Report on Airport Security Line Analysis and Optimization

Summary

In our study, we explored the efficiency of airport security lines through simulation. Our primary findings indicate that a single server queue works well within Little's Law, with sufficient buffer space to accommodate passenger fluctuations.

However, inadequate buffer space results in the dropping of passengers, which can significantly impact passenger satisfaction. To enhance the security screening process, we expanded to multi-server queues and implemented various server selection algorithms to improve system utilization.

The methods we used for server selection included Shortest Queue Length First, and Randomized allocation.

Simulation Setup

We initiated our study by modeling the airport security screening process as a single server queue with a security scanner. This served as the foundation for our subsequent explorations. We then expanded the model to incorporate multiple queues and analyzed the impact of varying buffer sizes on passenger satisfaction and system efficiency.

User-Defined Parameters

Our study demonstrated the critical importance of user-defined parameters, specifically the Arrival Rate (λ) and Service Rate (μ). When we implemented a low arrival rate and high service rate, the queue length was minimal. In contrast, as we increased the arrival rate, the queue length expanded, eventually leading to the dropping of passengers.

Data Collection and Analysis

Throughout our simulations, we collected essential statistics to evaluate the performance of the security screening process. These statistics included:

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

SINGLE SERVER

<u>CASE 1</u>: Single server, No buffer

a) Arrival rate=10
   Service rate=15

```
Total waiting time: 0
Total service time: 39.2263

Average waiting time: 0
Average service time: 0.0392263

Average queue length for checkpoint 0 is 0
System utilization for checkpoint 0 is 0.40862

Number of passengers served by checkpoint 0: 578

Number of packets dropped: 422
```

b) Arrival rate=10
Service rate=40

```
Total waiting time: 0
Total service time: 19.9603

Average waiting time: 0
Average service time: 0.0199603

Average queue length for checkpoint 0 is 0
System utilization for checkpoint 0 is 0.198619

Number of passengers served by checkpoint 0: 817

Number of packets dropped: 183
```

c) Arrival rate=10
Service rate = 1000 (perfectly optimal service rate)

```
Total waiting time: 0
Total service time: 1.0467

Average waiting time: 0
Average service time: 0.0010467

Average queue length for checkpoint 0 is 0
System utilization for checkpoint 0 is 0.0102304

Number of passengers served by checkpoint 0: 986

Number of packets dropped: 14
```

In case we have a single server with no buffer, packets will be dropped (or in this case passengers will be denied service at checkpoint) because while one passenger is being processed and some other passenger comes he will be denied service. But if we keep on increasing the service rate, i.e, more number of passengers are serviced per unit of time, then we observed that less number of packets will be dropped which makes sense intuitively as well.

<u>CASE 2:</u> Single server, finite buffer length

a) Arrival rate=10
   Service rate = 15
   Buffer size=50

```
Total waiting time: 140.583
Total service time: 64.7943

Average waiting time: 0.140583
Average service time: 0.0647943

Average queue length for checkpoint 0 is 1.4301
System utilization for checkpoint 0 is 0.659126

Number of passengers served by checkpoint 0: 1000

Number of packets dropped: 0
```

Here, we have observed that the service rate is sufficiently higher than the arrival rate, coupled with a sufficiuently large buffer to store the requests, to ensure that no passenger is denied service.

b) Arrival rate=20
   Service rate = 15
   Buffer size=50

```
Total waiting time: 2293.1
Total service time: 54.5958

Average waiting time: 2.2931
Average service time: 0.0545958

Average queue length for checkpoint 0 is 41.985
System utilization for checkpoint 0 is 0.999612

Number of passengers served by checkpoint 0: 801

Number of packets dropped: 199
```

In this case, now we have that the arrival rate is higher than the service rate and the buffer size is not sufficient to hold all the requests at any given point of time, so requests will be dropped accordingly.

<u>CASE 3</u>: Single Server, Infinite buffer

a) Arrival rate=10
   Service rate=15
   Buffer size=100000(infinite as compared to the number of passengers arriving)

```
Total waiting time: 13250.8
Total service time: 6653.45

Average waiting time: 0.132508
Average service time: 0.0665345

Average queue length for checkpoint 0 is 1.32966
System utilization for checkpoint 0 is 0.66764

Number of passengers served by checkpoint 0: 100000

Number of packets dropped: 0
```

In this case we have infinite buffer capacity and a single queue, so we can actually refer to little’s law to analyse our requests(service rate> arrival rate)

Average waiting time(according to little's law)=(1/(service rate- arrival rate) - 1/service rate)
=1/(15-10)- 1/15= 2/15= 0.133

Average service time= 1/(service time- arrival time) - avg waiting time= 0.2- 0.133= 0.066

Average queue length length= (arrival rate)*(average waiting time)=10*0.13= 1.33

System utilisation= arrival rate/ service rate= 10/15= 0.66

We observed that the values be obtained in our simulation are similar to those that are obtained according to the formulas.

We also observed that the in case of a single server, if we increase the buffer size then less number of packets are dropped and better service is provided.

MULTIPLE SERVERS

<u>CASE 1</u>: multiple servers with 0 buffer capacity

a) Arrival rate= 10
   Service rate= 15
   Number of servers= 5
   Randomized allcoation of servers

```
Average waiting time: 0
Average service time: 0.0586794

Average queue length for checkpoint 0 is 0
System utilization for checkpoint 0 is 0.121474
Average queue length for checkpoint 1 is 0
System utilization for checkpoint 1 is 0.127478
Average queue length for checkpoint 2 is 0
System utilization for checkpoint 2 is 0.107755
Average queue length for checkpoint 3 is 0
System utilization for checkpoint 3 is 0.108398
Average queue length for checkpoint 4 is 0
System utilization for checkpoint 4 is 0.123439
```

```
Number of passengers served by checkpoint 0: 177
Number of passengers served by checkpoint 1: 185
Number of passengers served by checkpoint 2: 184
Number of passengers served by checkpoint 3: 177
Number of passengers served by checkpoint 4: 170

Number of packets dropped: 107
```

b) Arrival rate= 10
   Service rate= 15
   Number of servers= 5
   Shortest queue first (Give to free server, drop if no one is free)

```
Average waiting time: 0
Average service time: 0.0631853

Average queue length for checkpoint 0 is 0
System utilization for checkpoint 0 is 0.000162353
Average queue length for checkpoint 1 is 0
System utilization for checkpoint 1 is 0.0119044
Average queue length for checkpoint 2 is 0
System utilization for checkpoint 2 is 0.0480789
Average queue length for checkpoint 3 is 0
System utilization for checkpoint 3 is 0.16737
Average queue length for checkpoint 4 is 0
System utilization for checkpoint 4 is 0.373886
```

```
Number of passengers served by checkpoint 0: 1
Number of passengers served by checkpoint 1: 12
Number of passengers served by checkpoint 2: 87
Number of passengers served by checkpoint 3: 275
Number of passengers served by checkpoint 4: 625

Number of packets dropped: 0
```

If we have randomized server allocation, then packets are dropped but if we have an optimised algo like shortest queue first, then no packets are dropped and the average service time is more in the latter case.

<u>CASE 2</u>: Finite Buffer (k=10)

a) Arrival rate=10
   Service rate=15
   Number of servers=5
   Randomized allocation of servers

```
Average waiting time: 0.0101784
Average service time: 0.0667705

Average queue length for checkpoint 0 is 0.0201815
System utilization for checkpoint 0 is 0.132196
Average queue length for checkpoint 1 is 0.0198831
System utilization for checkpoint 1 is 0.132782
Average queue length for checkpoint 2 is 0.0209239
System utilization for checkpoint 2 is 0.134095
Average queue length for checkpoint 3 is 0.0197486
System utilization for checkpoint 3 is 0.134075
Average queue length for checkpoint 4 is 0.0207648
System utilization for checkpoint 4 is 0.132707
```

```
Number of passengers served by checkpoint 0: 19960
Number of passengers served by checkpoint 1: 19954
Number of passengers served by checkpoint 2: 20159
Number of passengers served by checkpoint 3: 20157
Number of passengers served by checkpoint 4: 19770

Number of packets dropped: 0
```

Here, we can see that the average waiting time is less than that of single queue with finite buffer with same arrival rate and service rate.

b) Arrival rate = 50
   Service rate = 15
   Number of servers=5
   Randomized allocation of servers

```
Average waiting time: 0.134888
Average service time: 0.0674327

Average queue length for checkpoint 0 is 1.01969
System utilization for checkpoint 0 is 0.648624
Average queue length for checkpoint 1 is 1.5752
System utilization for checkpoint 1 is 0.684666
Average queue length for checkpoint 2 is 1.33894
System utilization for checkpoint 2 is 0.674155
Average queue length for checkpoint 3 is 1.33255
System utilization for checkpoint 3 is 0.679434
Average queue length for checkpoint 4 is 1.45893
System utilization for checkpoint 4 is 0.675214
```

```
Number of passengers served by checkpoint 0: 1939
Number of passengers served by checkpoint 1: 2006
Number of passengers served by checkpoint 2: 2034
Number of passengers served by checkpoint 3: 1983
Number of passengers served by checkpoint 4: 2001

Number of packets dropped: 37
```

Since service rate< arrival rate, so packets are dropped.

c) Arrival rate=50
   Service rate=15
   Number of servers=5
   k=10 for each server
   Shortest queue first (Give to free server, drop if no one is free)

```
Average waiting time: 0.0214442
Average service time: 0.0666547

Average queue length for checkpoint 0 is 0.39041
System utilization for checkpoint 0 is 0.566083
Average queue length for checkpoint 1 is 0.279275
System utilization for checkpoint 1 is 0.597352
Average queue length for checkpoint 2 is 0.188402
System utilization for checkpoint 2 is 0.65604
Average queue length for checkpoint 3 is 0.131991
System utilization for checkpoint 3 is 0.720826
Average queue length for checkpoint 4 is 0.0803382
System utilization for checkpoint 4 is 0.786854
```

```
Number of passengers served by checkpoint 0: 1651
Number of passengers served by checkpoint 1: 1776
Number of passengers served by checkpoint 2: 1992
Number of passengers served by checkpoint 3: 2185
Number of passengers served by checkpoint 4: 2396

Number of packets dropped: 0
```

The arrival rate, service rate and buffer size are similar to the previous case, but the server allocation algo is more optimal, so no packets are dropped and the service and waiting times are more efficient.

<u>CASE 3:</u> multiple servers, infinite buffer size

a) Arrival rate=50
   Service rate=15
   Number of servers=5
   k=10000 for each server
   Shortest queue first (Give to free server, drop if no one is free)

```
Average waiting time: 0.0213928
Average service time: 0.0666844

Average queue length for checkpoint 0 is 0.383393
System utilization for checkpoint 0 is 0.561358
Average queue length for checkpoint 1 is 0.275218
System utilization for checkpoint 1 is 0.600241
Average queue length for checkpoint 2 is 0.192866
System utilization for checkpoint 2 is 0.663531
Average queue length for checkpoint 3 is 0.131268
System utilization for checkpoint 3 is 0.730694
Average queue length for checkpoint 4 is 0.089474
System utilization for checkpoint 4 is 0.786439
```

```
Number of passengers served by checkpoint 0: 16782
Number of passengers served by checkpoint 1: 18013
Number of passengers served by checkpoint 2: 19858
Number of passengers served by checkpoint 3: 21802
Number of passengers served by checkpoint 4: 23545

Number of packets dropped: 0
```

b) Arrival rate=50
Service rate=15
Number of servers=5
k=10000 for each server
Random server selection algorithm

```
Average waiting time: 0.13655
Average service time: 0.0667632

Average queue length for checkpoint 0 is 1.42626
System utilization for checkpoint 0 is 0.667138
Average queue length for checkpoint 1 is 1.3512
System utilization for checkpoint 1 is 0.673943
Average queue length for checkpoint 2 is 1.3583
System utilization for checkpoint 2 is 0.671589
Average queue length for checkpoint 3 is 1.35247
System utilization for checkpoint 3 is 0.6743
Average queue length for checkpoint 4 is 1.37886
System utilization for checkpoint 4 is 0.670545
```

```
Number of passengers served by checkpoint 0: 19973
Number of passengers served by checkpoint 1: 19989
Number of passengers served by checkpoint 2: 20100
Number of passengers served by checkpoint 3: 20029
Number of passengers served by checkpoint 4: 19909

Number of packets dropped: 0
```

Server Selection Algorithms

To further optimize the multi-server queues, we applied two different server selection algorithms:
1. Shortest Queue Length First: Passengers are directed to the queue with the fewest passengers.
2. Randomized Allocation: Passengers are assigned to queues randomly.

Our findings revealed that the choice of server selection algorithm can significantly impact the overall system utilization and passenger waiting times. Shortest Queue Length First algo has better results.

Conclusions:

- Most efficient use of resources depends on passenger flow. Regularly analyze and adjust queue parameters based on passenger flow,i.e, increase the buffer size as rate of inflow of passengers increases
- Invest in buffer space to avoid passenger drops during peak periods.

- Consider multi-server queues for higher loads, because as shown by our results that will not only lead to better load balancing but also result in much higher quality of service
- Carefully select server selection algorithms to optimize system utilization.