Computer Networks Lab

Week 7

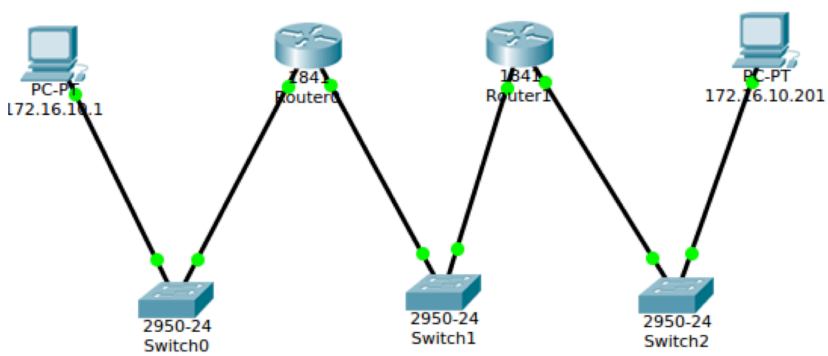
Vishal R

PES1UG19CS571

I Section

IPv4 Addressing and Static Routing

I. Topology



The following topology was used to rig up the network connections between four systems.

1. Task 1 : Assigning IP addresses to all four end systems (Ha,R1,R2,Hd)

1. 1. Assign IP address to system Ha

The IP address on end system Ha were manually added using the Edit Connections Menu in Linux.

End System Ha was assigned the IP address `172.16.10.1/24.`

```
student@pesu-OptiPlex-3070:~$ ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t qlen 1000
    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: enp1s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
group default qlen 1000
    link/ether 00:4e:01:a0:6b:6f brd ff:ff:ff:ff:ff:ff
    inet 172.16.10.1/24 brd 172.16.10.255 scope global enp1s0
       valid_lft forever preferred_lft forever
    inet6 fe80::d633:3a8f:8d02:ff29/64 scope link
       valid_lft forever preferred_lft forever
student@pesu-OptiPlex-3070:~$
```

1. 2. Assign IP address to system R1

On system R1, we will set the IP address manually. For the internal connection, we will use `172.16.10.201/24` as the IP address with device being set as `enp1s0`.

For the external connection (connection between routers R1 and R2), we will use `172.16.11.1/24` as the IP address with device being set as `enx00594d6e8e19`.

```
student@pesu-OptiPlex-3070:~$ ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    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: enp1s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:4e:01:a4:1e:7b brd ff:ff:ff:ff:ff:ff
    inet 172.16.10.201/24 brd 172.16.10.255 scope global enp1s0
       valid_lft forever preferred_lft forever
    inet6 fe80::8aff:b205:d4e4:1d10/64 scope link
       valid_lft forever preferred_lft forever
3: enx00594d6e8e19: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:59:4d:6e:8e:19 brd ff:ff:ff:ff:ff:ff
    inet 172.16.11.1/24 brd 172.16.11.255 scope global enx00594d6e8e19
       valid_lft forever preferred_lft forever
    inet6 fe80::e2df:1b81:5fb2:c84f/64 scope link
       valid_lft forever preferred_lft forever
```

1. 3. Assign IP address to system R2

On system R2, we will set the IP address manually as we did for others.
For internal connection, we will use `172.16.12.1/24` as the IP address with device being set as `enp1s0`.

For the external connection, we will use `172.16.11.201/24` as the IP address with device being set as `enx000ec6877201`.

```
student@pesu-OptiPlex-3070:~$ ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    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: enp1s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:4e:01:a0:63:92 brd ff:ff:ff:ff:ff:ff
    inet 172.16.12.1/24 brd 172.16.12.255 scope global enp1s0
       valid_lft forever preferred_lft forever
    inet6 fe80::e853:1969:648b:2214/64 scope link
       valid_lft forever preferred_lft forever
3: enx000ec6877201: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:0e:c6:87:72:01 brd ff:ff:ff:ff:ff:ff
    inet 172.16.11.201/24 brd 172.16.11.255 scope global enx000ec6877201
       valid_lft forever preferred_lft forever
    inet6 fe80::177f:1a1d:d40:ff64/64 scope link
       valid_lft forever preferred_lft forever
```

1. 4. Assign IP address to end system Hd

On system Hd, we will set the IP address as `172.16.12.201/24` using the manual method.

Finally, on host machines Ha and Hd, we will type the following command,
`$ sudo sysctl -w net.ipv4.conf.all.accept_redirects=0`. This will disable accepting the ICMP redirect packets.

Similarly, on the systems R1 and R2, we will type the following command,
`$ sudo sysctl -w net.ipv4.conf.all.send_redirects=0`. This will disable sending of the ICMP redirect packets by these routers with aliased interfaces.

2. Task 2 : Converting systems R1 and R2 to routers

In both the systems, we will enable IP forwarding. To do this, we will execute the command,
`$ sudo sysctl -w net.ipv4.ip_forward=1` on both the systems.

3. Task 3 : Verify the connection between Ha and Hd

Testing the connection of systems within the same networks.

At Ha : `$ ping 172.16.10.1 (local network)`

```
student@pesu-OptiPlex-3070:~$ ping -c 3 172.16.10.1
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=62 time=1.72 ms
64 bytes from 172.16.10.1: icmp_seq=2 ttl=62 time=1.68 ms
64 bytes from 172.16.10.1: icmp_seq=3 ttl=62 time=1.29 ms

--- 172.16.10.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 1.290/1.566/1.724/0.201 ms
```

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=0.301 ms
64 bytes from 172.16.12.1: icmp_seq=2 ttl=64 time=0.229 ms
64 bytes from 172.16.12.1: icmp_seq=3 ttl=64 time=0.220 ms
64 bytes from 172.16.12.1: icmp_seq=4 ttl=64 time=0.234 ms
64 bytes from 172.16.12.1: icmp_seq=5 ttl=64 time=0.227 ms
64 bytes from 172.16.12.1: icmp_seq=6 ttl=64 time=0.247 ms
64 bytes from 172.16.12.1: icmp_seq=7 ttl=64 time=0.248 ms
64 bytes from 172.16.12.1: icmp_seq=8 ttl=64 time=0.247 ms
64 bytes from 172.16.12.1: icmp_seq=9 ttl=64 time=0.251 ms
^C
--- 172.16.12.1 ping statistics ---
9 packets transmitted, 9 received, 0% packet loss, time 7999ms
rtt min/avg/max/mdev = 0.220/0.244/0.301/0.030 ms
```

4. Task 4 : Insert routing table entries on each system to direct IPv4 packets to ping across the networks.

In order for packets to be able to reach other end system, we need to add routing tables to both the routers and both the end systems as well.

At Ha : The following commands were entered.

```
$ 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@pesu-OptiPlex-3070:~$ ip route show
169.254.0.0/16 dev enp1s0  scope link  metric 1000
172.16.10.0/24 dev enp1s0  proto kernel  scope link  src 172.16.10.1

172.16.11.0/24 via 172.16.10.201 dev enp1s0
172.16.12.0/24 via 172.16.10.201 dev enp1s0
```

At R1 : The following commands were entered

```
$ sudo ip route add 172.16.12.0/24 via 172.16.11.201
$ ip route show
```

```
student@pesu-OptiPlex-3070:~$ sudo ip route add 172.16.12.0/24 via 172.16.11.201
student@pesu-OptiPlex-3070:~$ ip route show
169.254.0.0/16 dev enx00594d6e8e19  scope link  metric 1000
172.16.10.0/24 dev enp1s0  proto kernel  scope link  src 172.16.10.201  metric 100
172.16.11.0/24 dev enx00594d6e8e19  proto kernel  scope link  src 172.16.11.1  metric 100
172.16.12.0/24 via 172.16.11.201 dev enx00594d6e8e19
student@pesu-OptiPlex-3070:~$
```

At R2 : The following commands were executed

```
$ sudo ip route add 172.16.10.0/24 via 172.16.11.1
$ ip route show
```

```
student@pesu-OptiPlex-3070:~$ ip route show
169.254.0.0/16 dev enp1s0  scope link  metric 1000
172.16.10.0/24 via 172.16.11.1 dev enx000ec6877201
172.16.11.0/24 dev enx000ec6877201  proto kernel  scope link  src 172.16.11.201  metric 100
172.16.12.0/24 dev enp1s0  proto kernel  scope link  src 172.16.12.1  metric 100
```

At Hd : The following commands were executed

```
$ 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@pesu-OptiPlex-3070:~$ ip route show
169.254.0.0/16 dev enp1s0  scope link  metric 1000
172.16.10.0/24 via 172.16.12.1 dev enp1s0
172.16.11.0/24 via 172.16.12.1 dev enp1s0
172.16.12.0/24 dev enp1s0  proto kernel  scope link  src 172.16.12.201  metric 100
```

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

5. 1. Testing connectivity between Ha and Hd

To test if all our configuration steps we performed above is correct or not, we will ping the other end system from the first end system

To do this, **$ ping 172.16.12.201** was executed in the terminal of Ha

```
student@pesu-OptiPlex-3070:~$ 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=1.93 ms
64 bytes from 172.16.12.201: icmp_seq=2 ttl=62 time=1.42 ms
64 bytes from 172.16.12.201: icmp_seq=3 ttl=62 time=1.78 ms
64 bytes from 172.16.12.201: icmp_seq=4 ttl=62 time=2.16 ms
64 bytes from 172.16.12.201: icmp_seq=5 ttl=62 time=2.22 ms
64 bytes from 172.16.12.201: icmp_seq=6 ttl=62 time=2.01 ms
^C
--- 172.16.12.201 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5007ms
rtt min/avg/max/mdev = 1.427/1.924/2.226/0.270 ms
```

5. 2. Testing connectivity between Hd and Ha

To test the connectivity between Hd and Ha, we will ping Ha from Hd and check if we get a reply.

```
student@pesu-OptiPlex-3070:~$ ping -c 3 172.16.10.1
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=62 time=1.72 ms
64 bytes from 172.16.10.1: icmp_seq=2 ttl=62 time=1.68 ms
64 bytes from 172.16.10.1: icmp_seq=3 ttl=62 time=1.29 ms

--- 172.16.10.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 1.290/1.566/1.724/0.201 ms
```

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

To check the neighbor, we need to execute the command `$ ip neigh show`

At Ha : `$ ip neigh show`

```
student@pesu-OptiPlex-3070:~$ ip neigh show
172.16.10.201 dev enp1s0 lladdr 00:4e:01:a4:1e:7b STALE
student@pesu-OptiPlex-3070:~$
```

At R1 : `$ ip neigh show`

```
student@pesu-OptiPlex-3070:~$ ip neigh show
172.16.10.1 dev enp1s0 lladdr 00:4e:01:a0:6b:6f STALE
172.16.11.201 dev enx00594d6e8e19 lladdr 00:0e:c6:87:72:01 STALE
student@pesu-OptiPlex-3070:~$
```

At R2 : `$ ip neigh show`

```
student@pesu-OptiPlex-3070:~$ ip neigh show
172.16.12.201 dev enp1s0 lladdr 00:4e:01:a4:21:17 DELAY
172.16.11.1 dev enx000ec6877201 lladdr 00:59:4d:6e:8e:19 REACHABLE
student@pesu-OptiPlex-3070:~$
```

At Hd : `$ ip neigh show`

```
student@pesu-OptiPlex-3070:~$ ip neigh
172.16.12.1 dev enp1s0 lladdr 00:4e:01:a0:63:92 REACHABLE
student@pesu-OptiPlex-3070:~$
```

7. Task 7 : Capture packets from Ha and Hd using WireShark

7. 1. Capturing packets from Ha and Hd

To capture packets, we will open wireshark on all four systems and from Ha we will ping Hd
(`$ ping 172.16.12.201`)

```
student@pesu-OptiPlex-3070:~$ 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=1.93 ms
64 bytes from 172.16.12.201: icmp_seq=2 ttl=62 time=1.42 ms
64 bytes from 172.16.12.201: icmp_seq=3 ttl=62 time=1.78 ms
64 bytes from 172.16.12.201: icmp_seq=4 ttl=62 time=2.16 ms
64 bytes from 172.16.12.201: icmp_seq=5 ttl=62 time=2.22 ms
64 bytes from 172.16.12.201: icmp_seq=6 ttl=62 time=2.01 ms
^C
--- 172.16.12.201 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5007ms
rtt min/avg/max/mdev = 1.427/1.924/2.226/0.270 ms
```

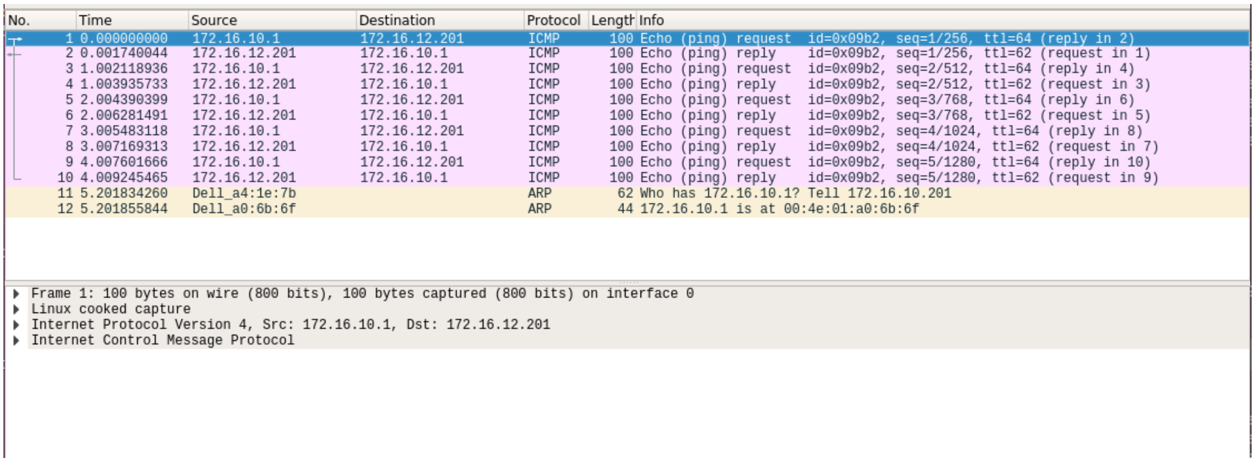


Figure : At Ha - Wireshark Packet Capture, capturing packets during the ping operation

At R1 :

We will open wireshark in both the interfaces using by internal and external connections and observe the packets captured during ping operation from Ha to Hd

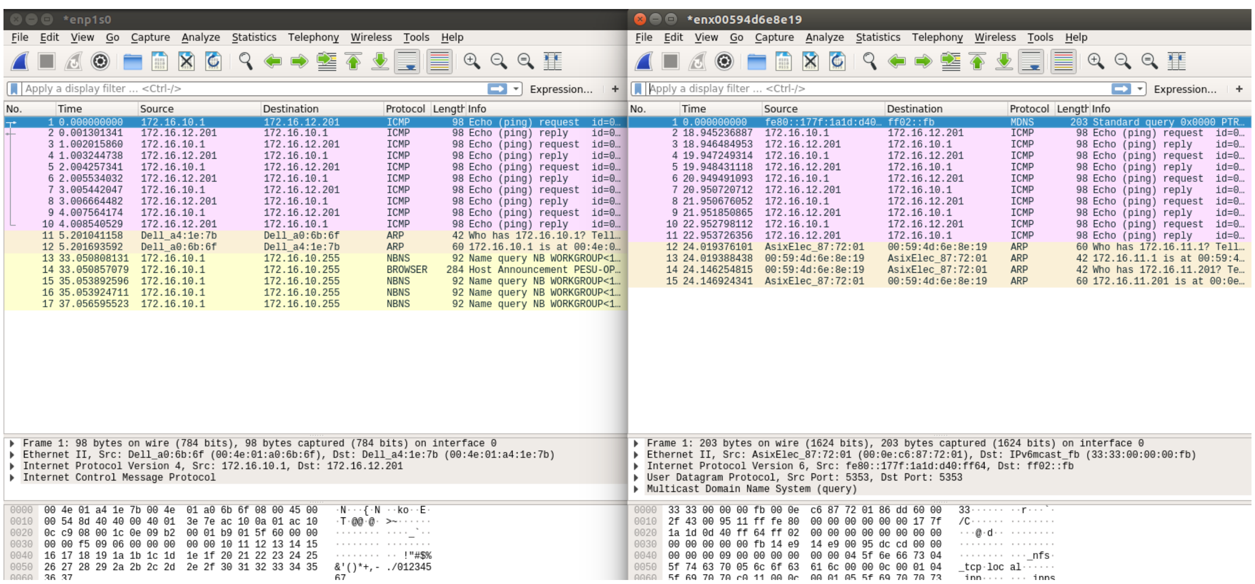


Figure : Wireshark Packet Capture in both interfaces used by internal (left) and external (right) connections.

At R2 :

We will open wireshark in both the interfaces using by internal and external connections and observe the packets captured during ping operation from Ha to Hd.

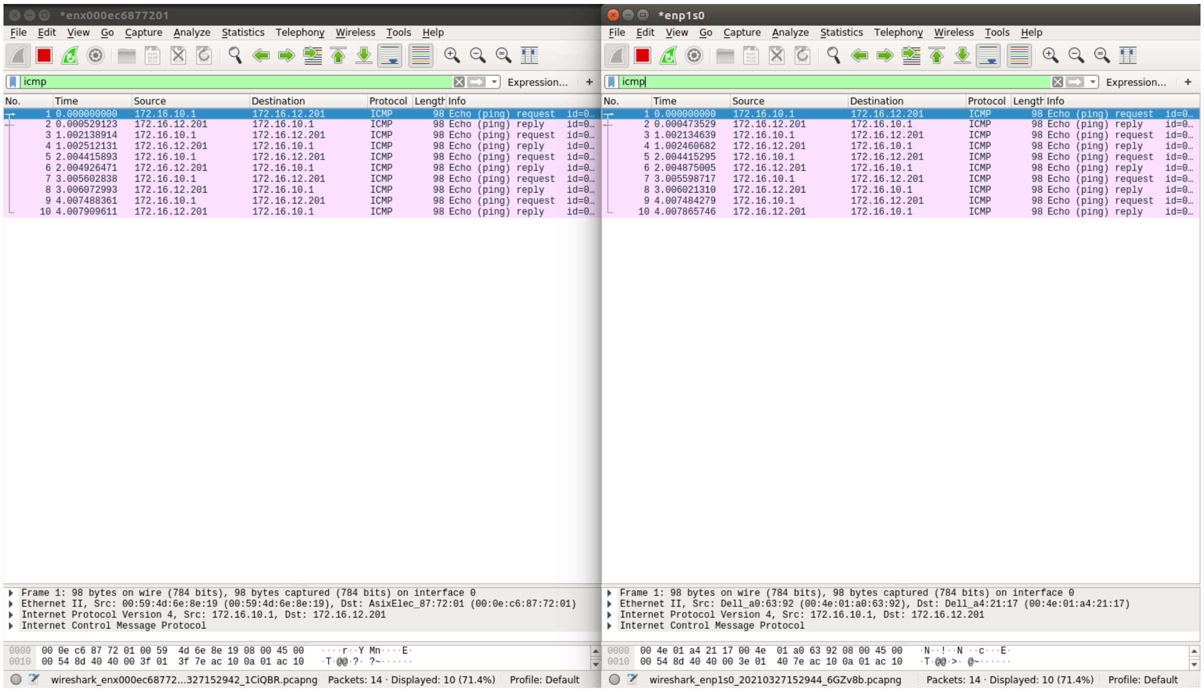


Figure : Wireshark Packet Capture in both interfaces used by internal (right) and external (left) connections.

At Hd :

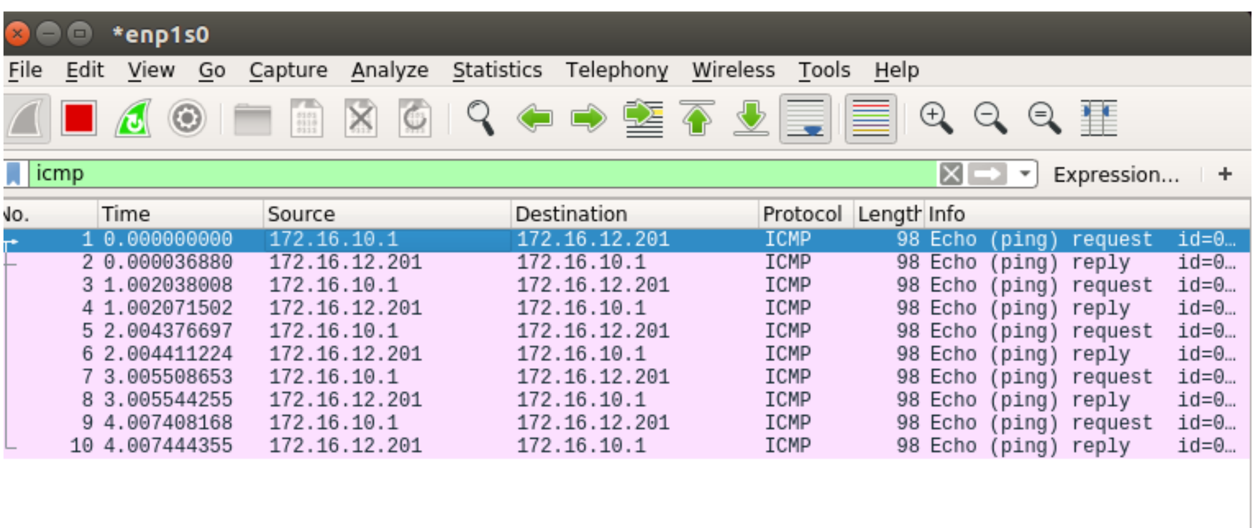


Figure : Wireshark Packet Capture at Hd capturing packets sent/received during ping.

7. 2. Capture packets from Hd and Ha

To do this, we will ping Ha from Hd and observe the packets we get in wireshark

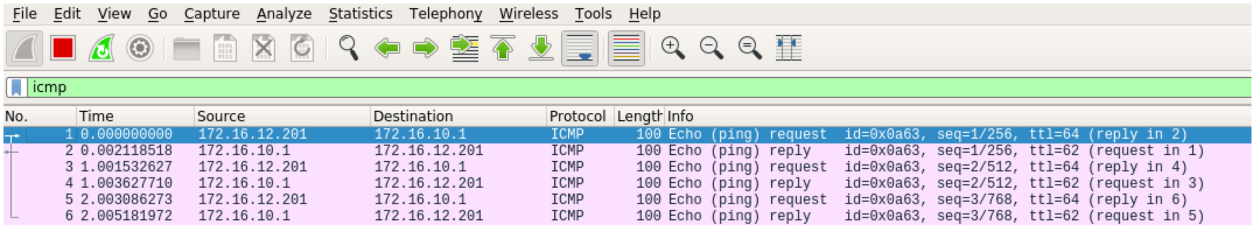


Figure : Wireshark Packet Capture at Hd during the ping operation

Additional Exercise

8. Send and Capture HTTP Packets from Ha to Hd

As we did before, we will request for a HTTP resource on the end system Hd from Ha. In the browser, we will request for `172.16.12.201/index.html` page. The packets we receive during this request will be observed in wireshark.

At Ha :

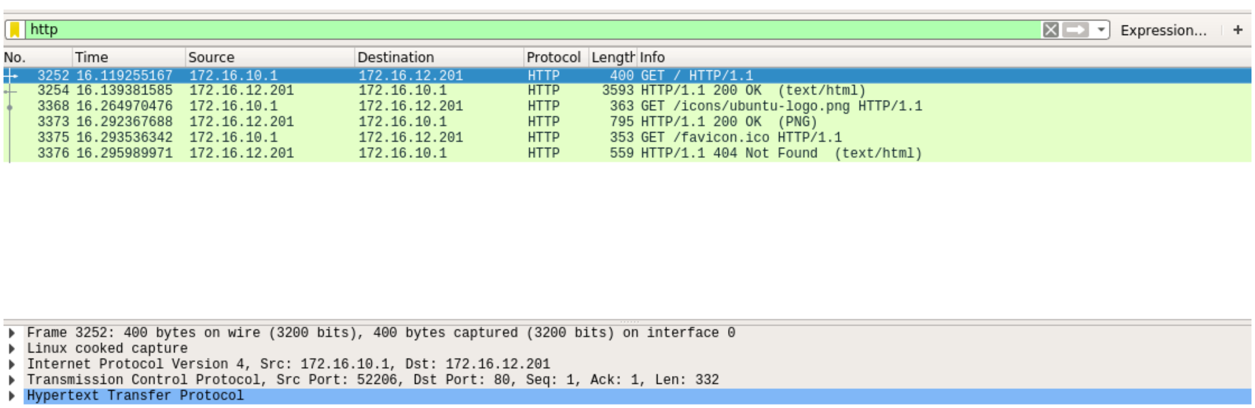


Figure : Wireshark packet capture at Ha.

At R1 :

Again we will open wireshark in both the interfaces using by internal and external connections and observe the packets captured during GET request made from Ha to Hd.

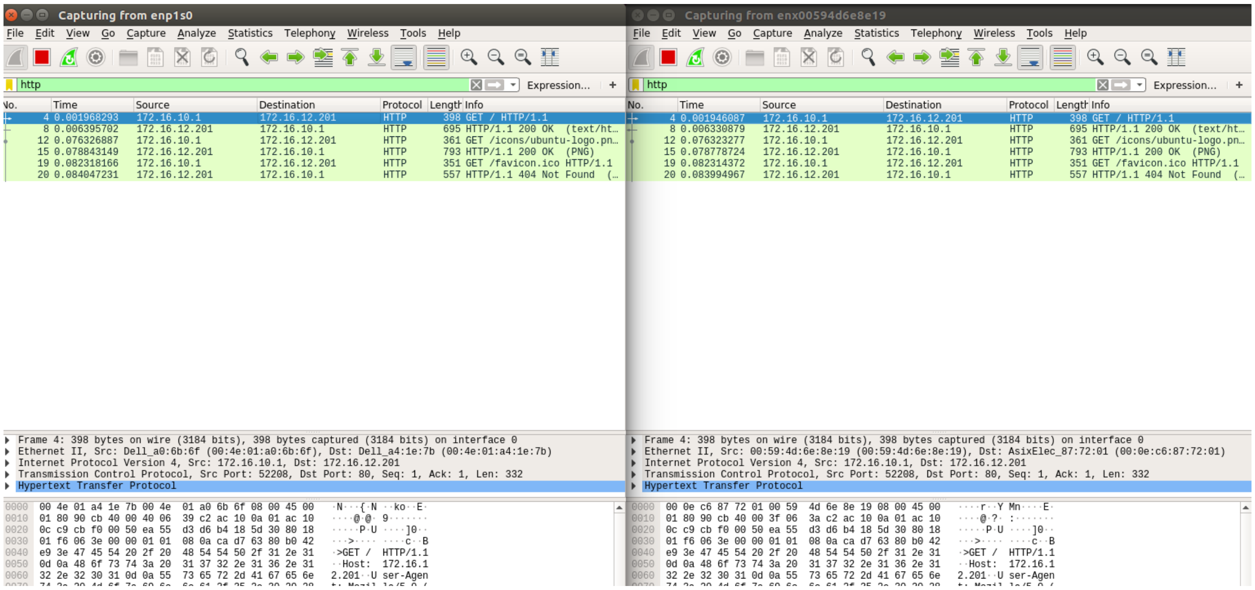


Figure : Wireshark packet capture at R1 in both interfaces internal (left) and external (right) connections.

At R2 :
As we did in R1, we will do the same in R2.

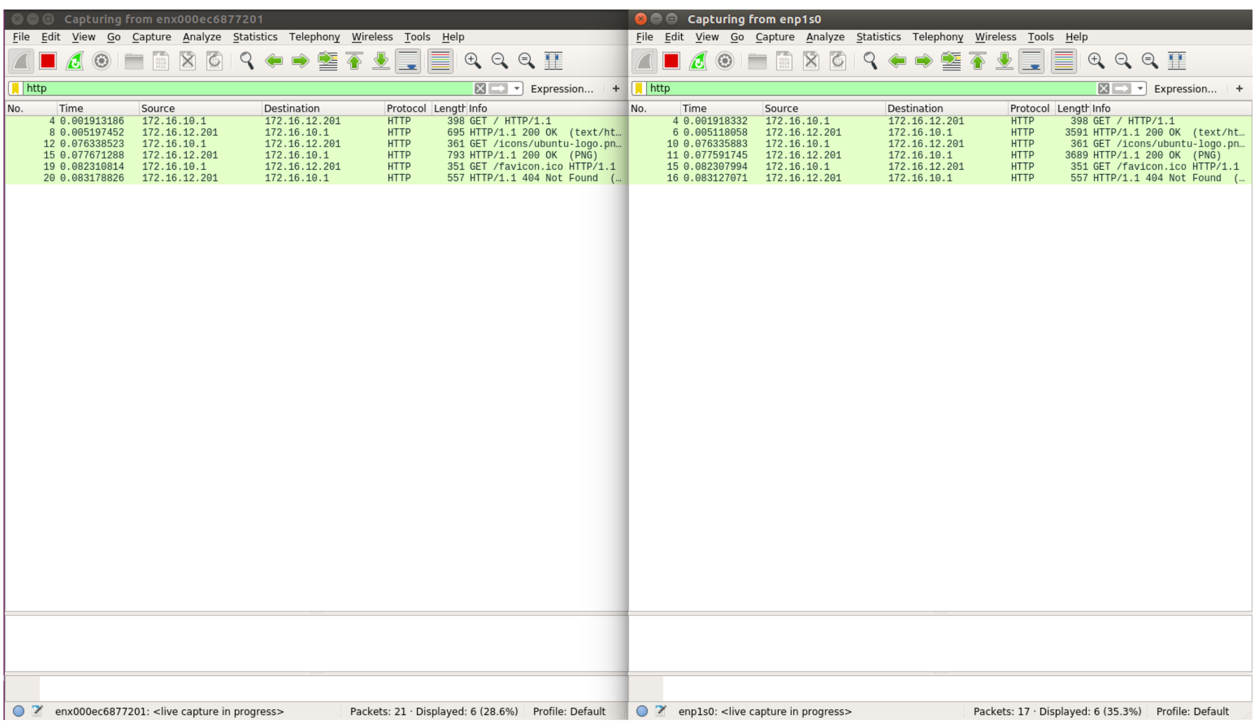


Figure : Wireshark Packet Capture at R2 in both interfaces internal (left) and external (right) connections.

At Hd :

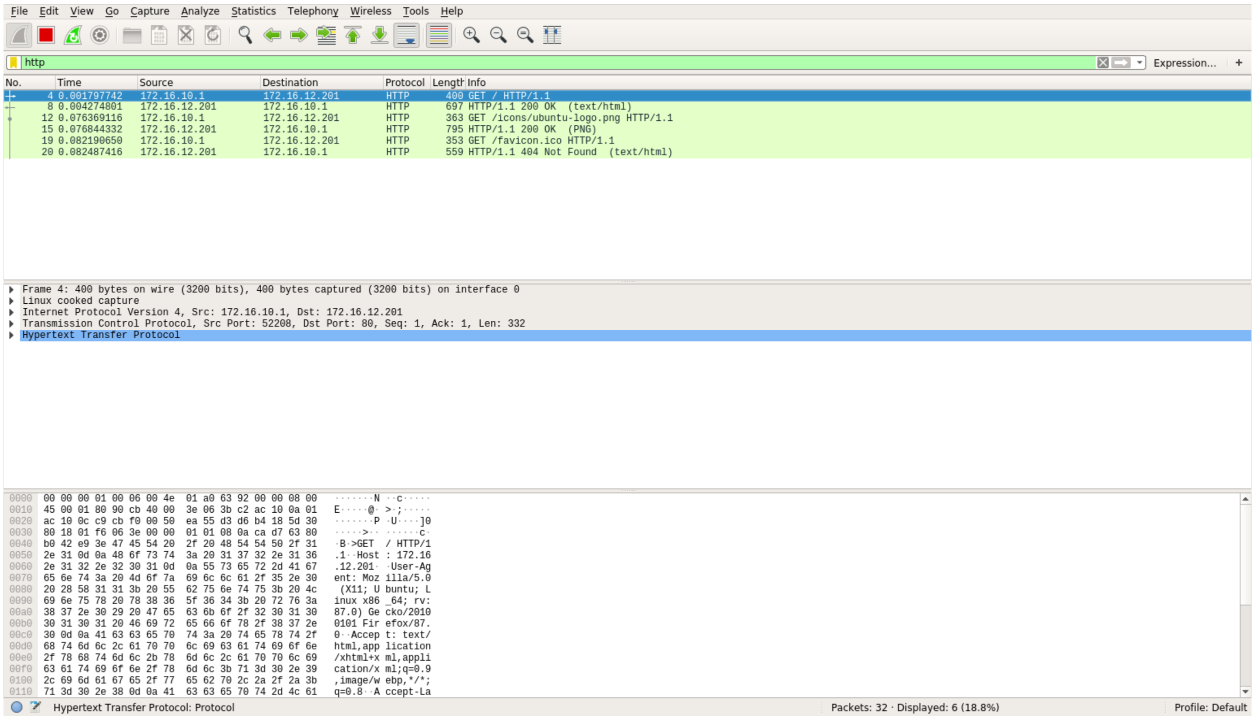


Figure : Wireshark Packet Capture at Hd showing the packets sent and received during GET request made from Ha