CS 342: Computer Networks Laboratory
Assignment 04

Name	**Roll No.**
Aasneh Prasad	210101001
Pranav Jain	210101078
Sanjana Siri Kolisetty	210101093

The assignment was to create a simulation that replicates and improves security screening processes in a busy airport environment. As in real-life airports, passengers arrive at irregular intervals and undergo varying durations of security checks. This simulation scenario mirrors the actual challenges of queuing in airport scenarios, emphasizing the need to optimize security lines for heightened passenger satisfaction and overall airport efficiency.

Considering the analogy to the queueing theory, let us consider m to be the number of servers or security scanners in the airport and k be the size of the buffer.

The user defined parameters are λ and μ. λ is the arrival rate of passengers and μ is the service rate of passengers by the security scanners.

Introduction to Queueing Theory

Queueing theory is a valuable tool for analysing and optimizing the performance of systems with waiting lines in various fields. This report provides an overview of queueing theory, its key concepts, and the relevant formulas for both single-queue and multi-queue systems.

Key Concepts of Queueing Theory

1. Queues

A queue, or waiting line, represents a collection of entities waiting for service.

2. Arrival Process

The arrival process characterizes how entities enter the queue, following patterns like Poisson or deterministic arrivals.

3. Service Process

The service process defines how entities are served once they enter the queue, considering various service disciplines and rates.

4. Queueing Models

The M/M/1 (Markovian single-server) and M/M/c (Markovian multi-server) models are commonly used to analyze queues with single and multiple servers, respectively.

5. Performance Metrics, Optimization and Impacts

Key performance metrics include utilization, average queue length, and average waiting time in the queue. These metrics are essential for assessing system efficiency and customer satisfaction.

In this simulation, we have taken the inter-arrival times to have exponential distribution. We have used the following function to generate values such that they are in exponential distribution.

```
// Function to generate values to form an exponential distribution
double generateExponentialTime(double lambda)
{
    random_device rd;
    mt19937 gen(rd());
    exponential_distribution <double> exponential(lambda);
    return exponential(gen);
}
```

We have generated the arrival times and service times using this function.

Now coming to the performance metrics, we had to calculate the following:

- **Average Waiting Time**: The average time passengers spend waiting in line before their security checks.
- **Average Queue Length**: The average number of passengers in the queue at any given time.
- **System Utilization**: The percentage of time the security scanner is actively processing passengers.

To calculate Average Waiting Time, we have taken the sum of waiting times of each passenger in each queue and then divided the sum by the total passengers serviced by the queue. We didn't take into account the dropped passengers. For finding the Average Waiting Time of the system, we took the sum of waiting times of all the passengers seriviced by the queues and divided it by the total number of passengers serviced.

To calculate Average Queue Length, we have taken the sum of queue lengths between every arrival and departure for every queue and then divided it by the time by which the last passenger in the corresponding queue was serviced. For finding the Average Queue Length of the system, we took the sum of the Average Queue Lengths of all the security scanners and divided it by the total time of simulation or the time by which all the passengers were serviced.

To calculate the System Utilization of a server/security scanner, we have taken the sum of times the security scanner was active or was servicing a passenger. Then, we divided this by the total simulation time and multiplied by 100 to get the System Utilization.

CASE 1:

In the base case, m = 1 and k = Infinity (for the simulation, we have taken to k = 1e9 to resemble infinity). In other words, we have one security scanner and an infinite buffer. So, the passengers can wait in the buffer till they get their chance to be served. Thus, this concludes that there will be no dropped passengers and we might have a high average queue length and waiting time.

CASE 1A: $\lambda < \mu$.

```
Printing statistics:

The number of security scanners in the system: 1
The size of buffer: 1000000000
Arrival rate: 5
Service rate: 8
Total number of passengers arrived: 50
Total number of passengers serviced: 50
Total number of passengers dropped: 0

Total waiting time for queue of Server 0: 10.2458
Total numbers of passengers serviced by the Server 0: 50
Average waiting time of the queue of Server 0: 0.204917

Average waiting time: 0.204917

Total time of simulation: 10.4171

Total queue length for queue of Server 0: 4.87535
Total time till which Server 0 was servicing: 10.4171
Average queue length of the queue of Server 0: 0.468016

Average queue length: 0.468016

Total time Server 0 was active: 6.55885
Percentage of time Server 0 was active: 62.9626
```

CASE 1B: $\lambda > \mu$

```
Printing statistics:

The number of security scanners in the system: 1
The size of buffer: 1000000000
Arrival rate: 8
Service rate: 5
Total number of passengers arrived: 50
Total number of passengers serviced: 50
Total number of passengers dropped: 0

Total waiting time for queue of Server 0: 86.6559
Total numbers of passengers serviced by the Server 0: 50
Average waiting time of the queue of Server 0: 1.73312

Average waiting time: 1.73312

Total time of simulation: 9.87868

Total queue length for queue of Server 0: 82.727
Total time till which Server 0 was servicing: 9.87868
Average queue length of the queue of Server 0: 8.37429

Average queue length: 8.37429

Total time Server 0 was active: 9.73443
Percentage of time Server 0 was active: 98.5397
```

As we can see, the average queue length and average waiting time are greater when arrival rate is greater than service rate. It is also because of infinite buffer which ensures that all passengers will be serviced.

CASE 2:

In this case, m = 1 and k = 2. In other words, we have one security scanner and a finite buffer. So, the passengers can wait in the buffer till they get their chance to be served. If an incoming passengers sees that all buffers are filled, it gets dropped.

CASE 2A: $\lambda < \mu$.

```
Printing statistics:

The number of security scanners in the system: 1
The size of buffer: 2
Arrival rate: 4
Service rate: 7
Total number of passengers arrived: 50
Total number of passengers serviced: 48
Total number of passengers dropped: 2

Total waiting time for queue of Server 0: 9.17727
Total numbers of passengers serviced by the Server 0: 48
Average waiting time of the queue of Server 0: 0.191193

Average waiting time: 0.191193

Total time of simulation: 11.2126

Total queue length for queue of Server 0: 4.43213
Total time till which Server 0 was servicing: 11.2126
Average queue length of the queue of Server 0: 0.395281

Average queue length: 0.395281

Total time Server 0 was active: 6.0797
Percentage of time Server 0 was active: 54.222
```

CASE 2B: $\lambda > \mu$

```
Printing statistics:

The number of security scanners in the system: 1
The size of buffer: 2
Arrival rate: 7
Service rate: 4
Total number of passengers arrived: 50
Total number of passengers serviced: 11
Total number of passengers dropped: 39

Total waiting time for queue of Server 0: 10.1433
Total numbers of passengers serviced by the Server 0: 11
Average waiting time of the queue of Server 0: 0.922116

Average waiting time: 0.922116

Total time of simulation: 5.34323

Total queue length for queue of Server 0: 9.18798
Total time till which Server 0 was servicing: 5.34323
Average queue length of the queue of Server 0: 1.71956

Average queue length: 1.71956

Total time Server 0 was active: 5.34323
Percentage of time Server 0 was active: 100
```

Since the servers have finite buffers, we observe that some passengers were dropped.
On comparing case 2A and 2B, average waiting time and average queue length are greater in the

latter because the number of passengers being served is less than the number of passengers arriving in given time interval. Moreover, the number of dropped packets in case 2A is less than 2B, because of the same reason.

CASE 3:

In this case, m = 3 and k = 1e9 (infinity). In other words, we have many security scanners and an infinite buffer. So, the passengers can choose the security scanner. To optimize the waiting time and queue length, an incoming passenger chooses the smallest queue. If all buffers are filled, it gets dropped.

CASE 3A: $\lambda < \mu$.

```
Printing statistics:

The number of security scanners in the system: 3
The size of buffer: 1000000000
Arrival rate: 4
Service rate: 5
Total number of passengers arrived: 500
Total number of passengers serviced: 500
Total number of passengers dropped: 0

Total waiting time for queue of Server 0: 7.15362
Total numbers of passengers serviced by the Server 0: 162
Average waiting time of the queue of Server 0: 0.0441581

Total waiting time for queue of Server 1: 1.82019
Total numbers of passengers serviced by the Server 1: 165
Average waiting time of the queue of Server 1: 0.0110314

Total waiting time for queue of Server 2: 0
Total numbers of passengers serviced by the Server 2: 173
Average waiting time of the queue of Server 2: 0

Average waiting time: 0.0179476

Total time of simulation: 122.685

Total queue length for queue of Server 0: 2.71455
Total time till which Server 0 was servicing: 122.685
Average queue length of the queue of Server 0: 0.0221263

Total queue length for queue of Server 1: 0.52746
Total time till which Server 1 was servicing: 122.628
Average queue length of the queue of Server 1: 0.00430131

Total queue length for queue of Server 2: 0
Total time till which Server 2 was servicing: 122.178
Average queue length of the queue of Server 2: 0

Average queue length: 0.00880919

Total time Server 0 was active: 33.563
Percentage of time Server 0 was active: 27.3571
Total time Server 1 was active: 32.3244
Percentage of time Server 1 was active: 26.3476
Total time Server 2 was active: 32.9212
Percentage of time Server 2 was active: 26.834
```

CASE 3B: $\lambda > \mu$

```
Printing statistics:

The number of security scanners in the system: 3
The size of buffer: 1000000000
Arrival rate: 5
Service rate: 4
Total number of passengers arrived: 500
Total number of passengers serviced: 500
Total number of passengers dropped: 0

Total waiting time for queue of Server 0: 18.8739
Total numbers of passengers serviced by the Server 0: 173
Average waiting time of the queue of Server 0: 0.109098

Total waiting time for queue of Server 1: 7.85722
Total numbers of passengers serviced by the Server 1: 174
Average waiting time of the queue of Server 1: 0.0451564

Total waiting time for queue of Server 2: 3.11
Total numbers of passengers serviced by the Server 2: 153
Average waiting time of the queue of Server 2: 0.0203268

Average waiting time: 0.0596822

Total time of simulation: 100.844

Total queue length for queue of Server 0: 10.3368
Total time till which Server 0 was servicing: 100.628
Average queue length of the queue of Server 0: 0.102722

Total queue length for queue of Server 1: 5.23789
Total time till which Server 1 was servicing: 100.844
Average queue length of the queue of Server 1: 0.0519404

Total queue length for queue of Server 2: 2.06548
Total time till which Server 2 was servicing: 100.754
Average queue length of the queue of Server 2: 0.0205002

Average queue length: 0.0583877

Total time Server 0 was active: 42.4931
Percentage of time Server 0 was active: 42.1374
Total time Server 1 was active: 43.7253
Percentage of time Server 1 was active: 43.3592
Total time Server 2 was active: 44.8729
Percentage of time Server 2 was active: 44.4972
```

The waiting time and queue lengths in the current case are much less than case 1, when we had only a single server. This is because now a passenger has more than one options to choose from.

No passengers dropped because infinite buffers.

Queue length and waiting time are lesser in A because $\lambda < \mu$.

CASE 4:

In this case, m = 3 and k = 5. In other words, we have many security scanner and a finite buffer. So, the passengers can choose the security scanner. To optimize the waiting time and queue length, an incoming passenger chooses the smallest queue. If all buffers are filled, it gets dropped.

CASE 4A: $\lambda < \mu$.

```
Printing statistics:

The number of security scanners in the system: 2
The size of buffer: 1
Arrival rate: 5
Service rate: 6
Total number of passengers arrived: 500
Total number of passengers serviced: 487
Total number of passengers dropped: 13

Total waiting time for queue of Server 0: 28.1488
Total numbers of passengers serviced by the Server 0: 256
Average waiting time of the queue of Server 0: 0.109956

Total waiting time for queue of Server 1: 12.4762
Total numbers of passengers serviced by the Server 1: 231
Average waiting time of the queue of Server 1: 0.0540095

Average waiting time: 0.0834189

Total time of simulation: 101.873

Total queue length for queue of Server 0: 12.0144
Total time till which Server 0 was servicing: 101.691
Average queue length of the queue of Server 0: 0.118147

Total queue length for queue of Server 1: 5.23854
Total time till which Server 1 was servicing: 101.873
Average queue length of the queue of Server 1: 0.0514223

Average queue length: 0.0847846

Total time Server 0 was active: 41.6018
Percentage of time Server 0 was active: 40.837
Total time Server 1 was active: 40.7537
Percentage of time Server 1 was active: 40.0045
```

CASE 4B: $\lambda > \mu$

```
The number of security scanners in the system: 3
The size of buffer: 2
Arrival rate: 8
Service rate: 2
Total number of passengers arrived: 500
Total number of passengers serviced: 375
Total number of passengers dropped: 125

Total waiting time for queue of Server 0: 98.3707
Total numbers of passengers serviced by the Server 0: 119
Average waiting time of the queue of Server 0: 0.826645

Total waiting time for queue of Server 1: 87.2546
Total numbers of passengers serviced by the Server 1: 119
Average waiting time of the queue of Server 1: 0.733232

Total waiting time for queue of Server 2: 76.8265
Total numbers of passengers serviced by the Server 2: 137
Average waiting time of the queue of Server 2: 0.560778

Average waiting time: 0.699872

Total time of simulation: 62.5313
Total queue length for queue of Server 0: 84.9605
Total time till which Server 0 was servicing: 61.6815
Average queue length of the queue of Server 0: 1.37741

Total queue length for queue of Server 1: 75.1002
Total time till which Server 1 was servicing: 62.5313
Average queue length of the queue of Server 1: 1.201

Total queue length for queue of Server 2: 64.7776
Total time till which Server 2 was servicing: 61.4928
Average queue length of the queue of Server 2: 1.05342

Average queue length: 1.21061

Total time Server 0 was active: 59.612
Percentage of time Server 0 was active: 95.3314
Total time Server 1 was active: 59.2552
Percentage of time Server 1 was active: 94.7608
Total time Server 2 was active: 56.1546
Percentage of time Server 2 was active: 89.8023
```

In this case, we can observe that both waiting times and queue lengths are smaller than the previous cases. Moreover, in case A, we see that server utilization, queue lengths and waiting times are less than B because in case A, $\lambda < \mu$, thus the number of passengers in the buffer is less than B.

Conclusion:

Understanding the performance metrics for both single-queue and multi-queue systems is crucial for optimizing system performance and resource allocation. These metrics play a vital role in providing efficient and reliable services, minimizing waiting times, and enhancing customer satisfaction. By analysing and applying the appropriate queueing models, organizations can improve operational efficiency and service quality.