Southern University of Science and Technology

Computer Networking Lab Report

唐润哲 11710418

- ***Introduction:***

#Problem7.1

Select one UDP packet from your trace. From this packet, determine

1) how many fields there are in the UDP header.
2) the name of each fields in the UDP header.
3) the length (in bytes) of each fields in the UDP header.
4) What is the maximum number of bytes that can be included in a UDP payload?(Hint: the answer to this question can be determined by your answer to 3) above)
5) What is the largest possible source port number? (Hint: same as the hint in 4) above.)
6) What is the protocol number for UDP?(Give your answer in both hexadecimal and decimal notation.)

# Problem 7.2

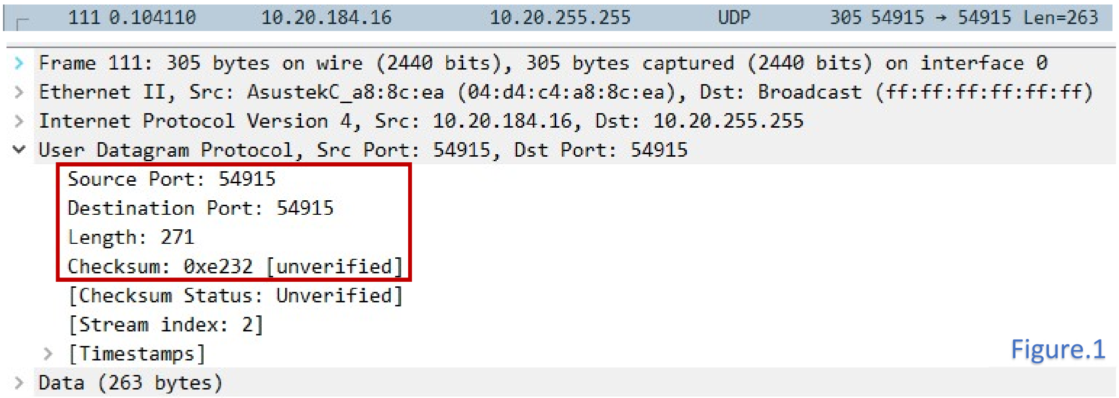
- 4.What is the sequence number of the TCP SYN segment that is used to initiate the TCP connection between the client computer and gaia.cs.umass.edu? What is it in the segment that identifies the segment as a SYN segment?
- 6. What is the sequence number of the TCP segment containing the HTTP POST command? 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.
- 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 242 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 Estimated RTT equation on page 242 for all subsequent segments.

- ➢ 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?
- ➢ 10. Are there any retransmitted segments in the trace file? What did you check for (in the trace) in order to answer this question?
- ➢ 12. What is the throughput (bytes transferred per unit time) for the TCP connection? Explain how you calculated this value.

- ***Result:***

Problem 7.1



```
  111 0.104110      10.20.184.16        10.20.255.255       UDP       305 54915 → 54915 Len=263

> Frame 111: 305 bytes on wire (2440 bits), 305 bytes captured (2440 bits) on interface 0
> Ethernet II, Src: AsustekC_a8:8c:ea (04:d4:c4:a8:8c:ea), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
> Internet Protocol Version 4, Src: 10.20.184.16, Dst: 10.20.255.255
v User Datagram Protocol, Src Port: 54915, Dst Port: 54915
    Source Port: 54915
    Destination Port: 54915
    Length: 271
    Checksum: 0xe232 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 2]
  > [Timestamps]
> Data (263 bytes)
```

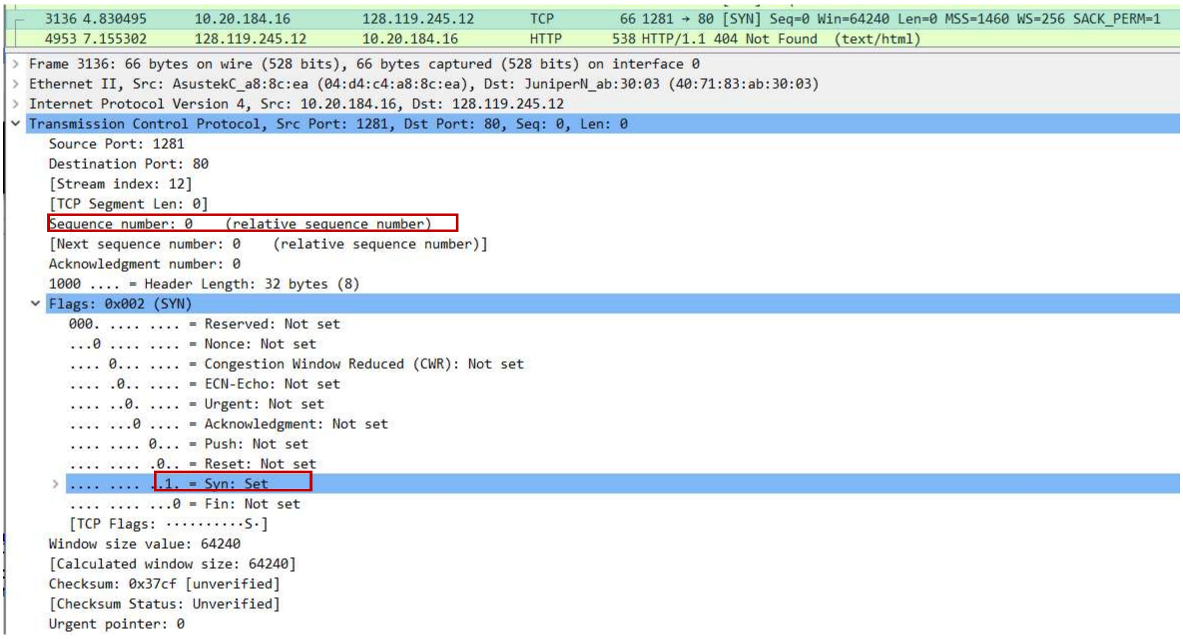
Figure.1

According to the Figure.1 captured above:

1) 4 fields in the UDP header
2) Source Port; Destination Port; Length; Checksum
3) 2 bytes for each filed
4) field: length = 2 bytes, and the UDP header has 8 bytes,
   so maximum number of bytes =2^{16}-1-8=65527 bytes
5) field: source port = 2 bytes,
   so maximum source port number = 2^{16}-1=65535
6) protocol number for UDP=0x11 in hexadecimal= 17 in decimal

# *Problem 7.2*

Q4：

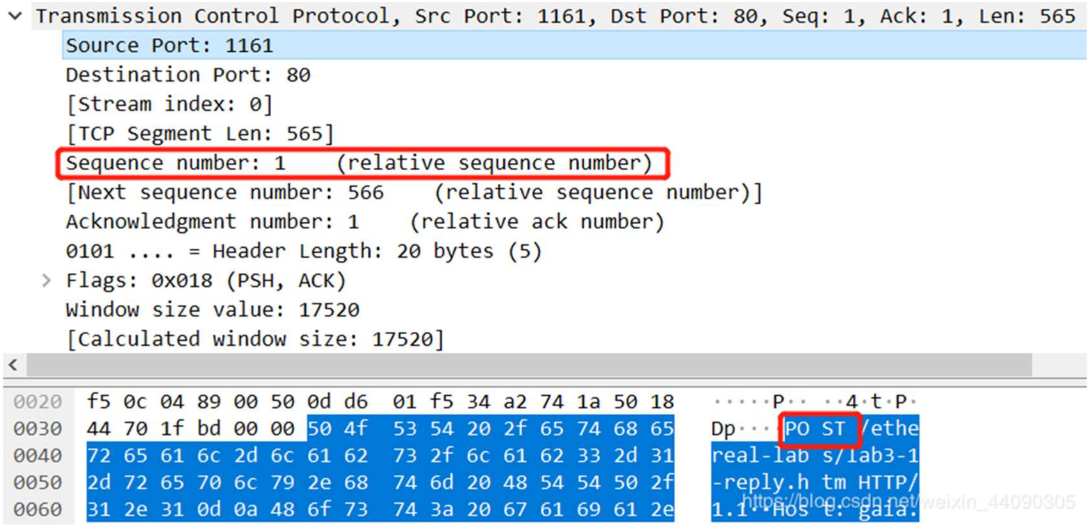


```
   3136 4.830495     10.20.184.16        128.119.245.12      TCP        66 1281 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
   4953 7.155302     128.119.245.12      10.20.184.16        HTTP      538 HTTP/1.1 404 Not Found  (text/html)
> Frame 3136: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
> Ethernet II, Src: AsustekC_a8:8c:ea (04:d4:c4:a8:8c:ea), Dst: JuniperN_ab:30:03 (40:71:83:ab:30:03)
> Internet Protocol Version 4, Src: 10.20.184.16, Dst: 128.119.245.12
v Transmission Control Protocol, Src Port: 1281, Dst Port: 80, Seq: 0, Len: 0
    Source Port: 1281
    Destination Port: 80
    [Stream index: 12]
    [TCP Segment Len: 0]
    Sequence number: 0    (relative sequence number)
    [Next sequence number: 0    (relative sequence number)]
    Acknowledgment number: 0
    1000 .... = Header Length: 32 bytes (8)
  v Flags: 0x002 (SYN)
      000. .... .... = Reserved: Not set
      ...0 .... .... = Nonce: Not set
      .... 0... .... = Congestion Window Reduced (CWR): 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
      .... .... ...0 = Fin: Not set
      [TCP Flags: ··········S·]
    Window size value: 64240
    [Calculated window size: 64240]
    Checksum: 0x37cf [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
```

Sequence number of the TCP SYN segment: 0.

SYN flag = 1 identifies the segment as a SYN segment

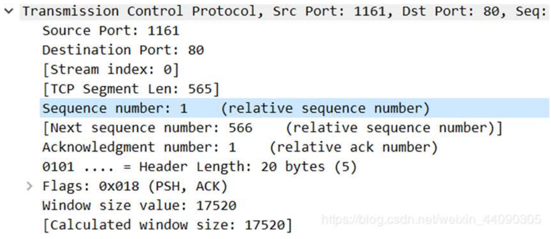
Q6：



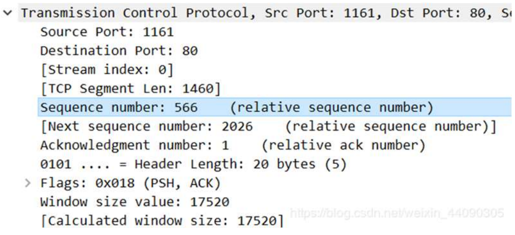
```
v Transmission Control Protocol, Src Port: 1161, Dst Port: 80, Seq: 1, Ack: 1, Len: 565
    Source Port: 1161
    Destination Port: 80
    [Stream index: 0]
    [TCP Segment Len: 565]
    Sequence number: 1    (relative sequence number)
    [Next sequence number: 566    (relative sequence number)]
    Acknowledgment number: 1    (relative ack number)
    0101 .... = Header Length: 20 bytes (5)
  > Flags: 0x018 (PSH, ACK)
    Window size value: 17520
    [Calculated window size: 17520]

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.
```

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

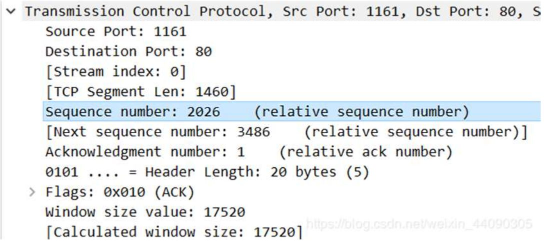
Q7：



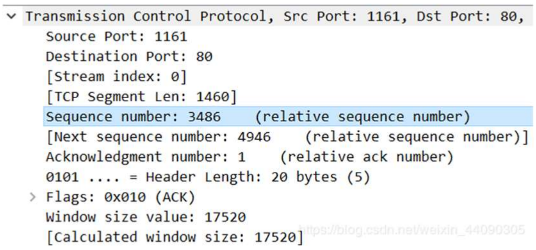
```
v Transmission Control Protocol, Src Port: 1161, Dst Port: 80, Seq:
    Source Port: 1161
    Destination Port: 80
    [Stream index: 0]
    [TCP Segment Len: 565]
    Sequence number: 1    (relative sequence number)
    [Next sequence number: 566    (relative sequence number)]
    Acknowledgment number: 1    (relative ack number)
    0101 .... = Header Length: 20 bytes (5)
  > Flags: 0x018 (PSH, ACK)
    Window size value: 17520
    [Calculated window size: 17520]
```



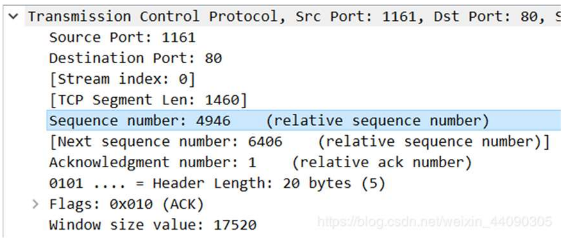
```
v Transmission Control Protocol, Src Port: 1161, Dst Port: 80, S
    Source Port: 1161
    Destination Port: 80
    [Stream index: 0]
    [TCP Segment Len: 1460]
    Sequence number: 566    (relative sequence number)
    [Next sequence number: 2026    (relative sequence number)]
    Acknowledgment number: 1    (relative ack number)
    0101 .... = Header Length: 20 bytes (5)
  > Flags: 0x018 (PSH, ACK)
    Window size value: 17520
    [Calculated window size: 17520]
```



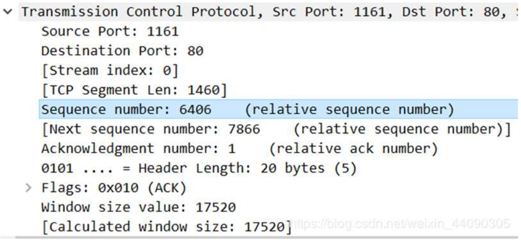
```
∨ Transmission Control Protocol, Src Port: 1161, Dst Port: 80, S
    Source Port: 1161
    Destination Port: 80
    [Stream index: 0]
    [TCP Segment Len: 1460]
    Sequence number: 2026    (relative sequence number)
    [Next sequence number: 3486    (relative sequence number)]
    Acknowledgment number: 1    (relative ack number)
    0101 .... = Header Length: 20 bytes (5)
  > Flags: 0x010 (ACK)
    Window size value: 17520
    [Calculated window size: 17520]
```



```
∨ Transmission Control Protocol, Src Port: 1161, Dst Port: 80,
    Source Port: 1161
    Destination Port: 80
    [Stream index: 0]
    [TCP Segment Len: 1460]
    Sequence number: 3486    (relative sequence number)
    [Next sequence number: 4946    (relative sequence number)]
    Acknowledgment number: 1    (relative ack number)
    0101 .... = Header Length: 20 bytes (5)
  > Flags: 0x010 (ACK)
    Window size value: 17520
    [Calculated window size: 17520]
```



```
∨ Transmission Control Protocol, Src Port: 1161, Dst Port: 80, S
    Source Port: 1161
    Destination Port: 80
    [Stream index: 0]
    [TCP Segment Len: 1460]
    Sequence number: 4946    (relative sequence number)
    [Next sequence number: 6406    (relative sequence number)]
    Acknowledgment number: 1    (relative ack number)
    0101 .... = Header Length: 20 bytes (5)
  > Flags: 0x010 (ACK)
    Window size value: 17520
```

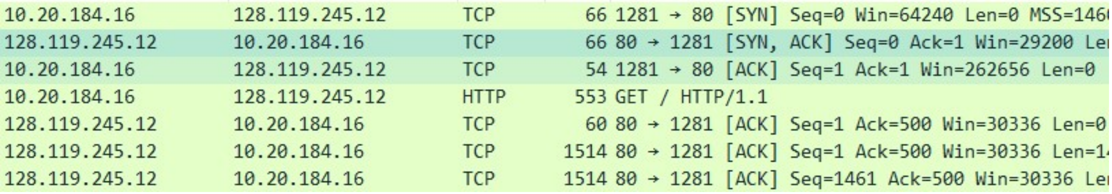


```
∨ Transmission Control Protocol, Src Port: 1161, Dst Port: 80,
    Source Port: 1161
    Destination Port: 80
    [Stream index: 0]
    [TCP Segment Len: 1460]
    Sequence number: 6406    (relative sequence number)
    [Next sequence number: 7866    (relative sequence number)]
    Acknowledgment number: 1    (relative ack number)
    0101 .... = Header Length: 20 bytes (5)
  > Flags: 0x010 (ACK)
    Window size value: 17520
    [Calculated window size: 17520]
```

EstimatedRTT= 7/8 LastEstimatedRTT+ 1/8 SampleRTT

Num	Sequence Number	Sent time(s)	Ack time(s)	RTT(s)
1	1	0.026477	0.053937	0.02746
2	566	0.041737	0.077294	0.02847
3	2026	0.054026	0.124085	0.03367
4	3486	0.054690	0.169118	0.04377
5	4946	0.077405	0.217299	0.05578
6	6406	0.078157	0.267802	0.07251

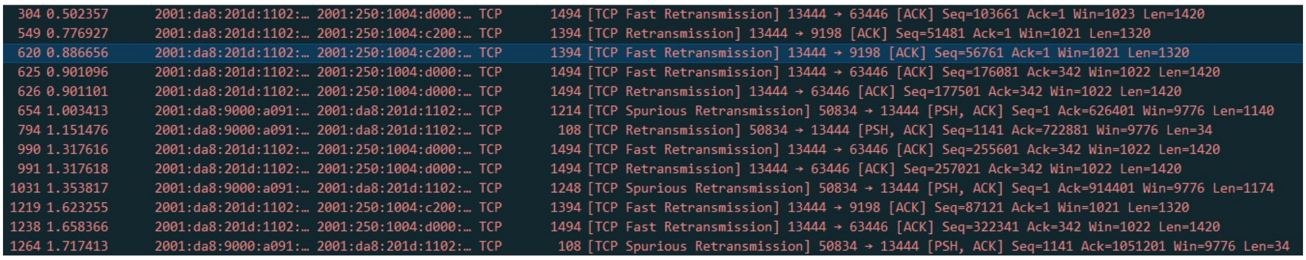
Q9：



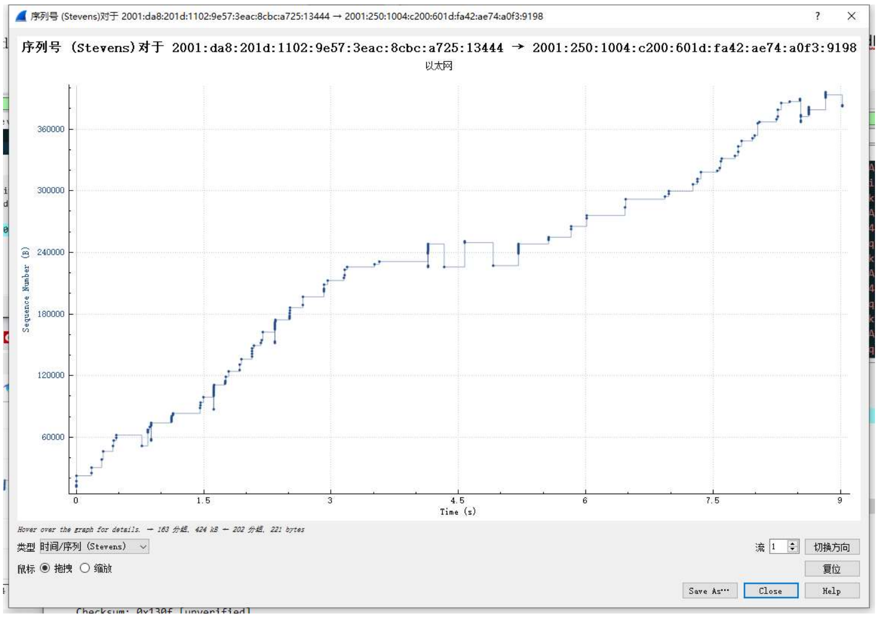
```
10.20.184.16     128.119.245.12   TCP     66 1281 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=146
128.119.245.12   10.20.184.16     TCP     66 80 → 1281 [SYN, ACK] Seq=0 Ack=1 Win=29200 Le
10.20.184.16     128.119.245.12   TCP     54 1281 → 80 [ACK] Seq=1 Ack=1 Win=262656 Len=0
10.20.184.16     128.119.245.12   HTTP   553 GET / HTTP/1.1
128.119.245.12   10.20.184.16     TCP     60 80 → 1281 [ACK] Seq=1 Ack=500 Win=30336 Len=0
128.119.245.12   10.20.184.16     TCP   1514 80 → 1281 [ACK] Seq=1 Ack=500 Win=30336 Len=1
128.119.245.12   10.20.184.16     TCP   1514 80 → 1281 [ACK] Seq=1461 Ack=500 Win=30336 Le
```

The minimum amount of available buffer space advertised is 29200. No throttle is made due to the lack of buffer space.

Q10：



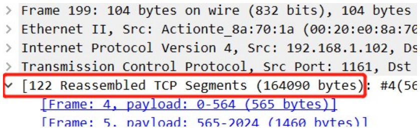
```
 304 0.502357  2001:da8:201d:1102:… 2001:250:1004:d000:… TCP  1494 [TCP Fast Retransmission] 13444 → 63446 [ACK] Seq=103661 Ack=1 Win=1023 Len=1420
 549 0.776927  2001:da8:201d:1102:… 2001:250:1004:c200:… TCP  1394 [TCP Retransmission] 13444 → 9198 [ACK] Seq=51481 Ack=1 Win=1021 Len=1320
 620 0.886656  2001:da8:201d:1102:… 2001:250:1004:c200:… TCP  1394 [TCP Fast Retransmission] 13444 → 9198 [ACK] Seq=56761 Ack=1 Win=1021 Len=1320
 625 0.901096  2001:da8:201d:1102:… 2001:250:1004:d000:… TCP  1494 [TCP Fast Retransmission] 13444 → 63446 [ACK] Seq=176081 Ack=342 Win=1022 Len=1420
 626 0.901101  2001:da8:201d:1102:… 2001:250:1004:d000:… TCP  1494 [TCP Retransmission] 13444 → 63446 [ACK] Seq=177501 Ack=342 Win=1022 Len=1420
 654 1.003413  2001:da8:9000:a091:… 2001:da8:201d:1102:… TCP  1214 [TCP Spurious Retransmission] 50834 → 13444 [PSH, ACK] Seq=1 Ack=626401 Win=9776 Len=1140
 794 1.151476  2001:da8:9000:a091:… 2001:da8:201d:1102:… TCP   108 [TCP Retransmission] 50834 → 13444 [PSH, ACK] Seq=1141 Ack=722881 Win=9776 Len=34
 990 1.317616  2001:da8:201d:1102:… 2001:250:1004:d000:… TCP  1494 [TCP Fast Retransmission] 13444 → 63446 [ACK] Seq=255601 Ack=342 Win=1022 Len=1420
 991 1.317618  2001:da8:201d:1102:… 2001:250:1004:d000:… TCP  1494 [TCP Retransmission] 13444 → 63446 [ACK] Seq=257021 Ack=342 Win=1022 Len=1420
1031 1.353817  2001:da8:9000:a091:… 2001:da8:201d:1102:… TCP  1248 [TCP Spurious Retransmission] 50834 → 13444 [PSH, ACK] Seq=1 Ack=914401 Win=9776 Len=1174
1219 1.623255  2001:da8:201d:1102:… 2001:250:1004:c200:… TCP  1394 [TCP Fast Retransmission] 13444 → 9198 [ACK] Seq=87121 Ack=1 Win=1021 Len=1320
1238 1.658366  2001:da8:201d:1102:… 2001:250:1004:d000:… TCP  1494 [TCP Fast Retransmission] 13444 → 63446 [ACK] Seq=322341 Ack=342 Win=1022 Len=1420
1264 1.717413  2001:da8:9000:a091:… 2001:da8:201d:1102:… TCP   108 [TCP Spurious Retransmission] 50834 → 13444 [PSH, ACK] Seq=1141 Ack=1051201 Win=9776 Len=34
```



Hover over the graph for details. — 163 分組, 424 kB — 202 分組, 221 bytes

There are many retransmitted segment in the trace file. I use filter” tcp.analysis.retransmission” and specific analysis tool to make a table . If the filter shows nothing and all sequence numbers are in ascending order, it indicates that there is no retransmitted segment, which is opposite to the truth.

Q12:



```
> Frame 199: 104 bytes on wire (832 bits), 104 bytes
> Ethernet II, Src: Actionte_8a:70:1a (00:20:e0:8a:70
> Internet Protocol Version 4, Src: 192.168.1.102, Ds
> Transmission Control Protocol, Src Port: 1161, Dst
v [122 Reassembled TCP Segments (164090 bytes): #4(56
    [Frame: 4, payload: 0-564 (565 bytes)]
    [Frame: 5, payload: 565-2024 (1460 bytes)]
```

File_size = 164090bytes

Download_time = 8.987656s-5.197529s=3.790127s

throughput = file_size/download_time=43294.06376bytes/s