Muhammad Ahsan Rahim
ERP-ID: 14817
Systems Modelling and Simulation

Telephone Lines Simulation

Goal: To find the minimum number of lines such that the percentage of dropped calls is less than 0.00 with the given simulation conditions

Simulation Conditions:

1) Calls from A to B with a mean inter-call arrival time of 10 seconds.

   To do that, I generated a normal distribution with mean = 10 and standard deviation of 5.

```
iatAB = np.random.normal(10, 5, 100)
iatAB= np.abs(iatAB.astype(int))
```

2) Calls from B to A with a mean inter-call arrival time of 12 seconds.

   To do that, I generated a normal distribution with mean = 12 and standard deviation of 5.

```
iatBA = np.random.normal(12, 5, 100)
iatBA= np.abs(iatBA.astype(int))
```

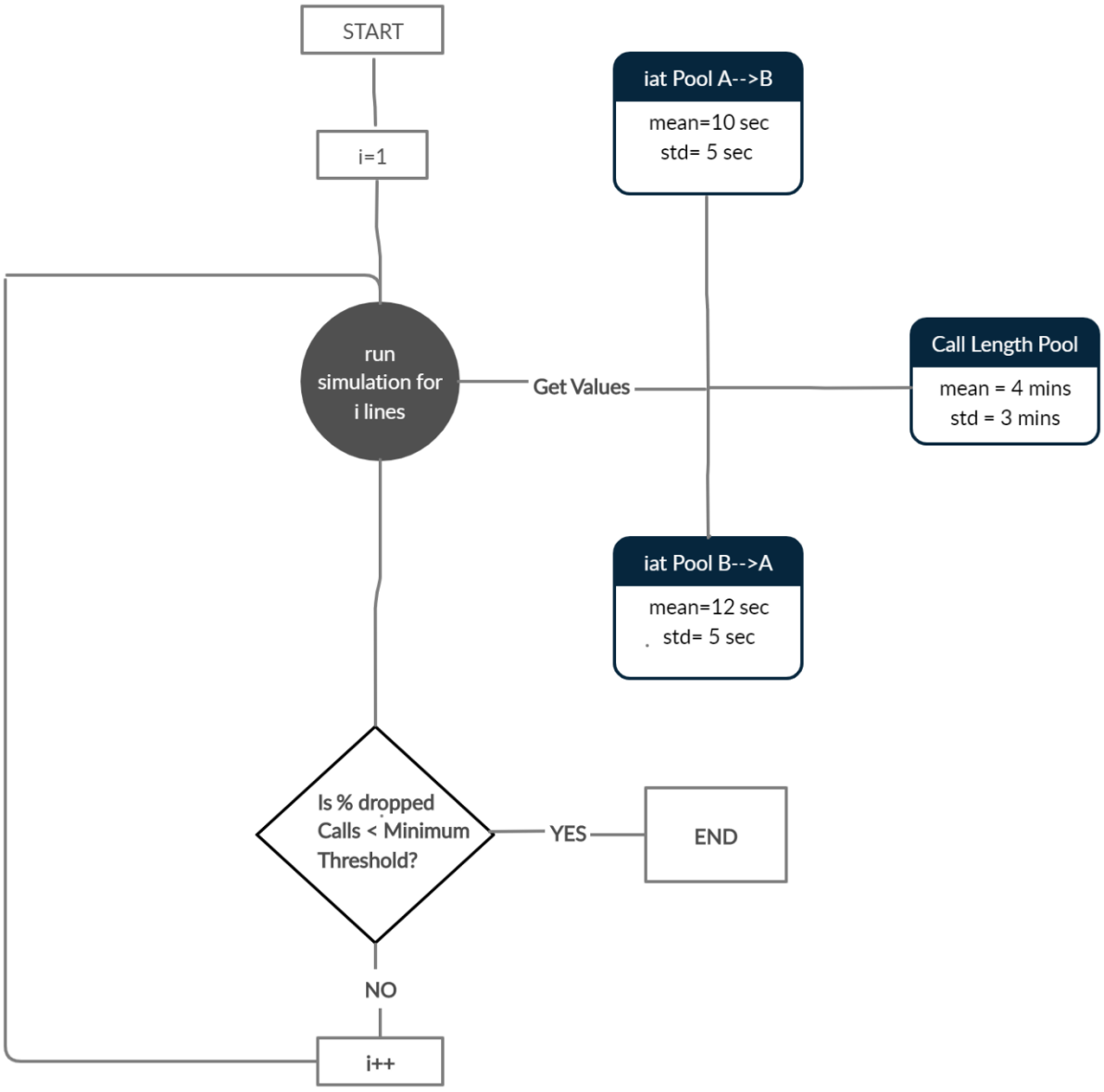
3) Length of conversation with a mean of 4 mins

   To do that, I generated a normal distribution with mean = 4 mins or 240 seconds and standard deviation of 3 mins or 180 seconds

```
callLength = np.random.normal(4*60, 3*60 , 100)
callLength=np.abs(callLength.astype(int))
```

Muhammad Ahsan Rahim
ERP-ID: 14817
Systems Modelling and Simulation

Dataflow:



Muhammad Ahsan Rahim
ERP-ID: 14817
Systems Modelling and Simulation

Base line code for 1 line:

```
connected=0
dropped=0
nextAB=choice(iatAB)
nextBA=choice(iatBA)
simTime=min(nextAB,nextBA)

while(simTime<twelveHours):

   freeLines= [i for i in lines if i<= simTime]

  if(nextAB==nextBA):
       simTime+=nextAB
       nextAB=choice(iatAB)
       nextBA=choice(iatBA)

      for m in range(2):

        if( len(freeLines)!=0):

          lines[lines.index(freeLines.pop(0))]=simTime+choice(callLength)
          connected+=1
        else:
             dropped+=1


   else:
       if( len(freeLines)!=0):
         lines[lines.index(freeLines.pop(0))]=simTime+choice(callLength)
         connected+=1
       else:
           dropped+=1


       if(nextAB>nextBA):
           simTime+=nextBA
           nextAB=nextAB-nextBA
           nextBA=choice(iatBA)

       else:
           simTime+=nextAB
           nextBA=nextBA-nextAB
           nextAB=choice(iatAB)
```

```
print("Calls connected = " , connected)
print("Calls dropped = " , dropped)
print("Total Calls = " , connected+dropped)
print("Percentage of Dropped Calls = " + str(dropPercentage))
```

Output:

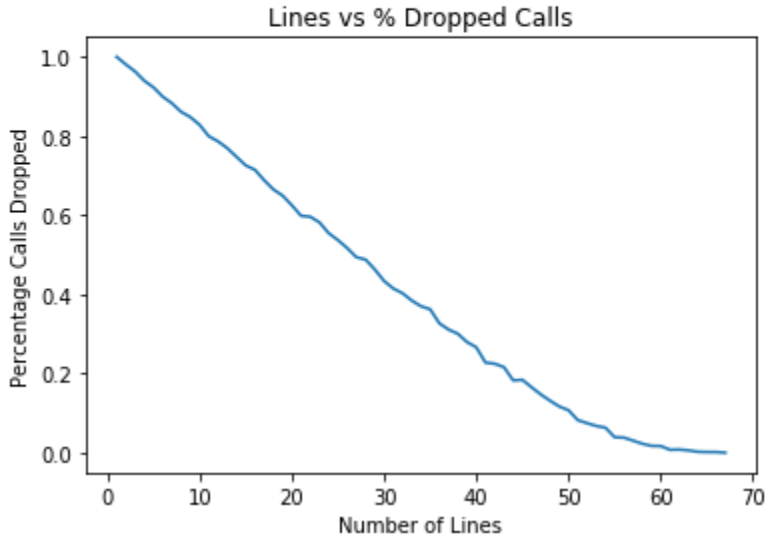
```
Calls connected =  162
Calls dropped =  8745
Total Calls =  8907
Percentage of Dropped Calls = 0.9818120579319636
```

As we would expect, on a single line system, most of the calls were dropped and less than 2% of the calls were connected.

Now we scale our simulation by increasing the number of lines and finding the out the minimum number of lines required to achieve our desired threshold that is 0% dropped calls while at the same time we plot a graph to visualize the relationship between number of lines and percentage dropped calls.

Simulation output on given conditions:

```
Lines = 66
Calls connected =  8810
Calls dropped =  0
Total Calls =  8810
Percentage of Dropped Calls = 0.0
```



As we can see from the graph, we achieve 0.0 percentage calls dropped with about 60 lines and achieve 0.00% of calls dropped with 66 lines.

NOTE: Simulation results might vary slightly each time we run the simulation as the inter call arrival times and call length times are random.

Muhammad Ahsan Rahim
ERP-ID: 14817
Systems Modelling and Simulation

Simulation on increased average call length:

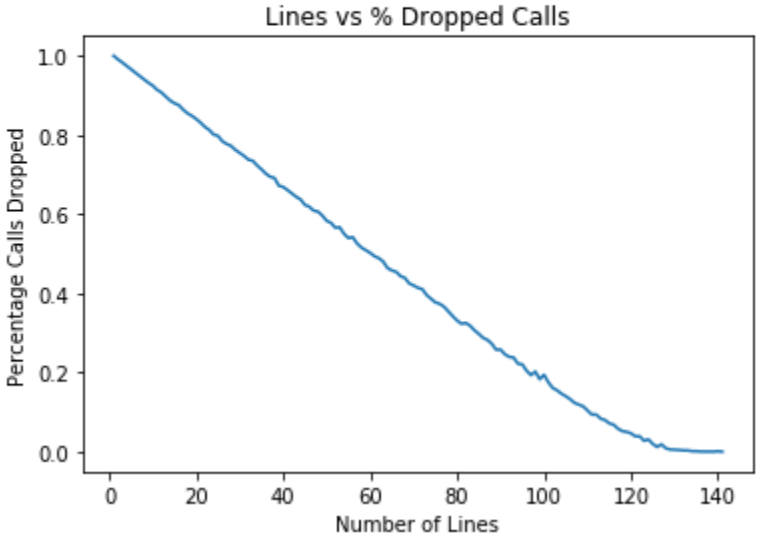
Let us now repeat our simulation by increasing the average call length to 10 minutes.

To do that, I generated a normal distribution with mean = 10 mins or 240 seconds and standard deviation of 3 mins or 180 seconds

```
callLength = np.random.normal(10*60, 3*60 , 100)
callLength=np.abs(callLength.astype(int))
```

My simulation gives the following output upon the increase in average call length.

```
Lines = 140
Calls connected =  8522
Calls dropped =  0
Total Calls =  8522
Percentage of Dropped Calls = 0.0
```



As expected, the number of lines required to achieve 0.00 percentage of dropped calls increased significantly, from around 65 lines previously to 140 with 10 mins of average call length.

And the number of lines required to achieve 0.0 percentage of dropped calls increased significantly, from around 60 lines previously to 127 lines with 10 mins of average call length.

NOTE: Simulation results might vary slightly each time we run the simulation as the inter call arrival times and call length times are random.

Muhammad Ahsan Rahim
ERP-ID: 14817
Systems Modelling and Simulation

APPENDIX:

Please find the simulation source code attached titled TelephoneSMS.py