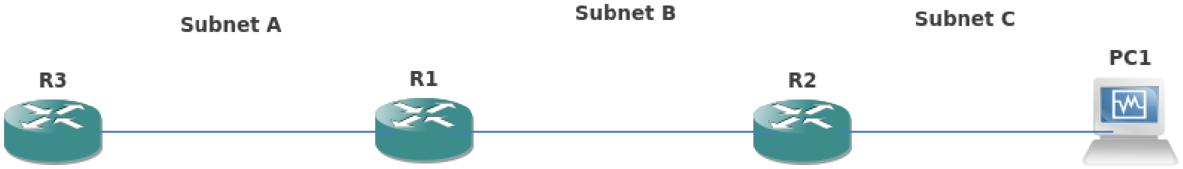
Sean McGrath

CME151 Double NAT Lab

All of the instructions for this lab are given in plain black text and each question in italicized red text. Our answers which serve as the answer key are in plain red text and include the appropriate images and tables.

Step 1:

Topology 1 A



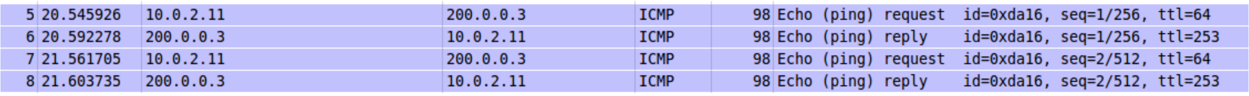
Setup topology 1 as shown above, with R3 and R1 connected via a public network. Configure R1 and R2 to both run NAT between their subnets.

1. *Include your addressing plan in the table below*

	Subnet A	Subnet B	Subnet C
IP prefix	200.0.0.0/24	10.0.1.0/24	10.0.2.0/24
R3	200.0.0.3		
R1	200.0.0.1	10.0.1.1	
R2		10.0.1.2	10.0.2.2
PC1			10.0.2.11

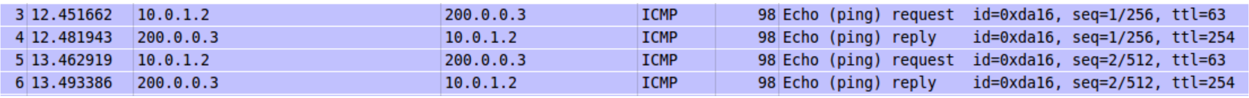
2 Ping R3 from PC1 and demonstrate connectivity using command output and packet captures.

Capture on R2’s fa1/0 interface:



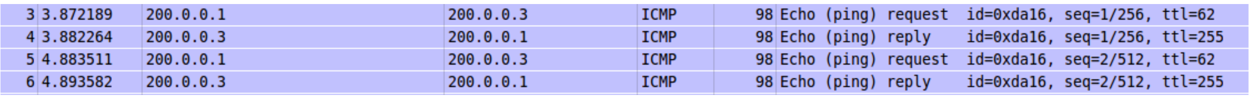
```
5 20.545926  10.0.2.11   200.0.0.3   ICMP   98 Echo (ping) request  id=0xda16, seq=1/256, ttl=64
6 20.592278  200.0.0.3   10.0.2.11   ICMP   98 Echo (ping) reply    id=0xda16, seq=1/256, ttl=253
7 21.561705  10.0.2.11   200.0.0.3   ICMP   98 Echo (ping) request  id=0xda16, seq=2/512, ttl=64
8 21.603735  200.0.0.3   10.0.2.11   ICMP   98 Echo (ping) reply    id=0xda16, seq=2/512, ttl=253
```

Capture on R1’s fa1/0 interface:



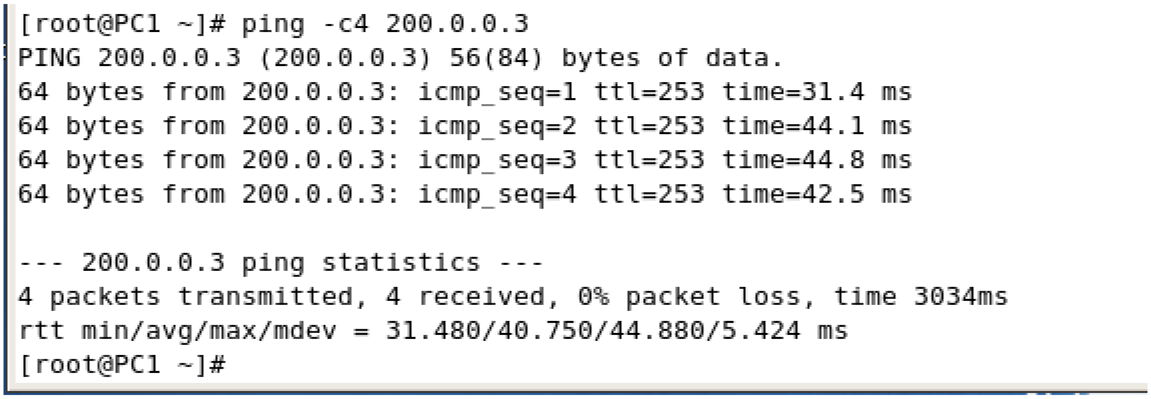
```
3 12.451662  10.0.1.2    200.0.0.3   ICMP   98 Echo (ping) request  id=0xda16, seq=1/256, ttl=63
4 12.481943  200.0.0.3   10.0.1.2    ICMP   98 Echo (ping) reply    id=0xda16, seq=1/256, ttl=254
5 13.462919  10.0.1.2    200.0.0.3   ICMP   98 Echo (ping) request  id=0xda16, seq=2/512, ttl=63
6 13.493386  200.0.0.3   10.0.1.2    ICMP   98 Echo (ping) reply    id=0xda16, seq=2/512, ttl=254
```

Capture on R3’s fa0/0 interface:



```
 3 3.872189  200.0.0.1      200.0.0.3    ICMP      98 Echo (ping) request  id=0xda16, seq=1/256, ttl=62
 4 3.882264  200.0.0.3      200.0.0.1    ICMP      98 Echo (ping) reply    id=0xda16, seq=1/256, ttl=255
 5 4.883511  200.0.0.1      200.0.0.3    ICMP      98 Echo (ping) request  id=0xda16, seq=2/512, ttl=62
 6 4.893582  200.0.0.3      200.0.0.1    ICMP      98 Echo (ping) reply    id=0xda16, seq=2/512, ttl=255
```

Output from *ping* on PC1:



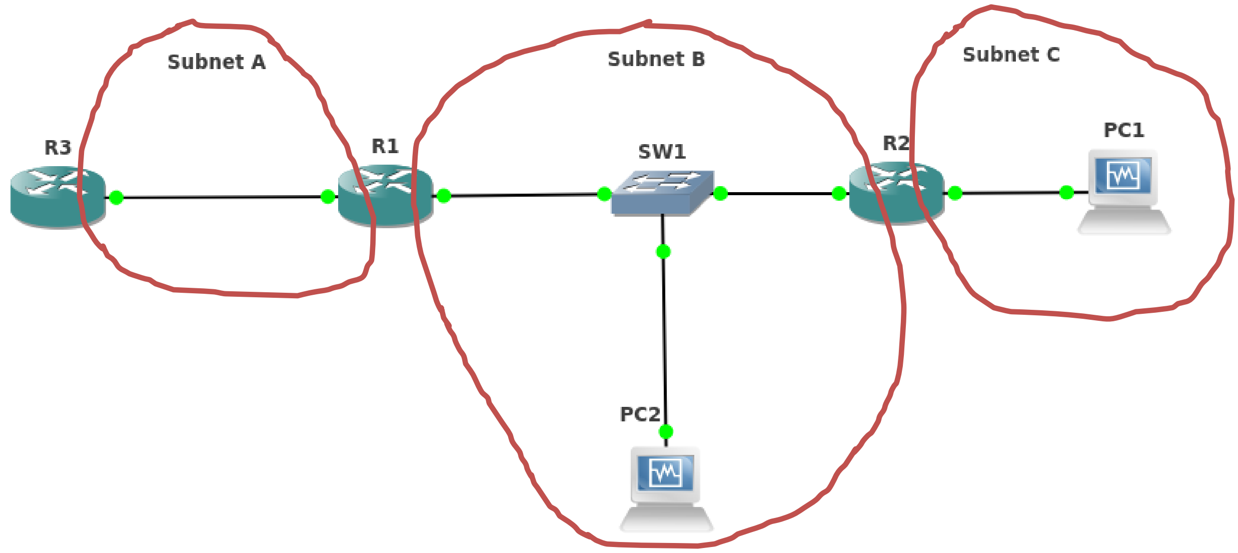
```
[root@PC1 ~]# ping -c4 200.0.0.3
PING 200.0.0.3 (200.0.0.3) 56(84) bytes of data.
64 bytes from 200.0.0.3: icmp_seq=1 ttl=253 time=31.4 ms
64 bytes from 200.0.0.3: icmp_seq=2 ttl=253 time=44.1 ms
64 bytes from 200.0.0.3: icmp_seq=3 ttl=253 time=44.8 ms
64 bytes from 200.0.0.3: icmp_seq=4 ttl=253 time=42.5 ms

--- 200.0.0.3 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3034ms
rtt min/avg/max/mdev = 31.480/40.750/44.880/5.424 ms
[root@PC1 ~]#
```

3 *Explain the address translation process as the ICMP packet is transported between each subnet. Include which node(s) R3 and R1 are aware of during this process. Support your answer with packet captures and NAT translation information.*

PC1 sends the ICMP packets destined for 200.0.0.3 to R2, its default gateway. Because R2 is running NAT, it consults its NAT translation table for the Inside Local address of the source (10.0.2.11). Now R2 knows to translate this address to 10.0.1.2 when it forwards the packet to its next hop. Now when the packet arrives at R1, R1 sees the source of the message as 10.0.2.2 (see capture from question 2), and performs the proper NAT translation before forwarding the packet to the 200.0.0.0 subnet. Finally, the message arrives at R3 with a source address of 200.0.0.1 (see capture from question 2), for which R3 is able to send a response to. Notice that R3 is only ever aware that it is communicating with R1, as its IP address is seen as the source. Similarly, R1 is unaware that the message originated from PC1, as it sees the source IP as that of R2.

Step 2:

Topology 1 B

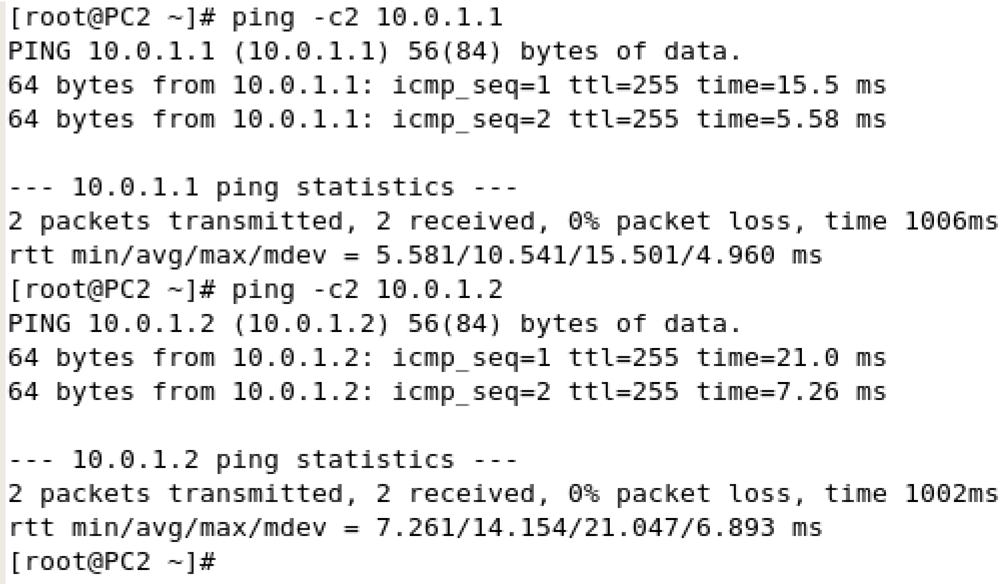


Now connect a second PC to R1 on Subnet B. Configure R1 to perform an appropriate NAT translation to map PC2's IP address onto the public subnet (Hint: you must translate PC2's IP to a different 192.168.0.0/24 address than R2, or configure Port Address Translation).

1. *Include addressing plan to include PC2 and demonstrate connectivity between the nodes of Subnet B with ping output and/or packet captures.*

	Subnet A	Subnet B	Subnet C
IP prefix	200.0.0.0/24	10.0.1.0/24	10.0.2.0/24
R3	200.0.0.3		
R1	200.0.0.1	10.0.1.1	
R2		10.0.1.2	10.0.2.2
PC1			10.0.2.11
PC2		10.0.1.22	

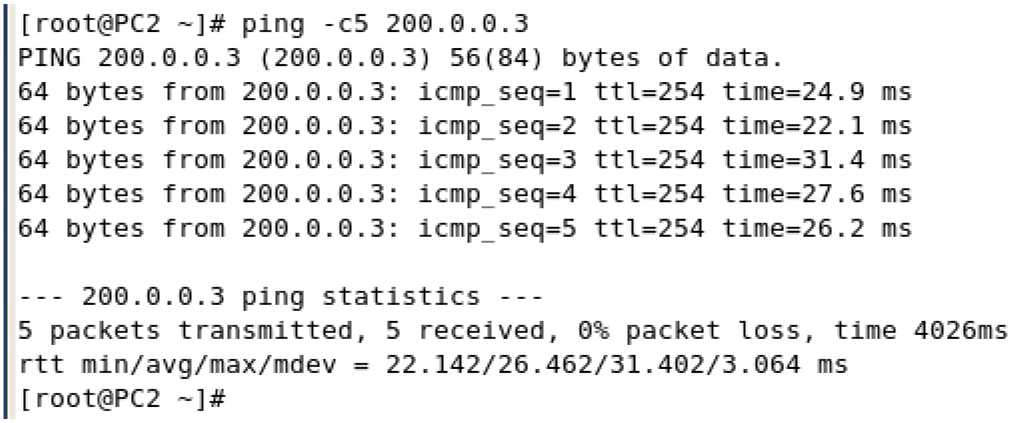
Ping output showing connectivity among Subnet B:



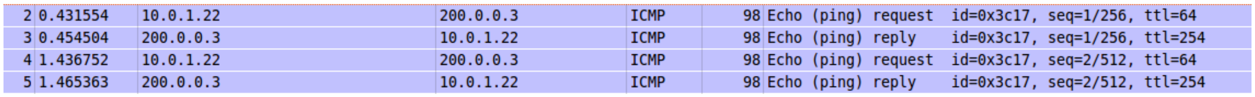
```
[root@PC2 ~]# ping -c2 10.0.1.1
PING 10.0.1.1 (10.0.1.1) 56(84) bytes of data.
64 bytes from 10.0.1.1: icmp_seq=1 ttl=255 time=15.5 ms
64 bytes from 10.0.1.1: icmp_seq=2 ttl=255 time=5.58 ms

--- 10.0.1.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1006ms
rtt min/avg/max/mdev = 5.581/10.541/15.501/4.960 ms
[root@PC2 ~]# ping -c2 10.0.1.2
PING 10.0.1.2 (10.0.1.2) 56(84) bytes of data.
64 bytes from 10.0.1.2: icmp_seq=1 ttl=255 time=21.0 ms
64 bytes from 10.0.1.2: icmp_seq=2 ttl=255 time=7.26 ms

--- 10.0.1.2 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 7.261/14.154/21.047/6.893 ms
[root@PC2 ~]#
```

2. *Issue a ping from PC2 to R3 and demonstrate NAT configuration via packet captures and NAT information (if NAT overload is not being used).*

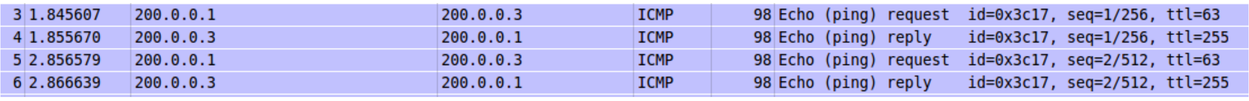
Here we demonstrate successful communication between PC2 and R3 via *ping* output and packet captures:



```
[root@PC2 ~]# ping -c5 200.0.0.3
PING 200.0.0.3 (200.0.0.3) 56(84) bytes of data.
64 bytes from 200.0.0.3: icmp_seq=1 ttl=254 time=24.9 ms
64 bytes from 200.0.0.3: icmp_seq=2 ttl=254 time=22.1 ms
64 bytes from 200.0.0.3: icmp_seq=3 ttl=254 time=31.4 ms
64 bytes from 200.0.0.3: icmp_seq=4 ttl=254 time=27.6 ms
64 bytes from 200.0.0.3: icmp_seq=5 ttl=254 time=26.2 ms

--- 200.0.0.3 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4026ms
rtt min/avg/max/mdev = 22.142/26.462/31.402/3.064 ms
[root@PC2 ~]#
```



```
2 0.431554  10.0.1.22   200.0.0.3   ICMP   98 Echo (ping) request  id=0x3c17, seq=1/256, ttl=64
3 0.454504  200.0.0.3   10.0.1.22   ICMP   98 Echo (ping) reply    id=0x3c17, seq=1/256, ttl=254
4 1.436752  10.0.1.22   200.0.0.3   ICMP   98 Echo (ping) request  id=0x3c17, seq=2/512, ttl=64
5 1.465363  200.0.0.3   10.0.1.22   ICMP   98 Echo (ping) reply    id=0x3c17, seq=2/512, ttl=254
```

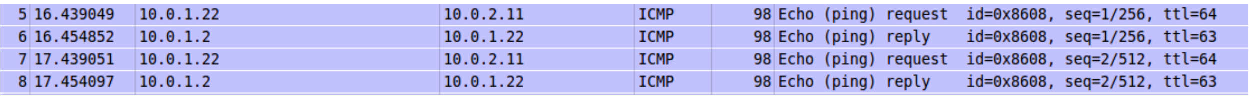


```
3 1.845607  200.0.0.1   200.0.0.3   ICMP   98 Echo (ping) request  id=0x3c17, seq=1/256, ttl=63
4 1.855670  200.0.0.3   200.0.0.1   ICMP   98 Echo (ping) reply    id=0x3c17, seq=1/256, ttl=255
5 2.856579  200.0.0.1   200.0.0.3   ICMP   98 Echo (ping) request  id=0x3c17, seq=2/512, ttl=63
6 2.866639  200.0.0.3   200.0.0.1   ICMP   98 Echo (ping) reply    id=0x3c17, seq=2/512, ttl=255
```

3. *Ping PC1 from PC2. Demonstrate connectivity via packet captures. Explain why it is not possible to have duplicate addresses exist on Subnet B and Subnet C. In other words, why is it that, as long as the hosts and routers are configured to know the next hop to the destination, that any host on Subnet B can communicate with any host on Subnet C, and vice versa?*

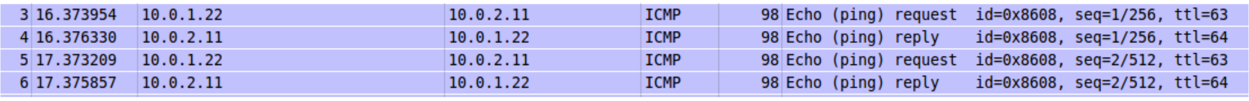
Packet captures from PC2 to PC1:

Capture on R2’s fa0/0 interface:



```
5 16.439049  10.0.1.22   10.0.2.11   ICMP   98 Echo (ping) request  id=0x8608, seq=1/256, ttl=64
6 16.454852  10.0.1.2    10.0.1.22   ICMP   98 Echo (ping) reply    id=0x8608, seq=1/256, ttl=63
7 17.439051  10.0.1.22   10.0.2.11   ICMP   98 Echo (ping) request  id=0x8608, seq=2/512, ttl=64
8 17.454097  10.0.1.2    10.0.1.22   ICMP   98 Echo (ping) reply    id=0x8608, seq=2/512, ttl=63
```

Capture on R2’s fa1/0 interface:

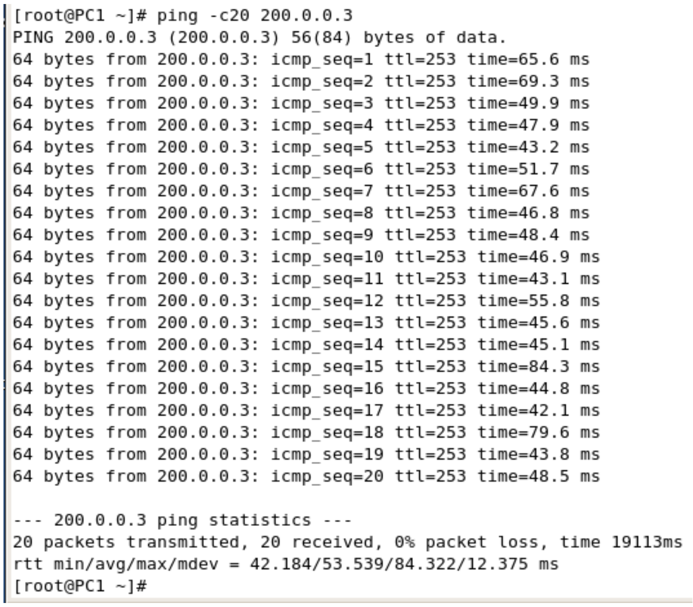


```
3 16.373954  10.0.1.22   10.0.2.11   ICMP   98 Echo (ping) request  id=0x8608, seq=1/256, ttl=63
4 16.376330  10.0.2.11   10.0.1.22   ICMP   98 Echo (ping) reply    id=0x8608, seq=1/256, ttl=64
5 17.373209  10.0.1.22   10.0.2.11   ICMP   98 Echo (ping) request  id=0x8608, seq=2/512, ttl=63
6 17.375857  10.0.2.11   10.0.1.22   ICMP   98 Echo (ping) reply    id=0x8608, seq=2/512, ttl=64
```

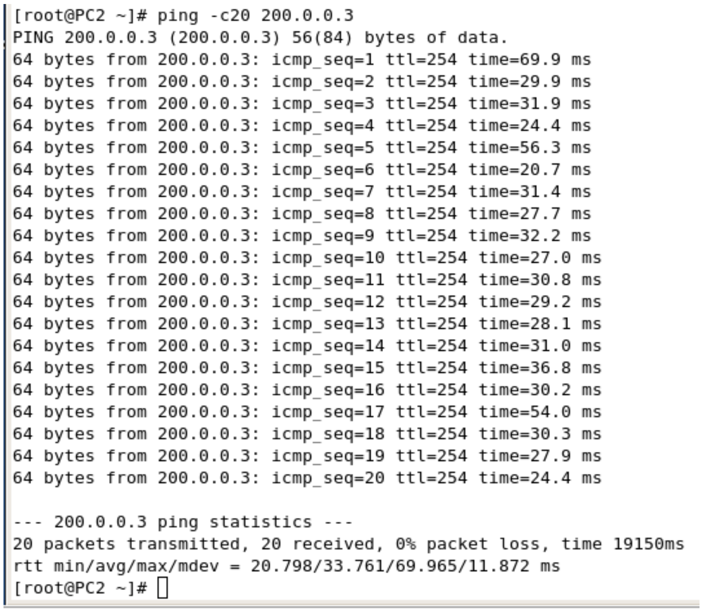
It is not possible for the nodes of Subnet B and Subnet C to be unable to communicate because the IP prefixes of the subnets on the different interfaces of R2 must be different. Because of this, the IP prefixes for each node on Subnet B are necessarily different from those on Subnet C, and communication can take place between the any devices on the two networks (assuming routing is properly configured and policy of each subnet allows for communication.)

4. *Issue simultaneous pings to R3 from PC1 and PC2. Analyze the transmission time of each link and determine if there is a significant delay added by going through a second NAT router.*

Below we issue two *pings* from PC1 and PC2 destined for PC3 simultaneously. The *ping* output is shown below:

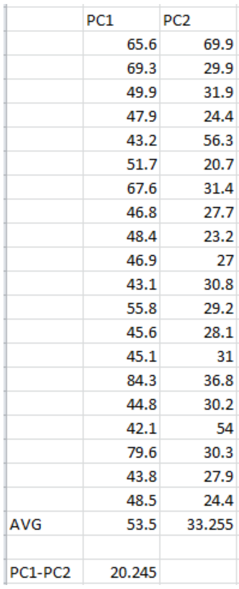


```
[root@PC1 ~]# ping -c20 200.0.0.3
PING 200.0.0.3 (200.0.0.3) 56(84) bytes of data.
64 bytes from 200.0.0.3: icmp_seq=1 ttl=253 time=65.6 ms
64 bytes from 200.0.0.3: icmp_seq=2 ttl=253 time=69.3 ms
64 bytes from 200.0.0.3: icmp_seq=3 ttl=253 time=49.9 ms
64 bytes from 200.0.0.3: icmp_seq=4 ttl=253 time=47.9 ms
64 bytes from 200.0.0.3: icmp_seq=5 ttl=253 time=43.2 ms
64 bytes from 200.0.0.3: icmp_seq=6 ttl=253 time=51.7 ms
64 bytes from 200.0.0.3: icmp_seq=7 ttl=253 time=67.6 ms
64 bytes from 200.0.0.3: icmp_seq=8 ttl=253 time=46.8 ms
64 bytes from 200.0.0.3: icmp_seq=9 ttl=253 time=48.4 ms
64 bytes from 200.0.0.3: icmp_seq=10 ttl=253 time=46.9 ms
64 bytes from 200.0.0.3: icmp_seq=11 ttl=253 time=43.1 ms
64 bytes from 200.0.0.3: icmp_seq=12 ttl=253 time=55.8 ms
64 bytes from 200.0.0.3: icmp_seq=13 ttl=253 time=45.6 ms
64 bytes from 200.0.0.3: icmp_seq=14 ttl=253 time=45.1 ms
64 bytes from 200.0.0.3: icmp_seq=15 ttl=253 time=84.3 ms
64 bytes from 200.0.0.3: icmp_seq=16 ttl=253 time=44.8 ms
64 bytes from 200.0.0.3: icmp_seq=17 ttl=253 time=42.1 ms
64 bytes from 200.0.0.3: icmp_seq=18 ttl=253 time=79.6 ms
64 bytes from 200.0.0.3: icmp_seq=19 ttl=253 time=43.8 ms
64 bytes from 200.0.0.3: icmp_seq=20 ttl=253 time=48.5 ms

--- 200.0.0.3 ping statistics ---
20 packets transmitted, 20 received, 0% packet loss, time 19113ms
rtt min/avg/max/mdev = 42.184/53.539/84.322/12.375 ms
[root@PC1 ~]#
```



```
[root@PC2 ~]# ping -c20 200.0.0.3
PING 200.0.0.3 (200.0.0.3) 56(84) bytes of data.
64 bytes from 200.0.0.3: icmp_seq=1 ttl=254 time=69.9 ms
64 bytes from 200.0.0.3: icmp_seq=2 ttl=254 time=29.9 ms
64 bytes from 200.0.0.3: icmp_seq=3 ttl=254 time=31.9 ms
64 bytes from 200.0.0.3: icmp_seq=4 ttl=254 time=24.4 ms
64 bytes from 200.0.0.3: icmp_seq=5 ttl=254 time=56.3 ms
64 bytes from 200.0.0.3: icmp_seq=6 ttl=254 time=20.7 ms
64 bytes from 200.0.0.3: icmp_seq=7 ttl=254 time=31.4 ms
64 bytes from 200.0.0.3: icmp_seq=8 ttl=254 time=27.7 ms
64 bytes from 200.0.0.3: icmp_seq=9 ttl=254 time=32.2 ms
64 bytes from 200.0.0.3: icmp_seq=10 ttl=254 time=27.0 ms
64 bytes from 200.0.0.3: icmp_seq=11 ttl=254 time=30.8 ms
64 bytes from 200.0.0.3: icmp_seq=12 ttl=254 time=29.2 ms
64 bytes from 200.0.0.3: icmp_seq=13 ttl=254 time=28.1 ms
64 bytes from 200.0.0.3: icmp_seq=14 ttl=254 time=31.0 ms
64 bytes from 200.0.0.3: icmp_seq=15 ttl=254 time=36.8 ms
64 bytes from 200.0.0.3: icmp_seq=16 ttl=254 time=30.2 ms
64 bytes from 200.0.0.3: icmp_seq=17 ttl=254 time=54.0 ms
64 bytes from 200.0.0.3: icmp_seq=18 ttl=254 time=30.3 ms
64 bytes from 200.0.0.3: icmp_seq=19 ttl=254 time=27.9 ms
64 bytes from 200.0.0.3: icmp_seq=20 ttl=254 time=24.4 ms

--- 200.0.0.3 ping statistics ---
20 packets transmitted, 20 received, 0% packet loss, time 19150ms
rtt min/avg/max/mdev = 20.798/33.761/69.965/11.872 ms
[root@PC2 ~]# 
```

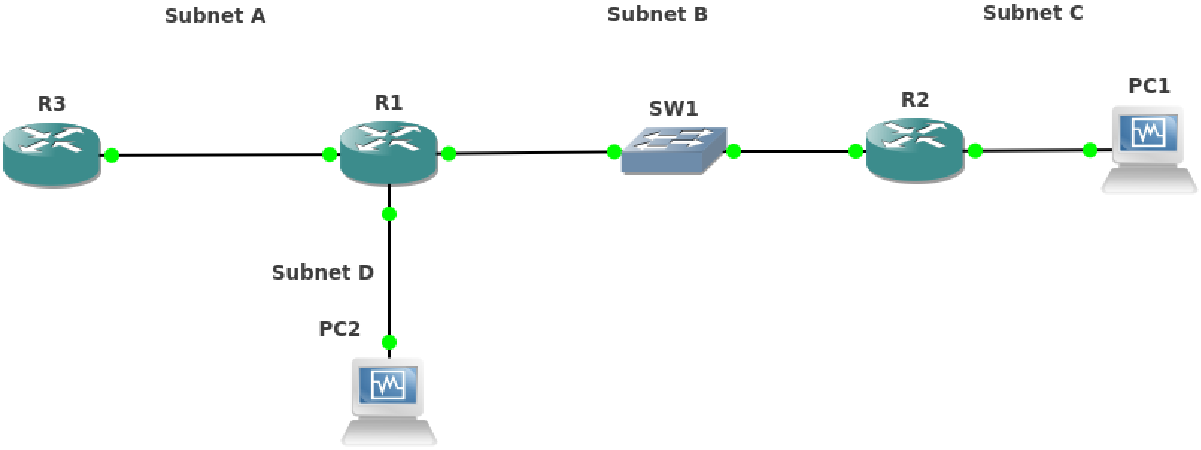
By preliminary inspection, it is clear that the time to receive ICMP packets back from R3 is significantly greater on PC1 than PC2. We wish to average the difference in delay, which we do via Excel. The result is shown below.



	PC1	PC2
	65.6	69.9
	69.3	29.9
	49.9	31.9
	47.9	24.4
	43.2	56.3
	51.7	20.7
	67.6	31.4
	46.8	27.7
	48.4	23.2
	46.9	27
	43.1	30.8
	55.8	29.2
	45.6	28.1
	45.1	31
	84.3	36.8
	44.8	30.2
	42.1	54
	79.6	30.3
	43.8	27.9
	48.5	24.4
AVG	53.5	33.255
PC1-PC2	20.245	

As is clear from our results, the average delay over 20 *pings* is 20.245 ms greater for PC1 than PC2. Considering the average delay for PC2 is 33.255, this represents a 60.8% added delay as a result of going through a second NAT router.

Topology 2



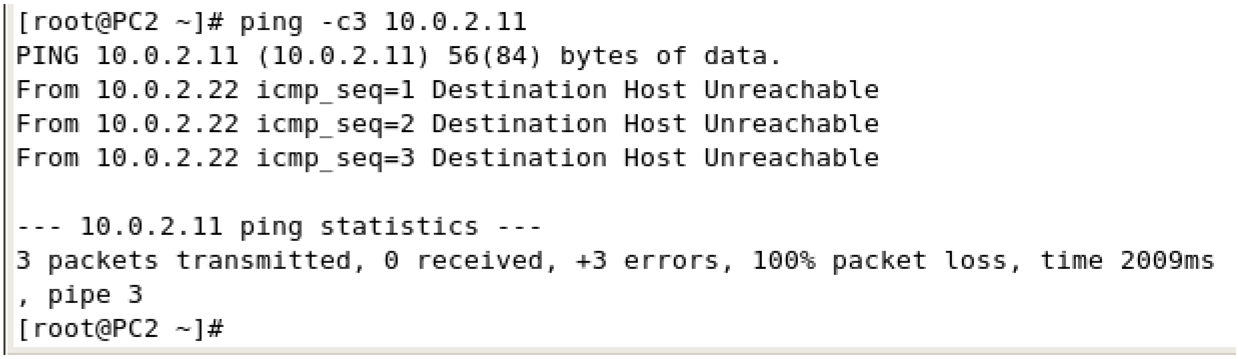
Remove PC2 from Subnet B and connect it to a third interface on R1, call this Subnet D. On R1, configure this subnet to have the same IP prefix as that of Subnet C.

1. Include the addressing plan in the table below.

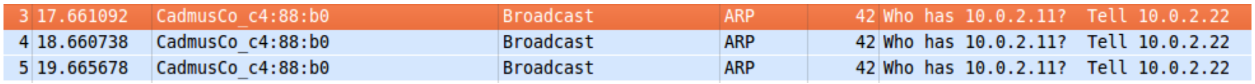
	Subnet A	Subnet B	Subnet C	Subnet D
IP prefix	200.0.0.0/24	10.0.1.0/24	10.0.2.0/24	10.0.2.0/24
R3	200.0.0.3			
R1	200.0.0.1	10.0.1.1		10.0.2.1
R2		10.0.1.2	10.0.2.2	
PC1			10.0.2.11	
PC2				10.0.2.22

2. View ip routing information on R1, what does this suggest about the capability of Subnet D to communicate with Subnet C? Issue a ping from PC2 to PC1, what do you observe? Support your answer with ping and traceroute output and packet captures.

Because the IP prefix of Subnet D and C are equivalent, communication between the two should not be possible. Below is ping output of an attempt at communication, and a packet capture on R1's fa2/0 interface:



```
[root@PC2 ~]# ping -c3 10.0.2.11
PING 10.0.2.11 (10.0.2.11) 56(84) bytes of data.
From 10.0.2.22 icmp_seq=1 Destination Host Unreachable
From 10.0.2.22 icmp_seq=2 Destination Host Unreachable
From 10.0.2.22 icmp_seq=3 Destination Host Unreachable

--- 10.0.2.11 ping statistics ---
3 packets transmitted, 0 received, +3 errors, 100% packet loss, time 2009ms
, pipe 3
[root@PC2 ~]#
```

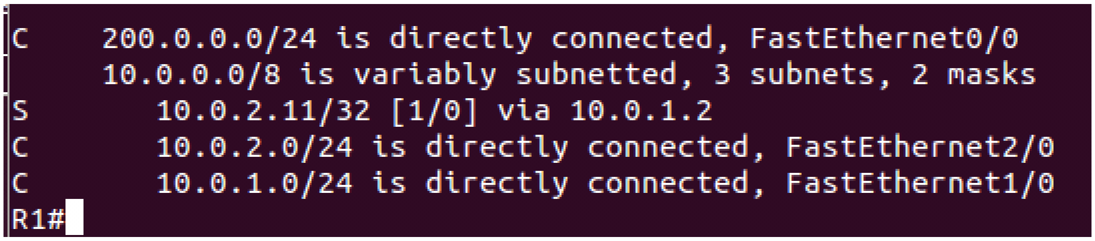


```
3 17.661092  CadmusCo_c4:88:b0   Broadcast   ARP   42 Who has 10.0.2.11?  Tell 10.0.2.22
4 18.660738  CadmusCo_c4:88:b0   Broadcast   ARP   42 Who has 10.0.2.11?  Tell 10.0.2.22
5 19.665678  CadmusCo_c4:88:b0   Broadcast   ARP   42 Who has 10.0.2.11?  Tell 10.0.2.22
```

As is clear from results above, communication is unsuccessful because PC2 assumes PC1 to be on the same subnet since their IP prefixes are the same. As a result, PC2 sends out an ARP Broadcast that never receives a response.

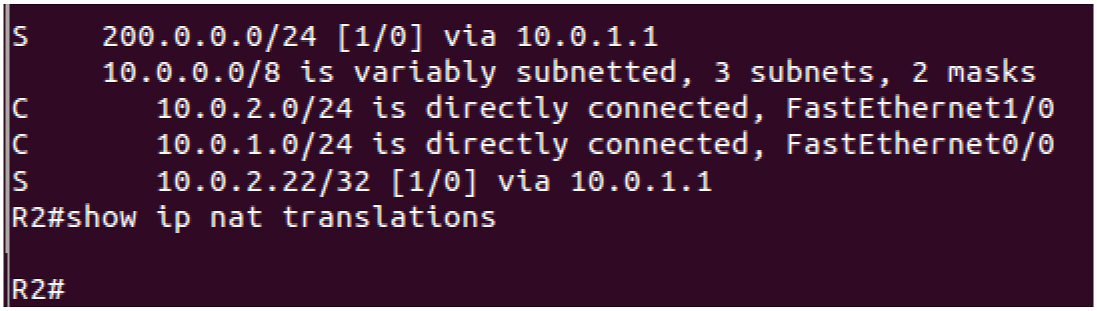
3. Statically configure a route to PC1 on PC2 and R1, as well as a route to PC2 on PC1 and R2. Demonstrate successful connectivity via traceroute output and packet captures. Include routing table information from R1 and R2.

Routing table information from R1:



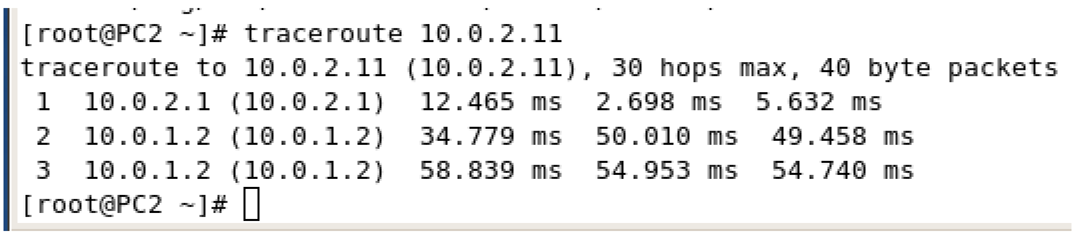
```
C    200.0.0.0/24 is directly connected, FastEthernet0/0
     10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
S       10.0.2.11/32 [1/0] via 10.0.1.2
C       10.0.2.0/24 is directly connected, FastEthernet2/0
C       10.0.1.0/24 is directly connected, FastEthernet1/0
R1#
```

Routing table information from R2:



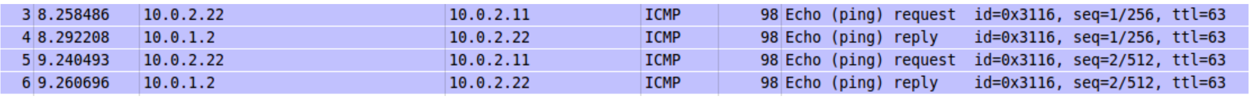
```
S    200.0.0.0/24 [1/0] via 10.0.1.1
     10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
C       10.0.2.0/24 is directly connected, FastEthernet1/0
C       10.0.1.0/24 is directly connected, FastEthernet0/0
S       10.0.2.22/32 [1/0] via 10.0.1.1
R2#show ip nat translations

R2#
```

Traceroute output from PC2 to PC1:



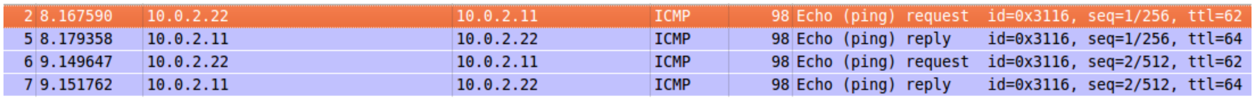
```
[root@PC2 ~]# traceroute 10.0.2.11
traceroute to 10.0.2.11 (10.0.2.11), 30 hops max, 40 byte packets
 1  10.0.2.1 (10.0.2.1)  12.465 ms  2.698 ms  5.632 ms
 2  10.0.1.2 (10.0.1.2)  34.779 ms  50.010 ms  49.458 ms
 3  10.0.1.2 (10.0.1.2)  58.839 ms  54.953 ms  54.740 ms
[root@PC2 ~]# 
```

Packet capture on R2's fa0/0 interface:



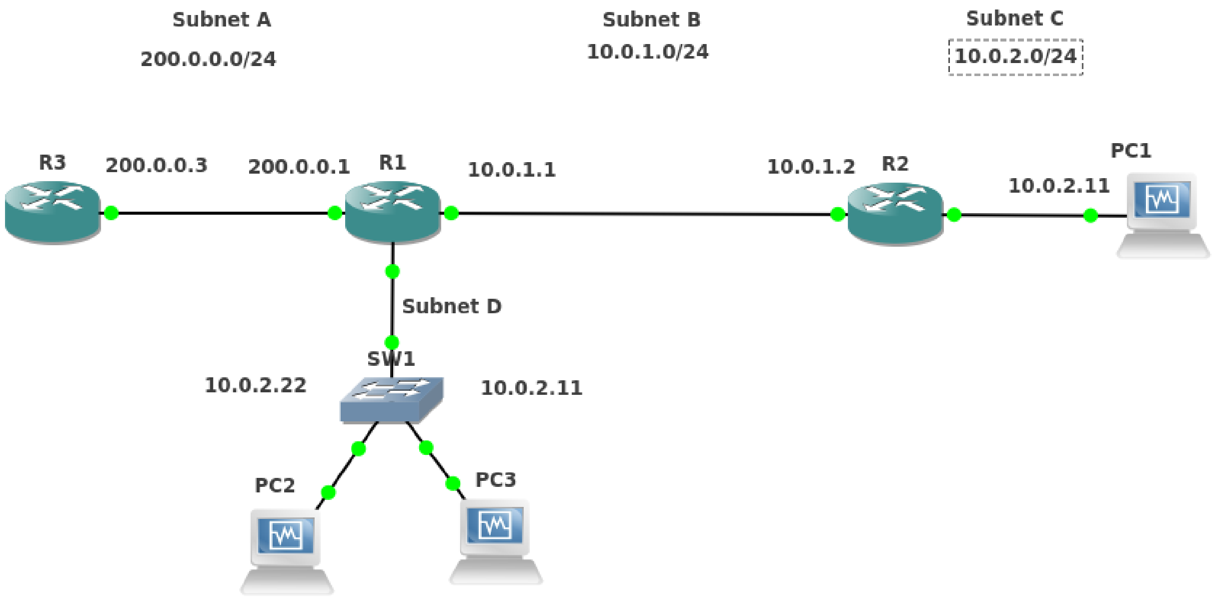
3	8.258486	10.0.2.22	10.0.2.11	ICMP	98	Echo (ping) request	id=0x3116, seq=1/256, ttl=63
4	8.292208	10.0.1.2	10.0.2.22	ICMP	98	Echo (ping) reply	id=0x3116, seq=1/256, ttl=63
5	9.240493	10.0.2.22	10.0.2.11	ICMP	98	Echo (ping) request	id=0x3116, seq=2/512, ttl=63
6	9.260696	10.0.1.2	10.0.2.22	ICMP	98	Echo (ping) reply	id=0x3116, seq=2/512, ttl=63

Packet capture on R2's fa1/0 interface:

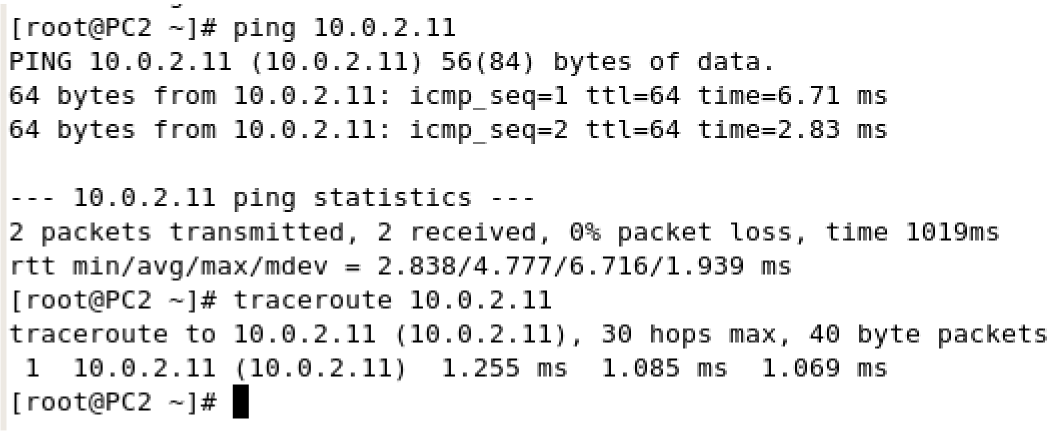


2	8.167590	10.0.2.22	10.0.2.11	ICMP	98	Echo (ping) request	id=0x3116, seq=1/256, ttl=62
5	8.179358	10.0.2.11	10.0.2.22	ICMP	98	Echo (ping) reply	id=0x3116, seq=1/256, ttl=64
6	9.149647	10.0.2.22	10.0.2.11	ICMP	98	Echo (ping) request	id=0x3116, seq=2/512, ttl=62
7	9.151762	10.0.2.11	10.0.2.22	ICMP	98	Echo (ping) reply	id=0x3116, seq=2/512, ttl=64

4. Now add a third PC to the topology on Subnet D. Give this PC the same IPv4 address as PC1. Observe what happens when you try to ping PC1 from PC2. Does the message go to PC3, PC2, or both?



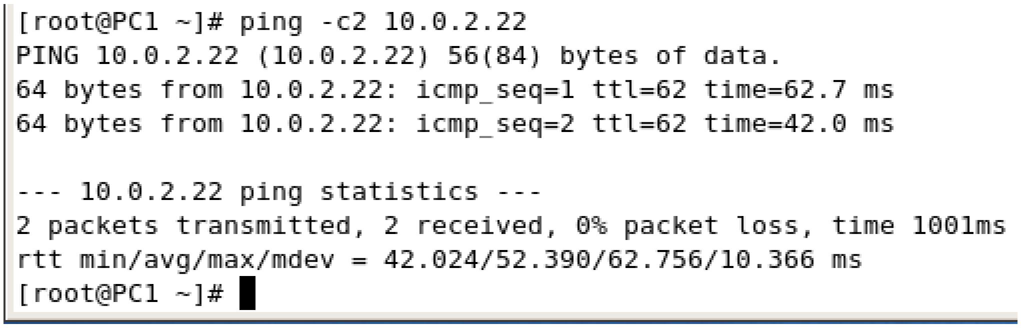
Below is *ping* and *traceroute* output from PC1 to 10.0.2.11 (PC3):



```
[root@PC2 ~]# ping 10.0.2.11
PING 10.0.2.11 (10.0.2.11) 56(84) bytes of data.
64 bytes from 10.0.2.11: icmp_seq=1 ttl=64 time=6.71 ms
64 bytes from 10.0.2.11: icmp_seq=2 ttl=64 time=2.83 ms

--- 10.0.2.11 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1019ms
rtt min/avg/max/mdev = 2.838/4.777/6.716/1.939 ms
[root@PC2 ~]# traceroute 10.0.2.11
traceroute to 10.0.2.11 (10.0.2.11), 30 hops max, 40 byte packets
 1  10.0.2.11 (10.0.2.11)  1.255 ms  1.085 ms  1.069 ms
[root@PC2 ~]# █
```

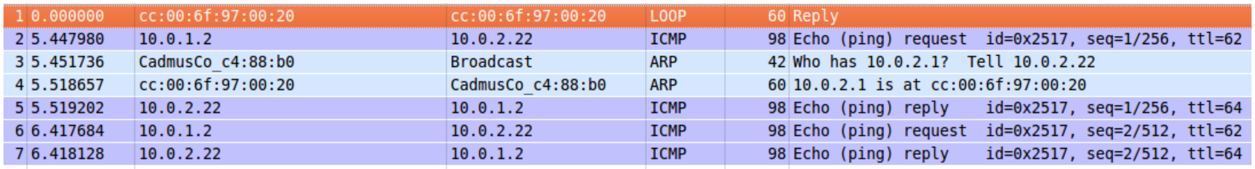
As is clear from the *traceroute* PC2 messaged PC3, not PC1. This is because PC3 is on the same subnet as PC2, so even though we have a static route to PC1, the message never leaves Subnet D.

However, notice what happens when we *ping* PC2 from PC1:



```
[root@PC1 ~]# ping -c2 10.0.2.22
PING 10.0.2.22 (10.0.2.22) 56(84) bytes of data.
64 bytes from 10.0.2.22: icmp_seq=1 ttl=62 time=62.7 ms
64 bytes from 10.0.2.22: icmp_seq=2 ttl=62 time=42.0 ms

--- 10.0.2.22 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 42.024/52.390/62.756/10.366 ms
[root@PC1 ~]# 
```

PC2 is able to send a reply back because NAT has translated the IP address of PC1 to that of R2's address on Subnet B. As proof see the below capture on R1's fa2/0 interface:



```
1 0.000000  cc:00:6f:97:00:20  cc:00:6f:97:00:20  LOOP  60 Reply
2 5.447980  10.0.1.2           10.0.2.22          ICMP  98 Echo (ping) request  id=0x2517, seq=1/256, ttl=62
3 5.451736  CadmusCo_c4:88:b0  Broadcast          ARP   42 Who has 10.0.2.1?  Tell 10.0.2.22
4 5.518657  cc:00:6f:97:00:20  CadmusCo_c4:88:b0  ARP   60 10.0.2.1 is at cc:00:6f:97:00:20
5 5.519202  10.0.2.22          10.0.1.2           ICMP  98 Echo (ping) reply    id=0x2517, seq=1/256, ttl=64
6 6.417684  10.0.1.2           10.0.2.22          ICMP  98 Echo (ping) request  id=0x2517, seq=2/512, ttl=62
7 6.418128  10.0.2.22          10.0.1.2           ICMP  98 Echo (ping) reply    id=0x2517, seq=2/512, ttl=64
```

We know the message is from PC1 because the ttl has been decremented by 2.