

Sai Valluru

Professor Bo Sheng

CS446

11/10/2023

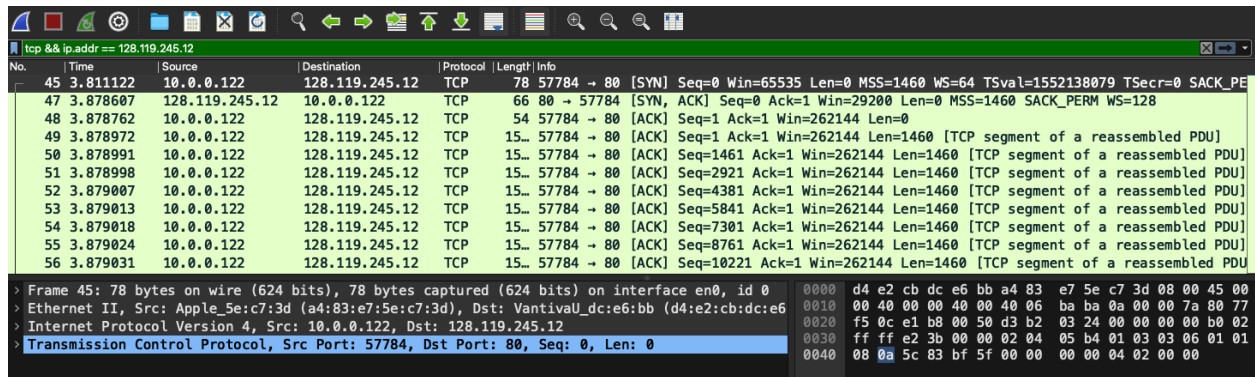
## Wireshark Lab: TCP

1. 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 and TCP port number used by the client computer (source) is:

IP address = 10.0.0.122

TCP port number = 57784



No.	Time	Source	Destination	Protocol	Length	Info
45	3.811122	10.0.0.122	128.119.245.12	TCP	78	57784 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=1552138079 TSecr=0 SACK_PERM
47	3.878607	128.119.245.12	10.0.0.122	TCP	66	80 → 57784 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM WS=128
48	3.878762	10.0.0.122	128.119.245.12	TCP	54	57784 → 80 [ACK] Seq=1 Ack=1 Win=262144 Len=0
49	3.878972	10.0.0.122	128.119.245.12	TCP	15	57784 → 80 [ACK] Seq=1 Ack=1 Win=262144 Len=1460 [TCP segment of a reassembled PDU]
50	3.878991	10.0.0.122	128.119.245.12	TCP	15	57784 → 80 [ACK] Seq=1461 Ack=1 Win=262144 Len=1460 [TCP segment of a reassembled PDU]
51	3.878998	10.0.0.122	128.119.245.12	TCP	15	57784 → 80 [ACK] Seq=2921 Ack=1 Win=262144 Len=1460 [TCP segment of a reassembled PDU]
52	3.879007	10.0.0.122	128.119.245.12	TCP	15	57784 → 80 [ACK] Seq=4381 Ack=1 Win=262144 Len=1460 [TCP segment of a reassembled PDU]
53	3.879013	10.0.0.122	128.119.245.12	TCP	15	57784 → 80 [ACK] Seq=5841 Ack=1 Win=262144 Len=1460 [TCP segment of a reassembled PDU]
54	3.879018	10.0.0.122	128.119.245.12	TCP	15	57784 → 80 [ACK] Seq=7301 Ack=1 Win=262144 Len=1460 [TCP segment of a reassembled PDU]
55	3.879024	10.0.0.122	128.119.245.12	TCP	15	57784 → 80 [ACK] Seq=8761 Ack=1 Win=262144 Len=1460 [TCP segment of a reassembled PDU]
56	3.879031	10.0.0.122	128.119.245.12	TCP	15	57784 → 80 [ACK] Seq=10221 Ack=1 Win=262144 Len=1460 [TCP segment of a reassembled PDU]

Frame 45: 78 bytes on wire (624 bits), 78 bytes captured (624 bits) on interface en0, id 0	
Ethernet II, Src: Apple_Se:c7:3d (a4:83:e7:5e:c7:3d), Dst: VantivaU_dc:e6:bb (d4:e2:cb:dc:e6:bb)	0000 d4 e2 cb dc e6 bb a4 83 e7 5e c7 3d 08 00 45 00
Internet Protocol Version 4, Src: 10.0.0.122, Dst: 128.119.245.12	0010 00 40 00 00 40 00 00 06 ba ba 0a 00 00 7a 80 77
Transmission Control Protocol, Src Port: 57784, Dst Port: 80, Seq: 0, Len: 0	0020 f5 0c e1 b8 00 50 d3 b2 03 24 00 00 00 00 b0 02
	0030 ff ff e2 3b 00 00 02 04 05 b4 01 03 03 06 01 01
	0040 08 0a 5c 83 bf 5f 00 00 00 00 04 02 00 00

Internet Protocol Version 4, Src: 10.0.0.122, Dst: 128.119.245.12

Transmission Control Protocol, Src Port: 57784, Dst Port: 80, Seq: 0, Len: 0

Source Port: 57784

2. What is the sequence number of the TCP SYN segment 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 of the TCP SYN segment used to initiate the TCP connection between the client computer and gaia.cs.umass.edu is 0.

The SYN Flag is in the segment and determines if it is a SYN segment. The SYN Flag is 1, showcasing that this segment is a SYN segment.

```
tcp && ip.addr == 128.119.245.12
No. | Time | Source | Destination | Protocol | Length | Info
---|---|---|---|---|---|---
45 | 3.811122 | 10.0.0.122 | 128.119.245.12 | TCP | 78 | 57784 → 80 [SYN, ACK] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=1552138079 TSecr=0 SACK_PERM
47 | 3.878607 | 128.119.245.12 | 10.0.0.122 | TCP | 66 | 80 → 57784 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM WS=128
48 | 3.878762 | 10.0.0.122 | 128.119.245.12 | TCP | 54 | 57784 → 80 [ACK] Seq=1 Ack=1 Win=262144 Len=0
49 | 3.878972 | 10.0.0.122 | 128.119.245.12 | TCP | 15... | 57784 → 80 [ACK] Seq=1 Ack=1 Win=262144 Len=1460 [TCP segment of a reassembled PDU]
50 | 3.878991 | 10.0.0.122 | 128.119.245.12 | TCP | 15... | 57784 → 80 [ACK] Seq=1461 Ack=1 Win=262144 Len=1460 [TCP segment of a reassembled PDU]
51 | 3.878998 | 10.0.0.122 | 128.119.245.12 | TCP | 15... | 57784 → 80 [ACK] Seq=2921 Ack=1 Win=262144 Len=1460 [TCP segment of a reassembled PDU]
52 | 3.879007 | 10.0.0.122 | 128.119.245.12 | TCP | 15... | 57784 → 80 [ACK] Seq=4381 Ack=1 Win=262144 Len=1460 [TCP segment of a reassembled PDU]
53 | 3.879013 | 10.0.0.122 | 128.119.245.12 | TCP | 15... | 57784 → 80 [ACK] Seq=5841 Ack=1 Win=262144 Len=1460 [TCP segment of a reassembled PDU]
54 | 3.879018 | 10.0.0.122 | 128.119.245.12 | TCP | 15... | 57784 → 80 [ACK] Seq=7301 Ack=1 Win=262144 Len=1460 [TCP segment of a reassembled PDU]
55 | 3.879024 | 10.0.0.122 | 128.119.245.12 | TCP | 15... | 57784 → 80 [ACK] Seq=8761 Ack=1 Win=262144 Len=1460 [TCP segment of a reassembled PDU]

> Frame 45: 78 bytes on wire (624 bits), 78 bytes captured (624 bits) on interface en0, id 0
> Ethernet II, Src: Apple_Sc:c7:3d (a4:83:e7:5e:c7:3d), Dst: Vantiva_dc:e6:bb (d4:e2:cb:dc:e6:bb)
> Internet Protocol Version 4, Src: 10.0.0.122, Dst: 128.119.245.12
> Transmission Control Protocol, Src Port: 57784, Dst Port: 80, Seq: 0, Len: 0
  Source Port: 57784
  Destination Port: 80
  [Stream index: 7]
  [Conversation completeness: Complete, WITH_DATA (31)]
  [TCP Segment Len: 0]
  Sequence Number: 0 (relative sequence number)
  Sequence Number (raw): 3551658788
  [Next Sequence Number: 1 (relative sequence number)]
  Acknowledgment Number: 0
  Acknowledgment number (raw): 0
  1011 .... = Header Length: 44 bytes (11)
  Flags: 0x002 (SYN)
    0000 .... = 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
    ....0... = Syn: Set
    ....0... = Fin: Not set
  [TCP Flags: .....S.]
  Window: 65535
```

```
Transmission Control Protocol, Src Port: 57784, Dst Port: 80, Seq: 0, Len: 0
Source Port: 57784
Destination Port: 80
[Stream index: 7]
[Conversation completeness: Complete, WITH_DATA (31)]
[TCP Segment Len: 0]
Sequence Number: 0      (relative sequence number)
```

```
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
```

3. 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 Acknowledgment 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 in reply to the SYN is 0.

The value of the Acknowledgment field in the SYNACK segment is 1.

In terms of how gaia.cs.umass.edu determined the value, it calculates the ACK number for the SYN-ACK response by adding 1 to the initial sequence number, which the client computer used in its SYN segment. Since the client's sequence number is 0, adding 1 will result in the server's ACK number to be 1.

The SYN Flag along with the Acknowledgement is in the segment and determines if it is a SYNACK segment. The SYN Flag and the ACK Flag are 1, showcasing that this segment is a SYN segment.

```
Transmission Control Protocol, Src Port: 80, Dst Port: 57784, Seq: 0, Ack: 1, Len: 0
Source Port: 80
Destination Port: 57784
[Stream index: 7]
[Conversation completeness: Complete, WITH_DATA (31)]
[TCP Segment Len: 0]
Sequence Number: 0 (relative sequence number)
Sequence Number (raw): 284851863
[Next Sequence Number: 1 (relative sequence number)]
Acknowledgment Number: 1 (relative ack number)
Acknowledgment number (raw): 3551658789
1000 .... = Header Length: 32 bytes (8)
```

## 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
[TCP Flags: .....A..S.]
```

The image shows a Wireshark packet capture analysis. The top pane displays a list of network packets. Packet 47 is selected, showing a SYN-ACK from 128.119.245.12 to 10.0.0.122. The middle pane shows the details of this packet, including Ethernet II, Internet Protocol Version 4, and Transmission Control Protocol. The bottom pane shows the raw data of the packet, which is a TCP segment with a SYN-ACK flag set.

tcp && ip.addr == 128.119.245.12

No.	Time	Source	Destination	Protocol	Length	Info
45	3.811122	10.0.0.122	128.119.245.12	TCP	78	57784 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=1552138079 TSecr=0 SACK_PERM
47	3.878607	128.119.245.12	10.0.0.122	TCP	66	80 → 57784 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM WS=128
48	3.878762	10.0.0.122	128.119.245.12	TCP	54	57784 → 80 [ACK] Seq=1 Ack=1 Win=262144 Len=0
49	3.878972	10.0.0.122	128.119.245.12	TCP	15	57784 → 80 [ACK] Seq=1 Ack=1 Win=262144 Len=1460 [TCP segment of a reassembled PDU]
50	3.878991	10.0.0.122	128.119.245.12	TCP	15	57784 → 80 [ACK] Seq=1461 Ack=1 Win=262144 Len=1460 [TCP segment of a reassembled PDU]
51	3.878998	10.0.0.122	128.119.245.12	TCP	15	57784 → 80 [ACK] Seq=2921 Ack=1 Win=262144 Len=1460 [TCP segment of a reassembled PDU]
52	3.879007	10.0.0.122	128.119.245.12	TCP	15	57784 → 80 [ACK] Seq=4381 Ack=1 Win=262144 Len=1460 [TCP segment of a reassembled PDU]
53	3.879013	10.0.0.122	128.119.245.12	TCP	15	57784 → 80 [ACK] Seq=5841 Ack=1 Win=262144 Len=1460 [TCP segment of a reassembled PDU]
54	3.879018	10.0.0.122	128.119.245.12	TCP	15	57784 → 80 [ACK] Seq=7301 Ack=1 Win=262144 Len=1460 [TCP segment of a reassembled PDU]
55	3.879024	10.0.0.122	128.119.245.12	TCP	15	57784 → 80 [ACK] Seq=8761 Ack=1 Win=262144 Len=1460 [TCP segment of a reassembled PDU]

> Frame 47: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface en0, id 0  
> Ethernet II, Src: VantivaU\_dc:e6:bb (d4:e2:cb:dc:e6:bb), Dst: Apple\_Se:c7:3d (a4:83:e7:5e:c7:3d)  
> Internet Protocol Version 4, Src: 128.119.245.12, Dst: 10.0.0.122  
> Transmission Control Protocol, Src Port: 80, Dst Port: 57784, Seq: 0, Ack: 1, Len: 0

Source Port: 80  
Destination Port: 57784  
[Stream index: 7]  
[Conversation completeness: Complete, WITH\_DATA (31)]  
[TCP Segment Len: 0]  
Sequence Number: 0 (relative sequence number)  
Sequence Number (raw): 284851863  
[Next Sequence Number: 1 (relative sequence number)]  
Acknowledgment Number: 1 (relative ack number)  
Acknowledgment number (raw): 3551658789  
1000 .... = Header Length: 32 bytes (8)  
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  
[TCP Flags: .....A..S.]  
Window: 29200

#### 4. What is the sequence number of the TCP segment containing the HTTP POST command?

The sequence number of the TCP segment containing the HTTP POST command is 1.

Looking into the packet content field in the bottom right, we can see the segment where it says “POST” in its data field. To find the HTTP POST command within a TCP segment, you would need to look in the packet data itself.

```
tcp && ip.addr == 128.119.245.12

No.  Time      Source                Destination           Protocol  Length  Info
45  3.811122  10.0.0.122           128.119.245.12       TCP      78      57784 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=1552138079 TSecr=0 SACK_PERM
47  3.878607  128.119.245.12       10.0.0.122           TCP      66      80 → 57784 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM WS=128
48  3.878762  10.0.0.122           128.119.245.12       TCP      54      57784 → 80 [ACK] Seq=1 Ack=1 Win=262144 Len=0
49  3.878972  10.0.0.122           128.119.245.12       TCP      15_     57784 → 80 [ACK] Seq=1 Ack=1 Win=262144 Len=1460 [TCP segment of a reassembled PDU]
50  3.878991  10.0.0.122           128.119.245.12       TCP      15_     57784 → 80 [ACK] Seq=1461 Ack=1 Win=262144 Len=1460 [TCP segment of a reassembled PDU]
51  3.878998  10.0.0.122           128.119.245.12       TCP      15_     57784 → 80 [ACK] Seq=2921 Ack=1 Win=262144 Len=1460 [TCP segment of a reassembled PDU]
52  3.879007  10.0.0.122           128.119.245.12       TCP      15_     57784 → 80 [ACK] Seq=4381 Ack=1 Win=262144 Len=1460 [TCP segment of a reassembled PDU]
53  3.879013  10.0.0.122           128.119.245.12       TCP      15_     57784 → 80 [ACK] Seq=5841 Ack=1 Win=262144 Len=1460 [TCP segment of a reassembled PDU]
54  3.879018  10.0.0.122           128.119.245.12       TCP      15_     57784 → 80 [ACK] Seq=7301 Ack=1 Win=262144 Len=1460 [TCP segment of a reassembled PDU]
55  3.879024  10.0.0.122           128.119.245.12       TCP      15_     57784 → 80 [ACK] Seq=8761 Ack=1 Win=262144 Len=1460 [TCP segment of a reassembled PDU]
56  3.879031  10.0.0.122           128.119.245.12       TCP      15_     57784 → 80 [ACK] Seq=10221 Ack=1 Win=262144 Len=1460 [TCP segment of a reassembled PDU]
57  3.879038  10.0.0.122           128.119.245.12       TCP      15_     57784 → 80 [ACK] Seq=11681 Ack=1 Win=262144 Len=1460 [TCP segment of a reassembled PDU]
58  3.879043  10.0.0.122           128.119.245.12       TCP      15_     57784 → 80 [ACK] Seq=13141 Ack=1 Win=262144 Len=1460 [TCP segment of a reassembled PDU]
59  3.990700  128.119.245.12       10.0.0.122           TCP      60      80 → 57784 [ACK] Seq=1 Ack=2921 Win=35072 Len=0
60  3.990795  10.0.0.122           128.119.245.12       TCP      15_     57784 → 80 [ACK] Seq=14601 Ack=1 Win=262144 Len=1460 [TCP segment of a reassembled PDU]
61  3.990826  10.0.0.122           128.119.245.12       TCP      15_     57784 → 80 [ACK] Seq=16061 Ack=1 Win=262144 Len=1460 [TCP segment of a reassembled PDU]
62  3.990847  10.0.0.122           128.119.245.12       TCP      15_     57784 → 80 [ACK] Seq=17521 Ack=1 Win=262144 Len=1460 [TCP segment of a reassembled PDU]
63  3.990857  10.0.0.122           128.119.245.12       TCP      15_     57784 → 80 [ACK] Seq=18981 Ack=1 Win=262144 Len=1460 [TCP segment of a reassembled PDU]
64  3.993215  128.119.245.12       10.0.0.122           TCP      60      80 → 57784 [ACK] Seq=1 Ack=4381 Win=38016 Len=0
65  3.993259  10.0.0.122           128.119.245.12       TCP      15_     57784 → 80 [ACK] Seq=20441 Ack=1 Win=262144 Len=1460 [TCP segment of a reassembled PDU]

> Frame 49: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits) on interface en0, id 0
> Ethernet II, Src: Apple_Sec7:3d (a4:83:e7:5e:c7:3d), Dst: VantivaU_dc:e6:bb (d4:e2:cb:dc:e6:bb)
> Internet Protocol Version 4, Src: 10.0.0.122, Dst: 128.119.245.12
> Transmission Control Protocol, Src Port: 57784, Dst Port: 80, Seq: 1, Ack: 1, Len: 1460
  Destination Port: 80
  Source Port: 57784
  [Stream index: 7]
  [Conversation completeness: Complete, WITH_DATA (31)]
  [TCP Segment Len: 1460]
  Sequence Number: 1 (relative sequence number)
  Sequence Number (raw): 3551658789
  [Next Sequence Number: 1461 (relative sequence number)]
  Acknowledgment Number: 1 (relative ack number)
  Acknowledgment number (raw): 284851864
  0101 .... = Header Length: 20 bytes (5)
  0020 f5 0c e1 b8 00 50 d3 b2 03 25 10 fa 7e 98 50 10 ....P...%...~P
  0030 10 00 eb 4d 00 00 50 4f 53 54 20 2f 77 69 72 65 ...M..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.umass.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 4d 61 63 69 6e 74 6f 73 la/S.0 ( Macintos
  00a0 68 3b 20 49 6e 74 65 6c 20 4d 61 63 20 4f 53 20 h; Intel Mac OS
  00b0 58 20 31 30 2e 31 35 3b 20 72 76 3a 31 30 39 2e X 10.15; rv:109
  00c0 30 29 20 47 65 63 6b 6f 2f 32 30 31 30 30 31 30 0) Gecko /2010010
  00d0 31 20 46 69 72 65 66 6f 78 2f 31 31 39 2e 30 0d 1 Firefo x/119.0
  00e0 0a 41 63 63 65 70 74 3a 20 74 65 78 74 2f 68 74 .Accept: text/ht
  00f0 6d 6c 2c 61 70 70 6c 69 63 61 74 69 6f 6e 2f 78 m,appli cation/x
  0100 68 74 6d 6c 2b 78 6d 6c 2c 61 70 70 6c 69 63 61 html+xml, applica
  0110 74 69 6f 6e 2f 78 6d 6c 3b 71 3d 30 2e 39 2c 60 tion/xml; q=0.9, i
  0120 6d 61 67 65 2f 61 76 69 66 7c 69 6d 61 67 65 2f mge/xv i f-image/
```

```
> Frame 49: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits) on interface en0, id 0
> Ethernet II, Src: Apple_Sec7:3d (a4:83:e7:5e:c7:3d), Dst: VantivaU_dc:e6:bb (d4:e2:cb:dc:e6:bb)
> Internet Protocol Version 4, Src: 10.0.0.122, Dst: 128.119.245.12
> Transmission Control Protocol, Src Port: 57784, Dst Port: 80, Seq: 1, Ack: 1, Len: 1460
  Source Port: 57784
  Destination Port: 80
  [Stream index: 7]
  [Conversation completeness: Complete, WITH_DATA (31)]
  [TCP Segment Len: 1460]
  Sequence Number: 1 (relative sequence number)
```

d4	e2	cb	dc	e6	bb	a4	83	e7	5e	c7	3d	08	00	45	00	...	...	..^.=..E..
05	dc	00	00	40	00	40	06	b5	1e	0a	00	00	7a	80	77	...	@..@..	.....z..w
f5	0c	e1	b8	00	50	d3	b2	03	25	10	fa	7e	98	50	10	...	...P..	..%...~..P..
10	00	eb	4d	00	00	50	4f	53	54	20	2f	77	69	72	65	...	M..	P0 ST/wire
73	68	61	72	6b	2d	6c	61	62	73	2f	6c	61	62	33	2d	shark-la	bs/lab3-	
31	2d	72	65	70	6c	79	2e	68	74	6d	20	48	54	54	50	1-reply.	htm HTTP	
2f	31	2e	31	0d	0a	48	6f	73	74	3a	20	67	61	69	61	/1.1..Ho	st: gaia	
2e	63	73	2e	75	6d	61	73	73	2e	65	64	75	0d	0a	55	.cs.umas	s.edu..U	

5. Identify the 3 TCP ACKs from the server, and find their corresponding TCP data segments from the client. Explain how you found these 3 TCP data segments. At what time was each TCP data segment sent by the client? When was each TCP ACK received? Given the difference, what are the RTT values for these 3 TCP segments?

(Note: Wireshark has a nice feature that allows you to plot the RTT for each of the TCP segments sent. You can use it to compare with your own calculation. Select a TCP segment in the “listing of captured packets” window that is being sent from the client to the gaia.cs.umass.edu server. Then select *Statistics -> TCP Stream Graph -> Round Trip Time Graph*.)

3 TCP ACKs from the server are frames No. 59, 64, and 67.

59	3.990709	128.119.245.12	10.0.0.122	TCP	60 80 → 57784 [ACK] Seq=1 Ack=2921 Win=35072 Len=0
64	3.993215	128.119.245.12	10.0.0.122	TCP	60 80 → 57784 [ACK] Seq=1 Ack=4381 Win=38016 Len=0
67	3.994257	128.119.245.12	10.0.0.122	TCP	60 80 → 57784 [ACK] Seq=1 Ack=5841 Win=40960 Len=0

Corresponding TCP data segments from the client are frames No. 50, 51, and 52.

50	3.878991	10.0.0.122	128.119.245.12	TCP	15... 57784 → 80 [ACK] Seq=1461 Ack=1 Win=262144 Len=1460 [TCP segment of a reassembled PDU]
51	3.878998	10.0.0.122	128.119.245.12	TCP	15... 57784 → 80 [ACK] Seq=2921 Ack=1 Win=262144 Len=1460 [TCP segment of a reassembled PDU]
52	3.879007	10.0.0.122	128.119.245.12	TCP	15... 57784 → 80 [ACK] Seq=4381 Ack=1 Win=262144 Len=1460 [TCP segment of a reassembled PDU]

Explanation of how I found the 3 TCP data Segments.

For example, frame 59 TCP ACK from the server has an acknowledgment number of 2921. Doing the calculation,  $2921 - 1 = 2920$ . This means it is confirming receipt of all data up to and including the byte 2920 and is now expecting the next byte, which would be byte 2921. When we look at frame 50, it has a sequence number of 1461 and carries 1460 bytes of data. This means it contains bytes numbered 1461 to 2920 ( $1461 + 1460 - 1$ ) of the data stream. So frame



59 is directly acknowledging the data sent in segment frame 50. Similar calculations go for Frame 64 and 67.

For Frame 64, it has an acknowledgment number of 4381. It confirms the receipt of all data up to and including byte 4380, and it's now expected byte would be 4381. If you look at the previous segment sent from the client that was not yet acknowledged, Frame 51, you would expect to find that Frame 51 sent data up to byte 4380.

For Frame 67, it has an acknowledgment number of 5841. It confirms the receipt of all data up to and including byte 5840, and it's now expected byte would be 5841. Again, looking back at the previously unacknowledged data sent from the client, Frame 52, you would expect Frame 52 to have sent data up to byte 5840.

#### Frame 59:

```

v Transmission Control Protocol, Src Port: 80, Dst Port: 57784, Seq: 1, Ack: 2921, Len: 0
  Source Port: 80
  Destination Port: 57784
  [Stream index: 7]
  [Conversation completeness: Complete, WITH_DATA (31)]
  [TCP Segment Len: 0]
  Sequence Number: 1      (relative sequence number)
  Sequence Number (raw): 284851864
  [Next Sequence Number: 1      (relative sequence number)]
  Acknowledgment Number: 2921      (relative ack number)
  Acknowledgment number (raw): 3551661709
  0101 .... = Header Length: 20 bytes (5)
  > Flags: 0x010 (ACK)
  Window: 274
  [Calculated window size: 35072]
  [Window size scaling factor: 128]
  Checksum: 0xdae9 [unverified]
  [Checksum Status: Unverified]
  Urgent Pointer: 0
  > [Timestamps]
  v [SEQ/ACK analysis]
    [This is an ACK to the segment in frame: 50]
    [The RTT to ACK the segment was: 0.111718000 seconds]
    [iRTT: 0.067640000 seconds]
```

## Frame 50:

```

▼ Transmission Control Protocol, Src Port: 57784, Dst Port: 80, Seq: 1461, Ack: 1, Len: 1460
  Source Port: 57784
  Destination Port: 80
  [Stream index: 7]
  [Conversation completeness: Complete, WITH_DATA (31)]
  [TCP Segment Len: 1460]
  Sequence Number: 1461      (relative sequence number)
  Sequence Number (raw): 3551660249
  [Next Sequence Number: 2921      (relative sequence number)]
  Acknowledgment Number: 1      (relative ack number)
  Acknowledgment number (raw): 284851864
  0101 .... = Header Length: 20 bytes (5)
  > Flags: 0x010 (ACK)
  Window: 4096
  [Calculated window size: 262144]
  [Window size scaling factor: 64]
  Checksum: 0x6e6a [unverified]
  [Checksum Status: Unverified]
  Urgent Pointer: 0
  > [Timestamps]
  ▼ [SEQ/ACK analysis]
    [iRTT: 0.067640000 seconds]
    [Bytes in flight: 2920]
    [Bytes sent since last PSH flag: 2920]
  TCP payload (1460 bytes)
  \[Reassembled PDU in frame: 206\]
  TCP segment data (1460 bytes)

```

## Frame 64:

```

▼ Transmission Control Protocol, Src Port: 80, Dst Port: 57784, Seq: 1, Ack: 4381, Len: 0
  Source Port: 80
  Destination Port: 57784
  [Stream index: 7]
  [Conversation completeness: Complete, WITH_DATA (31)]
  [TCP Segment Len: 0]
  Sequence Number: 1      (relative sequence number)
  Sequence Number (raw): 284851864
  [Next Sequence Number: 1      (relative sequence number)]
  Acknowledgment Number: 4381      (relative ack number)
  Acknowledgment number (raw): 3551663169
  0101 .... = Header Length: 20 bytes (5)
  > Flags: 0x010 (ACK)
  Window: 297
  [Calculated window size: 38016]
  [Window size scaling factor: 128]
  Checksum: 0xd51e [unverified]
  [Checksum Status: Unverified]
  Urgent Pointer: 0
  > [Timestamps]
  ▼ [SEQ/ACK analysis]
    \[This is an ACK to the segment in frame: 51\]
    [The RTT to ACK the segment was: 0.114217000 seconds]
    [iRTT: 0.067640000 seconds]

```

## Frame 51:

```

Transmission Control Protocol, Src Port: 57784, Dst Port: 80, Seq: 2921, Ack: 1, Len: 1460
  Source Port: 57784
  Destination Port: 80
  [Stream index: 7]
  [Conversation completeness: Complete, WITH_DATA (31)]
  [TCP Segment Len: 1460]
  Sequence Number: 2921      (relative sequence number)
  Sequence Number (raw): 3551661709
  [Next Sequence Number: 4381      (relative sequence number)]
  Acknowledgment Number: 1    (relative ack number)
  Acknowledgment number (raw): 284851864
  0101 .... = Header Length: 20 bytes (5)
  > Flags: 0x010 (ACK)
  Window: 4096
  [Calculated window size: 262144]
  [Window size scaling factor: 64]
  Checksum: 0x85ca [unverified]
  [Checksum Status: Unverified]
  Urgent Pointer: 0
  > [Timestamps]
  > [SEQ/ACK analysis]
    [iRTT: 0.067640000 seconds]
    [Bytes in flight: 4380]
    [Bytes sent since last PSH flag: 4380]
  TCP payload (1460 bytes)
  [Reassembled PDU in frame: 206]
  TCP segment data (1460 bytes)

```

## Frame 67:

```

Transmission Control Protocol, Src Port: 80, Dst Port: 57784, Seq: 1, Ack: 5841, Len: 0
  Source Port: 80
  Destination Port: 57784
  [Stream index: 7]
  [Conversation completeness: Complete, WITH_DATA (31)]
  [TCP Segment Len: 0]
  Sequence Number: 1      (relative sequence number)
  Sequence Number (raw): 284851864
  [Next Sequence Number: 1      (relative sequence number)]
  Acknowledgment Number: 5841      (relative ack number)
  Acknowledgment number (raw): 3551664629
  0101 .... = Header Length: 20 bytes (5)
  > Flags: 0x010 (ACK)
  Window: 320
  [Calculated window size: 40960]
  [Window size scaling factor: 128]
  Checksum: 0xcf53 [unverified]
  [Checksum Status: Unverified]
  Urgent Pointer: 0
  > [Timestamps]
  > [SEQ/ACK analysis]
    [This is an ACK to the segment in frame: 52]
    [The RTT to ACK the segment was: 0.115250000 seconds]
    [iRTT: 0.067640000 seconds]

```

## Frame 50:

```

▼ Transmission Control Protocol, Src Port: 57784, Dst Port: 80, Seq: 4381, Ack: 1, Len: 1460
  Source Port: 57784
  Destination Port: 80
  [Stream index: 7]
  [Conversation completeness: Complete, WITH_DATA (31)]
  [TCP Segment Len: 1460]
  Sequence Number: 4381      (relative sequence number)
  Sequence Number (raw): 3551663169
  [Next Sequence Number: 5841      (relative sequence number)]
  Acknowledgment Number: 1    (relative ack number)
  Acknowledgment number (raw): 284851864
  0101 .... = Header Length: 20 bytes (5)
  > Flags: 0x010 (ACK)
  Window: 4096
  [Calculated window size: 262144]
  [Window size scaling factor: 64]
  Checksum: 0xfb53 [unverified]
  [Checksum Status: Unverified]
  Urgent Pointer: 0
  > [Timestamps]
  ▼ [SEQ/ACK analysis]
    [iRTT: 0.067640000 seconds]
    [Bytes in flight: 5840]
    [Bytes sent since last PSH flag: 5840]
  TCP payload (1460 bytes)
  \[Reassembled PDU in frame: 206\]
  TCP segment data (1460 bytes)

```

The statement, "This is an ACK to the segment in Frame: No." also gives a direct reference to the frame that the acknowledgment is for.

At what time each TCP data segment was sent by the client, when was each TCP ACK received, and given the difference, the RTT values for the 3 TCP segments are all showcased below:

Segments & Frames	Time TCP data segment sent by the client	Time TCP ACK was received	RTT
Segment 1 [Frames 50, 59]	3.878991	3.990709	0.111718
Segment 2 [Frames 51, 64]	3.878998	3.993215	0.114217
Segment 3 [Frames 52, 67]	3.879007	3.994307	0.11525

#### How to Calculate RTT

RTT = ACK received time - Segment sent time

Seg 1 RTT = 3.990709 - 3.878991 = 0.111718 sec

Seg 2 RTT = 3.993215 - 3.878998 = 0.114217 sec

Seg 3 RTT = 3.994257 - 3.879007 = 0.115250 sec

```
50 3.878991 10.0.0.122 128.119.245.12 TCP 15_ 57784 → 80 [ACK] Seq=1461 Ack=1 Win=262144 Len=1460 [TCP segment of a reassembled PDU]
51 3.878998 10.0.0.122 128.119.245.12 TCP 15_ 57784 → 80 [ACK] Seq=2921 Ack=1 Win=262144 Len=1460 [TCP segment of a reassembled PDU]
52 3.879007 10.0.0.122 128.119.245.12 TCP 15_ 57784 → 80 [ACK] Seq=4381 Ack=1 Win=262144 Len=1460 [TCP segment of a reassembled PDU]
```

```
59 3.990709 128.119.245.12 10.0.0.122 TCP 60 80 → 57784 [ACK] Seq=1 Ack=2921 Win=35072 Len=0
```

```
64 3.993215 128.119.245.12 10.0.0.122 TCP 60 80 → 57784 [ACK] Seq=1 Ack=4381 Win=38016 Len=0
```

```
67 3.994257 128.119.245.12 10.0.0.122 TCP 60 80 → 57784 [ACK] Seq=1 Ack=5841 Win=40960 Len=0
```

6. Among the 3 TCP ACKs you identified in the previous question, 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?

Among the 3 TCP ACKs, the minimum amount of available buffer space advertised at the receiver for the entire trace are:

For 1st TCP ACK (Frame 59): 35072 bytes

For 2nd TCP ACK (Frame 64): 38016 bytes

For 3rd TCP ACK (Frame 67): 40960 bytes

As for if the lack of receiver buffer space ever throttles the sender, based on the TCP ACK packets info, the receiver's window size is either the same or increasing until it reaches a certain max buffer size. So it's an indication that the sender will not be throttled due to a lack of receiver buffer space.

```
59 3.990709 128.119.245.12 10.0.0.122 TCP 60 80 → 57784 [ACK] Seq=1 Ack=2921 Win=35072 Len=0
60 3.990795 10.0.0.122 128.119.245.12 TCP 15_ 57784 → 80 [ACK] Seq=14601 Ack=1 Win=262144 Len=
61 3.990826 10.0.0.122 128.119.245.12 TCP 15_ 57784 → 80 [ACK] Seq=16061 Ack=1 Win=262144 Len=
62 3.990847 10.0.0.122 128.119.245.12 TCP 15_ 57784 → 80 [ACK] Seq=17521 Ack=1 Win=262144 Len=
> Frame 59: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface en0, id 0
> Ethernet II, Src: VantivaU_dc:e6:bb (d4:e2:cb:dc:e6:bb), Dst: Apple_5e:c7:3d (a4:83:e7:5e:c7:3d)
> Internet Protocol Version 4, Src: 128.119.245.12, Dst: 10.0.0.122
> Transmission Control Protocol, Src Port: 80, Dst Port: 57784, Seq: 1, Ack: 2921, Len: 0
  Source Port: 80
  Destination Port: 57784
  [Stream index: 7]
  [Conversation completeness: Complete, WITH_DATA (31)]
  [TCP Segment Len: 0]
  Sequence Number: 1 (relative sequence number)
  Sequence Number (raw): 284851864
  [Next Sequence Number: 1 (relative sequence number)]
  Acknowledgment Number: 2921 (relative ack number)
  Acknowledgment number (raw): 3551661709
  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: 274
  [Calculated window size: 35072]
  [Window size scaling factor: 128]
```

64	3.993215	128.119.245.12	10.0.0.122	TCP	60	80 → 57784	[ACK]	Seq=1 Ack=4381 Win=38016 Len=0
65	3.993259	10.0.0.122	128.119.245.12	TCP	15...	57784 → 80	[ACK]	Seq=20441 Ack=1 Win=262144 Len=0
66	3.993273	10.0.0.122	128.119.245.12	TCP	15...	57784 → 80	[ACK]	Seq=21901 Ack=1 Win=262144 Len=0
67	3.994257	128.119.245.12	10.0.0.122	TCP	60	80 → 57784	[ACK]	Seq=1 Ack=5841 Win=40960 Len=0

```

> Frame 64: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface en0, id 0
> Ethernet II, Src: VantivaU_dc:e6:bb (d4:e2:cb:dc:e6:bb), Dst: Apple_Se:c7:3d (a4:83:e7:5e:c7:3d)
> Internet Protocol Version 4, Src: 128.119.245.12, Dst: 10.0.0.122
< Transmission Control Protocol, Src Port: 80, Dst Port: 57784, Seq: 1, Ack: 4381, Len: 0
  Source Port: 80
  Destination Port: 57784
  [Stream index: 7]
  [Conversation completeness: Complete, WITH_DATA (31)]
  [TCP Segment Len: 0]
  Sequence Number: 1 (relative sequence number)
  Sequence Number (raw): 284851864
  [Next Sequence Number: 1 (relative sequence number)]
  Acknowledgment Number: 4381 (relative ack number)
  Acknowledgment number (raw): 3551663169
  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: 297
  [Calculated window size: 38016]
  [Window size scaling factor: 128]

```

67	3.994257	128.119.245.12	10.0.0.122	TCP	60	80 → 57784	[ACK]	Seq=1 Ack=5841 Win=40960 Len=0
68	3.994294	10.0.0.122	128.119.245.12	TCP	15...	57784 → 80	[ACK]	Seq=23361 Ack=1 Win=262144 Len=0
69	3.994307	10.0.0.122	128.119.245.12	TCP	15...	57784 → 80	[ACK]	Seq=24821 Ack=1 Win=262144 Len=0
70	3.995018	128.119.245.12	10.0.0.122	TCP	60	80 → 57784	[ACK]	Seq=1 Ack=13141 Win=55552 Len=0

```

> Frame 67: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface en0, id 0
> Ethernet II, Src: VantivaU_dc:e6:bb (d4:e2:cb:dc:e6:bb), Dst: Apple_Se:c7:3d (a4:83:e7:5e:c7:3d)
> Internet Protocol Version 4, Src: 128.119.245.12, Dst: 10.0.0.122
< Transmission Control Protocol, Src Port: 80, Dst Port: 57784, Seq: 1, Ack: 5841, Len: 0
  Source Port: 80
  Destination Port: 57784
  [Stream index: 7]
  [Conversation completeness: Complete, WITH_DATA (31)]
  [TCP Segment Len: 0]
  Sequence Number: 1 (relative sequence number)
  Sequence Number (raw): 284851864
  [Next Sequence Number: 1 (relative sequence number)]
  Acknowledgment Number: 5841 (relative ack number)
  Acknowledgment number (raw): 3551664629
  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: 320
  [Calculated window size: 40960]
  [Window size scaling factor: 128]

```

## 7. What is the throughput (bytes transferred per unit time) for the TCP connection?

Explain how you calculated this value.

To calculate the throughput (bytes transferred per unit time) for the TCP connection, we need to determine the total amount of data transferred.

- Find the first packet in the file transfer, which is Frame 45
- Find the final Ack from the server, which is Frame 219

No.	Time	Source	Destination	Protocol	Length	Info
45	3.811122	10.0.0.122	128.119.245.12	TCP	78	57784 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=1552138079 TSecr=0 SACK_PERM
219	4.098282	128.119.245.12	10.0.0.122	TCP	60	80 → 57784 [ACK] Seq=1 Ack=152977 Win=227840 Len=0

To figure out the total amount of data transferred, we can take a look at the Ack when the payload is Len = 0. So in this scenario, the Ack is equal to 152,977 Bytes. Since the Len = 0 when the Seq = 1 at the start of the session, we can simply see that the bytes transferred is  $152,977 - 1 = 152,976$  Bytes.

[The Total Num of Bytes = Final Ack - Initial Seq]

219	4.098282	128.119.245.12	10.0.0.122	TCP	60	80 → 57784 [ACK] Seq=1 Ack=152977 Win=227840 Len=0
220	4.099447	128.119.245.12	10.0.0.122	HTTP	831	HTTP/1.1 200 OK (text/html)

```
> Frame 219: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface en0, id 0
> Ethernet II, Src: VantivaU_dc:e6:bb (d4:e2:cb:dc:e6:bb), Dst: Apple_5e:c7:3d (a4:83:e7:5e:c7:3d)
> Internet Protocol Version 4, Src: 128.119.245.12, Dst: 10.0.0.122
√ Transmission Control Protocol, Src Port: 80, Dst Port: 57784, Seq: 1, Ack: 152977, Len: 0
  Source Port: 80
  Destination Port: 57784
  [Stream index: 7]
  [Conversation completeness: Complete, WITH_DATA (31)]
  [TCP Segment Len: 0]
  Sequence Number: 1 (relative sequence number)
  Sequence Number (raw): 284851864
  [Next Sequence Number: 1 (relative sequence number)]
  Acknowledgment Number: 152977 (relative ack number)
```



Now, we need to calculate the total transmission time.

- Finding the timestamp of the first TCP segment that contains data.
- Finding the timestamp of the last ACK segment.
- Then, calculate the total transmission time by taking the difference between these two timestamps.

$\text{Time}(\text{Frame } 219) - \text{Time}(\text{Frame } 45) = \text{Total Time Transmission}$

$4.098282 - 3.811122 = 0.28716 \text{ seconds}$

Finally, to calculate the throughput, you need to divide the total amount of data transferred by the total transmission time. This will result in the TCP throughput using sequence numbers.

The total amount of data transferred = 152,976 bytes

The total transmission time = 0.28716 seconds

$\text{Throughput} = \text{Total amount of data transferred} / \text{Total transmission time}$

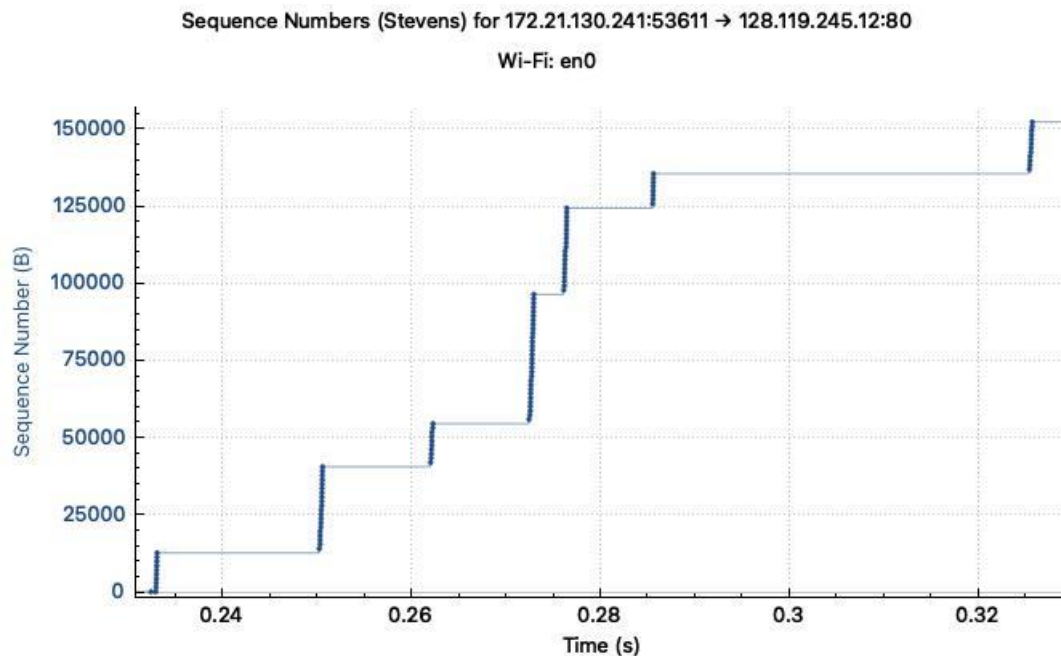
$152,976 \text{ bytes} / 0.28716 \text{ seconds} = \text{approximately } 532,720 \text{ bytes/second}$

(Exactly 532,720.434601 bytes/seconds)

Conversion from Bps to KBps:  $532,720 / 1024 = 520.23$

So the throughput for this TCP connection is approximately 520 KBps

8. Use the *Time-Sequence-Graph(Stevens)* plotting tool to view the sequence number versus the time plot of segments being sent from the client to the gaia.cs.umass.edu server. Can you identify where is TCP's slow start phase and congestion avoidance phase? Can you identify any congestion?



In order to identify where the TCP's slow start phase begins, we need to analyze the time plot to see the sequence number increasing exponentially, which is usually present at the beginning of the TCP connection. So we can make an inference that the slow start of the TCP approximately starts around 0.2325, because it is between 0.23 and 0.235, and ends approximately around 0.25, which is between 0.24 and 0.26.

As for Congestion avoidance, it usually starts after when the slow start threshold is reached. Instead of increasing exponentially, the increase in sequence numbers is more linear. We could guess that it starts at approximately around 0.2725 since it's between 0.27 and 0.275 because the amount of data being sent starts to reduce.