

实战作业4，实验环境下Linux主机之间实现跨网段通信 原创

青楼 | 2018-05-06 14:03:08

Linux学习从入门到打死也不放弃，完全笔记整理（持续更新，求收藏，求点赞~~~~）

<http://blog.51cto.com/13683480/2095439>

©著作权

实验目标：

用vmware多台虚拟机模拟，实现主机间的跨网段通信（相互之间能ping通）。

实验要求：

主机A 和 主机B 之间实现通信，中间至少跨越3个路由器（linux主机模拟路由器）

准备工作：

至少5台虚拟机。

由于本身只安装了centos6.9，centos7.4，redhat5.4 三台虚拟机，所以临时克隆了另外两台centos6，接下来将会以centos6充当主机A，centos7充当主机B

redhate5和另外两台克隆机将分别命名R1,R2,R3，充当临时路由器

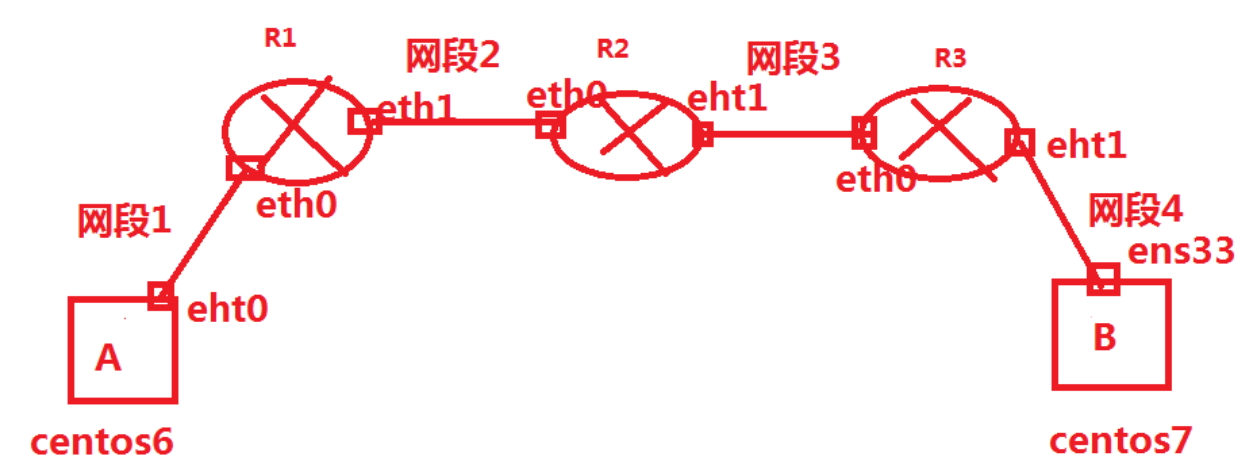
地址规划：

由于要求实现主机间的跨网段通信，所以需要将不同的终端配置不同网段的地址，

A: 192.168.6.45/24
B: 172.20.7.100/16
R1 eth0 : 192.168.6.1/24 eth1: 10.6.0.1/16
R2 eth0: 10.6.0.2/16 eth1: 10.0.0.1/8
R3 eht0: 10.0.0.2/8 eht1: 172.20.0.1/16

实现过程：

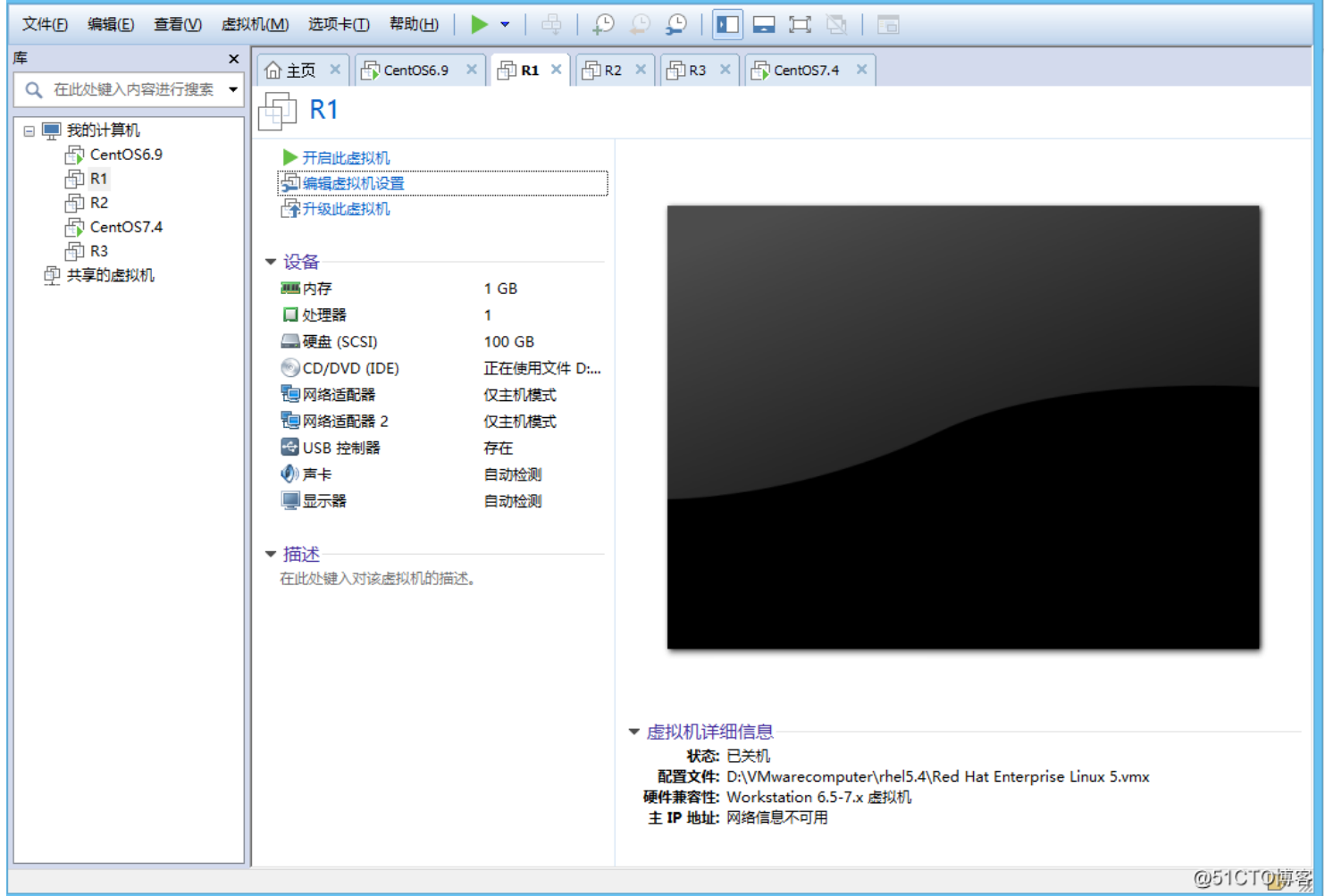
1. 大致流程图



@51CTO博客

2. 按照需要给不同主机加配网卡

充当路由使用的R1,R2,R3主机加配一块网卡，主机A 和主机B 有一块可用网卡即可



3. 开启虚拟机并登陆

为了查看方便，分别修改一下redhat5和两台克隆centos6的主机名为R1,R2,R3

由于是临时修改，所以使用hostname命令直接修改，而且重新登录即可，如图，3台主机重复执行即可

```
[root@station1 ~]# hostname R1
[root@station1 ~]#
[root@station1 ~]# exit
```

@51CTO博客

```
login: root
Password:
Last login: Mon Apr 30 16:09:15 on tty1
[root@R1 ~]#
[root@R1 ~]#
[root@R1 ~]#
```

@51CTO博客

需要注意的是，当在R2和R3上通过ifconfig命令查看的时候，发现此地没有eth0的网卡，而直接是 eth1 和eth2

这是由于R2和R3并非正常安装的虚拟机，而是通过克隆而来的

```

[root@R2 ~]# ifconfig
eth1      Link encap:Ethernet  HWaddr 00:0C:29:86:4B:69
          inet addr:192.168.65.135  Bcast:192.168.65.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe86:4b69/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:33 errors:0 dropped:0 overruns:0 frame:0
          TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:5739 (5.6 KiB)  TX bytes:1152 (1.1 KiB)

eth2      Link encap:Ethernet  HWaddr 00:0C:29:86:4B:73
          inet addr:192.168.65.138  Bcast:192.168.65.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe86:4b73/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:80 errors:0 dropped:0 overruns:0 frame:0
          TX packets:61 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:10559 (10.3 KiB)  TX bytes:5188 (5.0 KiB)

```

@51CTO博客

4. CentOS 6修改网卡名称：

当然，名称的不同使用起来并没有什么区别，但是我们已经不想去修改已经规划好的流程图了，所以这里我们直接修改网卡名称

centos6环境下，使用cd命令进入/etc/udev/rules.d/目录下，可以查看到：

```

[root@R2 ~]# cd /etc/udev/rules.d/
[root@R2 /etc/udev/rules.d]# ls
60-fprint-autosuspend.rules  60-raw.rules          90-alsa.rules            98-kexec.rules
60-openct.rules             70-persistent-cd.rules  90-hal.rules             99-fuse.rules
60-pcmcia.rules             70-persistent-net.rules  97-bluetooth-serial.rules
[root@R2 /etc/udev/rules.d]# _

```

@51CTO博客

其中70-persistent-net.rules就是网卡名称的配置文件

当然，也可以使用 `vim /etc/udev/rules.d/70-persistent-net.rules` 命令直接打开

```

1 # This file was automatically generated by the /lib/udev/write_net_rules
2 # program, run by the persistent-net-generator.rules rules file.
3 #
4 # You can modify it, as long as you keep each rule on a single
5 # line, and change only the value of the NAME= key.
6
7 # PCI device 0x0086:0x100f (e1000)
8 SUBSYSTEM=="net", ACTION=="add", DRIVERS=="*", ATTR{address}=="01:0c:29:6d:1c:1f", ATTR{type}=="
  "1", KERNEL=="eth*", NAME="eth0"
9
10 # PCI device 0x0086:0x100f (e1000)
11 SUBSYSTEM=="net", ACTION=="add", DRIVERS=="*", ATTR{address}=="00:0c:29:86:4b:69", ATTR{type}=="
  "1", KERNEL=="eth*", NAME="eth1"
12
13 # PCI device 0x0087:0x100f (e1000)
14 SUBSYSTEM=="net", ACTION=="add", DRIVERS=="*", ATTR{address}=="00:0c:29:86:4b:73", ATTR{type}=="
  "1", KERNEL=="eth*", NAME="eth2"

```

@51CTO博客

如图，直接删除第一行，将最后一行的eth2 改成 eth0 ，保存退出

修改不会立即生效，使用 `ethtool -i` 命令查看网卡使用的驱动，发现都是e1000

```

[root@R2 /etc/udev/rules.d]#sethtool -i eth1
driver: e1000
version: 7.3.21-k8-NAPI
firmware-version:
bus-info: 0000:02:01.0
supports-statistics: yes
supports-test: yes
supports-EEPROM-access: yes
supports-register-dump: yes
supports-priv-flags: no
[root@R2 /etc/udev/rules.d]#sethtool -i eth2
driver: e1000
version: 7.3.21-k8-NAPI
firmware-version:
bus-info: 0000:02:05.0
supports-statistics: yes
supports-test: yes
supports-EEPROM-access: yes
supports-register-dump: yes
supports-priv-flags: no
[root@R2 /etc/udev/rules.d]#

```

@51CTO博客

使用modprobe -r e1000 卸载网卡驱动

在使用modprobe e1000 启用网卡驱动，使用ifconfig命令，可以看见网卡名称已经修改：

接着在R3上重复相同的操作即可

```

[root@R2 /etc/udev/rules.d]#modprobe -r e1000
[root@R2 /etc/udev/rules.d]#modprobe e1000
[root@R2 /etc/udev/rules.d]#ifconfig
eth0
Link encap:Ethernet HWaddr 00:0C:29:86:4B:73
inet addr:192.168.65.138 Bcast:192.168.65.255 Mask:255.255.255.0
inet6 addr: fe80::20c:29ff:fe86:4b73/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:3 errors:0 dropped:0 overruns:0 frame:0
TX packets:7 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:746 (746.0 b) TX bytes:1082 (1.0 KiB)

eth1
Link encap:Ethernet HWaddr 00:0C:29:86:4B:69
inet addr:192.168.65.135 Bcast:192.168.65.255 Mask:255.255.255.0
inet6 addr: fe80::20c:29ff:fe86:4b69/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:4 errors:0 dropped:0 overruns:0 frame:0
TX packets:7 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:1088 (1.0 KiB) TX bytes:810 (810.0 b)

lo
Link encap:Local Loopback

```

@51CTO博客

5. 配置IP地址

当然，我们可以通过ifconfig eth0 直接配置IP地址，比如

```

[root@centos6 ~]#ifconfig eth0 192.168.6.45/24
[root@centos6 ~]#ifconfig
eth0
Link encap:Ethernet HWaddr 00:0C:29:6D:1C:1F
inet addr:192.168.6.45 Bcast:192.168.6.255 Mask:255.255.255.0
inet6 addr: fe80::20c:29ff:fe6d:1c1f/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:667 errors:0 dropped:0 overruns:0 frame:0
TX packets:8162 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:75560 (73.7 KiB) TX bytes:793883 (775.2 KiB)

```

@51CTO博客

但是这种情况下只要一重启，地址就会丢失，所以，这里我们通过修改配置文件的方式来修改地址

使用cd /etc/sysconfig/network-scripts

可以查看到此目录下有ifcfg-eth0的文件

```
lroot@centos6 ~]# cd /etc/sysconfig/network-scripts/
lroot@centos6 /etc/sysconfig/network-scripts]# ls
ifcfg-eth0  ifdown-ipp  ifdown-sit  ifup-ib     ifup-post   init.ipv6-global
ifcfg-lo    ifdown-ipv6 ifdown-tunnel ifup-ipp    ifup-ppp    net.hotplug
ifdown      ifdown-isd  ifup        ifup-ipv6   ifup-routes network-functions
ifdown-bnep ifdown-post ifup-aliases ifup-isdn   ifup-sit    network-functions-ipv6
ifdown-eth  ifdown-ppp  ifup-bnep   ifup-plip   ifup-tunnel
ifdown-ib   ifdown-routes ifup-eth    ifup-plusb  ifup-wireless
lroot@centos6 /etc/sysconfig/network-scripts]#
```

@51CTO博客

使用vim ifcfg-eth0打开此文件

```
1 DEVICE=eth0
2 TYPE=Ethernet
3 UUID=c5799ad4-8d31-4f67-9bb7-4c97c7a106c9
4 ONBOOT=yes
5 NM_CONTROLLED=yes
6 BOOTPROTO=dhcp
7 HWADDR=08:0C:29:6D:1C:1F
8 DEFROUTE=yes
9 PEERDNS=yes
10 PEERROUTES=yes
11 IPV4_FAILURE_FATAL=yes
12 IPV6INIT=no
13 NAME="System eth0"
14 LAST_CONNECT=1522150340
~
~
```

@51CTO博客

我们可以看到此文件包含的内容，几乎全是变量赋值，其他选项暂时不去关注，我们现在需要做的是把

BOOTPROTO=这一项默认的值是dhcp，需要改成 static 或者 none，

之后就可以添加IP地址了：

对于A主机而言，添上两行就可以了：

IPADDR=192.168.6.45

NETMASK=255.255.255.0 这一项也可以写成 PREFIX=24,效果一样

保存退出，使用

service network restart 重读配置文件

接着使用ifconfig查看，地址已经修改成功了

```
2 TYPE=Ethernet
3 UUID=c5799ad4-8d31-4f67-9bb7-4c97c7a106c9
4 ONBOOT=yes
5 NM_CONTROLLED=yes
6 BOOTPROTO=static
7 IPADDR=192.168.6.45
8 NETMASK=255.255.255.0
9 HWADDR=08:0C:29:6D:1C:1F
10 DEFROUTE=yes
11 PEERDNS=yes
12 PEERROUTES=yes
13 IPV4_FAILURE_FATAL=yes
14 IPV6INIT=no
15 NAME="System eth0"
16 LAST_CONNECT=1522150340
~
~
```

@51CTO博客

```

[root@centos6 ~]# service network restart
Shutting down interface eth0: Device state: 3 (disconnected)
[ OK ]
Shutting down loopback interface:
[ OK ]
Bringing up loopback interface:
[ OK ]
Bringing up interface eth0: Active connection state: activated
Active connection path: /org/freedesktop/NetworkManager/ActiveConnection/2
[ OK ]

[root@centos6 /etc/sysconfig/network-scripts]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0C:29:6D:1C:1F
          inet addr:192.168.6.45  Bcast:192.168.6.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe6d:1c1f/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:683 errors:0 dropped:0 overruns:0 frame:0
          TX packets:8164 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:77644 (75.8 KiB)  TX bytes:794003 (775.3 KiB)

```

@51CTO博客

同样的操作修改R1，由于redhat5.4的版本比较老，所有此文件中的内容会不一样，同样不去做过多关注，只修改我们需要的几项即可

```

# Advanced Micro Devices [AMD] 79c970 [PCnet32 LANCE]
DEVICE=eth0
BOOTPROTO=static
HWADDR=00:0C:29:5E:0F:2E
IPADDR=192.168.6.1
NETMASK=255.255.255.0
ONBOOT=yes

```

@51CTO博客

而且由于R1有两张网卡，所以eth0 和eth1 需要分别配置，然后依然执行service network restart命令重启网络服务，可以看到配置已经生效

```

[root@station1 network-scripts]# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:0C:29:5E:0F:2E
          inet addr:192.168.6.1  Bcast:192.168.6.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe5e:f2e/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:70 errors:0 dropped:0 overruns:0 frame:0
          TX packets:102 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:22008 (21.4 KiB)  TX bytes:22439 (21.9 KiB)
          Interrupt:67 Base address:0x2000

[root@station1 network-scripts]# ifconfig eth1
eth1      Link encap:Ethernet  HWaddr 00:0C:29:5E:0F:38
          inet addr:10.6.0.1  Bcast:10.6.255.255  Mask:255.255.0.0
          inet6 addr: fe80::20c:29ff:fe5e:f38/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:49 errors:0 dropped:0 overruns:0 frame:0
          TX packets:94 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:12455 (12.1 KiB)  TX bytes:22915 (22.3 KiB)
          Interrupt:67 Base address:0x2000

```

@51CTO博客

值得注意的是，R2和R3主机，默认是没有ifcfg-eth1 这个文件的，我们是用cp命令直接将ifcfg-eth0复制一份，然后修改可以

```

[root@R2 /etc/sysconfig/network-scripts]# cp ifcfg-eth0 ifcfg-eth1
[root@R2 /etc/sysconfig/network-scripts]# vim ifcfg-eth1_

```

@51CTO博客

另外，对于R2和R3，还是因为克隆的原因，使用service network restart命令的时候会出现错误

```

"ifcfg-eth1" 14L, 215C written
[root@R2 ~]# service network restart
Shutting down interface eth0: [ OK ]
Shutting down interface eth1: Device state: 3 (disconnected) [ OK ]

Shutting down loopback interface: [ OK ]
Bringing up loopback interface: [ OK ]
Bringing up interface eth0: Error: No suitable device found: no device found for connection 'System
eth0'. [ FAILED ]
Bringing up interface eth1: Active connection state: activated
Active connection path: /org/freedesktop/NetworkManager/ActiveConnection/4 [ OK ]

[root@R2 ~]# service network restart

```

@51CTO博客

特别提醒，这是由于NetworkManager 这个服务的原因，这个服务在centos6上有冲突，会导致网卡无法启动

使用service NetworkManager stop 命令可以关闭此服务

另外使用 chkconfig NetworkManager off 可以禁止此服务开机启动

再次使用service network restart命令，网卡启动成功，使用ifconfig命令成功获得地址

```

[root@R2 ~]# service NetworkManager stop
Stopping NetworkManager daemon: [ OK ]
[root@R2 ~]# chkconfig NetworkManager off
[root@R2 ~]# service network restart
Shutting down interface eth0: [ OK ]
Shutting down interface eth1: [ OK ]
Shutting down loopback interface: [ OK ]
Bringing up loopback interface: [ OK ]
Bringing up interface eth0: Determining if ip address 10.6.0.2 is already in use for device eth0... [ OK ]
Bringing up interface eth1: Determining if ip address 10.0.0.1 is already in use for device eth1... [ OK ]

[root@R2 ~]#

```

@51CTO博客

```

[root@R2 ~]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0C:29:86:4B:73
          inet addr:10.6.0.2  Bcast:10.6.255.255  Mask:255.255.0.0
          inet6 addr: fe80::20c:29ff:fe86:4b73/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:76 errors:0 dropped:0 overruns:0 frame:0
          TX packets:70 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:5306 (5.1 KiB)  TX bytes:5000 (4.8 KiB)

eth1      Link encap:Ethernet  HWaddr 00:0C:29:86:4B:69
          inet addr:10.0.0.1  Bcast:10.255.255.255  Mask:255.0.0.0
          inet6 addr: fe80::20c:29ff:fe86:4b69/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:47 errors:0 dropped:0 overruns:0 frame:0
          TX packets:74 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:3566 (3.4 KiB)  TX bytes:5240 (5.1 KiB)

```

@51CTO博客

同理在 R3 上做同样修改。

```

[root@R3 ~]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0C:29:81:B6:30
          inet addr:10.0.0.2  Bcast:10.255.255.255  Mask:255.0.0.0
          inet6 addr: fe80::20c:29ff:fe81:b630/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:69 errors:0 dropped:0 overruns:0 frame:0
          TX packets:47 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:5069 (4.9 KiB)  TX bytes:3294 (3.2 KiB)

eth1      Link encap:Ethernet  HWaddr 00:0C:29:81:B6:26
          inet addr:172.20.0.1  Bcast:172.20.255.255  Mask:255.255.0.0
          inet6 addr: fe80::20c:29ff:fe81:b626/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:66 errors:0 dropped:0 overruns:0 frame:0
          TX packets:15 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:4889 (4.7 KiB)  TX bytes:1086 (1.0 KiB)

```

@51CTO博客

最后是centos7，也就是A主机，需要注意centos7上的默认第一块网卡名称为ens33，且重启网络服务的命令为：

```
systemctl restart network
```

其他并没有大的区别：

```
[root@centos7 ~]# ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 172.20.7.100 netmask 255.255.0.0 broadcast 172.20.255.255
    inet6 fe80::cbd:8c18:143b:d67 prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:51:ca:5f txqueuelen 1000 (Ethernet)
    RX packets 684 bytes 99584 (97.2 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 227 bytes 41315 (40.3 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

@51CTO博客

6. 添加路由

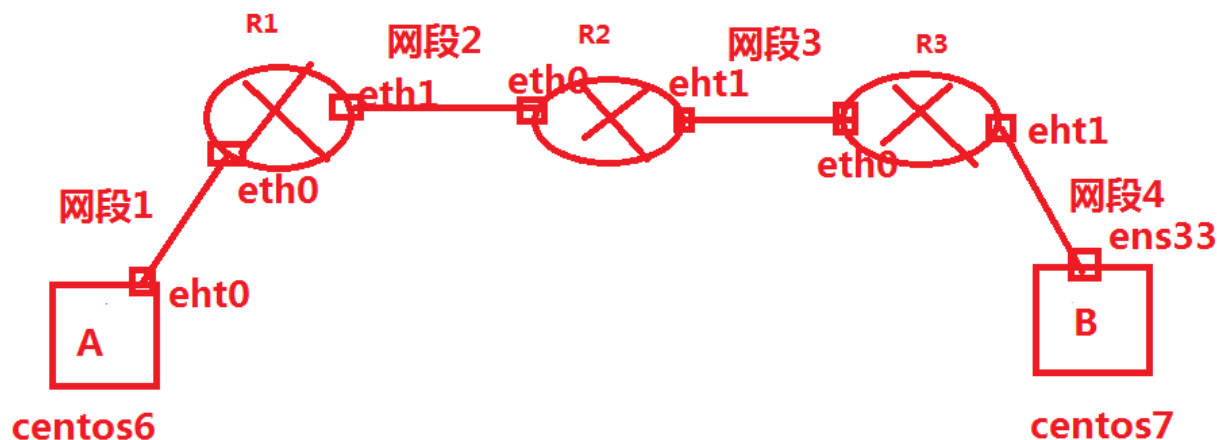
到这里地址配置的工作就完成了。

但是此时A和B两台主机之间是无法通信的，比如我们在A主机上ping B主机

```
[root@centos6 ~]# ping 172.20.7.100
connect: Network is unreachable
[root@centos6 ~]#
```

@51CTO博客

为了主机A和主机B 之间通信，必须配置路由，这是本次实验最重要的一环。我们在查看一下流程图和规划配置



@51CTO博客

A: 192.168.6.45/24

B: 172.20.7.100/16

R1 eth0 : 192.168.6.1/24 eth1: 10.6.0.1/16

R2 eth0: 10.6.0.2/16 eth1: 10.0.0.1/8

R3 eth0: 10.0.0.2/8 eth1: 172.20.0.1/16

主机A路由配置：

还是从A主机开始，由于规划的网络环境相对简单，A主机只与R1相连，因此在这里我们只需要添加一条路由就可以了，不管是主机路由，网络路由，默认路由都可以，但是使用的网关必须是R1路由器的eth0

使用route add -host 172.20.7.100 gw 192.168.6.1 添加主机路由

另外使用route add default gw 192.168.6.1 添加一条默认路由


```

[root@centos6 ~]# route add -host 172.20.7.100 gw 192.168.6.1
[root@centos6 ~]# route -n
Kernel IP routing table
Destination      Gateway         Genmask         Flags Metric Ref    Use Iface
172.20.7.100     192.168.6.1    255.255.255.255 UGH    0      0      0 eth0
192.168.6.0      0.0.0.0        255.255.255.0   U      0      0      0 eth0
[root@centos6 ~]# route add default gw 192.168.6.1
[root@centos6 ~]# route -n
Kernel IP routing table
Destination      Gateway         Genmask         Flags Metric Ref    Use Iface
172.20.7.100     192.168.6.1    255.255.255.255 UGH    0      0      0 eth0
192.168.6.0      0.0.0.0        255.255.255.0   U      0      0      0 eth0
0.0.0.0          192.168.6.1    0.0.0.0         UG     0      0      0 eth0
[root@centos6 ~]#

```

@51CTO博客

R1 路由配置

接下来，对于R1，由于R1和主机A在同一网段，所以不需要添加主机A路由，只需要添加主机B 或者网段4的路由记录即可，但是考虑到实际情况，网段3上也可能出现相连的主机，所以在此我们把网段3的路由也添加上

使用： route add -net 172.20.0.0/16 gw 10.6.0.2 添加网段4 路由记录 下一跳记录指向R2的eth0端口
route add -net 10.0.0.0/8 gw 10.6.0.2 添加网段3路由记录 下一跳记录指向R2的eth0端口

```

[root@R1 ~]# route add -net 172.20.0.0/16 gw 10.6.0.2
[root@R1 ~]# route add -net 10.0.0.0/8 gw 10.6.0.2
[root@R1 ~]# route -n
Kernel IP routing table
Destination      Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.6.0      0.0.0.0        255.255.255.0   U      0      0      0 eth0
10.6.0.0         0.0.0.0        255.255.0.0     U      0      0      0 eth1
169.254.0.0      0.0.0.0        255.255.0.0     U      0      0      0 eth1
172.20.0.0       10.6.0.2       255.255.0.0     UG     0      0      0 eth1
10.0.0.0         10.6.0.2       255.0.0.0       UG     0      0      0 eth1
[root@R1 ~]#

```

@51CTO博客

R2 路由配置

同理 R2 由于和网段2、网段3直接相连，所以只需要对应添加网段1 和网段4 的路由记录即可，

使用： route add -net 192.168.6.0/24 gw 10.6.0.1 添加网段1路由记录，下一跳记录指向R1的eth1端口
route add -net 172.20.0.0/16 gw 10.0.0.2 添加网段4路由记录，下一跳记录指向R3的eth0端口

```

[root@R2 ~]# route add -net 192.168.6.0/24 gw 10.6.0.1
[root@R2 ~]# route add -net 172.20.0.0/16 gw 10.0.0.2
[root@R2 ~]# route -n
Kernel IP routing table
Destination      Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.6.0      10.6.0.1       255.255.255.0   UG     0      0      0 eth0
10.6.0.0         0.0.0.0        255.255.0.0     U      0      0      0 eth0
169.254.0.0      0.0.0.0        255.255.0.0     U    1002    0      0 eth1
169.254.0.0      0.0.0.0        255.255.0.0     U    1003    0      0 eth0
172.20.0.0       10.0.0.2       255.255.0.0     UG     0      0      0 eth1
10.0.0.0         0.0.0.0        255.0.0.0       U      0      0      0 eth1

```

@51CTO博客

R3 路由配置

由于R3与网段3、网段4直接相连，所以需要添加网段1和网段2的路由记录

使用： route add -net 192.168.6.0/24 gw 10.0.0.1 添加网段1路由记录，下一跳指向R2的eth1端口
route add -net 10.6.0.0/16 gw 10.0.0.1 添加网段2路由记录，下一跳指向R2的eth1端口

```

[root@R3 ~]# route add -net 192.168.6.0/24 gw 10.0.0.1
[root@R3 ~]# route add -net 10.6.0.0/16 gw 10.0.0.1
[root@R3 ~]# route -n
Kernel IP routing table
Destination      Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.6.0      10.0.0.1        255.255.255.0   UG    0      0        0 eth0
10.6.0.0         10.0.0.1        255.255.0.0     UG    0      0        0 eth0
169.254.0.0      0.0.0.0         255.255.0.0     U      1002   0        0 eth1
169.254.0.0      0.0.0.0         255.255.0.0     U      1003   0        0 eth0
172.20.0.0        0.0.0.0         255.255.0.0     U      0      0        0 eth1
10.0.0.0          0.0.0.0         255.0.0.0       U      0      0        0 eth0

```

@51CTO博客

主机B 路由配置

由于B只有与R3相连这一条网络出口，所以只需要添加一条默认路由即可，这里我们同时也加上一条到主机A的主机路由
这里需要注意，对于centos7而言，如果在ifcfg-ens33配置静态地址的时候，如果不加上

GATEWAY=***

将无法配置默认路由，而如果加上这一行，使用 systemctl restart network重启网络服务的时候会自动添加一条默认路由

所以这里我们只需要将配置文件网关 (GATEWAY) =172.20.0.1 同时使用route add -host 192.168.6.45 gw 172.20.0.1 添加主机A的主机路由即可

```

[root@centos7 ~]# route -n
Kernel IP routing table
Destination      Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0          172.20.0.1      0.0.0.0         UG    100    0        0 ens33
172.20.0.0       0.0.0.0         255.255.0.0     U      100    0        0 ens33
192.168.65.0     0.0.0.0         255.255.255.0   U      100    0        0 ens37
192.168.122.0    0.0.0.0         255.255.255.0   U      0      0        0 virbr0

```

@51CTO博客

7. 连接测试

首先，直接在A主机上对主机B的IP地址使用ping命令

```

[root@centos6 /etc/sysconfig/network-scripts]# ping 172.20.7.100
connect: Network is unreachable
[root@centos6 /etc/sysconfig/network-scripts]#

```

@51CTO博客

很遗憾，第一次测试失败，原因就是，linux默认不支持直接当做路由器使用，即，linux主机可以接受到数据，但是如果数据的目标地址不是自己的话，会直接丢弃而不会转发出去

这里我们需要对R1,R2,R3这三台充当临时路由器的主机进行配置

使用 echo 1 > /proc/sys/net/ipv4/ip_forward 命令可以开启核心路由转发功能，对于目标地址非本机的数据，不会抛弃，而是根据路由地址进行转发。

```

[root@R1 ~]# echo 1 > /proc/sys/net/ipv4/ip_forward

```

```

[root@R2 ~]# echo 1 > /proc/sys/net/ipv4/ip_forward

```

```

[root@R3 ~]# echo 1 > /proc/sys/net/ipv4/ip_forward

```

配置完成之后再次尝试在A主机上ping 主机B的ip地址 172.20.7.100，结果发现还是不通

```

[root@centos6 ~]# ping 172.20.7.100
PING 172.20.7.100 (172.20.7.100) 56(84) bytes of data.
From 10.0.0.2 icmp_seq=1 Destination Host Prohibited
From 10.0.0.2 icmp_seq=2 Destination Host Prohibited
From 10.0.0.2 icmp_seq=3 Destination Host Prohibited
From 10.0.0.2 icmp_seq=4 Destination Host Prohibited

```

@51CTO博客

注意此时的报错提示：Destination Host Prohibited，目标主机禁止

这种情况一般是由于防火墙的原因

为了方便测试，使用 `service iptables stop` 命令分别关闭主机A，R1,R2,R3的防火墙功能

对于B主机，由于是centos7系统，需要使用 `systemctl stop firewalld` 命令关闭防火墙

再次测试：

```
[root@centos6 ~]# ping 172.20.7.100
PING 172.20.7.100 (172.20.7.100) 56(84) bytes of data.
64 bytes from 172.20.7.100: icmp_seq=1 ttl=61 time=1.97 ms
64 bytes from 172.20.7.100: icmp_seq=2 ttl=61 time=2.09 ms
64 bytes from 172.20.7.100: icmp_seq=3 ttl=61 time=2.18 ms
64 bytes from 172.20.7.100: icmp_seq=4 ttl=61 time=2.18 ms
^I^A64 bytes from 172.20.7.100: icmp_seq=5 ttl=61 time=2.10 ms
64 bytes from 172.20.7.100: icmp_seq=6 ttl=61 time=19.3 ms
^C
--- 172.20.7.100 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5870ms
rtt min/avg/max/mdev = 1.975/4.989/19.392/6.441 ms
[root@centos6 ~]#
```

@51CTO博客

可以看到结果已经OK，主机A(Centos6) 和主机B（centos7）经过4个网段，中间跨过3次路由器（R1,R2,R3）实现了通信功能。

8. 小结

虽然只是一个小小的实验，但在实现的工程中也并不是顺风顺水，一步到位的。中间出现了很多问题。

比如网卡名称修改之后一直无法启动，原因是centos6系统 NetworkManager服务不是很支持。

还有centos7添加默认路由一直不成功，原因是没有添加默认网关

还有就是最最最让我头疼的，一度困扰了很久的问题就是：Destination Host Prohibited，就是这个报错，这里我反复检查了N次路由配置，甚至重启重来，最后更换路由主机，都没有解决问题。

最后却发现问题出在防火墙上，原本很简单的一条命令就能解决的问题，却困扰了我最久。我以为我知道的，却还是忽略了。

事实上不管做什么事情都是这样，你以为你会的，不一定是真正掌握的。

送给你一句话，也是马永亮马哥反复强调的：

只有经过不断反复的练习和实践，知识才能变成真正有用的技能。

