

## IP Interfaces: Part 2

### Instructions

For the second lab, you will select IP addresses from 10.10.11.0/24 to configure R2, R3, and R4 in Area 1 as shown in the network diagram. Since R2, R3, and R4 are connected by different hubs, they are not in the same broadcast and collision domain, and therefore do not have direct access to each other via Ethernet.

You will configure four different subnets so that each pair of directly connected routers can communicate with each other. But you also must make sure that your subnets do not overlap. Before configuring your VMs, fill in the table and verify that you have assigned appropriate subnets for all three VMs. Each subnet should be large enough to accommodate its routers, but no larger than necessary.

Note that you will assign a /28 on R4 (eth2) to be used later for a subsequent DHCP assignment.

- Each subnet will hold 2 IP addresses. Will require a total of 4 IP addresses for each subnet to include the broadcast and network address. /30 holds 4 IP addresses.
  - R4 using /28 for DHCP
- Subnet IP addresses are distributed in powers of 2. R4 (eth2) has to start from an IP address of 16 to be distributed properly.
  - Subnet boundaries must be a multiple of their size.
  - /28 has to start at 16.
- Broadcast addresses are the highest IP address
- Network addresses are the first IP address

VM (Interface)	IP Address	Network Address	Broadcast Address	Range (usable addresses)
R2 (eth1)	10.10.11.1/30	10.10.11.0	10.10.11.3	10.10.11.1-10.10.11.2
R3 (eth0)	10.10.11.2/30			
R2 (eth2)	10.10.11.5/30	10.10.11.4	10.10.11.7	10.10.11.5-10.10.11.6
R4(eth1)	10.10.11.6/30			
R3(eth1)	10.10.11.9/30	10.10.11.8	10.10.11.11	10.10.11.9-10.10.11.10
R4(eth0)	10.10.11.10/30			
R4 (eth2)	10.10.11.17/28	10.10.11.16	10.10.11.31	10.10.11.16-10.10.11.31

## Part 1: Configuring Network Interfaces

Use vtysh to do the configurations (See Interface Commands under Zebra in the frrouting manual). Don't forget to write to memory or your changes will be lost.

You may use Linux's ifconfig command in order to verify that you have saved your network configuration (see man ifconfig).

## Part 2: Questions

- a) Why must we ensure that our subnets do not overlap? Discuss one example of something that could go wrong. (10 points)

The subnets must not overlap in order to ensure that every IP address in a network is unique and as such, accessible if it exists within the network and has a valid address.

Overlapping subnets will have overlapping IP addresses as well as IP routing tables<sup>1</sup>. If a router needs to send a packet to an IP address of a subnet that overlaps with another, then that packet may be forwarded to an address in the wrong subnet<sup>2</sup>.

- b) Suppose there is another Router (R5) directly connected to the HUB between R3 and R4. Explain whether or not we would need to reconfigure the IP subnets on R3 and R4 in order to communicate with R5. (10 points)

Yes, the subnet shared by R3 and R4 is of size /30 which contains 4 IP addresses. All 4 of the /30 subnet IP addresses are being occupied by an R3 interface, an R4 interface, the network address, and the broadcast address. They would need their subnets to be changed to a larger one to accommodate an additional IP Address.

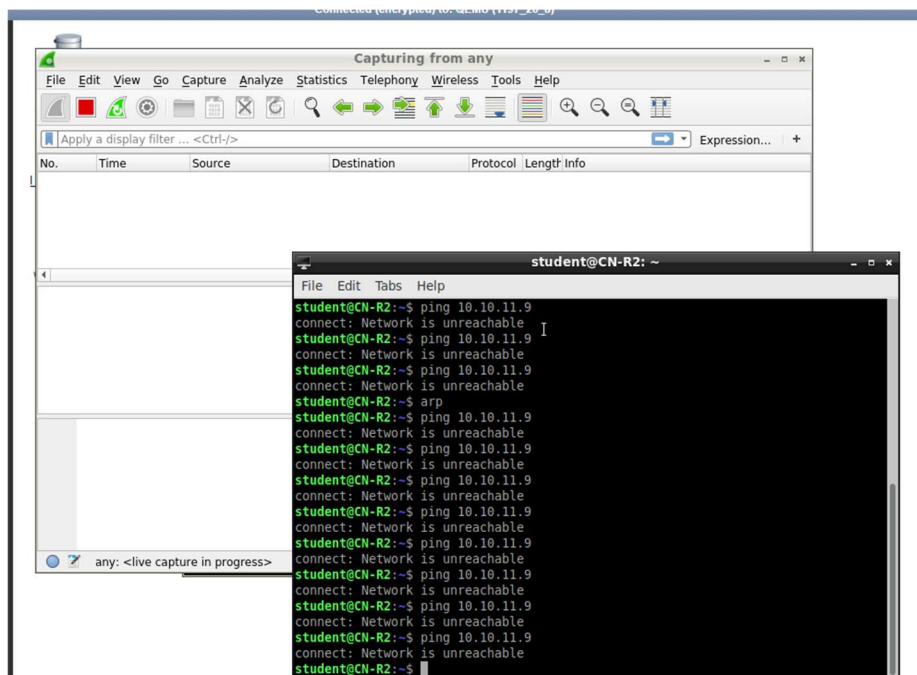
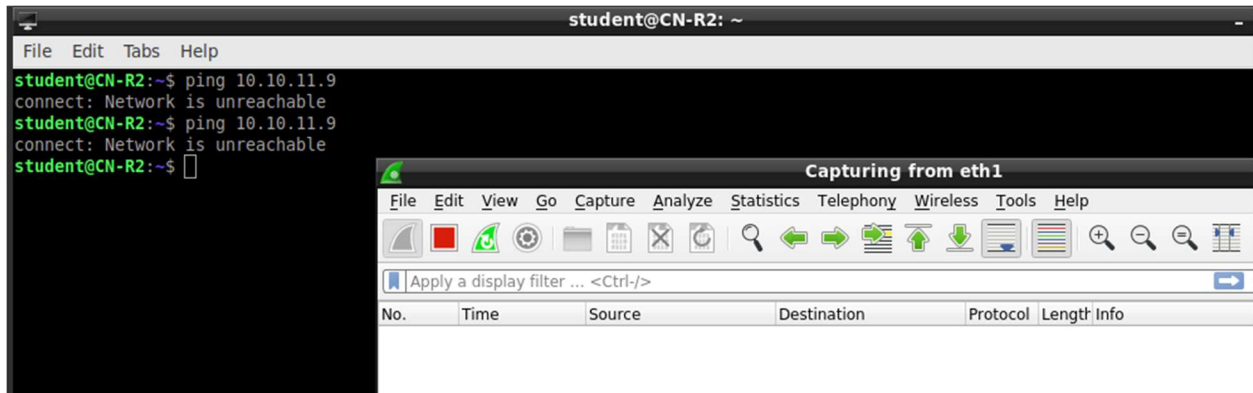
---

<sup>1</sup> <https://www.ciscopress.com/articles/article.asp?p=2738302&seqNum=2>

<sup>2</sup>

[https://blog.certskills.com/vlsmo\\_01\\_01/#:~:text=In%20real%20networks%2C%20if%20two,packet%20to%20the%20wrong%20subnet.](https://blog.certskills.com/vlsmo_01_01/#:~:text=In%20real%20networks%2C%20if%20two,packet%20to%20the%20wrong%20subnet.)

- c) Run Wireshark on R2 (eth1). Now ping R3(eth1) from R2. Identify what type of packet is used in ping. Why is R2 unable to reach R3 (eth1)? (10 points)



- Wireshark does not detect anything being sent on any interface, including eth1.

R2 does not see R3 within the eth1 interface, so R2 does not send anything through eth1. Wireshark does not see anything being sent through eth1 or any interface because R2 has been set to not broadcast an ARP request on all interfaces under this scenario.

R2 eth1 can't detect R3 eth1. They are in two different subnets which means a ping from R2 cannot jump from R2's subnet R3's subnet. Proxy ARPs and packet forwarding are not being used, and the two subnets have different broadcast addresses and domains, so R2 will not locate R3 through eth1.

Ping uses the ICMP echo request and ICMP echo reply packets. R2 pings by sending an ICMP echo request message to R3. R3 receives the request from R2 then replies by sending an ICMP echo reply message to R2.

R3 pinging R2 shows that the ping protocols being used below. It shows ICMP echo request and ICMP echo reply.

No.	Time	Source	Destination	Protocol	Length	Info
6	1.015309478	10.10.11.1	10.10.11.2	ICMP	98	Echo (ping) reply id=0x06c4, seq=6/1536, ttl=64 (requ
7	2.039265063	10.10.11.2	10.10.11.1	ICMP	98	Echo (ping) request id=0x06c4, seq=7/1792, ttl=64 (repl
8	2.039306429	10.10.11.1	10.10.11.2	ICMP	98	Echo (ping) reply id=0x06c4, seq=7/1792, ttl=64 (requ
9	3.063088661	10.10.11.2	10.10.11.1	ICMP	98	Echo (ping) request id=0x06c4, seq=8/2048, ttl=64 (repl
10	3.063127984	10.10.11.1	10.10.11.2	ICMP	98	Echo (ping) reply id=0x06c4, seq=8/2048, ttl=64 (requ
11	4.087193280	10.10.11.2	10.10.11.1	ICMP	98	Echo (ping) request id=0x06c4, seq=9/2304, ttl=64 (repl
12	4.087237193	10.10.11.1	10.10.11.2	ICMP	98	Echo (ping) reply id=0x06c4, seq=9/2304, ttl=64 (requ
13	5.111077895	10.10.11.2	10.10.11.1	ICMP	98	Echo (ping) request id=0x06c4, seq=10/2560, ttl=64 (rep
14	5.111113135	10.10.11.1	10.10.11.2	ICMP	98	Echo (ping) reply id=0x06c4, seq=10/2560, ttl=64 (requ
15	6.134956819	10.10.11.2	10.10.11.1	ICMP	98	Echo (ping) request id=0x06c4, seq=11/2816, ttl=64 (rep
16	6.135036504	10.10.11.1	10.10.11.2	ICMP	98	Echo (ping) reply id=0x06c4, seq=11/2816, ttl=64 (requ

Frame 8: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface 0

Ethernet II, Src: 00:00:00:00:00:05 (00:00:00:00:00:05), Dst: 00:00:00:00:00:06 (00:00:00:00:00:06)

Internet Protocol Version 4, Src: 10.10.11.1, Dst: 10.10.11.2

0100 .... = Version: 4

.... 0101 = Header Length: 20 bytes (5)

Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)

Total Length: 84

Identification: 0xc64 (53092)

Flags: 0x0000

Time to live: 64

Protocol: ICMP (1)

Header checksum: 0x812e [validation disabled]

[Header checksum status: Unverified]

Source: 10.10.11.1

- d) Briefly describe how Wireshark results compare when you ping R3 (eth0) from R2 (eth1). (5 points)

Pinging R3 from R2 eth0. R2 is able to locate R3 and its IP address. This is because those two IP addresses are within the same subnet and share the same broadcast address. R3 can broadcast its ARP request to R2 so that both devices can record each other's ARP information. R2 and R3 are able to locate each other in this subnet.

Connected (encrypted) to: QEMU (1197\_20\_8)

Capturing from eth1

No.	Time	Source	Destination	Protocol	Length	Info
2	0.000237091	00:00:00:00:00:06	00:00:00:00:00:05	ARP	42	10.10.11.2 is at 00:00:00:00:00:06
3	0.000266111	10.10.11.1	10.10.11.2	ICMP	98	Echo (ping) request id=0x06d5, seq
4	0.000570972	10.10.11.2	10.10.11.1	ICMP	98	Echo (ping) reply id=0x06d5, seq
5	1.001220939	10.10.11.1	10.10.11.2	ICMP	98	Echo (ping) request id=0x06d5, seq
6	1.001450423	10.10.11.2	10.10.11.1	ICMP	98	Echo (ping) reply id=0x06d5, seq
7	2.019775353	10.10.11.1	10.10.11.2	ICMP	98	Echo (ping) request id=0x06d5, seq
8	2.020159620	10.10.11.2	10.10.11.1	ICMP	98	Echo (ping) reply id=0x06d5, seq

Frame 1: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0

Ethernet II, Src: 00:00:00:00:00:05 (00:00:00:00:00:05), Dst: 00:00:00:00:00:06 (00:00:00:00:00:06)

Address Resolution Protocol (request)

0000 ff ff ff ff ff 00 00 00 00 05 08

0010 08 00 06 04 00 01 00 00 00 00 05 0a

0020 00 00 00 00 00 0a 0a 0b 02

student@CN-R2: ~

```

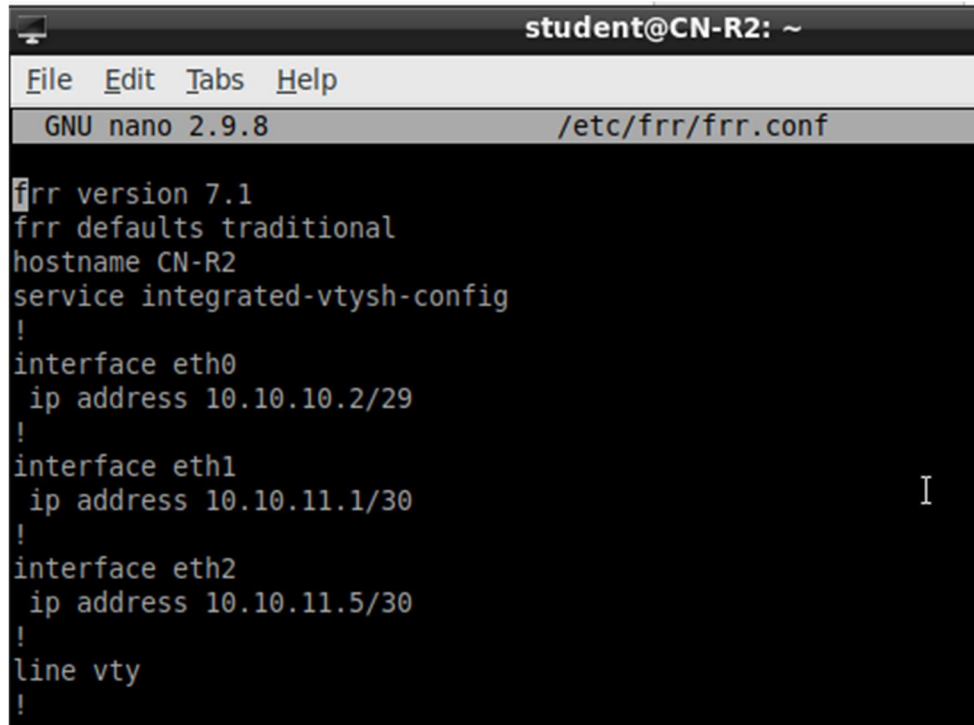
student@CN-R2:~$ ping 10.10.11.2
PING 10.10.11.2 (10.10.11.2) 56(84) bytes of data.
64 bytes from 10.10.11.2: icmp_seq=1 ttl=64 time=0.610 ms
64 bytes from 10.10.11.2: icmp_seq=2 ttl=64 time=0.268 ms
64 bytes from 10.10.11.2: icmp_seq=3 ttl=64 time=0.421 ms
64 bytes from 10.10.11.2: icmp_seq=4 ttl=64 time=0.536 ms
  
```

Brandon Vo

Submissions

[30 pts] Screenshot of the .conf file under /etc/frr/frr.conf from R2, R3, and R4

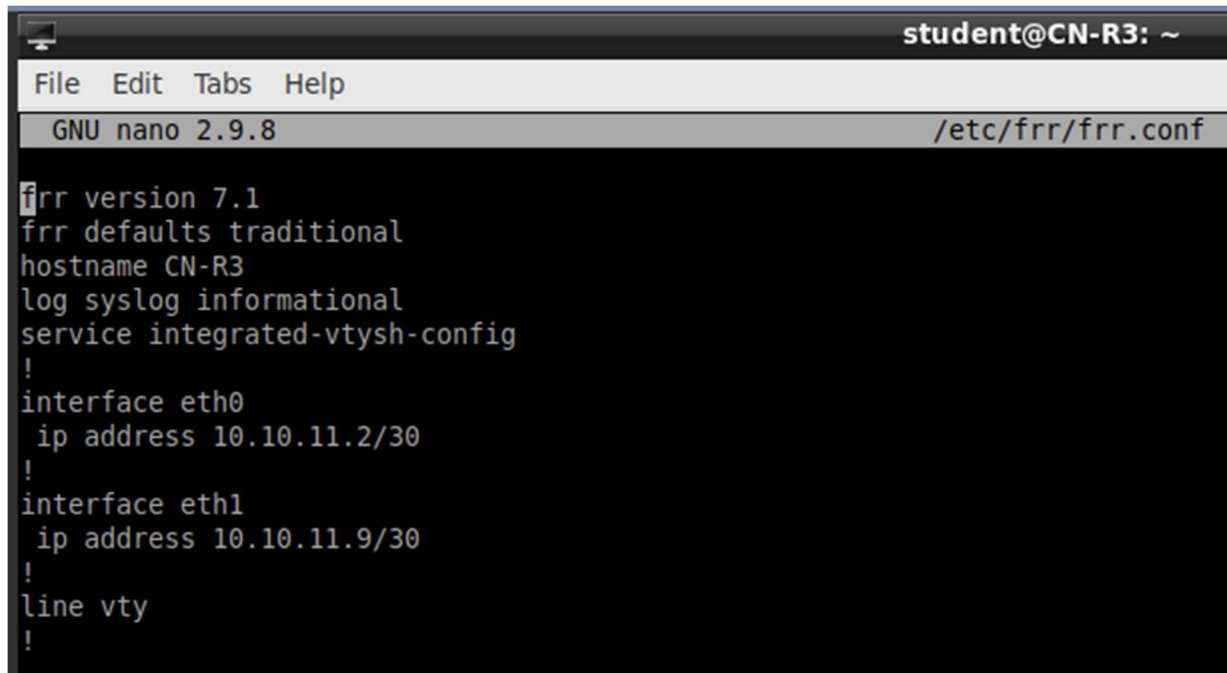
R2 conf file

A screenshot of a terminal window showing the configuration file for R2. The window title is 'student@CN-R2: ~'. The editor is GNU nano 2.9.8, editing /etc/frr/frr.conf. The configuration includes frr version 7.1, traditional defaults, hostname CN-R2, integrated-vtysh service, and three interfaces: eth0 (10.10.10.2/29), eth1 (10.10.11.1/30), and eth2 (10.10.11.5/30).

```
student@CN-R2: ~
File Edit Tabs Help
GNU nano 2.9.8 /etc/frr/frr.conf

frr version 7.1
frr defaults traditional
hostname CN-R2
service integrated-vtysh-config
!
interface eth0
  ip address 10.10.10.2/29
!
interface eth1
  ip address 10.10.11.1/30
!
interface eth2
  ip address 10.10.11.5/30
!
line vty
!
```

R3 conf file

A screenshot of a terminal window showing the configuration file for R3. The window title is 'student@CN-R3: ~'. The editor is GNU nano 2.9.8, editing /etc/frr/frr.conf. The configuration includes frr version 7.1, traditional defaults, hostname CN-R3, informational syslog, integrated-vtysh service, and two interfaces: eth0 (10.10.11.2/30) and eth1 (10.10.11.9/30).

```
student@CN-R3: ~
File Edit Tabs Help
GNU nano 2.9.8 /etc/frr/frr.conf

frr version 7.1
frr defaults traditional
hostname CN-R3
log syslog informational
service integrated-vtysh-config
!
interface eth0
  ip address 10.10.11.2/30
!
interface eth1
  ip address 10.10.11.9/30
!
line vty
!
```

Brandon Vo

R4 conf table

```
student@CN-R4: ~  
File Edit Tabs Help  
GNU nano 2.9.8 /etc/frr/frr.conf  
frr version 7.1  
frr defaults traditional  
hostname CN-R4  
log syslog informational  
service integrated-vtysh-config  
!  
interface eth0  
ip address 10.10.11.10/30  
!  
interface eth1  
ip address 10.10.11.6/30  
!  
interface eth2  
ip address 10.10.11.17/28  
!  
line vty  
!
```

[20 pts] Your IP subnet table

[10 pts] Screenshot showing that pinging works between R2, R3, and R4

R2 pinging R3 (eth0) and R4 (eth1)

```
student@CN-R2: ~  
File Edit Tabs Help  
student@CN-R2:~$ ping 10.10.11.6  
PING 10.10.11.6 (10.10.11.6) 56(84) bytes of data.  
64 bytes from 10.10.11.6: icmp_seq=1 ttl=64 time=0.497 ms  
64 bytes from 10.10.11.6: icmp_seq=2 ttl=64 time=0.383 ms  
64 bytes from 10.10.11.6: icmp_seq=3 ttl=64 time=0.636 ms  
^C  
--- 10.10.11.6 ping statistics ---  
3 packets transmitted, 3 received, 0% packet loss, time 48ms  
rtt min/avg/max/mdev = 0.383/0.505/0.636/0.105 ms  
student@CN-R2:~$ ping 10.10.11.2  
PING 10.10.11.2 (10.10.11.2) 56(84) bytes of data.  
64 bytes from 10.10.11.2: icmp_seq=1 ttl=64 time=0.548 ms  
64 bytes from 10.10.11.2: icmp_seq=2 ttl=64 time=0.661 ms  
64 bytes from 10.10.11.2: icmp_seq=3 ttl=64 time=0.351 ms  
^C  
--- 10.10.11.2 ping statistics ---  
3 packets transmitted, 3 received, 0% packet loss, time 58ms  
rtt min/avg/max/mdev = 0.351/0.520/0.661/0.128 ms  
student@CN-R2:~$
```



Brandon Vo

R3 pinging R2 (eth1) and R4 (eth0)

```
student@CN-R3: ~  
File Edit Tabs Help  
student@CN-R3:~$ ping 10.10.11.10  
PING 10.10.11.10 (10.10.11.10) 56(84) bytes of data.  
64 bytes from 10.10.11.10: icmp_seq=1 ttl=64 time=0.423 ms  
64 bytes from 10.10.11.10: icmp_seq=2 ttl=64 time=0.454 ms  
64 bytes from 10.10.11.10: icmp_seq=3 ttl=64 time=0.430 ms  
^C  
--- 10.10.11.10 ping statistics ---  
3 packets transmitted, 3 received, 0% packet loss, time 39ms  
rtt min/avg/max/mdev = 0.423/0.435/0.454/0.027 ms  
student@CN-R3:~$ ping 10.10.11.1  
PING 10.10.11.1 (10.10.11.1) 56(84) bytes of data.  
64 bytes from 10.10.11.1: icmp_seq=1 ttl=64 time=0.367 ms  
64 bytes from 10.10.11.1: icmp_seq=2 ttl=64 time=0.283 ms  
64 bytes from 10.10.11.1: icmp_seq=3 ttl=64 time=0.438 ms  
64 bytes from 10.10.11.1: icmp_seq=4 ttl=64 time=0.487 ms  
^C  
--- 10.10.11.1 ping statistics ---  
4 packets transmitted, 4 received, 0% packet loss, time 63ms  
rtt min/avg/max/mdev = 0.283/0.393/0.487/0.080 ms  
student@CN-R3:~$
```

R4 pinging R2 (eth2) and R3 (eth1)

```
student@CN-R4:~$ ping 10.10.11.9  
PING 10.10.11.9 (10.10.11.9) 56(84) bytes of data.  
64 bytes from 10.10.11.9: icmp_seq=1 ttl=64 time=0.726 ms  
64 bytes from 10.10.11.9: icmp_seq=2 ttl=64 time=0.304 ms  
64 bytes from 10.10.11.9: icmp_seq=3 ttl=64 time=0.504 ms  
64 bytes from 10.10.11.9: icmp_seq=4 ttl=64 time=0.319 ms  
^C  
--- 10.10.11.9 ping statistics ---  
4 packets transmitted, 4 received, 0% packet loss, time 68ms  
rtt min/avg/max/mdev = 0.304/0.463/0.726/0.171 ms  
student@CN-R4:~$ ping 10.10.11.5  
PING 10.10.11.5 (10.10.11.5) 56(84) bytes of data.  
64 bytes from 10.10.11.5: icmp_seq=1 ttl=64 time=0.366 ms  
64 bytes from 10.10.11.5: icmp_seq=2 ttl=64 time=0.437 ms  
64 bytes from 10.10.11.5: icmp_seq=3 ttl=64 time=0.638 ms  
64 bytes from 10.10.11.5: icmp_seq=4 ttl=64 time=0.473 ms  
^C  
--- 10.10.11.5 ping statistics ---  
4 packets transmitted, 4 received, 0% packet loss, time 82ms  
rtt min/avg/max/mdev = 0.366/0.478/0.638/0.102 ms  
student@CN-R4:~$
```

[5 pts] Screenshot of the ARP tables on R2, R3, and R4

```
student@CN-R3:~$ arp
Address          HWtype  HWaddress      Flags Mask    Iface
10.10.11.1       ether   00:00:00:00:00:05 C              eth0
10.10.11.10      ether   00:00:00:00:00:08 C              eth1
student@CN-R3:~$
```

```
student@CN-R2:~$ arp
Address          HWtype  HWaddress      Flags Mask    Iface
10.10.11.2       ether   00:00:00:00:00:06 C              eth1
10.10.11.6       ether   00:00:00:00:00:0a C              eth2
student@CN-R2:~$
```

```
student@CN-R4:~$ arp
Address          HWtype  HWaddress      Flags Mask    Iface
10.10.11.5       ether   00:00:00:00:00:09 C              eth1
10.10.11.9       ether   00:00:00:00:00:07 C              eth0
student@CN-R4:~$
```

[35 pts] Answers to questions 2a-2d.