REPORT

INTRODUCTION

Efficient management of airport security lines is a critical aspect of ensuring passenger satisfaction and smooth operations. This comprehensive report presents an in-depth analysis of various models and optimization strategies for the airport security screening process, focusing on the implementation of queue structures in a C++ simulation. The study aims to evaluate the impact of different arrival rates and departure rates on the average queue length, system utilization, and overall passenger experience.

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.
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.

REPORT

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.

METHODOLOGY

A C++ simulation was developed to model the airport security screening process as a queuing system, considering both single and multiple security lines. The simulation allowed for the adjustment of arrival rates and departure rates, facilitating the assessment of their effects on the overall efficiency of the security screening process. Key performance metrics such as average waiting time, average queue length, and system utilization were meticulously collected and analyzed for various scenarios and optimizations.

1.EFFECT OF CHANGING THE ARRIVAL AND SERVICE RATES::

A.PROCESS

- We collected the server utilisations in various cases as shown below
- We started with (0.2,0.3) as the arrival and service rates and increased the value of service rates by 0.1 and took 4 such readings .
- Then we increased the arrival rates by 0.1 till it became 0.8 and collected 7 such arrival rates and in total took 7*4=28 readings

B.OBSERVATION::

AS THE DEPARTURE RATES INCREASED FOR A FIXED LAMBDA;THE SERVICE UTILISATION WENT ON DECREASING THE SERVICE UTILISATION;WHEREAS AS THE LAMBDA VALUE WAS INCREASED THERE WAS A STARK INCREASE IN SERVER UTILISATION AS WELL.

C.CONCLUSION::

TAKE A LARGE LAMBDA VALUE(SAY 0.8) AND THEN KEEP THE DEPARTURE RATE CLOSE TO IT (SAY 0.9) ;ALSO ENSURE THAT LAMDA<MU(STABILITY CONSTRAINT).

```
Enter arrival rate:0.2
Enter Departure rate:0.3
Enter the simulated time:999999
Avg queue length in single server:1.44402
Avg waiting time in single server:6.54737
Server utilization in single server:0.629269
Note: we took int in place of double
```

```
himangshudeka@Himangshu-Dekas-Macbook-Air Week4 % ./a.out
Enter arrival rate:0.2
Enter Departure rate:0.4
Enter the simulated time:999999
Avg queue length in single server:0.54229
Avg waiting time in single server:2.4479
Server utilization in single server:0.449237
Note: we took int in place of double
Do you want to rerun?(y/n):
```

```
Enter arrival rate:0.8
Enter Departure rate:0.9
Enter the simulated time:999999
Avg queue length in single server:10.9274
Avg waiting time in single server:8.91708
Server utilization in single server:0.83927
```

```
Enter Departure rate:1
Enter the simulated time:999999
Avg queue length in single server:4.87825
Avg waiting time in single server:3.97738
Server utilization in single server:0.713565
Note: we took int in place of double
```

```
Enter arrival rate:0.3
Enter Departure rate:0.4
Enter the simulated time:999999
Avg queue length in single server:2.59261
Avg waiting time in single server:7.41215
Server utilization in single server:0.710708
```

```
Enter the simulated time:999999
Avg queue length in single server:3.9638
Avg waiting time in single server:8.04901
Server utilization in single server:0.75967
```

REPORT

2.EFFECT OF PUTTING A BUFFER OF OPTIMAL SIZE(I.E FINDING AN OTIMAL SIZE FOR BUFFER FOR A GIVEN λ and µ.

A. OBSERVATION:

- We fixed λ and µ values as 0.8 and 0.9 (because in the infinte buffer model this was the case we decided would yield us the largest server utilisation.
- Now we gradually increased the buffer size value from 10 to 10,000 and found out that the changes were stark and significant between 10 and 160 ,pointing to exponential or there-about increase,however after that the server utilisation values obtained evened out ;thus increasing our faith in the dependence being exponential.
- Thus we conclude that for the values of λ and µ ,that we took ;200(for safety margin) is a good buffer size,any increase beyond that in buffer size pays little dividends in server utilisation value.

```
Now running M/M/k queue model for this airport...
Enter buffer size:10
Enter the simulated time:999999
Avg queue length in single server:6.40001e-05
Avg waiting time in single server:1.6
Server utilization in single server with finite buffer of size 10:2e-05
Do you want to change the buffer size?(y/n):y
Enter buffer size:50
Enter the simulated time:999999
Avg queue length in single server:0.00989501
Avg waiting time in single server:6.71758
Server utilization in single server with finite buffer of size 50:0.000989001
Do you want to change the buffer size?(y/n):y
Enter buffer size:100
Enter the simulated time:999999
Avg queue length in single server:0.833816
Avg waiting time in single server:8.43772
Server utilization in single server with finite buffer of size 100:0.0671971
Do you want to change the buffer size?(y/n):y
Enter buffer size:125
Enter the simulated time:999999
Avg queue length in single server:10.5867
Avg waiting time in single server:8.62485
Server utilization in single server with finite buffer of size 125:0.840947
Do you want to change the buffer size?(y/n):y
Enter buffer size:150
Enter the simulated time:999999
Avg queue length in single server:8.71539
Avg waiting time in single server:8.75607
Server utilization in single server with finite buffer of size 150:0.681654
Do you want to change the buffer size?(y/n):y
Enter buffer size:175
Enter the simulated time:999999
Avg queue length in single server:10.7905
Avg waiting time in single server:8.79343
```

B CONCLUSION:Thus instead of wasting buffer space(by considering it infinite) we can improve upon our space constraint by introducing a buffer of finite size.

3.EFFECT OF INCREASING THE NO. OF SERVERS(M/M/C MODEL)

A.OBSERVATION

- As the no of servers are increased,server utilisation and the the avg. queue lengths in system and waiting times decreased pointing to a wastage of resources allocated.
- Also the fall from n=1 to n=5 is sharp and there after the curve sort of evens out pointing to an inverse exponential relation between the server utilisation.
- For n=5 we find that the server utilisation is the smallest(0.25) and thereafter evens out.

REPORT

B.CONCLUSION:

Thus introducing 5 servers is beneficial for it decreases the avg waiitng times for the passengers and any further increase in servers is a wastage of resource for it achieves little

4.EFFECT OF PUTTING A FINITE SIZE BUFFER NOW(M/M/C/K MODEL) :

A.OBSERVATION:

In this we observe that as the buffer size is increased from 1 to say around 25(while keeping the λ and μ values as 0.8 and 0.9 and 5 servers),the increase in server utility is sharp and thereafter it flattens out,so we can say that having a buffer of 25 is optimal

```
Enter buffer size:200
Enter the simulated time:999999
Enter the simulated time:99999
Enter the number of servers:5
Enter the buffer size:10
Avg queue length in multi server with buffer size 10:0
Avg waiting time in multi server with buffer size 10:0
Server utilization in multi server with buffer size 10:0.000142001
Do you want to enter number of servers and buffer size again?(y/n):y
Enter the simulated time:5
Enter the number of servers:5
Enter the buffer size:5
Avg queue length in multi server with buffer size 5:0
Avg waiting time in multi server with buffer size 5:0
Server utilization in multi server with buffer size 5:0.04
Do you want to enter number of servers and buffer size again?(y/n):y
Enter the simulated time:99999
Enter the number of servers:5
Enter the buffer size:15
Avg queue length in multi server with buffer size 15:0.000480005
Avg waiting time in multi server with buffer size 15:0.0162382
Server utilization in multi server with buffer size 15:0.00241402
Do you want to enter number of servers and buffer size again?(y/n):y
Enter the simulated time:99999
Enter the number of servers:5
Enter the buffer size:25
Avg queue length in multi server with buffer size 25:0.0143101
Avg waiting time in multi server with buffer size 25:0.0116302
Server utilization in multi server with buffer size 25:0.103547
Do you want to enter number of servers and buffer size again?(y/n):y
Enter the simulated time:99999
Enter the number of servers:5
Enter the buffer size:100
Avg queue length in multi server with buffer size 100:0.0134901
Avg waiting time in multi server with buffer size 100:0.0109879
Server utilization in multi server with buffer size 100:0.103815
```

5.ANALYSIS:

- Taking a larger value of λ and a value of μ near λ (probably a difference of 0.1) increases system utilisation in M/M/1 model of airport
- Then letting our airport have 5 servers instead of 1 achieves a lower probability of package drop and a shorter wait times in queue and in the system.
- Then finally we extend all the above to now constrain the buffer to a size of 25.

Results and Comprehensive Analysis and Optimization Strategies for Airport Security Screening Process

Impact of Arrival Rates on Queue Length and System Utilization

The simulation results demonstrated a direct correlation between higher arrival rates and increased average queue length, leading to longer waiting times for passengers. The system utilization was significantly impacted, with higher arrival rates resulting in decreased efficiency due to heightened congestion and increased idle time for security scanners. Conversely, lower arrival rates improved system utilization, leading to a smoother and more efficient screening process.

Effect of Departure Rates on Waiting Time and Passenger Flow

The departure rate played a crucial role in determining the average waiting time for passengers in the queue. Lower departure rates were found to increase waiting times, causing passenger dissatisfaction and congestion. However, higher departure rates facilitated a more streamlined flow through the security line, reducing waiting times and enhancing overall passenger satisfaction.

Optimization Strategies

Buffer Size Adjustment for Improved Passenger Flow

Increasing the buffer size proved to be an effective strategy in minimizing the number of rejected passengers due to a full buffer. By accommodating a greater number of waiting passengers, the optimization strategy contributed to smoother operations and reduced passenger dissatisfaction, ultimately enhancing the overall airport experience.

6.OPTIMISATION AND FURTHER POSSIBILITIES OF ENHANCEMENT::

Dynamic Staff Allocation for Adaptive Resource Management

Implementing dynamic staff allocation strategies based on real-time passenger flow data significantly improved resource utilization and queue management. By allocating staff according to fluctuating arrival rates and queue lengths, airports could optimize security operations and ensure efficient utilization of personnel, leading to a more seamless and satisfactory passenger experience.

QUEUEING THEORY PROBLEMS : MODEL – 2

Multiple server Model

M / M / K : ∞ / FCFS

Formulae

Multiple server Model

(1)

$$P_o = \frac{1}{\left[\sum_{n=0}^{K-1} \frac{1}{n!}\left(\frac{\lambda}{\mu}\right)^n\right] + \frac{1}{K!}\left(\frac{\lambda}{\mu}\right)^K \frac{k\mu}{(k\mu-\lambda)}}$$

(2)

$$P_n = \frac{1}{n!}\left(\frac{\lambda}{\mu}\right)^n . P_o$$

(3)

$$L_q = \frac{\lambda\mu\left(\frac{\lambda}{\mu}\right)^K}{(k-1)!(k\mu-\lambda)^2} \times P_o$$

(4)

$$L = L_q + \frac{\lambda}{\mu}$$

(5)

$$W_q = \frac{L_q}{\lambda}$$

Comprehensive System Utilization Analysis

The analysis of system utilization highlighted the importance of maintaining an optimal balance between passenger flow and resource utilization. A dynamic approach to managing the system utilization, considering the fluctuations in passenger traffic, emerged as a key factor in maximizing the overall efficiency of the security screening process.

Conclusion

The findings underscore the critical significance of implementing effective optimization strategies in managing the airport security screening process. By adjusting arrival rates, departure rates, buffer sizes, and staff allocations, airports can ensure smoother operations, reduced waiting times, and improved passenger satisfaction. The integration of dynamic management systems and real-time analytics is vital in facilitating adaptive security line management and ensuring a seamless airport experience for all passengers.

Recommendations for Future Implementation

Based on the extensive analysis, it is recommended that airports consider the integration of advanced technologies such as artificial intelligence and machine learning in their security line management systems. Real-time data analysis and predictive modeling can enable proactive decision-making, leading to more efficient resource utilization and enhanced passenger satisfaction. Additionally, the implementation of automated queue management systems and self-service kiosks can further optimize the screening process, reducing wait times and improving overall operational efficiency.

Conclusion

Taking arrival rate as 0.8 and departure rate as 0.9,with 5 servers and a buffer of 25 size ,ensured that probability of a packet being dropped was low (negligible in fact) and a proper mitigation between system utilisation and the waiting times was achieved.

Using a single server and a buffer size 200 ,a much higher system utilisation could be achieved.

It finally boils down to the requirement at hand,i.e. whether we prioritise each passengers waiting times or whether we prioritise the utility of the airport and the fact that no pax. should lose his flight owing to long waiting queues,we chose either M/M/C or M/M/C/k models with optimal C/K values;

Taking higher λ and μ values enable us to be prepared for the worst situations..