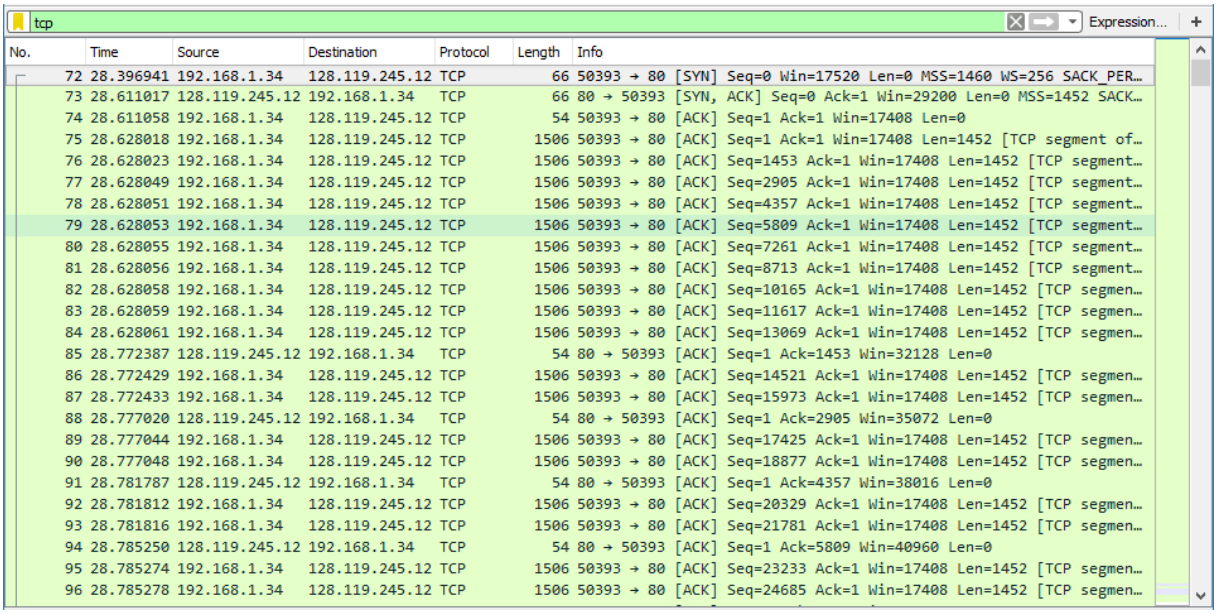
1. Capturing a bulk TCP transfer from your computer to a remote Server



tcp — Expression… +

No.	Time	Source	Destination	Protocol	Length	Info
72	28.396941	192.168.1.34	128.119.245.12	TCP	66	50393 → 80 [SYN] Seq=0 Win=17520 Len=0 MSS=1460 WS=256 SACK_PER…
73	28.611017	128.119.245.12	192.168.1.34	TCP	66	80 → 50393 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1452 SACK…
74	28.611058	192.168.1.34	128.119.245.12	TCP	54	50393 → 80 [ACK] Seq=1 Ack=1 Win=17408 Len=0
75	28.628018	192.168.1.34	128.119.245.12	TCP	1506	50393 → 80 [ACK] Seq=1 Ack=1 Win=17408 Len=1452 [TCP segment of…
76	28.628023	192.168.1.34	128.119.245.12	TCP	1506	50393 → 80 [ACK] Seq=1453 Ack=1 Win=17408 Len=1452 [TCP segment…
77	28.628049	192.168.1.34	128.119.245.12	TCP	1506	50393 → 80 [ACK] Seq=2905 Ack=1 Win=17408 Len=1452 [TCP segment…
78	28.628051	192.168.1.34	128.119.245.12	TCP	1506	50393 → 80 [ACK] Seq=4357 Ack=1 Win=17408 Len=1452 [TCP segment…
79	28.628053	192.168.1.34	128.119.245.12	TCP	1506	50393 → 80 [ACK] Seq=5809 Ack=1 Win=17408 Len=1452 [TCP segment…
80	28.628055	192.168.1.34	128.119.245.12	TCP	1506	50393 → 80 [ACK] Seq=7261 Ack=1 Win=17408 Len=1452 [TCP segment…
81	28.628056	192.168.1.34	128.119.245.12	TCP	1506	50393 → 80 [ACK] Seq=8713 Ack=1 Win=17408 Len=1452 [TCP segment…
82	28.628058	192.168.1.34	128.119.245.12	TCP	1506	50393 → 80 [ACK] Seq=10165 Ack=1 Win=17408 Len=1452 [TCP segmen…
83	28.628059	192.168.1.34	128.119.245.12	TCP	1506	50393 → 80 [ACK] Seq=11617 Ack=1 Win=17408 Len=1452 [TCP segmen…
84	28.628061	192.168.1.34	128.119.245.12	TCP	1506	50393 → 80 [ACK] Seq=13069 Ack=1 Win=17408 Len=1452 [TCP segmen…
85	28.772387	128.119.245.12	192.168.1.34	TCP	54	80 → 50393 [ACK] Seq=1 Ack=1453 Win=32128 Len=0
86	28.772429	192.168.1.34	128.119.245.12	TCP	1506	50393 → 80 [ACK] Seq=14521 Ack=1 Win=17408 Len=1452 [TCP segmen…
87	28.772433	192.168.1.34	128.119.245.12	TCP	1506	50393 → 80 [ACK] Seq=15973 Ack=1 Win=17408 Len=1452 [TCP segmen…
88	28.777020	128.119.245.12	192.168.1.34	TCP	54	80 → 50393 [ACK] Seq=1 Ack=2905 Win=35072 Len=0
89	28.777044	192.168.1.34	128.119.245.12	TCP	1506	50393 → 80 [ACK] Seq=17425 Ack=1 Win=17408 Len=1452 [TCP segmen…
90	28.777048	192.168.1.34	128.119.245.12	TCP	1506	50393 → 80 [ACK] Seq=18877 Ack=1 Win=17408 Len=1452 [TCP segmen…
91	28.781787	128.119.245.12	192.168.1.34	TCP	54	80 → 50393 [ACK] Seq=1 Ack=4357 Win=38016 Len=0
92	28.781812	192.168.1.34	128.119.245.12	TCP	1506	50393 → 80 [ACK] Seq=20329 Ack=1 Win=17408 Len=1452 [TCP segmen…
93	28.781816	192.168.1.34	128.119.245.12	TCP	1506	50393 → 80 [ACK] Seq=21781 Ack=1 Win=17408 Len=1452 [TCP segmen…
94	28.785250	128.119.245.12	192.168.1.34	TCP	54	80 → 50393 [ACK] Seq=1 Ack=5809 Win=40960 Len=0
95	28.785274	192.168.1.34	128.119.245.12	TCP	1506	50393 → 80 [ACK] Seq=23233 Ack=1 Win=17408 Len=1452 [TCP segmen…
96	28.785278	192.168.1.34	128.119.245.12	TCP	1506	50393 → 80 [ACK] Seq=24685 Ack=1 Win=17408 Len=1452 [TCP segmen…

Figure 1: Wireshark window after capturing packages

2. A first look at the captured trace

1. IP address used by client computer (source) is: *192.168.1.102*

```
Internet Protocol Version 4, Src: 192.168.1.102, Dst: 128.119.245.12
```

- TCP port number used by client computer (source) is: *1161*

```
Transmission Control Protocol, Src Port: 1161, Dst Port: 80, Seq: 0, Len: 0
```

2. IP address of gaia.cs.umass.edu is: *128.119.245.12*

```
Internet Protocol Version 4, Src: 192.168.1.102, Dst: 128.119.245.12
```

- TCP segments is sending and receiving for this connection on: *80*

```
Transmission Control Protocol, Src Port: 1161, Dst Port: 80, Seq: 0, Len: 0
```

3. IP address used by my client computer (source) is: *192.168.1.34*

```
Internet Protocol Version 4, Src: 192.168.1.34, Dst: 128.119.245.12
```

- TCP port number used by my client computer (source) is: *50393*

```
Transmission Control Protocol, Src Port: 50393, Dst Port: 80, Seq: 0, Len: 0
```

3. TCP Basics

4. Sequence number of the TCP SYN segment that is used to initiate the TCP connection between the client computer and gaia.cs.umass.edu is: *0*

```
1 0.000000   192.168.1.102    128.119.245.12  TCP        62 1161 → 80 [SYN] Seq=0 Win=16384 Len=0 MSS=1460
```

- The SYN flag is set to 1 and it indicates that this segment is a SYN segment

```
.... .... ..1. = Syn: Set
  [Expert Info (Chat/Sequence): Connection establish request (SYN): server port 80]
```

5. Sequence number of the SYNACK segment sent by gaia.cs.umass.edu to the client computer in reply to the SYN is: *0*

```
80 → 1161 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460
```

- Value of the Acknowledgement field in the SYNACK segment is: *1*

```
Acknowledgment number: 1    (relative ack number)
```

- The gaia.cs.umass.edu determines that value by adding 1 to the initial sequence number of SYN segment which is *0* from the client computer

- *Acknowledgement* and *Set* flags identifies the segment as a SYNACK segment

```
.... ..1. .... = Acknowledgment: Set
.... .... 0... = Push: Not set
.... .... .0.. = Reset: Not set
.... .... ..1. = Syn: Set
```

6. Sequence number of the TCP segment containing the HTTP POST command is: 1

In order to find the POST command, I have dug into the packet content field at the bottom of the Wireshark window and I found “POST” there.

```
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
```

```
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.
0070  63 73 2e 75 6d 61 73 73  2e 65 64 75 0d 0a 55 73   cs.umass .edu··Us
```

7. Sequence numbers of the first six segments in the TCP connection (including the segment containing the HTTP POST) are:

```
 619 1161 → 80 [PSH, ACK] Seq=1 Ack=1 Win=17520 Len=565
1514 1161 → 80 [PSH, ACK] Seq=566 Ack=1 Win=17520 Len=1460
  60 80 → 1161 [ACK] Seq=1 Ack=566 Win=6780 Len=0
1514 1161 → 80 [ACK] Seq=2026 Ack=1 Win=17520 Len=1460
1514 1161 → 80 [ACK] Seq=3486 Ack=1 Win=17520 Len=1460
  60 80 → 1161 [ACK] Seq=1 Ack=2026 Win=8760 Len=0
1514 1161 → 80 [ACK] Seq=4946 Ack=1 Win=17520 Len=1460
1514 1161 → 80 [ACK] Seq=6406 Ack=1 Win=17520 Len=1460
```

Each segment was sent at:

```
0.026477  192.168.1.102  128.119.245.12  TCP   619 1161 → 80 [PSH, ACK] Seq=1 Ack=1 Win=17520 Len=565
0.041737  192.168.1.102  128.119.245.12  TCP  1514 1161 → 80 [PSH, ACK] Seq=566 Ack=1 Win=17520 Len=1460
0.053937  128.119.245.12 192.168.1.102   TCP    60 80 → 1161 [ACK] Seq=1 Ack=566 Win=6780 Len=0
0.054026  192.168.1.102  128.119.245.12  TCP  1514 1161 → 80 [ACK] Seq=2026 Ack=1 Win=17520 Len=1460
0.054690  192.168.1.102  128.119.245.12  TCP  1514 1161 → 80 [ACK] Seq=3486 Ack=1 Win=17520 Len=1460
0.077294  128.119.245.12 192.168.1.102   TCP    60 80 → 1161 [ACK] Seq=1 Ack=2026 Win=8760 Len=0
0.077405  192.168.1.102  128.119.245.12  TCP  1514 1161 → 80 [ACK] Seq=4946 Ack=1 Win=17520 Len=1460
0.078157  192.168.1.102  128.119.245.12  TCP  1514 1161 → 80 [ACK] Seq=6406 Ack=1 Win=17520 Len=1460
```

ACK for each segment was received at:

```
 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=11680 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
```

RTT and EstimatedRTT values for each of the sixsegments are:

Segment Number	Sequence Number	Sent Time (seconds)	ACKed Time (seconds)	RTT Value (seconds)	EstimatedRTT Value (seconds)
1	1	0.026477	0.053937	0.027460	0.027460
2	566	0.041737	0.077294	0.035557	0.028472
3	2026	0.054026	0.124085	0.070059	0.033670
4	3486	0.054690	0.169118	0.114428	0.043764
5	4946	0.077405	0.217299	0.139894	0.055780
6	6406	0.078157	0.267802	0.189645	0.072513

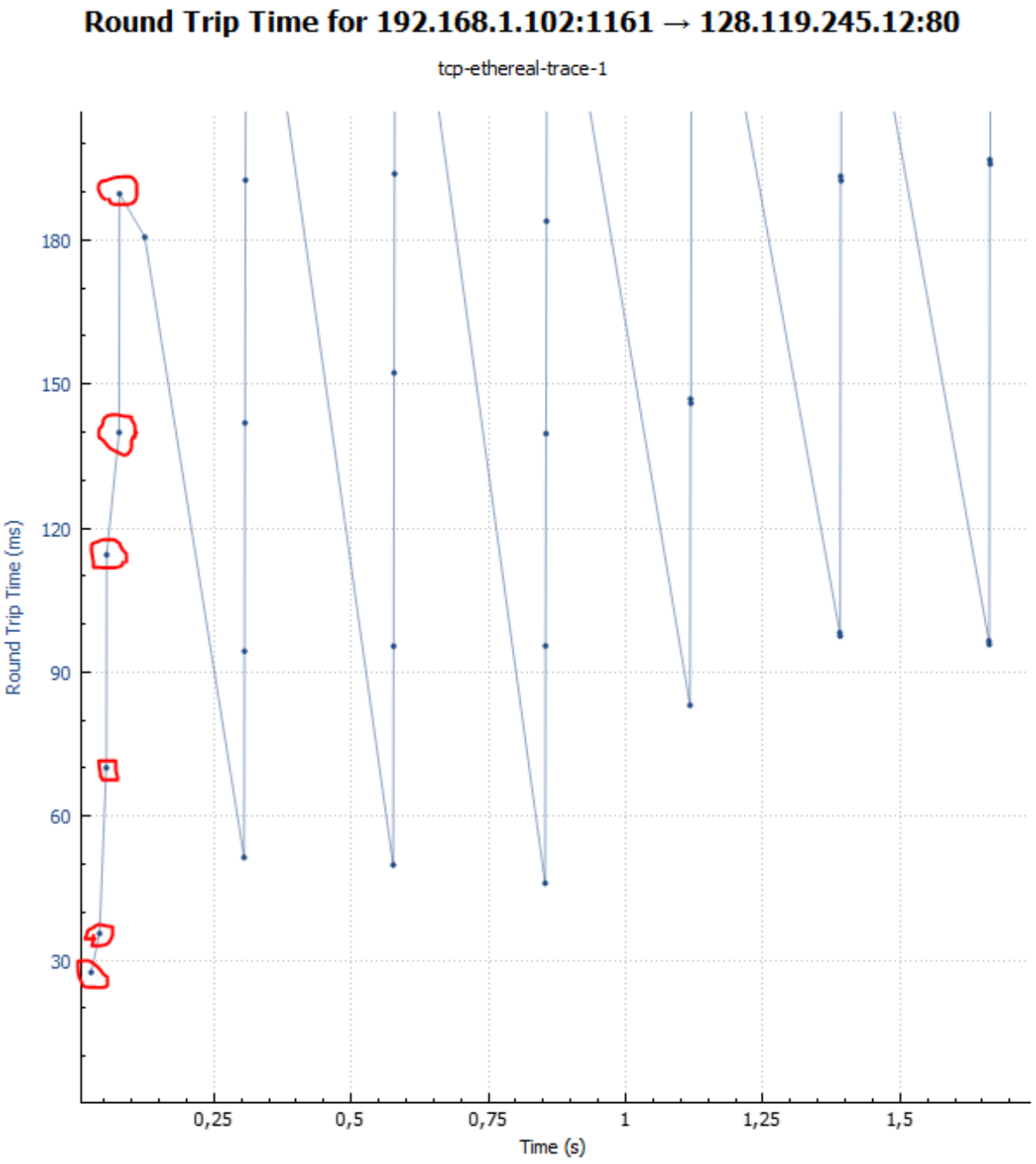


Figure 2: Round Trip Time Graph for First Six Segments

8. Length of each of the first TCP segment is: *565*

```
[TCP Segment Len: 565]
```

And the length of the remaining five TCP segments is: *1460*

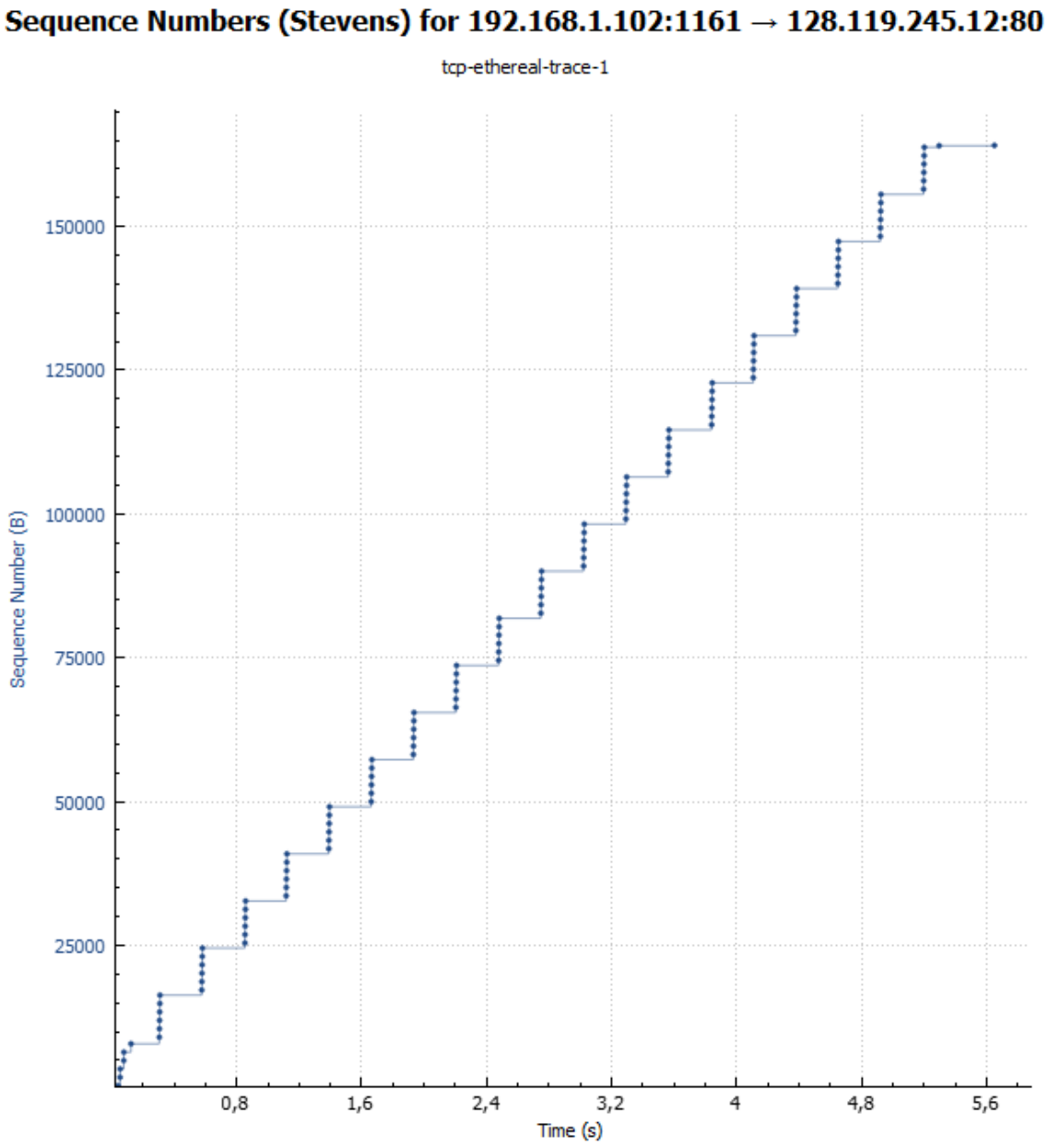
```
[TCP Segment Len: 1460]
```

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

```
Window size value: 5840
```

This receiver window grows steadily until a maximum receiver buffer size of 62780 bytes. According to the trace, the sender is never throttled due to lacking of receiver buffer space

10. There is no retransmitted segments in the trace file. I have checked sequence numbers of the ACKs for (in the trace) in order to answer this question. I have used Time-Sequence-Graph (Stevens) graph analyze method. In the Time-Sequence-Graph (Stevens) of this trace, all sequence numbers from the source (192.168.1.102) to the destination (128.119.245.12) are increasing monotonically with respect to time.



11. Data received by the server between these two ACKs is indicated with the difference between the acknowledged sequence numbers of two consecutive ACKs. There are cases where the receiver is ACKing every other segment by inspecting the amount of acknowledged data by each ACK.

Seq. number of the first ACK is 1 and the second ACK is 566. The data sended from server is 565 – 1 = 565 which is emphasised as “Len=565” below. The seq. number of the thirs ACK is 2026 and the second is 566, received data size is 2026 – 566 = 1460.

```
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
```

12. The “alice.txt” text file is 152.138 bytes but on the hard drive it is 155.648 bytes. Whereas, when it is uploaded to the server, also headers uploaded. Whole uploaded size can be calculated from sequence numbers. Seq. number of the first TCP segment is 1 and the last ACK is 164091. Thus, the uploaded size is 164090 bytes. Uploaded time can be calculated like this method. Time of the first TCP segment is 0.026477 seconds and the last ACK is 5.455830 seconds. The throughput (bytes transferred per unit time) for the TCP connection is 164090 bytes / (5.4558 - 0.026477) seconds = 30222.75398 bytes per second.

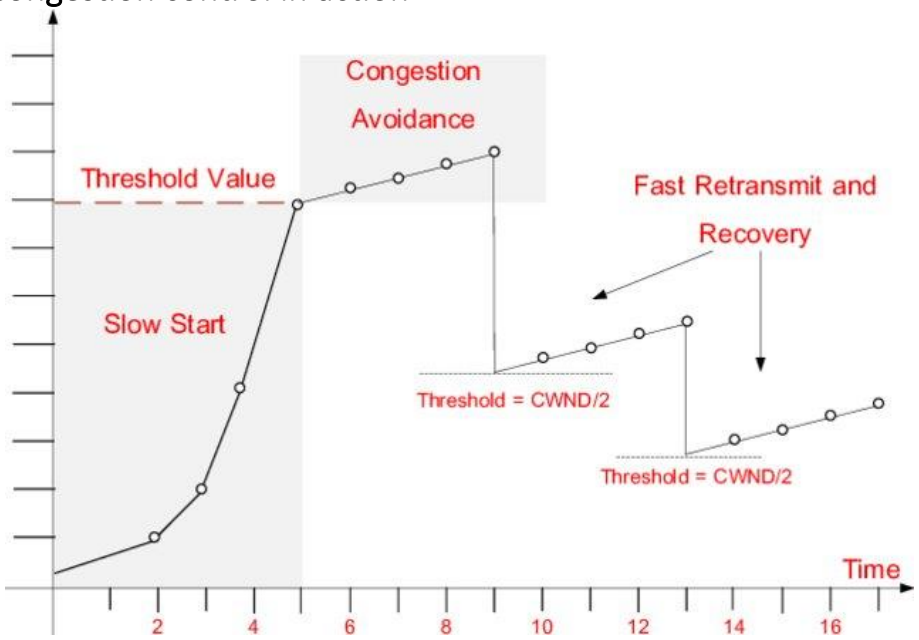
The first TCP segment:

```
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
```

The last ACK:

```
202 5.455830   128.119.245.12   192.168.1.102   TCP       60 80 → 1161 [ACK] Seq=1 Ack=164091 Win=62780 Len=0
```

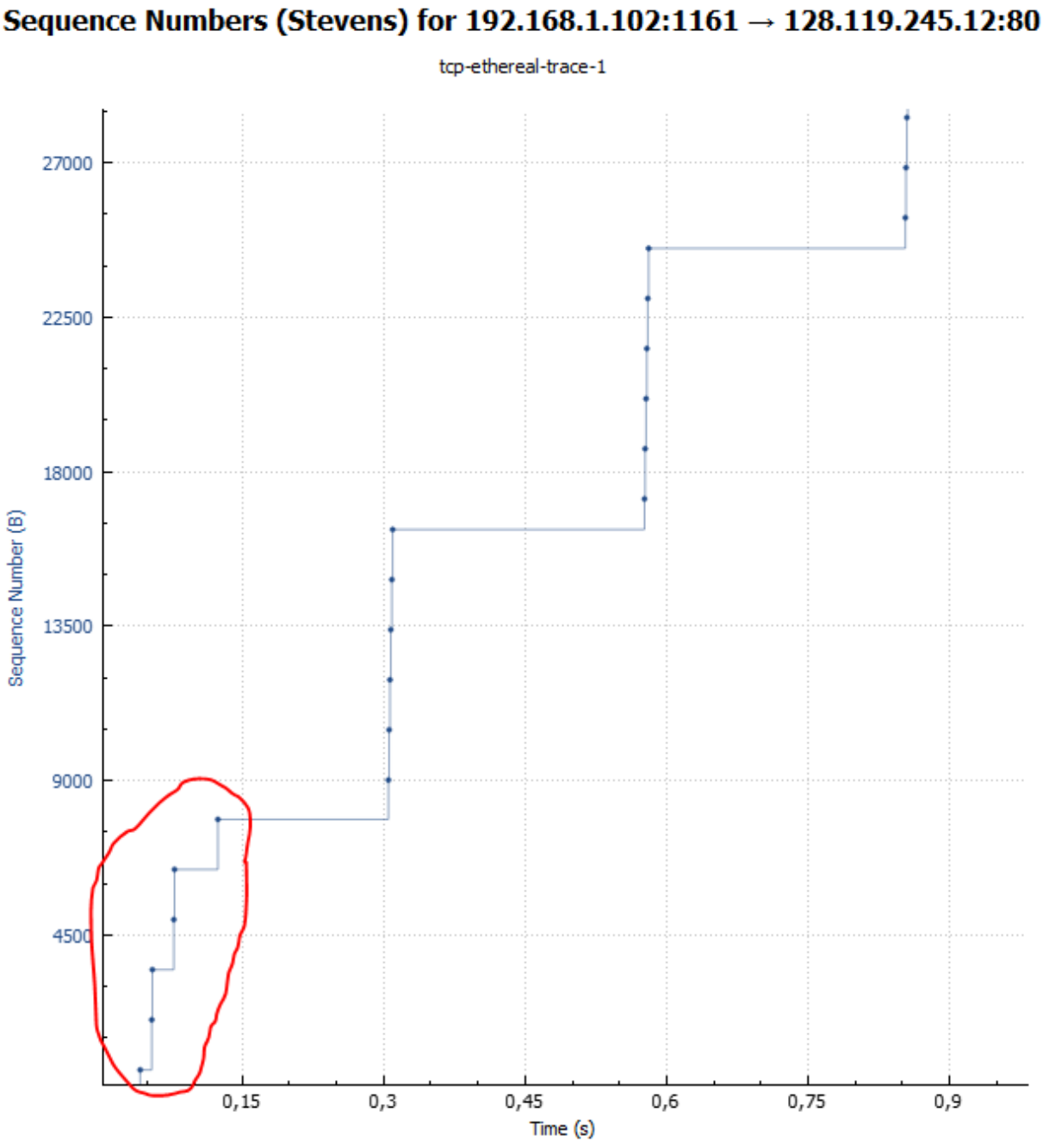
4. TCP congestion control in action



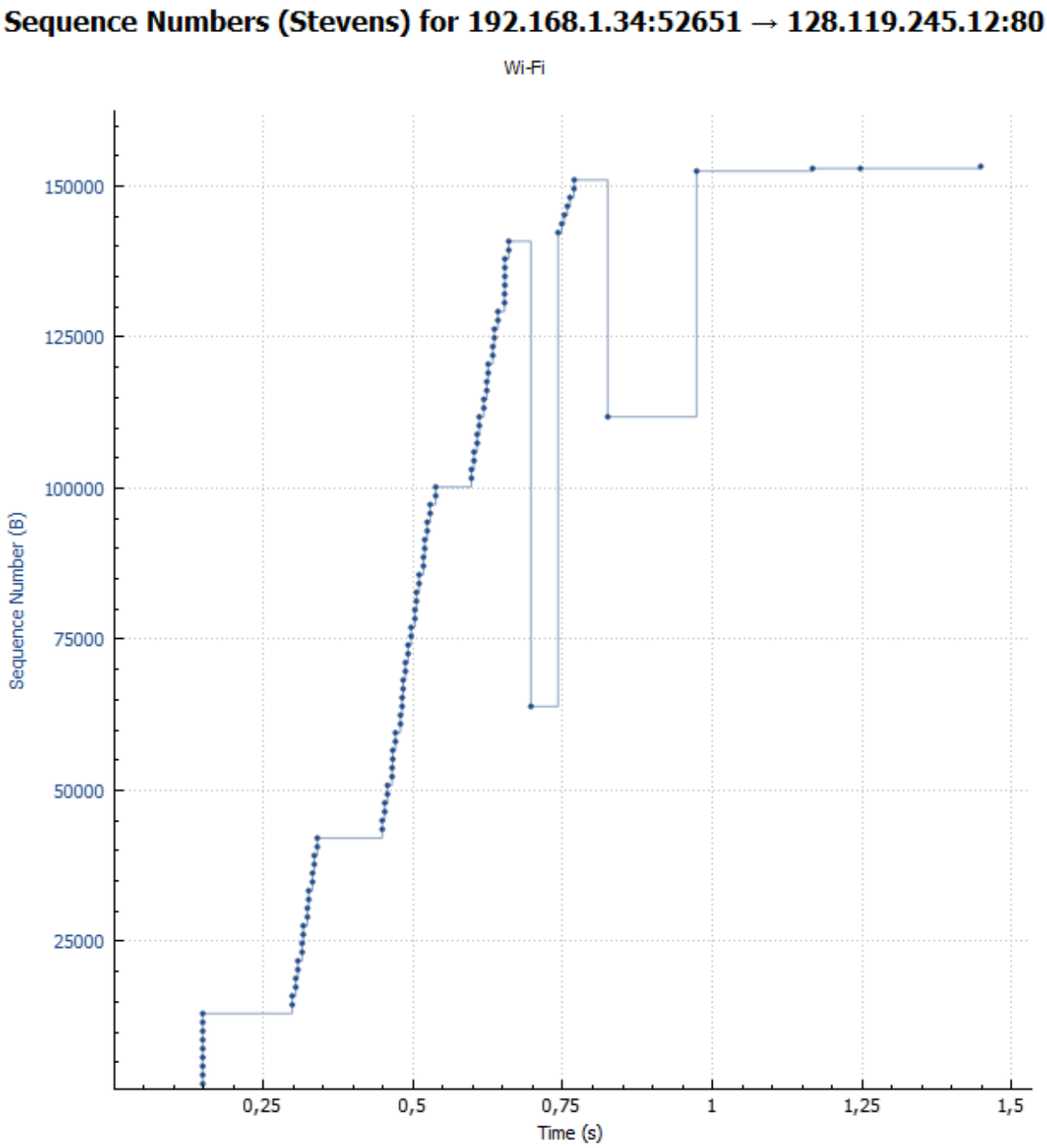
[1]

13. TCP's slowstart phase begins at 0 seconds and ends at 0.12 seconds. Congestion avoidance happens straight vertical increases in the graph below like at the 0.3, 0.55, 0.85 seconds.

    The measured data differs from the idealized behaivor of TCP that we have studied 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. TCP's slowstart phase begins at 0 seconds and ends at 0.15 seconds. Congestion avoidance happens straight vertical increases in the graph above like at the 0.15, 0.3-0.35, 0.45-0.55 seconds.



References:

[1]https://www.researchgate.net/profile/Luigi_Vanfretti/publication/256197047/figure/fig6/AS:614067353956354@1523416474316/Slow-Start-and-Congestion-Avoidance-in-TCP.png (Access date: 2018.12.02)