

Computer Networks CSE 4344/5344

Project 2

Transmission Control Protocol Analysis using Wireshark

Instructor: Dr. Sajib Datta

GTA (4344/5344): Jees Augustine

Summer 2020

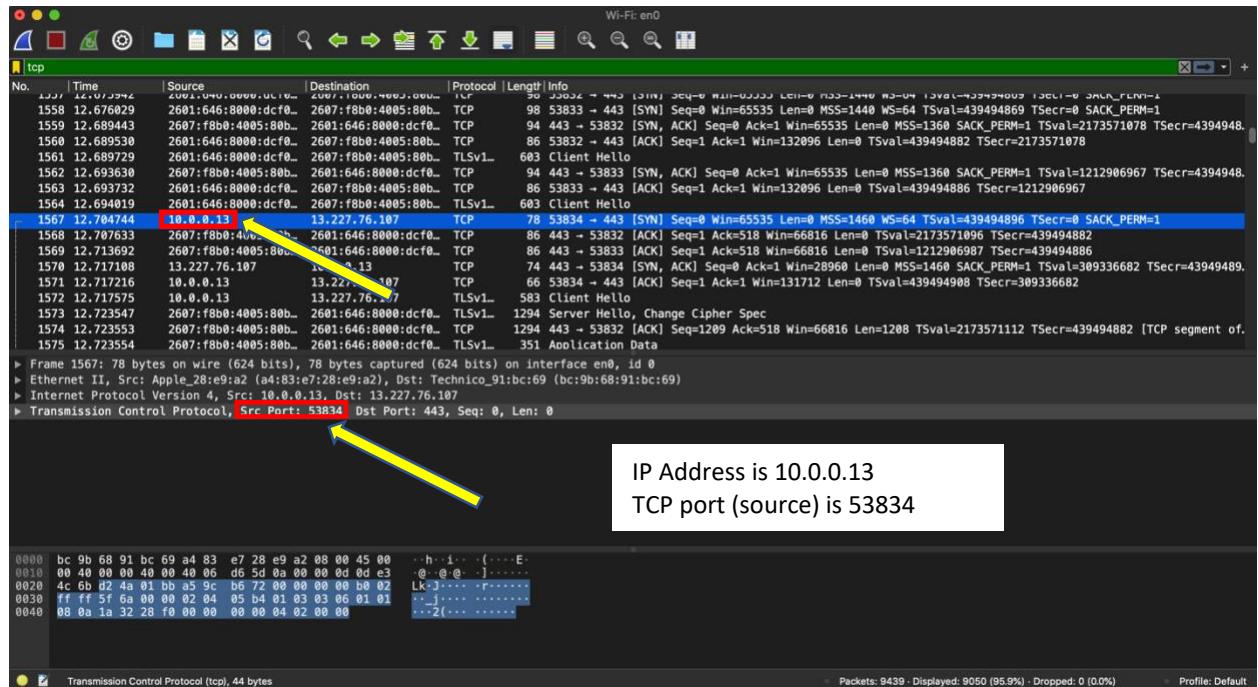
Name : Jahnavi Nuthalapati

Table of Contents

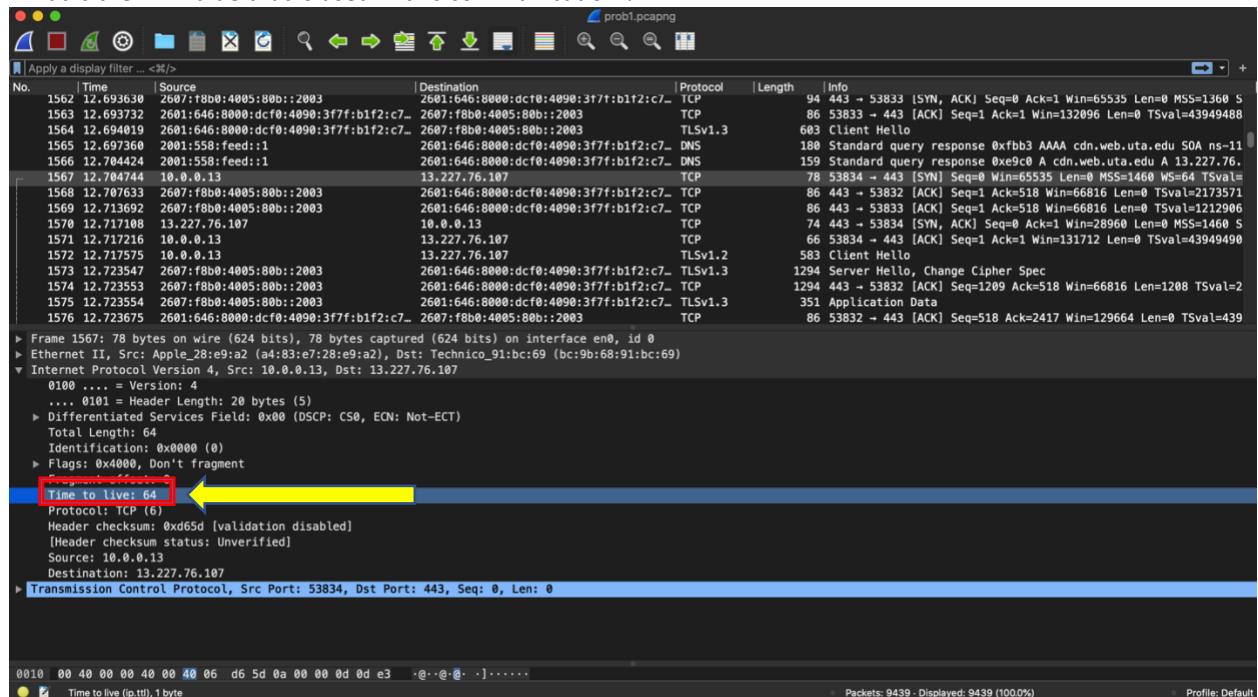
Sno.	Title	Page
1	Problem Set 1	2
2	Problem Set 2	7
3	Problem Set 3	10
4	Problem Set 4	20

Problem Set 1

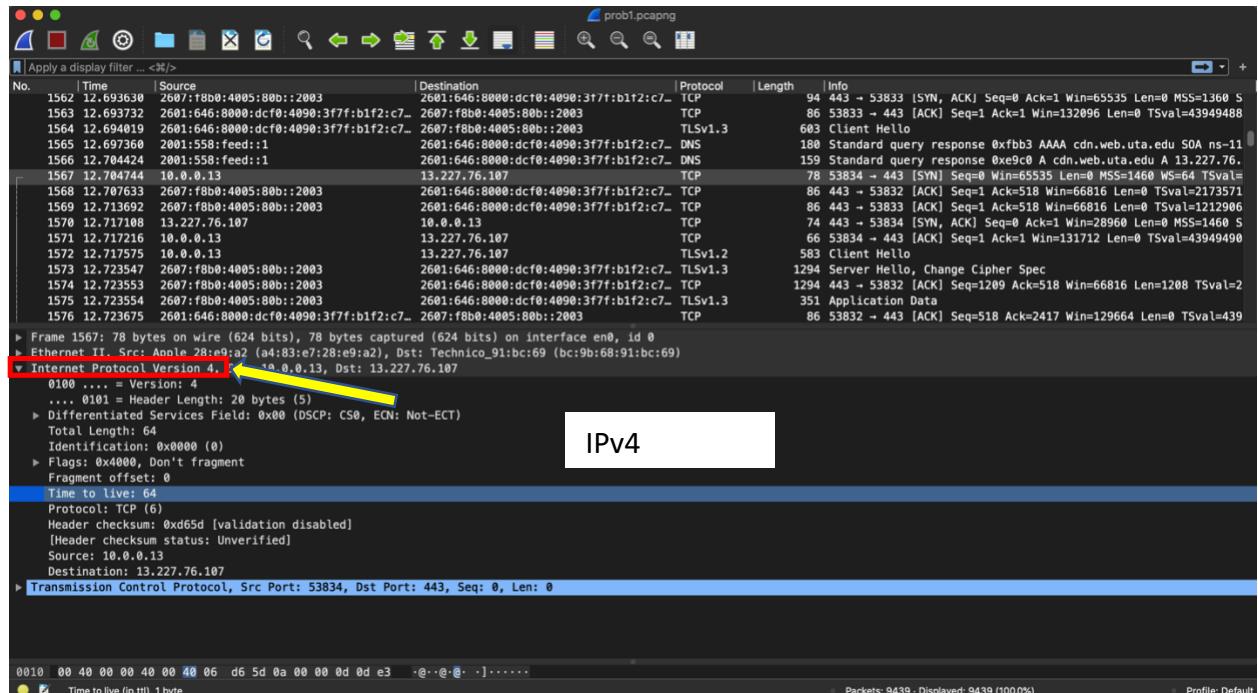
- What is the IP address and TCP port number used by your client computer (source) to browse the page uta.edu.



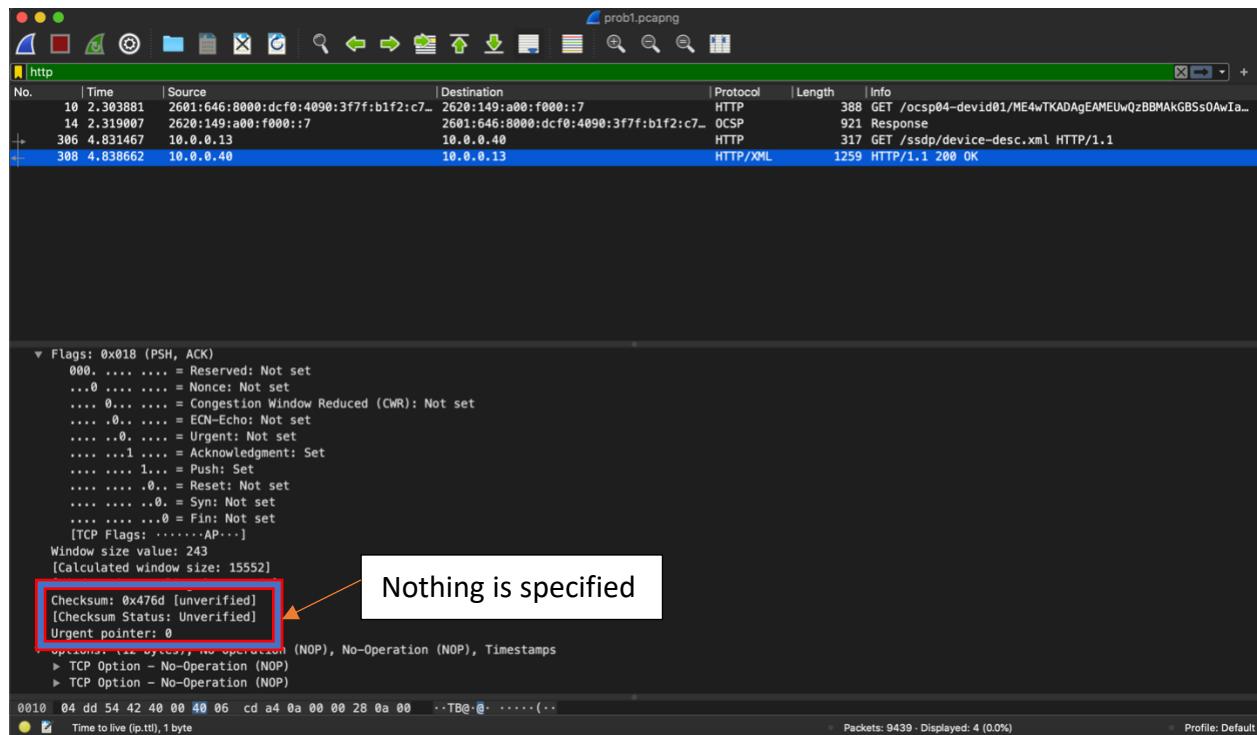
- What is the TTL value that is used in this communication?



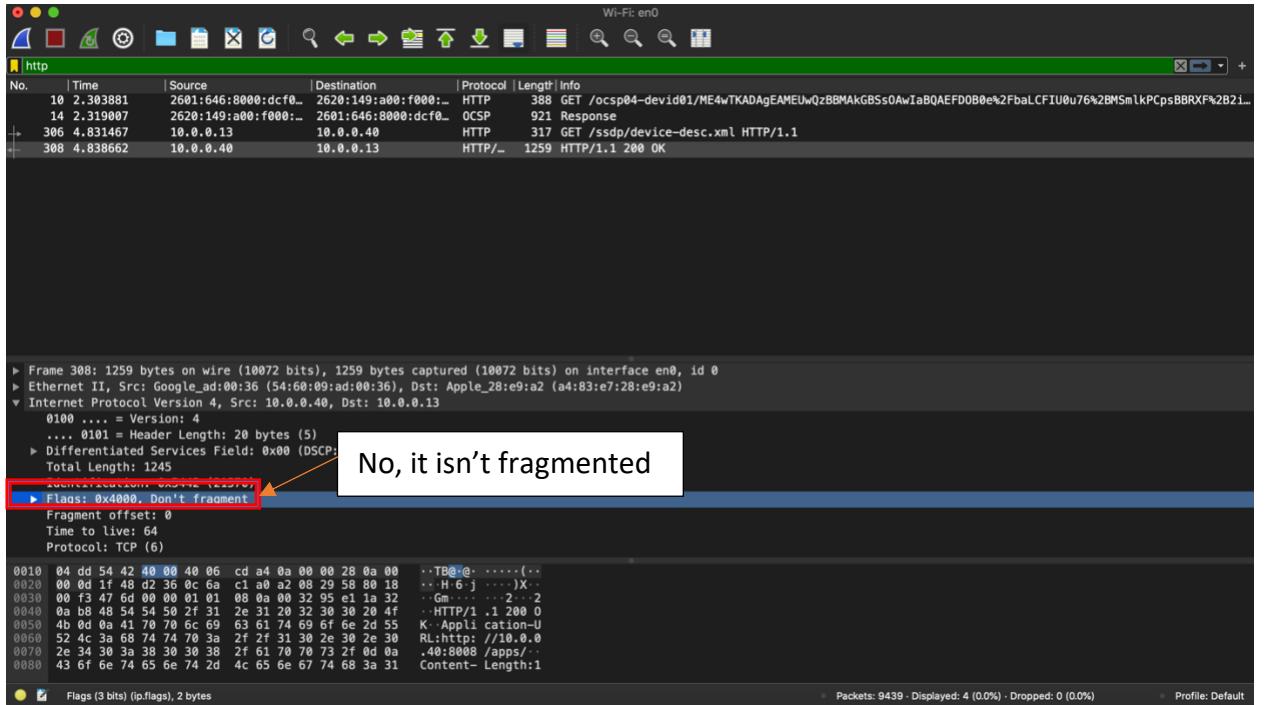
3. Did you Use IPv4 or IPv6 for communication ?



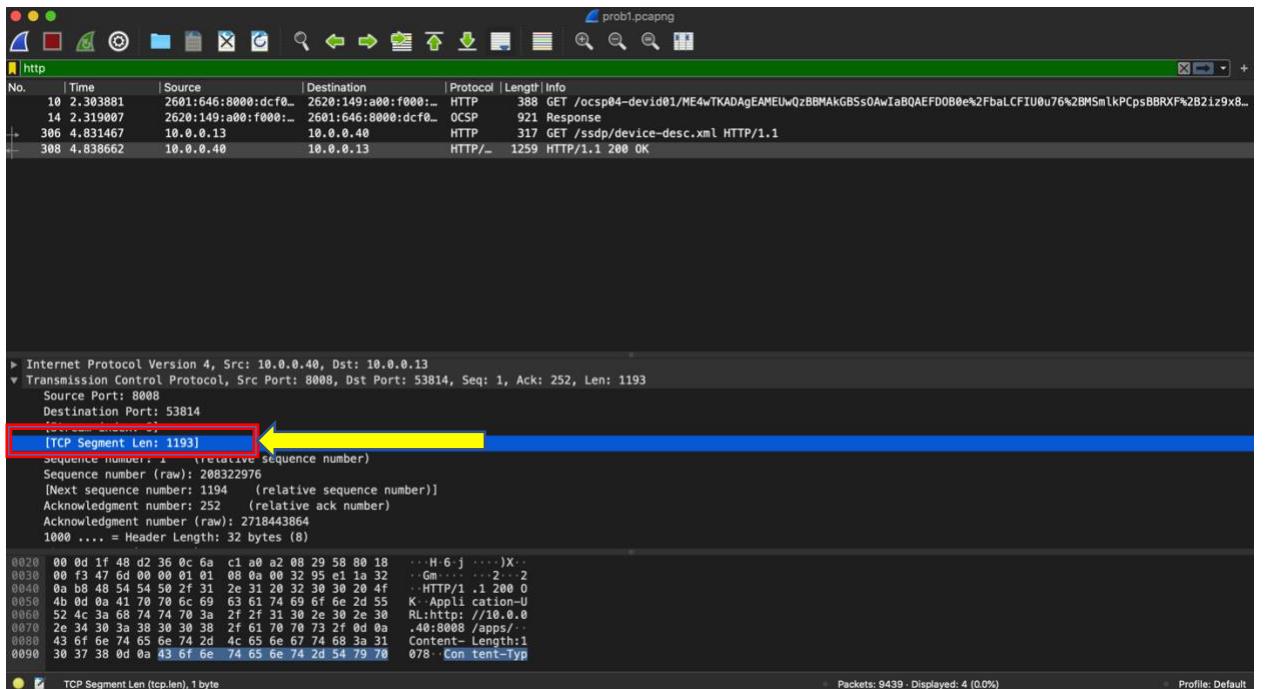
4. Does your optional field has some particular information or not.



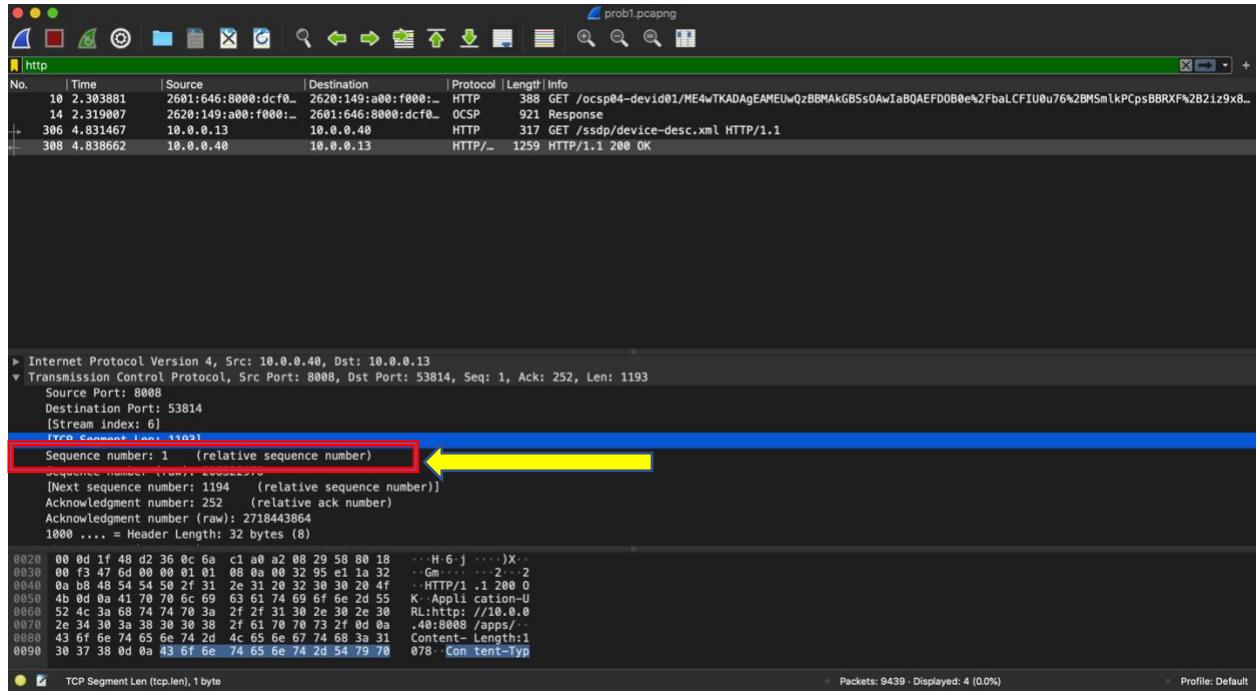
5. Is the Packet Fragmented



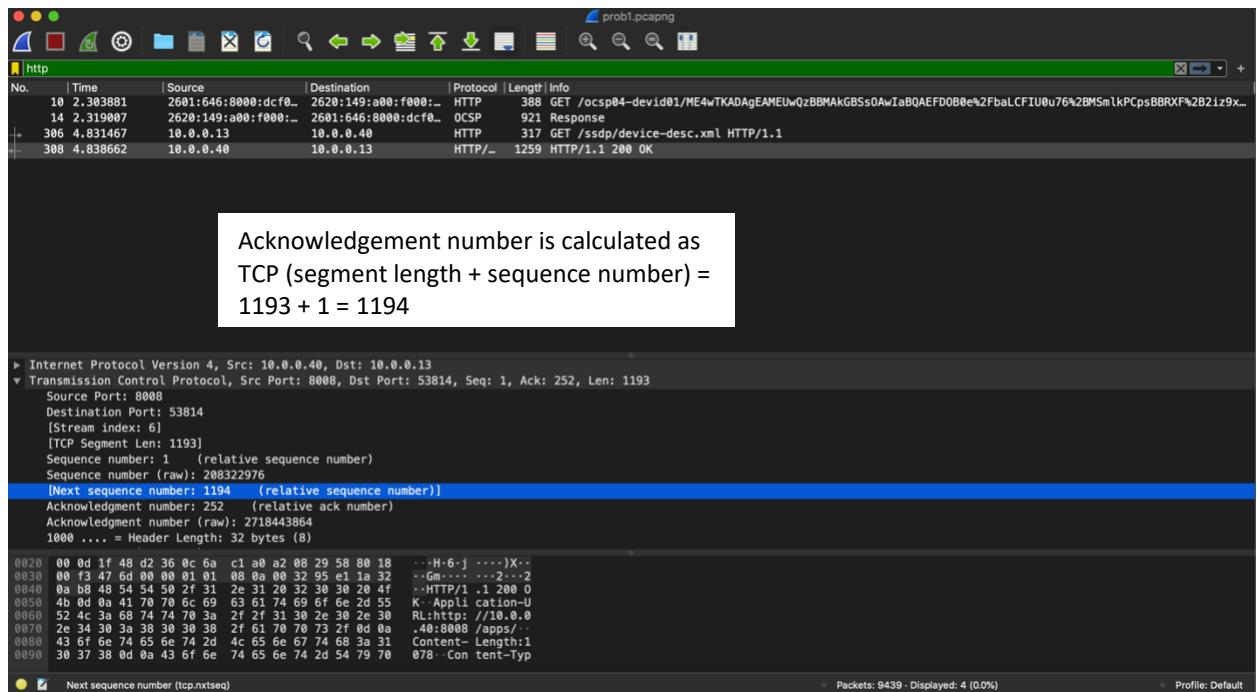
6. What is the TCP segment length ?



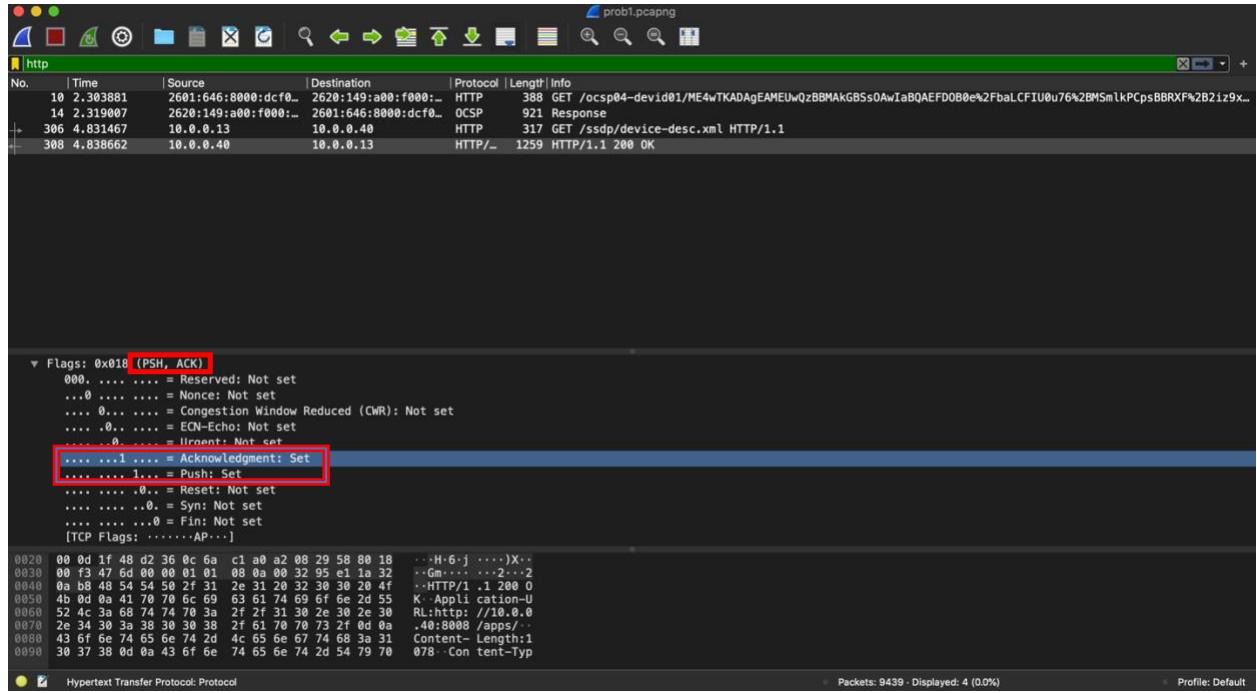
7. What is the Sequence Number of TCP segment (you can use the relative sequence number).



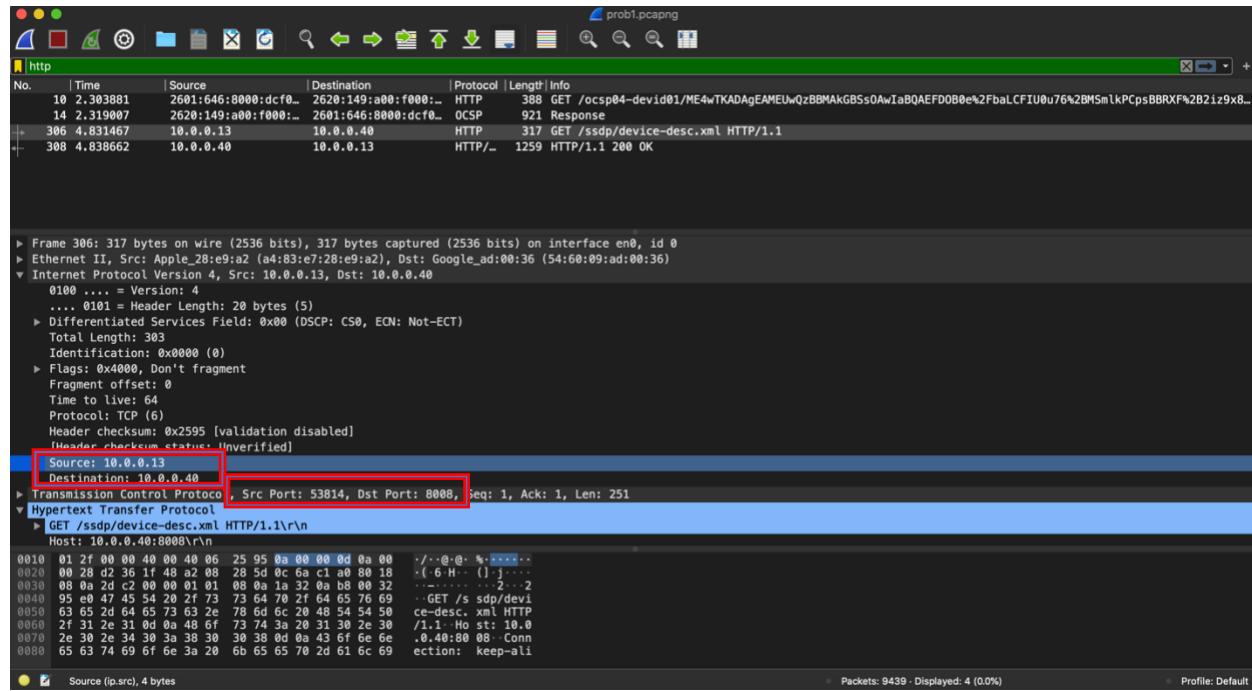
8. Calculate the acknowledgement number based on the two questions above. Verify your solution with the Wireshark values.



9. What are the fields in the TCP Flags. No need to give any values but give the field names given in Wireshark

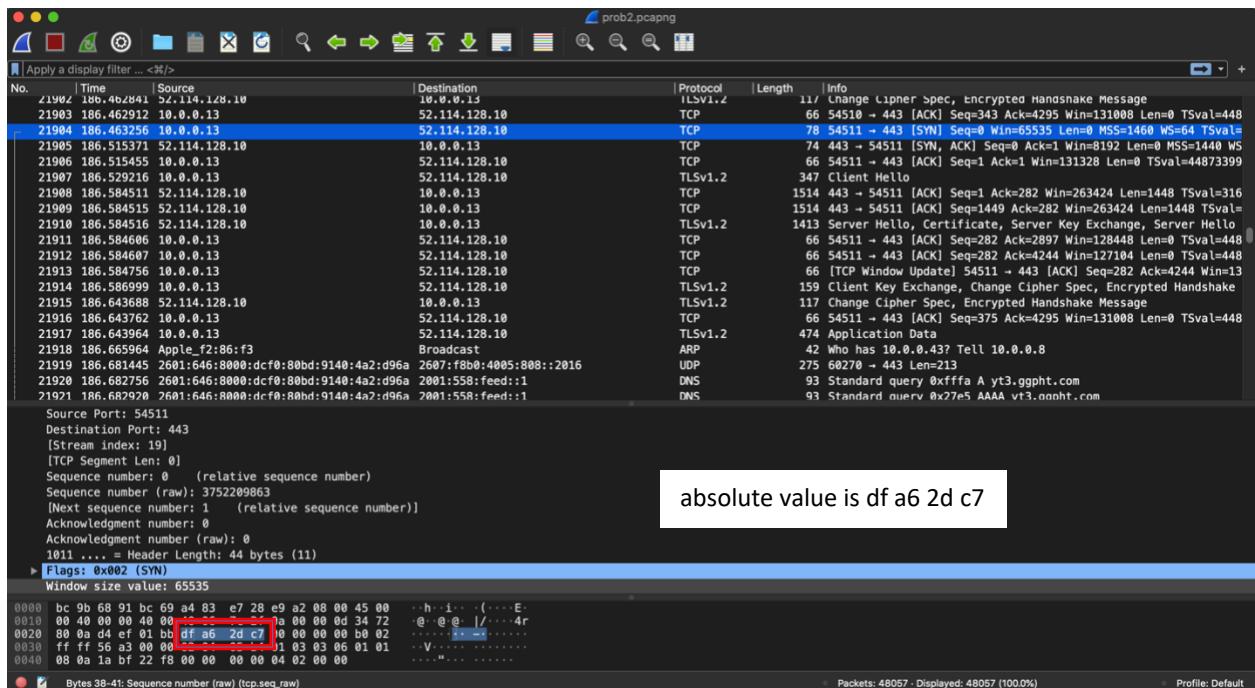
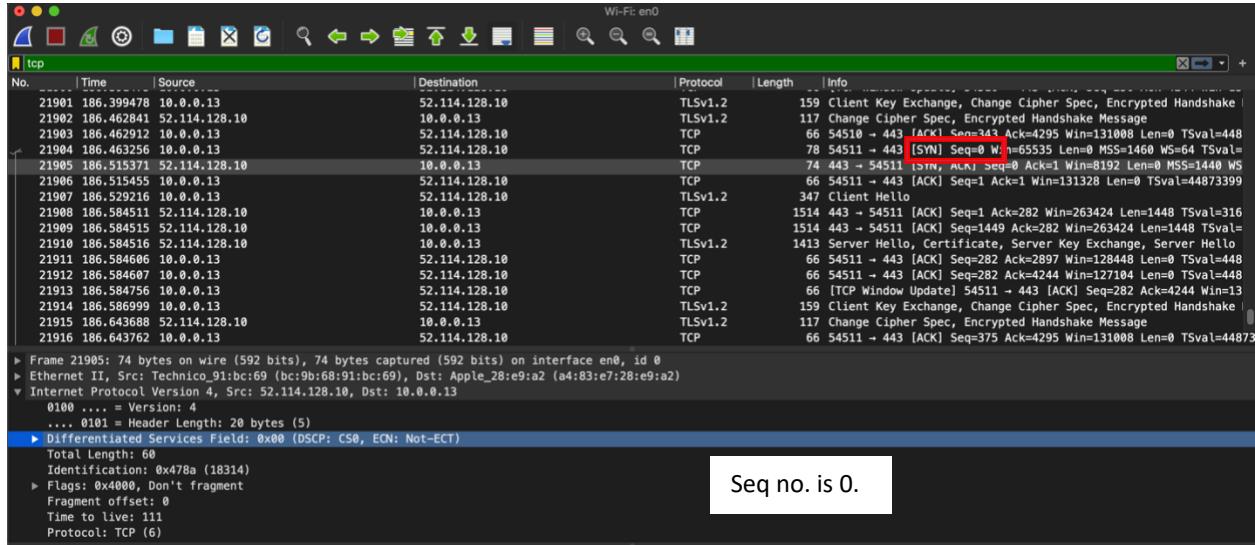


10. What is the IP address of uta.edu ? On what port number is it sending and receiving TCP segments for this connection?



Problem Set 2

- What is the sequence number (absolute) of the TCP SYN segment that is used to initiate the TCP connection between the client computer and youtube.com?



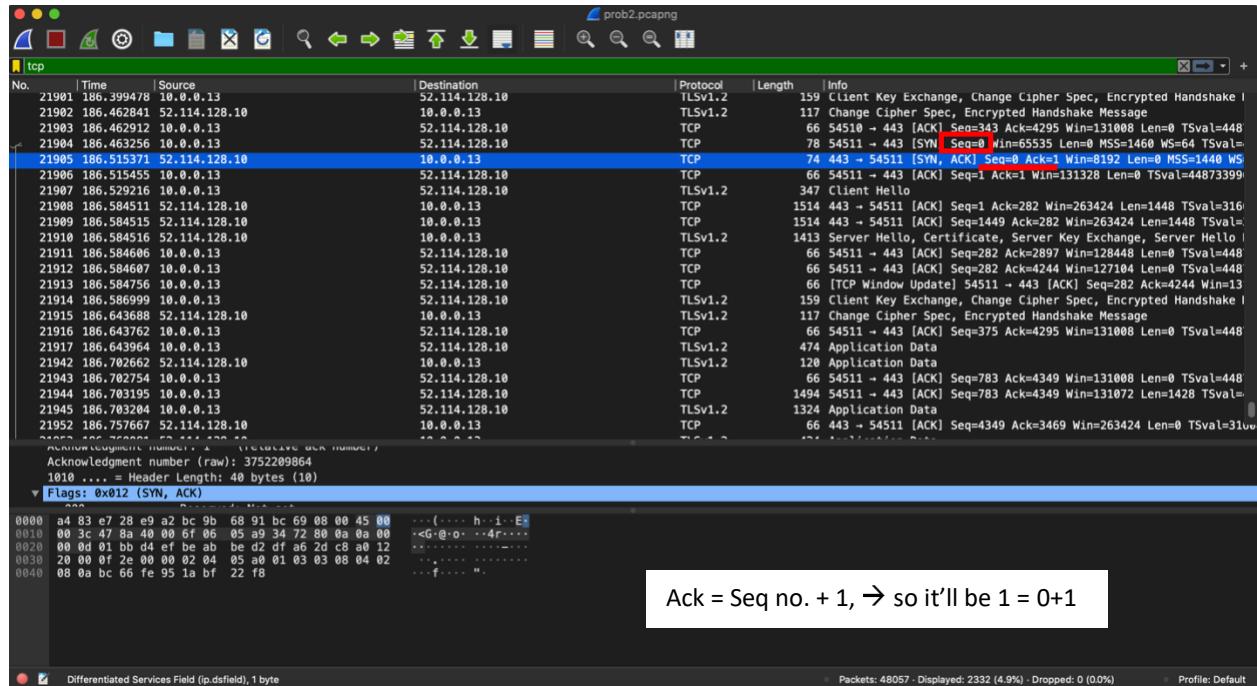
2. What is it in the segment that identifies the segment as a SYN segment?
 (The highlighted boxes)

Wi-Fi: en0						
No.	Time	Source	Destination	Protocol	Length	Info
[Header checksum status: Unverified]						
		Source: 10.0.0.13	Destination: 52.114.128.10			
▼		Transmission Control Protocol, Src Port: 54511, Dst Port: 443, Seq: 0, Len: 0				
		Source Port: 54511	Destination Port: 443			
		[Stream index: 19]				
		[TCP Segment Len: 0]				
		Sequence number: 0 (relative sequence number)				
		Sequence number (raw): 3752209863				
		[Next sequence number: 1 (relative sequence number)]				
		Acknowledgment number: 0				
		Acknowledgment number (raw): 0				
		Flags: 0x002 (SYN)				
	 = 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.... = Sync Set				
	0.... = RST: Not set				
		[TCP Flags: 0x002 S]				
		Window size value: 65535				
		[Calculated window size: 65535]				
0000	bc 9b 68 91 bc 69 a4 83 e7 28 e9 a2 08 00 45 00h...-i-.....E				
0010	00 40 00 00 40 00 40 06 7c 2f 0a 00 00 0d 34 72	@-@-@- /-.....4r-				
0020	80 0a d4 ef 01 bb df a6 2d c7 00 00 00 00 b0 02-.....-.....-				
0030	ff ff f6 a3 00 00 02 04 05 b4 01 03 03 06 01 01	..v.....				
0040	08 0a 1a bf 22 f8 00 00 00 00 04 02 00 00m.....				
Differentiated Services Field (ip.dsfield), 1 byte						
Packets: 48057 - Displayed: 2332 (4.9%) - Dropped: 0 (0.0%)						
Profile: Default						

3. What is the sequence number of the SYNACK segment sent by youtube.com to the client computer in reply to the SYN? What is the value of the Acknowledgement field in the SYNACK segment?

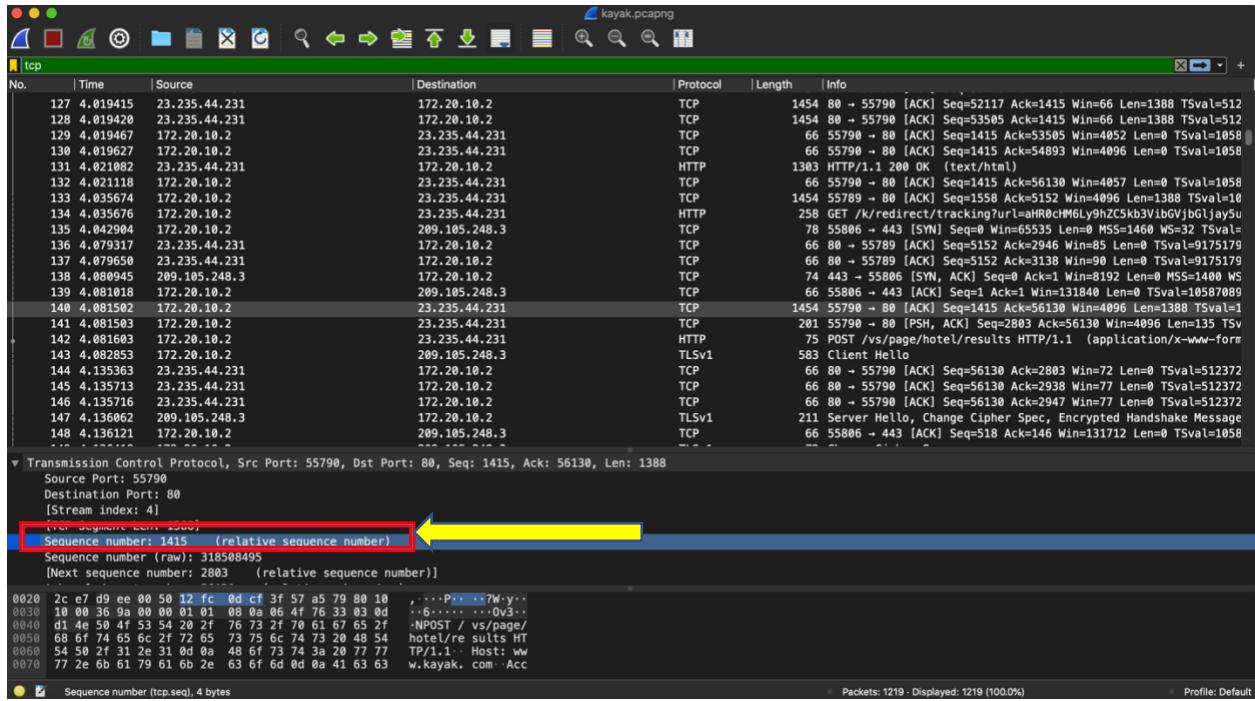
Wi-Fi: en0						
No.	Time	Source	Destination	Protocol	Length	Info
[Header checksum status: Unverified]						
21982	186.462841	52.114.128.10	10.0.0.13	TLSv1.2	117	Change Cipher Spec, Encrypted Handshake Message
21983	186.462912	10.0.0.13	52.114.128.10	TCP	66	54510 -> 443 [ACK] Seq=343 Ack=4295 Win=131008 Len=0 TSval=448
21984	186.463256	52.114.128.10	10.0.0.13	TCP	1413	Client Key Exchange, Change Cipher Spec, Encrypted Handshake
21985	186.515371	52.114.128.10	10.0.0.13	TCP	74	443 -> 54511 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1440 WS=64 TSval=448
21986	186.515455	10.0.0.13	52.114.128.10	TCP	00	54511 -> 443 [ACK] Seq=1 Ack=1 Win=131008 Len=0 TSval=44873399
21987	186.5159216	10.0.0.13	52.114.128.10	TLSv1.2	347	Client Hello
21988	186.584511	52.114.128.10	10.0.0.13	TCP	1514	443 -> 54511 [ACK] Seq=1 Ack=282 Win=263424 Len=1448 TSval=316
21989	186.584515	52.114.128.10	10.0.0.13	TCP	1514	443 -> 54511 [ACK] Seq=1449 Ack=282 Win=263424 Len=1448 TSval=316
21990	186.584516	52.114.128.10	10.0.0.13	TLSv1.2	1413	Server Hello, Certificate, Server Key Exchange, Server Hello
21991	186.584606	10.0.0.13	52.114.128.10	TCP	66	54511 -> 443 [ACK] Seq=282 Ack=2897 Win=128448 Len=0 TSval=448
21992	186.584607	10.0.0.13	52.114.128.10	TCP	66	54511 -> 443 [ACK] Seq=282 Ack=4244 Win=127104 Len=0 TSval=448
21993	186.584756	10.0.0.13	52.114.128.10	TCP	66	[TCP Window Update] 54511 -> 443 [ACK] Seq=282 Ack=4244 Win=448
21994	186.586999	10.0.0.13	52.114.128.10	TLSv1.2	159	Client Key Exchange, Change Cipher Spec, Encrypted Handshake
21995	186.643688	52.114.128.10	10.0.0.13	TLSv1.2	117	Change Cipher Spec, Encrypted Handshake Message
21996	186.643762	10.0.0.13	52.114.128.10	TCP	66	54511 -> 443 [ACK] Seq=375 Ack=4295 Win=131008 Len=0 TSval=448
21997	186.643964	10.0.0.13	52.114.128.10	TLSv1.2	474	Application Data
[Stream index: 19]						
		[TCP Segment Len: 0]				
		Sequence number: 0 (relative sequence number)				
		Sequence number (raw): 3198926546				
		[Next sequence number: 1 (relative sequence number)]				
		Acknowledgment number: 1 (relative ack number)				
		Acknowledgment number (raw): 3752209864				
		Flags: 0x012 (SYN, ACK)				
	 = Reserved: Not set				
0000	a4 83 e7 28 e9 a2 bc 9b 68 91 bc 69 08 00 45 00h...-i-.....E				
0010	00 3c 47 8a 40 00 6f 06 05 a9 34 72 00 0a 00	<-o-@-o-4r-				
0020	00 0d 01 bb d4 ef be ab be d2 df a6 2d c8 a0 12-.....-.....-				
0030	20 00 0f 2e 00 00 02 04 05 a0 01 03 03 08 04 02-.....-.....-				
0040	08 0a bc 66 fe 95 1a bf 22 f8f.....				
Differentiated Services Field (ip.dsfield), 1 byte						
Packets: 48057 - Displayed: 2332 (4.9%) - Dropped: 0 (0.0%)						
Profile: Default						

4. How did youtube.com determine that value? What is it in the segment that identifies the segment as a SYNACK segment?



Problem Set 3

- What is the sequence number of the TCP segment containing the first HTTP POST command?



- Consider the TCP segment containing the HTTP POST as the first segment in the TCP connection.

(The screenshots related to this question can be found below the calculations 2q)

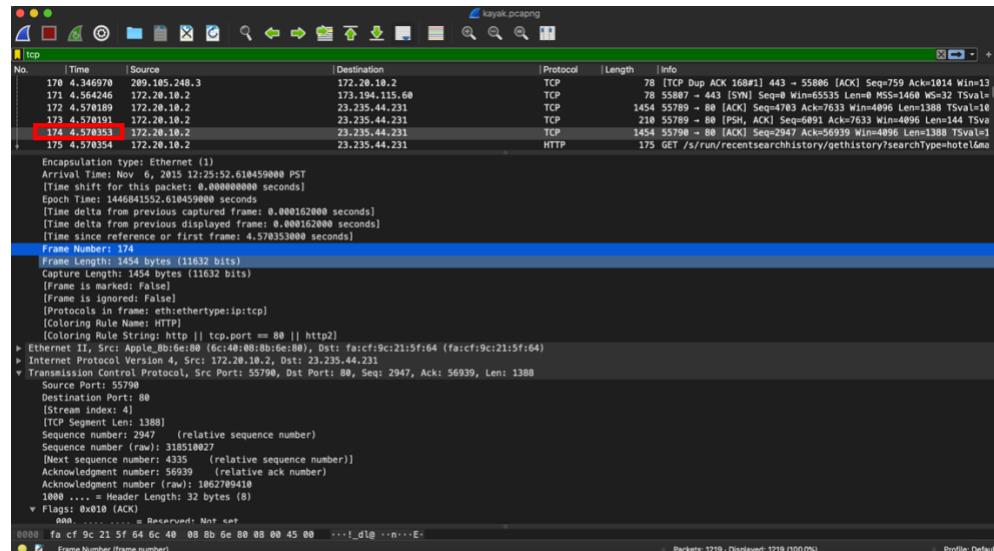
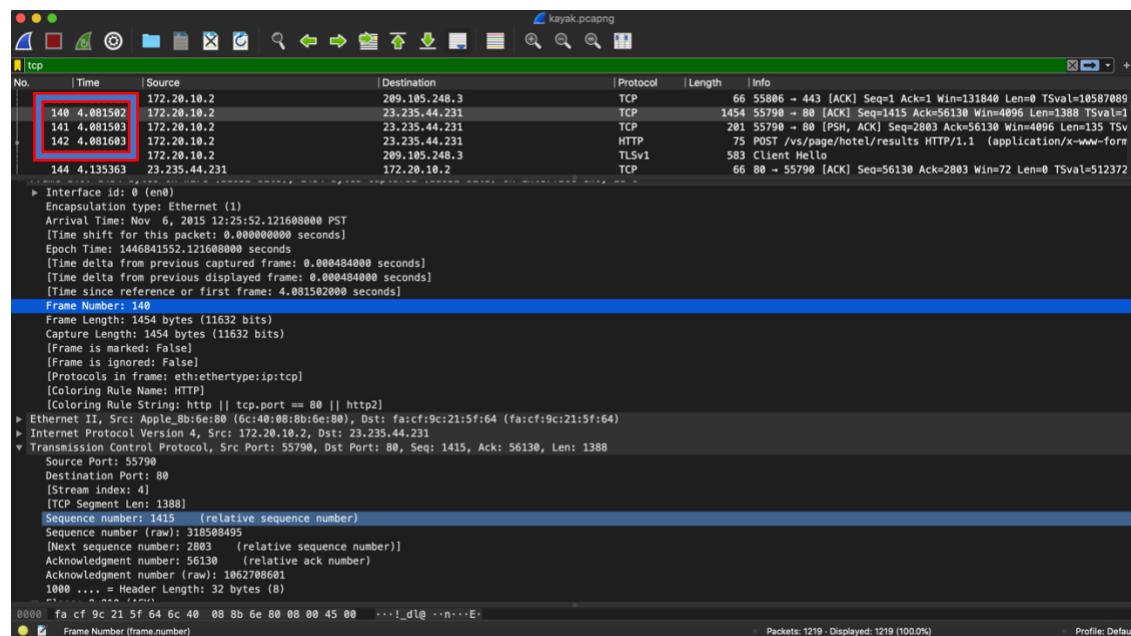
- (i) What are the sequence numbers of the first four segments in the TCP connection (including the segment containing the HTTP POST)?

Segment Number	Sequence Number
140	1415
141	2803
142	2938
174	2947

(ii) At what time was each segment sent?

(1st screenshot depicts the time for segment numbers 140, 141, 142 and 2nd for 174)

Segment Number	Time at which the segment was sent
140	4.081502
141	4.081503
142	4.081603
174	4.570353



(iii) When was the ACK for each segment received?

(See the highlighted boxes in the screenshots below)

Received time for ACK

4.135363

4.135713

4.135716

4.6468

No.	Time	Source	Destination	Protocol	Length	Info
139	4.081018	172.20.10.2	299.105.248.3	TCP	66	55806 - 443 [ACK] Seq=1 Ack=1 Win=131840 Len=0 Tsvl=10587089
140	4.081508	172.20.10.2	23.235.44.231	TCP	1454	55798 - 88 [ACK] Seq=1415 Ack=56130 Win=4096 Len=1388 Tsvl=1
141	4.081503	172.20.10.2	23.235.44.231	TCP	261	55798 - 88 [PSH, ACK] Seq=2803 Ack=56130 Win=4096 Len=135 Tsvl=1
142	4.081603	172.20.10.2	23.235.44.231	HTTP	75	POST /vs/page/hotel/results HTTP/1.1 (application/x-www-form
143	4.081603	172.20.10.2	299.105.248.3	TLSv1	583	Client Hello
144	4.081603	172.20.10.2	299.105.248.3	TCP	66	88 - 55798 [ACK] Seq=56130 Ack=2803 Win=72 Len=0 Tsvl=512372
145	4.081603	172.20.10.2	299.105.248.3	TCP	66	88 - 55798 [ACK] Seq=56130 Ack=2938 Win=77 Len=0 Tsvl=512372
146	4.081603	172.20.10.2	299.105.248.3	TLSv1	211	Server Hello, Change Cipher Spec, Encrypted Handshake Message
147	4.081603	172.20.10.2	299.105.248.3	TCP	66	55806 - 443 [ACK] Seq=518 Ack=146 Win=131712 Len=0 Tsvl=1058
148	4.081603	172.20.10.2	299.105.248.3	TLSv1	72	Change Cipher Spec
149	4.081603	172.20.10.2	299.105.248.3	TLSv1	119	Encrypted Handshake Message
150	4.081603	172.20.10.2	299.105.248.3	TLSv1	583	Application Data
151	4.081603	172.20.10.2	299.105.248.3	TCP	1016	88 - 55798 [PSH, ACK] Seq=5152 Ack=3138 Win=90 Len=950 Tsvl=
152	4.081603	172.20.10.2	299.105.248.3	HTTP	86	HTTP/1.1 302 Found (text/plain) (text/plain)
153	4.081603	172.20.10.2	299.105.248.3	TCP	66	55798 - 88 [ACK] Seq=3138 Ack=6102 Win=4066 Len=0 Tsvl=10587
154	4.081603	172.20.10.2	299.105.248.3	TCP	66	55798 - 88 [ACK] Seq=3138 Ack=6122 Win=4065 Len=0 Tsvl=10587
155	4.081603	172.20.10.2	299.105.248.3	TCP	1454	55798 - 88 [ACK] Seq=3138 Ack=6122 Win=4096 Len=1388 Tsvl=10
156	4.081603	172.20.10.2	299.105.248.3	HTTP	243	GET /s/cmp2ch/open?searchId=d0eAqPraE HTTP/1.1
157	4.081603	172.20.10.2	299.105.248.3	TCP	66	88 - 55798 [ACK] Seq=56130 Ack=2803 Len=0 Tsvl=512372
158	4.081603	172.20.10.2	299.105.248.3	TCP	177	... x x x ...
159	4.081603	172.20.10.2	299.105.248.3	TCP	177	... x x x ...
160	4.081603	172.20.10.2	299.105.248.3	TCP	177	... x x x ...
161	4.081603	172.20.10.2	299.105.248.3	TCP	177	... x x x ...
162	4.081603	172.20.10.2	299.105.248.3	TCP	177	... x x x ...
163	4.081603	172.20.10.2	299.105.248.3	TCP	177	... x x x ...
164	4.081603	172.20.10.2	299.105.248.3	TCP	177	... x x x ...
165	4.081603	172.20.10.2	299.105.248.3	TCP	177	... x x x ...
166	4.081603	172.20.10.2	299.105.248.3	TCP	177	... x x x ...
167	4.081603	172.20.10.2	299.105.248.3	TCP	177	... x x x ...
168	4.081603	172.20.10.2	299.105.248.3	TCP	177	... x x x ...
169	4.081603	172.20.10.2	299.105.248.3	TCP	177	... x x x ...
170	4.081603	172.20.10.2	299.105.248.3	TCP	177	... x x x ...
171	4.081603	172.20.10.2	299.105.248.3	TCP	177	... x x x ...
172	4.081603	172.20.10.2	299.105.248.3	TCP	177	... x x x ...
173	4.081603	172.20.10.2	299.105.248.3	TCP	177	... x x x ...
174	4.081603	172.20.10.2	299.105.248.3	TCP	177	... x x x ...
175	4.081603	172.20.10.2	299.105.248.3	HTTP	175	GET /s/run/recentsearchhistory/gethistory?searchType=search&url=
176	4.081603	172.20.10.2	299.105.248.3	HTTP	85	POST /s/run/hotelbookmsg HTTP/1.1 (application/x-www-form
177	4.081603	172.20.10.2	299.105.248.3	TLSv1.2	352	Application Data
178	4.081603	172.20.10.2	299.105.248.3	DNS	74	Standard query 0xcb2 A www.google.com
179	4.081603	172.20.10.2	299.105.248.3	TCP	74	443 - 55887 [SYN, ACK] Seq=0 Win=65535 Len=0 MSS=1460 WS=32 Tsvl=
180	4.081603	172.20.10.2	299.105.248.3	TCP	66	55887 - 443 [ACK] Seq=4783 Ack=7633 Win=4096 Len=1388 Tsvl=10
181	4.081603	172.20.10.2	299.105.248.3	DNS	154	Standard query response 0xcb2 A www.google.com A 173.194.115
182	4.081603	172.20.10.2	299.105.248.3	TCP	78	55888 - 88 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=32 Tsvl=1
183	4.081603	172.20.10.2	299.105.248.3	TLSv1.2	300	Client Hello
184	4.081603	172.20.10.2	299.105.248.3	TCP	66	88 - 55798 [ACK] Seq=67633 Ack=6891 Win=106 Len=0 Tsvl=17518
185	4.081603	172.20.10.2	299.105.248.3	TCP	66	88 - 55798 [ACK] Seq=67633 Ack=6235 Win=112 Len=0 Tsvl=17518
186	4.081603	172.20.10.2	299.105.248.3	TCP	66	88 - 55798 [ACK] Seq=56939 Ack=4335 Win=82 Len=0 Tsvl=512373
187	4.081603	172.20.10.2	299.105.248.3	TCP	66	443 - 55496 [SEQ] Seq=1 Ack=287 Win=455 Len=0 Tsvl=241802518
188	4.081603	172.20.10.2	299.105.248.3	TCP	66	88 - 55798 [ACK] Seq=56939 Ack=4444 Win=82 Len=0 Tsvl=512373
189	4.081603	172.20.10.2	299.105.248.3	TCP	66	88 - 55798 [ACK] Seq=56939 Ack=4444 Win=82 Len=0 Tsvl=512373
190	4.081603	172.20.10.2	299.105.248.3	TCP	66	88 - 55798 [ACK] Seq=56939 Ack=4444 Win=82 Len=0 Tsvl=512373
191	4.081603	172.20.10.2	299.105.248.3	TCP	66	88 - 55798 [ACK] Seq=56939 Ack=4444 Win=82 Len=0 Tsvl=512373
192	4.081603	172.20.10.2	299.105.248.3	TCP	66	88 - 55798 [ACK] Seq=56939 Ack=4444 Win=82 Len=0 Tsvl=512373
193	4.081603	172.20.10.2	299.105.248.3	TCP	66	88 - 55798 [ACK] Seq=56939 Ack=4444 Win=82 Len=0 Tsvl=512373
194	4.081603	172.20.10.2	299.105.248.3	TCP	66	88 - 55798 [ACK] Seq=56939 Ack=4444 Win=82 Len=0 Tsvl=512373
195	4.081603	172.20.10.2	299.105.248.3	TCP	66	88 - 55798 [ACK] Seq=56939 Ack=4444 Win=82 Len=0 Tsvl=512373
196	4.081603	172.20.10.2	299.105.248.3	TCP	66	88 - 55798 [ACK] Seq=56939 Ack=4444 Win=82 Len=0 Tsvl=512373
197	4.081603	172.20.10.2	299.105.248.3	TCP	66	88 - 55798 [ACK] Seq=56939 Ack=4444 Win=82 Len=0 Tsvl=512373
198	4.081603	172.20.10.2	299.105.248.3	TCP	66	88 - 55798 [ACK] Seq=56939 Ack=4444 Win=82 Len=0 Tsvl=512373
199	4.081603	172.20.10.2	299.105.248.3	TCP	66	88 - 55798 [ACK] Seq=56939 Ack=4444 Win=82 Len=0 Tsvl=512373
200	4.081603	172.20.10.2	299.105.248.3	TCP	66	88 - 55798 [ACK] Seq=56939 Ack=4444 Win=82 Len=0 Tsvl=512373
201	4.081603	172.20.10.2	299.105.248.3	TCP	66	88 - 55798 [ACK] Seq=56939 Ack=4444 Win=82 Len=0 Tsvl=512373
202	4.081603	172.20.10.2	299.105.248.3	TCP	66	88 - 55798 [ACK] Seq=56939 Ack=4444 Win=82 Len=0 Tsvl=512373
203	4.081603	172.20.10.2	299.105.248.3	TCP	66	88 - 55798 [ACK] Seq=56939 Ack=4444 Win=82 Len=0 Tsvl=512373
204	4.081603	172.20.10.2	299.105.248.3	TCP	66	88 - 55798 [ACK] Seq=56939 Ack=4444 Win=82 Len=0 Tsvl=512373
205	4.081603	172.20.10.2	299.105.248.3	TCP	66	88 - 55798 [ACK] Seq=56939 Ack=4444 Win=82 Len=0 Tsvl=512373
206	4.081603	172.20.10.2	299.105.248.3	TCP	66	88 - 55798 [ACK] Seq=56939 Ack=4444 Win=82 Len=0 Tsvl=512373
207	4.081603	172.20.10.2	299.105.248.3	TCP	66	88 - 55798 [ACK] Seq=56939 Ack=4444 Win=82 Len=0 Tsvl=512373
208	4.081603	172.20.10.2	299.105.248.3	TCP	66	88 - 55798 [ACK] Seq=56939 Ack=4444 Win=82 Len=0 Tsvl=512373
209	4.081603	172.20.10.2	299.105.248.3	TCP	66	88 - 55798 [ACK] Seq=56939 Ack=4444 Win=82 Len=0 Tsvl=512373
210	4.081603	172.20.10.2	299.105.248.3	TCP	66	88 - 55798 [ACK] Seq=56939 Ack=4444 Win=82 Len=0 Tsvl=512373
211	4.081603	172.20.10.2	299.105.248.3	TCP	66	88 - 55798 [ACK] Seq=56939 Ack=4444 Win=82 Len=0 Tsvl=512373
212	4.081603	172.20.10.2	299.105.248.3	TCP	66	88 - 55798 [ACK] Seq=56939 Ack=4444 Win=82 Len=0 Tsvl=512373
213	4.081603	172.20.10.2	299.105.248.3	TCP	66	88 - 55798 [ACK] Seq=56939 Ack=4444 Win=82 Len=0 Tsvl=512373
214	4.081603	172.20.10.2	299.105.248.3	TCP	66	88 - 55798 [ACK] Seq=56939 Ack=4444 Win=82 Len=0 Tsvl=512373
215	4.081603	172.20.10.2	299.105.248.3	TCP	66	88 - 55798 [ACK] Seq=56939 Ack=4444 Win=82 Len=0 Tsvl=512373
216	4.081603	172.20.10.2	299.105.248.3	TCP	66	88 - 55798 [ACK] Seq=56939 Ack=4444 Win=82 Len=0 Tsvl=512373
217	4.081603	172.20.10.2	299.105.248.3	TCP	66	88 - 55798 [ACK] Seq=56939 Ack=4444 Win=82 Len=0 Tsvl=512373
218	4.081603	172.20.10.2	299.105.248.3	TCP	66	88 - 55798 [ACK] Seq=56939 Ack=4444 Win=82 Len=0 Tsvl=512373
219	4.081603	172.20.10.2	299.105.248.3	TCP	66	88 - 55798 [ACK] Seq=56939 Ack=4444 Win=82 Len=0 Tsvl=512373
220	4.081603	172.20.10.2	299.105.248.3	TCP	66	88 - 55798 [ACK] Seq=56939 Ack=4444 Win=82 Len=0 Tsvl=512373
221	4.081603	172.20.10.2	299.105.248.3	TCP	66	88 - 55798 [ACK] Seq=56939 Ack=4444 Win=82 Len=0 Tsvl=512373
222	4.081603	172.20.10.2	299.105.248.3	TCP	66	88 - 55798 [ACK] Seq=56939 Ack=4444 Win=82 Len=0 Tsvl=512373
223	4.081603	172.20.10.2	299.105.248.3	TCP	66	88 - 55798 [ACK] Seq=56939 Ack=4444 Win=82 Len=0 Tsvl=512373
224	4.081603	172.20.10.2	299.105.248.3	TCP	66	88 - 55798 [ACK] Seq=56939 Ack=4444 Win=82 Len=0 Tsvl=512373
225	4.081603	172.20.10.2	299.105.248.3	TCP	66	88 - 55798 [ACK] Seq=56939 Ack=4444 Win=82 Len=0 Tsvl=512373
226	4.081603	172.20.10.2	299.105.248.3	TCP	66	88 - 55798 [ACK] Seq=56939 Ack=4444 Win=82 Len=0 Tsvl=512373
227	4.081603	172.20.10.2	299.105.248.3	TCP	66	88 - 55798 [ACK] Seq=56939 Ack=4444 Win=82 Len=0 Tsvl=512373
228	4.081603	172.20.10.2	299.105.248.3	TCP	66	88 - 55798 [ACK] Seq=56939 Ack=4444 Win=82 Len=0 Tsvl=512373
229	4.081603	172.20.10.2	299.105.248.3	TCP	66	88 - 55798 [ACK] Seq=56939 Ack=4444 Win=82 Len=0 Tsvl=512373
230	4.081603	172.20.10.2	299.105.248.3	TCP	66	88 - 55798 [ACK] Seq=56939 Ack=4444 Win=82 Len=0 Tsvl=512373
231	4.081603	172.20.10.2	299.105.248.3	TCP	66	88 - 55798 [ACK] Seq=56939 Ack=4444 Win=82 Len=0 Tsvl=512373
232	4.081603	172.20.10.2	299.105.248.3	TCP	66	88 - 55798 [ACK] Seq=56939 Ack=4444 Win=82 Len=0 Tsvl=512373
233	4.081603	172.20.10.2	299.105.248.3	TCP	66	88 - 55798 [ACK] Seq=56939 Ack=4444 Win=82 Len=0 Tsvl=512373
234	4.081603	172.20.10.2	299.105.248.3	TCP	66	88 - 55798 [ACK] Seq=56939 Ack=4444 Win=82 Len=0 Tsvl=512373
235	4.081603	172.20.10.2	299.105.248.3	TCP	66	88 - 55798 [ACK] Seq=56939 Ack=4444 Win=82 Len=0 Tsvl=512373
236	4.081603	172.20.10.2	299.105.248.3	TCP	66	88 - 55798 [ACK] Seq=56939 Ack=4444 Win=82 Len=0 Tsvl=512373
237	4.081603	172.20.10.2	299.105.248.3	TCP	66	88 - 55798 [ACK] Seq=56939 Ack=4444 Win=82

(iv) Given the difference between when each TCP segment was sent, and when its acknowledgement was received,

(v) what is the RTT value for each of the four segments?

(vi) What is the EstimatedRTT value (see Section 3.5.3, page 239 in text) after the receipt of each ACK?

(vii) Assume that the value of the EstimatedRTT is equal to the measured RTT for the first segment, and then is computed using the EstimatedRTT equation on page 239 for all subsequent segments

Ans for iv, v, vi and vii is – (tabular form)

The Estimated RTT is to be calculated by considering the previous values also, so for the segments S1, S2, S3 and S4 the values will be (According to the TextBook,)

Estimating the Round-Trip Time

Let's begin our study of TCP timer management by considering how TCP estimates the round-trip time between sender and receiver. This is accomplished as follows. The sample RTT, denoted `SampleRTT`, for a segment is the amount of time between when the segment is sent (that is, passed to IP) and when an acknowledgement for the segment is received. Instead of measuring a `SampleRTT` for every transmitted segment, most TCP implementations take only one `SampleRTT` measurement at a time. That is, at any point in time, the `SampleRTT` is being estimated for one segment. This is done because it is better to have one good estimate than a new value of `SampleRTT` approximately once every RTT. Also, TCP never computes a `SampleRTT` for a segment that has been retransmitted; it only measures `SampleRTT` for segments that have been transmitted once [Karn 1987]. (A problem at the end of the chapter asks you to consider why.)

Obviously, the `SampleRTT` values will fluctuate from segment to segment due to congestion in the routers and due to varying load on the end systems. Because of this fluctuation, any given `SampleRTT` value may be atypical. In order to estimate a typical RTT, TCP maintains a weighted average of the `SampleRTT`s, called `EstimatedRTT`. TCP maintains an average, called `EstimatedRTT`, of the `SampleRTT` values. Upon obtaining a new `SampleRTT`, TCP updates `EstimatedRTT` according to the following formula:

$$\text{EstimatedRTT} = (1 - \alpha) \cdot \text{EstimatedRTT} + \alpha \cdot \text{SampleRTT}$$

The formula above is written in the form of a programming-language statement—the new value of `EstimatedRTT` is a weighted combination of the previous value of `EstimatedRTT` and the new value for `SampleRTT`. The recommended value of α is $\alpha = 0.125$ (that is, $1/8$) [RFC 6298], in which case the formula above becomes:

$$\text{EstimatedRTT} = 0.875 \cdot \text{EstimatedRTT} + 0.125 \cdot \text{SampleRTT}$$

Note that `EstimatedRTT` is a weighted average of the `SampleRTT` values. As discussed in a homework problem at the end of this chapter, this weighted average puts more weight on recent samples than on old samples. This is natural, as the

NOW, RTT is calculated as (Received time for ACK – Time at which the segment was sent)

$$\text{So, } 4.135363 - 4.081502 = 0.053861$$

$$4.135713 - 4.081503 = 0.05421$$

$$4.135716 - 4.081603 = 0.054113$$

$$4.646868 - 4.570353 = 0.076515 \text{ will be the estimated RTT values.}$$

(Assuming the sample rtt to be the measured rtt, as given in the question) THE ESTIMATED RTT WILL BE –

segment1 (s1) → at first it will be the RTT value of the S1 itself = 0.05386 seconds

$$s2 \rightarrow 0.875 * 0.053861 + 0.125 * 0.05421 = 0.05390 \text{ seconds}$$

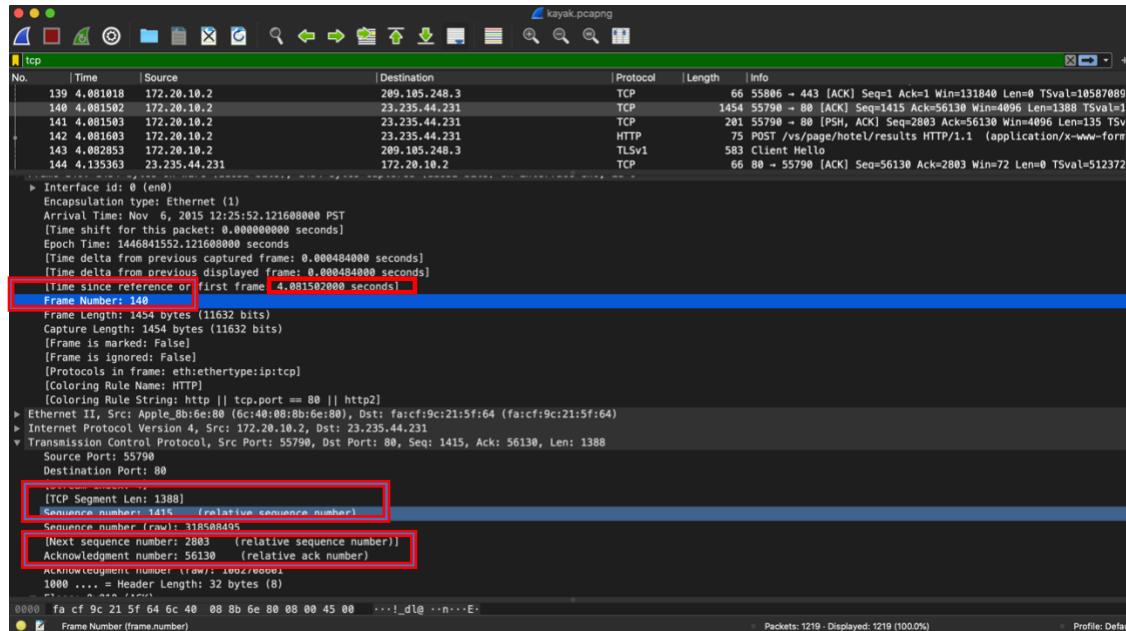
$$s3 \rightarrow 0.875 * 0.0539 + 0.125 * 0.054113 = 0.05392 \text{ seconds}$$

$$s4 \rightarrow 0.875 * 0.05392 + 0.125 * 0.0765 = 0.0567 \text{ seconds}$$

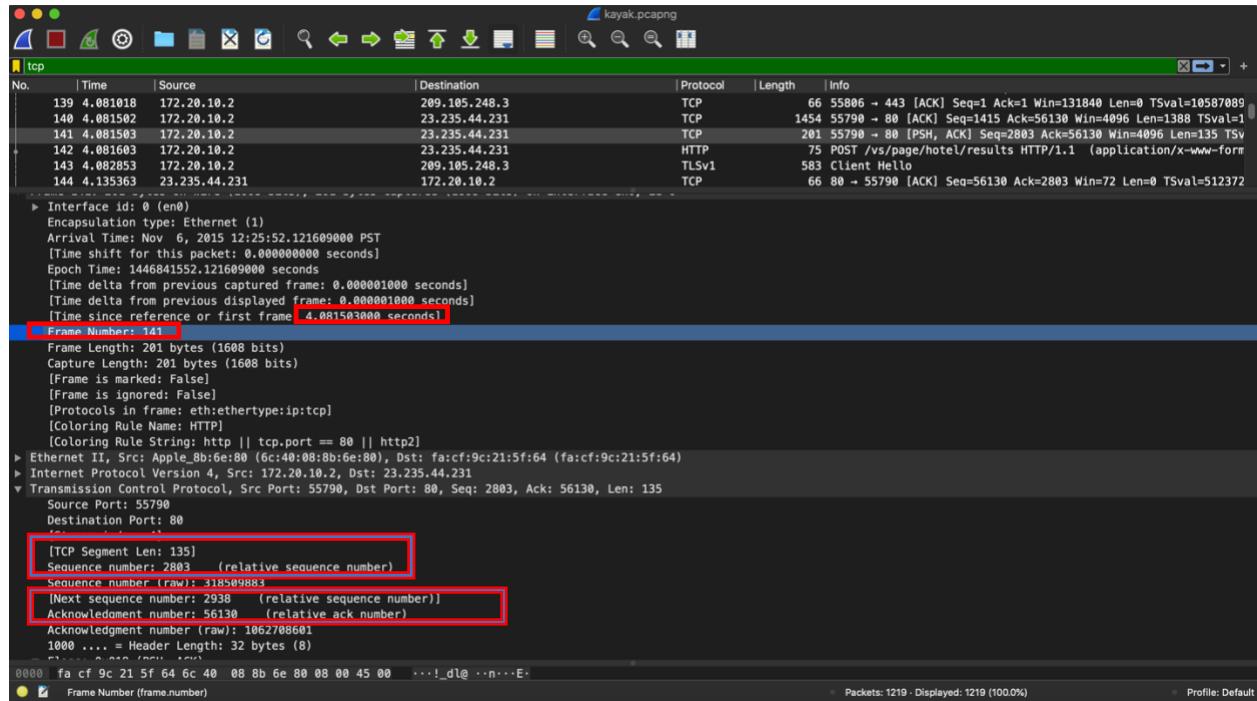
Received time for ACK	RTT (seconds) Value	Estimated RTT Value
4.135363	0.053861	0.05386
4.135713	0.05421	0.05390
4.135716	0.054113	0.05392
4.646868	0.076515	0.0567

Below are the screenshots – (2q)

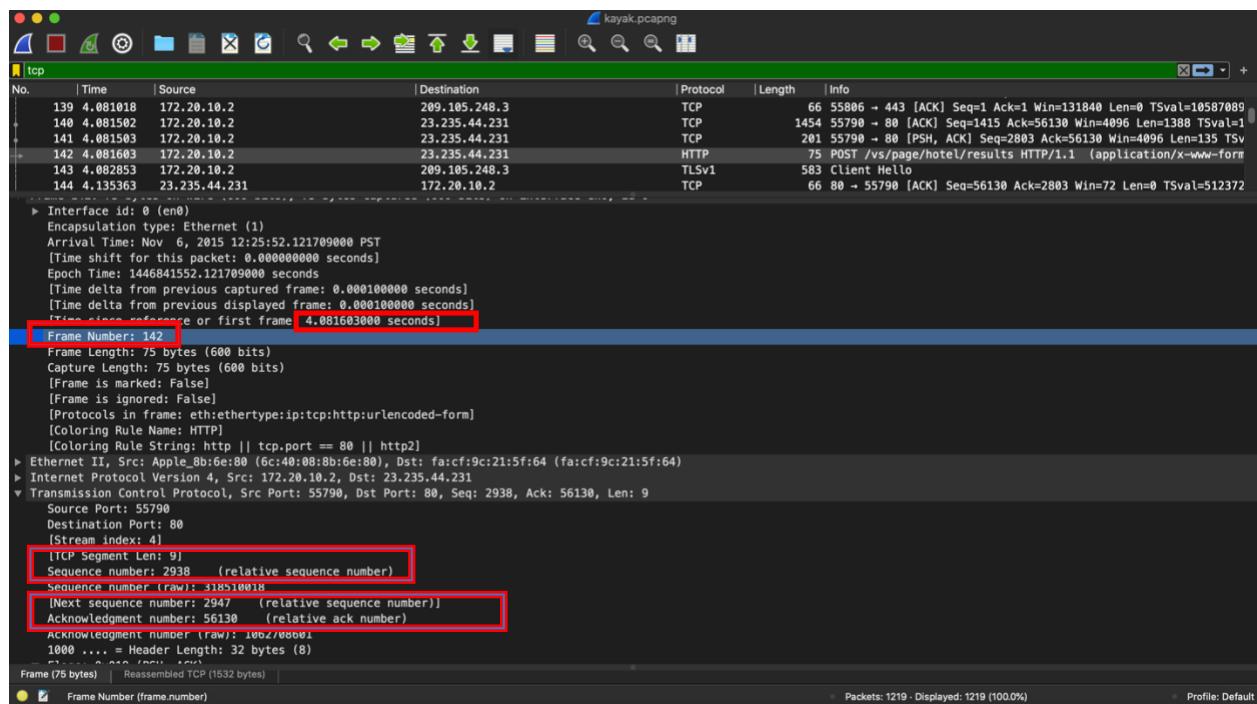
The Segment 140



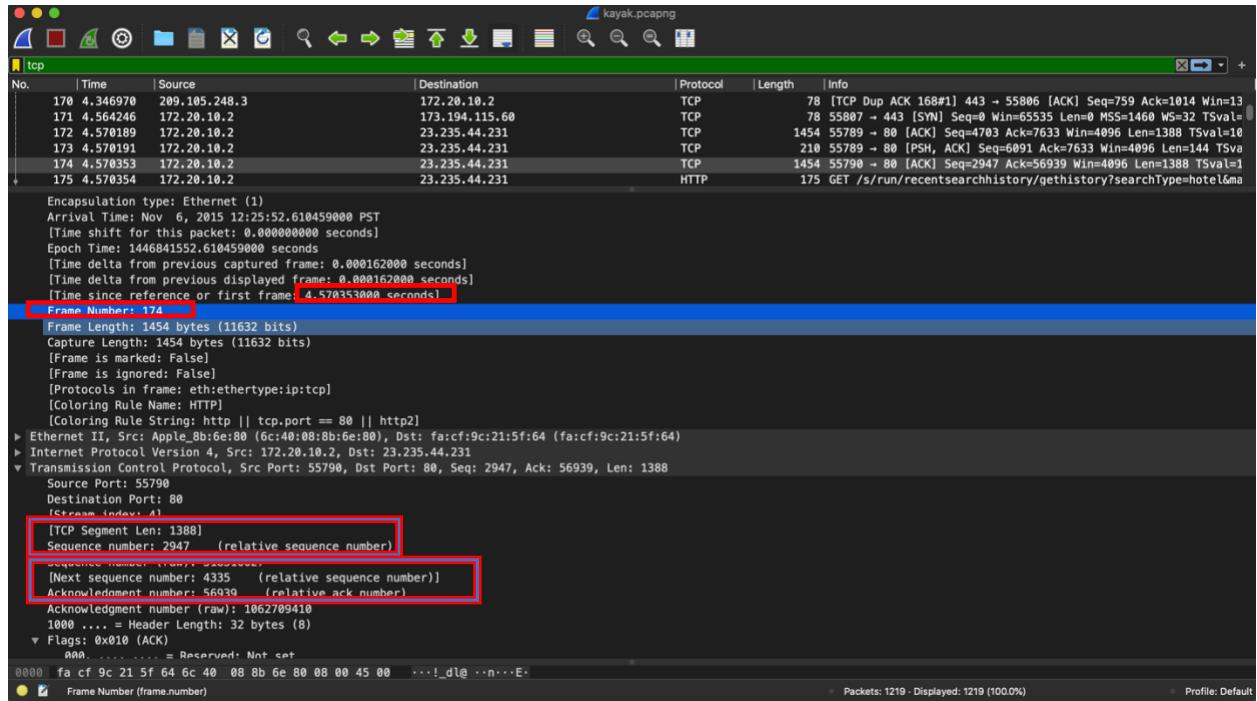
The Segment 141



The Segment 142



The Segment 174



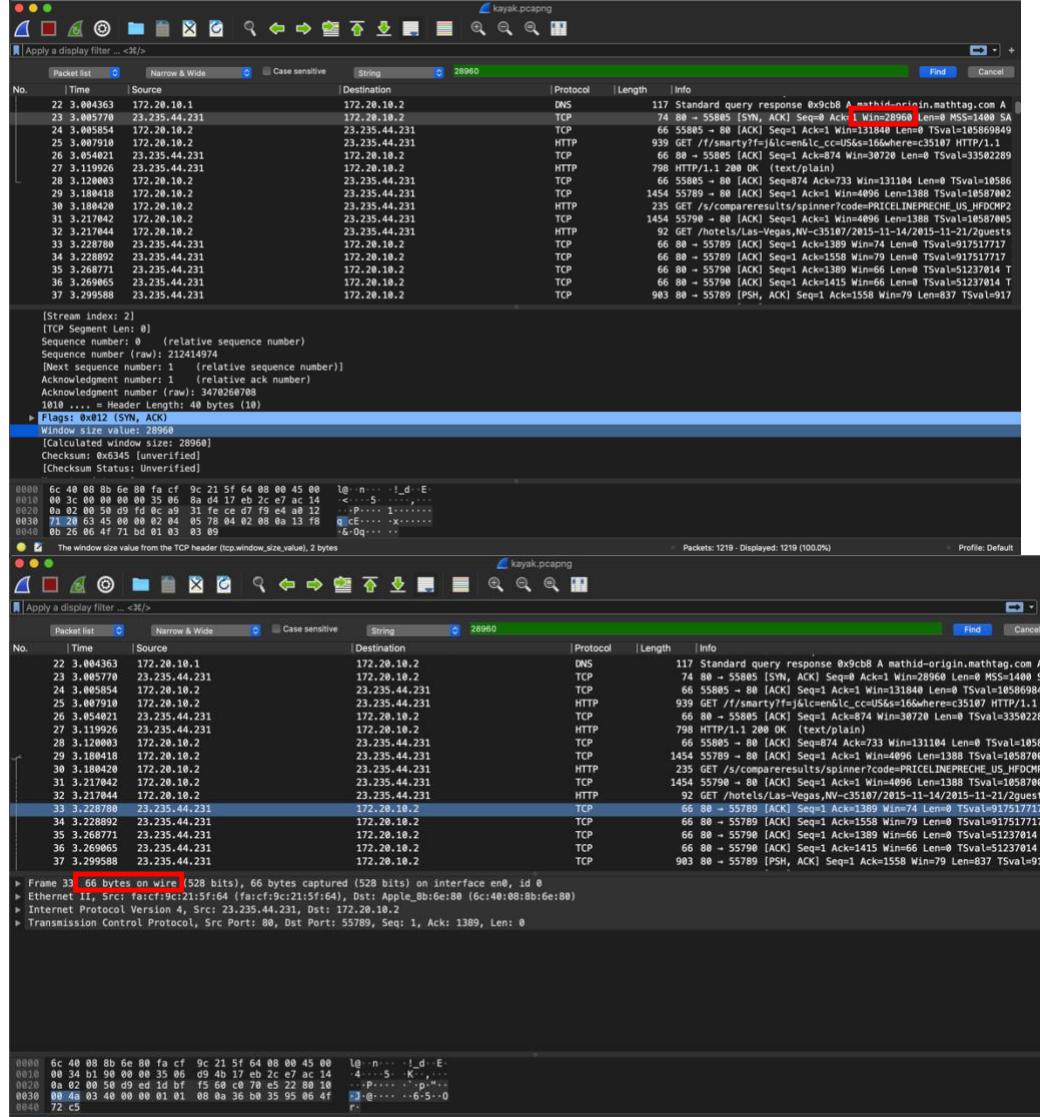
3) What is the length of each of the first four TCP Segments?

(The Segment lengths and numbers have also been highlighted in the above screenshots)

Segment Number	Segment Length
140	1388
141	135
142	9
174	1388

4) What is the minimum amount of available buffer space advertised at the receiver for the entire trace?

Ans) Here as highlighted, in the frame 23 it's the trace packet in the list with [SYN, ACK] respective source and destination address with window size 28960 in total. And with another frame as mentioned in the next screenshot with [ACK] the length is 66.



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

Ans) No, lack of receiver buffer space **will not** ever throttle the sender since it isn't possible to approach to close for the buffer size.

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

Ans) By checking and counting the retransmitted segments filter in the kayak trace. By which it can be said that YES!, retransmitted segments in the trace file exists, which are eight of them and can be found out by –

(press cmd + f or ctrl + f then type “Retransmission”(case sensitive R capital letter) then search in the packet list as shown.). So you can find eight of them.

The screenshot shows a network traffic analysis interface with the following details:

- Toolbar:** Includes icons for file operations (New, Open, Save, Import, Export, Find, Copy, Paste, etc.), zoom, and search.
- Header:** "kayak.pcapng" is displayed at the top center.
- Session List:** A table titled "tcp" showing session details. The columns are: No., Time, Source, Destination, Protocol, Length, and a detailed view column.
- Detailed View:** A large pane on the right displays detailed information for selected sessions, such as flags, window size, and checksums.
- Bottom Panel:** Shows raw hex and ASCII data for the selected session.

Apply a display filter: ...<6>/

No.	Time	Source	Destination	Protocol	Length	Info
160	4.210628	23.235.44.231	172.20.10.2	TCP	66	88 - 55789 [ACK] Seq=6122 Ack=4526 Win=96 Len=0 TSval=91751796
161	4.210628	23.235.44.231	172.20.10.2	TCP	66	88 - 55789 [ACK] Seq=6122 Ack=4703 Win=101 Len=0 TSval=91751796
162	4.222273	23.235.44.231	172.20.10.2	TCP	503	[TCP Retransmission] 55806 - 443 [PSH, ACK] Seq=577 Ack=146
163	4.300401	172.20.10.2	209.105.248.3	TCP	9/5 88 - 55789 [PSH, ACK] Seq=b122 Ack=4783 Win=101 Len=0 TSval=105871	
164	4.307809	23.235.44.231	172.20.10.2	TCP	66	55789 - 88 [ACK] Seq=4703 Ack=7031 Win=4067 Len=0 TSval=105871
165	4.307062	172.20.10.2	23.235.44.231	TCP	66	55789 - 88 [ACK] Seq=4703 Ack=7031 Win=4067 Len=0 TSval=105871
166	4.307230	23.235.44.231	172.20.10.2	HTTP	666	HTTP/1.1 200 OK (application/json)
167	4.307258	172.20.10.2	23.235.44.231	TCP	66	55789 - 88 [ACK] Seq=4703 Ack=7633 Win=4077 Len=0 TSval=105871
168	4.335350	209.105.248.3	172.20.10.2	TLSv1	679	Application Data
169	4.335424	172.20.10.2	209.105.248.3	TCP	66	55806 - 443 [ACK] Seq=1014 Ack=759 Win=131072 Len=0 TSval=105871
170	4.346970	209.105.248.3	172.20.10.2	TCP	78	[TCP Dup ACK 158#1] 443 - 55806 [ACK] Seq=759 Ack=1014 Win=131
171	4.564246	172.20.10.2	173.194.115.60	TCP	78	55807 - 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=32 TSval=1
172	4.570189	172.20.10.2	23.235.44.231	TCP	1454	55789 - 88 [ACK] Seq=4703 Ack=7633 Win=4096 Len=1388 TSval=105871
173	4.570191	172.20.10.2	23.235.44.231	TCP	210	55789 - 88 [PSH, ACK] Seq=0#1 Ack=7633 Win=4096 Len=144 TSval=105871
174	4.570353	172.20.10.2	23.235.44.231	TCP	1454	55790 - 88 [ACK] Seq=2947 Ack=56939 Win=4096 Len=1388 TSval=105871

Internet Protocol Version 4, Src: 172.20.10.2, Dst: 209.105.248.3

Transmission Control Protocol, Src Port: 55806, Dst Port: 443, Seq: 577, Ack: 146, Len: 437

Source Port: 55806
Destination Port: 443
[Stream index: 5]
[TCP Segment Len: 437]
Sequence number: 577 (relative sequence number)
Sequence number (raw): 12731616332
[Next sequence number: 1014 (relative sequence number)]
Acknowledgment number: 146 (relative ack number)
Acknowledgment number (raw): 408690151
1000 = Header Length: 32 bytes (8)

Window size value: 4116
[Calculated window size: 131722]
[Window size scaling factor: 32]

Checksum: 0x809c [Unverified]
[Checksum Status: Unverified]
Urgent pointer: 0

Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps

[SEQ/ACK analysis]

0030 10 14 76 9c 00 00 01 01 08 00 06 4f 77 0b 05 de ::v----- 0w--

The window size value from the TCP header (tcp.window.size.value), 2 bytes

Packets: 1219 - Displayed: 1219 (100.0%)

Profile: Default

7) How much data does the receiver typically acknowledge in an ACK? Can you identify cases where the receiver is ACK-ing every other received segment (see Table 3.2 on page 247 in the text).

The ACK data is given by the difference between the current ACK no. and previous ACK no.

Ans) Segment Number	Acknowledgement number	Acknowledged data in bytes
140	2803	0 (initial)
141	2938	135 (2938 - 2803)
142	2947	9 (2947 - 2938)
174	4335	1388 (4335 – 2947)

8) What is the throughput (bytes transferred per unit time) for the TCP connection (Just consider a single connection)? Think on how to calculate the throughput!

Ans) The ratio of quantity of transferred data to the whole transmission time is termed as the average throughput for a TCP connection. So to find the data that has been transferred it is given by the difference between sequence number of first TCP and the ACK no. of the last ACK. Therefore, it is given by, Throughput = (quantity of transferred data)/(total time) $\rightarrow 2920/0.56536 = 5164.79$ kilobytes per second

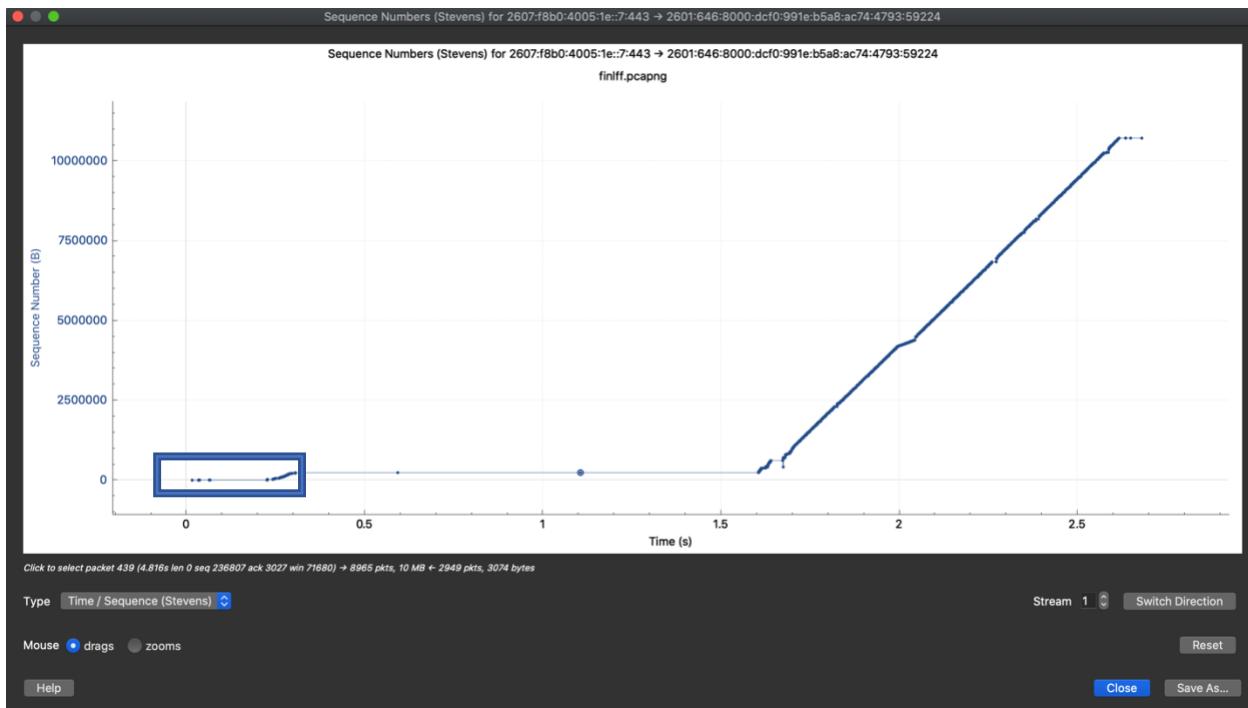
9) Explain how you calculated this value.

Ans) Initially as mentioned, the data that has been transferred it is given by the difference between sequence number of first TCP, i.e 140 and the ACK no. of the last ACK, i.e 186. So, it will be the difference of 4335 and 1415, which is 2920. Now for the time, it's the subtraction of first byte time sent and last ACK time, i.e difference of 4.64686 and 4.08150, which is 0.56536

Problem Set 4

1) Can you identify where TCP's slow-start phase begins and ends?

Ans) The TCP slow start phase begins at 0.016 and ends at 0.225



2) Where congestion avoidance takes over?

Ans) It takes over after when slow start phase finishes, so it'll be 0.225.

3) Comment on ways in which the measured data differs from the idealized behavior of TCP that we've studied in the text.

Ans) The main issue arises when there's a lot of traffic in the network, i.e. congestion, so in order to solve this the TCP protocol will have to work on its properties to reduce the size of the window as stated. But there are also numerous issues like sending the packets again, loss of the packets and delay in time. As TCP is capable of working on its properties like slow start and controlling the congestion, it's one of the plus point to it. The cause behind the objects setback or slowing down is because of the slow start stage and also the objects size is said to be small. And the communication will be terminated just before the slow start stage end.