CARNEGIE MELLON UNIVERSITY
APPLIED STOCHASTIC PROCESSES
(COURSE 18-751)
HOMEWORK 3

Daniel Marew

September 17, 2017

I collaborated with :

Nebyou Yismaw
Daniel Nkemelu
Agatha Niwomugizi

Q1

(a)

(b)

(c)

Q.2 Generalized CLT

(a) Show GCLT reduces to the usual i.i.d version of CLT

$$Yn = (X_1 + X_2 + \cdots + X_n - m_n)/Sn$$

where X_k is an independent random variable with $E[X_k] = \mu_k$ and $Var[X_k] = \sigma_k^2$, $m_n = (\mu_1 + \mu_2 + \cdots + \mu_n)/s_n$ and $s_n^2 = \sigma_1^2 + \sigma_2^2 + \cdots + \sigma_n^2$

since i.i.ds have identical distribution the have the same mean and variance
i.e $\mu_1 = \mu_2 = \cdots = \mu_n = \mu$ and $\sigma_1^2 = \sigma_2^2 = \cdots = \sigma_n^2 = \sigma^2$

$$m_n = (\mu_1 + \mu_2 + \cdots + \mu_n) = n\mu$$
$$s_n^2 = (\sigma_1^2 + \sigma_2^2 + \cdots + \sigma_n^2) = n\sigma^2$$
$$Yn = \frac{(X_1 + X_2 + \cdots + X_n - n\mu)}{\sqrt{n\sigma^2}}$$
$$Yn = \frac{(X_1 + X_2 + \cdots + X_n - n\mu)}{\sigma\sqrt{n}}$$

hence , GCLT reduces to CLT when $\{X_k\}$ are i.i.ds.

(b)

(c)

Average Number of packets in Buffer	3.36
Fraction of time the buffer is empty	0.33
The fraction of packet Arrivals that are blocked	0.0040

Table 1: Question 3b Answer

Q3 Buffers

(a) for $\lambda = 0.1$, $\mu = 0.12$, $BufferSize = 10$ and $NumberOfSteps = 1000$ we get table?? and figure ??

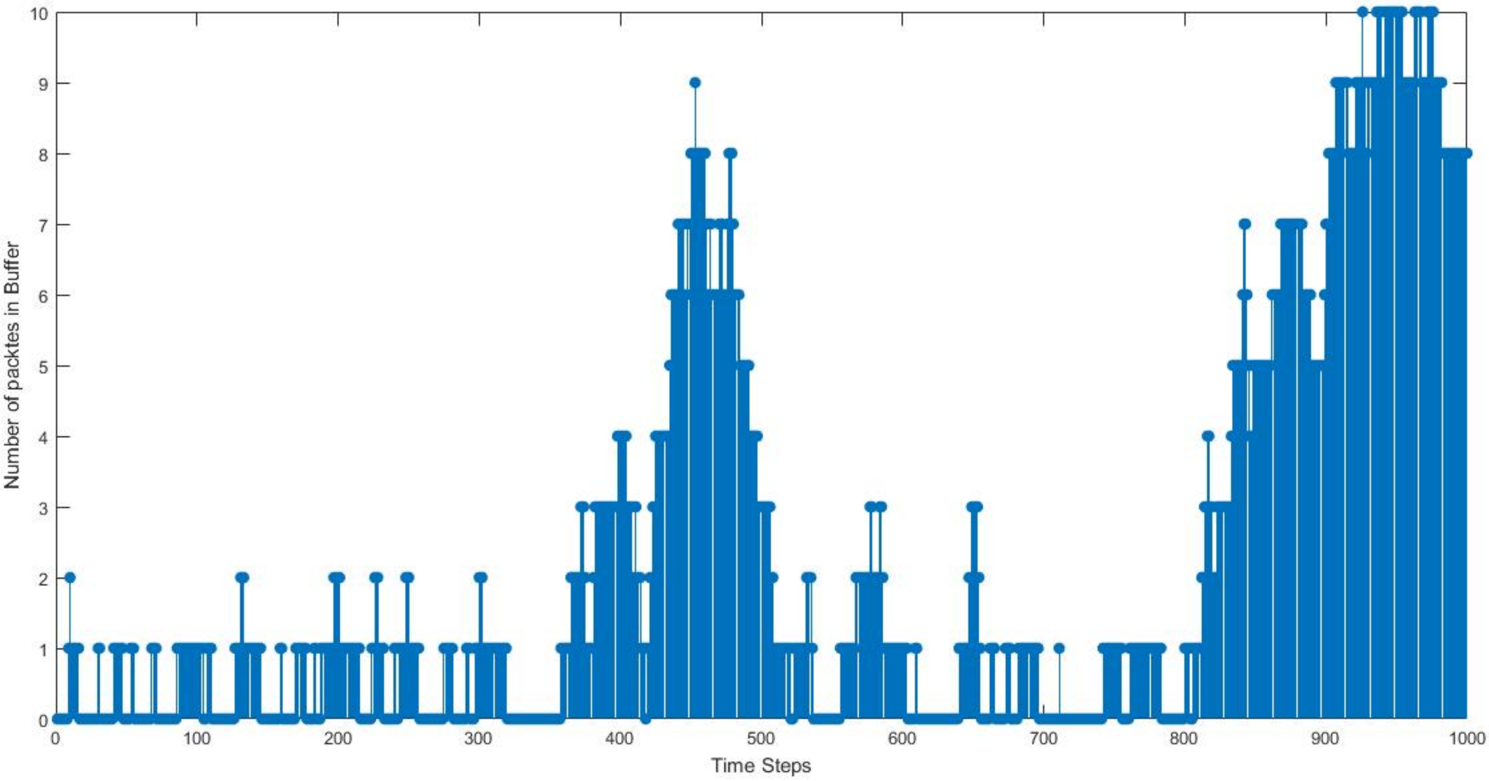

Figure 1: Number of packets in the buffer Vs Time steps

Average Number of packets in Buffer	6.87
Fraction of time the buffer is empty	0.03
The fraction of packet Arrivals that are blocked	0.2100

Table 2: Question 3 Answer

(b) for $\lambda = 0.1$**,** $\mu = 0.01$**,** $BufferSize = 10$ **and** $NumberOfSteps = 100$ **we get table?? and figure ??**

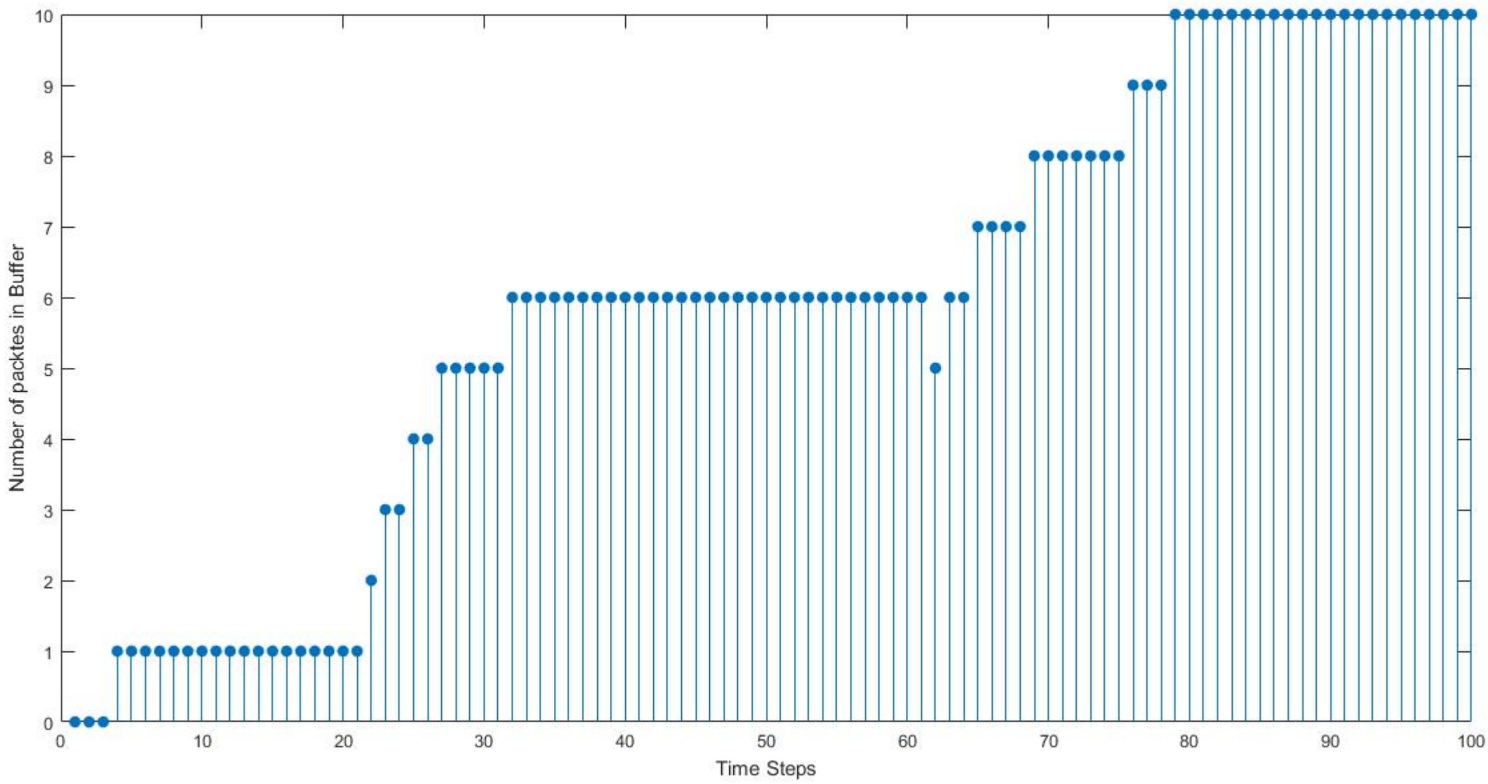

Figure 2: Number of packets in the buffer Vs Time steps

(c) Littel Law

Little Law is given by the following formula [**?**]

$$L = \lambda T \tag{1}$$

Where L is the average backlog(the average number of packets in the buffer) , T the delay in the system and λ is the average arrival rate.

hence to get the average delay in the system,

$$T = \frac{L}{\lambda} \tag{2}$$

for (a)

$$T = \frac{L}{\lambda} = \frac{3.36}{0.1} = 33.6$$

for (b)

$$T = \frac{L}{\lambda} = \frac{6.87}{0.1} = 68.7$$

Code Appendix

3. a

```
function P = get_stochastic_matrix(buffer_size,lamda,mu)
    P = zeros(buffer_size+2,buffer_size+2);
    a = lamda*(1-mu);
    b = mu*(1-lamda);
    c = 1-(a+b);
    P(1,1) = 1-a;
    P(1,2) = a;
    P(buffer_size+2,buffer_size+2) = 1-mu;
    P(buffer_size+2,buffer_size+1) = lamda*mu;
    P(buffer_size+2,buffer_size)   = b;
    for i=2:buffer_size+1
        P(i,i) = c;
        P(i,i+1) = a;
        P(i,i-1) = b;
    end

end
```

```
DEBUG = 0;
N = 100;%time steps
State0 = 1;
lamda = 0.1;
mu = 0.001:0.001:0.01; % 10% mu
%mu = 0.02:0.01:0.2;      % 1% mu
low_load_mu = 0;
BUFFER_SIZE = 10;
percentage = 10;
MU_FOUND = 0;
for i=1:length(mu)
    P = get_stochastic_matrix(BUFFER_SIZE,lamda,mu(i));
    StateTrans = simMC(N,State0,P);
    lost_packets = mean(StateTrans==(BUFFER_SIZE+2))*100;%loss of packets
    if DEBUG
    fprintf('mu %0.4f buffer size %i lost packets %4.4f percent\n',mu(i),BUFFER_SIZE,lost_packets);
    end
    if lost_packets>percentage
    %if lost_packets<percentage
        low_load_mu = mu(i);
        MU_FOUND = 1;
        if DEBUG
            fprintf('mu %0.3f satisfies loss value of %i percent with packet loss of %4.4f\n',...
            mu(i),percentage,lost_packets);
        end

    end
end
if MU_FOUND
```

```
    % Some stat before modifying StateTrans

    StateTrans(find(StateTrans==(BUFFER_SIZE+2)))=BUFFER_SIZE+1; % dropped == full
    Avg_Number_Of_Packets    = mean(StateTrans);
    Fraction_Of_Time_BEmpty = mean(StateTrans==1);
    Fraction_Of_Time_BBlocked = lost_packets/100;
    result = fopen('result_b.txt','w');
    fprintf(result,' Average Number of packets in Buffer: %2.2f \n Fraction of time the buffer is empty: %2.2f\n',...
        Avg_Number_Of_Packets,Fraction_Of_Time_BEmpty);
    fprintf(result,' The fraction of packet Arrivals that are blocked %2.4f\n',Fraction_Of_Time_BBlocked);
    fprintf(result,'\n MU:%2.2f\n Buffer Size: %i\n Lamda:%1.2f\n Number of Steps:%i\n Packet Loss:%2.2f percent\n',...
        low_load_mu,BUFFER_SIZE,lamda,N,lost_packets);
        StateTrans = StateTrans-1;% get rid of the bias so that it starts at state 0
    fclose(result);
    stem(1:N,StateTrans,'filled');
    xlabel('Time Steps');
    ylim([0 10]);
    ylabel('Number of packtes in Buffer');
else
    fprintf('appropriate mu not found try again!!\n')
end
```

```
function X = simMC(M,A,P)
X = zeros(1,M);
X(1) = A;
for m=1:M-1
       X(m+1) = discrete(P(X(m),:));
end
end
```

```
function T = discrete(P)
Pnorm = [0 P]/sum(P);
Pcum  = cumsum(Pnorm);
R = rand(1);
[~,T] = histc(R,Pcum);
end
```

References

[1] Jean Walrand. *Probability in Electrical Engineering and Computer science*. Jean Walrand, 2014.