Networks & Data Communications I

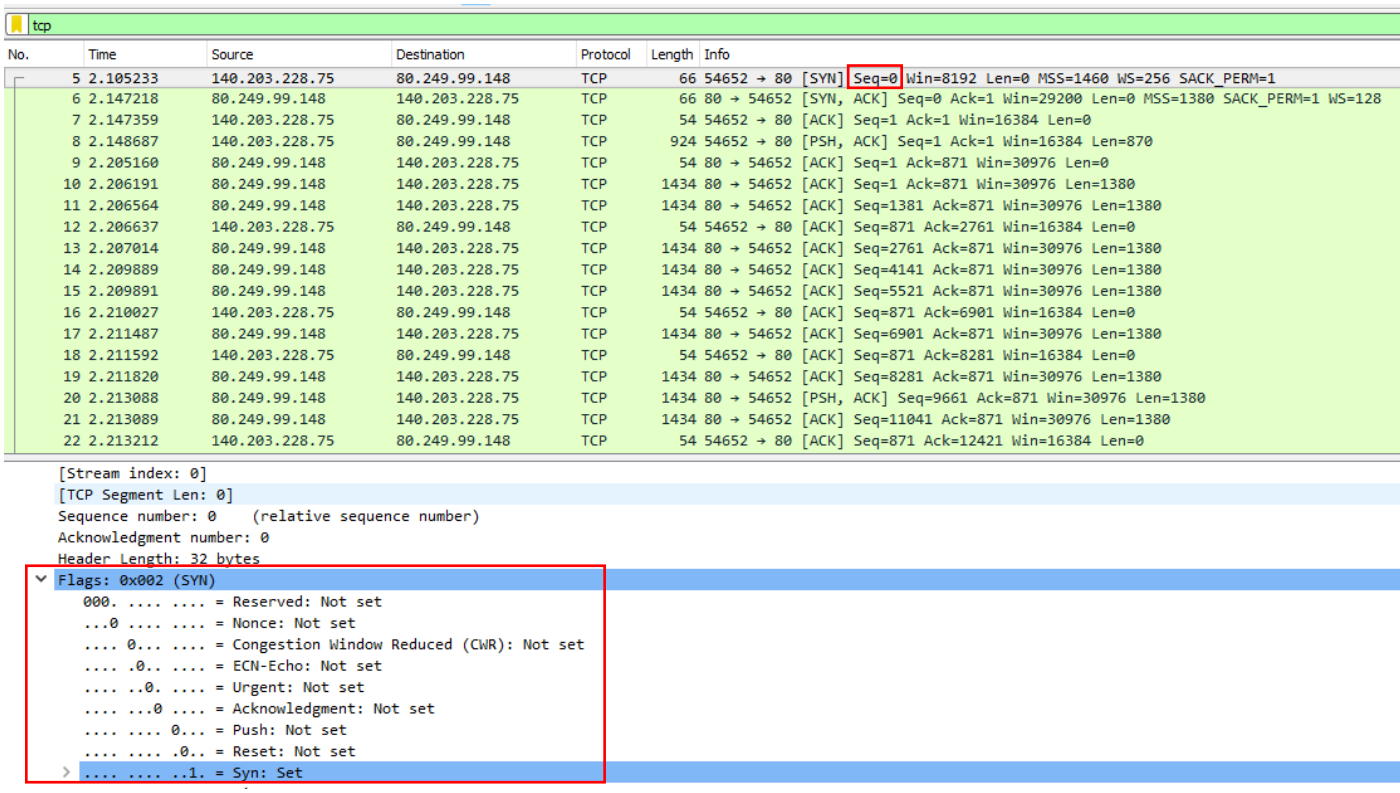
Assignment 1

TCP Protocol Analysis using Wireshark

- Cillian Fennell – 15407302 – C.FENNELL3@nuigalway.ie – 31/03/2017
- Andre Godinez – 15460718 – A.GODINEZ1@nuigalway.ie – 31/03/2017

Q1.

- The sequence number of the TCP SYN segment that was used to initiate a connection between my computer and the server is 0.
- The segment is identified as a SYN segment as its SYN flag has a value of 1.



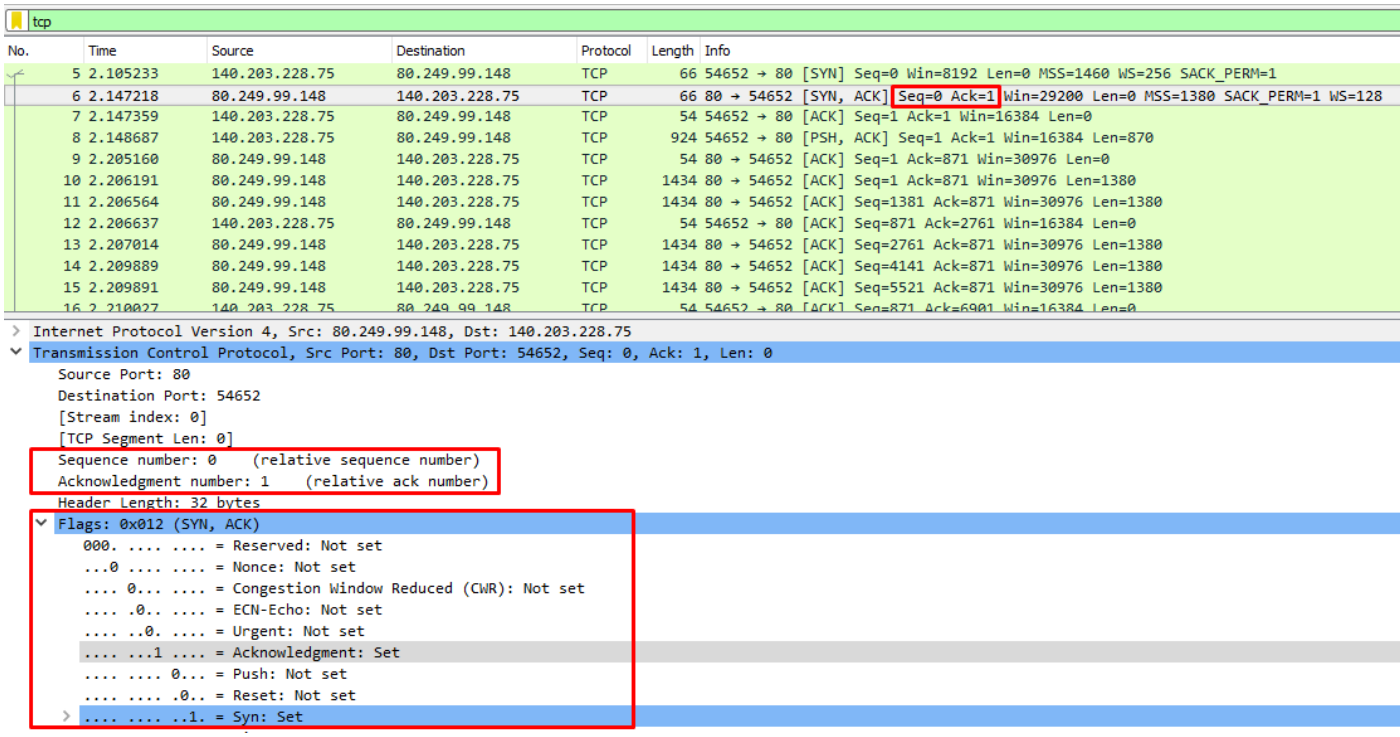
tcp

No.	Time	Source	Destination	Protocol	Length	Info
5	2.105233	140.203.228.75	80.249.99.148	TCP	66	54652 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
6	2.147218	80.249.99.148	140.203.228.75	TCP	66	80 → 54652 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1380 SACK_PERM=1 WS=128
7	2.147359	140.203.228.75	80.249.99.148	TCP	54	54652 → 80 [ACK] Seq=1 Ack=1 Win=16384 Len=0
8	2.148687	140.203.228.75	80.249.99.148	TCP	924	54652 → 80 [PSH, ACK] Seq=1 Ack=1 Win=16384 Len=870
9	2.205160	80.249.99.148	140.203.228.75	TCP	54	80 → 54652 [ACK] Seq=1 Ack=871 Win=30976 Len=0
10	2.206191	80.249.99.148	140.203.228.75	TCP	1434	80 → 54652 [ACK] Seq=1 Ack=871 Win=30976 Len=1380
11	2.206564	80.249.99.148	140.203.228.75	TCP	1434	80 → 54652 [ACK] Seq=1381 Ack=871 Win=30976 Len=1380
12	2.206637	140.203.228.75	80.249.99.148	TCP	54	54652 → 80 [ACK] Seq=871 Ack=2761 Win=16384 Len=0
13	2.207014	80.249.99.148	140.203.228.75	TCP	1434	80 → 54652 [ACK] Seq=2761 Ack=871 Win=30976 Len=1380
14	2.209889	80.249.99.148	140.203.228.75	TCP	1434	80 → 54652 [ACK] Seq=4141 Ack=871 Win=30976 Len=1380
15	2.209891	80.249.99.148	140.203.228.75	TCP	1434	80 → 54652 [ACK] Seq=5521 Ack=871 Win=30976 Len=1380
16	2.210027	140.203.228.75	80.249.99.148	TCP	54	54652 → 80 [ACK] Seq=871 Ack=6901 Win=16384 Len=0
17	2.211487	80.249.99.148	140.203.228.75	TCP	1434	80 → 54652 [ACK] Seq=6901 Ack=871 Win=30976 Len=1380
18	2.211592	140.203.228.75	80.249.99.148	TCP	54	54652 → 80 [ACK] Seq=871 Ack=8281 Win=16384 Len=0
19	2.211820	80.249.99.148	140.203.228.75	TCP	1434	80 → 54652 [ACK] Seq=8281 Ack=871 Win=30976 Len=1380
20	2.213088	80.249.99.148	140.203.228.75	TCP	1434	80 → 54652 [PSH, ACK] Seq=9661 Ack=871 Win=30976 Len=1380
21	2.213089	80.249.99.148	140.203.228.75	TCP	1434	80 → 54652 [ACK] Seq=11041 Ack=871 Win=30976 Len=1380
22	2.213212	140.203.228.75	80.249.99.148	TCP	54	54652 → 80 [ACK] Seq=871 Ack=12421 Win=16384 Len=0

```
[Stream index: 0]
[TCP Segment Len: 0]
Sequence number: 0    (relative sequence number)
Acknowledgment number: 0
Header Length: 32 bytes
Flags: 0x002 (SYN)
   000. .... .... = Reserved: Not set
   ...0 .... .... = Nonce: Not set
   .... 0... .... = Congestion Window Reduced (CWR): 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
```

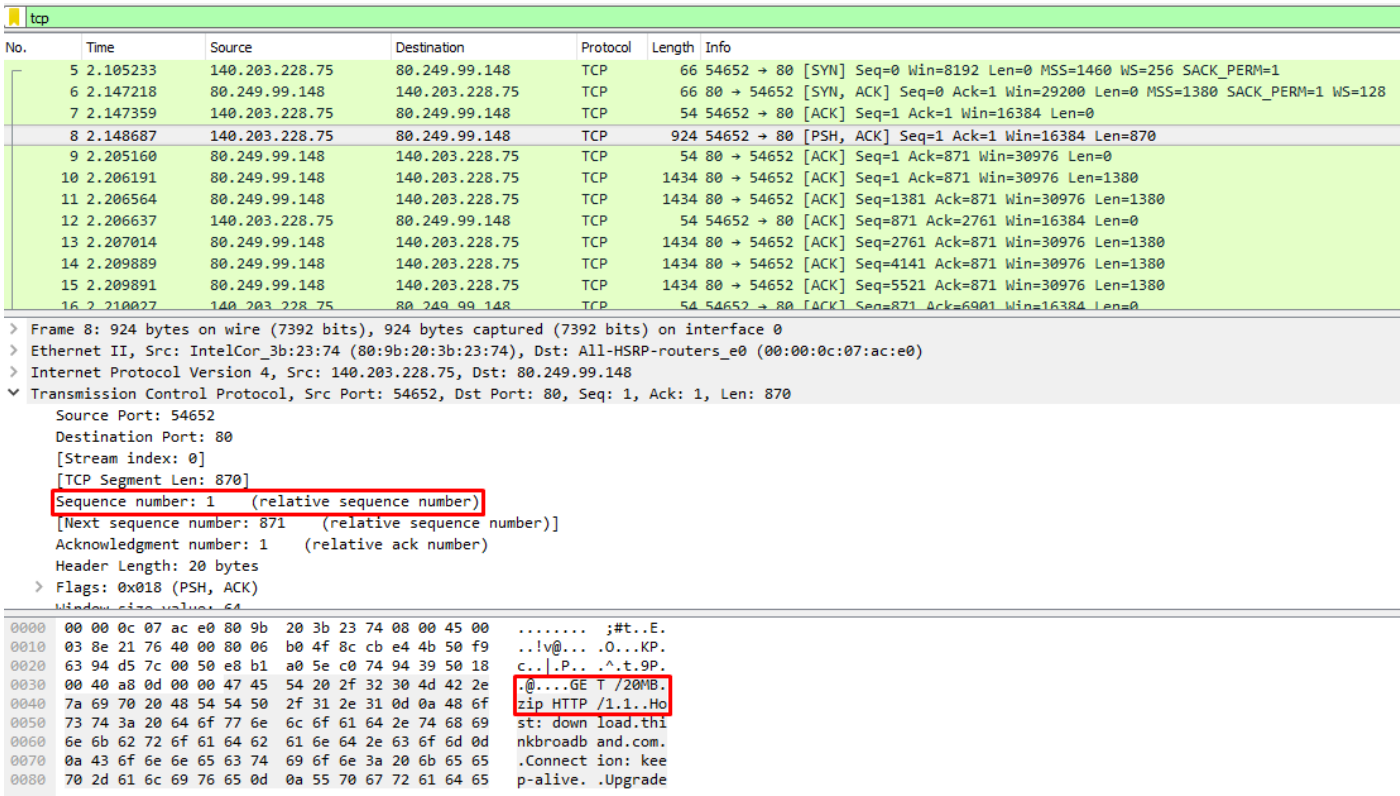
Q2.

- The sequence number of the SYN ACK segment sent by the server to my computer in reply to the SYN is 0.
- The value of the Acknowledgement field in the SYN ACK segment is 1.
- This value was found by the sequence number of the SYN segment from **Q1.** plus 1.
- This segment is identified as a SYN ACK segment as both its SYN flag and ACK flag have a value of 1.



Q3.

- The sequence number of the TCP segment containing the initial HTTP GET command is 1.



tcp

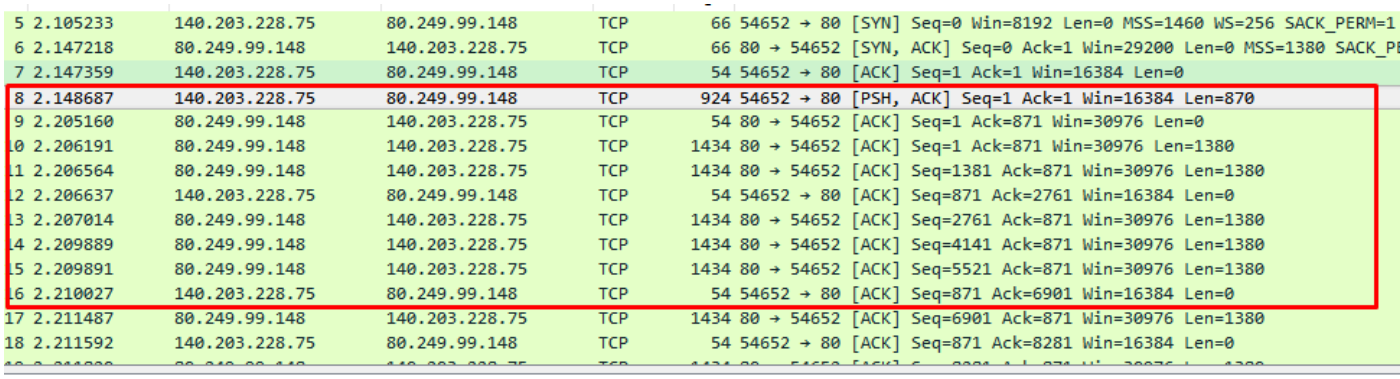
No.	Time	Source	Destination	Protocol	Length	Info
5	2.105233	140.203.228.75	80.249.99.148	TCP	66	54652 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
6	2.147218	80.249.99.148	140.203.228.75	TCP	66	80 → 54652 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1380 SACK_PERM=1 WS=128
7	2.147359	140.203.228.75	80.249.99.148	TCP	54	54652 → 80 [ACK] Seq=1 Ack=1 Win=16384 Len=0
8	2.148687	140.203.228.75	80.249.99.148	TCP	924	54652 → 80 [PSH, ACK] Seq=1 Ack=1 Win=16384 Len=870
9	2.205160	80.249.99.148	140.203.228.75	TCP	54	80 → 54652 [ACK] Seq=1 Ack=871 Win=30976 Len=0
10	2.206191	80.249.99.148	140.203.228.75	TCP	1434	80 → 54652 [ACK] Seq=1 Ack=871 Win=30976 Len=1380
11	2.206564	80.249.99.148	140.203.228.75	TCP	1434	80 → 54652 [ACK] Seq=1381 Ack=871 Win=30976 Len=1380
12	2.206637	140.203.228.75	80.249.99.148	TCP	54	54652 → 80 [ACK] Seq=871 Ack=2761 Win=16384 Len=0
13	2.207014	80.249.99.148	140.203.228.75	TCP	1434	80 → 54652 [ACK] Seq=2761 Ack=871 Win=30976 Len=1380
14	2.209889	80.249.99.148	140.203.228.75	TCP	1434	80 → 54652 [ACK] Seq=4141 Ack=871 Win=30976 Len=1380
15	2.209891	80.249.99.148	140.203.228.75	TCP	1434	80 → 54652 [ACK] Seq=5521 Ack=871 Win=30976 Len=1380
16	2.210027	140.203.228.75	80.249.99.148	TCP	54	54652 → 80 [ACK] Seq=871 Ack=6901 Win=16384 Len=0

```
> Frame 8: 924 bytes on wire (7392 bits), 924 bytes captured (7392 bits) on interface 0
> Ethernet II, Src: IntelCor_3b:23:74 (80:9b:20:3b:23:74), Dst: All-HSRP-routers_e0 (00:00:0c:07:ac:e0)
> Internet Protocol Version 4, Src: 140.203.228.75, Dst: 80.249.99.148
v Transmission Control Protocol, Src Port: 54652, Dst Port: 80, Seq: 1, Ack: 1, Len: 870
    Source Port: 54652
    Destination Port: 80
    [Stream index: 0]
    [TCP Segment Len: 870]
    Sequence number: 1    (relative sequence number)
    [Next sequence number: 871    (relative sequence number)]
    Acknowledgment number: 1    (relative ack number)
    Header Length: 20 bytes
  > Flags: 0x018 (PSH, ACK)
    Window size value: 64
```

```
0000  00 00 0c 07 ac e0 80 9b  20 3b 23 74 08 00 45 00   ........  ;#t..E.
0010  03 8e 21 76 40 00 80 06  b0 4f 8c cb e4 4b 50 f9   ..!v@... .O...KP.
0020  63 94 d5 7c 00 50 e8 b1  a0 5e c0 74 94 39 50 18   c..|.P.. .^.t.9P.
0030  00 40 a8 0d 00 00 47 45  54 20 2f 32 30 4d 42 2e   .@....GE T /20MB.
0040  7a 69 70 20 48 54 54 50  2f 31 2e 31 0d 0a 48 6f   zip HTTP /1.1..Ho
0050  73 74 3a 20 64 6f 77 6e  6c 6f 61 64 2e 74 68 69   st: down load.thi
0060  6e 6b 62 72 6f 61 64 62  61 6e 64 2e 63 6f 6d 0d   nkbroadb and.com.
0070  0a 43 6f 6e 6e 65 63 74  69 6f 6e 3a 20 6b 65 65   .Connect ion: kee
0080  70 2d 61 6c 69 76 65 0d  0a 55 70 67 72 61 64 65   p-alive. .Upgrade
```

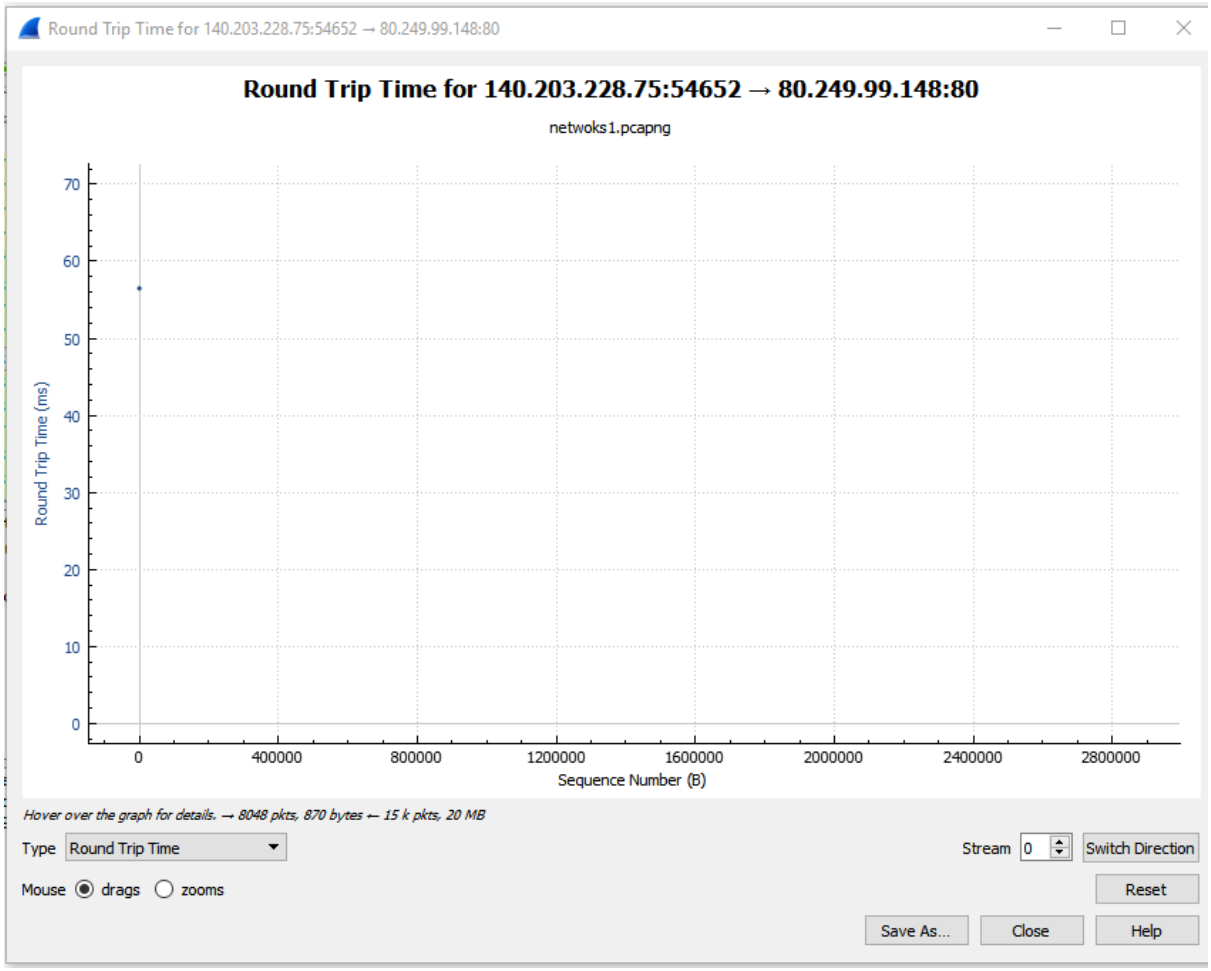
Q4.

- First 6 sequence numbers of the TCP connection are 1, 1, 1381, 2761, 4141 and 5521.
- Sent at 2.148687, 2.206191, 2.206564, 2.207014, 2.209889 and 2.209891 respectively.
- ACK segments received at 2.205160, 2.206564, 2.206637, 2.209889, 2.209891 and 2.210027 respectively.
- Note: I was confused by this question as I assumed each TCP segment would have a corresponding ACK segment directly after it was sent. Also the amount of TCP segments outweighs the ACK segments. After doing some research online I came to the assumption that data piggybacking is in action here where some ACK response are being sent with some TCP segments as to reduce the amount of segments needing to be sent. This is why some of my ACK segments are received at the same time some of the TCP segments are being sent (as seen above)



5	2.105233	140.203.228.75	80.249.99.148	TCP	66	54652 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
6	2.147218	80.249.99.148	140.203.228.75	TCP	66	80 → 54652 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1380 SACK_PE
7	2.147359	140.203.228.75	80.249.99.148	TCP	54	54652 → 80 [ACK] Seq=1 Ack=1 Win=16384 Len=0
8	2.148687	140.203.228.75	80.249.99.148	TCP	924	54652 → 80 [PSH, ACK] Seq=1 Ack=1 Win=16384 Len=870
9	2.205160	80.249.99.148	140.203.228.75	TCP	54	80 → 54652 [ACK] Seq=1 Ack=871 Win=30976 Len=0
10	2.206191	80.249.99.148	140.203.228.75	TCP	1434	80 → 54652 [ACK] Seq=1 Ack=871 Win=30976 Len=1380
11	2.206564	80.249.99.148	140.203.228.75	TCP	1434	80 → 54652 [ACK] Seq=1381 Ack=871 Win=30976 Len=1380
12	2.206637	140.203.228.75	80.249.99.148	TCP	54	54652 → 80 [ACK] Seq=871 Ack=2761 Win=16384 Len=0
13	2.207014	80.249.99.148	140.203.228.75	TCP	1434	80 → 54652 [ACK] Seq=2761 Ack=871 Win=30976 Len=1380
14	2.209889	80.249.99.148	140.203.228.75	TCP	1434	80 → 54652 [ACK] Seq=4141 Ack=871 Win=30976 Len=1380
15	2.209891	80.249.99.148	140.203.228.75	TCP	1434	80 → 54652 [ACK] Seq=5521 Ack=871 Win=30976 Len=1380
16	2.210027	140.203.228.75	80.249.99.148	TCP	54	54652 → 80 [ACK] Seq=871 Ack=6901 Win=16384 Len=0
17	2.211487	80.249.99.148	140.203.228.75	TCP	1434	80 → 54652 [ACK] Seq=6901 Ack=871 Win=30976 Len=1380
18	2.211592	140.203.228.75	80.249.99.148	TCP	54	54652 → 80 [ACK] Seq=871 Ack=8281 Win=16384 Len=0

Q5.



- The round trip time for this TCP segment is approximately 56ms

```
  7 2.147359     140.203.228.75     80.249.99.148      TCP       54 54652 → 80 [ACK] Seq=1 Ack=1 Win=16384 Len=0
  8 2.148687     140.203.228.75     80.249.99.148      TCP      924 54652 → 80 [PSH, ACK] Seq=1 Ack=1 Win=16384 Len=870
  9 2.205160     80.249.99.148      140.203.228.75     TCP       54 80 → 54652 [ACK] Seq=1 Ack=871 Win=30976 Len=0
 10 2.206191     80.249.99.148      140.203.228.75     TCP     1434 80 → 54652 [ACK] Seq=1 Ack=871 Win=30976 Len=1380
 11 2.206564     80.249.99.148      140.203.228.75     TCP     1434 80 → 54652 [ACK] Seq=1381 Ack=871 Win=30976 Len=1380
 12 2.206637     140.203.228.75     80.249.99.148      TCP       54 54652 → 80 [ACK] Seq=871 Ack=2761 Win=16384 Len=0
 13 2.207014     80.249.99.148      140.203.228.75     TCP     1434 80 → 54652 [ACK] Seq=2761 Ack=871 Win=30976 Len=1380
 14 2.209889     80.249.99.148      140.203.228.75     TCP     1434 80 → 54652 [ACK] Seq=4141 Ack=871 Win=30976 Len=1380
ame 9: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface 0
hernet II, Src: Cisco_fe:b4:40 (28:94:0f:fe:b4:40), Dst: IntelCor_3b:23:74 (80:9b:20:3b:23:74)
nternet Protocol Version 4, Src: 80.249.99.148, Dst: 140.203.228.75
ansmission Control Protocol, Src Port: 80, Dst Port: 54652, Seq: 1, Ack: 871, Len: 0
```

```
80 9b 20 3b 23 74 28 94  0f fe b4 40 08 00 45 a4   .. ;#t(. ...@..E.
00 28 a6 aa 40 00 38 06  75 dd 50 f9 63 94 8c cb   .(..@.8. u.P.c...
e4 4b 00 50 d5 7c c0 74  94 39 e8 b1 a3 c4 50 10   .K.P.|.t .9....P.
00 f2 d2 4c 00 00                                  ...L..
```

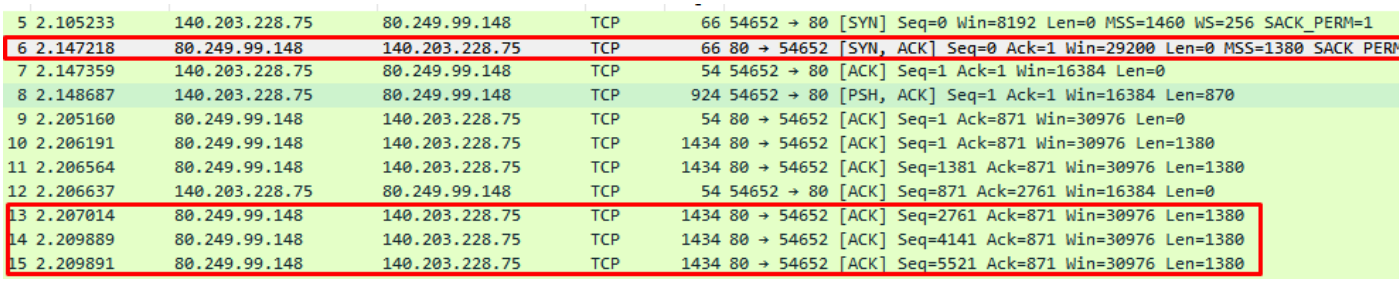
Q6.

- The length of the first segment is 870, the lengths of the following five segments are 1380.

5	2.105233	140.203.228.75	80.249.99.148	TCP	66 54652 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 S
6	2.147218	80.249.99.148	140.203.228.75	TCP	66 80 → 54652 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1
7	2.147359	140.203.228.75	80.249.99.148	TCP	54 54652 → 80 [ACK] Seq=1 Ack=1 Win=16384 Len=0
8	2.148687	140.203.228.75	80.249.99.148	TCP	924 54652 → 80 [PSH, ACK] Seq=1 Ack=1 Win=16384 Len=870
9	2.205160	80.249.99.148	140.203.228.75	TCP	54 80 → 54652 [ACK] Seq=1 Ack=871 Win=30976 Len=0
10	2.206191	80.249.99.148	140.203.228.75	TCP	1434 80 → 54652 [ACK] Seq=1 Ack=871 Win=30976 Len=1380
11	2.206564	80.249.99.148	140.203.228.75	TCP	1434 80 → 54652 [ACK] Seq=1381 Ack=871 Win=30976 Len=1380
12	2.206637	140.203.228.75	80.249.99.148	TCP	54 54652 → 80 [ACK] Seq=871 Ack=2761 Win=16384 Len=0
13	2.207014	80.249.99.148	140.203.228.75	TCP	1434 80 → 54652 [ACK] Seq=2761 Ack=871 Win=30976 Len=1380
14	2.209889	80.249.99.148	140.203.228.75	TCP	1434 80 → 54652 [ACK] Seq=4141 Ack=871 Win=30976 Len=1380
15	2.209891	80.249.99.148	140.203.228.75	TCP	1434 80 → 54652 [ACK] Seq=5521 Ack=871 Win=30976 Len=1380

Q7.

- The typical amount of available buffer space advertised at the receiver for the entire trace is 30976 bytes.

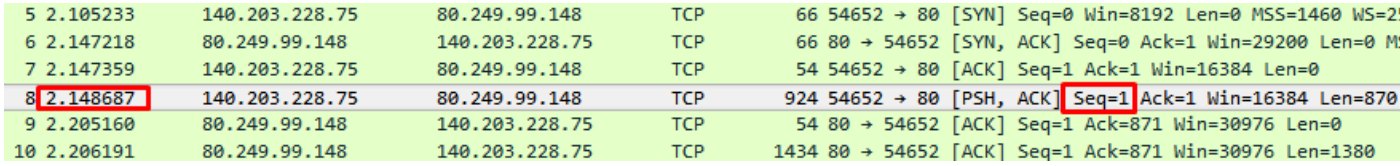


5	2.105233	140.203.228.75	80.249.99.148	TCP	66 54652 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
6	2.147218	80.249.99.148	140.203.228.75	TCP	66 80 → 54652 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1380 SACK_PERM
7	2.147359	140.203.228.75	80.249.99.148	TCP	54 54652 → 80 [ACK] Seq=1 Ack=1 Win=16384 Len=0
8	2.148687	140.203.228.75	80.249.99.148	TCP	924 54652 → 80 [PSH, ACK] Seq=1 Ack=1 Win=16384 Len=870
9	2.205160	80.249.99.148	140.203.228.75	TCP	54 80 → 54652 [ACK] Seq=1 Ack=871 Win=30976 Len=0
10	2.206191	80.249.99.148	140.203.228.75	TCP	1434 80 → 54652 [ACK] Seq=1 Ack=871 Win=30976 Len=1380
11	2.206564	80.249.99.148	140.203.228.75	TCP	1434 80 → 54652 [ACK] Seq=1381 Ack=871 Win=30976 Len=1380
12	2.206637	140.203.228.75	80.249.99.148	TCP	54 54652 → 80 [ACK] Seq=871 Ack=2761 Win=16384 Len=0
13	2.207014	80.249.99.148	140.203.228.75	TCP	1434 80 → 54652 [ACK] Seq=2761 Ack=871 Win=30976 Len=1380
14	2.209889	80.249.99.148	140.203.228.75	TCP	1434 80 → 54652 [ACK] Seq=4141 Ack=871 Win=30976 Len=1380
15	2.209891	80.249.99.148	140.203.228.75	TCP	1434 80 → 54652 [ACK] Seq=5521 Ack=871 Win=30976 Len=1380

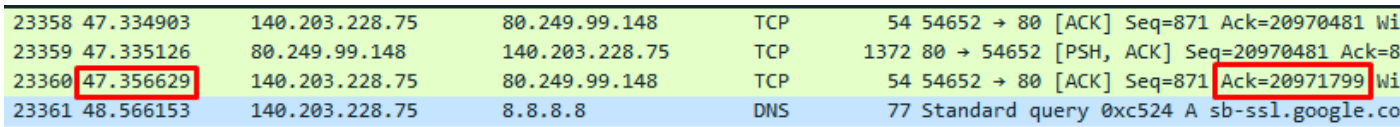
- As seen above the minimum amount of available buffer space advertised at the receiver is 29200 bytes, and also examples of the typical amount available, 30976 bytes.

Q8.

- Estimated throughput is 463 Kbytes/sec.
- The throughput is the data transmitted divided by the time taken for the data to be transmitted.
- The data transmitted is found by taking away the sequence number of the first TCP segment from the sequence number of the last ACK segment. In this case it is 20,971,799 – 1 = 20,971,798 bytes.
- The time taken is the time between the 2 respective segments. In this case it is 47.356629 – 2.148687 = 45.207942 seconds.
- Throughput = 20,971,798/45.207942 = 463,896.3215 bytes/sec, or, 463.896 Kbytes/seC.

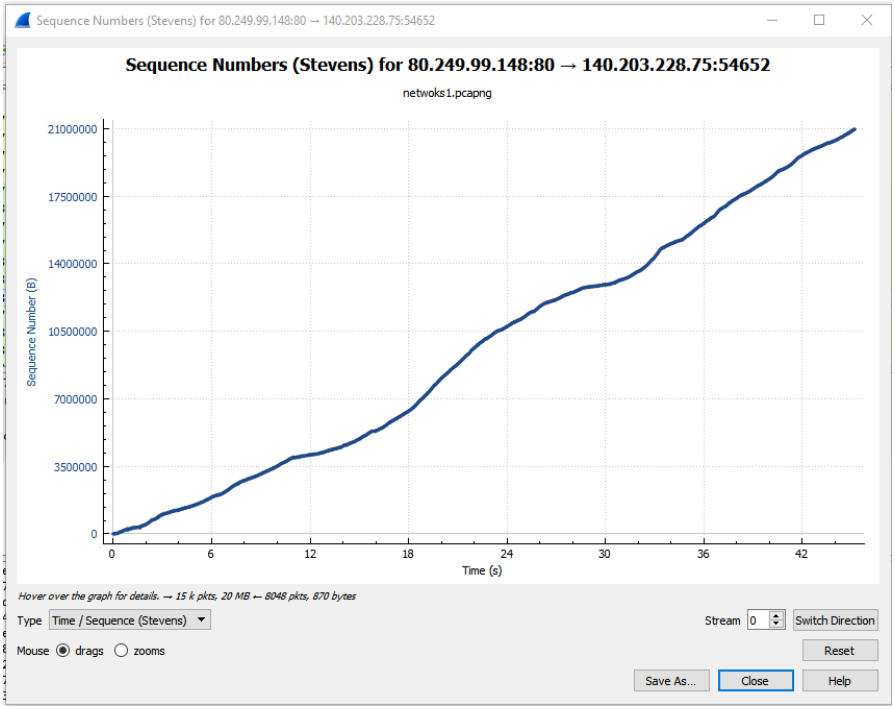


5	2.105233	140.203.228.75	80.249.99.148	TCP	66 54652 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=2
6	2.147218	80.249.99.148	140.203.228.75	TCP	66 80 → 54652 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 M
7	2.147359	140.203.228.75	80.249.99.148	TCP	54 54652 → 80 [ACK] Seq=1 Ack=1 Win=16384 Len=0
8	2.148687	140.203.228.75	80.249.99.148	TCP	924 54652 → 80 [PSH, ACK] Seq=1 Ack=1 Win=16384 Len=870
9	2.205160	80.249.99.148	140.203.228.75	TCP	54 80 → 54652 [ACK] Seq=1 Ack=871 Win=30976 Len=0
10	2.206191	80.249.99.148	140.203.228.75	TCP	1434 80 → 54652 [ACK] Seq=1 Ack=871 Win=30976 Len=1380

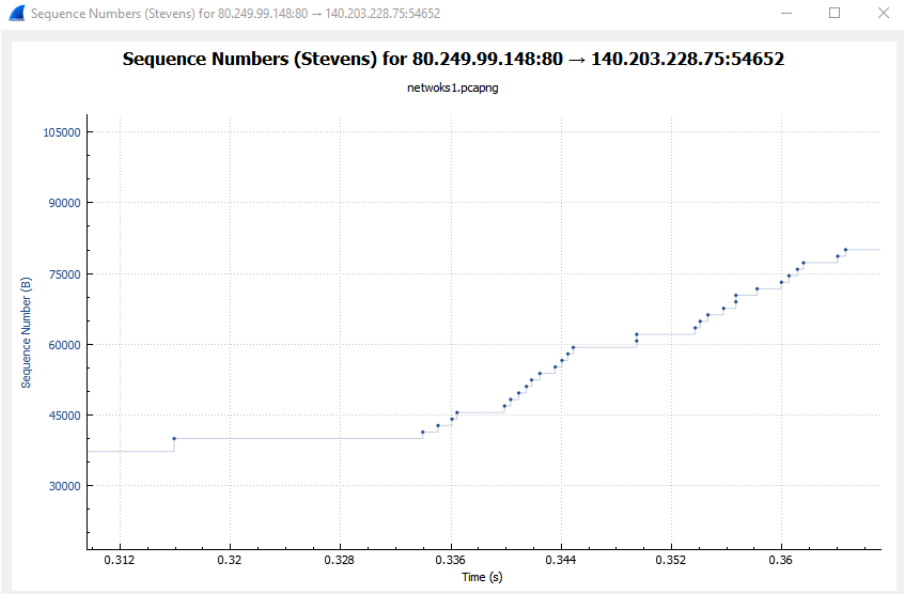


23358	47.334903	140.203.228.75	80.249.99.148	TCP	54 54652 → 80 [ACK] Seq=871 Ack=20970481 Wi
23359	47.335126	80.249.99.148	140.203.228.75	TCP	1372 80 → 54652 [PSH, ACK] Seq=20970481 Ack=8
23360	47.356629	140.203.228.75	80.249.99.148	TCP	54 54652 → 80 [ACK] Seq=871 Ack=20971799 Wi
23361	48.566153	140.203.228.75	8.8.8.8	DNS	77 Standard query 0xc524 A sb-ssl.google.co

Q9.



- The TCP slow start phase starts at the beginning of the connection



- Here the slow start ends at 0.336s and congestion control starts where packets start to be sent in groups

Q10.

- My results differ from the idealized behaviour of TCP as it only uses a fraction of the window size instead of the ideal value of around a half.
- The graph for congestion control is more uneven and gradual compared to the ideal vertical, steep graphs