NAME : DIVYANSHU SHARMA

PES1UG20CS806

CN LAB : 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.

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@CSELAB: ~
student@CSELAB:~$ ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp2s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
group default qlen 1000
    link/ether b8:ae:ed:a5:a5:d4 brd ff:ff:ff:ff:ff:ff
    inet 172.16.10.1/24 brd 172.16.10.255 scope global enp2s0
       valid_lft forever preferred_lft forever
    inet6 fe80::67ef:4c43:a82c:b073/64 scope link
       valid_lft forever preferred_lft forever
```

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@CSELAB: ~
puter
nt@CSELAB:~$ ifconfig
enp2s0    Link encap:Ethernet  HWaddr b8:ae:ed:a5:a5:e4
          inet addr:172.16.10.201  Bcast:172.16.10.255  Mask:255.255.255.0
          inet6 addr: fe80::58a4:9eab:6935:dea4/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:39 errors:0 dropped:0 overruns:0 frame:0
          TX packets:101 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:5247 (5.2 KB)  TX bytes:13288 (13.2 KB)

enxd03745b8dbb4 Link encap:Ethernet  HWaddr d0:37:45:b8:db:b4
          inet addr:172.16.11.1  Bcast:172.16.11.255  Mask:255.255.255.0
          inet6 addr: fe80::6a1e:fe5c:914:de01/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:65 errors:0 dropped:0 overruns:0 frame:0
          TX packets:159 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:6051 (6.0 KB)  TX bytes:25225 (25.2 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:2212 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2212 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:166112 (166.1 KB)  TX bytes:166112 (166.1 KB)
```

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

Step 4: Assign the IP address to the Hd.

$ sudo ip addr add 172.16.12.201/24 dev eth1

$ ip addr show

Note 1: As we know the machines are physically on the same LAN, so we can get ICMP redirect messages from other machines. Now disable accepting the ICMP redirect packets. On host machines Ha and Hd, give the following command:

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

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

Note 2: As we know the machines are physically on the same LAN, so we can get ICMP redirect messages from other machines.Thus, disable sending of the ICMP redirect packets by these routers with aliased interfaces. Give below command in router machines R1 and R2.

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

```
student@CSELAB:~$ sudo sysctl -w net.ipv4.conf.all.send_redirects=0
[sudo] password for student:
net.ipv4.conf.all.send_redirects = 0
```

Task 2: Converting machines B and C into routers.

Note 1: Check if IP forwarding is enabled or not. **$** sysctl net.ipv4.ip_forward 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@CSELAB:~$ sudo sysctl -w net.ipv4.ip_forward=1
net.ipv4.ip_forward = 1
student@CSELAB:~$ sudo ip route add 172.16.12.0/24 via
```

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

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

```
student@CSELAB:~$ 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_seq=1 ttl=64 time=0.023 ms
64 bytes from 172.16.10.1: icmp_seq=2 ttl=64 time=0.020 ms
64 bytes from 172.16.10.1: icmp_seq=3 ttl=64 time=0.022 ms
64 bytes from 172.16.10.1: icmp_seq=4 ttl=64 time=0.023 ms
64 bytes from 172.16.10.1: icmp_seq=5 ttl=64 time=0.022 ms
64 bytes from 172.16.10.1: icmp_seq=6 ttl=64 time=0.022 ms
^C
--- 172.16.10.1 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5002ms
rtt min/avg/max/mdev = 0.020/0.022/0.023/0.001 ms
student@CSELAB:~$
```

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

```
student@CSELAB:~$ 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_seq=1 ttl=64 time=1.48 ms
64 bytes from 172.16.12.1: icmp_seq=2 ttl=64 time=0.790 ms
64 bytes from 172.16.12.1: icmp_seq=3 ttl=64 time=0.818 ms
64 bytes from 172.16.12.1: icmp_seq=4 ttl=64 time=0.821 ms
^C
--- 172.16.12.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 2999ms
rtt min/avg/max/mdev = 0.790/0.979/1.488/0.294 ms
student@CSELAB:~$
```

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@CSELAB:~$ ip route show
169.254.0.0/16 dev enp2s0  scope link  metric 1000
172.16.10.0/24 dev enp2s0  proto kernel  scope link  src 172.16.10.1  metric 100

172.16.12.0/24 via 172.16.10.201 dev enp2s0
student@CSELAB:~$ sudo ip route add 172.16.11.0/24 via 172.16.10.201
student@CSELAB:~$ ip route show
169.254.0.0/16 dev enp2s0  scope link  metric 1000
172.16.10.0/24 dev enp2s0  proto kernel  scope link  src 172.16.10.1  metric 100

172.16.11.0/24 via 172.16.10.201 dev enp2s0
172.16.12.0/24 via 172.16.10.201 dev enp2s0
student@CSELAB:~$
```

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@CSELAB:~$ sudo ip route add 172.16.12.0/24 via 172.16.11.201
student@CSELAB:~$ ip route show
169.254.0.0/16 dev enp2s0  scope link  metric 1000
172.16.10.0/24 dev enp2s0  proto kernel  scope link  src 172.16.10.201  metric 100
172.16.11.0/24 dev enxd03745b8dbb4  proto kernel  scope link  src 172.16.11.1  metric 100
172.16.12.0/24 via 172.16.11.201 dev enxd03745b8dbb4
```

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@CSELAB:~$ sudo ip route add 172.16.10.0/24 via 172.16.11.1
student@CSELAB:~$ ip route show
169.254.0.0/16 dev enp2s0  scope link  metric 1000
172.16.10.0/24 via 172.16.11.1 dev enp2s0
172.16.11.0/24 dev enp2s0  proto kernel  scope link  src 172.16.11.201  metric 100
172.16.12.0/24 dev enxd03745b8d975  proto kernel  scope link  src 172.16.12.1  metric 100
student@CSELAB:~$
```

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@CSELAB:~$ ip route show
169.254.0.0/16 dev enp2s0  scope link  metric 1000
172.16.10.0/24 via 172.16.12.1 dev enp2s0
172.16.11.0/24 via 172.16.12.1 dev enp2s0
172.16.12.0/24 dev enp2s0  proto kernel  scope link  src 172.16.12.201  metric 1
00
```

Task 5: After adding routing table. 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

```
rtt min/avg/max/mdev = 0.689/0.802/0.924/0.074 ms
student@CSELAB:~$ 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_seq=1 ttl=62 time=0.934 ms
64 bytes from 172.16.12.201: icmp_seq=2 ttl=62 time=0.849 ms
64 bytes from 172.16.12.201: icmp_seq=3 ttl=62 time=0.853 ms
64 bytes from 172.16.12.201: icmp_seq=4 ttl=62 time=0.922 ms
64 bytes from 172.16.12.201: icmp_seq=5 ttl=62 time=0.869 ms
64 bytes from 172.16.12.201: icmp_seq=6 ttl=62 time=0.724 ms
64 bytes from 172.16.12.201: icmp_seq=7 ttl=62 time=0.790 ms
64 bytes from 172.16.12.201: icmp_seq=8 ttl=62 time=0.722 ms
64 bytes from 172.16.12.201: icmp_seq=9 ttl=62 time=0.753 ms
^C
--- 172.16.12.201 ping statistics ---
9 packets transmitted, 9 received, 0% packet loss, time 8000ms
rtt min/avg/max/mdev = 0.722/0.824/0.934/0.075 ms
student@CSELAB:~$
```

Step 2: Testing path from Hd and Ha

$ ping 172.16.12.1 and $ ping 172.16.12.201

```
connect: Network is unreachable
student@CSELAB:~$ 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_seq=1 ttl=64 time=1.48 ms
64 bytes from 172.16.12.1: icmp_seq=2 ttl=64 time=0.790 ms
64 bytes from 172.16.12.1: icmp_seq=3 ttl=64 time=0.818 ms
64 bytes from 172.16.12.1: icmp_seq=4 ttl=64 time=0.821 ms
^C
--- 172.16.12.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 2999ms
rtt min/avg/max/mdev = 0.790/0.979/1.488/0.294 ms
```

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

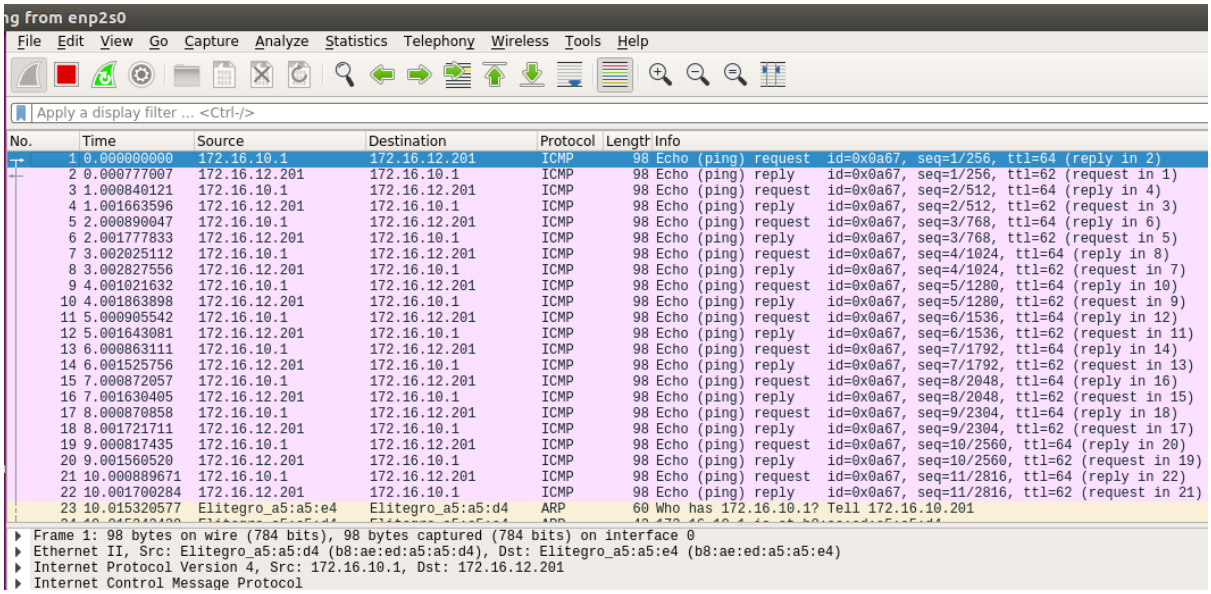
```
172.16.12.0/24 via 172.16.11.201 dev enxd03745b8dbb4
student@CSELAB:~$ ip neigh show
172.16.11.201 dev enxd03745b8dbb4 lladdr b8:ae:ed:a5:a5:a6 REACHABLE
172.16.10.1 dev enp2s0 lladdr b8:ae:ed:a5:a5:d4 REACHABLE
```

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

Step 1: Capture packets from Ha and Hd.

At Ha:

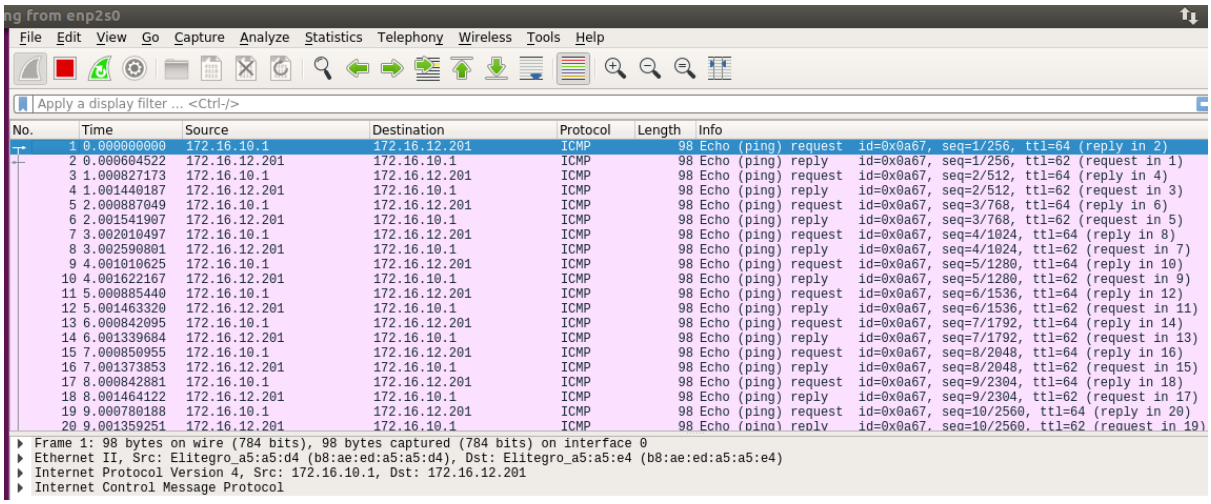
T1: $ sudo wireshark T2: $ ping 172.16.12.201



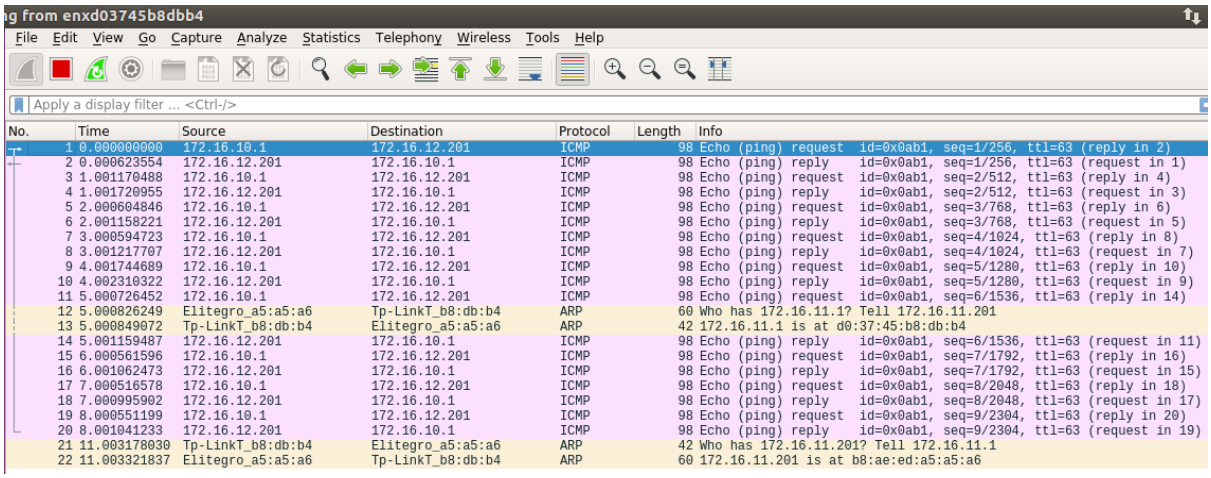
Step 2: Capture packets from R1 using both eth1 and eth2 interfaces.

$ sudo wireshark

At eth1:



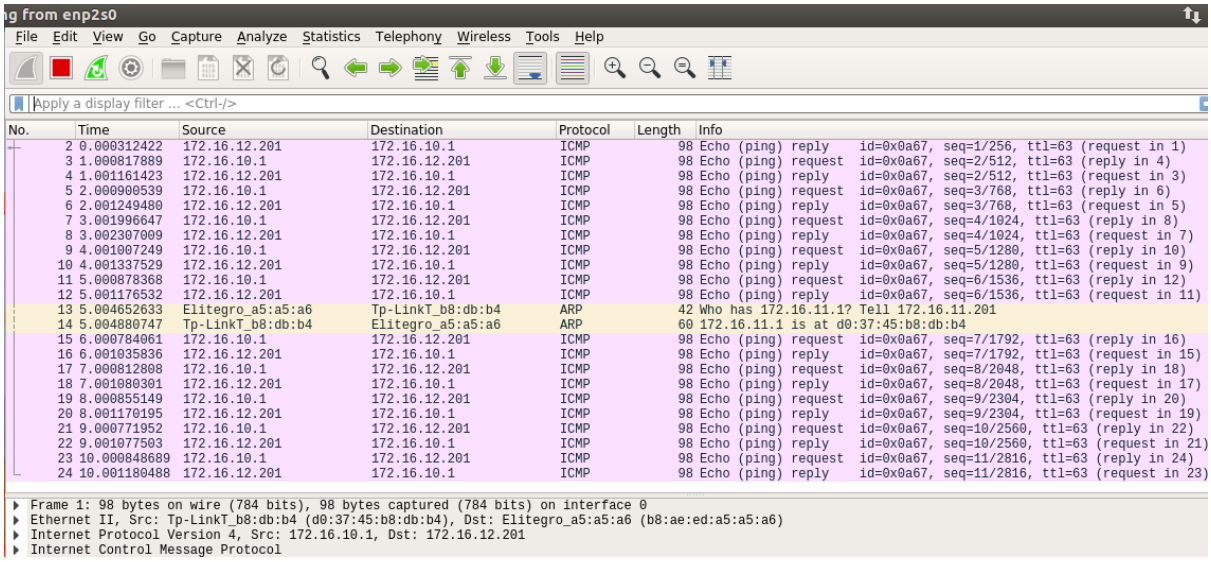
At eth2:



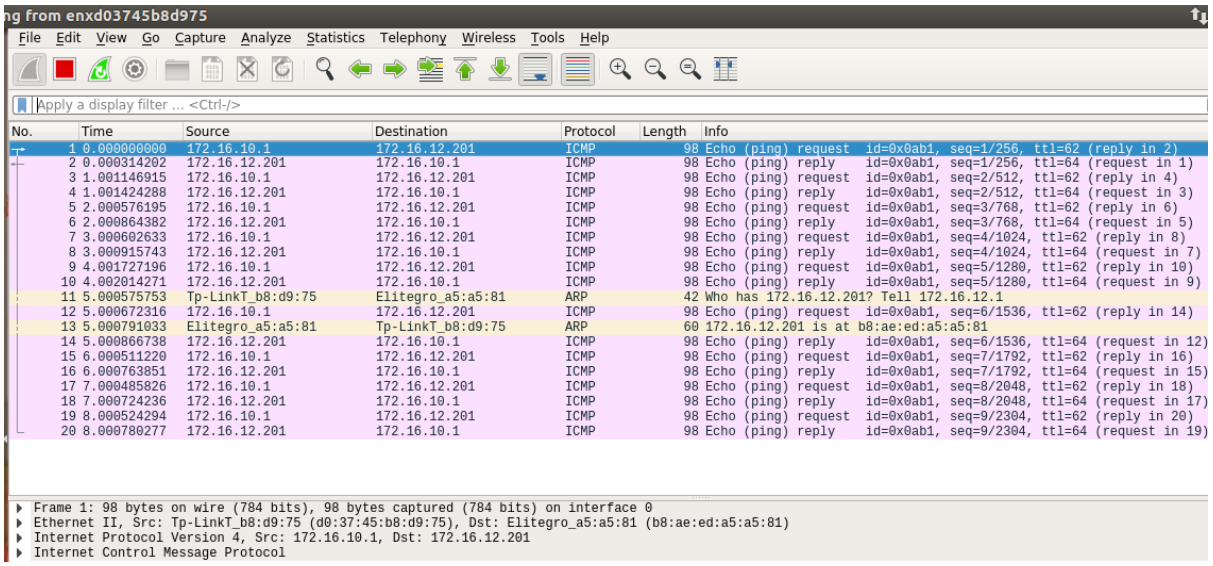
Step 3: Capture packets from R2 using both eth1 and eth2 interfaces.

$ sudo wireshark

At eth1:



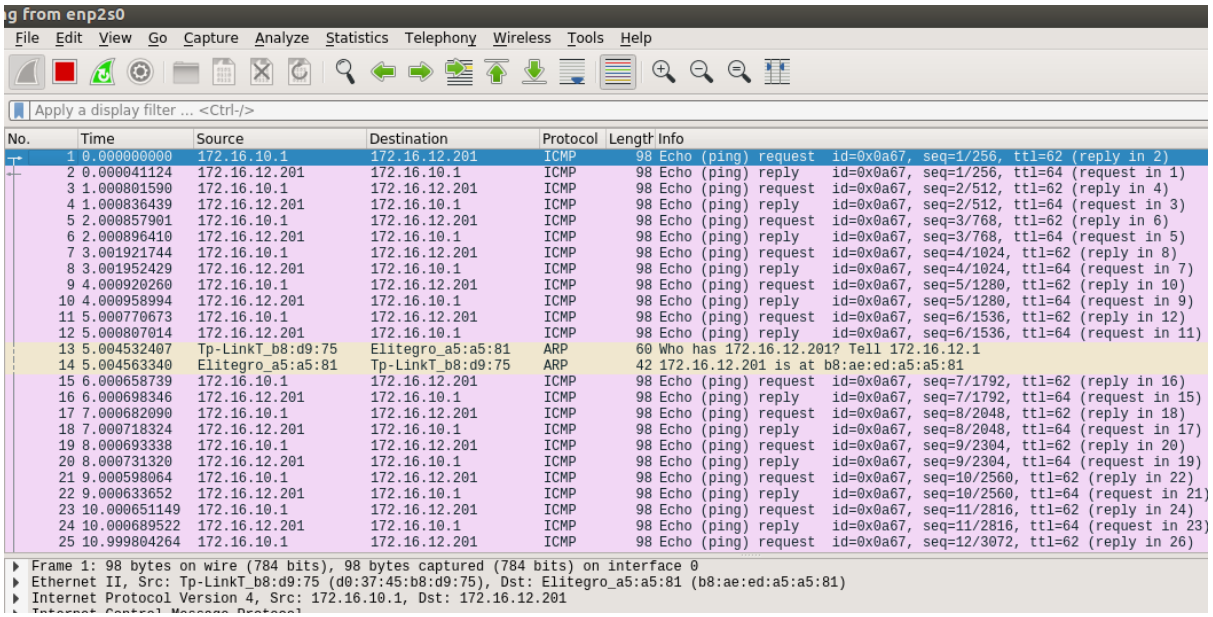
At eth2:



Step 4: Capture packets from Hd and Ha.

At Hd:

T1: $ sudo wireshark



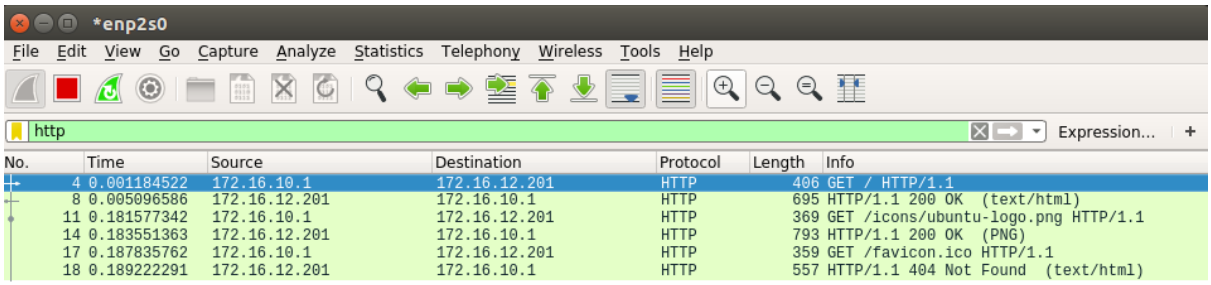
Exercises: Send http packets from Ha to Hd and capture Wireshark for Ha and Hd files

Apache Service at Ha

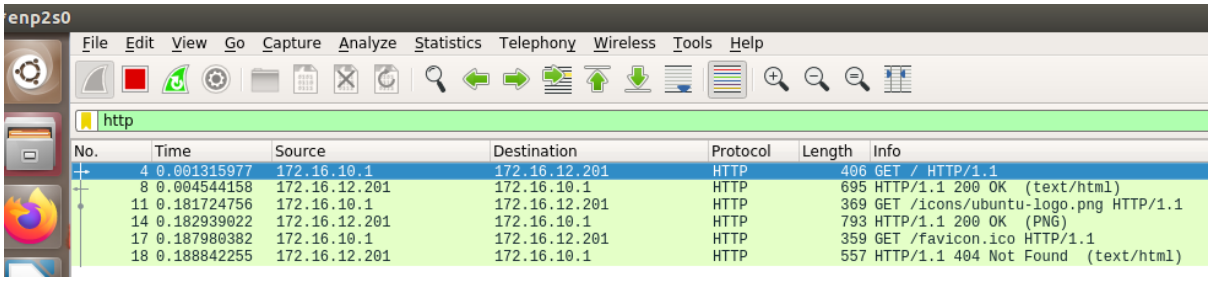
```
● apache2.service - LSB: Apache2 web server
   Loaded: loaded (/etc/init.d/apache2; bad; vendor preset: enabled)
  Drop-In: /lib/systemd/system/apache2.service.d
           └─apache2-systemd.conf
   Active: active (running) since Thu 2021-04-01 14:43:52 IST; 25min ago
     Docs: man:systemd-sysv-generator(8)
  Process: 975 ExecStart=/etc/init.d/apache2 start (code=exited, status=0/SUCCESS)
   CGroup: /system.slice/apache2.service
           ├─1083 /usr/sbin/apache2 -k start
           ├─1093 /usr/sbin/apache2 -k start
           ├─1094 /usr/sbin/apache2 -k start
           ├─1095 /usr/sbin/apache2 -k start
           ├─1096 /usr/sbin/apache2 -k start
           └─1097 /usr/sbin/apache2 -k start

Apr 01 14:43:47 CSELAB systemd[1]: Starting LSB: Apache2 web server...
Apr 01 14:43:49 CSELAB apache2[975]:  * Starting Apache httpd web server apache2
Apr 01 14:43:51 CSELAB apache2[975]: AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1.
Apr 01 14:43:52 CSELAB apache2[975]:  *
Apr 01 14:43:52 CSELAB systemd[1]: Started LSB: Apache2 web server.
```

R1 Capturing HTTP packet



R2 Capturing HTTP packet



Hd Capturing HTTP packet