

实验三 子网划分、静态路由和 NAT

141220132 银琦 141220132@smail.nju.edu.cn

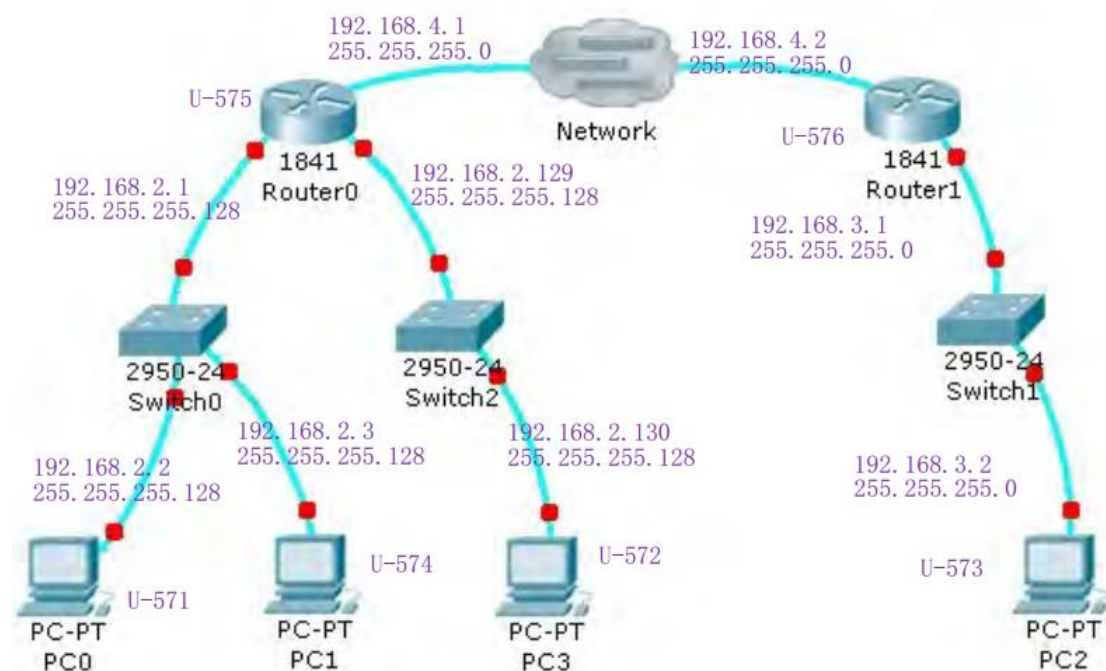
实验目的

本实验的主要目的是让学生能熟练地按照需求配置一个静态的包含多个子网的网络环境，并学会 NAT 的组网方式，为以后的实验过程中对组网的要求打下基础。

网络拓扑配置

节点名	虚拟设备名	Ip	Netmask
Router0	U-571	Eth0:192.168.2.1	255.255.255.128
		Eth1:192.168.2.129	255.255.255.128
		Eth2:192.168.4.1	255.255.255.0
Router1	U-576	Eth0:192.168.3.1	255.255.255.0
		Eth1:192.168.4.2	255.255.255.0
PC0	U-572	192.168.2.2	255.255.255.128
PC1	U-575	192.168.2.3	255.255.255.128
PC2	U-574	192.168.3.2	255.255.255.0
PC3	U-573	192.168.2.130	255.255.255.128

网络拓扑配置图



在打开每个虚拟机设置之前输入 `sudo service network-manager stop`

设置 IP 及网关

PC0:

```
sudo ifconfig eth0 192.168.2.2 netmask 255.255.255.128
```

```
sudo route add default gw 192.168.2.1
```

PC1:

```
sudo ifconfig eth0 192.168.2.3 netmask 255.255.255.128
```

```
sudo route add default gw 192.168.2.1
```

PC2:

```
sudo ifconfig eth0 192.168.3.2 netmask 255.255.255.0
```

```
sudo route add default gw 192.168.3.1
```

PC3:

```
sudo ifconfig eth0 192.168.2.130 netmask 255.255.255.128
```

```
sudo route add default gw 192.168.2.129
```

router0

```
sudo ifconfig eth0 192.168.4.1 netmask 255.255.255.0
```

```
sudo ifconfig eth1 192.168.2.1 netmask 255.255.255.128
```

```
sudo ifconfig eth2 192.168.2.129 netmask 255.255.255.128
```

router1

```
sudo ifconfig eth0 192.168.4.2 netmask 255.255.255.0
```

```
sudo ifconfig eth1 192.168.3.1 netmask 255.255.255.0
```

路由规则配置及 NAT 设置命令

router0

```
sudo iptables -t nat -A POSTROUTING -o eth0 -s 192.168.2.0/24 -j SNAT  
--to 210.28.130.166
```

```
sudo ip route add 192.168.3.0/24 via 192.168.4.2
```

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

router1

```
sudo ip route add 210.28.130.166 via 192.168.4.1
```

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

设置完成之后在每个虚拟机中输入 route 和 ip route 得到结果如下：

PC0

```
user@ubuntu: ~  
user@ubuntu:~$ route  
Kernel IP routing table  
Destination      Gateway          Genmask          Flags Metric Ref    Use Iface  
default          192.168.2.1     0.0.0.0          UG      0      0      0 eth0  
192.168.2.0      *               255.255.255.128 U        0      0      0 eth0  
user@ubuntu:~$ ip route  
default via 192.168.2.1 dev eth0  
192.168.2.0/25 dev eth0 proto kernel scope link src 192.168.2.2  
user@ubuntu:~$
```

PC1

```
user@ubuntu:~$ route  
Kernel IP routing table  
Destination      Gateway          Genmask          Flags Metric Ref    Use Iface  
default          192.168.2.1     0.0.0.0          UG      0      0      0 eth0  
192.168.2.0      *               255.255.255.128 U        0      0      0 eth0  
user@ubuntu:~$ ip route  
default via 192.168.2.1 dev eth0  
192.168.2.0/25 dev eth0 proto kernel scope link src 192.168.2.3  
user@ubuntu:~$
```

PC2

```
user@ubuntu:~$ route  
Kernel IP routing table  
Destination      Gateway          Genmask          Flags Metric Ref    Use Iface  
default          192.168.3.1     0.0.0.0          UG      0      0      0 eth0  
192.168.3.0      *               255.255.255.0    U        0      0      0 eth0  
user@ubuntu:~$ ip route  
default via 192.168.3.1 dev eth0  
192.168.3.0/24 dev eth0 proto kernel scope link src 192.168.3.2  
user@ubuntu:~$
```

PC3

```
user@ubuntu: ~  
user@ubuntu:~$ route  
Kernel IP routing table  
Destination      Gateway          Genmask          Flags Metric Ref    Use Iface  
default          192.168.2.129   0.0.0.0          UG      0      0      0 eth0  
192.168.2.128    *               255.255.255.128 U        0      0      0 eth0  
user@ubuntu:~$ ip route  
default via 192.168.2.129 dev eth0  
192.168.2.128/25 dev eth0 proto kernel scope link src 192.168.2.130  
user@ubuntu:~$
```

router0

```
user@ubuntu: ~  
user@ubuntu:~$ route  
Kernel IP routing table  
Destination      Gateway          Genmask          Flags Metric Ref    Use Iface  
192.168.2.0      *                255.255.255.128 U        0      0      0 eth1  
192.168.2.128    *                255.255.255.128 U        0      0      0 eth2  
192.168.3.0      192.168.4.2     255.255.255.0   UG       0      0      0 eth0  
192.168.4.0      *                255.255.255.0   U        0      0      0 eth0  
user@ubuntu:~$ ip route  
192.168.2.0/25 dev eth1 proto kernel scope link src 192.168.2.1  
192.168.2.128/25 dev eth2 proto kernel scope link src 192.168.2.129  
192.168.3.0/24 via 192.168.4.2 dev eth0  
192.168.4.0/24 dev eth0 proto kernel scope link src 192.168.4.1  
user@ubuntu:~$
```

router1

```
user@ubuntu:~$ route  
Kernel IP routing table  
Destination      Gateway          Genmask          Flags Metric Ref    Use Iface  
default          192.168.4.1     0.0.0.0          UG       0      0      0 eth1  
192.168.3.0      *                255.255.255.0   U        0      0      0 eth0  
192.168.4.0      *                255.255.255.0   U        0      0      0 eth1  
210.28.130.166  192.168.4.1     255.255.255.255 UGH      0      0      0 eth1  
user@ubuntu:~$ ip route  
default via 192.168.4.1 dev eth1  
192.168.3.0/24 dev eth0 proto kernel scope link src 192.168.3.1  
192.168.4.0/24 dev eth1 proto kernel scope link src 192.168.4.2  
210.28.130.166 via 192.168.4.1 dev eth1  
user@ubuntu:~$ _
```

数据包截图

PC0 ping PC1

No.	Time	Source	Destination	Protocol	Length	Info
4	1.004043	192.168.2.3	192.168.2.2	ICMP	98	Echo (ping) reply id=0x0aa5, seq=2/512, ttl=64
5	2.006060	192.168.2.2	192.168.2.3	ICMP	98	Echo (ping) request id=0x0aa5, seq=3/768, ttl=64
6	2.007611	192.168.2.3	192.168.2.2	ICMP	98	Echo (ping) reply id=0x0aa5, seq=3/768, ttl=64
7	3.010440	192.168.2.2	192.168.2.3	ICMP	98	Echo (ping) request id=0x0aa5, seq=4/1024, ttl=64
8	3.011415	192.168.2.3	192.168.2.2	ICMP	98	Echo (ping) reply id=0x0aa5, seq=4/1024, ttl=64
9	4.013755	192.168.2.2	192.168.2.3	ICMP	98	Echo (ping) request id=0x0aa5, seq=5/1280, ttl=64
10	4.015569	192.168.2.3	192.168.2.2	ICMP	98	Echo (ping) reply id=0x0aa5, seq=5/1280, ttl=64
11	5.004923	Vmware_c0:65:21	Vmware_95:57:cf	ARP	60	Who has 192.168.2.2? Tell 192.168.2.3
12	5.004969	Vmware_95:57:cf	Vmware_c0:65:21	ARP	42	192.168.2.2 is at 00:0c:29:95:57:cf
13	5.017242	192.168.2.2	192.168.2.3	ICMP	98	Echo (ping) request id=0x0aa5, seq=6/1536, ttl=64
14	5.018642	192.168.2.3	192.168.2.2	ICMP	98	Echo (ping) reply id=0x0aa5, seq=6/1536, ttl=64
15	6.018370	192.168.2.2	192.168.2.3	ICMP	98	Echo (ping) request id=0x0aa5, seq=7/1792, ttl=64
16	6.023014	192.168.2.3	192.168.2.2	ICMP	98	Echo (ping) reply id=0x0aa5, seq=7/1792, ttl=64

* Frame 1: 98 bytes on wire (784 bits), 98 bytes captured (784 bits)

Ethernet II, Src: Vmware_95:57:cf (00:0c:29:95:57:cf), Dst: Vmware_c0:65:21 (00:0c:29:c0:65:21)

Internet Protocol Version 4, Src: 192.168.2.2 (192.168.2.2), Dst: 192.168.2.3 (192.168.2.3)

Internet Control Message Protocol

0000	00 0c 29 c0 65 21 00 0c 29 95 57 cf 08 00 45 00	...ef..).W...E.
0010	00 54 00 00 40 00 40 01 b5 53 c0 a8 02 02 c0 a8	.T...@...S.....
0020	02 03 08 00 20 4b 0a a5 00 01 db e2 fd 56 fb d1	...K...V...
0030	0d 00 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15
0040	16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25!#\$%
0050	26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35	S'()*+,-./012345
0060	36 37	67

PC0 ping PC2 (在 router0 中使用 wireshark 抓包)

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	210.28.130.166	192.168.3.2	ICMP	98	Echo (ping) request id=0x0889, seq=15/3840, ttl=63
2	0.001393	192.168.3.2	210.28.130.166	ICMP	98	Echo (ping) reply id=0x0889, seq=15/3840, ttl=63
3	1.002203	210.28.130.166	192.168.3.2	ICMP	98	Echo (ping) request id=0x0889, seq=16/4096, ttl=63
4	1.003180	192.168.3.2	210.28.130.166	ICMP	98	Echo (ping) reply id=0x0889, seq=16/4096, ttl=63
5	2.006359	210.28.130.166	192.168.3.2	ICMP	98	Echo (ping) request id=0x0889, seq=17/4352, ttl=63
6	2.007561	192.168.3.2	210.28.130.166	ICMP	98	Echo (ping) reply id=0x0889, seq=17/4352, ttl=63
7	3.010403	210.28.130.166	192.168.3.2	ICMP	98	Echo (ping) request id=0x0889, seq=18/4608, ttl=63
8	3.011859	192.168.3.2	210.28.130.166	ICMP	98	Echo (ping) reply id=0x0889, seq=18/4608, ttl=63
9	4.014422	210.28.130.166	192.168.3.2	ICMP	98	Echo (ping) request id=0x0889, seq=19/4864, ttl=63
10	4.015775	192.168.3.2	210.28.130.166	ICMP	98	Echo (ping) reply id=0x0889, seq=19/4864, ttl=63
11	5.018359	210.28.130.166	192.168.3.2	ICMP	98	Echo (ping) request id=0x0889, seq=20/5120, ttl=63
12	5.019839	192.168.3.2	210.28.130.166	ICMP	98	Echo (ping) reply id=0x0889, seq=20/5120, ttl=63
13	6.022267	210.28.130.166	192.168.3.2	ICMP	98	Echo (ping) request id=0x0889, seq=21/5376, ttl=63
Frame 1: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface 0						
Ethernet II, Src: Vmware_95:57:cf (00:0c:29:95:57:cf), Dst: Vmware_5e:4e:61 (00:0c:29:5e:4e:61)						
Internet Protocol Version 4, Src: 210.28.130.166 (210.28.130.166), Dst: 192.168.3.2 (192.168.3.2)						
Internet Control Message Protocol						
0000 00 0c 29 5e 4e 61 00 0c 29 95 57 cf 00 00 45 00 ..)Na..).W...E.						
0010 00 54 00 00 00 40 00 3f 01 23 3c d2 1c 82 a6 c0 a8 .T..@.?.#<.....						
0020 03 02 08 00 e5 be 08 89 00 0f 0b 20 05 57 06 2fW./						
0030 08 00 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15!#\$%						
0040 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25^_`{~						
0050 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35 9'()*+,-./012345						
0060 36 3767						

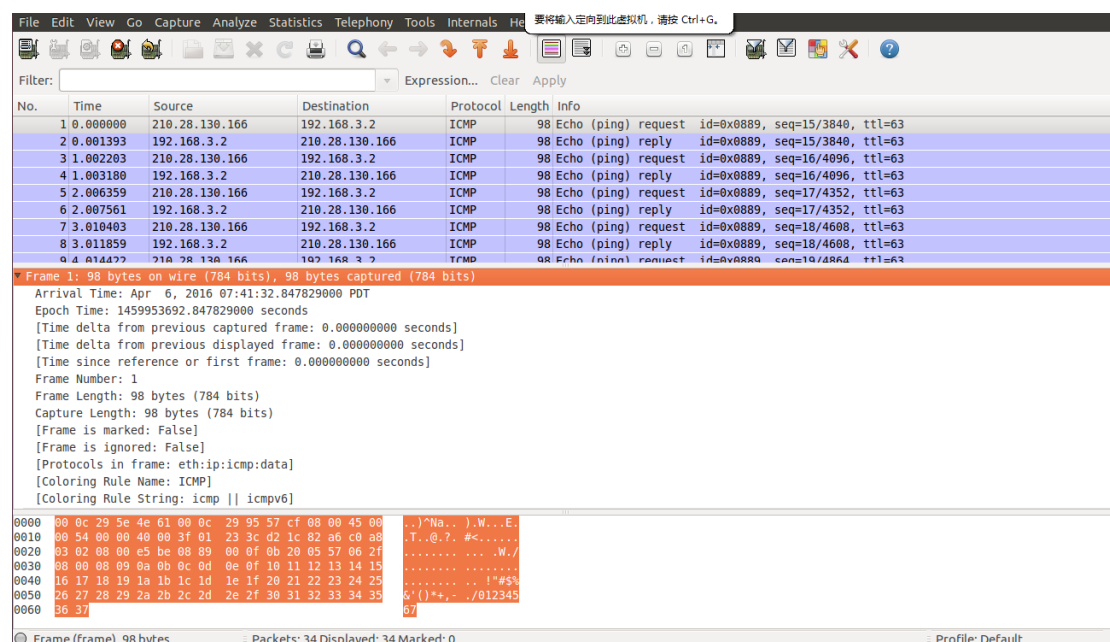
PC0 ping PC3

No.	Time	Source	Destination	Protocol	Length	Info
8	2.005052	192.168.2.130	192.168.2.2	ICMP	98	Echo (ping) reply id=0x0aaa, seq=3/768, ttl=63
9	3.008008	192.168.2.2	192.168.2.130	ICMP	98	Echo (ping) request id=0x0aaa, seq=4/1024, ttl=64
10	3.010705	192.168.2.130	192.168.2.2	ICMP	98	Echo (ping) reply id=0x0aaa, seq=4/1024, ttl=63
11	4.012491	192.168.2.2	192.168.2.130	ICMP	98	Echo (ping) request id=0x0aaa, seq=5/1280, ttl=64
12	4.014155	192.168.2.130	192.168.2.2	ICMP	98	Echo (ping) reply id=0x0aaa, seq=5/1280, ttl=63
13	5.012926	Vmware_6f:78:c9	Vmware_95:57:cf	ARP	60	Who has 192.168.2.2? Tell 192.168.2.1
14	5.012987	Vmware_95:57:cf	Vmware_6f:78:c9	ARP	42	192.168.2.2 is at 00:0c:29:95:57:cf
15	5.016819	192.168.2.2	192.168.2.130	ICMP	98	Echo (ping) request id=0x0aaa, seq=6/1536, ttl=64
16	5.019375	192.168.2.130	192.168.2.2	ICMP	98	Echo (ping) reply id=0x0aaa, seq=6/1536, ttl=63
17	6.021065	192.168.2.2	192.168.2.130	ICMP	98	Echo (ping) request id=0x0aaa, seq=7/1792, ttl=64
18	6.023066	192.168.2.130	192.168.2.2	ICMP	98	Echo (ping) reply id=0x0aaa, seq=7/1792, ttl=63
19	7.024736	192.168.2.2	192.168.2.130	ICMP	98	Echo (ping) request id=0x0aaa, seq=8/2048, ttl=64
20	7.029011	192.168.2.130	192.168.2.2	ICMP	98	Echo (ping) reply id=0x0aaa, seq=8/2048, ttl=63
Frame 1: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0						
Ethernet II, Src: Vmware_95:57:cf (00:0c:29:95:57:cf), Dst: Broadcast (ff:ff:ff:ff:ff:ff)						
Address Resolution Protocol (request)						
0000 ff ff ff ff ff ff 00 0c 29 95 57 cf 00 06 00 01).W....						
0010 08 00 06 04 00 01 00 0c 29 95 57 cf c0 a8 02 02).W....						
0020 00 00 00 00 00 00 c0 a8 02 01:						

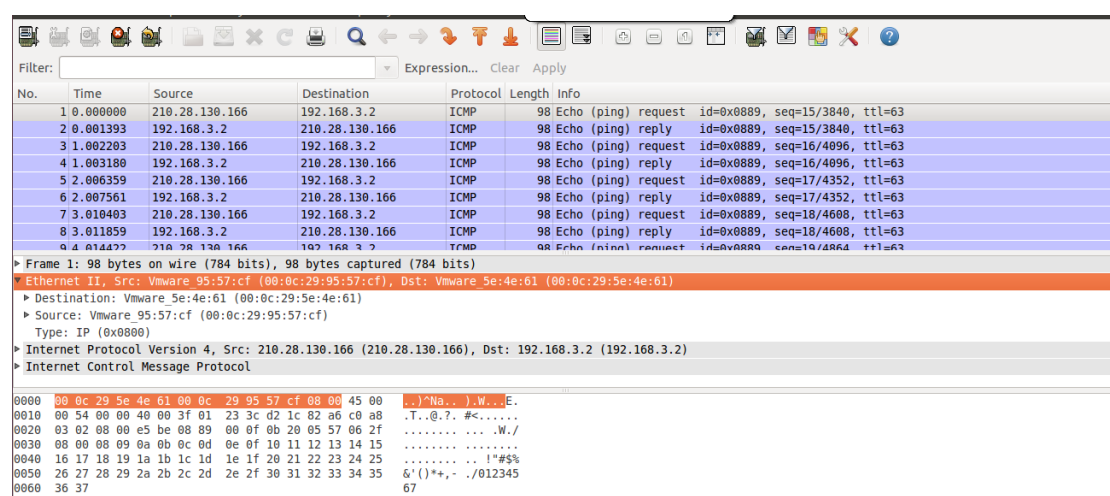
协议报文分析

在 PC0 中 ping 192.168.3.2: 若在 PC0 中抓包, 则地址没有被替换, 在 router0 中抓包才会体现出地址的替换。图中分为 3 栏, 第一栏为列表框, 第二栏为协议框, 第三栏为原始框, 首先在列表框中选择一个 request 项, 在协议框中选择第一个, 物理层帧, 在原始框中会选定所有数据。

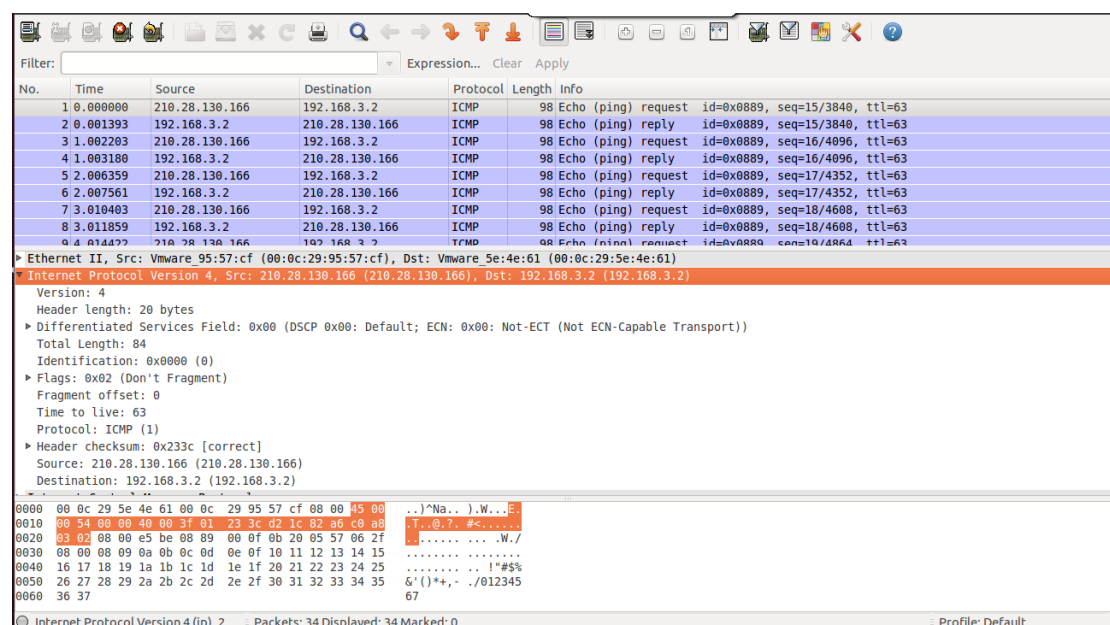
这里显示了帧的基本信息, 包括接收时间、帧编号、大小等。



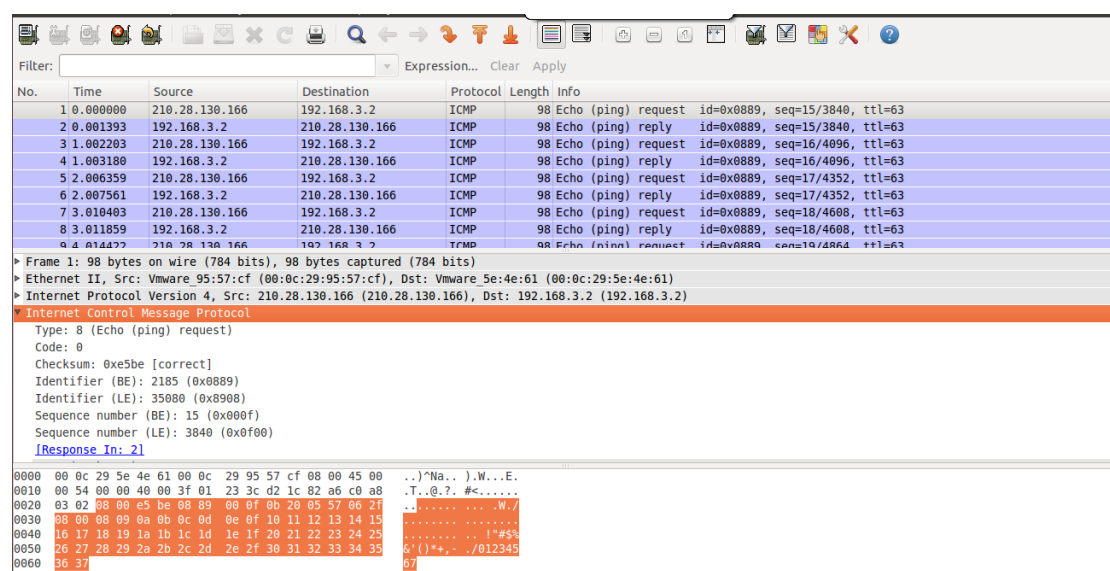
接下来选择协议框中的第二个：以太网帧及其首部，在原始框中选定相关的数据如下。这里说明了帧的源地址，目的地址以及帧类型。



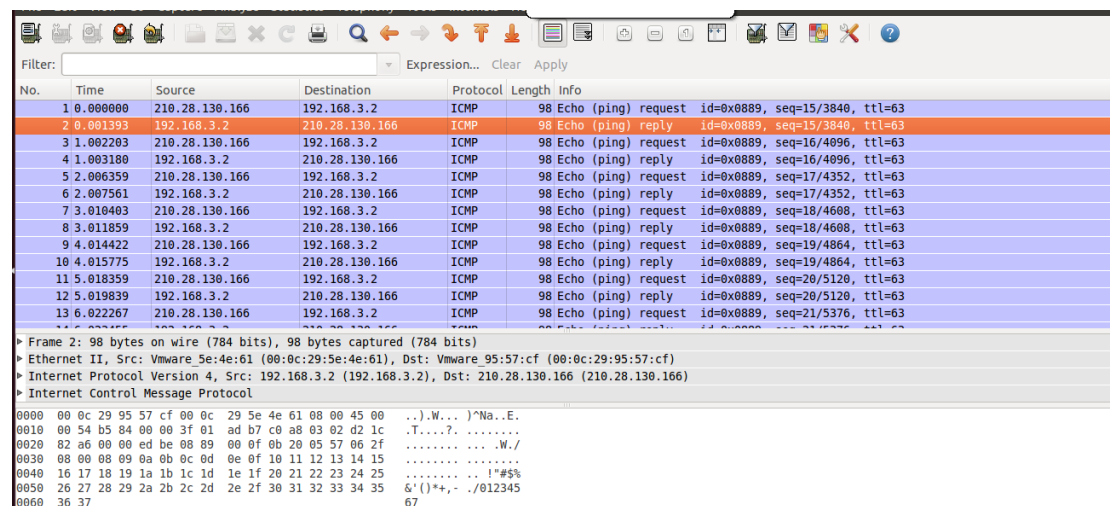
选定协议框中的第三项 IP 协议数据报及其首部，在原始框中也会选定相应数据如下。这里说明了 ip 数据报的大小、类型及其首部的地址等信息。



选定协议框中的第四项 Internet 控制报文协议，则结果如下。



选择列表框中 reply 一项，在协议框中也有四项，类似第一种情况：



遇到的问题

在进行 NAT 设置之后，我觉得理论上在用 PC0 ping PC2 的时候，Src 的地址不应该是 192.168.2.2，而应该是 210.28.130.166，但是我的 Src 的地址还是替换之前的，我在 Router 0 中使用 `iptables -L` 命令进行了查询，发现的确进行了 NAT 设置。

以上问题为第一次做实验时候出现的问题，后来，询问了大神才知道，在 router0 中抓包才可以看到地址的替换情况。

除此之外，我发现 eth0，eth1，eth2 与 net 的对应是有顺序的，一开始没有考虑顺序，结果 ping 不通，后来按照顺序设置了，就 ping 通了。