SCC 203 CW2 Report

Bikash Kumar Tiwary

34363718

Task 1.1: Network Measurement

Under this task, we have to measure the delay, loss and throughput in the following scenarios: a) between the two hosts connected to switch, b)two hosts connected to the hub and c) two hosts across the network(Using the router).

Measuring loss:

- We can measure loss by pinging.
- Ping uses ICMP echo messages that can be sent out to specific hosts.
- Measures end to end delay latency.
- Which also includes a measurement of packet loss by keeping how many messages were sent and how many responses received.

Measuring Delay:

- We can get delay through traceroute.
- This too uses an ICMP echo request message, but with an important modification: Time To Live(TTL) whose value is initially set to 1.
- We get 3 delay measures from each hop in the network.

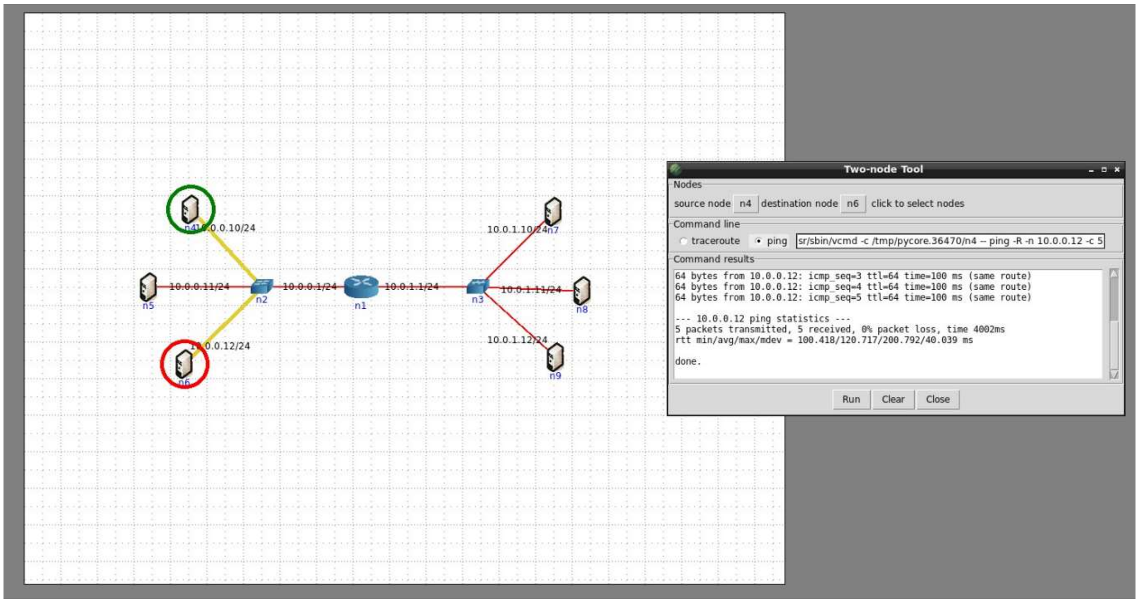
Measuring Throughput:

- Rate(bits/time unit) at which bits transfer between sender / receiver.
- Throughput is often restricted by a bottleneck
- Normally runs on a client-server model
    - One host requests the data, other serves it
    - Generate data streams across the network
    - Because we know exactly how large data is, and how long it takes to retrieve it, we can calculate throughput easily.

Below we observe the loss, delay and throughput of the asked scenarios.

a) Between two hosts connected to switch

- *Measuring loss*

- 

Nodes
source node n4 destination node n6 click to select nodes

Command line
○ traceroute ● ping `sr/sbin/vcmd -c /tmp/pycore.36470/n4 -- ping -R -n 10.0.0.12 -c 5`

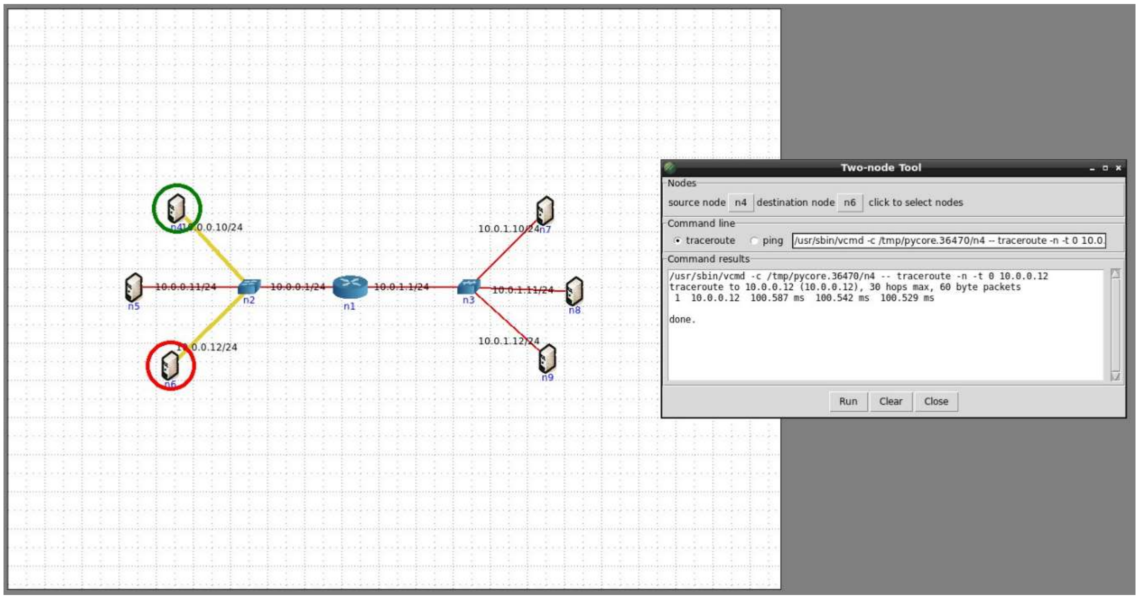
Command results

```
64 bytes from 10.0.0.12: icmp_seq=3 ttl=64 time=100 ms (same route)
64 bytes from 10.0.0.12: icmp_seq=4 ttl=64 time=100 ms (same route)
64 bytes from 10.0.0.12: icmp_seq=5 ttl=64 time=100 ms (same route)

--- 10.0.0.12 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4002ms
rtt min/avg/max/mdev = 100.418/120.717/200.792/40.039 ms

done.
```

As shown above, the packet loss is at 0% while pinging from source to destination i.e between the hosts connected to the switch, capped at 5 had 100% transmission rate.

Loss percentage in the links between the switch and host was set to null which resulted in no packet losses when pinged between two hosts connected to the switch.

- *Measuring delay*



Nodes
source node n4 destination node n6 click to select nodes

Command line
● traceroute ○ ping `/usr/sbin/vcmd -c /tmp/pycore.36470/n4 -- traceroute -n -t 0 10.0.`

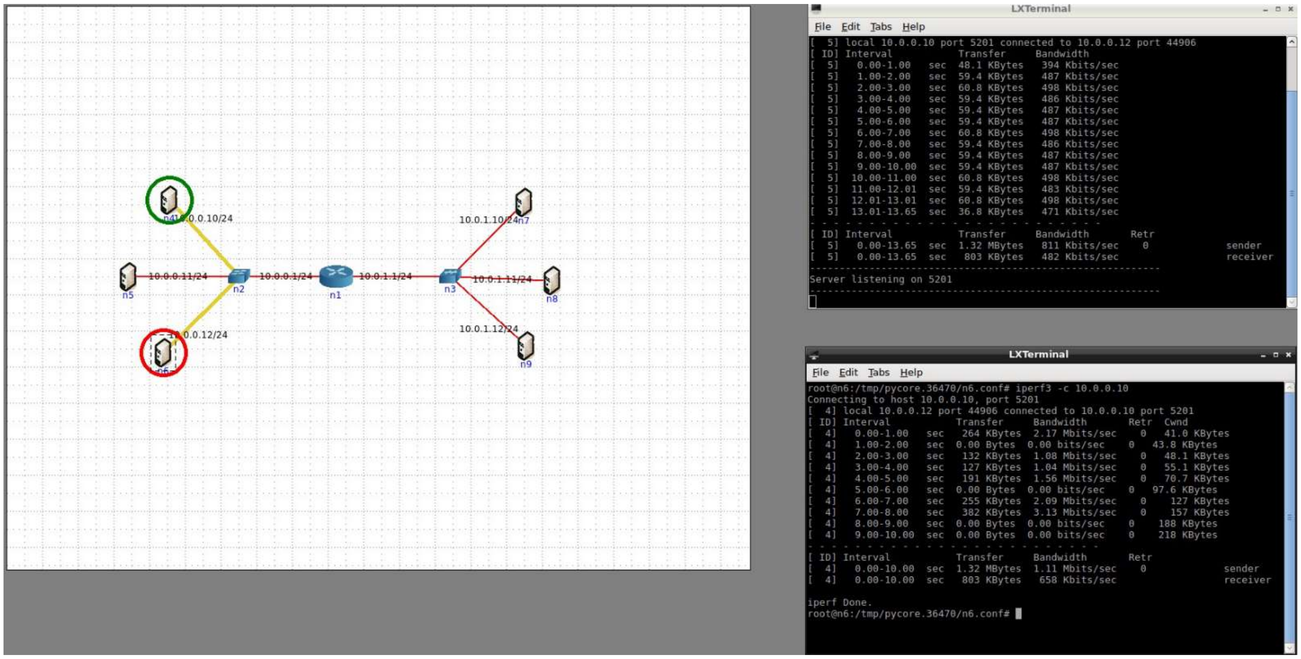
Command results

```
/usr/sbin/vcmd -c /tmp/pycore.36470/n4 -- traceroute -n -t 0 10.0.0.12
traceroute to 10.0.0.12 (10.0.0.12), 30 hops max, 60 byte packets
 1  10.0.0.12  100.587 ms  100.542 ms  100.529 ms

done.
```

Delay was measured with the help of traceroute in which the three delay measurements were approx 100ms each.

The round trip time was measured as such because of the delay for the links between hosts and switch was set to 50ms each.

- *Measuring throughput*



```
[  5] local 10.0.0.10 port 5201 connected to 10.0.0.12 port 44906
[ ID] Interval           Transfer     Bandwidth
[  5]   0.00-1.00   sec  48.1 KBytes   394 Kbits/sec
[  5]   1.00-2.00   sec  59.4 KBytes   487 Kbits/sec
[  5]   2.00-3.00   sec  60.8 KBytes   498 Kbits/sec
[  5]   3.00-4.00   sec  59.4 KBytes   486 Kbits/sec
[  5]   4.00-5.00   sec  59.4 KBytes   487 Kbits/sec
[  5]   5.00-6.00   sec  59.4 KBytes   487 Kbits/sec
[  5]   6.00-7.00   sec  60.8 KBytes   498 Kbits/sec
[  5]   7.00-8.00   sec  59.4 KBytes   486 Kbits/sec
[  5]   8.00-9.00   sec  59.4 KBytes   487 Kbits/sec
[  5]   9.00-10.00  sec  59.4 KBytes   487 Kbits/sec
[  5]  10.00-11.00  sec  60.8 KBytes   498 Kbits/sec
[  5]  11.00-12.01  sec  59.4 KBytes   483 Kbits/sec
[  5]  12.01-13.01  sec  60.8 KBytes   498 Kbits/sec
[  5]  13.01-13.65  sec  36.8 KBytes   471 Kbits/sec
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Retr
[  5]   0.00-13.65  sec  1.32 MBytes   811 Kbits/sec    0             sender
[  5]   0.00-13.65  sec   803 KBytes   482 Kbits/sec                  receiver
-----------------------------------------------------------
Server listening on 5201
-----------------------------------------------------------
```

```
root@n6:/tmp/pycore.36470/n6.conf# iperf3 -c 10.0.0.10
Connecting to host 10.0.0.10, port 5201
[  4] local 10.0.0.12 port 44906 connected to 10.0.0.10 port 5201
[ ID] Interval           Transfer     Bandwidth       Retr  Cwnd
[  4]   0.00-1.00   sec   264 KBytes  2.17 Mbits/sec    0   41.0 KBytes
[  4]   1.00-2.00   sec  0.00 Bytes  0.00 bits/sec    0   43.8 KBytes
[  4]   2.00-3.00   sec   132 KBytes  1.08 Mbits/sec    0   48.1 KBytes
[  4]   3.00-4.00   sec   127 KBytes  1.04 Mbits/sec    0   55.1 KBytes
[  4]   4.00-5.00   sec   191 KBytes  1.56 Mbits/sec    0   70.7 KBytes
[  4]   5.00-6.00   sec  0.00 Bytes  0.00 bits/sec    0   97.6 KBytes
[  4]   6.00-7.00   sec   255 KBytes  2.09 Mbits/sec    0    127 KBytes
[  4]   7.00-8.00   sec   382 KBytes  3.13 Mbits/sec    0    157 KBytes
[  4]   8.00-9.00   sec  0.00 Bytes  0.00 bits/sec    0    188 KBytes
[  4]   9.00-10.00  sec  0.00 Bytes  0.00 bits/sec    0    218 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Retr
[  4]   0.00-10.00  sec  1.32 MBytes  1.11 Mbits/sec    0             sender
[  4]   0.00-10.00  sec   803 KBytes   658 Kbits/sec                  receiver

iperf Done.
root@n6:/tmp/pycore.36470/n6.conf#
```

Bottlenecks affect network performance by slowing down the flow of information transmitted across networks.

Since the bandwidth across the links between host and switch is evenly distributed (i.e 512kbps).

The network between the hosts is the bottleneck.

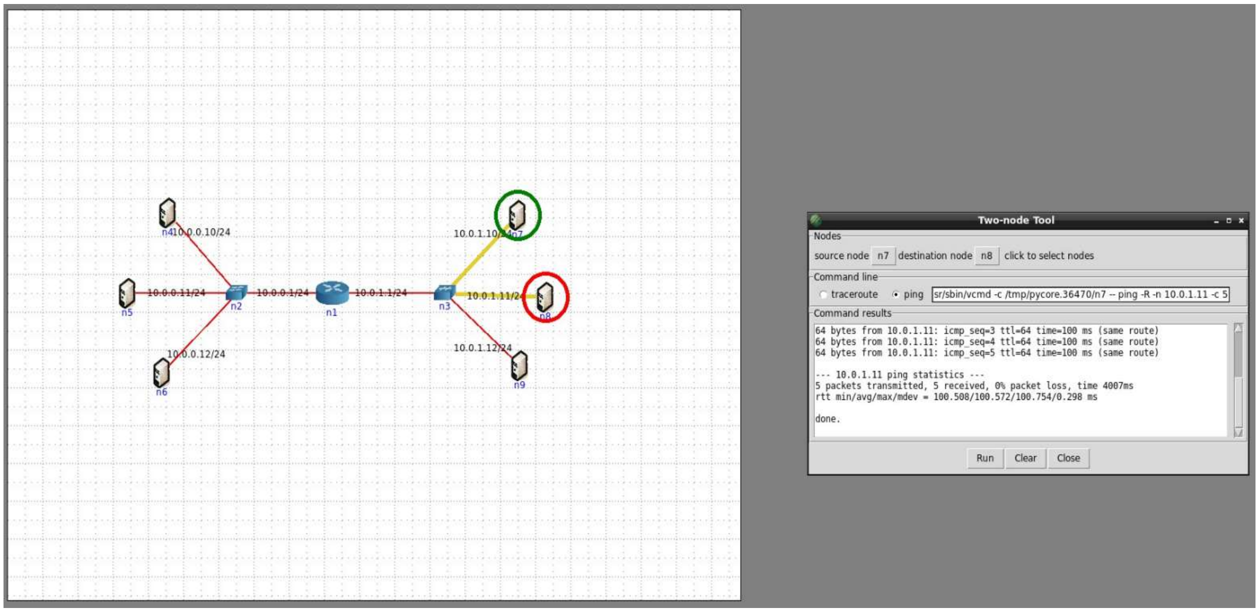
Throughput is measured by sender and receiver is as shown in

Sender- 1.11 M bits/sec

Receiver- 658 M bits/sec

b) Between two host connected to the hub

- *Measuring loss*



source node n7 destination node n8 click to select nodes

○ traceroute ● ping `sr/sbin/vcmd -c /tmp/pycore.36470/n7 -- ping -R -n 10.0.1.11 -c 5`

```
64 bytes from 10.0.1.11: icmp_seq=3 ttl=64 time=100 ms (same route)
64 bytes from 10.0.1.11: icmp_seq=4 ttl=64 time=100 ms (same route)
64 bytes from 10.0.1.11: icmp_seq=5 ttl=64 time=100 ms (same route)

--- 10.0.1.11 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4007ms
rtt min/avg/max/mdev = 100.508/100.572/100.754/0.298 ms

done.
```

As shown above, the packet loss is at 0% while pinging from source to destination i.e between the hosts connected to the hub, capped at 5 had 100% transmission rate.

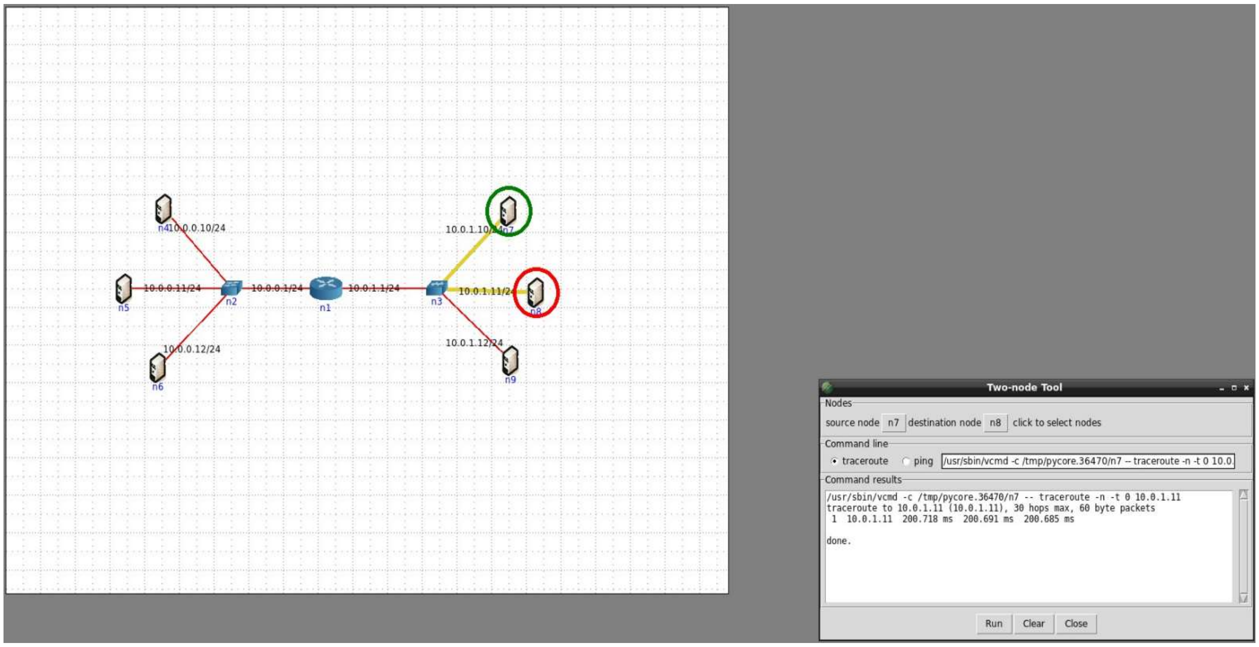
Loss percentage in the links between the hub and host was set to 1% each which means it has 1% chance of losing packets when pinged between two hosts connected to the hub.

- *Measuring delay*

Delay was measured with the help of traceroute in which the three delay measurements were approx 200ms each.

The round trip time was measured as such because of the delay for the links between hosts and switch was set to 50ms each.

In this observation, we can the delay is double the value expected.



source node n7 destination node n8 click to select nodes

● traceroute ○ ping /usr/sbin/vcmd -c /tmp/pycore.36470/n7 -- traceroute -n -t 0 10.0.

```
/usr/sbin/vcmd -c /tmp/pycore.36470/n7 -- traceroute -n -t 0 10.0.1.11
traceroute to 10.0.1.11 (10.0.1.11), 30 hops max, 60 byte packets
 1  10.0.1.11  200.718 ms  200.691 ms  200.685 ms

done.
```

- *Measuring throughput*

Bottlenecks affect network performance by slowing down the flow of information transmitted across networks.

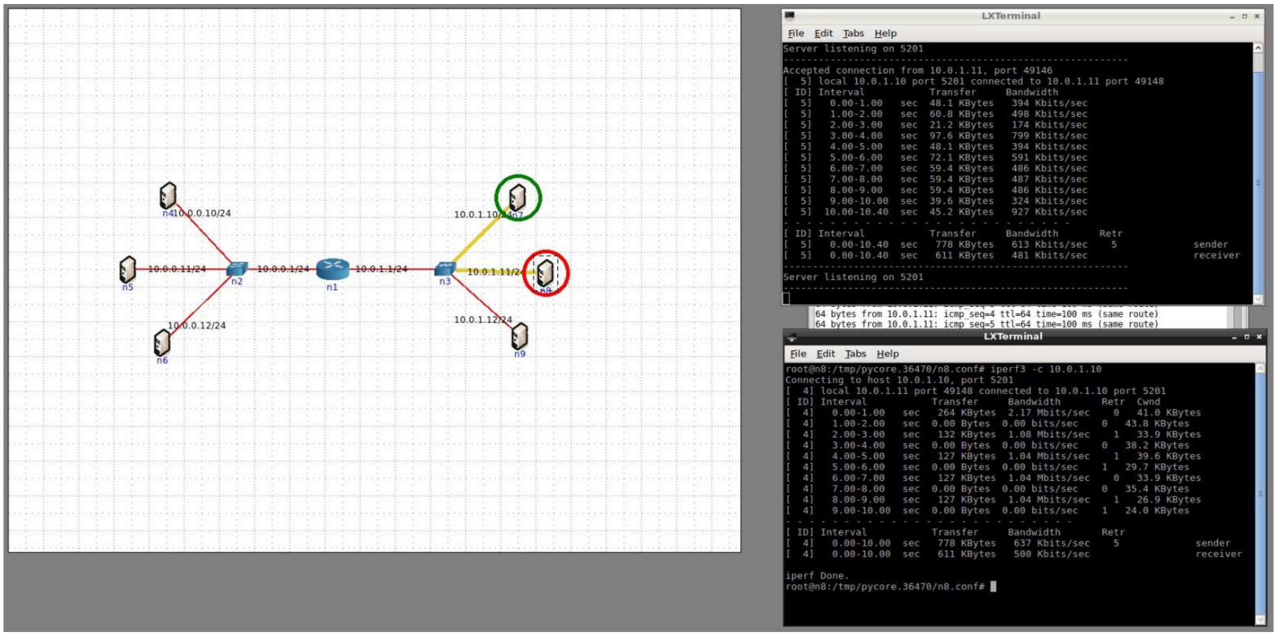
Since the bandwidth across the links between host and switch is evenly distributed (i.e 512kbps).

The network between the hosts is the bottleneck.

Throughput is measured by sender and receiver is as shown in

Sender- 637 k bits/sec

Receiver- 500 k bits/sec



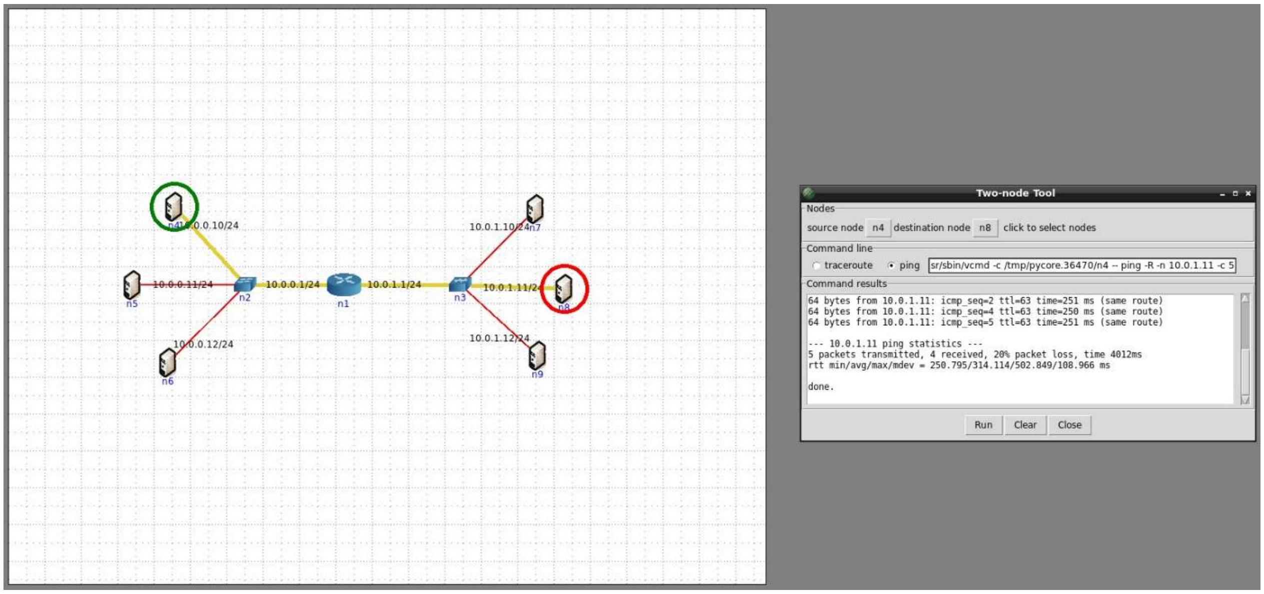
```
Server listening on 5201
-----------------------------------------------------------
Accepted connection from 10.0.1.11, port 49146
[  5] local 10.0.1.10 port 5201 connected to 10.0.1.11 port 49148
[ ID] Interval           Transfer     Bandwidth
[  5]   0.00-1.00   sec  48.1 KBytes   394 Kbits/sec
[  5]   1.00-2.00   sec  60.8 KBytes   498 Kbits/sec
[  5]   2.00-3.00   sec  21.2 KBytes   174 Kbits/sec
[  5]   3.00-4.00   sec  97.6 KBytes   799 Kbits/sec
[  5]   4.00-5.00   sec  48.1 KBytes   394 Kbits/sec
[  5]   5.00-6.00   sec  72.1 KBytes   591 Kbits/sec
[  5]   6.00-7.00   sec  59.4 KBytes   486 Kbits/sec
[  5]   7.00-8.00   sec  59.4 KBytes   487 Kbits/sec
[  5]   8.00-9.00   sec  59.4 KBytes   486 Kbits/sec
[  5]   9.00-10.00  sec  39.6 KBytes   324 Kbits/sec
[  5]  10.00-10.40  sec  45.2 KBytes   927 Kbits/sec
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Retr
[  5]   0.00-10.40  sec   778 KBytes   613 Kbits/sec    5             sender
[  5]   0.00-10.40  sec   611 KBytes   481 Kbits/sec                  receiver
-----------------------------------------------------------
Server listening on 5201
-----------------------------------------------------------
```

```
64 bytes from 10.0.1.11: icmp_seq=4 ttl=64 time=100 ms (same route)
64 bytes from 10.0.1.11: icmp_seq=5 ttl=64 time=100 ms (same route)
```

```
root@n8:/tmp/pycore.36470/n8.conf# iperf3 -c 10.0.1.10
Connecting to host 10.0.1.10, port 5201
[  4] local 10.0.1.11 port 49148 connected to 10.0.1.10 port 5201
[ ID] Interval           Transfer     Bandwidth       Retr  Cwnd
[  4]   0.00-1.00   sec   264 KBytes  2.17 Mbits/sec    0   41.0 KBytes
[  4]   1.00-2.00   sec  0.00 Bytes  0.00 bits/sec    0   43.8 KBytes
[  4]   2.00-3.00   sec   132 KBytes  1.08 Mbits/sec    1   33.9 KBytes
[  4]   3.00-4.00   sec  0.00 Bytes  0.00 bits/sec    0   38.2 KBytes
[  4]   4.00-5.00   sec   127 KBytes  1.04 Mbits/sec    1   39.6 KBytes
[  4]   5.00-6.00   sec  0.00 Bytes  0.00 bits/sec    1   29.7 KBytes
[  4]   6.00-7.00   sec   127 KBytes  1.04 Mbits/sec    0   33.9 KBytes
[  4]   7.00-8.00   sec  0.00 Bytes  0.00 bits/sec    0   35.4 KBytes
[  4]   8.00-9.00   sec   127 KBytes  1.04 Mbits/sec    1   26.9 KBytes
[  4]   9.00-10.00  sec  0.00 Bytes  0.00 bits/sec    1   24.0 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Retr
[  4]   0.00-10.00  sec   778 KBytes   637 Kbits/sec    5             sender
[  4]   0.00-10.00  sec   611 KBytes   500 Kbits/sec                  receiver

iperf Done.
root@n8:/tmp/pycore.36470/n8.conf# 
```

b) Two hosts across the network

- *Measuring loss*



source node n4 destination node n8 click to select nodes

○ traceroute ● ping sr/sbin/vcmd -c /tmp/pycore.36470/n4 -- ping -R -n 10.0.1.11 -c 5

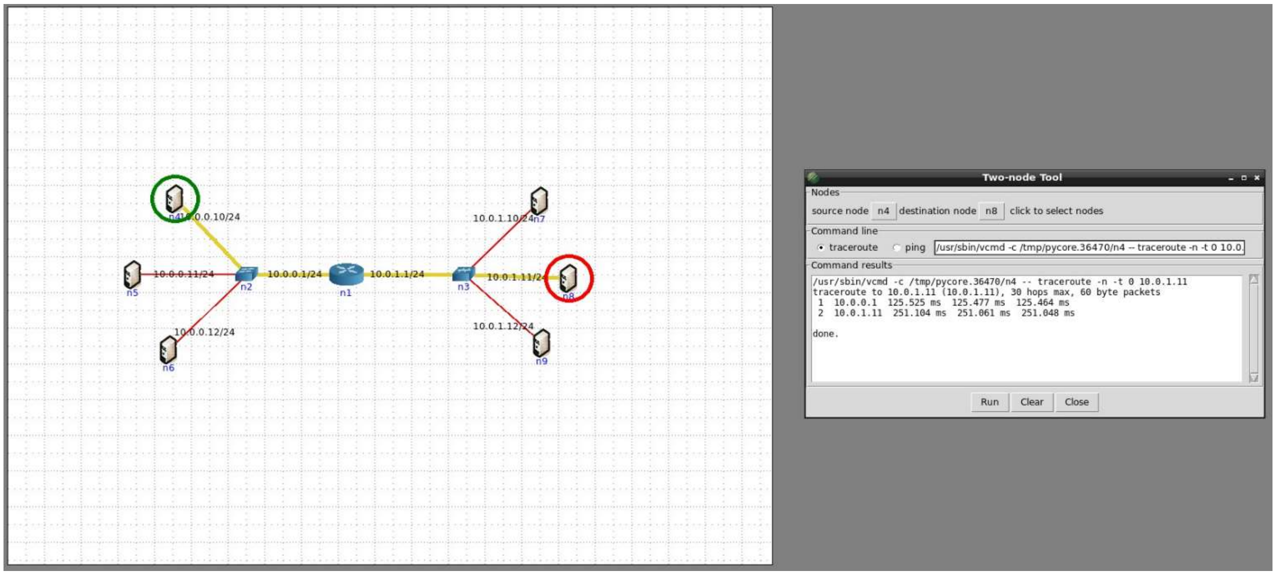
```
64 bytes from 10.0.1.11: icmp_seq=2 ttl=63 time=251 ms (same route)
64 bytes from 10.0.1.11: icmp_seq=4 ttl=63 time=250 ms (same route)
64 bytes from 10.0.1.11: icmp_seq=5 ttl=63 time=251 ms (same route)

--- 10.0.1.11 ping statistics ---
5 packets transmitted, 4 received, 20% packet loss, time 4012ms
rtt min/avg/max/mdev = 250.795/314.114/502.849/108.966 ms

done.
```

Here two hosts are connected across the network through a switch, router and hub.

As we can see when pinging there is 20% loss of packets across the network which results in 80% transmission rate.

This happens due to the links between router-switch and router-hub have a loss percentage set as 5% each and also there is a loss of 1% set between the hub and the host resulting in loss of packets.

- *Measuring Delay*



source node n4 destination node n8 click to select nodes

● traceroute ○ ping /usr/sbin/vcmd -c /tmp/pycore.36470/n4 -- traceroute -n -t 0 10.0.

```
/usr/sbin/vcmd -c /tmp/pycore.36470/n4 -- traceroute -n -t 0 10.0.1.11
traceroute to 10.0.1.11 (10.0.1.11), 30 hops max, 60 byte packets
 1  10.0.0.1  125.525 ms  125.477 ms  125.464 ms
 2  10.0.1.11  251.104 ms  251.061 ms  251.048 ms

done.
```

With the help of traceroute we got the delay from each hop and as shown above

The delay from the 1st hop that is router was around 125ms and the delay from 2nd hop that is the other host was approx 251ms.

Given the link between switch-router and switch-hub has 75ms set as a delay, and host-switch and host-hub have 50ms delay set, the results were justifiable.

- *Measuring Throughput*

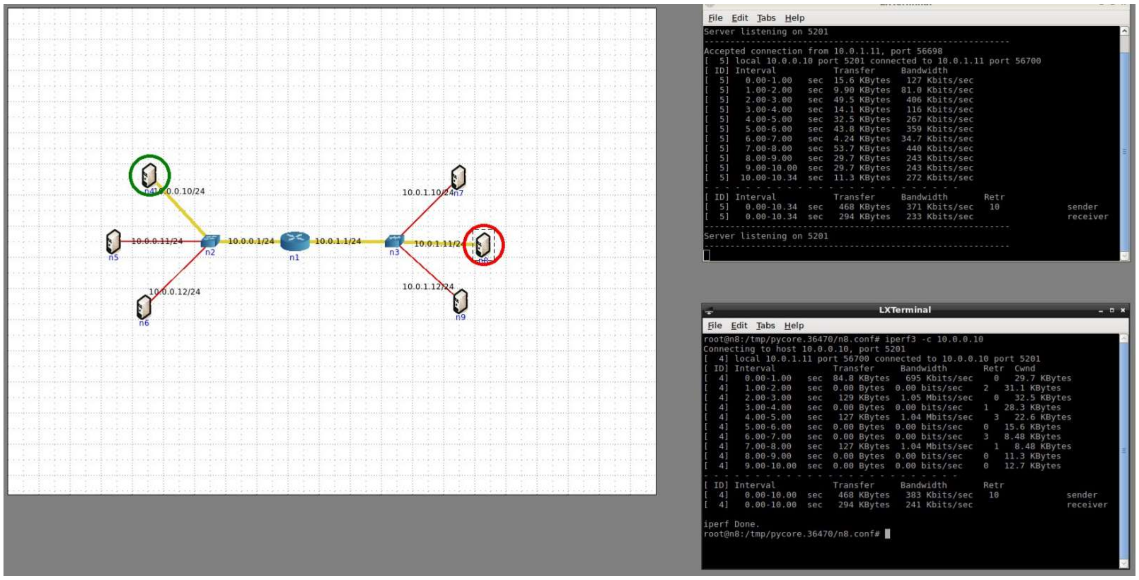
Bottlenecks affect network performance by slowing down the flow of information transmitted across networks.

Throughput is measured by sender and receiver is as shown in

Sender- 383 k bits/sec

Receiver- 241 k bits/sec

The network link between switch and hub is the bottleneck.



```
Server listening on 5201
-----------------------------------------------------------
Accepted connection from 10.0.1.11, port 56698
[  5] local 10.0.0.10 port 5201 connected to 10.0.1.11 port 56700
[ ID] Interval           Transfer     Bandwidth
[  5]   0.00-1.00   sec  15.6 KBytes   127 Kbits/sec
[  5]   1.00-2.00   sec  9.90 KBytes  81.0 Kbits/sec
[  5]   2.00-3.00   sec  49.5 KBytes   406 Kbits/sec
[  5]   3.00-4.00   sec  14.1 KBytes   116 Kbits/sec
[  5]   4.00-5.00   sec  32.5 KBytes   267 Kbits/sec
[  5]   5.00-6.00   sec  43.8 KBytes   359 Kbits/sec
[  5]   6.00-7.00   sec  4.24 KBytes  34.7 Kbits/sec
[  5]   7.00-8.00   sec  53.7 KBytes   440 Kbits/sec
[  5]   8.00-9.00   sec  29.7 KBytes   243 Kbits/sec
[  5]   9.00-10.00  sec  29.7 KBytes   243 Kbits/sec
[  5]  10.00-10.34  sec  11.3 KBytes   272 Kbits/sec
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Retr
[  5]   0.00-10.34  sec   468 KBytes   371 Kbits/sec   10             sender
[  5]   0.00-10.34  sec   294 KBytes   233 Kbits/sec                  receiver
-----------------------------------------------------------
Server listening on 5201
-----------------------------------------------------------
```

```
root@n8:/tmp/pycore.36470/n8.conf# iperf3 -c 10.0.0.10
Connecting to host 10.0.0.10, port 5201
[  4] local 10.0.1.11 port 56700 connected to 10.0.0.10 port 5201
[ ID] Interval           Transfer     Bandwidth       Retr  Cwnd
[  4]   0.00-1.00   sec  84.8 KBytes   695 Kbits/sec    0   29.7 KBytes
[  4]   1.00-2.00   sec  0.00 Bytes  0.00 bits/sec    2   31.1 KBytes
[  4]   2.00-3.00   sec   129 KBytes  1.05 Mbits/sec    0   32.5 KBytes
[  4]   3.00-4.00   sec  0.00 Bytes  0.00 bits/sec    1   28.3 KBytes
[  4]   4.00-5.00   sec   127 KBytes  1.04 Mbits/sec    3   22.6 KBytes
[  4]   5.00-6.00   sec  0.00 Bytes  0.00 bits/sec    0   15.6 KBytes
[  4]   6.00-7.00   sec  0.00 Bytes  0.00 bits/sec    3   8.48 KBytes
[  4]   7.00-8.00   sec   127 KBytes  1.04 Mbits/sec    1   8.48 KBytes
[  4]   8.00-9.00   sec  0.00 Bytes  0.00 bits/sec    0   11.3 KBytes
[  4]   9.00-10.00  sec  0.00 Bytes  0.00 bits/sec    0   12.7 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Retr
[  4]   0.00-10.00  sec   468 KBytes   383 Kbits/sec   10             sender
[  4]   0.00-10.00  sec   294 KBytes   241 Kbits/sec                  receiver

iperf Done.
root@n8:/tmp/pycore.36470/n8.conf#
```

Task 1.2: Link Layer Transport

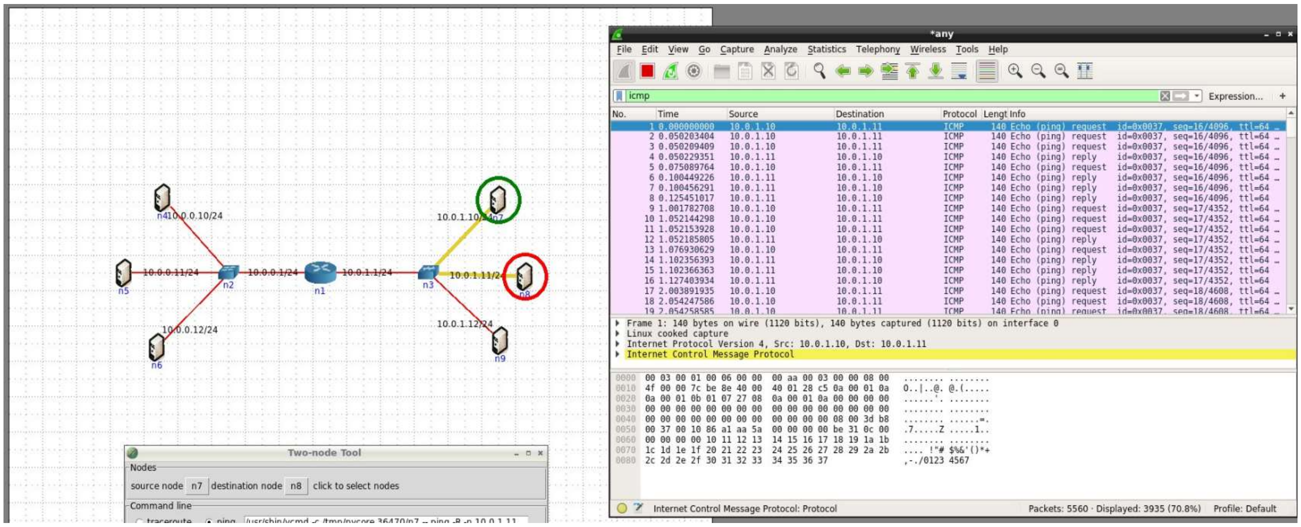
A Switch can connect to various network segments. It is a hardware device that joins multiple hosts within a network whereas a hub connects multiple hosts together, making them act as a single segment, in other words, when a hub receives a packet it copies and distributes the copies to all its host.

As shown below in the picture

- There are 3 hosts attached to the switch and the hub each.
- With the help of Wireshark, you can observe the difference between the hub and switch.
- When you ping we can observe the ICMP messages in Wireshark and differentiate between the hub and the switch.

Below is the evidence for the hub and switch with Wireshark.

Hub



source node n7 destination node n8 click to select nodes

```
/usr/sbin/vcmd -c /tmp/pycore.36470/n7 -- ping -R -n 10.0.1.11
```

icmp

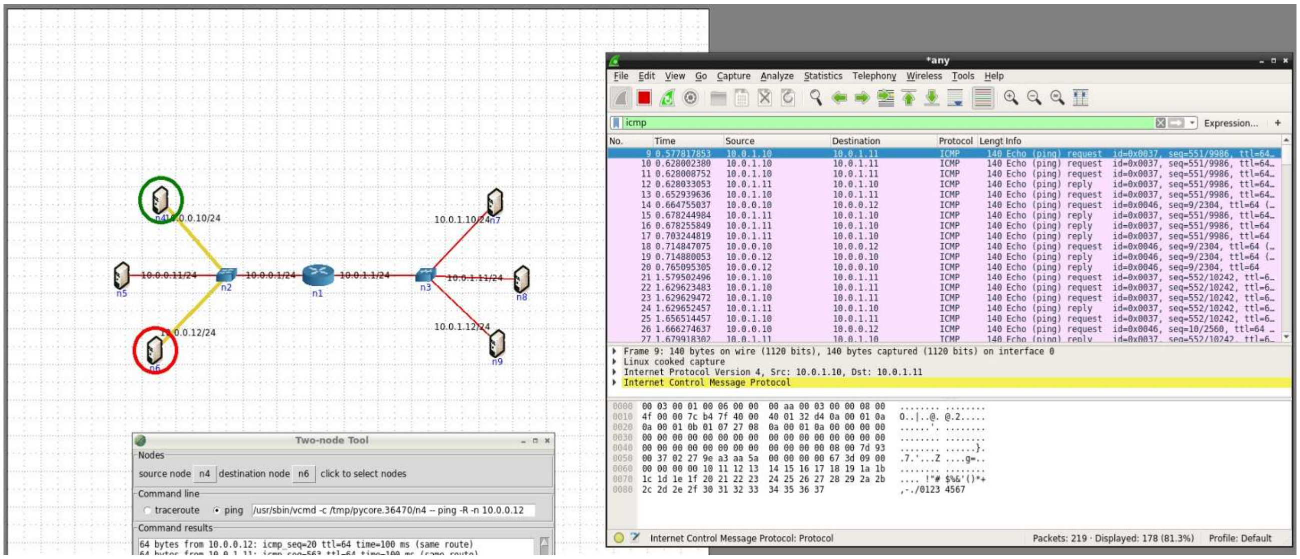
```
No.  Time         Source       Destination  Protocol Lengt Info
   1 0.000000000  10.0.1.10    10.0.1.11    ICMP     140 Echo (ping) request  id=0x0037, seq=16/4096, ttl=64 …
   2 0.050203404  10.0.1.10    10.0.1.11    ICMP     140 Echo (ping) request  id=0x0037, seq=16/4096, ttl=64 …
   3 0.050209409  10.0.1.10    10.0.1.11    ICMP     140 Echo (ping) request  id=0x0037, seq=16/4096, ttl=64 …
   4 0.050229351  10.0.1.11    10.0.1.10    ICMP     140 Echo (ping) reply    id=0x0037, seq=16/4096, ttl=64 …
   5 0.075089764  10.0.1.10    10.0.1.11    ICMP     140 Echo (ping) request  id=0x0037, seq=16/4096, ttl=64 …
   6 0.100449226  10.0.1.11    10.0.1.10    ICMP     140 Echo (ping) reply    id=0x0037, seq=16/4096, ttl=64 …
   7 0.100456291  10.0.1.11    10.0.1.10    ICMP     140 Echo (ping) reply    id=0x0037, seq=16/4096, ttl=64
   8 0.125451017  10.0.1.11    10.0.1.10    ICMP     140 Echo (ping) reply    id=0x0037, seq=16/4096, ttl=64
   9 1.001782708  10.0.1.10    10.0.1.11    ICMP     140 Echo (ping) request  id=0x0037, seq=17/4352, ttl=64 …
  10 1.052144298  10.0.1.10    10.0.1.11    ICMP     140 Echo (ping) request  id=0x0037, seq=17/4352, ttl=64 …
  11 1.052153928  10.0.1.10    10.0.1.11    ICMP     140 Echo (ping) request  id=0x0037, seq=17/4352, ttl=64 …
  12 1.052185805  10.0.1.11    10.0.1.10    ICMP     140 Echo (ping) reply    id=0x0037, seq=17/4352, ttl=64 …
  13 1.076930629  10.0.1.10    10.0.1.11    ICMP     140 Echo (ping) request  id=0x0037, seq=17/4352, ttl=64 …
  14 1.102356393  10.0.1.11    10.0.1.10    ICMP     140 Echo (ping) reply    id=0x0037, seq=17/4352, ttl=64 …
  15 1.102366363  10.0.1.11    10.0.1.10    ICMP     140 Echo (ping) reply    id=0x0037, seq=17/4352, ttl=64
  16 1.127403934  10.0.1.11    10.0.1.10    ICMP     140 Echo (ping) reply    id=0x0037, seq=17/4352, ttl=64
  17 2.003891935  10.0.1.10    10.0.1.11    ICMP     140 Echo (ping) request  id=0x0037, seq=18/4608, ttl=64 …
  18 2.054247586  10.0.1.10    10.0.1.11    ICMP     140 Echo (ping) request  id=0x0037, seq=18/4608, ttl=64 …
  19 2.054258585  10.0.1.10    10.0.1.11    ICMP     140 Echo (ping) request  id=0x0037, seq=18/4608, ttl=64 …
```

```
▸ Frame 1: 140 bytes on wire (1120 bits), 140 bytes captured (1120 bits) on interface 0
▸ Linux cooked capture
▸ Internet Protocol Version 4, Src: 10.0.1.10, Dst: 10.0.1.11
▸ Internet Control Message Protocol
```

```
0000  00 03 00 01 00 06 00 00  00 aa 00 03 00 00 08 00   ........ ........
0010  4f 00 00 7c be 8e 40 00  40 01 28 c5 0a 00 01 0a   O..|..@. @.(.....
0020  0a 00 01 0b 01 07 27 08  0a 00 01 0a 00 00 00 00   ......'. ........
0030  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0040  00 00 00 00 00 00 00 00  00 00 00 00 08 00 3d b8   ........ ......=.
0050  00 37 00 10 86 a1 aa 5a  00 00 00 00 be 31 0c 00   .7.....Z .....1..
0060  00 00 00 00 10 11 12 13  14 15 16 17 18 19 1a 1b   ........ ........
0070  1c 1d 1e 1f 20 21 22 23  24 25 26 27 28 29 2a 2b   .... !"# $%&'()*+
0080  2c 2d 2e 2f 30 31 32 33  34 35 36 37               ,-./0123 4567
```

Internet Control Message Protocol: Protocol   Packets: 5560 · Displayed: 3935 (70.8%)   Profile: Default

Switch



source node n4 destination node n6 click to select nodes

```
/usr/sbin/vcmd -c /tmp/pycore.36470/n4 -- ping -R -n 10.0.0.12
```

```
64 bytes from 10.0.0.12: icmp_seq=20 ttl=64 time=100 ms (same route)
64 bytes from 10.0.1.11: icmp_seq=563 ttl=64 time=100 ms (same route)
```

icmp

```
No.  Time         Source       Destination  Protocol Lengt Info
   9 0.577817853  10.0.1.10    10.0.1.11    ICMP     140 Echo (ping) request  id=0x0037, seq=551/9986, ttl=64…
  10 0.628002380  10.0.1.10    10.0.1.11    ICMP     140 Echo (ping) request  id=0x0037, seq=551/9986, ttl=64…
  11 0.628008752  10.0.1.10    10.0.1.11    ICMP     140 Echo (ping) request  id=0x0037, seq=551/9986, ttl=64…
  12 0.628033053  10.0.1.11    10.0.1.10    ICMP     140 Echo (ping) reply    id=0x0037, seq=551/9986, ttl=64…
  13 0.652939636  10.0.1.10    10.0.1.11    ICMP     140 Echo (ping) request  id=0x0037, seq=551/9986, ttl=64…
  14 0.664755037  10.0.0.10    10.0.0.12    ICMP     140 Echo (ping) request  id=0x0046, seq=9/2304, ttl=64 (…
  15 0.678244984  10.0.1.11    10.0.1.10    ICMP     140 Echo (ping) reply    id=0x0037, seq=551/9986, ttl=64…
  16 0.678255849  10.0.1.11    10.0.1.10    ICMP     140 Echo (ping) reply    id=0x0037, seq=551/9986, ttl=64
  17 0.703244819  10.0.1.11    10.0.1.10    ICMP     140 Echo (ping) reply    id=0x0037, seq=551/9986, ttl=64
  18 0.714847075  10.0.0.10    10.0.0.12    ICMP     140 Echo (ping) request  id=0x0046, seq=9/2304, ttl=64 (…
  19 0.714880053  10.0.0.12    10.0.0.10    ICMP     140 Echo (ping) reply    id=0x0046, seq=9/2304, ttl=64 (…
  20 0.765095305  10.0.0.12    10.0.0.10    ICMP     140 Echo (ping) reply    id=0x0046, seq=9/2304, ttl=64
  21 1.579502496  10.0.1.10    10.0.1.11    ICMP     140 Echo (ping) request  id=0x0037, seq=552/10242, ttl=6…
  22 1.629623483  10.0.1.10    10.0.1.11    ICMP     140 Echo (ping) request  id=0x0037, seq=552/10242, ttl=6…
  23 1.629629472  10.0.1.10    10.0.1.11    ICMP     140 Echo (ping) request  id=0x0037, seq=552/10242, ttl=6…
  24 1.629652457  10.0.1.11    10.0.1.10    ICMP     140 Echo (ping) reply    id=0x0037, seq=552/10242, ttl=6…
  25 1.656514457  10.0.1.10    10.0.1.11    ICMP     140 Echo (ping) request  id=0x0037, seq=552/10242, ttl=6…
  26 1.666274637  10.0.0.10    10.0.0.12    ICMP     140 Echo (ping) request  id=0x0046, seq=10/2560, ttl=64 …
  27 1.679918302  10.0.1.11    10.0.1.10    ICMP     140 Echo (ping) reply    id=0x0037, seq=552/10242, ttl=6…
```

```
▸ Frame 9: 140 bytes on wire (1120 bits), 140 bytes captured (1120 bits) on interface 0
▸ Linux cooked capture
▸ Internet Protocol Version 4, Src: 10.0.1.10, Dst: 10.0.1.11
▸ Internet Control Message Protocol
```

```
0000  00 03 00 01 00 06 00 00  00 aa 00 03 00 00 08 00   ........ ........
0010  4f 00 00 7c b4 7f 40 00  40 01 32 d4 0a 00 01 0a   O..|..@. @.2.....
0020  0a 00 01 0b 01 07 27 08  0a 00 01 0a 00 00 00 00   ......'. ........
0030  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0040  00 00 00 00 00 00 00 00  00 00 00 00 08 00 7d 93   ........ ......}.
0050  00 37 02 27 9e a3 aa 5a  00 00 00 00 67 3d 09 00   .7.'...Z ....g=..
0060  00 00 00 00 10 11 12 13  14 15 16 17 18 19 1a 1b   ........ ........
0070  1c 1d 1e 1f 20 21 22 23  24 25 26 27 28 29 2a 2b   .... !"# $%&'()*+
0080  2c 2d 2e 2f 30 31 32 33  34 35 36 37               ,-./0123 4567
```

Internet Control Message Protocol: Protocol   Packets: 219 · Displayed: 178 (81.3%)   Profile: Default

Task 2.1: Building a Bigger Network

In this task, we are going to expand our network from our already existing network that we have seen in task 1.1.

Here we will be including

- 4 additional routers (making 5 in total) in a full mesh configuration; that is, each router is connected to every other router
- A small network attached to each of these new routers. These should each contain three hosts attached to one switch. Each switch is then attached to one router.

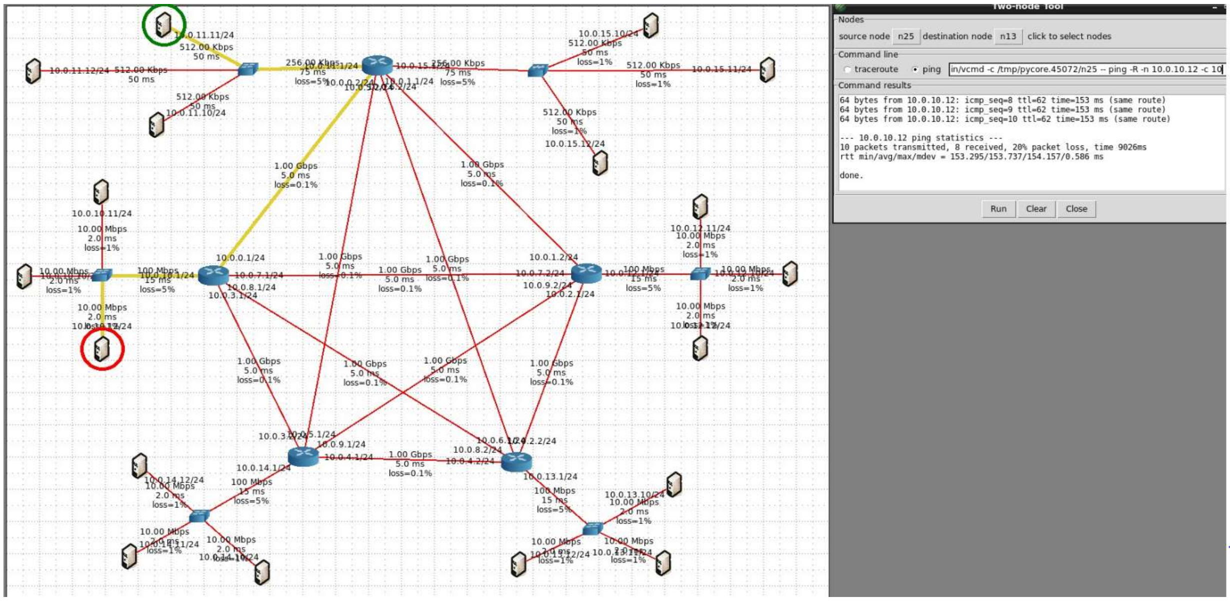
Once the topology is set, we need to set the value for bandwidth, delay, loss

- For links between routers, bandwidth should be set to 1gbps, delay to 5ms and loss to 0.1%
- For links between the switches and the routers, bandwidth should be set to 100mbps, delay to 15ms and loss to 5 %.
- For links between the hosts and switches, bandwidth should be set to 10mbps, delay to 2ms and loss to 1%

Now once the bigger network is built we need to run multiple measurements like we did in task1.1

A few Measurements are shown below

- *Measuring Loss*



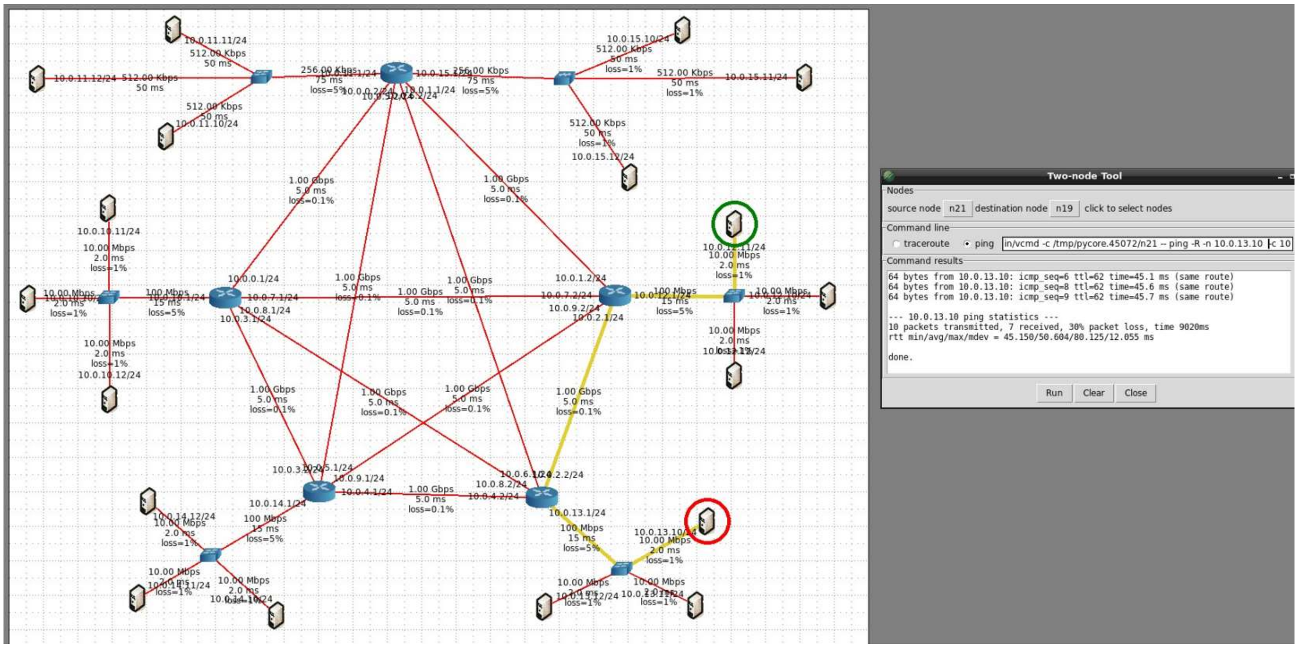
source node n25 destination node n13 click to select nodes

○ traceroute ● ping `in/vcmd -c /tmp/pycore.45072/n25 -- ping -R -n 10.0.10.12 -c 10`

```
64 bytes from 10.0.10.12: icmp_seq=8 ttl=62 time=153 ms (same route)
64 bytes from 10.0.10.12: icmp_seq=9 ttl=62 time=153 ms (same route)
64 bytes from 10.0.10.12: icmp_seq=10 ttl=62 time=153 ms (same route)

--- 10.0.10.12 ping statistics ---
10 packets transmitted, 8 received, 20% packet loss, time 9026ms
rtt min/avg/max/mdev = 153.295/153.737/154.157/0.586 ms

done.
```

Here as we can observe, the packet loss percentage is 20% which means 80% of packets are transmitted.

Loss of packets occurs due to the link loss percentage values set between new switch- new router at 5% and between the routers as 0.1%.



Two-node Tool

Nodes

source node n21 destination node n19 click to select nodes

Command line

○ traceroute ● ping in/vcmd -c /tmp/pycore.45072/n21 -- ping -R -n 10.0.13.10 -c 10

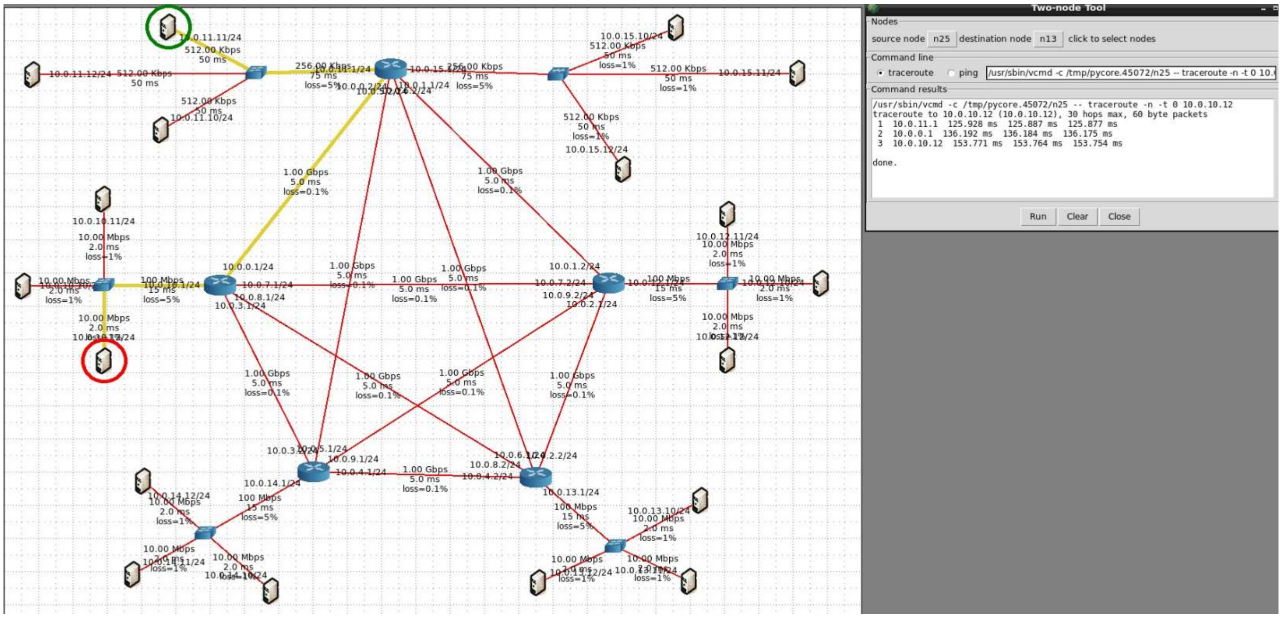
Command results

```
64 bytes from 10.0.13.10: icmp_seq=6 ttl=62 time=45.1 ms (same route)
64 bytes from 10.0.13.10: icmp_seq=8 ttl=62 time=45.6 ms (same route)
64 bytes from 10.0.13.10: icmp_seq=9 ttl=62 time=45.7 ms (same route)

--- 10.0.13.10 ping statistics ---
10 packets transmitted, 7 received, 30% packet loss, time 9020ms
rtt min/avg/max/mdev = 45.150/50.604/80.125/12.055 ms

done.
```

Here as we can observe, the packet loss percentage is 30% which means 70% of packets are transmitted.

Loss of packets occurs due to the link loss percentage values set between new switch- new router at 5% and between the routers as 0.1%.

- *Measuring Delay*



Two-node Tool

Nodes

source node n25 destination node n13 click to select nodes

Command line

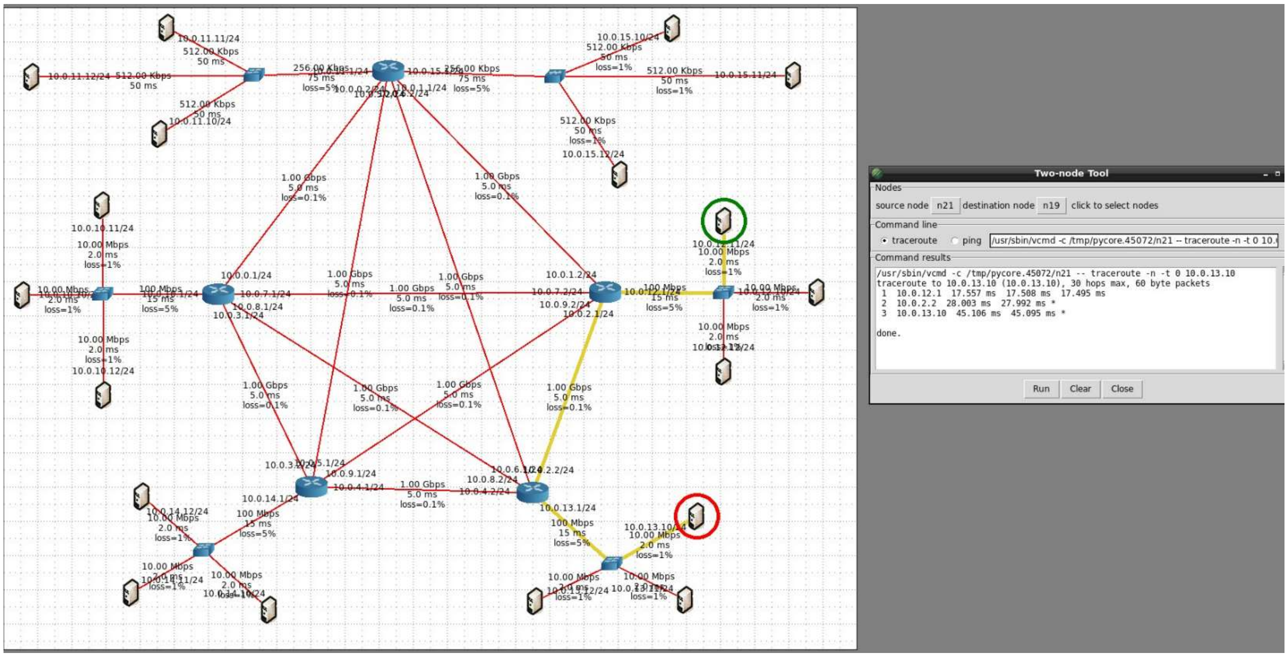
● traceroute ○ ping /usr/sbin/vcmd -c /tmp/pycore.45072/n25 -- traceroute -n -t 0 10.0

Command results

```
/usr/sbin/vcmd -c /tmp/pycore.45072/n25 -- traceroute -n -t 0 10.0.10.12
traceroute to 10.0.10.12 (10.0.10.12), 30 hops max, 60 byte packets
 1  10.0.11.1  125.928 ms  125.887 ms  125.877 ms
 2  10.0.0.1  136.192 ms  136.184 ms  136.175 ms
 3  10.0.10.12  153.771 ms  153.764 ms  153.754 ms

done.
```

As we can observe the delay is set to 50ms, 75ms, 5ms, 15ms, 2ms for the link between host-old_switch, old_switch-router,router-router,router-new_switch,new_switch-host respectively.

With each hop, we can observe the correct delay output from traceroute.



```
/usr/sbin/vcmd -c /tmp/pycore.45072/n21 -- traceroute -n -t 0 10.0
```

```
/usr/sbin/vcmd -c /tmp/pycore.45072/n21 -- traceroute -n -t 0 10.0.13.10
traceroute to 10.0.13.10 (10.0.13.10), 30 hops max, 60 byte packets
 1  10.0.12.1  17.557 ms  17.508 ms  17.495 ms
 2  10.0.2.2  28.003 ms  27.992 ms *
 3  10.0.13.10  45.106 ms  45.095 ms *

done.
```

As we can observe the delay is set to 2ms, 15ms, 5ms, 15ms, 2ms for the link between host-switch, switch-router,router-router,router-switch,switch-host respectively.

With each hop, we can observe the correct delay output from traceroute. And we can also observe lost packets.

- *Measuring Throughput*

we can observe in the shown two throughput cases below.

In the first case, the bandwidth rate is seen as

Sender- 581 k bits/sec

Receiver- 445 k bits/sec

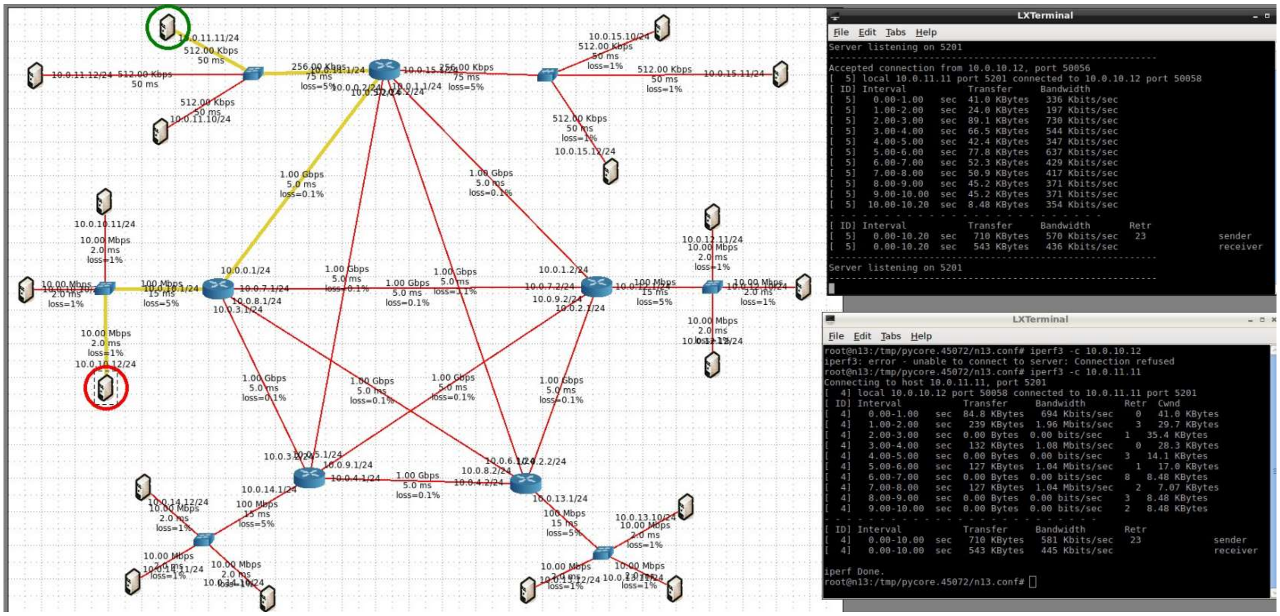
The sender_host-switch part of the network act as a bottleneck in this case.

In the Second case, the bandwidth rate is seen as

Sender- 1.66 M bits/sec

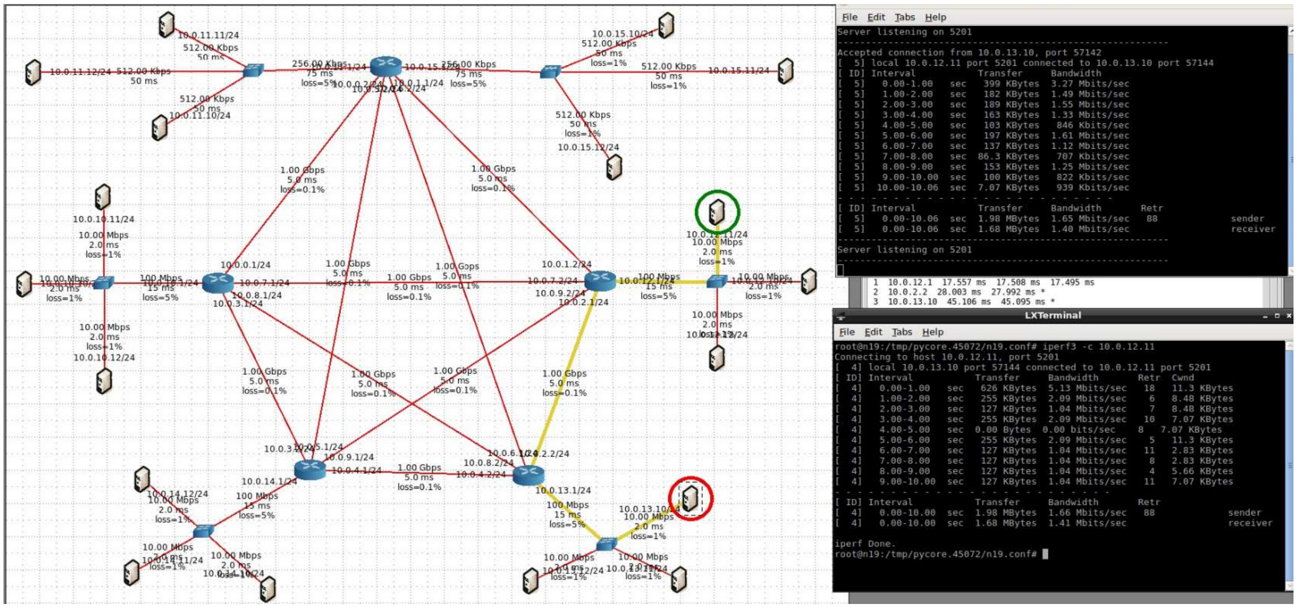
Receiver- 1.41 M bits/sec

The router-router part of the network act as a bottleneck in this case.



```
Server listening on 5201
-----------------------------------------------------------
Accepted connection from 10.0.10.12, port 50056
[  5] local 10.0.11.11 port 5201 connected to 10.0.10.12 port 50058
[ ID] Interval           Transfer     Bandwidth
[  5]   0.00-1.00   sec  41.0 KBytes   336 Kbits/sec
[  5]   1.00-2.00   sec  24.0 KBytes   197 Kbits/sec
[  5]   2.00-3.00   sec  89.1 KBytes   730 Kbits/sec
[  5]   3.00-4.00   sec  66.5 KBytes   544 Kbits/sec
[  5]   4.00-5.00   sec  42.4 KBytes   347 Kbits/sec
[  5]   5.00-6.00   sec  77.8 KBytes   637 Kbits/sec
[  5]   6.00-7.00   sec  52.3 KBytes   429 Kbits/sec
[  5]   7.00-8.00   sec  50.9 KBytes   417 Kbits/sec
[  5]   8.00-9.00   sec  45.2 KBytes   371 Kbits/sec
[  5]   9.00-10.00  sec  45.2 KBytes   371 Kbits/sec
[  5]  10.00-10.20  sec  8.48 KBytes   354 Kbits/sec
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Retr
[  5]   0.00-10.20  sec   710 KBytes   570 Kbits/sec   23             sender
[  5]   0.00-10.20  sec   543 KBytes   436 Kbits/sec                  receiver
-----------------------------------------------------------
Server listening on 5201
-----------------------------------------------------------
```

```
root@n13:/tmp/pycore.45072/n13.conf# iperf3 -c 10.0.10.12
iperf3: error - unable to connect to server: Connection refused
root@n13:/tmp/pycore.45072/n13.conf# iperf3 -c 10.0.11.11
Connecting to host 10.0.11.11, port 5201
[  4] local 10.0.10.12 port 50058 connected to 10.0.11.11 port 5201
[ ID] Interval           Transfer     Bandwidth       Retr  Cwnd
[  4]   0.00-1.00   sec  84.8 KBytes   694 Kbits/sec    0   41.0 KBytes
[  4]   1.00-2.00   sec   239 KBytes  1.96 Mbits/sec    3   29.7 KBytes
[  4]   2.00-3.00   sec  0.00 Bytes  0.00 bits/sec    1   35.4 KBytes
[  4]   3.00-4.00   sec   132 KBytes  1.08 Mbits/sec    0   28.3 KBytes
[  4]   4.00-5.00   sec  0.00 Bytes  0.00 bits/sec    3   14.1 KBytes
[  4]   5.00-6.00   sec   127 KBytes  1.04 Mbits/sec    1   17.0 KBytes
[  4]   6.00-7.00   sec  0.00 Bytes  0.00 bits/sec    8   8.48 KBytes
[  4]   7.00-8.00   sec   127 KBytes  1.04 Mbits/sec    2   7.07 KBytes
[  4]   8.00-9.00   sec  0.00 Bytes  0.00 bits/sec    3   8.48 KBytes
[  4]   9.00-10.00  sec  0.00 Bytes  0.00 bits/sec    2   8.48 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Retr
[  4]   0.00-10.00  sec   710 KBytes   581 Kbits/sec   23             sender
[  4]   0.00-10.00  sec   543 KBytes   445 Kbits/sec                  receiver

iperf Done.
root@n13:/tmp/pycore.45072/n13.conf#
```



```
Server listening on 5201
-----------------------------------------------------------
Accepted connection from 10.0.13.10, port 57142
[  5] local 10.0.12.11 port 5201 connected to 10.0.13.10 port 57144
[ ID] Interval           Transfer     Bandwidth
[  5]   0.00-1.00   sec   399 KBytes  3.27 Mbits/sec
[  5]   1.00-2.00   sec   182 KBytes  1.49 Mbits/sec
[  5]   2.00-3.00   sec   189 KBytes  1.55 Mbits/sec
[  5]   3.00-4.00   sec   163 KBytes  1.33 Mbits/sec
[  5]   4.00-5.00   sec   103 KBytes   846 Kbits/sec
[  5]   5.00-6.00   sec   197 KBytes  1.61 Mbits/sec
[  5]   6.00-7.00   sec   137 KBytes  1.12 Mbits/sec
[  5]   7.00-8.00   sec  86.3 KBytes   707 Kbits/sec
[  5]   8.00-9.00   sec   153 KBytes  1.25 Mbits/sec
[  5]   9.00-10.00  sec   100 KBytes   822 Kbits/sec
[  5]  10.00-10.06  sec  7.07 KBytes   939 Kbits/sec
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Retr
[  5]   0.00-10.06  sec  1.98 MBytes  1.65 Mbits/sec   88             sender
[  5]   0.00-10.06  sec  1.68 MBytes  1.40 Mbits/sec                  receiver
-----------------------------------------------------------
Server listening on 5201
-----------------------------------------------------------
```

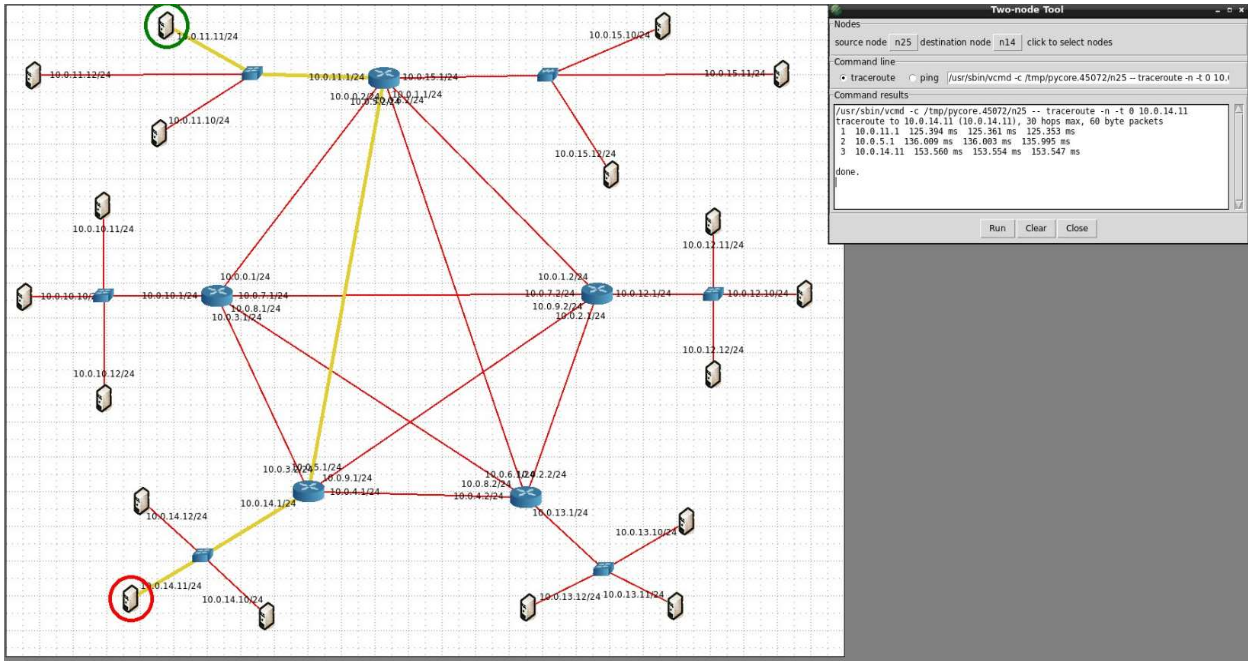
```
 1  10.0.12.1  17.557 ms  17.508 ms  17.495 ms
 2  10.0.2.2  28.003 ms  27.992 ms *
 3  10.0.13.10  45.106 ms  45.095 ms *
```

```
root@n19:/tmp/pycore.45072/n19.conf# iperf3 -c 10.0.12.11
Connecting to host 10.0.12.11, port 5201
[  4] local 10.0.13.10 port 57144 connected to 10.0.12.11 port 5201
[ ID] Interval           Transfer     Bandwidth       Retr  Cwnd
[  4]   0.00-1.00   sec   626 KBytes  5.13 Mbits/sec   18   11.3 KBytes
[  4]   1.00-2.00   sec   255 KBytes  2.09 Mbits/sec    6   8.48 KBytes
[  4]   2.00-3.00   sec   127 KBytes  1.04 Mbits/sec    7   8.48 KBytes
[  4]   3.00-4.00   sec   255 KBytes  2.09 Mbits/sec   10   7.07 KBytes
[  4]   4.00-5.00   sec  0.00 Bytes  0.00 bits/sec    8   7.07 KBytes
[  4]   5.00-6.00   sec   255 KBytes  2.09 Mbits/sec    5   11.3 KBytes
[  4]   6.00-7.00   sec   127 KBytes  1.04 Mbits/sec   11   2.83 KBytes
[  4]   7.00-8.00   sec   127 KBytes  1.04 Mbits/sec    8   2.83 KBytes
[  4]   8.00-9.00   sec   127 KBytes  1.04 Mbits/sec    4   5.66 KBytes
[  4]   9.00-10.00  sec   127 KBytes  1.04 Mbits/sec   11   7.07 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Retr
[  4]   0.00-10.00  sec  1.98 MBytes  1.66 Mbits/sec   88             sender
[  4]   0.00-10.00  sec  1.68 MBytes  1.41 Mbits/sec                  receiver

iperf Done.
root@n19:/tmp/pycore.45072/n19.conf#
```

Task2.2: Configuring Routing

This task involves manipulating the routing configuration of the routers contained within the topology.

- Before manipulating the route configuration

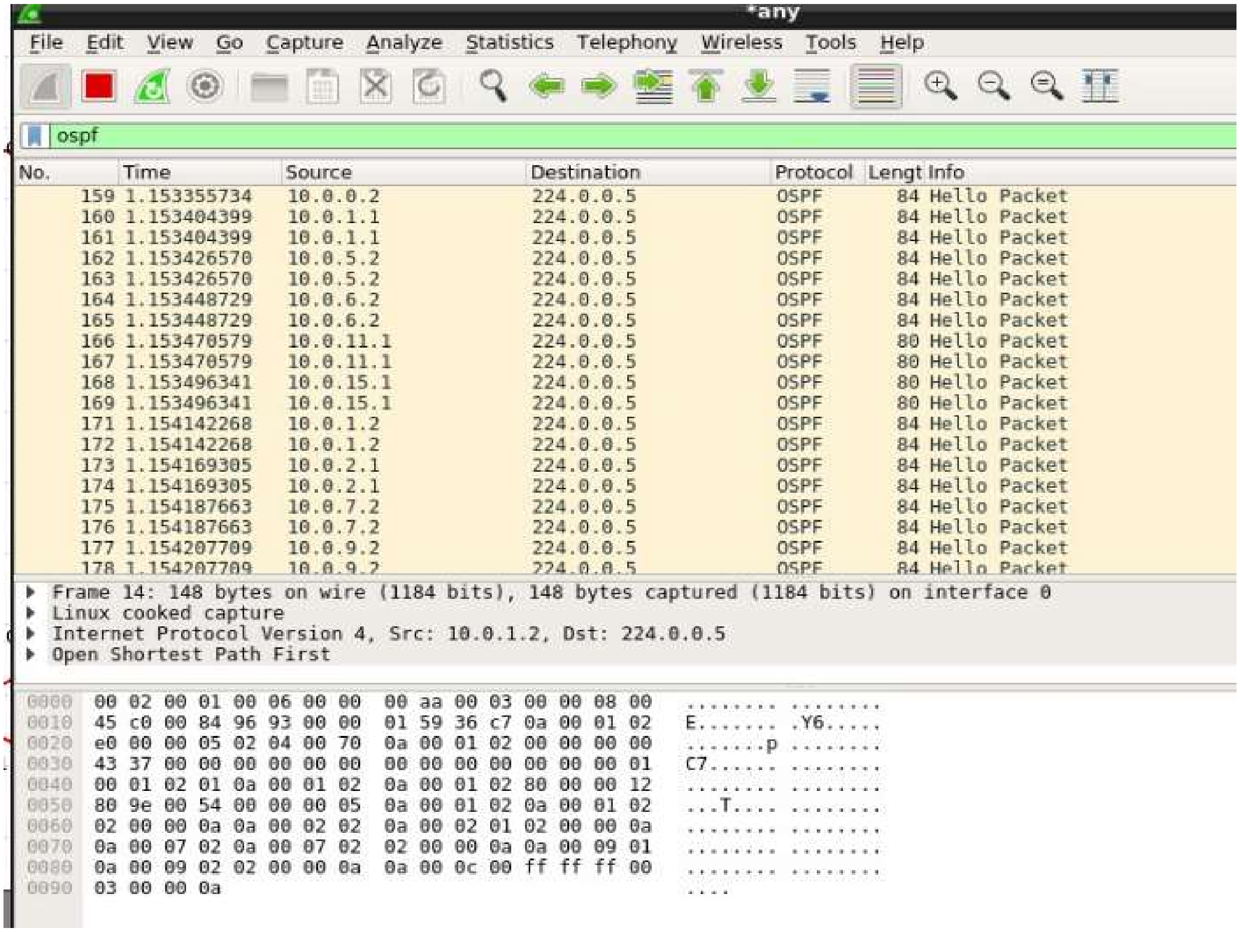


source node n25 destination node n14 click to select nodes

● traceroute ○ ping /usr/sbin/vcmd -c /tmp/pycore.45072/n25 -- traceroute -n -t 0 10.0

```
/usr/sbin/vcmd -c /tmp/pycore.45072/n25 -- traceroute -n -t 0 10.0.14.11
traceroute to 10.0.14.11 (10.0.14.11), 30 hops max, 60 byte packets
 1  10.0.11.1  125.394 ms  125.361 ms  125.353 ms
 2  10.0.5.1  136.009 ms  136.003 ms  135.995 ms
 3  10.0.14.11  153.560 ms  153.554 ms  153.547 ms

done.
```

Before manipulating the route between the hosts, the path taken is the shortest path cost between the host.



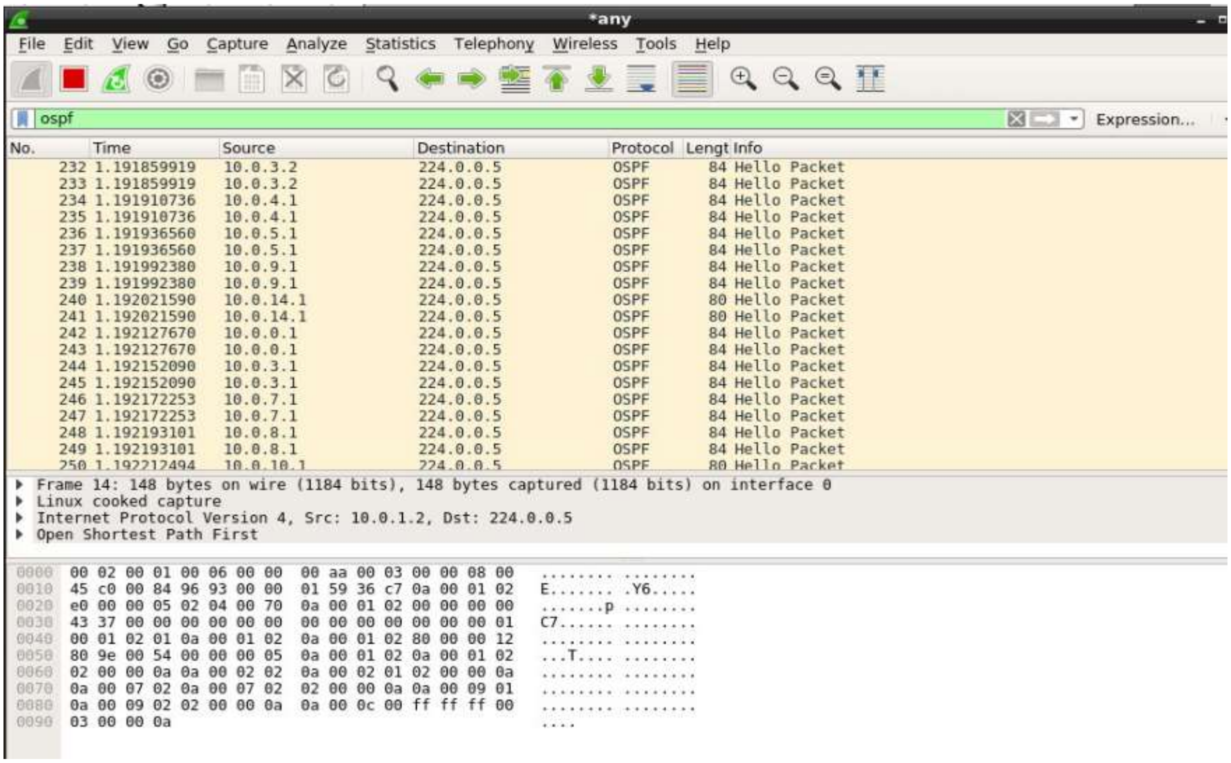
ospf

```
No.       Time          Source         Destination     Protocol Lengt Info
    159 1.153355734  10.0.0.2       224.0.0.5       OSPF        84 Hello Packet
    160 1.153404399  10.0.1.1       224.0.0.5       OSPF        84 Hello Packet
    161 1.153404399  10.0.1.1       224.0.0.5       OSPF        84 Hello Packet
    162 1.153426570  10.0.5.2       224.0.0.5       OSPF        84 Hello Packet
    163 1.153426570  10.0.5.2       224.0.0.5       OSPF        84 Hello Packet
    164 1.153448729  10.0.6.2       224.0.0.5       OSPF        84 Hello Packet
    165 1.153448729  10.0.6.2       224.0.0.5       OSPF        84 Hello Packet
    166 1.153470579  10.0.11.1      224.0.0.5       OSPF        80 Hello Packet
    167 1.153470579  10.0.11.1      224.0.0.5       OSPF        80 Hello Packet
    168 1.153496341  10.0.15.1      224.0.0.5       OSPF        80 Hello Packet
    169 1.153496341  10.0.15.1      224.0.0.5       OSPF        80 Hello Packet
    171 1.154142268  10.0.1.2       224.0.0.5       OSPF        84 Hello Packet
    172 1.154142268  10.0.1.2       224.0.0.5       OSPF        84 Hello Packet
    173 1.154169305  10.0.2.1       224.0.0.5       OSPF        84 Hello Packet
    174 1.154169305  10.0.2.1       224.0.0.5       OSPF        84 Hello Packet
    175 1.154187663  10.0.7.2       224.0.0.5       OSPF        84 Hello Packet
    176 1.154187663  10.0.7.2       224.0.0.5       OSPF        84 Hello Packet
    177 1.154207709  10.0.9.2       224.0.0.5       OSPF        84 Hello Packet
    178 1.154207709  10.0.9.2       224.0.0.5       OSPF        84 Hello Packet
```

```
▸ Frame 14: 148 bytes on wire (1184 bits), 148 bytes captured (1184 bits) on interface 0
▸ Linux cooked capture
▸ Internet Protocol Version 4, Src: 10.0.1.2, Dst: 224.0.0.5
▸ Open Shortest Path First
```

```
0000  00 02 00 01 00 06 00 00  00 aa 00 03 00 00 08 00   ........ ........
0010  45 c0 00 84 96 93 00 00  01 59 36 c7 0a 00 01 02   E....... .Y6.....
0020  e0 00 00 05 02 04 00 70  0a 00 01 02 00 00 00 00   .......p ........
0030  43 37 00 00 00 00 00 00  00 00 00 00 00 00 00 01   C7...... ........
0040  00 01 02 01 0a 00 01 02  0a 00 01 02 80 00 00 12   ........ ........
0050  80 9e 00 54 00 00 00 05  0a 00 01 02 0a 00 01 02   ...T.... ........
0060  02 00 00 0a 0a 00 02 02  0a 00 02 01 02 00 00 0a   ........ ........
0070  0a 00 07 02 0a 00 07 02  02 00 00 0a 0a 00 09 01   ........ ........
0080  0a 00 09 02 02 00 00 0a  0a 00 0c 00 ff ff ff 00   ........ ........
0090  03 00 00 0a                                        ....
```

And with the help of Wireshark, we can observe through which routers OSPF messages pass



*any
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Expression...

ospf

```
No.     Time          Source       Destination   Protocol Lengt Info
  232 1.191859919    10.0.3.2     224.0.0.5     OSPF     84 Hello Packet
  233 1.191859919    10.0.3.2     224.0.0.5     OSPF     84 Hello Packet
  234 1.191910736    10.0.4.1     224.0.0.5     OSPF     84 Hello Packet
  235 1.191910736    10.0.4.1     224.0.0.5     OSPF     84 Hello Packet
  236 1.191936560    10.0.5.1     224.0.0.5     OSPF     84 Hello Packet
  237 1.191936560    10.0.5.1     224.0.0.5     OSPF     84 Hello Packet
  238 1.191992380    10.0.9.1     224.0.0.5     OSPF     84 Hello Packet
  239 1.191992380    10.0.9.1     224.0.0.5     OSPF     84 Hello Packet
  240 1.192021590    10.0.14.1    224.0.0.5     OSPF     80 Hello Packet
  241 1.192021590    10.0.14.1    224.0.0.5     OSPF     80 Hello Packet
  242 1.192127670    10.0.0.1     224.0.0.5     OSPF     84 Hello Packet
  243 1.192127670    10.0.0.1     224.0.0.5     OSPF     84 Hello Packet
  244 1.192152090    10.0.3.1     224.0.0.5     OSPF     84 Hello Packet
  245 1.192152090    10.0.3.1     224.0.0.5     OSPF     84 Hello Packet
  246 1.192172253    10.0.7.1     224.0.0.5     OSPF     84 Hello Packet
  247 1.192172253    10.0.7.1     224.0.0.5     OSPF     84 Hello Packet
  248 1.192193101    10.0.8.1     224.0.0.5     OSPF     84 Hello Packet
  249 1.192193101    10.0.8.1     224.0.0.5     OSPF     84 Hello Packet
  250 1.192212494    10.0.10.1    224.0.0.5     OSPF     80 Hello Packet
```

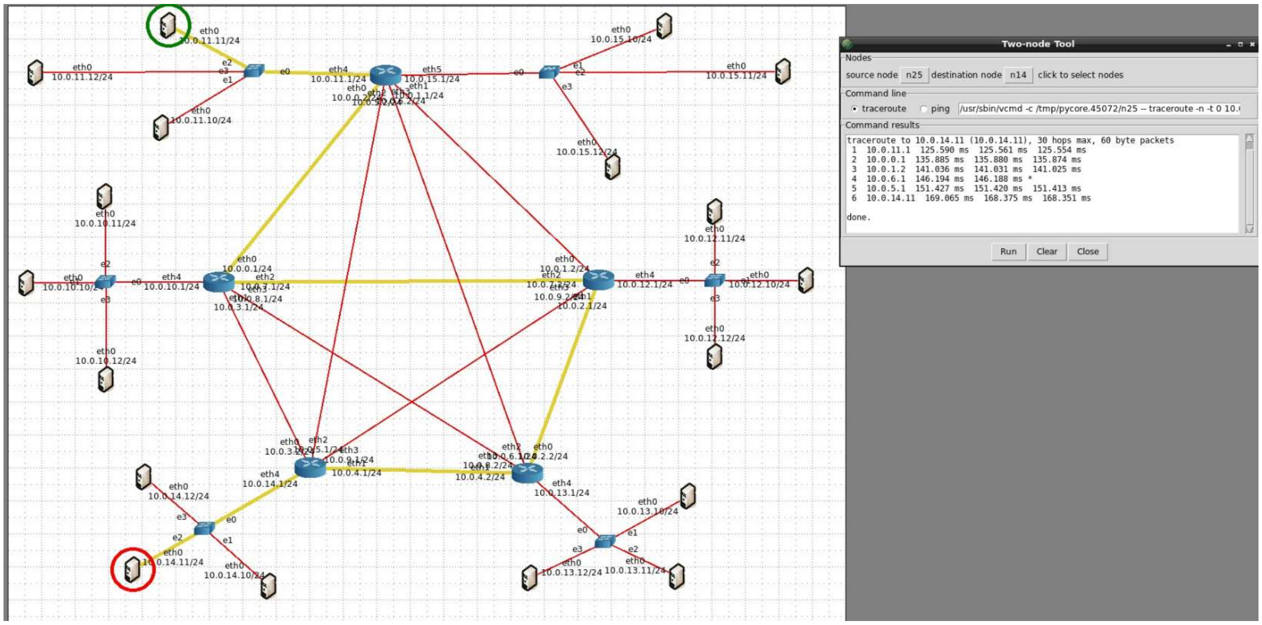
```
▸ Frame 14: 148 bytes on wire (1184 bits), 148 bytes captured (1184 bits) on interface 0
▸ Linux cooked capture
▸ Internet Protocol Version 4, Src: 10.0.1.2, Dst: 224.0.0.5
▸ Open Shortest Path First
```

```
0000  00 02 00 01 00 06 00 00  00 aa 00 03 00 00 08 00   ........ ........
0010  45 c0 00 84 96 93 00 00  01 59 36 c7 0a 00 01 02   E....... .Y6.....
0020  e0 00 00 05 02 04 00 70  0a 00 01 02 00 00 00 00   .......p ........
0030  43 37 00 00 00 00 00 00  00 00 00 00 00 00 00 01   C7...... ........
0040  00 01 02 01 0a 00 01 02  0a 00 01 02 80 00 00 12   ........ ........
0050  80 9e 00 54 00 00 00 05  0a 00 01 02 0a 00 01 02   ...T.... ........
0060  02 00 00 0a 0a 00 02 02  0a 00 02 01 02 00 00 0a   ........ ........
0070  0a 00 07 02 0a 00 07 02  02 00 00 0a 0a 00 09 01   ........ ........
0080  0a 00 09 02 02 00 00 0a  0a 00 0c 00 ff ff ff 00   ........ ........
0090  03 00 00 0a                                        ....
```

Above pictures show the generated traffic between the hosts before manipulating the routes.

- After manipulating the route configuration

In this case, we have manipulated the routes configuration



source node n25 destination node n14 click to select nodes

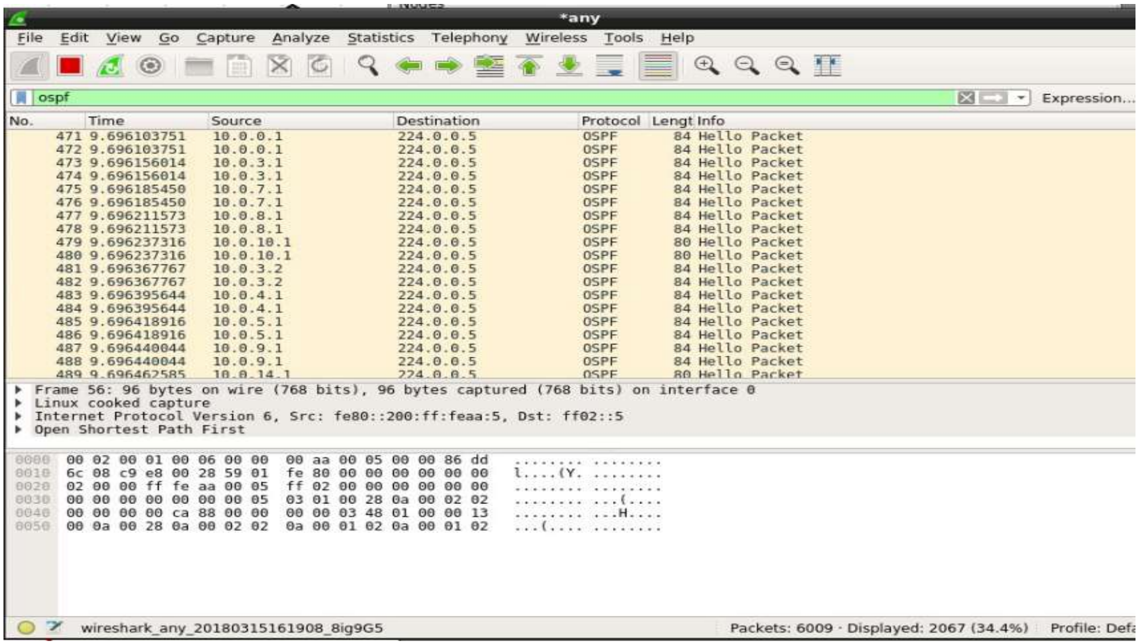
● traceroute ○ ping /usr/sbin/vcmd -c /tmp/pycore.45072/n25 -- traceroute -n -t 0 10.(

```
traceroute to 10.0.14.11 (10.0.14.11), 30 hops max, 60 byte packets
 1  10.0.11.1  125.590 ms  125.561 ms  125.554 ms
 2  10.0.0.1  135.885 ms  135.880 ms  135.874 ms
 3  10.0.1.2  141.036 ms  141.031 ms  141.025 ms
 4  10.0.6.1  146.194 ms  146.188 ms *
 5  10.0.5.1  151.427 ms  151.420 ms  151.413 ms
 6  10.0.14.11  169.065 ms  168.375 ms  168.351 ms

done.
```

- We have established the costs appropriately in order to route traffic accordingly.
- This includes generating traffic between hosts to demonstrate that the paths have been changed.

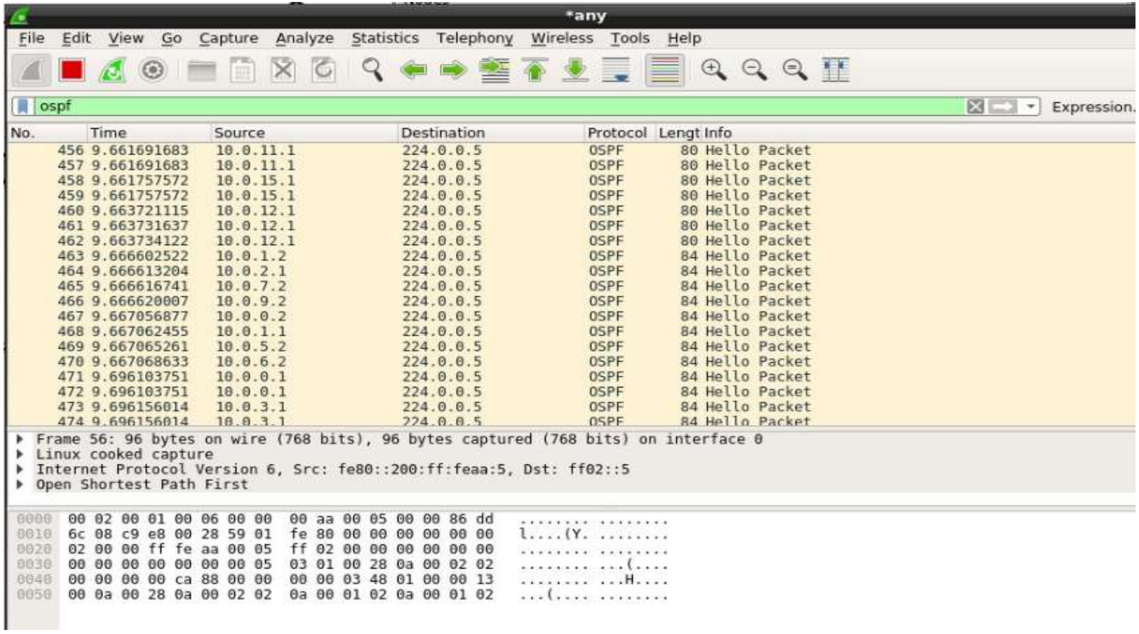
we want you to steer traffic around the maximum number of hops before reaching its destination; that is, traffic from one host (of your choice) travelling towards another host (of your choice, but at least attached to another router) should traverse every other router before arriving there.

As shown below, we can observe from Wireshark and understands which path OSPF messages have taken.



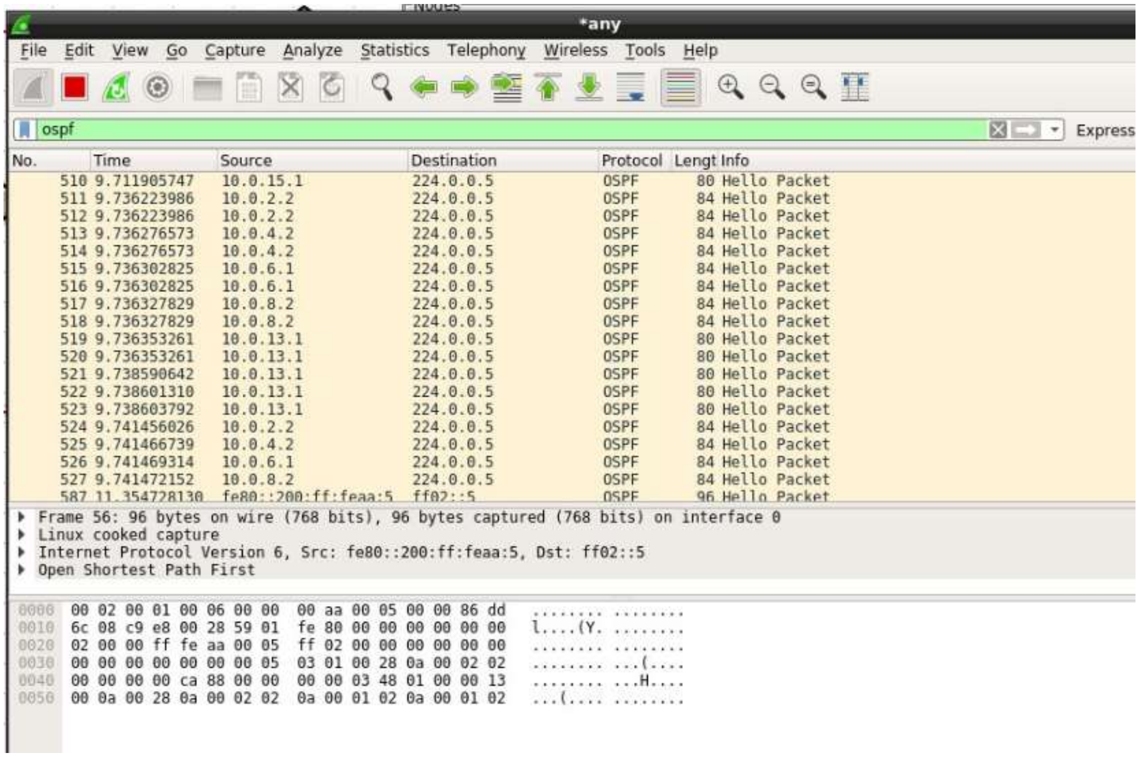
*any
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Expression...

```
ospf

No.   Time         Source       Destination  Protocol Lengt Info
  471 9.696103751  10.0.0.1     224.0.0.5    OSPF      84 Hello Packet
  472 9.696103751  10.0.0.1     224.0.0.5    OSPF      84 Hello Packet
  473 9.696156014  10.0.3.1     224.0.0.5    OSPF      84 Hello Packet
  474 9.696156014  10.0.3.1     224.0.0.5    OSPF      84 Hello Packet
  475 9.696185450  10.0.7.1     224.0.0.5    OSPF      84 Hello Packet
  476 9.696185450  10.0.7.1     224.0.0.5    OSPF      84 Hello Packet
  477 9.696211573  10.0.8.1     224.0.0.5    OSPF      84 Hello Packet
  478 9.696211573  10.0.8.1     224.0.0.5    OSPF      84 Hello Packet
  479 9.696237316  10.0.10.1    224.0.0.5    OSPF      80 Hello Packet
  480 9.696237316  10.0.10.1    224.0.0.5    OSPF      80 Hello Packet
  481 9.696367767  10.0.3.2     224.0.0.5    OSPF      84 Hello Packet
  482 9.696367767  10.0.3.2     224.0.0.5    OSPF      84 Hello Packet
  483 9.696395644  10.0.4.1     224.0.0.5    OSPF      84 Hello Packet
  484 9.696395644  10.0.4.1     224.0.0.5    OSPF      84 Hello Packet
  485 9.696418916  10.0.5.1     224.0.0.5    OSPF      84 Hello Packet
  486 9.696418916  10.0.5.1     224.0.0.5    OSPF      84 Hello Packet
  487 9.696440044  10.0.9.1     224.0.0.5    OSPF      84 Hello Packet
  488 9.696440044  10.0.9.1     224.0.0.5    OSPF      84 Hello Packet
  489 9.696462585  10.0.14.1    224.0.0.5    OSPF      80 Hello Packet

▸ Frame 56: 96 bytes on wire (768 bits), 96 bytes captured (768 bits) on interface 0
▸ Linux cooked capture
▸ Internet Protocol Version 6, Src: fe80::200:ff:feaa:5, Dst: ff02::5
▸ Open Shortest Path First

0000  00 02 00 01 00 06 00 00  00 aa 00 05 00 00 86 dd   ........ ........
0010  6c 08 c9 e8 00 28 59 01  fe 80 00 00 00 00 00 00   l....(Y. ........
0020  02 00 00 ff fe aa 00 05  ff 02 00 00 00 00 00 00   ........ ........
0030  00 00 00 00 00 00 00 05  03 01 00 28 0a 00 02 02   ........ ...(....
0040  00 00 00 00 ca 88 00 00  00 00 03 48 01 00 00 13   ........ ...H....
0050  00 0a 00 28 0a 00 02 02  0a 00 01 02 0a 00 01 02   ...(.... ........

wireshark_any_20180315161908_8ig9G5      Packets: 6009 · Displayed: 2067 (34.4%)      Profile: Defa
```



*any
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Expression.

```
ospf

No.   Time         Source       Destination  Protocol Lengt Info
  456 9.661691683  10.0.11.1    224.0.0.5    OSPF      80 Hello Packet
  457 9.661691683  10.0.11.1    224.0.0.5    OSPF      80 Hello Packet
  458 9.661757572  10.0.15.1    224.0.0.5    OSPF      80 Hello Packet
  459 9.661757572  10.0.15.1    224.0.0.5    OSPF      80 Hello Packet
  460 9.663721115  10.0.12.1    224.0.0.5    OSPF      80 Hello Packet
  461 9.663731637  10.0.12.1    224.0.0.5    OSPF      80 Hello Packet
  462 9.663734122  10.0.12.1    224.0.0.5    OSPF      80 Hello Packet
  463 9.666602522  10.0.1.2     224.0.0.5    OSPF      84 Hello Packet
  464 9.666613204  10.0.2.1     224.0.0.5    OSPF      84 Hello Packet
  465 9.666616741  10.0.7.2     224.0.0.5    OSPF      84 Hello Packet
  466 9.666620007  10.0.9.2     224.0.0.5    OSPF      84 Hello Packet
  467 9.667056877  10.0.0.2     224.0.0.5    OSPF      84 Hello Packet
  468 9.667062455  10.0.1.1     224.0.0.5    OSPF      84 Hello Packet
  469 9.667065261  10.0.5.2     224.0.0.5    OSPF      84 Hello Packet
  470 9.667068633  10.0.6.2     224.0.0.5    OSPF      84 Hello Packet
  471 9.696103751  10.0.0.1     224.0.0.5    OSPF      84 Hello Packet
  472 9.696103751  10.0.0.1     224.0.0.5    OSPF      84 Hello Packet
  473 9.696156014  10.0.3.1     224.0.0.5    OSPF      84 Hello Packet
  474 9.696156014  10.0.3.1     224.0.0.5    OSPF      84 Hello Packet

▸ Frame 56: 96 bytes on wire (768 bits), 96 bytes captured (768 bits) on interface 0
▸ Linux cooked capture
▸ Internet Protocol Version 6, Src: fe80::200:ff:feaa:5, Dst: ff02::5
▸ Open Shortest Path First

0000  00 02 00 01 00 06 00 00  00 aa 00 05 00 00 86 dd   ........ ........
0010  6c 08 c9 e8 00 28 59 01  fe 80 00 00 00 00 00 00   l....(Y. ........
0020  02 00 00 ff fe aa 00 05  ff 02 00 00 00 00 00 00   ........ ........
0030  00 00 00 00 00 00 00 05  03 01 00 28 0a 00 02 02   ........ ...(....
0040  00 00 00 00 ca 88 00 00  00 00 03 48 01 00 00 13   ........ ...H....
0050  00 0a 00 28 0a 00 02 02  0a 00 01 02 0a 00 01 02   ...(.... ........
```

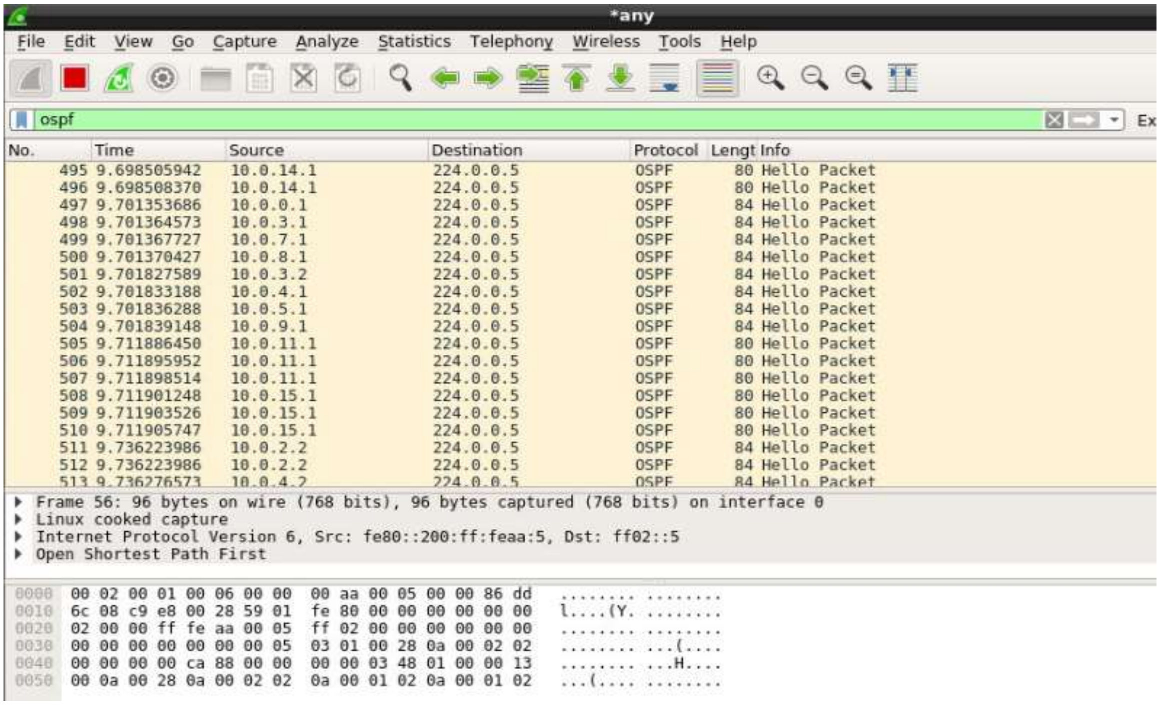

*any
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Express

```
ospf
```

```
No.      Time          Source               Destination   Protocol Lengt Info
     510 9.711905747   10.0.15.1            224.0.0.5     OSPF      80 Hello Packet
     511 9.736223986   10.0.2.2             224.0.0.5     OSPF      84 Hello Packet
     512 9.736223986   10.0.2.2             224.0.0.5     OSPF      84 Hello Packet
     513 9.736276573   10.0.4.2             224.0.0.5     OSPF      84 Hello Packet
     514 9.736276573   10.0.4.2             224.0.0.5     OSPF      84 Hello Packet
     515 9.736302825   10.0.6.1             224.0.0.5     OSPF      84 Hello Packet
     516 9.736302825   10.0.6.1             224.0.0.5     OSPF      84 Hello Packet
     517 9.736327829   10.0.8.2             224.0.0.5     OSPF      84 Hello Packet
     518 9.736327829   10.0.8.2             224.0.0.5     OSPF      84 Hello Packet
     519 9.736353261   10.0.13.1            224.0.0.5     OSPF      80 Hello Packet
     520 9.736353261   10.0.13.1            224.0.0.5     OSPF      80 Hello Packet
     521 9.738590642   10.0.13.1            224.0.0.5     OSPF      80 Hello Packet
     522 9.738601310   10.0.13.1            224.0.0.5     OSPF      80 Hello Packet
     523 9.738603792   10.0.13.1            224.0.0.5     OSPF      80 Hello Packet
     524 9.741456026   10.0.2.2             224.0.0.5     OSPF      84 Hello Packet
     525 9.741466739   10.0.4.2             224.0.0.5     OSPF      84 Hello Packet
     526 9.741469314   10.0.6.1             224.0.0.5     OSPF      84 Hello Packet
     527 9.741472152   10.0.8.2             224.0.0.5     OSPF      84 Hello Packet
     587 11.354728130  fe80::200:ff:feaa:5  ff02::5       OSPF      96 Hello Packet
```

```
▸ Frame 56: 96 bytes on wire (768 bits), 96 bytes captured (768 bits) on interface 0
▸ Linux cooked capture
▸ Internet Protocol Version 6, Src: fe80::200:ff:feaa:5, Dst: ff02::5
▸ Open Shortest Path First
```

```
0000  00 02 00 01 00 06 00 00  00 aa 00 05 00 00 86 dd   ........ ........
0010  6c 08 c9 e8 00 28 59 01  fe 80 00 00 00 00 00 00   l....(Y. ........
0020  02 00 00 ff fe aa 00 05  ff 02 00 00 00 00 00 00   ........ ........
0030  00 00 00 00 00 00 00 05  03 01 00 28 0a 00 02 02   ........ ...(....
0040  00 00 00 00 ca 88 00 00  00 00 03 48 01 00 00 13   ........ ...H....
0050  00 0a 00 28 0a 00 02 02  0a 00 01 02 0a 00 01 02   ...(.... ........
```


*any
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Ex

```
ospf
```

```
No.      Time          Source               Destination   Protocol Lengt Info
     495 9.698505942   10.0.14.1            224.0.0.5     OSPF      80 Hello Packet
     496 9.698508370   10.0.14.1            224.0.0.5     OSPF      80 Hello Packet
     497 9.701353686   10.0.0.1             224.0.0.5     OSPF      84 Hello Packet
     498 9.701364573   10.0.3.1             224.0.0.5     OSPF      84 Hello Packet
     499 9.701367727   10.0.7.1             224.0.0.5     OSPF      84 Hello Packet
     500 9.701370427   10.0.8.1             224.0.0.5     OSPF      84 Hello Packet
     501 9.701827589   10.0.3.2             224.0.0.5     OSPF      84 Hello Packet
     502 9.701833188   10.0.4.1             224.0.0.5     OSPF      84 Hello Packet
     503 9.701836288   10.0.5.1             224.0.0.5     OSPF      84 Hello Packet
     504 9.701839148   10.0.9.1             224.0.0.5     OSPF      84 Hello Packet
     505 9.711886450   10.0.11.1            224.0.0.5     OSPF      80 Hello Packet
     506 9.711895952   10.0.11.1            224.0.0.5     OSPF      80 Hello Packet
     507 9.711898514   10.0.11.1            224.0.0.5     OSPF      80 Hello Packet
     508 9.711901248   10.0.15.1            224.0.0.5     OSPF      80 Hello Packet
     509 9.711903526   10.0.15.1            224.0.0.5     OSPF      80 Hello Packet
     510 9.711905747   10.0.15.1            224.0.0.5     OSPF      80 Hello Packet
     511 9.736223986   10.0.2.2             224.0.0.5     OSPF      84 Hello Packet
     512 9.736223986   10.0.2.2             224.0.0.5     OSPF      84 Hello Packet
     513 9.736276573   10.0.4.2             224.0.0.5     OSPF      84 Hello Packet
```

```
▸ Frame 56: 96 bytes on wire (768 bits), 96 bytes captured (768 bits) on interface 0
▸ Linux cooked capture
▸ Internet Protocol Version 6, Src: fe80::200:ff:feaa:5, Dst: ff02::5
▸ Open Shortest Path First
```

```
0000  00 02 00 01 00 06 00 00  00 aa 00 05 00 00 86 dd   ........ ........
0010  6c 08 c9 e8 00 28 59 01  fe 80 00 00 00 00 00 00   l....(Y. ........
0020  02 00 00 ff fe aa 00 05  ff 02 00 00 00 00 00 00   ........ ........
0030  00 00 00 00 00 00 00 05  03 01 00 28 0a 00 02 02   ........ ...(....
0040  00 00 00 00 ca 88 00 00  00 00 03 48 01 00 00 13   ........ ...H....
0050  00 0a 00 28 0a 00 02 02  0a 00 01 02 0a 00 01 02   ...(.... ........
```

We can compare the Wireshark results and observe the change in routes before and after manipulation.