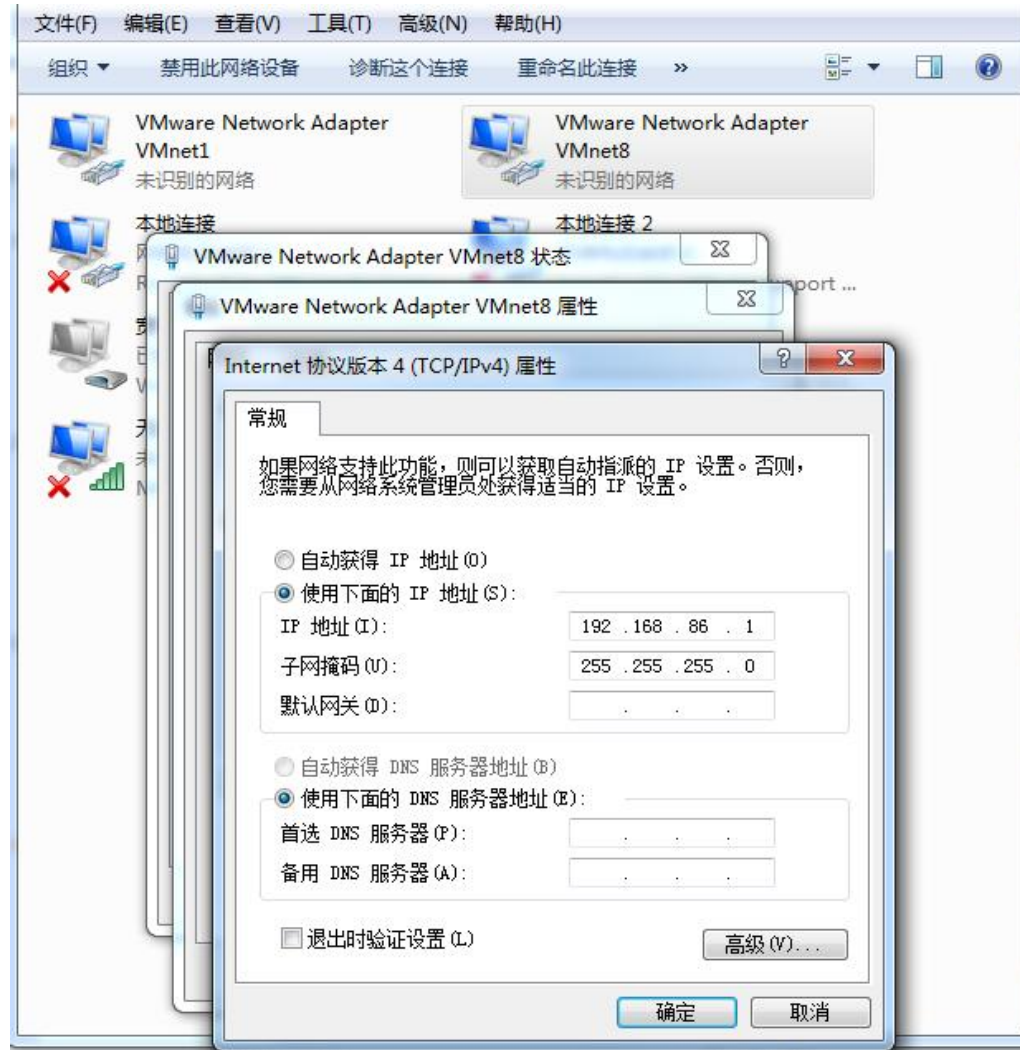
- 2. 启动虚拟机中的Linux，查看Linux的IP地址
ifconfig命令查看IP地址，此时IP是192.168.86.128

```
[root@localhost ~]# ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.86.128 netmask 255.255.255.0 broadcast 192.168.86.255
    inet6 fe80::20c:29ff:fef3:fafa prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:f3:fa:fa txqueuelen 1000 (Ethernet)
    RX packets 63 bytes 15427 (15.0 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 75 bytes 8234 (8.0 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
    device interrupt 19 base 0x2000

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 0 (Local Loopback)
    RX packets 8 bytes 800 (800.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 8 bytes 800 (800.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

[root@localhost ~]# _
```

- 3.虚拟网卡VMnet8的ip地址应与Linux在同一网段



- 在windows下ping 192.168.86.128

```
管理员: 命令提示符

Microsoft Windows [版本 6.1.7601]
版权所有 (c) 2009 Microsoft Corporation。保留所有权利。

C:\Users\Administrator>ping 192.168.86.128

正在 Ping 192.168.86.128 具有 32 字节的数据:
来自 192.168.86.128 的回复: 字节=32 时间=1ms TTL=64
来自 192.168.86.128 的回复: 字节=32 时间<1ms TTL=64
来自 192.168.86.128 的回复: 字节=32 时间=1ms TTL=64
来自 192.168.86.128 的回复: 字节=32 时间<1ms TTL=64

192.168.86.128 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
    往返行程的估计时间(以毫秒为单位):
        最短 = 0ms, 最长 = 1ms, 平均 = 0ms

C:\Users\Administrator>
```

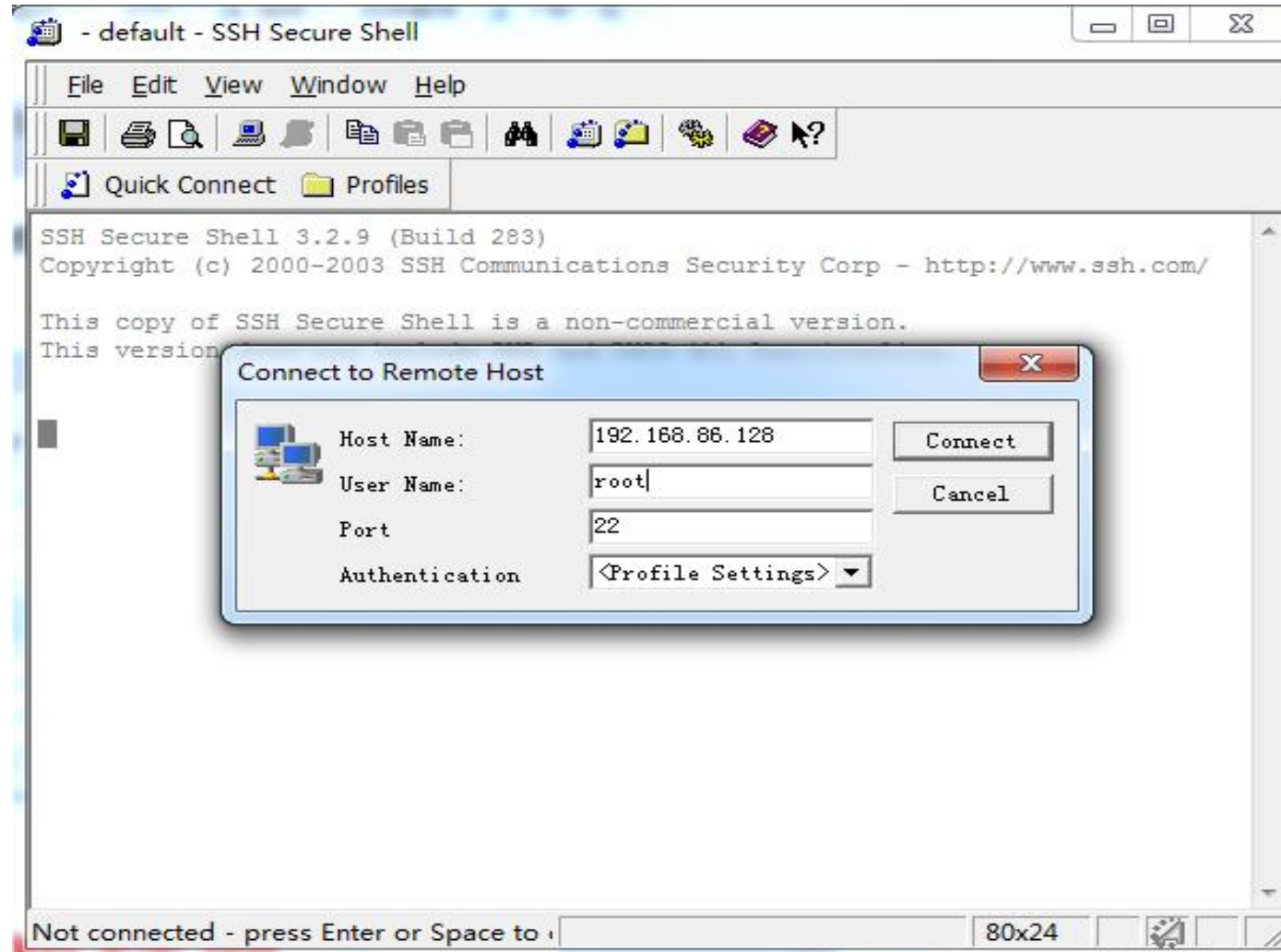
- 4. 使用命令开启Linux的SSHD服务:

service sshd start或#chkconfig sshd on开启服务

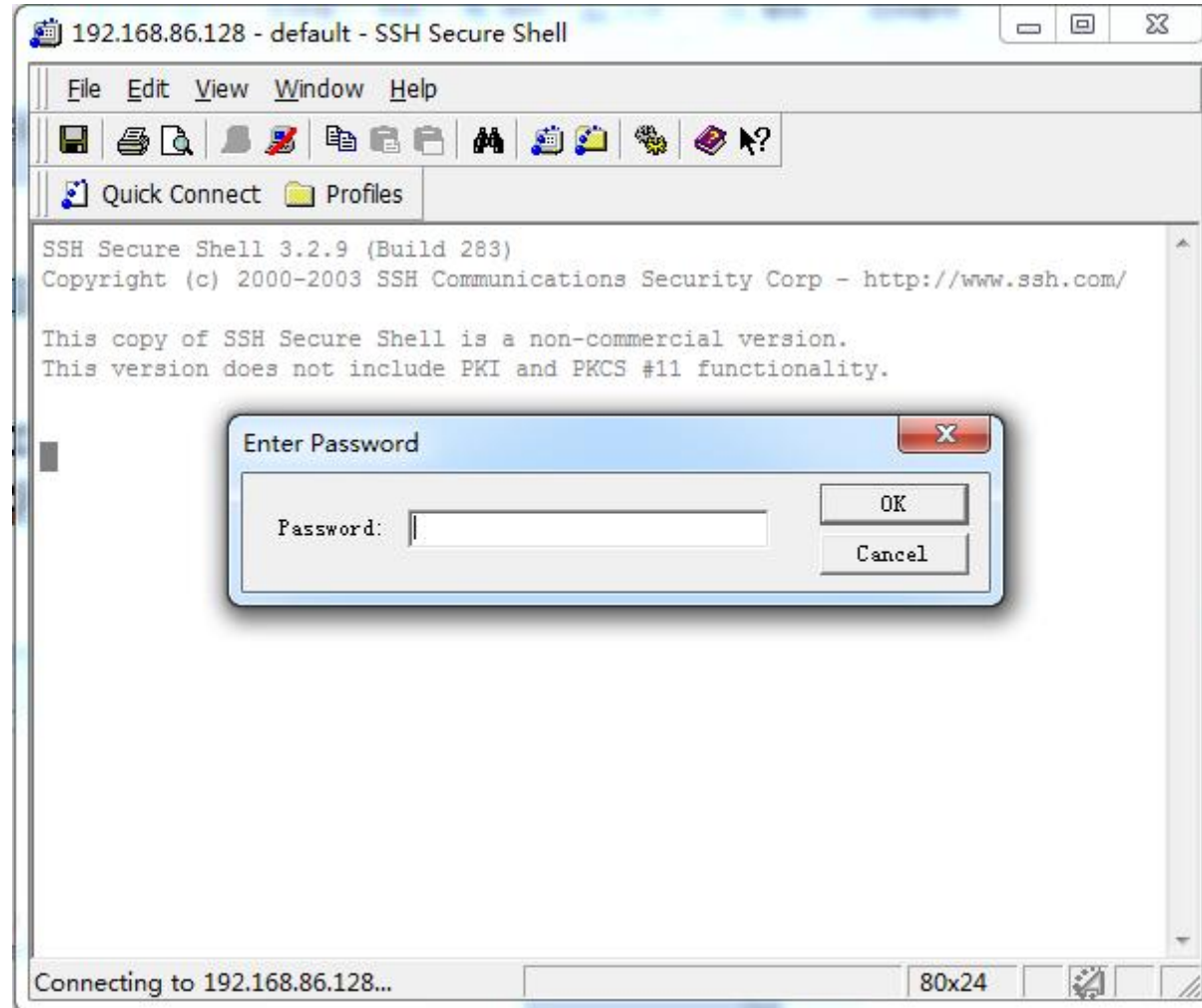
```
[root@localhost ~]# service sshd start
Redirecting to /bin/systemctl start  sshd.service
[root@localhost ~]# /bin/systemctl start sshd.service
[root@localhost ~]# service sshd status
Redirecting to /bin/systemctl status  sshd.service
■ sshd.service - OpenSSH server daemon
   Loaded: loaded (/usr/lib/systemd/system/ssh.service; enabled)
   Active: active (running) since Tue 2016-01-05 19:18:49 MST; 9min ago
     Docs: man:sshd(8)
           man:sshd_config(5)
  Main PID: 852 (sshd)
    CGroup: /system.slice/ssh.service
            └─852 /usr/sbin/sshd -D

Jan 05 19:18:51 localhost.localdomain sshd[852]: Server listening on 0.0.0.0 port 22.
Jan 05 19:18:51 localhost.localdomain sshd[852]: Server listening on :: port 22.
[root@localhost ~]# _
```

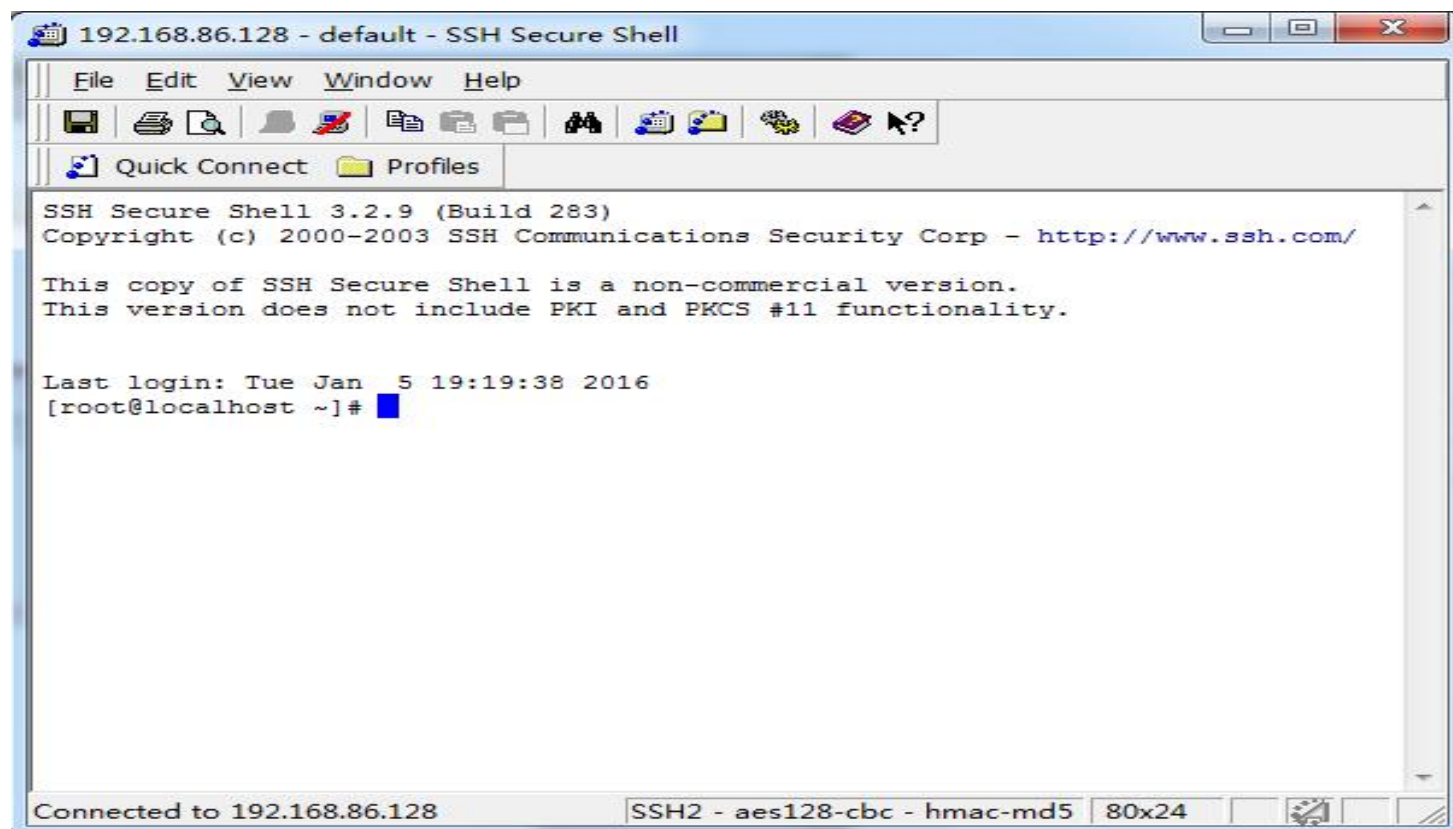
- 5. 通过SSHSecureShellClient-3.2.9连接Linux，单击Quick Connect按钮，输入linux IP地址及用户名。



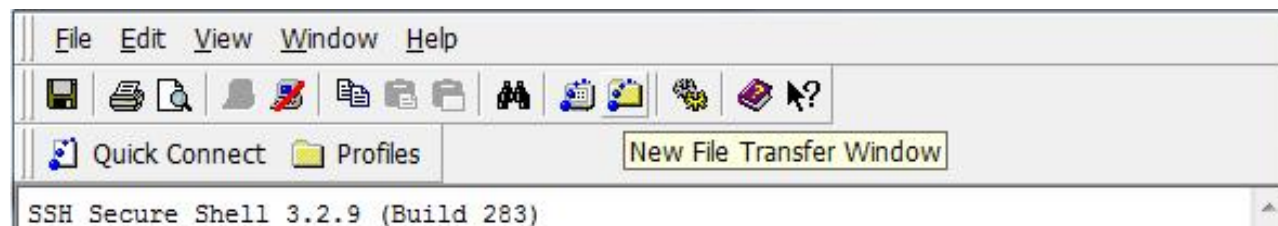
- 点击**connect**连接，在弹出的对话框中输入Linux用户的密码。

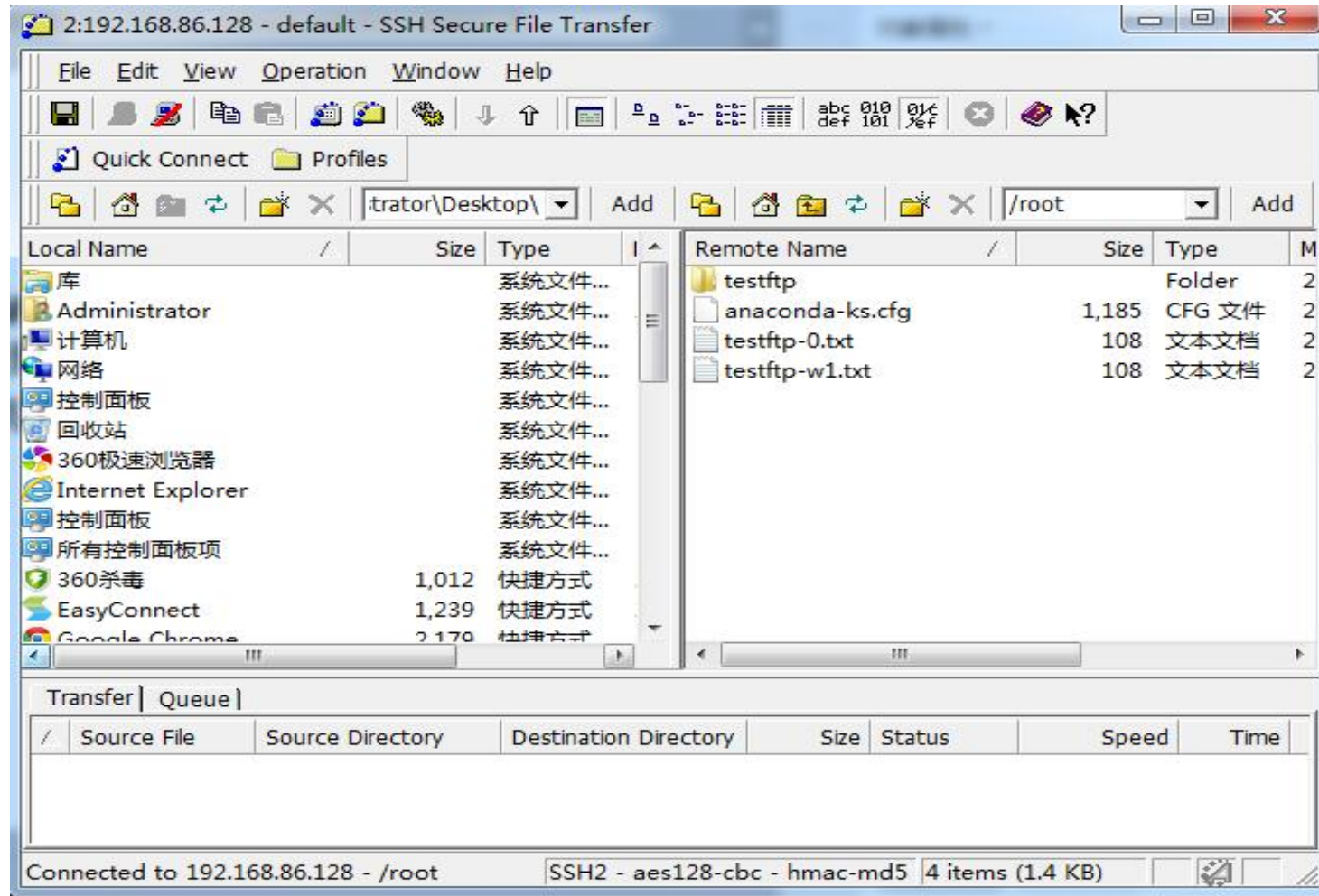


- 连接成功，显示如图



点击工具栏文件传输按钮
**New FileTransfer
Window**





- 已经连接到Linux，左边是Windows资源管理器，右边是Linux目录，默认打开到/root目录。可以在windows和虚拟机的linux间拖拽传输文件。