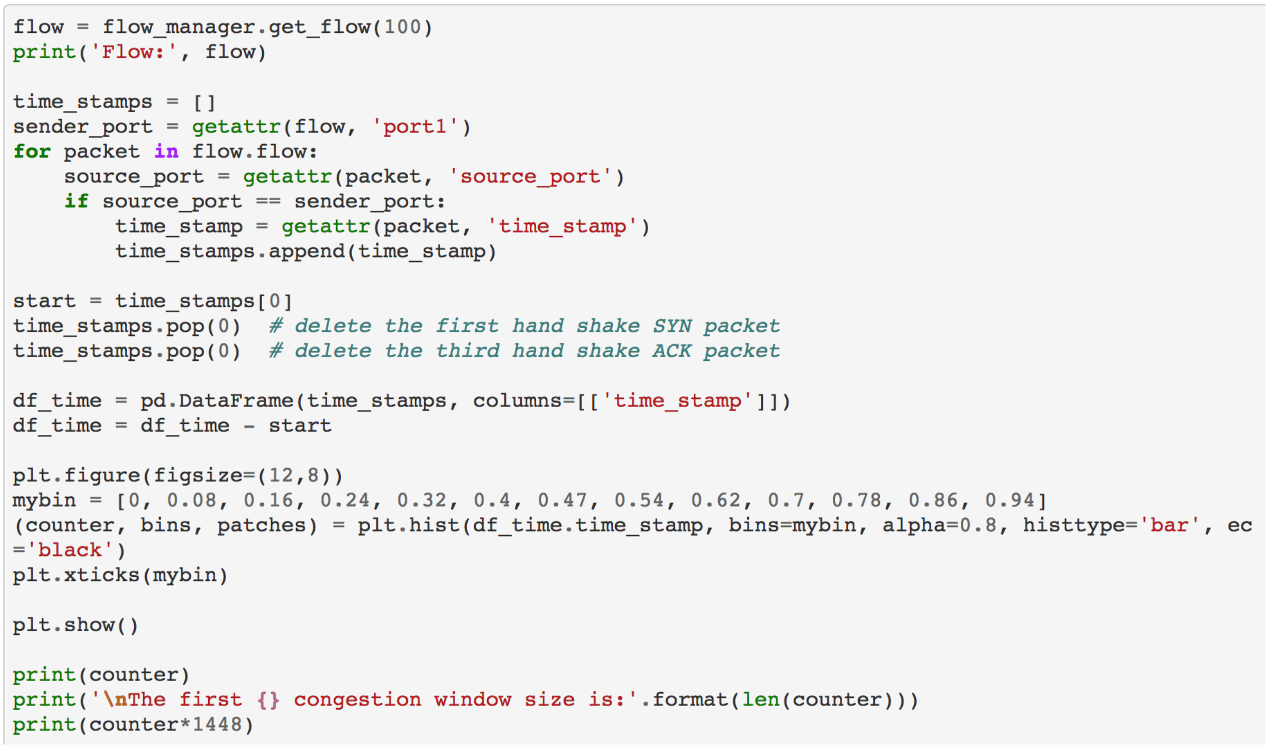
Table of Contents

1 Part B (1) 1
1.1 Estimate Method 1
1.2 First Flow 2
1.3 Second Flow 3
1.4 Third Flow 4
2 Part B (2) 5
2.1 Estimate Method 5
2.2 Answer 5

1 Part B (1)

1.1 Estimate Method

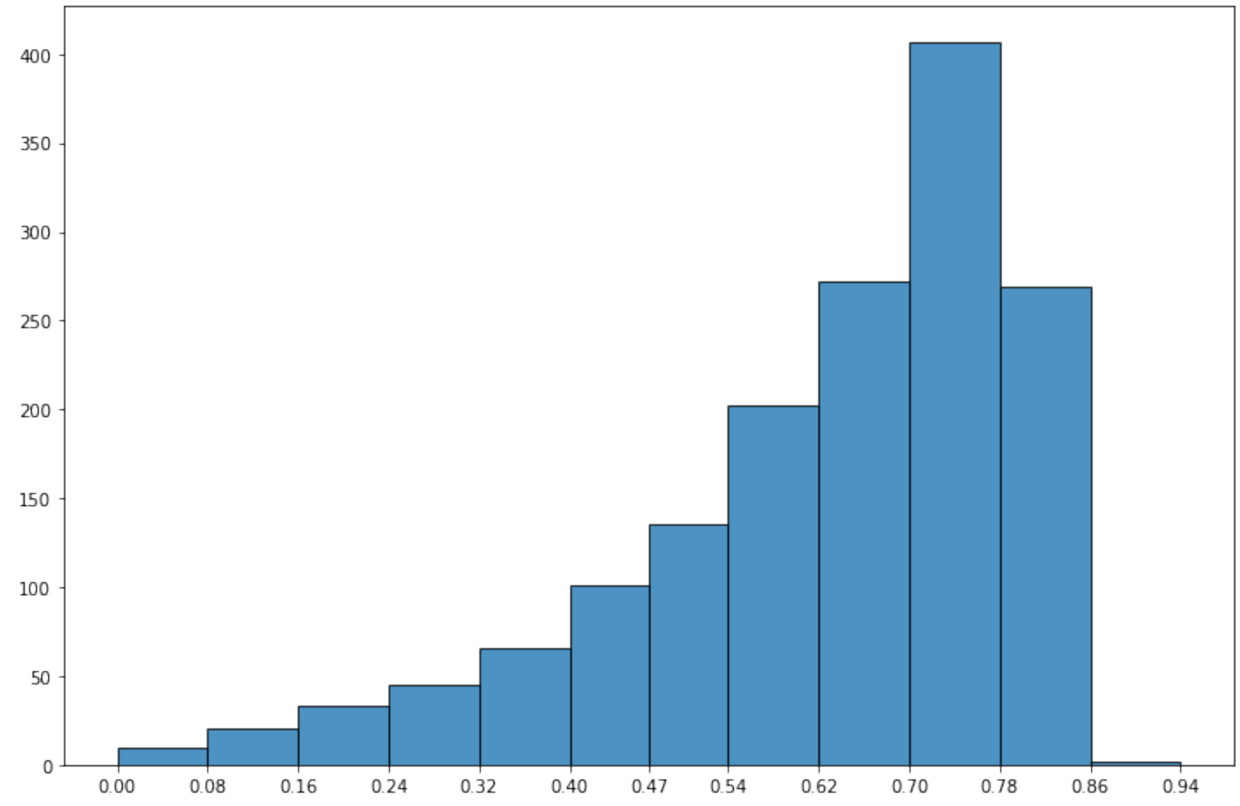
The sender sends some packets every round of RTT. In part A, we already get the RTT is a little bigger than 0.07 second. So, we put all the timestamps of **sender's** packet into **Pandas.Dataframe**, and **set the bins according to RTT**, and Matplotlib will do everything left. Code is in Jupyter Notebook. Code for the first flow is below.



```
flow = flow_manager.get_flow(100)
print('Flow:', flow)

time_stamps = []
sender_port = getattr(flow, 'port1')
for packet in flow.flow:
    source_port = getattr(packet, 'source_port')
    if source_port == sender_port:
        time_stamp = getattr(packet, 'time_stamp')
        time_stamps.append(time_stamp)

start = time_stamps[0]
time_stamps.pop(0)  # delete the first hand shake SYN packet
time_stamps.pop(0)  # delete the third hand shake ACK packet

df_time = pd.DataFrame(time_stamps, columns=[['time_stamp']])
df_time = df_time - start

plt.figure(figsize=(12,8))
mybin = [0, 0.08, 0.16, 0.24, 0.32, 0.4, 0.47, 0.54, 0.62, 0.7, 0.78, 0.86, 0.94]
(counter, bins, patches) = plt.hist(df_time.time_stamp, bins=mybin, alpha=0.8, histtype='bar', ec
='black')
plt.xticks(mybin)

plt.show()

print(counter)
print('\nThe first {} congestion window size is:'.format(len(counter)))
print(counter*1448)
```

1.2 First Flow

```
Flow: ID=100  port1=43498  port2=80  # of packets=11106
```



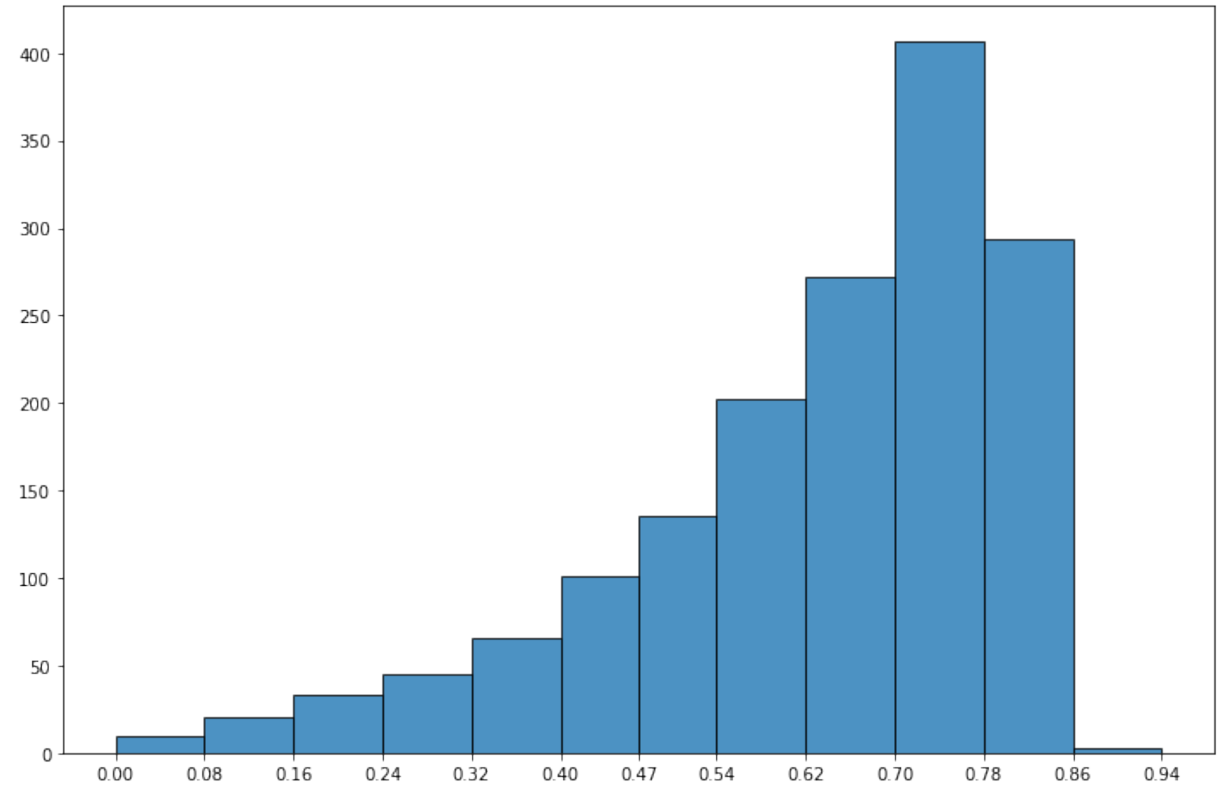
```
[  10.   20.   33.   45.   66.  101.  135.  202.  272.  407.  269.    2.]

The first 12 congestion window size is:
[  14480.   28960.   47784.   65160.   95568.  146248.  195480.  292496.
  393856.  589336.  389512.    2896.]
```

Only printing the first 10 congestion window size is NOT interesting, since they keep growing. What is interesting is that a triple duplicate ack occurred in the 10th RTT, so the congestion window decreased in the 11th RTT. And in the 11th RTT, a timeout unfortunately occurred, so the congestion window decreased to a very small number in the 12th RTT. Therefore, I printed first 12 congestion window size.

The initial congestion window size is **14480**. It seems that the first ten size grows **multiplicatively**, though **not by a factor of 2**. The first ten grow rate is [2, 1.65, 1.36, 1.47, 1.53, 1.34, 1.49, 1.35, 0.66]

1.3 Second Flow

```
Flow: ID=101  port1=43500  port2=80  # of packets=11834
```



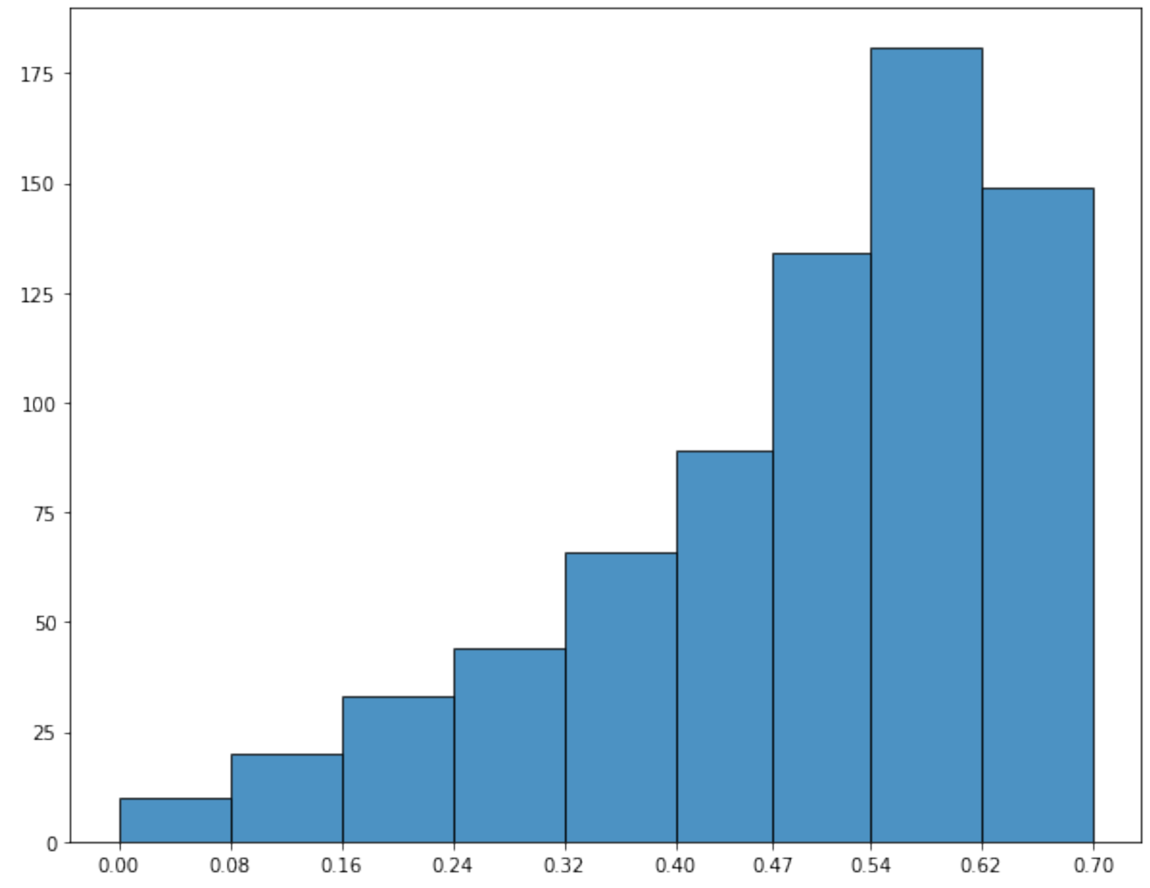
```
[  10.   20.   33.   45.   66.  101.  135.  202.  272.  407.  294.    3.]

The first 12 congestion window size is:
[  14480.   28960.   47784.   65160.   95568.  146248.  195480.  292496.
  393856.  589336.  425712.    4344.]
```

Only printing the first 10 congestion window size is NOT interesting, since they keep growing. What is interesting is that a triple duplicate ack occurred in the 10th RTT, so the congestion window decreased in the 11th RTT. And in the 11th RTT, a timeout unfortunately occurred, so the congestion window decreased to a very small number in the 12th RTT. Therefore, I printed first 12 congestion window size.

The initial congestion window size is **14480**. It seems that the first ten size grows **multiplicatively**, though **not by a factor of 2**. The first ten grow rate is [2, 1.65, 1.36, 1.47, 1.53, 1.34, 1.49, 1.35, 0.72]

1.4 Third Flow

```
Flow: ID=102  port1=43502  port2=80  # of packets=1185
```



```
[  10.   20.   33.   44.   66.   89.  134.  181.  149.]

The first 9 congestion window size is:
[  14480.   28960.   47784.   63712.   95568.  128872.  194032.  262088.
  215752.]
```

This flow didn't encounter a loss event.

The initial congestion window size is **14480**. It seems that the first eight size grows **multiplicatively**, though **not by a factor of 2**. The first eight grow rate is [2, 1.65, 1.33, 1.5, 1.35, 1.51, 1.35].

Note that in the **ninth RTT**, the sender finished sending its data, thus the flow ends without utilizing its full potential. This causes the ninth window is smaller than the eighth window. However, real congestion window size (full potential) of the **ninth window** is estimated around **230**.

2 Part B (2)

2.1 Estimate Method

First, we get the total number of retransmission, which is the total number of loss event $total\_loss$.
Then we compute the number of triple duplicate ack TDA.
Then the number of timeout is $timeout = total\_loss - TDA$

2.2 Answer

```
PART B(2)

***Flow 100***
# of triple duplicate ack = 2
# of timeout = 1

***Flow 101***
# of triple duplicate ack = 4
# of timeout = 90

***Flow 102***
# of triple duplicate ack = 0
# of timeout = 0
```