Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών
Ακ. έτος 2020-2021, 6ο Εξάμηνο: Συστήματα Αναμονής

Παναγιώτα-Μικαέλα Ξυλιά
ΑΜ:03118859

4η Ομάδα Ασκήσεων

Δίκτυο με εναλλακτική δρομολόγηση

(1) Απαραίτητες παραδοχές ώστε οι σύνδεσμοι (γραμμές) να μπορούν να μοντελοποιηθούν σαν M/M/1 ουρές είναι:

- Ο μέσος ρυθμός ροής πακέτων λ πρέπει να ακολουθεί κατανομή Poisson
- Το μέγεθος των πακέτων πρέπει να ακολουθεί εκθετική κατανομή

Επομένως, θα έχουμε $\lambda_1=\alpha\lambda$, $\lambda_2=(1-\alpha)\lambda$ και $\mu i = \frac{\mathrm{Ci}}{\mathrm{E(L)}}$.

$$\mu 1 = \frac{15 * 10^6 bps}{128 * 8\ bits} \approx 14648Hz \quad \kappa\alpha\iota \quad \mu 2 = \frac{12 * 10^6 bps}{128 * 8\ bits} \approx 11718Hz$$

$$\rho 1 = \frac{\lambda 1}{\mu 1} = \frac{\alpha\lambda}{\mu 1} = 0.682\alpha < 1 \ \ \kappa\alpha\iota \ \ \rho 2 = \frac{\lambda 2}{\mu 2} = \frac{(1-\alpha)\lambda}{\mu 2} = 0.682(1-\alpha) < 1$$

(2) Από το θεώρημα Jackson έχουμε:

$$E(n) = E(n1) + E(n2) = \frac{\rho 1}{1-\rho 1} + \frac{\rho 2}{1-\rho 2} = \frac{\frac{\alpha\lambda}{\mu 1}}{1-\frac{\alpha\lambda}{\mu 1}} + \frac{\frac{(1-\alpha)\lambda}{\mu 2}}{1-\frac{(1-\alpha)\lambda}{\mu 2}} = \frac{\alpha\lambda}{\mu 1-\alpha\lambda} + \frac{(1-\alpha)\lambda}{\mu 2-(1-\alpha)\lambda}.$$

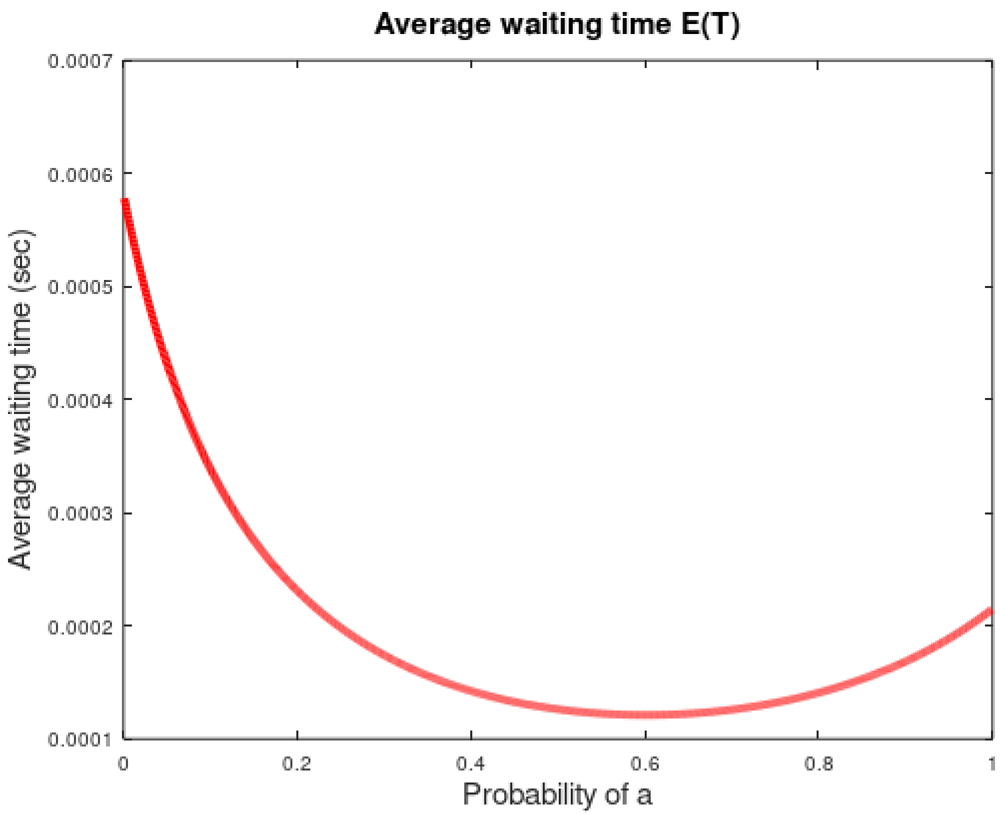
Από την φόρμουλα του Little :

$$E(T) = \frac{E(n)}{\gamma} = \frac{E(n)}{\lambda}$$

Με την χρήση του octave φτιάχνουμε το διάγραμμα του μέσου χρόνου καθυστέρησης E(T) ενός τυχαίου πακέτου στο σύστημα συναρτήσει του α, για το οποίο θα ισχύει

```
minR = 1.2120e-04
position = 601
```

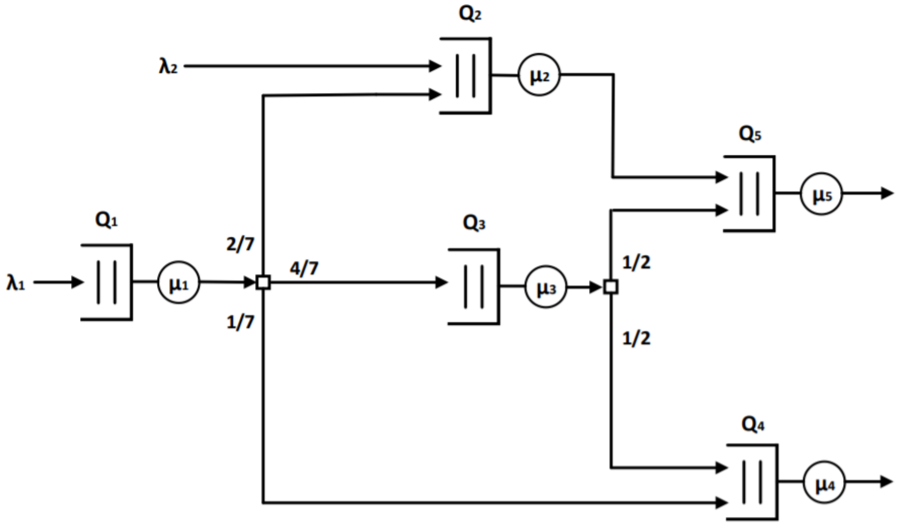
Δηλαδή το ελάχιστο $E(T)=1.212*10^{-4}$ και α=0.601



Κώδικας:

```
pkg load queueing

clc;
clear all;
close all;

a = 0.001:0.001:0.999;
lambda = 10000;

mu1 = (15 * 10^6) / (128 * 8);
mu2 = (12 * 10^6) / (128 * 8);

lambda1 = a.*lambda;
lambda2 = (1-a).*lambda;

[U1 R1 Q1 X1 P1] = qsmm1(lambda1,mu1);
[U2 R2 Q2 X2 P2] = qsmm1(lambda2,mu2);

R = a.*R1 + (1-a).*R2;

figure(1);
plot(a,R,'r',"linewidth",2);
title("Average waiting time E(T)","fontsize", 15);
xlabel("Probability of a","fontsize", 15);
ylabel("Average waiting time (sec)","fontsize", 15);

[minR,position] = min(R);
display(minR);
display(position);
```

Ανοιχτό δίκτυο ουρών αναμονής



Όλες οι αφίξεις με ρυθμό λi ακολουθούν κατανομή Poisson, ενώ οι εξυπηρετήσεις με ρυθμό μi ακολουθούν εκθετική κατανομή.

(1) Για να χρησιμοποιήσουμε το θεώρημα Jackson αρκεί:

- Οι γεννήσεις να ακολουθούν ανεξάρτητη κατανομή Poisson
- Οι θάνατοι να ακολουθούν ανεξάρτητη εκθετική κατανομή
- Οι χρόνοι εξυπηρέτησης πελατών όπως διαπερνούν το δίκτυο δεν διατηρούν την τιμή τους (έλλειψη μνήμης) αλλά αποκτούν χρόνο εξυπηρέτησης ανάλογο με την κατανομή του κάθε εξυπηρετητη
- Η εσωτερική δρομολόγηση να γίνεται με τυχαίο τρόπο με πιθανότητα δρομολόγησης πελάτη από τον κόμβο (ουρά) Qi στον κόμβο Qj : rij

(2) Γνωρίζουμε ότι $\rho = \frac{\lambda}{\mu}$ και άρα

Q1: $\rho 1 = \frac{\lambda 1}{\mu 1}$

Q2: $\rho 2 = \frac{\lambda 2 + \rho 12 \lambda 1}{\mu 2} = \frac{\lambda 2 + \frac{2}{7}\lambda 1}{\mu 2}$

Q3: $\rho 3 = \frac{\rho 13 \lambda 1}{\mu 3} = \frac{4\lambda 1}{7\mu 3}$

Q4: $\rho 4 = \frac{(\rho 14 + \rho 34 \rho 13)\lambda 1}{\mu 4} = \frac{3\lambda 1}{7\mu 4}$

Q1: $\rho 5 = \frac{\lambda 2 + (\rho 12 + \rho 35 \rho 13)\lambda 1}{\mu 5} = \frac{\lambda 2 + \frac{4}{7}\lambda 1}{\mu 5}$

Κώδικας:

```
% erotima 2
function [r,is_ergodic] = intensities(lambda,mu)
r(1) = lambda(1)/mu(1);
r(2) = (lambda(2) + 2*lambda(1)/7)/mu(2);
r(3) = (4*lambda(1)/7)/mu(3);
r(4) = (3*lambda(1)/7)/mu(4);
r(5) = (lambda(2) + (4/7)*lambda(1))/mu(5);
is_ergodic = true;
for i=1:5
  printf('Q%d: %f\n',i,r(i));
  is_ergodic  = is_ergodic && (r(i) < 1)
endfor
printf("Ergodicity: %d \n",is_ergodic)
endfunction
```

(3) Κώδικας:

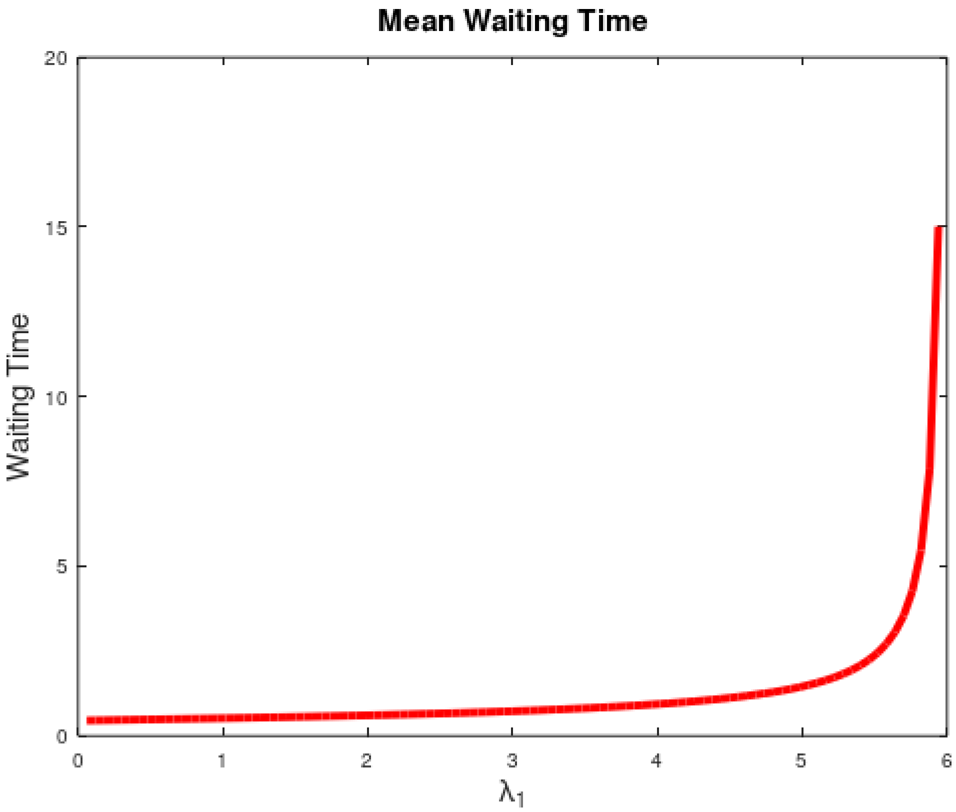
```
% erotima 3
function [R] = mean_clients(lambda,mu)
[r,is_ergodic] = intensities(lambda,mu);
R = r ./ (1-r);
for i=1:5
  printf("Mean Clients at Q%d: %d\n",i,R(i))
endfor
endfunction
```

(4) Κώδικας:

```
% erotima 4
l = 4;
lambda = [l,1];
mu = [6,5,8,7,6];
R = mean_clients(lambda,mu);
average = sum(R)/sum(lambda);
printf("Average service time: %d", average);
```

(5) Η ουρά Q1 θα είναι η στενωπός του δικτύου εφόσον παρατηρούμε προφανώς μεγαλύτερο φορτίο σε αυτήν. Άρα η μέγιστη τιμή λ1 θα είναι λ1max= ρ1max * μ1 = 6 , γιατί ρ1max=1 αφού το σύστημα είναι εργοδικό.

(6)



Συνολικός κώδικας:

```
addpath(pwd);
% erotima 2
function [r,is_ergodic] = intensities(lambda,mu)
r(1) = lambda(1)/mu(1);
r(2) = (lambda(2) + 2*lambda(1)/7)/mu(2);
r(3) = (4*lambda(1)/7)/mu(3);
r(4) = (3*lambda(1)/7)/mu(4);
r(5) = (lambda(2) + (4/7)*lambda(1))/mu(5);
is_ergodic = true;
for i=1:5
  printf('Q%d: %f\n',i,r(i));
  is_ergodic  = is_ergodic && (r(i) < 1)
endfor
printf("Ergodicity: %d \n",is_ergodic)
endfunction


% erotima 3
function [R] = mean_clients(lambda,mu)
[r,is_ergodic] = intensities(lambda,mu);
R = r ./ (1-r);
for i=1:5
  printf("Mean Clients at Q%d: %d\n",i,R(i))
endfor
endfunction

% erotima 4
```

```
l = 4;
lambda = [l,1];
mu = [6,5,8,7,6];
R = mean_clients(lambda,mu);
average = sum(R)/sum(lambda);
printf("Average service time: %d", average);

% erotima 6

max_lambda = 6
for i=1:99
   l = max_lambda*i/100;
   vec_lambda(i) = l;
   lambda = [l,1];
   mu = [6,5,8,7,6];
   vec_sum(i) = sum(mean_clients(lambda,mu))/sum(lambda);
endfor

figure(1);
plot(vec_lambda,vec_sum,"r","linewidth",2);
title("Mean Waiting Time","fontsize", 15);
xlabel('\lambda_1',"fontsize", 15);
ylabel("Waiting Time","fontsize", 15);
```