Networking

Exercise 1: Review and preparation

1-1
Review the documentation for the ping utility and answer the following questions:
(a)What is the ping utility used for?
ping命令用来测试主机之间网络的连通性。执行ping指令会使用ICMP传输协议，发出要求回应的信息，若远端主机的网络功能没有问题，就会回应该信息，因而得知该主机运作正常。

(b) How does the ping utility work?
Ping命令发送一个ICMP请求报文给目的IP，然后目的IP回复一个ICMP报文。
原理：网络上的机器都有唯一确定的IP地址，我们给目标IP地址发送一个数据包，对方就要返回一个同样大小的数据包，根据返回的数据包我们可以确定目标主机的存在，可以初步判断目标主机的操作系统等。因为ping命令是使用ICMP协议，所以没有端口号，但是有两个域：类型和代码。

(c) How can you detect if the host 10.17.1.1 is up?
ping 10.17.1.1
若有正确的回应信息，则说明主机之间能够连通，如果返回超时信息，则无法连通，有可能远程主机开启防火墙，拒绝接收icmp报文。

(d) If you want to send a ping to all computers on a subnet, how can you do it using the ping utility?
ping 192.168.1.255 子网广播

(e) What would happen if you could ping the address 255.255.255.255?
255.255.255.255 is a broadcast address, you are sending a ping to every device on your local network and you will get a reply from every device. The ping command is only showing the first reply it gets, in your case your own PC (127.0.0.1 is loopback) was the quickest. If you use a packet sniffer (like Wireshark) you will be able to see all replies. Some devices will reply to a normal ping but will not reply to a ping sent to a broadcast address. This is to prevent an exploit called a Smurf attack.
255.255.255.255 will also broadcast to every device on the internet. For obvious reason this is blocked, the message will not leave your local network.

(f) What command-line option to ping causes it to print numerical addresses?
ping -n

1-2
Review the documentation for the traceroute utility and answer the following questions:
(a)What is the traceroute utility used for?
traceroute命令用于追踪数据包在网络上的传输时的全部路径，它默认发送的数据包大小是40字节。
通过traceroute我们可以知道信息从你的计算机到互联网另一端的主机是走的什么路径。当然每次数据包由某一同样的出发点（source）到达某一同样的目的地(destination)走的路径可能会不一样，但基本上来说大部分时候所走的路由是相同的。
traceroute通过发送小的数据包到目的设备直到其返回，来测量其需要多长时间。一条路径上的每个设备traceroute要测3次。输出结果中包括每次测试的时间(ms)和设备的名称（如有的话）及其ip地址。

(b) How does traceroute work?
traceroute是利用ICMP及IP头部的TTL。首先，traceroute送出一个TTL是1的IP数据包（其实，每次送出的为3个40字节的包，包括源地址，目的地址和包发出的时间标签）到目的地，当路径上的第一个路由器收到IP数据包时，将TTL减1。此时，TTL变为0，所以将该路由器会将此IP数据包丢掉，并返回一个ICMP数据包（包括发IP包的源地址，IP包的所有内容及路由器的IP地址），当traceroute收到这个消息后，接着继续发生TTL为2的IP数据包给第二个路由器。以此类推，直到IP数据包达到最后一台路由器时，会返回一个ICMP echo reply的数据包。

(c) What command-line option causes traceroute to print numerical addresses?
Traceroute -n

(d) What is the third-to-last hop on the route to 150.203.99.8?

1-3
Review the documentation for the ifconfig, route, netstat and ip commands
and answer the following questions:
(a) How do you set the address of interface eth0 to 130.236.189.14/24 (netmask 255.255.255.0) and broadcast address 130.236.189.255 using ifconfig? How do you set it using ip?

```
ifconfig eth0 130.236.189.14 netmask 255.255.255.0
route add default gw 192.168.5.1
ip addr add 130.236.189.14/24 brd + dev eth1 label eth1:1
```

(b) How do you display the current routing table using route? How do you display it using ip? Using netstat?

```
route
ip route show
netstat -r
```

1-4
Review the documentation for the sysctl command and answer the following:
(a)What is a sysctl (not what does the command do, but what a sysctl actually is)?
sysctl用于配置与显示在/proc/sys目录中的内核参数．可以用sysctl来设置或重新设置联网功能，如IP转发、IP碎片去除以及源路由检查等。用户只需要编辑/etc/sysctl.conf文件，即可手工或自动执行由sysctl控制的功能。

(b) In what file can you place sysctl values so they are loaded at boot time?
/etc/sysctl.conf

(c) Which sysctl controls IP forwarding?
sysctl net.ipv4.ip_forward

1-5
Which addresses are your hosts supposed to use? Which names? Which netmasks and broadcast addresses? You can find this information on the course home page.
inet addr:192.168.1.125 Bcast:192.168.1.255 Mask:255.255.255.0
hostname: ubuntu

Exercise 2: Setting the hostname

2-1
Set the hostname on all your hosts. The router should be named gw, the server should be named server and the clients named client-1 and client-2. Don’t forget the FQDN. Please use the recommended names – it simplifies things for your lab assistant.

```
#router gw
root@/etc/hostname:/etc$ cat hosts
127.0.0.1 localhost gw
192.168.0.4 server
192.168.0.2 client-1
192.168.0.3 client-2
root@/etc/hostname:/etc$ cat hostname
gw#host client-2
root@/etc/hostname:/etc$ cat hosts
127.0.0.1 localhost client-2
192.168.0.4 server
192.168.0.2 client-1
root@/etc/hostname:/etc$ cat hostname
client-2#host server
root@/etc/hostname:/etc$ cat hosts
127.0.0.1 localhost server
192.168.0.3 client-2
192.168.0.2 client-1
root@/etc/hostname:/etc$ cat hostname
server#host client-1
root@/etc/hostname:/etc$ cat hosts
127.0.0.1 localhost client-1
192.168.0.3 client-2
192.168.0.4 server
root@/etc/hostname:/etc$ cat hostname
client-1
```

Exercise 3: Interface configuration (gateway)

3-1
Configure addresses and anything else that is needed on each network interface. The configuration must survive a restart of the system.

```
root@/etc/hostname:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr FE:FD:00:00:8A:DE
          inet addr:192.168.0.1  Bcast:192.168.0.255  Mask:255.0.0.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
          Interrupt:5
```

3-2
Bring each interface up, so that it can be used.

```
#host client-2
root@/etc/hostname:/etc$ ifconfig
eth0    Link encap:Ethernet  HWaddr FE:FD:00:00:D4:D6
      inet addr:192.168.0.3  Bcast:192.168.0.255
Mask:255.255.255.0
      UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
      RX packets:0 errors:0 dropped:0 overruns:0 frame:0
      TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:1000
      RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
      Interrupt:5
```

3-3
Use ifconfig to verify your configuration.

```
#host server
root@/etc/hostname:/etc$ ifconfig
eth0    Link encap:Ethernet  HWaddr FE:FD:00:00:EF:DB
      inet addr:192.168.0.4  Bcast:192.168.0.255
Mask:255.255.255.0
      UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
      RX packets:0 errors:0 dropped:0 overruns:0 frame:0
      TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:1000
      RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
      Interrupt:5
```

3-4
If ping is available, test that you can reach 130.236.189.1 from your gateway.

```
#host client-1
root@/etc/hostname:/etc$ ifconfig
eth0    Link encap:Ethernet  HWaddr FE:FD:00:00:BD:92
      inet addr:192.168.0.2  Bcast:192.168.0.255
Mask:255.255.255.0
      UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
      RX packets:0 errors:0 dropped:0 overruns:0 frame:0
      TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:1000
      RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
      Interrupt:5
```

Exercise 4: Interface configuration (client)

4-1

Configure addresses and anything else that is needed on your clients (one client is sufficient at this time; you can do the other later).

```
root@/etc/hostname:~$ ifconfig
eth1    Link encap:Ethernet  HWaddr FE:FD:00:00:34:71
      inet addr:192.168.194.2  Bcast:192.168.194.255
Mask:255.255.255.0
      UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
      RX packets:0 errors:0 dropped:0 overruns:0 frame:0
      TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:1000
      RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
      Interrupt:5

      lo    Link encap:Local Loopback
      inet addr:127.0.0.1  Mask:255.0.0.0
      UP LOOPBACK RUNNING  MTU:16436  Metric:1
      RX packets:0 errors:0 dropped:0 overruns:0 frame:0
      TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:0
      RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

4-2

Bring the client’s interfaces up, so they can be used.

```
root@/etc/hostname:/etc$ ifconfig
lo    Link encap:Local Loopback
      inet addr:127.0.0.1  Mask:255.0.0.0
      UP LOOPBACK RUNNING  MTU:16436  Metric:1
      RX packets:0 errors:0 dropped:0 overruns:0 frame:0
      TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:0
      RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

4-3

Use ifconfig to verify your configuration.

```
root@/etc/hostname:/etc$ ifconfig
lo    Link encap:Local Loopback
      inet addr:127.0.0.1  Mask:255.0.0.0
      UP LOOPBACK RUNNING  MTU:16436  Metric:1
      RX packets:0 errors:0 dropped:0 overruns:0 frame:0
      TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:0
      RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

4-4
If ping is available, test that you can reach your gateway (the address it has on eth0).

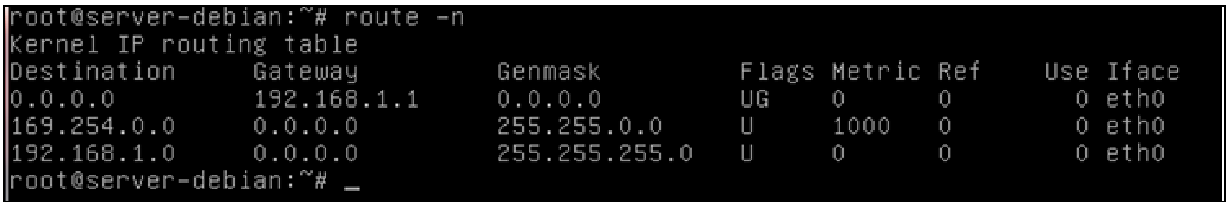
```
root@/etc/hostname:/etc$ ifconfig
lo    Link encap:Local Loopback
      inet addr:127.0.0.1  Mask:255.0.0.0
      UP LOOPBACK RUNNING  MTU:16436  Metric:1
      RX packets:0 errors:0 dropped:0 overruns:0 frame:0
      TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:0
      RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

Exercise 5: Default gateway/route (gateway)

5-1
Configure a suitable default route on your gateway.
5-2
Bring the appropriate interface down and then back up again to install the default route.
5-3
Use the route command to check that the default route has been installed.
5-4
If ping is available, test that you can reach 130.236.1.1 from your gateway.

Exercise 6: Default gateway/route (client)

6-1
Configure a suitable default route on your clients.
6-2
Bring the appropriate interface down and then back up again to install the default route.
6-3
Use the route command to check that the default route has been installed.



```
root@server-debian:~# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         192.168.1.1     0.0.0.0         UG    0      0        0 eth0
169.254.0.0     0.0.0.0         255.255.0.0     U     1000   0        0 eth0
192.168.1.0     0.0.0.0         255.255.255.0   U     0      0        0 eth0
root@server-debian:~# _
```

Exercise 7: Resolver and NSS configuration

7-1

Answer the following questions:

(a)

What configuration file controls NSS?

(b)

What configuration file controls the resolver?

(c)

What does the “search” keyword in the resolver configuration file do?

(d)

Which order is more appropriate: look for hosts in local files first, then DNS or look for hosts in DNS first, then local files? Motivate your answer.

7-2

Configure the resolver on all systems to use 130.236.189.1 as its name server.

7-3

Configure the name service switch on all systems to use DNS as a source for host names.

7-4

Test that name resolution works by attempting to communicate with ida-gw.sysinst.ida.liu.se (use ping or telnet if they are available).

ofile: /etc/default/nss

ofile: /etc/recolv.conf

okeywords:

□nameserver #定义DNS服务器的IP地址

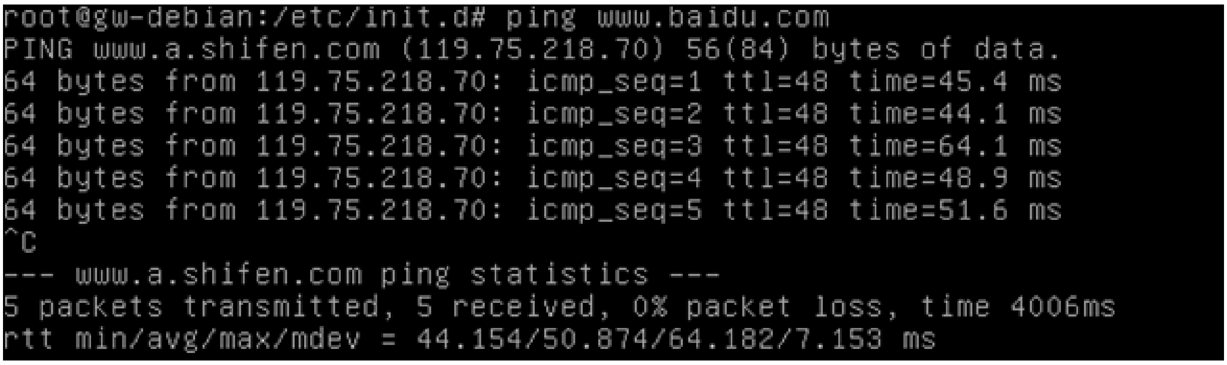
□domain #定义本地域名

□search #定义域名的搜索列表

□sortlist #对返回的域名进行排序

o(1) local file (2) DNS server

□reason: 可以避免GW的DNS污染之类的



```
root@gw-debian:/etc/init.d# ping www.baidu.com
PING www.a.shifen.com (119.75.218.70) 56(84) bytes of data.
64 bytes from 119.75.218.70: icmp_seq=1 ttl=48 time=45.4 ms
64 bytes from 119.75.218.70: icmp_seq=2 ttl=48 time=44.1 ms
64 bytes from 119.75.218.70: icmp_seq=3 ttl=48 time=64.1 ms
64 bytes from 119.75.218.70: icmp_seq=4 ttl=48 time=48.9 ms
64 bytes from 119.75.218.70: icmp_seq=5 ttl=48 time=51.6 ms
^C
--- www.a.shifen.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 44.154/50.874/64.182/7.153 ms
```

Exercise 8: Quagga configuration files

8-1
Quagga consists of several separate processes. Describe how they fit together.
zebra - kernel interface, static routes, zserv server
ripd ripngd - RIPv1/RIPv2 for IPv4 and RIPng for IPv6
ospfd ospf6d - OSPFv2 and OSPFv3
bgpd - BGPv4+ (including address family support for multicast and IPv6)
isisd - IS-IS with support for IPv4 and IPv6
under development or unmaintained:
olsrd - OLSR wireless mesh routing through a plugin for olsrd
ldpd - MPLS Label Distribution Protocol
bfdd - Bidirectional Forwarding Detection

8-2
What configuration files does the Debian installation of Quagga use?
/etc/quagga/debian.conf

8-3
How do you configure Quagga interactively?
(1) with an integrated user interface shell called vtysh.g
(2) Each daemon has it’s own configuration file and terminal interface. When you configure a static route, it must be done in zebra configuration file. When you configure BGP network it must be done in bgpd configuration file.
(3) telnet localhost (service name)

Exercise 9: RIP basics

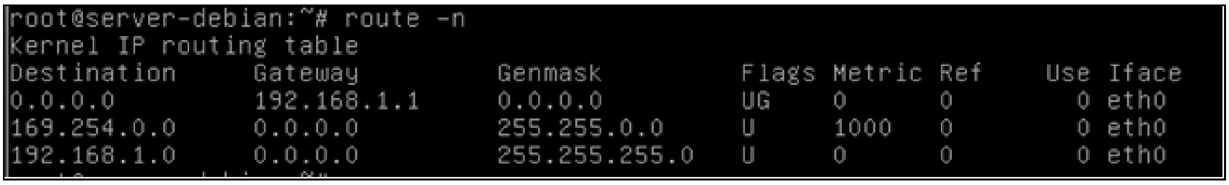
9-1
Explain, in broad terms, how RIP works.
路由信息协议（Routing Information Protocol，缩写：RIP）是一种使用最广泛的内部网关协议（IGP）。（IGP）是在内部网络上使用的路由协议(在少数情形下,也可以用于连接到因特网的网络)，它可以通过不断的交换信息让路由器动态的适应网络连接的变化，这些信息包括每个路由器可以到达哪些网络，这些网络有多远等。 IGP是应用层协议，并使用UDP作为传输协议。

9-2
What should your RIP router send out in the packets it sends, and to which networks?
content: prefix of the network; to: 10.0.0.0/24

Exercise 10: RIP configuration

10-1

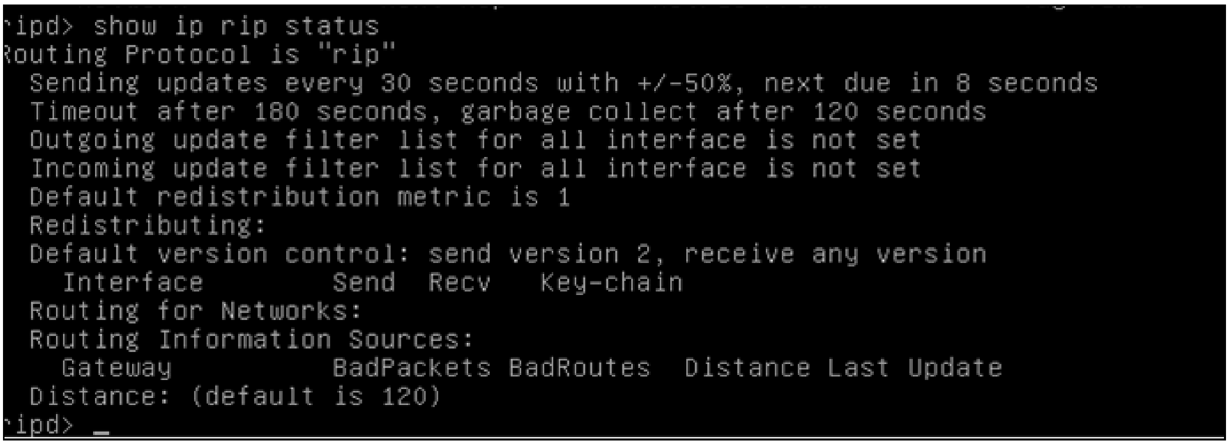
Enable RIP on the external interface of your router. Use RIP version 2 with no authentication (you may have to explicitly disable authentication). Select a sensible enable password.



```
root@server-debian:~# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         192.168.1.1     0.0.0.0         UG    0      0        0 eth0
169.254.0.0     0.0.0.0         255.255.0.0     U     1000   0        0 eth0
192.168.1.0     0.0.0.0         255.255.255.0   U     0      0        0 eth0
```

10-2

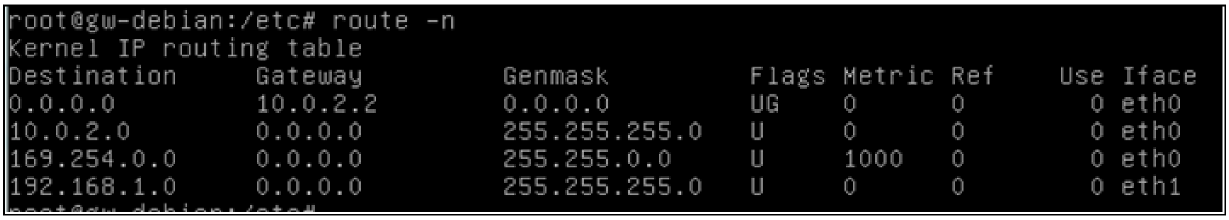
(Re)start all quagga servers, wait a few seconds and examine your routing tables. If you have successfully enabled ripd, you should see an additional default route and possibly one or more other routes.



```
ripd> show ip rip status
Routing Protocol is "rip"
  Sending updates every 30 seconds with +/-50%, next due in 8 seconds
  Timeout after 180 seconds, garbage collect after 120 seconds
  Outgoing update filter list for all interface is not set
  Incoming update filter list for all interface is not set
  Default redistribution metric is 1
  Redistributing:
  Default version control: send version 2, receive any version
    Interface        Send  Recv   Key-chain
  Routing for Networks:
  Routing Information Sources:
    Gateway          BadPackets BadRoutes  Distance Last Update
  Distance: (default is 120)
ripd> _
```

10-3

Connect to ripd interactively and run show ip rip status to display the current RIP status. You should see at least one routing information source.



```
root@gw-debian:/etc# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         10.0.2.2        0.0.0.0         UG    0      0        0 eth0
10.0.2.0        0.0.0.0         255.255.255.0   U     0      0        0 eth0
169.254.0.0     0.0.0.0         255.255.0.0     U     1000   0        0 eth0
192.168.1.0     0.0.0.0         255.255.255.0   U     0      0        0 eth1
```

Exercise 11: RIP announcements

11-1

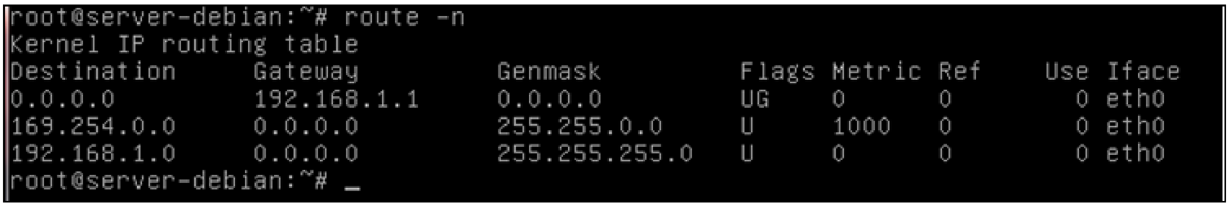
Configure ripd to announce your network prefix.

11-2

(Re)start all quagga servers and wait a few seconds. Check that your router is still accepting announcements from other servers by using show ip rip status.

11-3

Connect to the ripd port on 130.236.189.1 using telnet and check that it has your router as a routing information source. Connect to the zebra port and check that it has your prefix in its routing table.

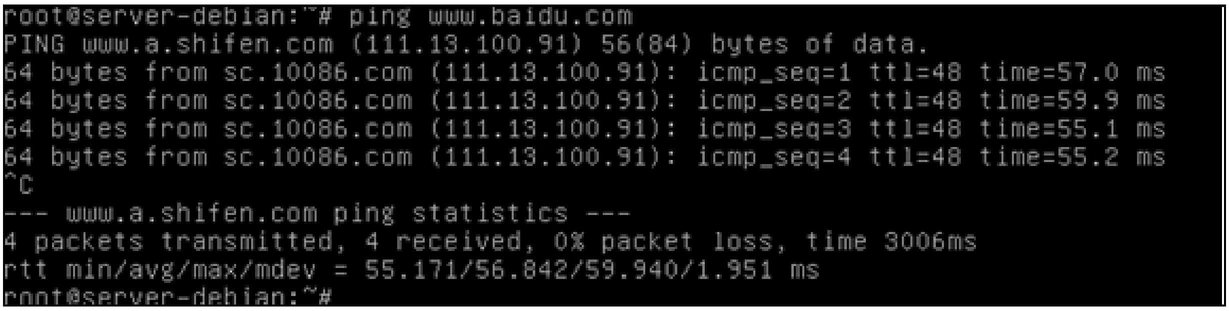


```
root@server-debian:~# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         192.168.1.1     0.0.0.0         UG    0      0        0 eth0
169.254.0.0     0.0.0.0         255.255.0.0     U     1000   0        0 eth0
192.168.1.0     0.0.0.0         255.255.255.0   U     0      0        0 eth0
root@server-debian:~# _
```

Exercise 12: Final checks

12-1

Test that you have full network connectivity to your gateway and clients.



```
root@server-debian:~# ping www.baidu.com
PING www.a.shifen.com (111.13.100.91) 56(84) bytes of data.
64 bytes from sc.10086.com (111.13.100.91): icmp_seq=1 ttl=48 time=57.0 ms
64 bytes from sc.10086.com (111.13.100.91): icmp_seq=2 ttl=48 time=59.9 ms
64 bytes from sc.10086.com (111.13.100.91): icmp_seq=3 ttl=48 time=55.1 ms
64 bytes from sc.10086.com (111.13.100.91): icmp_seq=4 ttl=48 time=55.2 ms
^C
--- www.a.shifen.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3006ms
rtt min/avg/max/mdev = 55.171/56.842/59.940/1.951 ms
root@server-debian:~#
```

Exercise 13: Intermediate RIP configuration (optional)

This exercise is optional, but might save you a lot of trouble later. If you do not do this, it is possible that another group’s configuration errors might prevent you from accessing the network. You should use the ip prefix-list command for this.

13-1

Configure your router to filter incoming announcements (use a prefix list). You should

(a) Accept routes for 10.17.1.0/24 (but not longer prefixes).

(b) Accept routes for the other groups’ address space.

(c) Not accept anything else.

bash # if other groups’ networks are like 10.17.2.0 or 10.17.254.0 ip prefix-list allowList permit 10.17.0.0 ge 16 ip prefix-list denyList deny 10.17.1.0 ge 25

13-2

Configure your RIP router to announce a default route on your internal network only. This may entail setting up a distribute list for announcements (again, using a prefix list).

bash # didn’t find the prefix-way and here’s the bgp way bgp router-id 192.168.1.1