CZ4015 Assignment 1 part 2 report

Seungju Lee (N1904291J)

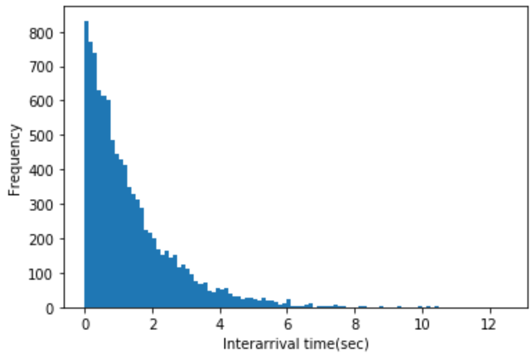
For this assignment, I made a simulation model in python using jupyter notebook. I used numpy, pandas, matplotlib.pyplot, scipy.stats module in my code.

1. Input Analysis

The measured data is provided in the excel file “PCS_TEST_DETERMINISTIC” so I used this data to find what distributions the inter-arrival time of calls, the locations where calls are generated(from now on I will call it base station), the call durations, and car speeds follow respectively and figure out the parameter values of these distributions.

1 - 1 Interarrival time

In the measured data, interarrival time is not directly given so I computed interarrival time by subtracting previous call’s initiation time from current call’s initiation time. The image below is the histogram of the interarrival time with square root of the number of data number of intervals.



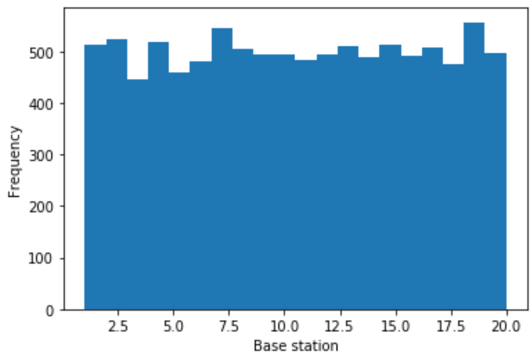
The histogram of interarrival time looks like the pdf of an exponential distribution. Therefore, I hypothesized the data has an exponential distribution and conducted a chi-square test with MLE (this case, mean of interarrival times, which was 1.3698). The result of the chi-square test was like below.

```
Test statistic: 111.64999999999999
Critical value with alpha=0.05: 122.10773460981942
The null hypothesis cannot be rejected. Use exponential distribution with current parameter!
```

In other words, it is logical to say that interarrival time follows an exponential distribution

1 - 2 Base stations

The image below is the histogram of base stations.



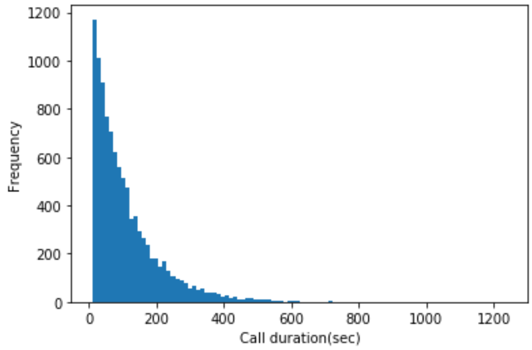
The histogram of base station looks like it is uniformly distributed so I hypothesized the data has a uniform distribution and conducted a chi-square test. The result was like below.

```
Test statistic: 25.656
Critical value with alpha=0.05: 30.14352720564616
The null hypothesis cannot be rejected. Use uniform distribution!
```

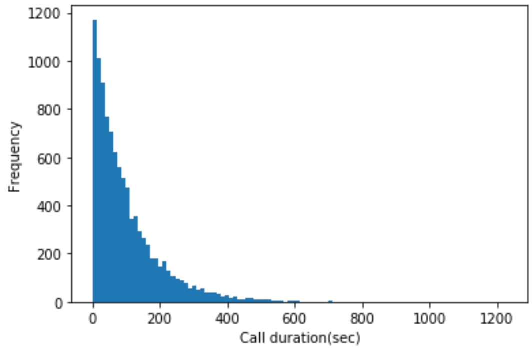
In other words, it is logical to say that base station is uniformly distributed from 1 to 20(integer values).

1 - 3 Call duration

The image below is the histogram of call duration.



Looking at the histogram, there is a slight gap, about the length of 10, between the minimum value of call duration and 0. Although the overall shape of the histogram looks like it follows exponential distribution, it is necessary to deal with this gap. The image below is the histogram of the data: duration – gap (minimum duration).



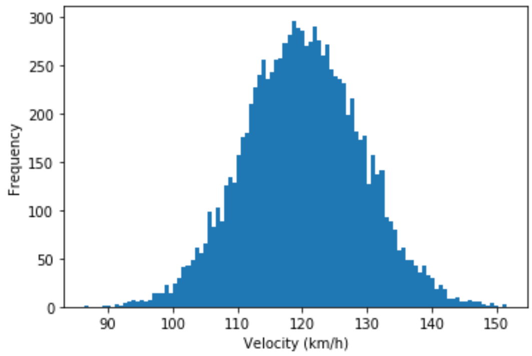
It would be much more logical to say that (duration – gap) follows an exponential distribution because there is no gap between 0 and the left of the first interval in the histogram. Therefore, I hypothesized that (duration – gap) follows exponential distribution with parameter (mean of this data) and the result of chi square test was like below.

```
Test statistic: 97.01999999999998
Critical value with alpha=0.05: 122.10773460981942
The null hypothesis cannot be rejected. Use exponential distribution with current parameter!
```

In other words, it is logical to say that (duration – gap) follows exponential distribution.

1 - 4 Velocity

The image below is the histogram of velocity.



The histogram of velocity looks like it follows the normal distribution, so I hypothesized that velocity of the car follows the normal distribution with estimated parameter (MLE) and conducted a normality test. I used scipy.stats.normaltest for this goodness-of-fit test. The result of the test was like below.

```
The null hypothesis cannot be rejected. Use normal distribution with current parameter!
```

In other words, it is logical to say that velocity follows normal distribution with estimated parameters.

2. Corrections and changes from the pseudocode

There were some points in my pseudocode that are needed to be corrected or modified. First, I did not generate random calls in my pseudocode. I only read the data from the excel file and generated random direction and position for each call. It is necessary to generate random calls not only using the deterministic calls. In my simulation model, I did not use the deterministic call events and only used randomly generated calls.

Also, I created all the calls in the main function at time 0. Only the first call initiation event should be created in main function and putted into the future event list. In my simulation model, I generated the first call in the main function (__init__ function in python) and generated certain number of random calls after time 0 with random interarrival times.

Due to the change in the main function, overall logic of the simulation is modified. In my simulation model, initialization step would be done first including generation of the first call initialization event and putting it into the future event list. Then, with this first call, I generated number of calls, of course their event type would be "Call Initialization" and inserted them into the future event list. After that, the event handling routine will begin from the very first event in the future event list and move on to the next scheduled event. If the new event is generated, it will be inserted into the future event list which is sorted in time order. The simulation will continue until the system meets the ending condition: future event list is empty or certain number of call initialization is handled.

In my call initiation and call handover handling routine in pseudocode, I made the blocked calls and dropped calls to remain in the system after they are blocked or dropped. This should not be handled like this so in my actual simulation model, if the call is blocked or dropped, they leave the event handling function right away.

In addition, I calculated the stats in a wrong way and ignored warm up period in the pseudocode. I defined blocked rate as number of blocked calls divided by number of total calls and dropped rate as number of dropped calls divided by number of handovers. However, in the problem description, blocked rate and dropped rate is not as I defined so I modified in my simulation model. Other than that, I calculated the number of blocks, number of drops, number of total calls after the warm-up period in my actual model to make the stat more reliable.

Moreover, I differently defined the random variate "position" in my actual model. Before, I set the range of position value from 0 to 40. Which means that the position is demonstrated like a coordinate in my pseudocode. However, in my actual model, I made the position's range from 0 to 2. This shows the actual position of the call's initialization in the cell not in the overall highway. Due to this change, I changed the logic of calculating the call's remaining distance to the end of the station and time spent in the cell for the call's initialization. In terms of remaining distance, I made a mistake in calculating it in call handover handling routine. Before, I calculated remaining distance identical to that process in call initialization. However, for the handover remaining distance should always be 2.

In the event handling routine, I changed the condition check and parameter updating step when deciding the event type of the next event. In my actual model, I firstly check whether the call ends in the current station. If it ends in the current station, I scheduled call termination event after the call's remaining duration in the current station. If the call does not end in the current station, then I check whether the next station of the call is out of the highway stations' index. If so, call should be terminated after the car's time spent in the cell. If the call does not end in the current station and the next station is still in the highway, then I scheduled the handover event after the car's time spent in cell with updated duration (subtract car's time spent in the cell from the original value). This condition check and updating step is much more logical than that from my pseudocode.

Finally, there was an error in the expression of scheduling future event. In my pseudocode, I said that if certain conditions are satisfied, "call" certain event handling routine. This expression is vague. It should be "schedule" the future event. In my simulation model, the system checks the condition of the event and decide the event type of its next event. Then, it inserts this event with new event type to the future event list, which is literally scheduling the next event in the future event list.

3. Discussion of warm up period

As I mentioned in part 2 of this report, warm up period is necessary to get the reliable statistic. I set the warm-up period as calculation of stats after the simulation clock pass 1000(sec). The reason why I chose like this is that interarrival time follows exponential distribution with mean of about 1.3. Roughly, I can estimate that until the simulation clock becomes 1000, about 800 calls (calculation: 1000 / 1.3) are generated. Also, there are 20 stations with 10 channels each, so there are 200 available channels in the highway right now. Therefore, 800 calls are much larger than the total number of channels in the highway. Some termination and handover events would have been done before 1000 seconds so by comparing the size of channel and estimated generated calls, I thought that after 1000 seconds the system might be stabilized and appropriate to get some statistics. Therefore, I set the warm-up period to 1000 seconds.

Other than that, I considered about the cooling period in the simulation. If I just made the simulation model ends when the future event list is empty, in the very last stage of the simulation the system is in the different situation from the stabilized version. No more call initializations will happen and there would be only termination and handover at the very last. This would definitely affect the reliability of the stats. So, I made the ending condition of the simulation model. The simulation model will end when the future event list is empty, or all generated calls are initialized.

4. Simulation results

For the simulation, it is necessary to decide the number of iterations, the number of simulations per iteration, FCA scheme, total number of calls to be generated, and warm up period. So, I made my simulation model to have these variables as input. This time, I set the number of iterations to 10, number of simulations per iteration to 10, total number of calls to be generated to 10000, warm up period to 1000. The purpose of the simulation was to check the effect of channel reservation for handover, so I did 4 simulations to check the effect of channel reservation.

The images below are the result of simulation with different schemes. I made it able to check the mean, standard deviation, and 95% confidence interval. The average rate of blocks and average rate of drops follow normal distribution respectively.

4 - 1 No reservation

```
simulate(number_of_iteration = 10, number_of_simulation = 10, scheme = FCA_scheme.NO_RESERVATION,
         total_calls = 10000, warm_up_period = 1000)
```

```
Iteration 1 complete. Blocked rate: 0.366414%, Dropped rate: 0.605678%
Iteration 2 complete. Blocked rate: 0.353808%, Dropped rate: 0.535044%
Iteration 3 complete. Blocked rate: 0.376815%, Dropped rate: 0.543145%
Iteration 4 complete. Blocked rate: 0.341220%, Dropped rate: 0.569930%
Iteration 5 complete. Blocked rate: 0.355377%, Dropped rate: 0.546518%
Iteration 6 complete. Blocked rate: 0.346549%, Dropped rate: 0.585073%
Iteration 7 complete. Blocked rate: 0.311902%, Dropped rate: 0.598842%
Iteration 8 complete. Blocked rate: 0.361447%, Dropped rate: 0.591229%
Iteration 9 complete. Blocked rate: 0.339992%, Dropped rate: 0.611101%
Iteration 10 complete. Blocked rate: 0.429272%, Dropped rate: 0.589976%
----------------------------------------------------------------------
Total number of simulation: 100
Requirement fulfilled: 100
Failed in fulfilling requirement for block: 0
Failed in fulfilling requirement for drop: 0
----------------------Statistics----------------------
Block
Mean : 0.358%
Std : 0.031%
95% confidence interval : (0.3352397150583172, 0.38131890387892176)
Drop
Mean : 0.578%
Std : 0.027%
95% confidence interval : (0.5569575784173084, 0.5983497101930386)
```

4 - 2 One reservation

```
simulate(number_of_iteration = 10, number_of_simulation = 10, scheme = FCA_scheme.ONE_RESERVATION,
         total_calls = 10000, warm_up_period = 1000)
```

```
Iteration 1 complete. Blocked rate: 1.073783%, Dropped rate: 0.349753%
Iteration 2 complete. Blocked rate: 1.087229%, Dropped rate: 0.318126%
Iteration 3 complete. Blocked rate: 1.200301%, Dropped rate: 0.360057%
Iteration 4 complete. Blocked rate: 1.085256%, Dropped rate: 0.356772%
Iteration 5 complete. Blocked rate: 1.098618%, Dropped rate: 0.336839%
Iteration 6 complete. Blocked rate: 1.119039%, Dropped rate: 0.361514%
Iteration 7 complete. Blocked rate: 1.091428%, Dropped rate: 0.334516%
Iteration 8 complete. Blocked rate: 1.131188%, Dropped rate: 0.362665%
Iteration 9 complete. Blocked rate: 1.069893%, Dropped rate: 0.341981%
Iteration 10 complete. Blocked rate: 1.142855%, Dropped rate: 0.359556%
----------------------------------------------------------------------
Total number of simulation: 100
Requirement fulfilled: 100
Failed in fulfilling requirement for block: 0
Failed in fulfilling requirement for drop: 0
----------------------Statistics---------------------
Block
Mean : 1.110%
Std : 0.040%
95% confidence interval : (1.0799123895732783, 1.1400054743113266)
Drop
Mean : 0.348%
Std : 0.015%
95% confidence interval : (0.3369613732275319, 0.359394390464891)
```

4 - 3 Two reservation

```
simulate(number_of_iteration = 10, number_of_simulation = 10, scheme = FCA_scheme.TWO_RESERVATION,
         total_calls = 10000, warm_up_period = 1000)
```

```
Iteration 1 complete. Blocked rate: 2.724669%, Dropped rate: 0.215653%
Iteration 2 complete. Blocked rate: 2.752310%, Dropped rate: 0.216845%
Iteration 3 complete. Blocked rate: 2.629506%, Dropped rate: 0.176976%
Iteration 4 complete. Blocked rate: 2.528904%, Dropped rate: 0.177898%
Iteration 5 complete. Blocked rate: 2.739787%, Dropped rate: 0.185873%
Iteration 6 complete. Blocked rate: 2.577617%, Dropped rate: 0.153325%
Iteration 7 complete. Blocked rate: 2.648416%, Dropped rate: 0.175817%
Iteration 8 complete. Blocked rate: 2.814946%, Dropped rate: 0.162742%
Iteration 9 complete. Blocked rate: 2.668993%, Dropped rate: 0.198452%
Iteration 10 complete. Blocked rate: 2.808219%, Dropped rate: 0.169527%
----------------------------------------------------------------------
Total number of simulation: 100
Requirement fulfilled: 0
Failed in fulfilling requirement for block: 100
Failed in fulfilling requirement for drop: 0
----------------------Statistics---------------------
Block
Mean : 2.689%
Std : 0.095%
95% confidence interval : (2.617554990167624, 2.761118580591351)
Drop
Mean : 0.183%
Std : 0.021%
95% confidence interval : (0.16732646164998552, 0.19929502508293614)
```

4 - 4 Three reservation

```
simulate(number_of_iteration = 10, number_of_simulation = 10, scheme = FCA_scheme.THREE_RESERVATION,
         total_calls = 10000, warm_up_period = 1000)
```

```
Iteration 1 complete. Blocked rate: 5.706824%, Dropped rate: 0.085251%
Iteration 2 complete. Blocked rate: 5.622155%, Dropped rate: 0.096144%
Iteration 3 complete. Blocked rate: 5.769753%, Dropped rate: 0.080847%
Iteration 4 complete. Blocked rate: 5.528281%, Dropped rate: 0.089364%
Iteration 5 complete. Blocked rate: 5.409834%, Dropped rate: 0.077748%
Iteration 6 complete. Blocked rate: 5.717411%, Dropped rate: 0.086189%
Iteration 7 complete. Blocked rate: 5.711233%, Dropped rate: 0.068015%
Iteration 8 complete. Blocked rate: 5.506865%, Dropped rate: 0.066850%
Iteration 9 complete. Blocked rate: 5.410857%, Dropped rate: 0.069043%
Iteration 10 complete. Blocked rate: 5.470574%, Dropped rate: 0.090591%
----------------------------------------------------------------------
Total number of simulation: 100
Requirement fulfilled: 0
Failed in fulfilling requirement for block: 100
Failed in fulfilling requirement for drop: 0
----------------------Statistics---------------------
Block
Mean : 5.585%
Std : 0.136%
95% confidence interval : (5.482586968542703, 5.688170174085371)
Drop
Mean : 0.081%
Std : 0.010%
95% confidence interval : (0.07322177721828345, 0.08878673144584819)
```

The result shows that when there is no reservation and one reservation, throughout the whole simulation requirement was always fulfilled. Also, the right-hand side of the confidence interval is still below the minimum requirement for the blocked and dropped rate. This implies that the current system meets the quality of service requirements quite well. To compare between these two schemes, when there is one reservation, blocked rate seemed to increase more than 1%. Under the current quality of service requirement, it is okay but comparing to the result with no reservation, reserving a channel for handover is not that preferable.

When there are more than one channels reserved, dropped rate showed significant decrease but blocked rate showed huge increase in its mean value. Throughout the whole simulation, when there are multiple channels reserved, none of the simulation attempt fulfilled the quality of service requirements. This definitely implies that remaining the current scheme would be the best.

In conclusion, both current FCA schemes meet the quality of service requirements very well. However, the best option is not to have a reserved channel because when a channel is reserved although the QoS requirement is fulfilled but there is a significant increase in the blocked rate. Moreover, making more reserved channels would be a bad decision because the simulation result showed that blocked rate almost always surpass 2%. Thus, no channels should be reserved for handover service for best service.