Capacity Analysis

Project report

Project by: A.Lakshmikanth

19991213-T217

Multiple server queuing model- M/M/k

Index

Introduction:

This project is aimed to develop content in Covid-19 category. The impact of the covid -19 pandemic is drastically on people. More than ever the need for personal healthcare and hygienic life is gaining more importance. Increasing problems related to health in this digital era.

Queuing theory refers to the mathematical study of the formation, function, and congestion of waiting lines, or queues. Queuing theory uses the Kendall notation to classify the different types of queuing systems. Most commonly used are M/M/1 (single server system) and M/M/s (multiple server system). Waiting in line is a common occurrence in daily life and it has important functions. Queues are logical and necessary way of coping with the flow of customers. With an in effective queue management, negative outcomes will happen.

MATLAB is an interactive programming environment for scientific computing. For data processing, problem solving, experimentation and algorithm development, MATLAB is widely used in many technical fields. MATLAB has a number of features that separate it from standard scientific programming languages.

The recent Pandemic situation which resulted with spreading corona due to social gatherings, long queue at heavy populated areas are one of the main reasons for global pandemic.

Aim of the project:

Recently I went for a covid test at a nearby community health care center and I had to wait for 50 min in the queue in an enclosed hall before the sample was collected for the test. Being a victim to this situation and resulted with Covid +ve. I tried to understand the situation and wanted to model a multiple server queuing system and observe the queue with respect to time. And also to find, when would it be applicable for me/anyone to go for the test again and at which time it would be faster?

Used Materials:

- Matlab version R2021a and its tools and functions.
- Queuing theory

Used Method:

The multiple server queuing system is used for analyzing the situation. In this model two or more servers are accessible to handle arriving customers.

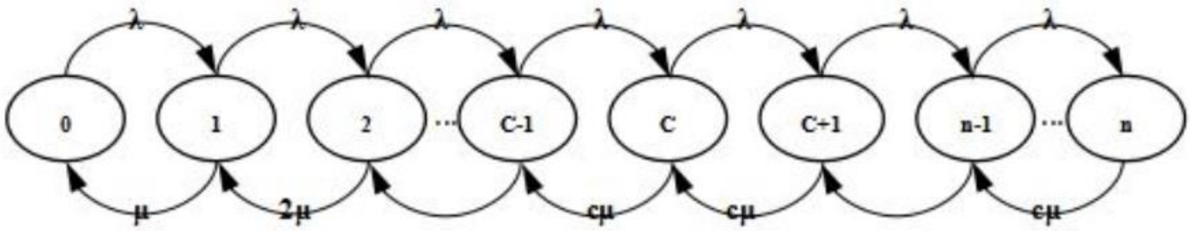
To solve the situation I tried to design a model with 3 servers which follows first come first serve condition. Where the arrival rates is lambda and the service rate is mu with a finite population so as to know the average time spent in the queue, average queue length.

I am assuming a situation where the customers who are waiting for the service form a single line and then continue to the server which is available first. For this model, it is assumed that the arrivals and the service of the system follow Poisson and exponential distribution.

The arrival times of each customer is calculated by adding the inter arrival times and the arrival time of the previous customer. At the initial conditions there won't be any customers in the system so the first customers will directly go to the servers and gets the service. When the next customer arrives, they check the availability of the server and if it's available the customer proceeds to that respected server. The customer waits in the queue if the server is not available. So by this we can analyze the system and the length of the queue with respect to time which gives the efficient times for the customers to get their services done.

M/M/K Queuing system:

M represents that the states are Markovian. The state diagram is the best way to represent functioning of queue with a multiple servers.



This is a state transition diagram of M/M/k Queuing Model.

The model is a type of birth and death process. The server utilization $\rho = \lambda/(c\,\mu)$ has to be less than unity for the queue to be stable and has stationary distribution.

$$\pi_0 = \left[\sum_{k=0}^{c} \frac{\lambda^k}{\mu^k k!} + \frac{\lambda^c}{\mu^c c!} \sum_{k=c+1}^{K} \frac{\lambda^{k-c}}{\mu^{k-c} c^{k-c}}\right]^{-1}$$

$$\pi_k = \begin{cases} \frac{(\lambda/\mu)^k}{k!}\pi_0 & \text{for } k = 1, 2, \ldots, c \\ \frac{(\lambda/\mu)^k}{c^{k-c} c!}\pi_0 & \text{for } k = c+1, \ldots, K. \end{cases}$$

Where πk is the probability that the system has k customers. And $\pi 0$ represents the probability that there is no one in the queue.

The average number of customers in the system is

$$\mathbb{E}[N] = \frac{\lambda}{\mu} + \pi_0 \frac{\rho(c\rho)^c}{(1-\rho)^2 c!}$$

The average response time for a customer is

$$\mathbb{E}[T] = \frac{1}{\mu} + \pi_0 \frac{\rho(c\rho)^c}{\lambda(1-\rho)^2 c!}$$

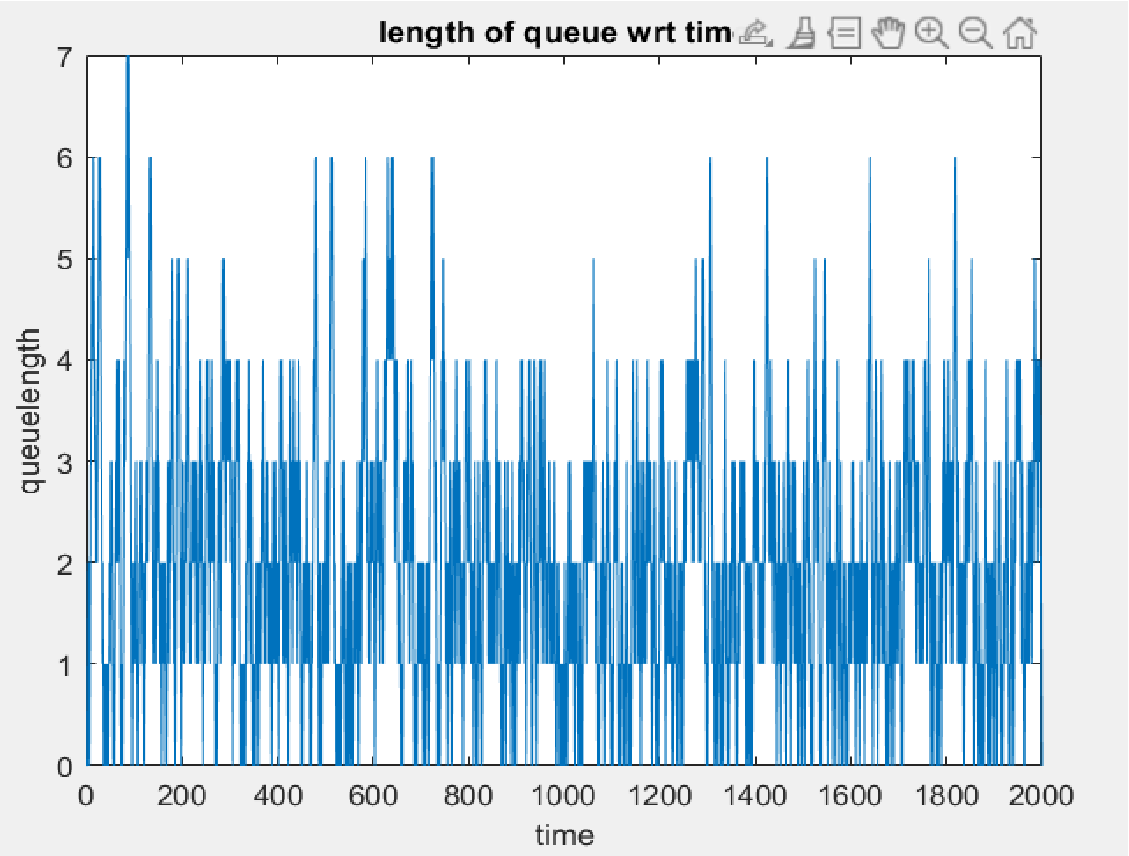
Result:

For,

- Arrival rate = 5 per minute.
- Service rate= 8 per minute.
- Number of servers = 3.

We get a graph which represents the length of queue with respect to time. By this we can pinpoint the time where the length of queue is maximum and schedule the time accordingly. We also know the waiting time of the customer in the queue by which we can estimate and manage our visit to the center.

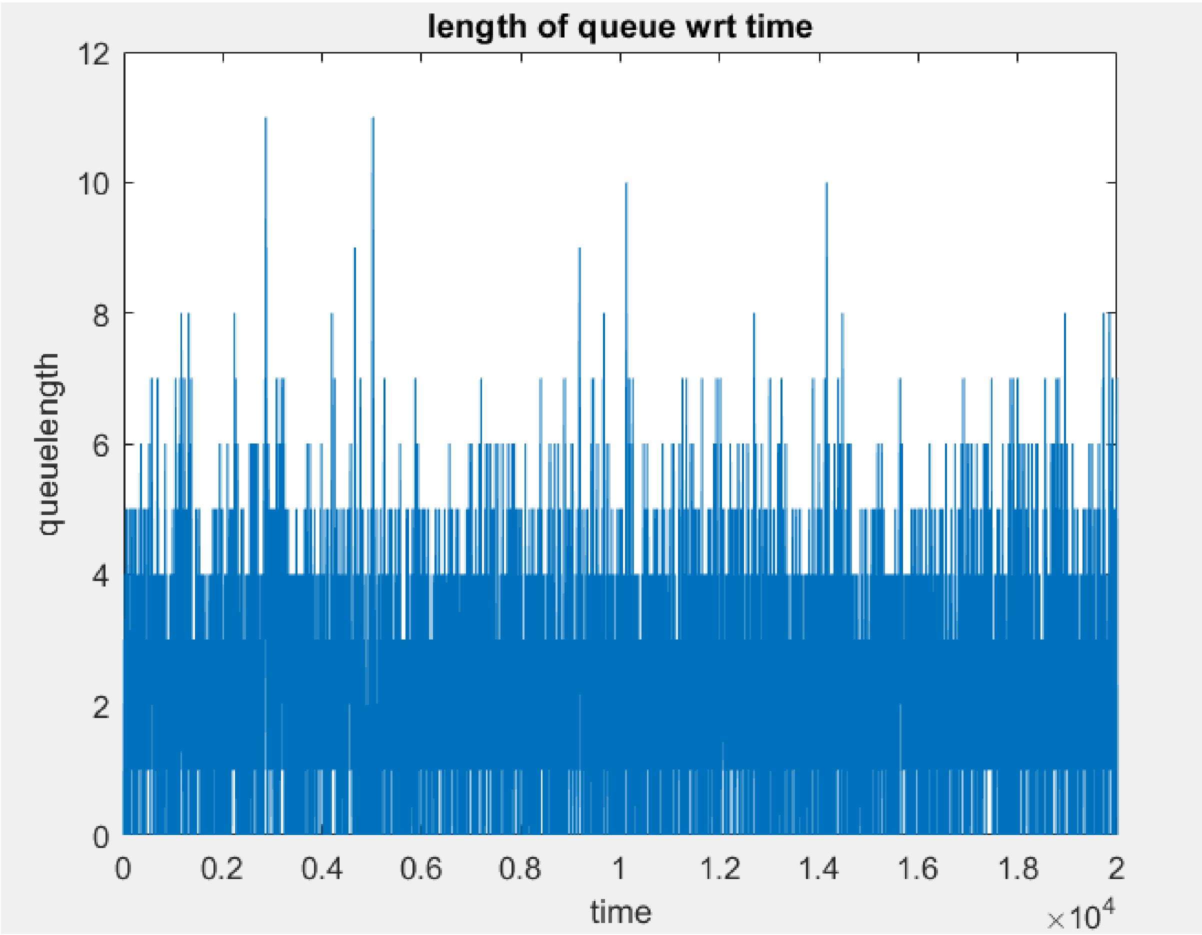
By this study, we can observe and analyze the queue of the test center at certain times which helps in scheduling the service and getting it done effectively.



```
The arrival rate for this system per minute is: 5
What is the service rate of the system per minute?:8
How many servers are there in this system?:3

 The average waiting time for the customer in the queue is: 2401.887171
The average length of queue in the system at any time is : 2.174500
The average time spent by a customer in the system is : 8.024999
```

Even though this model is for finite population. When the population increases the service rate has to be increased so as to cope up with the customers. When the population increases it is a bit difficult to find the minimum queue and schedule accordingly so, what we can do is to find the average length of queue and avoid the times where the length is maximum. We can also make this as a user defined function in Matlab and use it for various scenarios and other applications.



```
The arrival rate for this system per minute is: 4
What is the service rate of the system per minute?:7
How many servers are there in this system?:3

 The average waiting time for the customer in the queue is: 20227.946328
The average length of queue in the system at any time is : 2.255050
The average time spent by a customer in the system is : 7.081747
```

Appendix

```
clear all
close all
clc

lambda=input('The arrival rate for this system per minute is:
');
mu=input('What is the service rate of the system per
minute?:');
ser=input('How many servers are there in this system?:');
%this represents number of servers in the system.
num=1000; % this represents the population.

%Initializing the arrival,queue and departure times.
arrivaltime=zeros(1,num);
queuetime=zeros(1,num);
departuretime=zeros(1,num);

interarrivaltimes=exprnd(lambda,1,num);

servicetime=exprnd(mu,1,num);
for c=2:ser
    servicetime=[servicetime;exprnd(mu,1,num)];
end
%finding the arrivaltimes of the population
arrivaltime(1)=interarrivaltimes(1);
for t=2:num
    arrivaltime(t)=arrivaltime(t-1)+interarrivaltimes(t);
end
%This represents the index number for the servers.
index=zeros(1,ser);
index(1)=1;
recsernum=ones(1,ser);    % this represents to record the
customers who ever leaves.
%queuetime(1)=0;    %As we know at the initial times the queue
will be empty.
departuretime(1)=queuetime(1)+servicetime(1,1);
leav=0;
leavpos=0;
```

```
%This represents calculation of the queuetimes and the
departure times when
%customers are less than number of servers.
for c=1:ser
    queuetime(c)=arrivaltime(c);
    departuretime(c)=queuetime(c)+servicetime(c,recsernum(1));
    recsernum(c)=recsernum(c)+1;
    index(c)=c;
end
minimumtime=0;
minimumtime(1,1)=departuretime(index(1));
for i=2:ser
        minimumtime=[minimumtime;departuretime(index(i))];
end
minimumtime=minimumtime.';

% This loop is to find the queuetime,departure times of the
system when
% customers are greater than the servers.
for c=(ser+1):num

    [leav, leavpos]=min(minimumtime); % this records the time
where the customer leaves which tells that the server is
empty.

    if arrivaltime(c)>leav
        queuetime(c)=arrivaltime(c);
    else
        last=index(leavpos);
        queuetime(c)=servicetime(last);
    end

departuretime(c)=queuetime(c)+servicetime(leavpos,recsernum(le
avpos));
    recsernum(leavpos)=recsernum(leavpos)+1;
    index(leavpos)=c;
end

lenat = zeros(num,2);
lendt = zeros(num,2);

%Here we are noting the arrivaltimes and the departure times
of the
%customers and taking 1 when an arrival happens and -1 when a
departure
%happens.
for i = 1:num
    lenat(i,1) = arrivaltime(1,i);
    lenat(i,2) = 1;

    lendt(i,1) = departuretime(1,i);
```

```
    lendt(i,2) = -1;

end

length=[lenat;lendt];
length=sortrows(length,1);%sorting because the next arrival
happens when the service in any of the servers is done.
lengthfin=zeros(size(length,1),1);%the length final represents
the length of the queue at the respective time stamps.
for c=2:size(length,1)
    if length(c,2)>0
        lengthfin(c,1)=lengthfin(c-1,1) + 1;
    else
        if lengthfin(c-1,1)>0
            lengthfin(c,1)=lengthfin(c-1,1)-1;
        else
            lengthfin(c,1)=0;
        end
    end
end
figure,plot(queuetime);% represents the waitimg time as the
customers keeps joining.
avgwaitinqueue=mean(queuetime);
fprintf('\n The average waiting time for the customer in the
queue is: %f \n' ,avgwaitinqueue);
t=1:size(length,1);
figure,plot(t,lengthfin);title('length of queue wrt
time');xlabel('time');ylabel('queuelength');
averagequeuelength=mean(lengthfin);
fprintf('The average length of queue in the system at any time
is : %f \n' ,averagequeuelength);
avgtimeinsys=mean(departuretime-arrivaltime);
fprintf('The average time spent by a customer in the system is
: %f \n' ,avgtimeinsys);
```

References:

1. https://en.wikipedia.org/wiki/M/M/c_queue

2. https://www.win.tue.nl/~resing/SOR/eng/college10_09_eng.pdf

3. Matlab Documentation.

4. Lecture PPt's and materials provided by Siamak Khatibi.