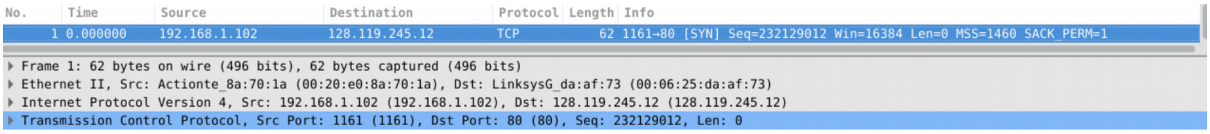
Question 1

Question 1. What is the IP address of gaia.cs.umass.edu? On what port number is it sending and receiving TCP segments for this connection? 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?

Answer



```
No.   Time        Source           Destination      Protocol Length Info
    1 0.000000    192.168.1.102    128.119.245.12   TCP          62 1161→80 [SYN] Seq=232129012 Win=16384 Len=0 MSS=1460 SACK_PERM=1

▸ Frame 1: 62 bytes on wire (496 bits), 62 bytes captured (496 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 (192.168.1.102), Dst: 128.119.245.12 (128.119.245.12)
▸ Transmission Control Protocol, Src Port: 1161 (1161), Dst Port: 80 (80), Seq: 232129012, Len: 0
```

Source: My computer

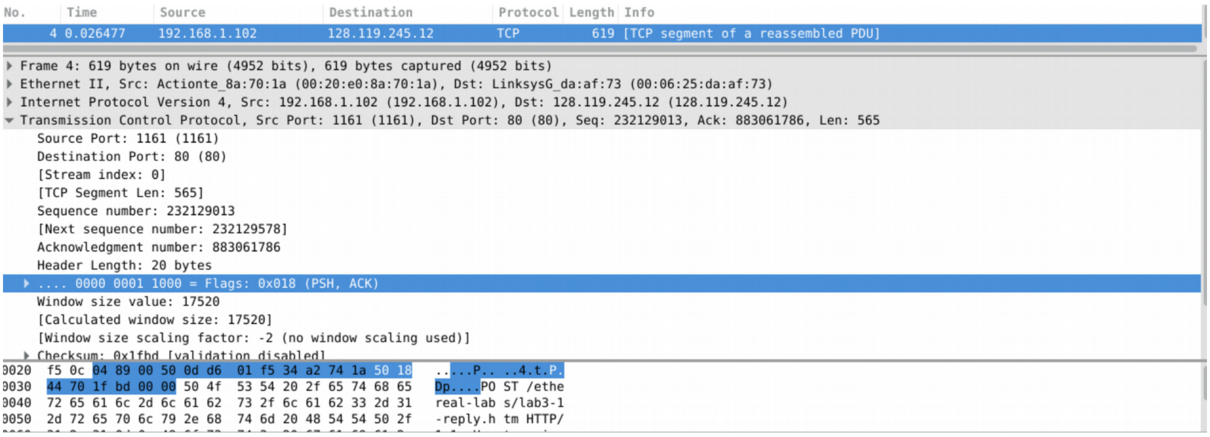
IP: 192.168.1.102 **TCP port number:** 1161

Destination: gaia.cs.umass.ed

IP: 128.119.245.112 **TCP port number:** 80

Question 2. 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 Ethereal window, looking for a segment with a "POST" within its DATA field.

Answer



```
No.   Time        Source           Destination      Protocol Length Info
    4 0.026477    192.168.1.102    128.119.245.12   TCP         619 [TCP segment of a reassembled PDU]

▸ Frame 4: 619 bytes on wire (4952 bits), 619 bytes captured (4952 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 (192.168.1.102), Dst: 128.119.245.12 (128.119.245.12)
▾ Transmission Control Protocol, Src Port: 1161 (1161), Dst Port: 80 (80), Seq: 232129013, Ack: 883061786, Len: 565
    Source Port: 1161 (1161)
    Destination Port: 80 (80)
    [Stream index: 0]
    [TCP Segment Len: 565]
    Sequence number: 232129013
    [Next sequence number: 232129578]
    Acknowledgment number: 883061786
    Header Length: 20 bytes
  ▸ .... 0000 0001 1000 = Flags: 0x018 (PSH, ACK)
    Window size value: 17520
    [Calculated window size: 17520]
    [Window size scaling factor: -2 (no window scaling used)]
  ▸ Checksum: 0x1fbd [validation disabled]
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/
```

Segment #4 contains the HTTP POST command. The sequence number of the packet is 232129013

Question 3. 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) sent from the client to the web server (Do not consider the ACKs received from the server as part of these six segments)?

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 relevant parts of Section 3.5 or lecture slides) after the receipt of each ACK?

Assume that the initial value of EstimatedRTT is equal to the measured RTT (SampleRTT) for the first segment, and then is computed using the EstimatedRTT equation for all subsequent segments. Set alpha to 0.125.

Answer

Since the HTTP Post segment is considered to be the first segment in the TCP connection, segments 1 to 6 are: 4,5,7,8 and 10 respectively in the trace.

Segment 1 sequence number: 232129013

Segment 2 sequence number: 232129578

Segment 3 sequence number: 232131038

Segment 4 sequence number: 232132498

Segment 5 sequence number: 232133958

Segment 6 sequence number: 232135418

Segment	Sent time	ACK received	RTT (secs)
Segment 1	0.026477	0.053937	0.02746
Segment 2	0.041737	0.077294	0.035557000
Segment 3	0.054026	0.124085	0.070059000
Segment 4	0.054690	0.169188	0.114428000
Segment 5	0.077405	0.217299	0.139894000
Segment 6	0.078157	0.267802	0.189645000

$$Estimated\ RTT = (1-\alpha) * Estimated\ RTT + \alpha * Sample\ RTT$$

Segment 1:

$$Estimated\ RTT = RTT = 0.02746\ secs$$

Segment 2:

$$Estimated\ RTT = 0.875 * 0.02746 + 0.125 * 0.035557 = 0.0285\ secs$$

Segment 3:

$$Estimated\ RTT = 0.875 * 0.\ 0285 + 0.125 * 0.\ 070059 = 0.\ 0337\ secs$$

Segment 4:

$$Estimated\ RTT=0.875*0.0337+0.125*0.11443=0.0438\ sec\ s$$

Segment 5:

$$Estimated\ RTT=0.875*0.0438+0.125*0.1389=0.0558\ secs$$

Segment 6:

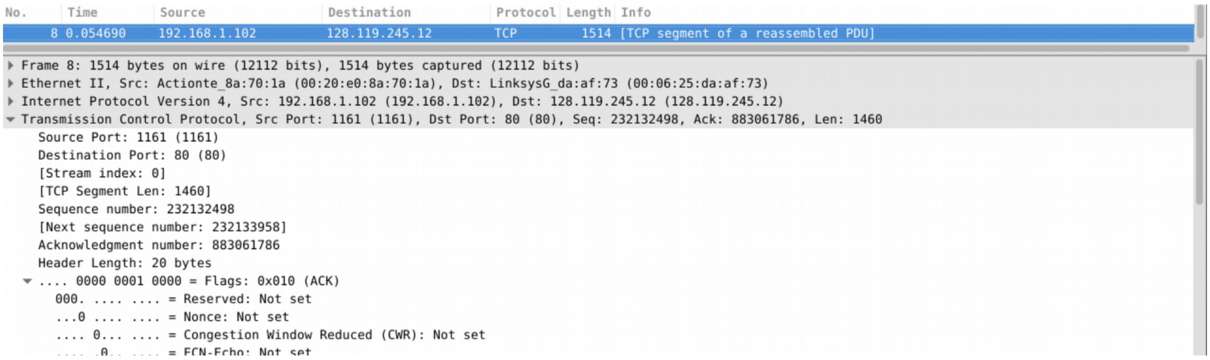
$$Estimated\ RTT=0.875*0.0558+0.125*0.18964=0.0725\ secs$$

Question 4. What is the length of each of the first six TCP segments?

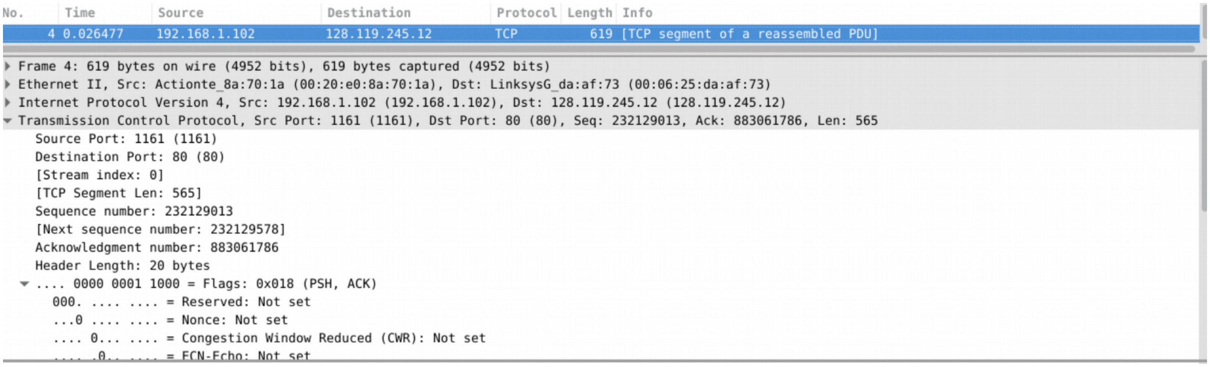
Answer

Length of first TCP segment (sending the HTTP Post request): 565 bytes

Length of other segments: 1460 bytes



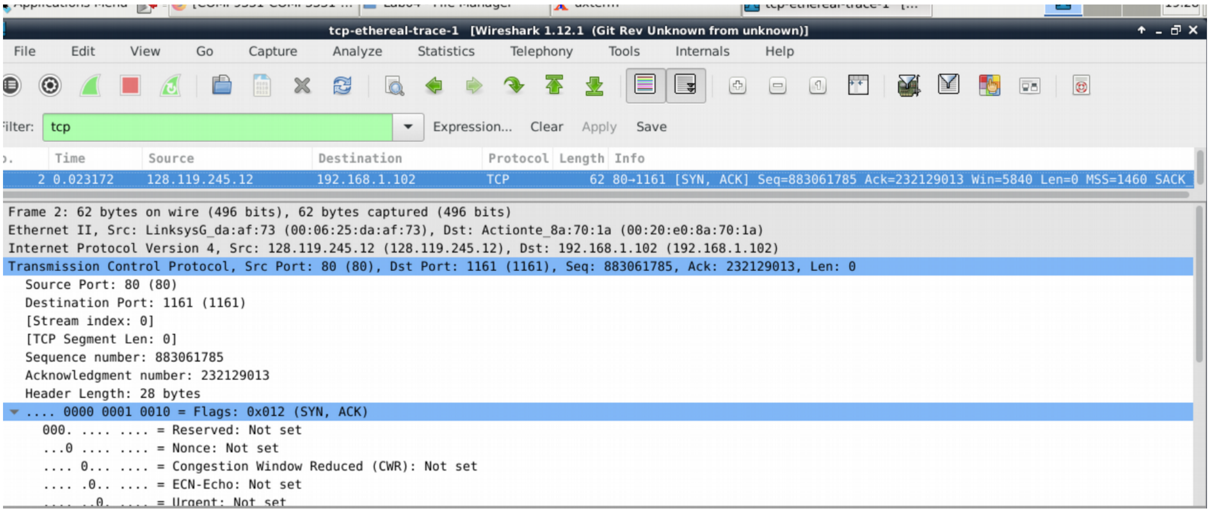
```
No.     Time          Source              Destination         Protocol Length Info
      8 0.054690      192.168.1.102       128.119.245.12      TCP        1514 [TCP segment of a reassembled PDU]

▸ Frame 8: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 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 (192.168.1.102), Dst: 128.119.245.12 (128.119.245.12)
▾ Transmission Control Protocol, Src Port: 1161 (1161), Dst Port: 80 (80), Seq: 232132498, Ack: 883061786, Len: 1460
    Source Port: 1161 (1161)
    Destination Port: 80 (80)
    [Stream index: 0]
    [TCP Segment Len: 1460]
    Sequence number: 232132498
    [Next sequence number: 232133958]
    Acknowledgment number: 883061786
    Header Length: 20 bytes
  ▾ .... 0000 0001 0000 = Flags: 0x010 (ACK)
      000. .... .... = Reserved: Not set
      ...0 .... .... = Nonce: Not set
      .... 0... .... = Congestion Window Reduced (CWR): Not set
      .... .0.. .... = ECN-Echo: Not set
```



```
No.     Time          Source              Destination         Protocol Length Info
      4 0.026477      192.168.1.102       128.119.245.12      TCP         619 [TCP segment of a reassembled PDU]

▸ Frame 4: 619 bytes on wire (4952 bits), 619 bytes captured (4952 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 (192.168.1.102), Dst: 128.119.245.12 (128.119.245.12)
▾ Transmission Control Protocol, Src Port: 1161 (1161), Dst Port: 80 (80), Seq: 232129013, Ack: 883061786, Len: 565
    Source Port: 1161 (1161)
    Destination Port: 80 (80)
    [Stream index: 0]
    [TCP Segment Len: 565]
    Sequence number: 232129013
    [Next sequence number: 232129578]
    Acknowledgment number: 883061786
    Header Length: 20 bytes
  ▾ .... 0000 0001 1000 = Flags: 0x018 (PSH, ACK)
      000. .... .... = Reserved: Not set
      ...0 .... .... = Nonce: Not set
      .... 0... .... = Congestion Window Reduced (CWR): Not set
      .... .0.. .... = ECN-Echo: Not set
```

Question 5. 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?

Answer


tcp-ethereal-trace-1 [Wireshark 1.12.1 (Git Rev Unknown from unknown)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter: tcp
Expression... Clear Apply Save

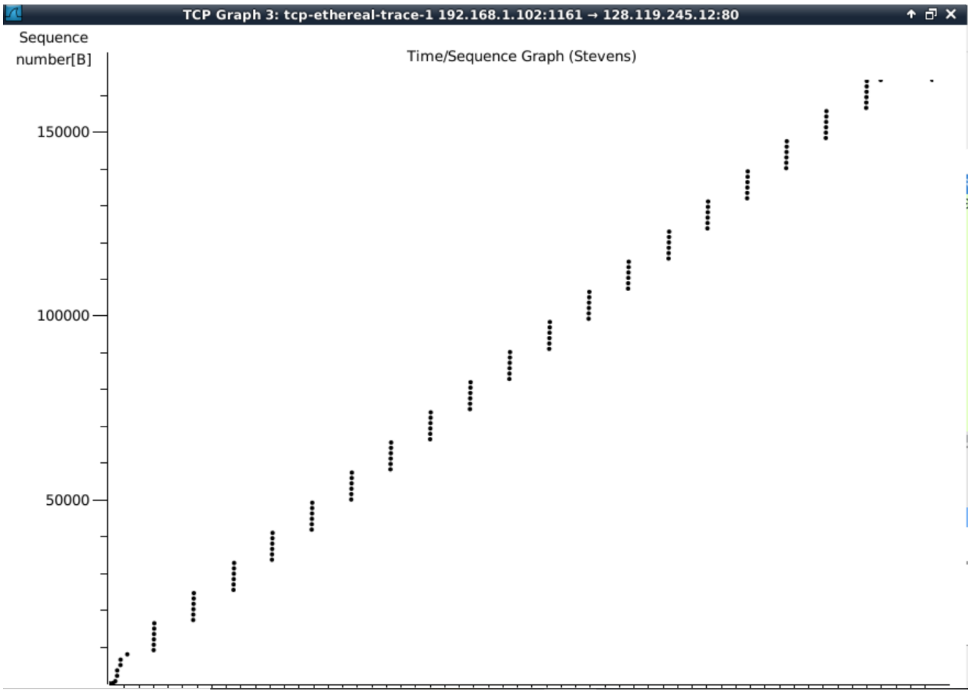
```
   Time       Source            Destination       Protocol Length Info
 2 0.023172   128.119.245.12    192.168.1.102     TCP          62 80→1161 [SYN, ACK] Seq=883061785 Ack=232129013 Win=5840 Len=0 MSS=1460 SACK

Frame 2: 62 bytes on wire (496 bits), 62 bytes captured (496 bits)
Ethernet II, Src: LinksysG_da:af:73 (00:06:25:da:af:73), Dst: Actionte_8a:70:1a (00:20:e0:8a:70:1a)
Internet Protocol Version 4, Src: 128.119.245.12 (128.119.245.12), Dst: 192.168.1.102 (192.168.1.102)
Transmission Control Protocol, Src Port: 80 (80), Dst Port: 1161 (1161), Seq: 883061785, Ack: 232129013, Len: 0
  Source Port: 80 (80)
  Destination Port: 1161 (1161)
  [Stream index: 0]
  [TCP Segment Len: 0]
  Sequence number: 883061785
  Acknowledgment number: 232129013
  Header Length: 28 bytes
  .... 0000 0001 0010 = 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
```

The minimum amount of receiver buffer space offered is the first acknowledgement of the server. It is 5840 bytes. The receiver window gradually grows over time to a maximum of 62780 bytes. Upon inspecting, the sender is never throttled.

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

Answer

There are no retransmitted segments in the trace file. This is verified by checking the sequence numbers of TCP segments in the trace file. If there is a retransmitted segment, the sequence number of the segment that is retransmitted has to be smaller than that of its neighbouring segments.



One can also look at the *Time-Sequence Graph* above. The graph shows that all sequence numbers are increasing over time, thus indicating the absence of retransmitted segments.

Question 7. How much data does the receiver typically acknowledge in an ACK? Can you identify cases where the receiver is ACKing every other received segment (recall the discussion about delayed acks from the lecture notes or Section 3.5 of the text).

Answer

ACK Number	ACK Sequence Number	Acknowledged data size
ACK 1	232129578	565
ACK 2	232131038	1460
ACK 3	23132498	1460
ACK 4	232133958	1460
ACK 5	232135418	1460
ACK 6	232136878	1460
ACK 7	232138025	1147
ACK 8	232139485	1460
ACK 9	232140945	1460
ACK 10	232142405	1460
ACK 11	232143865	1460
ACK 12	232145325	1460
And so on…		

The receiver typically acknowledges 1460 bytes as seen above. However, there are some exceptions (such as ACK 7). The difference between ACK number indicates the data received by the server between two ACKs.

Question 8. What is the throughput (bytes transferred per unit time) for the TCP connection? Explain how you calculated this value.

Answer

TCP throughput depends mainly on the selection of the average time period. The average throughput is calculated as follows: $Throughput = \frac{Total\, amount\ of\ data\ received}{Total\ transmission\, time}$

For this question, the total amount of data transmitted can be calculated by taking the difference between the sequence number of the first TCP segment transmitted (No. 4 segment) and the acknowledged sequence number of the lack ACK.

$$Total\ Data\, Transmitted = 232293103 - 232\,12\,9013 = 164090\, bytes$$

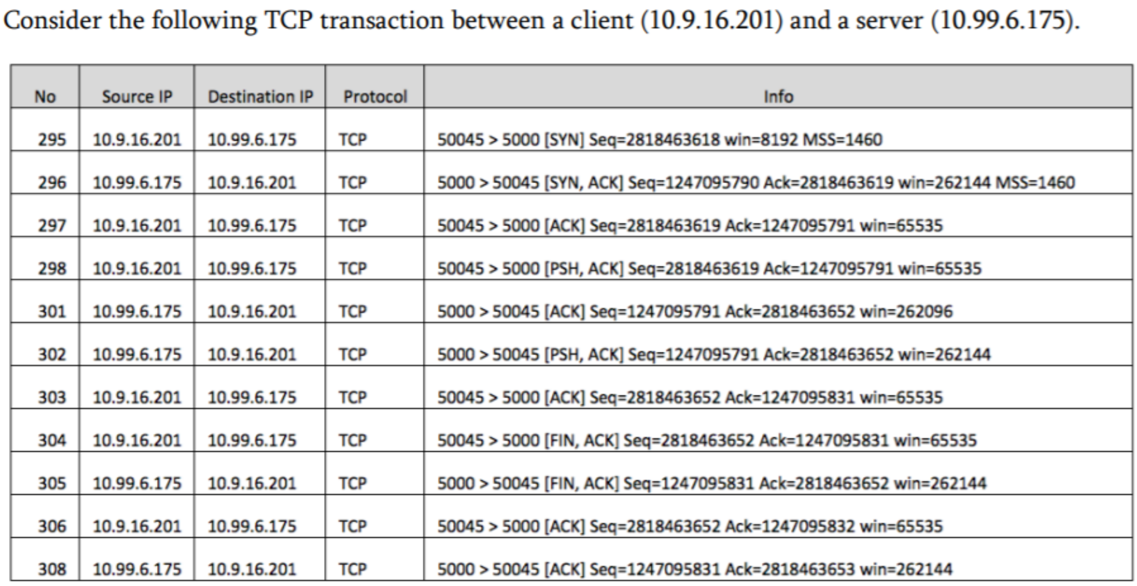
The total transmission time is the difference between the first TCP segment (Segment No. 4) and the instance of the last ACK.

$$Total\, Time\ Taken = 5.455830 - 0.026477 = 5.4294\ secs$$

So, the throughput is:

T $hroughput = \frac{164090}{5.4294} = 30.222\ KByte/sec$

Question 2



Consider the following TCP transaction between a client (10.9.16.201) and a server (10.99.6.175).

No	Source IP	Destination IP	Protocol	Info
295	10.9.16.201	10.99.6.175	TCP	50045 > 5000 [SYN] Seq=2818463618 win=8192 MSS=1460
296	10.99.6.175	10.9.16.201	TCP	5000 > 50045 [SYN, ACK] Seq=1247095790 Ack=2818463619 win=262144 MSS=1460
297	10.9.16.201	10.99.6.175	TCP	50045 > 5000 [ACK] Seq=2818463619 Ack=1247095791 win=65535
298	10.9.16.201	10.99.6.175	TCP	50045 > 5000 [PSH, ACK] Seq=2818463619 Ack=1247095791 win=65535
301	10.99.6.175	10.9.16.201	TCP	5000 > 50045 [ACK] Seq=1247095791 Ack=2818463652 win=262096
302	10.99.6.175	10.9.16.201	TCP	5000 > 50045 [PSH, ACK] Seq=1247095791 Ack=2818463652 win=262144
303	10.9.16.201	10.99.6.175	TCP	50045 > 5000 [ACK] Seq=2818463652 Ack=1247095831 win=65535
304	10.9.16.201	10.99.6.175	TCP	50045 > 5000 [FIN, ACK] Seq=2818463652 Ack=1247095831 win=65535
305	10.99.6.175	10.9.16.201	TCP	5000 > 50045 [FIN, ACK] Seq=1247095831 Ack=2818463652 win=262144
306	10.9.16.201	10.99.6.175	TCP	50045 > 5000 [ACK] Seq=2818463652 Ack=1247095832 win=65535
308	10.99.6.175	10.9.16.201	TCP	5000 > 50045 [ACK] Seq=1247095831 Ack=2818463653 win=262144

Question 1. What is the sequence number of the TCP SYN segment that is used to initiate the TCP connection between the client computer and server?

Answer

Sequence number: 2818463618

Question 2. What is the sequence number of the SYNACK segment sent by the server to the client computer in reply to the SYN? What is the value of the Acknowledgement field in the SYNACK segment? How did the server determine that value?

Answer

Sequence number: 1247095790

ACK field value: 2818463619

The value of the acknowledgement field in the SYNACK segment is determined by the server. The server adds 1 to the initial sequence number of SYN segment form the client computer. For this case, the initial sequence number of SYN segment from the client computer is 2818463618, thus the value of the ACKnowledgement field in the SYNACK segment is 2818463619.

Question 3. What is the sequence number of the ACK segment sent by the client computer in response to the SYNACK? What is the value of the Acknowledgment field in this ACK segment? Does this segment contain any data?

Answer

Sequence number: 2818463619

Acknowledgement field value: 1247095791

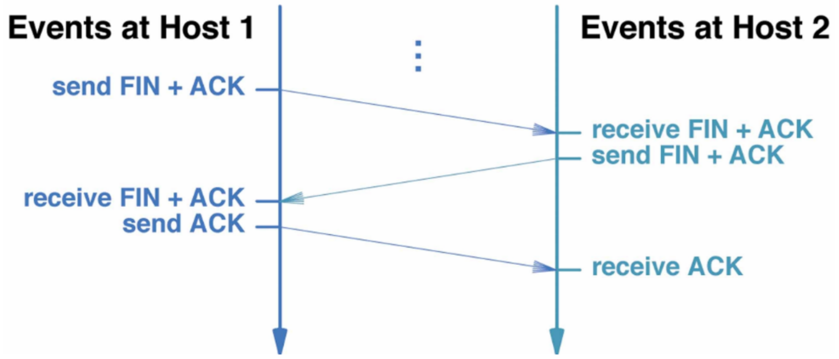
The segment does not contain 1 byte of data. This can be seen because the ACK field value of 1247095791 is one higher than the previous segment's sequence number value of 1247095790.

Question 4. Who has done the active close? client or the server? how you have determined this? What type of closure has been performed? 3 Segment (FIN/FINACK/ACK), 4 Segment (FIN/ACK/FIN/ACK) or Simultaneous close?

Answer

The client (10.9.16.201) sent the first [FIN,ACK] segment, indicating that it wanted to close the connection. Thus, the client performed the active close.

This is a 3 segment close, as it follows the example for a 3-way handshake termination of a connection.



Question 5. How many data bytes have been transferred from the client to the server and from the server to the client during the whole duration of the connection? What relationship does this have with the Initial Sequence Number and the final ACK received from the other side?

Answer

The total amount of data transmitted can be calculated by taking the difference between the sequence number of the first TCP segment transmitted and the acknowledged sequence number of the lack ACK.

$$Total\ Data\ Transmitted = 2818463653 - 2818463618 = 35\ bytes$$