CS359 Computer Networks Lab Assignment 11

Name: Chandrawanshi Mangesh Shivaji
Roll No: 1801cs16
Date: 22/04/2021
Filename: Assignment11_1801cs16_Report.pdf

Problem:

Calculate the following statistics from your traces while performing experiments at 2 different times of the day: Throughput, RTT, Packet size, number of packets lost, number of UDP and TCP packets, number of responses received with respect to one request sent. Report the observed values in your traces, preferable using tables.

I have captured data packets from and to www.facebook.com using Wireshark at two different times of a day. Using Wireshark all the above statistics are calculated.

Packet Capturing

1. Open Wireshark and Select appropriate network
2. Capture Filter: host www.facebook.com
3. Start packet capturing and switch to any browser
   (OS: Linux (Ubuntu 20.10), Browser: Mozilla Firefox used for this assignment)
4. Go to www.facebook.com
5. Sign in to your account and scroll for about 2-3 minutes
6. Stop Packet Capturing and save it
7. Now we can perform further analysis

Note: Instead of (www.facebook.com) we can use IP address of Facebook in above capture filter, though for that we have to it beforehand. As Facebook has large network, there is no fixed single IP address for it (multiple IP addresses exist), we can get IP for our connection using ping command.

Wireshark Capture Filenames

For Facebook Capture1:	**capture1_4pm.pcapng**
For Facebook Capture2:	**capture2_1pm.pcapng**
For Random Capture:	**random.pcapng**
For http request-response Capture:	**http_request_disney_capture.pcapng**

Summary:

Table for host www.facebook.com:

Statistic	Capture 1 (Time: 4pm)	Capture 2 (Time: 1pm)
Avg. Throughput	377K bits/s	219K bits/s
Range for Throughput (Facebook server to system)	2 x 10^5 bits/s to 1.3 x 10^6 bits/s.	2.5 x 10^5 bits/s to 1.25 x 10^6 bits/s
Range for Throughput (system to Facebook server)	2 x 10^4 bits/s to 3 x 10^5 bits/s	2.5 x 10^4 bits/s to 1.8 x 10^5 bits/s
Range for RTT (Facebook server to system)	0 ms to 50 ms	0 ms to 50 ms
Range for RTT (system to Facebook server)	0 ms to 800 ms	0 ms to 800 ms
Avg. Packet Length	839.57 bytes	850.46 bytes
Dropped Packets	0	0
Total Captured Packets	6080	6025
TCP Packets	6080	6080
UDP Packets	0	0
Number of HTTP requests	0	0
Number of HTTP request responses	0	0

As Facebook uses only TCP no UDP packets are captured. Also, Facebook uses HTTPS (not HTTP) therefore no request-response pair packets are captured.

To cover TCP and UDP statistics I have captured packets without any Capture Filter.
Total packets captured = 6621
Number of TCP packets = 6475 (97.8%)
Number of UDP packets =142 (2.1%)
remaining ARP 4 packets

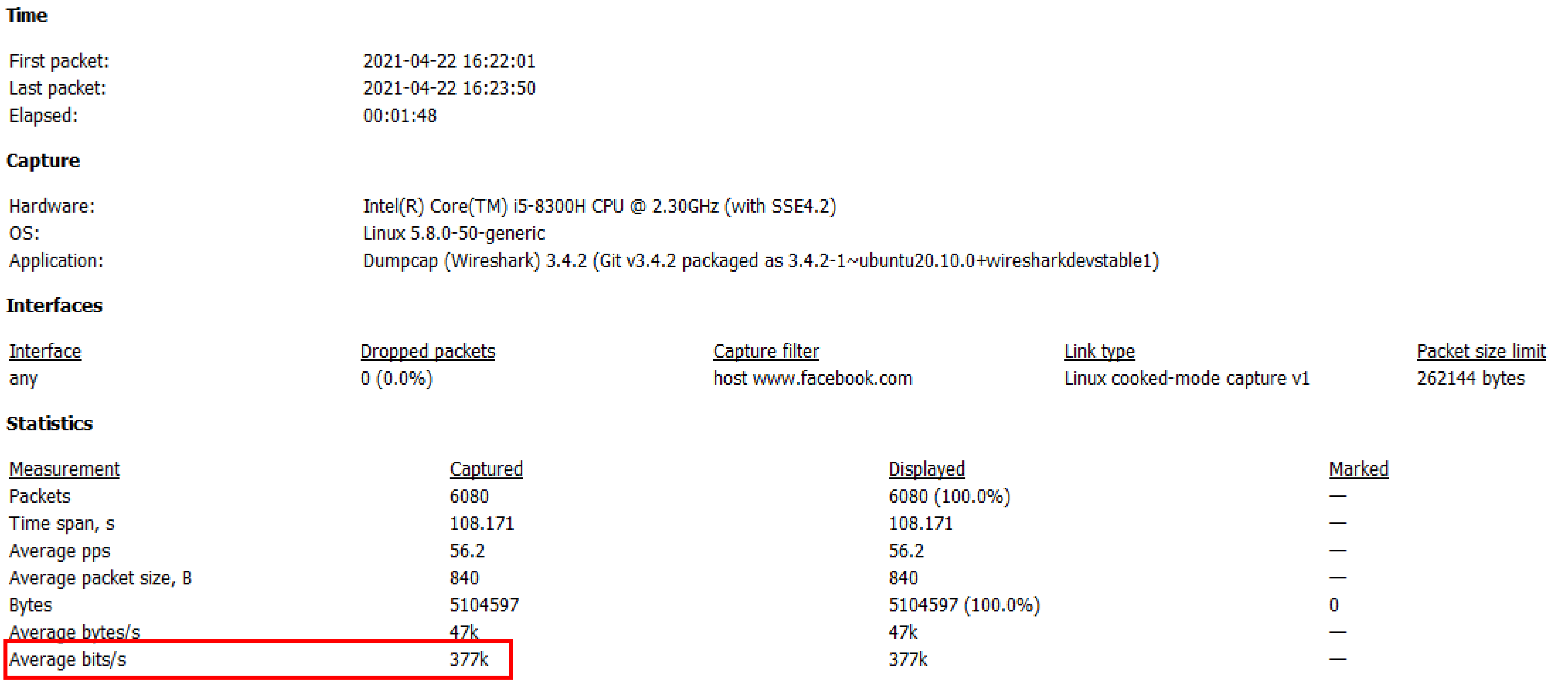
To cover HTTP request-response pair I have captured packets from www.disney.in.
I got one GET request (using filter: http_request) and 200 OK as response for that (using display filter: http_request_in). More details are in description.

Descriptive Analysis:

Throughput

Throughput is the amount of data that was actually transferred from a source to a destination at any given time whereas bandwidth is the data which can be transferred theoretically from a source to a destination at any given time. Wireshark has inbuilt tools which allow us to calculate average throughout and plot graphs of average throughput for whole time interval.

To get summary of **our first capture**, go to **“Statistics -> Capture File Properties”**



Time

First packet:	2021-04-22 16:22:01
Last packet:	2021-04-22 16:23:50
Elapsed:	00:01:48

Capture

Hardware:	Intel(R) Core(TM) i5-8300H CPU @ 2.30GHz (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~ubuntu20.10.0+wiresharkdevstable1)

Interfaces

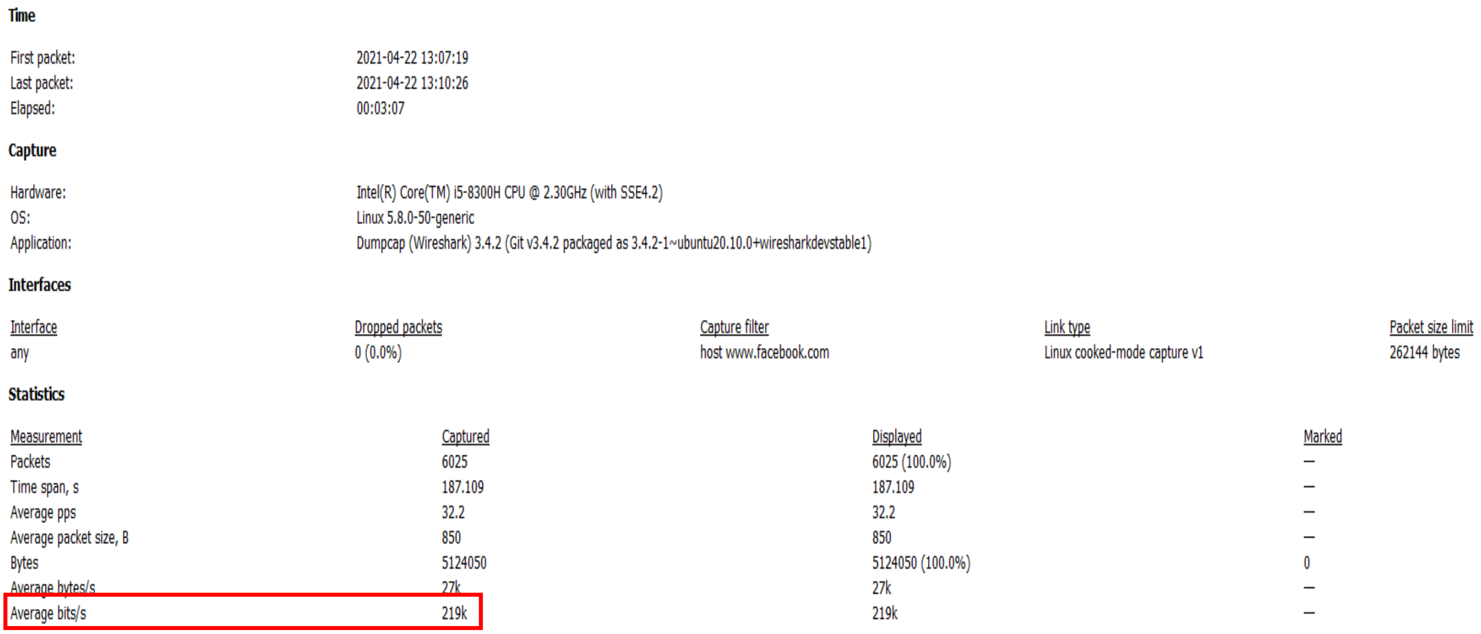
Interface	Dropped packets	Capture filter	Link type	Packet size limit
any	0 (0.0%)	host www.facebook.com	Linux cooked-mode capture v1	262144 bytes

Statistics

Measurement	Captured	Displayed	Marked
Packets	6080	6080 (100.0%)	—
Time span, s	108.171	108.171	—
Average pps	56.2	56.2	—
Average packet size, B	840	840	—
Bytes	5104597	5104597 (100.0%)	0
Average bytes/s	47k	47k	—
Average bits/s	377k	377k	—

Average Throughput = **377k bits/s**

To get summary of **our second capture**, go to **“Statistics -> Capture File Properties”**



Time

First packet:	2021-04-22 13:07:19
Last packet:	2021-04-22 13:10:26
Elapsed:	00:03:07

Capture

Hardware:	Intel(R) Core(TM) i5-8300H CPU @ 2.30GHz (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~ubuntu20.10.0+wiresharkdevstable1)

Interfaces

Interface	Dropped packets	Capture filter	Link type	Packet size limit
any	0 (0.0%)	host www.facebook.com	Linux cooked-mode capture v1	262144 bytes

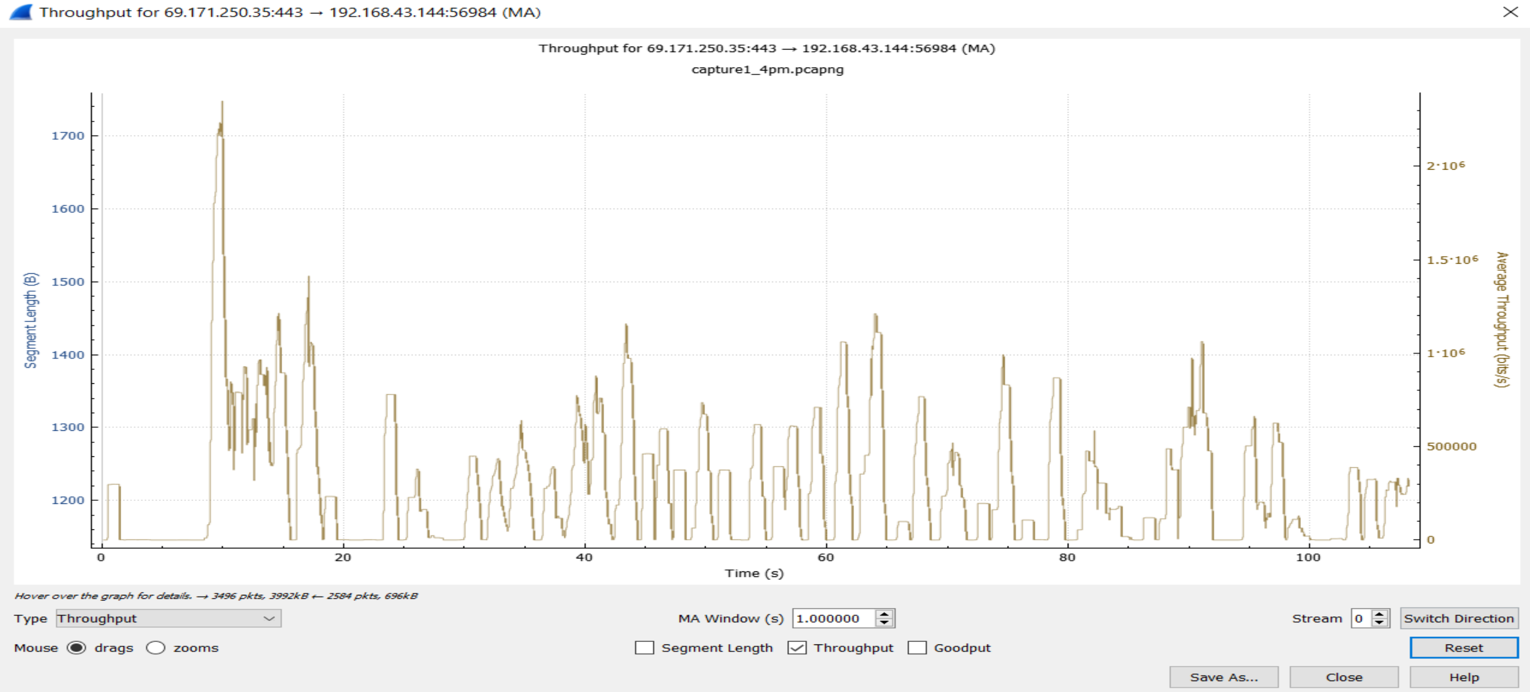
Statistics

Measurement	Captured	Displayed	Marked
Packets	6025	6025 (100.0%)	—
Time span, s	187.109	187.109	—
Average pps	32.2	32.2	—
Average packet size, B	850	850	—
Bytes	5124050	5124050 (100.0%)	0
Average bytes/s	27k	27k	—
Average bits/s	219k	219k	—

Average Throughput = **219k bits/s**

First capture

To view the distribution of average throughput in the whole time interval, received from Facebook servers to our system. Go to **“Statistics -> TCP Stream Graphs -> Throughput”**



Throughput for 69.171.250.35:443 → 192.168.43.144:56984 (MA)

Throughput for 69.171.250.35:443 → 192.168.43.144:56984 (MA)
capture1_4pm.pcapng

Hover over the graph for details. → 3496 pkts, 3992kB ← 2584 pkts, 696kB

Type Throughput

MA Window (s) 1.000000

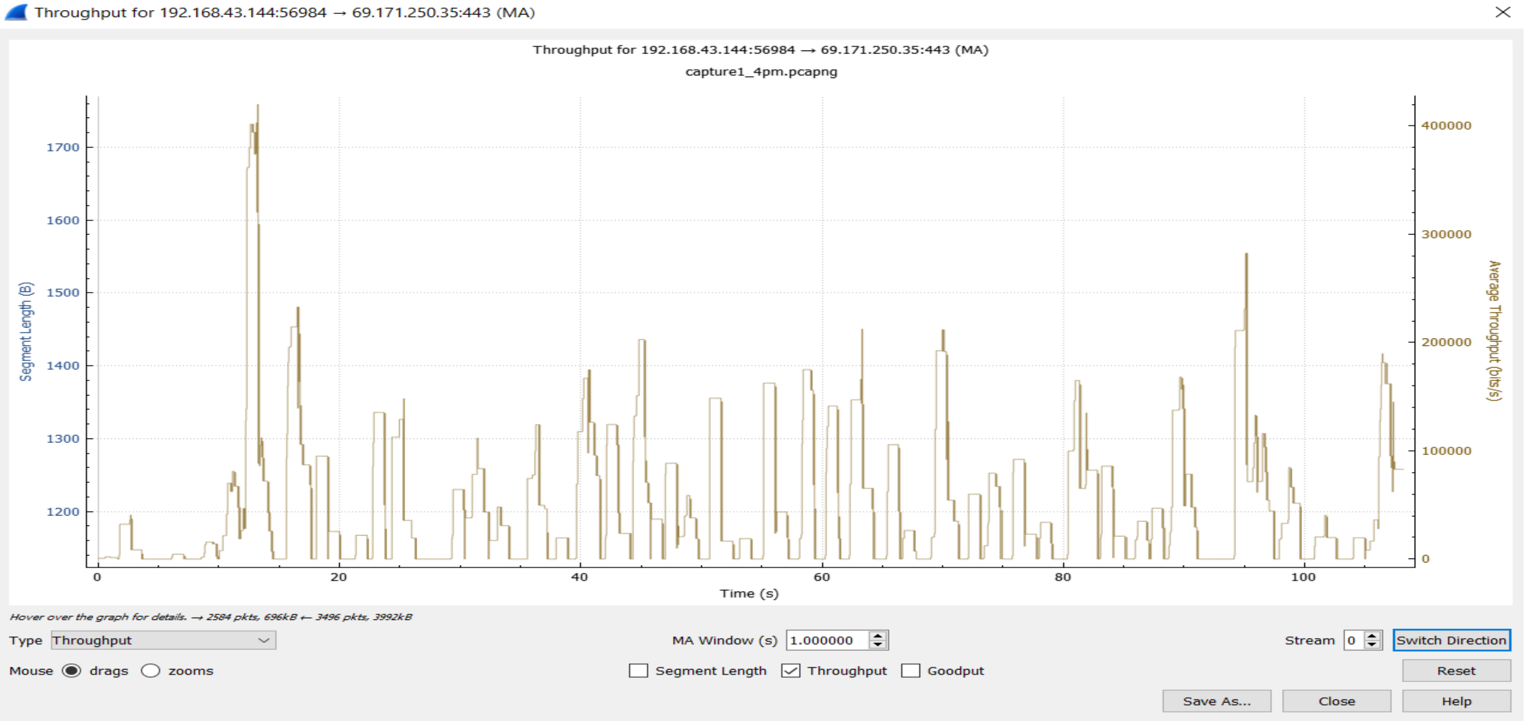
Stream 0

Mouse ● drags ○ zooms

□ Segment Length ☑ Throughput □ Goodput

Apart from intervals where avg. throghput is 0 (due to no scrolling while capture, this is also responsible lower avg. in total interval), avg. throughput ranges mostly from **2 x 10^5 bits/s to 1.3 x 10^6 bits/s**.

To view the distribution of average throughput in the whole time interval, sent from our system to Facebook servers. Click on **“Switch Direction”**



Throughput for 192.168.43.144:56984 → 69.171.250.35:443 (MA)

Throughput for 192.168.43.144:56984 → 69.171.250.35:443 (MA)
capture1_4pm.pcapng

Hover over the graph for details. → 2584 pkts, 696kB ← 3496 pkts, 3992kB

Type Throughput

MA Window (s) 1.000000

Stream 0

Mouse ● drags ○ zooms

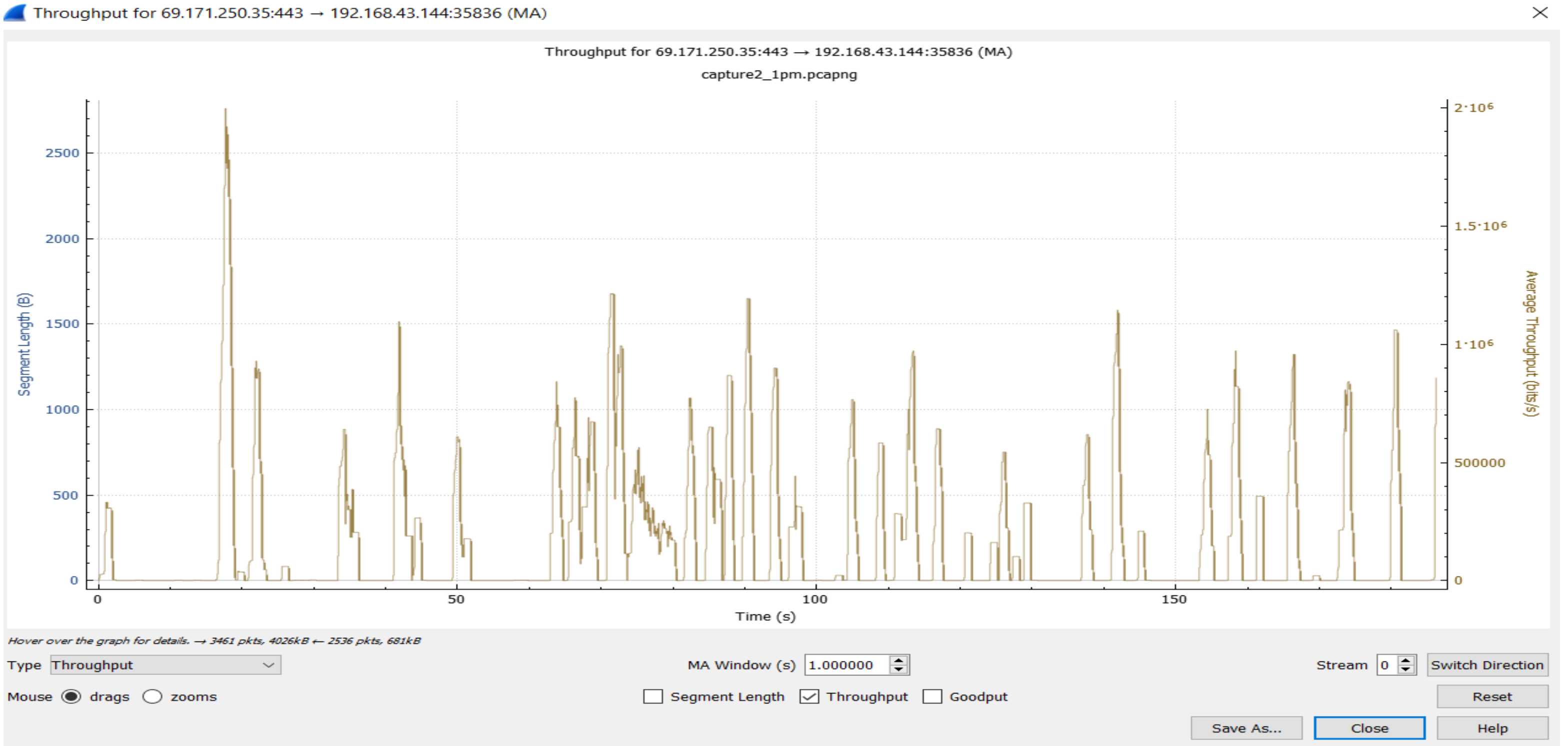
□ Segment Length ☑ Throughput □ Goodput

Apart from intervals where avg. throghput is 0 (due to no scrolling while capture), avg. throughput ranges mostly from **2 x 10^4 bits/s to 3 x 10^5 bits/s**.

Since upload speeds are much slower than download speeds (in general internet connections), we can notice throughput is also higher when packets are received from Facebook to system rathen than when sent to Facebook from system.

Second Capture

To view the distribution of average throughput in the whole time interval, received from Facebook servers to our system. Go to **“Statistics -> TCP Stream Graphs -> Throughput”**



Throughput for 69.171.250.35:443 → 192.168.43.144:35836 (MA)

Hover over the graph for details. → 3461 pkts, 4026kB ← 2536 pkts, 681kB

Type Throughput

MA Window (s) 1.000000

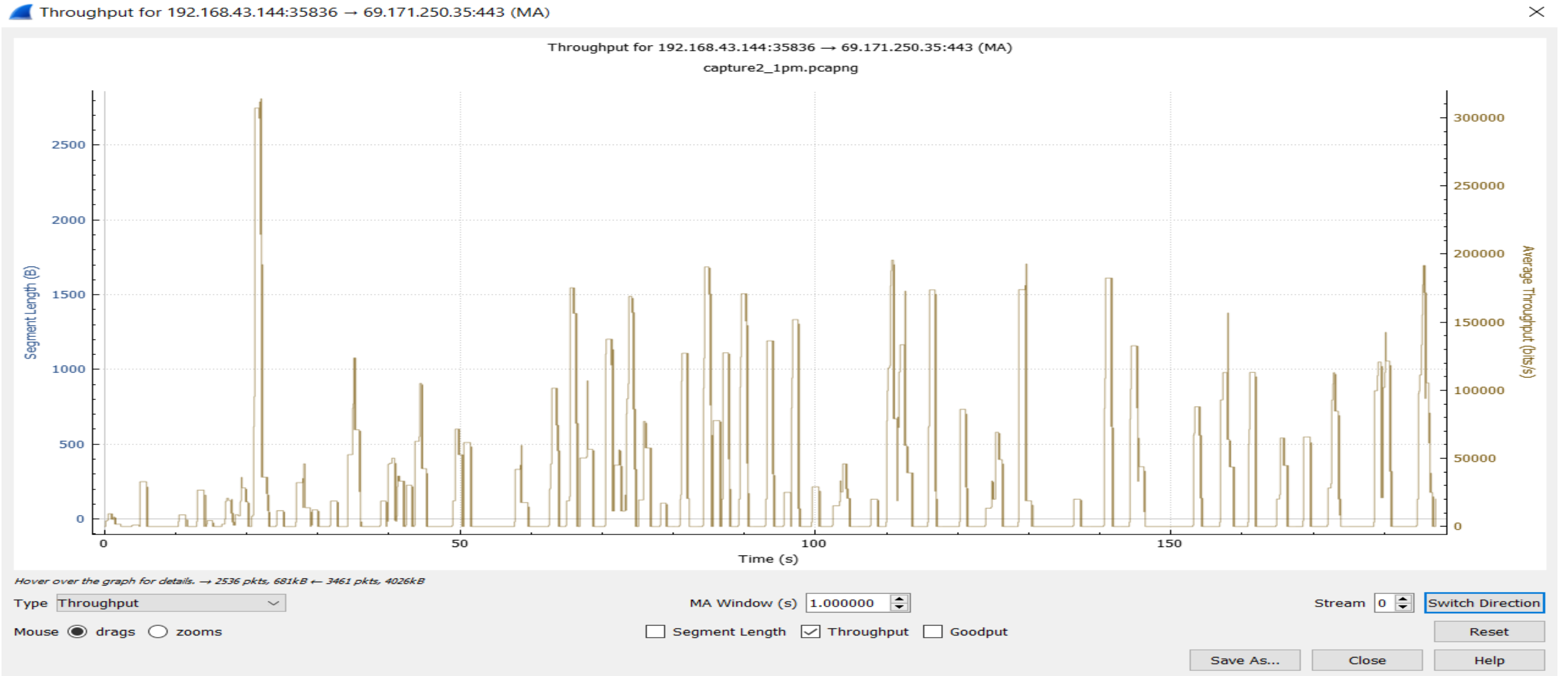
Stream 0

Mouse ● drags ○ zooms

□ Segment Length ☑ Throughput □ Goodput

Apart from intervals where avg. throghput is 0 (due to no scrolling while capture, this is also responsible lower avg. in total interval), avg. throughput ranges mostly from **2.5 x 10^5 bits/s to 1.25 x 10^6 bits/s**.

To view the distribution of average throughput in the whole time interval, sent from our system to Facebook servers. Click on **“Switch Direction”**



Throughput for 192.168.43.144:35836 → 69.171.250.35:443 (MA)

Hover over the graph for details. → 2536 pkts, 681kB ← 3461 pkts, 4026kB

Type Throughput

MA Window (s) 1.000000

Stream 0

Mouse ● drags ○ zooms

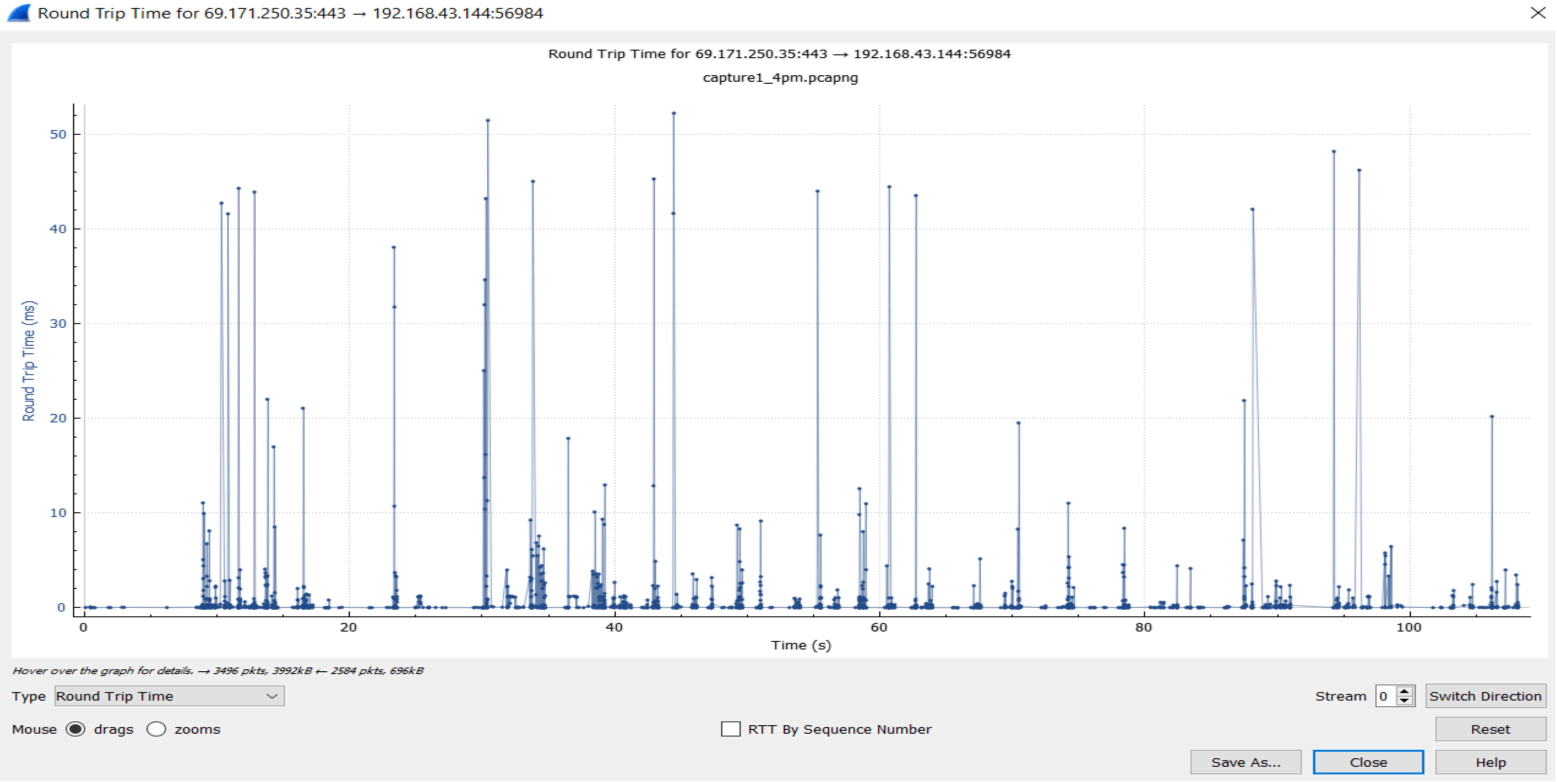
□ Segment Length ☑ Throughput □ Goodput

Apart from intervals where avg. throghput is 0 (due to no scrolling while capture), avg. throughput ranges mostly from **2.5 x 10^4 bits/s to 1.8 x 10^5 bits/s**.

Since upload speeds are much slower than download speeds (in general internet connections), we can notice throughput is also higher when packets are received from Facebook to system rathen than when sent to Facebook from system.

Round Trip Time (RTT) : Round Trip Time (RTT) is the 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 source. It Is possible to plot the graphs similar for RRT of packets in Wireshark. Go to **"Statistics -> TCP Stream Graphs -> Round Trip Time"**

First Capture: Plot of RRT for Packets received from Facebook Servers to our system, RTT Range for most packets: 0 to 50 ms (faster)



Round Trip Time for 69.171.250.35:443 → 192.168.43.144:56984

Round Trip Time for 69.171.250.35:443 → 192.168.43.144:56984

capture1_4pm.pcapng

Hover over the graph for details. → 3496 pkts, 3992kB ← 2584 pkts, 696kB

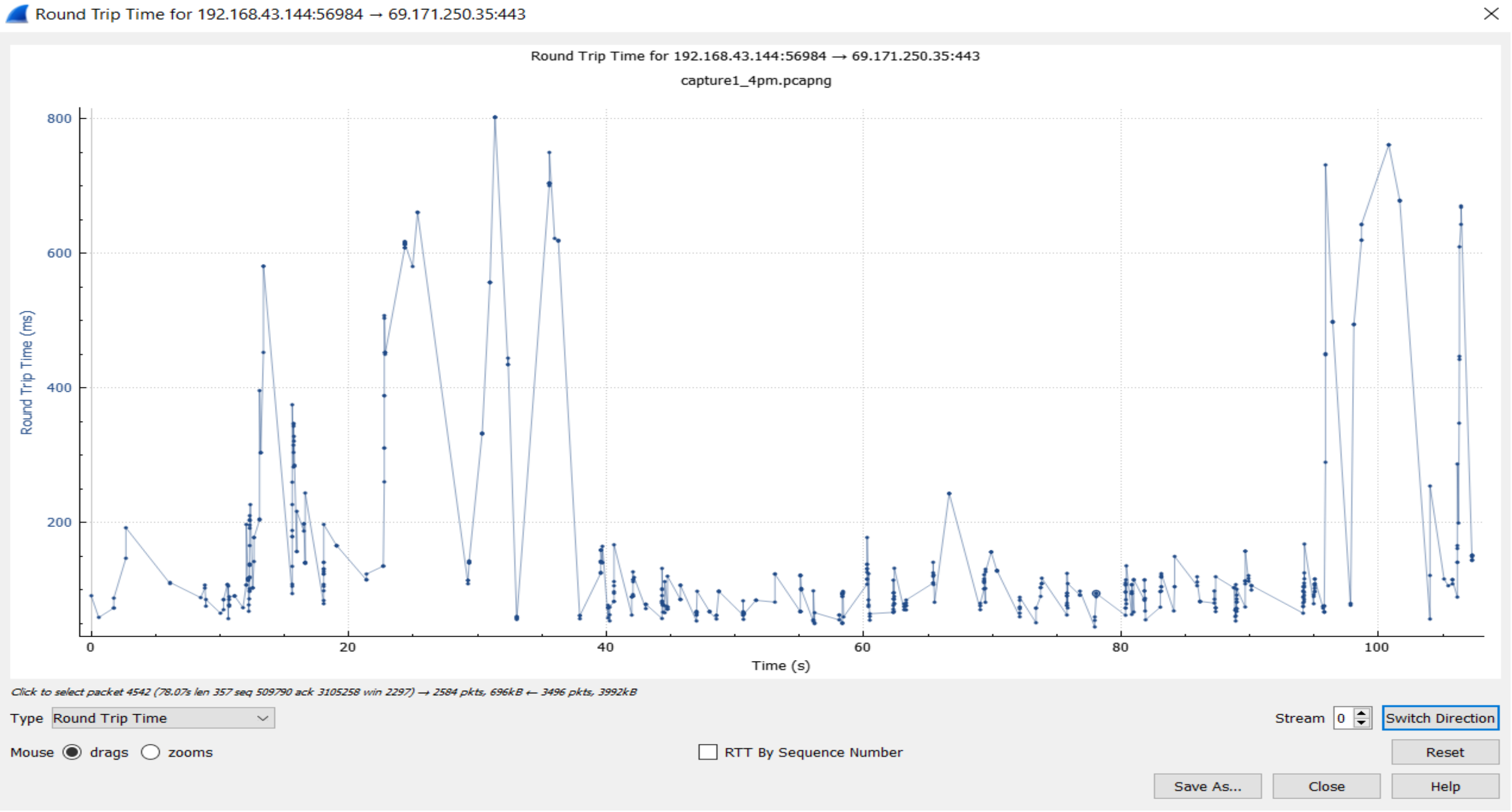
Type Round Trip Time

Stream 0

Mouse ● drags ○ zooms

□ RTT By Sequence Number

Plot of RRT for Packets sent from our system to Facebook Servers RTT Range for most packets: 0 to 800 ms (slower)



Round Trip Time for 192.168.43.144:56984 → 69.171.250.35:443

Round Trip Time for 192.168.43.144:56984 → 69.171.250.35:443

capture1_4pm.pcapng

Click to select packet 4542 (78.07s len 357 seq 509790 ack 3105258 win 2297) → 2584 pkts, 696kB ← 3496 pkts, 3992kB

Type Round Trip Time

Stream 0

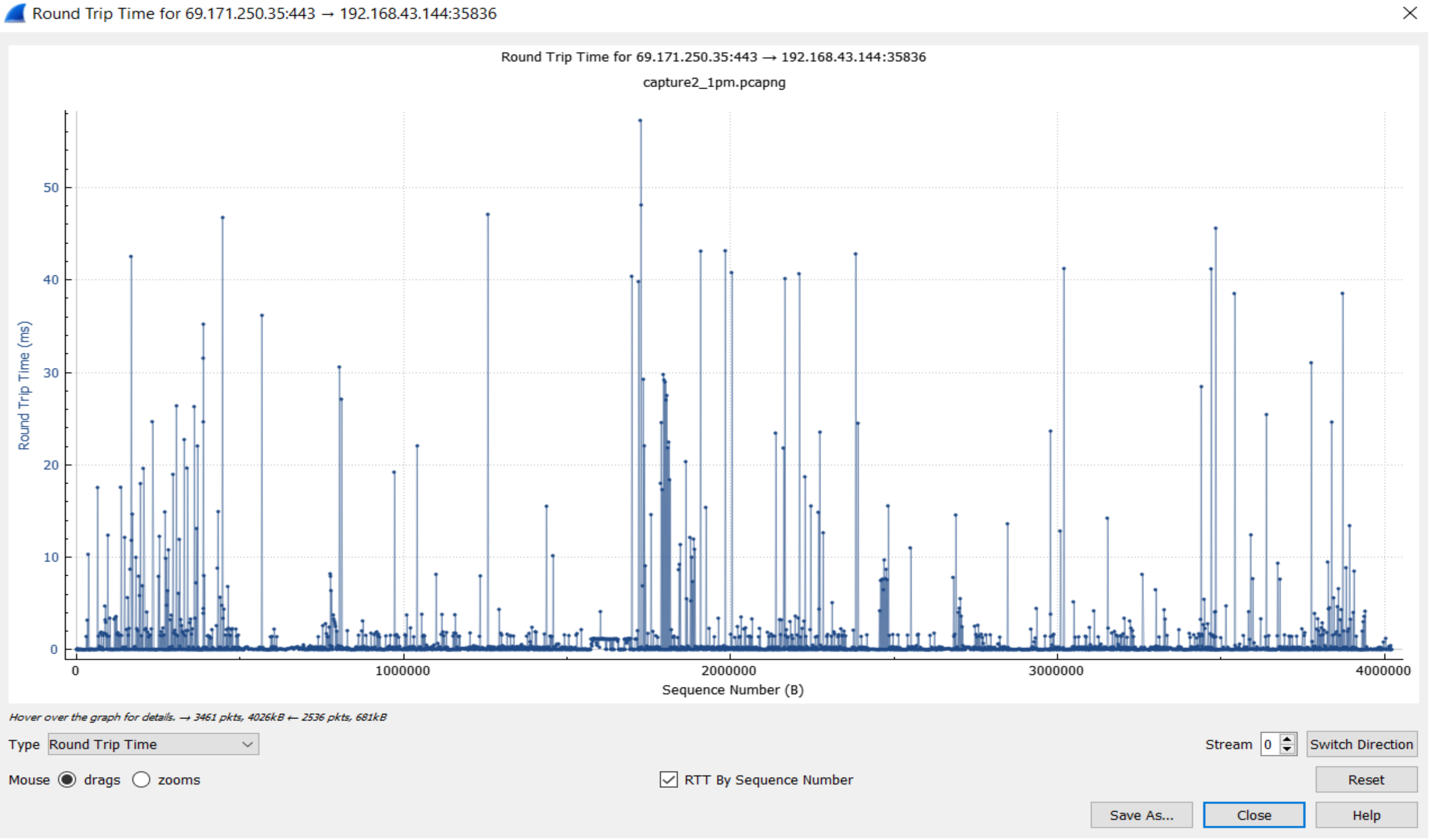
Mouse ● drags ○ zooms

□ RTT By Sequence Number

Second Capture:

Plot of RRT for Packets received from Facebook Servers to our system,

RTT Range for most packets: 0 to 50 ms (faster)



Round Trip Time for 69.171.250.35:443 → 192.168.43.144:35836

Round Trip Time for 69.171.250.35:443 → 192.168.43.144:35836

capture2_1pm.pcapng

Hover over the graph for details. → 3461 pkts, 4026kB ← 2536 pkts, 681kB

Type Round Trip Time

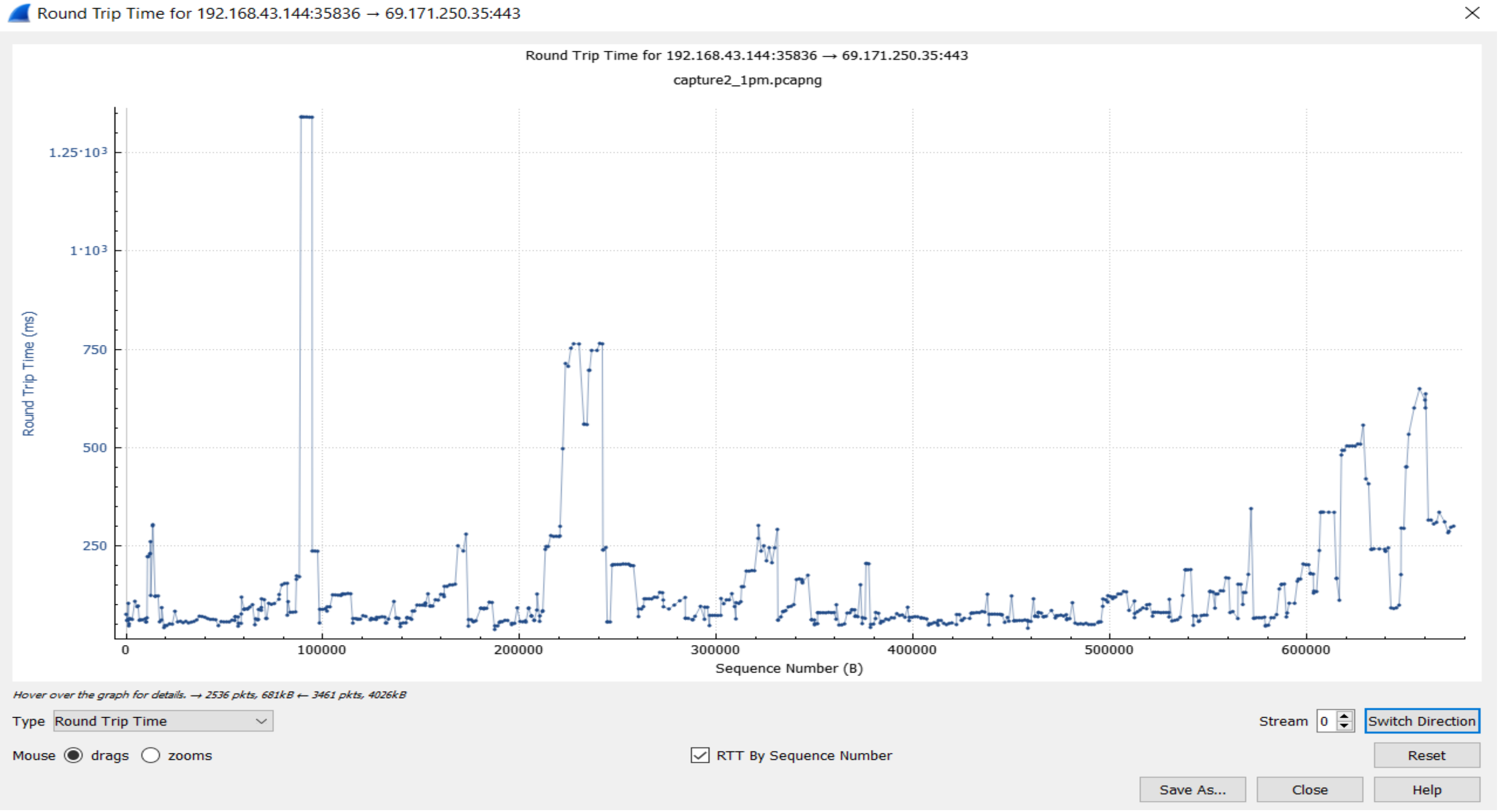
Stream 0

Mouse ● drags ○ zooms

☑ RTT By Sequence Number

Plot of RRT for Packets sent from our system to Facebook Servers

RTT Range for most packets: 0 to 800 ms (slower)



Round Trip Time for 192.168.43.144:35836 → 69.171.250.35:443

Round Trip Time for 192.168.43.144:35836 → 69.171.250.35:443

capture2_1pm.pcapng

Hover over the graph for details. → 2536 pkts, 681kB ← 3461 pkts, 4026kB

Type Round Trip Time

Stream 0

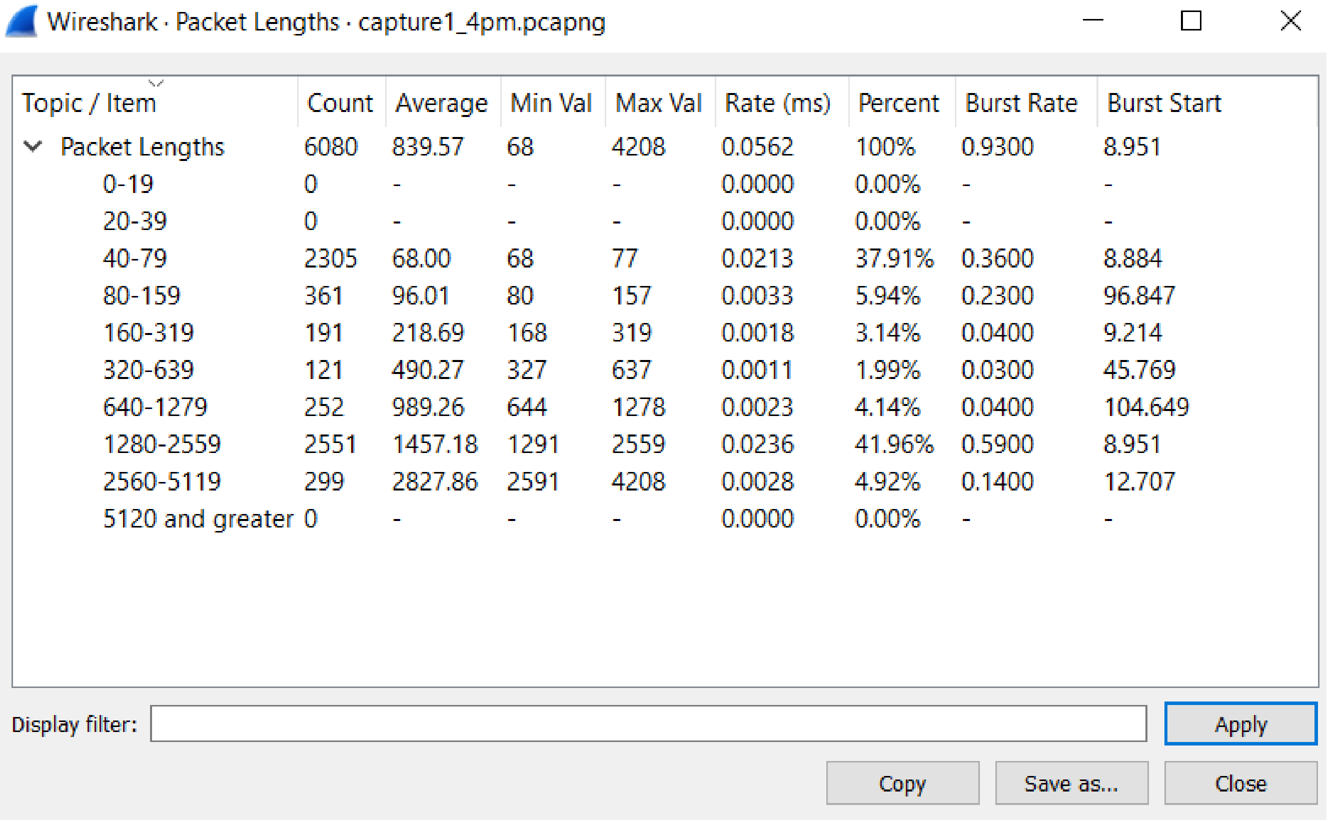
Mouse ● drags ○ zooms

☑ RTT By Sequence Number

Packet Length

In Wireshark, the amount of data carried by packets can be displayed using **"Statistics -> Packet lengths"**

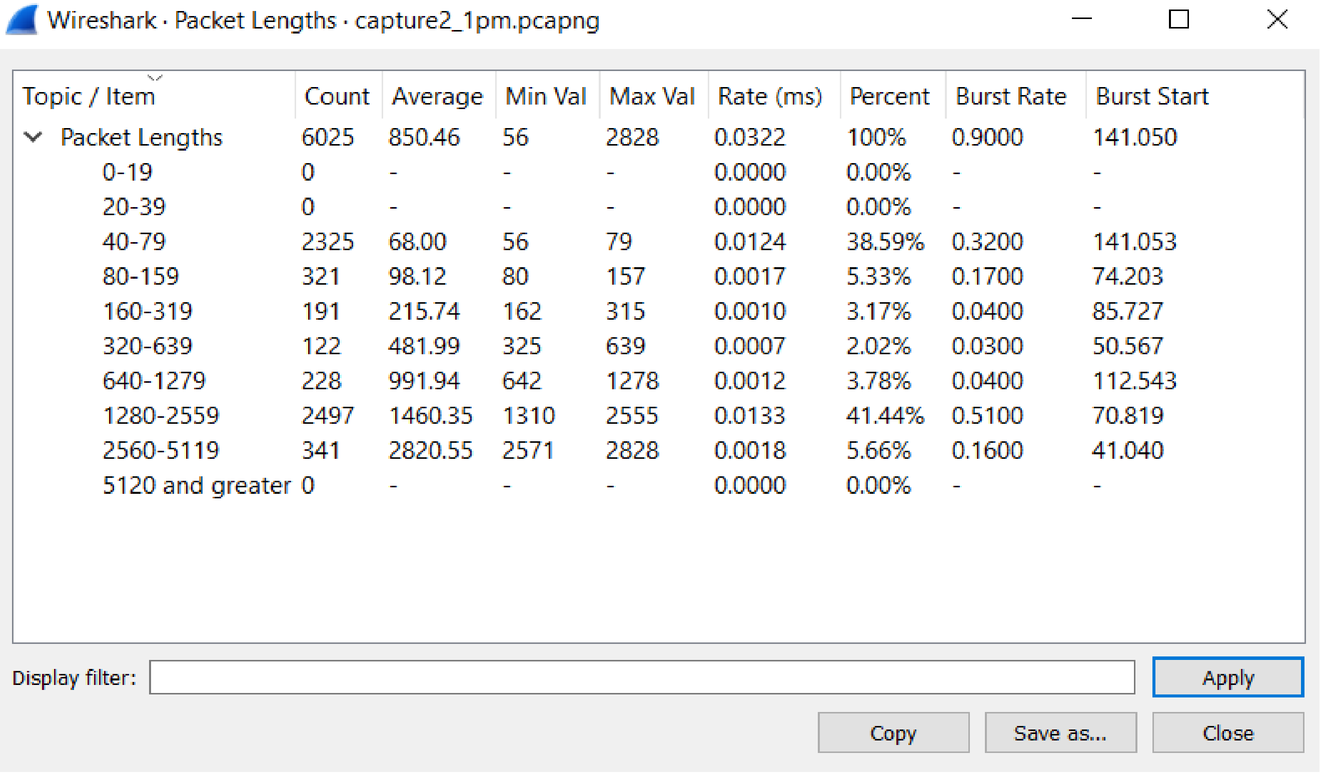
First Capture:



Topic / Item	Count	Average	Min Val	Max Val	Rate (ms)	Percent	Burst Rate	Burst Start
Packet Lengths	6080	839.57	68	4208	0.0562	100%	0.9300	8.951
0-19	0	-	-	-	0.0000	0.00%	-	-
20-39	0	-	-	-	0.0000	0.00%	-	-
40-79	2305	68.00	68	77	0.0213	37.91%	0.3600	8.884
80-159	361	96.01	80	157	0.0033	5.94%	0.2300	96.847
160-319	191	218.69	168	319	0.0018	3.14%	0.0400	9.214
320-639	121	490.27	327	637	0.0011	1.99%	0.0300	45.769
640-1279	252	989.26	644	1278	0.0023	4.14%	0.0400	104.649
1280-2559	2551	1457.18	1291	2559	0.0236	41.96%	0.5900	8.951
2560-5119	299	2827.86	2591	4208	0.0028	4.92%	0.1400	12.707
5120 and greater	0	-	-	-	0.0000	0.00%	-	-

Average Packet Length: 839.57 bytes

Second Capture:



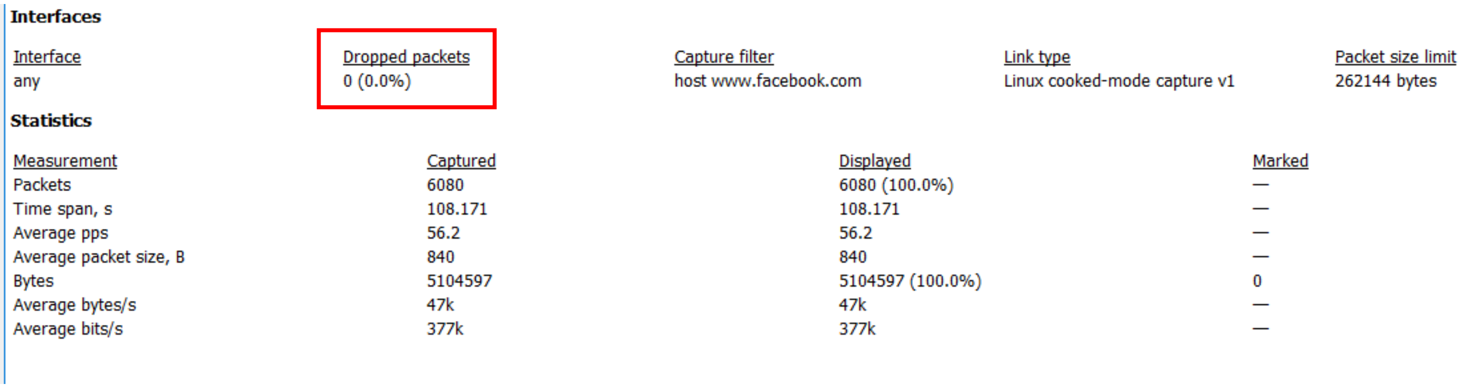
Topic / Item	Count	Average	Min Val	Max Val	Rate (ms)	Percent	Burst Rate	Burst Start
Packet Lengths	6025	850.46	56	2828	0.0322	100%	0.9000	141.050
0-19	0	-	-	-	0.0000	0.00%	-	-
20-39	0	-	-	-	0.0000	0.00%	-	-
40-79	2325	68.00	56	79	0.0124	38.59%	0.3200	141.053
80-159	321	98.12	80	157	0.0017	5.33%	0.1700	74.203
160-319	191	215.74	162	315	0.0010	3.17%	0.0400	85.727
320-639	122	481.99	325	639	0.0007	2.02%	0.0300	50.567
640-1279	228	991.94	642	1278	0.0012	3.78%	0.0400	112.543
1280-2559	2497	1460.35	1310	2555	0.0133	41.44%	0.5100	70.819
2560-5119	341	2820.55	2571	2828	0.0018	5.66%	0.1600	41.040
5120 and greater	0	-	-	-	0.0000	0.00%	-	-

Average Packet Length: 850.46 bytes

Packets Lost

We can check the number of packets lost using the **“Statistics -> Capture File Properties”** Summary shows many other details including Dropped packets.

First Capture:



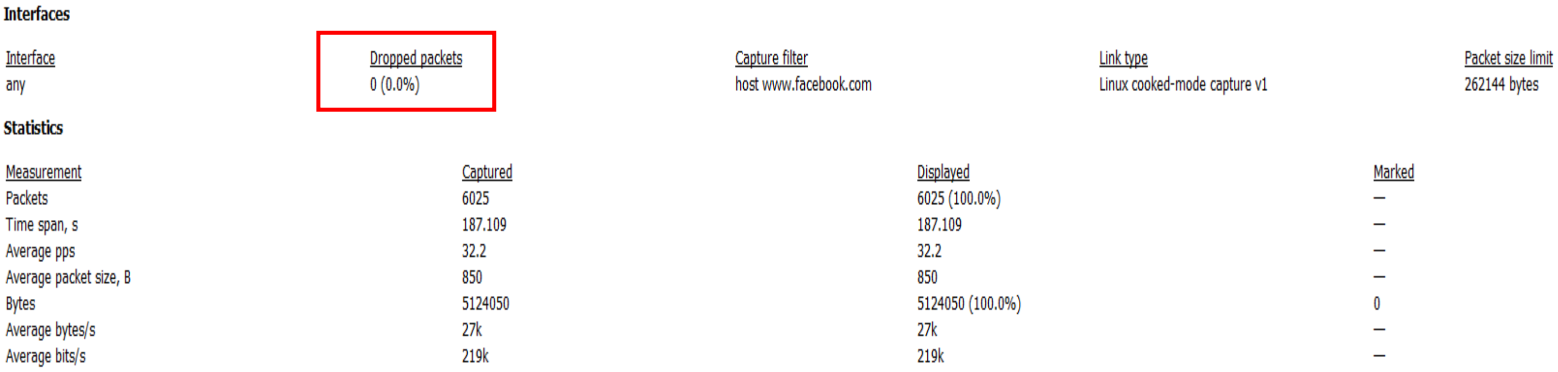
Interfaces

Interface	Dropped packets	Capture filter	Link type	Packet size limit
any	0 (0.0%)	host www.facebook.com	Linux cooked-mode capture v1	262144 bytes

Statistics

Measurement	Captured	Displayed	Marked
Packets	6080	6080 (100.0%)	—
Time span, s	108.171	108.171	—
Average pps	56.2	56.2	—
Average packet size, B	840	840	—
Bytes	5104597	5104597 (100.0%)	0
Average bytes/s	47k	47k	—
Average bits/s	377k	377k	—

Second Capture:



Interfaces

Interface	Dropped packets	Capture filter	Link type	Packet size limit
any	0 (0.0%)	host www.facebook.com	Linux cooked-mode capture v1	262144 bytes

Statistics

Measurement	Captured	Displayed	Marked
Packets	6025	6025 (100.0%)	—
Time span, s	187.109	187.109	—
Average pps	32.2	32.2	—
Average packet size, B	850	850	—
Bytes	5124050	5124050 (100.0%)	0
Average bytes/s	27k	27k	—
Average bits/s	219k	219k	—

Due to reliable connection, we have 0 packets dropped/lost in both the cases

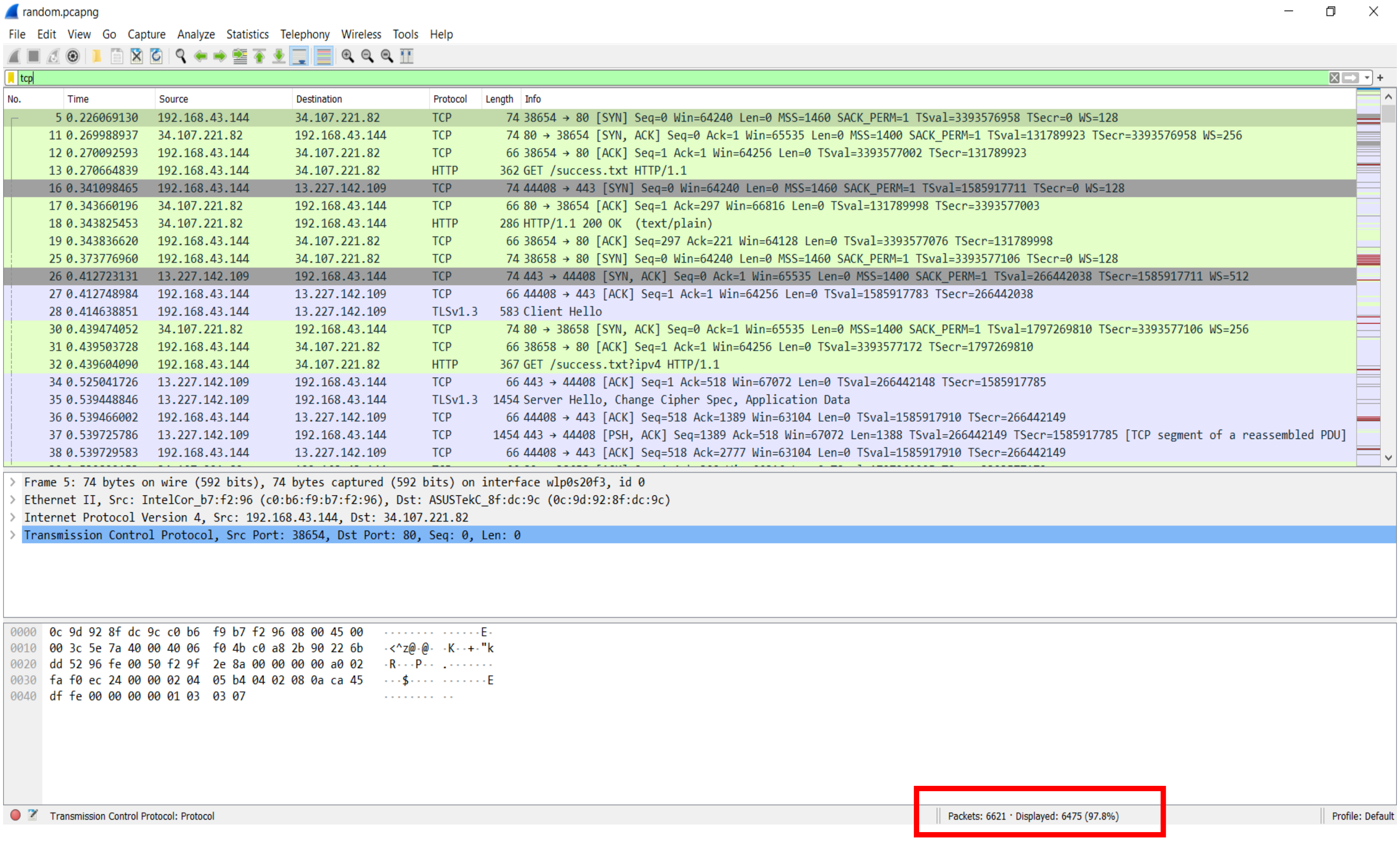
Count of TCP and UDP Packets

Captured Packets from www.facebook.com contain only TCP Packets, there isn't much use of this data from now on. Therefore, for the experiment of Capturing both TCP Packets and UDP Packets, we remove the capture filter and collect general data from web surfing.

TCP

Capture Filename: random.pcapng

Display Filter: tcp



random.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
tcp

```
No.   Time         Source          Destination     Protocol Length Info
    5 0.226069130  192.168.43.144  34.107.221.82   TCP        74 38654 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=3393576958 TSecr=0 WS=128
   11 0.269988937  34.107.221.82   192.168.43.144  TCP        74 80 → 38654 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1400 SACK_PERM=1 TSval=131789923 TSecr=3393576958 WS=256
   12 0.270092593  192.168.43.144  34.107.221.82   TCP        66 38654 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=3393577002 TSecr=131789923
   13 0.270664839  192.168.43.144  34.107.221.82   HTTP      362 GET /success.txt HTTP/1.1
   16 0.341098465  192.168.43.144  13.227.142.109  TCP        74 44408 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=1585917711 TSecr=0 WS=128
   17 0.343660196  34.107.221.82   192.168.43.144  TCP        66 80 → 38654 [ACK] Seq=1 Ack=297 Win=66816 Len=0 TSval=131789998 TSecr=3393577003
   18 0.343825453  34.107.221.82   192.168.43.144  HTTP      286 HTTP/1.1 200 OK  (text/plain)
   19 0.343836620  192.168.43.144  34.107.221.82   TCP        66 38654 → 80 [ACK] Seq=297 Ack=221 Win=64128 Len=0 TSval=3393577076 TSecr=131789998
   25 0.373776960  192.168.43.144  34.107.221.82   TCP        74 38658 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=3393577106 TSecr=0 WS=128
   26 0.412723131  13.227.142.109  192.168.43.144  TCP        74 443 → 44408 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1400 SACK_PERM=1 TSval=266442038 TSecr=1585917711 WS=512
   27 0.412748984  192.168.43.144  13.227.142.109  TCP        66 44408 → 443 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=1585917783 TSecr=266442038
   28 0.414638851  192.168.43.144  13.227.142.109  TLSv1.3   583 Client Hello
   30 0.439474052  34.107.221.82   192.168.43.144  TCP        74 80 → 38658 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1400 SACK_PERM=1 TSval=1797269810 TSecr=3393577106 WS=256
   31 0.439503728  192.168.43.144  34.107.221.82   TCP        66 38658 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=3393577172 TSecr=1797269810
   32 0.439604090  192.168.43.144  34.107.221.82   HTTP      367 GET /success.txt?ipv4 HTTP/1.1
   34 0.525041726  13.227.142.109  192.168.43.144  TCP        66 443 → 44408 [ACK] Seq=1 Ack=518 Win=67072 Len=0 TSval=266442148 TSecr=1585917785
   35 0.539448846  13.227.142.109  192.168.43.144  TLSv1.3  1454 Server Hello, Change Cipher Spec, Application Data
   36 0.539466002  192.168.43.144  13.227.142.109  TCP        66 44408 → 443 [ACK] Seq=518 Ack=1389 Win=63104 Len=0 TSval=1585917910 TSecr=266442149
   37 0.539725786  13.227.142.109  192.168.43.144  TCP      1454 443 → 44408 [PSH, ACK] Seq=1389 Ack=518 Win=67072 Len=1388 TSval=266442149 TSecr=1585917785 [TCP segment of a reassembled PDU]
   38 0.539729583  192.168.43.144  13.227.142.109  TCP        66 44408 → 443 [ACK] Seq=518 Ack=2777 Win=63104 Len=0 TSval=1585917910 TSecr=266442149
```

```
> Frame 5: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface wlp0s20f3, id 0
> Ethernet II, Src: IntelCor_b7:f2:96 (c0:b6:f9:b7:f2:96), Dst: ASUSTekC_8f:dc:9c (0c:9d:92:8f:dc:9c)
> Internet Protocol Version 4, Src: 192.168.43.144, Dst: 34.107.221.82
> Transmission Control Protocol, Src Port: 38654, Dst Port: 80, Seq: 0, Len: 0
```

```
0000  0c 9d 92 8f dc 9c c0 b6  f9 b7 f2 96 08 00 45 00   ········ ······E·
0010  00 3c 5e 7a 40 00 40 06  f0 4b c0 a8 2b 90 22 6b   ·<^z@·@· ·K··+·"k
0020  dd 52 96 fe 00 50 f2 9f  2e 8a 00 00 00 00 a0 02   ·R···P·· .·······
0030  fa f0 ec 24 00 00 02 04  05 b4 04 02 08 0a ca 45   ···$···· ·······E
0040  df fe 00 00 00 00 01 03  03 07                     ········ ··
```

```
Transmission Control Protocol: Protocol        Packets: 6621 · Displayed: 6475 (97.8%)        Profile: Default
```

Total packets captured = 6621

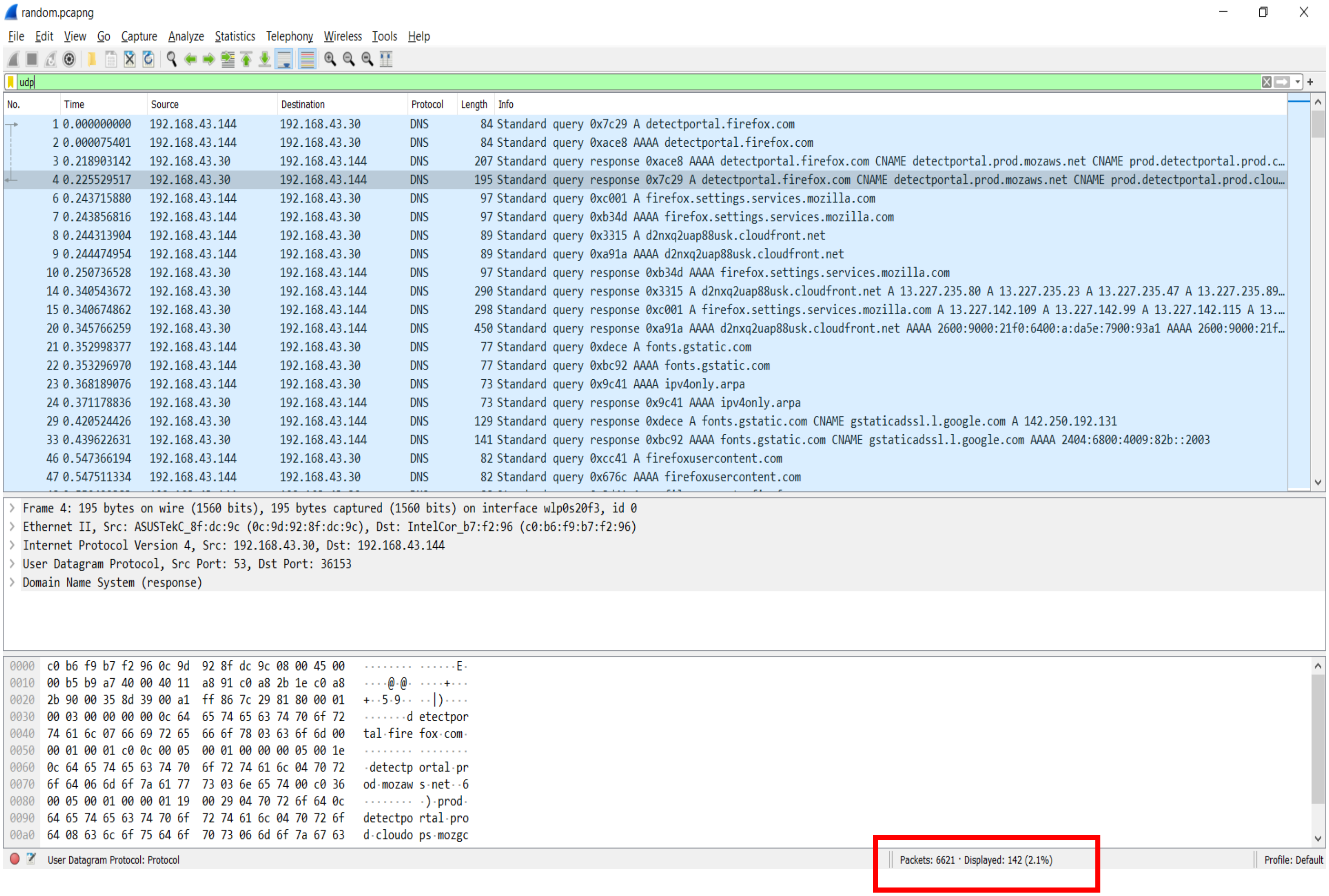
Number of TCP packets = 6475

Percentage of TCP packets = 97.8 %

UDP

Capture Filename: random.pcapng

Display Filter: udp


random.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
udp

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.43.144	192.168.43.30	DNS	84	Standard query 0x7c29 A detectportal.firefox.com
2	0.000075401	192.168.43.144	192.168.43.30	DNS	84	Standard query 0xace8 AAAA detectportal.firefox.com
3	0.218903142	192.168.43.30	192.168.43.144	DNS	207	Standard query response 0xace8 AAAA detectportal.firefox.com CNAME detectportal.prod.mozaws.net CNAME prod.detectportal.prod.c…
4	0.225529517	192.168.43.30	192.168.43.144	DNS	195	Standard query response 0x7c29 A detectportal.firefox.com CNAME detectportal.prod.mozaws.net CNAME prod.detectportal.prod.clou…
6	0.243715880	192.168.43.144	192.168.43.30	DNS	97	Standard query 0xc001 A firefox.settings.services.mozilla.com
7	0.243856816	192.168.43.144	192.168.43.30	DNS	97	Standard query 0xb34d AAAA firefox.settings.services.mozilla.com
8	0.244313904	192.168.43.144	192.168.43.30	DNS	89	Standard query 0x3315 A d2nxq2uap88usk.cloudfront.net
9	0.244474954	192.168.43.144	192.168.43.30	DNS	89	Standard query 0xa91a AAAA d2nxq2uap88usk.cloudfront.net
10	0.250736528	192.168.43.30	192.168.43.144	DNS	97	Standard query response 0xb34d AAAA firefox.settings.services.mozilla.com
14	0.340543672	192.168.43.30	192.168.43.144	DNS	290	Standard query response 0x3315 A d2nxq2uap88usk.cloudfront.net A 13.227.235.80 A 13.227.235.23 A 13.227.235.47 A 13.227.235.89…
15	0.340674862	192.168.43.30	192.168.43.144	DNS	298	Standard query response 0xc001 A firefox.settings.services.mozilla.com A 13.227.142.109 A 13.227.142.99 A 13.227.142.115 A 13.…
20	0.345766259	192.168.43.30	192.168.43.144	DNS	450	Standard query response 0xa91a AAAA d2nxq2uap88usk.cloudfront.net AAAA 2600:9000:21f0:6400:a:da5e:7900:93a1 AAAA 2600:9000:21f…
21	0.352998377	192.168.43.144	192.168.43.30	DNS	77	Standard query 0xdece A fonts.gstatic.com
22	0.353296970	192.168.43.144	192.168.43.30	DNS	77	Standard query 0xbc92 AAAA fonts.gstatic.com
23	0.368189076	192.168.43.144	192.168.43.30	DNS	73	Standard query 0x9c41 AAAA ipv4only.arpa
24	0.371178836	192.168.43.30	192.168.43.144	DNS	73	Standard query response 0x9c41 AAAA ipv4only.arpa
29	0.420524426	192.168.43.30	192.168.43.144	DNS	129	Standard query response 0xdece A fonts.gstatic.com CNAME gstaticadssl.l.google.com A 142.250.192.131
33	0.439622631	192.168.43.30	192.168.43.144	DNS	141	Standard query response 0xbc92 AAAA fonts.gstatic.com CNAME gstaticadssl.l.google.com AAAA 2404:6800:4009:82b::2003
46	0.547366194	192.168.43.144	192.168.43.30	DNS	82	Standard query 0xcc41 A firefoxusercontent.com
47	0.547511334	192.168.43.144	192.168.43.30	DNS	82	Standard query 0x676c AAAA firefoxusercontent.com

```
> Frame 4: 195 bytes on wire (1560 bits), 195 bytes captured (1560 bits) on interface wlp0s20f3, id 0
> Ethernet II, Src: ASUSTekC_8f:dc:9c (0c:9d:92:8f:dc:9c), Dst: IntelCor_b7:f2:96 (c0:b6:f9:b7:f2:96)
> Internet Protocol Version 4, Src: 192.168.43.30, Dst: 192.168.43.144
> User Datagram Protocol, Src Port: 53, Dst Port: 36153
> Domain Name System (response)
```

```
0000  c0 b6 f9 b7 f2 96 0c 9d  92 8f dc 9c 08 00 45 00   ········ ······E·
0010  00 b5 b9 a7 40 00 40 11  a8 91 c0 a8 2b 1e c0 a8   ····@·@· ····+···
0020  2b 90 00 35 8d 39 00 a1  ff 86 7c 29 81 80 00 01   +··5·9·· ··|)····
0030  00 03 00 00 00 00 0c 64  65 74 65 63 74 70 6f 72   ·······d etectpor
0040  74 61 6c 07 66 69 72 65  66 6f 78 03 63 6f 6d 00   tal·fire fox·com·
0050  00 01 00 01 c0 0c 00 05  00 01 00 00 00 05 00 1e   ········ ········
0060  0c 64 65 74 65 63 74 70  6f 72 74 61 6c 04 70 72   ·detectp ortal·pr
0070  6f 64 06 6d 6f 7a 61 77  73 03 6e 65 74 00 c0 36   od·mozaw s·net··6
0080  00 05 00 01 00 00 01 19  00 29 04 70 72 6f 64 0c   ········ ·)·prod·
0090  64 65 74 65 63 74 70 6f  72 74 61 6c 04 70 72 6f   detectpo rtal·pro
00a0  64 08 63 6c 6f 75 64 6f  70 73 06 6d 6f 7a 67 63   d·cloudo ps·mozgc
```

User Datagram Protocol: Protocol

Packets: 6621 · Displayed: 142 (2.1%)

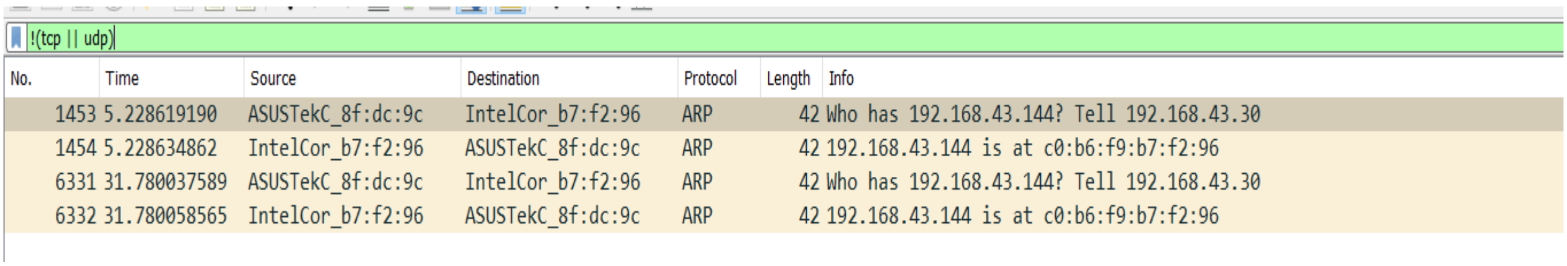
Profile: Default

Total packets captured = 6621

Number of TCP packets =142

Percentage of TCP packets = 2.1 %

Note: Remaining 4 packets are of ARP protocol



No.	Time	Source	Destination	Protocol	Length	Info
1453	5.228619190	ASUSTekC_8f:dc:9c	IntelCor_b7:f2:96	ARP	42	Who has 192.168.43.144? Tell 192.168.43.30
1454	5.228634862	IntelCor_b7:f2:96	ASUSTekC_8f:dc:9c	ARP	42	192.168.43.144 is at c0:b6:f9:b7:f2:96
6331	31.780037589	ASUSTekC_8f:dc:9c	IntelCor_b7:f2:96	ARP	42	Who has 192.168.43.144? Tell 192.168.43.30
6332	31.780058565	IntelCor_b7:f2:96	ASUSTekC_8f:dc:9c	ARP	42	192.168.43.144 is at c0:b6:f9:b7:f2:96

HTTP Request and Response

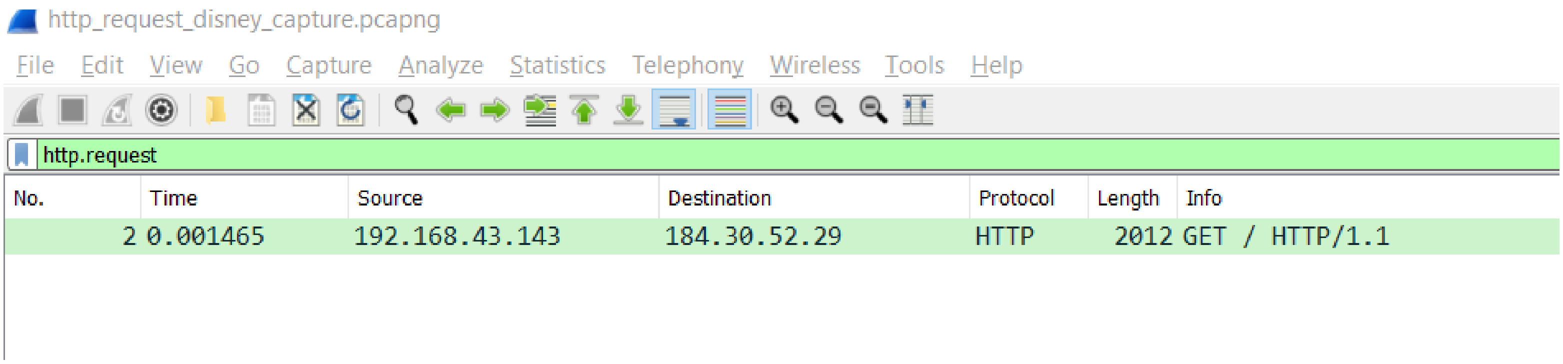
For this part, I have captured packets from a site using HTTP as an application layer protocol.

Capture Filename: **http_request_disney_capture.pcapng**

Capture Filter: **host www.disney.in**

Captured Packets using similar procedure as above

To get HTTP request among packets, use
Display Filter: **http.request**

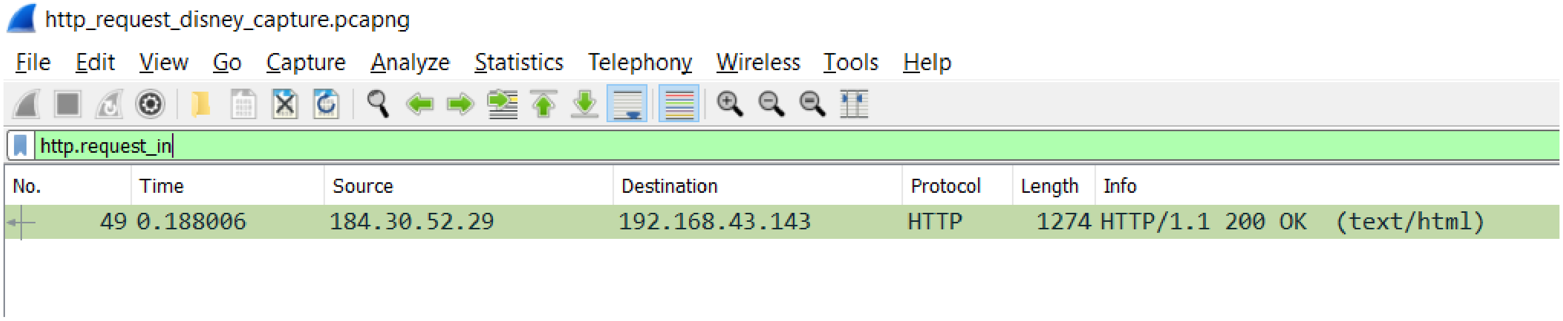


http_request_disney_capture.pcapng

http.request

No.	Time	Source	Destination	Protocol	Length	Info
2	0.001465	192.168.43.143	184.30.52.29	HTTP	2012	GET / HTTP/1.1

To get Response for the HTTP request, use
Display Filter: **http.request_in**



http_request_disney_capture.pcapng

http.request_in

No.	Time	Source	Destination	Protocol	Length	Info
49	0.188006	184.30.52.29	192.168.43.143	HTTP	1274	HTTP/1.1 200 OK (text/html)

So, for the above HTTP GET request we have got single response i.e. 200 OK