Cairo University Faculty of Engineering
Dept. of Electronics and Electrical Communications Engineering

ELC-3080

Networks Project

Code	BN.	Sec.	Name
9210695	6	3	علياء عصام الدين نجيب محمد
9210776	16	3	عمرو عبد المتجلي احمد متولي

2.Effect of TCP window size: Question 2.i: n11 as server & n7 as client

```
root@n7:/tmp/pycore.42687/n7.conf# iperf3 -c 10.0.13.20 -t 40 -i 10 -w 1k
Connecting to host 10.0.13.20, port 5201
[  4] local 10.0.12.20 port 36488 connected to 10.0.13.20 port 5201
[ ID] Interval           Transfer     Bandwidth       Retr  Cwnd
[  4]   0.00-10.00  sec  1.20 MBytes  1.00 Mbits/sec    0   5.45 KBytes
[  4]  10.00-20.00  sec  1.40 MBytes  1.18 Mbits/sec    0   5.45 KBytes
[  4]  20.00-30.00  sec  1.37 MBytes  1.15 Mbits/sec    0   5.45 KBytes
[  4]  30.00-40.00  sec  1.40 MBytes  1.17 Mbits/sec    0   5.45 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Retr
[  4]   0.00-40.00  sec  5.37 MBytes  1.13 Mbits/sec    0             sender
[  4]   0.00-40.00  sec  5.37 MBytes  1.13 Mbits/sec                  receiver

iperf Done.
```

Window of size 1KB

```
root@n7:/tmp/pycore.42687/n7.conf# iperf3 -c 10.0.13.20 -t 40 -i 10 -w 2k
Connecting to host 10.0.13.20, port 5201
[  4] local 10.0.12.20 port 36490 connected to 10.0.13.20 port 5201
[ ID] Interval           Transfer     Bandwidth       Retr  Cwnd
[  4]   0.00-10.00  sec  3.51 MBytes  2.94 Mbits/sec    0   7.07 KBytes
[  4]  10.00-20.00  sec  3.72 MBytes  3.12 Mbits/sec    0   4.24 KBytes
[  4]  20.00-30.00  sec  3.70 MBytes  3.11 Mbits/sec    0   4.24 KBytes
[  4]  30.00-40.00  sec  3.69 MBytes  3.10 Mbits/sec    0   4.24 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Retr
[  4]   0.00-40.00  sec  14.6 MBytes  3.07 Mbits/sec    0             sender
[  4]   0.00-40.00  sec  14.6 MBytes  3.07 Mbits/sec                  receiver

iperf Done.
```

Window size of 2KB

```
root@n7:/tmp/pycore.42687/n7.conf# iperf3 -c 10.0.13.20 -t 40 -i 10 -w 3k
Connecting to host 10.0.13.20, port 5201
[  4] local 10.0.12.20 port 36492 connected to 10.0.13.20 port 5201
[ ID] Interval           Transfer     Bandwidth       Retr  Cwnd
[  4]   0.00-10.00  sec  6.44 MBytes  5.40 Mbits/sec    0   14.1 KBytes
[  4]  10.00-20.00  sec  7.34 MBytes  6.16 Mbits/sec    0   14.1 KBytes
[  4]  20.00-30.00  sec  6.89 MBytes  5.78 Mbits/sec    0   14.1 KBytes
[  4]  30.00-40.00  sec  6.02 MBytes  5.05 Mbits/sec    0   14.1 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Retr
[  4]   0.00-40.00  sec  26.7 MBytes  5.60 Mbits/sec    0             sender
[  4]   0.00-40.00  sec  26.7 MBytes  5.60 Mbits/sec                  receiver

iperf Done.
```

Window size of 3KB

```
root@n7:/tmp/pycore.42687/n7.conf# iperf3 -c 10.0.13.20 -t 40 -i 10 -w 4k
Connecting to host 10.0.13.20, port 5201
[  4] local 10.0.12.20 port 36494 connected to 10.0.13.20 port 5201
[ ID] Interval           Transfer     Bandwidth       Retr  Cwnd
[  4]   0.00-10.00  sec  6.44 MBytes  5.40 Mbits/sec    0   14.1 KBytes
[  4]  10.00-20.00  sec  5.50 MBytes  4.61 Mbits/sec    0   14.1 KBytes
[  4]  20.00-30.00  sec  6.89 MBytes  5.78 Mbits/sec    0   14.1 KBytes
[  4]  30.00-40.00  sec  7.00 MBytes  5.87 Mbits/sec    0   14.1 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Retr
[  4]   0.00-40.00  sec  25.8 MBytes  5.42 Mbits/sec    0             sender
[  4]   0.00-40.00  sec  25.8 MBytes  5.42 Mbits/sec                  receiver

iperf Done.
```

Window size of 4KB

```
root@n7:/tmp/pycore.42687/n7.conf# iperf3 -c 10.0.13.20 -t 40 -i 10 -w 5k
Connecting to host 10.0.13.20, port 5201
[  4] local 10.0.12.20 port 36496 connected to 10.0.13.20 port 5201
[ ID] Interval           Transfer     Bandwidth       Retr  Cwnd
[  4]   0.00-10.00  sec  6.35 MBytes  5.32 Mbits/sec    0   14.1 KBytes
[  4]  10.00-20.00  sec  6.20 MBytes  5.20 Mbits/sec    0   14.1 KBytes
[  4]  20.00-30.00  sec  6.95 MBytes  5.83 Mbits/sec    0   14.1 KBytes
[  4]  30.00-40.00  sec  6.96 MBytes  5.84 Mbits/sec    0   14.1 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Retr
[  4]   0.00-40.00  sec  26.5 MBytes  5.55 Mbits/sec    0             sender
[  4]   0.00-40.00  sec  26.5 MBytes  5.55 Mbits/sec                  receiver

iperf Done.
```

Window size of 5KB

```
root@n7:/tmp/pycore.42687/n7.conf# iperf3 -c 10.0.13.20 -t 40 -i 10 -w 6k
Connecting to host 10.0.13.20, port 5201
[  4] local 10.0.12.20 port 36498 connected to 10.0.13.20 port 5201
[ ID] Interval           Transfer     Bandwidth       Retr  Cwnd
[  4]   0.00-10.00  sec   154 KBytes   126 Kbits/sec   86   4.24 KBytes
[  4]  10.00-20.00  sec   139 KBytes   114 Kbits/sec   85   4.24 KBytes
[  4]  20.00-30.00  sec   148 KBytes   122 Kbits/sec   89   4.24 KBytes
[  4]  30.00-40.00  sec   139 KBytes   114 Kbits/sec   85   4.24 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Retr
[  4]   0.00-40.00  sec   580 KBytes   119 Kbits/sec  345             sender
[  4]   0.00-40.00  sec   570 KBytes   117 Kbits/sec                  receiver

iperf Done.
```

Window size of 6KB

```
root@n7:/tmp/pycore.42687/n7.conf# iperf3 -c 10.0.13.20 -t 40 -i 10 -w 12k
Connecting to host 10.0.13.20, port 5201
[  4] local 10.0.12.20 port 36500 connected to 10.0.13.20 port 5201
[ ID] Interval           Transfer     Bandwidth       Retr  Cwnd
[  4]   0.00-10.00  sec  1.84 MBytes  1.54 Mbits/sec  317   4.24 KBytes
[  4]  10.00-20.00  sec  2.41 MBytes  2.02 Mbits/sec  437   2.83 KBytes
[  4]  20.00-30.00  sec  2.46 MBytes  2.07 Mbits/sec  446   2.83 KBytes
[  4]  30.00-40.00  sec  2.51 MBytes  2.10 Mbits/sec  453   4.24 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Retr
[  4]   0.00-40.00  sec  9.21 MBytes  1.93 Mbits/sec  1653             sender
[  4]   0.00-40.00  sec  9.19 MBytes  1.93 Mbits/sec                  receiver

iperf Done.
```

Window size of 12KB

```
root@n7:/tmp/pycore.42687/n7.conf# iperf3 -c 10.0.13.20 -t 40 -i 10 -w 16k
Connecting to host 10.0.13.20, port 5201
[  4] local 10.0.12.20 port 36502 connected to 10.0.13.20 port 5201
[ ID] Interval           Transfer     Bandwidth       Retr  Cwnd
[  4]   0.00-10.00  sec  1.64 MBytes  1.38 Mbits/sec  249   2.83 KBytes
[  4]  10.00-20.00  sec   573 KBytes   469 Kbits/sec  140   4.24 KBytes
[  4]  20.00-30.00  sec   460 KBytes   376 Kbits/sec  130   4.24 KBytes
[  4]  30.00-40.00  sec   445 KBytes   365 Kbits/sec  128   5.66 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Retr
[  4]   0.00-40.00  sec  3.09 MBytes   648 Kbits/sec  647             sender
[  4]   0.00-40.00  sec  3.06 MBytes   642 Kbits/sec                  receiver

iperf Done.
```

Window size of 16KB

```
root@n7:/tmp/pycore.42687/n7.conf# iperf3 -c 10.0.13.20 -t 40 -i 10 -w 24k
Connecting to host 10.0.13.20, port 5201
[  4] local 10.0.12.20 port 36504 connected to 10.0.13.20 port 5201
[ ID] Interval           Transfer     Bandwidth       Retr  Cwnd
[  4]   0.00-10.00  sec   310 KBytes   254 Kbits/sec  114   8.48 KBytes
[  4]  10.00-20.00  sec   277 KBytes   227 Kbits/sec  112   8.48 KBytes
[  4]  20.00-30.00  sec   297 KBytes   243 Kbits/sec  118   5.66 KBytes
[  4]  30.00-40.00  sec   277 KBytes   227 Kbits/sec  112   5.66 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Retr
[  4]   0.00-40.00  sec  1.13 MBytes   238 Kbits/sec  456             sender
[  4]   0.00-40.00  sec  1.09 MBytes   229 Kbits/sec                  receiver

iperf Done.
```

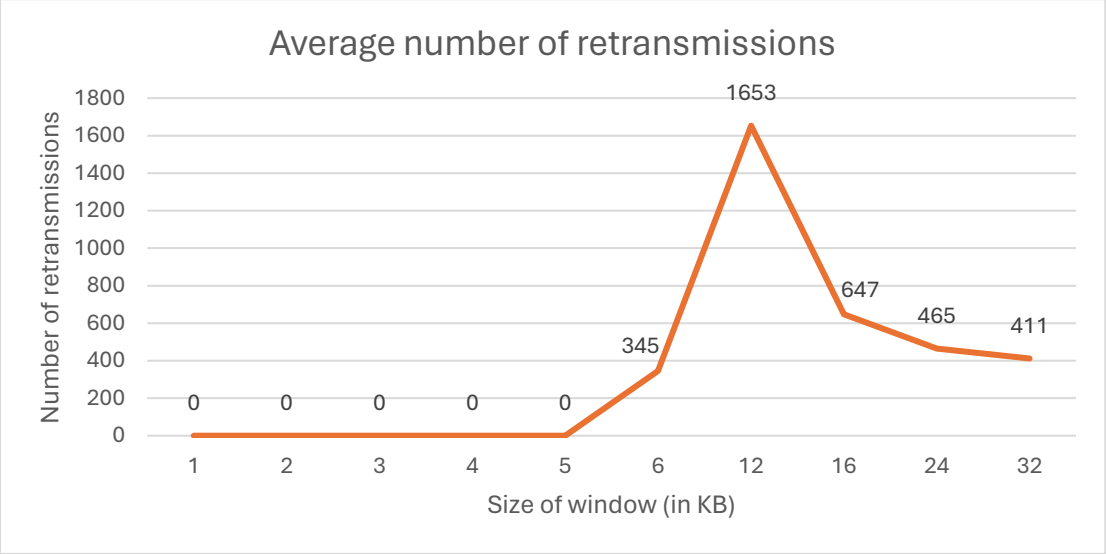
Window size of 24KB

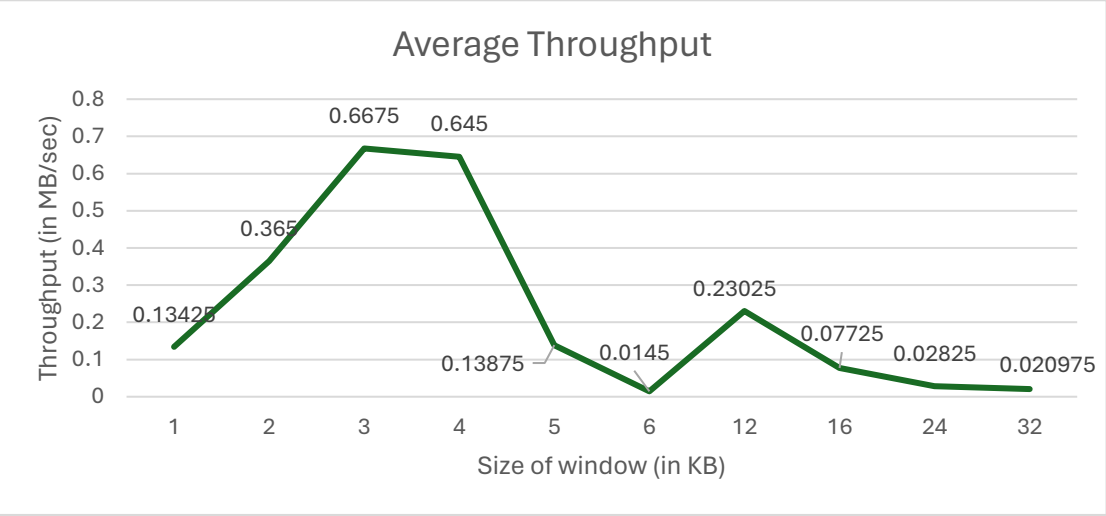
```
root@n7:/tmp/pycore.42687/n7.conf# iperf3 -c 10.0.13.20 -t 40 -i 10 -w 32k
Connecting to host 10.0.13.20, port 5201
[  4] local 10.0.12.20 port 36506 connected to 10.0.13.20 port 5201
[ ID] Interval           Transfer     Bandwidth       Retr  Cwnd
[  4]   0.00-10.00  sec   246 KBytes   202 Kbits/sec  105   7.07 KBytes
[  4]  10.00-20.00  sec   189 KBytes   155 Kbits/sec   97   14.1 KBytes
[  4]  20.00-30.00  sec   161 KBytes   132 Kbits/sec   93   22.6 KBytes
[  4]  30.00-40.00  sec   242 KBytes   198 Kbits/sec  116   5.66 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Retr
[  4]   0.00-40.00  sec   839 KBytes   172 Kbits/sec  411             sender
[  4]   0.00-40.00  sec   779 KBytes   160 Kbits/sec                  receiver

iperf Done.
```

Window size of 32KB

From the previous results we can calculate: Throughput = $\frac{\text{data transered by sender}}{\text{total time}}$

Size of window (in KB)	1	2	3	4	5	6	12	16	24	32
Average number of retransmissions	0	0	0	0	0	345	1653	647	465	411
Throughput (in MB/sec)	0.13425	0.365	0.6675	0.645	0.13875	0.0145	0.23025	0.07725	0.02825	0.02098

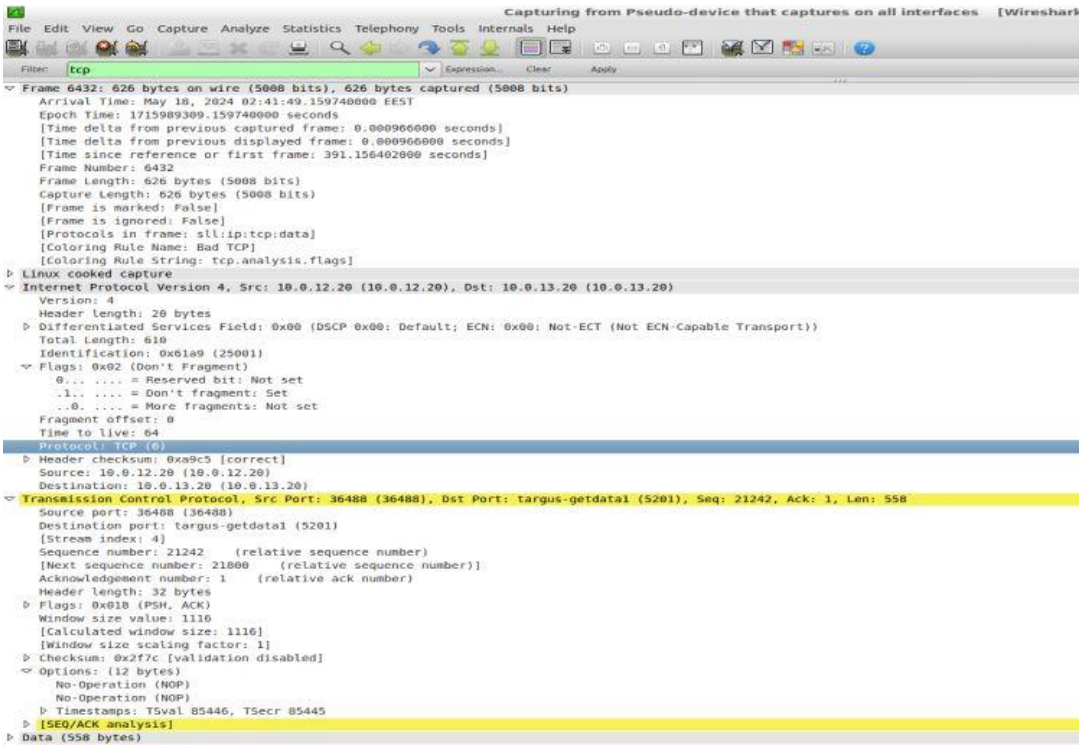




❖ **Comments:** From the 2 graphs we see that in the case of window sizes are in the range:

- From 1KB to 3 KB, the throughput increases as the number of window size increases while the number of retransmissions is zero and this makes sense.
- From 4KB to 6KB, the throughput decreases as the number of window size increases while the number of retransmissions is zero except at window size of 6KB it becomes 345, we can say that in this case the receiver informed the transmitter to reduce the throughput it is sending with in order to adjust the flow control to reduce the segment losses and prevent retransmission (wasting more time & resources), when the window size increased to 6KB we see that number of retransmissions is not equal to zero this means that the receiver buffer is full so the receiver needs to allocate more buffer if available.
- From 6KB to 12 KB, the throughput increased & the number of retransmissions also increased much as a result of this the sender reduced the throughput this explains the zigzag behavior as at higher window sizes as the window size increases, congestion is more likely to happen so sender reduces throughput thus number of retransmissions may probably decrease till a certain size at which buffer is full & number of retransmissions increases much repeating the scenario at 6KB & 12KB.

Question 2.ii: TCP data segment whose source is node n7:



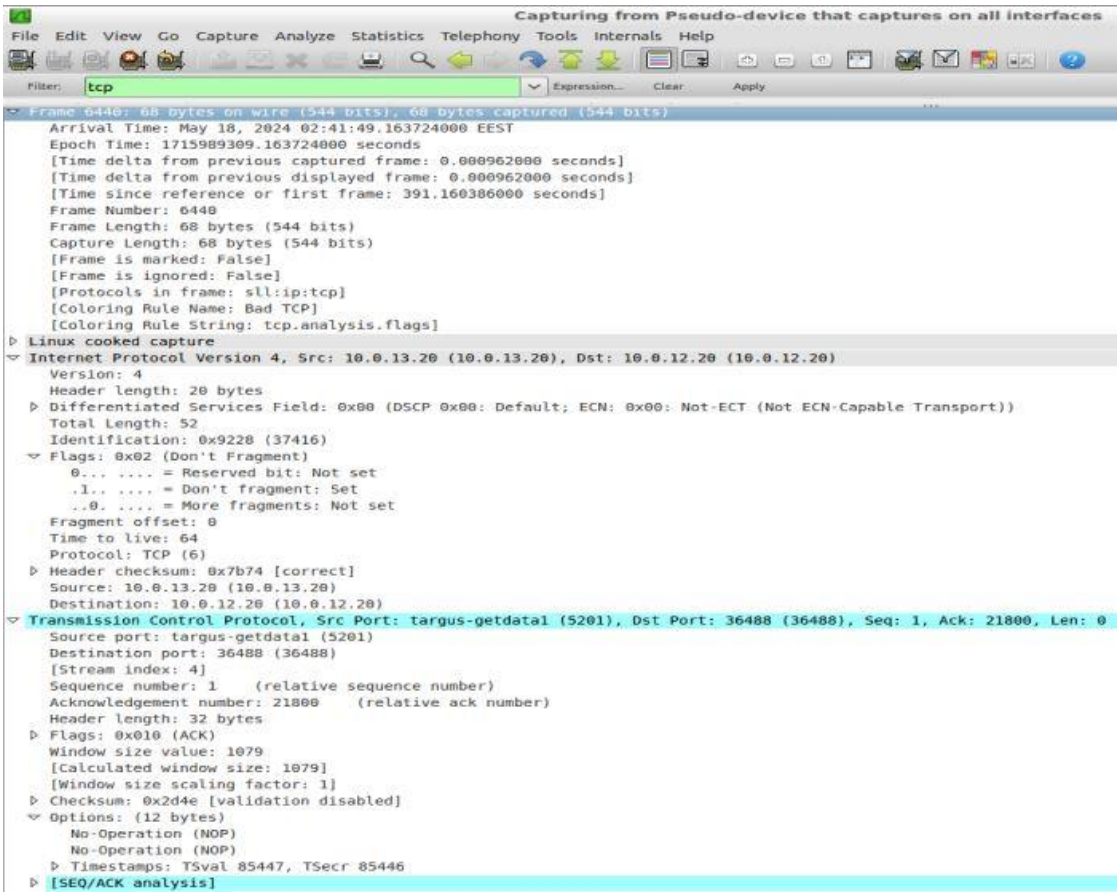
TCP packet from Wireshark

Frame length = 626 bytes, IP header length = 20 bytes, TCP header length = 32 bytes,

Ethernet header length = 16 bytes, TCP options length = 12 bytes

TCP options has only timestamps which help in calculating the Round-Trip Time (RTT) between the sender and receiver accurately. They are used for various purposes, including congestion control algorithms, retransmission timeouts, and performance measurements.

Question 2.ii: TCP acknowledgement segment whose source is node n11:



ACK packet from Wireshark

Frame length = 68 bytes, IP header length = 20 bytes, TCP header length = 32 bytes,

Ethernet header length = 16 bytes, TCP options length = 12 bytes

3.TCP short versus long paths:

Question 3.i:

```
root@n7:/tmp/pycore.42687/n7.conf# iperf3 -c 10.0.11.20 -t 40 -i 10 -w 4k
Connecting to host 10.0.11.20, port 5201
[  4] local 10.0.12.20 port 33522 connected to 10.0.11.20 port 5201
[ ID] Interval           Transfer     Bandwidth       Retr  Cwnd
[  4]   0.00-10.00  sec  4.02 MBytes  3.37 Mbits/sec    0   14.1 KBytes
[  4]  10.00-20.00  sec  4.24 MBytes  3.56 Mbits/sec    0   14.1 KBytes
[  4]  20.00-30.00  sec  3.06 MBytes  2.57 Mbits/sec    0   14.1 KBytes
[  4]  30.00-40.00  sec  4.66 MBytes  3.91 Mbits/sec    0   14.1 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Retr
[  4]   0.00-40.00  sec  16.0 MBytes  3.35 Mbits/sec    0             sender
[  4]   0.00-40.00  sec  16.0 MBytes  3.35 Mbits/sec                  receiver

iperf Done.
```

n7 to n8

```
root@n7:/tmp/pycore.42687/n7.conf# iperf3 -c 10.0.13.20 -t 40 -i 10 -w 4k
Connecting to host 10.0.13.20, port 5201
[  4] local 10.0.12.20 port 36494 connected to 10.0.13.20 port 5201
[ ID] Interval           Transfer     Bandwidth       Retr  Cwnd
[  4]   0.00-10.00  sec  6.44 MBytes  5.40 Mbits/sec    0   14.1 KBytes
[  4]  10.00-20.00  sec  5.50 MBytes  4.61 Mbits/sec    0   14.1 KBytes
[  4]  20.00-30.00  sec  6.89 MBytes  5.78 Mbits/sec    0   14.1 KBytes
[  4]  30.00-40.00  sec  7.00 MBytes  5.87 Mbits/sec    0   14.1 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Retr
[  4]   0.00-40.00  sec  25.8 MBytes  5.42 Mbits/sec    0             sender
[  4]   0.00-40.00  sec  25.8 MBytes  5.42 Mbits/sec                  receiver

iperf Done.
```

n7 to n11

- Throughput from n7 to n8 = 0.4 MB/sec
- Throughput from n7 to n11 = 0.645 MB/sec
- Throughput drops when connecting to n8 instead of n11 although capacities on the two paths are the same because the path from n7 to n8 is longer than that from n7 to n11 thus there are more delays.

4.Higher Link Capacity with Drops versus Reliable Lower Capacity:

From n7 to n11 with window size 4KB, configuring the link from n4 to n5:

```
root@n7:/tmp/pycore.42687/n7.conf# iperf3 -c 10.0.13.20 -t 40 -i 10 -w 4k
Connecting to host 10.0.13.20, port 5201
[  4] local 10.0.12.20 port 36510 connected to 10.0.13.20 port 5201
[ ID] Interval           Transfer     Bandwidth       Retr  Cwnd
[  4]   0.00-10.00  sec  5.65 MBytes  4.74 Mbits/sec    0   14.1 KBytes
[  4]  10.00-20.00  sec  6.85 MBytes  5.75 Mbits/sec    0   14.1 KBytes
[  4]  20.00-30.00  sec  6.26 MBytes  5.25 Mbits/sec    0   14.1 KBytes
[  4]  30.00-40.00  sec  7.08 MBytes  5.94 Mbits/sec    0   14.1 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Retr
[  4]   0.00-40.00  sec  25.8 MBytes  5.42 Mbits/sec    0             sender
[  4]   0.00-40.00  sec  25.8 MBytes  5.42 Mbits/sec                  receiver

iperf Done.
```

a. capacity of 10 Mbps with zero loss in both directions, Throughput = 0.645 MB/sec

```
root@n7:/tmp/pycore.42687/n7.conf# iperf3 -c 10.0.13.20 -t 40 -i 10 -w 4k
Connecting to host 10.0.13.20, port 5201
[  4] local 10.0.12.20 port 36512 connected to 10.0.13.20 port 5201
[ ID] Interval           Transfer     Bandwidth       Retr  Cwnd
[  4]   0.00-10.00  sec  3.39 MBytes  2.85 Mbits/sec    0   14.1 KBytes
[  4]  10.00-20.00  sec  3.41 MBytes  2.86 Mbits/sec    0   14.1 KBytes
[  4]  20.00-30.00  sec  3.41 MBytes  2.86 Mbits/sec    0   14.1 KBytes
[  4]  30.00-40.00  sec  3.40 MBytes  2.85 Mbits/sec    0   14.1 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Retr
[  4]   0.00-40.00  sec  13.6 MBytes  2.85 Mbits/sec    0             sender
[  4]   0.00-40.00  sec  13.6 MBytes  2.85 Mbits/sec                  receiver

iperf Done.
```

b. capacity of 3 Mbps with zero loss in both directions, Throughput = 0.34 MB/sec

```
root@n7:/tmp/pycore.42687/n7.conf# iperf3 -c 10.0.13.20 -t 40 -i 10 -w 4k
Connecting to host 10.0.13.20, port 5201
[  4] local 10.0.12.20 port 36514 connected to 10.0.13.20 port 5201
[ ID] Interval           Transfer     Bandwidth       Retr  Cwnd
[  4]   0.00-10.00  sec   556 KBytes   455 Kbits/sec   35   4.24 KBytes
[  4]  10.00-20.00  sec   656 KBytes   538 Kbits/sec   35   4.24 KBytes
[  4]  20.00-30.00  sec   611 KBytes   500 Kbits/sec   36   4.24 KBytes
[  4]  30.00-40.00  sec   699 KBytes   572 Kbits/sec   32   4.24 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Retr
[  4]   0.00-40.00  sec  2.46 MBytes   516 Kbits/sec  138             sender
[  4]   0.00-40.00  sec  2.46 MBytes   515 Kbits/sec                  receiver

iperf Done.
```

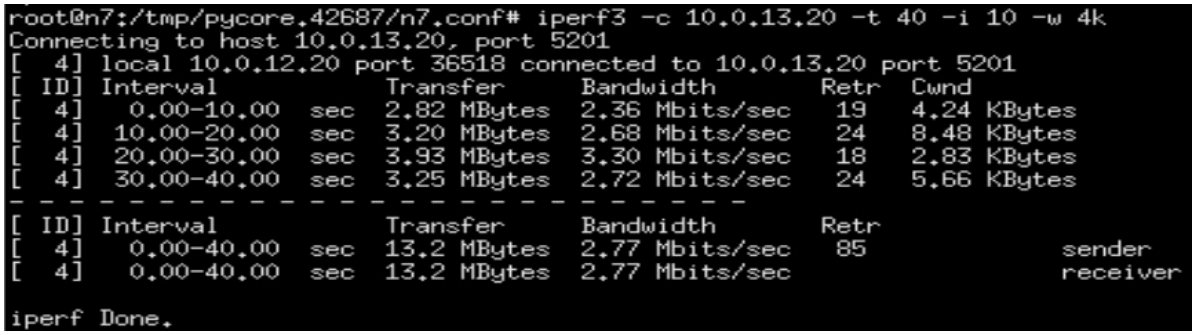
c. capacity of 10Mbps with 5% loss in both directions, Throughput = 0.0615 MB/sec

```
root@n7:/tmp/pycore.42687/n7.conf# iperf3 -c 10.0.13.20 -t 40 -i 10 -w 4k
Connecting to host 10.0.13.20, port 5201
[  4] local 10.0.12.20 port 36516 connected to 10.0.13.20 port 5201
[ ID] Interval           Transfer     Bandwidth       Retr  Cwnd
[  4]   0.00-10.00  sec   301 KBytes   247 Kbits/sec   28   4.24 KBytes
[  4]  10.00-20.00  sec   274 KBytes   225 Kbits/sec   33   4.24 KBytes
[  4]  20.00-30.00  sec   300 KBytes   246 Kbits/sec   32   2.83 KBytes
[  4]  30.00-40.00  sec  29.7 KBytes  24.3 Kbits/sec    9   2.83 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Retr
[  4]   0.00-40.00  sec   905 KBytes   185 Kbits/sec  102             sender
[  4]   0.00-40.00  sec   899 KBytes   184 Kbits/sec                  receiver

iperf Done.
```

d. capacity of 100Mbps with 10% loss in both directions, Throughput = 0.022625 MB/sec

```
iperf Done.
root@n7:/tmp/pycore.42687/n7.conf# iperf3 -c 10.0.13.20 -t 40 -i 10 -w 4k
Connecting to host 10.0.13.20, port 5201
[  4] local 10.0.12.20 port 36520 connected to 10.0.13.20 port 5201
[ ID] Interval           Transfer     Bandwidth       Retr  Cwnd
[  4]   0.00-10.00  sec  4.64 MBytes  3.89 Mbits/sec    7   8.48 KBytes
[  4]  10.00-20.00  sec  4.92 MBytes  4.13 Mbits/sec   15   8.48 KBytes
[  4]  20.00-30.00  sec  4.50 MBytes  3.77 Mbits/sec   17   8.48 KBytes
[  4]  30.00-40.00  sec  4.23 MBytes  3.55 Mbits/sec   16   8.48 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Retr
[  4]   0.00-40.00  sec  18.3 MBytes  3.84 Mbits/sec   55             sender
[  4]   0.00-40.00  sec  18.3 MBytes  3.83 Mbits/sec                  receiver

iperf Done.
```

e. capacity of 10Mbps with 1% loss in direction from n4 to n5 and 0% loss in other direction, Throughput = 0.4575 MB/sec



f. capacity of 10Mbps with 0% loss in direction from n4 to n5 and 1% loss in other direction, Throughput = 0.33 MB/sec

Question 4.i:

a) Throughput = 0.645 MB/sec
b) Throughput = 0.34 MB/sec
c) Throughput = 0.0615 MB/sec

Throughput in case b is greater than throughput in case c although link capacity in b is greater than that in c but in b there is no loss so zero retransmissions while in case c there is 5% loss in the link so there are number of retransmissions because data didn't reach the receiver or the ack didn't reach the sender.

Throughput in case a is greater than throughput in case b as link loss equals zero in both cases while the link capacity in case a is greater than that in case b so it makes sense that throughput in case a is greater than that in case b.

Question 4.ii:

b) Throughput = 0.34 MB/sec

c) Throughput = 0.0615 MB/sec

d) Throughput = 0.022625 MB/sec

Throughput in case b is greater than both cases c & d although link capacity is lower than them because link loss is zero, so case b is still better as we see (highest throughput & lowest number of retransmissions), also as we see throughput in case c is greater than that in case d.

Question 4.iii:

e) Throughput = 0.4575 MB/sec

f) Throughput = 0.33 MB/sec

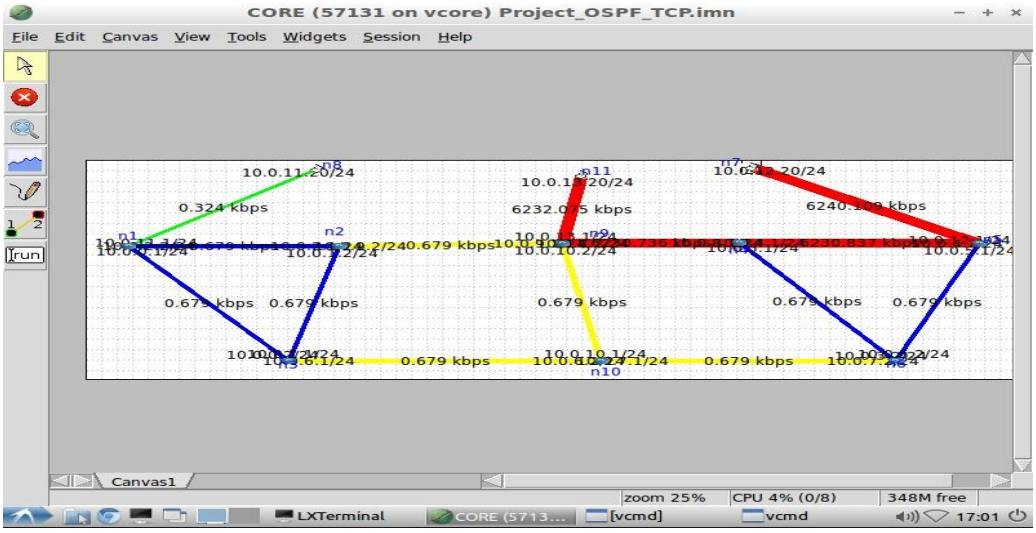
Throughput in case e is better than that in case f because the data segments that are large in sizes are sent on this link in the direction from n5 to n4 while the other direction sends

acknowledgments from the receiver to the sender where these acknowledgments are very small in size, this direction transmitting large length segments (from n5 to n4) has zero loss in case e while it has 1% loss in case f, this loss leads to retransmission thus reducing the throughput so it is better to retransmit the shorter length segments (ACKs) that's why case e is better than case f.

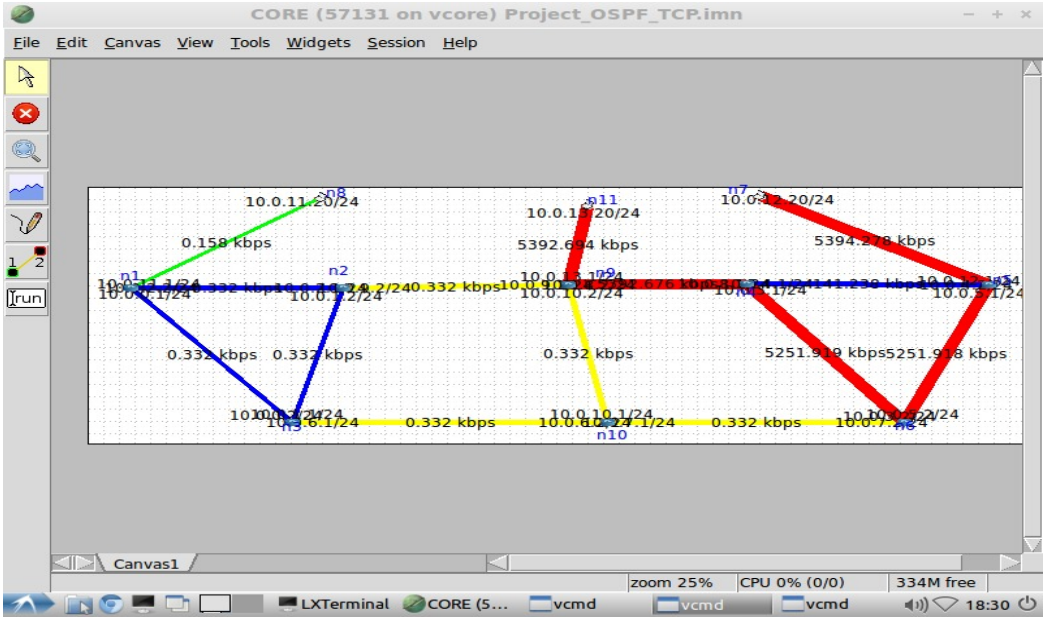
5.OSPF link cost changes:

Question 5.i:

The path between n7 and n11 for the network before changing the cost of the eth1 link:



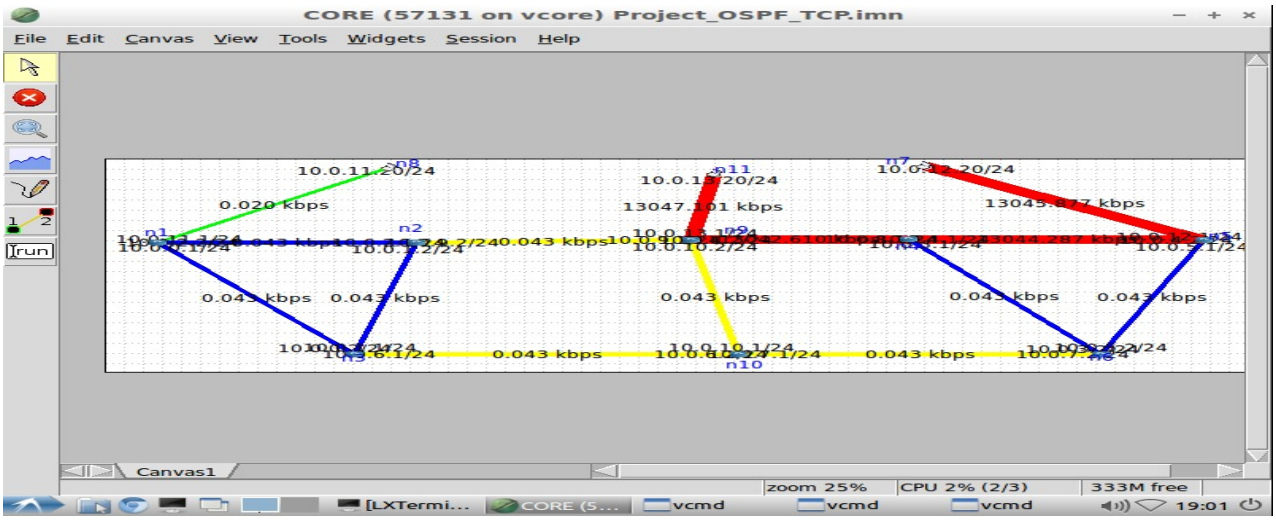
The path between n7 and n11 for the network after changing the cost of the eth1 link to be 40:



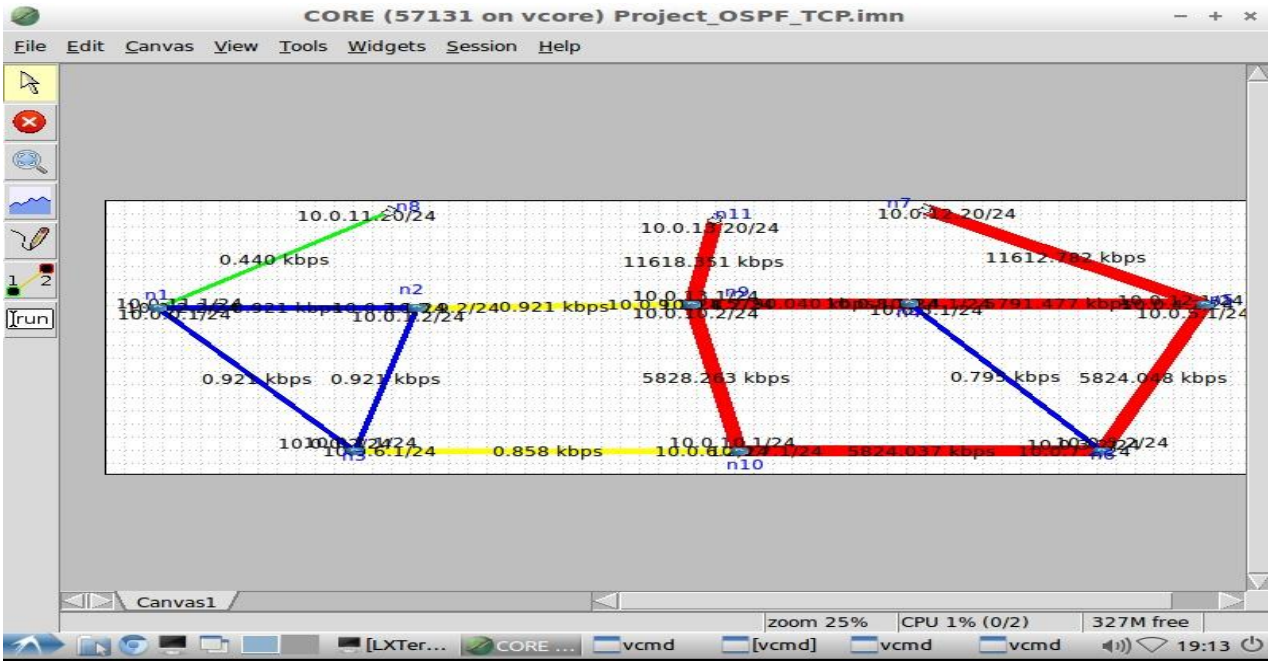
Comment on the above figures: Before increasing the link cost of eth1 link, the packets were sent through it with the least cost but after we increase the link cost for eth1 link connecting between node 4 and node 5, the congestion in this link increased, so the router selects an alternative path between node 7 and node 11 with lower cost using Dijkstra's Algorithm so router 5 choose eth0 link to send the packets with the least cost.

Question 5.ii:

The path between n7 and n11 for the network before changing the cost of the eth1 link:



The path between n7 and n11 for the network after changing the cost of the eth1 link to be 40:



Comment on the above figures:

- As we can see the path becomes so much longer after we increase the link cost for eth1 link connecting between node 4 and node 5 so the congestion in this link increased, as increasing the cost of a link makes the router select another path to send the packets from node 7 to n11 even if it's longer to decrease the congestion also to reduce collisions but the packets sent from node 11 to node 7 took the same path as the cost changed from n7 perspective only.
- So, we concluded that not all interfaces have the same link cost even if they are connected point to point as we can see in the case above, the link cost from node 4 to node 5 is 40, but the link cost from node 5 to node 4 is 10.

6.OSPF database updates:

Question 6.i:

```
root@n2:/tmp/pycore.54953/n2.conf# vtysh

Hello, this is Quagga (version 0.99.20.1).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

n2# show ip ospf database

       OSPF Router with ID (10.0.1.2)

                Router Link States (Area 0.0.0.0)

Link ID         ADV Router      Age  Seq#       CkSum  Link count
10.0.0.1        10.0.0.1       1108 0x80000008 0xf6bb 3
10.0.0.2        10.0.0.2       1107 0x8000000b 0x9809 3
10.0.1.2        10.0.1.2       1101 0x8000000a 0x8d0a 3
10.0.3.1        10.0.3.1         44 0x80000010 0x3b30 3
10.0.3.2        10.0.3.2       1109 0x80000009 0xafdd 3
10.0.5.1        10.0.5.1       1108 0x80000008 0x9405 3
10.0.6.2        10.0.6.2       1108 0x8000000a 0xcaac 3
10.0.8.2        10.0.8.2       1107 0x8000000b 0x2414 4

                Net Link States (Area 0.0.0.0)

Link ID         ADV Router      Age  Seq#       CkSum
10.0.0.2        10.0.0.2       1107 0x80000001 0x5fcb
10.0.1.2        10.0.1.2       1111 0x80000001 0x64c2
10.0.2.1        10.0.1.2       1101 0x80000001 0x51d6
10.0.3.2        10.0.3.2       1114 0x80000001 0x6bb3
10.0.4.2        10.0.5.1       1114 0x80000001 0x64b7
10.0.5.1        10.0.5.1         85 0x80000002 0x67b2
10.0.6.2        10.0.6.2       1113 0x80000001 0x4bcc
10.0.7.1        10.0.6.2       1108 0x80000001 0x59bb
10.0.8.2        10.0.8.2       1112 0x80000001 0x52bd
10.0.9.1        10.0.8.2       1107 0x80000001 0x2de3
10.0.10.2       10.0.8.2         81 0x80000002 0x5fa9
```

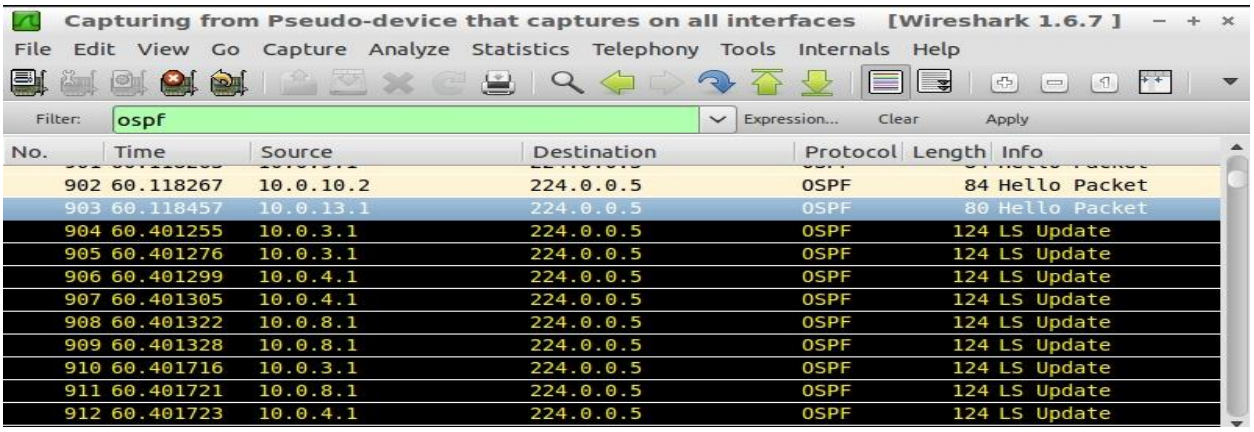
OSPF database

```
n2# show ip ospf route
============ OSPF network routing table ============
N    10.0.0.0/24           [20] area: 0.0.0.0
                           via 10.0.1.1, eth0
                           via 10.0.2.2, eth1
N    10.0.1.0/24           [10] area: 0.0.0.0
                           directly attached to eth0
N    10.0.2.0/24           [10] area: 0.0.0.0
                           directly attached to eth1
N    10.0.3.0/24           [30] area: 0.0.0.0
                           via 10.0.9.1, eth2
N    10.0.4.0/24           [50] area: 0.0.0.0
                           via 10.0.1.1, eth0
                           via 10.0.9.1, eth2
N    10.0.5.0/24           [40] area: 0.0.0.0
                           via 10.0.1.1, eth0
                           via 10.0.9.1, eth2
N    10.0.6.0/24           [20] area: 0.0.0.0
                           via 10.0.1.1, eth0
N    10.0.7.0/24           [30] area: 0.0.0.0
                           via 10.0.1.1, eth0
                           via 10.0.9.1, eth2
N    10.0.8.0/24           [20] area: 0.0.0.0
                           via 10.0.9.1, eth2
N    10.0.9.0/24           [10] area: 0.0.0.0
                           directly attached to eth2
N    10.0.10.0/24          [20] area: 0.0.0.0
                           via 10.0.9.1, eth2
N    10.0.11.0/24          [20] area: 0.0.0.0
                           via 10.0.2.2, eth1
N    10.0.12.0/24          [50] area: 0.0.0.0
                           via 10.0.1.1, eth0
                           via 10.0.9.1, eth2
N    10.0.13.0/24          [20] area: 0.0.0.0
                           via 10.0.9.1, eth2

============ OSPF router routing table =============

============ OSPF external routing table ===========
```

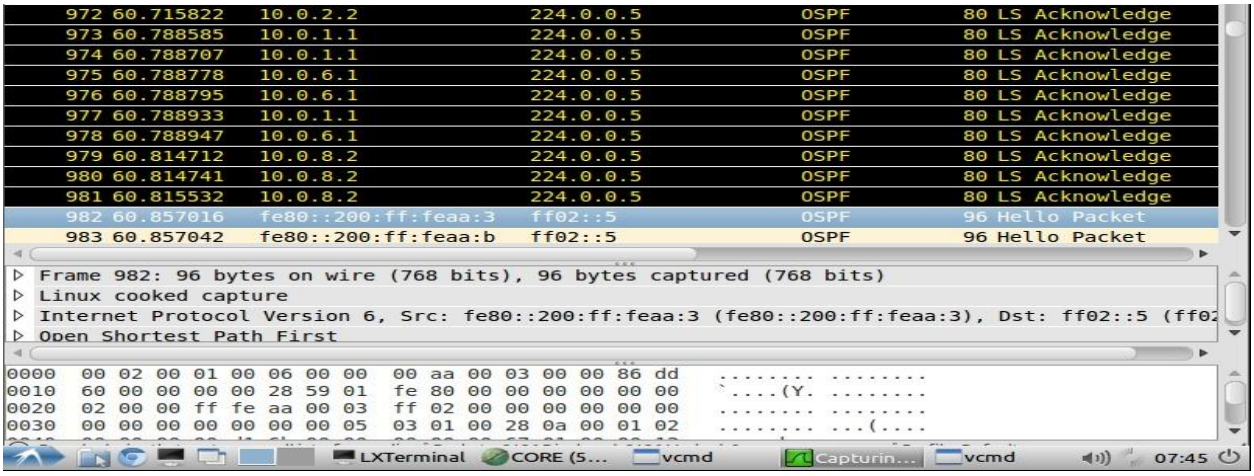
OSPF Route

Comment on the above figures: As we can see there are many paths having the same cost so the switch can forward the packets via any one of these paths and in case of one of these paths are down the switch will forward the packets via the other one having the same cost.

Question 6.ii:



:



The required time for exchanging link state packets and adjusting routing tables can be calculated from above figures:

Time taken = 60.815532 – 60.401255 = 0.414277 **sec**

Question 6.iii:

```
n2# show ip ospf database

       OSPF Router with ID (10.0.1.2)

                Router Link States (Area 0.0.0.0)

Link ID         ADV Router      Age  Seq#       CkSum  Link count
10.0.0.1        10.0.0.1         82 0x8000000a 0xf2bd 3
10.0.0.2        10.0.0.2       1834 0x8000000c 0x960a 3
10.0.1.2        10.0.1.2       1823 0x8000000b 0x8b0b 3
10.0.3.1        10.0.3.1         57 0x80000018 0xa649 1
10.0.3.2        10.0.3.2         74 0x8000000c 0xf2a7 3
10.0.5.1        10.0.5.1         25 0x8000000b 0xfda9 3
10.0.6.2        10.0.6.2       1835 0x8000000b 0xc8ad 3
10.0.8.2        10.0.8.2          7 0x8000000d 0x2823 4

                Net Link States (Area 0.0.0.0)

Link ID         ADV Router      Age  Seq#       CkSum
10.0.0.2        10.0.0.2       1214 0x80000002 0x5dcc
10.0.1.2        10.0.1.2       1753 0x80000002 0x62c3
10.0.2.1        10.0.1.2       1203 0x80000002 0x4fd7
10.0.5.1        10.0.5.1        806 0x80000003 0x65b3
10.0.6.2        10.0.6.2       1755 0x80000002 0x49cd
10.0.7.1        10.0.6.2       1215 0x80000002 0x57bc
10.0.8.2        10.0.8.2       3600 0x80000002 0x50be
10.0.9.1        10.0.8.2        152 0x80000003 0x29e5
10.0.10.2       10.0.8.2        834 0x80000003 0x5daa
```

OSPF database

```
n2# show ip ospf route
============ OSPF network routing table ============
N    10.0.0.0/24           [20] area: 0.0.0.0
                           via 10.0.1.1, eth0
                           via 10.0.2.2, eth1
N    10.0.1.0/24           [10] area: 0.0.0.0
                           directly attached to eth0
N    10.0.2.0/24           [10] area: 0.0.0.0
                           directly attached to eth1
N    10.0.3.0/24           [40] area: 0.0.0.0
                           via 10.0.1.1, eth0
                           via 10.0.9.1, eth2
N    10.0.4.0/24           [50] area: 0.0.0.0
                           via 10.0.1.1, eth0
                           via 10.0.9.1, eth2
N    10.0.5.0/24           [40] area: 0.0.0.0
                           via 10.0.1.1, eth0
                           via 10.0.9.1, eth2
N    10.0.6.0/24           [20] area: 0.0.0.0
                           via 10.0.1.1, eth0
N    10.0.7.0/24           [30] area: 0.0.0.0
                           via 10.0.1.1, eth0
                           via 10.0.9.1, eth2
N    10.0.8.0/24           [20] area: 0.0.0.0
                           via 10.0.9.1, eth2
N    10.0.9.0/24           [10] area: 0.0.0.0
                           directly attached to eth2
N    10.0.10.0/24          [20] area: 0.0.0.0
                           via 10.0.9.1, eth2
N    10.0.11.0/24          [20] area: 0.0.0.0
                           via 10.0.2.2, eth1
N    10.0.12.0/24          [50] area: 0.0.0.0
                           via 10.0.1.1, eth0
                           via 10.0.9.1, eth2
N    10.0.13.0/24          [20] area: 0.0.0.0
                           via 10.0.9.1, eth2

============ OSPF router routing table =============
```

OSPF route

Comment on the above figures:

- After disconnecting node 4, there will be another node take its place which is node 2 which has higher cost than node 4, so if we look to the OSPF database we will see there are a lot of net link states are removed from it as they were connected to the interface of node 4.
- We conclude that OSPF protocol allows the routers to modify link states and routing tables dynamically.