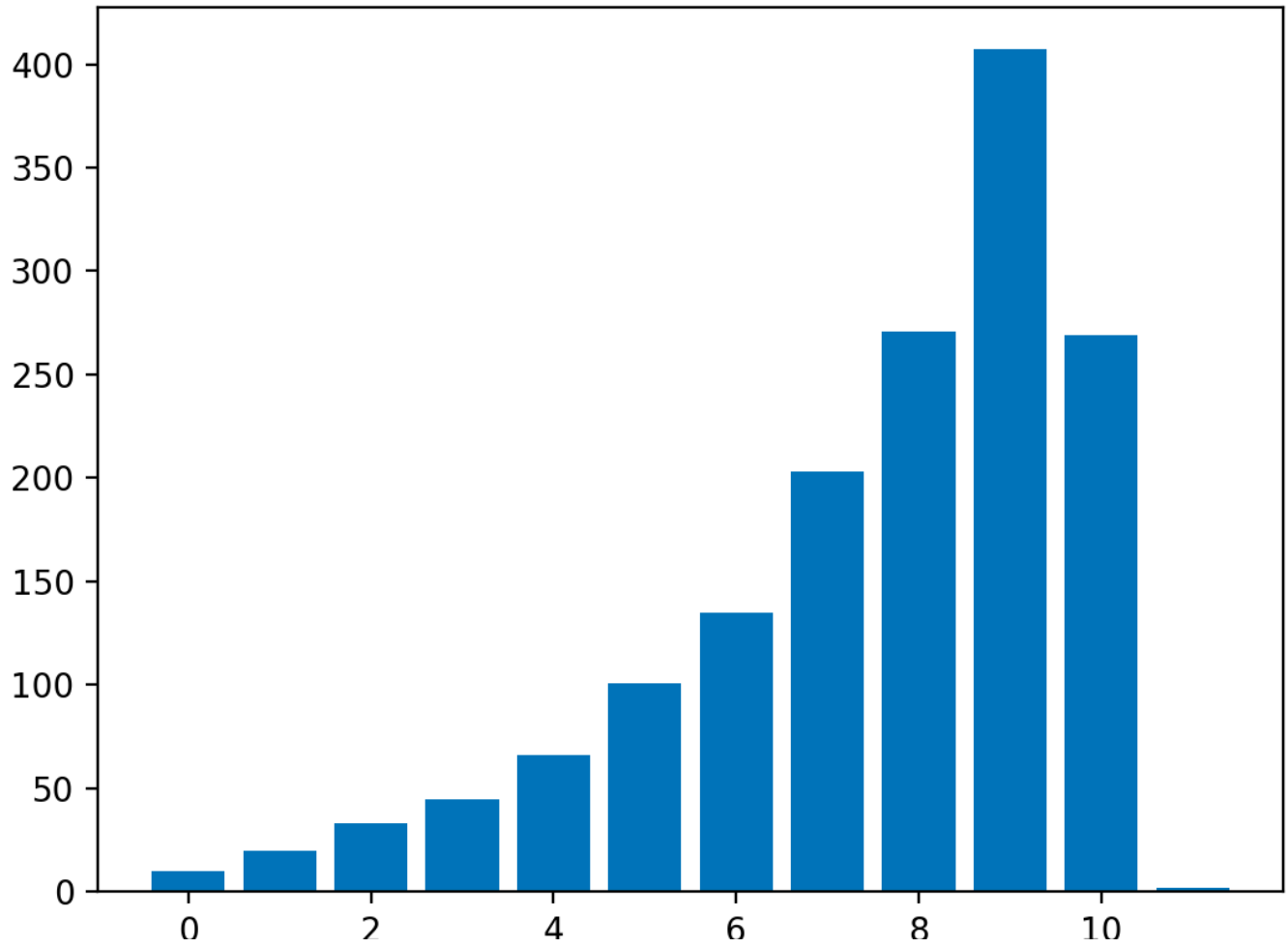
Answers:

Task B-1:

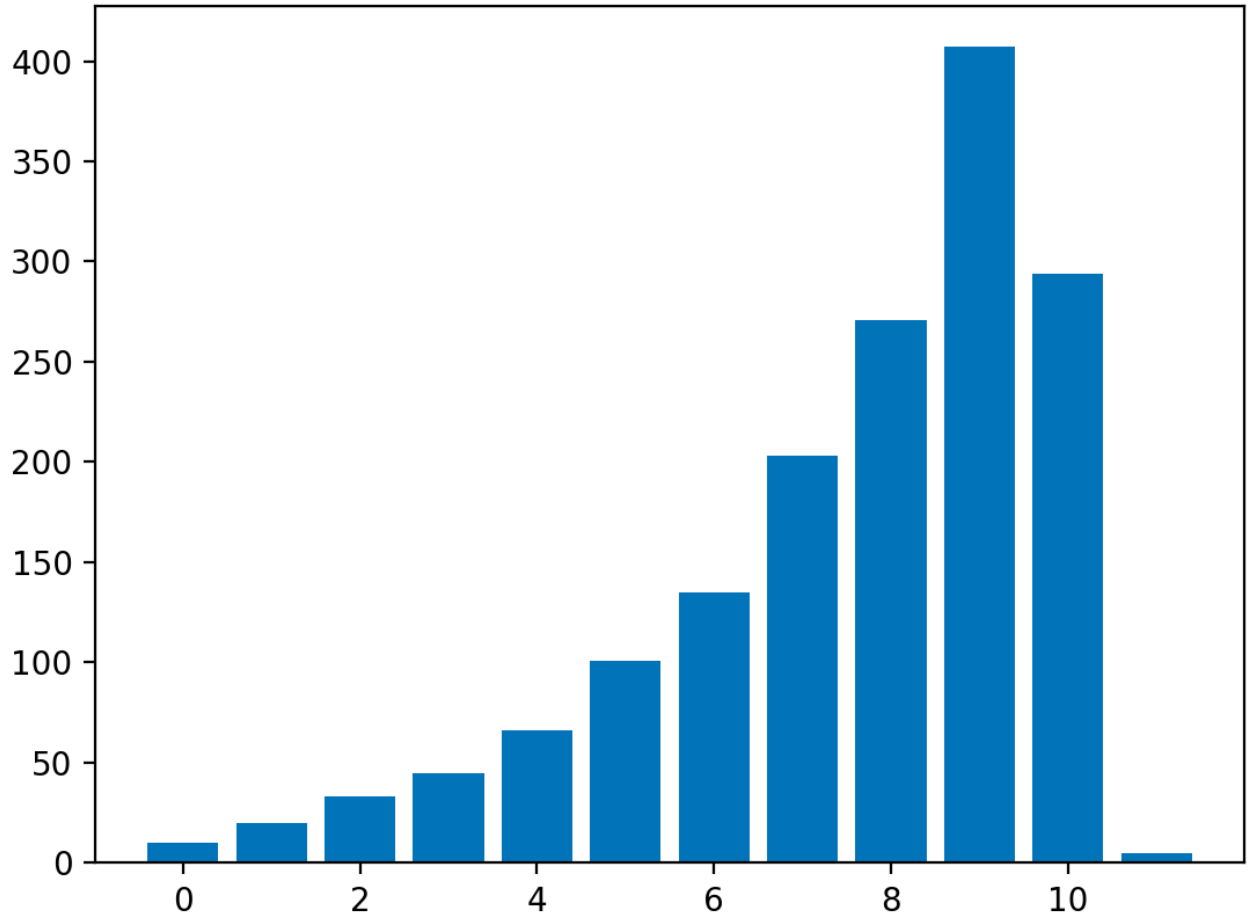
In part A, we have computed the RTT is a little bigger than 0.07 second, I set the RTT as 0.08 and divide the packets into each window list based on the time difference. I used the ways for part A to process the packets and flows for Drawing the graph. The detail code is in B1.py.

Flow 1 result:



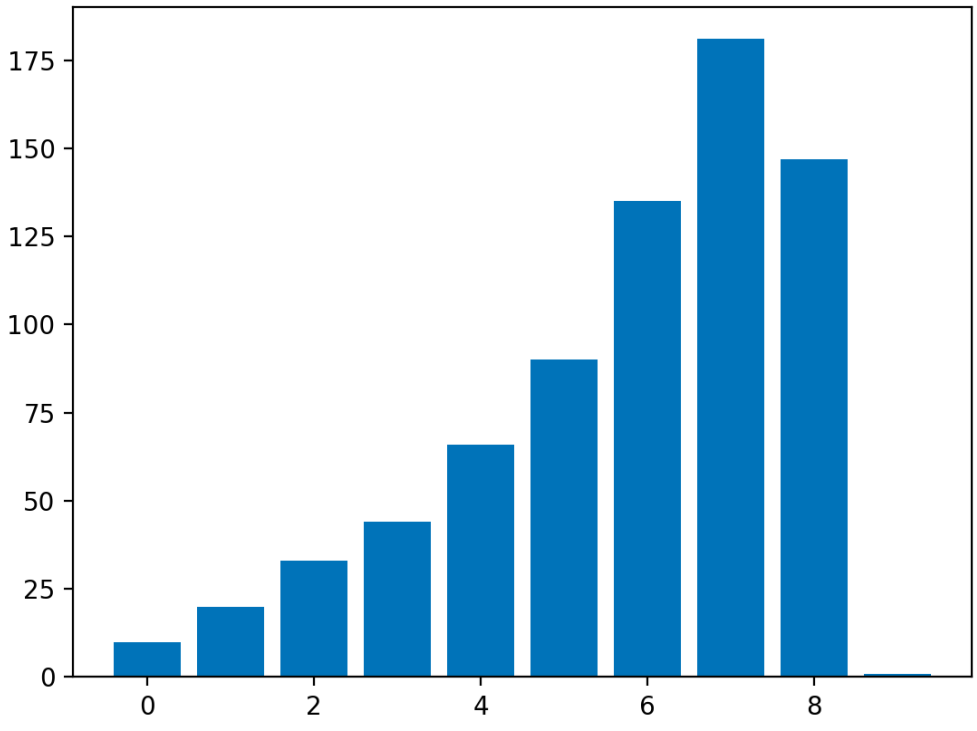
mss is 1460
[10, 20, 33, 45, 66, 101, 135, 203, 271, 407, 269, 2]
[14600, 29200, 48180, 65700, 96360, 147460, 197100, 296380, 395660, 594220, 392740, 2920]

Flow 2 result:




```
mss is 1460
[10, 20, 33, 45, 66, 101, 135, 203, 271, 407, 294, 5]
[14600, 29200, 48180, 65700, 96360, 147460, 197100, 296380, 395660, 594220,
429240, 7300]

Flow 3 result:
```



mss is 1460
[10, 20, 33, 44, 66, 90, 135, 181, 147, 1]
[14600, 29200, 48180, 64240, 96360, 131400, 197100, 264260, 214620, 1460]

For each flow: the first list is the number of packets, and the second list is the result of congestion window size: congestion window size=number of packets*mss. I read the mss from the packet, it is 1460, but in practice, it seems 1448.

For the first two graphs, the first ten size grow multiplicatively to reach the almost same window size, which should the bandwidth. For the third flow, since it doesn't have much data, it didn't make use of bandwidth fully.

B2:

There are two types of packet loss: triple duplicate ack and time out. Since time out is hard to identify. I use this way to finish it: identify the total loss and triple duplicate ack loss; then time out loss=total loss-triple duplicate ack loss.

This task is finished by the part A program:

Number of triple duplicate ack = 2
Number of timeout = 1

Number of triple duplicate ack = 3
Number of timeout = 91

Number of triple duplicate ack = 0
Number of timeout = 0