Typical solution for Project No. 2

ECE 642
Dr. Bijan Jabbari

I Project description: Simulation of an M/M/1 Queue

Consider a high speed transmission link represented by an M/M/1 model of single-server queue with Poisson arrival and exponential service time as depicted below. The link is assumed to have a capacity of 155 Mbps and the length of packets are exponentially distributed with mean packet size of 2325 bits. Plot the queue size (including server), expected queue size, mean delay through the system vs. time for $\rho = 0.50$. Also plot the probability density of the queue size for $\rho = 0.50$. Compare the simulation results with analytical results. Now consider $\rho = 0.3, 0.50, 0.9, 0.95$ and obtain delay vs. throughout of the system. In the same graph plot the analytical results.

II Introduction

The M/M/1 queue is simulated for $\rho = 0.5$. The capacity of the link is specified as 155 Mbps and the length of packets are exponentially distributed with mean packet size 2325 bits. This gives the service rate, $\mu = 155000000/2325$. Hence, we can calculate the arrival rate λ, which is equal to $\mu * \rho$. The inter-arrival and service times are exponentially distributed.

The random number generator function, rand, is used to generate uniform random numbers and with transformation to provide the exponentially distributed inter-arrival and service times:

$$inter - arrival = (-1/\lambda) * log(rand); \tag{1}$$

$$service - time = (-1/\mu) * log(rand); \tag{2}$$

The code simulates an M/M/1 queue for these inter-arrival and service times and then plots the graphs of queue size, expected queue size and mean delay through the system vs time for $\rho = 0.5$. If there is an arrival before the server has finished serving packet, the arrival is buffered and it waits in the queue until the server becomes free. The queue in M/M/1 is infinite and so all arrivals will be accepted but will be buffered or processed right away depending on whether the server is busy or free. We will discuss simulation details in next section.

To make it seed-independent, the simulation should be repeated on multiple seeds.

III Design thoughts

Step 1: Generate random numbers that are exponentially distributed for both customer service time and customer interarrival time. To do this, we need to assign different seeds for the $rand()$ function in Matlab. Note: the unit time is $'msec'$ in this simulation. The Matlab program is listed below:

```
rand('state',seed1); u1 = rand(NumberOfSamples,1);
rand('state',seed2); u2 = rand(NumberOfSamples,1);

x = -E_service*log(u1); % service time vector
tau = -E_interarr*log(u2); % inter-arrival time vector
```

Step 2: In order to get mean queue size and mean delay through the system, we need to obtain the parameters listed below:

- t: t(i) is the arrival time for ith customer
- s: s(i) is the departure time for ith customer
- w: w(i) is the waiting time in queue for ith customer
- T: T(i) is the system delay for ith customer
- N: N(i) is the number of customers in system (including server) at time i

We list the program to show how to get t,s,w and T:

```
t =  cumsum(tau); % the arrival time vector
s(1) = x(1)+tau(1); % the departure time vector
w(1) = 0; % the waiting time vector
for i=2:NumberOfSamples,
    s(i) = max(t(i),s(i-1)) + x(i);
    w(i) = max(s(i-1)-t(i),0);
end;
for i=1:NumberOfSamples,
    T(i) = w(i) + x(i); % the delay time vector
end;
```

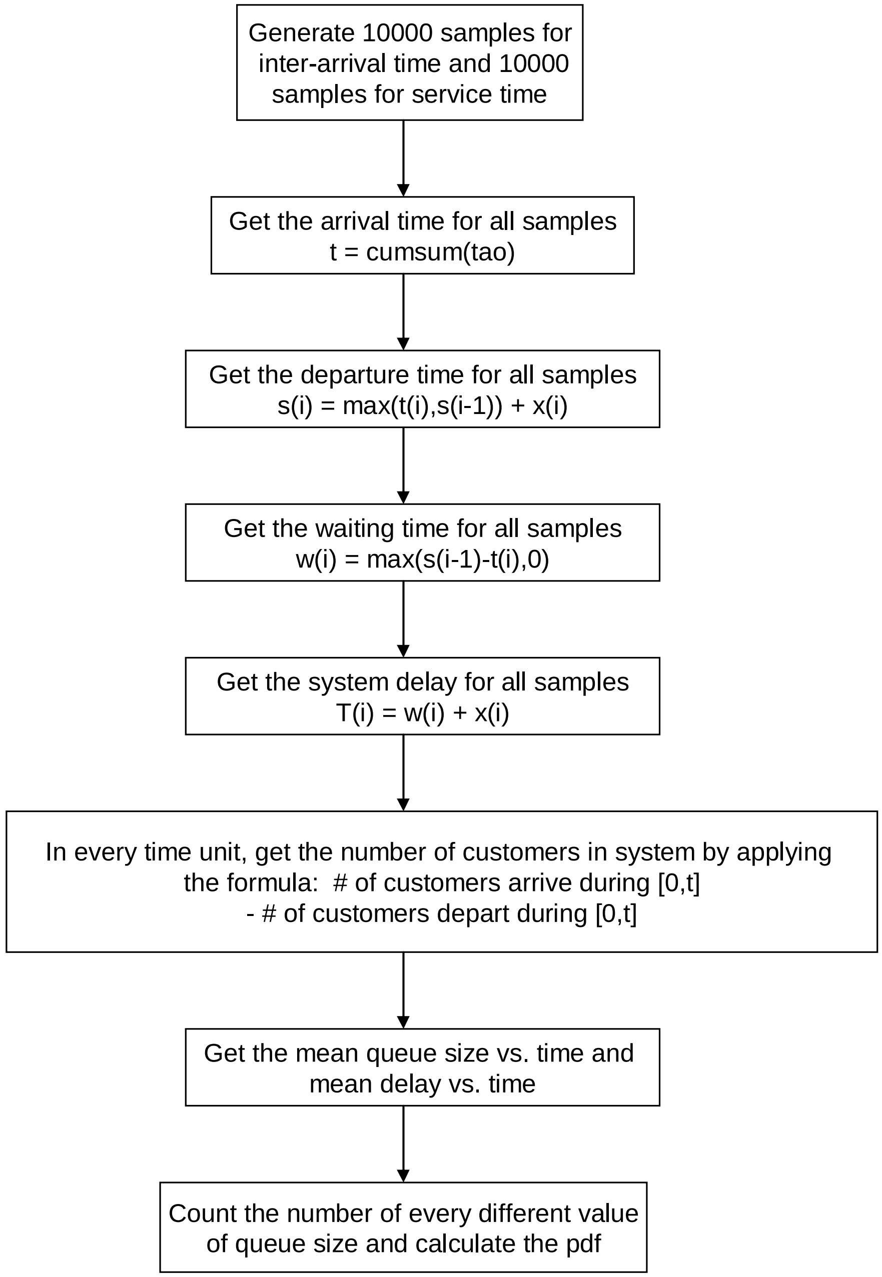
Step 3: After obtaining the vectors above, we proceed to find the queue size. Going through the arrival time vector t and departure time vector s, whenever an event occurs (an arrival or a departure), update the number of arrivals or departures in the corresponding time. Then we can get the number of packets in system by this equation:

$$N(t) = arrival\_num(t) - depart\_num(t) \tag{3}$$

Step 4: The expected queue size over time is found by using the following program:

```
E_n(1) = N(1); % average number of queue size over time
X1(1) = 1; total = N(1); % x axis is for time
for i=2:size(N),
    total = total+N(i);
    E_n(i) = total/i;
    X1(i) = i;
end;
```

Step 5: The simulation runs for 5 pairs of different seeds in order to make it seed-independent. So the 'meanvalues' function should be run for each ρ with 5 pairs different seeds and the final results should be averaged out.



VI Appendix: Flow chart

The flow chart of the simulation is shown below:

IV Program list

```
%%%%%%%%%% ECE 642 Project 2%%%%%%%%%%%%%%%%
%Description: Consider a high speed transmission link represented by an M/M/1
%model of single-server queue with Poisson arrival and exponential service
%time as depicted below. The link is assumed to have a capacity of 155 Mbps
%and the length of packets are exponentially distributed with mean packet
```

```
%size of 2325 bits. We want to simulate the system for the desired utilizations.
%Plot the queue size (including server), expected queue size, mean delay through
%the system vs. time for load=0.50. Also plot the probability density of the
%queue size for load=0.50. Compare the simulation results with analytical
%results. Now consider load=0.3,0.5,0.9 and load=0.95 and obtain delay vs.
%throughout of the system. In the same graph plot the analytical results.

%Assume the time unit is msec
NumberOfSamples = 10000;

%for rho=0.5, run simulation with 5 pair of different seeds
[N1,E_n1,E_T1,X11,pr1,X1,t1,delay11] =
meanvalues(0.5,8,7,NumberOfSamples);
[N2,E_n2,E_T2,X12,pr2,X2,t2,delay12] =
meanvalues(0.5,9,10,NumberOfSamples);
[N3,E_n3,E_T3,X13,pr3,X3,t3,delay13] =
meanvalues(0.5,15,16,NumberOfSamples);
[N4,E_n4,E_T4,X14,pr4,X4,t4,delay14] =
meanvalues(0.5,21,22,NumberOfSamples);
[N5,E_n5,E_T5,X15,pr5,X5,t5,delay15] =
meanvalues(0.5,3,4,NumberOfSamples);

%because the vectors may have different length, so we need to make them the same length
%before doing average on them
mini2=min([length(pr1),length(pr2),length(pr3),length(pr4),length(pr5)]);
pr1 = pr1(1:mini2);
pr2 = pr2(1:mini2);
pr3 = pr3(1:mini2);
pr4=pr4(1:mini2);
pr5 = pr5(1:mini2);
X1 = X1(1:mini2);

%average out the results from five simulations
pr = (pr1+pr2+pr3+pr4+pr5)/5;
Average_queue_size =mean([E_n1(length(E_n1)),E_n2(length(E_n2)),
E_n3(length(E_n3)),E_n4(length(E_n4)),E_n5(length(E_n5))])
Average_delay = mean([delay11,delay12,delay13,delay14,delay15])

%ploting.....
figure
plot(X11,N1);
xlabel('time(msec)');
ylabel('queue size (packets)');
title('The queue size(including server) for an M/M/1 model');

figure
plot(X11,E_n1);
```

```
xlabel('time(msec)');
ylabel('mean queue size (packets)');
title('The expected queue size for an M/M/1 model');

figure
plot(t1,E_T1);
xlabel('time(msec)');
ylabel('mean delay (msec)');
title('The expected delay through the system for an M/M/1 model');

figure
bar(X1,pr);
xlabel('queue size (packets)');
ylabel('Pr');
title('Probability density of the queue size for an M/M/1 model');

%for rho=0.3, run simulation with 5 pair of different seeds
[N,E_n,E_T,X1,pr,X,X_T,delay21]
=meanvalues(0.3,8,7,NumberOfSamples);
[N,E_n,E_T,X1,pr,X,X_T,delay22]
=meanvalues(0.3,9,10,NumberOfSamples);
[N,E_n,E_T,X1,pr,X,X_T,delay23]
=meanvalues(0.3,15,16,NumberOfSamples);
[N,E_n,E_T,X1,pr,X,X_T,delay24]
=meanvalues(0.3,21,22,NumberOfSamples);
[N,E_n,E_T,X1,pr,X,X_T,delay25]
=meanvalues(0.3,3,4,NumberOfSamples);

%for rho=0.9, run simulation with 5 pair of different seeds
[N,E_n,E_T,X1,pr,X,X_T,delay31]
=meanvalues(0.9,8,7,NumberOfSamples);
[N,E_n,E_T,X1,pr,X,X_T,delay32]
=meanvalues(0.9,9,10,NumberOfSamples);
[N,E_n,E_T,X1,pr,X,X_T,delay33]
=meanvalues(0.9,15,16,NumberOfSamples);
[N,E_n,E_T,X1,pr,X,X_T,delay34]
=meanvalues(0.9,21,22,NumberOfSamples);
[N,E_n,E_T,X1,pr,X,X_T,delay35]
=meanvalues(0.9,3,4,NumberOfSamples);

%for rho=0.95, run simulation with 5 pair of different seeds
[N,E_n,E_T,X1,pr,X,X_T,delay41]
=meanvalues(0.95,8,7,NumberOfSamples);
[N,E_n,E_T,X1,pr,X,X_T,delay42]
=meanvalues(0.95,9,10,NumberOfSamples);
[N,E_n,E_T,X1,pr,X,X_T,delay43]
=meanvalues(0.95,15,16,NumberOfSamples);
```

```
[N,E_n,E_T,X1,pr,X,X_T,delay44]
=meanvalues(0.95,21,22,NumberOfSamples);
[N,E_n,E_T,X1,pr,X,X_T,delay45]
=meanvalues(0.95,3,4,NumberOfSamples);

%average out the results from 5 simulations
delay(1)=(delay21+delay22+delay23+delay24+delay25)/5;
delay(2)=(delay11+delay12+delay13+delay14+delay25)/5;
delay(3)=(delay31+delay32+delay33+delay34+delay35)/5;
delay(4)=(delay41+delay42+delay43+delay44+delay45)/5;

%analytical results
delay_ana=[0.0214,0.03,0.15,0.3];
throughput_ana=[0.3,0.5,0.9,0.95];

%ploting......
figure
plot(throughput_ana,delay,'g+:',throughput_ana,delay_ana,'ro:');
xlabel('throughput(normalized)');
ylabel('mean delay (msec)');
legend('simulation results','analytical results');
title('Delay vs. throughput for rho=0.3,0.5,0.9,0.95');


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% N: the number of customers in system over time                               %%
%% E_n: the average number of customers in system over time                     %%
%% E_T: the average time delay in system over time                              %%
%% X1: the time axis for E_n,E_T and N                                          %%
%% mean_load: the network utilization                                           %%
%% mean_delay: the mean delay for packets                                       %%
%% pr: pdf                                                                      %%
%% X: the x axis for pr                                                         %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [N,E_n,E_T,X1,pr,X,t,mean_delay]
=meanvalues(rho,seed1,seed2,NumberOfSamples)

E_service = 2325/(155*10^3); % mean service time
E_interarr = E_service/rho; % mean interarrival time

%generate two vector of random numbers with uniformly distribution
rand('state',seed1);
u1 = rand(NumberOfSamples,1);
rand('state',seed2);
u2 = rand(NumberOfSamples,1);
```

```
x = -E_service*log(u1); % service time vector
tau = -E_interarr*log(u2); % inter-arrival time vector
t =  cumsum(tau); % the arrival time vector
s(1) = x(1)+tau(1); % the departure time vector
w(1) = 0; % the waiting time vector
for i=2:NumberOfSamples,
    s(i) = max(t(i),s(i-1)) + x(i);
    w(i) = max(s(i-1)-t(i),0);
end;
for i=1:NumberOfSamples,
    T(i) = w(i) + x(i); % the delay time vector
end;

% Get the queue size and average delay (event_driven)
arrival_num = zeros(1,5*NumberOfSamples);
depart_num = zeros(1,5*NumberOfSamples);
%arrival_num is a vector to record # of packets arrive in each time unit
arrival_num(ceil(t(1)))=1;
%depart_num is a vector to record # of packets depart in each time unit
depart_num(ceil(s(1)))=1;

for i=2:NumberOfSamples,
    arrival_num(ceil(t(i))) = arrival_num(ceil(t(i))) + 1;
    depart_num(ceil(s(i))) = depart_num(ceil(s(i))) + 1;
end;
arrival_num = arrival_num(1:ceil(t(NumberOfSamples)));
depart_num = depart_num(1:ceil(t(NumberOfSamples)));
arrival_num = cumsum(arrival_num);
depart_num = cumsum(depart_num);
N = arrival_num-depart_num;

for k=1:NumberOfSamples,
    E_T(k)=mean(T(1:k));
end;
mean_delay = E_T(NumberOfSamples);

%Get the expected queue size vs. time
E_n(1) = N(1); % average number of queue size over time
X1(1) = 1; % x axis is for time
total = N(1); for i=2:length(N),
    total = total+N(i);
    E_n(i) = total/i;
    X1(i) = i;
end;

%Get Pr (Probability of n customers in system during 1 msec)
for i=min(N):max(N),
```

```
    count = 0;
    for j=1:length(N), % count for each different queue size
        if(N(j) == i)
            count = count+1;
        end;
    end;
    % the state probability
    pr(i+1) = count/length(N);
    % x axis is for queue size
    X(i+1) = i;
end;
```

V Simulation results:

Queue size Vs Time:

The N vector contains the value for the queue size in each time unit which we get by counting the number of customers at the end of every msec (discussed in section III step 3). The $X1$ vector contains the time. A graph is then plotted between N and $X1$ which is the plot for the queue size(including the server) Vs time.
This graph is shown in Figure 1.

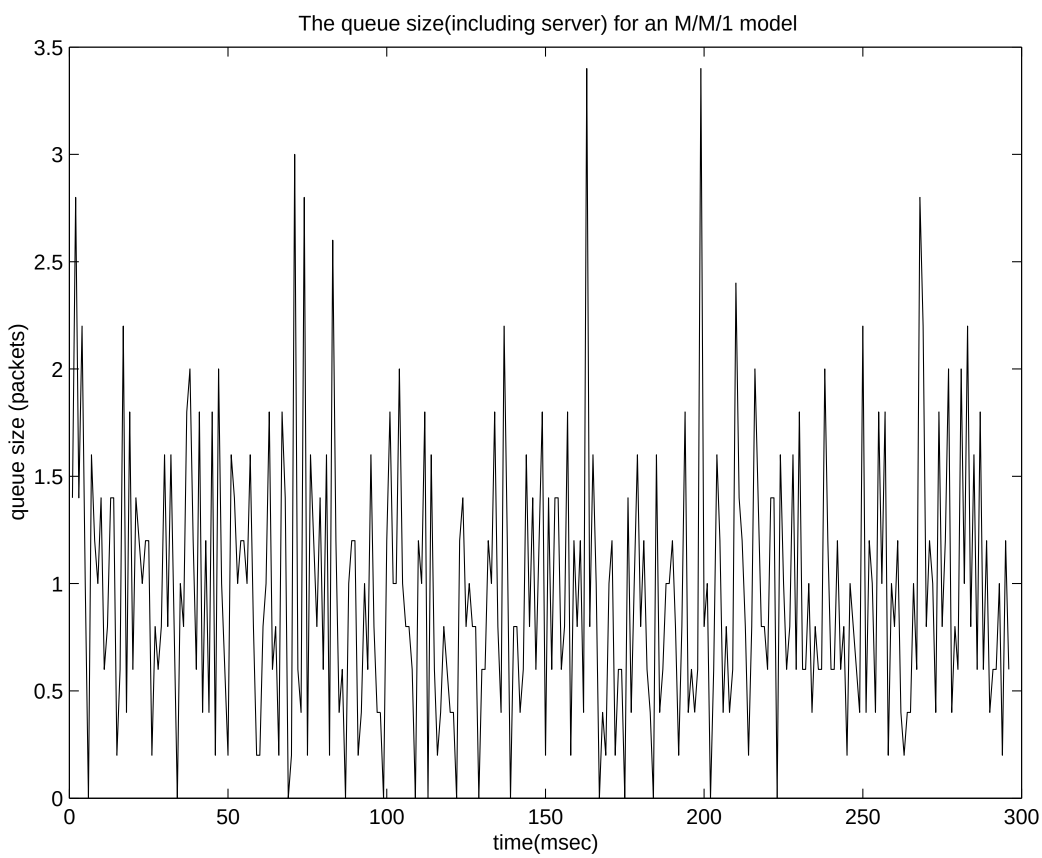


Figure 1: The queue size for an Infinite M/M/1 model $NumberOfSamples = 50000$

Expected Queue size Vs Time:

The vector of expected queue size is the mean of the number of customers in the queue for each individual time unit. For every msec, we find out the expected queue size and plot it against the time unit. Analytically, we can calculate expected queue size as,

$ExpectedQueuesize = \rho/(1-\rho)$

Hence, for $\rho = 0.5$, the $Expectedqueuesize = 1$. If compared with the simulated value, the mean of the graph is close to the analytical expected queue size. A graph is then plotted between E_n and $X1$ which is the plot for the mean queue size(including the server) Vs time.
This graph is shown in Figure 2.

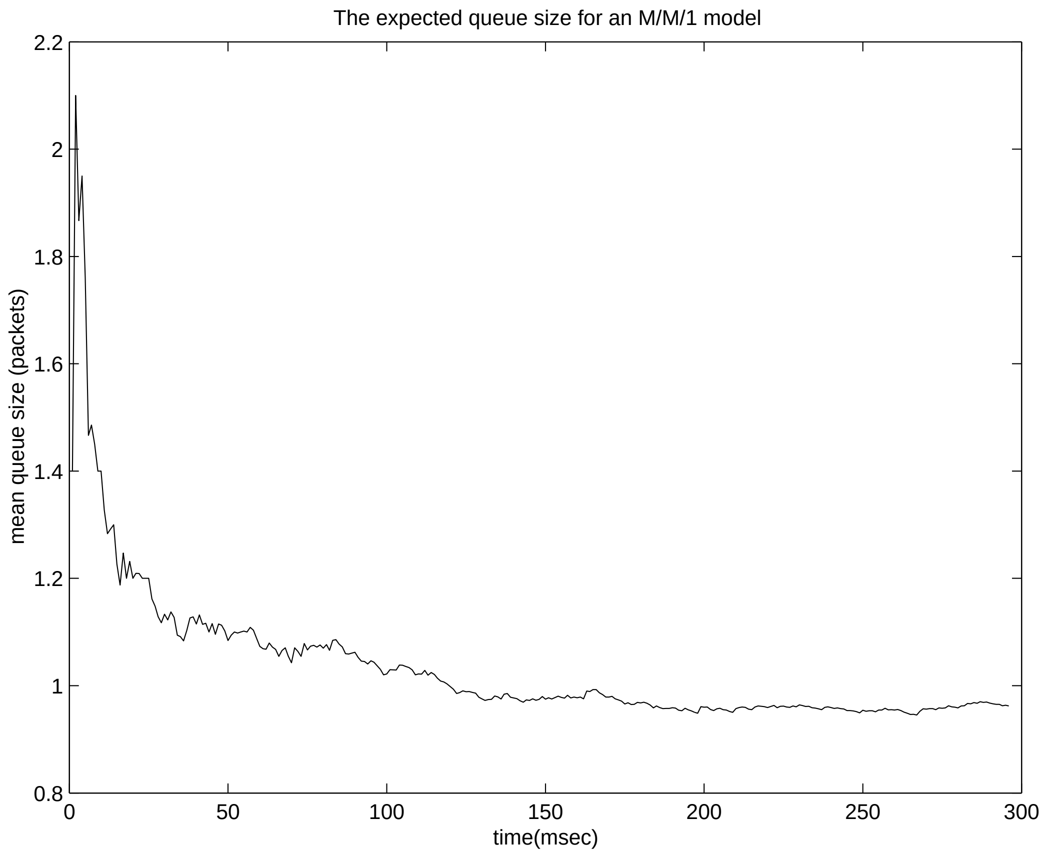


Figure 2: The expected queue size for an Infinite M/M/1 model $NumberOfSamples = 50000$

Mean delay Vs Time:

Mean delay is $average\_queuesize/arrivalrate$. For every time unit (msec), we find out the average delay by adding delay of each customer in the system during that particular time unit and divided by the number of customers. Analytically, we can calculate mean delay as,

$Meandelay = 1/(\mu * (1-\rho))$

Hence, for $\rho = 0.5$, the $MeanDelay = 3.0000e-002$ msec. If compared with the simulated value, the mean of the graph is close to the analytical mean delay obtained as time increased.
This graph is shown in Figure 3.

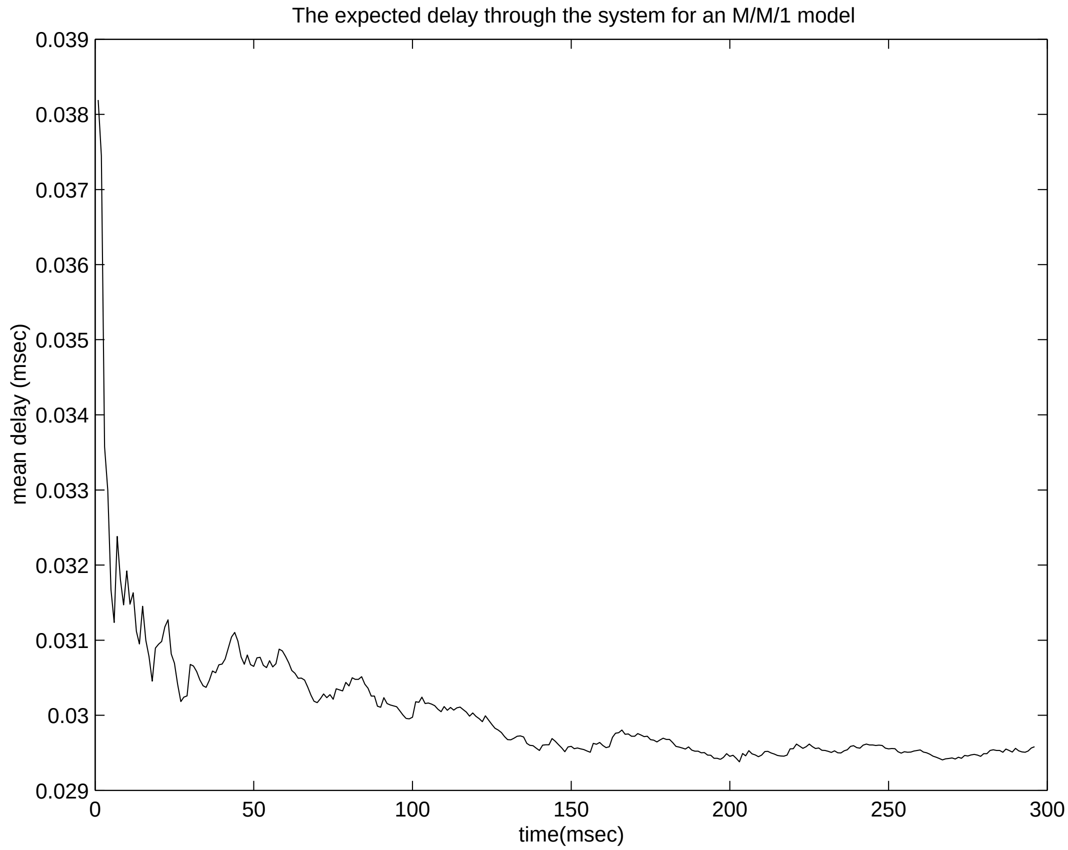


Figure 3: The expected delay through system for an Infinite M/M/1 model $NumberOfSamples = 50000$

Probability density of the queue size:

The probability density of the queue size is the state probability of the queue. For $\rho = 0.5$, the analytical result can get by applying the formula $p_n = \rho^n(1-\rho)$, that is:

$$p_0 = 0.5$$

$$p_1 = 0.25$$

$$p_2 = 0.125$$

$$p_3 = 0.0625$$

$$p_4 \approx 0.0313$$

$$p_5 \approx 0.0156$$

$$p_6 \approx 0.0078$$

$$p_7 \approx 0.0039$$

This graph is shown in Figure 4.

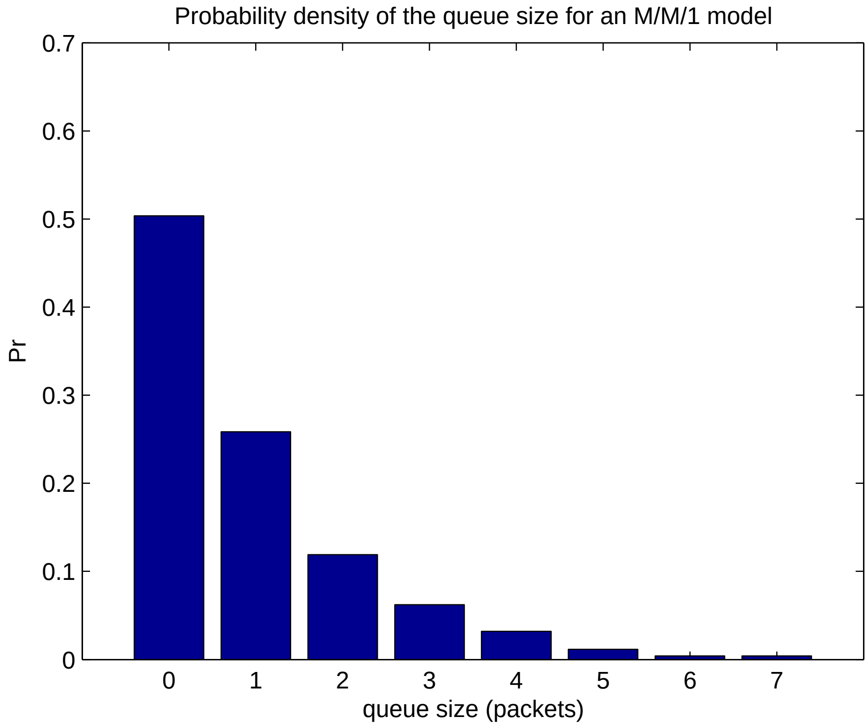


Figure 4: The state probability density of queue size for an Infinite M/M/1 model $NumberOfSamples = 50000$

Delay vs Throughput:

We know that throughput is equal to rho. Hence we plot the graph between rho and delay. The delay is found by the formula $Delay = 1/(\mu - \lambda)$. This is found for $rho = 0.3, 0.5, 0.9, 0.95$.
This graph is shown in Figure 5.

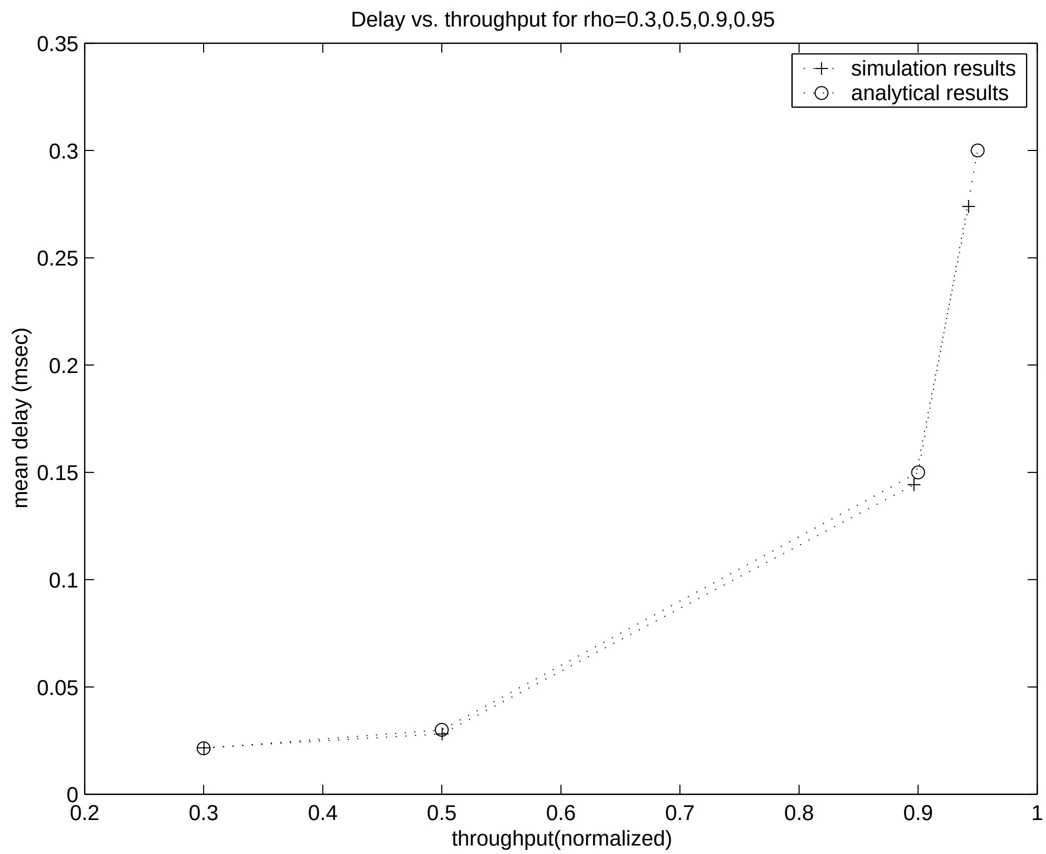


Figure 5: Delay over throughput for an Infinite M/M/1 model $NumberOfSamples = 50000$