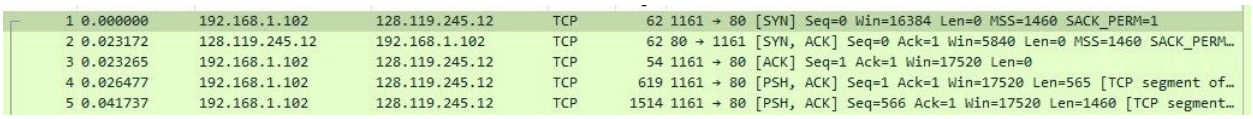
Lab Exercise 4: Exploring TCP

By John Dao z5258962

Exercise 1: Understanding TCP using Wireshark

Q1 . What is the IP address of gaia.cs.umass.edu? On what port number is it sending and receiving TCP segments for this connection? 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?

Sample Output

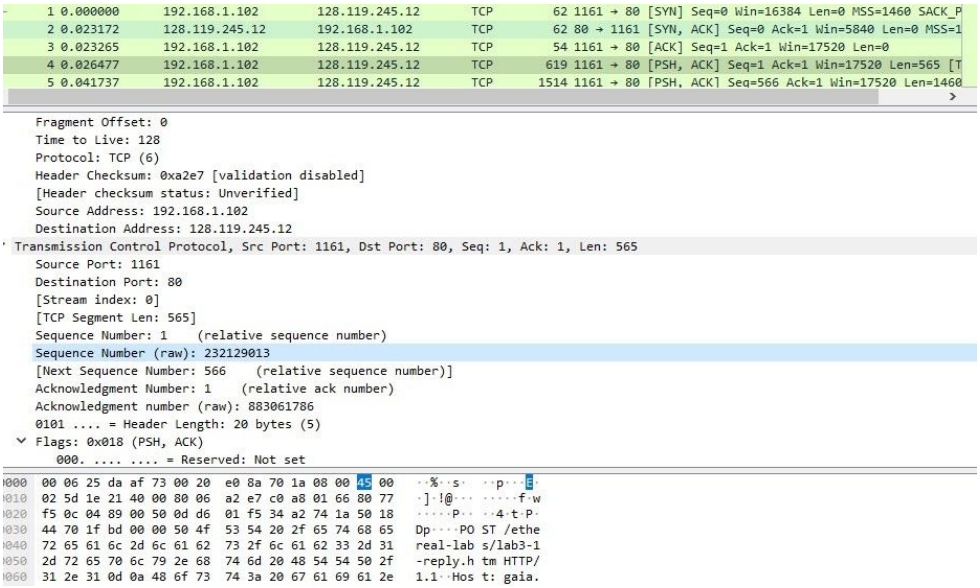


1	0.000000	192.168.1.102	128.119.245.12	TCP	62 1161 → 80 [SYN] Seq=0 Win=16384 Len=0 MSS=1460 SACK_PERM=1
2	0.023172	128.119.245.12	192.168.1.102	TCP	62 80 → 1161 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460 SACK_PERM…
3	0.023265	192.168.1.102	128.119.245.12	TCP	54 1161 → 80 [ACK] Seq=1 Ack=1 Win=17520 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…
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…

IP of the host is 192.168.0.102 using port 1161
IP of gaia.cs.umass.edu is 128.119.245.12 using port 80

Q2. 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.

Sample Output of post search for HTTP POST



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_P
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=1
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 [T
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

```
    Fragment Offset: 0
    Time to Live: 128
    Protocol: TCP (6)
    Header Checksum: 0xa2e7 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 192.168.1.102
    Destination Address: 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]
    [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)
      000. .... .... = Reserved: Not set
```

```
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.
```

The Sequence Number is **232129013**

Q3. 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) sent from the client to the web server (Do not consider the ACKs received from the server as part of these six segments)? 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 EstimatedRTT value (see relevant parts of Section 3.5 or lecture slides) after the receipt of each ACK? Assume that the initial value of EstimatedRTT is equal to the measured RTT (SampleRTT) for the first segment, and then is computed using the EstimatedRTT equation for all subsequent segments. Set alpha to 0.125.

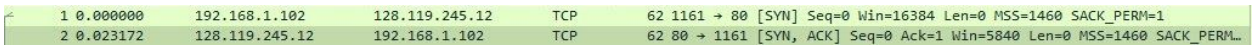
No.	*Sequence Num*	*Time sent (s)*	*ACK rcv (s)*	*RTT (diff)*	*EST RTT*
4	232129013	0.026477	0.053937	0.02746	0.02746
5	232129578	0.041737	0.077294	0.035557	0.028472
7	232131038	0.054026	0.124085	0.070059	0.03367
8	232132498	0.054690	0.169118	0.114428	0.043765
10	232133958	0.077405	0.217299	0.139894	0.055781
11	232135418	0/078157	0.267802	0.189645	0.072514

Q4. What is the length of each of the first six TCP segments? (same six segments as Q3)

No.	*Length*
4	565
5	1460
7	1460
8	1460
10	1460
11	1460

Q5. What is the minimum amount of available buffer space advertised at the receiver for the entire trace? Does the lack of receiver buffer space ever throttle the sender?

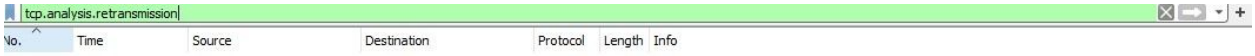
Sample output



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=1
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…

Buffer space available is 5840 bytes. The lack of buffer space will never throttle the sender and will grow until it reaches the maximum buffer size of 65536 bytes.

Q6. Are there any retransmitted segments in the trace file? What did you check for (in the trace) in order to answer this question?

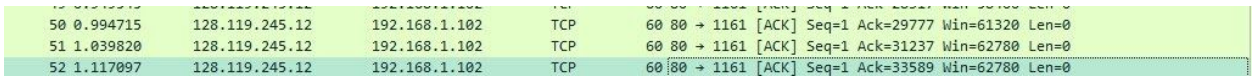


No. I used tcp.analysis.retransmission to check.

Q7. How much data does the receiver typically acknowledge in an ACK? Can you identify cases where the receiver is ACKing every other received segment (recall the discussion about delayed acks from the lecture notes or Section 3.5 of the text).

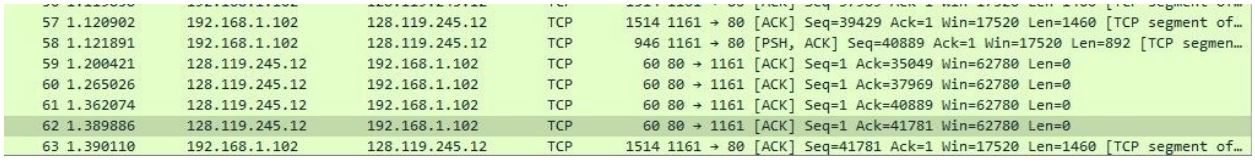
The receiver is typically acknowledging 1460 bytes of data.

A example of a case where the receiver is ACKing every other time is in around 52 where the receiver gas ACKed every other received segment



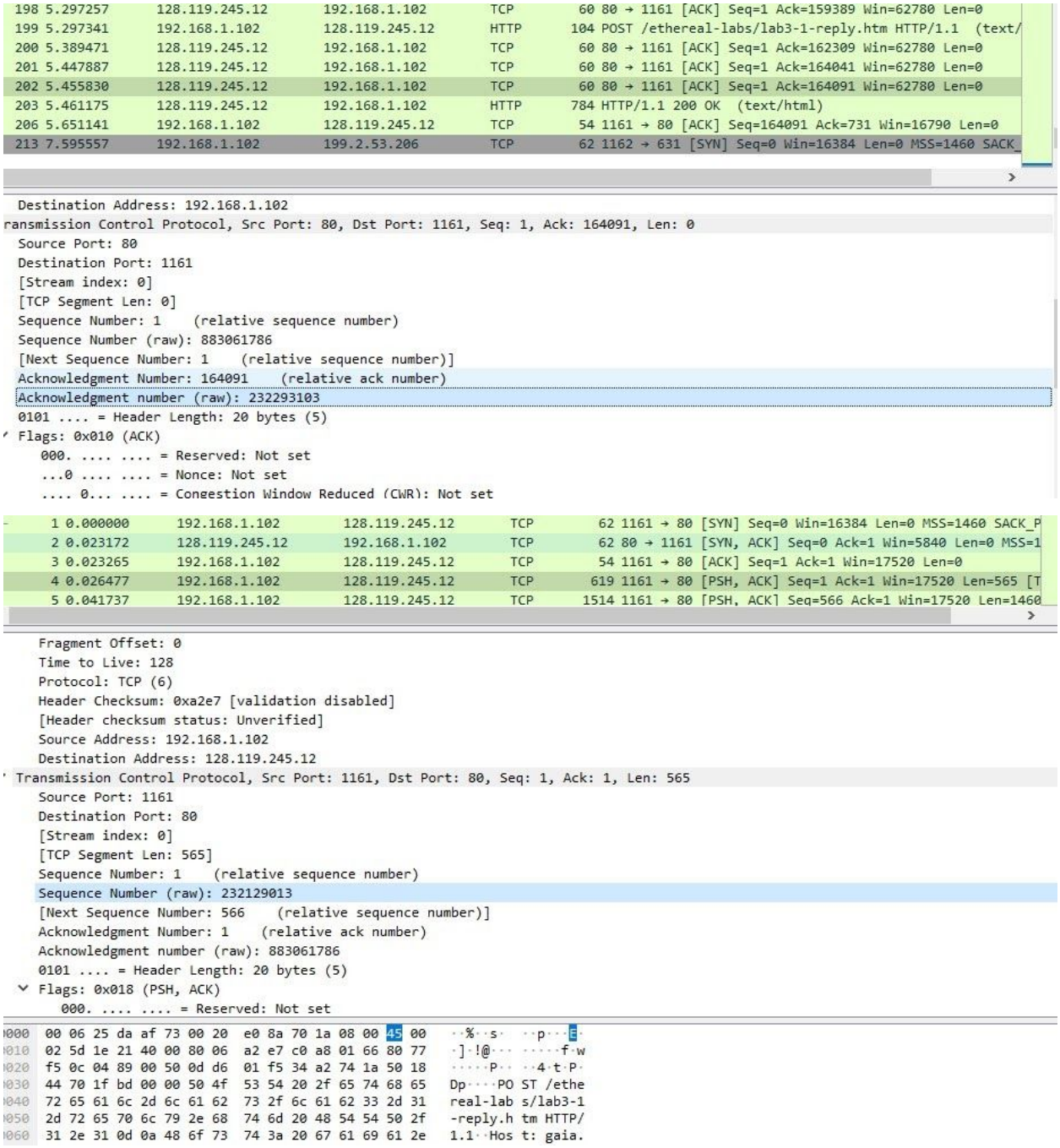
50	0.994715	128.119.245.12	192.168.1.102	TCP	60 80 → 1161 [ACK] Seq=1 Ack=29777 Win=61320 Len=0
51	1.039820	128.119.245.12	192.168.1.102	TCP	60 80 → 1161 [ACK] Seq=1 Ack=31237 Win=62780 Len=0
52	1.117097	128.119.245.12	192.168.1.102	TCP	60 80 → 1161 [ACK] Seq=1 Ack=33589 Win=62780 Len=0

Another example would be around 60 where the receiver has send ACKs for two segments.



57	1.120902	192.168.1.102	128.119.245.12	TCP	1514 1161 → 80 [ACK] Seq=39429 Ack=1 Win=17520 Len=1460 [TCP segment of…
58	1.121891	192.168.1.102	128.119.245.12	TCP	946 1161 → 80 [PSH, ACK] Seq=40889 Ack=1 Win=17520 Len=892 [TCP segmen…
59	1.200421	128.119.245.12	192.168.1.102	TCP	60 80 → 1161 [ACK] Seq=1 Ack=35049 Win=62780 Len=0
60	1.265026	128.119.245.12	192.168.1.102	TCP	60 80 → 1161 [ACK] Seq=1 Ack=37969 Win=62780 Len=0
61	1.362074	128.119.245.12	192.168.1.102	TCP	60 80 → 1161 [ACK] Seq=1 Ack=40889 Win=62780 Len=0
62	1.389886	128.119.245.12	192.168.1.102	TCP	60 80 → 1161 [ACK] Seq=1 Ack=41781 Win=62780 Len=0
63	1.390110	192.168.1.102	128.119.245.12	TCP	1514 1161 → 80 [ACK] Seq=41781 Ack=1 Win=17520 Len=1460 [TCP segment of…

Q8. What is the throughput (bytes transferred per unit time) for the TCP connection? Explain how you calculated this value.



Throughput = amount of data (D) / transmission time (T).

From last ACK #202 -> first SEQ #4 = 232293103 − 232129013 = 164090 *bytes transmitted*

Transmission time = ACK #202 -> first SEQ #4 = 5.455830 − 0.026477 = 5.429353 *seconds*

$\therefore T_{Trans} = D/T = 164090/5.429353 = 30222.75398\ byte/sec = 30.223\ kb/sec$

Exercise 2: TCP Connection Management

Given output

No	Source IP	Destination IP	Protocol	Info
295	10.9.16.201	10.99.6.175	TCP	50045 > 5000 [SYN] Seq=2818463618 win=8192 MSS=1460
296	10.99.6.175	10.9.16.201	TCP	5000 > 50045 [SYN, ACK] Seq=1247095790 Ack=2818463619 win=262144 MSS=1460
297	10.9.16.201	10.99.6.175	TCP	50045 > 5000 [ACK] Seq=2818463619 Ack=1247095791 win=65535
298	10.9.16.201	10.99.6.175	TCP	50045 > 5000 [PSH, ACK] Seq=2818463619 Ack=1247095791 win=65535
301	10.99.6.175	10.9.16.201	TCP	5000 > 50045 [ACK] Seq=1247095791 Ack=2818463652 win=262096
302	10.99.6.175	10.9.16.201	TCP	5000 > 50045 [PSH, ACK] Seq=1247095791 Ack=2818463652 win=262144
303	10.9.16.201	10.99.6.175	TCP	50045 > 5000 [ACK] Seq=2818463652 Ack=1247095831 win=65535
304	10.9.16.201	10.99.6.175	TCP	50045 > 5000 [FIN, ACK] Seq=2818463652 Ack=1247095831 win=65535
305	10.99.6.175	10.9.16.201	TCP	5000 > 50045 [FIN, ACK] Seq=1247095831 Ack=2818463652 win=262144
306	10.9.16.201	10.99.6.175	TCP	50045 > 5000 [ACK] Seq=2818463652 Ack=1247095832 win=65535
308	10.99.6.175	10.9.16.201	TCP	5000 > 50045 [ACK] Seq=1247095831 Ack=2818463653 win=262144

Q1 . What is the sequence number of the TCP SYN segment that is used to initiate the TCP connection between the client computer and server?

2818463618

Q2. What is the sequence number of the SYNACK segment sent by the server to the client computer in reply to the SYN? What is the value of the Acknowledgement field in the SYNACK segment? How did the server determine that value?

SYNACK segment is **1247095790**

ACK is **2818463619**

New ACK number = client x + 1 (incremented by 1)

Q3 . What is the sequence number of the ACK segment sent by the client computer in response to the SYNACK? What is the value of the Acknowledgment field in this ACK segment? Does this segment contain any data?

SEQ numb = **2818463619**

ACK num = **1247095791**

No. as the sequence number is the same as the previous ACK

Question 4 . Who has done the active close? client or the server? how you have determined this? What type of closure has been performed? 3 Segment (FIN/FINACK/ACK), 4 Segment (FIN/ACK/FIN/ACK) or Simultaneous close?

Both the client and server closed the connection as both sent [FIN,ACK] to each other.
Thus it is a simultaneous close.

Q5 . How many data bytes have been transferred from the client to the server and from the server to the client during the whole duration of the connection? What relationship does this have with the Initial Sequence Number and the final ACK received from the other side?

$$D_{Client -> server} = S_{304} - S_{295} - 1 = 2818463652 - 2818463618 - 1 = 33\ bytes$$
$$D_{Server -> client} = S_{305} - S_{296} - 1 = 1247095831 - 1247095790 - 1 = 40 bytes$$

Initial sequence number will be added with sent data to result in the final ACK. SYN and FYM segments increment ACK by 1 while no data is sent so 1 is removed to get the actual data transmitted. This can all be combined to calculate the total data transmitted between client and server.