LAB 2: Understanding of TCP(Transmission Control Protocol) using wireshark and netsim.

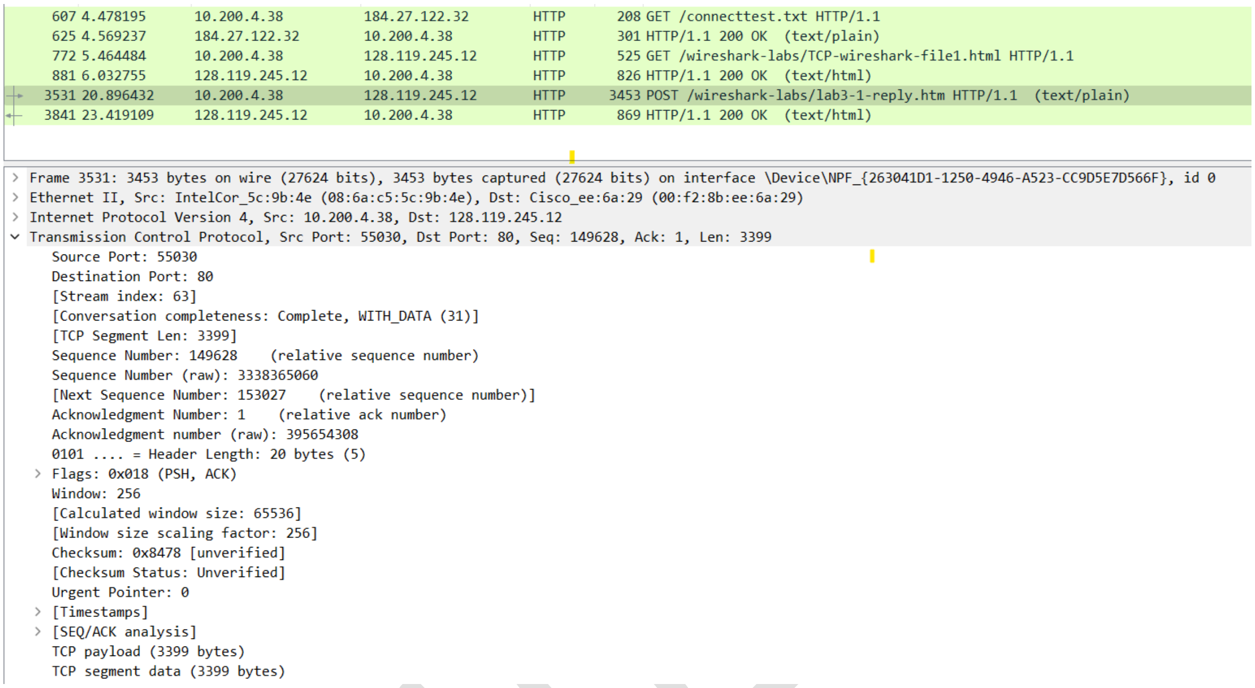
Akshar Panchani ID- 202101522

IT304 Computer Networks

8/22/23

Exercise:

1.2:



```
  607 4.478195    10.200.4.38       184.27.122.32     HTTP      208 GET /connecttest.txt HTTP/1.1
  625 4.569237    184.27.122.32     10.200.4.38       HTTP      301 HTTP/1.1 200 OK  (text/plain)
  772 5.464484    10.200.4.38       128.119.245.12    HTTP      525 GET /wireshark-labs/TCP-wireshark-file1.html HTTP/1.1
  881 6.032755    128.119.245.12    10.200.4.38       HTTP      826 HTTP/1.1 200 OK  (text/html)
 3531 20.896432   10.200.4.38       128.119.245.12    HTTP     3453 POST /wireshark-labs/lab3-1-reply.htm HTTP/1.1  (text/plain)
 3841 23.419109   128.119.245.12    10.200.4.38       HTTP      869 HTTP/1.1 200 OK  (text/html)

> Frame 3531: 3453 bytes on wire (27624 bits), 3453 bytes captured (27624 bits) on interface \Device\NPF_{263041D1-1250-4946-A523-CC9D5E7D566F}, id 0
> Ethernet II, Src: IntelCor_5c:9b:4e (08:6a:c5:5c:9b:4e), Dst: Cisco_ee:6a:29 (00:f2:8b:ee:6a:29)
> Internet Protocol Version 4, Src: 10.200.4.38, Dst: 128.119.245.12
v Transmission Control Protocol, Src Port: 55030, Dst Port: 80, Seq: 149628, Ack: 1, Len: 3399
     Source Port: 55030
     Destination Port: 80
     [Stream index: 63]
     [Conversation completeness: Complete, WITH_DATA (31)]
     [TCP Segment Len: 3399]
     Sequence Number: 149628    (relative sequence number)
     Sequence Number (raw): 3338365060
     [Next Sequence Number: 153027    (relative sequence number)]
     Acknowledgment Number: 1    (relative ack number)
     Acknowledgment number (raw): 395654308
     0101 .... = Header Length: 20 bytes (5)
   > Flags: 0x018 (PSH, ACK)
     Window: 256
     [Calculated window size: 65536]
     [Window size scaling factor: 256]
     Checksum: 0x8478 [unverified]
     [Checksum Status: Unverified]
     Urgent Pointer: 0
   > [Timestamps]
   > [SEQ/ACK analysis]
     TCP payload (3399 bytes)
     TCP segment data (3399 bytes)
```

Answer below is based on this image reference .

1.What is the IP address and TCP port number used by the client computer (source) that is transferring the file to gaia.cs.umass.edu? To answer this question, it’s probably easiest to select an HTTP message and explore the details of the TCP packet used to carry this HTTP message, using the “details of the selected packet header window”.

IP address of client = 10.200.4.38

Port number of client = 55030

2. What is the IP address of gaia.cs.umass.edu? On what port number is it sending and receiving TCP segments for this connection?

IP address of server = 128.119.245.12

Port number of server = 80

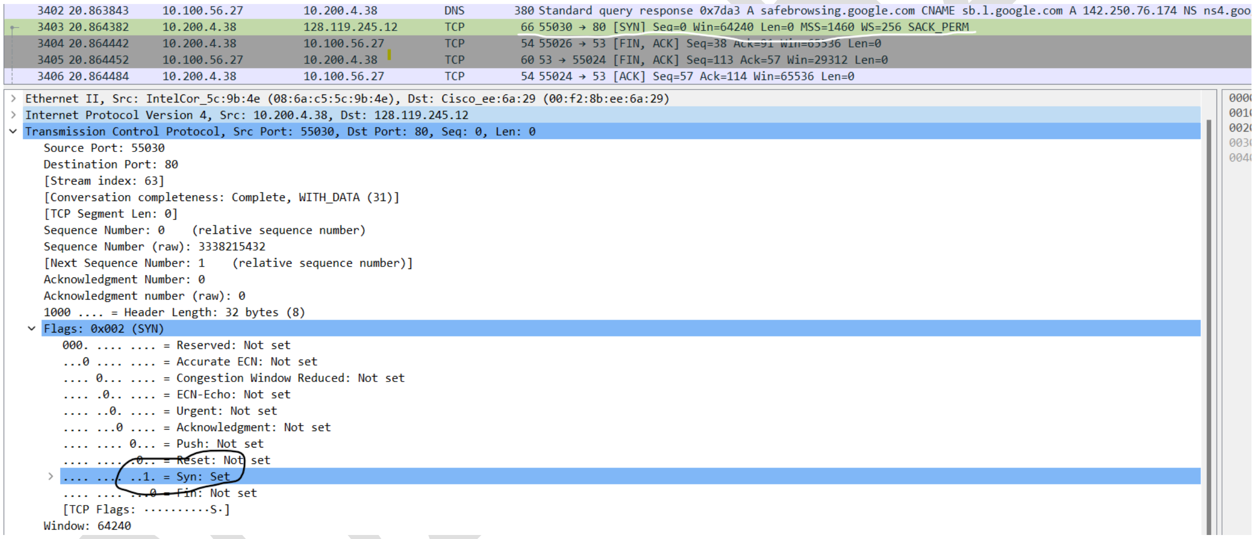
3. What is the IP address and TCP port number used by your client computer (source) to transfer the file to gaia.cs.umass.edu?

Same as Q1 IP address and port number is same that is:

IP address of client = 10.200.4.38 and Port number of client = 55030

2.2:

1. What is the sequence number of the TCP SYN segment that is used to initiate the TCP connection between the client computer and gaia.cs.umass.edu? What is it in the segment that identifies the segment as a SYN segment?



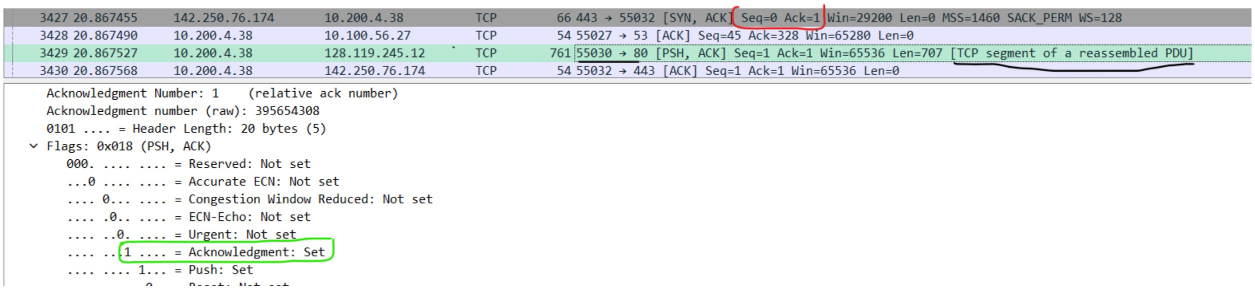
```
3402 20.863843   10.100.56.27    10.200.4.38      DNS    380 Standard query response 0x7da3 A safebrowsing.google.com CNAME sb.l.google.com A 142.250.76.174 NS ns4.goo
3403 20.864382   10.200.4.38     128.119.245.12   TCP     66 55030 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
3404 20.864442   10.200.4.38     10.100.56.27     TCP     54 55026 → 53 [FIN, ACK] Seq=38 Ack=91 Win=65536 Len=0
3405 20.864452   10.100.56.27    10.200.4.38      TCP     60 53 → 55024 [FIN, ACK] Seq=113 Ack=57 Win=29312 Len=0
3406 20.864484   10.200.4.38     10.100.56.27     TCP     54 55024 → 53 [ACK] Seq=57 Ack=114 Win=65536 Len=0

> Ethernet II, Src: IntelCor_5c:9b:4e (08:6a:c5:5c:9b:4e), Dst: Cisco_ee:6a:29 (00:f2:8b:ee:6a:29)
> Internet Protocol Version 4, Src: 10.200.4.38, Dst: 128.119.245.12
v Transmission Control Protocol, Src Port: 55030, Dst Port: 80, Seq: 0, Len: 0
    Source Port: 55030
    Destination Port: 80
    [Stream index: 63]
    [Conversation completeness: Complete, WITH_DATA (31)]
    [TCP Segment Len: 0]
    Sequence Number: 0    (relative sequence number)
    Sequence Number (raw): 3338215432
    [Next Sequence Number: 1    (relative sequence number)]
    Acknowledgment Number: 0
    Acknowledgment number (raw): 0
    1000 .... = Header Length: 32 bytes (8)
  v Flags: 0x002 (SYN)
      000. .... .... = Reserved: Not set
      ...0 .... .... = Accurate ECN: Not set
      .... 0... .... = Congestion Window Reduced: Not set
      .... .0.. .... = ECN-Echo: Not set
      .... ..0. .... = Urgent: Not set
      .... ...0 .... = Acknowledgment: Not set
      .... .... 0... = Push: Not set
      .... .... .0.. = Reset: Not set
    > .... .... ..1. = Syn: Set
      .... .... ...0 = Fin: Not set
      [TCP Flags: ··········S·]
    Window: 64240
```

Sequence number of TCP SYN: 0

SYN segment which is flag as: 1 as Set

2. What is the sequence number of the SYNACK segment sent by gaia.cs.umass.edu to the client computer in reply to the SYN? What is the value of the Acknowledgement field in the SYNACK segment? How did gaia.cs.umass.edu determine that value? What is it in the segment that identifies the segment as a SYNACK segment?



```
3427 20.867455   142.250.76.174   10.200.4.38      TCP   66 443 → 55032 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM WS=128
3428 20.867490   10.200.4.38      10.100.56.27     TCP   54 55027 → 53 [ACK] Seq=45 Ack=328 Win=65280 Len=0
3429 20.867527   10.200.4.38      128.119.245.12   TCP  761 55030 → 80 [PSH, ACK] Seq=1 Ack=1 Win=65536 Len=707 [TCP segment of a reassembled PDU]
3430 20.867568   10.200.4.38      142.250.76.174   TCP   54 55032 → 443 [ACK] Seq=1 Ack=1 Win=65536 Len=0

    Acknowledgment Number: 1    (relative ack number)
    Acknowledgment number (raw): 395654308
    0101 .... = Header Length: 20 bytes (5)
  ⌄ Flags: 0x018 (PSH, ACK)
      000. .... .... = Reserved: Not set
      ...0 .... .... = Accurate ECN: Not set
      .... 0... .... = Congestion Window Reduced: Not set
      .... .0.. .... = ECN-Echo: Not set
      .... ..0. .... = Urgent: Not set
      .... ...1 .... = Acknowledgment: Set
      .... .... 1... = Push: Set
```

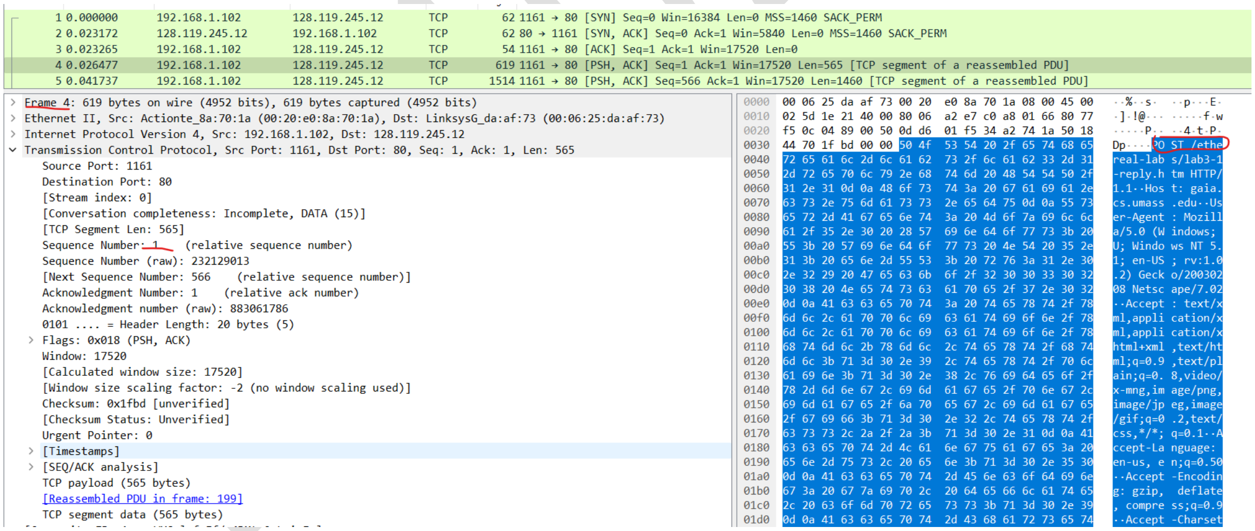
Seq Number of SYNACK in reply to SYN: 0

Value of Acknowledgement of SYNACK: 1

The site determine value by Seq number SYN segment + 1, which is 0 + 1

Here TCP is SYNACK as Acknowledgement is set to 1.

3. What is the sequence number of the TCP segment containing the HTTP POST command? Note that in order to find the POST command, you'll need to dig into the packet content field at the bottom of the Wireshark window, looking for a segment with a "POST" within its DATA field.



```
1 0.000000   192.168.1.102   128.119.245.12   TCP    62 1161 → 80 [SYN] Seq=0 Win=16384 Len=0 MSS=1460 SACK_PERM
2 0.023172   128.119.245.12  192.168.1.102    TCP    62 80 → 1161 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460 SACK_PERM
3 0.023265   192.168.1.102   128.119.245.12   TCP    54 1161 → 80 [ACK] Seq=1 Ack=1 Win=17520 Len=0
4 0.026477   192.168.1.102   128.119.245.12   TCP   619 1161 → 80 [PSH, ACK] Seq=1 Ack=1 Win=17520 Len=565 [TCP segment of a reassembled PDU]
5 0.041737   192.168.1.102   128.119.245.12   TCP  1514 1161 → 80 [PSH, ACK] Seq=566 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]
```

```
> Frame 4: 619 bytes on wire (4952 bits), 619 bytes captured (4952 bits)
> Ethernet II, Src: Actionte_8a:70:1a (00:20:e0:8a:70:1a), Dst: LinksysG_da:af:73 (00:06:25:da:af:73)
> Internet Protocol Version 4, Src: 192.168.1.102, Dst: 128.119.245.12
⌄ Transmission Control Protocol, Src Port: 1161, Dst Port: 80, Seq: 1, Ack: 1, Len: 565
    Source Port: 1161
    Destination Port: 80
    [Stream index: 0]
    [Conversation completeness: Incomplete, DATA (15)]
    [TCP Segment Len: 565]
    Sequence Number: 1    (relative sequence number)
    Sequence Number (raw): 232129013
    [Next Sequence Number: 566    (relative sequence number)]
    Acknowledgment Number: 1    (relative ack number)
    Acknowledgment number (raw): 883061786
    0101 .... = Header Length: 20 bytes (5)
  > Flags: 0x018 (PSH, ACK)
    Window: 17520
    [Calculated window size: 17520]
    [Window size scaling factor: -2 (no window scaling used)]
    Checksum: 0x1fbd [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0
  > [Timestamps]
  > [SEQ/ACK analysis]
    TCP payload (565 bytes)
    [Reassembled PDU in frame: 199]
    TCP segment data (565 bytes)
```

```
0000  00 06 25 da af 73 00 20  e0 8a 70 1a 08 00 45 00   ··%··s·  ··p···E·
0010  02 5d 1e 21 40 00 80 06  a2 e7 c0 a8 01 66 80 77   ·]·!@···  ·····f·w
0020  f5 0c 04 89 00 50 0d d6  01 f5 34 a2 74 1a 50 18   ·····P··  ··4·t·P·
0030  44 70 1f bd 00 00 50 4f  53 54 20 2f 65 74 68 65   Dp····PO ST /ethe
0040  72 65 61 6c 2d 6c 61 62  73 2f 6c 61 62 33 2d 31   real-lab s/lab3-1
0050  2d 72 65 70 6c 79 2e 68  74 6d 20 48 54 54 50 2f   -reply.h tm HTTP/
0060  31 2e 31 0d 0a 48 6f 73  74 3a 20 67 61 69 61 2e   1.1··Hos t: gaia.
0070  63 73 2e 75 6d 61 73 73  2e 65 64 75 0d 0a 55 73   cs.umass .edu··Us
0080  65 72 2d 41 67 65 6e 74  3a 20 4d 6f 7a 69 6c 6c   er-Agent : Mozill
0090  61 2f 35 2e 30 20 28 57  69 6e 64 6f 77 73 3b 20   a/5.0 (W indows;
00a0  55 3b 20 57 69 6e 64 6f  77 73 20 4e 54 20 35 2e   U; Windo ws NT 5.
00b0  31 3b 20 65 6e 2d 55 53  3b 20 72 76 3a 31 2e 30   1; en-US ; rv:1.0
00c0  2e 32 29 20 47 65 63 6b  6f 2f 32 30 30 33 30 32   .2) Geck o/200302
00d0  30 38 20 4e 65 74 73 63  61 70 65 2f 37 2e 30 32   08 Netsc ape/7.02
00e0  0d 0a 41 63 63 65 70 74  3a 20 74 65 78 74 2f 78   ··Accept : text/x
00f0  6d 6c 2c 61 70 70 6c 69  63 61 74 69 6f 6e 2f 78   ml,appli cation/x
0100  6d 6c 2c 61 70 70 6c 69  63 61 74 69 6f 6e 2f 78   ml,appli cation/x
0110  68 74 6d 6c 2b 78 6d 6c  2c 74 65 78 74 2f 68 74   html+xml ,text/ht
0120  6d 6c 3b 71 3d 30 2e 39  2c 74 65 78 74 2f 70 6c   ml;q=0.9 ,text/pl
0130  61 69 6e 3b 71 3d 30 2e  38 2c 76 69 64 65 6f 2f   ain;q=0. 8,video/
0140  78 2d 6d 6e 67 2c 69 6d  61 67 65 2f 70 6e 67 2c   x-mng,im age/png,
0150  69 6d 61 67 65 2f 6a 70  65 67 2c 69 6d 61 67 65   image/jp eg,image
0160  2f 67 69 66 3b 71 3d 30  2e 32 2c 74 65 78 74 2f   /gif;q=0 .2,text/
0170  63 73 73 2c 2a 2f 2a 3b  71 3d 30 2e 31 0d 0a 41   css,*/*; q=0.1··A
0180  63 63 65 70 74 2d 4c 61  6e 67 75 61 67 65 3a 20   ccept-La nguage:
0190  65 6e 2d 75 73 2c 20 65  6e 3b 71 3d 30 2e 35 30   en-us, e n;q=0.50
01a0  0d 0a 41 63 63 65 70 74  2d 45 6e 63 6f 64 69 6e   ··Accept -Encodin
01b0  67 3a 20 67 7a 69 70 2c  20 64 65 66 6c 61 74 65   g: gzip,  deflate
01c0  2c 20 63 6f 6d 70 72 65  73 73 3b 71 3d 30 2e 39   , compre ss;q=0.9
01d0  0d 0a 41 63 63 65 70 74  2d 43 68 61 72 73 65 74   ··Accept -Charset
```

Seq Number is 1. Which is relative and the frame no is 4.

4. Consider the TCP segment containing the HTTP POST as the first segment in the TCP connection. What are the sequence numbers of the first six segments in the TCP connection (including the segment containing the HTTP POST)? At what time was each segment sent? When was the ACK for each segment received? Given

the difference between when each TCP segment was sent, and when its acknowledgement was received, what is the RTT value for each of the six segments? What is the Estimated RTT value after the receipt of each ACK? Assume that the value of the Estimated RTT is equal to the measured RTT for the first segment.

Six Segment:

Segment 1 Seq no. is :1

Segment 2 Seq no. is :2026

Segment 3 Seq no. is :3486

Segment 4 Seq no. is :4946

Segment 5 Seq no. is :6406

Segment 6 Seq no. is :7866

Time:

Segment 1: 0.077924 s

Segment 2: 0.077924 s

Segment 3: 0.088034 s

Segment 4: 0.089023 s

Segment 5: 0.169105 s

Segment 6: 0.267801 s

Ack for each received :

Segment 1: 0.054026000 seconds

Segment 2: 0.054027000 seconds

Segment 3: 0.054027000 seconds

Segment 4: 0.054127100 seconds

Segment 5: 0.054128600 seconds

Segment 6: 0.054128600 seconds

RTT value :

Segment 1: 0.023265000 s

Segment 2: 0.023265000 s

Segment 3: 0.023265000 s

Segment 4: 0.023265000 s

Segment 5: 0.023266000 s

Segment 6: 0.023266000 s

5. What is the length of each of the first six TCP segments?

Length of the TCP segment is given as 707 bytes.

Length of each TCP segment: 1460 bytes (MSS).

6. What is the minimum amount of available buffer space advertised at the received for the entire trace? Does the lack of receiver buffer space ever throttle the sender?

The amount available is 29200 bytes for server.

7. Are there any retransmitted segments in the trace file? What did you check for (in the trace) in order to answer this question?

No retransmitted signal is found with any such label.

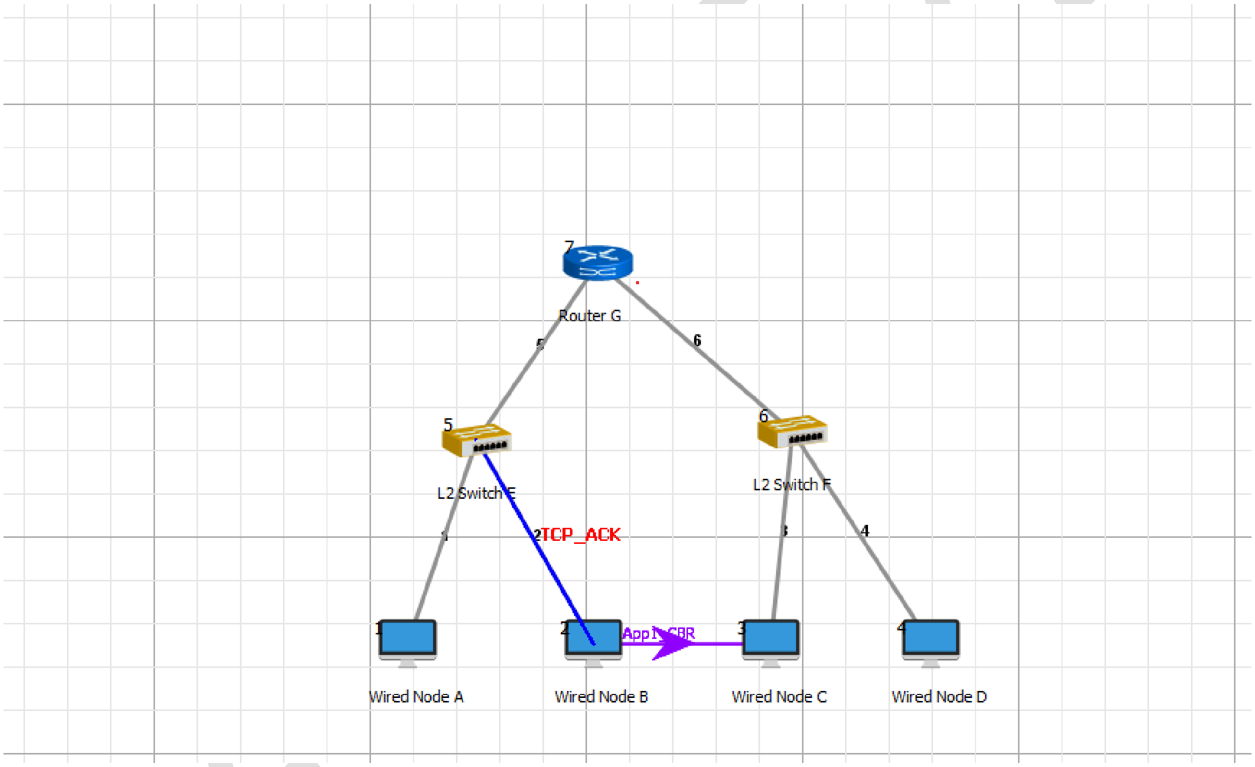
8. How much data does the receiver typically acknowledge in an ACK? Can you identify cases where the receiver is ACKing every other received segment.

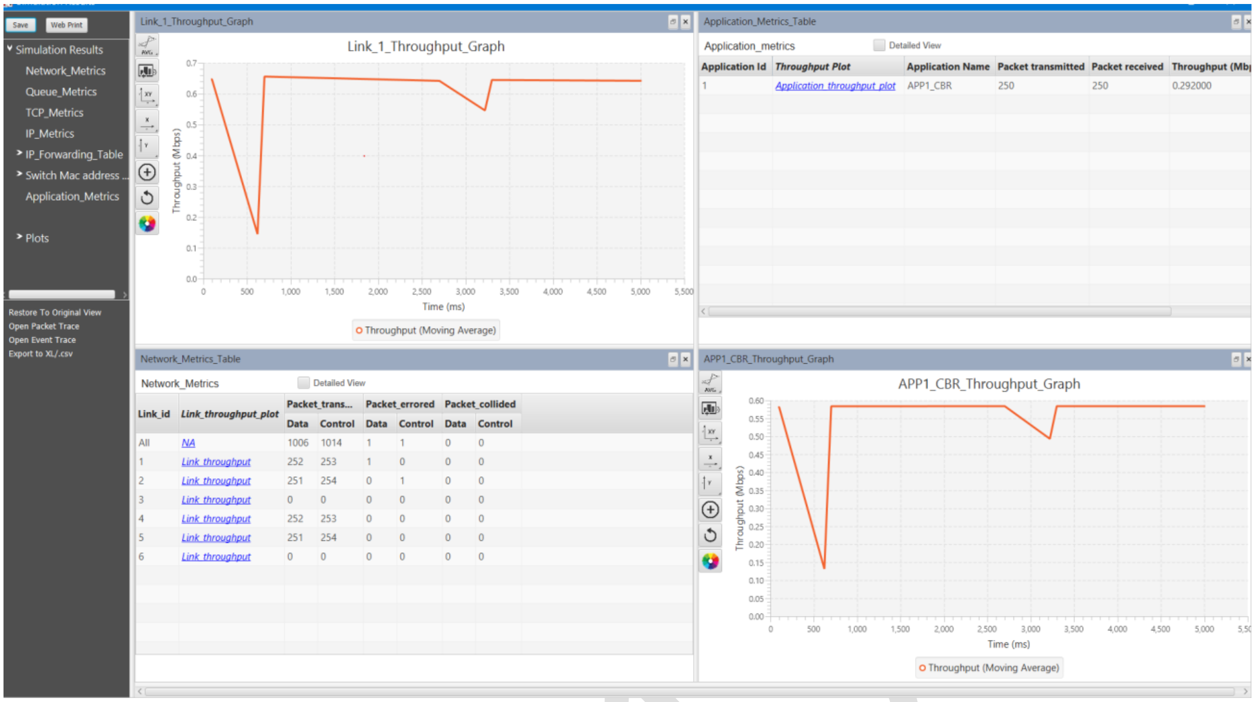
In segment 60 data is acknowledge of 1460 bytes is acknowledge in ACK. Yes receiver is ACKing every other segment.

9. What is the throughput (bytes transferred per unit time) for the TCP connection? Explain how you calculated this value.

It is calculated as (transferred bytes)/(required time- Delta of time) = 160600 / 0.00481 ~ 3.3Mbps

5.3:



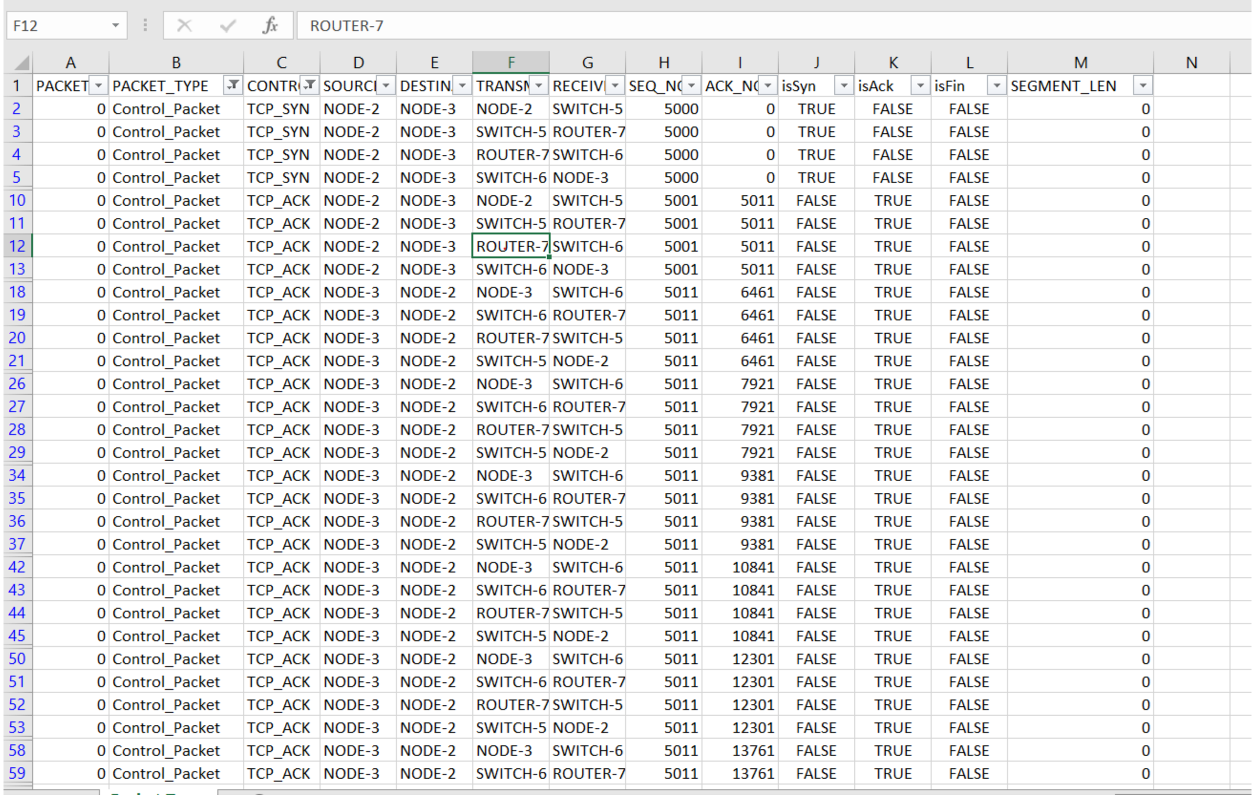


Application_metrics ☐ Detailed View

Application Id	Throughput Plot	Application Name	Packet transmitted	Packet received	Throughput (Mb
1	Application_throughput_plot	APP1_CBR	250	250	0.292000

Network_Metrics ☐ Detailed View

Link_id	Link_throughput_plot	Packet_trans...		Packet_errored		Packet_collided	
		Data	Control	Data	Control	Data	Control
All	NA	1006	1014	1	1	0	0
1	Link_throughput	252	253	1	0	0	0
2	Link_throughput	251	254	0	1	0	0
3	Link_throughput	0	0	0	0	0	0
4	Link_throughput	252	253	0	0	0	0
5	Link_throughput	251	254	0	0	0	0
6	Link_throughput	0	0	0	0	0	0



ROUTER-7

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	PACKET	PACKET_TYPE	CONTR	SOURC	DESTIN	TRANSN	RECEIV	SEQ_NC	ACK_NC	isSyn	isAck	isFin	SEGMENT_LEN	
2	0	Control_Packet	TCP_SYN	NODE-2	NODE-3	NODE-2	SWITCH-5	5000	0	TRUE	FALSE	FALSE	0	
3	0	Control_Packet	TCP_SYN	NODE-2	NODE-3	SWITCH-5	ROUTER-7	5000	0	TRUE	FALSE	FALSE	0	
4	0	Control_Packet	TCP_SYN	NODE-2	NODE-3	ROUTER-7	SWITCH-6	5000	0	TRUE	FALSE	FALSE	0	
5	0	Control_Packet	TCP_SYN	NODE-2	NODE-3	SWITCH-6	NODE-3	5000	0	TRUE	FALSE	FALSE	0	
10	0	Control_Packet	TCP_ACK	NODE-2	NODE-3	NODE-2	SWITCH-5	5001	5011	FALSE	TRUE	FALSE	0	
11	0	Control_Packet	TCP_ACK	NODE-2	NODE-3	SWITCH-5	ROUTER-7	5001	5011	FALSE	TRUE	FALSE	0	
12	0	Control_Packet	TCP_ACK	NODE-2	NODE-3	ROUTER-7	SWITCH-6	5001	5011	FALSE	TRUE	FALSE	0	
13	0	Control_Packet	TCP_ACK	NODE-2	NODE-3	SWITCH-6	NODE-3	5001	5011	FALSE	TRUE	FALSE	0	
18	0	Control_Packet	TCP_ACK	NODE-3	NODE-2	NODE-3	SWITCH-6	5011	6461	FALSE	TRUE	FALSE	0	
19	0	Control_Packet	TCP_ACK	NODE-3	NODE-2	SWITCH-6	ROUTER-7	5011	6461	FALSE	TRUE	FALSE	0	
20	0	Control_Packet	TCP_ACK	NODE-3	NODE-2	ROUTER-7	SWITCH-5	5011	6461	FALSE	TRUE	FALSE	0	
21	0	Control_Packet	TCP_ACK	NODE-3	NODE-2	SWITCH-5	NODE-2	5011	6461	FALSE	TRUE	FALSE	0	
26	0	Control_Packet	TCP_ACK	NODE-3	NODE-2	NODE-3	SWITCH-6	5011	7921	FALSE	TRUE	FALSE	0	
27	0	Control_Packet	TCP_ACK	NODE-3	NODE-2	SWITCH-6	ROUTER-7	5011	7921	FALSE	TRUE	FALSE	0	
28	0	Control_Packet	TCP_ACK	NODE-3	NODE-2	ROUTER-7	SWITCH-5	5011	7921	FALSE	TRUE	FALSE	0	
29	0	Control_Packet	TCP_ACK	NODE-3	NODE-2	SWITCH-5	NODE-2	5011	7921	FALSE	TRUE	FALSE	0	
34	0	Control_Packet	TCP_ACK	NODE-3	NODE-2	NODE-3	SWITCH-6	5011	9381	FALSE	TRUE	FALSE	0	
35	0	Control_Packet	TCP_ACK	NODE-3	NODE-2	SWITCH-6	ROUTER-7	5011	9381	FALSE	TRUE	FALSE	0	
36	0	Control_Packet	TCP_ACK	NODE-3	NODE-2	ROUTER-7	SWITCH-5	5011	9381	FALSE	TRUE	FALSE	0	
37	0	Control_Packet	TCP_ACK	NODE-3	NODE-2	SWITCH-5	NODE-2	5011	9381	FALSE	TRUE	FALSE	0	
42	0	Control_Packet	TCP_ACK	NODE-3	NODE-2	NODE-3	SWITCH-6	5011	10841	FALSE	TRUE	FALSE	0	
43	0	Control_Packet	TCP_ACK	NODE-3	NODE-2	SWITCH-6	ROUTER-7	5011	10841	FALSE	TRUE	FALSE	0	
44	0	Control_Packet	TCP_ACK	NODE-3	NODE-2	ROUTER-7	SWITCH-5	5011	10841	FALSE	TRUE	FALSE	0	
45	0	Control_Packet	TCP_ACK	NODE-3	NODE-2	SWITCH-5	NODE-2	5011	10841	FALSE	TRUE	FALSE	0	
50	0	Control_Packet	TCP_ACK	NODE-3	NODE-2	NODE-3	SWITCH-6	5011	12301	FALSE	TRUE	FALSE	0	
51	0	Control_Packet	TCP_ACK	NODE-3	NODE-2	SWITCH-6	ROUTER-7	5011	12301	FALSE	TRUE	FALSE	0	
52	0	Control_Packet	TCP_ACK	NODE-3	NODE-2	ROUTER-7	SWITCH-5	5011	12301	FALSE	TRUE	FALSE	0	
53	0	Control_Packet	TCP_ACK	NODE-3	NODE-2	SWITCH-5	NODE-2	5011	12301	FALSE	TRUE	FALSE	0	
58	0	Control_Packet	TCP_ACK	NODE-3	NODE-2	NODE-3	SWITCH-6	5011	13761	FALSE	TRUE	FALSE	0	
59	0	Control_Packet	TCP_ACK	NODE-3	NODE-2	SWITCH-6	ROUTER-7	5011	13761	FALSE	TRUE	FALSE	0	

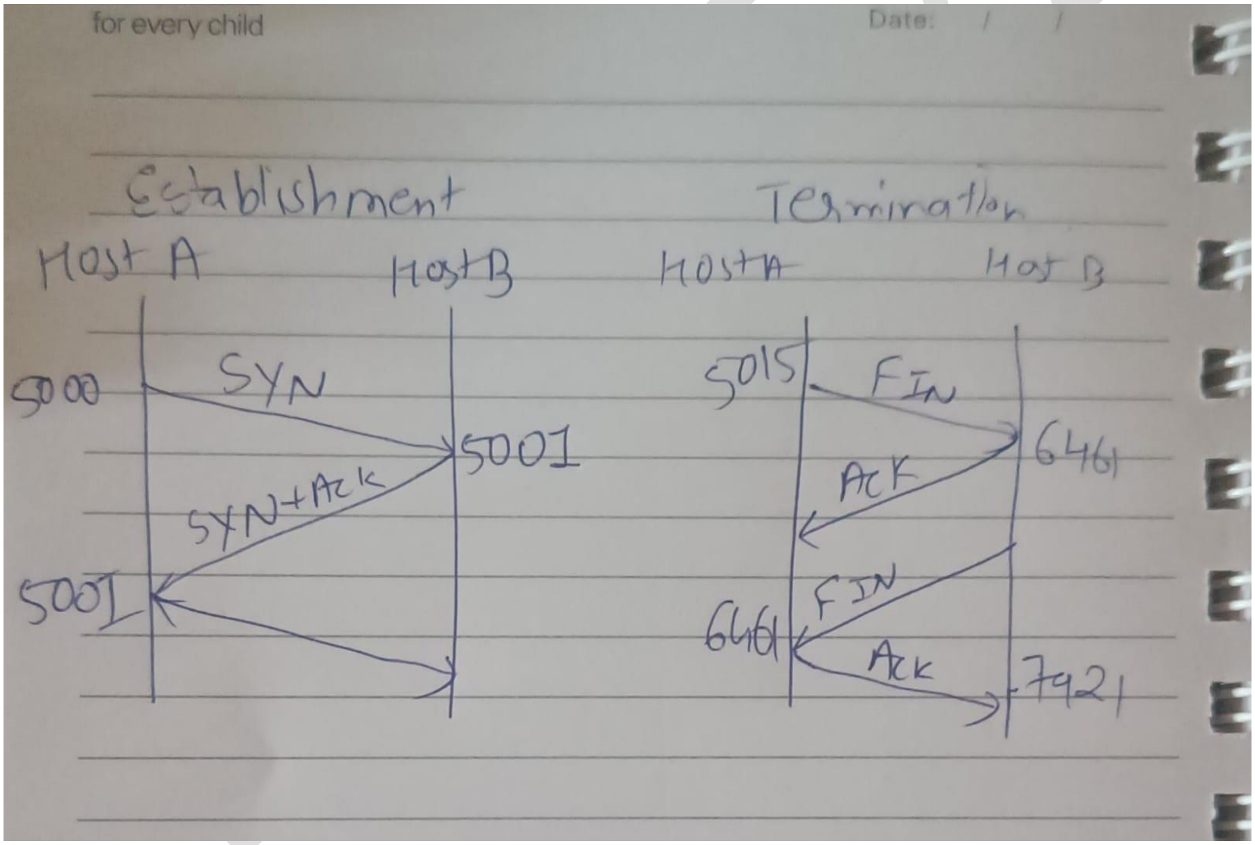
1.What is the Sequence number of the 1st SYN control packet and its acknowledgement?

It is 5000 and isAck is False

2. What is the sequence number of the 1st FIN control packet and its acknowledgement?

It is 5015 and isAck is False

3. Draw the Diagram of Connection establishment and termination as shown in figure 8 and 10 only with sequence number of each packet in you log book..



4. Why TCP uses 4 way finishing for connection termination instead of 3way like connection establishment?

In termination both sender and receiver may not want to end connection at same time so they need separate FIN-ACK while in establishment the sender make sure that receiver is ready to connect which can be done in just 3 ways which is justified.

5. How many sessions it takes to transfer all data in this application?

The TRUE value of is SYN is 16 times so in all, total of 4 session is taken.