



LAB-2

DATA COMMUNICATIONS NETWORKS

Prof. Dr. Karim Banawan - Prof. Dr. Noha ElKorany
Communication & Electronics Department

Asmaa Gamal Abdel-Halem Mabrouk Nagy
أسماء جمال عبد الحليم مبروك ناجي
15010473 - section 8

Wireshark Lab: TCP v6.0

By: "Computer Networking: A Top Down Approach, 6th edition"



Section 1 : Capturing a bulk TCP transfer from your computer to a remote Server

The Task:



ALICE'S ADVENTURES IN WONDERLAND

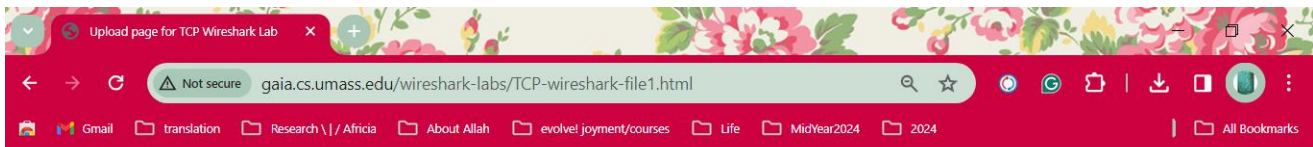
Lewis Carroll

THE MILLENNIUM FULCRUM EDITION 3.0

CHAPTER I

Down the Rabbit-Hole

Alice was beginning to get very tired of sitting by her sister on the bank, and of having nothing to do: once or twice she had peeped into the book her sister was reading, but it had no pictures or conversations in it, 'and what is the use of a book,' thought Alice 'without pictures or conversation?'



Upload page for TCP Wireshark Lab

Computer Networking: A Top Down Approach, 6th edition

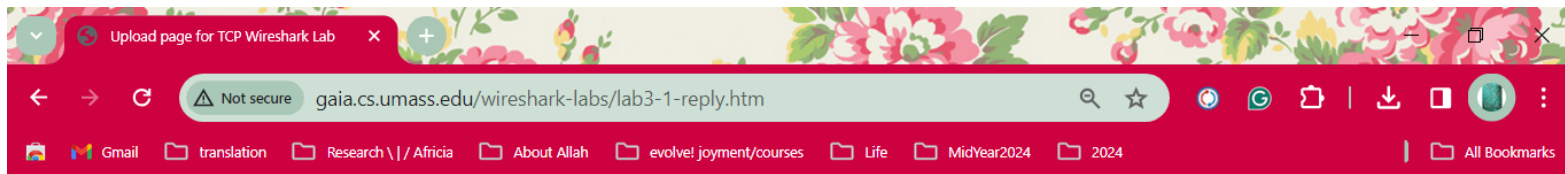
Copyright 2012 J.F. Kurose and K.W. Ross, All Rights Reserved

If you have followed the instructions for the TCP Wireshark Lab, you have *already* downloaded an ASCII copy of Alice and Wonderland from <http://gaia.cs.umass.edu/wireshark-labs/alice.txt> and you also *already* have the Wireshark packet sniffer running and capturing packets on your computer.

Click on the Browse button below to select the directory/file name for the copy of alice.txt that is stored on your computer.

No file chosen

Once you have selected the file, click on the "Upload alice.txt file" button below. This will cause your browser to send a copy of alice.txt over an HTTP connection (using TCP) to the web server at gaia.cs.umass.edu. After clicking on the button, wait until a short message is displayed indicating the the upload is complete. Then stop your Wireshark packet sniffer - you're ready to begin analyzing the TCP transfer of alice.txt from your computer to gaia.cs.umass.edu!!



Congratulations!

You've now transferred a copy of alice.txt from your computer to gaia.cs.umass.edu. You should now stop Wireshark packet capture. It's time to start analyzing the captured Wireshark packets!

Section 2: A first look at the captured trace:

The Questions & Answers:

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?

The book author's client computer (source):

Source IP Address: 192.168.1.102

Source TCP port number: 1161

The image shows a Wireshark packet capture of an HTTP POST request. The packet list shows a packet from 192.168.1.102 to 128.119.245.12 on port 80. The packet details show the source IP address as 192.168.1.102 and the source port as 1161. A green arrow points from the source IP in the packet list to the source IP in the packet details. A red box highlights the source port 1161 in the packet details.

No.	Time	Source	Destination	Protocol	Length	Info
199	16:44:25.867722	192.168.1.102	128.119.245.12	HTTP	104	POST /ethereal-labs/lab3-1-reply.htm HTTP/1.1 (text/plain)
203	16:44:26.031556	128.119.245.12	192.168.1.102	HTTP	784	HTTP/1.1 200 OK (text/html)

Frame 199: 104 bytes on wire (832 bits), 104 bytes captured (832 bits) on interface ...
Ethernet II, Src: ActiontecEle_8a:70:1a (00:20:00:8a:70:1a), Dst: LinksysGroup_da:af:73 (00:06:25:da:af:73)
Internet Protocol Version 4, Src: 192.168.1.102, Dst: 128.119.245.12
0100 = Version: 4
.... 0101 = Header Length: 20 bytes (5)
> Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
Total Length: 90
Identification: 0x1e9a (7834)
> 010. = Flags: 0x2, Don't fragment
...0 0000 0000 0000 = Fragment Offset: 0
Time to Live: 128
Protocol: TCP (6)
Header Checksum: 0xa471 [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: 164041, Ack: 1, Len: 50
Source Port: 1161
Destination Port: 80
[Stream index: 0]

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?

Destination Server: gaia.cs.umass.edu

Destination IP Address: 128.119.245.12

Destination TCP port number: 80

The image shows a Wireshark packet capture of an HTTP POST request. The packet list shows a packet from 192.168.1.5 to 128.119.245.12 on port 80. The packet details show the destination IP address as 128.119.245.12 and the destination port as 80. A green arrow points from the destination IP in the packet list to the destination IP in the packet details. A red box highlights the destination port 80 in the packet details.

No.	Time	Source	Destination	Protocol	Length	Info
283	00:28:54.856051	192.168.1.5	128.119.245.12	HTTP	1229	POST /wireshark-labs/lab3-1-reply.htm HTTP/1.1 (text/plain)
333	00:28:55.075631	128.119.245.12	192.168.1.5	HTTP	831	HTTP/1.1 200 OK (text/html)

Frame 283: 1229 bytes on wire (9832 bits), 1229 bytes captured (9832 bits) on interface ...
Ethernet II, Src: Intel_db:e3:45 (80:00:0b:db:e3:45), Dst: TpLinkTechno_6c:42:46 (cc:32:e5:6c:42:46)
Internet Protocol Version 4, Src: 192.168.1.5, Dst: 128.119.245.12
0100 = Version: 4
.... 0101 = Header Length: 20 bytes (5)
> Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
Total Length: 1215
Identification: 0xe0a2 (57506)
> 010. = Flags: 0x2, Don't fragment
...0 0000 0000 0000 = Fragment Offset: 0
Time to Live: 128
Protocol: TCP (6)
Header Checksum: 0xde64 [validation disabled]
[Header checksum status: Unverified]
Source Address: 192.168.1.5
Destination Address: 128.119.245.12
Transmission Control Protocol, Src Port: 1952, Dst Port: 80, Seq: 151929, Ack: 1, Len: 1175
Source Port: 1952
Destination Port: 80
[Stream index: 4]
> [Conversation completeness: Complete, WITH_DATA (31)]

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?

My client computer (source):

Source IP Address: 192.168.1.5

Source TCP port number: 1952

The image shows a Wireshark packet capture of an HTTP POST request. The packet list pane shows two packets: packet 283 (1229 bytes) and packet 333 (831 bytes). The packet details pane for packet 283 shows the source IP address 192.168.1.5 and the source port 1952 highlighted with red boxes. The packet bytes pane shows the raw data of the packet.

The Task:

The image shows the 'Enabled Protocols' dialog box in Wireshark. The 'Search' field is set to 'http'. The 'Protocol' list on the left shows various protocols, with 'HTTP' and 'HTTP2' highlighted with red boxes. The 'Description' list on the right shows the corresponding descriptions for each protocol. The 'Enable All', 'Disable All', and 'Invert' buttons are at the bottom left, and the 'OK', 'Cancel', and 'Help' buttons are at the bottom right.

Section 3: TCP Basics

The Questions & Answers:

4. 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?

- Sequence number of the TCP SYN segment is equal to 0 here and it is used to initiate the TCP connection between the client computer and the server.
- The SYN flag is equal to 1 (set) and it indicates that this segment is a SYN segment.

The image shows a Wireshark packet capture of a TCP SYN segment. The packet list at the top shows a packet from 192.168.1.5 to 128.119.245.12 with sequence number 0 and the SYN flag set. The packet details pane shows the following information:

- Sequence Number: 0 (relative sequence number) - 0 is the sequence number of the TCP SYN segment
- Next Sequence Number: 1 (relative sequence number)
- Acknowledgment Number: 0
- Acknowledgment number (raw): 0
- 1000 = Header Length: 32 bytes (8)
- 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 - 1 to identify that it is a SYN segment
 -0... = Fin: Not set
- [TCP Flags:S.]
- Window: 64240

The packet bytes pane shows the raw data of the packet, with the sequence number 0 and the SYN flag set to 1.

5. 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?

Sequence number of the SYNACK segment from server to the client in replying to the SYN is equal to 0.

The value of the Acknowledgement field in the SYN-ACK segment is 1. The value of the Acknowledgement field in the SYN-ACK segment is determined by the server **by adding 1** to the initial sequence number of SYN segment from the client (i.e. the sequence number of the original SYN segment initiated by the client is 0 as shown before).

To indicate/identify the segment as a SYN-ACK segment : the SYN flag and Acknowledgement flag in the segment are both set to 1.

tcp

No.	Time	Source	Destination	Protocol	Length	Info
23	00:28:54.180862	192.168.1.5	128.119.245.12	TCP	66	1952 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
65	00:28:54.325511	128.119.245.12	192.168.1.5	TCP	66	80 → 1952 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1432 SACK_PERM WS=12
66	00:28:54.325615	192.168.1.5	128.119.245.12	TCP	54	1952 → 80 [ACK] Seq=1 Ack=1 Win=65792 Len=0
70	00:28:54.326683	192.168.1.5	128.119.245.12	TCP	838	1952 → 80 [PSH, ACK] Seq=1 Ack=1 Win=65792 Len=784
71	00:28:54.326835	192.168.1.5	128.119.245.12	TCP	1486	1952 → 80 [ACK] Seq=785 Ack=1 Win=65792 Len=1432
72	00:28:54.326835	192.168.1.5	128.119.245.12	TCP	1486	1952 → 80 [ACK] Seq=2217 Ack=1 Win=65792 Len=1432
73	00:28:54.326835	192.168.1.5	128.119.245.12	TCP	1486	1952 → 80 [ACK] Seq=3649 Ack=1 Win=65792 Len=1432
74	00:28:54.326835	192.168.1.5	128.119.245.12	TCP	1486	1952 → 80 [ACK] Seq=5081 Ack=1 Win=65792 Len=1432
75	00:28:54.326835	192.168.1.5	128.119.245.12	TCP	1486	1952 → 80 [ACK] Seq=6513 Ack=1 Win=65792 Len=1432
76	00:28:54.326835	192.168.1.5	128.119.245.12	TCP	1486	1952 → 80 [ACK] Seq=7945 Ack=1 Win=65792 Len=1432

[Conversation completeness: Complete, WITH_DATA (31)]
 [TCP Segment Len: 0]
 Sequence Number: 0 (relative sequence number) the Sequence number of the SYNACK segment
 Sequence Number (raw): 2689007689
 [Next Sequence Number: 1 (relative sequence number)]
 Acknowledgment Number: 1 (relative ack number) The value of the ACKnowledgement field
 Acknowledgment number (raw): 4025216167
 1000 = Header Length: 32 bytes (8)
 Flags: 0x012 (SYN, 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
0... = Push: Not set
0... = Reset: Not set
1... = Syn: Set
0... = Fin: Not set
 [TCP Flags:A..S..]
 both flags are set to identify the SYN-ACK segment

6. 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.

After investigating packets , the below packet contains POST method, its sequence number is 1.

- Using the author's capture trace:

tcp

No.	Time	Source	Destination	Protocol	Length	Info
1	16:44:20.570381	192.168.1.102	128.119.245.12	TCP	62	1161 → 80 [SYN] Seq=0 Win=16384 Len=0 MSS=1460 SACK_PERM
2	16:44:20.593553	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	16:44:20.593646	192.168.1.102	128.119.245.12	TCP	54	1161 → 80 [ACK] Seq=1 Ack=1 Win=17520 Len=0
4	16:44:20.596858	192.168.1.102	128.119.245.12	TCP	619	1161 → 80 [PSH, ACK] Seq=1 Ack=1 Win=17520 Len=565
5	16:44:20.612118	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [PSH, ACK] Seq=566 Ack=1 Win=17520 Len=1460
6	16:44:20.624318	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=566 Win=6780 Len=0
7	16:44:20.624407	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=2026 Ack=1 Win=17520 Len=1460
8	16:44:20.625071	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=3486 Ack=1 Win=17520 Len=1460
9	16:44:20.647675	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=2026 Win=8760 Len=0

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)
 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
0... = Reset: Not set
0... = Syn: Not set
0... = Fin: Not set
 [TCP Flags:AP...]

0020 f5 0c 04 89 00 50 0d d6 01 f5 34 a2 74 1a 50 18 ... P...t.P
 0030 44 70 1f bd 00 00 50 4f 53 54 20 2f 65 74 68 65 ... PO ST /ethe
 0040 72 65 61 6c 2d 6c 61 62 70 1f 0c 01 62 33 2d 31 ... ear-lab s/lab3-1
 0050 31 72 05 70 6c 79 2e 68 74 6d 20 48 54 04 50 2f ... -reply.h tm HTTP/
 0060 31 2e 31 0d 0a 48 6f 73 74 0d 20 67 61 69 61 2e ... 1.1 ·Hos t: gaia.
 0070 63 73 2e 75 6d 63 73 73 2e 65 64 75 0d 0a 55 73 ... cs.umass .edu·Us
 0080 65 73 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 30 322) 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 6f 6e 2f 78 ... 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 ... /gzip, deflate
 01c0 2c 20 63 6f 6d 70 72 65 73 73 3b 71 3d 30 2e 39 ... , compre ss;q=0.9

- Using my own capture trace:

No.	Time	Source	Destination	Protocol	Length	Info
23	00:28:54.180862	192.168.1.5	128.119.245.12	TCP	66	1952 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
65	00:28:54.325511	128.119.245.12	192.168.1.5	TCP	66	80 → 1952 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1432 SACK_PERM WS=128
66	00:28:54.325615	192.168.1.5	128.119.245.12	TCP	54	1952 → 80 [ACK] Seq=1 Ack=1 Win=65792 Len=0
70	00:28:54.326683	192.168.1.5	128.119.245.12	TCP	838	1952 → 80 [PSH, ACK] Seq=1 Ack=1 Win=65792 Len=784
71	00:28:54.326835	192.168.1.5	128.119.245.12	TCP	1486	1952 → 80 [ACK] Seq=785 Ack=1 Win=65792 Len=1432
72	00:28:54.326835	192.168.1.5	128.119.245.12	TCP	1486	1952 → 80 [ACK] Seq=2217 Ack=1 Win=65792 Len=1432
73	00:28:54.326835	192.168.1.5	128.119.245.12	TCP	1486	1952 → 80 [ACK] Seq=3649 Ack=1 Win=65792 Len=1432
74	00:28:54.326835	192.168.1.5	128.119.245.12	TCP	1486	1952 → 80 [ACK] Seq=5081 Ack=1 Win=65792 Len=1432
75	00:28:54.326835	192.168.1.5	128.119.245.12	TCP	1486	1952 → 80 [ACK] Seq=6513 Ack=1 Win=65792 Len=1432
76	00:28:54.326835	192.168.1.5	128.119.245.12	TCP	1486	1952 → 80 [ACK] Seq=7945 Ack=1 Win=65792 Len=1432

> [Conversation completeness: Complete, WITH_DATA (31)]	0020	f5 0c 07 a0 00 50 ef eb ec a7 9f 1e fa f6 50 18P.....P
[TCP Segment Len: 784]	0030	01 01 d7 22 00 00 50 4f 53 54 20 2f 77 69 72 65PO ST /wire
Sequence Number: 1 (relative sequence number)	0040	73 60 01 72 00 20 0c 01 82 73 21 bc b1 b2 71 27	shark-lab/lab3-
Sequence Number (raw): 4025216167	0050	31 2d 72 65 70 6c 79 2e 68 51 0d 20 48 54 54 50	1-reply. htm HTTP
[Next Sequence Number: 785 (relative sequence number)]	0060	2f 31 2a 31 3d 0a 48 6f 73 74 3a 20 67 61 69 61	/1.1 Ho st: gaia
Acknowledgment Number: 1 (relative ack number)	0070	2e 63 73 2e 75 6d 61 73 73 2e 65 64 75 0d 0a 43	.cs.umass.s.edu C
Acknowledgment number (raw): 2669607670	0080	6f 6e 6e 65 63 74 69 6f 6e 3a 20 6b 65 65 70 2d	onnectio n: keep-
0101 = Header Length: 20 bytes (5)	0090	61 6c 69 76 65 0d 0a 43 6f 6e 74 65 6e 74 2d 4c	alive C ontent-L
▼ Flags: 0x018 (PSH, ACK)	00a0	65 6e 67 74 68 3a 20 31 35 32 33 31 39 0d 0a 43	length: 1 52319 C
0000 = Reserved: Not set	00b0	61 63 68 65 2d 43 6f 6e 74 72 6f 6c 3a 20 6d 61	ache-Con trol: ma
...0 = Accurate ECN: Not set	00c0	78 2d 61 67 65 3d 30 0d 0a 55 70 67 72 61 64 65	x-age=0 Upgrade
...0 = Congestion Window Reduced: Not set	00d0	2d 49 6e 73 65 63 75 72 65 2d 52 65 71 75 65 73	-Insecu e-Reques
...0 = ECN-Echo: Not set	00e0	74 73 3a 20 31 0d 0a 4f 72 69 67 69 6e 3a 20 68	ts: 1 0 rigin: h
...0 = Urgent: Not set	00f0	74 74 70 3a 2f 2f 67 61 69 61 2e 63 73 2e 75 6d	ttp://ga ia.cs.um
...0 = Acknowledgment: Set	0100	61 73 73 2e 65 64 75 0d 0a 43 6f 6e 74 65 6e 74	ass.edu Content
...1 = Push: Set	0110	2d 54 79 70 65 3a 20 6d 75 6c 74 69 70 61 72 74	-Type: m ultipart
...1 = Reset: Not set	0120	2f 66 6f 72 6d 2d 64 61 74 61 3b 20 62 6f 75 6e	/form-da ta; boun
...0 = Syn: Not set	0130	64 61 72 79 3d 2d 2d 2d 2d 57 65 62 4b 69 74 46	dary= WebKitF
...0 = Fin: Not set	0140	6f 72 6d 42 6f 75 6e 64 61 72 79 70 56 75 7a 30	ormBound arypVuz0
[TCP Flags:AP....]	0150	4f 30 65 66 31 6c 56 65 4a 46 77 0d 0a 55 73 65	00ef1lve JFw Use
Window: 257	0160	72 2d 41 67 65 6e 74 3a 20 4d 6f 7a 69 6c 6c 61	r-Agent: Mozilla
	0170	2f 35 2e 30 20 28 57 69 6e 64 6f 77 73 20 4e 54	/5.0 (Wi ndows NT
	0180	20 31 30 2e 30 3b 20 57 69 6e 36 34 3b 20 78 3f	10.0; W in64; x6
	0190	24 70 70 a1 70 70 6c 65 57 65 63 4b 60 74 7f 25	A) Apple WebKit/C

7. 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 EstimatedRTT value (see Section 3.5.3, page 239 in text) after the receipt of each ACK? 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.

Note: Wireshark has a nice feature that allows you to plot the RTT for each of the TCP segments sent. Select a TCP segment in the “listing of captured packets” window that is being sent from the client to the gaia.cs.umass.edu server. Then select: Statistics->TCP Stream Graph->Round Trip Time Graph.

The HTTP POST segment is considered as the first segments from the client computer to the server in the TCP connection. Segments 1 – 6 are No. 4, 5, 7, 8, 10, and 11 in this trace respectively.

The ACKs of segments 1 – 6 are No. 6, 9, 12, 14, 15, and 16 in this trace.

1	16:44:20.570381	192.168.1.102	128.119.245.12	TCP	62	1161 → 80 [SYN] Seq=0 Win=16384 Len=0 MSS=1460 SACK_PERM
2	16:44:20.593553	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	16:44:20.593646	192.168.1.102	128.119.245.12	TCP	54	1161 → 80 [ACK] Seq=1 Ack=1 Win=17520 Len=0
4	16:44:20.596858	192.168.1.102	128.119.245.12	TCP	619	1161 → 80 [PSH, ACK] Seq=1 Ack=1 Win=17520 Len=565
5	16:44:20.612118	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [PSH, ACK] Seq=566 Ack=1 Win=17520 Len=1460
6	16:44:20.624318	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=566 Win=6780 Len=0
7	16:44:20.624407	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=2026 Ack=1 Win=17520 Len=1460
8	16:44:20.625071	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=3486 Ack=1 Win=17520 Len=1460
9	16:44:20.647675	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=2026 Win=8760 Len=0
10	16:44:20.647786	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=4946 Ack=1 Win=17520 Len=1460
11	16:44:20.648538	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=6406 Ack=1 Win=17520 Len=1460
12	16:44:20.694466	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=3486 Win=11680 Len=0
13	16:44:20.694566	192.168.1.102	128.119.245.12	TCP	1201	1161 → 80 [PSH, ACK] Seq=7866 Ack=1 Win=17520 Len=1147
14	16:44:20.739499	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=4946 Win=14600 Len=0
15	16:44:20.787680	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=6406 Win=17520 Len=0
16	16:44:20.838183	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=7866 Win=20440 Len=0

The sequence number , sending time, and the received time of ACKs are in the following table:

Seq num	Sent time	ACK time	RTT (seconds)
1	0.026477	0.053937	0.02746
566	0.041737	0.077294	0.035557
2026	0.054026	0.124085	0.070059
3486	0.054690	0.169118	0.11443
4946	0.077405	0.217299	0.13989
6406	0.078157	0.267802	0.18964

$$RTT_{est_i} = \frac{7}{8}RTT_{est(i-1)} + \frac{1}{8}RTT_{sample}$$

EstimatedRTT after the receipt of the ACK of segment 1:

EstimatedRTT = RTT for Segment 1 = 0.02746 seconds

EstimatedRTT after the receipt of the ACK of segment 2:

EstimatedRTT = 0.875 * 0.02746 + 0.125 * 0.035557 = 0.0285 seconds

EstimatedRTT after the receipt of the ACK of segment 3:

EstimatedRTT = 0.875 * 0.0285 + 0.125 * 0.070059 = 0.0337 seconds

EstimatedRTT after the receipt of the ACK of segment 4:

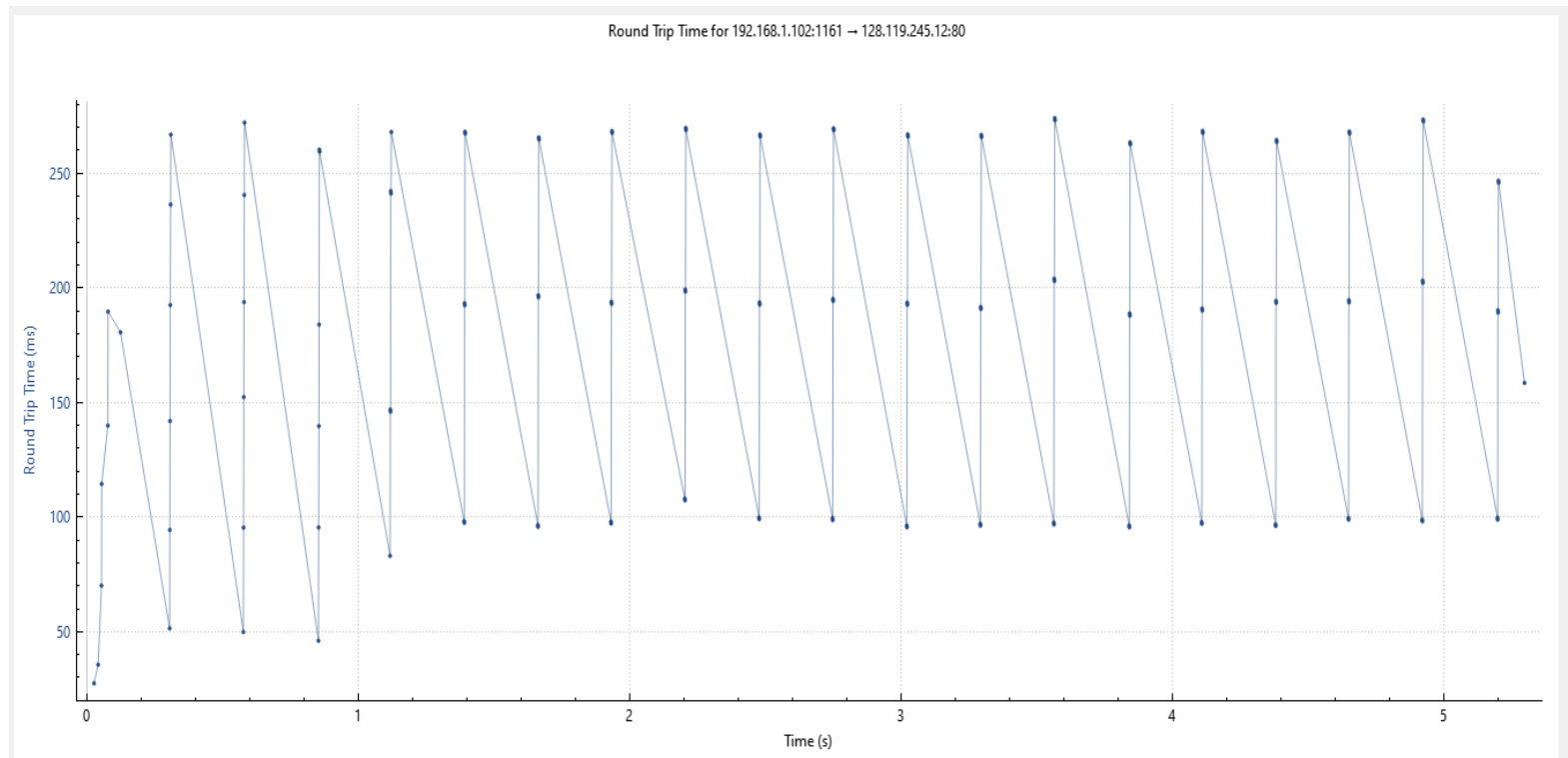
EstimatedRTT = 0.875 * 0.0337 + 0.125 * 0.11443 = 0.0438 seconds

EstimatedRTT after the receipt of the ACK of segment 5:

EstimatedRTT = 0.875 * 0.0438 + 0.125 * 0.13989 = 0.0558 seconds

EstimatedRTT after the receipt of the ACK of segment 6:

EstimatedRTT = 0.875 * 0.0558 + 0.125 * 0.18964 = 0.0725 seconds



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

The first TCP segment length of the HTTP POST: 565 bytes -> [TCP Segment Len: 565]

Length of each of the other five TCP segments: 1460 bytes -> [TCP Segment Len: 1460]

No.	Time	Source	Destination	Protocol	Length	Info
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
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
6	0.053937	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=566 Win=6780 Len=0
7	0.054026	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=2026 Ack=1 Win=17520 Len=1460
8	0.054690	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=3486 Ack=1 Win=17520 Len=1460
9	0.077294	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=566 Win=8760 Len=0

Frame 7: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits)
> Ethernet II, Src: ActiontecEle_8a:70:1a (00:20:e0:8a:70:1a), Dst: LinksysGroup_da:af:73 (00:0c:29:af:73:00)
> Internet Protocol Version 4, Src: 192.168.1.102, Dst: 128.119.245.12
▼ Transmission Control Protocol, Src Port: 1161, Dst Port: 80, Seq: 2026, Ack: 1, Len: 1460
Source Port: 1161
Destination Port: 80
[Stream index: 0]
> [Conversation completeness: Incomplete, DATA (15)]
[TCP Segment Len: 1460]

9. 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 minimum amount of buffer space (receiver window) advertised at gaia.cs.umass.edu for the entire trace is 5840 bytes, which shows in the first acknowledgement from the server(in the SYN-ACK).

This receiver window grows steadily until a maximum receiver buffer size of 62780 bytes. Therefore, the sender will never be throttled due to lacking of receiver buffer space by inspecting this trace.

Because, the size of what the Tx sends is always smaller than the allowed windows size by the server here:

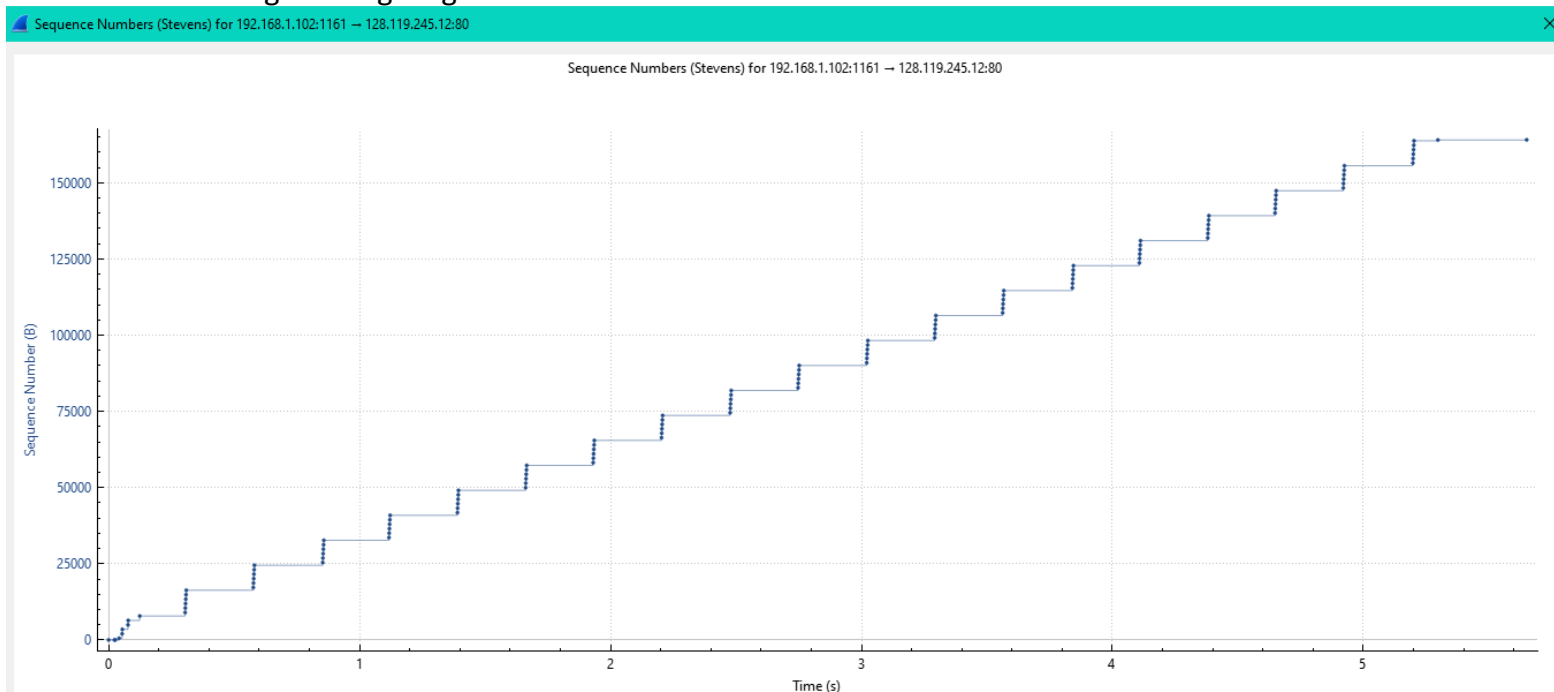
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.1.102	128.119.245.12	TCP	62	1161 → 80 [SYN] Seq=0 Win=16384 Len=0 MSS=1460 SACK_PERM
2	0.023172	128.119.245.12	192.168.1.102	TCP	62	80 → 1161 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460 SACK_PERM
3	0.023265	192.168.1.102	128.119.245.12	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
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
6	0.053937	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=566 Win=6780 Len=0

[Stream index: 0]
> [Conversation completeness: Incomplete, DATA (15)]
[TCP Segment Len: 0]
Sequence Number: 0 (relative sequence number)
Sequence Number (raw): 883061785
[Next Sequence Number: 1 (relative sequence number)]
Acknowledgment Number: 1 (relative ack number)
Acknowledgment number (raw): 232129013
0111 = Header Length: 28 bytes (7)
▼ Flags: 0x012 (SYN, 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
...0 = Push: Not set
...0 = Reset: Not set
> ...1 = Syn: Set
...0 = Fin: Not set
Window: 5840
[Calculated window size: 5840]
Checksum: 0x74d [unverified]

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

No. There is no retransmitted segments in the trace file since in the time sequence graph (stevens), all sequence numbers are monotonically increasing. But, if there is retransmission there will be a drop in the curve, or the Seq number will decrease suddenly then returned to increase again. so to conclude, There are no retransmitted segments in the trace file. We can verify this by

checking the sequence numbers of the TCP segments in the trace file or using graph. If there is a retransmitted segment, the sequence number of this retransmitted segment should be smaller than those of its neighboring segments.



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

In the first acknowledgement, the Rx ACK a segment with size 566 bytes, but all the other sizes were 1460 bytes. So, for example in The acknowledged sequence numbers of the first 6 ACKs are listed as follows.

	acknowledged sequence number	acknowledged data size
ACK 1	566	566
ACK 2	2026	1460
ACK 3	3486	1460
ACK 4	4946	1460
ACK 5	6406	1460
ACK 6	7866	1460

The difference between the acknowledged sequence numbers of two consecutive ACKs indicates the data received by the server between these two ACKs. By inspecting the amount of acknowledged data size by each ACK, there are a lot of cases where the receiver is ACKing every other segment. For example, after the first TCP ACK segment, We can conclude that the ACK increases each time by 1460.

No.	Time	Source	Destination	Protocol	Length	Info
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
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
6	0.053937	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=566 Win=6780 Len=0
7	0.054026	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=2026 Ack=1 Win=17520 Len=1460
8	0.054690	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=3486 Ack=1 Win=17520 Len=1460
9	0.077294	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=2026 Win=8760 Len=0
10	0.077405	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=4946 Ack=1 Win=17520 Len=1460
11	0.078157	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=6406 Ack=1 Win=17520 Len=1460
12	0.124085	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=3486 Win=14680 Len=0
13	0.124185	192.168.1.102	128.119.245.12	TCP	1201	1161 → 80 [PSH, ACK] Seq=7866 Ack=1 Win=17520 Len=1147
14	0.169118	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=4946 Win=14600 Len=0
15	0.217299	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=6406 Win=17520 Len=0
16	0.267802	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=7866 Win=20440 Len=0
17	0.304807	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=9013 Win=23360 Len=0

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

$$\text{Throughput} = \frac{\text{Data}}{\text{Time}}$$

Data = ACK-Seq num of Last ACK (before the HTTP OK)- Seq. of 1st ACK TCP Segment of HTTP POST from client to server= 164091-1 = 164090 bytes

Time = Time of Last ACK (before the HTTP OK)- Time of First ACK TCP Seg of HTTP POST= 5.455830-0.026477= 5.4294 sec

the throughput for the TCP connection = 164090/5.4294 = 30.222 KByte/sec.

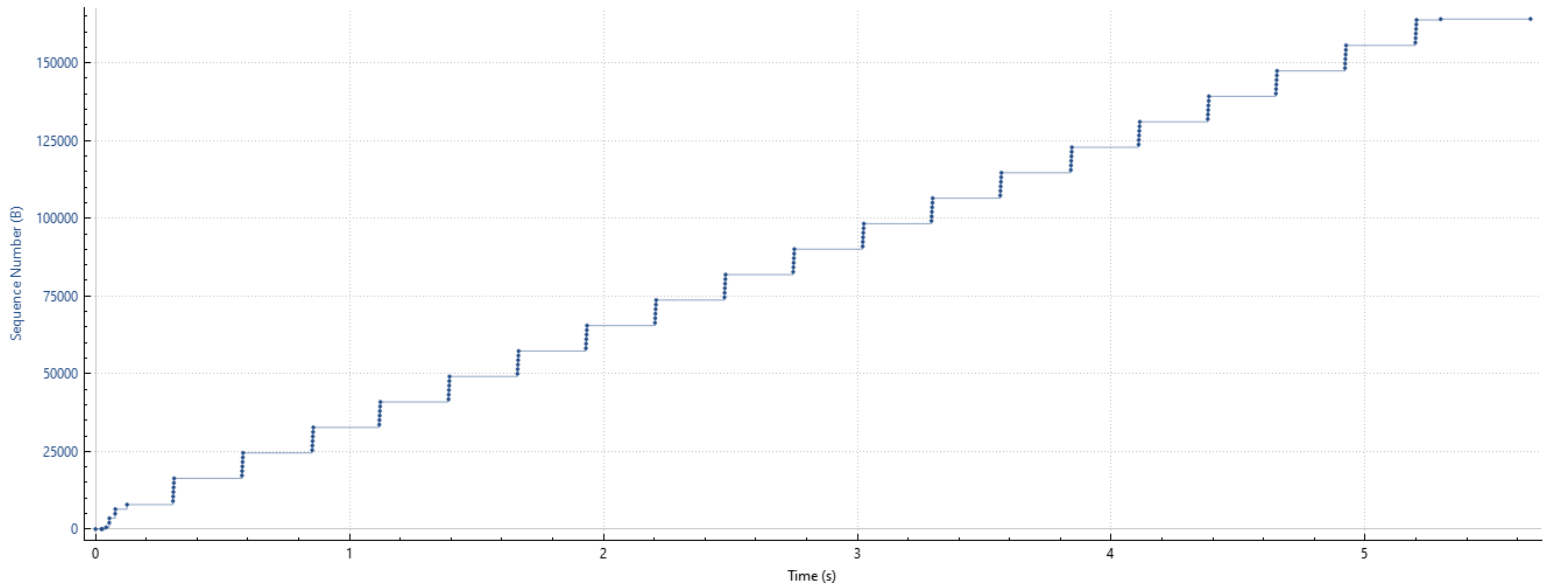
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.1.102	128.119.245.12	TCP	62	1161 → 80 [SYN] Seq=0 Win=16384 Len=0 MSS=1460 SACK_PERM
4	0.026477	192.168.1.102	128.119.245.12	TCP	619	1161 → 80 [PSH, ACK] Seq=1 Ack=1 Win=17520 Len=565
5	0.041737	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [PSH, ACK] Seq=566 Ack=1 Win=17520 Len=1460
6	0.053937	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=566 Win=6780 Len=0
5.455830		128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=164091 Win=62780 Len=0
5.461175		192.168.1.102	128.119.245.12	HTTP	784	HTTP/1.1 200 OK (text/html)
5.598090		192.168.1.102	128.119.245.12	SSDP	174	M-SEARCH * HTTP/1.1
5.599082		192.168.1.102	128.119.245.12	SSDP	175	M-SEARCH * HTTP/1.1
5.651141		192.168.1.102	128.119.245.12	TCP	54	1161 → 80 [ACK] Seq=164091 Ack=731 Win=16790 Len=0

Section 4: TCP congestion control in action

The Task:

The screenshot shows the Wireshark interface with the following details:

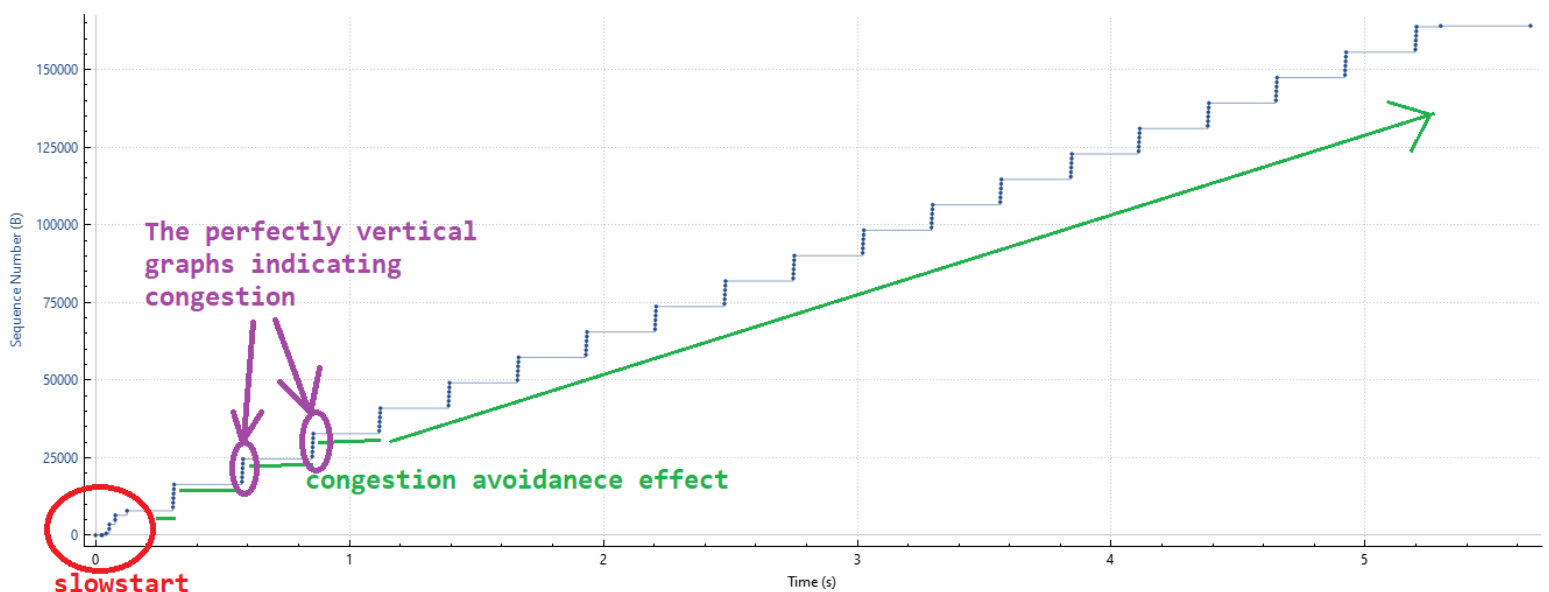
- Packet List:** Shows a sequence of TCP segments. The selected packet (No. 4) is a [PSH, ACK] segment from 192.168.1.102 to 128.119.245.12.
- Packet Details:** The 'Time Sequence (Stevens)' section is expanded, showing the 'Throughput' section. The 'Throughput' section shows the calculated throughput for the selected packet.
- Packet Bytes:** The raw data of the selected packet is displayed in hexadecimal and ASCII.



The Questions & Answers:

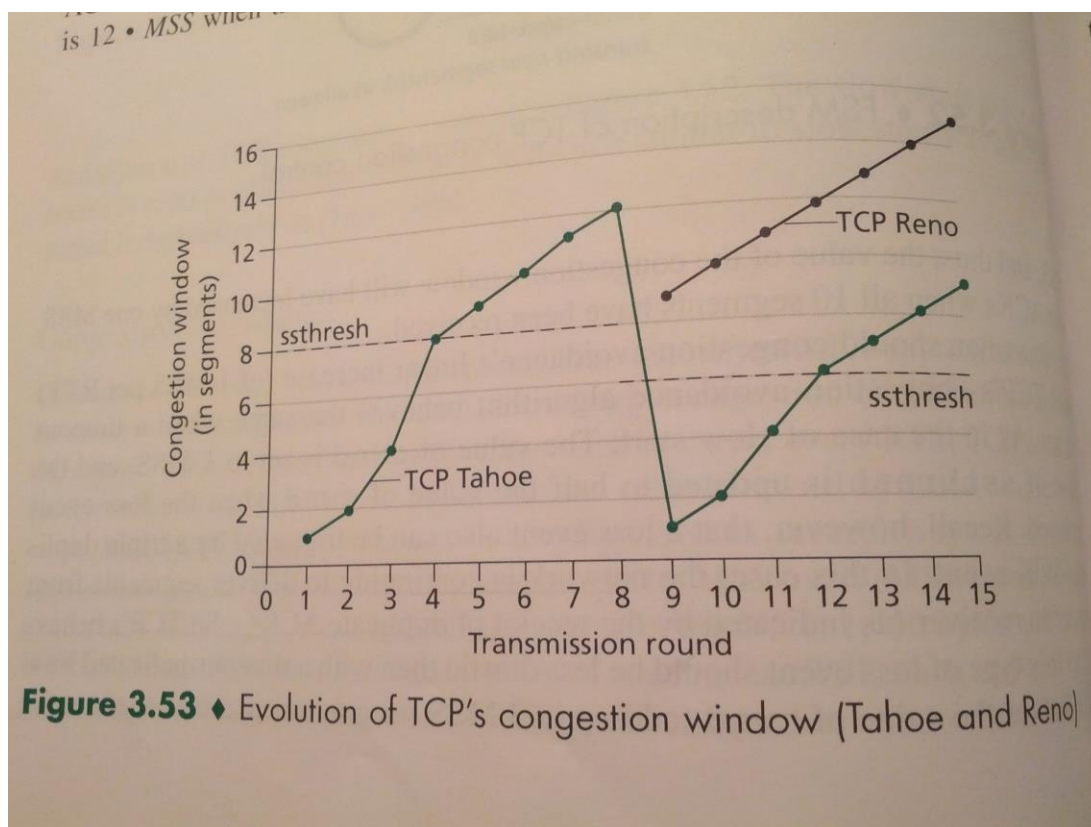
13. Use the Time-Sequence-Graph(Stevens) plotting tool to view the sequence number versus time plot of segments being sent from the client to the gaia.cs.umass.edu server. Can you identify where TCP's slowstart phase begins and ends, and where congestion avoidance takes over? Comment on ways in which the measured data differs from the idealized behavior of TCP that we've studied in the text.

Slow start : prevents a network from becoming congested by regulating the amount of data that's sent over it. It negotiates the connection between a sender and receiver by defining the amount of data that can be transmitted with each packet, and slowly increases the amount of data until the network's capacity is reached.



Slow Start starts at the beginning Sequence Number = 566 and ends at Sequence Number = 7866
Congestion Avoidance starts to take place at Sequence Number = 7866.

Comment: This differs from the perfectly exponentially plotted slow start graphs seen earlier in the text, as the plotted graph is a lot more jagged and uneven, as well as the perfectly vertical graphs indicating congestion avoidance compared to the more gradual graphs shown in the text.

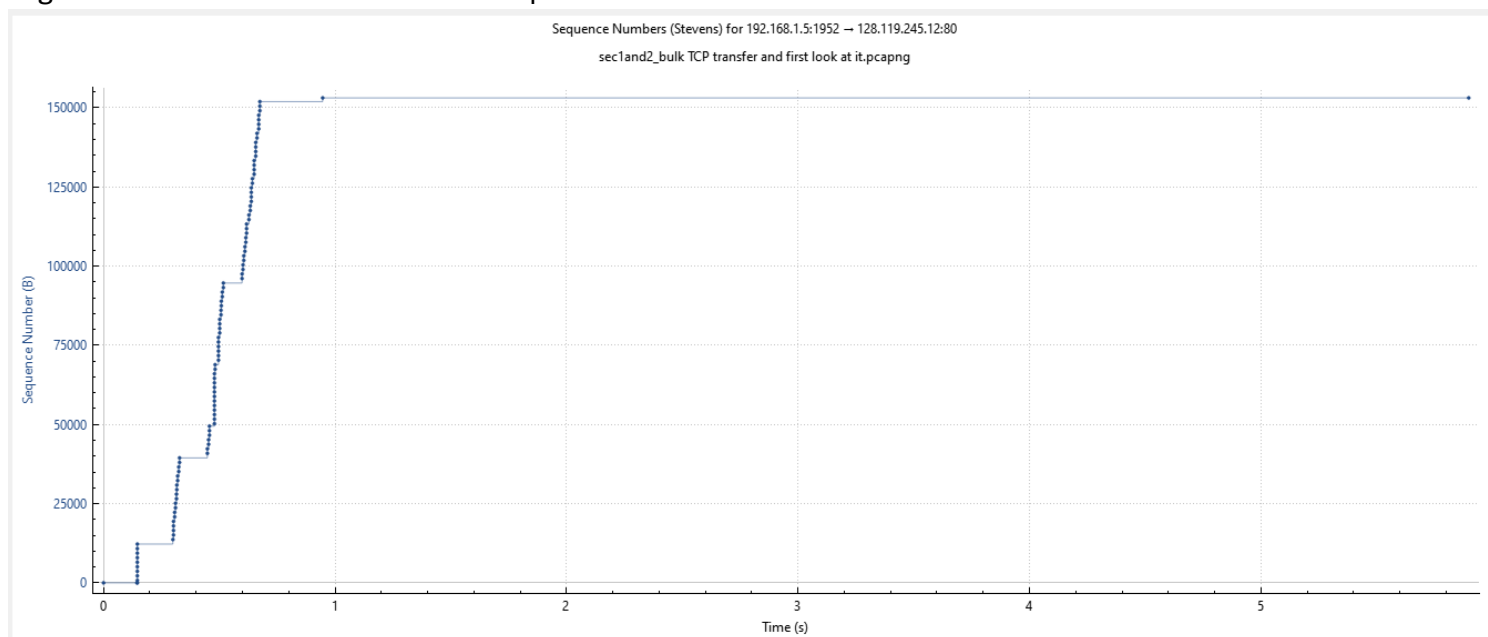


14. Answer each of two questions above for the trace that you have gathered when you transferred a file from your computer to gaia.cs.umass.edu?

First solution:

Slow Start starts at Sequence Number = 785 and ends at Sequence Number = 12241

Congestion Avoidance starts at Sequence Number = 12241



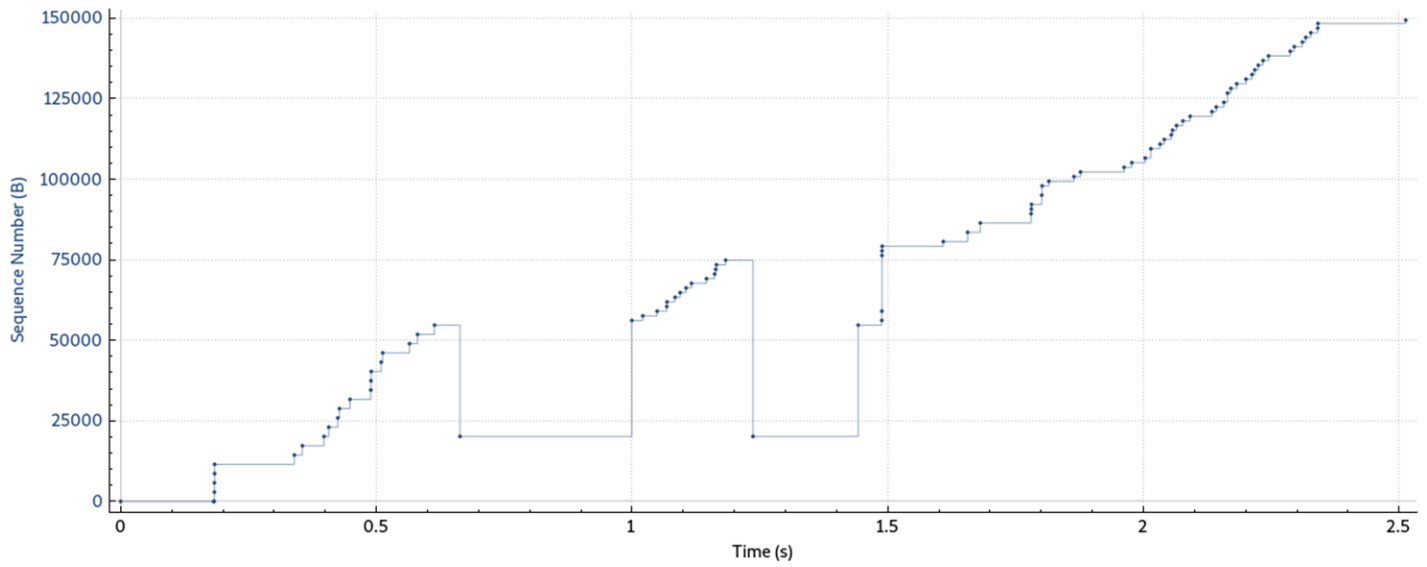
Second solution:

Slow Start starts at Sequence Number = 566 and ends at Sequence Number = 7866

Congestion Avoidance starts at Sequence Number = 7866

Sequence Numbers (Stevens) for 192.168.1.5:52640 → 128.119.245.12:80

lab2 trial1.pcapng



Comment:

if there is retransmission there will be a drop in the curve, or the Seq number will decrease suddenly then returned to increase again.

