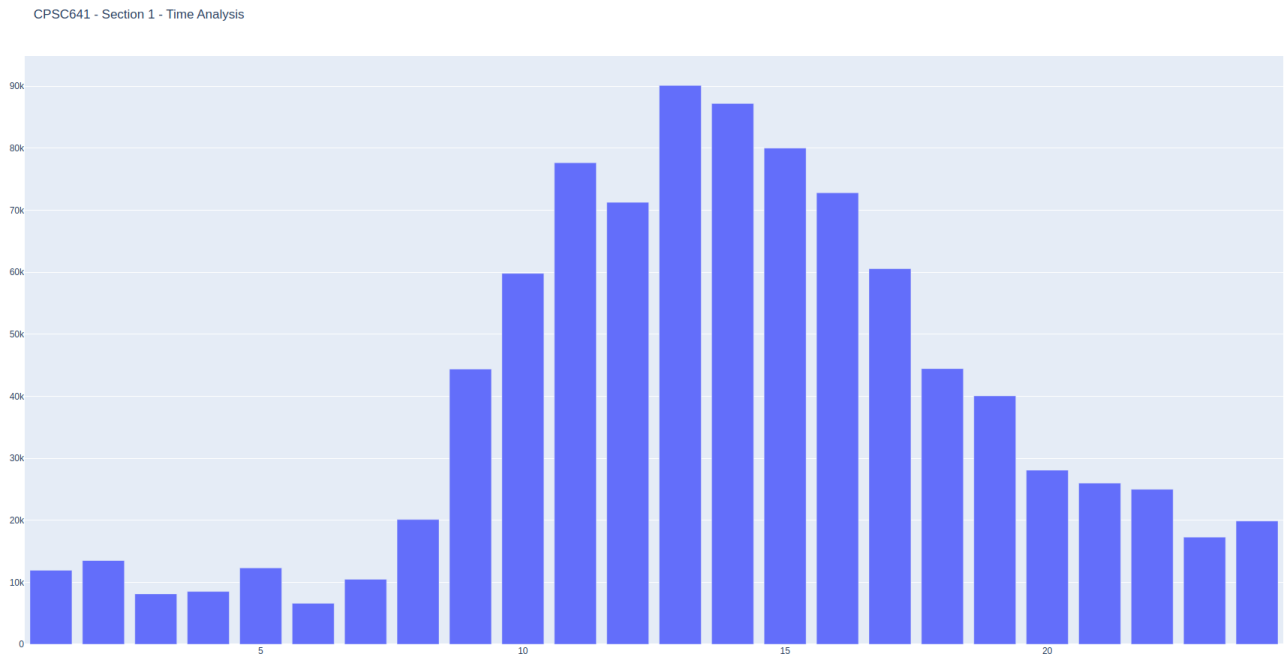
CPSC 641: High Speed Networks
First Assignment
Amirhossein Sefati

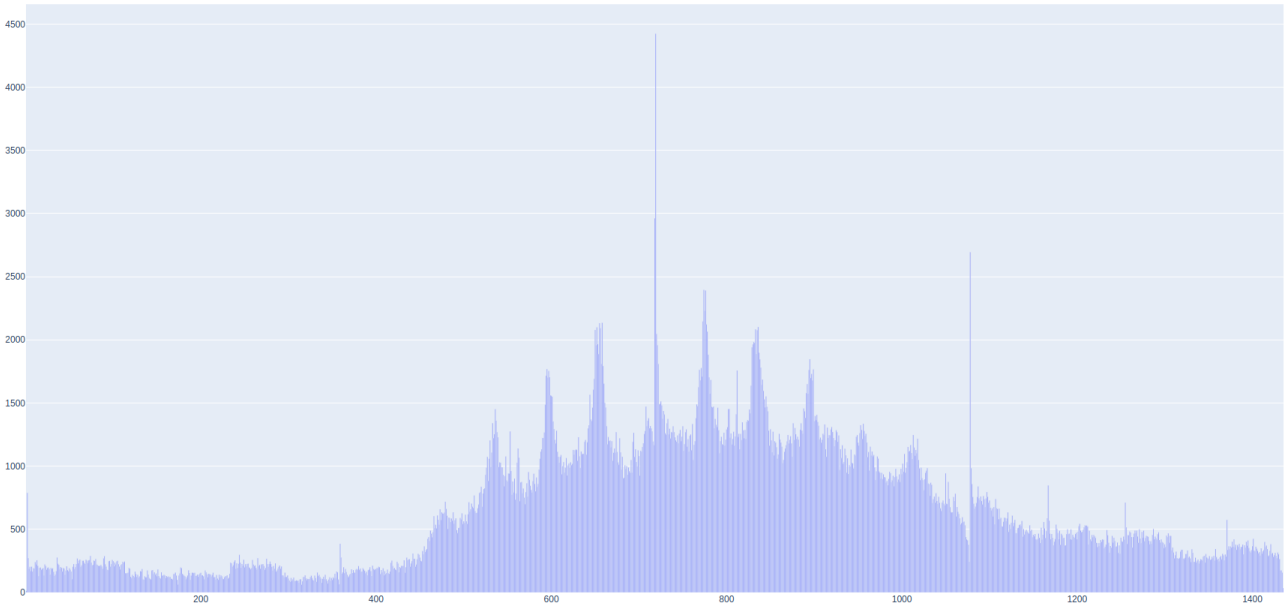
Section 1:



- Figure 1: Arrival Time Analysis (1 hour window size)

As we can see in the Figure 1, from 00:00 to 23:59, the number of TCP connections is shown. We can result to things, the first and obvious one is the number of users using the internet is likely to be higher when it comes to mid of the day (the maximum number was 91000 which happened at 13pm.)
The second important thing to notice is the way of increasing the network usage which is kind of similar to exponentially increasing. So, it is not linear but exponential and it starts suddenly to increase after 6am (when usually a human starts its day) and then gradually (kind of linearly) decrease in the night but it remains higher than the midnight. (For example, 23pm has a more network requests compared to 2am, and it is because normally people will sleep after 23pm-0am)
Of course we have to analyze so many other days and just one single day shouldn't be the reference.

For the next analysis, I use a 1 minutes window size for the plot and Figure 2 shows how the result is. As we can see, the result has a gaussian or normal distribution which in the middle is the highest points and it starts with low values and ends with low values too.



- Figure 2: Arrival Time Analysis (1 minute window size)

Another thing is similar to the point of Figure 1 which it indicates that the nights have more TCP connections compared to days. It is that most of the people are whether sleep or not working with internet during the early times of the day.

Section 2:

At first, I just get some statistical analysis from the data. It is how python helps me through that, the number of all of the “Durations” is 868889 which is not the same as the number of the total files because some of them just miss the duration field. (I detected them by the code and not manual). Then, the mean is 189.14 which it means that in the average, users would spend ~3 minutes when they used a new TCP connection. Standard derivation is 590, minimum is 0 and maximum is 21504. Also, I calculated three quartiles which is better than just having the median. I delete the rows

```
amirh@AHS:~/Desktop/Winter_2022/Courses/CPSC 641/A1$ python s2.py
            Duration
count  868889.000000
mean      189.140636
std       590.896629
min         0.000000
25%         1.350423
50%        22.420548
75%       125.489980
max     21504.610639
```

Figure 3: Connection Duration Analysis – Simple Analysis

```
amirh@AHS:~/Desktop/Winter_2022/Courses/CPSC 641/A1$ python s2.py
            Duration  frequency       pdf       cdf
0           0.000000       4640  0.005340  0.005340
1           0.000001         69  0.000079  0.005420
2           0.000002         68  0.000078  0.005498
3           0.000003        107  0.000123  0.005621
4           0.000004        167  0.000192  0.005813
...              ...        ...       ...       ...
742577  21104.241648          1  0.000001  0.999995
742578  21503.599426          1  0.000001  0.999997
742579  21503.614942          1  0.000001  0.999998
742580  21504.301788          1  0.000001  0.999999
742581  21504.610639          1  0.000001  1.000000
```

Figure 4: Connection Duration Analysis – PDF and CDF

As it can be seen by Figure 4, I got the frequencies of each of duration data in the raw data and after getting the frequencies, I calculated pdf and cdf of duration of each TCP connection. It can be seen that the number of 0s in the duration is a much more than other durations so I tried to delete them (as they are probably a problem of measurement). Now, it is time for recalculate all the before mentioned statistics.

```
amirh@AHS:~/Desktop/Winter_2022/Courses/CPSC 641/A1$ python s2.py
            Duration
count  864249.000000
mean      190.156099
std       592.317739
min         0.000001
25%         1.470451
50%        24.138791
75%       128.590764
max     21504.610639
```

Figure 5: Connection Duration Analysis – Simple Analysis after removing 0s.

It can be seen from Figure 5 that the average of duration increases which is normal because we just remove all the 0s from our data. All other statistics also changes. Now, for PDF and CDF we have:

	Duration	frequency	pdf	cdf
0	0.000001	69	0.000080	0.000080
1	0.000002	68	0.000079	0.000159
2	0.000003	107	0.000124	0.000282
3	0.000004	167	0.000193	0.000476
4	0.000005	201	0.000233	0.000708
...	...	...	...	...
742576	21104.241648	1	0.000001	0.999995
742577	21503.599426	1	0.000001	0.999997
742578	21503.614942	1	0.000001	0.999998
742579	21504.301788	1	0.000001	0.999999
742580	21504.610639	1	0.000001	1.000000

Figure 6: Connection Duration Analysis – PDF and CDF after removing the 0s.

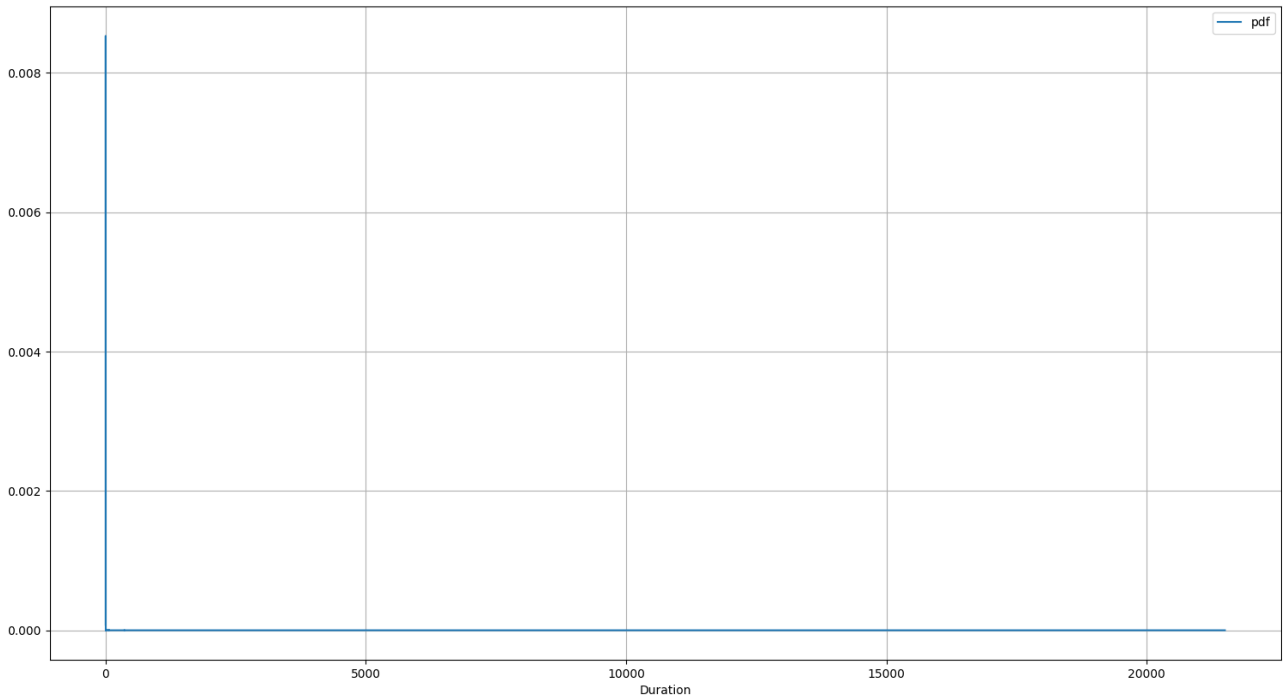


Figure 7: Connection Duration Analysis – Plotting PDF

Figure 7 shows that duration is usually small and the chances for having a long communication with one TCP connection is low. Also, from PDF we can see the range is from 0.000001 second to more than 21500 seconds with is amazing.

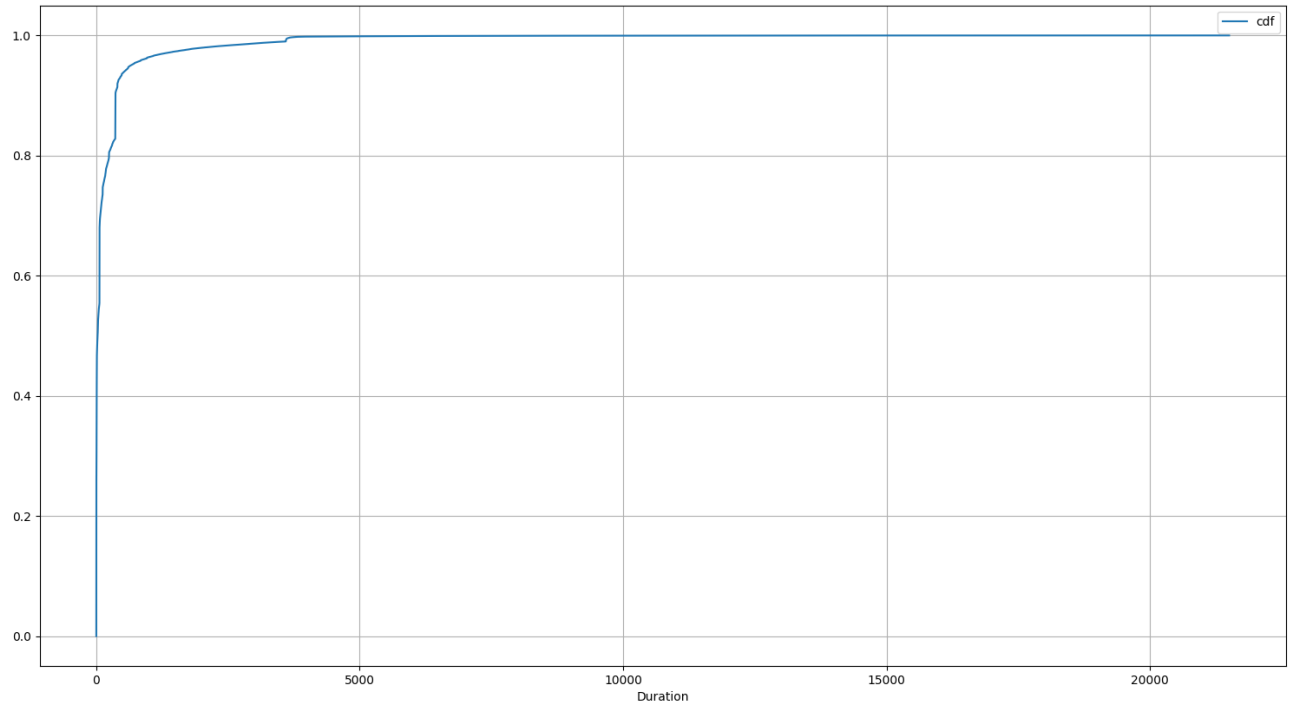


Figure 8: Connection Duration Analysis – Plotting CDF

Figure 8 shows us how is the sum of probabilities. It says that between 0s to 5000s we have more probability and after that we will reach the 99.99% of the total 100%.

Section 3:

```
amirh@AHS:~/Desktop/Winter_2022/Courses/CPSC 641/A1$ python s3.py
                Size
count   8.090020e+05
mean    5.007359e+04
std     1.444318e+06
min     1.000000e+00
25%     6.370000e+02
50%     1.634000e+03
75%     5.104000e+03
max     4.298259e+08
```

Figure 9: Transfer Size Analysis – Simple Analysis – Sent bytes

Figure 9 shows the statistics for bytes sent. As we can see, the average sent bytes is around 50000 bytes where the minimum for that is just 1 byte and the maximum is around 430000000 bytes. We can see the standard derivation too. Now, it is time to calculate PDF and CDF. Figure 10 shows them.

	Size	frequency	pdf	cdf
0	1.0	64249	0.079418	0.079418
1	2.0	21	0.000026	0.079444
2	40.0	11595	0.014332	0.093776
3	41.0	178	0.000220	0.093996
4	48.0	24	0.000030	0.094026
...	...	...	...	...
92728	283690097.0	1	0.000001	0.999995
92729	325894234.0	1	0.000001	0.999996
92730	329936770.0	1	0.000001	0.999998
92731	330975808.0	1	0.000001	0.999999
92732	429825851.0	1	0.000001	1.000000

Figure 10: Transfer Size Analysis – PDF and CDF – Sent bytes

Figure 10 means there is some transfer sizes that are more frequently used in communications. For example, size of 1 byte is frequently used (and I guess it is used for keep-alive concept) and 40 bytes is the next size which is used way more than other similar sizes (I guess it happens maybe because the headers have 40 bytes size). This was so interesting that make me to calculate and observe another thing, the most frequently used sizes. I sort the dataframe by the column of “frequency” and I got the Figure 11. We can see from Figure 11 that the top 3 most frequent ones are some rounded numbers like 1, 40, 80.

Size	frequency	pdf	cdf
1.0	64249	0.079418	0.079418
40.0	11595	0.014332	0.093776
80.0	4769	0.005895	0.106521
104.0	3802	0.004700	0.113529
52.0	2868	0.003545	0.097571
64.0	2077	0.002567	0.100426
208.0	2004	0.002477	0.137965
132.0	1739	0.002150	0.120289
120.0	1728	0.002136	0.116686
2001.0	1704	0.002106	0.565235
172.0	1644	0.002032	0.130005
212.0	1495	0.001848	0.139935
156.0	1466	0.001812	0.124369
1980.0	1439	0.001779	0.559674
92.0	1286	0.001590	0.108572
252.0	1245	0.001539	0.149144
144.0	1199	0.001482	0.122210
160.0	1184	0.001464	0.125929
168.0	1138	0.001407	0.127935
224.0	1018	0.001258	0.143343

Figure 11: Transfer Size Analysis – The most used transfer sizes – Sent bytes

Now, Figures 12 and 13 will show the PDF and CDF plots.

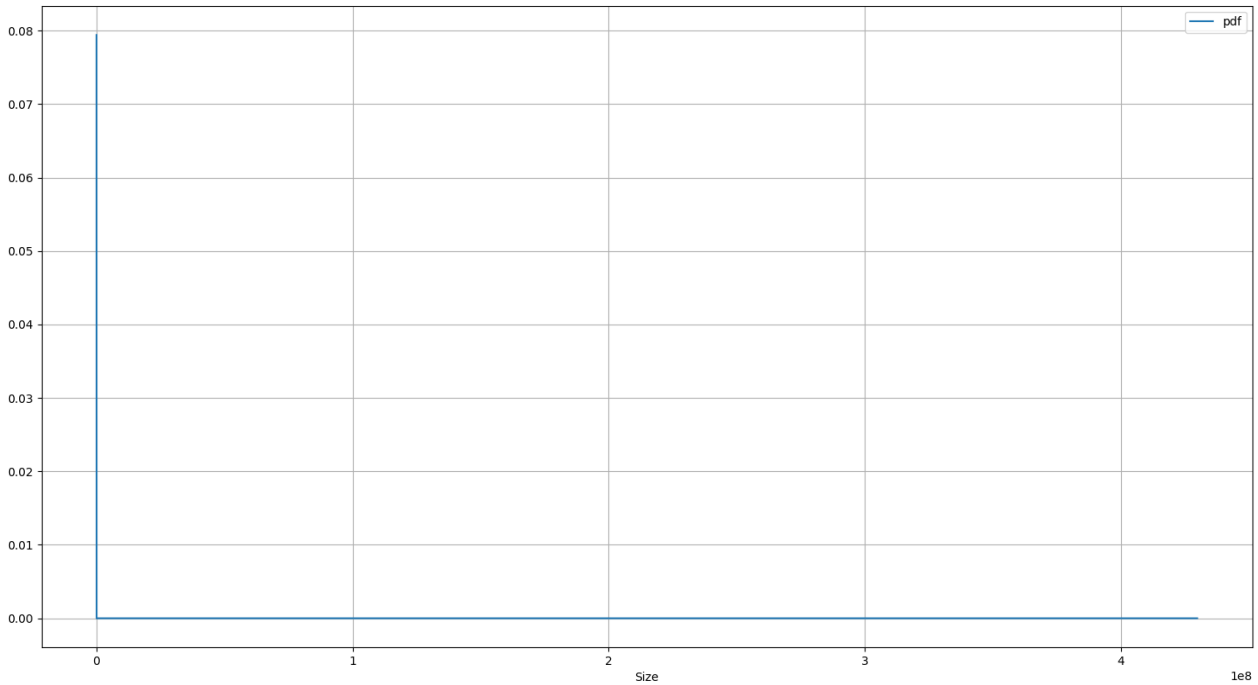


Figure 12: Transfer Size Analysis – PDF plot – Sent bytes

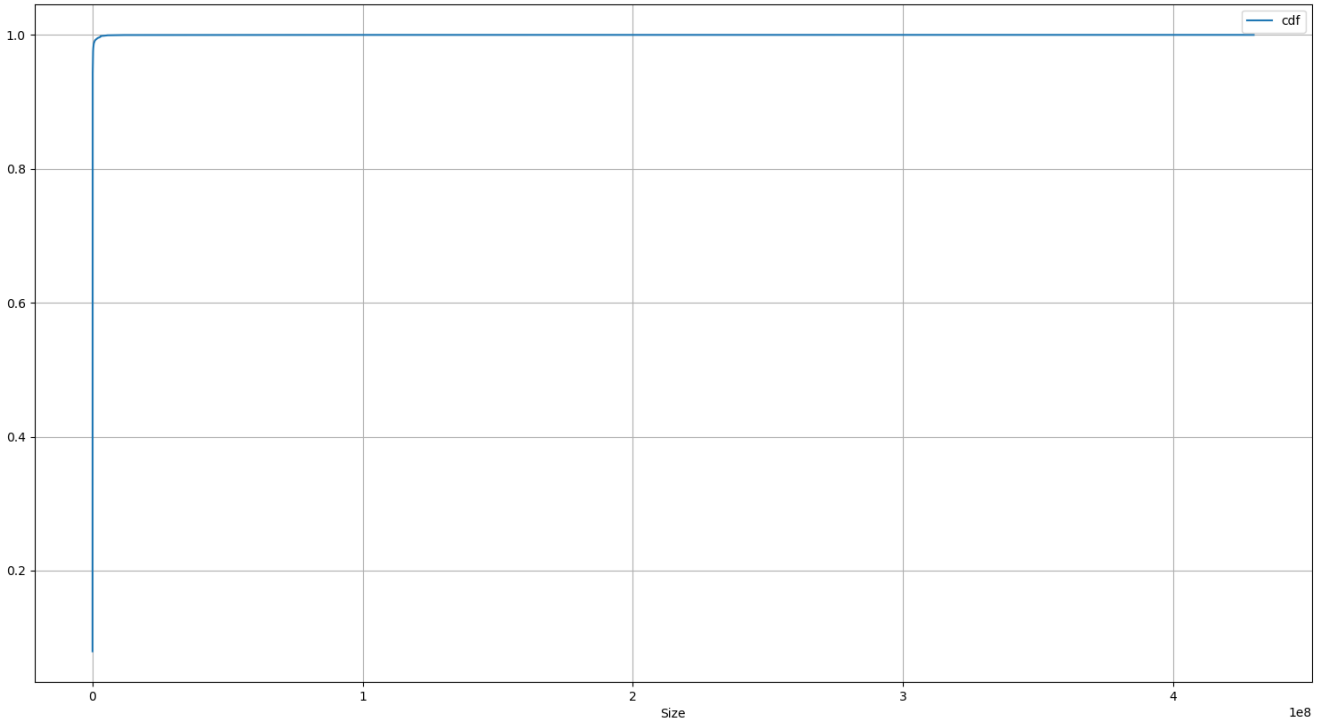


Figure 13: Transfer Size Analysis – CDF plot – Sent bytes

```
amirh@AHS:~/Desktop/Winter_2022/Courses/CPSC 641/A1$ python s3.py
                Size
count   9.341570e+05
mean    1.184950e+05
std     3.983581e+06
min     0.000000e+00
25%     8.760000e+02
50%     3.043000e+03
75%     9.109000e+03
max     3.131881e+09
```

Figure 14: Transfer Size Analysis – Simple Analysis – Received bytes

	Size	frequency	pdf	cdf
0	0.000000e+00	2580	0.002762	0.002762
1	4.000000e+01	3379	0.003617	0.006379
2	4.800000e+01	1	0.000001	0.006380
3	5.200000e+01	1538	0.001646	0.008026
4	5.800000e+01	1	0.000001	0.008028
...	...	...	...	...
130525	3.820034e+08	1	0.000001	0.999996
130526	4.434133e+08	1	0.000001	0.999997
130527	4.498261e+08	1	0.000001	0.999998
130528	4.597860e+08	1	0.000001	0.999999
130529	3.131881e+09	1	0.000001	1.000000

Figure 15: Transfer Size Analysis – PDF and CDF – Received bytes

	Size	frequency	pdf	cdf
69	181.0	57919	0.062001	0.099178
241	362.0	49131	0.052594	0.179545
30	125.0	7167	0.007672	0.027674
131	250.0	5697	0.006099	0.113547
1957	2078.0	3659	0.003917	0.415134
1	40.0	3379	0.003617	0.006379
61	169.0	3268	0.003498	0.034947
10	80.0	3005	0.003217	0.014131
27	120.0	2889	0.003093	0.019998
1196	1317.0	2648	0.002835	0.331188
217	338.0	2637	0.002823	0.124247
0	0.0	2580	0.002762	0.002762
1193	1314.0	2429	0.002600	0.328183
5	60.0	2359	0.002525	0.010553
918	1039.0	2179	0.002333	0.278923
1043	1164.0	2110	0.002259	0.300220
1168	1289.0	1862	0.001993	0.323724
793	914.0	1861	0.001992	0.258001
19	104.0	1642	0.001758	0.016583
3	52.0	1538	0.001646	0.008026

Figure 16: Transfer Size Analysis – The most used transfer sizes – Received bytes

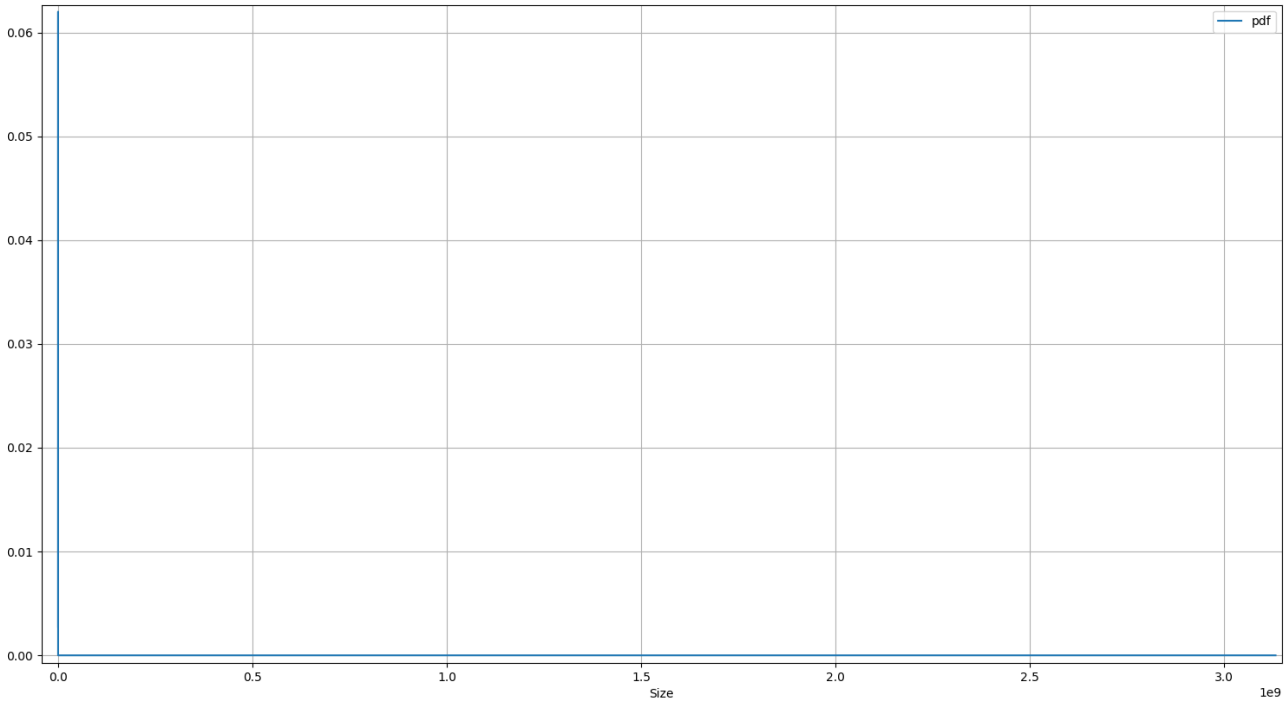


Figure 17: Transfer Size Analysis – PDF plot – Received bytes

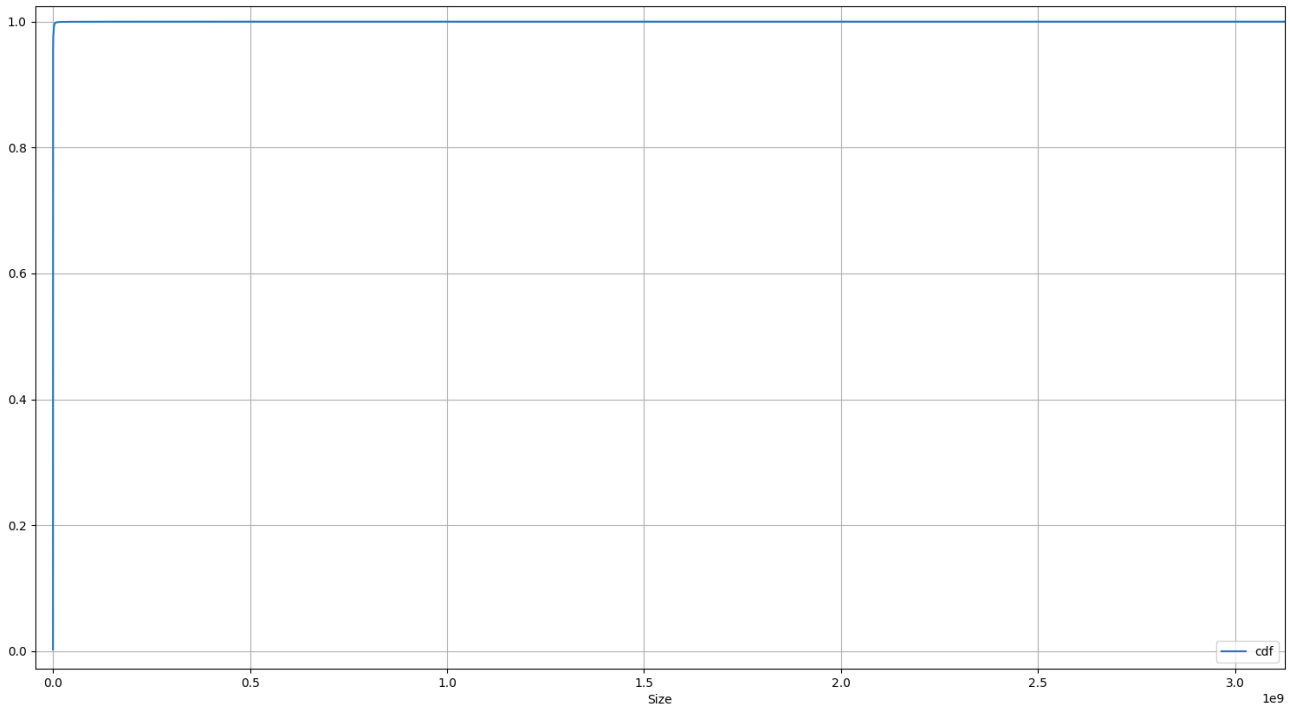


Figure 18: Transfer Size Analysis – CDF plot – Received bytes

Figure 12 and 13 show all the observations that had been discussed in the previous paragraph. Now, it is time to focus on the bytes received instead of bytes sent. Figure 14, 15, 17, and 18 show statistical analysis of bytes received.

Section 4:

```
amirh@AHS:~/Desktop/Winter_2022/Courses/CPSC 641/A1$ python s4.py
              ClientIP  counts
0     136.159.160.125   15715
1     136.159.160.121   12065
2     198.161.243.253    9522
3     198.161.243.251    9384
4     198.161.243.254    8559
...               ...     ...
2099  136.159.146.140       1
2100    136.159.85.55       1
2101     136.159.73.9       1
2102  136.159.222.133       1
2103    136.159.85.60       1
```

Figure 19: Client Analysis - Number of connections used by each client IP address

Figure 19 shows how many connections were used by each client. So, we can see what clients are using more TCP connections. But it can't show what client is using more network bandwidth because it is possible to send or receive more data just by one TCP connection. Figure 20 shows the plot of previous data analyze. (Clients ranks – Number of connections)

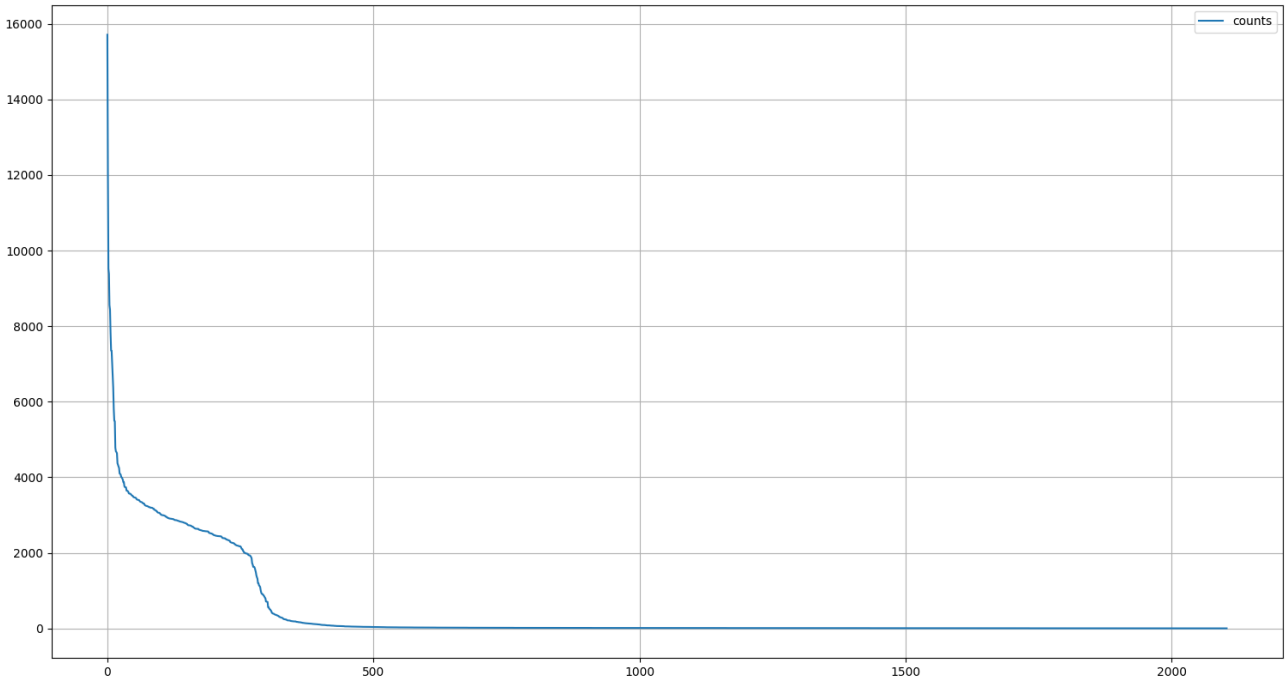


Figure 20: Client Analysis – Rank of clients vs number of connections created by them.

Now, for comparing the number of connections with sent/received bytes of data for each of the clients, I will try to make a sum function for each client IP and calculate all of sent and received bytes each of clients use. Then I combine two data frames with each other to make an understandable data frame. The final data frame is shown in the Figure 21.

```
amirh@AHS:~/Desktop/Winter_2022/Courses/CPSC 641/A1$ python s4.py
             ClientIP  counts        sent        recv
0     136.159.160.125   15715    89911471   383505457
1     136.159.160.121   12065  1736229965  2922491497
2     198.161.243.253    9522    86275403   473307977
3     198.161.243.251    9384   133145984  2709284843
4     198.161.243.254    8559   209085399   769117881
...               ...     ...         ...         ...
2099  136.159.146.140       1        2887        7106
2100    136.159.85.55       1         491        1116
2101     136.159.73.9       1         755        1194
2102  136.159.222.133       1         812       25821
2103    136.159.85.60       1        1422         920
```

Figure 21: Client Analysis – Sent and Received bytes for each client.

Figure 21 shows a very interesting point, the client which used the most number of TCP connections is not necessarily the same client which sent or received the most bytes. For example, the client with index of 0, has used 15715 TCP connections in one day of analysis while the client with index of 4 has used 8559 TCP connections in the same period of time. But, the latter client has sent more than the first one and also has received more bytes too. (209085399 bytes compared to 89911471 bytes and 769117881 bytes compared to 383505457 bytes)

Figure 22 and 23 show the clients sorted by number of connections and also their sent and received bytes all together to show us how they can be different.

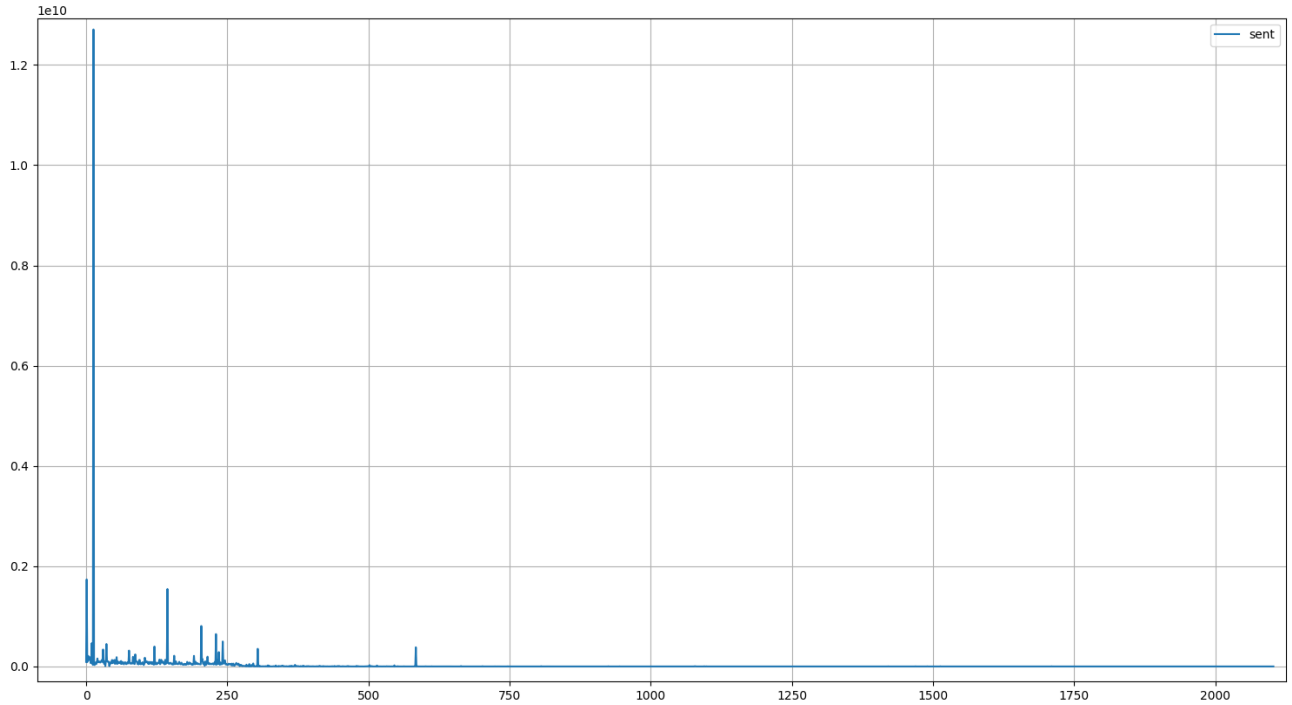


Figure 22: Client Analysis – Sent bytes vs Client rank sorted by number of connections

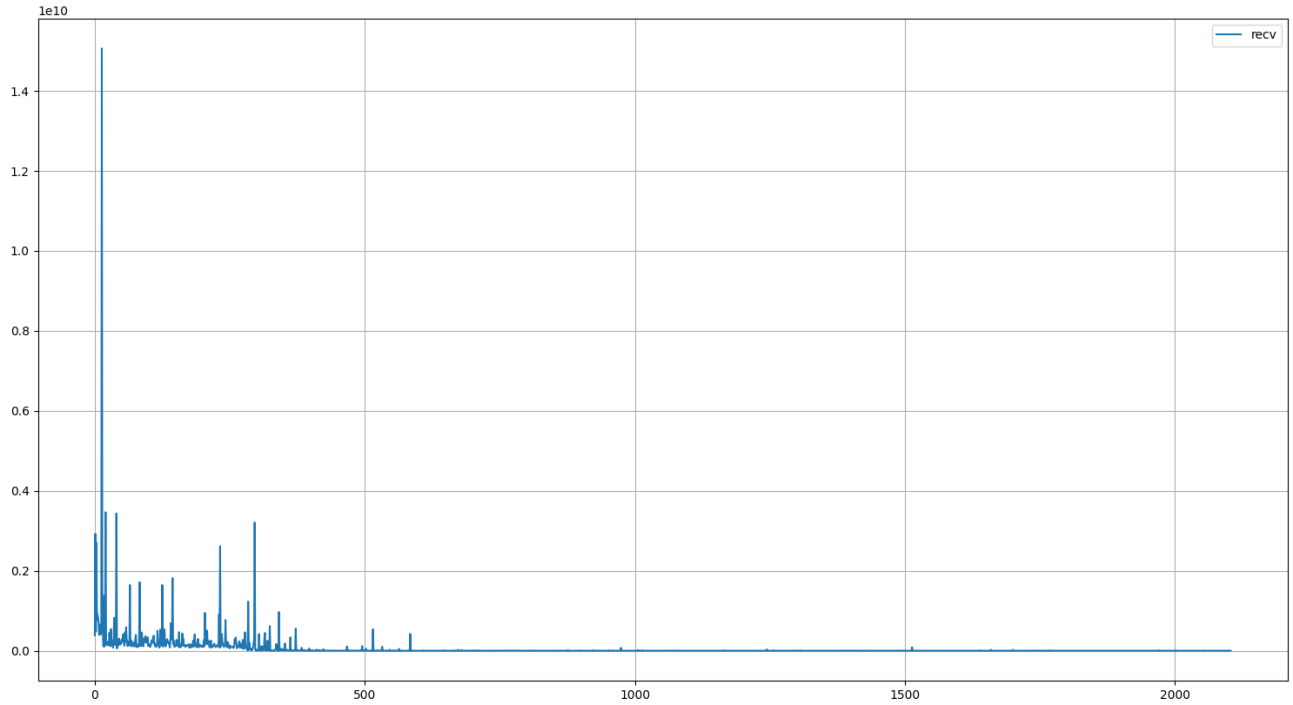


Figure 23: Client Analysis – Received bytes vs Client rank sorted by number of connections

Figure 22 and 23 show us that although we can say there is a relation between “Number of TCP connections” and “Sent/Received byte”, but we can’t say this relation is too strict because we can see in the top 500 clients this relation is not always correct and we have many clients which didn’t used many TCP connections but they sent or received many bytes. For example, from Figure 18 we can see there are some clients in the range between 500-1000 (which is sorted by the number of used TCP connections) received more bytes than some clients in the range between 0-500.

The same thing happens in Figure 22 as well, we can see there are some clients in the range between 500-750 (which is sorted by the number of used TCP connections) sent more bytes than some clients in the range between 0-500.

Section 5:

```
amirh@AHS:~/Desktop/Winter_2022/Courses/CPSC 641/A1$ python s5.py
            ServerIP  counts
0     35.186.224.25  566019
1       34.98.74.57  116220
2     23.220.167.17   99921
3     35.186.224.19   74924
4     23.220.167.66   32731
5     104.16.148.64   23818
6     35.186.224.44   19061
7     151.101.65.21    1798
8     35.186.224.18     431
9     35.186.224.45     414
10   151.101.124.84     188
11  151.101.126.248     120
12   151.101.126.91      79
13  151.101.124.201      13
14  151.101.126.249       2
```

Figure 24: Server Analysis - Number of connections used by each server IP address

Figure 24 has two important points. First is the number of servers are lower than the clients which is normal and obvious because there should be one server for Spotify and 14 other servers which they work as CDN of their network. The second point is I locate all of these 15 servers and I found that 13 of them were located in USA and 2 of them were in Canada. It is kind of normal because the population of USA is 10 times more than Canada's population. (And probably it can help them to make their service fair in orders of speed and availability for all)

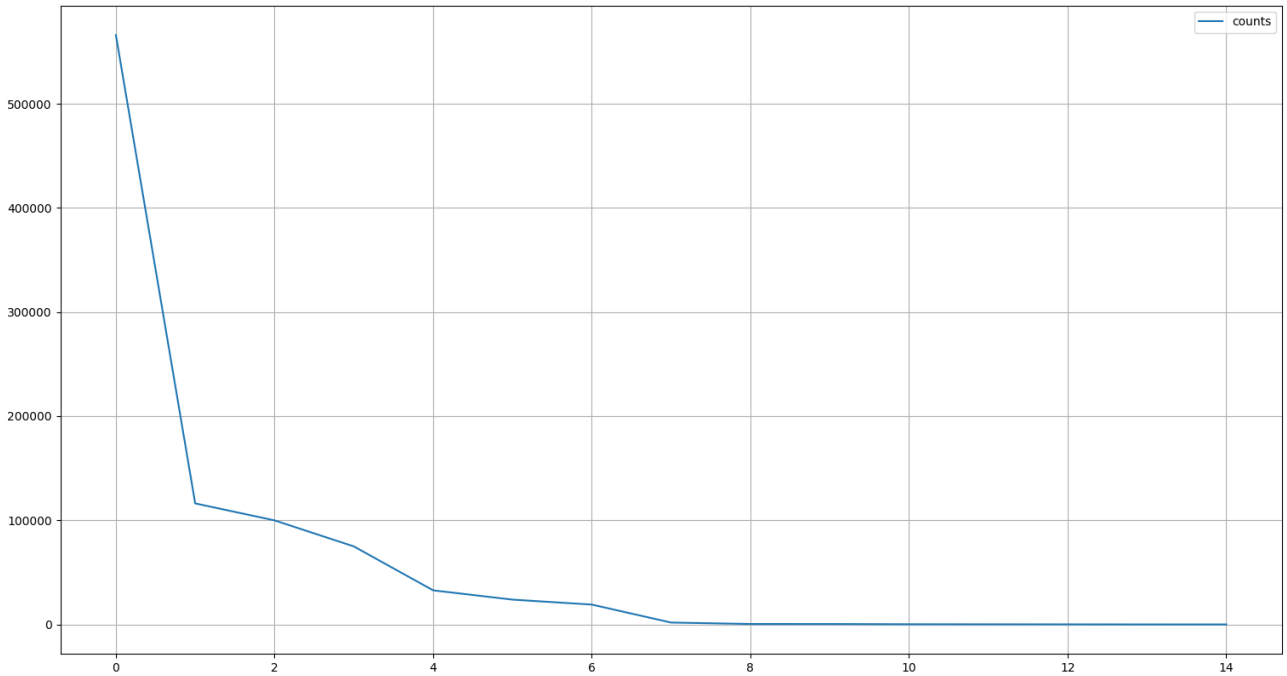


Figure 25: Server Analysis – Rank of servers vs number of connections used by them.

Figure 25 shows a visualized ranking of each server versus the number of TCP connections they used.

```
amirh@AHS:~/Desktop/Winter_2022/Courses/CPSC 641/A1$ python s5.py
            ServerIP  counts        sent        recv
0     35.186.224.25  566019  38595503389  54574087209
1       34.98.74.57  116220    103056562    296157721
2     23.220.167.17   99921    395667715  17209811859
3     35.186.224.19   74924    510283004    625249563
4     23.220.167.66   32731    725213367  35991797573
5     104.16.148.64   23818     44748105    820613017
6     35.186.224.44   19061    114226237   1091284106
7     151.101.65.21    1798     17938213     55571965
8     35.186.224.18     431       578696      1571906
9     35.186.224.45     414       299232      1362941
10   151.101.124.84     188      1094124      5038241
11  151.101.126.248     120       674832     19342417
12   151.101.126.91      79       320119       743185
13  151.101.124.201      13        24530       289606
14  151.101.126.249       2         3113         4815
```

Figure 26: Server Analysis – Sent and Received bytes for each server.

Figure 26 shows how many bytes a server sent or received, again we can say there is a “not strict” relation between the number of TCP connections and sent/received bytes. The only difference here is the number of servers are limited so the bytes received or sent is much higher because the servers actually act as single points where many clients would communicate through them.

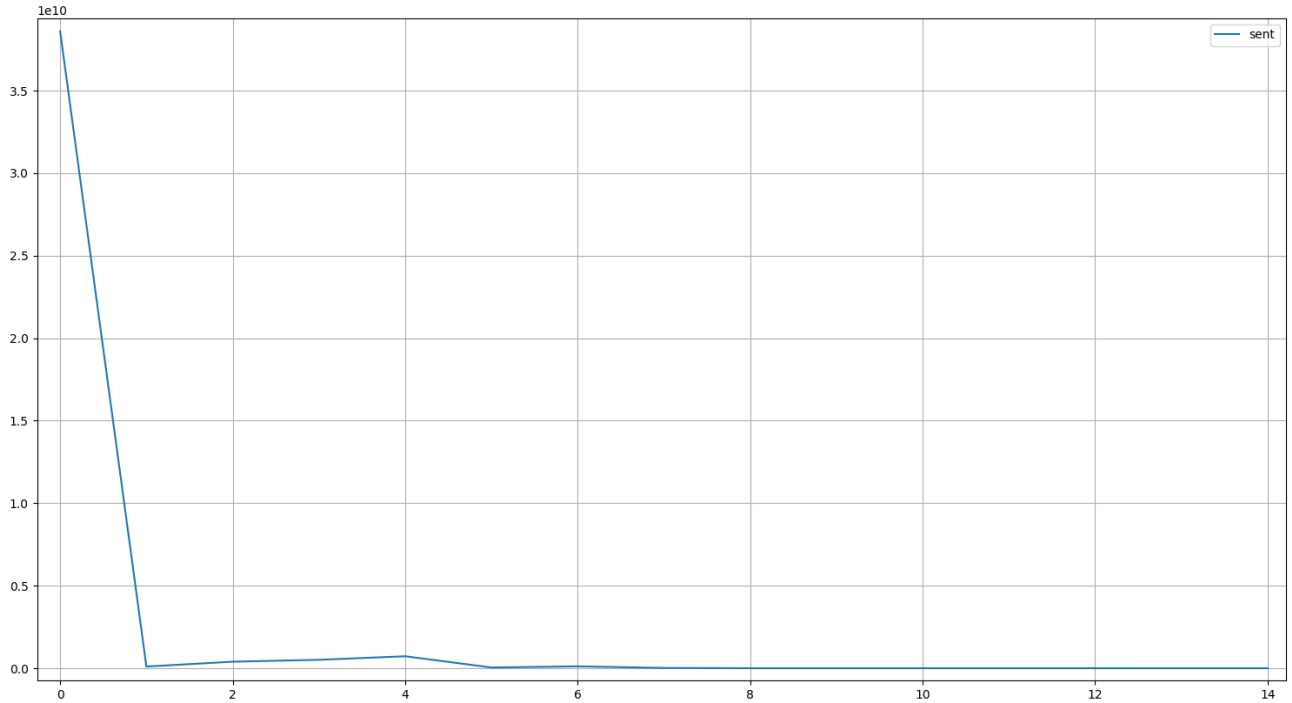


Figure 27: Server Analysis – Sent bytes vs Server rank sorted by number of connections

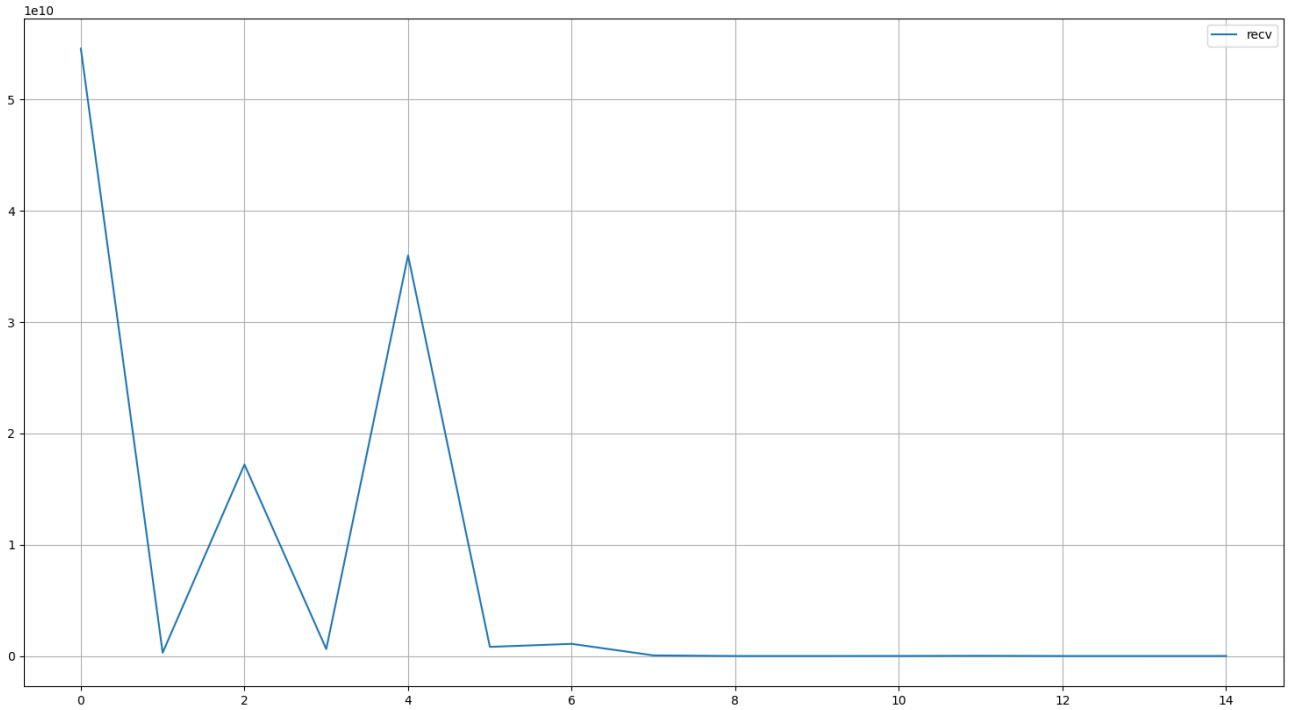


Figure 28: Server Analysis – Received bytes vs Server rank sorted by number of connections

As Figure 28 shows, the received bytes is not so related to the number of connections a server used. For example, the server with index 4 received much more bytes than servers with indexes 1, 2, and 3.

Section 6:

For the idea of this section, I choose to use the column that get us the code for the connection status. First, I should know about all of these signs in that column. Table 1 shows definition of each of these signs.

Table 1: Definitions of signs in the connection-level data [1]

Sign	Definition
S0	Connection attempt seen, no reply.
S1	Connection established, not terminated.
SF	Normal establishment and termination. Note that this is the same symbol as for state S1. You can tell the two apart because for S1 there will not be any byte counts in the summary, while for SF there will be.
REJ	Connection attempt rejected.
S2	Connection established and close attempt by originator seen (but no reply from responder).
S3	Connection established and close attempt by responder seen (but no reply from originator).
RST0	Connection established, originator aborted (sent a RST).
RSTR	Responder sent a RST.
RST0S0	Originator sent a SYN followed by a RST, we never saw a SYN-ACK from the responder.
RSTRH	Responder sent a SYN ACK followed by a RST, we never saw a SYN from the (purported) originator.
SH	Originator sent a SYN followed by a FIN, we never saw a SYN ACK from the responder (hence the connection was “half” open).
SHR	Responder sent a SYN ACK followed by a FIN, we never saw a SYN from the originator.
OTH	No SYN seen, just midstream traffic (a “partial connection” that was not later closed).

Now, the number of connections which have specified sign has been counted. Figure 29 shows this.

```
amirh@AHS:~/Desktop/Winter_2022/Courses/CPSC 641/A1$ python s6.py
      Sign  counts
0       SF  311184
1      SHR  193173
2       S1  121221
3       S3  109018
4     RSTO   84087
5      OTH   38125
6       S2   34922
7   RSTOS0   24525
8       SH    9583
9     RSTR    6112
10   RSTRH    1900
11      S0    1347
12     REJ     542
```

Figure 29: Type of Connection Analysis – Number of each type

As we can see from Figure 29, SF type connection is the most one. SF is used when the connection and termination is normal, so it shows us most of the communications were normal. After that, SHR is the most used one. It is for a normal connection too and the only problem was a SYN from originator is lost. S1 connections are the next most used one in which they are the connections that didn't happen to terminate. So, the top 3 of the list is for the connections where the communication were normal. Figure 30 shows the visualization of Figure 29 data.

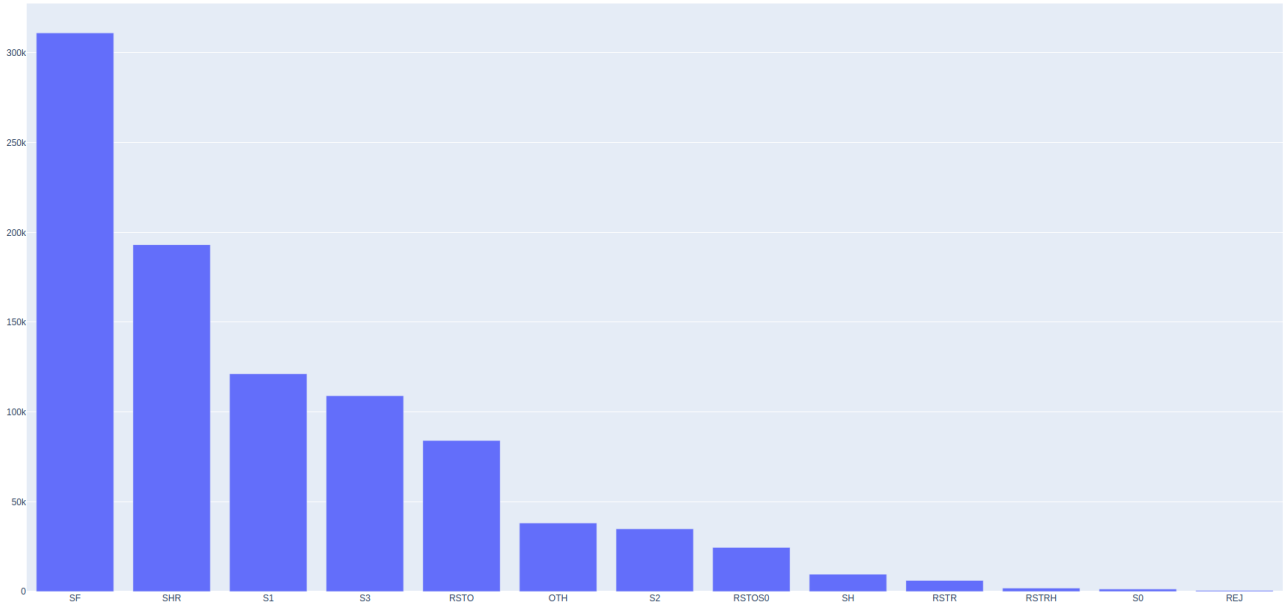


Figure 30: Type of Connection Analysis – Type of connection vs Number of occurrence

Also, the other thing was about the bytes sent or received when we are focusing on the type of the connection. For example, I thought that because type of "S0" occurs when connection attempt is seen but there is no reply, it should have less bytes sent and received compared to type of "REJ", because in the second type, we get the result so the bytes sent/received should be higher. Figure 31 surprisingly shows that actually happens.

```
amirh@AHS:~/Desktop/Winter_2022/Courses/CPSC 641/A1$ python s6.py
      Sign  counts          sent          recv
0       SF  311184  22796063820  59885608817
1      SHR  193173    391854818    966900363
2       S1  121221   4905406464  11448575614
3       S3  109018   2845436533   7197494616
4     RSTO   84087   5590991228  18763110451
5      OTH   38125    973773415   2083518631
6       S2   34922   1707589621   3660647768
7   RSTOS0   24525    817305844   1746142310
8       SH    9583    260389069    562734384
9     RSTR    6112    199677054   4022796074
10   RSTRH    1900     10431310     70567700
11      S0    1347       676919        61528
12     REJ     542     10035143    284767868
```

Figure 31: Type of Connection Analysis – Type of connections sorted by counts with bytes sent/received

Figure 31 shows that the comparison between “S0” and “REJ” types are actually correct and the bytes sent/received by type of “REJ” is much more than type of “S0”.

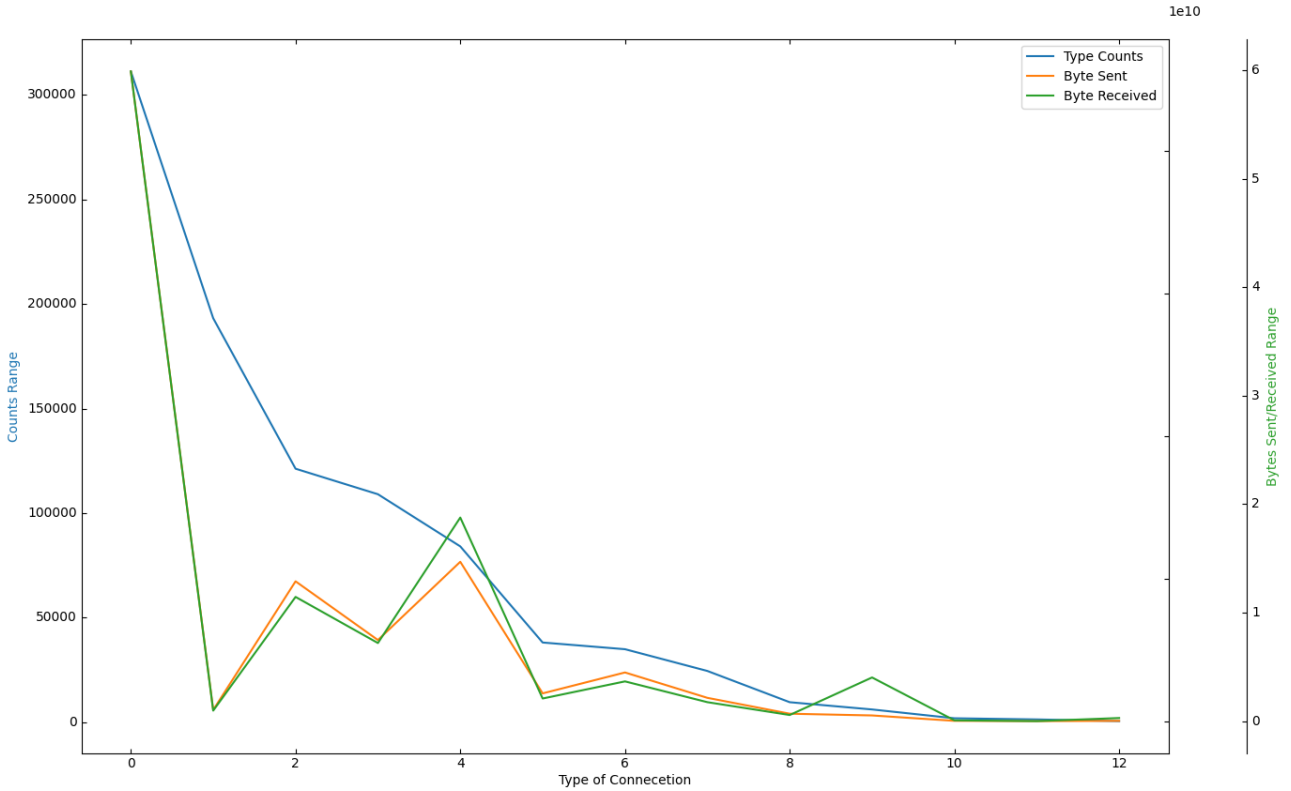


Figure 31: Count of Types vs Bytes Received vs Bytes Sent – Types are sorted by counts.

References

[1] Ian Howson, “conn-log connection state values and descriptions”