

# Wireshark Lab: TCP v6.0

Vandré Leal Cândido

September 1, 2017

## 1 A first look at the captured trace

Answer the following questions, by opening the Wireshark captured packet file `tcpethereal-trace-1` in <http://gaia.cs.umass.edu/wireshark-labs/wireshark-traces.zip> (that is download the trace and open that trace in Wireshark; see footnote 2). Whenever possible, when answering a question you should hand in a printout of the packet(s) within the trace that you used to answer the question asked. Annotate the 3 printout to explain your answer. To print a packet, use File - Print, choose Selected packet only, choose Packet summary line, and select the minimum amount of packet detail that you need to answer the question.

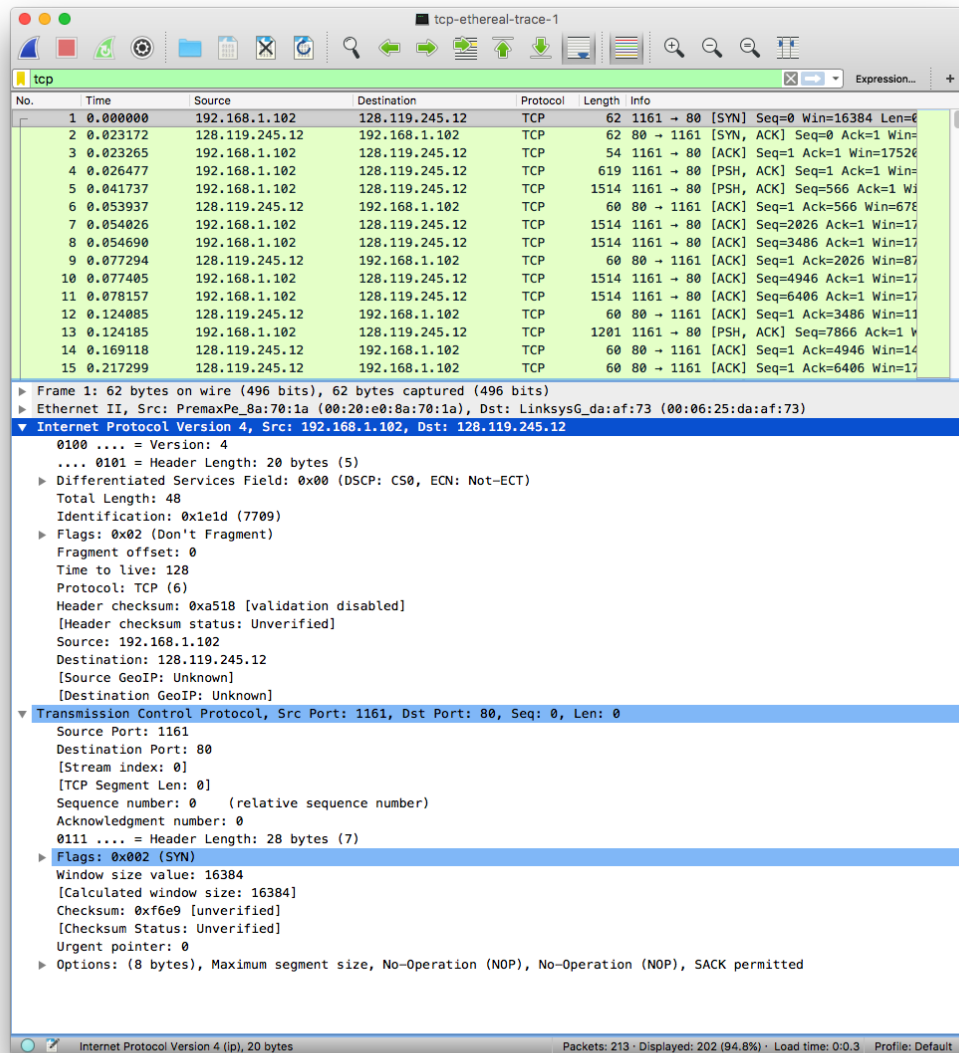
- *What is the IP address and TCP port number used by the client computer (source) that is transferring the file to `gaia.cs.umass.edu`? To answer this question, it's probably easiest to select an HTTP message and explore the details of the TCP packet used to carry this HTTP message, using the “details of the selected packet header window” (refer to Figure 2 in the “Getting Started with Wireshark” Lab if you're uncertain about the Wireshark windows).*

The IP address used by the client computer is 192.168.1.102 and the source TCP port number is 1161.

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

The IP address of `gaia.cs.umass.edu` is 128.119.245.12 and the TCP port is 80.

Figure 1: tcpethereal-trace-1



If you have been able to create your own trace, answer the following question:

- What is the IP address and TCP port number used by your client computer (source) to transfer the file to *gaia.cs.umass.edu*?

The IP address used by my computer is 192.168.0.15 and the TCP port is 64813.

Figure 2: tcplocal-trace-1

The image shows a Wireshark packet capture window titled 'tcp\_exp1.pcapng'. The top toolbar includes icons for file operations, packet list, packet details, packet bytes, and search. Below the toolbar is a display filter bar with the text 'Apply a display filter ... <[filter]>'. The main packet list table shows the following data:

No.	Time	Source	Destination	Protocol	Length	Info
21	2.955816	192.168.0.15	8.8.8.8	DNS	77	Standard query 0x5c8c A gaia.cs.umass
22	3.054902	8.8.8.8	192.168.0.15	DNS	93	Standard query response 0x5c8c A gaia
23	3.055301	192.168.0.15	128.119.245.12	TCP	78	64813 → 80 [SYN] Seq=0 Win=65535 Len=
24	3.127242	192.168.0.11	239.255.255.250	SSDP	167	M-SEARCH * HTTP/1.1
25	3.139748	128.119.245.12	192.168.0.15	TCP	66	80 → 64808 [FIN, ACK] Seq=1 Ack=2 Win=
26	3.139839	192.168.0.15	128.119.245.12	TCP	66	64808 → 80 [ACK] Seq=2 Ack=2 Win=4117
27	3.206234	192.168.0.15	8.8.8.8	DNS	77	Standard query 0x9791 A gaia.cs.umass
28	3.224867	128.119.245.12	192.168.0.15	TCP	74	80 → 64813 [SYN, ACK] Seq=0 Ack=1 Win=
29	3.224944	192.168.0.15	128.119.245.12	TCP	66	64813 → 80 [ACK] Seq=1 Ack=1 Win=1317
30	3.225385	192.168.0.15	128.119.245.12	TCP	762	64813 → 80 [PSH, ACK] Seq=1 Ack=1 Win=
31	3.225473	192.168.0.15	128.119.245.12	TCP	1514	64813 → 80 [ACK] Seq=697 Ack=1 Win=13
32	3.225473	192.168.0.15	128.119.245.12	TCP	1514	64813 → 80 [ACK] Seq=2145 Ack=1 Win=1
33	3.281283	8.8.8.8	192.168.0.15	DNS	93	Standard query response 0x9791 A gaia
34	3.281562	192.168.0.15	128.119.245.12	TCP	78	64814 → 80 [SYN] Seq=0 Win=65535 Len=
35	3.436221	192.168.0.11	239.255.255.250	SSDP	167	M-SEARCH * HTTP/1.1

Below the packet list, the details pane shows the structure of the selected packet (Frame 23):

- Frame 23: 78 bytes on wire (624 bits), 78 bytes captured (624 bits) on interface 0
- Ethernet II, Src: Apple\_c0:87:6a (78:31:c1:c0:87:6a), Dst: CiscoSpv\_3a:63:77 (bc:d1:65:3a:63:77)
- Internet Protocol Version 4, Src: 192.168.0.15, 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: 64
  - Identification: 0x2783 (10115)
  - Flags: 0x02 (Don't Fragment)
  - Fragment offset: 0
  - Time to live: 64
  - Protocol: TCP (6)
  - Header checksum: 0xdcf9 [validation disabled]
  - [Header checksum status: Unverified]
  - Source: 192.168.0.15
  - Destination: 128.119.245.12
  - [Source GeoIP: Unknown]
  - [Destination GeoIP: Unknown]
- Transmission Control Protocol, Src Port: 64813, Dst Port: 80, Seq: 0, Len: 0
  - Source Port: 64813
  - Destination Port: 80
  - [Stream index: 5]
  - [TCP Segment Len: 0]
  - Sequence number: 0 (relative sequence number)
  - Acknowledgment number: 0
  - 1011 .... = Header Length: 44 bytes (11)
  - Flags: 0x002 (SYN)
  - Window size value: 65535
  - [Calculated window size: 65535]
  - Checksum: 0xdb78 [unverified]
  - [Checksum Status: Unverified]
  - Urgent pointer: 0
  - Options: (24 bytes), Maximum segment size, No-Operation (NOP), Window scale, No-Operation (NOP), No-Operation (NOP),

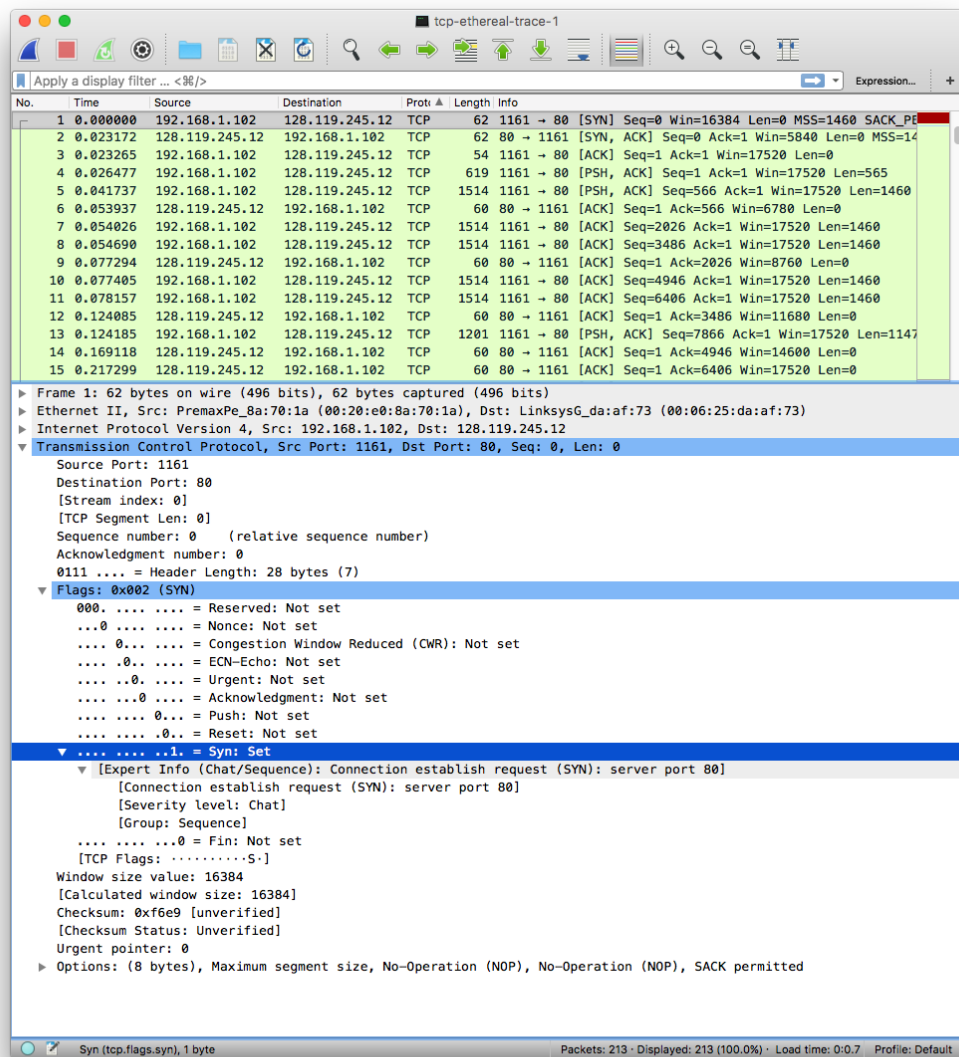
The status bar at the bottom indicates: Internet Protocol Version 4 (ip), 20 bytes. Packets: 333 · Displayed: 333 (100.0%) · Dropped: 0 (0.0%) · Profile: Default

## 2 TCP Basics

- 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?

The sequence number used to initiate the TCP connection is 0. The **SYN** flag highlighted below indicates that the segment is a SYN segment.

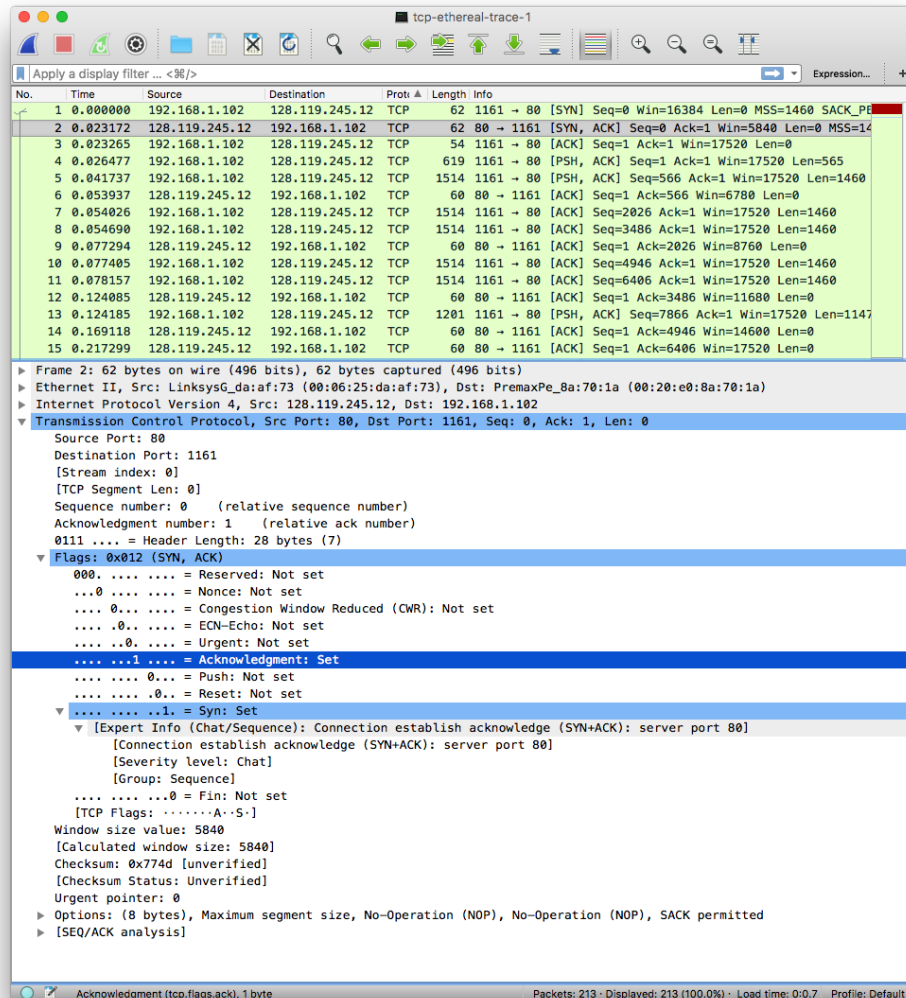
Figure 3: tcplocal-trace-1 (Question 04)



- 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?

The sequence number of the SYNACK segment sent by gaia.cs.umass.edu to the client computer is 0. The value of the ACK filed in the SYNACK segment is 1 as highlighted below. gaia.cs.umass.edu determined the value by adding 1 to the initial sequence number of SYN segment (0). Both the **SYN** and the **Acknowledgment** flags are set to 1, which indicates that this is a SYNACK segment.

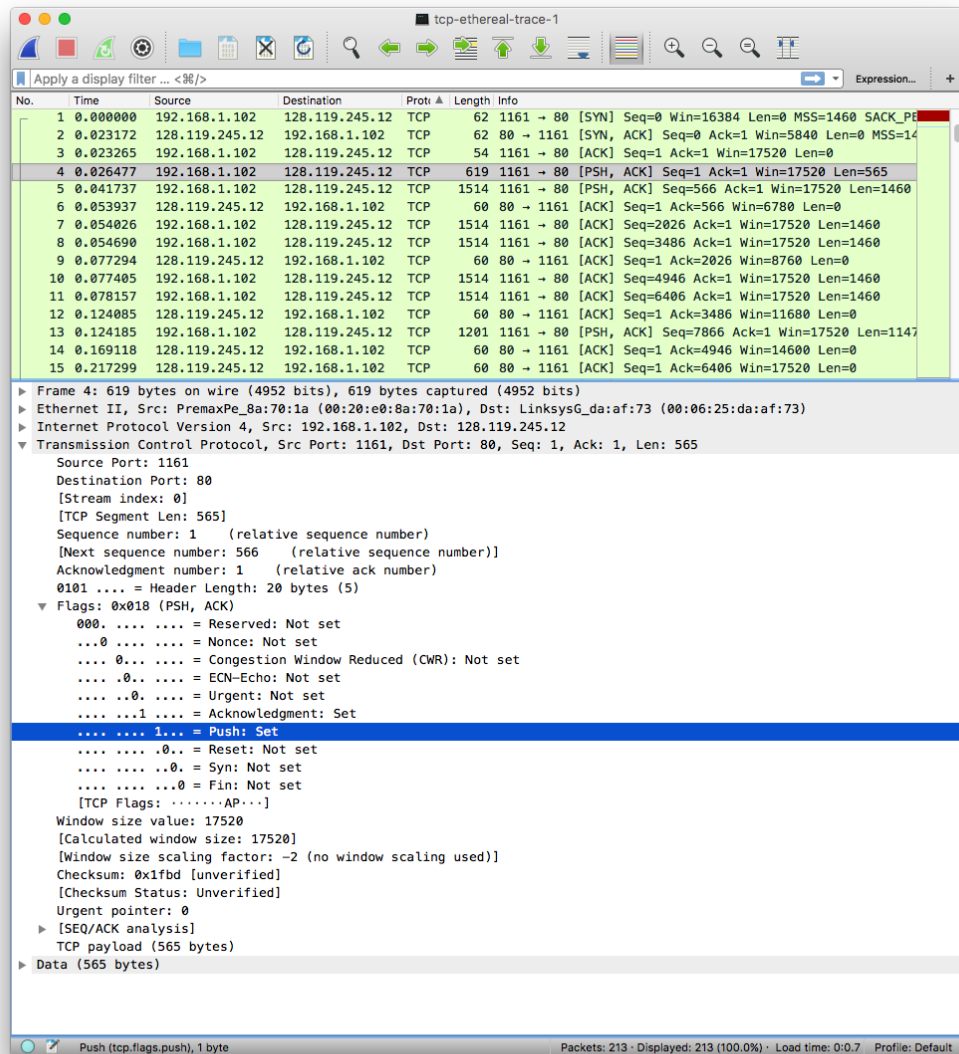
Figure 4: tcplocal-trace-1 (Question 05)



- 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.

The sequence number of the TCP segment containing the POST command is 1. The segment is highlighted below and has the flag **Push** set to 1 indicating that the client is sending data.

Figure 5: tcplocal-trace-1 (Question 06)



- 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.

The sequence numbers of the first six segments in the TCP connection are 1, 566, 2026, 3486, 4946 and 6406 (frames 4, 5, 7, 8, 9 and 10) in the trace. The ACK received for each one the segments are frames 6, 9, 12, 14, 15 and 16, respectively.

Table 1: RTT values (seconds)

Segment	Frame	Sent time (seconds)	ACK received (seconds)	RTT (seconds)
1	4	0.026477	0.053937	0.02746
2	5	0.041737	0.077294	0.035557
3	7	0.054026	0.124085	0.070059
4	8	0.054690	0.169118	0.11443
5	10	0.077405	0.217299	0.13989
6	11	0.078157	0.267802	0.18964

The EstimatedRTT equation on page 239 is

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

Segment 1:  $\text{EstimatedRTT} = 0.02746 \text{ secs}$

Segment 2:  $\text{EstimatedRTT} = 0.875 * 0.02746 + 0.125 * 0.035557 = 0.0285 \text{ secs}$

Segment 3:  $\text{EstimatedRTT} = 0.875 * 0.0285 + 0.125 * 0.070059 = 0.0337 \text{ secs}$

Segment 4:  $\text{EstimatedRTT} = 0.875 * 0.0337 + 0.125 * 0.11443 = 0.0438 \text{ secs}$

Segment 5:  $\text{EstimatedRTT} = 0.875 * 0.0438 + 0.125 * 0.13989 = 0.0558 \text{ secs}$

Segment 6:  $\text{EstimatedRTT} = 0.875 * 0.0558 + 0.125 * 0.18964 = 0.0725 \text{ secs}$



Figure 6: tcplocal-trace-1 (Question 07)

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_P
2	0.023172	128.119.245.12	192.168.1.102	TCP	62	80 → 1161 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=14
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=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=114
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
18	0.305040	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=9013 Ack=1 Win=17520 Len=1460
19	0.305813	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=10473 Ack=1 Win=17520 Len=1460
20	0.306692	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=11933 Ack=1 Win=17520 Len=1460
21	0.307571	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=13393 Ack=1 Win=17520 Len=1460
22	0.308699	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=14853 Ack=1 Win=17520 Len=1460
23	0.309553	192.168.1.102	128.119.245.12	TCP	946	1161 → 80 [PSH, ACK] Seq=16313 Ack=1 Win=17520 Len=89
24	0.356437	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=10473 Win=26280 Len=0
25	0.400164	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=11933 Win=29200 Len=0
26	0.448613	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=13393 Win=32120 Len=0
27	0.500029	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=14853 Win=35040 Len=0
28	0.545052	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=16313 Win=37960 Len=0
29	0.576417	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=17205 Win=37960 Len=0
30	0.576671	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=17205 Ack=1 Win=17520 Len=1460
31	0.577385	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=18665 Ack=1 Win=17520 Len=1460
32	0.578329	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=20125 Ack=1 Win=17520 Len=1460
33	0.579195	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=21585 Ack=1 Win=17520 Len=1460
34	0.580149	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=23045 Ack=1 Win=17520 Len=1460
35	0.581074	192.168.1.102	128.119.245.12	TCP	946	1161 → 80 [PSH, ACK] Seq=24505 Ack=1 Win=17520 Len=89
36	0.626496	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=18665 Win=40880 Len=0
37	0.672796	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=20125 Win=43800 Len=0
38	0.730684	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=21585 Win=46720 Len=0
39	0.772990	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=23045 Win=49640 Len=0
40	0.820622	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=24505 Win=52560 Len=0

▶ Frame 16: 60 bytes on wire (480 bits), 60 bytes captured (480 bits)  
 ▶ Ethernet II, Src: LinksysG\_da:af:73 (00:06:25:da:af:73), Dst: PremaxPe\_8a:70:1a (00:20:e0:8a:70:1a)  
 ▶ Internet Protocol Version 4, Src: 128.119.245.12, Dst: 192.168.1.102  
 ▶ Transmission Control Protocol, Src Port: 80, Dst Port: 1161, Seq: 1, Ack: 7866, Len: 0

Internet Protocol Version 4 (ip), 20 bytes      Packets: 213 · Displayed: 213 (100.0%) · Load time: 0:0.3 · Profile: Default



- What is the length of each of the first six TCP segments?

The length of the first TCP segment (including the HTTP POST) is 565 bytes. The length of each one of the other five segments is 1460 bytes. This value is included within the TCP section as highlighted below but can also be quickly identified by checking the value of **Len** in the column **Info** for each one of the segments in the trace.

Figure 7: tcplocal-trace-1 (Question 08)

No.	Time	Source	Destination	Prot	Len	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=1
2	0.023172	128.119.245.12	192.168.1.102	TCP	62	80 → 1161 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460
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=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

Frame 4: 619 bytes on wire (4952 bits), 619 bytes captured (4952 bits)

Ethernet II, Src: PremaxPe\_Ba:70:1a (00:20:e0:8a:70:1a), Dst: LinksysG\_da:af:73 (00:06:25:da:af:73)

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

Transmission Control Protocol, Src Port: 1161, Dst Port: 80, Seq: 1, Ack: 1, Len: 565

Source Port: 1161  
Destination Port: 80  
[Stream index: 0]

[TCP Segment Len: 565]

Sequence number: 1 (relative sequence number)  
[Next sequence number: 566 (relative sequence number)]  
Acknowledgment number: 1 (relative ack number)  
0101 .... = Header Length: 20 bytes (5)  
▼ Flags: 0x018 (PSH, ACK)  
000. .... = Reserved: Not set  
...0 .... = Nonce: Not set  
...0 .... = Congestion Window Reduced (CWR): 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...]  
Window size value: 17520  
[Calculated window size: 17520]  
[Window size scaling factor: -2 (no window scaling used)]  
Checksum: 0x1fbd [unverified]  
[Checksum Status: Unverified]  
Urgent pointer: 0  
► [SEQ/ACK analysis]  
TCP payload (565 bytes)  
► Data (565 bytes)

TCP Segment Len (tcp.len), 1 byte

Packets: 213 · Displayed: 213 (100.0%) · Load time: 0:0.7 · Profile: Default

- 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 advertised at the received (gaia.cs.umass.edu) for the entire trace is 5840 bytes as indicated by the field **Window size value** in the first ACK. The window size value grows until it reaches the value 62780 bytes as indicated in the last POST segment. By inspecting the trace it doesn't seem like the sender ever throttle due to lack of buffer space.

Figure 8: tcplocal-trace-1 (Question 09 - First ACK)

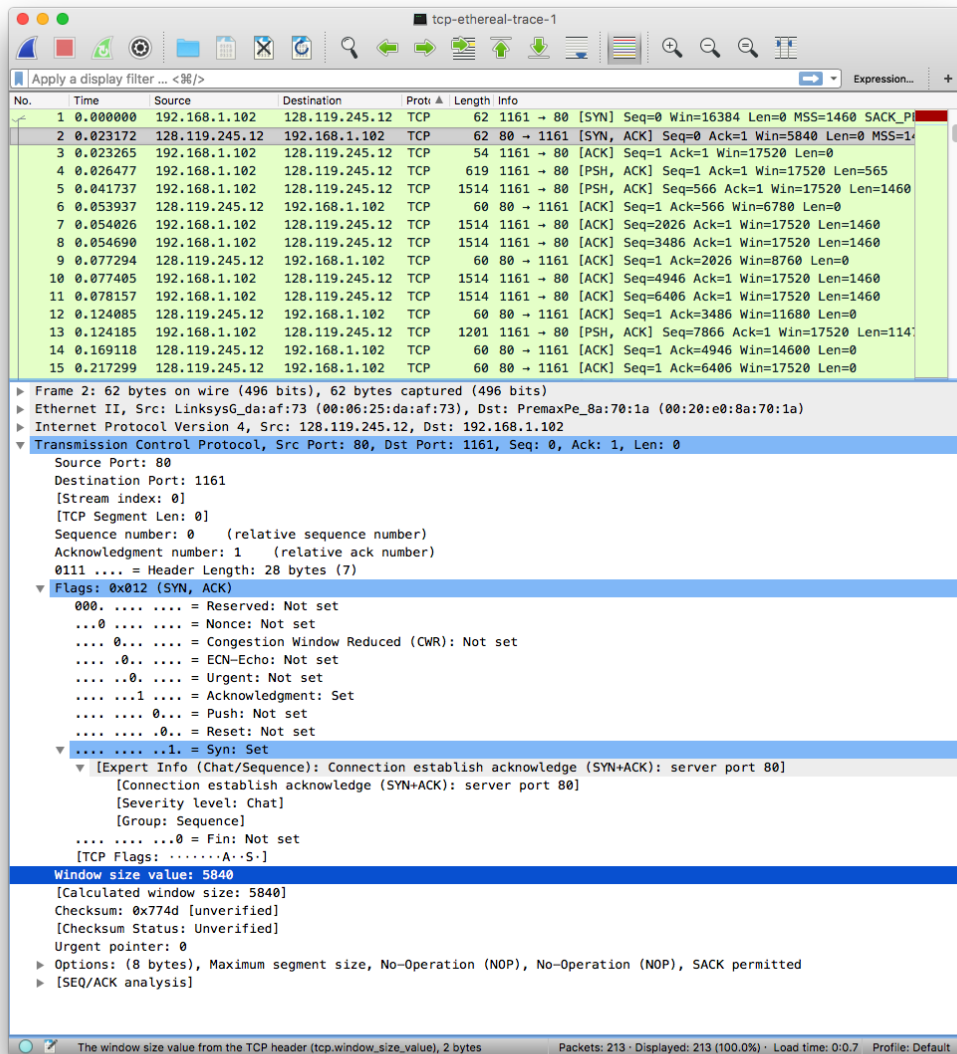
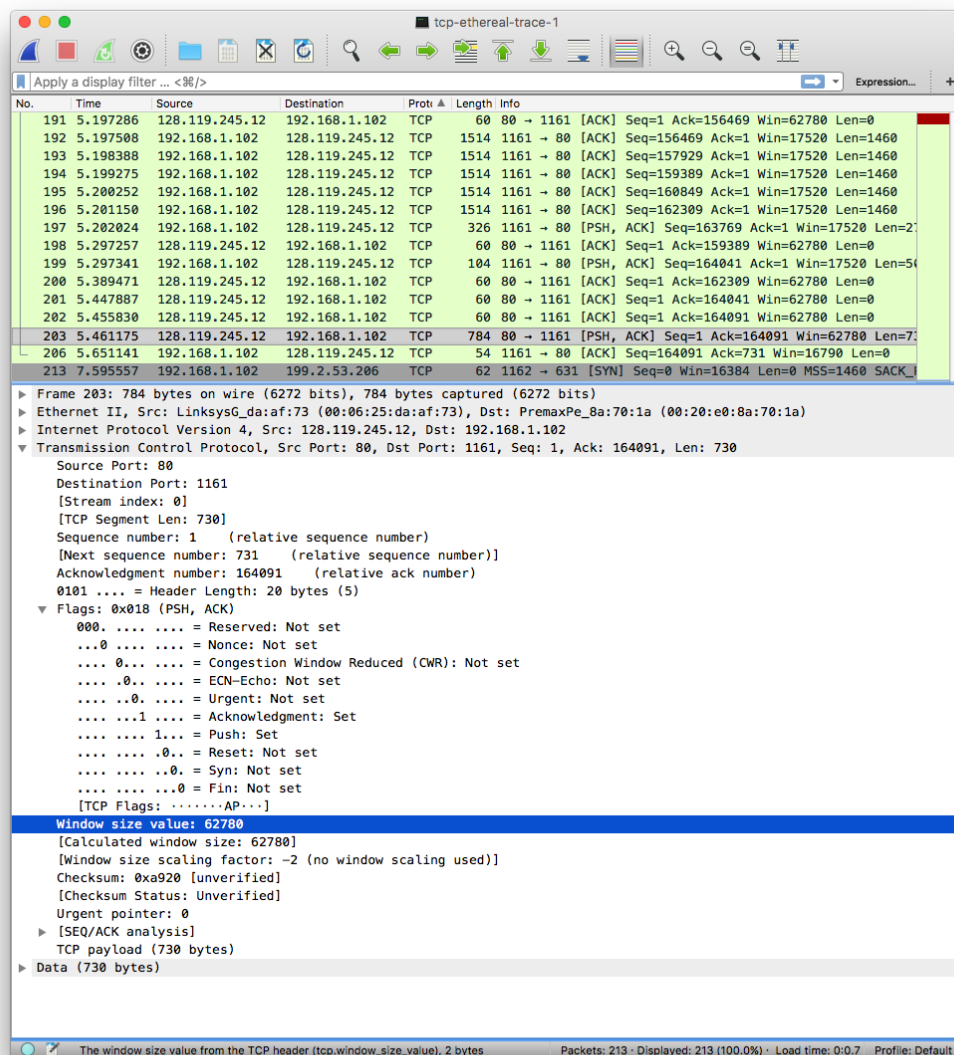


Figure 9: tcplocal-trace-1 (Question 09 - Last POST segment)



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

There are no retransmitted segments in the trace file. There aren't any wireshark frames highlighted as retransmitted [TCP Retransmission]. Additionally, all sequence numbers increases as packets are sent. When a segment is retransmitted, the sequence number of the segment is smaller than the sequence number of its nearby segments.

- *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).*

The receiver typically acknowledge 1460 bytes in an ACK.

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

The average throughput is computed as the ratio between the total amount data and the total transmission time.

**Total amount data:**

Sequence number of the last ACK minus the seq number of the first TCP segment

164091 - 1

164090 bytes.

**Total transmission time:**

Time instant of the last ACK minus the time instant of the first TCP segment

5.455830 - 0.026477

5.4294 seconds.

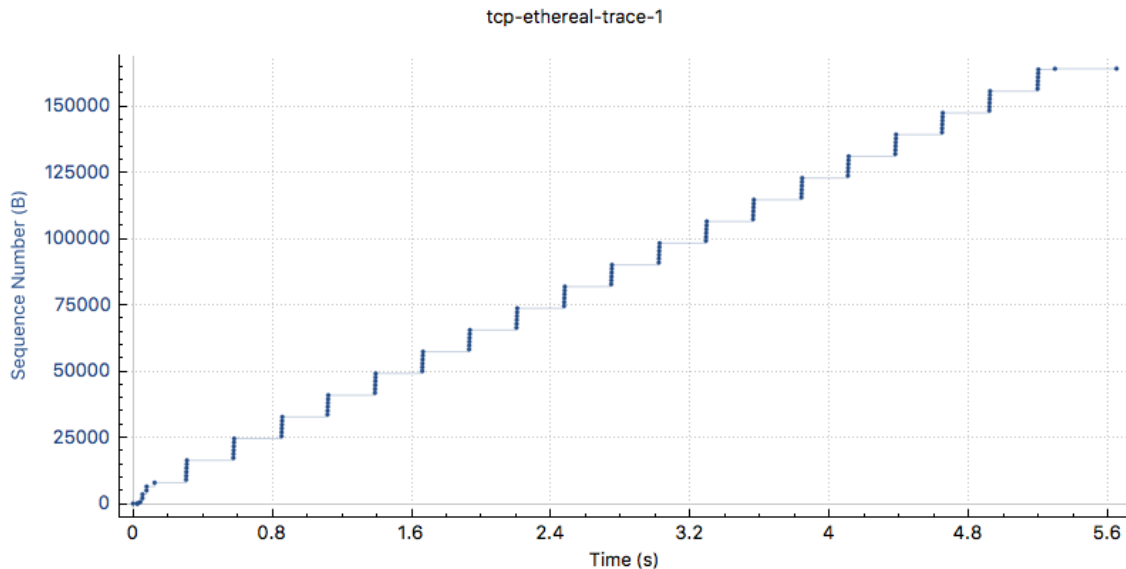
Therefore, the throughput for the TCP connection is  $164090/5.4294 = \mathbf{30.222 \text{ Kbs/sec}}$ .

### 3 TCP congestion control in action

- 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.

Figure 10: tcplocal-trace-1 (Question 13 - Time-Sequence-Graph(Stevens))

**Sequence Numbers (Stevens) for 192.168.1.102:1161 → 128.119.245.12:80**



- 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`

There's no way to determine the end of the slow start phase and the start of the congestion avoidance since the client is not sending enough data to force a congestion state. The client stops transmitting data before any congestion.