

CS-359 Assignment-11

Name: Ammaar Ahmad

Roll No.: 1801CS08

In this assignment, we experimented with various statistical values which related to the performance of Internet Protocols like Throughput, Round trip time, Packet Size, Number of packets lost, Number of TCP, UDP Packets, Number of responses with respect to requests etc.

For this purpose, we captured data packets from and to facebook.com at two different parts of the day. And on this data, we applied various functionalities of Wireshark to analyze performance.

Capturing Packets

1. Capture Filter: **host www.facebook.com** and then start capturing
7. Open a tab and go to facebook.com and perform actions until at least 2000 packets.

Throughput

Throughput tells you how much data was transferred from a source at any given time and bandwidth tells you how much data could theoretically be transferred from a source at any given time.

We can measure average throughput and graphical plot of throughputs of various packets using Wireshark

For summary of transfers, got to **“Statistics -> Capture File Properties”**

Wireshark - Capture File Properties - wlp8s0 (host www.facebook.com)

Details

File

Name: /tmp/wireshark_wlp8s0AHFD20.pcapng
 Length: 7,383kB
 Hash (SHA256): 1352571c5ccc93cff6d56ae2e1e0083f9a365e3fd812ffd18f56166321c88d55
 Hash (RIPEMD160): 82b0106ec7916d75bba51ca1c7d1aa3844005100
 Hash (SHA1): 004b2c965ab115de99d060f6627f66f3765fc54c
 Format: Wireshark/... - pcapng
 Encapsulation: Ethernet

Time

First packet: 2021-04-26 23:18:13
 Last packet: 2021-04-26 23:43:48
 Elapsed: 00:25:35

Capture

Hardware: Intel(R) Core(TM) i5-9300H CPU @ 2.40GHz (with SSE4.2)
 OS: Linux 5.8.0-50-generic
 Application: Dumpcap (Wireshark) 3.4.2 (Git v3.4.2 packaged as 3.4.2-1~ubuntu20.04.0+wiresharkdevstable1)

Interfaces

Interface	Dropped packets	Capture filter	Link type	Packet size limit
wlp8s0	Unknown	host www.facebook.com	Ethernet	262144 bytes

Statistics

Measurement	Captured	Displayed	Marked
Packets	8530	8530 (100.0%)	—
Time span, s	1535.023	1535.023	—
Average pps	5.6	5.6	—
Average packet size, B	832	832	—
Bytes	7094794	7094794 (100.0%)	0
Average bytes/s	4,621	4,621	—
Average bits/s	36k	36k	—

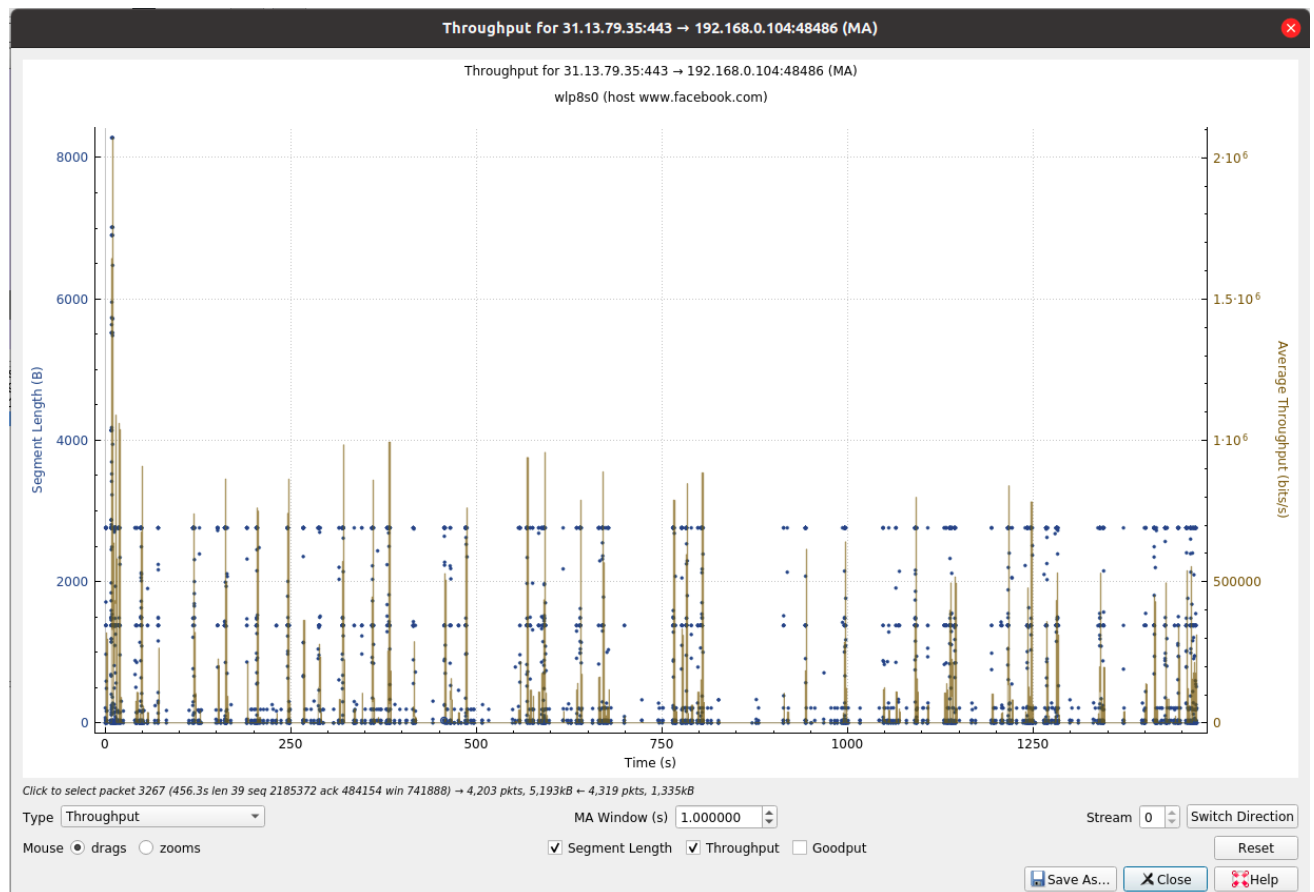
Capture file comments

Refresh Save Comments Close Copy To Clipboard Help

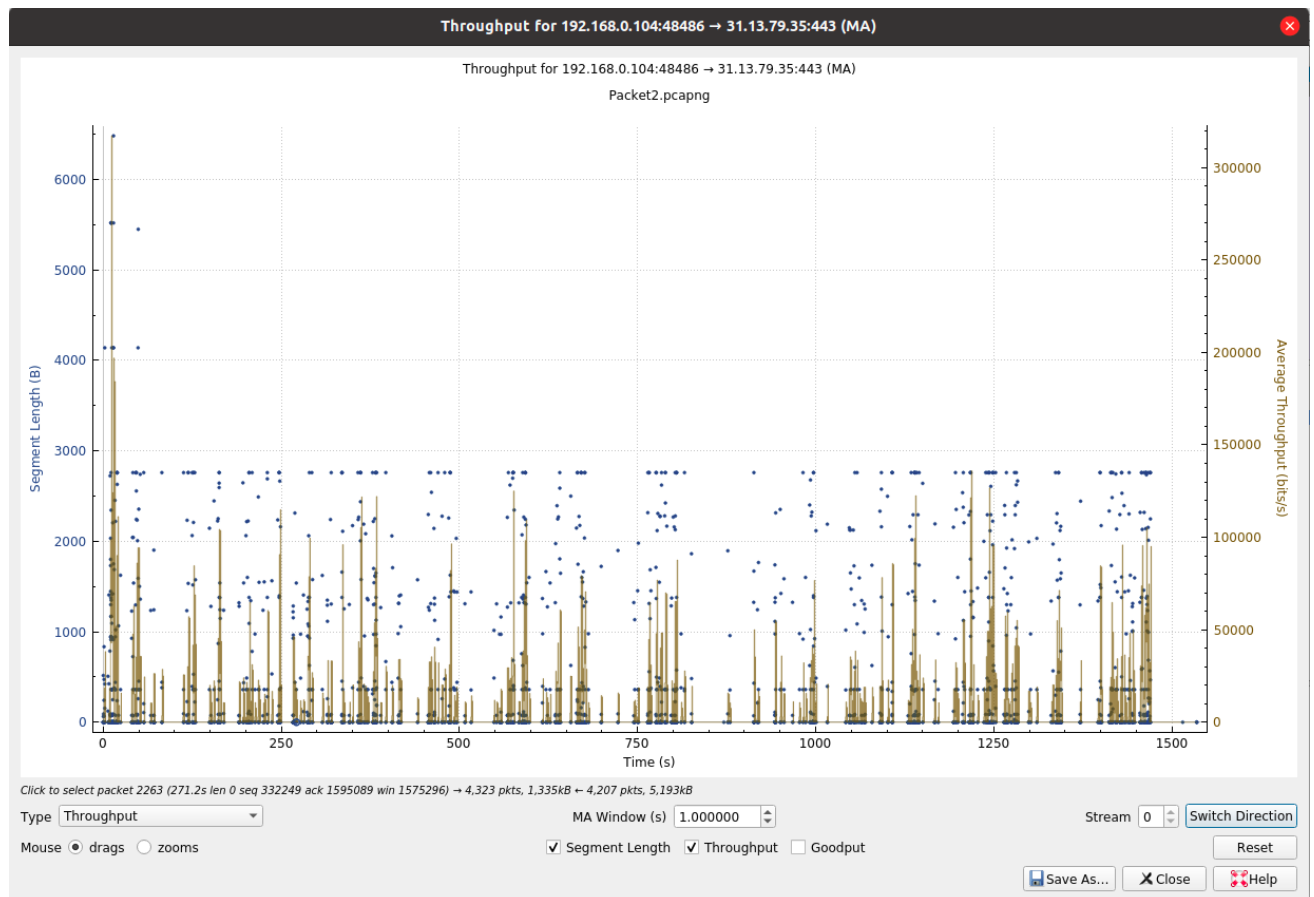
The above picture shows that Average rate of transfer in **bits/sec** is **36k**. Which is average throughput

We can also view the distribution of Throughputs of various packets sent and received from Facebook to the machine.

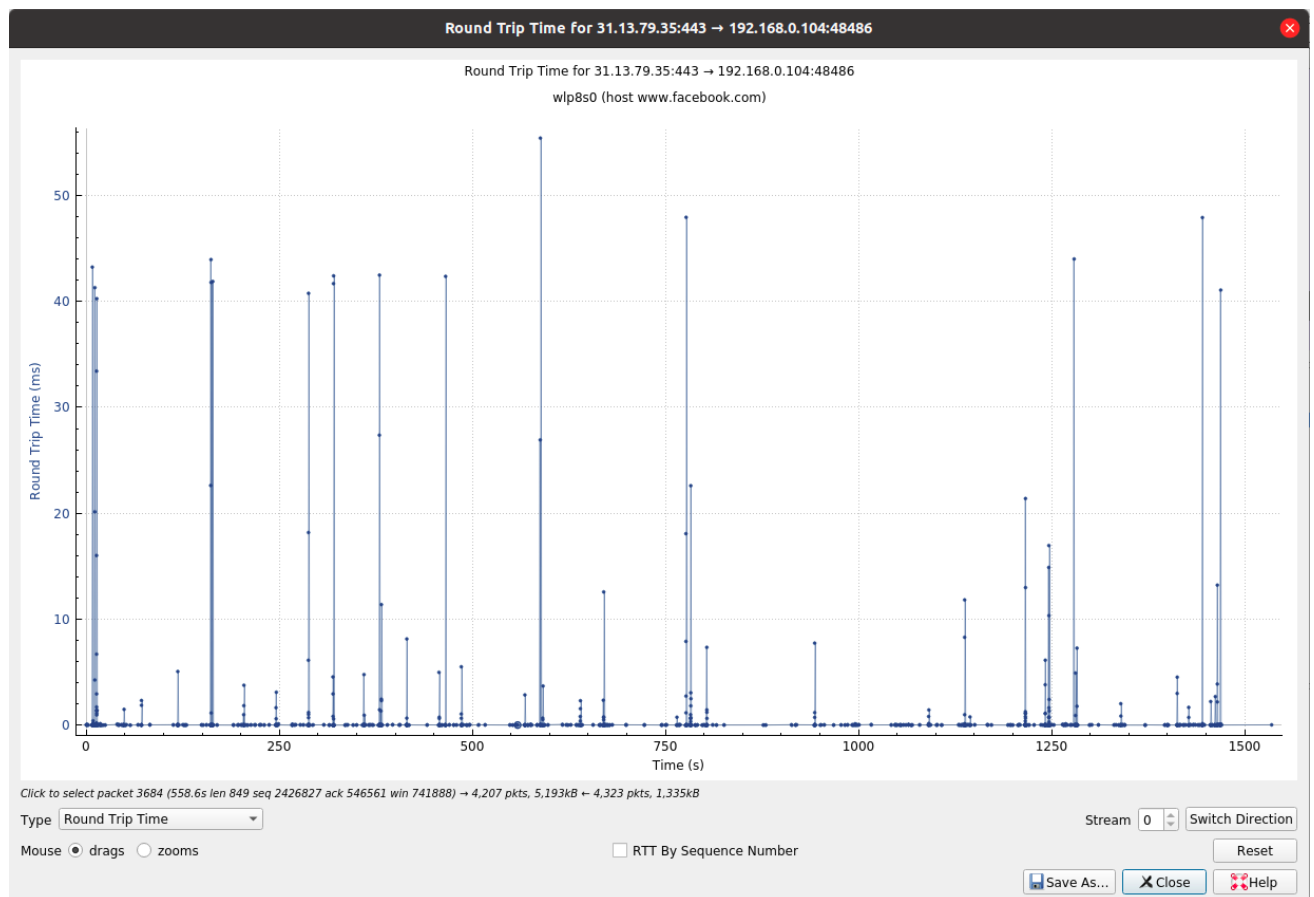
We can do this by **“Statistics -> TCP Stream Graphs -> Throughput”**



We can see in the above graph that there are a lot of spikes in speeds varying from 10^4 range to $8 \cdot 10^5$



We can see in the above graph that there are a lot of spikes in speeds varying from $5 \cdot 10^4$ range to $15 \cdot 10^4$



Round Trip Time

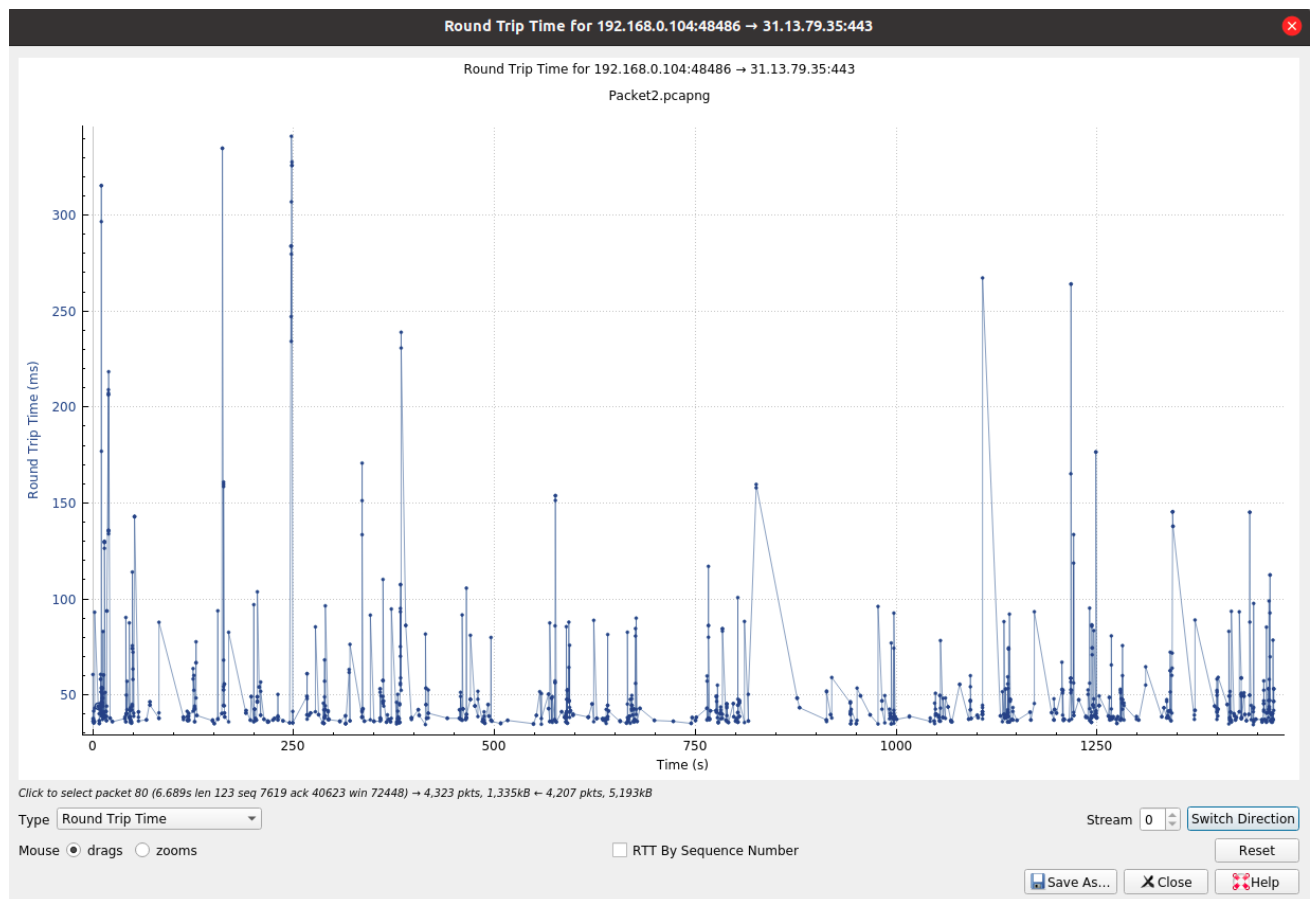
Round Trip Time (RTT) is the length time it takes for a data packet to be sent to a destination plus the time it takes for an acknowledgment of that packet to be received back at the origin.

It is possible to plot the graphs similar to those of throughputs for to and fro packets in Wireshark.

We can do so by **“Statistics -> TCP Stream Graphs -> Round Trip Time”**

The following graph plots the RTT of packets going from machine to facebook. We can see they vary between 0.1ms to 10 ms with major chunks lying from 0 to 4.

The packets coming from Facebook have very less RTT. Because of the high speed servers of Facebook and the acknowledgement is also sent immediately from our machine



The following graph plots the RTT of packets going from machine to facebook. We can see they vary between 0.1 ms to 100 ms with major chunks lying from 0 to 10.

Packet Length

The summary of amount of data carried by packets can be displayed using “**Statistics -> Packet lengths**”

Following are the results for data collected at different times

Wireshark · Packet Lengths · wlp8s0 (host www.facebook.com)								
Topic / Item	Count	Average	Min Val	Max Val	Rate (ms)	Percent	Burst Rate	Burst Start
▼ Packet Lengths	8530	831.75	66	8346	0.0056	100%	0.9700	8.072
0-19	0	-	-	-	0.0000	0.00%	-	-
20-39	0	-	-	-	0.0000	0.00%	-	-
40-79	3983	66.66	66	79	0.0026	46.69%	0.4900	8.072
80-159	765	112.98	80	159	0.0005	8.97%	0.0700	361.375
160-319	368	251.68	160	318	0.0002	4.31%	0.0400	8.846
320-639	346	444.56	324	635	0.0002	4.06%	0.0300	12.099
640-1279	278	982.67	643	1278	0.0002	3.26%	0.0400	8.091
1280-2559	1447	1593.94	1287	2559	0.0009	16.96%	0.1600	319.363
2560-5119	1315	2843.45	2564	4246	0.0009	15.42%	0.3000	8.076
5120 and greater	28	6345.07	5516	8346	0.0000	0.33%	0.0900	9.836

Display filter:

Apply

Copy

Save as...

Close

In the above data, we can see the average length of packet is 831.75 bytes

Packets Lost

We can check the number of packets lost using the “**Statistics -> Capture File Properties**”

The summary shows many details including Dropped packets.

Luckily, due to the reliable Facebook servers, we get 0 dropped packets.

Interface wlp8s0	Dropped packets 0 (0.0%)	Capture filter host www.facebook.com	Link type Ethernet	Packet size limit 262144 bytes
---------------------	-----------------------------	---	-----------------------	-----------------------------------

TCP Packets

Since Facebook Only uses TCP Protocol, there isn't much use of data from only Facebook. Therefore for the experiment of Capturing TCP Packets and UDP Packets, we remove the capture filter and collect general data.

We can use the Display filter "**tcp**" to view only TCP Packets.

The image shows a Wireshark packet capture interface. The top bar indicates the host is 'wlp8s0 (host www.facebook.com)'. The main pane displays a list of captured packets, all of which are TCP. The bottom pane shows the detailed view of the selected packet (No. 1), which is a SYN packet from 192.168.0.104 to 192.168.0.104. The packet details include Ethernet II, Internet Protocol Version 4, and Transmission Control Protocol. The packet bytes pane shows the raw data in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.0.104	192.168.0.104	TCP	74	48486 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=1847807792 TSecr=0 WS=129
2	0.038427897	192.168.0.104	192.168.0.104	TCP	76	443 → 48486 [ACK] Seq=8 Ack=1 Win=65535 Len=0 MSS=1392 SACK_PERM=1 TSval=560199692 TSecr=1847807792 WS=256
3	0.038475650	192.168.0.104	192.168.0.104	TCP	68	48486 → 443 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=1847807631 TSecr=560199692
4	0.039010335	192.168.0.104	192.168.0.104	TLV1.3	583	Client Hello
5	0.099610318	192.168.0.104	192.168.0.104	TCP	68	443 → 48486 [ACK] Seq=1 Ack=518 Win=66816 Len=0 TSval=560199731 TSecr=1847807831
6	0.099610563	192.168.0.104	192.168.0.104	TLV1.3	278	Server Hello, Change Cipher Spec, Application Data
7	0.099647880	192.168.0.104	192.168.0.104	TCP	66	48486 → 443 [ACK] Seq=518 Ack=213 Win=64128 Len=0 TSval=1847807892 TSecr=560199732
8	0.100372493	192.168.0.104	192.168.0.104	TLV1.3	130	Change Cipher Spec, Application Data
9	0.136592244	192.168.0.104	192.168.0.104	TCP	68	443 → 48486 [ACK] Seq=213 Ack=582 Win=66816 Len=0 TSval=560199792 TSecr=1847807893
10	0.136593512	192.168.0.104	192.168.0.104	TLV1.3	237	Application Data
11	0.136621140	192.168.0.104	192.168.0.104	TCP	66	48486 → 443 [ACK] Seq=582 Ack=384 Win=64000 Len=0 TSval=1847807929 TSecr=560199792
12	0.136593596	192.168.0.104	192.168.0.104	TLV1.3	140	Application Data
13	0.136643567	192.168.0.104	192.168.0.104	TCP	66	48486 → 443 [ACK] Seq=582 Ack=458 Win=64000 Len=0 TSval=1847807929 TSecr=560199792
14	0.437832920	192.168.0.104	192.168.0.104	TLV1.3	158	Application Data
15	0.438807697	192.168.0.104	192.168.0.104	TLV1.3	675	Application Data, Application Data
16	0.475548075	192.168.0.104	192.168.0.104	TCP	68	443 → 48486 [ACK] Seq=458 Ack=674 Win=66816 Len=0 TSval=560200129 TSecr=1847808230
17	0.475548079	192.168.0.104	192.168.0.104	TCP	68	443 → 48486 [ACK] Seq=458 Ack=1283 Win=67840 Len=0 TSval=560200129 TSecr=1847808230
18	0.475549002	192.168.0.104	192.168.0.104	TLV1.3	110	Application Data
19	0.475560305	192.168.0.104	192.168.0.104	TCP	66	48486 → 443 [ACK] Seq=1283 Ack=502 Win=64128 Len=0 TSval=1847808268 TSecr=560200130
20	0.905510307	192.168.0.104	192.168.0.104	TLV1.3	2826	Application Data

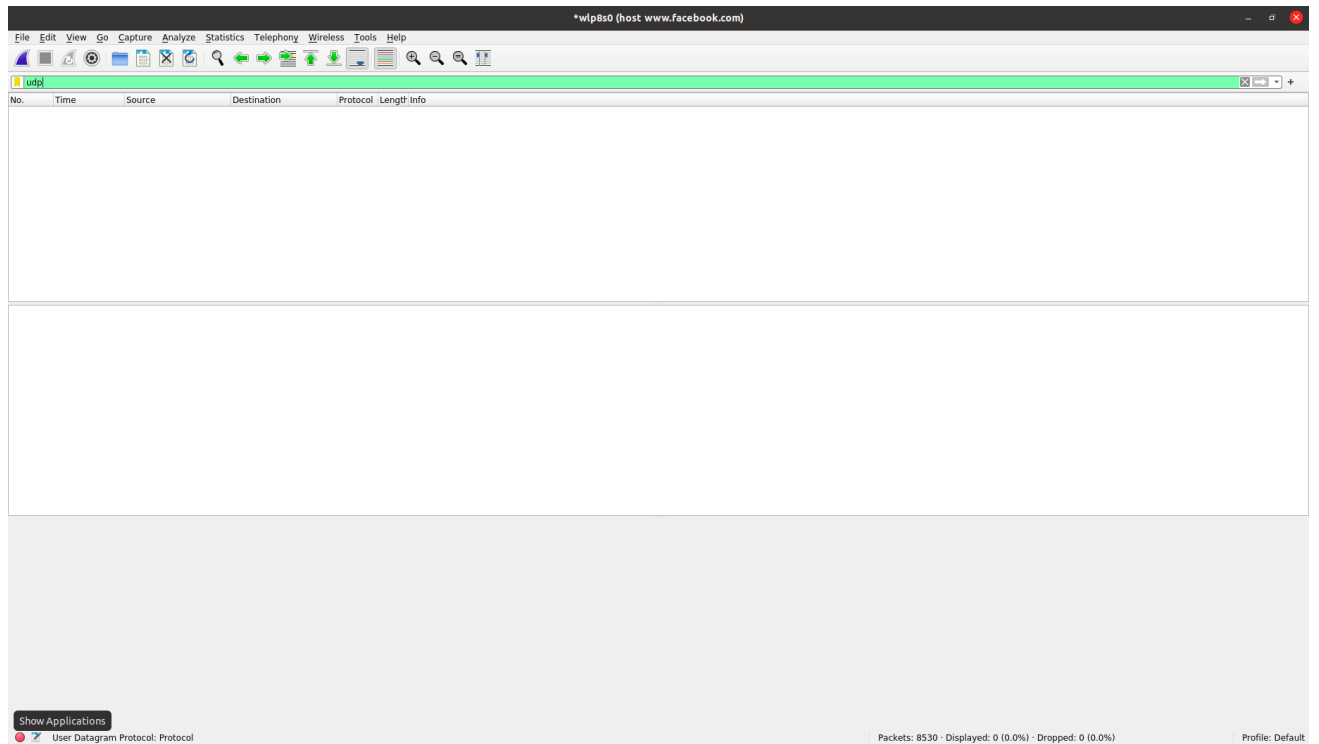
Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface wlp8s0, id 0
Ethernet II, Src: IntelCor, If: 1b:59:6c:6a:77:9f, Dst: Tp-LinT, 68:d5:a8 (f4:f2:6d:68:d5:a8)
Internet Protocol Version 4, Src: 192.168.0.104, Dst: 192.168.0.104
Transmission Control Protocol, Src Port: 48486, Dst Port: 443, Seq: 0, Len: 0

Packets: 8530 · Displayed: 8530 (100.0%) · Dropped: 0 (0.0%) Profile: Default

We can see that 8530 packets out of 8530 captured packets are TCP protocol.

i.e. 100% of packets follow TCP protocol.

UDP Packets



Similarly, we can capture the UDP Packets using the display filter “**udp**” We can see that none of the packets (0% of packets) out of 8530 packets follow UDP Protocol.

All the above experiments were performed 2nd time next morning

Throughput

Throughput tells you how much data was transferred from a source at any given time and bandwidth tells you how much data could theoretically be transferred from a source at any given time.

We can measure average throughput and graphical plot of throughputs of various packets using Wireshark

For summary of transfers, got to **“Statistics -> Capture File Properties”**

Wireshark · Capture File Properties · wlp8s0 (host www.facebook.com)

Details

File

Name: /tmp/wireshark_wlp8s06VUM20.pcapng
 Length: 2,850kB
 Hash (SHA256): c75665e3f3d34a2882fdf193db4d9e6054272932fb2b3ebb39504a6a9df2a6eb
 Hash (RIPEMD160): af482079846065cee12138110bfc712de4de8b1b
 Hash (SHA1): f68c4a250e7cc1dd5def6eb7f38bb8a790a2cda4
 Format: Wireshark/... - pcapng
 Encapsulation: Ethernet

Time

First packet: 2021-04-27 09:48:07
 Last packet: 2021-04-27 09:54:56
 Elapsed: 00:06:48

Capture

Hardware: Intel(R) Core(TM) i5-9300H CPU @ 2.40GHz (with SSE4.2)
 OS: Linux 5.8.0-50-generic
 Application: Dumpcap (Wireshark) 3.4.2 (Git v3.4.2 packaged as 3.4.2-1~ubuntu20.04.0+wiresharkdevstable1)

Interfaces

Interface	Dropped packets	Capture filter	Link type	Packet size limit
wlp8s0	0 (0.0%)	host www.facebook.com	Ethernet	262144 bytes

Statistics

Measurement	Captured	Displayed	Marked
Packets	3786	3786 (100.0%)	—
Time span, s	408.679	408.679	—
Average pps	9.3	9.3	—
Average packet size, B	719	719	—
Bytes	2722412	2722412 (100.0%)	0
Average bytes/s	6,661	6,661	—
Average bits/s	53k	53k	—

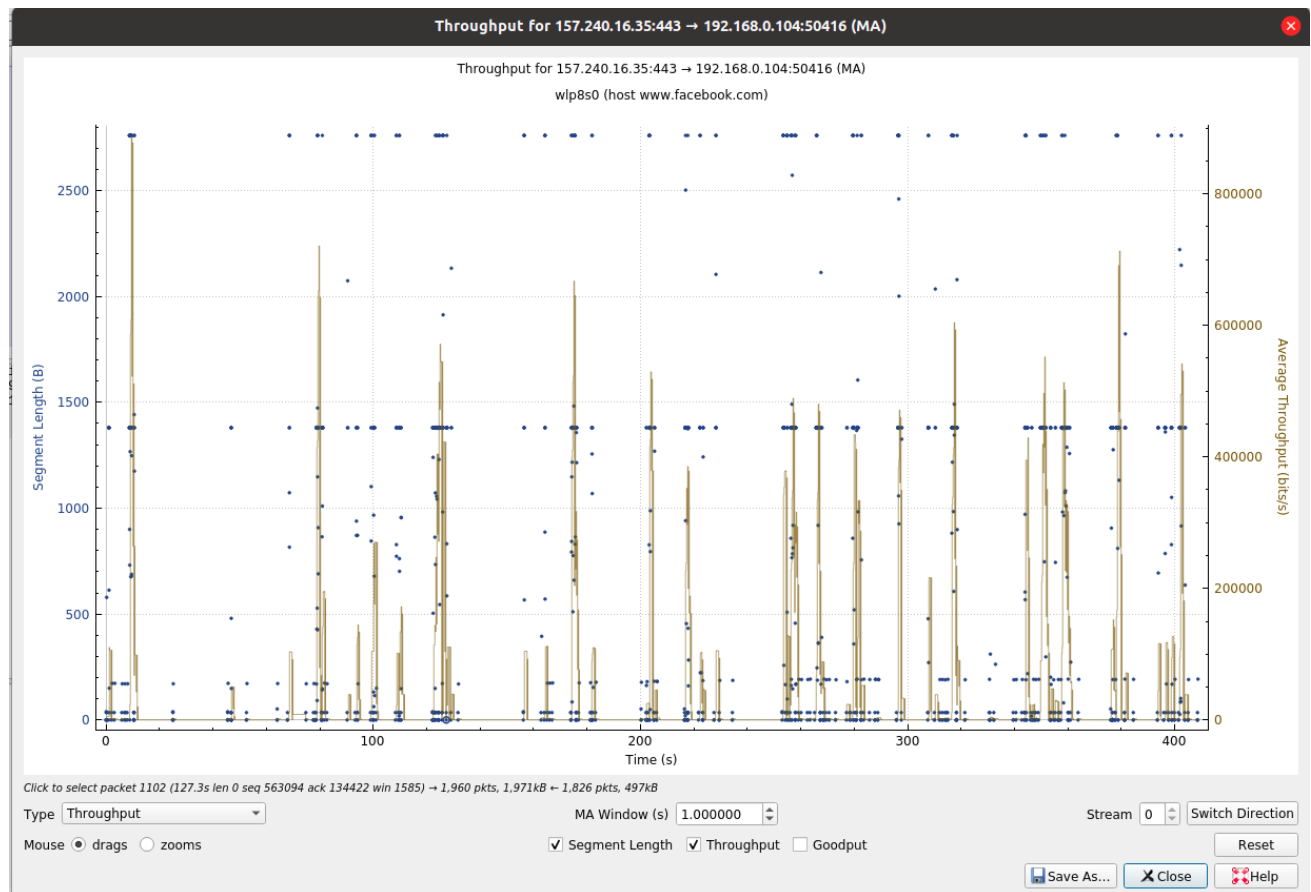
Capture file comments

Refresh Save Comments Close Copy To Clipboard Help

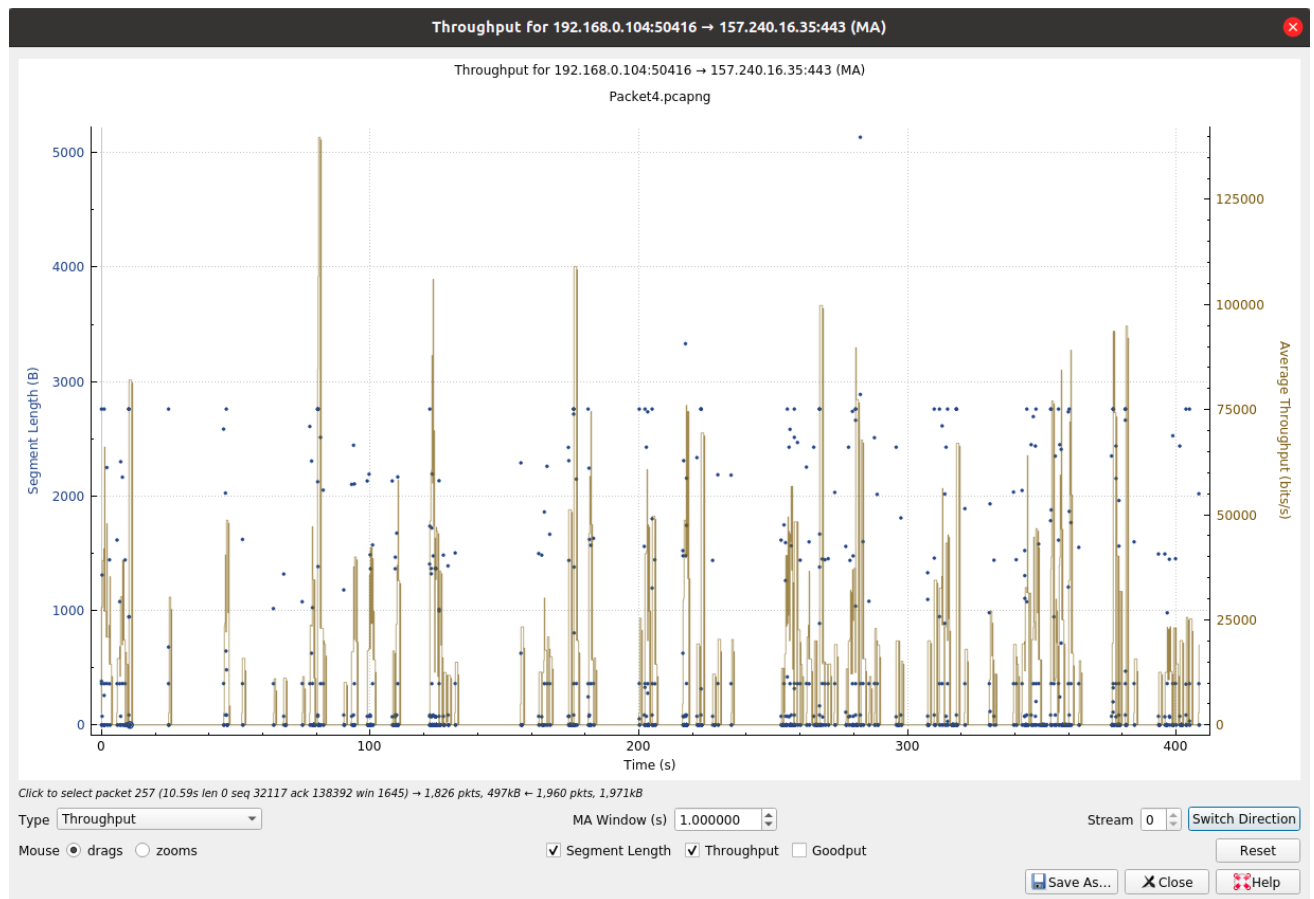
The above picture shows that Average rate of transfer in **bits/sec is 53k**. Which is average throughput

We can also view the distribution of Throughputs of various packets sent and received from Facebook to the machine.

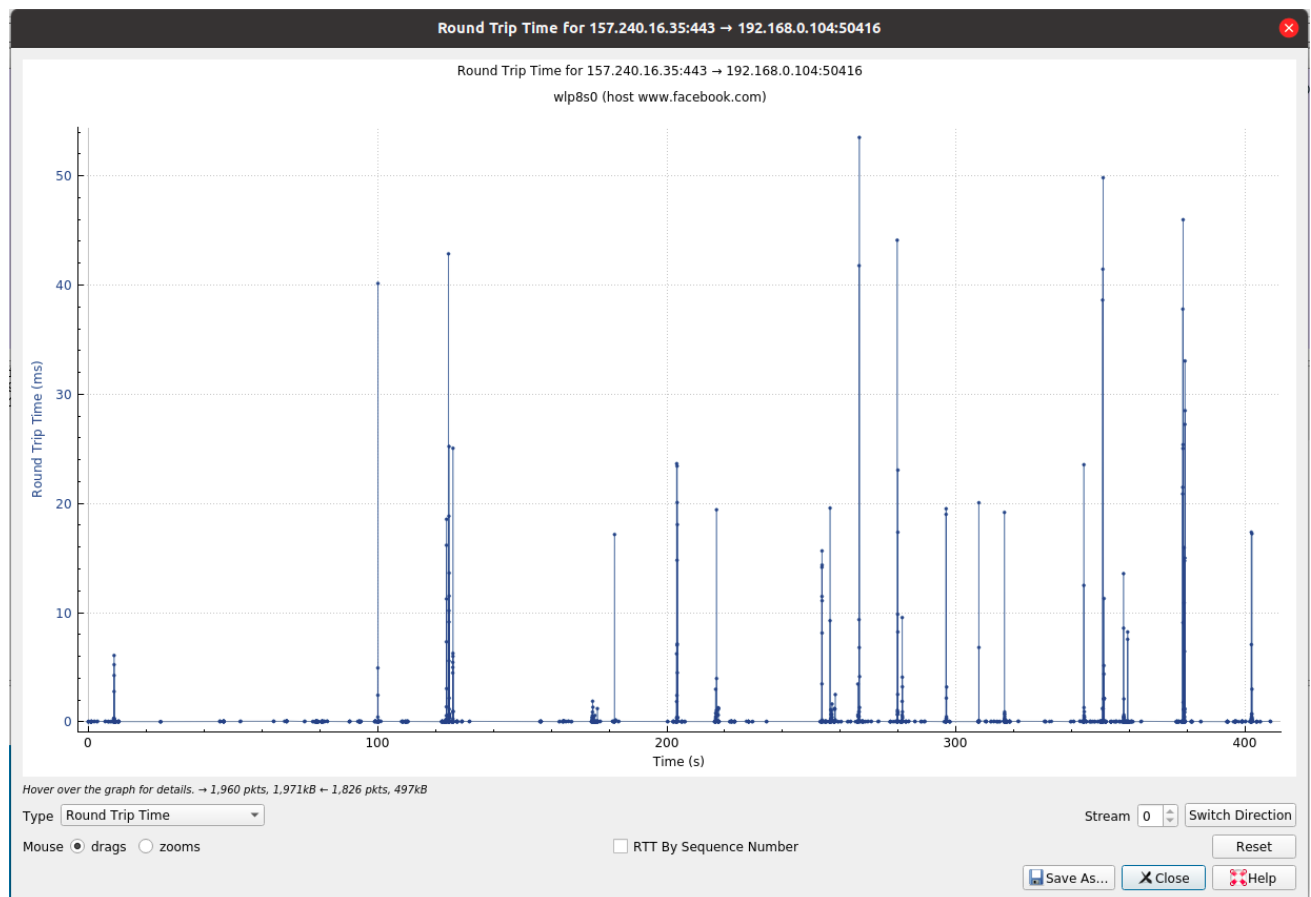
We can do this by **“Statistics -> TCP Stream Graphs -> Throughput”**



We can see in the above graph that there are a lot of spikes in speeds varying from 2×10^5 range to 8×10^5



We can see in the above graph that there are a lot of spikes in speeds varying from 2.5×10^4 range to 7.5×10^4



Round Trip Time

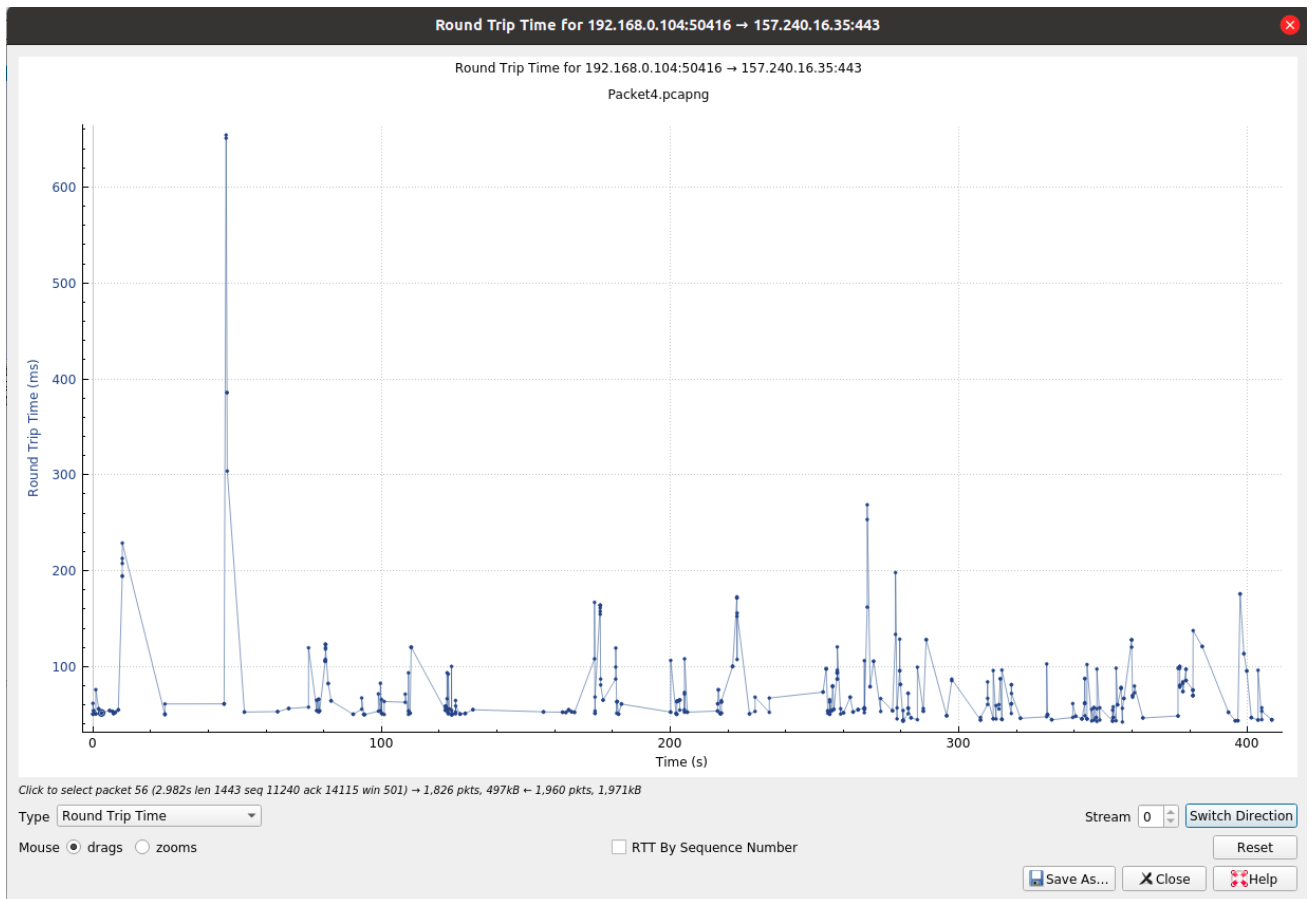
Round Trip Time (RTT) is the length time it takes for a data packet to be sent to a destination plus the time it takes for an acknowledgment of that packet to be received back at the origin.

It is possible to plot the graphs similar to those of throughputs for to and fro packets in Wireshark.

We can do so by **“Statistics -> TCP Stream Graphs -> Round Trip Time”**

The following graph plots the RTT of packets going from machine to facebook. We can see they vary between 0.1ms to 10 ms with major chunks lying from 0 to 4.

The packets coming from Facebook have very less RTT. Because of the high speed servers of Facebook and the acknowledgement is also sent immediately from our machine



The following graph plots the RTT of packets going from machine to facebook. We can see they vary between 10 ms to 40 ms with major chunks lying from 20 to 30 .

Packet Length

The summary of amount of data carried by packets can be displayed using “**Statistics -> Packet lengths**”

Following are the results for data collected at different times

Wireshark · Packet Lengths · wlp8s0 (host www.facebook.com)								
Topic / Item	Count	Average	Min Val	Max Val	Rate (ms)	Percent	Burst Rate	Burst Start
▼ Packet Lengths	3786	719.07	66	5201	0.0093	100%	0.6000	78.770
0-19	0	-	-	-	0.0000	0.00%	-	-
20-39	0	-	-	-	0.0000	0.00%	-	-
40-79	1700	67.24	66	78	0.0042	44.90%	0.3000	78.770
80-159	329	112.94	84	158	0.0008	8.69%	0.1000	8.803
160-319	114	243.38	165	312	0.0003	3.01%	0.0200	80.903
320-639	123	441.32	324	637	0.0003	3.25%	0.0300	257.876
640-1279	111	951.05	645	1262	0.0003	2.93%	0.0300	174.218
1280-2559	1205	1498.91	1281	2534	0.0029	31.83%	0.3000	78.770
2560-5119	203	2817.57	2568	3397	0.0005	5.36%	0.0500	8.803
5120 and greater	1	5201.00	5201	5201	0.0000	0.03%	0.0100	282.440

Display filter:

Apply

Copy

Save as...

Close

In the above data, we can see the average length of packet is 719.07 bytes

Packets Lost

We can check the number of packets lost using the “**Statistics -> Capture File Properties**”

The summary shows many details including Dropped packets.

Luckily, due to the reliable Facebook servers, we get 0 dropped packets.

Interface wlp8s0	Dropped packets 0 (0.0%)	Capture filter host www.facebook.com	Link type Ethernet	Packet size limit 262144 bytes
---------------------	-----------------------------	---	-----------------------	-----------------------------------

TCP Packets

Since Facebook Only uses TCP Protocol, there isn't much use of data from only Facebook. Therefore for the experiment of Capturing TCP Packets and UDP Packets, we remove the capture filter and collect general data.

We can use the Display filter "**tcp**" to view only TCP Packets.

The screenshot shows the Wireshark network protocol analyzer interface. The title bar indicates the host is www.facebook.com. The menu bar includes File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Wireless, Tools, and Help. The toolbar contains icons for various functions. The packet list pane shows 20 captured packets, all of which are TCP. The packet details pane shows the selected packet (No. 1) as a TCP Reset (RST) with Seq=1, Ack=1, Len=0. The packet bytes pane shows the raw data in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.0.104	157.240.16.35	TLSv1.2	427	Application Data
2	0.000126559	192.168.0.104	157.240.16.35	TLSv1.2	2826	Application Data
3	0.000172786	192.168.0.104	157.240.16.35	TLSv1.2	449	Application Data
4	0.050661981	157.240.16.35	192.168.0.104	TCP	68	443 -> 50416 [ACK] Seq=1 Ack=362 Win=287 Len=0 TSval=3914338896 TSecr=4180017995
5	0.050661298	157.240.16.35	192.168.0.104	TCP	68	443 -> 50416 [ACK] Seq=1 Ack=3122 Win=389 Len=0 TSval=3914338896 TSecr=4180017996
6	0.050661320	157.240.16.35	192.168.0.104	TCP	68	443 -> 50416 [ACK] Seq=1 Ack=3122 Win=389 Len=0 TSval=3914338896 TSecr=4180017996
7	0.050684993	157.240.16.35	192.168.0.104	TLSv1.2	101	Application Data
8	0.050692380	192.168.0.104	157.240.16.35	TCP	66	50416 -> 443 [ACK] Seq=3504 Ack=36 Win=501 Len=0 TSval=4180018046 TSecr=3914338897
9	0.051153331	157.240.16.35	192.168.0.104	TLSv1.2	105	Application Data
10	0.051153382	192.168.0.104	157.240.16.35	TCP	66	50416 -> 443 [ACK] Seq=3504 Ack=75 Win=501 Len=0 TSval=4180018047 TSecr=3914338897
11	0.052518146	157.240.16.35	192.168.0.104	TCP	68	443 -> 50416 [ACK] Seq=75 Ack=3504 Win=320 Len=0 TSval=3914338897 TSecr=4180017996
12	0.212232442	157.240.16.35	192.168.0.104	TLSv1.2	645	Application Data
13	0.212251818	192.168.0.104	157.240.16.35	TCP	66	50416 -> 443 [ACK] Seq=3504 Ack=654 Win=497 Len=0 TSval=4180018208 TSecr=3914331040
14	0.231801084	192.168.0.104	157.240.16.35	TLSv1.2	142	Application Data
15	0.231834004	192.168.0.104	157.240.16.35	TLSv1.2	1376	Application Data
16	0.285017820	157.240.16.35	192.168.0.104	TCP	68	443 -> 50416 [ACK] Seq=654 Ack=3580 Win=320 Len=0 TSval=3914331128 TSecr=4180018227
17	0.285018213	157.240.16.35	192.168.0.104	TCP	68	443 -> 50416 [ACK] Seq=654 Ack=4890 Win=330 Len=0 TSval=3914331128 TSecr=4180018227
18	0.286262140	157.240.16.35	192.168.0.104	TLSv1.2	101	Application Data
19	0.286282116	192.168.0.104	157.240.16.35	TCP	66	50416 -> 443 [ACK] Seq=4890 Ack=689 Win=501 Len=0 TSval=4180018282 TSecr=3914331129
20	0.991449729	192.168.0.104	157.240.16.35	TLSv1.2	427	Application Data

Frame 1: 427 bytes on wire (3416 bits), 427 bytes captured (3416 bits) on interface wlan0, id 0
Ethernet II, Src: IntelCor_8f:1b:59 (6c:6a:77:9f:1b:59), Dst: Tp-LinkT_68:d5:a8 (f4:f2:6d:68:d5:a8)
Internet Protocol Version 4, Src: 192.168.0.104, Dst: 157.240.16.35
Transmission Control Protocol, Src Port: 50416, Dst Port: 443, Seq: 1, Ack: 1, Len: 361
Transport Layer Security

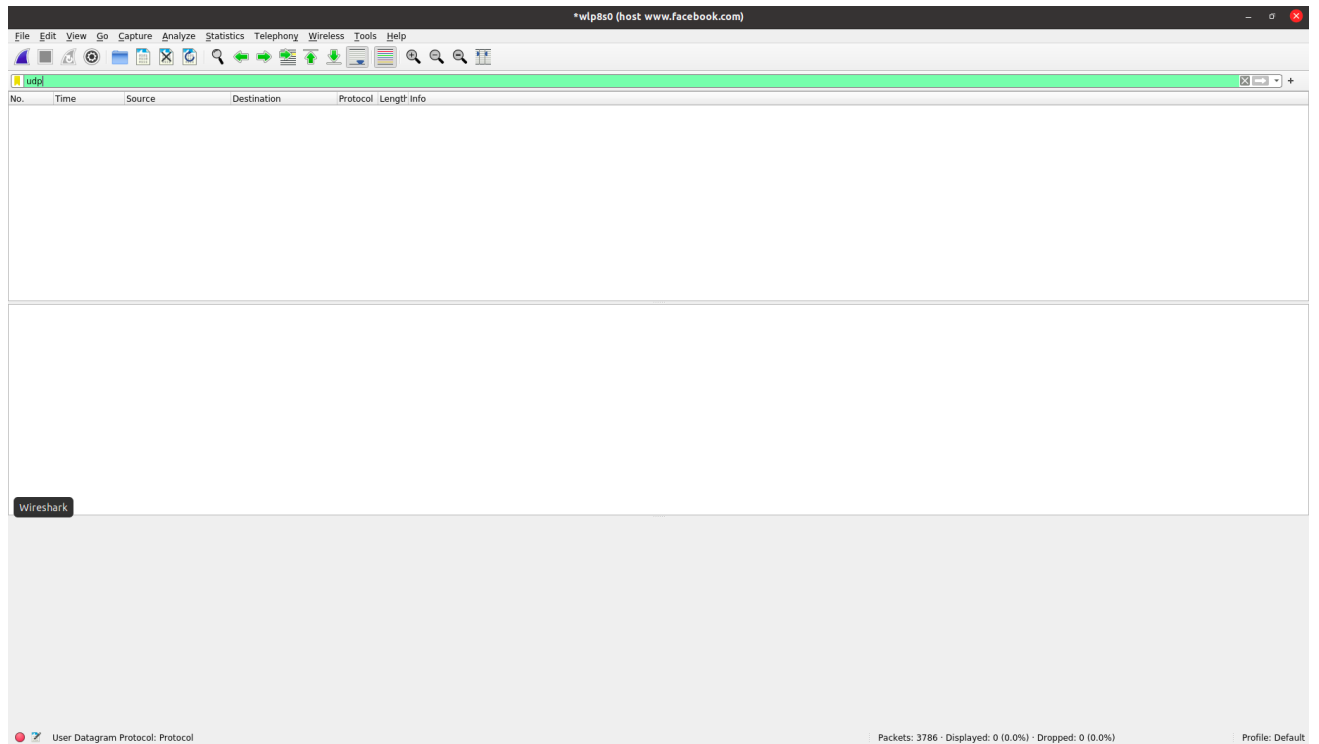
Wireshark

0000 f4 f2 6d 68 d5 a8 6c 6a 77 9f 1b 59 08 00 45 00 --mh-[]
0010 01 90 9f 95 48 09 40 08 2a a2 c0 a8 08 08 9d f9 --000
0020 10 23 c4 f0 01 bb b2 9b 7d 8c e6 06 a3 d5 80 18 --#-...
0030 01 f5 70 b3 08 00 01 01 08 8a f9 26 03 4b e9 4f --p-...
0040 9c f0 17 03 03 91 64 97 0b 32 de 8b 5d be 98 3f --d-...
0050 a8 e5 a4 d3 e6 91 ce 53 4d 41 16 c5 f8 a7 ba 5e --S-...
0060 b3 1a 36 36 f1 96 df 54 8c 26 27 22 8f f2 9a 28 --B-...T
0070 72 54 21 f9 f0 79 cf 1f 09 0b 05 c4 40 45 8d af --v-...
0080 22 83 91 5a 97 24 f5 e2 05 b4 8a 8c 05 f1 fa ea --Z-S-...
0090 cc 51 17 80 e4 bf ca 8d 11 43 a3 76 94 c4 8a a7 --Q-...
00a0 3a 17 e5 43 81 8e 18 e6 fa e9 3c f6 52 54 84 f6 --C-...
00b0 3b c0 c9 5f 2e 4c 4e 9b 91 90 6c 59 cc 96 7d 72 --LM-...Y-...
00c0 f7 37 8c da a6 59 40 7c 5b 11 73 40 5e e3 85 a3 --7-...Y-...
00d0 07 02 7c 73 78 a6 eb fe ad c9 29 d8 27 62 40 fa --[sx-...
00e0 a1 1c 33 66 06 46 b3 23 3b 06 25 fb 1f 97 c4 --3F-F-#
00f0 ec ee 58 ca 41 62 62 7b 2e c4 7c bc 9f 95 ec 9b --V-Abb[...]
0100 52 1f ec 50 3a 7a 79 59 ac e3 00 00 94 ea 55 3e 78 --R-P-pY
0110 f4 c3 57 36 f2 f4 e7 4b af 5f f6 a1 e5 29 87 0e --W-...K
0120 26 79 83 44 43 88 86 67 6c 8c 43 3f ee c0 1a d7 --&y-DC-g 1-C7
0130 68 8f 14 27 cf 99 c6 7e b3 d1 b2 a8 08 9d 39 c5 --e-...-
0140 39 c5 23 a4 f6 26 47 05 28 7e 4b 17 05 13 8f eb --0-#-...G-...
Packets: 3786 · Displayed: 3786 (100.0%) · Dropped: 0 (0.0%) Profile: Default

We can see that 3786 packets out of 3786 captured packets are TCP protocol.

i.e. 100% of packets follow TCP protocol.

UDP Packets



Similarly, we can capture the UDP Packets using the display filter “**udp**” We can see that none of the packets (0% of packets) out of 3786 packets follow UDP Protocol.

Request-Response

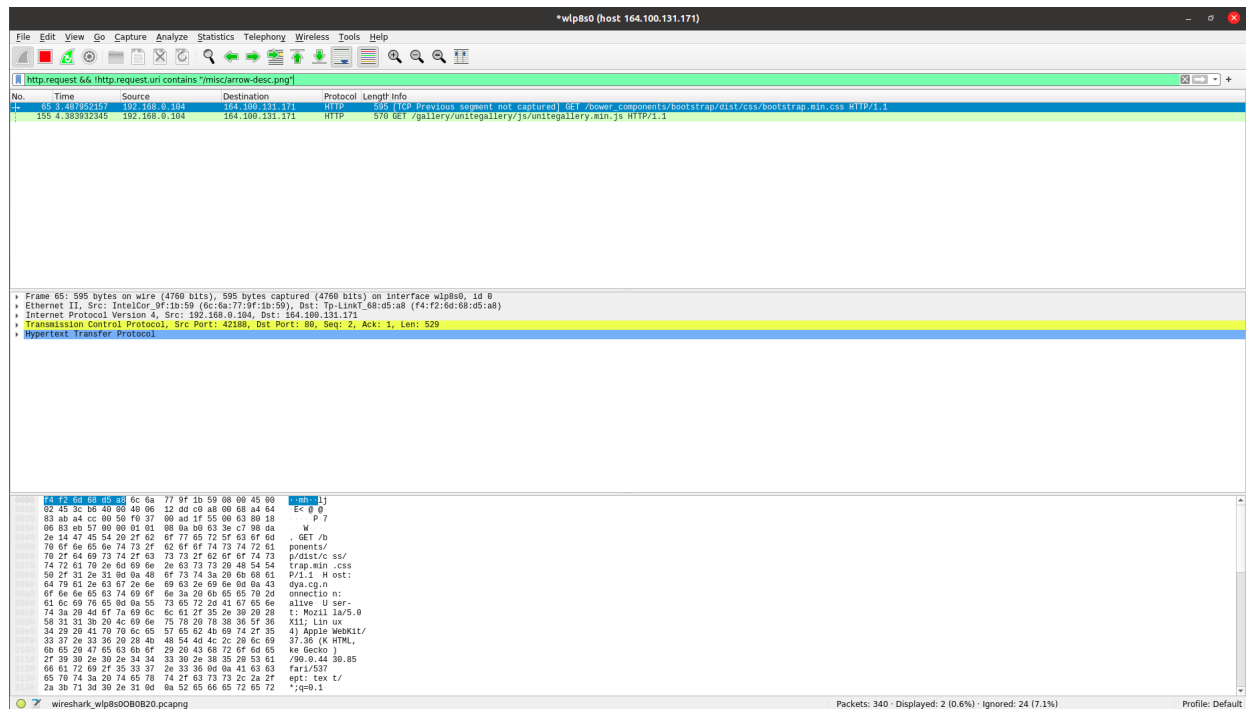
As a part of this experiment, we are also required to capture Request Response pairs. i.e. How many Responses would be there for one particular Request.

Therefore, we need HTTP GET data. Unfortunately, Facebook follows HTTPS protocol, and therefore we need to change our source for this experiment. A website which uses HTTP is essential.

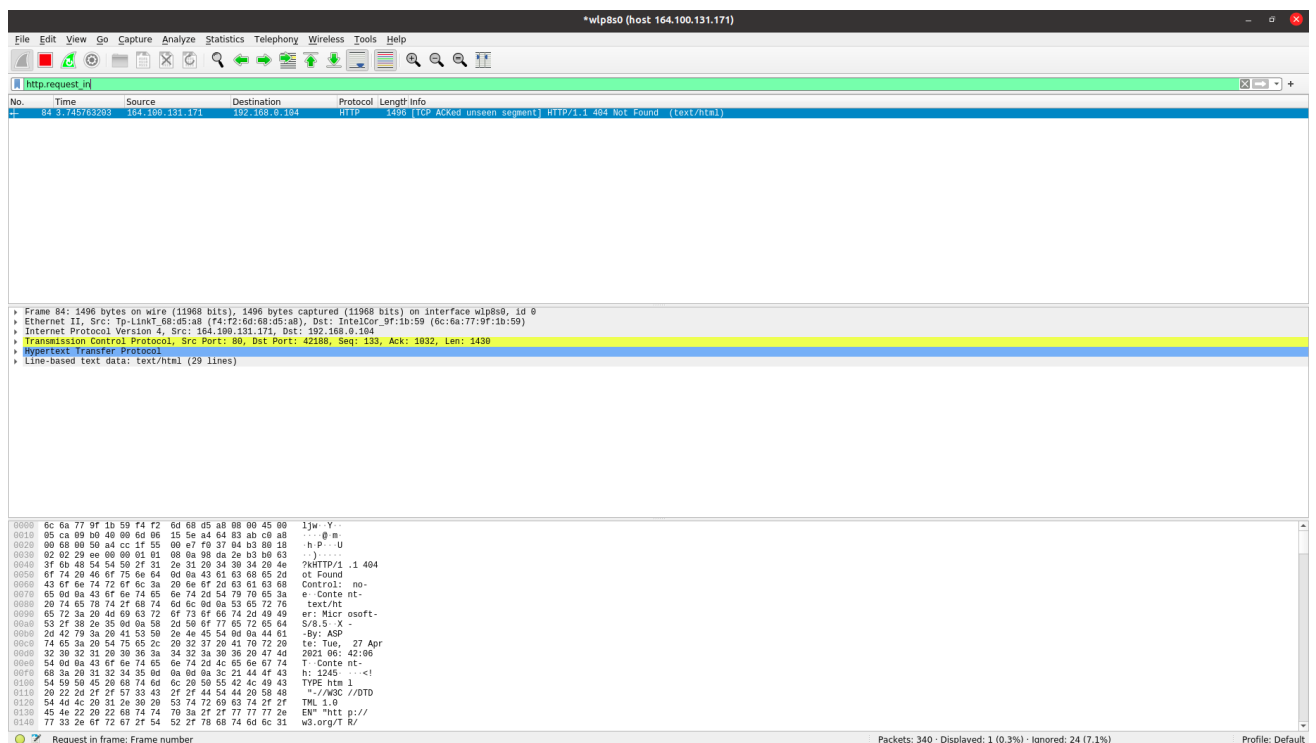
For this experiment, we use a site http://khadya.cg.nic.in/Directorate_AboutUs_Hn.aspx IP address is 157.140.2.239. We can capture its data using the “**host 164.100.131.171**” capture filter. Now that we have the data, we need to perform the following steps to capture its response packets

1. From the data, display the request packets using “**http.request**” display filter
2. From the resulting list, choose one random request and display it. I have used “http.request.uri contains “/misc/arrow-desc.png”” for this experiment.

3. Next step involves removing all the other requests and their responses. This could be done using “**http.request && !http.request.uri contains "/misc/arrow-desc.png"**” display filter and remove that list using “**Edit -> Ignore all displayed**”



4. Now there is only the response for the request we are considering. We can display the response using “**http.request_in**” display filter



We can now see the singular response for our request.