

CS359 Computer Networks - Assignment 11

Name: M Maheeth Reddy

Roll No.: 1801CS31

Date: 21-Apr-2021

Objective of the lab: Using Wireshark to calculate the following statistics:

- Throughput
- RTT
- Packet size
- number of packets lost
- number of UDP and TCP packets
- Number of responses received with respect to one request sent.

We need to perform this experiment at two different times in the day and report the observed values.

Procedure:

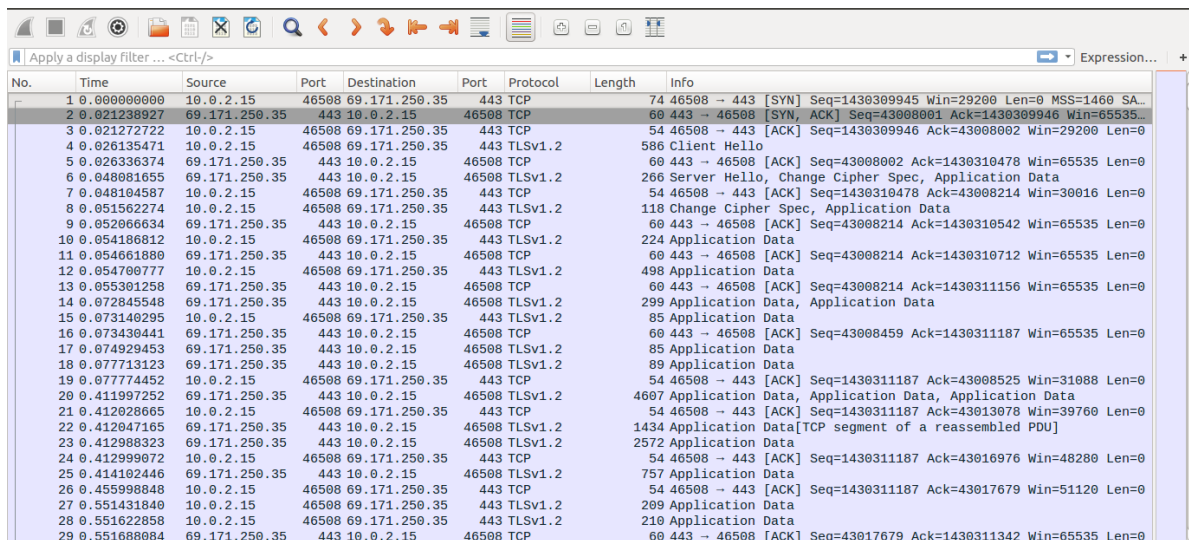
Step 1: Capturing the packets from Facebook using Wireshark

1. Find the IP address of Facebook by executing **ping www.facebook.com** in the terminal. I found the IP address to be 69.171.250.35
2. Open Wireshark
3. Select Network Interface and enter capture filter expression **host 69.171.250.35**
4. Start Packet Capture
5. Open the Web Browser and open www.facebook.com
6. After sufficient packets have been captured, stop the packet capture.

Step 2: Calculating the various metrics mentioned above for both the trials, using procedures described below.

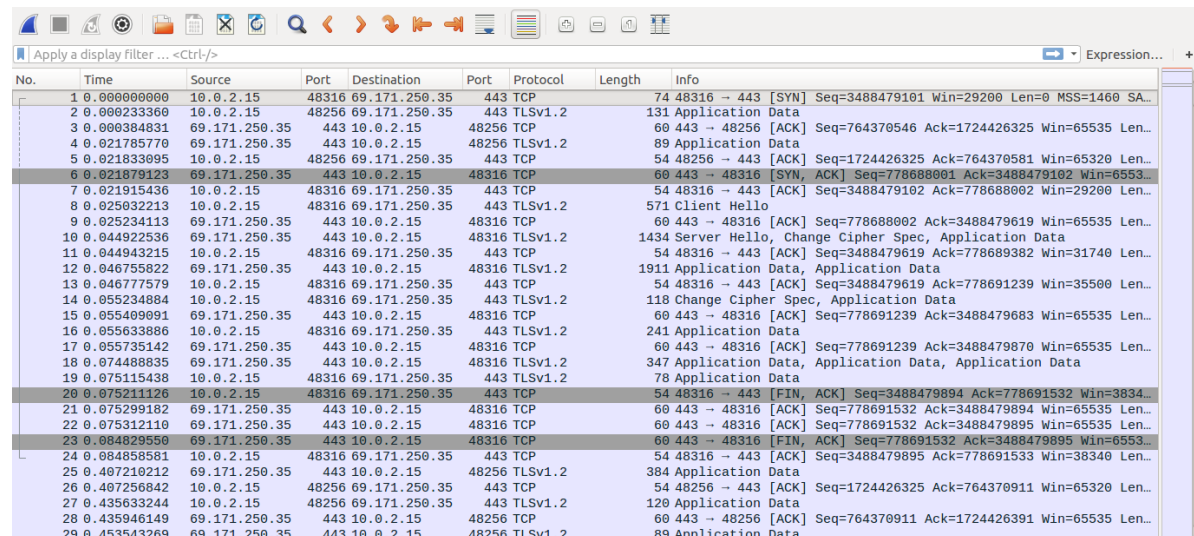
Experiment

Packet Capture for Trial 1 – 2317 packets have been captured



No.	Time	Source	Port	Destination	Port	Protocol	Length	Info
1	0.000000000	10.0.2.15	46508	69.171.250.35	443	TCP	74	69.171.250.35 → 443 [SYN] Seq=1430309945 Win=29200 Len=0 MSS=1460 SA=10.0.2.15
2	0.021238927	69.171.250.35	443	10.0.2.15	46508	TCP	60	443 → 46508 [SYN, ACK] Seq=43008001 Ack=1430309946 Win=65535 Len=0
3	0.021272722	10.0.2.15	46508	69.171.250.35	443	TCP	54	46508 → 443 [ACK] Seq=1430309946 Ack=43008002 Win=29200 Len=0
4	0.026135471	10.0.2.15	46508	69.171.250.35	443	TLSv1.2	586	Client Hello
5	0.026336374	69.171.250.35	443	10.0.2.15	46508	TCP	60	443 → 46508 [ACK] Seq=43008002 Ack=1430310478 Win=65535 Len=0
6	0.040801655	69.171.250.35	443	10.0.2.15	46508	TLSv1.2	266	Server Hello, Change Cipher Spec, Application Data
7	0.048104587	10.0.2.15	46508	69.171.250.35	443	TCP	54	46508 → 443 [ACK] Seq=1430310478 Ack=43008214 Win=30016 Len=0
8	0.051562274	10.0.2.15	46508	69.171.250.35	443	TLSv1.2	118	Change Cipher Spec, Application Data
9	0.052066634	69.171.250.35	443	10.0.2.15	46508	TCP	60	443 → 46508 [ACK] Seq=43008214 Ack=1430310542 Win=65535 Len=0
10	0.054186812	10.0.2.15	46508	69.171.250.35	443	TLSv1.2	224	Application Data
11	0.054661880	69.171.250.35	443	10.0.2.15	46508	TCP	60	443 → 46508 [ACK] Seq=43008214 Ack=1430310712 Win=65535 Len=0
12	0.054708777	10.0.2.15	46508	69.171.250.35	443	TLSv1.2	498	Application Data
13	0.055301258	69.171.250.35	443	10.0.2.15	46508	TCP	60	443 → 46508 [ACK] Seq=43008214 Ack=1430311156 Win=65535 Len=0
14	0.072845548	69.171.250.35	443	10.0.2.15	46508	TLSv1.2	299	Application Data, Application Data
15	0.073140295	10.0.2.15	46508	69.171.250.35	443	TLSv1.2	85	Application Data
16	0.073430441	69.171.250.35	443	10.0.2.15	46508	TCP	60	443 → 46508 [ACK] Seq=43008459 Ack=1430311187 Win=65535 Len=0
17	0.074929453	69.171.250.35	443	10.0.2.15	46508	TLSv1.2	85	Application Data
18	0.077713123	69.171.250.35	443	10.0.2.15	46508	TLSv1.2	89	Application Data
19	0.077774452	10.0.2.15	46508	69.171.250.35	443	TCP	54	46508 → 443 [ACK] Seq=1430311187 Ack=43008525 Win=31088 Len=0
20	0.411997252	69.171.250.35	443	10.0.2.15	46508	TLSv1.2	4607	Application Data, Application Data, Application Data
21	0.412028665	10.0.2.15	46508	69.171.250.35	443	TCP	54	46508 → 443 [ACK] Seq=1430311187 Ack=43013078 Win=39760 Len=0
22	0.412047165	69.171.250.35	443	10.0.2.15	46508	TLSv1.2	1434	Application Data[TCP segment of a reassembled PDU]
23	0.412988323	69.171.250.35	443	10.0.2.15	46508	TLSv1.2	2572	Application Data
24	0.412999072	10.0.2.15	46508	69.171.250.35	443	TCP	54	46508 → 443 [ACK] Seq=1430311187 Ack=43016976 Win=48280 Len=0
25	0.414102446	69.171.250.35	443	10.0.2.15	46508	TLSv1.2	757	Application Data
26	0.455998848	10.0.2.15	46508	69.171.250.35	443	TCP	54	46508 → 443 [ACK] Seq=1430311187 Ack=43017679 Win=51120 Len=0
27	0.551431840	10.0.2.15	46508	69.171.250.35	443	TLSv1.2	209	Application Data
28	0.551622858	10.0.2.15	46508	69.171.250.35	443	TLSv1.2	210	Application Data
29	0.551688084	69.171.250.35	443	10.0.2.15	46508	TCP	60	443 → 46508 [ACK] Seq=43017679 Ack=1430311342 Win=65535 Len=0

Packet Capture for Trial 2 – 3087 packets have been captured



No.	Time	Source	Port	Destination	Port	Protocol	Length	Info
1	0.000000000	10.0.2.15	48316	69.171.250.35	443	TCP	74	48316 → 443 [SYN] Seq=3488479101 Win=29200 Len=0 MSS=1460 SA...
2	0.000233360	10.0.2.15	48256	69.171.250.35	443	TLSv1.2	131	Application Data
3	0.000384831	69.171.250.35	443	10.0.2.15	48256	TCP	60	443 → 48256 [ACK] Seq=764370546 Ack=1724426325 Win=65535 Len...
4	0.021785770	69.171.250.35	443	10.0.2.15	48256	TLSv1.2	89	Application Data
5	0.021833895	10.0.2.15	48256	69.171.250.35	443	TCP	54	48256 → 443 [ACK] Seq=1724426325 Ack=764370581 Win=65320 Len...
6	0.021879123	69.171.250.35	443	10.0.2.15	48316	TCP	60	443 → 48316 [SYN, ACK] Seq=778688001 Ack=3488479102 Win=6553...
7	0.021915436	10.0.2.15	48316	69.171.250.35	443	TCP	54	48316 → 443 [ACK] Seq=3488479102 Ack=778688002 Win=29200 Len...
8	0.025032213	10.0.2.15	48316	69.171.250.35	443	TLSv1.2	571	Client Hello
9	0.025234113	69.171.250.35	443	10.0.2.15	48316	TCP	60	443 → 48316 [ACK] Seq=778688002 Ack=3488479619 Win=65535 Len...
10	0.044922536	69.171.250.35	443	10.0.2.15	48316	TLSv1.2	1434	Server Hello, Change Cipher Spec, Application Data
11	0.044943215	10.0.2.15	48316	69.171.250.35	443	TCP	54	48316 → 443 [ACK] Seq=3488479619 Ack=778689382 Win=31740 Len...
12	0.046755822	69.171.250.35	443	10.0.2.15	48316	TLSv1.2	1911	Application Data, Application Data
13	0.046777579	10.0.2.15	48316	69.171.250.35	443	TCP	54	48316 → 443 [ACK] Seq=3488479619 Ack=778691239 Win=35500 Len...
14	0.055234884	10.0.2.15	48316	69.171.250.35	443	TLSv1.2	118	Change Cipher Spec, Application Data
15	0.055409091	69.171.250.35	443	10.0.2.15	48316	TCP	60	443 → 48316 [ACK] Seq=778691239 Ack=3488479683 Win=65535 Len...
16	0.055633886	10.0.2.15	48316	69.171.250.35	443	TLSv1.2	241	Application Data
17	0.055735142	69.171.250.35	443	10.0.2.15	48316	TCP	60	443 → 48316 [ACK] Seq=778691239 Ack=3488479870 Win=65535 Len...
18	0.074488835	69.171.250.35	443	10.0.2.15	48316	TLSv1.2	347	Application Data, Application Data, Application Data
19	0.075115438	10.0.2.15	48316	69.171.250.35	443	TLSv1.2	78	Application Data
20	0.075211126	10.0.2.15	48316	69.171.250.35	443	TCP	54	48316 → 443 [FIN, ACK] Seq=3488479894 Ack=778691532 Win=3834...
21	0.075299182	69.171.250.35	443	10.0.2.15	48316	TCP	60	443 → 48316 [ACK] Seq=778691532 Ack=3488479894 Win=65535 Len...
22	0.075312110	69.171.250.35	443	10.0.2.15	48316	TCP	60	443 → 48316 [ACK] Seq=778691532 Ack=3488479895 Win=65535 Len...
23	0.084829550	69.171.250.35	443	10.0.2.15	48316	TCP	60	443 → 48316 [FIN, ACK] Seq=778691532 Ack=3488479895 Win=6553...
24	0.084859581	10.0.2.15	48316	69.171.250.35	443	TCP	54	48316 → 443 [ACK] Seq=3488479895 Ack=778691533 Win=38340 Len...
25	0.407210212	69.171.250.35	443	10.0.2.15	48256	TLSv1.2	384	Application Data
26	0.407256642	10.0.2.15	48256	69.171.250.35	443	TCP	54	48256 → 443 [ACK] Seq=1724426325 Ack=764370911 Win=65320 Len...
27	0.435633244	10.0.2.15	48256	69.171.250.35	443	TLSv1.2	120	Application Data
28	0.435946149	69.171.250.35	443	10.0.2.15	48256	TCP	60	443 → 48256 [ACK] Seq=764370911 Ack=1724426391 Win=65535 Len...
29	0.453543269	69.171.250.35	443	10.0.2.15	48256	TLSv1.2	89	Application Data

Measurement of Throughput:

Throughput is the amount of data is transmitted during a specified time period via a network, interface or channel. It is measured in bits/s or bytes/s.

Throughput for Trial 1:

After capturing packets using Wireshark, go to **Statistics > Capture File Properties**. A window opens up, which shows statistics of the packet capture. I have put the screenshot here for Trial 1.

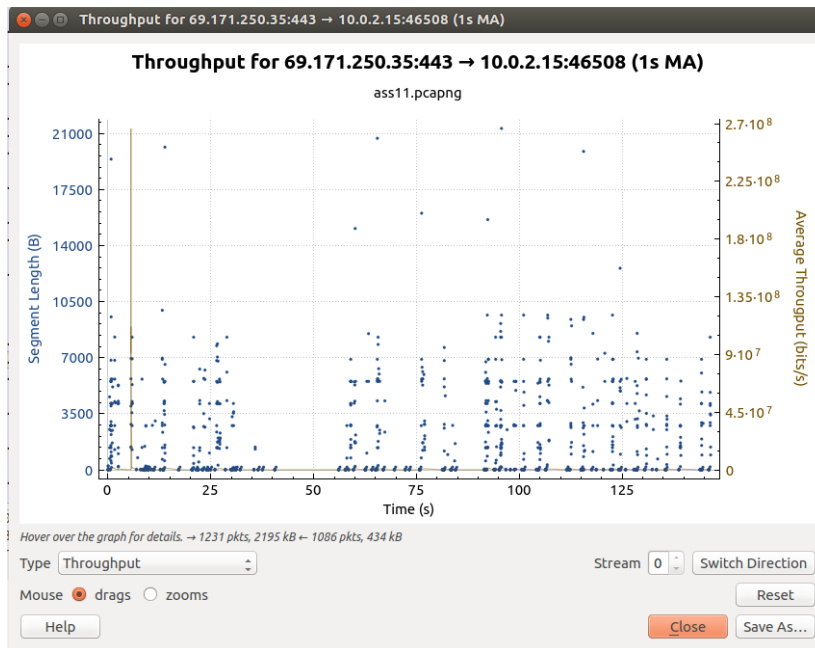
Interfaces				
Interface	Dropped packets	Capture filter	Link type	Packet size limit
enp0s3	0 (0.0%)	host 69.171.250.35	Ethernet	262144 bytes

Statistics			
Measurement	Captured	Displayed	Marked
Packets	2317	2317 (100.0%)	N/A
Time span, s	146.520	146.520	N/A
Average pps	15.8	15.8	N/A
Average packet size, B	1190.5	1190.5	N/A
Bytes	2758015	2758015 (100.0%)	0
Average bytes/s	18 k	18 k	N/A
Average bits/s	150 k	150 k	N/A

It shows the average data transferred is 150 kbits/s. Therefore, the average throughput is 150 kbits/s.

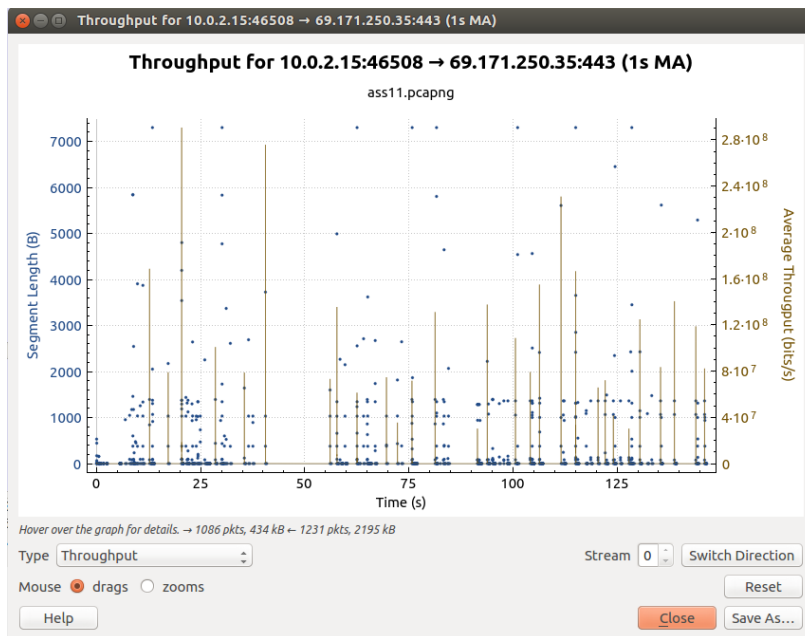
Actually, throughput changes from time to time during the packet capture. Wireshark allows us to see the variation of throughput graphically. Click on any one of the captured packets, go to **Statistics > TCP Stream Graphs > Throughput** to view the graphs.

Variation of Average Throughput when receiving packets from Facebook:



Variation of Average Throughput when sending packets to Facebook:

I got this graph by clicking **Switch Direction** option in the above graph.



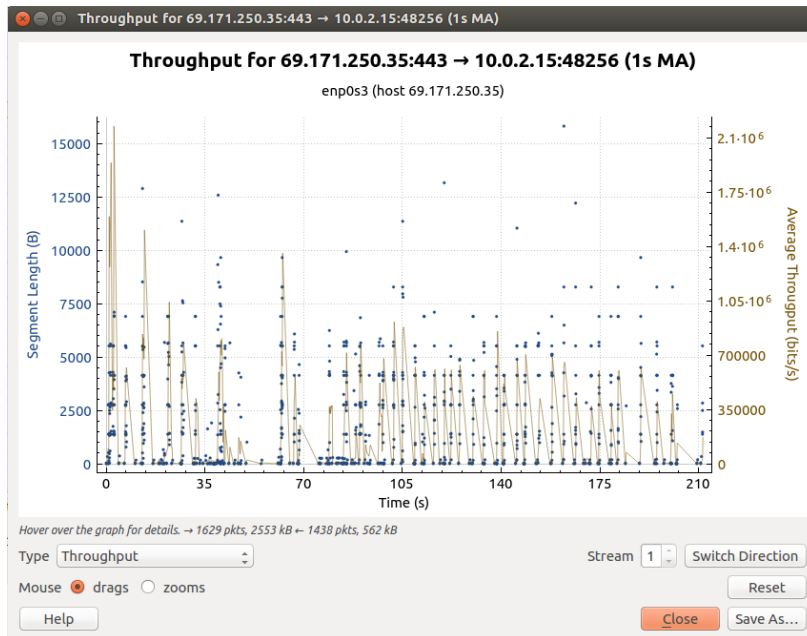
Throughput for Trial 2:

For Trial 2, the average throughput was found out to be 124 kbits/s. It is evident from below screenshot.

Interfaces				
Interface	Dropped packets	Capture filter	Link type	Packet size limit
enp0s3	0 (0.0%)	host 69.171.250.35	Ethernet	262144 bytes

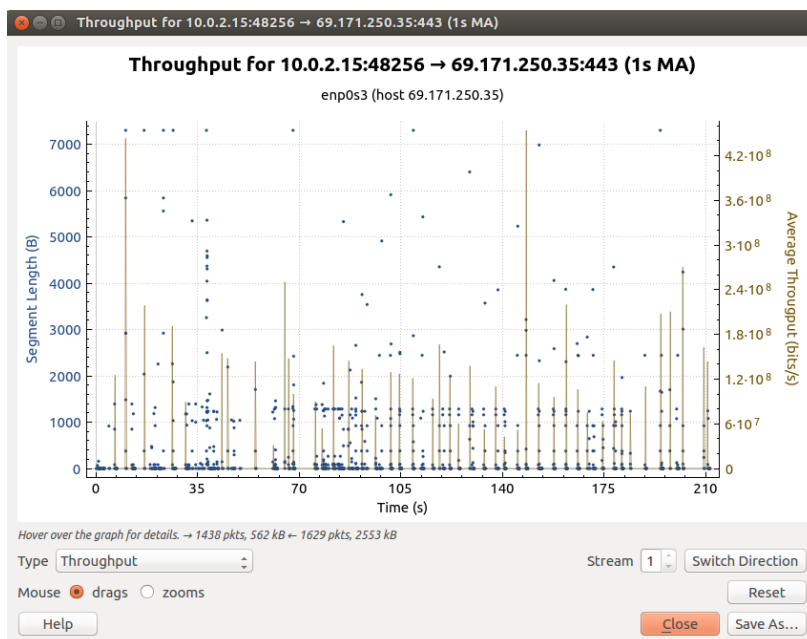
Statistics			
Measurement	Captured	Displayed	Marked
Packets	3087	3087 (100.0%)	N/A
Time span, s	211.406	211.406	N/A
Average pps	14.6	14.6	N/A
Average packet size, B	1066.5	1066.5	N/A
Bytes	3291333	3291333 (100.0%)	0
Average bytes/s	15 k	15 k	N/A
Average bits/s	124 k	124 k	N/A

Variation of Average Throughput when receiving packets from Facebook:



Variation of Average Throughput when sending packets to Facebook:

I got this graph by clicking **Switch Direction** option in the above graph.



Measurement of Round Trip Time (RTT):

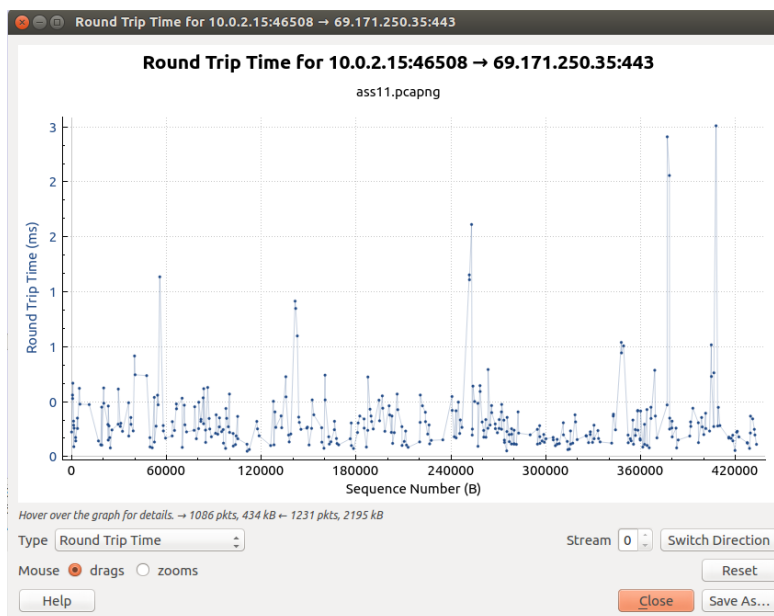
Round-trip time (RTT) is the duration from when a browser sends a request to when it receives a response from a server. RTT will vary continuously over the course of the packet capture. Wireshark provides a graph to look at the variation. Go to **Statistics > TCP Stream Graphs > Round Trip Time**.

RTT variation for Trial 1:

Variation when receiving packets from Facebook: RTT varies between 0 and 50ms as seen in this graph but majority of the packets have RTT near zero. We can count the number of packets whose RTT is higher.



Variation when sending packets from Facebook: RTT varies between 0.1 and 3ms as seen in this graph. But RTT for most of the packets lies in the range of 0.1 to 0.7ms.



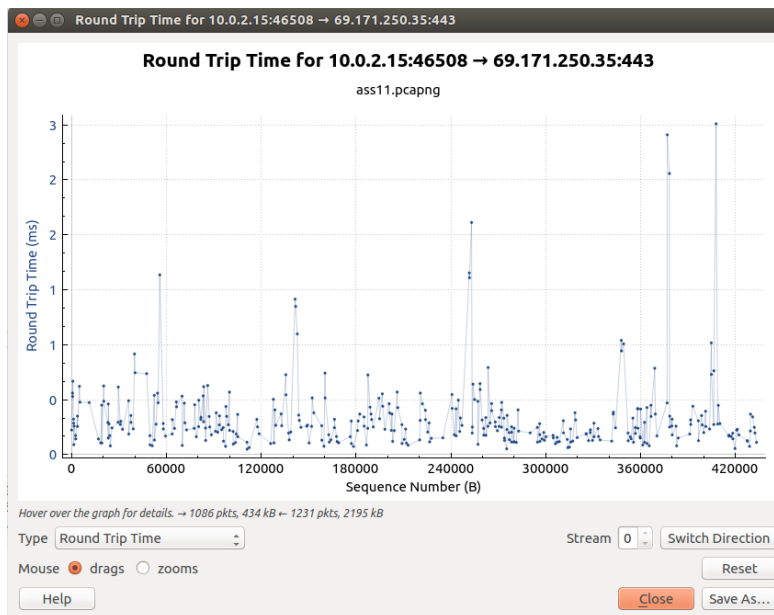
Well, the RTT for packets being received is significantly lower than those being sent. I infer this is because Facebook is a big company and would have really fast servers that can handle many packets in less amount of time. No surprise that my machine isn't that fast. So, the skewed graph was obtained for sending packets to Facebook.

RTT variation for Trial 1:

Variation when receiving packets from Facebook: RTT varies between 0 and 50ms as seen in this graph but majority of the packets have RTT near zero. We can count by ourselves the number of packets whose RTT is higher.

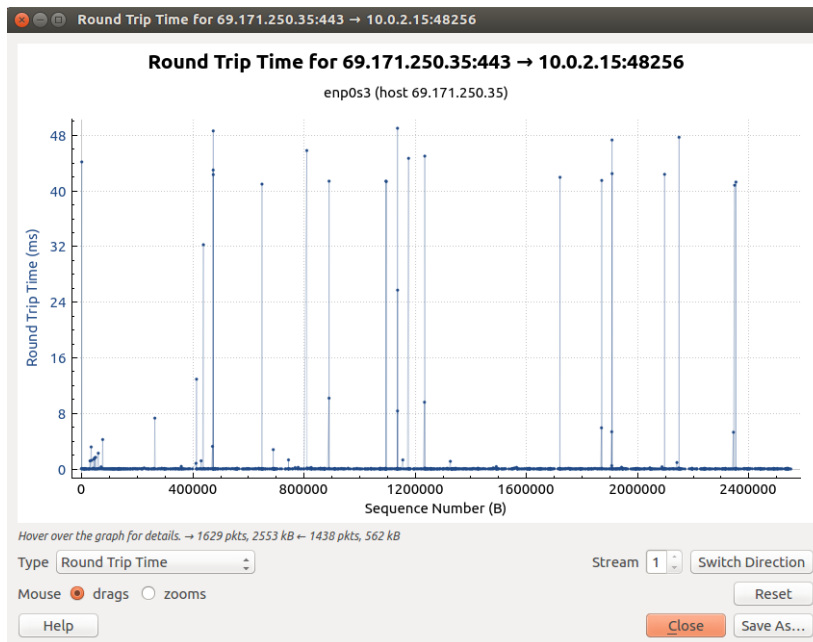


Variation when sending packets from Facebook: RTT varies between 0.1 and 3ms as seen in this graph. But RTT for most of the packets lies in the range of 0.1 to 0.7ms. This graph can be obtained by clicking on **Switch Direction** button in the above dialog box.

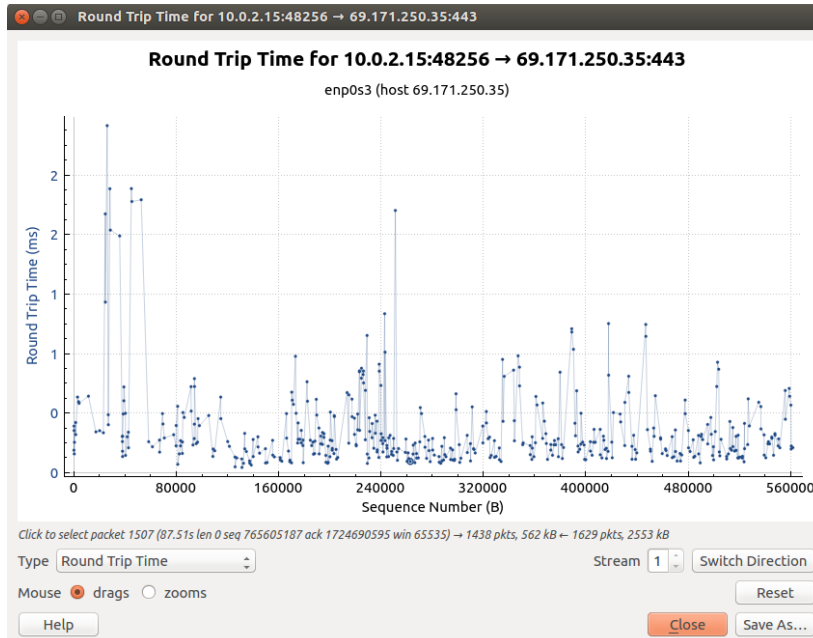


RTT variation for Trial 2:

Variation when receiving packets from Facebook: RTT varies between 0 and 50ms as seen in this graph but majority of the packets have RTT near zero. Again, we can count by ourselves the number of packets whose RTT is higher.



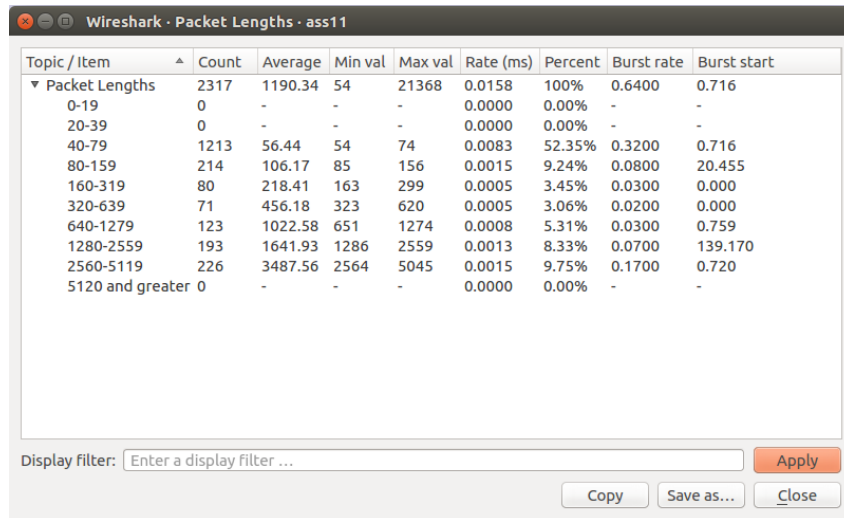
Variation when sending packets from Facebook: RTT varies between 0.1 and 2.5ms as seen in this graph. But RTT for most of the packets lies in the range of 0.1 to 1ms.



Packet size

One can see the size distribution of the captured packets in Wireshark. Go to **Statistics > Packet Lengths**.

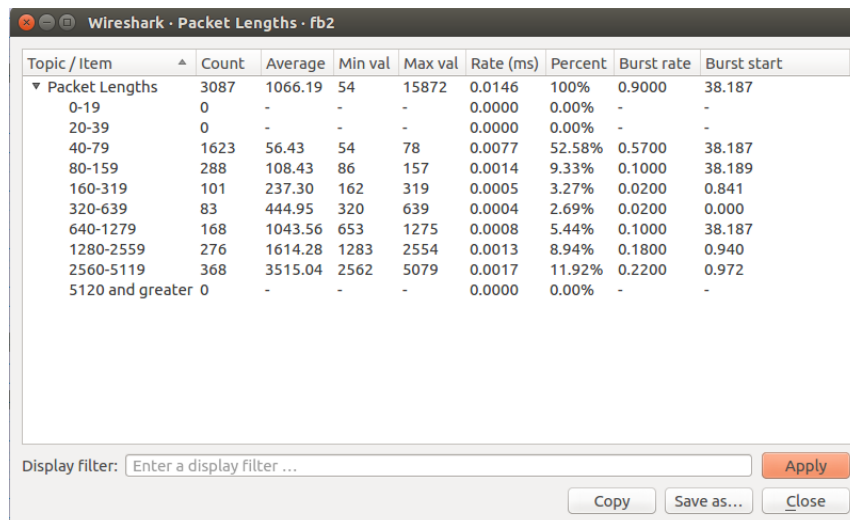
For Trial 1, the average packet size is 1190.34. The screenshot below shows the distribution of packet lengths.



The screenshot shows the Wireshark 'Packet Lengths' window for capture 'ass11'. The table displays the distribution of packet lengths across various ranges. The average packet size is 1190.34 bytes.

Topic / Item	Count	Average	Min val	Max val	Rate (ms)	Percent	Burst rate	Burst start
▼ Packet Lengths	2317	1190.34	54	21368	0.0158	100%	0.6400	0.716
0-19	0	-	-	-	0.0000	0.00%	-	-
20-39	0	-	-	-	0.0000	0.00%	-	-
40-79	1213	56.44	54	74	0.0083	52.35%	0.3200	0.716
80-159	214	106.17	85	156	0.0015	9.24%	0.0800	20.455
160-319	80	218.41	163	299	0.0005	3.45%	0.0300	0.000
320-639	71	456.18	323	620	0.0005	3.06%	0.0200	0.000
640-1279	123	1022.58	651	1274	0.0008	5.31%	0.0300	0.759
1280-2559	193	1641.93	1286	2559	0.0013	8.33%	0.0700	139.170
2560-5119	226	3487.56	2564	5045	0.0015	9.75%	0.1700	0.720
5120 and greater	0	-	-	-	0.0000	0.00%	-	-

For Trial 2, the average packet size is 1066.19. The screenshot below shows the distribution of packet lengths.



The screenshot shows the Wireshark 'Packet Lengths' window for capture 'fb2'. The table displays the distribution of packet lengths across various ranges. The average packet size is 1066.19 bytes.

Topic / Item	Count	Average	Min val	Max val	Rate (ms)	Percent	Burst rate	Burst start
▼ Packet Lengths	3087	1066.19	54	15872	0.0146	100%	0.9000	38.187
0-19	0	-	-	-	0.0000	0.00%	-	-
20-39	0	-	-	-	0.0000	0.00%	-	-
40-79	1623	56.43	54	78	0.0077	52.58%	0.5700	38.187
80-159	288	108.43	86	157	0.0014	9.33%	0.1000	38.189
160-319	101	237.30	162	319	0.0005	3.27%	0.0200	0.841
320-639	83	444.95	320	639	0.0004	2.69%	0.0200	0.000
640-1279	168	1043.56	653	1275	0.0008	5.44%	0.1000	38.187
1280-2559	276	1614.28	1283	2554	0.0013	8.94%	0.1800	0.940
2560-5119	368	3515.04	2562	5079	0.0017	11.92%	0.2200	0.972
5120 and greater	0	-	-	-	0.0000	0.00%	-	-

Number of Packets lost for Trial 1: No packets were dropped as in the screenshot below. I obtained the info by going to **Statistics > Capture File Properties**.

Interfaces				
Interface	Dropped packets	Capture filter	Link type	Packet size limit
enp0s3	0 (0%)	host 69.171.250.35	Ethernet	262144 bytes
Statistics				
Measurement	Captured	Displayed	Marked	
Packets	2317	2317 (100.0%)	N/A	
Time span, s	146.520	146.520	N/A	
Average pps	15.8	15.8	N/A	
Average packet size, B	1190.5	1190.5	N/A	
Bytes	2758015	2758015 (100.0%)	0	
Average bytes/s	18 k	18 k	N/A	
Average bits/s	150 k	150 k	N/A	

Number of Packets lost for Trial 2: No packets were dropped as in the screenshot below.

Interfaces				
<u>Interface</u>	<u>Dropped packets</u>	<u>Capture filter</u>	<u>Link type</u>	<u>Packet size limit</u>
enp0s3	0 (0%)	host 69.171.250.35	Ethernet	262144 bytes
Statistics				
<u>Measurement</u>	<u>Captured</u>	<u>Displayed</u>	<u>Marked</u>	
Packets	3087	3087 (100.0%)	N/A	
Time span, s	211.406	211.406	N/A	
Average pps	14.6	14.6	N/A	
Average packet size, B	1066.5	1066.5	N/A	
Bytes	3291333	3291333 (100.0%)	0	
Average bytes/s	15 k	15 k	N/A	
Average bits/s	124 k	124 k	N/A	

Number of TCP Packets for Trial 1: All 2317 packets captured were TCP packets

No.	Time	Source	Port	Destination	Port	Protocol	Length	Info
3	0.021272722	10.0.2.15	46508	69.171.250.35	443	TCP	54	46508 -> 443 [ACK] Seq=1430309946 Ack=43008002 Win=29200 Len=0
4	0.026135471	10.0.2.15	46508	69.171.250.35	443	TLSv1.2	586	Client Hello
5	0.026336374	69.171.250.35	443	10.0.2.15	46508	TCP	60	443 -> 46508 [ACK] Seq=43008002 Ack=1430310478 Win=65535 Len=0
6	0.040801655	69.171.250.35	443	10.0.2.15	46508	TLSv1.2	266	Server Hello, Change Cipher Spec, Application Data
7	0.048104587	10.0.2.15	46508	69.171.250.35	443	TCP	54	46508 -> 443 [ACK] Seq=1430310478 Ack=43008214 Win=30016 Len=0
8	0.051562274	10.0.2.15	46508	69.171.250.35	443	TLSv1.2	118	Change Cipher Spec, Application Data
9	0.052066634	69.171.250.35	443	10.0.2.15	46508	TCP	60	443 -> 46508 [ACK] Seq=43008214 Ack=1430310542 Win=65535 Len=0
10	0.054186812	10.0.2.15	46508	69.171.250.35	443	TLSv1.2	224	Application Data
11	0.054606180	69.171.250.35	443	10.0.2.15	46508	TCP	60	443 -> 46508 [ACK] Seq=43008214 Ack=1430310712 Win=65535 Len=0
12	0.054706707	10.0.2.15	46508	69.171.250.35	443	TLSv1.2	498	Application Data
13	0.055301258	69.171.250.35	443	10.0.2.15	46508	TCP	60	443 -> 46508 [ACK] Seq=43008214 Ack=1430311156 Win=65535 Len=0
14	0.072845548	69.171.250.35	443	10.0.2.15	46508	TLSv1.2	299	Application Data, Application Data
15	0.073140295	10.0.2.15	46508	69.171.250.35	443	TLSv1.2	85	Application Data
16	0.073430441	69.171.250.35	443	10.0.2.15	46508	TCP	60	443 -> 46508 [ACK] Seq=43008459 Ack=1430311187 Win=65535 Len=0
17	0.074928453	69.171.250.35	443	10.0.2.15	46508	TLSv1.2	95	Application Data
18	0.077713123	69.171.250.35	443	10.0.2.15	46508	TLSv1.2	89	Application Data
19	0.077774452	10.0.2.15	46508	69.171.250.35	443	TCP	54	46508 -> 443 [ACK] Seq=1430311187 Ack=43008525 Win=31088 Len=0
20	0.011397252	69.171.250.35	443	10.0.2.15	46508	TLSv1.2	4607	Application Data, Application Data, Application Data
21	0.412028665	10.0.2.15	46508	69.171.250.35	443	TCP	54	46508 -> 443 [ACK] Seq=1430311187 Ack=43013078 Win=39760 Len=0
22	0.412047165	69.171.250.35	443	10.0.2.15	46508	TLSv1.2	1434	Application Data[TCP segment of a reassembled PDU]
23	0.412988323	69.171.250.35	443	10.0.2.15	46508	TLSv1.2	2572	Application Data
24	0.412999672	10.0.2.15	46508	69.171.250.35	443	TCP	54	46508 -> 443 [ACK] Seq=1430311187 Ack=43016976 Win=48280 Len=0
25	0.414102446	69.171.250.35	443	10.0.2.15	46508	TLSv1.2	757	Application Data
26	0.455998848	10.0.2.15	46508	69.171.250.35	443	TCP	54	46508 -> 443 [ACK] Seq=1430311187 Ack=43017679 Win=51120 Len=0
27	0.4551431840	10.0.2.15	46508	69.171.250.35	443	TLSv1.2	200	Application Data
28	0.4551622858	10.0.2.15	46508	69.171.250.35	443	TLSv1.2	210	Application Data
29	0.4551688084	69.171.250.35	443	10.0.2.15	46508	TCP	60	443 -> 46508 [ACK] Seq=43017679 Ack=1430311342 Win=65535 Len=0
30	0.4551703635	69.171.250.35	443	10.0.2.15	46508	TCP	60	443 -> 46508 [ACK] Seq=43017679 Ack=1430311498 Win=65535 Len=0
31	0.4576194643	69.171.250.35	443	10.0.2.15	46508	TLSv1.2	89	Application Data

▶ Frame 38: 69 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
 ▶ Ethernet II, Src: RealtekU12:35:02 (52:54:00:12:35:02), Dst: PcsCompu_5d:9d:56 (08:00:27:5d:9d:56)
 ▶ Internet Protocol Version 4, Src: 69.171.250.

Number of UDP Packets for Trial 1: None of the packets captured were UDP packets

[illegible]

Number of TCP Packets for Trial 2: All 3087 packets captured were TCP packets

No.	Time	Source	Port	Destination	Port	Protocol	Length	Info
145	1.068910913	10.0.2.15	48256	69.171.250.35	443	TCP	54	48256 → 443 [ACK] Seq=1724426701 Ack=764495820 Win=65535 Len=
146	1.072184703	69.171.250.35	443	10.0.2.15	48256	TLSv1.2	2814	Application Data, Application Data
147	1.072203306	10.0.2.15	48256	69.171.250.35	443	TCP	54	48256 → 443 [ACK] Seq=1724426701 Ack=764498580 Win=65535 Len=
148	1.073397645	69.171.250.35	443	10.0.2.15	48256	TLSv1.2	1434	Application Data[TCP segment of a reassembled PDU]
149	1.073482424	10.0.2.15	48256	69.171.250.35	443	TCP	54	48256 → 443 [ACK] Seq=1724426701 Ack=764499960 Win=65535 Len=
150	1.074750268	69.171.250.35	443	10.0.2.15	48256	TCP	1434	[TCP segment of a reassembled PDU]
151	1.074765145	10.0.2.15	48256	69.171.250.35	443	TCP	54	48256 → 443 [ACK] Seq=1724426701 Ack=764501340 Win=65535 Len=
152	1.076256677	69.171.250.35	443	10.0.2.15	48256	TLSv1.2	1434	Application Data, Application Data, Application Data
153	1.076261141	10.0.2.15	48256	69.171.250.35	443	TCP	54	48256 → 443 [ACK] Seq=1724426701 Ack=764502720 Win=65535 Len=
154	1.077920932	69.171.250.35	443	10.0.2.15	48256	TLSv1.2	1385	Application Data, Application Data
155	1.077929954	10.0.2.15	48256	69.171.250.35	443	TCP	54	48256 → 443 [ACK] Seq=1724426701 Ack=764503971 Win=65535 Len=
156	1.079286182	69.171.250.35	443	10.0.2.15	48256	TLSv1.2	523	Application Data
157	1.079291820	10.0.2.15	48256	69.171.250.35	443	TCP	54	48256 → 443 [ACK] Seq=1724426701 Ack=764504440 Win=65535 Len=
158	1.136878373	69.171.250.35	443	10.0.2.15	48256	TLSv1.2	330	Application Data
159	1.136884313	10.0.2.15	48256	69.171.250.35	443	TCP	54	48256 → 443 [ACK] Seq=1724426701 Ack=764504716 Win=65535 Len=
160	1.139507478	69.171.250.35	443	10.0.2.15	48256	TLSv1.2	86	Application Data
161	1.139562905	10.0.2.15	48256	69.171.250.35	443	TCP	54	48256 → 443 [ACK] Seq=1724426701 Ack=764504748 Win=65535 Len=
162	1.163776390	69.171.250.35	443	10.0.2.15	48256	TLSv1.2	2814	Application Data
163	1.163794839	10.0.2.15	48256	69.171.250.35	443	TCP	54	48256 → 443 [ACK] Seq=1724426701 Ack=764507508 Win=65535 Len=
164	1.164870781	69.171.250.35	443	10.0.2.15	48256	TLSv1.2	1375	Application Data
165	1.164885567	10.0.2.15	48256	69.171.250.35	443	TCP	54	48256 → 443 [ACK] Seq=1724426701 Ack=764508829 Win=65535 Len=
166	1.167833867	69.171.250.35	443	10.0.2.15	48256	TLSv1.2	2552	Application Data, Application Data
167	1.167848510	10.0.2.15	48256	69.171.250.35	443	TCP	54	48256 → 443 [ACK] Seq=1724426701 Ack=764511327 Win=65535 Len=
168	1.205754455	69.171.250.35	443	10.0.2.15	48256	TCP	1434	[TCP segment of a reassembled PDU]
169	1.205762723	10.0.2.15	48256	69.171.250.35	443	TCP	54	48256 → 443 [ACK] Seq=1724426701 Ack=764512707 Win=65535 Len=
170	1.206810937	69.171.250.35	443	10.0.2.15	48256	TLSv1.2	2404	Application Data, Application Data
171	1.206821437	10.0.2.15	48256	69.171.250.35	443	TCP	54	48256 → 443 [ACK] Seq=1724426701 Ack=764515057 Win=65535 Len=
172	1.336760740	69.171.250.35	443	10.0.2.15	48256	TLSv1.2	2814	Application Data
173	1.336777920	10.0.2.15	48256	69.171.250.35	443	TCP	54	48256 → 443 [ACK] Seq=1724426701 Ack=764517817 Win=65535 Len=

Number of UDP Packets for Trial 2: None of the packets captured were UDP packets

The image shows the Wireshark network protocol analyzer interface. The top status bar indicates the selected protocol is 'udp'. Below this, the packet list pane, packet details pane, and packet bytes pane are all empty. The bottom status bar displays the protocol name 'User Datagram Protocol: Protocol' and statistics: 'Packets: 3087 · Displayed: 0 (0.0%) · Dropped: 0 (0.0%) · Profile: Default'.

Number of responses received with respect to one request sent

To get this information, we need to know what packets are requests sent and what packets are the responses.

We could use the **http.request** filter in Wireshark to find out which packets were request packets. Then by using some more commands, we could identify the number of responses for a given request packet.

But this method works only for HTTP protocols. For HTTPS protocol, Wireshark can't show which packets are request and which ones are response.

Trial 1:

So, I took the liberty to run the packet capture on a different website, bio.acousti.ca whose IP address is 157.140.2.239, which I found by executing **ping bio.acousti.ca** in the terminal.

No.	Time	Source	Port	Destination	Port	Protocol	Length	Info
1	0.000000000	10.0.2.15	34968	157.140.2.239	80	TCP	74	34968 → 80 [SYN] Seq=4241374293 Win=29200 Len=0 MSS=1460 SAC...
2	0.039335765	10.0.2.15	34972	157.140.2.239	80	TCP	74	34972 → 80 [SYN] Seq=3193334508 Win=29200 Len=0 MSS=1460 SAC...
3	0.137663548	157.140.2.239	80	10.0.2.15	34968	TCP	60	80 → 34968 [SYN, ACK] Seq=763008001 Ack=4241374294 Win=65535...
4	0.137704926	10.0.2.15	34968	157.140.2.239	80	TCP	54	34968 → 80 [ACK] Seq=4241374294 Ack=763008002 Win=29200 Len=0
5	0.138303840	10.0.2.15	34968	157.140.2.239	80	HTTP	688	GET / HTTP/1.1
6	0.138532432	157.140.2.239	80	10.0.2.15	34968	TCP	60	80 → 34968 [ACK] Seq=763008002 Ack=4241374928 Win=65535 Len=0
7	0.184213183	157.140.2.239	80	10.0.2.15	34972	TCP	60	80 → 34972 [SYN, ACK] Seq=763072001 Ack=3193334509 Win=65535...
8	0.184252535	10.0.2.15	34972	157.140.2.239	80	TCP	54	34972 → 80 [ACK] Seq=3193334509 Ack=763072002 Win=29200 Len=0
9	0.278047413	157.140.2.239	80	10.0.2.15	34968	HTTP	684	HTTP/1.1 304 Not Modified
10	0.278067386	10.0.2.15	34968	157.140.2.239	80	TCP	54	34968 → 80 [ACK] Seq=4241374928 Ack=763008632 Win=30240 Len=0
11	0.419846535	10.0.2.15	34972	157.140.2.239	80	HTTP	672	GET /sites/all/libraries/mediaelement/build/mediaelement-and...
12	0.419927839	10.0.2.15	34968	157.140.2.239	80	HTTP	664	GET /sites/all/modules/contrib/views_slideshow/js/views_slid...
13	0.420087793	157.140.2.239	80	10.0.2.15	34972	TCP	60	80 → 34972 [ACK] Seq=763072002 Ack=3193335127 Win=65535 Len=0
14	0.420099340	157.140.2.239	80	10.0.2.15	34968	TCP	60	80 → 34968 [ACK] Seq=763008632 Ack=4241375538 Win=65535 Len=0
15	0.420214568	10.0.2.15	34974	157.140.2.239	80	TCP	74	34974 → 80 [SYN] Seq=1135083405 Win=29200 Len=0 MSS=1460 SAC...
16	0.420275948	10.0.2.15	34976	157.140.2.239	80	TCP	74	34976 → 80 [SYN] Seq=2131970462 Win=29200 Len=0 MSS=1460 SAC...
17	0.420330528	10.0.2.15	34978	157.140.2.239	80	TCP	74	34978 → 80 [SYN] Seq=141816690 Win=29200 Len=0 MSS=1460 SACK...
18	0.420386311	10.0.2.15	34980	157.140.2.239	80	TCP	74	34980 → 80 [SYN] Seq=2274291710 Win=29200 Len=0 MSS=1460 SAC...
19	0.424736533	10.0.2.15	34968	157.140.2.239	80	TCP	54	34968 → 80 [FIN, ACK] Seq=4241375538 Ack=763008632 Win=30240...
20	0.424967126	157.140.2.239	80	10.0.2.15	34968	TCP	60	80 → 34968 [ACK] Seq=763008632 Ack=4241375539 Win=65535 Len=0
21	0.425138557	10.0.2.15	34982	157.140.2.239	80	TCP	74	34982 → 80 [SYN] Seq=2456681006 Win=29200 Len=0 MSS=1460 SAC...
22	0.425212465	10.0.2.15	34984	157.140.2.239	80	TCP	74	34984 → 80 [SYN] Seq=3656492764 Win=29200 Len=0 MSS=1460 SAC...
23	0.425269612	10.0.2.15	34986	157.140.2.239	80	TCP	74	34986 → 80 [SYN] Seq=2208127739 Win=29200 Len=0 MSS=1460 SAC...
24	0.429839727	10.0.2.15	34988	157.140.2.239	80	TCP	74	34988 → 80 [SYN] Seq=3399921067 Win=29200 Len=0 MSS=1460 SAC...
25	0.429950734	10.0.2.15	34990	157.140.2.239	80	TCP	74	34990 → 80 [SYN] Seq=1627649963 Win=29200 Len=0 MSS=1460 SAC...
26	0.430025775	10.0.2.15	34992	157.140.2.239	80	TCP	74	34992 → 80 [SYN] Seq=1654476378 Win=29200 Len=0 MSS=1460 SAC...
27	0.437756745	10.0.2.15	34994	157.140.2.239	80	TCP	74	34994 → 80 [SYN] Seq=2016889421 Win=29200 Len=0 MSS=1460 SAC...
28	0.437929609	10.0.2.15	34996	157.140.2.239	80	TCP	74	34996 → 80 [SYN] Seq=3607365638 Win=29200 Len=0 MSS=1460 SAC...
29	0.447512660	10.0.2.15	34998	157.140.2.239	80	TCP	74	34998 → 80 [SYN] Seq=1309135770 Win=29200 Len=0 MSS=1460 SAC...

I filtered out the request packets using **http.request** display filter. I chose one of the filtered packets and tried to get its corresponding responses.

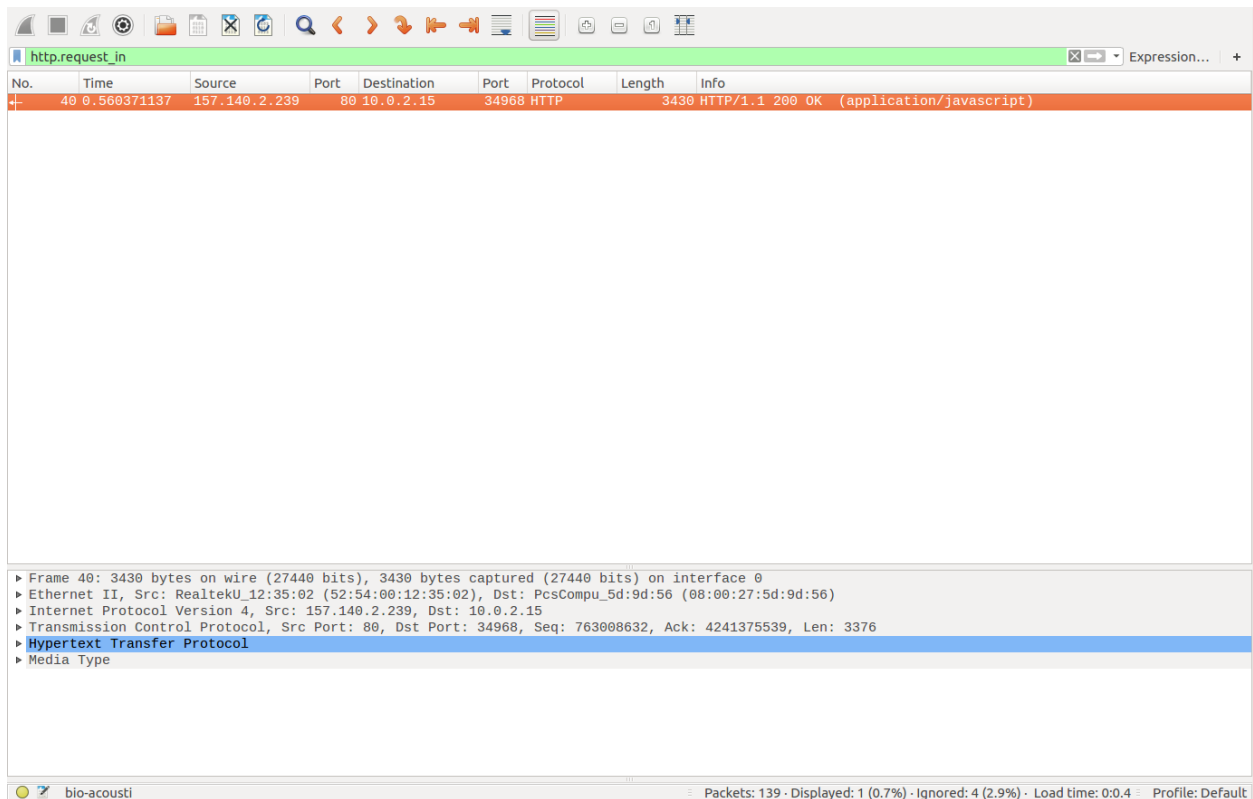
I chose the 12th packet in the above list whose info is:

"GET /sites/all/modules/contrib/views_slideshow/js/views_slideshow.js?v=1.0 HTTP/1.1"

Now, I have to isolate the response packets for the selected request packet. So, I cleared the display filter and selected the remaining packets using the display filter: **http.request && !http.request.uri contains "/sites/all/modules/contrib/views_slideshow/js/views_slideshow.js?v=1.0"**

Now, ignore all the packets that have been filtered out, by opening **Edit > Ignore all displayed**. This leaves only the aforementioned request packet and its corresponding response packets.

By using the display filter **http.request_in**, we can finally view the response packets for the selected request packet. I noticed that there is only one response packet.



So, I used the procedure mentioned above to perform a second trial described below.

Trial 2:

I repeated the above procedure on the same website for a second time. This time, I tried to get responses for the 68th packet in the packet capture screenshot below.

Wireshark packet capture screenshot showing a list of network packets. The display filter is set to `http.request`. Packet 68 is highlighted in red, showing an HTTP GET request for `/misc/drupal.js?qrtaf5`. The packet details pane shows the structure of the packet: Ethernet II, Internet Protocol Version 4, Transmission Control Protocol, and Hypertext Transfer Protocol.

No.	Time	Source	Port	Destination	Port	Protocol	Length	Info
50	1.762477246	157.140.2.239	80	10.0.2.15	47664	HTTP	944	HTTP/1.1 200 OK (text/css)
51	1.762501343	10.0.2.15	47664	157.140.2.239	80	TCP	54	47664 → 80 [RST] Seq=1181868869 Win=0 Len=0
52	1.786867991	10.0.2.15	47718	157.140.2.239	80	TCP	74	47718 → 80 [SYN] Seq=3920615976 Win=29200 Len=0 MSS=1460 SAC...
53	1.848571404	10.0.2.15	47720	157.140.2.239	80	TCP	74	47720 → 80 [SYN] Seq=1739709988 Win=29200 Len=0 MSS=1460 SAC...
54	1.924709569	10.0.2.15	47722	157.140.2.239	80	TCP	74	47722 → 80 [SYN] Seq=861162032 Win=29200 Len=0 MSS=1460 SAC...
55	1.979352862	10.0.2.15	47724	157.140.2.239	80	TCP	74	47724 → 80 [SYN] Seq=4032027519 Win=29200 Len=0 MSS=1460 SAC...
56	1.989516417	10.0.2.15	47726	157.140.2.239	80	TCP	74	47726 → 80 [SYN] Seq=3866161336 Win=29200 Len=0 MSS=1460 SAC...
57	2.000709208	157.140.2.239	80	10.0.2.15	47720	TCP	60	80 → 47720 [SYN, ACK] Seq=264768001 Ack=1739709989 Win=65535...
58	2.000768630	10.0.2.15	47720	157.140.2.239	80	TCP	54	47720 → 80 [ACK] Seq=1739709989 Ack=264768002 Win=29200 Len=0
59	2.001844702	10.0.2.15	47720	157.140.2.239	80	HTTP	622	GET /misc/autocomplete.js?v=7.78 HTTP/1.1
60	2.002001303	157.140.2.239	80	10.0.2.15	47720	TCP	60	80 → 47720 [ACK] Seq=264768002 Ack=1739710557 Win=65535 Len=0
61	2.038285652	10.0.2.15	47728	157.140.2.239	80	TCP	74	47728 → 80 [SYN] Seq=2258386406 Win=29200 Len=0 MSS=1460 SAC...
62	2.099606079	157.140.2.239	80	10.0.2.15	47722	TCP	60	80 → 47722 [SYN, ACK] Seq=264832001 Ack=861162033 Win=65535...
63	2.099674816	10.0.2.15	47722	157.140.2.239	80	TCP	54	47722 → 80 [ACK] Seq=861162033 Ack=264832002 Win=29200 Len=0
64	2.099901811	10.0.2.15	47722	157.140.2.239	80	HTTP	600	GET /sites/default/files/css/css_tcVOMd1RMjTsBkm7ZJABjZ30ct1...
65	2.100092752	157.140.2.239	80	10.0.2.15	47722	TCP	60	80 → 47722 [ACK] Seq=264832002 Ack=861162579 Win=65535 Len=0
66	2.155698342	157.140.2.239	80	10.0.2.15	47720	HTTP	428	HTTP/1.1 304 Not Modified
67	2.155721744	10.0.2.15	47720	157.140.2.239	80	TCP	54	47720 → 80 [ACK] Seq=1739710557 Ack=264768376 Win=30016 Len=0
68	2.150700094	10.0.2.15	47720	157.140.2.239	80	HTTP	524	GET /misc/drupal.js?qrtaf5 HTTP/1.1
69	2.157101582	157.140.2.239	80	10.0.2.15	47720	TCP	60	80 → 47720 [ACK] Seq=264768376 Ack=1739711027 Win=65535 Len=0
70	2.259787586	157.140.2.239	80	10.0.2.15	47722	HTTP	1080	HTTP/1.1 200 OK (text/css)
71	2.259837807	10.0.2.15	47722	157.140.2.239	80	TCP	54	47722 → 80 [ACK] Seq=861162579 Ack=264833028 Win=31806 Len=0
72	2.260292086	10.0.2.15	47722	157.140.2.239	80	HTTP	580	GET /sites/all/libraries/mediaelement/build/mediaelement-and...
73	2.260493371	157.140.2.239	80	10.0.2.15	47722	TCP	60	80 → 47722 [ACK] Seq=264833028 Ack=861163105 Win=65535 Len=0
74	2.308766650	157.140.2.239	80	10.0.2.15	47720	TCP	2790	[TCP segment of a reassembled PDU]
75	2.308787182	10.0.2.15	47720	157.140.2.239	80	TCP	54	47720 → 80 [ACK] Seq=1739711027 Ack=264771112 Win=35500 Len=0
76	2.309770149	157.140.2.239	80	10.0.2.15	47720	HTTP	4746	HTTP/1.1 200 OK (application/javascript)
77	2.309798124	10.0.2.15	47720	157.140.2.239	80	TCP	54	47720 → 80 [ACK] Seq=1739711027 Ack=264775804 Win=44020 Len=0
78	2.310924566	10.0.2.15	47720	157.140.2.239	80	HTTP	647	GET /misc/jquery.html-prefilter-3.5.0-backport.js?v=1.8.3 HT...

Frame 68: 524 bytes on wire (4192 bits), 524 bytes captured (4192 bits) on interface 0
Ethernet II, Src: PcsCompu_5d:9d:56 (08:00:27:5d:9d:56), Dst: RealtekU_12:35:02 (52:54:00:12:35:02)
Internet Protocol Version 4, Src: 10.0.2.15, Dst: 157.140.2.239
Transmission Control Protocol, Src Port: 47720, Dst Port: 80, Seq: 1739710557, Ack: 264768376, Len: 470
Hypertext Transfer Protocol

After ignoring the irrelevant packets using the display filter: `http.request && !http.request.uri contains /misc/drupal.js?qrtaf5`, I used the display filter `http.request_in` to find that there was only one response packet.

Wireshark packet capture screenshot showing a single network packet. The display filter is set to `http.request_in`. Packet 76 is highlighted in red, showing an HTTP GET request for `/misc/drupal.js?qrtaf5`. The packet details pane shows the structure of the packet: Ethernet II, Internet Protocol Version 4, Transmission Control Protocol, and Hypertext Transfer Protocol.

No.	Time	Source	Port	Destination	Port	Protocol	Length	Info
76	2.309770149	157.140.2.239	80	10.0.2.15	47720	HTTP	4746	HTTP/1.1 200 OK (application/javascript)

Frame 76: 4746 bytes on wire (37968 bits), 4746 bytes captured (37968 bits) on interface 0
Ethernet II, Src: RealtekU_12:35:02 (52:54:00:12:35:02), Dst: PcsCompu_5d:9d:56 (08:00:27:5d:9d:56)
Internet Protocol Version 4, Src: 157.140.2.239, Dst: 10.0.2.15
Transmission Control Protocol, Src Port: 80, Dst Port: 47720, Seq: 264771112, Ack: 1739711027, Len: 4692
[2 Reassembled TCP Segments (7428 bytes): #74(2736), #76(4692)]
Hypertext Transfer Protocol
Media Type

----- End of report -----