ECE 6101

Computer Communication Network

Class Project 1 Report

Instructor : Ness B. Shroff

Reported by :Siyang Li

li.9868

Introduction

Learn how to generate exponential random variable, Poisson random variable and Erlang random variable from uniform variable. And simulate M/M1 , M/Ek/1 and M/D/1 queueing system and calculate, analyze and solve different related problems.

Problem

1.

(a) The function called uniform is to generate uniform random variable U[0,1]. There are 3 parameters :

```
def uniform(start,end,size):
    x = np.random.uniform(start,end,size)
    return x
```

start → according to this problem, it should be 0.

end → according to this problem, it should be 1.

size → input how many values you need to generate.

(b) The function called cumsum is to estimate and plot P(U>x). And I choose 5000 values between [0,1].The figure is below:

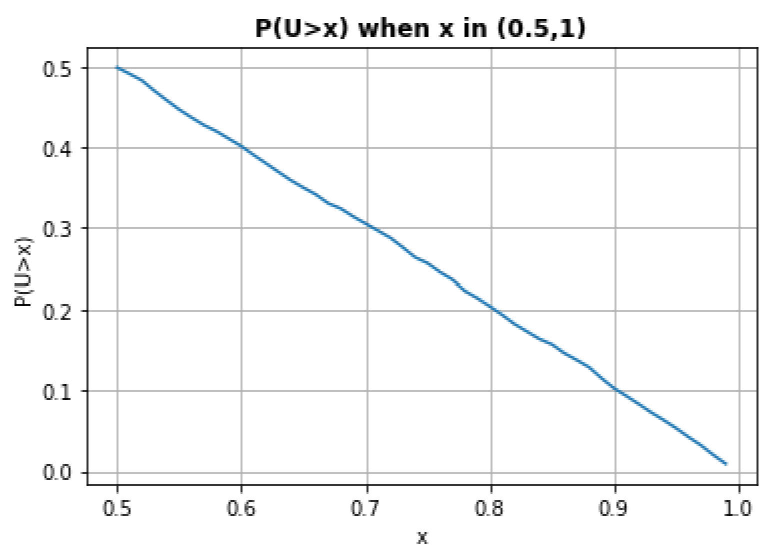

Figure 1.b P（U>x）when x in (0.5,1)

2.

(a) I use inverse distribution function to generate exponential random variables.Since for exponential distribution

$F(x) = 1 - e^{-\lambda * \mathrm{x}}$,

so when we calculate its inverse distribution function

$X = (-\lambda) * \log(1 - Y)$.

The function called ExpRand is to generate Exponential random variables.And the figure below shows 5000 values of Exponential random variables:

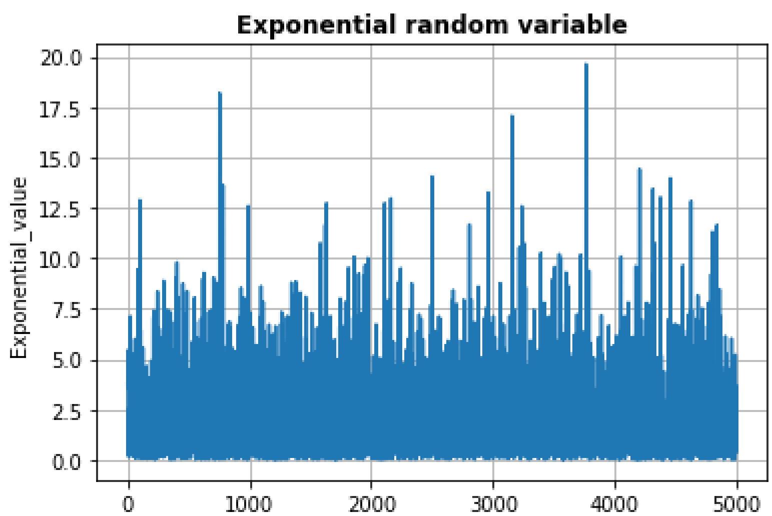

Figure 2.a.1 5000 Exponential random variables

Use following algorithm to generate Poisson random variable:

```
algorithm poisson random number (Knuth):
    init:
         Let L ← e^-λ, k ← 0 and p ← 1.
    do:
         k ← k + 1.
         Generate uniform random number u in
[0,1] and let p ← p×u.
    while p > L.
```

```
    return k - 1.
```

The function called Poisson is to generate Poisson random variables.And the figure below shows 500 values of Poisson random variables:

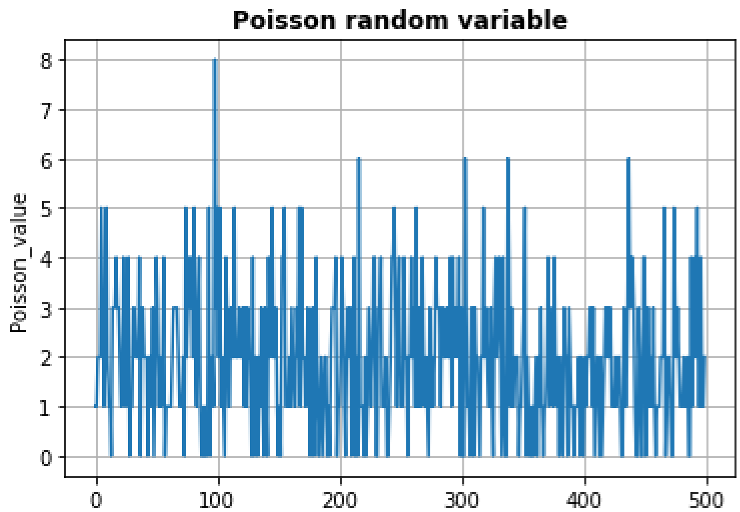

Figure 2.a.2 500 Exponential random variables

(b) The function ExpPlot is to plot the $P(X > x)$for exponential random variables. I choose range from 0 to 15 and step is 0.1, the figure is like below:

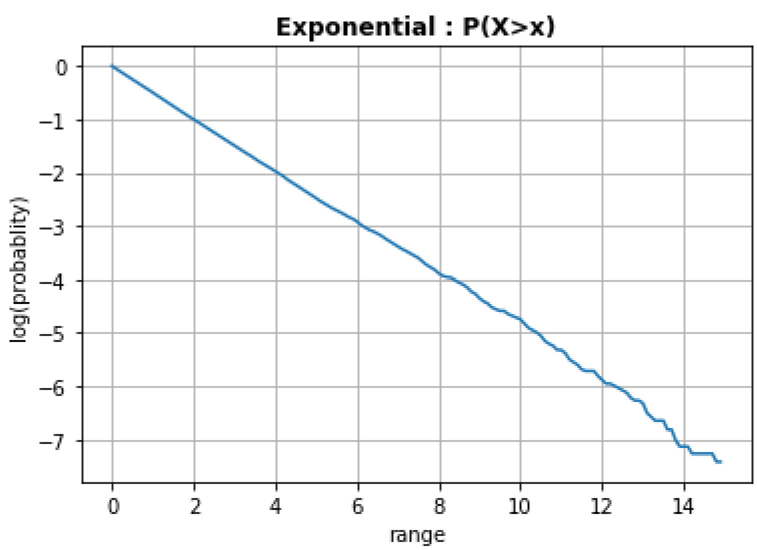

Figure 2.b.1 P(X>x) for exponential random variables

The function Poissonp is to plot the $P(Y > x)$ for Poisson random variables. I choose range from 0 to 15 and step is 0.1, the figure is like below:

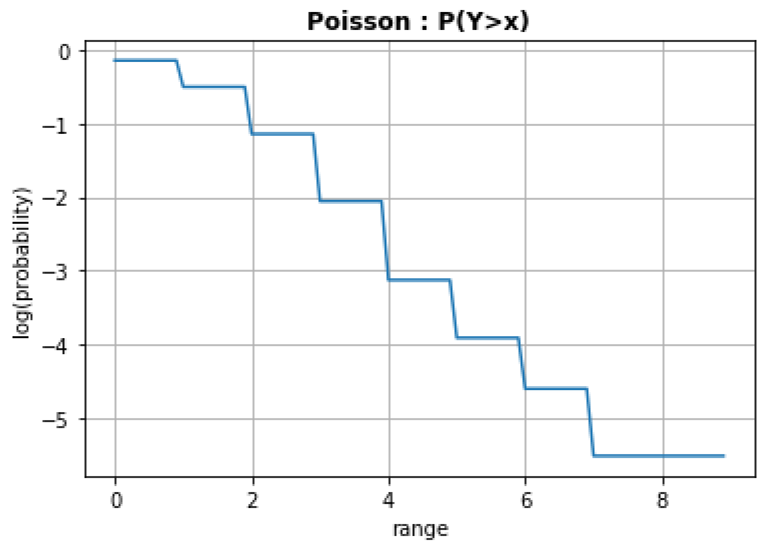

Figure 2.b.2 500 Exponential random variables

Note: Because the Possion random variables can only be integers. So when the probability reaches like 5.1 / 5.21, the probability will still hold on 5 until the probability reaches 6.

3.

(a) The idea I came up with for the simulation of M/M/1 system is like below:

①Generate two lists of exponential random variables *InterArrivalT* of 9999 and *ServiceT* of 10000 as the Inter-arrival distribution and Service Time distribution.

②There are two cases for the procedure of M/M/1 system:

1st : when (i+1)st packet arrive, the ith packet has departured. In this case, the departure time of (i+1) = arrival time of (i+1) + service time of (i+1).

2nd: when (i+1)st packet arrive, the ith packet has not departured. In this case, the departure time of (i+1) = depature time of i + service time of (i+1).

Thus we could get two sequence, arrival-time sequence and departure-time sequence.

③ I add an attribute *state* to record its state. If one packet arrives, the *state +1* ,else if one packet departures, the *state -1.*And I combine both of the arrival-time and departure-time sequence together to generate a new sequence *Timeline* and sort it*,*and this sequence record all the time point and state of all arrival and departure time of all packets.The *Timeline* sequence should be like this:

```
[[0, 1],
 [0.0653164159768464, 2],
 [0.285337906476816, 2],
 [0.3280934038301168, 3],
 [0.39269585768269016, 3],
 [0.440118105977953, 4],
 [0.6448055095783987, 4],
 [0.8242676379877061, 4],
 [1.16518924555565, 3],
 [1.1732063987219663, 3],
 [1.433050332239987, 3],
 [1.5569761691845645, 3],
 [1.6847597694231757, 3],
 [1.8987020873723635, 3],
 [2.0031718106692917, 3],
 [2.0055484766615495, 4],
 [2.0224699195629317, 5],
 [2.5024993750159856, 3],
 [2.5641154465245135, 4],
```

Figure 3.a.1 Timeline Sequence Sample

(b) The idea for this problem is that when we need to calculate P(0), we should use the time of state next to the time of state '0' – the time of state '0' and sum this time period together and divides it by the total time. And then we get P(0) and so on. The figure is like below:

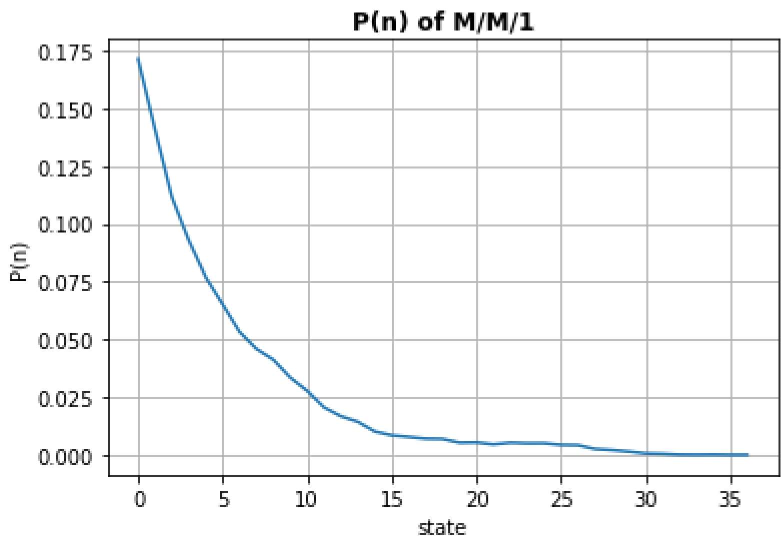

Figure 3.b P(n) of M/M/1 System

(c) When I calculate the P(n) , I calculate the EN the same time.And then use Little's Law to calculate the expected delay when ρ=5/6

EN = 5.09464097963829

Expected Delay = 1.018928195927658

4 .

(a) The function erlang is to generate erlang random variables. I use the algorithm below:

$$E(k, \lambda) \approx -\frac{1}{\lambda} \ln \prod_{i=1}^{k} U_i$$

The figure below is 10000 erlang random variables:

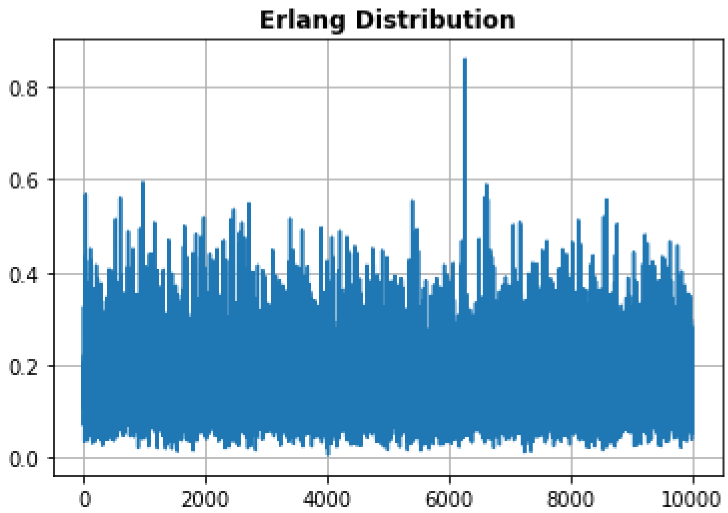

Figure 4.a 10000 Erlang random variables

And I change the service time distribution from exponential to erlang and get the M/EK/1 queuing system.

(b) When k=4 , ρ=5/6 , overall service rate is 6 and arrival rate is 5. Because mean of erlang is k/λ. So 4/λ = 1/6→λ=24.So the P(n) of M/Ek/1 in this case is as below:

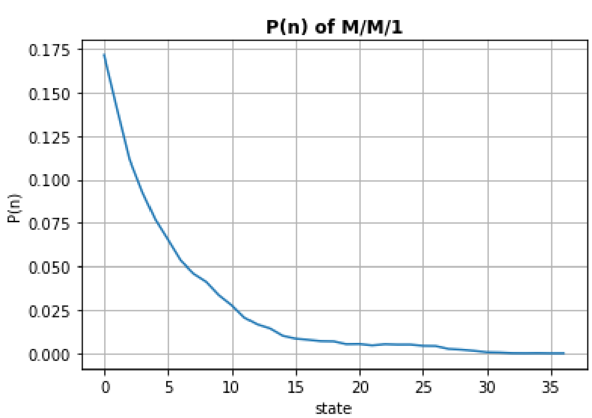

Figure 4.b.1 P(n) of M/M/1

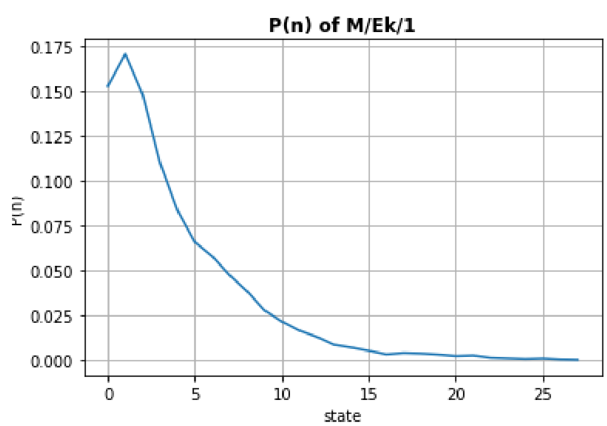

Figure 4.b.2 p(n) of M/Ek/1

M/M/1 : **E(N)** = 5.09464097963829

M/Ek/1 : **E(N)** = 4.0183276966658354

The E(N) of M/M/1 is greater than M/Ek/1 with the same ρ=5/6. The trend of P(n) of M/M/1 reaches its peak at x=0 goes down to 0 as the number of state increases .And the the trend of P(n) of M/Ek/1 is going up before x=1 and reaches its peak at x=1 and goes down to 0 as the P(n) of M/M/1 after x=1 .

(c)

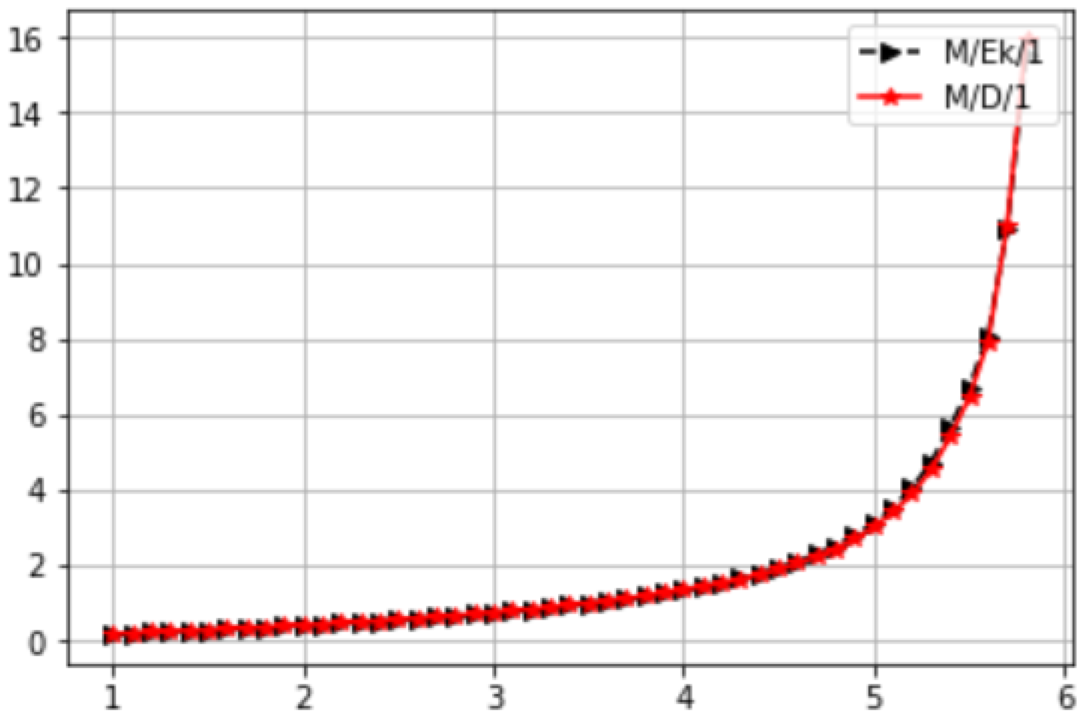

Figure 4.c E(N) of M/Ek/1 and M/D/1

The line of M/Ek/1 and M/D/1 nearly completely overlaps.

According to M/G/1 Mean Queue Length formula :

$$E(N) = \frac{\rho}{1-\rho}\left[1 - \frac{\rho}{2}\left(1 - \mu^2\delta^2\right)\right]$$

For M/D/1 , $\boldsymbol{\delta^2}$ = 0.

For M/Ek/1 , $\boldsymbol{\delta^2} = \frac{\boldsymbol{k}}{\lambda^2} = \frac{\mathbf{40}}{\mathbf{240^2}}$ It is a small value.

So the E(N) of M/Ek/1 should be larger than M/D/1.

Thus the as λ increases, the line of M/Ek/1 should be higher than M/D/1 but the difference of M/Ek/1 and M/D/1 is small. M/D/1 system provides a lower bound on congestion for a system with Poisson arrivals.

Conclusion

From this project, I have a better understanding of the relation and difference between some common random variables and how these different variables build up different queueing system. What's more, based on the simulation procedure and result, I also get a clear view of what are the properties and features between different queueing system, like PN between M/M/1 and M/Ek/1 with the same $\boldsymbol{\rho}$ and the trend between M/Ek/1 and M/D/1 with arrival rate increases.

Reference

1. https://en.wikipedia.org/wiki/Exponential_distribution
2. https://en.wikipedia.org/wiki/Poisson_distribution
3. https://en.wikipedia.org/wiki/Erlang_distribution