

Task 1:

1. How many UDP packets are exchanged in this communication between iperf3 client and remote server ?

We find that the IP address of ping.online.net is 62.210.18.40. We use the filter `ip.addr == 62.210.18.40 && udp` to filter out the UDP packets that are exchanged between client and server.

We find 995 packets were exchanged between client and server and this can be seen in both the filtered list as well as the endpoints.

Even though Wireshark shows multiple IPv4 frames using proto = UDP, these frames are combined to form a single UDP packet which we have counted here.

The image shows a Wireshark packet capture window titled "task1.pcapng". The filter bar at the top contains the filter `ip.addr == 62.210.18.40 && udp`. The packet list shows a series of UDP packets from source 62.210.18.40 to destination 192.168.43.17. Packet 297 is selected, showing a length of 1402 bytes. The packet details pane for packet 297 shows: Frame 297: 1402 bytes on wire (11216 bits), 1402 bytes captured (11216 bits) on interface any, id 0; Linux cooked capture; Internet Protocol Version 4, Src: 62.210.18.40, Dst: 192.168.43.17; User Datagram Protocol, Src Port: 5208, Dst Port: 48519; Data (1358 bytes). The packet bytes pane shows the raw data in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Length	Info
285	26.525907870	192.168.43.17	62.210.18.40	UDP	48	48519 → 5208 Len=4
286	27.032599371	62.210.18.40	192.168.43.17	UDP	48	5208 → 48519 Len=4
287	27.032638679	62.210.18.40	192.168.43.17	UDP	1402	5208 → 48519 Len=1358
290	27.032714093	62.210.18.40	192.168.43.17	UDP	1402	5208 → 48519 Len=1358
291	27.032722600	62.210.18.40	192.168.43.17	UDP	1402	5208 → 48519 Len=1358
292	27.032728621	62.210.18.40	192.168.43.17	UDP	1402	5208 → 48519 Len=1358
293	27.032733934	62.210.18.40	192.168.43.17	UDP	1402	5208 → 48519 Len=1358
294	27.041855188	62.210.18.40	192.168.43.17	UDP	1402	5208 → 48519 Len=1358
295	27.056004486	62.210.18.40	192.168.43.17	UDP	1402	5208 → 48519 Len=1358
296	27.073073227	62.210.18.40	192.168.43.17	UDP	1402	5208 → 48519 Len=1358
297	27.085790111	62.210.18.40	192.168.43.17	UDP	1402	5208 → 48519 Len=1358
298	27.100897390	62.210.18.40	192.168.43.17	UDP	1402	5208 → 48519 Len=1358
299	27.116150156	62.210.18.40	192.168.43.17	UDP	1402	5208 → 48519 Len=1358
300	27.141907460	62.210.18.40	192.168.43.17	UDP	1402	5208 → 48519 Len=1358
301	27.152221775	62.210.18.40	192.168.43.17	UDP	1402	5208 → 48519 Len=1358
302	27.172293227	62.210.18.40	192.168.43.17	UDP	1402	5208 → 48519 Len=1358
303	27.181947227	62.210.18.40	192.168.43.17	UDP	1402	5208 → 48519 Len=1358
304	27.216225214	62.210.18.40	192.168.43.17	UDP	1402	5208 → 48519 Len=1358
305	27.241033431	62.210.18.40	192.168.43.17	UDP	1402	5208 → 48519 Len=1358
306	27.250824722	62.210.18.40	192.168.43.17	UDP	1402	5208 → 48519 Len=1358
307	27.266930759	62.210.18.40	192.168.43.17	UDP	1402	5208 → 48519 Len=1358
308	27.331958815	62.210.18.40	192.168.43.17	UDP	1402	5208 → 48519 Len=1358
309	27.331970140	62.210.18.40	192.168.43.17	UDP	1402	5208 → 48519 Len=1358

> Frame 297: 1402 bytes on wire (11216 bits), 1402 bytes captured (11216 bits) on interface any, id 0
> Linux cooked capture
> Internet Protocol Version 4, Src: 62.210.18.40, Dst: 192.168.43.17
> User Datagram Protocol, Src Port: 5208, Dst Port: 48519
> Data (1358 bytes)

```
0000 00 00 00 01 00 06 08 25 25 5b 7e 58 00 00 08 00 .....%[~X...
0010 45 28 05 6a 05 b4 40 00 32 11 00 f4 3e d2 12 28 E(.j...@.2...>...
0020 c0 a8 2b 11 14 58 bd 87 05 56 52 f1 5f 95 6d 3f ..+...X...VR...m?
0030 00 00 a8 f6 00 00 00 0c bf 7a 4b d2 4a 43 9e d3 .....zK-JC...
0040 55 91 9f aa 58 51 08 06 64 f2 89 96 3b e6 0e d1 U...XQ...d...;...
0050 59 30 8b f2 c4 40 c1 83 ba 0c 56 04 50 f4 d8 a5 Y0...@...-V-P...
0060 86 77 4f de c9 57 e4 2d 4a 6d c3 85 53 d1 56 ac wO...W...-Jm...S-V
0070 01 e1 9f c5 22 60 49 dc 6c 9f e1 bc 93 b9 62 19 ....I...l....b...
```

User Datagram Protocol: Protocol | Packets: 1377 · Displayed: 995 (72.3%) | Profile: Default

Wireshark · Endpoints · task1.pcapng

Ethernet		IPv4 · 2		IPv6	TCP	UDP · 2	
Address	Port	Packets	Bytes	Tx Packets	Tx Bytes	Rx Packets	Rx Bytes
62.210.18.40	5208	995	1392 k	994	1392 k	1	48
192.168.43.17	48519	995	1392 k	1	48	994	1392 k

☐ Name resolution
 ☒ Limit to display filter
 Endpoint Types

2. Who is sending bulk data to whom ? What is the average size of the packet sent?

We use the filter `ip.addr == 62.210.18.40` to filter out packets that are transferred between client and server. We use conversation statistics to find the data transferred between the two.

We see that 48 bytes are transferred from A to B whereas 1392k bytes are transferred B to A where A is the client system and B is the server. We can conclude from this that a bulk of data is transferred from server to client.

For packet size,

we have 1 packet from client to server and 48 bytes of data is transferred. Therefore, the size of packet sent from client to server is 48 bytes.

we have 995 packets from server to client and 1392k bytes of data is transferred. Therefore, size of packet from server to client is 1432.57 bytes.

$(1392k/995 \Rightarrow 1392 * 1024 / 995 \Rightarrow 1,432.5708542713567839195979899497)$

Wireshark · Conversations · task1.pcapng

Ethernet		IPv4 · 1		IPv6	TCP	UDP · 1											
Address A	Port A	Address B	Port B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s B → A				
192.168.43.17	48519	62.210.18.40	5208	995	1392 k	1	48	994	1392 k	26.525908	10.6948	35	1041 k				

☐ Name resolution
 ☒ Limit to display filter
 ☐ Absolute start time
 Conversation Types ▾

Copy ▾
Follow Stream...
Graph...
Close
Help

- Calculate the throughput (bytes transferred per unit time) for this UDP conversation using UDP's length field. Explain how you calculated this value using Wireshark capture in this experiment along with relevant screenshots. Verify your calculation with the one done by Wireshark using "Capture File properties" as well with the one displayed by iperf3 terminal. If you observe the major difference in your calculation and with the other two listed here, comment why and how?

task1.pcapng

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

ip.addr == 62.210.18.40 && udp

No.	Time	Source	Destination	Protocol	Length	Info
285	26.525907870	192.168.43.17	62.210.18.40	UDP	48	48519 → 5208 Len=4
286	27.032599371	62.210.18.40	192.168.43.17	UDP	48	5208 → 48519 Len=4
287	27.032638679	62.210.18.40	192.168.43.17	UDP	1402	5208 → 48519 Len=1358
290	27.032714093	62.210.18.40	192.168.43.17	UDP	1402	5208 → 48519 Len=1358
291	27.032722600	62.210.18.40	192.168.43.17	UDP	1402	5208 → 48519 Len=1358
292	27.032728621	62.210.18.40	192.168.43.17	UDP	1402	5208 → 48519 Len=1358
293	27.032733934	62.210.18.40	192.168.43.17	UDP	1402	5208 → 48519 Len=1358
294	27.041855188	62.210.18.40	192.168.43.17	UDP	1402	5208 → 48519 Len=1358
295	27.056004486	62.210.18.40	192.168.43.17	UDP	1402	5208 → 48519 Len=1358
296	27.073073227	62.210.18.40	192.168.43.17	UDP	1402	5208 → 48519 Len=1358
297	27.085790111	62.210.18.40	192.168.43.17	UDP	1402	5208 → 48519 Len=1358
298	27.100897390	62.210.18.40	192.168.43.17	UDP	1402	5208 → 48519 Len=1358
299	27.116150156	62.210.18.40	192.168.43.17	UDP	1402	5208 → 48519 Len=1358
300	27.141907460	62.210.18.40	192.168.43.17	UDP	1402	5208 → 48519 Len=1358
301	27.152221775	62.210.18.40	192.168.43.17	UDP	1402	5208 → 48519 Len=1358
302	27.172293227	62.210.18.40	192.168.43.17	UDP	1402	5208 → 48519 Len=1358
303	27.181947227	62.210.18.40	192.168.43.17	UDP	1402	5208 → 48519 Len=1358
304	27.216225214	62.210.18.40	192.168.43.17	UDP	1402	5208 → 48519 Len=1358
305	27.241033431	62.210.18.40	192.168.43.17	UDP	1402	5208 → 48519 Len=1358
306	27.250824722	62.210.18.40	192.168.43.17	UDP	1402	5208 → 48519 Len=1358
307	27.266930759	62.210.18.40	192.168.43.17	UDP	1402	5208 → 48519 Len=1358
308	27.331958815	62.210.18.40	192.168.43.17	UDP	1402	5208 → 48519 Len=1358
309	27.331970140	62.210.18.40	192.168.43.17	UDP	1402	5208 → 48519 Len=1358

< >

> Frame 297: 1402 bytes on wire (11216 bits), 1402 bytes captured (11216 bits) on interface any, id 0

> Linux cooked capture

> Internet Protocol Version 4, Src: 62.210.18.40, Dst: 192.168.43.17

> User Datagram Protocol, Src Port: 5208, Dst Port: 48519

Source Port: 5208

Destination Port: 48519

Length: 1366

Checksum: 0x52f1 [unverified]

[Checksum Status: Unverified]

[Stream index: 7]

```

0000  00 00 00 01 00 06 08 25 25 5b 7e 58 00 00 08 00  ....%[~X...
0010  45 28 05 6a 05 b4 40 00 32 11 00 f4 3e d2 12 28  E(.j..@. 2...>
0020  c0 a8 2b 11 14 58 bd 87 05 56 52 f1 5f 95 6d 3f  ..+..X...VR...m?
0030  00 00 a8 f6 00 00 00 0c bf 7a 4b d2 4a 43 9e d3  .......zK-JC...
0040  55 91 9f aa 58 51 08 06 64 f2 89 96 3b e6 0e d1  U...XQ...d...;...
0050  59 30 8b f2 c4 40 c1 83 ba 0c 56 04 50 f4 d8 a5  Y0...@...V.P...
0060  86 77 4f de c9 57 e4 2d 4a 6d c3 85 53 d1 56 ac  wO..W.-Jm..S.V.
0070  01 e1 9f c5 22 60 49 dc 6c 9f e1 bc 93 b9 62 19  ....I. 1....b.

```

User Datagram Protocol: Protocol

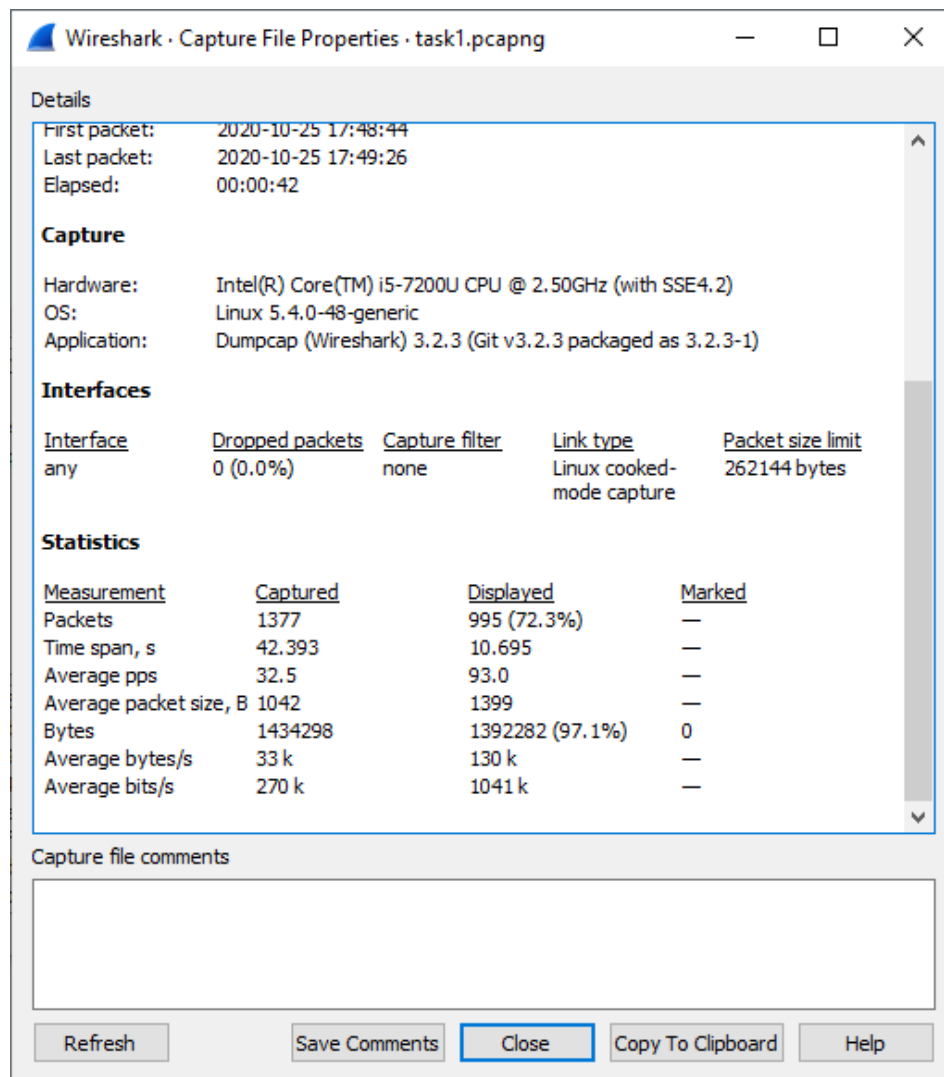
Packets: 1377 · Displayed: 995 (72.3%)

Profile: Default

From the Wireshark capture we can see the length of each packet is 1366 bytes. Since we had 995 packets transferred the total data transferred is $1366 * 995 = 1359170$ bytes.

The time required for this transmission is 10 seconds (defined in the command executed).

Therefore, the throughput is $1359170 / (1024 * 10) = 132.7314453125$ Kb/s



The same throughput of 130 k bytes/s is shown in Capture file properties as well.

$132.7314453125 \text{ Kbyte/s} = 132.7314453125 * 8 / 1024 \text{ Mbits/sec} = 1.03696441650390625 \text{ Mbits/s}$ which is close to the throughput reported by iperf.

```

tokudai@tokudai-msi: /mnt/c/L  × + ▾
(base) tokudai@tokudai-msi:/mnt/c/Users/visha$ iperf3 -u -t 10 -c ping.online.net -p 5208 -R
Connecting to host ping.online.net, port 5208
Reverse mode, remote host ping.online.net is sending
^[[A[ 5] local 192.168.43.17 port 48519 connected to 62.210.18.40 port 5208
[ ID] Interval          Transfer      Bitrate      Jitter      Lost/Total Datagrams
[ 5] 0.00-1.00 sec      128 KBytes    1.05 Mbits/sec 282080451.505 ms 0/92 (0%)
[ 5] 1.00-2.00 sec      128 KBytes    1.05 Mbits/sec 744270.772 ms 0/92 (0%)
[ 5] 2.00-3.00 sec      128 KBytes    1.05 Mbits/sec 1965.365 ms 0/92 (0%)
[ 5] 3.00-4.00 sec      128 KBytes    1.05 Mbits/sec 6.006 ms 0/92 (0%)
[ 5] 4.00-5.00 sec      127 KBytes    1.04 Mbits/sec 1.468 ms 0/91 (0%)
[ 5] 5.00-6.00 sec      128 KBytes    1.05 Mbits/sec 0.531 ms 0/92 (0%)
[ 5] 6.00-7.00 sec      128 KBytes    1.05 Mbits/sec 0.639 ms 0/92 (0%)
[ 5] 7.00-8.00 sec      128 KBytes    1.05 Mbits/sec 1.813 ms 0/92 (0%)
[ 5] 8.00-9.00 sec      128 KBytes    1.05 Mbits/sec 0.569 ms 0/92 (0%)
[ 5] 9.00-10.00 sec     127 KBytes    1.04 Mbits/sec 1.340 ms 0/91 (0%)
- - - - -
[ ID] Interval          Transfer      Bitrate      Jitter      Lost/Total Datagrams
[ 5] 0.00-10.00 sec     1.28 MBytes    1.07 Mbits/sec 0.000 ms 0/918 (0%) sender
[SUM] 0.0-10.0 sec    18 datagrams received out-of-order
[ 5] 0.00-10.00 sec     1.25 MBytes    1.05 Mbits/sec 1.340 ms 0/918 (0%) receiver

iperf Done.
(base) tokudai@tokudai-msi:/mnt/c/Users/visha$ iperf3 -u -t 10 -c ping.online.net -p 5208 -R

```

Task 2:

1. How many TCP packets are exchanged in this communication client and remote server?

We use the filter `ip.addr == 62.210.18.40 && tcp` to filter out the TCP packets transferred between server and client.

For 2mb file, we find that a total of 956 packets are transferred between the client and server.

The image shows a Wireshark packet capture analysis of a file named `task2.2m.pcapng.gz`. The filter applied is `ip.addr == 62.210.18.40 && tcp`. The packet list shows a series of TCP and HTTP packets. The packet details pane for packet 46 (HTTP GET) is expanded, showing the request for `/2Mo.dat`. The packet bytes pane shows the raw data in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Length	Info
33	1.966556	192.168.9.87	62.210.18.40	TCP	66	52626 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK
34	1.967095	192.168.9.87	62.210.18.40	TCP	66	52627 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK
42	2.137480	62.210.18.40	192.168.9.87	TCP	66	80 → 52626 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1440
43	2.137480	62.210.18.40	192.168.9.87	TCP	66	80 → 52627 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1440
44	2.137571	192.168.9.87	62.210.18.40	TCP	54	52626 → 80 [ACK] Seq=1 Ack=1 Win=66048 Len=0
45	2.137624	192.168.9.87	62.210.18.40	TCP	54	52627 → 80 [ACK] Seq=1 Ack=1 Win=66048 Len=0
46	2.137927	192.168.9.87	62.210.18.40	HTTP	542	GET /2Mo.dat HTTP/1.1
47	2.143003	62.210.18.40	192.168.9.87	TCP	54	80 → 52627 [ACK] Seq=1 Ack=489 Win=30720 Len=0
60	2.307063	62.210.18.40	192.168.9.87	TCP	2934	80 → 52627 [ACK] Seq=1 Ack=489 Win=30720 Len=2880 [TCP segm
61	2.307149	192.168.9.87	62.210.18.40	TCP	54	52627 → 80 [ACK] Seq=489 Ack=2881 Win=66048 Len=0
62	2.309947	62.210.18.40	192.168.9.87	TCP	1494	80 → 52627 [ACK] Seq=2881 Ack=489 Win=30720 Len=1440 [TCP se
63	2.309947	62.210.18.40	192.168.9.87	TCP	1494	80 → 52627 [ACK] Seq=4321 Ack=489 Win=30720 Len=1440 [TCP se
64	2.309947	62.210.18.40	192.168.9.87	TCP	1494	80 → 52627 [ACK] Seq=5761 Ack=489 Win=30720 Len=1440 [TCP se
65	2.309947	62.210.18.40	192.168.9.87	TCP	1045	80 → 52627 [PSH, ACK] Seq=7201 Ack=489 Win=30720 Len=991 [T
66	2.309947	62.210.18.40	192.168.9.87	TCP	4374	80 → 52627 [ACK] Seq=8192 Ack=489 Win=30720 Len=4320 [TCP se
67	2.310059	192.168.9.87	62.210.18.40	TCP	54	52627 → 80 [ACK] Seq=489 Ack=12512 Win=66048 Len=0
68	2.324282	62.210.18.40	192.168.9.87	TCP	1494	80 → 52627 [ACK] Seq=12512 Ack=489 Win=30720 Len=1440 [TCP
69	2.324282	62.210.18.40	192.168.9.87	TCP	503	80 → 52627 [PSH, ACK] Seq=13952 Ack=489 Win=30720 Len=449 [
70	2.324583	192.168.9.87	62.210.18.40	TCP	54	52627 → 80 [ACK] Seq=489 Ack=14401 Win=66048 Len=0
82	2.485897	62.210.18.40	192.168.9.87	TCP	1494	80 → 52627 [ACK] Seq=14401 Ack=489 Win=30720 Len=1440 [TCP
83	2.485897	62.210.18.40	192.168.9.87	TCP	1494	80 → 52627 [ACK] Seq=15841 Ack=489 Win=30720 Len=1440 [TCP
84	2.485962	192.168.9.87	62.210.18.40	TCP	54	52627 → 80 [ACK] Seq=489 Ack=17281 Win=66048 Len=0

Frame 46: 542 bytes on wire (4336 bits), 542 bytes captured (4336 bits) on interface \Device\NPF_{69905F9E-06EF-4127-9DBE-D20CCD8FAE4E}, id 0
> Ethernet II, Src: 4e:2c:8d:82:7a:0c (4e:2c:8d:82:7a:0c), Dst: TendaTec_9a:b7:a0 (04:95:e6:9a:b7:a0)
> Internet Protocol Version 4, Src: 192.168.9.87, Dst: 62.210.18.40
> Transmission Control Protocol, Src Port: 52627, Dst Port: 80, Seq: 1, Ack: 1, Len: 488
> Hypertext Transfer Protocol

0000 04 95 e6 9a b7 a0 4e 2c 8d 82 7a 0c 08 00 45 00N, ..z...E
0010 02 10 c1 c4 40 00 00 06 00 00 c0 a8 09 57 3e d2@... ..W>
0020 12 28 cd 93 00 50 e1 54 7e 34 da 0f eb 6b 50 18 .(...P-T ~4...kP
0030 01 02 1c fc 00 00 47 45 54 20 2f 32 4d 6f 2e 64GE T /2Mo,d
0040 61 74 20 48 54 54 50 2f 31 2e 31 0d 0a 48 6f 73 at HTTP/ 1.1..Hos
0050 74 3a 20 70 69 6e 67 2e 6f 6e 6c 69 6e 65 2e 6e t: ping. online.n
0060 65 74 0d 0a 43 6f 6e 6e 65 63 74 69 6f 6e 3a 20 et..Conn ectio:n
0070 6b 65 65 70 2d 61 6c 69 76 65 0d 0a 55 70 67 72 keep-ali ve..Upgr

Transmission Control Protocol: Protocol | Packets: 1101 · Displayed: 956 (86.8%) · Dropped: 0 (0.0%) | Profile: Default

Wireshark · Endpoints · task2.2m.pcapng.gz

Ethernet · 2		IPv4 · 2		IPv6	TCP · 3		UDP
Address	Port	Packets	Bytes	Tx Packets	Tx Bytes	Rx Packets	Rx Bytes
62.210.18.40	80	956	2052 k	700	2038 k	256	14 k
192.168.9.87	52626	3	186	2	120	1	66
192.168.9.87	52627	953	2052 k	254	14 k	699	2038 k

☐ Name resolution
 ☒ Limit to display filter
 Endpoint Types

Copy ▼ Map ▼ Close Help

For 50 mb file, we find that a total of 23296 packets are transferred between the client and server.

task2.50m.pcapng.gz

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

ip.addr == 62.210.18.40 && tcp

No.	Time	Source	Destination	Protocol	Length	Info
24270	49.948872	192.168.9.87	62.210.18.40	TCP	54	52663 → 80 [ACK] Seq=490 Ack=49972390 Win=3432960 Len=0
24271	49.952988	62.210.18.40	192.168.9.87	TCP	4374	80 → 52663 [ACK] Seq=49972390 Ack=490 Win=30720 Len=4320 [T...
24272	49.953042	192.168.9.87	62.210.18.40	TCP	54	52663 → 80 [ACK] Seq=490 Ack=49976710 Win=3432960 Len=0
24273	49.968592	62.210.18.40	192.168.9.87	TCP	1494	80 → 52663 [ACK] Seq=49976710 Ack=490 Win=30720 Len=1440 [T...
24274	49.968592	62.210.18.40	192.168.9.87	TCP	2934	80 → 52663 [ACK] Seq=49978150 Ack=490 Win=30720 Len=2880 [T...
24275	49.968667	192.168.9.87	62.210.18.40	TCP	54	52663 → 80 [ACK] Seq=490 Ack=49981030 Win=3432960 Len=0
24276	49.987516	62.210.18.40	192.168.9.87	TCP	1494	80 → 52663 [ACK] Seq=49981030 Ack=490 Win=30720 Len=1440 [T...
24277	49.987516	62.210.18.40	192.168.9.87	TCP	1494	80 → 52663 [ACK] Seq=49982470 Ack=490 Win=30720 Len=1440 [T...
24278	49.987564	192.168.9.87	62.210.18.40	TCP	54	52663 → 80 [ACK] Seq=490 Ack=49983910 Win=3432960 Len=0
24279	49.990628	62.210.18.40	192.168.9.87	TCP	1494	80 → 52663 [ACK] Seq=49983910 Ack=490 Win=30720 Len=1440 [T...
24280	49.990628	62.210.18.40	192.168.9.87	TCP	1494	80 → 52663 [ACK] Seq=49985350 Ack=490 Win=30720 Len=1440 [T...
24281	49.990653	192.168.9.87	62.210.18.40	TCP	54	52663 → 80 [ACK] Seq=490 Ack=49986790 Win=3432960 Len=0
24282	50.000887	62.210.18.40	192.168.9.87	TCP	1494	80 → 52663 [ACK] Seq=49986790 Ack=490 Win=30720 Len=1440 [T...
24283	50.000887	62.210.18.40	192.168.9.87	TCP	1494	80 → 52663 [ACK] Seq=49988230 Ack=490 Win=30720 Len=1440 [T...
24284	50.000933	192.168.9.87	62.210.18.40	TCP	54	52663 → 80 [ACK] Seq=490 Ack=49989670 Win=3432960 Len=0
24285	50.009592	62.210.18.40	192.168.9.87	TCP	1494	80 → 52663 [ACK] Seq=49989670 Ack=490 Win=30720 Len=1440 [T...
24286	50.009592	62.210.18.40	192.168.9.87	TCP	1494	80 → 52663 [ACK] Seq=49991110 Ack=490 Win=30720 Len=1440 [T...
24287	50.009592	62.210.18.40	192.168.9.87	TCP	1494	80 → 52663 [ACK] Seq=49992550 Ack=490 Win=30720 Len=1440 [T...
24288	50.009644	192.168.9.87	62.210.18.40	TCP	54	52663 → 80 [ACK] Seq=490 Ack=49993990 Win=3432960 Len=0
24289	50.014273	62.210.18.40	192.168.9.87	TCP	5814	80 → 52663 [ACK] Seq=49993990 Ack=490 Win=30720 Len=5760 [T...
24290	50.014424	192.168.9.87	62.210.18.40	TCP	54	52663 → 80 [ACK] Seq=490 Ack=49999750 Win=3432960 Len=0
24291	50.023080	62.210.18.40	192.168.9.87	HTTP	576	HTTP/1.1 200 OK

> Frame 680: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface \Device\NPF_{69905F9E-06EF-4127-9DBE-D20CCD8FAE4E}, id 0
 > Ethernet II, Src: 4e:2c:8d:82:7a:0c (4e:2c:8d:82:7a:0c), Dst: TendaTec_9a:b7:a0 (04:95:e6:9a:b7:a0)
 > Internet Protocol Version 4, Src: 192.168.9.87, Dst: 62.210.18.40
 > Transmission Control Protocol, Src Port: 52663, Dst Port: 80, Seq: 0, Len: 0

```

0000  04 95 e6 9a b7 a0 4e 2c 8d 82 7a 0c 08 00 45 00  ....N, ..z...E.
0010  00 34 c2 d1 40 00 00 06 00 00 c0 a8 09 57 3e d2  .4..@.....W>
0020  12 28 cd b7 00 50 a7 57 64 e9 00 00 00 00 80 02  .(...P.W d.....
0030  fa f0 1b 20 00 00 02 04 05 b4 01 03 03 08 01 01  .. ... ..
0040  04 02
  
```

Transmission Control Protocol: Protocol | Packets: 24304 · Displayed: 23236 (95.6%) | Profile: Default

Wireshark · Endpoints · task2.50m.pcapng.gz

Ethernet · 2 | IPv4 · 2 | IPv6 | TCP · 3 | UDP

Address	Port	Packets	Bytes	Tx Packets	Tx Bytes	Rx Packets	Rx Bytes
62.210.18.40	80	23,236	51 M	15,665	50 M	7,571	428 k
192.168.9.87	52663	23,227	51 M	7,566	428 k	15,661	50 M
192.168.9.87	52665	9	523	5	283	4	240

☐ Name resolution ☒ Limit to display filter Endpoint Types

Copy Map Close Help

2. What is the minimum amount of available buffer space advertised at the client/receiver for the entire trace?
- The minimum buffer space is advertised in the first SYN packet for client side and SYN, ACK packet for the server side.

For both the 2mb file and 50mb file, the buffer size at both server and client is the same.

For the client, the buffer size is 64240 bytes are shown by the window size value in first SYN packet.

The image shows a Wireshark packet capture analysis of a file named `task2.2m.pcapng.gz`. The filter applied is `ip.addr == 62.210.18.40 && tcp`. The packet list shows a series of TCP packets between 192.168.9.87 and 62.210.18.40. Packet 33 is a SYN packet from 192.168.9.87 to 62.210.18.40 with a window size of 64240. The packet details pane shows the following information for packet 33:

- Acknowledgment number (raw): 0
- 1000 = Header Length: 32 bytes (8)
- Flags: 0x002 (SYN)
- Window size value: 64240
- [Calculated window size: 64240]
- Checksum: 0x1b20 [unverified]
- [Checksum Status: Unverified]
- Urgent pointer: 0
- Options: (12 bytes), Maximum segment size, No-Operation (NOP), Window scale, No-Operation (NOP), No-Operation (NOP), SACK permitted
- [Timestamps]

The packet bytes pane shows the raw data of the packet, with the window size value (64240) highlighted in blue. The status bar at the bottom indicates that the window size value from the TCP header (tcp.window_size_value), 2 bytes, is displayed. The status bar also shows that 1101 packets are displayed, representing 956 (86.8%) of the total packets. The profile is set to Default.

task2.50m.pcapng.gz

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

ip.addr == 62.210.18.40 && tcp

No.	Time	Source	Destination	Protocol	Length	Info
689	2.624925	192.168.9.87	62.210.18.40	TCP	66	52663 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK
692	2.669902	192.168.9.87	62.210.18.40	TCP	66	52665 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK
696	2.789591	62.210.18.40	192.168.9.87	TCP	66	80 → 52663 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1440
697	2.789662	192.168.9.87	62.210.18.40	TCP	54	52663 → 80 [ACK] Seq=1 Ack=1 Win=66048 Len=0
698	2.789874	192.168.9.87	62.210.18.40	HTTP	543	GET /50%o.dat HTTP/1.1
699	2.802027	62.210.18.40	192.168.9.87	TCP	54	80 → 52663 [ACK] Seq=1 Ack=490 Win=30720 Len=0
704	2.838429	62.210.18.40	192.168.9.87	TCP	66	80 → 52665 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1440
705	2.838515	192.168.9.87	62.210.18.40	TCP	54	52665 → 80 [ACK] Seq=1 Ack=1 Win=66048 Len=0
710	2.957724	62.210.18.40	192.168.9.87	TCP	5814	80 → 52663 [ACK] Seq=1 Ack=490 Win=30720 Len=5760 [TCP segment
711	2.957832	192.168.9.87	62.210.18.40	TCP	54	52663 → 80 [ACK] Seq=490 Ack=5761 Win=66048 Len=0
712	2.960769	62.210.18.40	192.168.9.87	TCP	1494	80 → 52663 [ACK] Seq=5761 Ack=490 Win=30720 Len=1440 [TCP segment
713	2.960769	62.210.18.40	192.168.9.87	TCP	1045	80 → 52663 [PSH, ACK] Seq=7201 Ack=490 Win=30720 Len=991 [TCP segment
714	2.960769	192.168.9.87	192.168.9.87	TCP	2934	80 → 52663 [ACK] Seq=8192 Ack=490 Win=30720 Len=2880 [TCP segment
715	2.960927	192.168.9.87	62.210.18.40	TCP	54	52663 → 80 [ACK] Seq=490 Ack=11072 Win=66048 Len=0
716	2.962031	62.210.18.40	192.168.9.87	TCP	1494	80 → 52663 [ACK] Seq=11072 Ack=490 Win=30720 Len=1440 [TCP segment
717	2.962031	62.210.18.40	192.168.9.87	TCP	1494	80 → 52663 [ACK] Seq=12512 Ack=490 Win=30720 Len=1440 [TCP segment
718	2.962031	62.210.18.40	192.168.9.87	TCP	503	80 → 52663 [PSH, ACK] Seq=13952 Ack=490 Win=30720 Len=449 [TCP segment
719	2.962134	192.168.9.87	62.210.18.40	TCP	54	52663 → 80 [ACK] Seq=490 Ack=14401 Win=66048 Len=0
730	3.123779	62.210.18.40	192.168.9.87	TCP	1494	80 → 52663 [PSH, ACK] Seq=14401 Ack=490 Win=30720 Len=1440 [TCP segment
738	3.168067	192.168.9.87	62.210.18.40	TCP	54	52663 → 80 [ACK] Seq=490 Ack=15841 Win=66048 Len=0
742	3.283212	62.210.18.40	192.168.9.87	TCP	5814	80 → 52663 [PSH, ACK] Seq=15841 Ack=490 Win=30720 Len=5760 [TCP segment
743	3.283261	192.168.9.87	62.210.18.40	TCP	54	52663 → 80 [ACK] Seq=490 Ack=21601 Win=66048 Len=0

Acknowledgment number (raw): 0
1000 = Header Length: 32 bytes (8)
Flags: 0x002 (SYN)
Window size value: 64240
[Calculated window size: 64240]
Checksum: 0xb20 [unverified]
[Checksum Status: Unverified]
Urgent pointer: 0
Options: (12 bytes), Maximum segment size, No-Operation (NOP), Window scale, No-Operation (NOP), No-Operation (NOP), SACK permitted
[Timestamps]

0000 04 95 e6 9a b7 a0 4e 2c 8d 82 7a 0c 08 00 45 00N, ..z...E
0010 00 34 c2 d1 40 00 80 06 00 00 c0 a8 09 57 3e d2@... ..W..
0020 12 28 cd b7 00 50 a7 57 64 e9 00 00 00 00 02P.W d.....
0030 fa f0 1b 20 00 00 02 04 05 b4 01 03 03 01 015.....
0040 04 02 ..

The window size value from the TCP header (tcp.window_size_value), 2 bytes

Packets: 24304 · Displayed: 23236 (95.6%)

Profile: Default

For the server, the buffer size is 29200 bytes as shown by the window size value in first SYN, ACK packet.

task2.2m.pcapng.gz

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

ip.addr == 62.210.18.40 && tcp

No.	Time	Source	Destination	Protocol	Length	Info
33	1.966556	192.168.9.87	62.210.18.40	TCP	66	52626 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK
34	1.967095	192.168.9.87	62.210.18.40	TCP	66	52627 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK
42	2.137480	62.210.18.40	192.168.9.87	TCP	66	80 → 52626 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1440
43	2.137480	62.210.18.40	192.168.9.87	TCP	66	80 → 52627 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1440
44	2.137571	192.168.9.87	62.210.18.40	TCP	54	52626 → 80 [ACK] Seq=1 Ack=1 Win=66048 Len=0
45	2.137624	192.168.9.87	62.210.18.40	TCP	54	52627 → 80 [ACK] Seq=1 Ack=1 Win=66048 Len=0
46	2.137927	192.168.9.87	62.210.18.40	HTTP	542	GET /2%o.dat HTTP/1.1
47	2.143003	62.210.18.40	192.168.9.87	TCP	54	80 → 52627 [ACK] Seq=1 Ack=489 Win=30720 Len=0
60	2.307063	62.210.18.40	192.168.9.87	TCP	2934	80 → 52627 [ACK] Seq=1 Ack=489 Win=30720 Len=2880 [TCP segment
61	2.307149	192.168.9.87	62.210.18.40	TCP	54	52627 → 80 [ACK] Seq=489 Ack=2881 Win=66048 Len=0
62	2.309947	62.210.18.40	192.168.9.87	TCP	1494	80 → 52627 [ACK] Seq=2881 Ack=489 Win=30720 Len=1440 [TCP segment
63	2.309947	62.210.18.40	192.168.9.87	TCP	1494	80 → 52627 [ACK] Seq=4321 Ack=489 Win=30720 Len=1440 [TCP segment
64	2.309947	62.210.18.40	192.168.9.87	TCP	1494	80 → 52627 [ACK] Seq=5761 Ack=489 Win=30720 Len=1440 [TCP segment
65	2.309947	62.210.18.40	192.168.9.87	TCP	1045	80 → 52627 [PSH, ACK] Seq=7201 Ack=489 Win=30720 Len=991 [TCP segment
66	2.309947	62.210.18.40	192.168.9.87	TCP	4374	80 → 52627 [ACK] Seq=8192 Ack=489 Win=30720 Len=4320 [TCP segment
67	2.310059	192.168.9.87	62.210.18.40	TCP	54	52627 → 80 [ACK] Seq=489 Ack=12512 Win=66048 Len=0
68	2.324282	62.210.18.40	192.168.9.87	TCP	1494	80 → 52627 [ACK] Seq=12512 Ack=489 Win=30720 Len=1440 [TCP segment
69	2.324282	62.210.18.40	192.168.9.87	TCP	503	80 → 52627 [PSH, ACK] Seq=13952 Ack=489 Win=30720 Len=449 [TCP segment
70	2.324583	192.168.9.87	62.210.18.40	TCP	54	52627 → 80 [ACK] Seq=489 Ack=14401 Win=66048 Len=0
82	2.485897	62.210.18.40	192.168.9.87	TCP	1494	80 → 52627 [ACK] Seq=14401 Ack=489 Win=30720 Len=1440 [TCP segment
83	2.485897	62.210.18.40	192.168.9.87	TCP	1494	80 → 52627 [ACK] Seq=15841 Ack=489 Win=30720 Len=1440 [TCP segment
84	2.485962	192.168.9.87	62.210.18.40	TCP	54	52627 → 80 [ACK] Seq=489 Ack=17281 Win=66048 Len=0
85	2.486133	62.210.18.40	192.168.9.87	TCP	1494	80 → 52627 [ACK] Seq=17281 Ack=489 Win=30720 Len=1440 [TCP segment

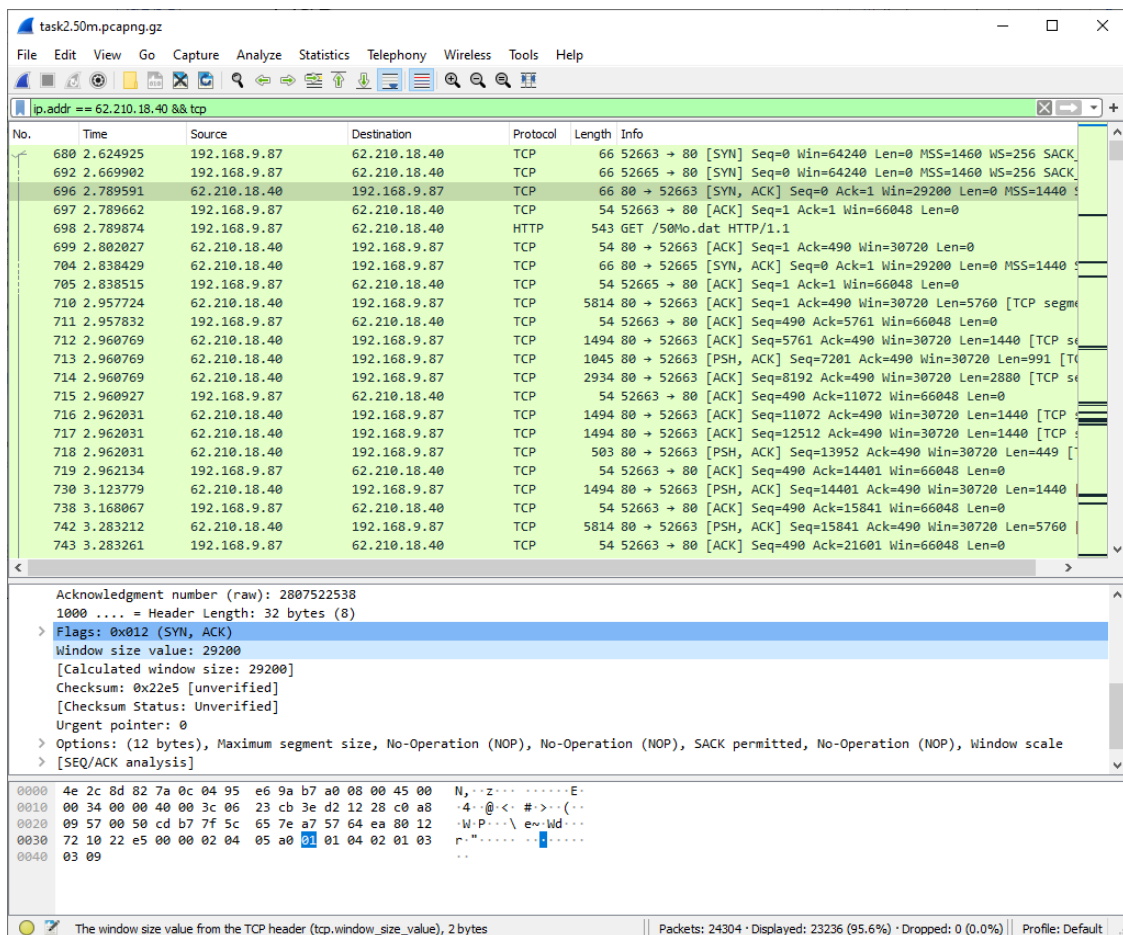
Acknowledgment number (raw): 1760766997
1000 = Header Length: 32 bytes (8)
Flags: 0x012 (SYN, ACK)
Window size value: 29200
[Calculated window size: 29200]
Checksum: 0x5757 [unverified]
[Checksum Status: Unverified]
Urgent pointer: 0
Options: (12 bytes), Maximum segment size, No-Operation (NOP), No-Operation (NOP), SACK permitted, No-Operation (NOP), Window scale
[SEQ/ACK analysis]

0000 4e 2c 8d 82 7a 0c 04 95 e6 9a b7 a0 08 00 45 00N, ..z... ..E
0010 00 34 00 00 40 00 3c 06 23 cb 3e d2 12 28 c0 a8@... ..W..
0020 09 57 00 50 fd 92 10 86 17 41 68 f3 2c 15 00 12P... ..Ah,..
0030 72 10 57 57 00 00 02 04 05 a0 01 01 04 02 01 03W..
0040 03 09 ..

The window size value from the TCP header (tcp.window_size_value), 2 bytes

Packets: 1101 · Displayed: 956 (86.8%)

Profile: Default



3. Pick any 5 TCP segments from server to client which are not part of initial TCP connection establishment and final connection termination.

1. Make a table listing for each of these segments, the length of each of these TCP segments, the sequence number, time when the segment was sent, time when the respective ACK for each segment was received, length of the respective ACK segment. Place the screenshot of Wireshark of at least one such segment with respective ACK as a proof of observation and calculation. What is the maximum length out of all?

For the 2m file,

Segment no	Length of segment	Sequence no (relative)	Time when segment sent	Time of ACK	Length of ACK
315	1494	326881	3.112960	3.113096	54
324	1494	338401	3.123928	3.124020	54
326	11574	339841	3.128035	3.128035	54
335	1494	362881	3.139622	3.139699	54
338	1494	364321	3.160456	3.160504	54

task2.2m.pcapng.gz

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp.stream eq 6

No.	Time	Source	Destination	Protocol	Length	Info
316	3.113036	192.168.9.87	62.210.18.40	TCP	54	52627 → 80 [ACK] Seq=489 Ack=328321 Win=532736 Len=0
317	3.123928	62.210.18.40	192.168.9.87	TCP	1494	80 → 52627 [ACK] Seq=328321 Ack=489 Win=30720 Len=1440 [TCP
318	3.123928	62.210.18.40	192.168.9.87	TCP	1494	80 → 52627 [ACK] Seq=329761 Ack=489 Win=30720 Len=1440 [TCP
319	3.123928	62.210.18.40	192.168.9.87	TCP	1494	80 → 52627 [ACK] Seq=331201 Ack=489 Win=30720 Len=1440 [TCP
320	3.123928	62.210.18.40	192.168.9.87	TCP	1494	80 → 52627 [ACK] Seq=332641 Ack=489 Win=30720 Len=1440 [TCP
321	3.123928	62.210.18.40	192.168.9.87	TCP	1494	80 → 52627 [ACK] Seq=334081 Ack=489 Win=30720 Len=1440 [TCP
322	3.123928	62.210.18.40	192.168.9.87	TCP	1494	80 → 52627 [ACK] Seq=335521 Ack=489 Win=30720 Len=1440 [TCP
323	3.123928	62.210.18.40	192.168.9.87	TCP	1494	80 → 52627 [ACK] Seq=336961 Ack=489 Win=30720 Len=1440 [TCP
324	3.123928	62.210.18.40	192.168.9.87	TCP	1494	80 → 52627 [ACK] Seq=338401 Ack=489 Win=30720 Len=1440 [TCP
325	3.124020	192.168.9.87	62.210.18.40	TCP	54	52627 → 80 [ACK] Seq=489 Ack=339841 Win=532736 Len=0
326	3.127972	62.210.18.40	192.168.9.87	TCP	11574	80 → 52627 [ACK] Seq=339841 Ack=489 Win=30720 Len=11520 [TCP
327	3.128035	192.168.9.87	62.210.18.40	TCP	54	52627 → 80 [ACK] Seq=489 Ack=351361 Win=532736 Len=0
328	3.139622	62.210.18.40	192.168.9.87	TCP	1494	80 → 52627 [ACK] Seq=351361 Ack=489 Win=30720 Len=1440 [TCP
329	3.139622	62.210.18.40	192.168.9.87	TCP	1494	80 → 52627 [PSH, ACK] Seq=352801 Ack=489 Win=30720 Len=1440 [TCP
330	3.139622	62.210.18.40	192.168.9.87	TCP	1494	80 → 52627 [ACK] Seq=354241 Ack=489 Win=30720 Len=1440 [TCP
331	3.139622	62.210.18.40	192.168.9.87	TCP	1494	80 → 52627 [ACK] Seq=355681 Ack=489 Win=30720 Len=1440 [TCP
332	3.139622	62.210.18.40	192.168.9.87	TCP	1494	80 → 52627 [ACK] Seq=357121 Ack=489 Win=30720 Len=1440 [TCP
333	3.139622	62.210.18.40	192.168.9.87	TCP	1494	80 → 52627 [ACK] Seq=358561 Ack=489 Win=30720 Len=1440 [TCP
334	3.139622	62.210.18.40	192.168.9.87	TCP	1494	80 → 52627 [ACK] Seq=360001 Ack=489 Win=30720 Len=1440 [TCP
335	3.139622	62.210.18.40	192.168.9.87	TCP	1494	80 → 52627 [ACK] Seq=361441 Ack=489 Win=30720 Len=1440 [TCP
336	3.139699	192.168.9.87	62.210.18.40	TCP	54	52627 → 80 [ACK] Seq=489 Ack=362881 Win=532736 Len=0
337	3.160456	62.210.18.40	192.168.9.87	TCP	1494	80 → 52627 [ACK] Seq=362881 Ack=489 Win=30720 Len=1440 [TCP

Sequence number (raw): 3658814795
 [Next sequence number: 339841 (relative sequence number)]
 Acknowledgment number: 489 (relative ack number)
 Acknowledgment number (raw): 3780411420
 0101 = Header Length: 20 bytes (5)
 > Flags: 0x010 (ACK)
 Window size value: 60
 [Calculated window size: 30720]
 [Window size scaling factor: 512]

0030 00 3c 73 ab 00 00 48 06 70 5b 13 4e 60 fb ae a7 ...s...H...p[.N'...
 0040 df b8 f7 a4 31 a8 5f b4 e7 25 e5 c6 ae ac 48 47 ...1...%...HG
 0050 a7 fc 5c df f3 3b 9f c4 ec 9b 02 f5 fa 37 b8 cd ...;...7...
 0060 82 c4 e2 a9 5f 7f 92 13 2f f4 db c5 76 1d e2 41 .../...v...A
 0070 94 9f 11 5b dd 00 fa a5 bf c1 79 2f cf a7 8e 53 ...[...y/...S
 0080 e3 fc 0b 55 23 6e 8d 6c 7f 22 2e 20 e0 7e 17 72 ...U#n-1 "...r
 0090 3a 82 4e ed 2d 52 46 a3 1b 49 1d 9a 28 53 e7 44 ...N-RF...I...(S-D
 00a0 9d 88 33 d5 90 74 02 f4 ff fe dc b9 e9 d6 98 77 ...3...t...w

The window size scaling factor (-1 when unknown, -2 when no scaling is used) (tcp.window_size_scalingfactor), 2 bytes | Packets: 1101 · Displayed: 953 (86.6%) | Profile: Default

Wireshark · Packet 324 · task2.2m.pcapng.gz

> Frame 324: 1494 bytes on wire (11952 bits), 1494 bytes captured (11952 bits) on interface \Device\NPF_{6990...}

> Ethernet II, Src: TendaTec_9a:b7:a0 (04:95:e6:9a:b7:a0), Dst: 4e:2c:8d:82:7a:0c (4e:2c:8d:82:7a:0c)

> Internet Protocol Version 4, Src: 62.210.18.40, Dst: 192.168.9.87

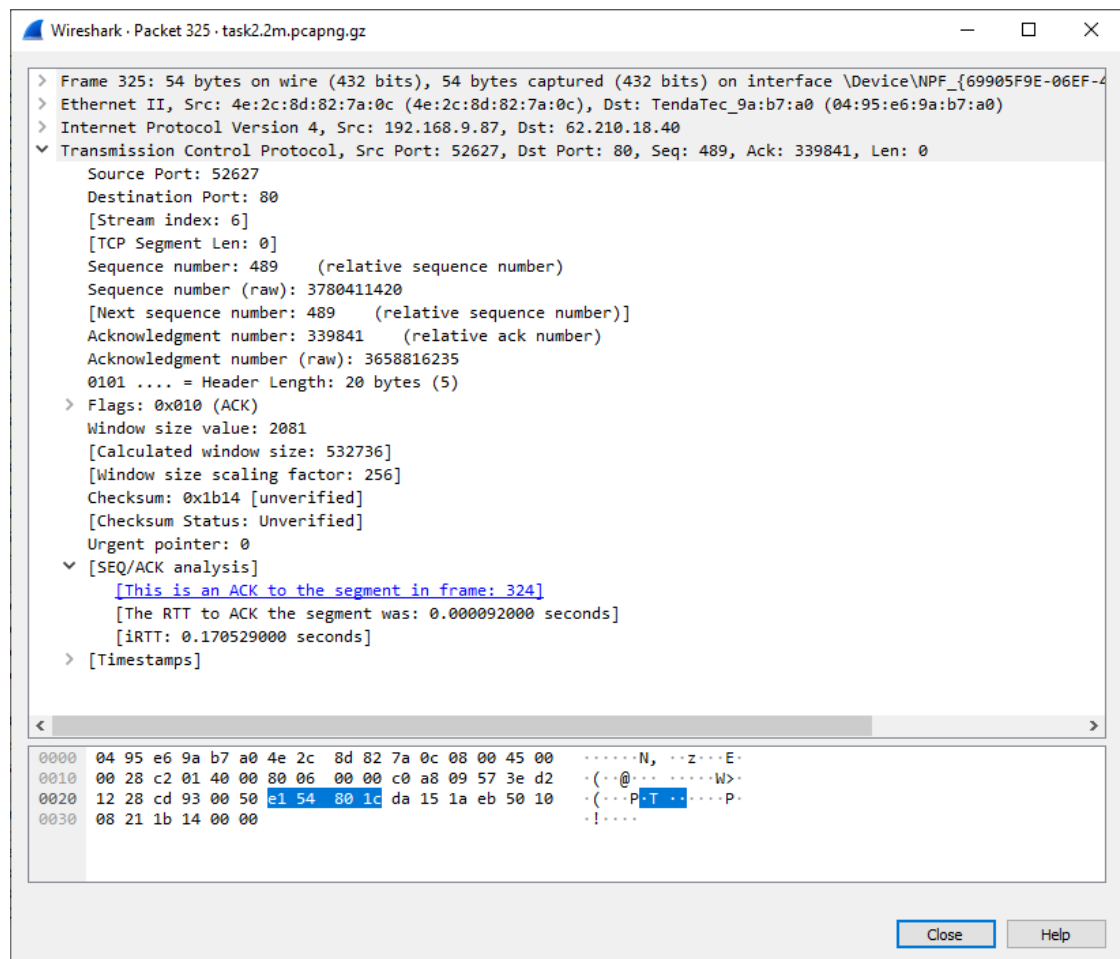
> Transmission Control Protocol, Src Port: 80, Dst Port: 52627, Seq: 338401, Ack: 489, Len: 1440

Source Port: 80
 Destination Port: 52627
 [Stream index: 6]
 [TCP Segment Len: 1440]
 Sequence number: 338401 (relative sequence number)
 Sequence number (raw): 3658814795
 [Next sequence number: 339841 (relative sequence number)]
 Acknowledgment number: 489 (relative ack number)
 Acknowledgment number (raw): 3780411420
 0101 = Header Length: 20 bytes (5)
 > Flags: 0x010 (ACK)

0000 4e 2c 8d 82 7a 0c 04 95 e6 9a b7 a0 08 00 45 00 N...z...E...
 0010 05 c8 e9 5f 40 00 3c 06 34 d7 3e d2 12 28 c0 a8 ...@< 4>...(
 0020 09 57 00 50 cd 93 da 15 15 4b e1 54 80 1c 50 10 ...W.P...K.T.P.
 0030 00 3c 73 ab 00 00 48 06 70 5b 13 4e 60 fb ae a7 ...s...H...p[.N'...
 0040 df b8 f7 a4 31 a8 5f b4 e7 25 e5 c6 ae ac 48 47 ...1...%...HG
 0050 a7 fc 5c df f3 3b 9f c4 ec 9b 02 f5 fa 37 b8 cd ...;...7...
 0060 82 c4 e2 a9 5f 7f 92 13 2f f4 db c5 76 1d e2 41 .../...v...A
 0070 94 9f 11 5b dd 00 fa a5 bf c1 79 2f cf a7 8e 53 ...[...y/...S
 0080 e3 fc 0b 55 23 6e 8d 6c 7f 22 2e 20 e0 7e 17 72 ...U#n-1 "...r
 0090 3a 82 4e ed 2d 52 46 a3 1b 49 1d 9a 28 53 e7 44 ...N-RF...I...(S-D
 00a0 9d 88 33 d5 90 74 02 f4 ff fe dc b9 e9 d6 98 77 ...3...t...w
 00b0 fd 01 66 ed 3f 76 df 49 37 00 fd 48 4c ed aa 27 ...f?v-I 7...HL...
 00c0 02 18 96 da 29 c2 b2 46 3b a9 54 e6 a6 28 7a 53 ...)...F ;.T...(zS
 00d0 5f 53 93 b8 67 e3 26 26 59 7a a8 a3 55 64 3d 2f ...S.g.&& Yz..Ud=/
 00e0 0a 83 e4 c8 f9 2f 9a cc e4 38 0b 53 8e c7 79 20 .../...8.S.y
 00f0 68 e1 28 27 74 f1 cf 9e d1 b4 1c d1 83 d5 7f d5 ...h('t...
 0100 77 1f de 44 87 8e 4c af 97 84 71 ac 02 c7 a0 70 ...w.D.O.L...q...p
 0110 c4 d3 24 6c 7e 5b d7 48 ea b3 f3 90 f7 38 2f f7 ...\$1...[.H ...8/...
 0120 56 c4 55 93 b8 b8 40 ab 30 d0 71 00 47 1b b7 3f ...V.U.8:@ 0-q.G-?
 0130 3f ff 58 be fd ef 83 95 5f e9 5b 40 93 2b 30 05 ...?X... ..[...+0...
 0140 d0 5a ac d3 e2 de 8a c5 88 b4 67 af 4f d0 d9 53 ...Z... ..g.O..S

Bytes 54-1493: TCP segment data (tcp.segment_data)

Close Help



For the 50m file,

Segment no	Length of segment	Sequence no	Time when segment sent	Time of ACK	Length of ACK
8422	5814	17342464	14.954143	14.954160	54
8426	5814	17348320	14.973745	14.973796	54
8428	2934	17354080	14.975647	14.975697	54
8432	597	17364160	14.979039	14.979114	54
8441	1494	17389183	14.984490	14.984552	54

task2.50m.pcapng.gz

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

ip.addr == 62.210.18.40 && tcp

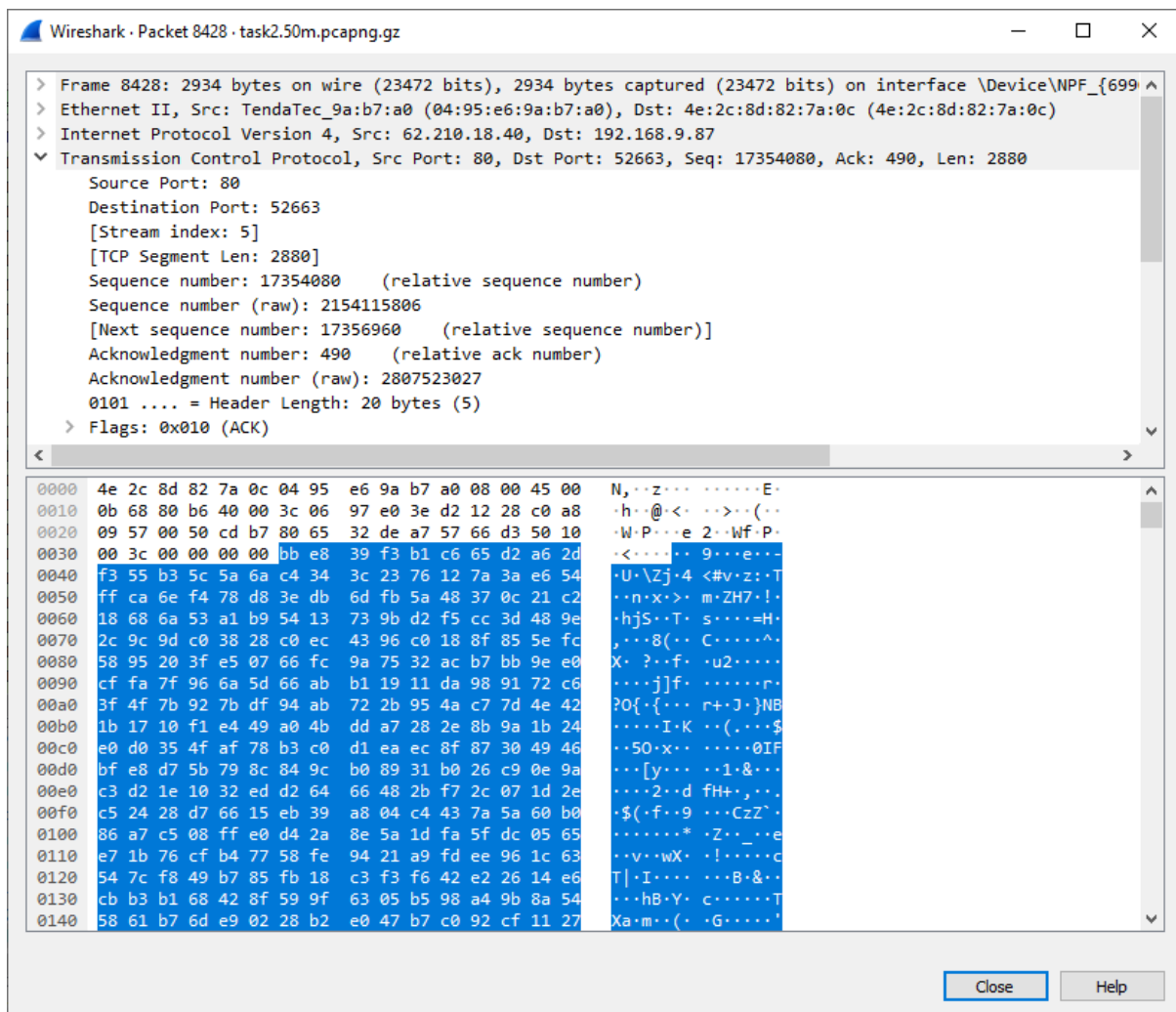
No.	Time	Source	Destination	Protocol	Length	Info
8416	14.948399	62.210.18.40	192.168.9.87	TCP	7254	80 → 52663 [ACK] Seq=17322304 Ack=490 Win=30720 Len=7200 [T...
8417	14.948431	192.168.9.87	62.210.18.40	TCP	66	52663 → 80 [ACK] Seq=490 Ack=17319424 Win=3432960 Len=0 SLE...
8418	14.948453	192.168.9.87	62.210.18.40	TCP	54	52663 → 80 [ACK] Seq=490 Ack=17322304 Win=3432960 Len=0
8419	14.948467	192.168.9.87	62.210.18.40	TCP	54	52663 → 80 [ACK] Seq=490 Ack=17329504 Win=3432960 Len=0
8420	14.950669	62.210.18.40	192.168.9.87	TCP	13014	80 → 52663 [ACK] Seq=17329504 Ack=490 Win=30720 Len=12960 [T...
8421	14.950688	192.168.9.87	62.210.18.40	TCP	54	52663 → 80 [ACK] Seq=490 Ack=17342464 Win=3432960 Len=0
8422	14.954143	62.210.18.40	192.168.9.87	TCP	5814	80 → 52663 [ACK] Seq=17342464 Ack=490 Win=30720 Len=5760 [T...
8423	14.954160	192.168.9.87	62.210.18.40	TCP	54	52663 → 80 [ACK] Seq=490 Ack=17348224 Win=3432960 Len=0
8424	14.963322	62.210.18.40	192.168.9.87	TCP	150	80 → 52663 [PSH, ACK] Seq=17348224 Ack=490 Win=30720 Len=96
8425	14.963366	192.168.9.87	62.210.18.40	TCP	54	52663 → 80 [ACK] Seq=490 Ack=17348320 Win=3432704 Len=0
8426	14.973745	62.210.18.40	192.168.9.87	TCP	5814	80 → 52663 [ACK] Seq=17348320 Ack=490 Win=30720 Len=5760 [T...
8427	14.973796	192.168.9.87	62.210.18.40	TCP	54	52663 → 80 [ACK] Seq=490 Ack=17354080 Win=3432960 Len=0
8428	14.975647	62.210.18.40	192.168.9.87	TCP	2934	80 → 52663 [ACK] Seq=17354080 Ack=490 Win=30720 Len=2880 [T...
8429	14.975697	192.168.9.87	62.210.18.40	TCP	54	52663 → 80 [ACK] Seq=490 Ack=17356960 Win=3432960 Len=0
8430	14.979039	62.210.18.40	192.168.9.87	TCP	1494	80 → 52663 [ACK] Seq=17356960 Ack=490 Win=30720 Len=1440 [T...
8431	14.979039	62.210.18.40	192.168.9.87	TCP	5814	80 → 52663 [ACK] Seq=17358400 Ack=490 Win=30720 Len=5760 [T...
8432	14.979039	62.210.18.40	192.168.9.87	TCP	597	80 → 52663 [PSH, ACK] Seq=17364160 Ack=490 Win=30720 Len=54
8433	14.979114	192.168.9.87	62.210.18.40	TCP	54	52663 → 80 [ACK] Seq=490 Ack=17364703 Win=3432960 Len=0
8434	14.984490	62.210.18.40	192.168.9.87	TCP	1494	80 → 52663 [ACK] Seq=17364703 Ack=490 Win=30720 Len=1440 [T...
8435	14.984490	62.210.18.40	192.168.9.87	TCP	14454	80 → 52663 [ACK] Seq=17366143 Ack=490 Win=30720 Len=14400 [T...
8436	14.984490	62.210.18.40	192.168.9.87	TCP	1494	80 → 52663 [ACK] Seq=17380543 Ack=490 Win=30720 Len=1440 [T...
8437	14.984490	62.210.18.40	192.168.9.87	TCP	1494	80 → 52663 [ACK] Seq=17381983 Ack=490 Win=30720 Len=1440 [T...

[TCP Segment Len: 2880]
Sequence number: 17354080 (relative sequence number)
Sequence number (raw): 2154115806
[Next sequence number: 17356960 (relative sequence number)]
Acknowledgment number: 490 (relative ack number)
Acknowledgment number (raw): 2807523027
0101 = Header Length: 20 bytes (5)
> Flags: 0x010 (ACK)
Window size value: 60

0030 00 3c 00 00 00 00 bb e8 39 f3 b1 c6 65 d2 a6 2d ... 9...e...
0040 f3 55 b3 5c 5a 6a c4 34 3c 23 76 12 7a 3a e6 54 ...U\Zj.4 <#v.z:~T
0050 ff ca 6e f4 78 d8 3e db 6d fb 5a 48 37 0c 21 c2 ...n~x> m~ZH7.!~
0060 18 68 6a 53 a1 b9 54 13 73 9b d2 f5 cc 3d 48 9e ...hJS~T s~...=H~
0070 2c 9c 9d c0 38 28 c0 ec 43 96 c0 18 8f 85 5e fc ...8(C~...^~
0080 58 95 20 3f e5 07 66 fc 9a 75 32 ac b7 bb 9e e0 ...X~?~f~u2~...~
0090 cf fa 7f 96 6a 5d 66 ab b1 19 11 da 98 91 72 c6 ...~j]f~...~r~
00a0 3f 4f 7b 92 7b df 94 ab 72 2b 95 4a c7 7d 4e 42 ...?Q{~...r+J~}NB

The window size value from the TCP header (tcp.window_size_value), 2 bytes

Packets: 24304 · Displayed: 24277 (99.9%) · Dropped: 0 (0.0%) Profile: Default



2. Given the difference between when each TCP segment was sent, and when its acknowledgement was received, what is the RTT value for each of these segments? What is the *EstimatedRTT* value 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 for all subsequent segments.

Place these calculated values appropriately in the table formed in #b above. $EstimatedRTT = (1 - \alpha) \times EstimatedRTT + \alpha \times SampleRTT$ where $\alpha = 0.125$ (that is, $1/8$) [RFC 6298]

For 2m capture,

Sequence no.	RTT	Estimated RTT
315	0.000136	0.000136
324	0.000092	0.000130
326	0.000063	0.000122
335	0.000077	0.000116
338	0.000048	0.000108

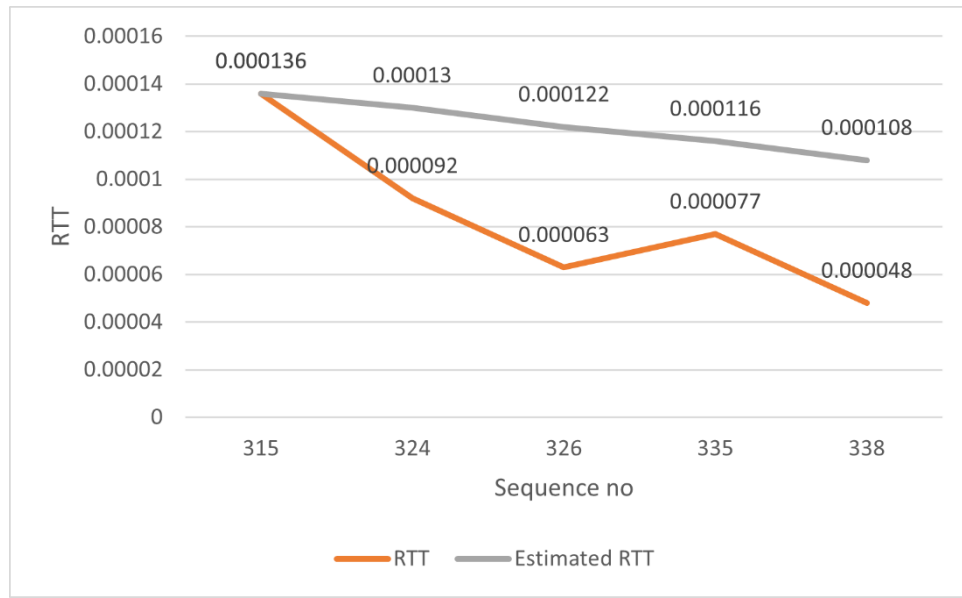
For 50m capture,

Sequence no.	RTT	Estimated RTT
8422	0.000017	0.000017

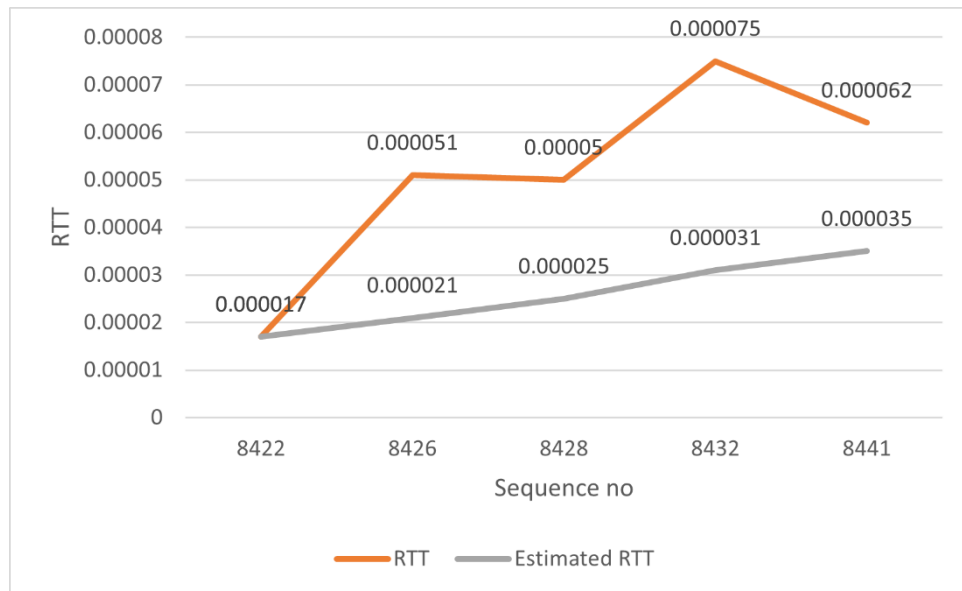
8426	0.000051	0.000021
8428	0.000050	0.000025
8432	0.000075	0.000031
8441	0.000062	0.000035

3. Plot the RTT Graph for any TCP segment out of these using the graph feature of Wireshark. Plot another graph manually from the table above for Sample RTT and estimated RTT.

For 2m capture,



For 50m capture,



4. Comment on your understanding of Estimated RTT calculation and plotted RTT graphs.

We find that RTT time is increasing/ decreasing which may show increasing / decreasing congestion. This may be due to network congestion or CPU bottlenecks where the system may be busy with other tasks and as such not able to provide response immediately.

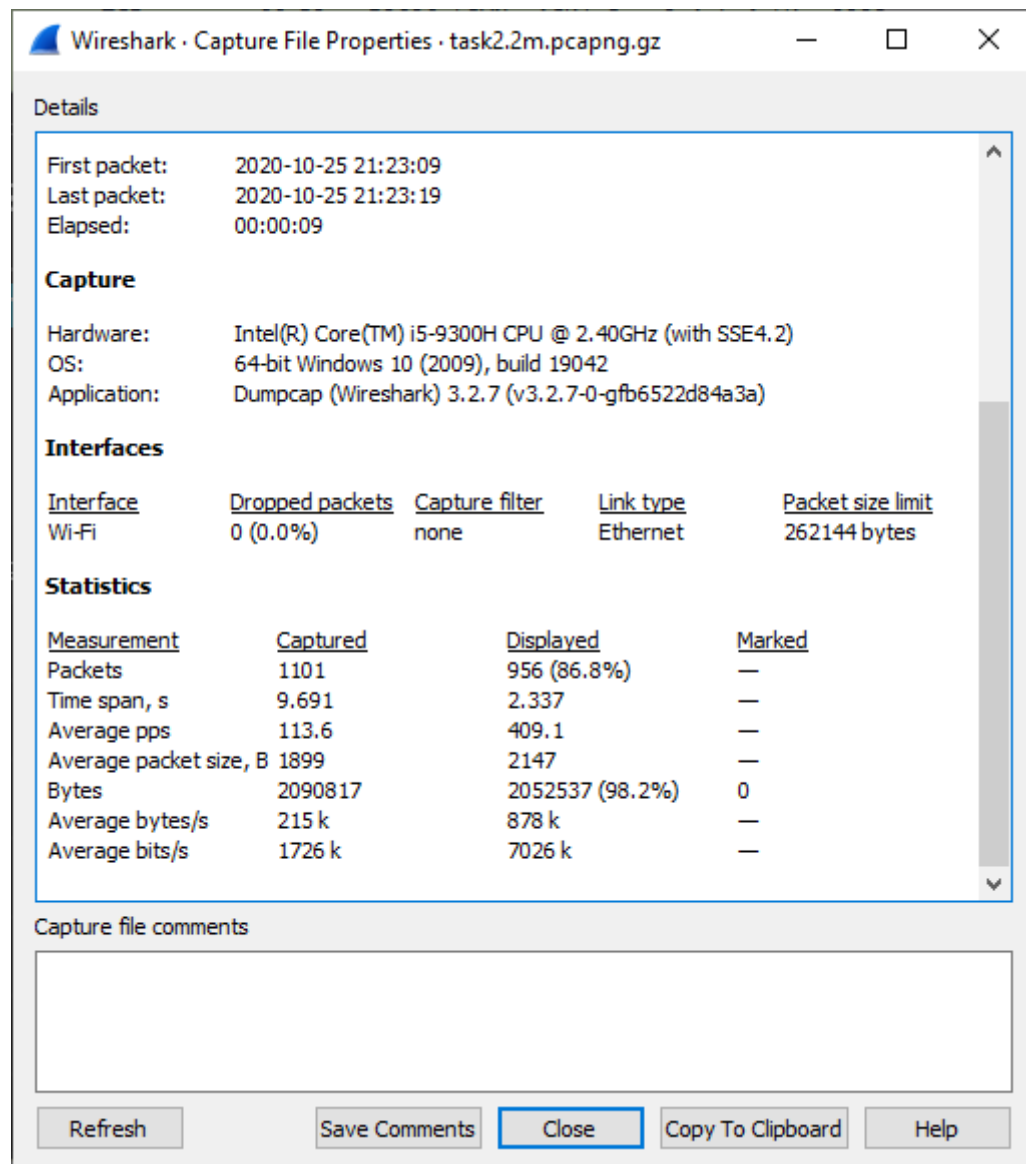
4. Calculate the overall throughput (bytes transferred per unit time) for this TCP conversation using different fields of TCP from the captured file. Explain how you calculated this value using Wireshark capture in this experiment along with relevant screenshots. Verify your calculation with the one done by Wireshark using "Capture File properties". If you observe the major difference in your calculation and one calculated by Wireshark, comment why and how ?

The total data transmitted can be computed by the difference between the sequence number of first TCP segment and last segment divided by the time difference between those segments.

For 2m file, first sequence number is 1 and last segment is 1998721. The data is $1998721 - 1 = 1998720$ bytes.

The transmission time is the time difference of time instant of first TCP segment and last ACK segment. The total time is $4.303398 - 2.307063 = 1.996335$ seconds.

Therefore, the throughput for the TCP transmission is $1998720 / 1.996335 = 10,01,194.6892680837634966075333048$ bytes / second = 977.729 Kbytes / second.

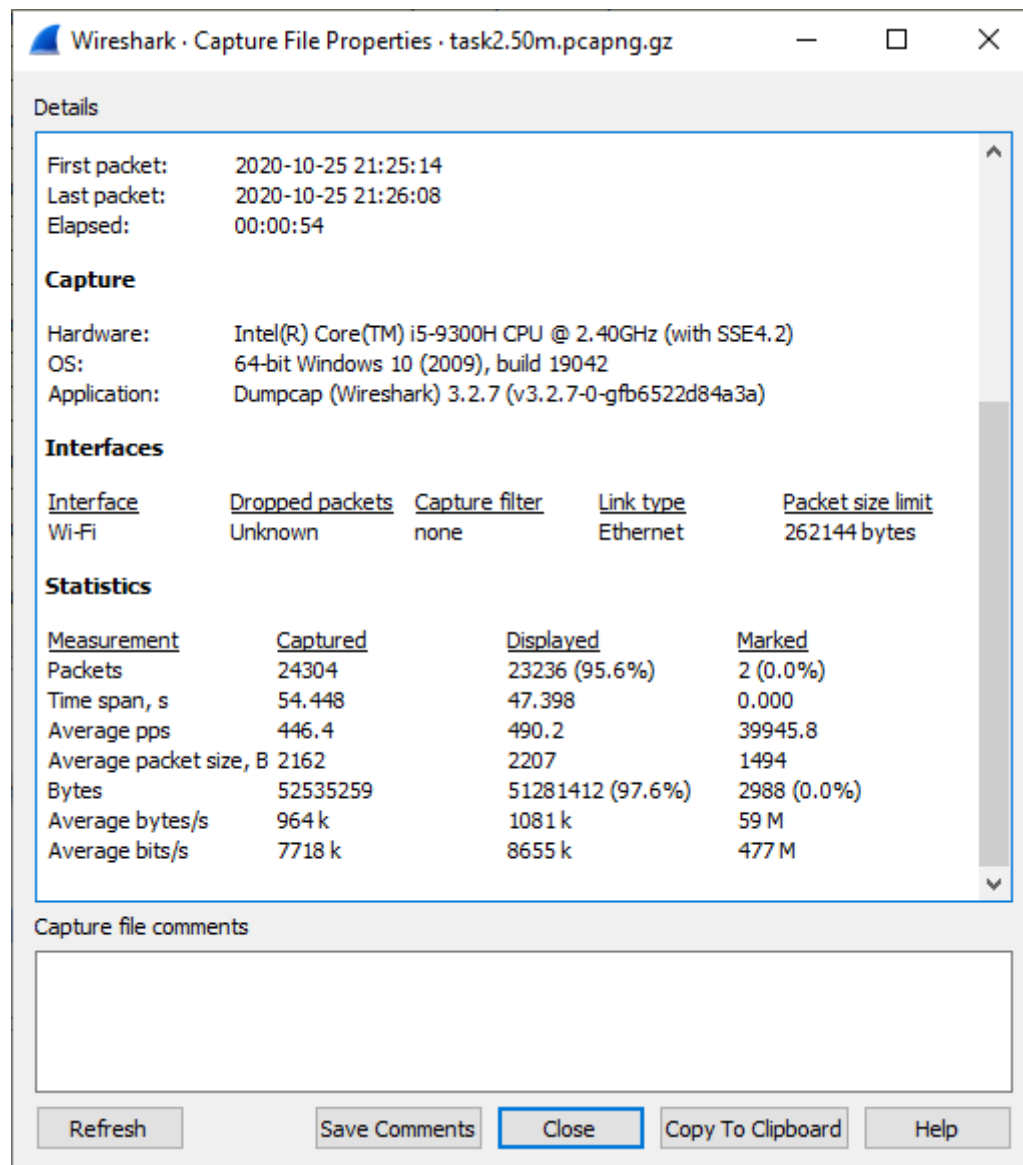


Comparing this to wireshark file properties, the reported value is close to the calculated value.

For 50m file, first sequence number is 1 and last segment is 49993990. The data is $49993990 - 1 = 49993989$ bytes.

The transmission time is the time difference of time instant of first TCP segment and last ACK segment. The total time is $50.014273 - 2.957724 = 47.056549$ seconds.

Therefore, the throughput for the TCP transmission is $49993989 / 47.056549 = 10,62,423.6171675062699561754942973$ bytes / second = 1,037.523 Kbytes / second.

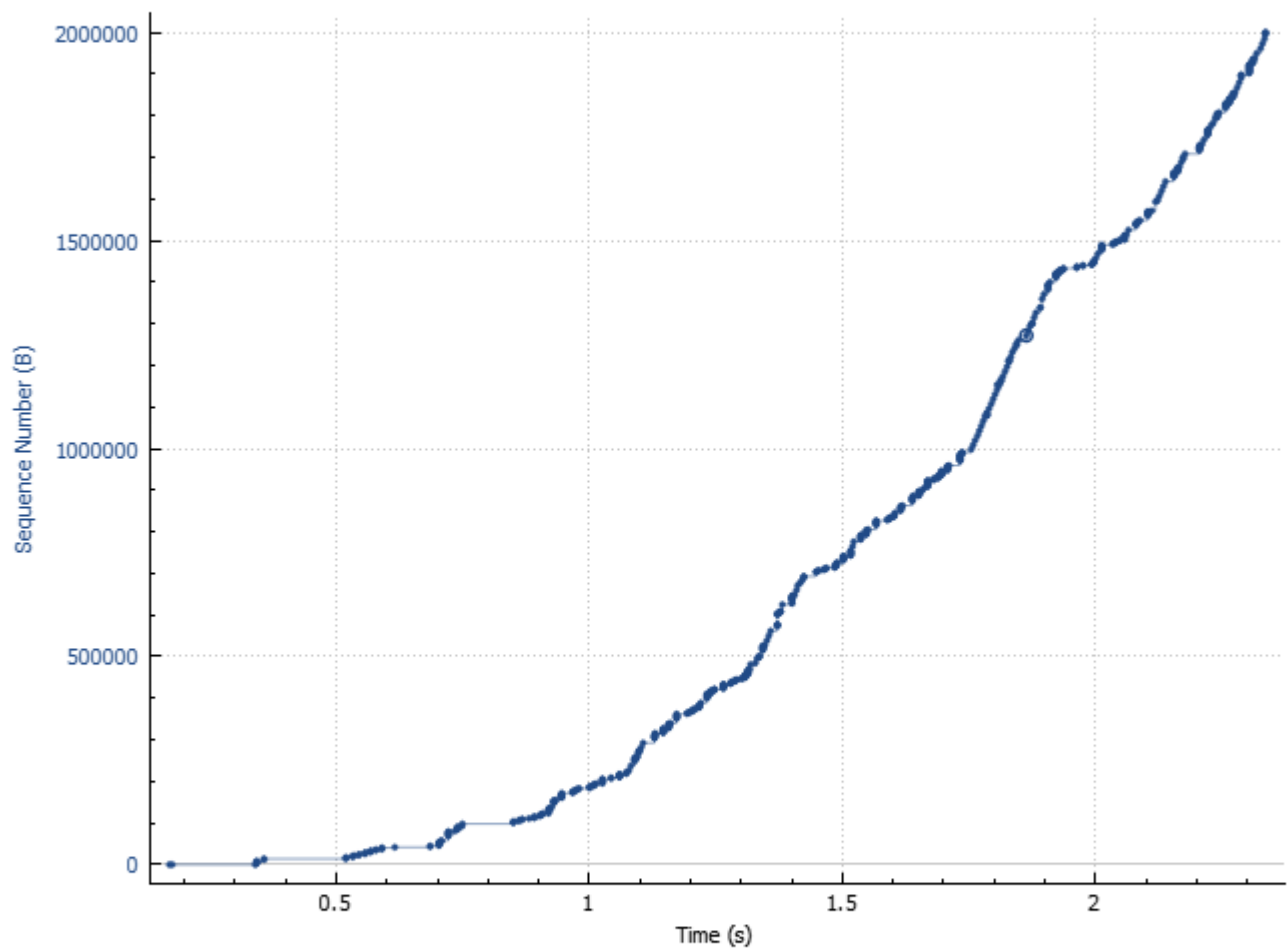


Comparing this to wireshark file properties, the reported value is close to the calculated value.

- Using any active TCP segment (pick the packet of bulk data length, e.g: 5668) involved in the download process from server to client, capture the TCP's functioning using the Time-Sequence-Graph (Stevens) plotting tool to view the sequence number versus time plot of segments being sent from the server to the client. Can you identify where TCP's slow start phase begins and ends, and where congestion avoidance takes over? If not possible, why?

Sequence Numbers (Stevens) for 62.210.18.40:80 → 192.168.9.87:52627

task2.2m.pcapng.gz



Click to select packet 761 (3.831s len 11520 seq 1271521 ack 489 win 30720) → 699 pkts, 2000 kB ← 254 pkts, 488 bytes

Type Time / Sequence (Stevens) ▾

Stream 6 ▾

Switch Direction

Mouse ☒ drags ☐ zooms

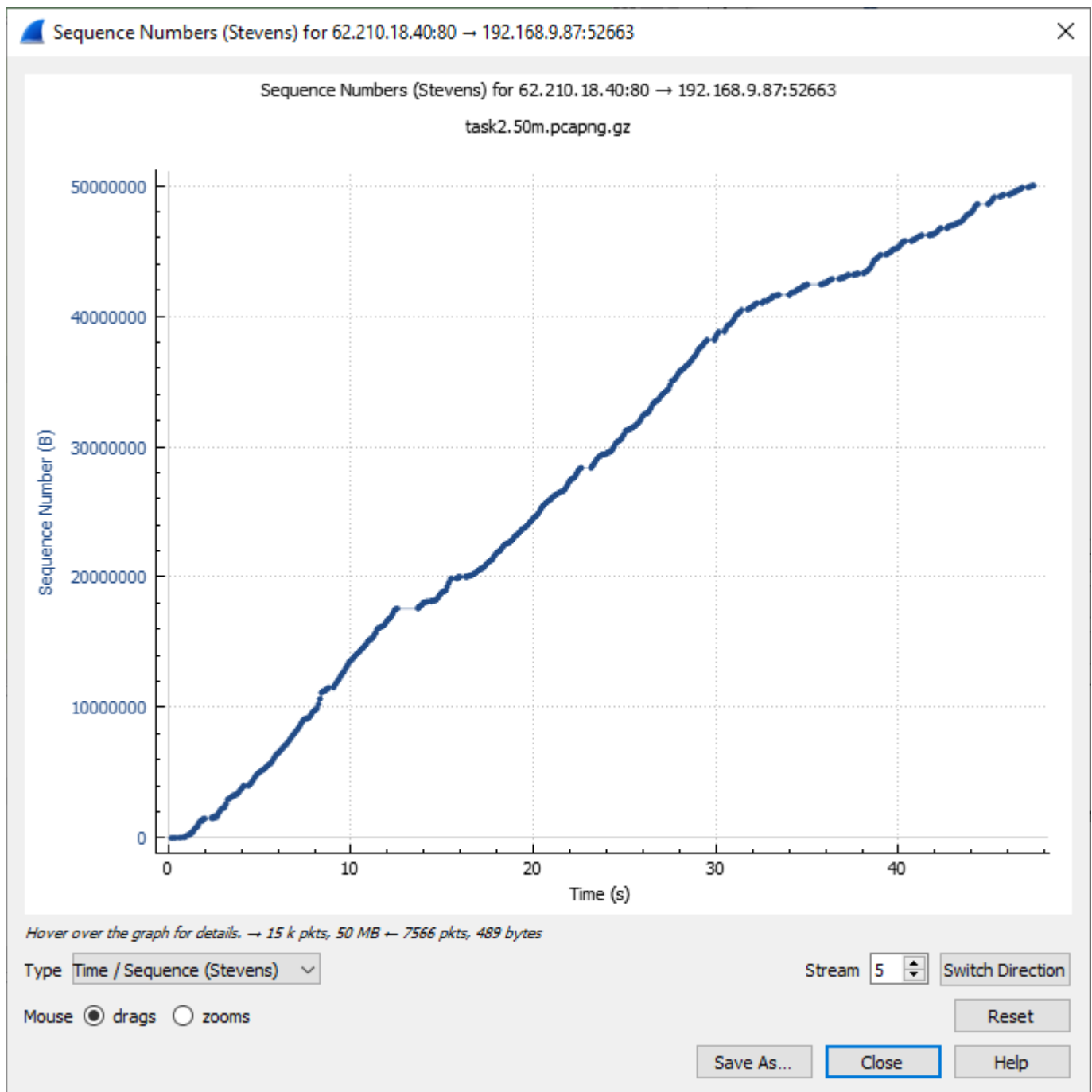
Reset

Save As...

Close

Help

For 50m file,



We see that there are some places the distance between packets increase rapidly. This can be considered as a congestion control where the segments are stopped to avoid congestion. When we zoom in, we see the distance decreasing slowly showing the multiplicative increase.

6. Only for #c above, observe and clearly explain with screenshots, how TCP connection gets terminated in this case, as well as which fields of TCP influence this, due to cancelling of the download in between.

When a download is stopped midway, we see that there are a few a few packets missing where segment is not captured. Also we find that there is no HTTP/1.1 200 OK packet received from the server which can be found in other files where download is completed properly.

I certify that this assignment/report is my own work, based on my personal study and/or research and that I have acknowledged all material and sources used in its preparation, whether they be books, articles, reports, lecture notes, and any other kind of document, electronic or personal communication. I also certify that this assignment/report has not previously been submitted for assessment in any other course, except where specific permission has been granted from all course instructors involved, or at any other time in this course, and that I

have not copied in part or whole or otherwise plagiarised the work of other students and/or persons. I pledge to uphold the principles of honesty and responsibility at CSE@IITH. In addition, I understand my responsibility to report honour violations by other students if I become aware of it.