COL334 Computer Networks

Assignment 3 Packet Trace Analysis

Submitted By: Kumar Prithvi Mishra (2016MT10618), Ayush Chaurasia (2016MT10617)

Questions

Ques. What different FTP commands and response codes to you see? Use the FTP RFC to describe these commands briefly.

Ans. FTP commands have two types.

1) Request Commands

2) Response Codes

FTP server return codes always have three digits, and each digit has a special meaning.

https://en.wikipedia.org/wiki/List_of_FTP_server_return_codes

Ques 1. How many unique server IPs do you see? How many unique client IPs?

Ans.

Day	03-01-11	03-01-14	03-01-18
Client IPs	522	939	510
Server IPs	45	50	89

Ques 2. How many unique TCP flows do you see?

Ans. Assuming Server-client & Client-Server considered different flows. And a unique 4-tuple is a different flow.

Day	03-01-11	03-01-14	03-01-18
TCP Flows	3256	5422	3280

Ques 3. Draw a plot of the number of connections (TCP flows) opened to any FTP server within 60min windows over the 24-hour duration. How can you use these traffic profiles to detect if the system is under a DoS attack?

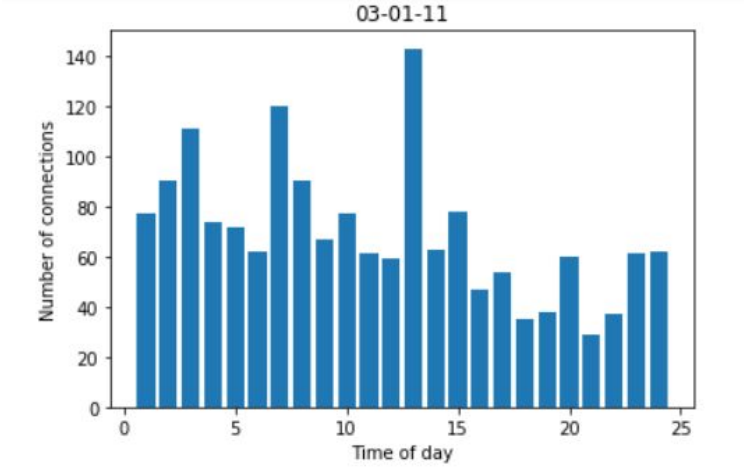
Ans. The flows are displayed on the next page.

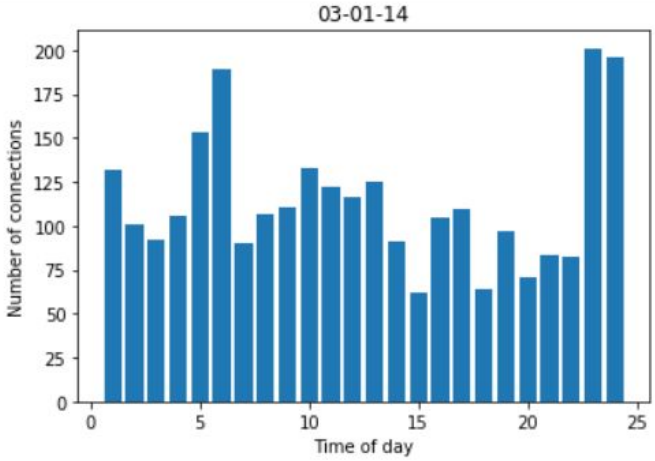
In computing, a **denial-of-service attack** (**DoS attack**) is a cyber-attack in which the perpetrator seeks to make a machine or network resource unavailable to its intended users by temporarily or indefinitely disrupting services of a host connected to the Internet. Denial of service is typically accomplished by flooding the targeted machine or resource with superfluous requests in an attempt to overload systems and prevent some or all legitimate requests from being fulfilled.

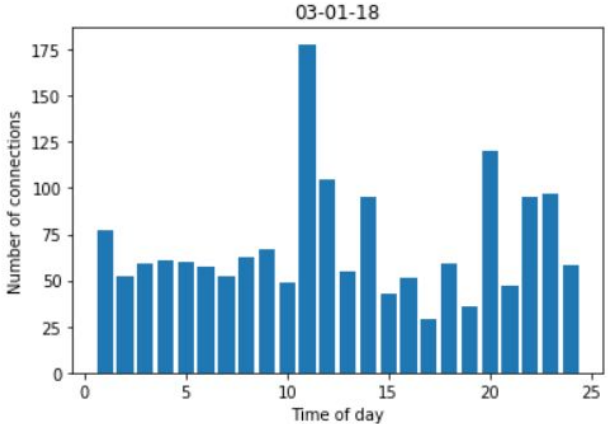
(Ref: https://en.wikipedia.org/wiki/Denial-of-service_attack)

From the flows, we might say that the traffic profiles of 3-01-11 and 3-01-18 are quite similar. And since they are 7 days apart, we may guess that traffic on a particular day of the week is quite similar to last week. Accordingly, we can collect traffic data for each day from previous weeks, and compare the traffic for any particular day's data to previous week, if it is dissimilar to that data then we may declare it as suspicious.

Also. as we can see, the flow count may jump up or down abruptly at any hour in a day. Hence, using big jumps as a marker for suspicion is a bad idea.


03-01-11
Number of connections
Time of day
0
20
40
60
80
100
120
140
0
5
10
15
20
25


03-01-14
Number of connections
Time of day
0
25
50
75
100
125
150
175
200
0
5
10
15
20
25


03-01-18
Number of connections
Time of day
0
25
50
75
100
125
150
175
0
5
10
15
20
25

Ques 4. For all the connections, find the duration over which a connection was kept open, and plot the CDF of these connection durations. Notice how most connections are of a short duration. Why do you think so?

Ans. The CDFs are displayed on the next page.

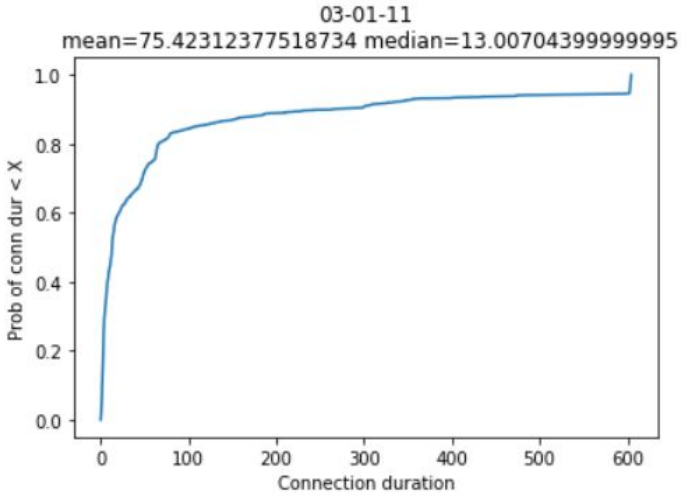
The mean and median is mentioned along with the plot.

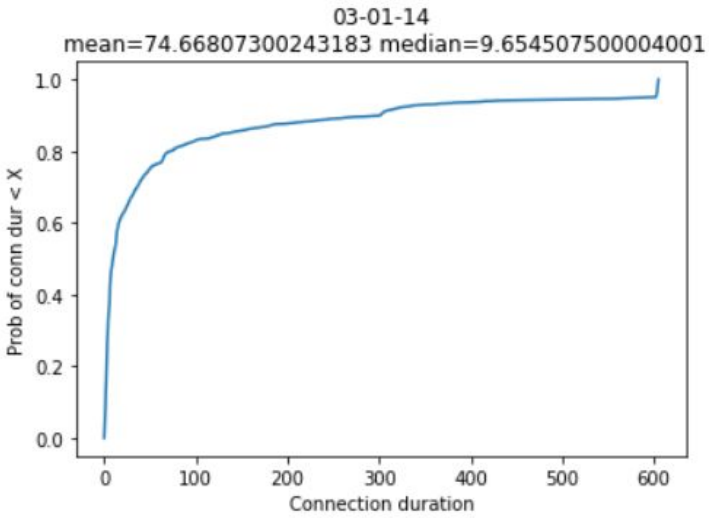
The mean is high because of the outliers.

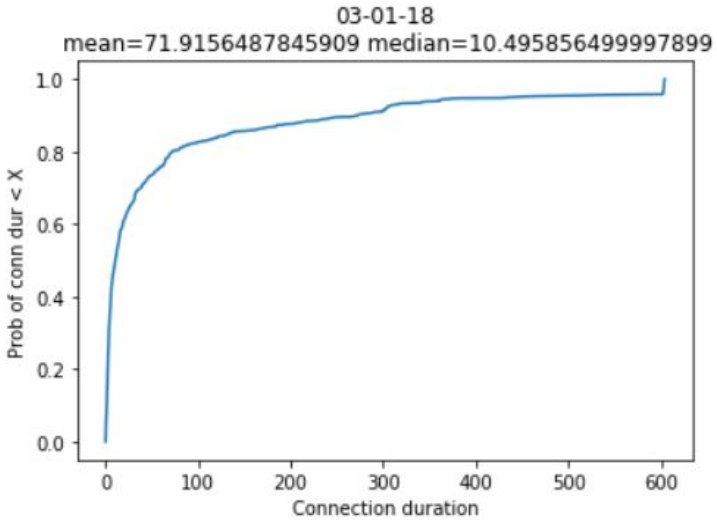
From the median, we can infer that most connections are of a short duration.

This might be because of using ftp.

Most of the connections are short because they are from a non-persistent HTTP(control packets have non-persistent HTTP connections).


03-01-11
mean=75.42312377518734 median=13.00704399999995
Prob of conn dur < X
Connection duration
0.0
0.2
0.4
0.6
0.8
1.0
0
100
200
300
400
500
600


03-01-14
mean=74.66807300243183 median=9.654507500004001
Prob of conn dur < X
Connection duration
0.0
0.2
0.4
0.6
0.8
1.0
0
100
200
300
400
500
600

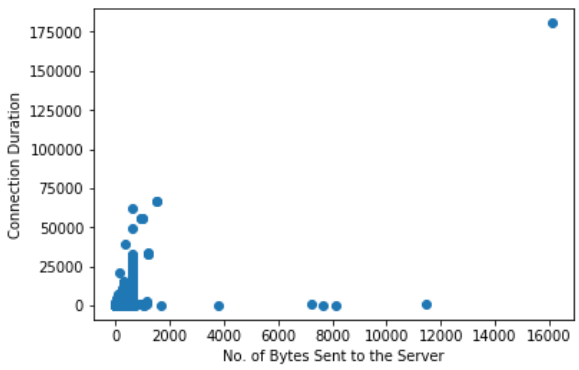

03-01-18
mean=71.9156487845909 median=10.495856499997899
Prob of conn dur < X
Connection duration
0.0
0.2
0.4
0.6
0.8
1.0
0
100
200
300
400
500
600

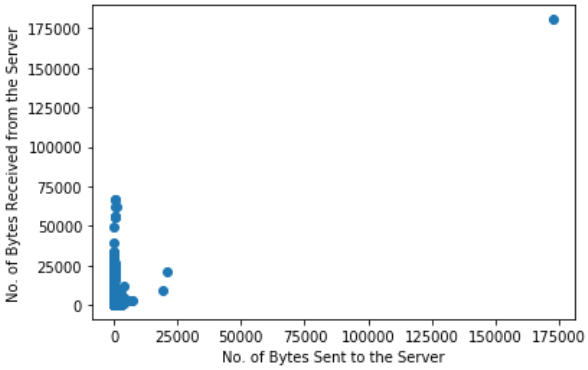
Ques 5. On the same lines as above, for all the connections, find the number of bytes sent and received over the connection, and check if there is a correlation between the connection duration and the number of bytes sent to the servers. Similarly check if there is a correlation between the number of bytes sent to the servers and the number of bytes received from the servers.

To find the correlation, you can use the Pearson's correlation coefficient assuming that it is likely to find a linear relationship between the variables. You should also make a scatterplot of the two variables.

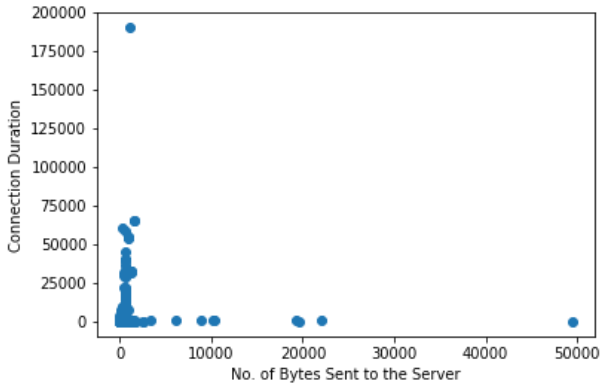
Do you find that there is a correlation? Does the scatterplot help you identify any outliers? If you eliminate the outliers, does your correlation improve?

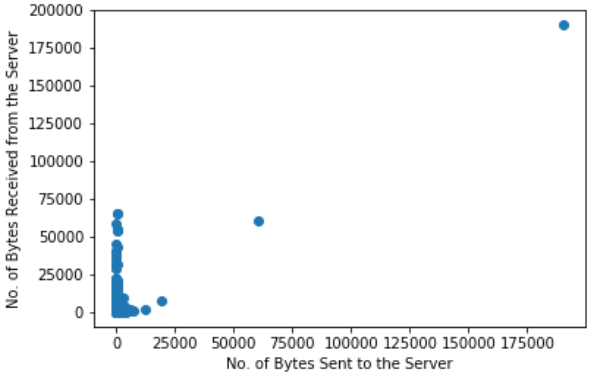
Ans. Day 1



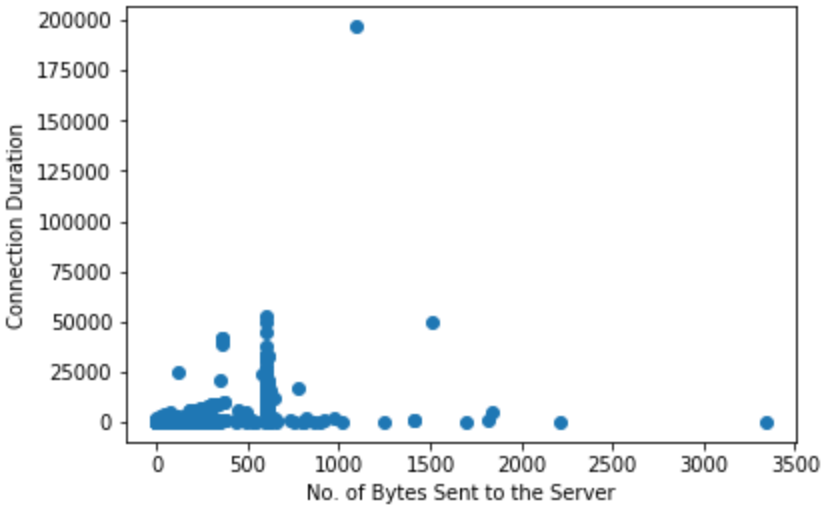


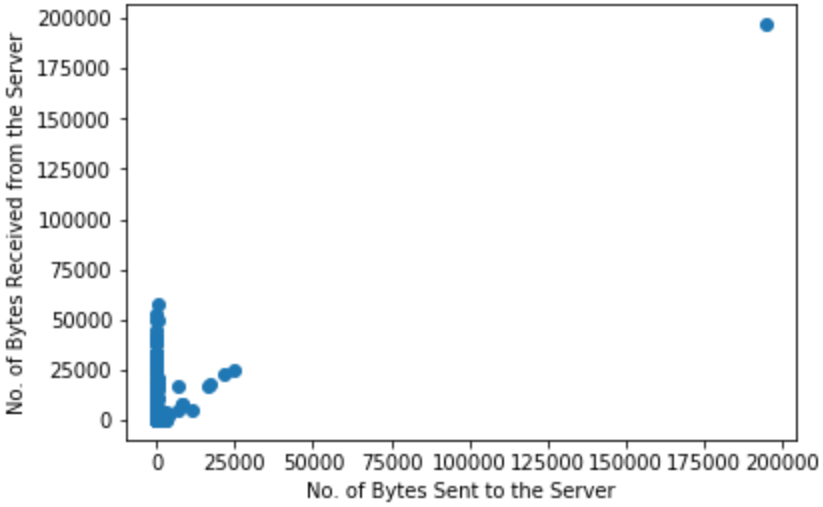
Day 2





Day 3





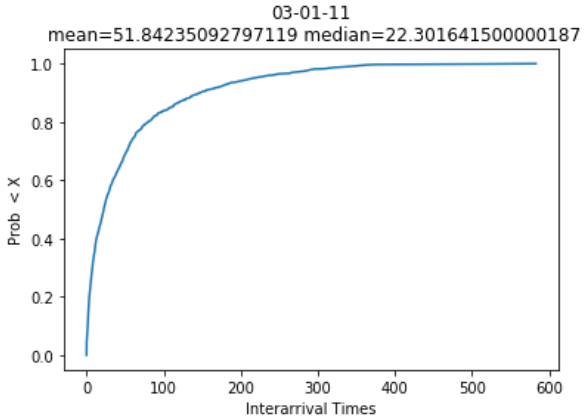
We couldn’t infer anything from these scatter plots.

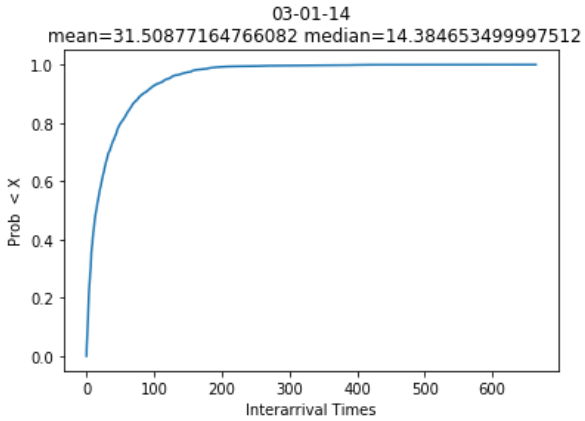
Ques 6. Plot the CDF of the inter-arrival times between two consecutive connections being opened, along with the mean and median values.

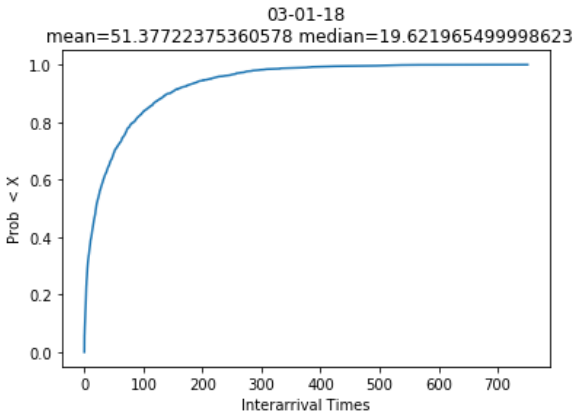
Ans. The plots below describe the CDF of the Inter-Arrival time on different days.

First, observe that these plots perfectly match the CDF of an exponential distribution, which makes sense, since Inter-arrival time in such a scenario can be expected to match an Exponential Distribution.

Also, since 03-01-11 and 03-01-18 occur on the same weekday and henceforth follow the same CDF also evident from almost the same mean and median of both dates.





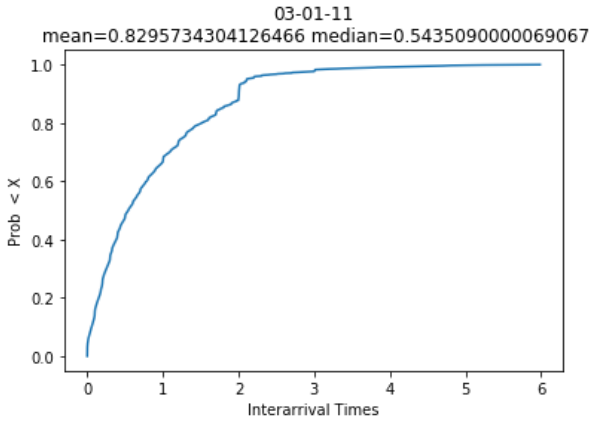


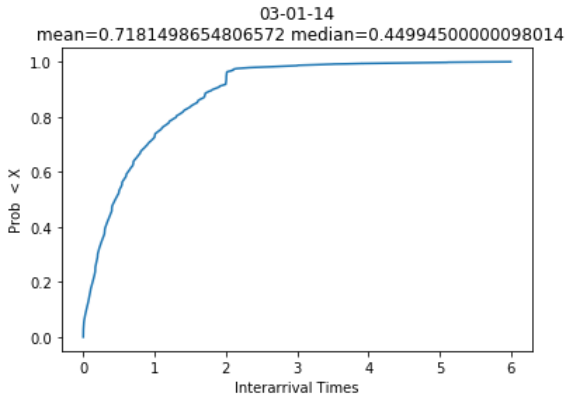
Ques 7. Plot the CDF of the inter-arrivals times of incoming packets to the servers, and report the mean and median values.

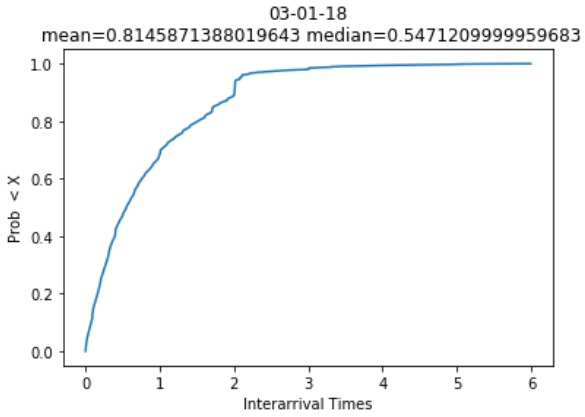
Again notice that most interarrival times are short, but the range is quite wide between the maximum and minimum interarrival times. Why do you think this is the case?

Ans. The next 3 plots show the CDF of the inter-arrival time of packets arriving to the servers.

From question 3, we saw that number of connections varied a lot with time, hence the interarrival times are expected to have high variance. Also, the exponential distribution allows all values to occur, be it large or small with different probabilities. Hence these outliers are quite likely to exist because of our huge sample space.





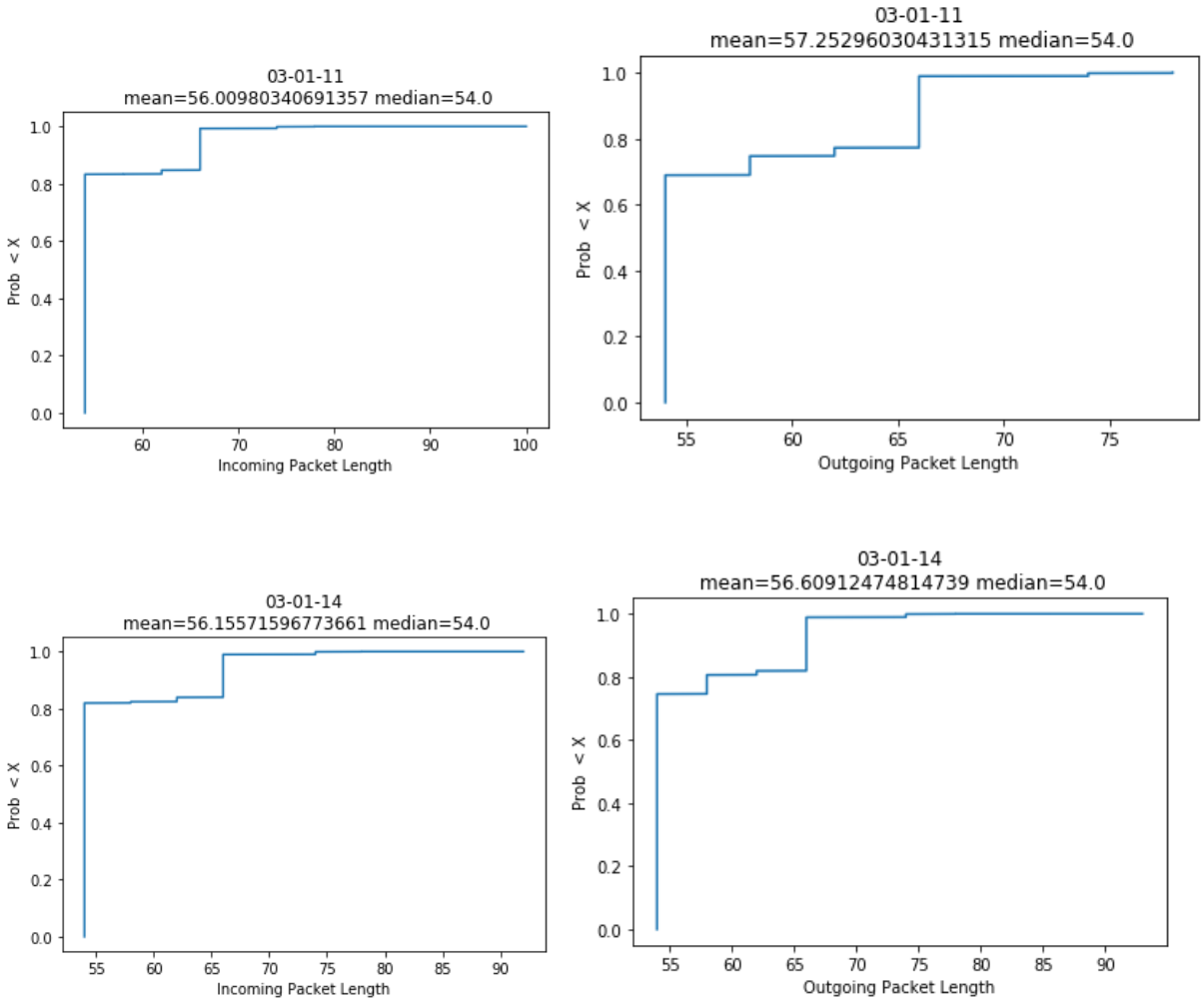


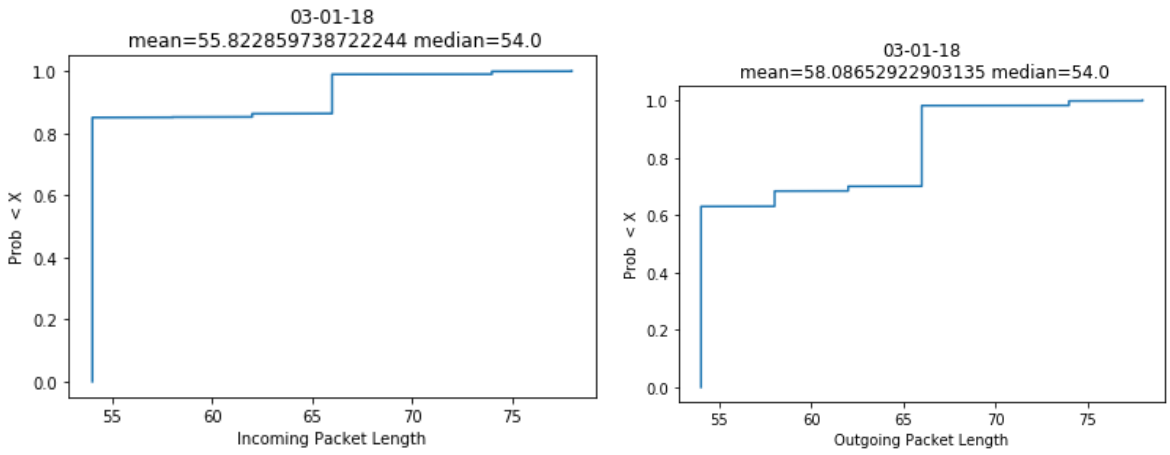
Ques 8. Plot the CDF of the packet lengths for the incoming and outgoing packets separately. Does the CDF appear to be clustered around a few values? Why do you think this is the case?

Ans. The plots below contain the required CDFs.

Yes, the CDF appear to be clustered around a few values, this might be due to the design of the header systems. The TCP packets for a connection setup are expected to cluster around a value, with the variance from the optional packet settings.

Likewise, along the control line for FTP, clustering based on standard commands sent and received is expected.





Ques 9. Write a tool that given a flow identified by the four tuple, you can generate a sequence number plot for the flow.

Show the sequence number plots for any two of the most data-intensive connections identified in question 5 above that appear interesting.

Identify some shorter connections where packet retransmissions took place, and show the sequence number plots for two of these connections. Explain what the retransmissions look like in the plot. You can write a separate analysis tool to identify flows in which retransmissions happened, and then show the sequence number plot for these flows.

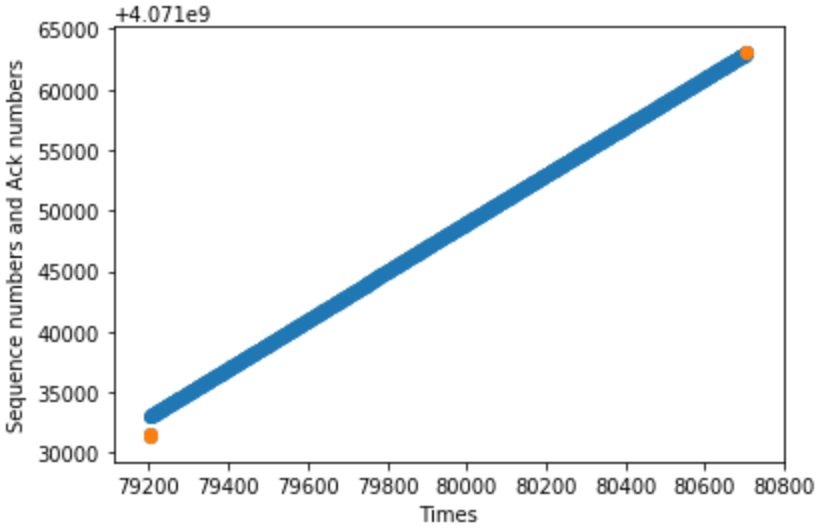
Similarly, identify some connections where spurious retransmissions happened, and show the sequence number plots for two of these flows. Explain the plots too. You can spot spurious retransmissions in this trace because it has captured both acknowledgements as well as data packets. You can write a separate analysis tool to identify flows in which spurious retransmissions happened.

In the same way, show two sequence number plots for flows where duplicate acknowledgements are seen, and explain the plots.

Show two sequence number plots for flows where out-of-order delivery has happened, and explain the plots.

Ans. Most Data Intensive connections.

For Day 1

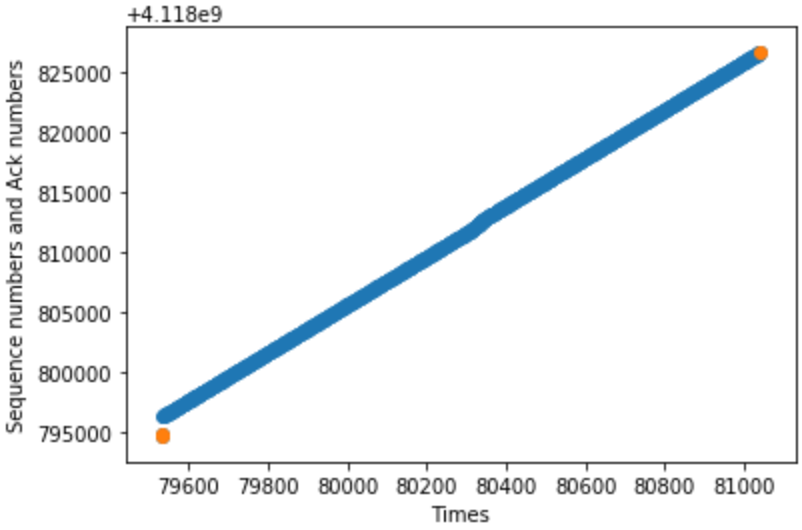


client_ip = '92.209.69.32'

server_ip = '131.243.2.12'

client_port = '1772'

server_port = '21'



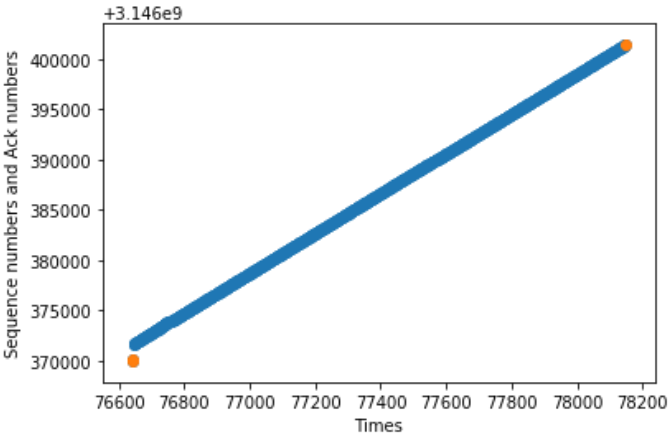
client_ip = '92.209.69.32'

server_ip = '131.243.2.12'

client_port = '1550'

server_port = '21'

Day 2

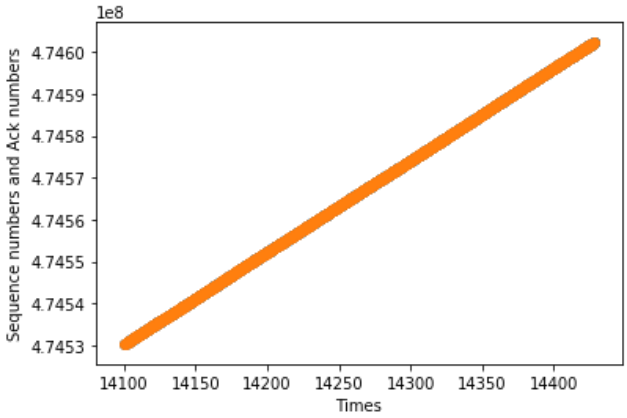


client_ip = '92.209.69.32'

server_ip = '131.243.2.12'

client_port = '1690'

server_port = '21'



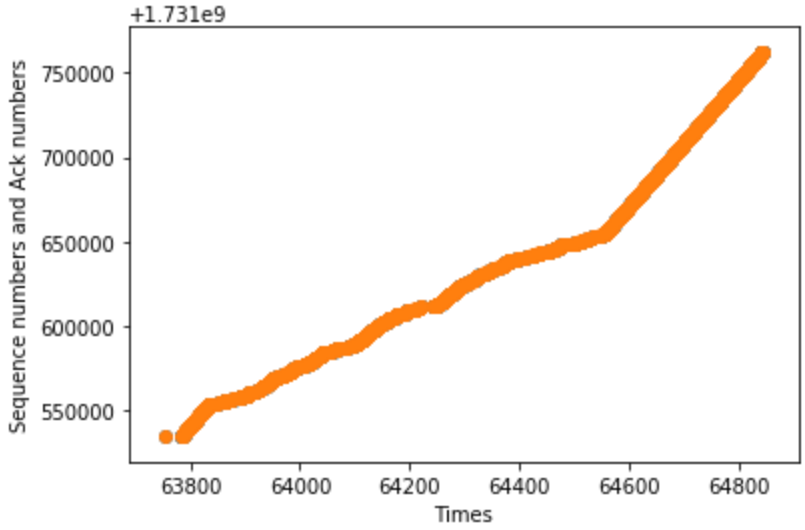
client_ip = '207.190.63.72'

server_ip = '128.3.28.48'

client_port = '34973'

server_port = '21'

Day 3



client_ip = '171.1.7.162'

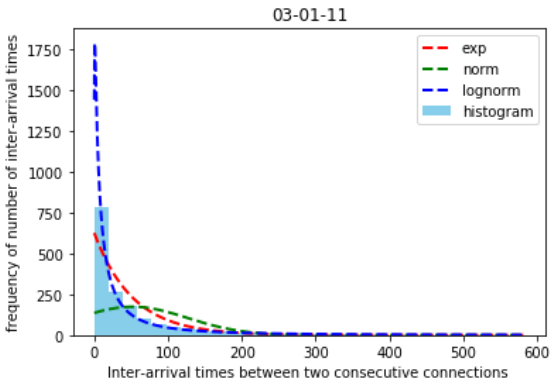
server_ip = '128.3.28.48'

client_port = '33682'

server_port = '21'

Ques 10. For questions 6 and 7 above, we next want to fit probability distributions to the data. Report the plots and parameter estimates.

Ans. PDF fitting for question 6th



Parameters:

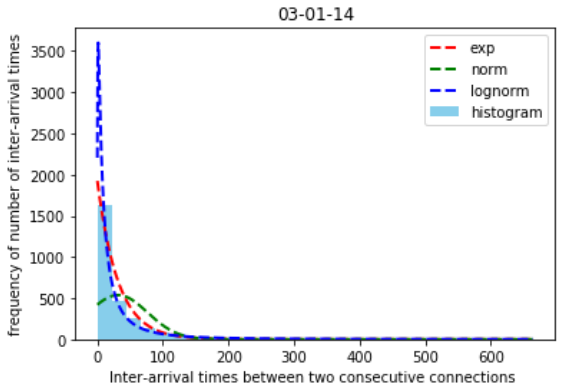
Exp mean(1/lambda)= 51.84235092797119

Normal mean= 51.84235092797119

Normal std. dev = 74.48032317547579

LNorm shape(exp(mean)) = 1.7004312462006066

Scale(std. dev)= 18.038994387688962



Parameters:

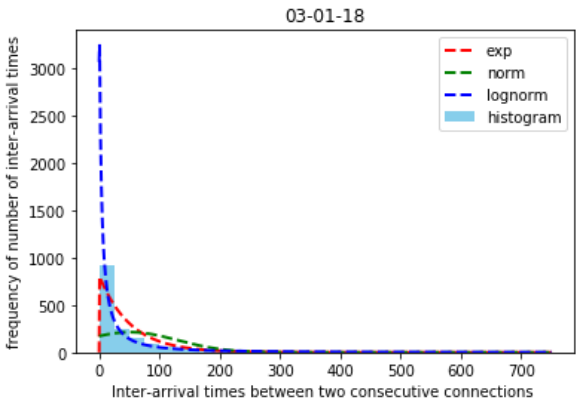
Exp mean(1/lambda)= 31.50877164766082

Normal mean= 31.50877164766082

Normal std. dev = 45.02624291039452

LNorm shape(exp(mean)) = 1.4619920135974123

Scale(std. dev)= 13.35898425167829



Parameters:

Exp mean(1/lambda)= 51.377222753605444

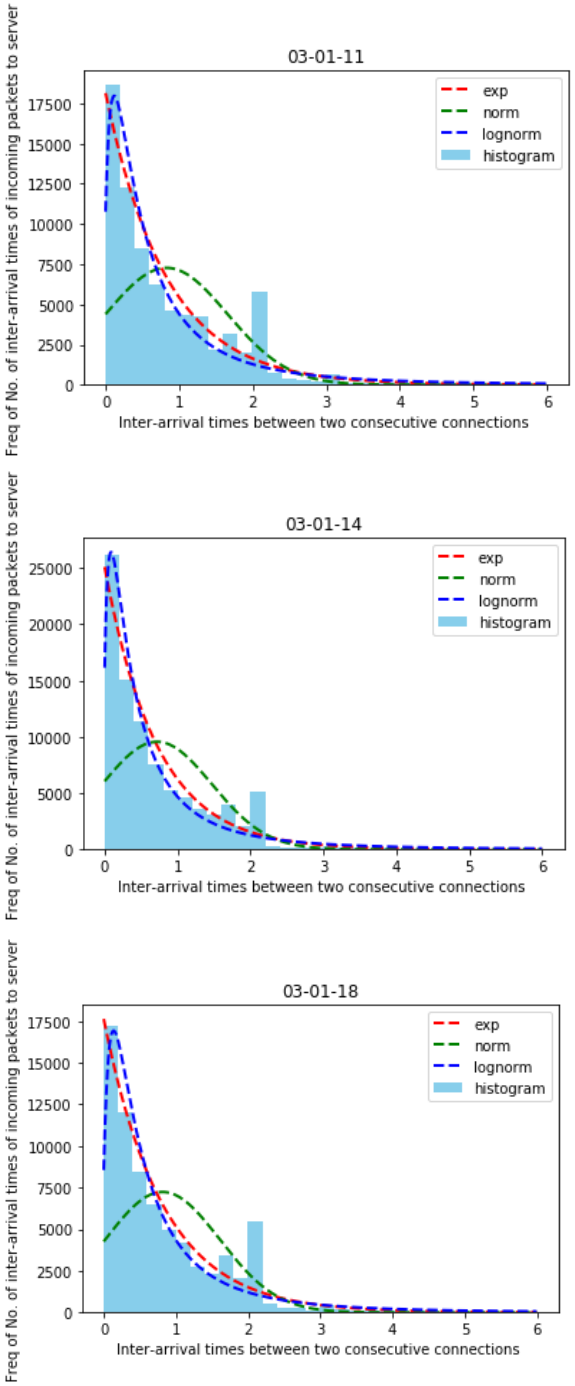
Normal mean= 51.37722375360578

Normal std. dev = 77.83938176925382

LNorm shape(exp(mean)) = 1.901307533279424

Scale(std. dev)= 15.079280112267412

PDF fitting for question 7th



Parameters:

Exp mean(1/lambda)= 0.8295734304126466

Normal mean= 0.8295734304126466

Normal std. dev = 0.825869195765223

LNorm shape(exp(mean)) = 1.063171908437548

Scale(std. dev)= 0.5531504746540603

Parameters:

Exp mean(1/lambda)= 0.7181498654806572

Normal mean= 0.7181498654806572

Normal std. dev = 0.7516120717906161

LNorm shape(exp(mean)) = 1.109861874473586

Scale(std. dev)= 0.4540307618172925

Parameters:

Exp mean(1/lambda)= 0.8145871388019643

Normal mean= 0.8145871388019643

Normal std. dev = 0.7915521195626273

LNorm shape(exp(mean)) = 1.0230117218501713

Scale(std. dev)= 0.5584904548995082

Ques 11. Use the value of lambda as estimated by the exponential fit for the packet interarrival times for the incoming packets.

To calculate the value of mu, assume that the link can transmit data at 128Kbps. Then use the mean value of packet length as found in question 8 above to calculate mu in terms of packets that can be serviced per second.

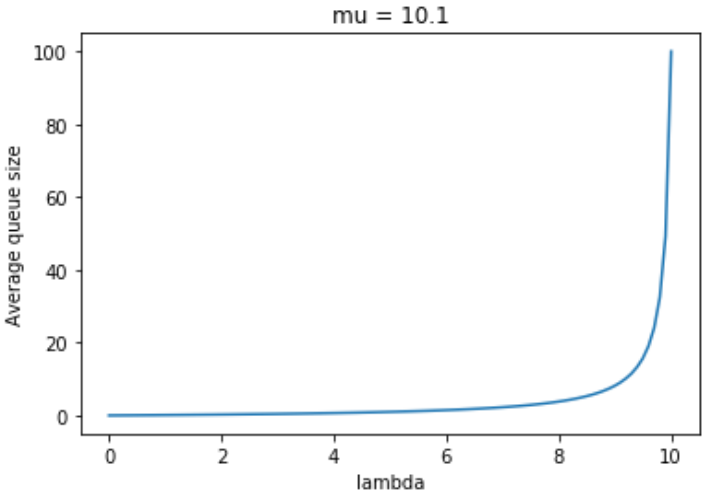
What is the utilization factor rho? For this utilization factor, what is the average queue size and the average waiting time for a packet in the queue?

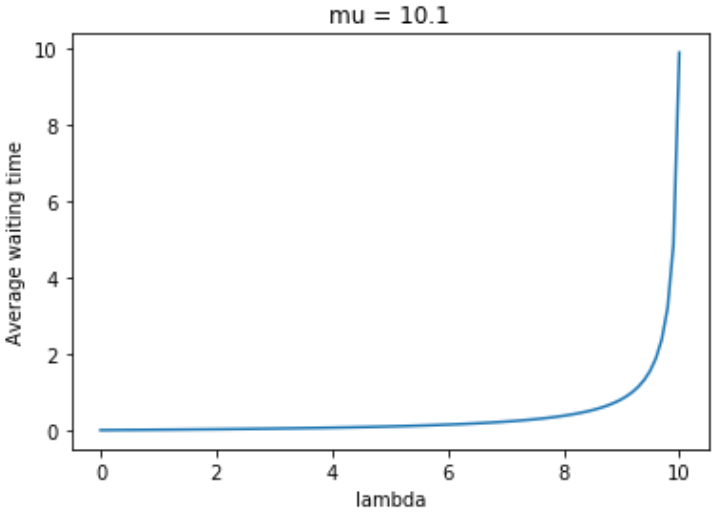
Now graph the queue size and the average waiting time (on separate graphs) for different values of lambda, starting from 0 to getting close to mu. What does this tell you about how the queue sizes and delays change with an increasing load in the network?

Ans. Increasing load will decrease mean inter arrival time and which will increase lambda. Henceforth avg queue time and waiting time will increase, that can be inferred from the graphs given below.

```
#11. queue setup
lambda_inv = [51.8423, 31.5087, 51.3772] #these are the mean interarrival time or inverse of lambdas found in previous parts
lambdas = [1/x for x in lambda_inv]
mean_packet_size = [57.2529, 56.6091, 58.0865] #these are the mean packet sizes found in 8th question
transmit_data_speed = 128*1000/8
mu = [transmit_data_speed/x for x in mean_packet_size]
rho = [(l/m) for l,m in zip(lambdas, mu)]
avg_queue_size = [x/(1-x) for x in rho]
avg_waiting_time = [(1/(m-l))-(1/m) for l,m in zip(lambdas,mu)]
print("lambda           = "+str(lambdas))
print("mu               = "+str(mu))
print("rho              = "+str(rho))
print("avg_queue_size   = "+str(avg_queue_size))
print("avg_waiting_time = "+str(avg_waiting_time))
```

```
lambda           = [0.019289267644375347, 0.03173726621536274, 0.01946388670460827]
mu               = [279.46182638783364, 282.64007023605745, 275.451266645434]
rho              = [6.902290696979108e-05, 0.00011228862980700568, 7.066181594170177e-05]
avg_queue_size   = [6.902767146033662e-05, 0.00011230123995936635, 7.066680938677985e-05]
avg_waiting_time = [2.470021482092223e-07, 3.973295076863982e-07, 2.565492264652164e-07]
```


mu = 10.1
Average queue size
lambda
0
20
40
60
80
100
0
2
4
6
8
10


mu = 10.1
Average waiting time
lambda
0
2
4
6
8
10
0
2
4
6
8
10