Lab 04 - Ryan McClue (z5346008)

1. Understanding TCP using Wireshark

1. *What is the IP address of gaia.cs.umass.edu?*

`128.119.245.12`

On what port number is it sending and receiving TCP segments for this connection?

`80`

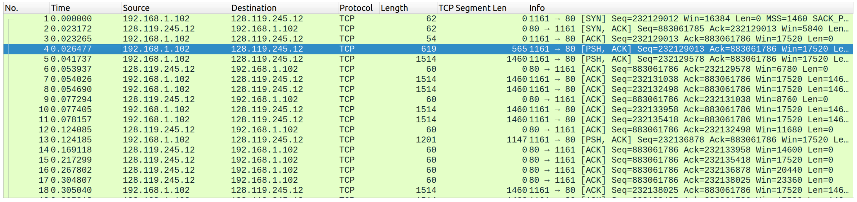
What are the IP address and TCP port numbers used by the client computer (source) that is transferring the file to gaia.cs.umass.edu?

`192.168.1.102, 1161`

2. *What is the sequence number of the TCP segment containing the HTTP POST command?*

`232129013`

3. *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 webserver 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 after the receipt of each ACK?*



No.	Time	Source	Destination	Protocol	Length	TCP Segment Len	Info
1	0.000000	192.168.1.102	128.119.245.12	TCP	62	0	1161 → 80 [SYN] Seq=232129012 Win=16384 Len=0 MSS=1460 SACK_P…
2	0.023172	128.119.245.12	192.168.1.102	TCP	62	0	80 → 1161 [SYN, ACK] Seq=883061785 Ack=232129013 Win=5840 Len…
3	0.023265	192.168.1.102	128.119.245.12	TCP	54	0	1161 → 80 [ACK] Seq=232129013 Ack=883061786 Win=17520 Len=0
4	0.026477	192.168.1.102	128.119.245.12	TCP	619	565	1161 → 80 [PSH, ACK] Seq=232129013 Ack=883061786 Win=17520 Le…
5	0.041737	192.168.1.102	128.119.245.12	TCP	1514	1460	1161 → 80 [PSH, ACK] Seq=232129578 Ack=883061786 Win=17520 Le…
6	0.053937	128.119.245.12	192.168.1.102	TCP	60	0	80 → 1161 [ACK] Seq=883061786 Ack=232129578 Win=6780 Len=0
7	0.054026	192.168.1.102	128.119.245.12	TCP	1514	1460	1161 → 80 [ACK] Seq=232131038 Ack=883061786 Win=17520 Len=146…
8	0.054690	192.168.1.102	128.119.245.12	TCP	1514	1460	1161 → 80 [ACK] Seq=232132498 Ack=883061786 Win=17520 Len=146…
9	0.077294	128.119.245.12	192.168.1.102	TCP	60	0	80 → 1161 [ACK] Seq=883061786 Ack=232131038 Win=8760 Len=0
10	0.077405	192.168.1.102	128.119.245.12	TCP	1514	1460	1161 → 80 [ACK] Seq=232133958 Ack=883061786 Win=17520 Len=146…
11	0.078157	192.168.1.102	128.119.245.12	TCP	1514	1460	1161 → 80 [ACK] Seq=232135418 Ack=883061786 Win=17520 Len=146…
12	0.124085	128.119.245.12	192.168.1.102	TCP	60	0	80 → 1161 [ACK] Seq=883061786 Ack=232132498 Win=11680 Len=0
13	0.124185	192.168.1.102	128.119.245.12	TCP	1201	1147	1161 → 80 [PSH, ACK] Seq=232136878 Ack=883061786 Win=17520 Le…
14	0.169118	128.119.245.12	192.168.1.102	TCP	60	0	80 → 1161 [ACK] Seq=883061786 Ack=232133958 Win=14600 Len=0
15	0.217299	128.119.245.12	192.168.1.102	TCP	60	0	80 → 1161 [ACK] Seq=883061786 Ack=232135418 Win=17520 Len=0
16	0.267802	128.119.245.12	192.168.1.102	TCP	60	0	80 → 1161 [ACK] Seq=883061786 Ack=232136878 Win=20440 Len=0
17	0.304807	128.119.245.12	192.168.1.102	TCP	60	0	80 → 1161 [ACK] Seq=883061786 Ack=232138025 Win=23360 Len=0
18	0.305040	192.168.1.102	128.119.245.12	TCP	1514	1460	1161 → 80 [ACK] Seq=232138025 Ack=883061786 Win=17520 Len=146…

1: seq: `232129013`, time-sent: `0.026477`, ack-time-recieved: `0.053937` , rtt: `0.027460` , estimated-rtt: `0.02746`

2: seq: `232129578`, time-sent: `0.041737`, ack-time-recieved: `0.077294` , rtt: `0.035557` , estimated-rtt: `0.02847`

3: seq: `232131038`, time-sent: `0.054026`, ack-time-recieved: `0.124085` , rtt: `0.070059` , estimated-rtt: `0.03367`

4: seq: `232132498`, time-sent: `0.054690`, ack-time-recieved: `0.169118` , rtt: `0.114428` , estimated-rtt: `0.04377`

5: seq: `232133958`, time-sent: `0.077405`, ack-time-recieved: `0.217299` , rtt: `0.139894` , estimated-rtt: `0.05578`

6: seq: `232135418`, time-sent: `0.078157`, ack-time-recieved: `0.267802` , rtt: `0.189645` , estimated-rtt: `0.07251`

```
#!/bin/python

def est(est_rtt, sample_rtt):
  return ((1 - 0.125)*(est_rtt)) + (0.125 * sample_rtt)

est_rtt = est(0.027460, 0.027460)
print(est_rtt)
est_rtt = est(est_rtt, 0.035557)
print(est_rtt)
est_rtt = est(est_rtt, 0.070059)
print(est_rtt)
est_rtt = est(est_rtt, 0.114428)
print(est_rtt)
est_rtt = est(est_rtt, 0.139894)
print(est_rtt)
est_rtt = est(est_rtt, 0.189645)
print(est_rtt)
```

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

1: `565` **2:** `1460` **3:** `1460` **4:** `1460` **5:** `1460` **6:** `1460`

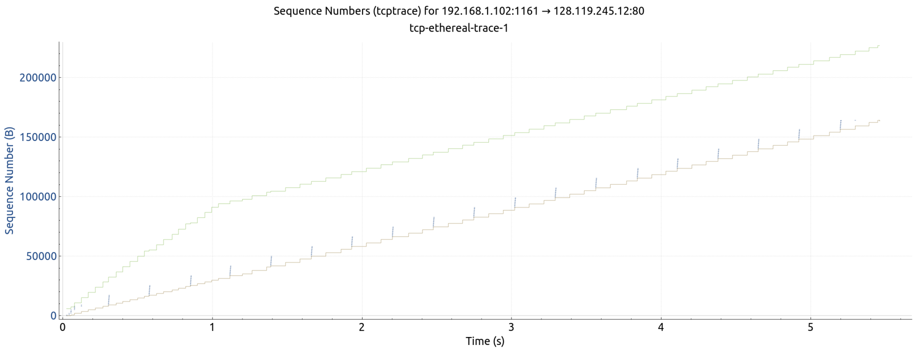
5. *What is the minimum amount of available buffer space advertised at the receiver for the entire trace?*

Observing initial window size of first packet from server, see `5480` bytes

Does the lack of receiver buffer space ever throttle the sender?

No. Inspecting the window size of sequential packets from the server, we see it's always increasing. Therefore no throttling

6. *Are there any retransmitted segments in the trace file?*



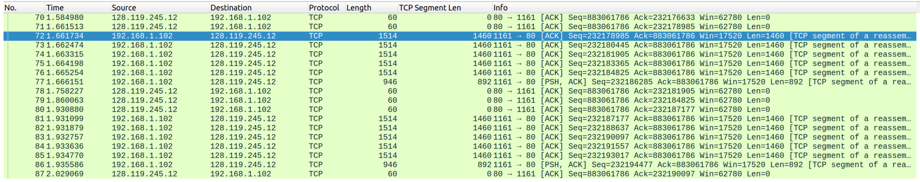
Wireshark menu navigation: `Statistics -> TCP Stream Graphs -> Time`

`Sequence`. There are no duplicated sequence numbers, so no retransmission

7. *How much data does the receiver typically acknowledge in an ACK?*

1460

Can you identify cases where the receiver is ACKing every other received segment:



No.	Time	Source	Destination	Protocol	Length	TCP Segment Len	Info
70	1.584980	128.119.245.12	192.168.1.102	TCP	60	0	80 → 1161 [ACK] Seq=883061786 Ack=232176633 Win=62780 Len=0
71	1.661513	128.119.245.12	192.168.1.102	TCP	60	0	80 → 1161 [ACK] Seq=883061786 Ack=232178985 Win=62780 Len=0
72	1.661734	192.168.1.102	128.119.245.12	TCP	1514	1460	1161 → 80 [ACK] Seq=232178985 Ack=883061786 Win=17520 Len=1460 [TCP segment of a reassem…
73	1.662474	192.168.1.102	128.119.245.12	TCP	1514	1460	1161 → 80 [ACK] Seq=232180445 Ack=883061786 Win=17520 Len=1460 [TCP segment of a reassem…
74	1.663315	192.168.1.102	128.119.245.12	TCP	1514	1460	1161 → 80 [ACK] Seq=232181905 Ack=883061786 Win=17520 Len=1460 [TCP segment of a reassem…
75	1.664198	192.168.1.102	128.119.245.12	TCP	1514	1460	1161 → 80 [ACK] Seq=232183365 Ack=883061786 Win=17520 Len=1460 [TCP segment of a reassem…
76	1.665254	192.168.1.102	128.119.245.12	TCP	1514	1460	1161 → 80 [ACK] Seq=232184825 Ack=883061786 Win=17520 Len=1460 [TCP segment of a reassem…
77	1.666151	192.168.1.102	128.119.245.12	TCP	946	892	1161 → 80 [PSH, ACK] Seq=232186285 Ack=883061786 Win=17520 Len=892 [TCP segment of a rea…
78	1.758227	128.119.245.12	192.168.1.102	TCP	60	0	80 → 1161 [ACK] Seq=883061786 Ack=232181905 Win=62780 Len=0
79	1.860063	128.119.245.12	192.168.1.102	TCP	60	0	80 → 1161 [ACK] Seq=883061786 Ack=232184825 Win=62780 Len=0
80	1.930880	128.119.245.12	192.168.1.102	TCP	60	0	80 → 1161 [ACK] Seq=883061786 Ack=232187177 Win=62780 Len=0
81	1.931099	192.168.1.102	128.119.245.12	TCP	1514	1460	1161 → 80 [ACK] Seq=232187177 Ack=883061786 Win=17520 Len=1460 [TCP segment of a reassem…
82	1.931879	192.168.1.102	128.119.245.12	TCP	1514	1460	1161 → 80 [ACK] Seq=232188637 Ack=883061786 Win=17520 Len=1460 [TCP segment of a reassem…
83	1.932757	192.168.1.102	128.119.245.12	TCP	1514	1460	1161 → 80 [ACK] Seq=232190097 Ack=883061786 Win=17520 Len=1460 [TCP segment of a reassem…
84	1.933636	192.168.1.102	128.119.245.12	TCP	1514	1460	1161 → 80 [ACK] Seq=232191557 Ack=883061786 Win=17520 Len=1460 [TCP segment of a reassem…
85	1.934770	192.168.1.102	128.119.245.12	TCP	1514	1460	1161 → 80 [ACK] Seq=232193017 Ack=883061786 Win=17520 Len=1460 [TCP segment of a reassem…
86	1.935586	192.168.1.102	128.119.245.12	TCP	946	892	1161 → 80 [PSH, ACK] Seq=232194477 Ack=883061786 Win=17520 Len=892 [TCP segment of a rea…
87	2.029069	128.119.245.12	192.168.1.102	TCP	60	0	80 → 1161 [ACK] Seq=883061786 Ack=232190097 Win=62780 Len=0

From packet 72-77 observe sending 6 packets. However, only recieve 3 distinct ACKs (packets 78-80) This is an example of cumulutive ACKs as a result of employing a delayed ACK protocol

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

Packet 202 last from server to client. Subtract the first packet's SEQ number: `232129012` from this packet's ACK number: `232293103`. Divide by total time, which is the time of packet 202 being recieved. This gives `(232293103 - 232129012) / 5.455830` a throughput of approximately 30076 bytes per unit time

2. TCP Connection Management

1. *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`

2. *What is the sequence number of the SYNACK segment sent by the server to the client computer in reply to the SYN?*

`1247095790`

What is the value of the Acknowledgement field in the SYNACK segment?

`2818463619`

How did the server determine that value?

Added 1 to the sequence number provided by the client's SYN packet, to account for the 1 byte of data used in the SYN packet

3. *What is the sequence number of the ACK segment sent by the client computer in response to the SYNACK?*

`2818463619`

What is the value of the Acknowledgment field in this ACK segment?

`1247095791`

Does this segment contain any data?

No

4. *Who has done the active close? client or the server? how you have determined this? What type of closure has been performed?*

A simultaneous close has been peformed, meaning the client and server simultaneous initiated the closure. We can see this by looking at the order of packets being FIN/FIN/ACK/ACK, i.e. neither client or server waits for the other's FIN before sending its own.

5. *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?*

`2818463653 - 2818463618 - 2 = 33`

What relationship does this have with the Initial Sequence Number and the final ACK received from the other side?

It's the final ACK number subtracted by the Initial Sequence Number. However, this includes the 1 byte for SYN packet and 1 byte for FIN packet. Therefore, subtract this value by 2 to get number of data bytes.