

IP Interfaces: Part 1

Part 0: Warm Up (not for credit)

- a. What is the slash notation representation of 255.255.255.0?

255.255.255.0/24

- b. What is the dot-decimal representation of /30?

255.255.255.252

- c. What is the smallest subnet size that would accommodate 5 hosts?

Size 8, netmask: 255.255.255.248/29

- d. Fill in the blank cells in the table below (R1 eth0 filled in by Vital system)

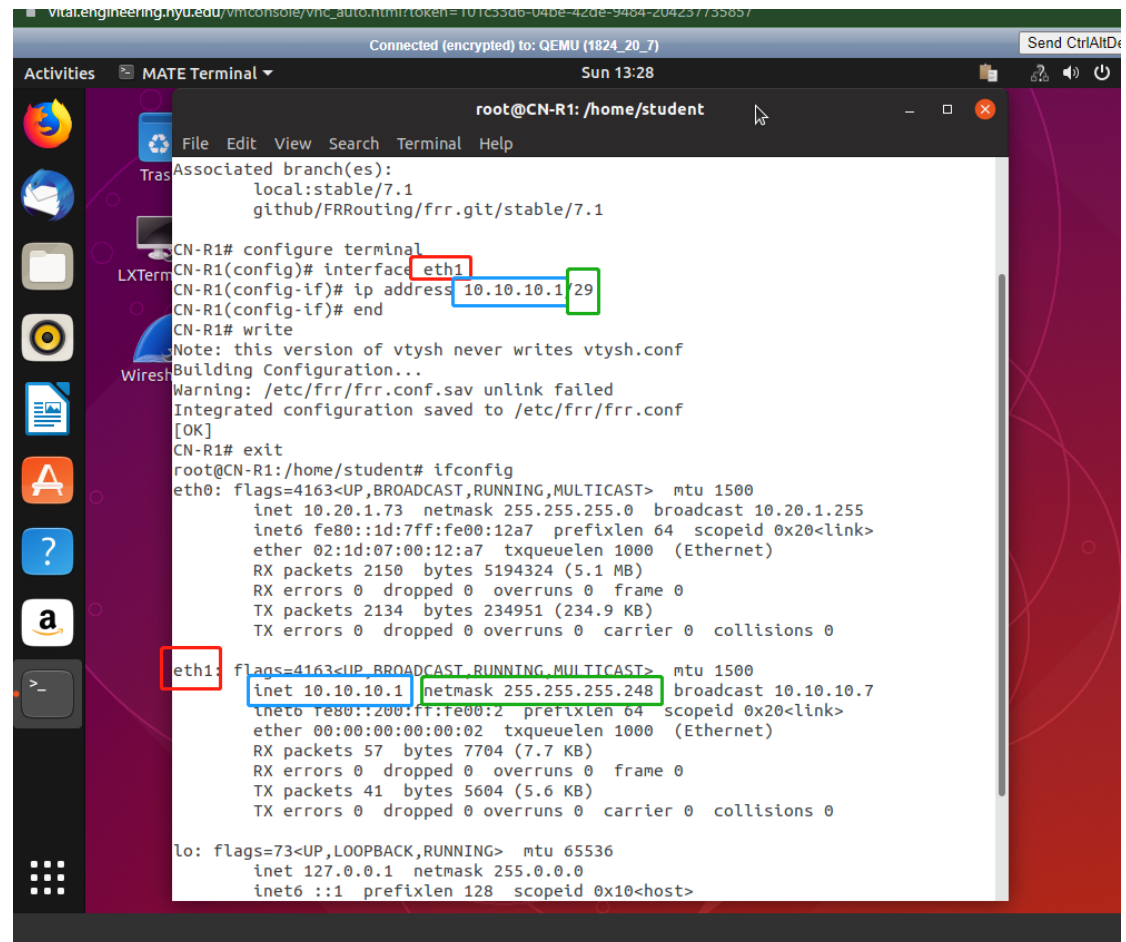
VM (Interface)	IP Address (CIDR Notation)
R1 (eth1)	10.10.10.1/29
R2 (eth0)	10.10.10.2/29
Kali (eth0)	10.10.10.3/29

Part1

For R1

Configuration processing and the ifconfig double check

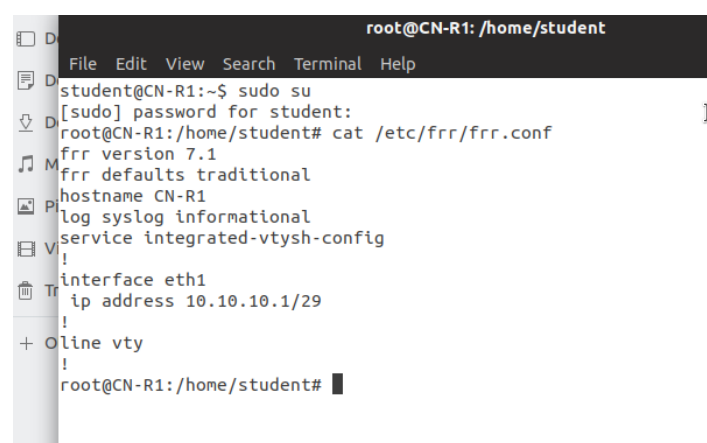
Screenshot of .conf file under /etc/frr/frr.conf



```
root@CN-R1: /home/student
File Edit View Search Terminal Help
Associated branch(es):
  local:stable/7.1
  github:FRRouting/frr.git/stable/7.1
CN-R1# configure terminal
CN-R1(config)# interface eth1
CN-R1(config-if)# ip address 10.10.10.1/29
CN-R1(config-if)# end
CN-R1# write
Note: this version of vtysh never writes vtysh.conf
Building Configuration...
Warning: /etc/frr/frr.conf.sav unlink failed
Integrated configuration saved to /etc/frr/frr.conf
[OK]
CN-R1# exit
root@CN-R1: /home/student# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.20.1.73 netmask 255.255.255.0 broadcast 10.20.1.255
    inet6 fe80::1d:7ff:fe00:12a7 prefixlen 64 scopeid 0x20<link>
    ether 02:1d:07:00:12:a7 txqueuelen 1000 (Ethernet)
    RX packets 2150 bytes 5194324 (5.1 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 2134 bytes 234951 (234.9 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.10.10.1 netmask 255.255.255.248 broadcast 10.10.10.7
    inet6 fe80::200:ff:fe00:2 prefixlen 64 scopeid 0x20<link>
    ether 00:00:00:00:00:02 txqueuelen 1000 (Ethernet)
    RX packets 57 bytes 7704 (7.7 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 41 bytes 5604 (5.6 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
```

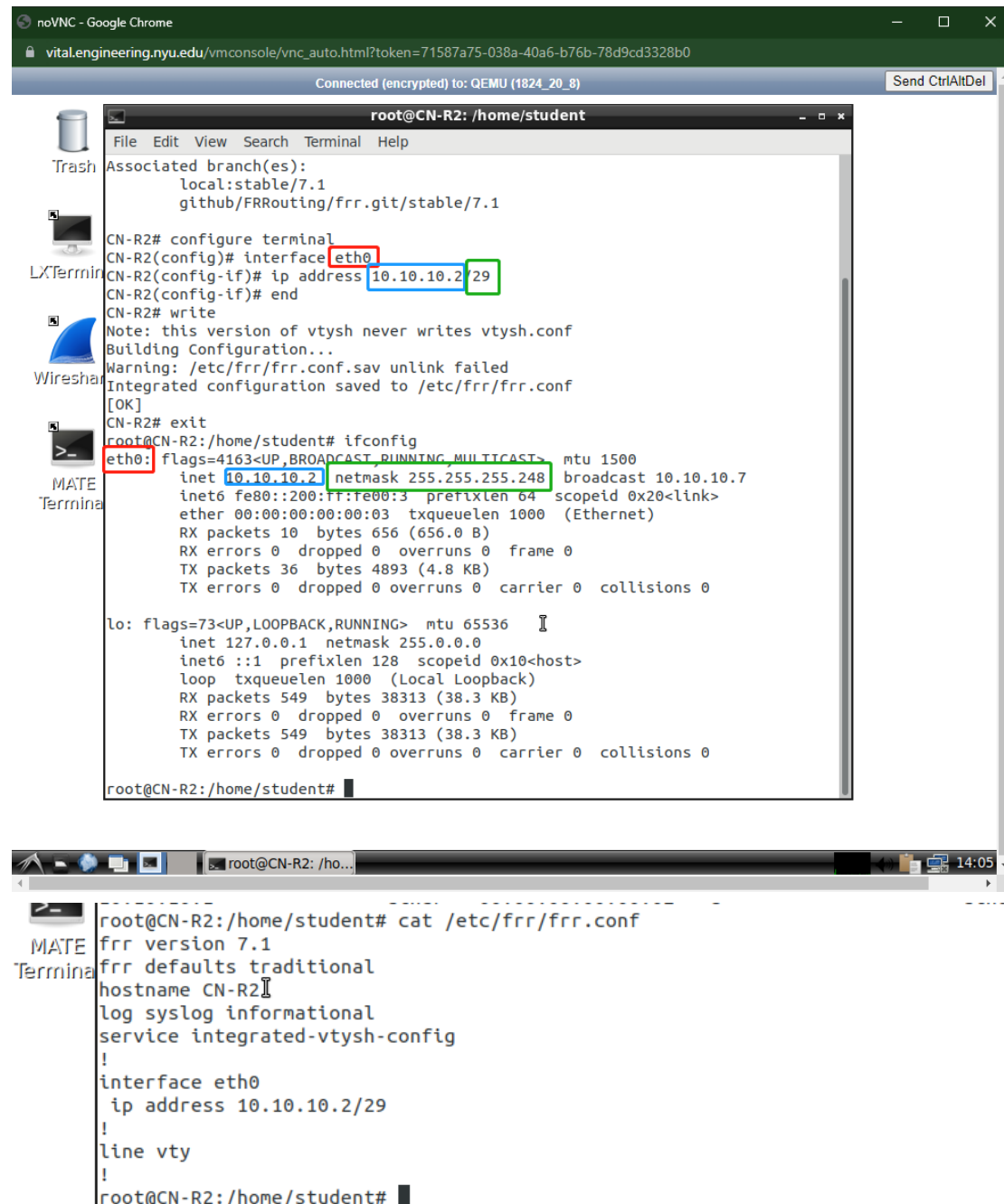


```
root@CN-R1: /home/student
File Edit View Search Terminal Help
student@CN-R1:~$ sudo su
[sudo] password for student:
root@CN-R1: /home/student# cat /etc/frr/frr.conf
frr version 7.1
frr defaults traditional
hostname CN-R1
log syslog informational
service integrated-vtysh-config
!
interface eth1
ip address 10.10.10.1/29
!
+
!
root@CN-R1: /home/student#
```

For R2

Configuration processing and the ifconfig double check

Screenshot of .conf file under /etc/frr/frr.conf



```
noVNC - Google Chrome
vital.engineering.nyu.edu/vmconsole/vnc_auto.html?token=71587a75-038a-40a6-b76b-78d9cd3328b0
Connected (encrypted) to: QEMU (1824_20_8) Send CtrlAltDel

root@CN-R2: /home/student
File Edit View Search Terminal Help
Associated branch(es):
  local:stable/7.1
  github:FRRouting/frr.git/stable/7.1
CN-R2# configure terminal
CN-R2(config)# interface eth0
CN-R2(config-if)# ip address 10.10.10.2/29
CN-R2(config-if)# end
CN-R2# write
Note: this version of vtysh never writes vtysh.conf
Building Configuration...
Warning: /etc/frr/frr.conf.sav unlink failed
Integrated configuration saved to /etc/frr/frr.conf
[OK]
CN-R2# exit
root@CN-R2:/home/student# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.10.10.2 netmask 255.255.255.248 broadcast 10.10.10.7
    inet6 fe80::200:ff:fe00:3 prefixlen 64 scopeid 0x20<link>
    ether 00:00:00:00:00:03 txqueuelen 1000 (Ethernet)
    RX packets 10 bytes 656 (656.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 36 bytes 4893 (4.8 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 549 bytes 38313 (38.3 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 549 bytes 38313 (38.3 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@CN-R2:/home/student#

root@CN-R2:/home/student# cat /etc/frr/frr.conf
frr version 7.1
frr defaults traditional
hostname CN-R2
log syslog informational
service integrated-vtysh-config
!
interface eth0
ip address 10.10.10.2/29
!
line vty
!
root@CN-R2:/home/student#
```

Part 2

Auto eth0

```
iface eth0 inet static
```

```
    address 10.10.10.3
```

```
    netmask 255.255.255.248
```

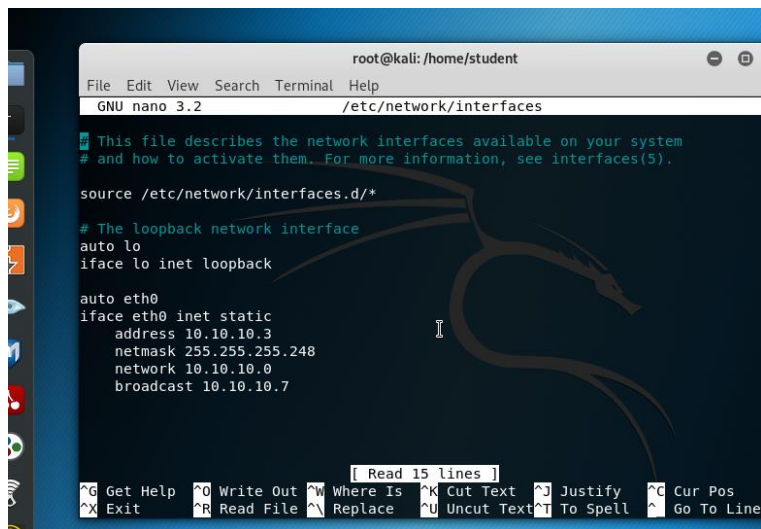
```
    network 10.10.10.0
```

```
    broadcast 10.10.10.7
```

Configuration processing and the ifconfig double check

First screenshot is the code write into the interfaces

Second screenshot is after reboot and the ifconfig



The screenshot shows a terminal window with a nano editor editing the file /etc/network/interfaces. The user is root at kali. The file content is as follows:

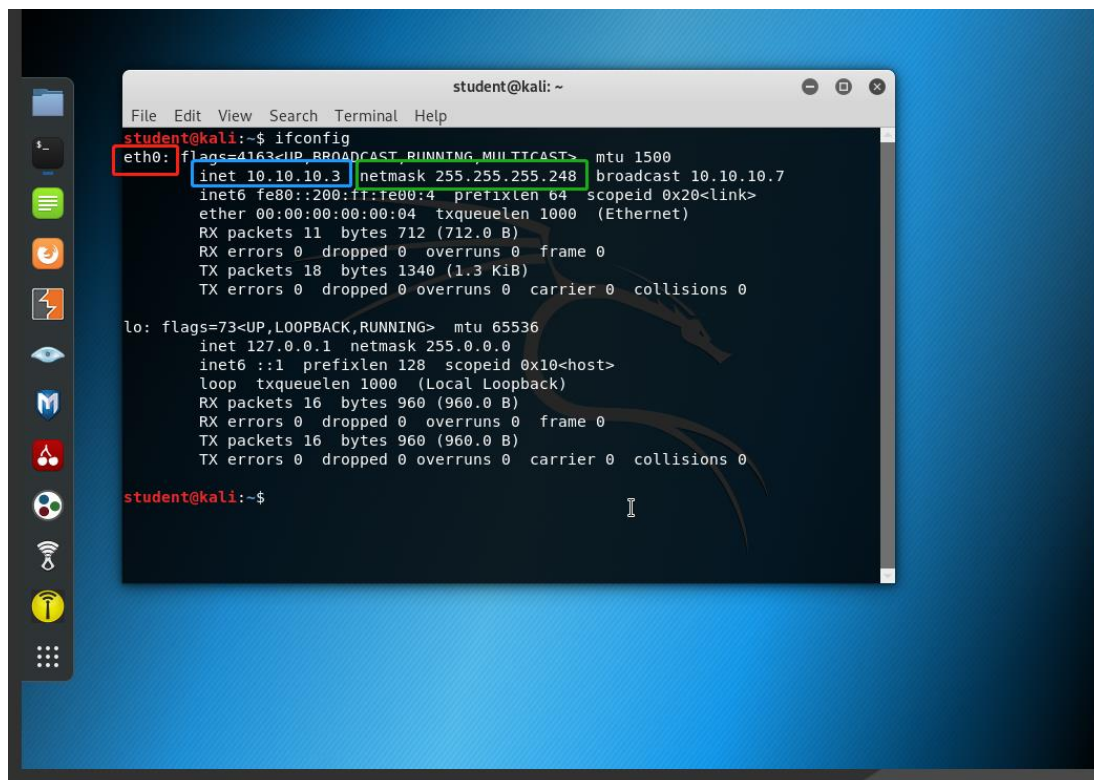
```
root@kali: /home/student
File Edit View Search Terminal Help
GNU nano 3.2 /etc/network/interfaces

This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
    address 10.10.10.3
    netmask 255.255.255.248
    network 10.10.10.0
    broadcast 10.10.10.7
```



The screenshot shows a terminal window with the user student at kali. The user has run the ifconfig command, and the output is displayed. The output for eth0 is highlighted with a red box, and the output for lo is highlighted with a green box.

```
student@kali:~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.10.10.3 netmask 255.255.255.248 broadcast 10.10.10.7
    inet6 fe80::200:ff:fe00:4 prefixlen 64 scopeid 0x20<link>
    ether 00:00:00:00:00:04 txqueuelen 1000 (Ethernet)
    RX packets 11 bytes 712 (712.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 18 bytes 1340 (1.3 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 16 bytes 960 (960.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 16 bytes 960 (960.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

student@kali:~$
```

Part 3

Q a) Why did we choose the /29 subnet mask for Area 0? (10 points)

because in area 0 we have 3 ip address need to be assigned. And with netmask 255.255.255.248 or /29, we can have a working range of 6 machine»

Netmask: 255.255.255.248 or /29 Starts with multiple of 8;

Network	Range	Broadcast
0	1-6	7

So I take out the first 3 from the range of /29 and assign to R1, R2 and Kali.

If we choose Netmask: 255.255.255.252 or /30, it only contains 2 work ip except broadcast which is not large enough here. And Netmask: 255.255.255.240 or /28 was unnecessary large for the Area 0 networking.

Q b) The Linux ARP (see man ARP) command will print the current entries in the machine's address resolution protocol table. Now that you have configured Area 0, what entries are currently in R1, R2, and Kali? (10 points)

ARP before ping for R1, R2 and Kali

```
root@CN-R1:/home/student# arp
Address          HWtype  HWaddress      Flags Mask    Iface
10.20.1.1        ether    96:ce:ab:7b:67:a9  C             eth0
root@CN-R1:/home/student#
```

```
root@CN-R2:/home/student# arp
root@CN-R2:/home/student#
```

```
[sudo] password for student:
root@kali:/home/student# nano /etc/network/interfaces
root@kali:/home/student# arp
root@kali:/home/student#
```

Currently R1, R2 and Kali's addresses are not in each other's ARP table.

Q c) Now ping both R2 and Kali from R1. Note the changes on each machine's ARP tables. At this point, R2 should be aware of R1, but why doesn't R2 have a table entry for Kali? (10 points)

Under R1, ping R2

The first ARP cmd shows that R2/10.10.10.2 is not in the ARP table, and after pinging, R2's address is in the ARP table.

```
root@CN-R1:/home/student# arp
Address          HWtype  HWaddress      Flags Mask    Iface
10.20.1.1        ether    96:ce:ab:7b:67:a9  C             eth0
root@CN-R1:/home/student# ping 10.10.10.2
PING 10.10.10.2 (10.10.10.2) 56(84) bytes of data:
64 bytes from 10.10.10.2: icmp_seq=1 ttl=64 time=0.703 ms
64 bytes from 10.10.10.2: icmp_seq=2 ttl=64 time=0.275 ms
64 bytes from 10.10.10.2: icmp_seq=3 ttl=64 time=0.364 ms
64 bytes from 10.10.10.2: icmp_seq=4 ttl=64 time=0.397 ms
64 bytes from 10.10.10.2: icmp_seq=5 ttl=64 time=0.348 ms
64 bytes from 10.10.10.2: icmp_seq=6 ttl=64 time=0.379 ms
^C
--- 10.10.10.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 95ms
rtt min/avg/max/mdev = 0.275/0.411/0.703/0.136 ms
root@CN-R1:/home/student# arp
Address          HWtype  HWaddress      Flags Mask    Iface
10.20.1.1        ether    96:ce:ab:7b:67:a9  C             eth0
10.10.10.2        ether    00:00:00:00:00:03  C             eth1
root@CN-R1:/home/student#
```

Under R1, ping Kali

The first ARP shows that Kali machine /10.10.10.3 is not in the ARP table, and after pinging, Kali's address is in the ARP table.

```
root@CN-R1:/home/student# arp
Address          HWtype  HWaddress      Flags Mask    Iface
10.20.1.1        ether   96:ce:ab:7b:67:a9  C             eth0
10.10.10.2        ether   00:00:00:00:00:03  C             eth1
root@CN-R1:/home/student# ping 10.10.10.3
PING 10.10.10.3 (10.10.10.3) 56(84) bytes of data.
64 bytes from 10.10.10.3: icmp_seq=1 ttl=64 time=0.931 ms
64 bytes from 10.10.10.3: icmp_seq=2 ttl=64 time=0.605 ms
64 bytes from 10.10.10.3: icmp_seq=3 ttl=64 time=0.623 ms
64 bytes from 10.10.10.3: icmp_seq=4 ttl=64 time=0.529 ms
^C
--- 10.10.10.3 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 81ms
rtt min/avg/max/mdev = 0.529/0.672/0.931/0.153 ms
root@CN-R1:/home/student# arp
Address          HWtype  HWaddress      Flags Mask    Iface
10.10.10.3        ether   00:00:00:00:00:04  C             eth1
10.20.1.1        ether   96:ce:ab:7b:67:a9  C             eth0
10.10.10.2        ether   00:00:00:00:00:03  C             eth1
root@CN-R1:/home/student#
```

Screenshot from R2 and Kali (the ARP table before and after pinged from R1)

```
root@CN-R2:/home/student# arp
root@CN-R2:/home/student# arp
Address          HWtype  HWaddress      Flags Mask    Iface
10.10.10.1        ether   00:00:00:00:00:02  C             eth0
root@CN-R2:/home/student#
```

```
root@kali:/home/student# nano /etc/network/interfaces
root@kali:/home/student# arp
root@kali:/home/student# arp
Address          HWtype  HWaddress      Flags Mask    Iface
10.10.10.1        ether   00:00:00:00:00:02  C             eth0
root@kali:/home/student#
```

Both machines are aware of R1

The Reason that R2 ARP table doesn't contain an entry for Kali is because either Kali or R2 doesn't send out any ARP request to each other so for now R2 doesn't have a table entry for Kali. But if I directly ping from R2 to Kali or reversely ping from Kali to R2. Both Kali and R2's ARP table will have the entry for each other.

Below is I use R2 to ping Kali, and ARP table is being updated, Kali now is in R2's ARP table after pinging, they are aware of each other.

```
student@CN-R2:~$ arp
student@CN-R2:~$ arp
Address          Hwtype  Hwaddress      Flags Mask       Iface
10.10.10.1       ether   00:00:00:00:00:02 C                eth0
student@CN-R2:~$ ping 10.10.10.3
PING 10.10.10.3 (10.10.10.3) 56(84) bytes of data:
64 bytes from 10.10.10.3: icmp_seq=1 ttl=64 time=0.874 ms
64 bytes from 10.10.10.3: icmp_seq=2 ttl=64 time=0.470 ms
64 bytes from 10.10.10.3: icmp_seq=3 ttl=64 time=0.445 ms
64 bytes from 10.10.10.3: icmp_seq=4 ttl=64 time=0.385 ms
^C
--- 10.10.10.3 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 54ms
rtt min/avg/max/mdev = 0.385/0.543/0.874/0.194 ms
student@CN-R2:~$ arp
Address          Hwtype  Hwaddress      Flags Mask       Iface
10.10.10.1       ether   00:00:00:00:00:02 C                eth0
10.10.10.3       ether   00:00:00:00:00:04 C                eth0
student@CN-R2:~$
```

And after ping, Kali's ARP table is also being updated with R2's information.

```
student@kali:~$ arp
student@kali:~$ arp
Address          Hwtype  Hwaddress      Flags Mask       Iface
10.10.10.1       ether   00:00:00:00:00:02 C                eth0
student@kali:~$ arp
Address          Hwtype  Hwaddress      Flags Mask       Iface
10.10.10.2       ether   00:00:00:00:00:03 C                eth0
10.10.10.1       ether   00:00:00:00:00:02 C                eth0
student@kali:~$
```

Screenshot of ping between R1, R2 and Kali

