ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ
ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΕΜΠ

“QUEUING SYSTEMS”

Ορφανουδάκης Φίλιππος 03113140

5η Ομάδα Ασκήσεων

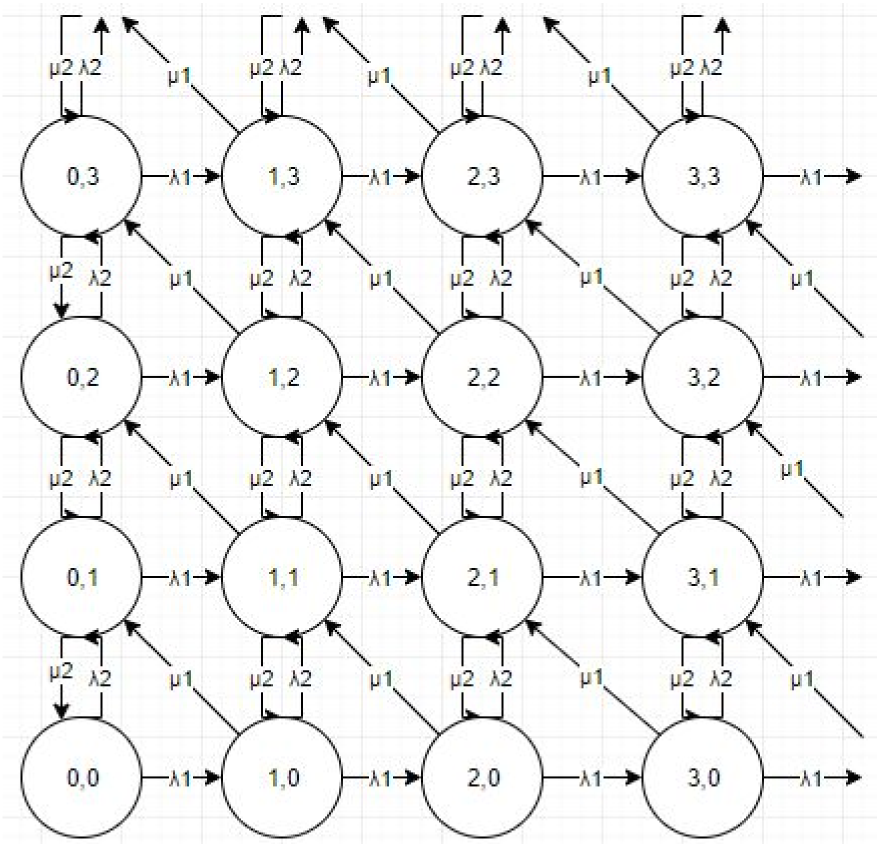
Δίκτυο δύο εκθετικών ουρών εν σειρά

(1) Για να έχουν την μορφή γινομένου πρέπει να τηρούνται οι εξής προυποθέσεις:
- Μαρκοβιανες συνθήκες , δηλαδή Poisson αφίξεις πελατών , εκθετικές εξυπηρετήσεις πελατών και ανεξάρτητη η μία από την άλλη και ανεξαρτησία των πελατών όπου και αν βρίσκονται.
- Παραδοχή Kleinrock ανεξαρτησίας εξυπηρετήσεων, οι χρόνοι εξυπηρέτησης (ανάλογοι του μήκους πακέτου) δεν διατηρούν τα μεγέθη τους όταν προωθούνται μεταξύ συστημάτων (ουρών) εξυπηρέτησης . Οι χρόνοι εξυπηρέτησης ανατίθενται σε κάθε σύστημα σαν ανεξάρτητες εκθετικές τυχαίες μεταβλητές .
- Απειρη χωρητικότητα , δηλαδή δεν έχω μπλοκαρισματα πελατών, επομένως οι αφίξεις στην δεύτερη ουρά είναι λ1+λ2 , αφου δεν πρόκεται να χαθεί απο την πρώτη στη δεύτερη.
- Για να έχω εργοδικότητα πρέπει να ισχύει ότι λ1<μ1 και λ1+λ2<μ2.

(2) Για το φορτιο 1 : ρ1=λ1/μ1 και για το φορτίο 2 : ρ2 = λ1+λ2/μ2

Αφού όπως εξηγούμε παραπάνω έχουμε στις προυποθέσεις μας άπειρη χωρητικότητα επομένως όλοι οι πελάτες που φτάνουν στην πρώτη ουρά θα μπούν και στη δεύτερη (λ1) και με εφαρμογή άθροισης δυο ανεξάρτητων ροών Poisson λ1 , λ2 έχω λ=λ1+λ2.

(3)



(4) Για την επαλήθευση θα πάρουμε 3 αντιπροσωπευτικές καταστάσεις :

την (2,2) :

$$P(2,2)*(\lambda_2+\mu_2+\mu_1+\lambda_1)=P(2,3)*\mu_2+P(1,2)*\lambda_1+P(2,1)*\lambda_2+P(3,1)*\mu_1$$

πάμε να δούμε αν ισχύει το γινόμενο δηλαδή :

$$(\lambda_2+\mu_2+\mu_1+\lambda_1)*K*\rho_1^2 * \rho_2^2= K*(\lambda_2+\lambda_1)*\rho_1^2 * \rho_2^2 + K*\mu_2*\rho_1^2 * \rho_2^2 + K*\mu_1*\rho_1^2 * \rho_2^2 =K*\mu_2*\rho_1^2 * \rho_2^3 + K*(\lambda_1+\lambda_2)\rho_1^2 * \rho_1 + K*\lambda_1*p_1*p_2^2= \mu_2*P(2,3)+\lambda_1*(1,2)+ \lambda_2*P(2,1)+ \mu_1*P(3,1)$$ **ισχύει .**

την (1,0) :

$$P(1,0)*(\lambda_1+\mu_1+\lambda_2)=P(0,0)*\lambda_1+P(1,1)*\mu_2$$

πάμε να δούμε αν ισχύει το γινόμενο δηλαδή :

$$(\lambda_1+\mu_1+\lambda_2)*K*p_1=K*(\lambda_1+\lambda_2)*p_1 + K*\mu_1*p_1= K*\mu_2*p_1*p_2 + K*\lambda_1= P(0,0)*\lambda_1+P(1,1)*\mu_2$$
ισχύει .

την (0,1) :

$$P(0,1)*(\lambda_1+\lambda_2+\mu_2)=P(0,0)*\lambda_2+ P(1,0)*\mu_1+P(0,2)*\mu_2$$

πάμε να δούμε αν ισχύει το γινόμενο δηλαδή :

(λ1+λ2+μ2)*Κ*ρ2=Κ*(λ1+λ2)*ρ2 + Κ*μ2*ρ2=μ2*P(0,2) + Κ*(λ1+λ2)P(0,0)= P(0,0)*λ2+ P(1,0)*μ1+P(0,2)*μ2 ισχύει .

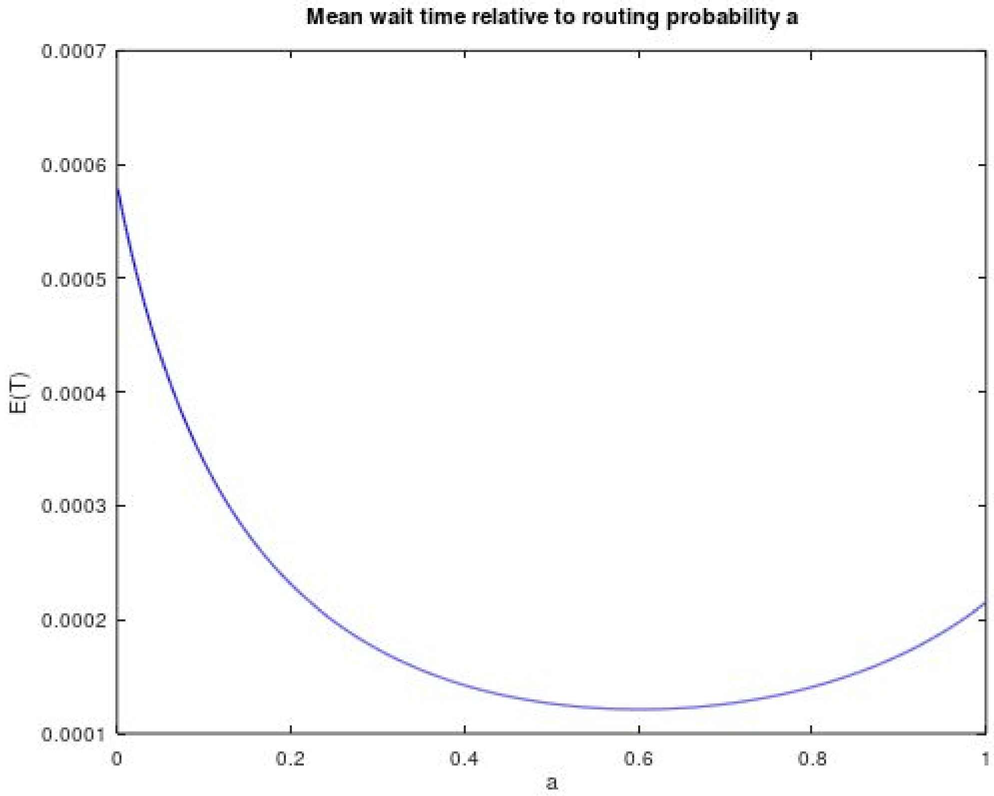
(5) E(T)= E(T1)+ E(T2). Όπου E(T1)=1/μ1/1-ρ1 και E(T2)=1/μ2/1-ρ2

Δίκτυο με εναλλακτική δρομολόγηση

(1)

- Τυχαίες εξωτερικές αφίξεις – Poisson.
- Τυχαία δρομολόγηση πακέτων βάσει των πιθανοτήτων a,1-a .
- Ανεξάρτητες εκθετικές εξυπηρετήσεις πακέτων, παραδοχή Kleinrock ανεξαρτησίας εξυπηρετήσεων.
- Άπειρες ουρές FIFO, χωρίς απώλειες.

(2)



minimum E(T) is 0.000121198 for a = 0.601

Ανοιχτό δίκτυο ουρών αναμονής

(1)

- Τυχαίες εξωτερικές αφίξεις – Poisson.
- Τυχαία δρομολόγηση πακέτων βάσει των αντίστοιχων πιθανοτήτων .
- Ανεξάρτητες εκθετικές εξυπηρετήσεις πακέτων, παραδοχή Kleinrock ανεξαρτησίας εξυπηρετήσεων.
- Άπειρες ουρές FIFO, χωρίς απώλειες.

(2)

ρ1=λ1/μ1 , ρ2=λ2+(2/7)λ1/μ2 , ρ3=(4/7)λ1/μ3 , ρ4=(3/7)λ1/μ4 , ρ5=λ2+(4/7)λ1/μ5

Η συνθήκη για εργοδικότητα είναι κάθε μια ουρά να είναι εργοδική.

```
function [r, ergodic] = intensities(lambda, mu)
```

```
  r(1) = lambda(1)/mu(1);
  r(2) = (lambda(2) + 2/7*lambda(1))/mu(2);
  r(3) = 4/7*lambda(1)/mu(3);
  r(4) = 3/7*lambda(1)/mu(4);
  r(5) = (lambda(2) + 4/7*lambda(1))/mu(5);
  ergodic = 1;
  disp("Intensities are:");
  fprintf("r(1)=%d\n",r(1));
  fprintf("r(2)=%d\n",r(2));
  fprintf("r(3)=%d\n",r(3));
  fprintf("r(4)=%d\n",r(4));
  fprintf("r(5)=%d\n",r(5));
  for i=1:length(r)
    if r(i) >= 1
      ergodic = 0;
      break;
    endif
  endfor
endfunction
```

(3)
Όπως φαίνεται και στις διαφάνειες :

$$\sum_{i=1}^{M} \mathrm{E}(n_i) = \sum_{i=1}^{M} \frac{\rho_i}{1-\rho_i}$$

Επομένως μπορούμε εύκολα να υπολογίσουμε τα mean_clients :

```
function [E_m] = mean_clients (lambda, mu)
  [r,ergodic] = intensities(lambda,mu);
  if ergodic == 0
    error("System is not ergodic, cannot calculate mean number of
clients");
  endif
  for i=1:columns(r)
    E_m(i) = r(i)/(1-r(i));
  endfor
endfunction
```

(4)

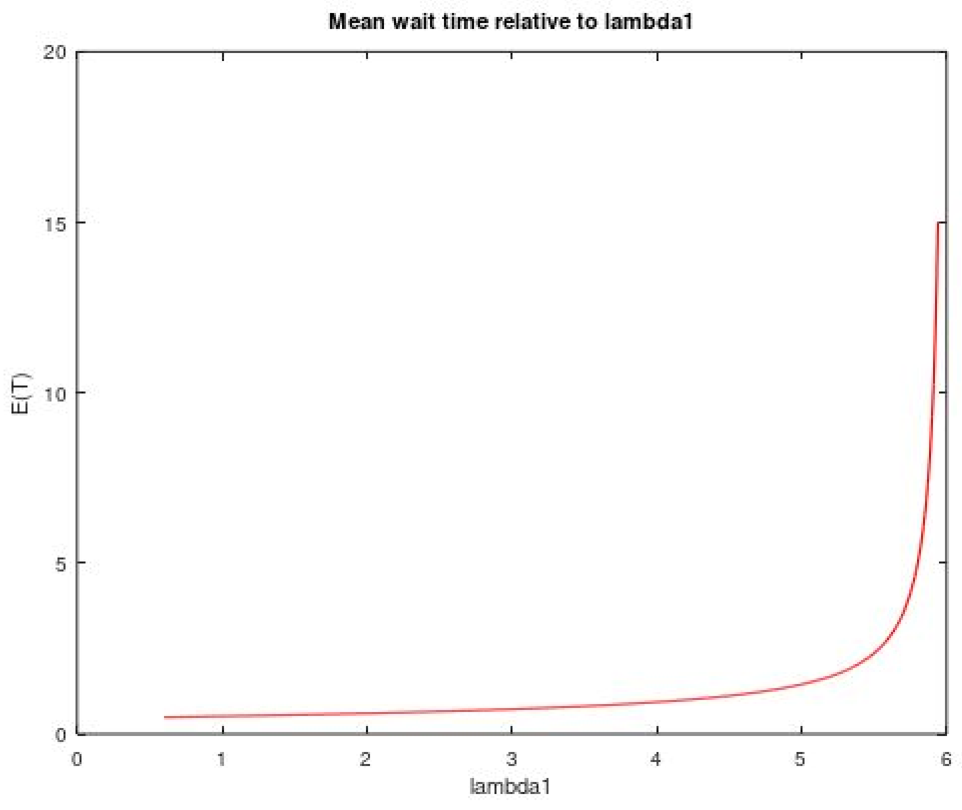
Intensities are:

r(1)=0.666667
r(2)=0.428571
r(3)=0.285714
r(4)=0.244898
r(5)=0.547619
Average wait time in the system is 0.93697

(5)

Βλέπουμε ότι η το μεγαλύτερο φόρτο το έχει η πρώτη ουρά , επομένως αυτή είναι η στενωπός του δικτύου.
Για να παραμείνει εργοδικό το σύστημα πρέπει λ1<μ1 , αρα λ1max=6.

(6)



ο κώδικας που χρησιμοποιήθηκε είναι :

```
clc;
clear all;
close all;
pkg load queueing;
```

```
#2
a = 0.001:0.001:0.999;
C1 = 15000000;
C2 = 12000000;
mean_packet_size = 128*8;
m1 = C1/mean_packet_size;
m2 = C2/mean_packet_size;
min_T = 1000000;
for i=1:length(a)
  lambda = 10000.*[a(i),1-a(i)];
  E_n = lambda(1)/(m1-lambda(1)) + lambda(2)/(m2-lambda(2));
  gamma = sum(lambda);
  E_T(i) = E_n/gamma;
  if E_T(i) < min_T
    min_T = E_T(i);
    min_a = a(i);
  endif
endfor

figure(1);
plot(a,E_T,"b","linewidth",1.2);
title("Mean wait time relative to routing probability a");
xlabel("a");
ylabel("E(T)");
fprintf("minimum E(T) is %d for a = %d\n",min_T, min_a);

function [r, ergodic] = intensities(lambda, mu)
  r(1) = lambda(1)/mu(1);
  r(2) = (lambda(2) + 2/7*lambda(1))/mu(2);
  r(3) = 4/7*lambda(1)/mu(3);
  r(4) = 3/7*lambda(1)/mu(4);
  r(5) = (lambda(2) + 4/7*lambda(1))/mu(5);
  ergodic = 1;
  disp("Intensities are:");
  fprintf("r(1)=%d\n",r(1));
  fprintf("r(2)=%d\n",r(2));
  fprintf("r(3)=%d\n",r(3));
  fprintf("r(4)=%d\n",r(4));
  fprintf("r(5)=%d\n",r(5));
  for i=1:length(r)
    if r(i) >= 1
      ergodic = 0;
      break;
    endif
  endfor
```

```
endfunction


function [E_m] = mean_clients (lambda, mu)
  [r,ergodic] = intensities(lambda,mu);
  if ergodic == 0
    error("System is not ergodic, cannot calculate mean number of clients");
  endif
  for i=1:columns(r)
    E_m(i) = r(i)/(1-r(i));
  endfor
endfunction




#3/4
lambda = [4,1];
mu = [6,5,8,7,6];
E_n_total = sum(mean_clients(lambda,mu));
gamma = sum(lambda);
fprintf("Average wait time in the system is %d\n",E_n_total/gamma);
#3/6
max = 6;
lambda1=(0.1*max):0.01:(0.99*max);
lambda2 = 1;
mu = [6,5,8,7,6];
for i=1:length(lambda1)
  E_T_new(i) = sum(mean_clients([lambda1(i),lambda2],mu))/(lambda1(i)+lambda2);
endfor;

figure(2);
plot(lambda1,E_T_new,"r","linewidth",1.2);
title("Mean wait time relative to lambda1");
xlabel("lambda1");
ylabel("E(T)");
```