Wireshark Lab #1, Intro

He Tianyang, 3022001441

October 20, 2024

1 Preperation

Following the instructions, we used `wget` to retrieve an ASCII copy of *Alice in Wonderland* and stored it in `~/Documents/alice.txt`.

```
wget -O ~/Documents/alice.txt http://gaia.cs.umass.edu/wireshark-
labs/alice.txt
```

Next, we prevented the Large Send Offload (LSO) feature from interfering with the lab by running the following command:

```
sudo ethtool -K wlp0s20f3 tso off gso off lro off
```

To check the status of the LSO feature, we used the following command:

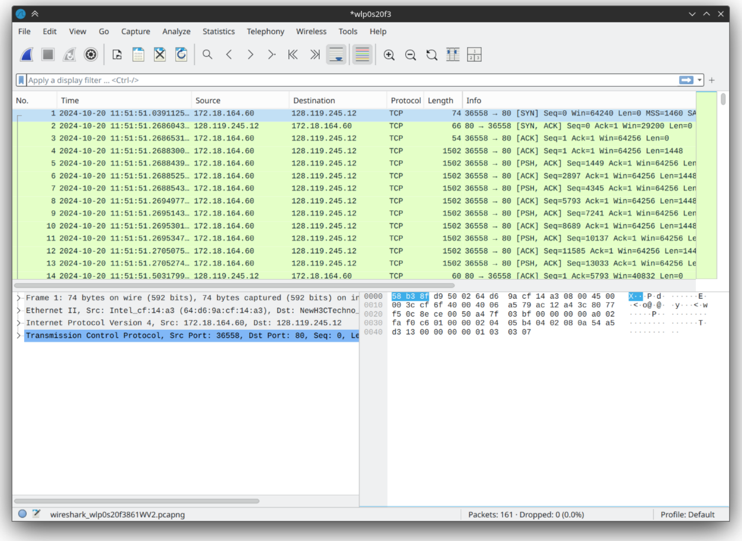
```
sudo ethtool -k wlp0s20f3
```

At last, we uploaded the file to the server as per the instructions, and the result is shown in Fig. 1.

2 A first look at the captured trace

Question 1.

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?



```
No. Time                           Source          Destination     Protocol Length Info
  1 2024-10-20 11:51:51.0391125… 172.18.164.60   128.119.245.12  TCP        74 36558 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SA
  2 2024-10-20 11:51:51.2686043… 128.119.245.12  172.18.164.60   TCP        66 80 → 36558 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0
  3 2024-10-20 11:51:51.2686531… 172.18.164.60   128.119.245.12  TCP        54 36558 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0
  4 2024-10-20 11:51:51.2688300… 172.18.164.60   128.119.245.12  TCP      1502 36558 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=1448
  5 2024-10-20 11:51:51.2688439… 172.18.164.60   128.119.245.12  TCP      1502 36558 → 80 [PSH, ACK] Seq=1449 Ack=1 Win=64256 Len
  6 2024-10-20 11:51:51.2688525… 172.18.164.60   128.119.245.12  TCP      1502 36558 → 80 [ACK] Seq=2897 Ack=1 Win=64256 Len=1448
  7 2024-10-20 11:51:51.2688543… 172.18.164.60   128.119.245.12  TCP      1502 36558 → 80 [PSH, ACK] Seq=4345 Ack=1 Win=64256 Len
  8 2024-10-20 11:51:51.2694977… 172.18.164.60   128.119.245.12  TCP      1502 36558 → 80 [ACK] Seq=5793 Ack=1 Win=64256 Len=1448
  9 2024-10-20 11:51:51.2695143… 172.18.164.60   128.119.245.12  TCP      1502 36558 → 80 [PSH, ACK] Seq=7241 Ack=1 Win=64256 Len
 10 2024-10-20 11:51:51.2695301… 172.18.164.60   128.119.245.12  TCP      1502 36558 → 80 [ACK] Seq=8689 Ack=1 Win=64256 Len=1448
 11 2024-10-20 11:51:51.2695347… 172.18.164.60   128.119.245.12  TCP      1502 36558 → 80 [PSH, ACK] Seq=10137 Ack=1 Win=64256 Le
 12 2024-10-20 11:51:51.2705075… 172.18.164.60   128.119.245.12  TCP      1502 36558 → 80 [ACK] Seq=11585 Ack=1 Win=64256 Len=144
 13 2024-10-20 11:51:51.2705274… 172.18.164.60   128.119.245.12  TCP      1502 36558 → 80 [PSH, ACK] Seq=13033 Ack=1 Win=64256 Le
 14 2024-10-20 11:51:51.5031799… 128.119.245.12  172.18.164.60   TCP        60 80 → 36558 [ACK] Seq=1 Ack=5793 Win=40832 Len=0
```

```
> Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on ir
> Ethernet II, Src: Intel_cf:14:a3 (64:d6:9a:cf:14:a3), Dst: NewH3CTechno_
> Internet Protocol Version 4, Src: 172.18.164.60, Dst: 128.119.245.12
> Transmission Control Protocol, Src Port: 36558, Dst Port: 80, Seq: 0, Le
```

```
0000  58 b3 8f d9 50 02 64 d6  9a cf 14 a3 08 00 45 00   X···P·d· ······E·
0010  00 3c cf 6f 40 00 40 06  a5 79 ac 12 a4 3c 80 77   ·<·o@·@· ·y···<·w
0020  f5 0c 8e ce 00 50 a4 7f  03 bf 00 00 00 00 a0 02   ·····P·· ········
0030  fa f0 c6 01 00 00 02 04  05 b4 04 02 08 0a 54 a5   ········ ······T·
0040  d3 13 00 00 00 00 01 03  03 07                     ········ ··
```

*wlp0s20f3
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>
wireshark_wlp0s20f3861WV2.pcapng
Packets: 161 · Dropped: 0 (0.0%)
Profile: Default

Figure 1: Uploading to the server

Inspecting the first three packets, we can observe the handshake process. By examining the `SYN` packet from the client, we can see that **the source IP address is `172.18.164.60` and the source port number is `36558`**. The details are shown in Fig. 2.

Question 2.

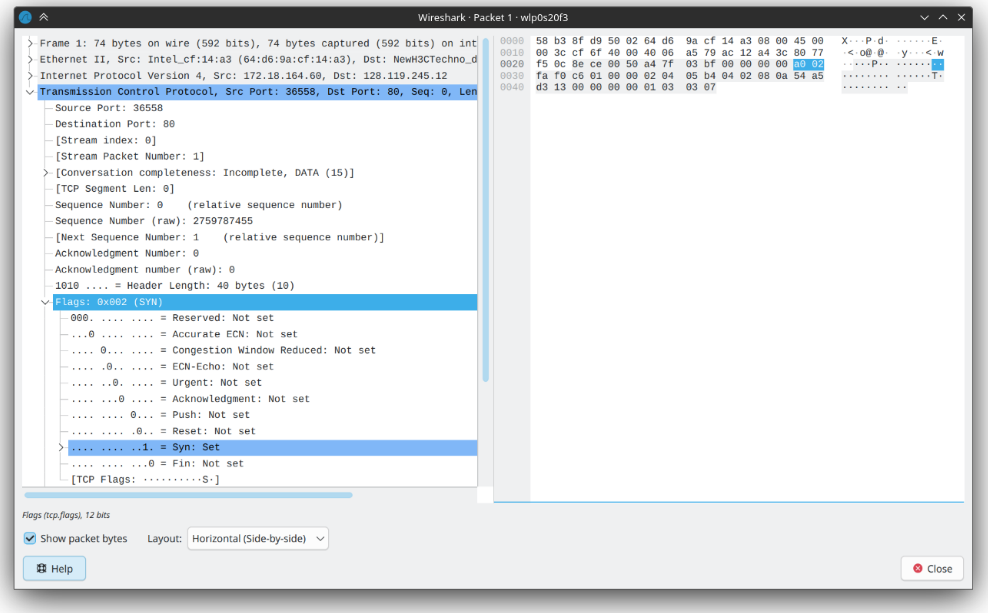
What is the IP address of gaia.cs.umass.edu? On what port number is it sending and receiving TCP segments for this connection?

Also from Fig. 2, we can see the **destination IP address is `128.119.245.12` and the destination port number is `80`.**

Question 3.

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

Question 3 is similar to Question 1. Inspect the `SYN` packet from the client, we can see **the source IP address is `172.18.164.60` and the source port number is `36558`**.



Wireshark · Packet 1 · wlp0s20f3

```
> Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on int
> Ethernet II, Src: Intel_cf:14:a3 (64:d6:9a:cf:14:a3), Dst: NewH3CTechno_d
> Internet Protocol Version 4, Src: 172.18.164.60, Dst: 128.119.245.12
v Transmission Control Protocol, Src Port: 36558, Dst Port: 80, Seq: 0, Len
    Source Port: 36558
    Destination Port: 80
    [Stream index: 0]
    [Stream Packet Number: 1]
  > [Conversation completeness: Incomplete, DATA (15)]
    [TCP Segment Len: 0]
    Sequence Number: 0    (relative sequence number)
    Sequence Number (raw): 2759787455
    [Next Sequence Number: 1    (relative sequence number)]
    Acknowledgment Number: 0
    Acknowledgment number (raw): 0
    1010 .... = Header Length: 40 bytes (10)
  v Flags: 0x002 (SYN)
      000. .... .... = Reserved: Not set
      ...0 .... .... = Accurate ECN: Not set
      .... 0... .... = Congestion Window Reduced: 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·]
```

```
0000  58 b3 8f d9 50 02 64 d6  9a cf 14 a3 08 00 45 00   X···P·d· ······E·
0010  00 3c cf 6f 40 00 40 06  a5 79 ac 12 a4 3c 80 77   ·<·o@·@· ·y···<·w
0020  f5 0c 8e ce 00 50 a4 7f  03 bf 00 00 00 00 a0 02   ·····P·· ········
0030  fa f0 c6 01 00 00 02 04  05 b4 04 02 08 0a 54 a5   ········ ······T·
0040  d3 13 00 00 00 00 01 03  03 07                     ········ ··
```

Flags (tcp.flags), 12 bits

Show packet bytes ✓ Layout: Horizontal (Side-by-side)

Figure 2: SYN packet

3 TCP Basics

Question 4.

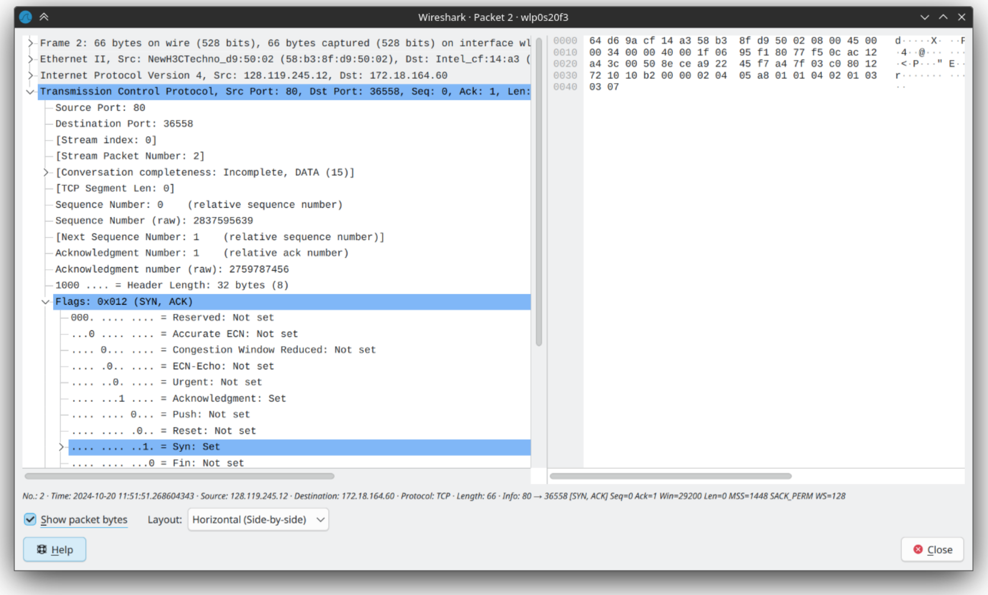
What is the sequence number of the TCP SYN segment that is used to initiate the TCP connection between the client computer and gaia.cs.umass.edu? What is it in the segment that identifies the segment as a SYN segment?

Also from Fig. 2, we can see the **sequence number is `2759787455` and the flag is set to `0x002` which only `SYN` Flag is set.**

Question 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?

The `SYN,ACK` packet is shown in Fig. 3. The **sequence number is `2837595639` and the Acknowledgement field is `2759787456`.** The server determined the value by adding 1 to the sequence number of the SYN packet. The flag is set to `0x012` which `SYN` and `ACK` flags are set.



```
> Frame 2: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface wl
> Ethernet II, Src: NewH3CTechno_d9:50:02 (58:b3:8f:d9:50:02), Dst: Intel_cf:14:a3 (
> Internet Protocol Version 4, Src: 128.119.245.12, Dst: 172.18.164.60
v Transmission Control Protocol, Src Port: 80, Dst Port: 36558, Seq: 0, Ack: 1, Len:
    Source Port: 80
    Destination Port: 36558
    [Stream index: 0]
    [Stream Packet Number: 2]
  > [Conversation completeness: Incomplete, DATA (15)]
    [TCP Segment Len: 0]
    Sequence Number: 0    (relative sequence number)
    Sequence Number (raw): 2837595639
    [Next Sequence Number: 1    (relative sequence number)]
    Acknowledgment Number: 1    (relative ack number)
    Acknowledgment number (raw): 2759787456
    1000 .... = Header Length: 32 bytes (8)
  v Flags: 0x012 (SYN, ACK)
      000. .... .... = Reserved: Not set
      ...0 .... .... = Accurate ECN: Not set
      .... 0... .... = Congestion Window Reduced: 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
      .... .... ...0 = Fin: Not set
```

```
0000  64 d6 9a cf 14 a3 58 b3  8f d9 50 02 08 00 45 00   d·····X· ··P
0010  00 34 00 00 40 00 1f 06  95 f1 80 77 f5 0c ac 12   ·4··@··· ···
0020  a4 3c 00 50 8e ce a9 22  45 f7 a4 7f 03 c0 80 12   ·<·P···" E··
0030  72 10 10 b2 00 00 02 04  05 a8 01 01 04 02 01 03   r······· ···
0040  03 07                                              ··
```

No.: 2 · Time: 2024-10-20 11:51:51.268604343 · Source: 128.119.245.12 · Destination: 172.18.164.60 · Protocol: TCP · Length: 66 · Info: 80 → 36558 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1448 SACK_PERM WS=128

[✓] Show packet bytes   Layout: Horizontal (Side-by-side)

Figure 3: SYNACK packet

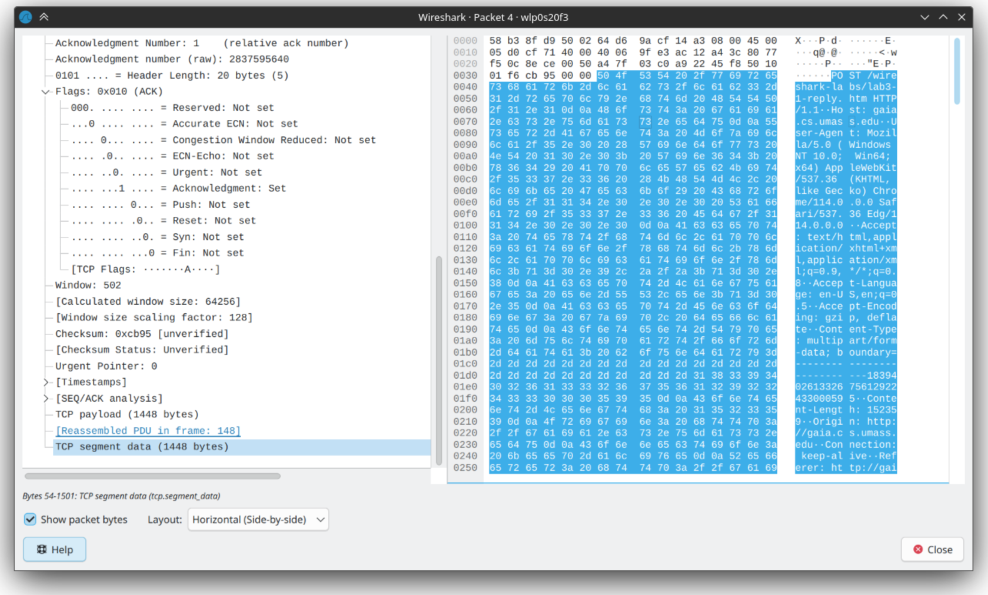
Question 6.

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.

Inspecting the first TCP packet after the handshake, the details are shown in Fig. 4. The **sequence number is** `2759787456`. We can see the `POST` command at the start of the DATA field.

Question 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 `EstimatedRTT` equation on page 242 for all subsequent segments.



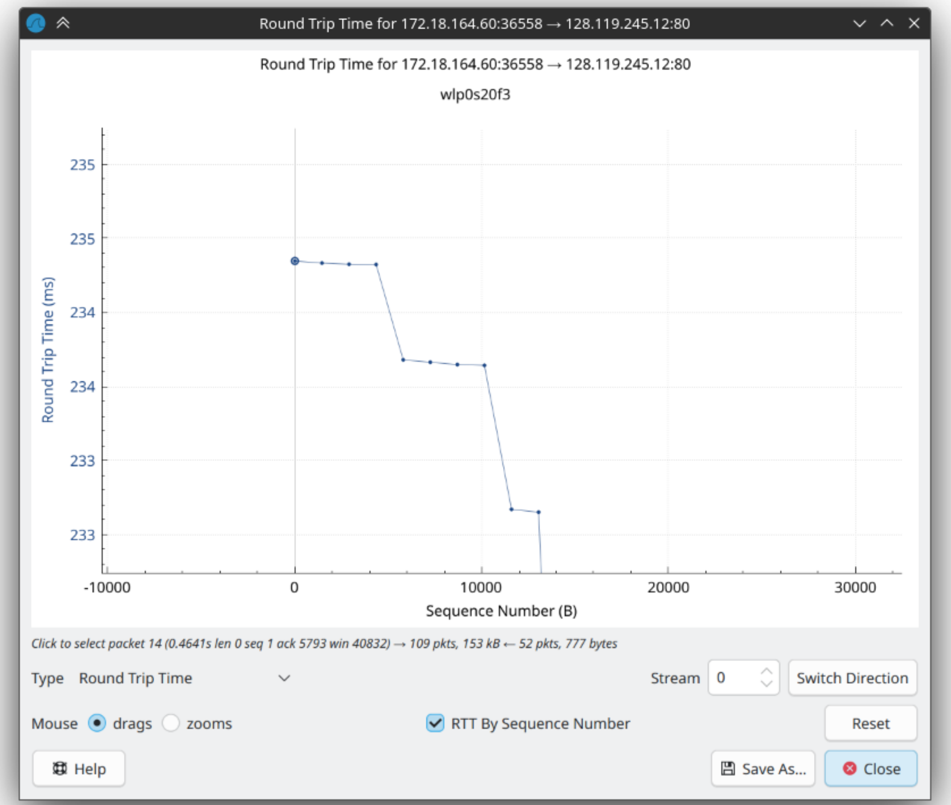
```
Acknowledgment Number: 1    (relative ack number)
Acknowledgment number (raw): 2837595640
0101 .... = Header Length: 20 bytes (5)
Flags: 0x010 (ACK)
    000. .... .... = Reserved: Not set
    ...0 .... .... = Accurate ECN: Not set
    .... 0... .... = Congestion Window Reduced: Not set
    .... .0.. .... = ECN-Echo: Not set
    .... ..0. .... = Urgent: Not set
    .... ...1 .... = Acknowledgment: Set
    .... .... 0... = Push: Not set
    .... .... .0.. = Reset: Not set
    .... .... ..0. = Syn: Not set
    .... .... ...0 = Fin: Not set
    [TCP Flags: ·······A····]
Window: 502
[Calculated window size: 64256]
[Window size scaling factor: 128]
Checksum: 0xcb95 [unverified]
[Checksum Status: Unverified]
Urgent Pointer: 0
[Timestamps]
[SEQ/ACK analysis]
TCP payload (1448 bytes)
[Reassembled PDU in frame: 148]
TCP segment data (1448 bytes)
```

```
0000  58 b3 8f d9 50 02 64 d6  9a cf 14 a3 08 00 45 00   X···P·d· ······E·
0010  05 d0 cf 71 40 00 40 06  9f e3 ac 12 a4 3c 80 77   ···q@·@· ·····<·w
0020  f5 0c 8e ce 00 50 a4 7f  03 c0 a9 22 45 f8 50 10   ·····P·· ···"E·P·
0030  01 f6 cb 95 00 00 50 4f  53 54 20 2f 77 69 72 65   ······PO ST /wire
0040  73 68 61 72 6b 2d 6c 61  62 73 2f 6c 61 62 33 2d   shark-la bs/lab3-
0050  31 2d 72 65 70 6c 79 2e  68 74 6d 20 48 54 54 50   1-reply. htm HTTP
0060  2f 31 2e 31 0d 0a 48 6f  73 74 3a 20 67 61 69 61   /1.1··Ho st: gaia
0070  2e 63 73 2e 75 6d 61 73  73 2e 65 64 75 0d 0a 55   .cs.umas s.edu··U
0080  73 65 72 2d 41 67 65 6e  74 3a 20 4d 6f 7a 69 6c   ser-Agen t: Mozil
0090  6c 61 2f 35 2e 30 20 28  57 69 6e 64 6f 77 73 20   la/5.0 ( Windows 
00a0  4e 54 20 31 30 2e 30 3b  20 57 69 6e 36 34 3b 20   NT 10.0;  Win64; 
00b0  78 36 34 29 20 41 70 70  6c 65 57 65 62 4b 69 74   x64) App leWebKit
00c0  2f 35 33 37 2e 33 36 20  28 4b 48 54 4d 4c 2c 20   /537.36  (KHTML, 
00d0  6c 69 6b 65 20 47 65 63  6b 6f 29 20 43 68 72 6f   like Gec ko) Chro
00e0  6d 65 2f 31 31 34 2e 30  2e 30 2e 30 20 53 61 66   me/114.0 .0.0 Saf
00f0  61 72 69 2f 35 33 37 2e  33 36 20 45 64 67 2f 31   ari/537. 36 Edg/1
0100  31 34 2e 30 2e 30 2e 30  0d 0a 41 63 63 65 70 74   14.0.0.0 ··Accept
0110  3a 20 74 65 78 74 2f 68  74 6d 6c 2c 61 70 70 6c   : text/h tml,appl
0120  69 63 61 74 69 6f 6e 2f  78 68 74 6d 6c 2b 78 6d   ication/ xhtml+xm
0130  6c 2c 61 70 70 6c 69 63  61 74 69 6f 6e 2f 78 6d   l,applic ation/xm
0140  6c 3b 71 3d 30 2e 39 2c  2a 2f 2a 3b 71 3d 30 2e   l;q=0.9, */*;q=0.
0150  38 0d 0a 41 63 63 65 70  74 2d 4c 61 6e 67 75 61   8··Accep t-Langua
0160  67 65 3a 20 65 6e 2d 55  53 2c 65 6e 3b 71 3d 30   ge: en-U S,en;q=0
0170  2e 35 0d 0a 41 63 63 65  70 74 2d 45 6e 63 6f 64   .5··Acce pt-Encod
0180  69 6e 67 3a 20 67 7a 69  70 2c 20 64 65 66 6c 61   ing: gzi p, defla
0190  74 65 0d 0a 43 6f 6e 74  65 6e 74 2d 54 79 70 65   te··Cont ent-Type
01a0  3a 20 6d 75 6c 74 69 70  61 72 74 2f 66 6f 72 6d   : multip art/form
01b0  2d 64 61 74 61 3b 20 62  6f 75 6e 64 61 72 79 3d   -data; b oundary=
01c0  2d 2d 2d 2d 2d 2d 2d 2d  2d 2d 2d 2d 2d 2d 2d 2d   -------- --------
01d0  2d 2d 2d 2d 2d 2d 2d 2d  2d 2d 2d 31 38 33 39 34   -------- ---18394
01e0  30 32 36 31 33 33 32 36  37 35 36 31 32 39 32 32   02613326 75612922
01f0  34 33 33 30 30 30 35 39  35 0d 0a 43 6f 6e 74 65   43300059 5··Conte
0200  6e 74 2d 4c 65 6e 67 74  68 3a 20 31 35 32 33 35   nt-Lengt h: 15235
0210  39 0d 0a 4f 72 69 67 69  6e 3a 20 68 74 74 70 3a   9··Origi n: http:
0220  2f 2f 67 61 69 61 2e 63  73 2e 75 6d 61 73 73 2e   //gaia.c s.umass.
0230  65 64 75 0d 0a 43 6f 6e  6e 65 63 74 69 6f 6e 3a   edu··Con nection:
0240  20 6b 65 65 70 2d 61 6c  69 76 65 0d 0a 52 65 66    keep-al ive··Ref
0250  65 72 65 72 3a 20 68 74  74 70 3a 2f 2f 67 61 69   erer: ht tp://gai
```

Bytes 54-1501: TCP segment data (tcp.segment_data)

[✓] Show packet bytes Layout: Horizontal (Side-by-side)

Figure 4: POST packet

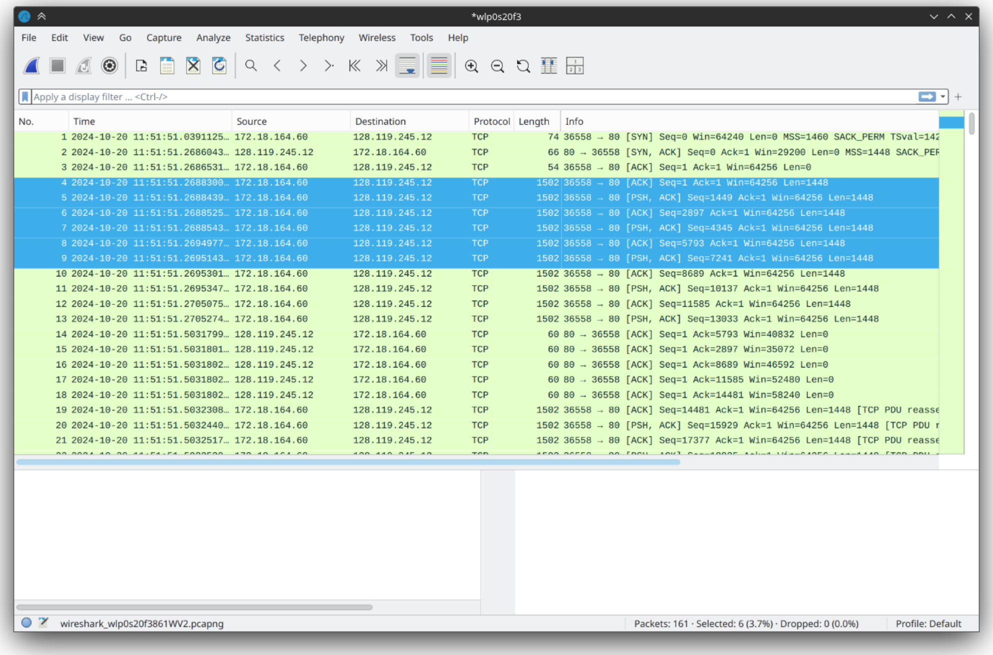


Click to select packet 14 (0.4641s len 0 seq 1 ack 5793 win 40832) → 109 pkts, 153 kB ← 52 pkts, 777 bytes

Type: Round Trip Time Stream: 0

Mouse [✓ drags] zooms [✓] RTT By Sequence Number

Figure 5: RTT statistic


*wlp0s20f3
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
wireshark_wlp0s20f3861WV2.pcapng
Packets: 161 · Selected: 6 (3.7%) · Dropped: 0 (0.0%)
Profile: Default

```
No. Time                           Source          Destination     Protocol Length Info
  1 2024-10-20 11:51:51.0391125… 172.18.164.60   128.119.245.12  TCP        74 36558 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=142
  2 2024-10-20 11:51:51.2686043… 128.119.245.12  172.18.164.60   TCP        66 80 → 36558 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1448 SACK_PER
  3 2024-10-20 11:51:51.2686531… 172.18.164.60   128.119.245.12  TCP        54 36558 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0
  4 2024-10-20 11:51:51.2688300… 172.18.164.60   128.119.245.12  TCP      1502 36558 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=1448
  5 2024-10-20 11:51:51.2688439… 172.18.164.60   128.119.245.12  TCP      1502 36558 → 80 [PSH, ACK] Seq=1449 Ack=1 Win=64256 Len=1448
  6 2024-10-20 11:51:51.2688525… 172.18.164.60   128.119.245.12  TCP      1502 36558 → 80 [ACK] Seq=2897 Ack=1 Win=64256 Len=1448
  7 2024-10-20 11:51:51.2688543… 172.18.164.60   128.119.245.12  TCP      1502 36558 → 80 [PSH, ACK] Seq=4345 Ack=1 Win=64256 Len=1448
  8 2024-10-20 11:51:51.2694977… 172.18.164.60   128.119.245.12  TCP      1502 36558 → 80 [ACK] Seq=5793 Ack=1 Win=64256 Len=1448
  9 2024-10-20 11:51:51.2695143… 172.18.164.60   128.119.245.12  TCP      1502 36558 → 80 [PSH, ACK] Seq=7241 Ack=1 Win=64256 Len=1448
 10 2024-10-20 11:51:51.2695301… 172.18.164.60   128.119.245.12  TCP      1502 36558 → 80 [ACK] Seq=8689 Ack=1 Win=64256 Len=1448
 11 2024-10-20 11:51:51.2695347… 172.18.164.60   128.119.245.12  TCP      1502 36558 → 80 [PSH, ACK] Seq=10137 Ack=1 Win=64256 Len=1448
 12 2024-10-20 11:51:51.2705075… 172.18.164.60   128.119.245.12  TCP      1502 36558 → 80 [ACK] Seq=11585 Ack=1 Win=64256 Len=1448
 13 2024-10-20 11:51:51.2705274… 172.18.164.60   128.119.245.12  TCP      1502 36558 → 80 [PSH, ACK] Seq=13033 Ack=1 Win=64256 Len=1448
 14 2024-10-20 11:51:51.5031799… 128.119.245.12  172.18.164.60   TCP        60 80 → 36558 [ACK] Seq=1 Ack=5793 Win=40832 Len=0
 15 2024-10-20 11:51:51.5031801… 128.119.245.12  172.18.164.60   TCP        60 80 → 36558 [ACK] Seq=1 Ack=2897 Win=35072 Len=0
 16 2024-10-20 11:51:51.5031802… 128.119.245.12  172.18.164.60   TCP        60 80 → 36558 [ACK] Seq=1 Ack=8689 Win=46592 Len=0
 17 2024-10-20 11:51:51.5031802… 128.119.245.12  172.18.164.60   TCP        60 80 → 36558 [ACK] Seq=1 Ack=11585 Win=52480 Len=0
 18 2024-10-20 11:51:51.5031802… 128.119.245.12  172.18.164.60   TCP        60 80 → 36558 [ACK] Seq=1 Ack=14481 Win=58240 Len=0
 19 2024-10-20 11:51:51.5032308… 172.18.164.60   128.119.245.12  TCP      1502 36558 → 80 [ACK] Seq=14481 Ack=1 Win=64256 Len=1448 [TCP PDU reasse
 20 2024-10-20 11:51:51.5032440… 172.18.164.60   128.119.245.12  TCP      1502 36558 → 80 [PSH, ACK] Seq=15929 Ack=1 Win=64256 Len=1448 [TCP PDU r
 21 2024-10-20 11:51:51.5032517… 172.18.164.60   128.119.245.12  TCP      1502 36558 → 80 [ACK] Seq=17377 Ack=1 Win=64256 Len=1448 [TCP PDU reasse
```

Figure 6: First 6 packets

Using the statistics feature in Wireshark, the RTT statistics are shown in Fig. 5, and the `EstimatedRTT` is presented in Fig. 2.

Since the absolute sequence number is not important, we use relative sequence numbers instead.

a. The sequence numbers of the first six segments are `1`, `1449`, `2897`, `4345`, `5793`, and `7241`. Each one equals the previous sequence number plus the length of the previous segment, which is `1448`. The details are shown in Fig. 6 with the selected packets.

b&c. The time of each segment and its corresponding ACK is shown in Tab. 1 and Fig. 5.

Table 1: Time of each segment and its ack

Segment	Sent Time	Ack Time	RTT (ms)
1	2024-10-20 11:51:51.268830094	2024-10-20 11:51:51.503179933	234.349839
2	2024-10-20 11:51:51.268843992	2024-10-20 11:51:51.503179933	234.335941
3	2024-10-20 11:51:51.268852535	2024-10-20 11:51:51.503179933	234.327398
4	2024-10-20 11:51:51.268854358	2024-10-20 11:51:51.503179933	234.325575
5	2024-10-20 11:51:51.269497706	2024-10-20 11:51:51.503180205	233.682499
6	2024-10-20 11:51:51.269514380	2024-10-20 11:51:51.503180205	233.665825

We can caluclated the `EstimatedRTT` using the formula

$$\texttt{EstimatedRTT} = (1 - \alpha) \times \texttt{EstimatedRTT} + \alpha \times \texttt{SampleRTT}$$

where $\alpha = 0.125$.

The `EstimatedRTT` is shown in Tab. 2

Table 2: EstimatedRTT

Segment	EstimatedRTT (ms)
1	234.349839
2	234.348102
3	234.345514
4	234.343021
5	234.260456
6	234.186127

Question 8.

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

From Fig. 6, we can see the length of each segment is `1448` bytes.

Question 9.

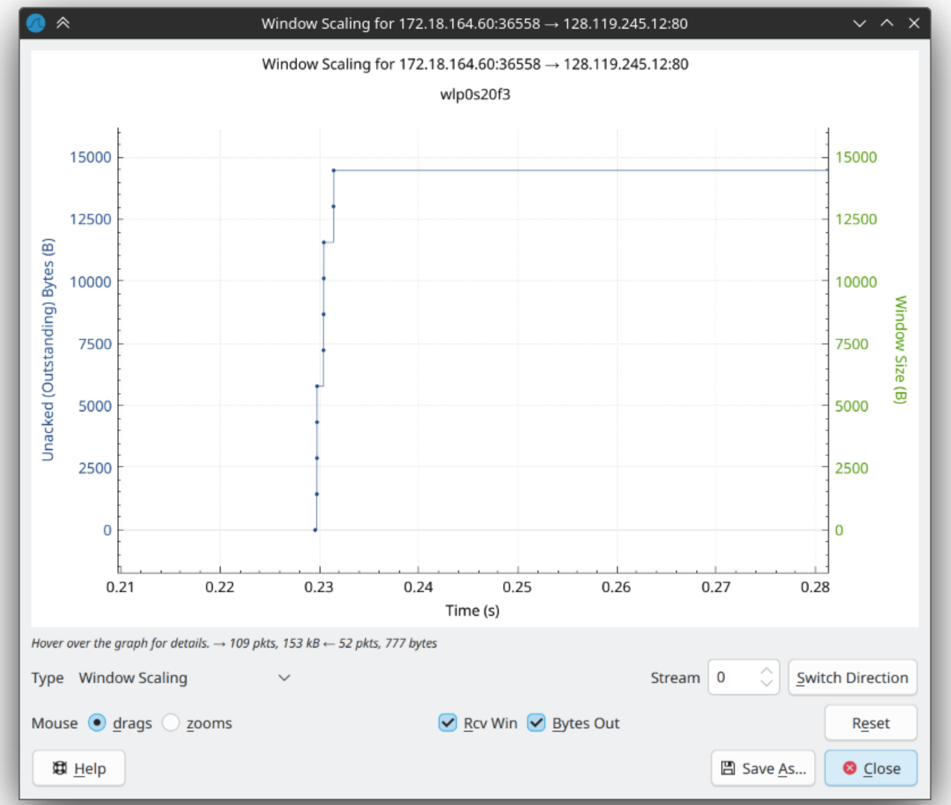
What is the minimum amount of available buffer space advertised at the received for the entire trace? Does the lack of receiver buffer space ever throttle the sender?

We can use the statistics feature in Wireshark to obtain the window size. The window size is shown in Fig. 7. As we can see, the window size keeps growing, indicating that the lack of receiver buffer space never throttled the sender. The window size details are also presented in Tab. 3.

Question 10.

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

We can use the filter `tcp.analysis.retransmission` to find the retransmitted segments. The result is shown in Fig. 8. As shown, there are no retransmitted segments in the trace.


Window Scaling for 172.18.164.60:36558 → 128.119.245.12:80

wlp0s20f3

Hover over the graph for details. → 109 pkts, 153 kB ← 52 pkts, 777 bytes

Type: Window Scaling

Stream: 0

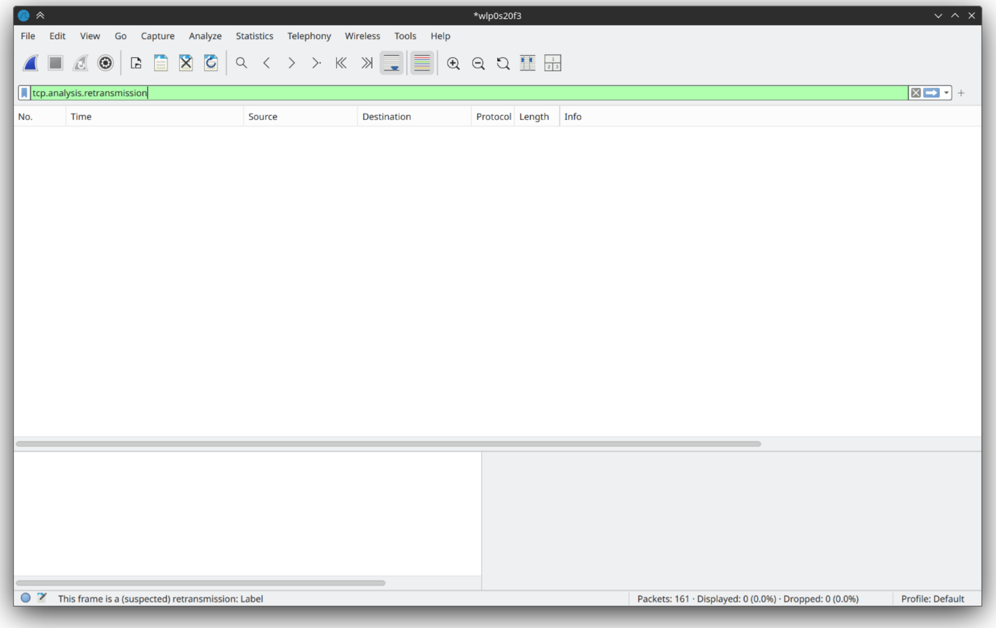
Mouse: [✓ drags] zooms

Rcv Win ✓ Bytes Out ✓

Figure 7: Window Size

Table 3: Window Size

Segment	Window Size (Bytes)
1	1448
2	2896
3	4344
4	5792
5	7240
6	8688



tcp.analysis.retransmission

This frame is a (suspected) retransmission: Label

Packets: 161 · Displayed: 0 (0.0%) · Dropped: 0 (0.0%)

Figure 8: Retransmission

Question 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.

We can use the filter `tcp.analysis.ack_rtt` to find the ACK packets. The result is shown in Fig. 9. We can examine the `Ack` field to confirm the acknowledgment sequence numbers. The ACK numbers are `5793`, `8689`, `11585`, and `14481`, among others. This indicates that the **first ACK acknowledges 5793 bytes, and the subsequent ACKs acknowledge 2896 bytes each**. Fig. 10 shows the time of each ACK packet, suggesting that there may be delayed ACKs.

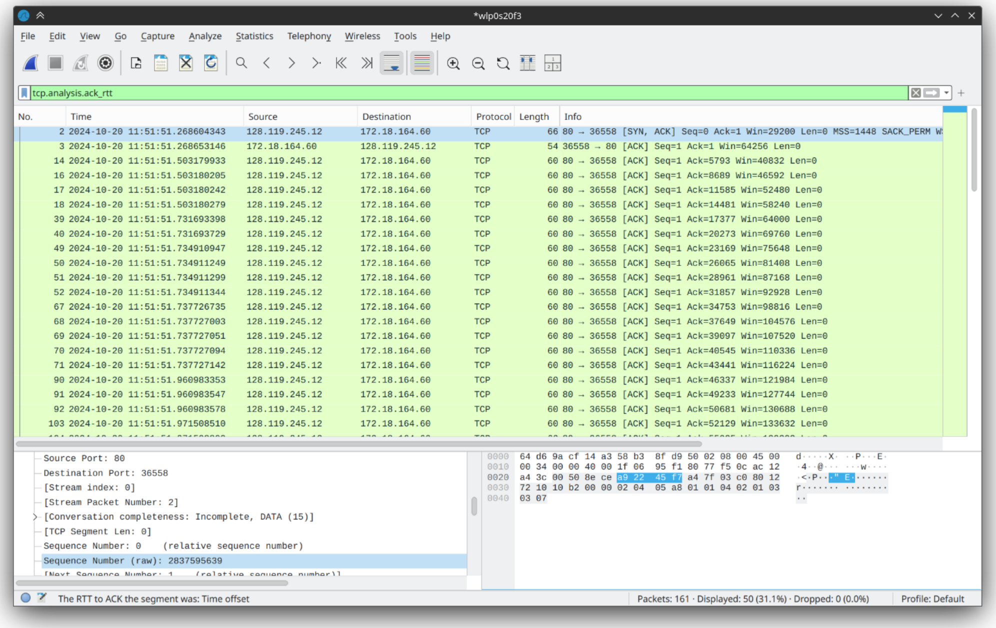
Question 12.

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

First, we need to calculate how much data is transferred. By locating the HTTP request, we can see that it indicates the total transferred bytes as 153,141 bytes, as shown in Fig. 11.

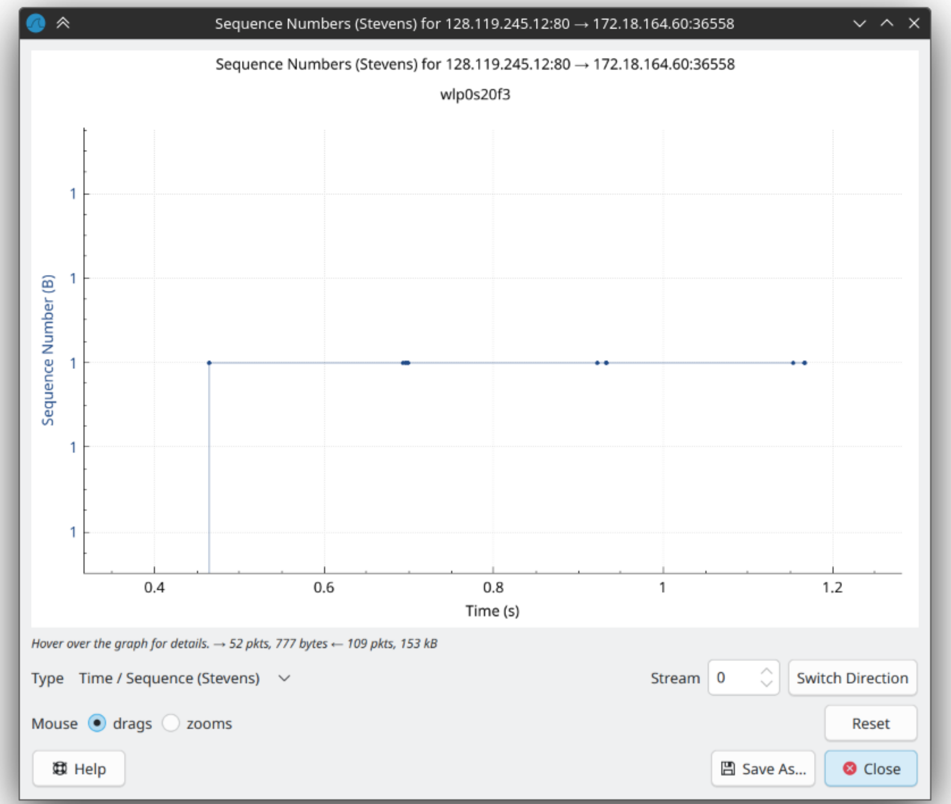
Next, we calculate the transfer time using the timestamps of the first and last packets. The transfer occurred between 2024-10-20 11:51:51.039112562 and 2024-10-20 11:51:51.973625259, as shown in Fig. 1 and Fig. 11. The transfer time can be calculated as:

$$\text{Transfer Time} = 973.625259 - 39.112562 = 934.512697\text{ms}$$



```
tcp.analysis.ack_rtt

No.  Time                           Source          Destination     Protocol Length Info
  2 2024-10-20 11:51:51.268604343   128.119.245.12  172.18.164.60   TCP      66 80 → 36558 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1448 SACK_PERM W
  3 2024-10-20 11:51:51.268653146   172.18.164.60   128.119.245.12  TCP      54 36558 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0
 14 2024-10-20 11:51:51.503179933   128.119.245.12  172.18.164.60   TCP      60 80 → 36558 [ACK] Seq=1 Ack=5793 Win=40832 Len=0
 16 2024-10-20 11:51:51.503180205   128.119.245.12  172.18.164.60   TCP      60 80 → 36558 [ACK] Seq=1 Ack=8689 Win=46592 Len=0
 17 2024-10-20 11:51:51.503180242   128.119.245.12  172.18.164.60   TCP      60 80 → 36558 [ACK] Seq=1 Ack=11585 Win=52480 Len=0
 18 2024-10-20 11:51:51.503180279   128.119.245.12  172.18.164.60   TCP      60 80 → 36558 [ACK] Seq=1 Ack=14481 Win=58240 Len=0
 39 2024-10-20 11:51:51.731693398   128.119.245.12  172.18.164.60   TCP      60 80 → 36558 [ACK] Seq=1 Ack=17377 Win=64000 Len=0
 40 2024-10-20 11:51:51.731693729   128.119.245.12  172.18.164.60   TCP      60 80 → 36558 [ACK] Seq=1 Ack=20273 Win=69760 Len=0
 49 2024-10-20 11:51:51.734910947   128.119.245.12  172.18.164.60   TCP      60 80 → 36558 [ACK] Seq=1 Ack=23169 Win=75648 Len=0
 50 2024-10-20 11:51:51.734911249   128.119.245.12  172.18.164.60   TCP      60 80 → 36558 [ACK] Seq=1 Ack=26065 Win=81408 Len=0
 51 2024-10-20 11:51:51.734911299   128.119.245.12  172.18.164.60   TCP      60 80 → 36558 [ACK] Seq=1 Ack=28961 Win=87168 Len=0
 52 2024-10-20 11:51:51.734911344   128.119.245.12  172.18.164.60   TCP      60 80 → 36558 [ACK] Seq=1 Ack=31857 Win=92928 Len=0
 67 2024-10-20 11:51:51.737726735   128.119.245.12  172.18.164.60   TCP      60 80 → 36558 [ACK] Seq=1 Ack=34753 Win=98816 Len=0
 68 2024-10-20 11:51:51.737727003   128.119.245.12  172.18.164.60   TCP      60 80 → 36558 [ACK] Seq=1 Ack=37649 Win=104576 Len=0
 69 2024-10-20 11:51:51.737727051   128.119.245.12  172.18.164.60   TCP      60 80 → 36558 [ACK] Seq=1 Ack=39097 Win=107520 Len=0
 70 2024-10-20 11:51:51.737727094   128.119.245.12  172.18.164.60   TCP      60 80 → 36558 [ACK] Seq=1 Ack=40545 Win=110336 Len=0
 71 2024-10-20 11:51:51.737727142   128.119.245.12  172.18.164.60   TCP      60 80 → 36558 [ACK] Seq=1 Ack=43441 Win=116224 Len=0
 90 2024-10-20 11:51:51.960983353   128.119.245.12  172.18.164.60   TCP      60 80 → 36558 [ACK] Seq=1 Ack=46337 Win=121984 Len=0
 91 2024-10-20 11:51:51.960983547   128.119.245.12  172.18.164.60   TCP      60 80 → 36558 [ACK] Seq=1 Ack=49233 Win=127744 Len=0
 92 2024-10-20 11:51:51.960983578   128.119.245.12  172.18.164.60   TCP      60 80 → 36558 [ACK] Seq=1 Ack=50681 Win=130688 Len=0
103 2024-10-20 11:51:51.971508510   128.119.245.12  172.18.164.60   TCP      60 80 → 36558 [ACK] Seq=1 Ack=52129 Win=133632 Len=0

  Source Port: 80
  Destination Port: 36558
  [Stream index: 0]
  [Stream Packet Number: 2]
> [Conversation completeness: Incomplete, DATA (15)]
  [TCP Segment Len: 0]
  Sequence Number: 0    (relative sequence number)
  Sequence Number (raw): 2837595639

0000  64 d6 9a cf 14 a3 58 b3  8f d9 50 02 08 00 45 00   d·····X· ··P···E·
0010  00 34 00 00 40 00 1f 06  95 f1 80 77 f5 0c ac 12   ·4··@··· ···w····
0020  a4 3c 00 50 8e ce a9 22  45 f7 a4 7f 03 c0 80 12   ·<·P···" E·······
0030  72 10 10 b2 00 00 02 04  05 a8 01 01 04 02 01 03   r······· ········
0040  03 07                                              ··

The RTT to ACK the segment was: Time offset    Packets: 161 · Displayed: 50 (31.1%) · Dropped: 0 (0.0%)    Profile: Default
```

Figure 9: ACK packets

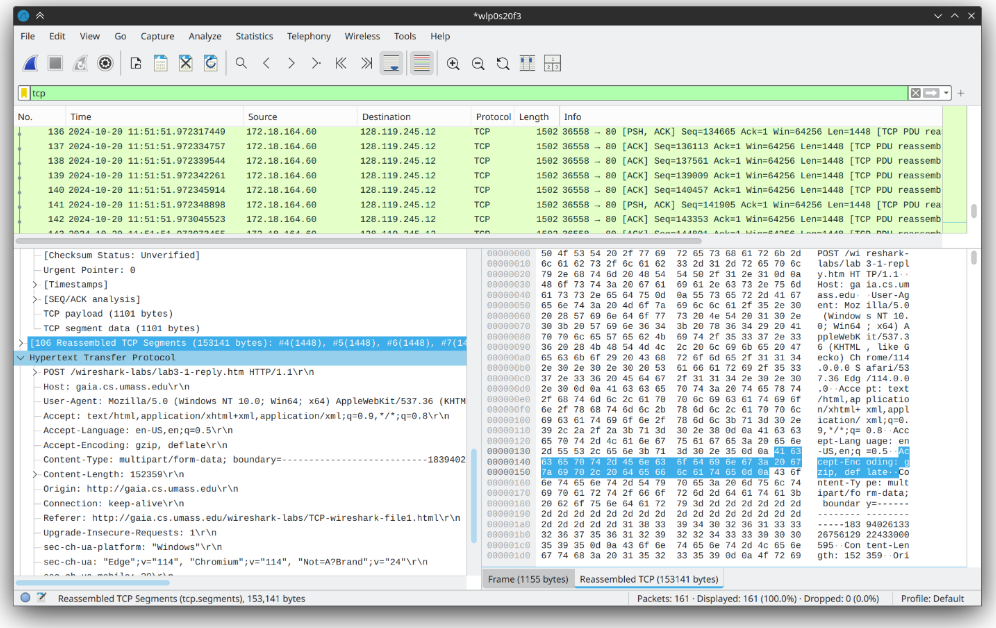


Hover over the graph for details. → 52 pkts, 777 bytes ← 109 pkts, 153 kB

Type Time / Sequence (Stevens)   Stream 0

Mouse [✓ drags] zooms

Figure 10: ACK packets


*wlp0s20f3
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
tcp

```
No.  Time                           Source         Destination     Protocol Length Info
 136 2024-10-20 11:51:51.972317449 172.18.164.60  128.119.245.12  TCP      1502 36558 → 80 [PSH, ACK] Seq=134665 Ack=1 Win=64256 Len=1448 [TCP PDU rea
 137 2024-10-20 11:51:51.972334757 172.18.164.60  128.119.245.12  TCP      1502 36558 → 80 [ACK] Seq=136113 Ack=1 Win=64256 Len=1448 [TCP PDU reassemb
 138 2024-10-20 11:51:51.972339544 172.18.164.60  128.119.245.12  TCP      1502 36558 → 80 [ACK] Seq=137561 Ack=1 Win=64256 Len=1448 [TCP PDU reassemb
 139 2024-10-20 11:51:51.972342261 172.18.164.60  128.119.245.12  TCP      1502 36558 → 80 [ACK] Seq=139009 Ack=1 Win=64256 Len=1448 [TCP PDU reassemb
 140 2024-10-20 11:51:51.972345914 172.18.164.60  128.119.245.12  TCP      1502 36558 → 80 [ACK] Seq=140457 Ack=1 Win=64256 Len=1448 [TCP PDU reassemb
 141 2024-10-20 11:51:51.972348898 172.18.164.60  128.119.245.12  TCP      1502 36558 → 80 [PSH, ACK] Seq=141905 Ack=1 Win=64256 Len=1448 [TCP PDU rea
 142 2024-10-20 11:51:51.973045523 172.18.164.60  128.119.245.12  TCP      1502 36558 → 80 [ACK] Seq=143353 Ack=1 Win=64256 Len=1448 [TCP PDU reassemb
```

```
    [Checksum Status: Unverified]
    Urgent Pointer: 0
  > [Timestamps]
  > [SEQ/ACK analysis]
    TCP payload (1101 bytes)
    TCP segment data (1101 bytes)
> [106 Reassembled TCP Segments (153141 bytes): #4(1448), #5(1448), #6(1448), #7(14
v Hypertext Transfer Protocol
  > POST /wireshark-labs/lab3-1-reply.htm HTTP/1.1\r\n
    Host: gaia.cs.umass.edu\r\n
    User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTM
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\n
    Accept-Language: en-US,en;q=0.5\r\n
    Accept-Encoding: gzip, deflate\r\n
    Content-Type: multipart/form-data; boundary=---------------------------1839402
  > Content-Length: 152359\r\n
    Origin: http://gaia.cs.umass.edu\r\n
    Connection: keep-alive\r\n
    Referer: http://gaia.cs.umass.edu/wireshark-labs/TCP-wireshark-file1.html\r\n
    Upgrade-Insecure-Requests: 1\r\n
    sec-ch-ua-platform: "Windows"\r\n
    sec-ch-ua: "Edge";v="114", "Chromium";v="114", "Not=A?Brand";v="24"\r\n
```

```
00000000  50 4f 53 54 20 2f 77 69  72 65 73 68 61 72 6b 2d   POST /wi reshark-
00000010  6c 61 62 73 2f 6c 61 62  33 2d 31 2d 72 65 70 6c   labs/lab 3-1-repl
00000020  79 2e 68 74 6d 20 48 54  54 50 2f 31 2e 31 0d 0a   y.htm HT TP/1.1··
00000030  48 6f 73 74 3a 20 67 61  69 61 2e 63 73 2e 75 6d   Host: ga ia.cs.um
00000040  61 73 73 2e 65 64 75 0d  0a 55 73 65 72 2d 41 67   ass.edu· ·User-Ag
00000050  65 6e 74 3a 20 4d 6f 7a  69 6c 6c 61 2f 35 2e 30   ent: Moz illa/5.0
00000060  20 28 57 69 6e 64 6f 77  73 20 4e 54 20 31 30 2e    (Window s NT 10.
00000070  30 3b 20 57 69 6e 36 34  3b 20 78 36 34 29 20 41   0; Win64 ; x64) A
00000080  70 70 6c 65 57 65 62 4b  69 74 2f 35 33 37 2e 33   ppleWebK it/537.3
00000090  36 20 28 4b 48 54 4d 4c  2c 20 6c 69 6b 65 20 47   6 (KHTML , like G
000000a0  65 63 6b 6f 29 20 43 68  72 6f 6d 65 2f 31 31 34   ecko) Ch rome/114
000000b0  2e 30 2e 30 2e 30 20 53  61 66 61 72 69 2f 35 33   .0.0.0 S afari/53
000000c0  37 2e 33 36 20 45 64 67  2f 31 31 34 2e 30 2e 30   7.36 Edg /114.0.0
000000d0  2e 30 0d 0a 41 63 63 65  70 74 3a 20 74 65 78 74   .0··Acce pt: text
000000e0  2f 68 74 6d 6c 2c 61 70  70 6c 69 63 61 74 69 6f   /html,ap plicatio
000000f0  6e 2f 78 68 74 6d 6c 2b  78 6d 6c 2c 61 70 70 6c   n/xhtml+ xml,appl
00000100  69 63 61 74 69 6f 6e 2f  78 6d 6c 3b 71 3d 30 2e   ication/ xml;q=0.
00000110  39 2c 2a 2f 2a 3b 71 3d  30 2e 38 0d 0a 41 63 63   9,*/*;q= 0.8··Acc
00000120  65 70 74 2d 4c 61 6e 67  75 61 67 65 3a 20 65 6e   ept-Lang uage: en
00000130  2d 55 53 2c 65 6e 3b 71  3d 30 2e 35 0d 0a 41 63   -US,en;q =0.5··Ac
00000140  63 65 70 74 2d 45 6e 63  6f 64 69 6e 67 3a 20 67   cept-Enc oding: g
00000150  7a 69 70 2c 20 64 65 66  6c 61 74 65 0d 0a 43 6f   zip, def late··Co
00000160  6e 74 65 6e 74 2d 54 79  70 65 3a 20 6d 75 6c 74   ntent-Ty pe: mult
00000170  69 70 61 72 74 2f 66 6f  72 6d 2d 64 61 74 61 3b   ipart/fo rm-data;
00000180  20 62 6f 75 6e 64 61 72  79 3d 2d 2d 2d 2d 2d 2d    boundar y=------
00000190  2d 2d 2d 2d 2d 2d 2d 2d  2d 2d 2d 2d 2d 2d 2d 2d   -------- --------
000001a0  2d 2d 2d 2d 2d 31 38 33  39 34 30 32 36 31 33 33   -----183 94026133
000001b0  32 36 37 35 36 31 32 39  32 32 34 33 33 30 30 30   26756129 22433000
000001c0  35 39 35 0d 0a 43 6f 6e  74 65 6e 74 2d 4c 65 6e   595··Con tent-Len
000001d0  67 74 68 3a 20 31 35 32  33 35 39 0d 0a 4f 72 69   gth: 152 359··Ori
```

Frame (1155 bytes)   Reassembled TCP (153141 bytes)

Reassembled TCP Segments (tcp.segments), 153,141 bytes   Packets: 161 · Displayed: 161 (100.0%) · Dropped: 0 (0.0%)   Profile: Default

Figure 11: HTTP request

Therefore, the throughput is:

$$\text{Throughput} = \frac{BytesTransferred}{TransferTime} = \frac{153141}{0.934512697} = 163.9\text{KB/s}$$

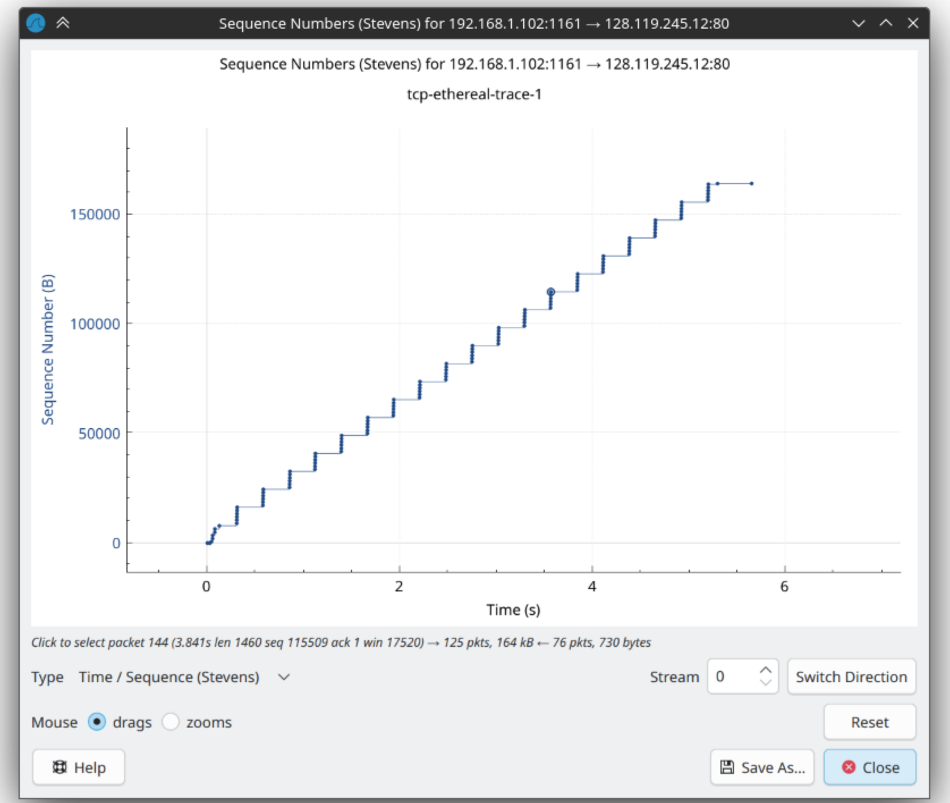
4 TCP congestion control in action

Question 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 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.

From the given trace file, we used the Time-Sequence-Graph to plot the sequence number versus time, as shown in Fig. 12.

The slow start phase begins at the start of the connection and ends around packet 13 at approximately 0.125 seconds. After this point, **congestion avoidance** takes over, but the increase in the congestion window (`cwnd`) is minimal, and it remains relatively constant for the remainder of the connection.



Click to select packet 144 (3.841s len 1460 seq 115509 ack 1 win 17520) → 125 pkts, 164 kB ← 76 pkts, 730 bytes

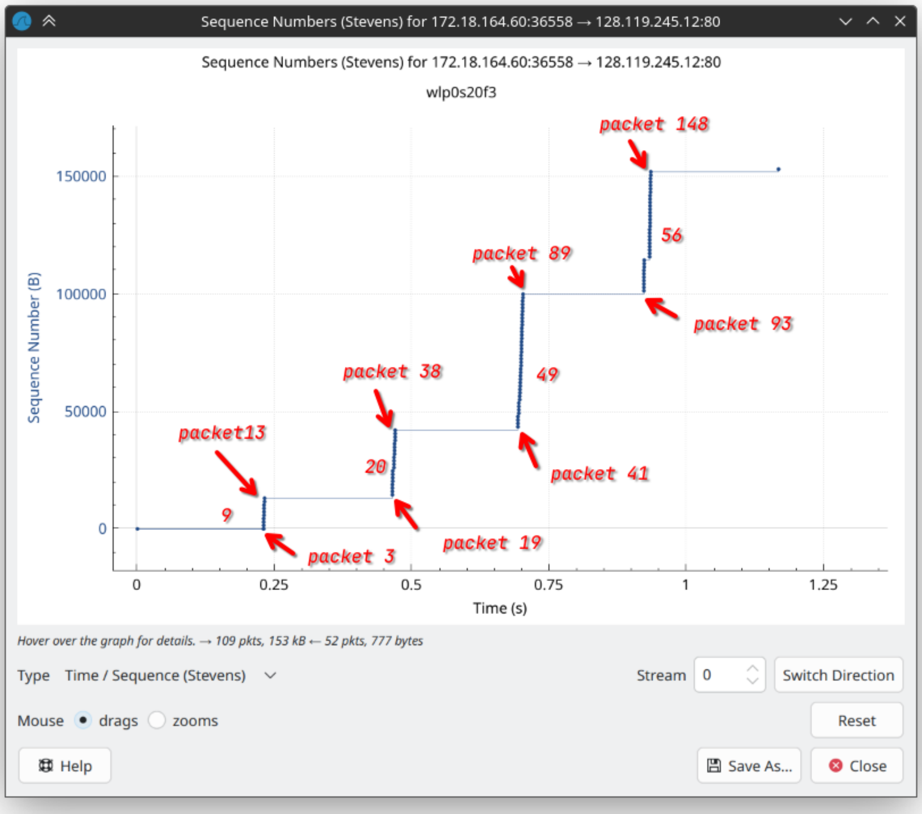
Type: Time / Sequence (Stevens)
Stream: 0
Mouse: [✓ drags] zooms

Figure 12: Time-Sequence-Graph

Compared to the expected behavior, the increase in the congestion window during the congestion avoidance phase is not as pronounced. The `cwnd` seems to stabilize at a value that is likely best suited for the current network conditions.

Question 14.

> 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.ed



Hover over the graph for details. → 109 pkts, 153 kB ← 52 pkts, 777 bytes

Figure 13: Time-Sequence-Graph

Using the Time-Sequence-Graph, we can observe the sequence number versus time plot of segments being sent from the client to the server. The result is shown in Fig. 13. The **slow start** phase begins at the start of the connection, with the `cwnd` increasing almost exponentially from 9 to 49 (though not perfectly exponential). **Congestion avoidance** takes over at packet 93, where the `cwnd` increases linearly from 49 to 56.

Compared to the expected behavior, the slow start phase is not exactly exponential, and the congestion avoidance phase is not precisely linear either. The `cwnd` increases in a manner that is less smooth than the idealized behavior.