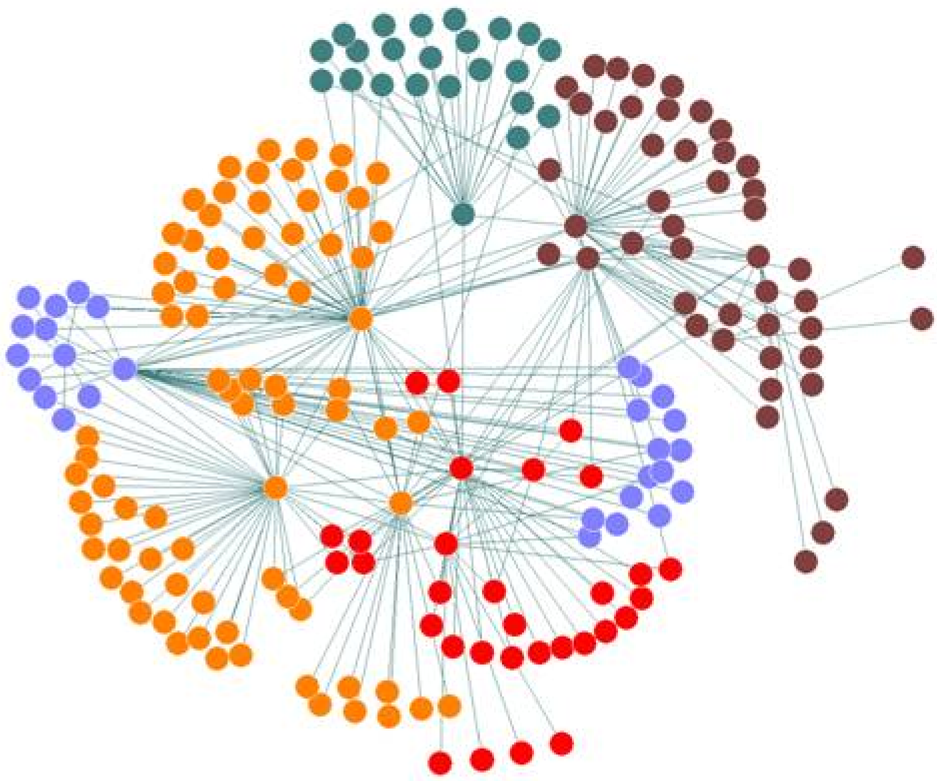
Escola Tècnica Superior
d’**Enginyeria de Telecomunicació de Barcelona**

Laboratory Session 2.

Network Performance Analysis and Evaluation



Erik Solé & Gokarna Dumre
Lluís Llopis

GRUP 51
2022-23

Departament d'Enginyeria Telemàtica
UNIVERSITAT POLITÈCNICA DE CATALUNYA

Index

1. Objectives

The objectives of this session may be summarized in the following points:

- Consolidate the theoretical knowledge, previously learnt by the students, about transmission systems modeled as M/M/1.
- Become familiar with the simulation environment provided by Scalev Lite.
  - Single simulations.
  - Result files.
- Analysis of the results with MATLAB.

2. Previous work.

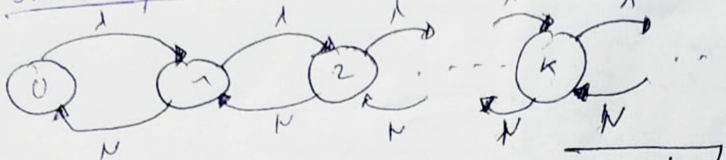
Do the following previous work:

a) Read the Scalev Lite user guide

b) Obtain, for the M/M/1 queue system:

Erik & Gokarna — AAX: Estudi previ 2 — 28/09/2022

- Obtain for M/M/1 queue system
  - State probabilities



$$\boxed{P_k = P_0 \prod_{i=0}^{k-1} \frac{\lambda_i}{\mu_{i+1}} = P_0 \prod_{i=0}^{k-1} \frac{\lambda}{\mu} = P_0 \left(\frac{\lambda}{\mu}\right)^k}$$

$$\boxed{P_0 = \frac{1}{1+\sum_{k=1}^{\infty} \left(\frac{\lambda}{\mu}\right)^k} = \frac{1}{\sum_{k=0}^{\infty} \left(\frac{\lambda}{\mu}\right)^k} = \frac{1}{\frac{1-0}{1-\rho}} = 1-\frac{\lambda}{\mu}}$$

$$\boxed{P_k = \left(1-\frac{\lambda}{\mu}\right)\left(\frac{\lambda}{\mu}\right)^k \xrightarrow[\frac{\lambda}{\mu}=\rho]{} P_k = (1-\rho)\rho^k}$$

- Average number of client in queue and in server

$$\boxed{N_{queue}} = \sum_{k=0}^{\infty} k \cdot P_k = \sum_{k=0}^{\infty} k \cdot (1-\rho)\rho^k = \frac{\rho}{1-\rho}$$

$$\boxed{N_{server}} = \lambda \cdot T_s = \lambda \cdot \frac{1}{\mu_s} = \rho$$

- Transmission, waiting & transfer times

$$\left.\begin{aligned} &\cdot f_{ts} = \mu e^{-\mu t} \\ &\cdot f_{tw} = (1-\rho)\delta(t) + \frac{\rho}{T} e^{-\frac{t}{T}} \\ &\cdot f_t = \mu(1-\rho) e^{-\mu(1-\rho)t} \end{aligned}\right\} t \geq 0$$

3. M/M/1 System Simulation

Using the Scalev Lite simulator, do the following simulation:

- Name: MM1
- Length: 5000000.
- Type: Single simulation.
- Scheduler: FCFS.
- Traffic sources: 1.
- Categories: 1.
- Traffic 1:
- Interarrival time: exponential, 0.2 s.
- Packet length: exponential, 120 bits.
- Category 1 queue size: infinite.
- 1 server (SS, Single Server)
- Channel capacity: 1200 bps.

3.1 Average values analysis

a) Check the values provided by the simulator:

- Traffic source 1 and Category 1.
    - Channel utilization: • Transmission time: **0.10069302966237427**
    - Transmission time:**0.09898531160608105**
    - Transfer time: **0.20654111625365604**
    - Waiting time: **0.10755580464757498**
    - Loss probability: **0.0**
    - Served packets**:20742**

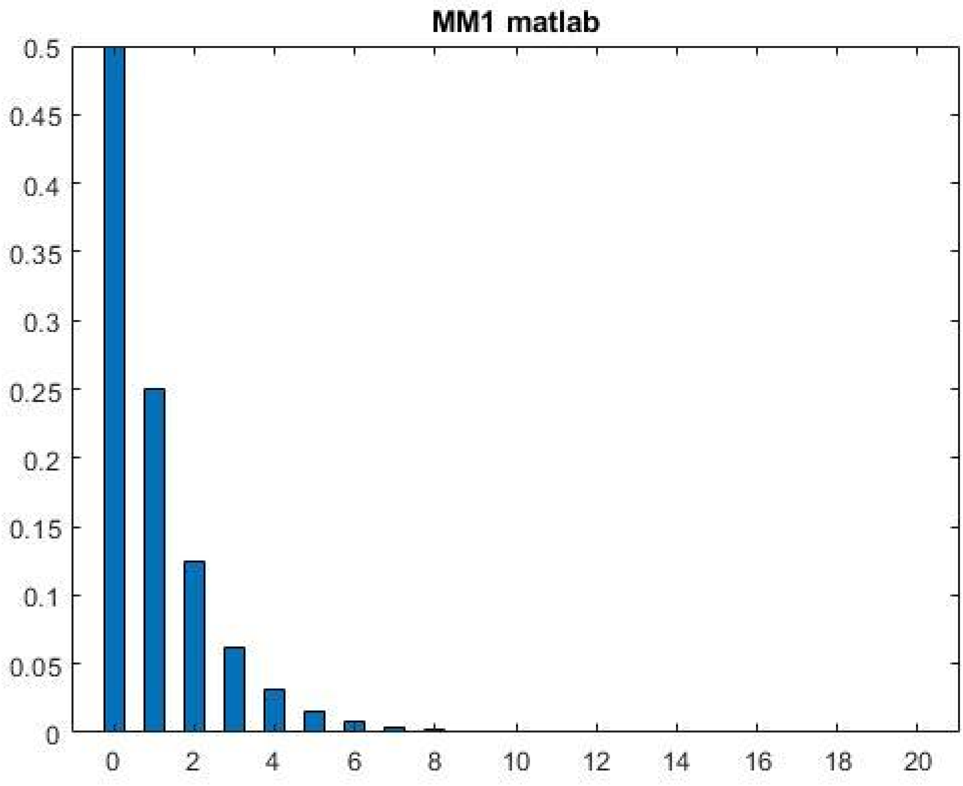
- Category 1
    - Channel utilization. **0.49275630724369274**
    - Transmission time: **0.09898531160608105**
    - Transfer time: **0.20654111625365604**
    - Waiting time: **0.10755580464757498**
    - Number of packets in queue: **0.5355114644885366**
    - Number of packets in server: **0.49279510720489367**
    - Loss probability: **0.0**

b) Open and analyze the MATLAB script mm1.m. Next, run it using the same values as in the simulation. Check the theoretical mean values and compare them with the simulation results.

Arrival rate (packets/s): **1/0.2=5**
Packet length: **120**
Server capacity: **1200**

RESULTS FOR THE M/M/1 QUEUE

Utilization factor:**0.5**
Waiting time:**0.1**
Service time:**0.1**
Transfer time:**0.2**
Number of packets in queue:**0.5**
Number of packets in server:**0.5**
Number of packets in system:**1**



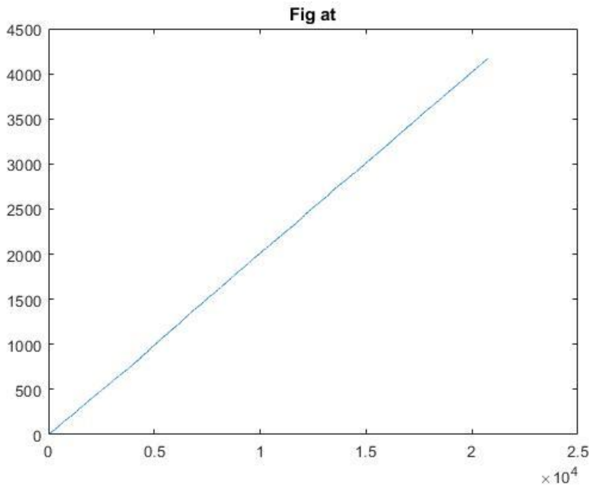
Comparing the both datas, Scalev data and matlab data. We can conclude, the datas are identical but more precise in Scalev with more decimals

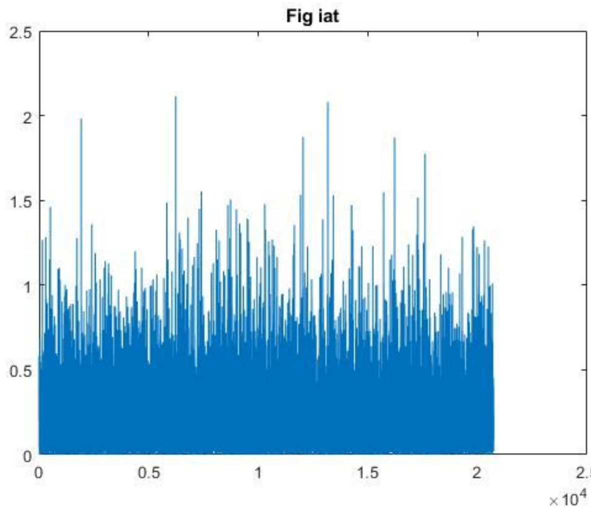
3.2 Arrival process analysis

In this section, the simulated arrival process will be checked and compared with the theoretical.

a) Execute in MATLAB the following commands, paying attention and understanding the results.

```
load output_MM1_source_1.txt;
at=output_MM1_source_1(:,2);
plot(at)
iat=diff(at);
plot(iat)
```





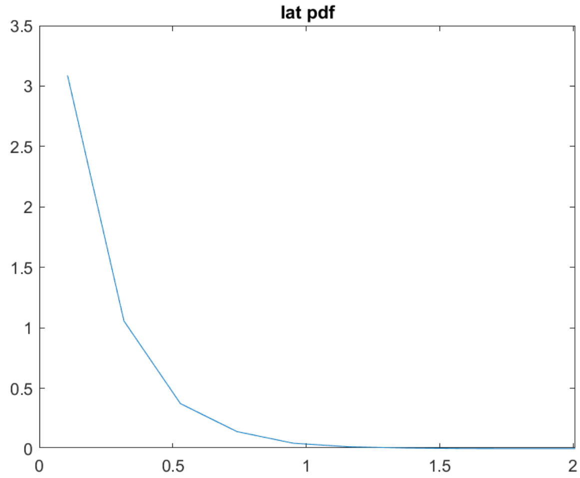
b) What does the vector iat mean? For the random variable iat compute the maximum, the minimum, the mean value, the variance, the standard deviation and the coefficient of variation.

```
max(iat)=2.1142
min(iat)=0
mean(iat)= 0.200857480352924
var(iat)=0.0409594730922919
std(var)= 0.2024
Coef=1,007679674164985   %std(timeService)/mean(timeService)
```

The vector iat is the difference between the current value (at()) and the previous, of all the samples. Doing this we obtain the random variable of interarrival time.

c) Obtain and plot the pdf of the vector iat

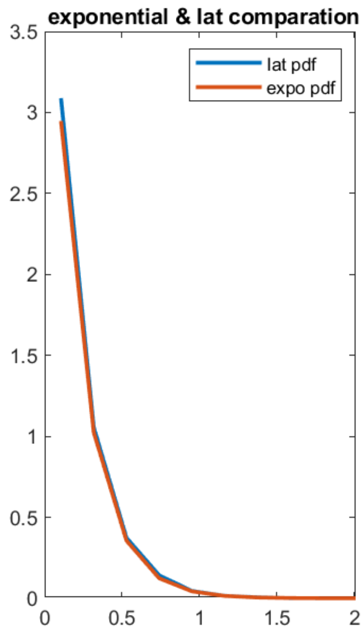
```
load output_MM1_source_1.txt;
at=output_MM1_source_1(:,2);
iat=diff(at);

%we creat iat pdf
[ht,x]=hist(iat);
area=(x(2)-x(1))*sum(ht);
pdf=ht./area;
subplot(1,2,1);
bar(x,pdf);
```



d) Plot the pdf of the vector iat and the exponential with the same interarrival rate. Do the plots match?

```
%we creat expo pdf
expo2=lambda*exp(-x*lambda);
subplot(1,2,2);

%plot the pdfs
plot(x,pdf, 'LineWidth',1.75)
hold on
plot(x,expo2,'LineWidth',1.75)
title('exponential & Iat comparation')
hold off
```

We can observe, both pdfs are practically identical but with some differences in the maximum.

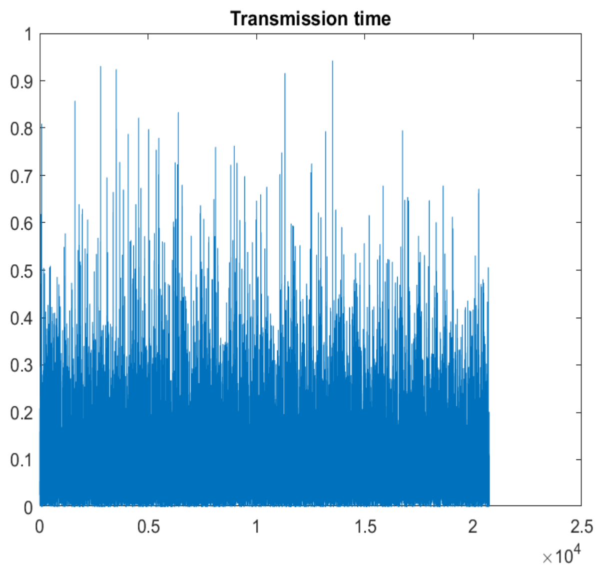


3.3 Transmission time distribution analysis

Repeat similar steps done in the preceding section, but now to analyze the service time distribution.

To analyze the service time we have to get the third number of the file outpu_MM1_source.

```
% 3.3 Transmission time distribution
 analysis
%arrival time=2
%transmission time=3
%transfer time=4
%waiting time=5

name='Transmission time';
number=3;
load output_MM1_source_1.txt;
time=output_MM1_source_1(:,number);
plot(time);
title(name);
figure
```

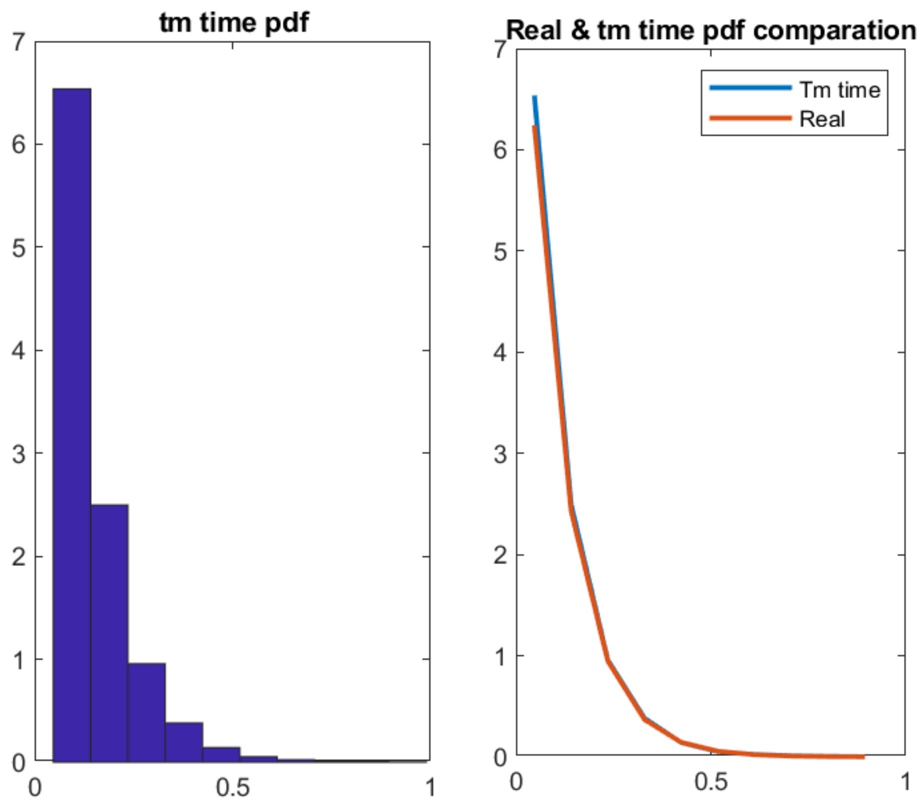


- What does the vector mean? For the random variables compute the maximum, the minimum, the mean value, the variance, the standard deviation and the coefficient of variation.

```
max(tm_time)=0.9425
min(tm_time)=0
mean(tm_time)=0.0990
var(tm_time)=0.0100
std(tm_time)=0.1002
Coef=1.0127 %std(timeService)/mean(timeService)
```

- Obtain and plot the pdf of the vector
- Plot the pdf of the vector and the (in this case gaussian) with the rate. Do the plots match?

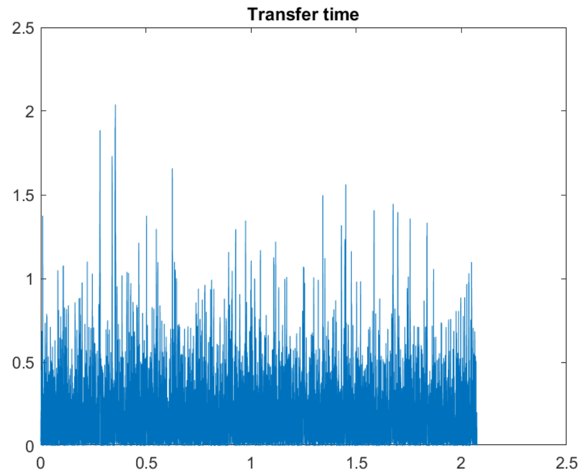
```
%we create timeservice pdf
[ht,x]=hist(tm_time);
area=(x(2)-x(1))*sum(ht);
pdf=ht./area;
subplot(1,2,1);
bar(x,pdf,'histc');
title( 'tm_time pdf');
%we creat expo pdf
lambda=10
compared=lambda*exp(-x*lambda);
subplot(1,2,2);
plot(x,pdf,'LineWidth',1.75);
hold on
plot(x,compared,'LineWidth',1.75);
title('Real & tm time pdf comparation');
legend('Tm time','Real')

hold off
```



3.4 Transfer time distribution analysis

Repeat the same operations to study and analyze the transfer time distribution.
We take the fourth number of the file output_MM1_source.

```
% 3.3 Transfer time distribution analysis
%arrival time=2
%transmission time=3
%transfer time=4
%waiting time=5
name='Transfer time';
number=4;
load output_MM1_source_1.txt;
tt_time=output_MM1_source_1(:,number);
plot(tt_time);
title(name);
```

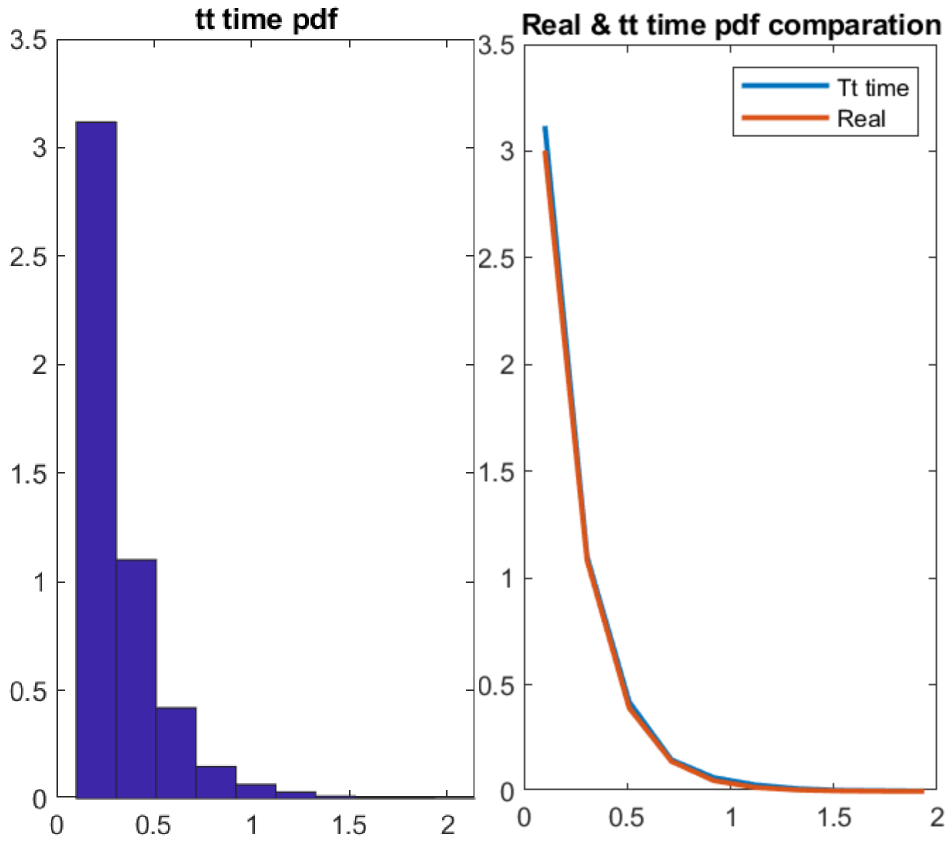


- What does the vector mean? For the random variable compute the maximum, the minimum, the mean value, the variance, the standard deviation and the coefficient of variation.

```
max(tt_time=2.0400
min(tt_time)=0
mean(tt_time)=0.2065
var(tt_time)=0.0477
std(tt_time)=0.2183
Coef=std(tt_time)/mean(tt_time)=1.0571
```

- Obtain and plot the pdf of the vector
- Plot the pdf of the vector and the (in this case gaussian) with the same interarrival rate. Do the plots match?

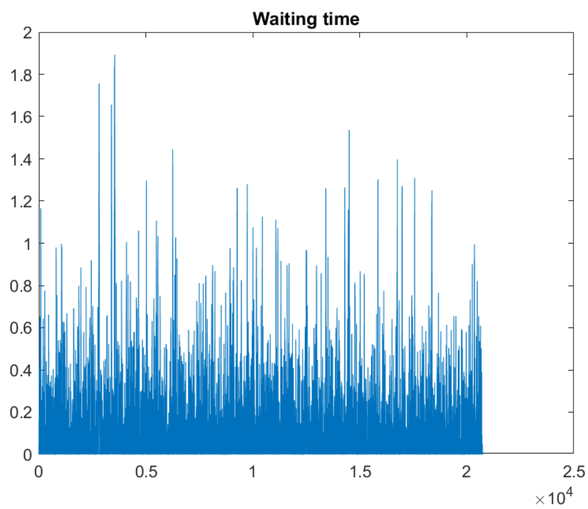
```
%we create timetransfer pdf
[ht,x]=hist(tt_time);
area=(x(2)-x(1))*sum(ht);
pdf=ht./area;
subplot(1,2,1);
bar(x,pdf,'histc');
title( 'tt time pdf');
%we creat expo pdf
lambda=1/mean(tt_time) %5
compared=lambda*exp(-x*lambda);
subplot(1,2,2);
plot(x,pdf,'LineWidth',1.75);
hold on
plot(x,compared,'LineWidth',1.75);
title('Real & tt time pdf comparation');
legend('Tt time','Real')
hold off
```



3.5 Waiting time distribution analysis

a) Repeat the same operations to analyze the waiting time.

```
% 3.5 Waiting time distribution analysis
%arrival time=2
%transmission time=3
%transfer time=4
%waiting time=5
name='Waiting time';
number=5;
load output_MM1_source_1.txt;
Wt_time=output_MM1_source_1(:,number);
plot(Wt_time);
title(name);
```

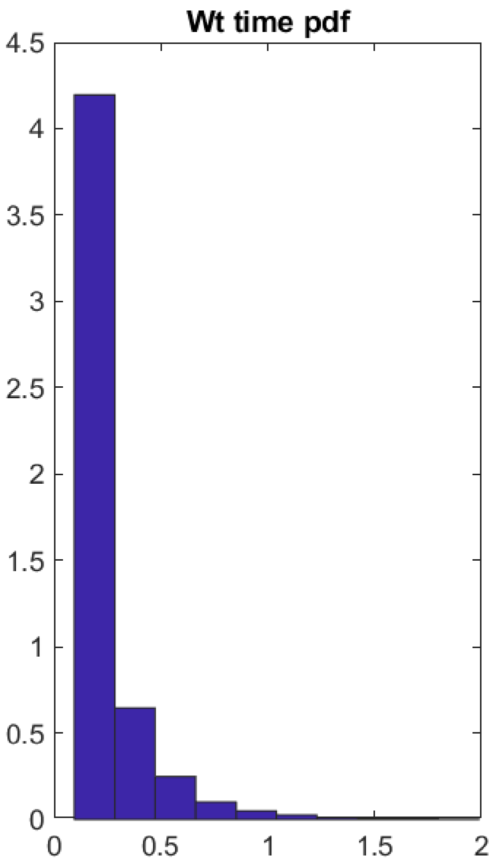


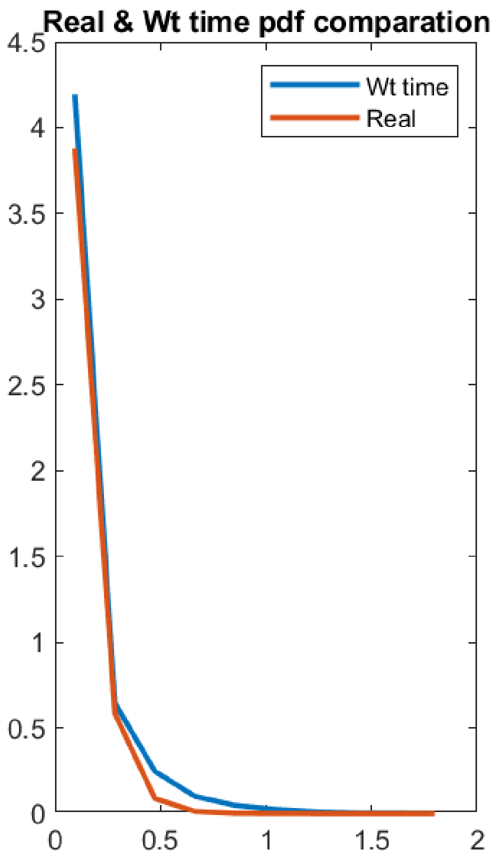
- What does the vector mean? For the random variable compute the maximum, the minimum, the mean value, the variance, the standard deviation and the coefficient of variation.

```
max(Wt_time=1.8933
min(Wt_time)=0
mean(Wt_time)=0.1076
var(Wt_time)=0.0372
std(Wt_time)=0.1930
Coef=std(Wt_time)/mean(tt_time)=1.7944
```

- Obtain and plot the pdf of the vector
- Plot the pdf of the vector and the (in this case gaussian) with the same interarrival rate. Do the plots match?

```
%we create timeservice pdf
[ht,x]=hist(Wt_time);
area=(x(2)-x(1))*sum(ht);
pdf=ht./area;
subplot(1,2,1);
bar(x,pdf,'histc');
title( 'Wt time pdf');
%we creat expo pdf
lambda=1/mean(Wt_time) %10
compared=lambda*exp(-x*lambda);
subplot(1,2,2);
plot(x,pdf,'LineWidth',1.75);
hold on
plot(x,compared,'LineWidth',1.75);
title('Real & Wt time pdf comparation');
legend('Wt time','Real')
hold off
```





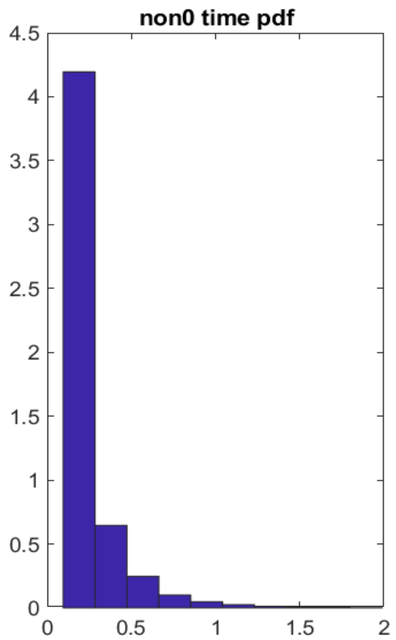
b) Using the find function, obtain the probability of null waiting time (favorable outcomes/possible outcomes) and non-null waiting time. Relate the result with some parameter of section 3.1.

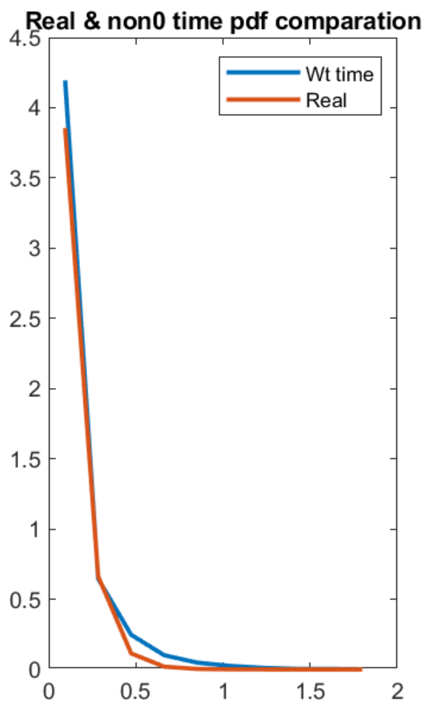
```
%probabilities
found=find(Wt_time);
sizet=size(Wt_time);
sizef=size(found);
non_null_prob=sizef/sizet
null_prob=1-non_nullprob

non_null_prob =  0.4941
null_prob =    0.5059
```

The non-null probability is very similar to the parameter of number of packets in queue which is 0.507622766459831.

c) Using only the non-null values, repeat the operations made in section a).

```
%3.5 c
name='non-null waiting time';
number=5;
load output_MM1_source_1.txt;
non0_time=output_MM1_source_1(:,number);
plot(non0_time);
title(name);
%we create non0_time pdf
[ht,x]=hist(non0_time);
area=(x(2)-x(1))*sum(ht);
pdf=ht./area;
subplot(1,2,1);
bar(x,pdf,'histc');
title( 'non0 time pdf');
%we creat expo pdf
lambda=1/mean(non0_time) %9.2975
compared=lambda*exp(-x*lambda);
subplot(1,2,2);
plot(x,pdf,'LineWidth',1.75);
hold on
plot(x,compared,'LineWidth',1.75);
title('Real & non0 time pdf comparation');
legend('Wt time','Real')
hold off
```





d) Compute E(tw) using the Total Probability Theorem.
E(tw)= p(tw = 0)·0 + p(tw ≠ 0)· E(tw / tw ≠ 0)

```
Etw=null_prob*0+non_null_prob*mean(non0_time)

Etw =0.509*0+0.4941*0.197555804647575

Etw= 0.0976
```

3.6 System occupancy analysis

As previously seen, the mm1.m script provides also a vector named prob with the system state probabilities. In this section, those probabilities will be obtained from the results provided by the simulator and compared with the theoretical values. Scalev Lite provides the needed values in the occupancy files.

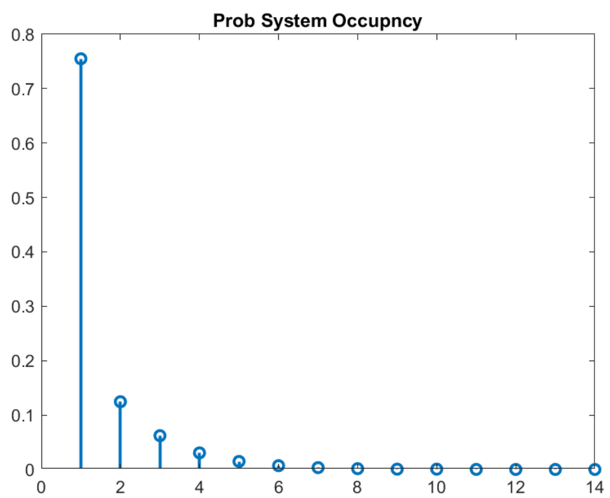
a) Using the file occupancy_MM1_1.txt construct the matrix t_n. Remember that the first column of matrix t_n holds the time instants of the events (arrival or departure), while the second column holds the number of packets in the system after that event.

We build a matrix with the first column of time instants and the second with the number of packets.

In the file occupancy_MM1_1.txt the first column refers to the time instants, the second to the number of packets in the waiting queue and the third is the number of packets in the server.

```
clc
clear
load occupancy_MM1_1.txt
time=occupancy_MM1_1(:,1);
packets=occupancy_MM1_1(:,2);
t_n=[time packets];
Max_time=max(time);
Max_stats=max(packets);
prob=zeros(1,max(packets)+1);
for i=1:length(time)-1
  prob(packets(i)+1)=prob(packets(i)+1)+(time(i+1)-time(i));
end
prob=prob./Max_time;
stem(prob)
```

b) Use the function written in Lab. Session 1 to obtain the simulated system state probabilities.



c) Compare the theoretical and simulated system state probabilities.

In this case, the probability of occupancy of the system is also an exponential distribution. High occupancy states are less likely to happen, as the system is stable.

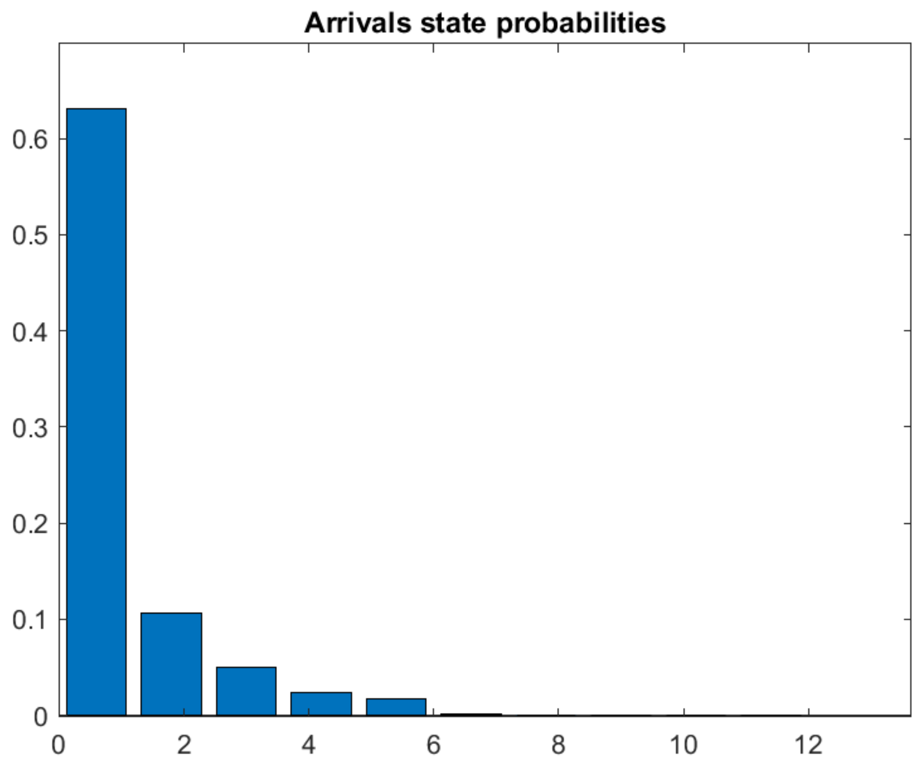
3.7 PASTA property

a) Execute the following commands using the matrix t_n obtained in the previous section. At the end, the matrix state_at_arrival will be obtained, which holds the state seen by the arrivals.

```
n_change=diff(t_n(:,2));
arrival_index=find(n_change==1);
state_at_arrival=t_n(arrival_index,2);
```

b) Obtain the histogram, and normalize it to obtain the state probabilities seen by the arrivals. Compare these probabilities with the system state probabilities.

```
[ht,x]=hist(state_at_arrival);
area=(x(2)-x(1))*sum(ht);
pdf=ht./area;
bar(x,pdf)
title( 'Arrivals state probabilities');
```



The arrivals should have Poisson distributions, in order to the PASTA property. Therefore the probabilities of the state at arrival show the same function as the system state probabilities.

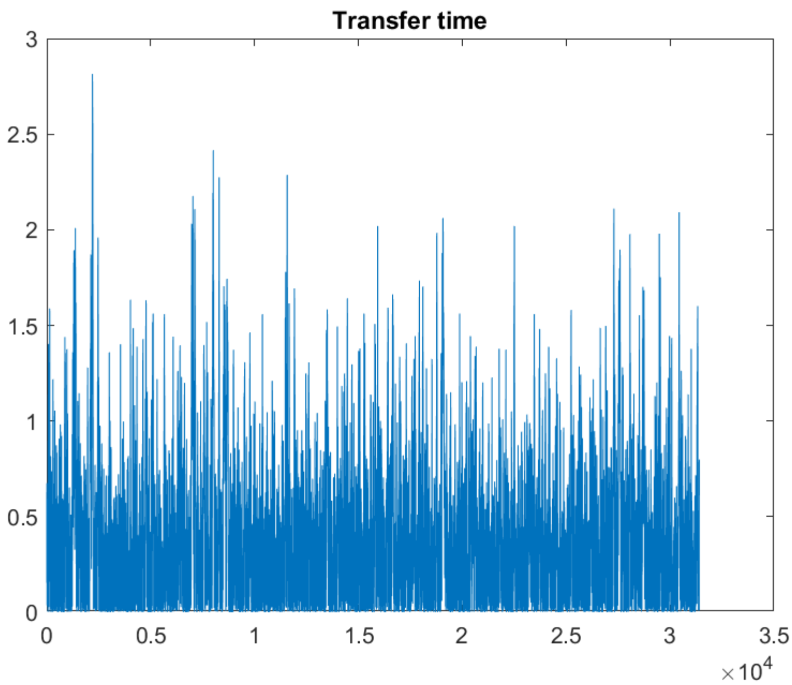
3.8 Additional exercise

Change the arrival rate to obtain a utilization factor equal to 0,75. Do the analysis of:

- Transfer time distribution.

```
Arrival rate (packets/s): 1/0.75
Packet length: 120
Server capacity: 1200

RESULTS FOR THE M/M/1 QUEUE
----------------

Utilization factor:0.13333        %this value to put in scalevLite
Waiting time:0.015385
Service time:0.1
Transfer time:0.11538
Number of packets in queue:0.020513
Number of packets in server:0.13333
Number of packets in system:0.15385
```

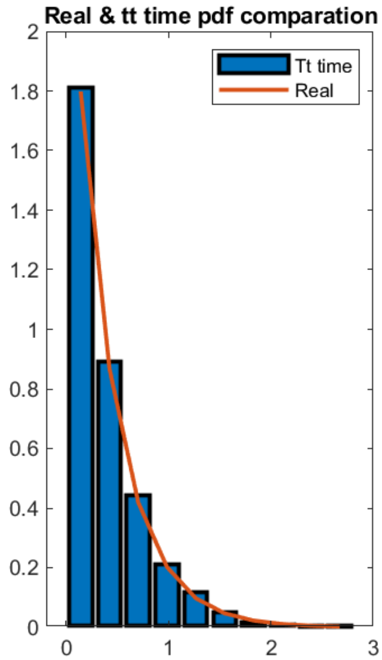
```
load output_ex8_MM1_ex8_source_1.txt;
transf_time=output_ex8_MM1_ex8_source_1(:,4);
plot(transf_time);
title('Transfer time')
```




```
max(tt_time=2.8150
min(tt_time)=0
mean(tt_time)= 0.3863
var(tt_time)= 0.1363
std(tt_time)=  0.3691
Coef=std(tt_time)/mean(tt_time)=0.9557
```

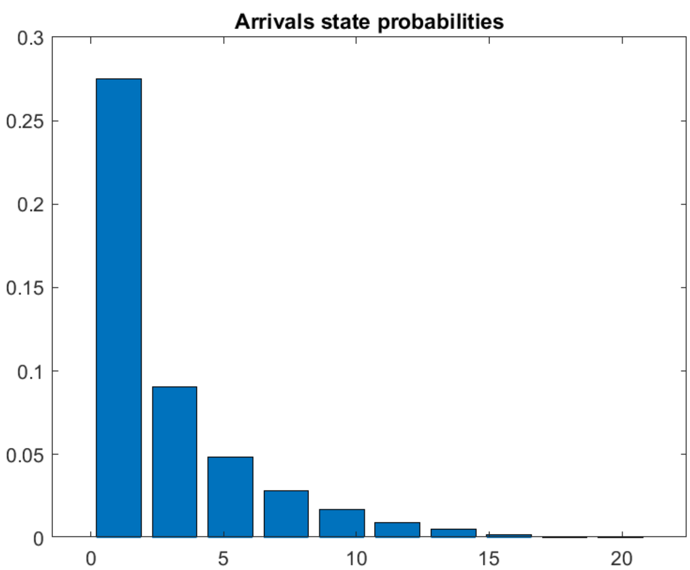
lambda = 2.5889

```
%we create timet pdf
[ht,x]=hist(tt_time);
area=(x(2)-x(1))*sum(ht);
pdf=ht./area;
subplot(1,2,1);
bar(x,pdf,'histc');
title( 'tt time pdf');
%we creat expo pdf
lambda=1/mean(tt_time) %2.5889
compared=lambda*exp(-x*lambda);
plot(x,compared,'LineWidth',1.75);
title('Real & tt time pdf comparation');
legend('Tt time','Real')
hold off
```



- **System occupancy.**

```
load occupancy_ex8_MM1_ex8_1.txt
time=occupancy_ex8_MM1_ex8_1(:,1);
packets=occupancy_ex8_MM1_ex8_1(:,2);
t_n=[time packets];
Max_time=max(time);
Max_stats=max(packets);
prob=zeros(1,max(packets)+1);
for i=1:length(time)-1
 prob(packets(i)+1)=prob(packets(i)+1)+(time(i+1)-time(i));
end
prob=prob./Max_time;
stem (prob,'LineWidth',1.75)
title('Prob System Occupncy')
n_change=diff(t_n(:,2));
arrival_index=find(n_change==1);
state_at_arrival=t_n(arrival_index,2);
[ht,x]=hist(state_at_arrival);
area=(x(2)-x(1))*sum(ht);
pdf=ht./area;
bar(x,pdf)
title( 'Arrivals state probabilities');
```



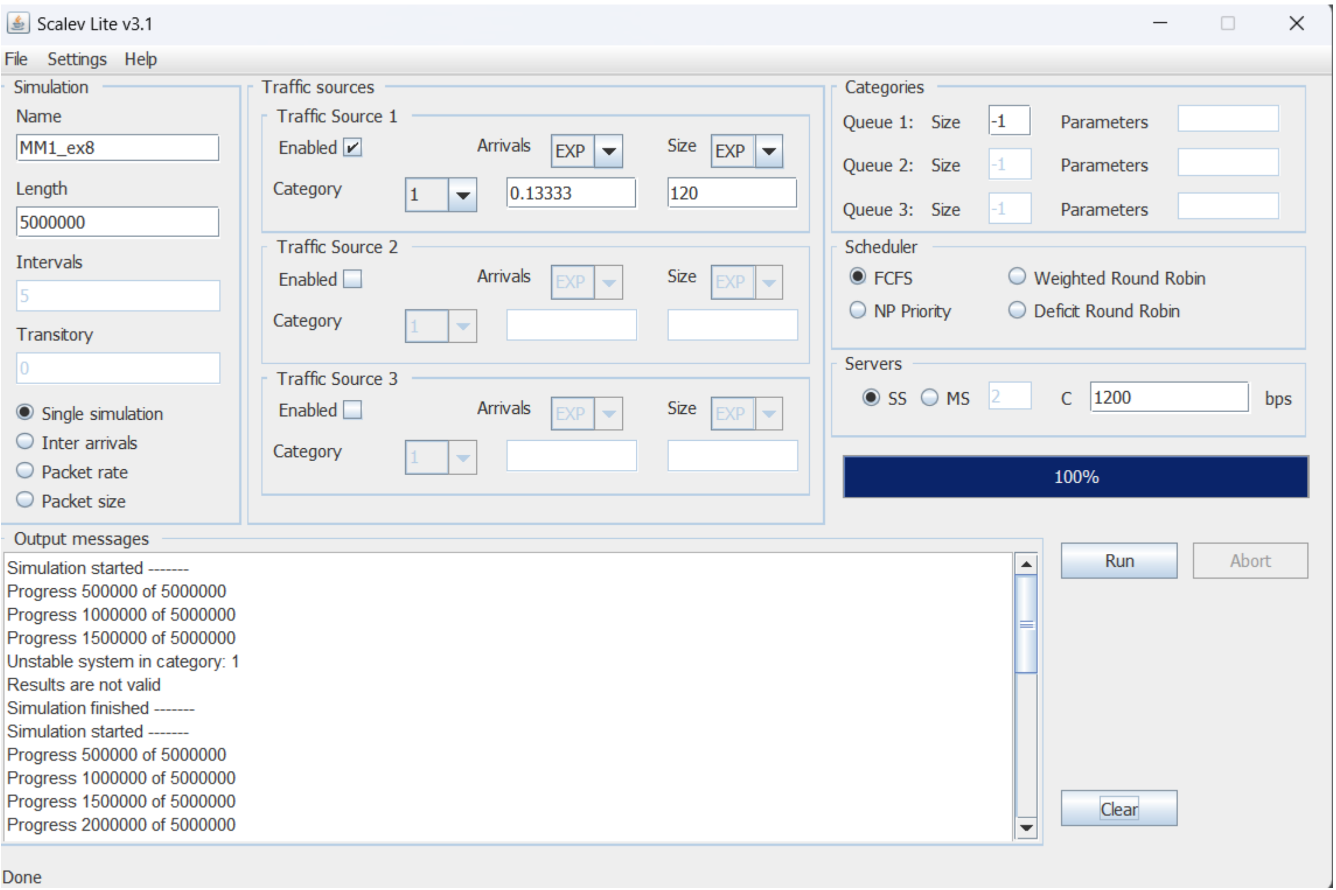
4. Annex

```
% 3.2 Arrival process analysis
load output_MM1_source_1.txt;
at=output_MM1_source_1(:,2);
plot(at)
iat=diff(at);
plot(iat)
title("Ia vector")

%we creat iat pdf
[ht,x]=hist(iat);
bar(x,ht)
area=(x(2)-x(1))*sum(ht);
pdf=ht./area;
figure
plot(x,pdf);
title("Ia pdf")

%we creat expo pdf
expo2=lambda*exp(-x*lambda)
subplot(1,2,2)
%plot the pdfs
plot(x,pdf, 'LineWidth',1.75)
hold on
plot(x,expo2,'LineWidth',1.75)
title('exponential Iat comparation')
hold off

% 3.3 Transmission time distribution analysis
%arrival time=2, %transmission time=3, %transfer time=4
%waiting time=5
name='Transmission time';
number=3;
load output_MM1_source_1.txt;
tm_time=output_MM1_source_1(:,number);
plot(tm_time);
title(name);
figure
max(tm_time)
min(tm_time)
mean(tm_time)
var(tm_time)
std(tm_time)
coef=std(tm_time)/mean(tm_time)

%we create timeservice pdf
[ht,x]=hist(tm_time);
area=(x(2)-x(1))*sum(ht);
pdf=ht./area;
subplot(1,2,1);
bar(x,pdf,'histc');
title( 'tm_time pdf');

%we creat expo pdf
lambda=10
compared=lambda*exp(-x*lambda);
subplot(1,2,2);
plot(x,pdf,'LineWidth',1.75);
hold on
plot(x,compared,'LineWidth',1.75);
title('Real & tm time pdf comparation');
legend('Tm time','Real')
```

```
hold off

% 3.4 Transfer time distribution analysis
name='Transfer time';
number=4;
load output_MM1_source_1.txt;
tt_time=output_MM1_source_1(:,number);
plot(tt_time);
title(name);
max_tt=max(tt_time)
min_tt=min(tt_time)
mean_tt=mean(tt_time)
var_tt=var(tt_time)
std_tt=std(tt_time)
coef_tt=std(tt_time)/mean(tt_time)

%we create timeservice pdf
[ht,x]=hist(tt_time);
area=(x(2)-x(1))*sum(ht);
pdf=ht./area;
subplot(1,2,1);
bar(x,pdf,'histc');
title( 'tt time pdf');

%we creat expo pdf
lambda=5
compared=lambda*exp(-x*lambda);
subplot(1,2,2);
plot(x,pdf,'LineWidth',1.75);
hold on
plot(x,compared,'LineWidth',1.75);
title('Real & tt time pdf comparation');
legend('Tt time','Real')
hold off

% 3.5 Waiting time distribution analysis
name='Waiting time';
number=5;
load output_MM1_source_1.txt;
Wt_time=output_MM1_source_1(:,number);
plot(Wt_time);
title(name);
max_Wt=max(Wt_time)
min_Wt=min(Wt_time)
mean_Wt=mean(Wt_time)
var_Wt=var(Wt_time)
std_Wt=std(Wt_time)
coef_Wt=std(Wt_time)/mean(Wt_time)

%we create timeservice pdf
[ht,x]=hist(Wt_time);
area=(x(2)-x(1))*sum(ht);
pdf=ht./area;
subplot(1,2,1);
bar(x,pdf,'histc');
title( 'Wt time pdf');

%we creat expo pdf
lambda=10
compared=lambda*exp(-x*lambda);
subplot(1,2,2);
plot(x,pdf,'LineWidth',1.75);
hold on
plot(x,compared,'LineWidth',1.75);
```

```
title('Real & Wt time pdf comparation');
legend('Wt time','Real')
hold off
found=find(Wt_time);
sizet=size(Wt_time);
sizef=size(found);
non_null_prob=sizef/sizet
null_prob=1-non_null_prob

%3.5 c
name='non-null waiting time';
number=5;
load output_MM1_source_1.txt;
non0_time=output_MM1_source_1(:,number);
plot(non0_time);
title(name);

%we create non0_time pdf
[ht,x]=hist(non0_time);
area=(x(2)-x(1))*sum(ht);
pdf=ht./area;
subplot(1,2,1);
bar(x,pdf,'histc');
title( 'non0 time pdf');

%we creat expo pdf
lambda=1/mean(non0_time)
compared=lambda*exp(-x*lambda);
subplot(1,2,2);
plot(x,pdf,'LineWidth',1.75);
hold on
plot(x,compared,'LineWidth',1.75);
title('Real & non0 time pdf comparation');
legend('Wt time','Real')
hold off
%
Etw= null_prob*0+non_null_prob*mean(non0_time)


%3.6 System occupancy analysis
load occupancy_MM1_1.txt
time=occupancy_MM1_1(:,1);
packets=occupancy_MM1_1(:,2);
t_n=[time packets];
Max_time=max(time);
Max_stats=max(packets);
prob=zeros(1,max(packets)+1);
for i=1:length(time)-1
 prob(packets(i)+1)=prob(packets(i)+1)+(time(i+1)-time(i));
end
prob=prob./Max_time;
stem (prob,'LineWidth',1.75)
title('Prob System Occupancy')

%3.7 PASTA property
n_change=diff(t_n(:,2));
arrival_index=find(n_change==1);
state_at_arrival=t_n(arrival_index,2);
[ht,x]=hist(state_at_arrival);
area=(x(2)-x(1))*sum(ht);
pdf=ht./area;
bar(x,pdf)
title( 'Arrivals state probabilities');
```

```
%3.8 Transfer time distribution.
Scalev.Lite to generate the .txt file of occupancy and the output
file
```




```
%the matlab script for this exercise is the same as 3.4 but changing
these lines

load output_ex8_MM1_ex8_source_1.txt;
tt_time=output_ex8_MM1_ex8_source_1(:,4);


%3.8 Transfer time distribution.

%the matlab script for this exercise is the same as 3.6 but changing
these lines

load occupancy_ex8_MM1_ex8_1.txt
time=occupancy_ex8_MM1_ex8_1(:,1);
packets=occupancy_ex8_MM1_ex8_1(:,2);
```