1. **What are the first and last packets for the POST request?**

   First: 199, Last: 203

```
199 5.297341  192.168.1.102  128.119.245.12 HTTP   104 POST /ethereal-labs/lab3-1-reply.htm HTTP/1.1  (text/plain)
200 5.389471  128.119.245.12 192.168.1.102  TCP     60 80 → 1161 [ACK] Seq=1 Ack=162309 Win=62780 Len=0
201 5.447887  128.119.245.12 192.168.1.102  TCP     60 80 → 1161 [ACK] Seq=1 Ack=164041 Win=62780 Len=0
202 5.455830  128.119.245.12 192.168.1.102  TCP     60 80 → 1161 [ACK] Seq=1 Ack=164091 Win=62780 Len=0
203 5.461175  128.119.245.12 192.168.1.102  HTTP   784 HTTP/1.1 200 OK  (text/html)
```

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

   IP: 192.168.0.102

   Port: 1161

```
Transmission Control Protocol, Src Port: 1161, Dst Port: 80, Seq: 164041, Ack: 1, Len: 50
  Source Port: 1161
```

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

   IP: 128.119.245.12

   Port: 80

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: 232129012.

   The SYN flag is set to 1.

```
Transmission Control Protocol, Src Port: 1161, Dst Port: 80, Seq: 232129012, Len: 0
  Source Port: 1161
  Destination Port: 80
  [Stream index: 0]
  [TCP Segment Len: 0]
  Sequence Number: 232129012
  [Next Sequence Number: 232129013]
  Acknowledgment Number: 0
  Acknowledgment number (raw): 0
  0111 .... = Header Length: 28 bytes (7)
  Flags: 0x002 (SYN)
```

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: 883061785.

   ACKnowledgement field: 232129013

   It determined that value by adding 1 to the last sequence number.

   SYN and ACK flags are set to 1.

```
Transmission Control Protocol, Src Port: 80, Dst Port: 1161, Seq: 883061785, Ack: 232129013, Len: 0
  Source Port: 80
  Destination Port: 1161
  [Stream index: 0]
  [TCP Segment Len: 0]
  Sequence Number: 883061785
  [Next Sequence Number: 883061786]
  Acknowledgment Number: 232129013
  0111 .... = Header Length: 28 bytes (7)
  Flags: 0x012 (SYN, 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
    .... .... 0... = Push: Not set
    .... .... .0.. = Reset: Not set
    .... .... ..1. = Syn: Set
```

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

   Sequence number: 232293053

```
199 5.297341  192.168.1.102  128.119.245.12  HTTP  104 POST /ethereal-labs/lab3-1-reply.htm HTTP/1.1  (text/plain)

Frame 199: 104 bytes on wire (832 bits), 104 bytes captured (832 bits)
Ethernet II, Src: Actionte_8a: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: 232293053, Ack: 883061786, Len: 50
  Source Port: 1161
  Destination Port: 80
  [Stream index: 0]
  [TCP Segment Len: 50]
  Sequence Number: 232293053
```

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 269 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 270 for all subsequent segments.**

   6 first segments sequence numbers:
   1, 566, 2026, 3486, 4946, 6406.

   Times:
   0.026477, 0.041737, 0.054026, 0.054690, 0.077405, 0.078157.

   ACK segments received:
   0.053937, 0.077294, 0.124085, 0.169118, 0.217299, 0.267802.

   RTT:
   0.027460, 0.0356, 0.0701, 0.1144, 0.1399, 0.1896.

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 [TCP segment of a reassembled PDU]
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 [TCP segment of a reassembled PDU]
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 [TCP segment of a reassembled PDU]
8	0.054690	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=3486 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]
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 [TCP segment of a reassembled PDU]
11	0.078157	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=6406 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]
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 [TCP segment of a reassembled PDU]
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

8. **What is the length of each of the first six TCP segments?**
   565, 1460, 1460, 1460, 1460, 1460

9. **What is the minimum amount of available buffer space advertised at the receiver for the entire trace? Does the lack of receiver buffer space ever throttle the sender?**
   5840 is the smallest window. No segment is larger than this.

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 SAC
3	0.023265	192.168.1.102	128.119.245.12	TCP	54	1161 → 80 [ACK] Seq=1 Ack=1 Win=17520 Len=0

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, because no sequence number appear twice.

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 278 in the text).**
    A burst of 6 packets contains 8192 bytes. Three ACKs for these gives 2730 bytes per ACK. And we find that 6 packets are transmitted and are ACKed by three packets.

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

    The entire transfer took 5.461175 seconds, and the total number of bytes is 163788.
    163788 / 5,434698 = 30kB/s

Connections at a high level: TCP uses ACKs to keep track of sent data. It can be used to estimate to total number of sent bytes. The available buffer space decides how many packets can be sent at once.

The impact of RTT estimates, packet losses, and interpreted packet loss events: We can use RTT estimates to optimize how we send the data. If the RTT is too long, we might want to increase the congestion-window to minimize the amount of ACKs between each packet. Packet losses are an indication that there are some congestion issues stopping the packets from reaching the receiver. If this is the case, we might want to decrease the congestion-window.

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 if and where TCP's *slow start* phase begins and ends, as well as if 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.**

    Starts at: 0.026477, Ends at: 0.07816.
    Congestion avoidance starts after the TCP slow start.
    It does no seem to differ from the text book examples.

14. **Explain the relationship between (i) the congestion window (`cwnd`), (ii) the receiver advertised window (`rwnd`), (iii) the number of unacknowledged bytes, and (iv) the effective window at the sender (i.e., the window effectively limiting the data transmission).**

    (i) The congestion window is the number of packets that can be sent at once. (ii) The receiver advertised window is the maximum number of packets that the receiver can receive. (iv) is the smallest of (i) and (ii). (iii) is a deciding factor in the size of cwnd.

15. **Is it generally possible to find the congestion window size (`cwnd`) and how it changes with time, from the captured trace files? If so, please explain how. If not, please explain when and when not. Motivate your answer and give examples.**

    No, because it is hard to know if the cwnd is determined by the senders capacity or the capacity of the network. For example, is the current transmitting speed determined by packet loss or the fact that we reached maximum capacity of the sender.

16. **What is the throughput of each of the connections in bps (bits per second)? What is the total bandwidth of the host on which the clients are running? Discuss the TCP fairness for this case**

1.	316882 b/s	3.8M
2.	318316 b/s	3.8M
3.	321904 b/s	3.8M
4.	318819 b/s	3.8M

Total bandwidth: 1275921b/s (sum of connections)

It seems fair, because the connections have roughly the same throughput. The formula gives ~3800000 for all of them so its fair.

17. **What is the throughput of each of the connections in bps (bits per second)? What is the total bandwidth of the host on which the clients are running? Discuss the TCP fairness for this case.**

1.	2.9 Mb/s	38M
2.	2 Mb/s	70M
3.	1.7 Mb/s	116M
4.	1.6 Mb/s	117M
5.	1.2 Mb/s	59M
6.	0.8 Mb/s	26M
7.	0.7 Mb/s	95M
8.	0.5 Mb/s	163M
9.	0.4 Mb/s	129M

Total bandwidth: 11.8 mb/s (sum of connections)
Since the only difference is RTT, some connections have much lower throughput. Since fairness is based on a connection's access to a network's resources, this is an unfair network. The formula gives very different numbers so this is a unfair network.

18. **Discuss the TCP fairness for this case. How does it differ from the previous cases, and how is it affected by the use of BitTorrent?**
It is hard to determine fairness in a peer-to-peer system, since the peers might have different network connections. In this case, packet loss and maximum segment size are the same and only RTT varies, but in the real world, it would be even more complex to determine the fairness, since these variables would vary. A host would probably distribute their own bandwidth equally, and not take responsibility for how this affects the throughput for the individual clients.