Week #7
IPv4 Addressing and Static Routing

Objective: To setup a network with two routers and exchange packets across routers.

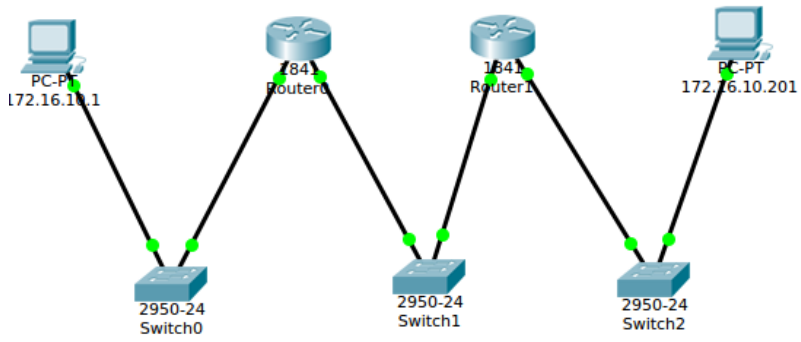
Hardware Requirements:

- Desktops/Laptops : 4
- Switch : 3
- Patch Cords (1.5m) : 6
- External NIC : 2

Software Requirements:

- Wireshark Tool
- Ubuntu Linux Operating System

Topology Description: Design a network with at least 2 router networks. Host **Ha** should be able to communicate with Host **Hd** using newly assigned addresses.



Note:

- Experiment to be accomplished in a group of 4.
- Make sure connections are flawless.
- Assign the IP address using commands or ‘Edit connections’.
- Don’t disturb existing hardware setup while setting IP address or doing experiment.
- Choose your ethernet interface according to your machine.

Task 1: Assign IP addresses to all computers A, B, C and D (Source Host Ha, Router R1, Router R2 & Destination Host Hd).

Step 1: Assign the IP address to the Ha.

$ sudo ip addr add 172.16.10.1/24 dev eth1

$ ip addr show

```
student@pesit-To-be-filled-by-O-E-M:~$ ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP ql
en 1000
    link/ether 50:e5:49:1c:64:c0 brd ff:ff:ff:ff:ff:ff
    inet 172.16.10.1/24 brd 172.16.10.255 scope global eth1
    inet6 fe80::52e5:49ff:fe1c:64c0/64 scope link
       valid_lft forever preferred_lft forever
student@pesit-To-be-filled-by-O-E-M:~$
```

Step 2: Assign the IP address to R1.

$ sudo ip addr add 172.16.10.201/24 dev eth1

$ sudo ip addr add 172.16.11.1/24 dev eth2

$ ip addr show

```
student@pesit-To-be-filled-by-O-E-M:~$ ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP ql
en 1000
    link/ether b8:a3:86:98:42:73 brd ff:ff:ff:ff:ff:ff
    inet 172.16.11.1/24 brd 172.16.11.255 scope global eth2
    inet6 fe80::baa3:86ff:fe98:4273/64 scope link
       valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP ql
en 1000
    link/ether 50:e5:49:1d:4a:ad brd ff:ff:ff:ff:ff:ff
    inet 172.16.10.201/24 brd 172.16.10.255 scope global eth1
    inet6 fe80::52e5:49ff:fe1d:4aad/64 scope link
       valid_lft forever preferred_lft forever
student@pesit-To-be-filled-by-O-E-M:~$
```

Step 3: Assign the IP address to R2.

$ sudo ip addr add 172.16.11.201/24 dev eth2

$ sudo ip addr add 172.16.12.1/24 dev eth1

$ ip addr show

```
student@pesit-To-be-filled-by-O-E-M:~$ ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP q
en 1000
    link/ether 28:10:7b:4b:15:05 brd ff:ff:ff:ff:ff:ff
    inet 172.16.11.201/24 brd 172.16.11.255 scope global eth2
    inet6 fe80::2a10:7bff:fe4b:1505/64 scope link
       valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP q
en 1000
    link/ether 50:e5:49:1c:d3:ae brd ff:ff:ff:ff:ff:ff
    inet 172.16.12.1/24 brd 172.16.12.255 scope global eth1
    inet6 fe80::52e5:49ff:fe1c:d3ae/64 scope link
       valid_lft forever preferred_lft forever
student@pesit-To-be-filled-by-O-E-M:~$
```

Step 4: Assign the IP address to the Hd.

$ sudo ip addr add 172.16.12.201/24 dev eth1

$ ip addr show

```
student@pesit-To-be-filled-by-O-E-M:~$ sudo ip addr show
[sudo] password for student:
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP q
    link/ether 50:e5:49:1b:f0:c4 brd ff:ff:ff:ff:ff:ff
    inet 172.16.12.201/24 brd 172.16.12.255 scope global eth1
    inet6 fe80::52e5:49ff:fe1b:f0c4/64 scope link
       valid_lft forever preferred_lft forever
student@pesit-To-be-filled-by-O-E-M:~$
```

Note 1: The machines are physically on the same LAN, thus you may get ICMP redirect messages from other machines (in case you make some configuration mistakes). So, as a precautionary measure disable accepting the ICMP redirect packets. By default, the Ubuntu Linux enables accepting the ICMP redirect packets. On host machines Ha and Hd, issue the following command:

$ sudo sysctl –w net.ipv4.conf.all.accept_redirects=0

```
student@student-H81H3-I:~$ sudo sysctl -w net.ipv4.conf.all.accept_redirects=0
net.ipv4.conf.all.accept_redirects = 0
student@student-H81H3-I:~$
```

Note 2: Since machines are on same physical interface, the router is going to send ICMP redirect message disturbing the routing decision by hosts. Thus, disable sending of the ICMP redirect packets by these routers with aliased interfaces. To have precautionary measures issue below command in router machines R1 and R2.

$ sudo sysctl –w net.ipv4.conf.all.send_redirects=0

```
student@student-H81H3-I:~$ sudo sysctl -w net.ipv4.conf.all.send_redirects=0
net.ipv4.conf.all.send_redirects = 0
student@student-H81H3-I:~$
```

Task 2: Convert the machines B and C into routers.

Note 1: Check if IP forwarding is enabled or not. We need to query the sysctl kernel value ***net.ipv4.ip_forward*** to see if forwarding is enabled or not using sysctl:

$ sysctl net.ipv4.ip_forward
net.ipv4.ip_forward = 0

Other alternative to check out if IP forwarding is enabled or not through the value in the ***/proc/sys***:

$cat /proc/sys/net/ipv4/ip_forward
0

Command to set the value of ***net.ipv4.ip_forward*** in R1 & R2 is given below:

At R1: **$ sudo sysctl –w net.ipv4.ip_forward=1**

At R2: **$ sudo sysctl –w net.ipv4.ip_forward=1**

```
student@student-H81H3-I:~$ sudo sysctl -w net.ipv4.ip_forward=1
net.ipv4.ip_forward = 1
```

Task 3: Verify the connection between Ha and Hd using ping command.

Initially test the connection of systems within the same network.

At Ha: **$ ping 172.16.10.1 (Local network)**

```
student@pesit-To-be-filled-by-O-E-M:~$ ping 172.16.10.1
PING 172.16.10.1 (172.16.10.1) 56(84) bytes of data.
64 bytes from 172.16.10.1: icmp_req=1 ttl=64 time=0.027 ms
64 bytes from 172.16.10.1: icmp_req=2 ttl=64 time=0.020 ms
64 bytes from 172.16.10.1: icmp_req=3 ttl=64 time=0.018 ms
64 bytes from 172.16.10.1: icmp_req=4 ttl=64 time=0.018 ms
64 bytes from 172.16.10.1: icmp_req=5 ttl=64 time=0.022 ms
^C
--- 172.16.10.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 3996ms
rtt min/avg/max/mdev = 0.018/0.021/0.027/0.003 ms
student@pesit-To-be-filled-by-O-E-M:~$
```

At Hd: **$ ping 172.16.12.1 (Local network)**

```
student@pesit-To-be-filled-by-O-E-M:~$ ping 172.16.10.201
PING 172.16.10.201 (172.16.10.201) 56(84) bytes of data.
64 bytes from 172.16.10.201: icmp_req=1 ttl=64 time=0.248 ms
64 bytes from 172.16.10.201: icmp_req=2 ttl=64 time=0.237 ms
64 bytes from 172.16.10.201: icmp_req=3 ttl=64 time=0.223 ms
64 bytes from 172.16.10.201: icmp_req=4 ttl=64 time=0.219 ms
64 bytes from 172.16.10.201: icmp_req=5 ttl=64 time=0.216 ms
64 bytes from 172.16.10.201: icmp_req=6 ttl=64 time=0.211 ms
^C
--- 172.16.10.201 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 4997ms
rtt min/avg/max/mdev = 0.211/0.225/0.248/0.021 ms
student@pesit-To-be-filled-by-O-E-M:~$
```

Task 4: Insert routing table entries on each system to direct ipv4 packets to ping across the networks.

At Ha: **$ sudo ip route add 172.16.12.0/24 via 172.16.10.201**

$ sudo ip route add 172.16.11.0/24 via 172.16.10.201

$ ip route show

```
student@pesit-To-be-filled-by-O-E-M:~$ ip route show
default via 172.16.10.201 dev eth1  proto static
169.254.0.0/16 dev eth1  scope link  metric 1000
172.16.10.0/24 dev eth1  proto kernel  scope link  src 172.16.10.1  metric 1
172.16.12.0/24 via 172.16.10.201 dev eth1
student@pesit-To-be-filled-by-O-E-M:~$
```

In the first host since 172.16.10.0/24 network is local network we don't need any routing table entry. We need to have routing table entries for other networks such as 172.16.11.0/24 and 172.16.12.0/24.

At R1: **$ sudo ip route add 172.16.12.0/24 via 172.16.11.201**

$ ip route show

```
student@pesit-To-be-filled-by-O-E-M:~$ sudo ip route show
default via 172.16.10.1 dev eth1  proto static
169.254.0.0/16 dev eth1  scope link  metric 1000
172.16.10.0/24 dev eth1  proto kernel  scope link  src 172.16.10.201  metric 1
172.16.11.0/24 dev eth2  proto kernel  scope link  src 172.16.11.1  metric 1
172.16.12.0/24 via 172.16.11.201 dev eth2
student@pesit-To-be-filled-by-O-E-M:~$
```

Since R1 is connected to 172.16.10.0/24 and 172.16.11.0/24 networks we need to have one routing table entry to 172.16.12.0/24.

At R2: **$ sudo ip route add 172.16.10.0/24 via 172.16.11.1**

$ ip route show

```
student@pesit-To-be-filled-by-O-E-M:~$ ip route show
default via 172.16.12.201 dev eth1  proto static
169.254.0.0/16 dev eth2  scope link  metric 1000
172.16.10.0/24 via 172.16.11.1 dev eth2
172.16.11.0/24 dev eth2  proto kernel  scope link  src 172.16.11.201  metric 1
172.16.12.0/24 dev eth1  proto kernel  scope link  src 172.16.12.1  metric 1
student@pesit-To-be-filled-by-O-E-M:~$
```

At Hd: **$ sudo ip route add 172.16.10.0/24 via 172.16.12.1**

$ sudo ip route add 172.16.11.0/24 via 172.16.12.1

$ ip route show

```
student@pesit-To-be-filled-by-O-E-M:~$ sudo ip route show
default via 172.16.12.1 dev eth1  proto static
169.254.0.0/16 dev eth1  scope link  metric 1000
172.16.10.0/24 via 172.16.12.1 dev eth1
172.16.12.0/24 dev eth1  proto kernel  scope link  src 172.16.12.201  metric 1
student@pesit-To-be-filled-by-O-E-M:~$
```

Task 5: After adding routing table entries again verify the connection from Ha and Hd using ping command.

Step 1: Testing path from Ha and Hd

$ ping 172.16.12.1 and $ ping 172.16.12.201

```
student@pesit-To-be-filled-by-O-E-M:~$ ping 172.16.12.201
PING 172.16.12.201 (172.16.12.201) 56(84) bytes of data.
64 bytes from 172.16.12.201: icmp_req=1 ttl=62 time=0.661 ms
64 bytes from 172.16.12.201: icmp_req=2 ttl=62 time=0.643 ms
64 bytes from 172.16.12.201: icmp_req=3 ttl=62 time=0.640 ms
64 bytes from 172.16.12.201: icmp_req=4 ttl=62 time=0.630 ms
64 bytes from 172.16.12.201: icmp_req=5 ttl=62 time=0.660 ms
^C
--- 172.16.12.201 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 3998ms
rtt min/avg/max/mdev = 0.630/0.646/0.661/0.034 ms
student@pesit-To-be-filled-by-O-E-M:~$ ping 172.16.12.1
PING 172.16.12.1 (172.16.12.1) 56(84) bytes of data.
64 bytes from 172.16.12.1: icmp_req=1 ttl=63 time=0.486 ms
64 bytes from 172.16.12.1: icmp_req=2 ttl=63 time=0.446 ms
64 bytes from 172.16.12.1: icmp_req=3 ttl=63 time=0.442 ms
64 bytes from 172.16.12.1: icmp_req=4 ttl=63 time=0.451 ms
64 bytes from 172.16.12.1: icmp_req=5 ttl=63 time=0.447 ms
^C
--- 172.16.12.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 3998ms
rtt min/avg/max/mdev = 0.442/0.454/0.486/0.024 ms
student@pesit-To-be-filled-by-O-E-M:~$
```

Step 2: Testing path from Hd and Ha

$ ping 172.16.12.1 and $ ping 172.16.12.201

```
student@pesit-To-be-filled-by-O-E-M:~$ ping 172.16.10.1
PING 172.16.10.1 (172.16.10.1) 56(84) bytes of data.
64 bytes from 172.16.10.1: icmp_req=1 ttl=62 time=0.627 ms
64 bytes from 172.16.10.1: icmp_req=2 ttl=62 time=0.619 ms
64 bytes from 172.16.10.1: icmp_req=3 ttl=62 time=0.590 ms
64 bytes from 172.16.10.1: icmp_req=4 ttl=62 time=0.612 ms
64 bytes from 172.16.10.1: icmp_req=5 ttl=62 time=0.613 ms
^C
--- 172.16.10.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 3999ms
rtt min/avg/max/mdev = 0.590/0.612/0.627/0.019 ms
student@pesit-To-be-filled-by-O-E-M:~$ ping 172.16.10.201
PING 172.16.10.201 (172.16.10.201) 56(84) bytes of data.
64 bytes from 172.16.10.201: icmp_req=1 ttl=63 time=0.458 ms
64 bytes from 172.16.10.201: icmp_req=2 ttl=63 time=0.443 ms
64 bytes from 172.16.10.201: icmp_req=3 ttl=63 time=0.404 ms
64 bytes from 172.16.10.201: icmp_req=4 ttl=63 time=0.398 ms
64 bytes from 172.16.10.201: icmp_req=5 ttl=63 time=0.423 ms
^C
--- 172.16.10.201 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 3999ms
rtt min/avg/max/mdev = 0.398/0.425/0.458/0.026 ms
student@pesit-To-be-filled-by-O-E-M:~$
```

Task 6: Check each system neighbor to verify the connection.

ip neighbor provides a command line interface to display the neighbor table (ARP cache)

At Ha: **$ ip neigh show**

```
student@pesit-To-be-filled-by-O-E-M:~$ ip neigh  show
172.16.10.201 dev eth1 lladdr 50:e5:49:1d:4a:ad REACHABLE
student@pesit-To-be-filled-by-O-E-M:~$
```

At R1: **$ ip neigh show**

```
student@pesit-To-be-filled-by-O-E-M:~$ ip neigh show
'172.16.11.201 dev eth2 lladdr 28:10:7b:4b:15:05 STALE
172.16.10.1 dev eth1 lladdr 50:e5:49:1c:64:c0 STALE
student@pesit-To-be-filled-by-O-E-M:~$ '
```

At R2: **$ ip neigh show**

```
student@pesit-To-be-filled-by-O-E-M:~$ ip neigh show
172.16.11.1 dev eth2 lladdr b8:a3:86:98:42:73 STALE
172.16.12.201 dev eth1 lladdr 50:e5:49:1b:f0:c4 REACHABLE
student@pesit-To-be-filled-by-O-E-M:~$
```

At Hd: **$ ip neigh show**

```
student@pesit-To-be-filled-by-O-E-M:~$ ip neigh show
172.16.12.1 dev eth1 lladdr 50:e5:49:1c:d3:ae REACHABLE
student@pesit-To-be-filled-by-O-E-M:~$
```

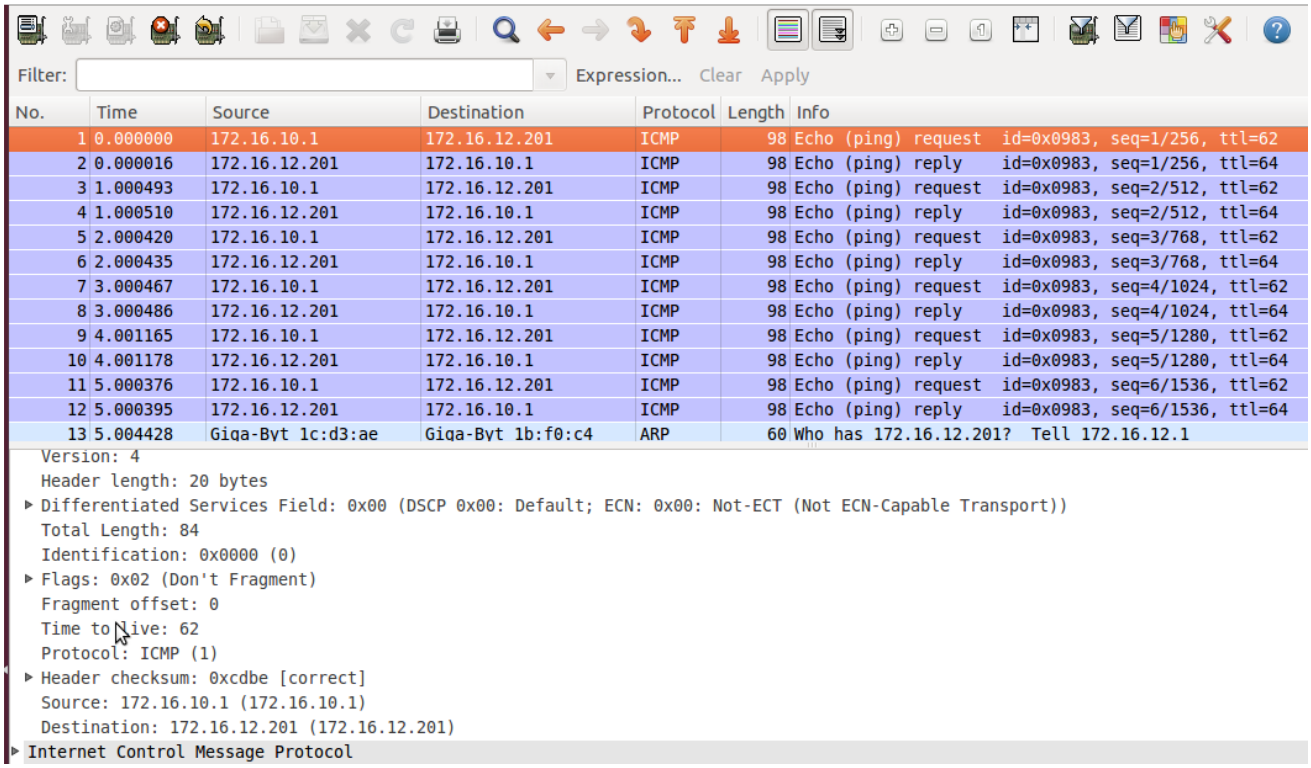
Task 7: Capture packets from Ha and Hb using Wireshark tool.

Step 1: Capture packets from Ha and Hd.

At Ha:

T1: $ sudo wireshark

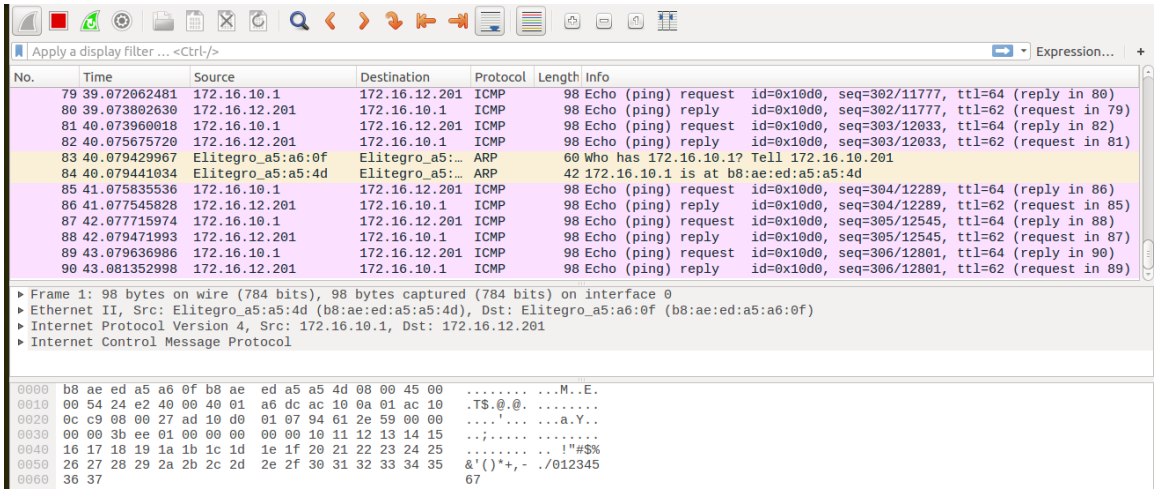
T2: $ ping 172.16.12.201



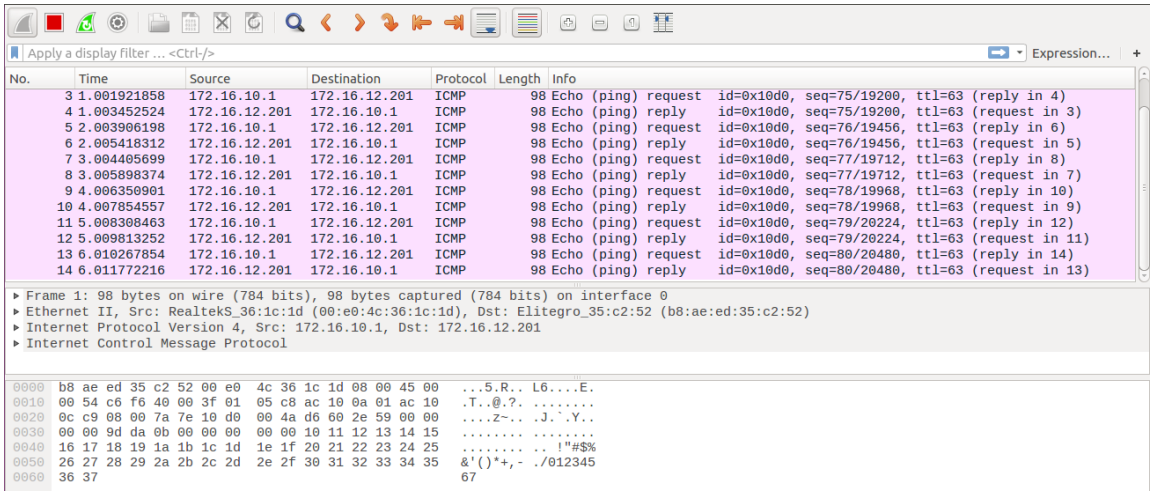
Step 2: Capture packets from R1 using both eth1 and eth2 interfaces.

$ sudo wireshark

At eth1:



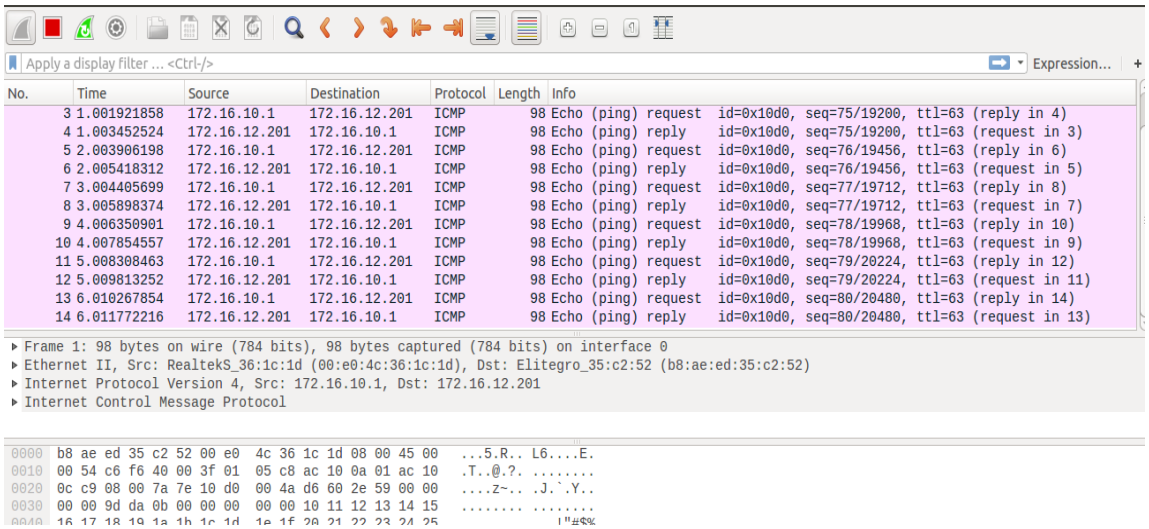
At eth2:



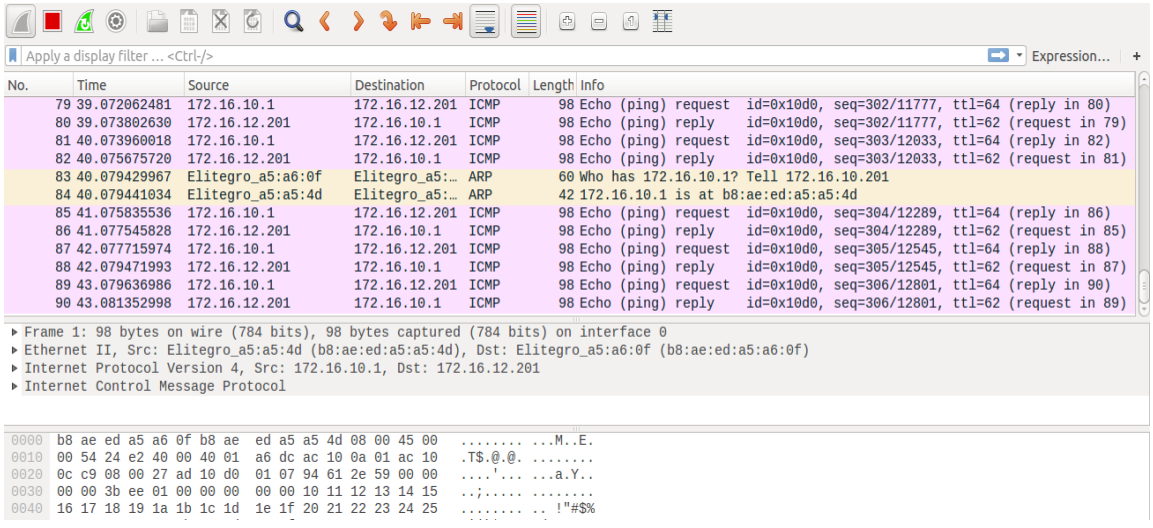
Step 3: Capture packets from R2 using both eth1 and eth2 interfaces.

$ sudo wireshark

At eth1:



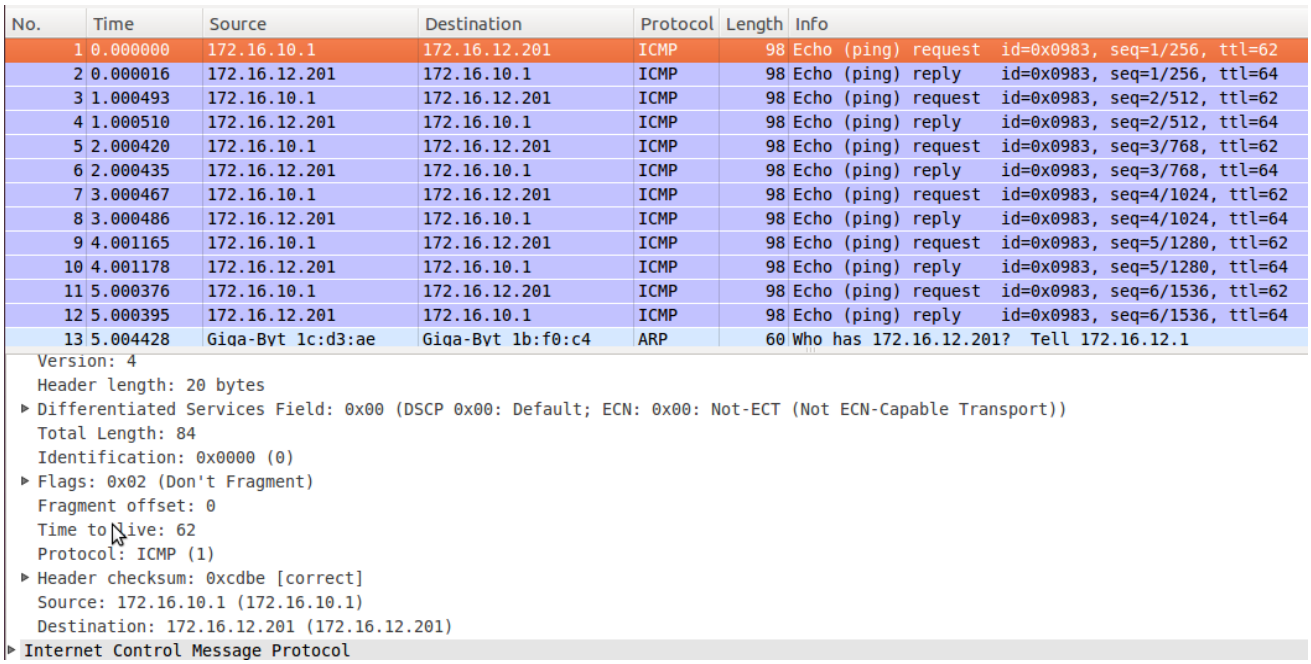
At eth2:



Step 4: Capture packets from Hd and Ha.

At Hd:

T1: $ sudo wireshark



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	172.16.10.1	172.16.12.201	ICMP	98	Echo (ping) request id=0x0983, seq=1/256, ttl=62
2	0.000016	172.16.12.201	172.16.10.1	ICMP	98	Echo (ping) reply id=0x0983, seq=1/256, ttl=64
3	1.000493	172.16.10.1	172.16.12.201	ICMP	98	Echo (ping) request id=0x0983, seq=2/512, ttl=62
4	1.000510	172.16.12.201	172.16.10.1	ICMP	98	Echo (ping) reply id=0x0983, seq=2/512, ttl=64
5	2.000420	172.16.10.1	172.16.12.201	ICMP	98	Echo (ping) request id=0x0983, seq=3/768, ttl=62
6	2.000435	172.16.12.201	172.16.10.1	ICMP	98	Echo (ping) reply id=0x0983, seq=3/768, ttl=64
7	3.000467	172.16.10.1	172.16.12.201	ICMP	98	Echo (ping) request id=0x0983, seq=4/1024, ttl=62
8	3.000486	172.16.12.201	172.16.10.1	ICMP	98	Echo (ping) reply id=0x0983, seq=4/1024, ttl=64
9	4.001165	172.16.10.1	172.16.12.201	ICMP	98	Echo (ping) request id=0x0983, seq=5/1280, ttl=62
10	4.001178	172.16.12.201	172.16.10.1	ICMP	98	Echo (ping) reply id=0x0983, seq=5/1280, ttl=64
11	5.000376	172.16.10.1	172.16.12.201	ICMP	98	Echo (ping) request id=0x0983, seq=6/1536, ttl=62
12	5.000395	172.16.12.201	172.16.10.1	ICMP	98	Echo (ping) reply id=0x0983, seq=6/1536, ttl=64
13	5.004428	Giga-Byt 1c:d3:ae	Giga-Byt 1b:f0:c4	ARP	60	Who has 172.16.12.201? Tell 172.16.12.1

```
Version: 4
Header length: 20 bytes
▸ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))
Total Length: 84
Identification: 0x0000 (0)
▸ Flags: 0x02 (Don't Fragment)
Fragment offset: 0
Time to live: 62
Protocol: ICMP (1)
▸ Header checksum: 0xcdbe [correct]
Source: 172.16.10.1 (172.16.10.1)
Destination: 172.16.12.201 (172.16.12.201)
▸ Internet Control Message Protocol
```

Exercises:

Send http packets from Ha to Hd and capture Wireshark for Ha and Hd files.