Συστήματα Αναμονής

# 5η Εργαστηριακή Άσκηση

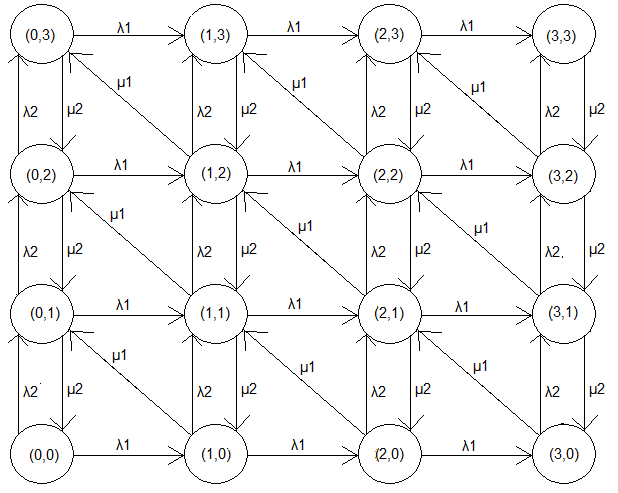
**Όνομα:** Σταύρος Σταύρου

**ΑΜ:** 03115701

**Εξάμηνο:** 6ο-ΣΗΜΜΥ

**Δίκτυο δύο εκθετικών ουρών εν σειρά**

1. Πέραν της απαίτησης για εργοδικότητα (από την οποία εξάγουμε την απαίτηση και ), κάνουμε την παραδοχή πως οι ρυθμοί εξυπηρέτησης και δεν εξαρτώνται μεταξύ τους. Δηλαδή, πως οι 2 τυχαίες μεταβλητές εκθετικής κατανομής που τις αναπαριστούν είναι ανεξάρτητες μεταξύ τους. Παρόλο που η παραδοχή αυτή, προφανώς επηρεάζει τα αποτελέσματα μας (ένας πελάτης συγκεκριμένης ανάγκης εξυπηρέτησης θα έχει ανάλογο χρόνο στις 2 ουρές) εν τούτοις μας δίνει μια αρκετά ακριβή προσέγγιση. Επίσης θεωρούμε άπειρες ουρές FIFO χωρίς απώλειες.
2. Η πρώτη ουρά δέχεται πελάτες με ρυθμό και εξυπηρετεί με ρυθμό 🡺 . Στην έξοδο της πρώτης ουράς εμφανίζονται με ρυθμό , πελάτες που εισέρχονται στην δεύτερη. Επίσης, στη δεύτερη εισέρχεται ένα δεύτερο κύμα πελατών με ρυθμό και ως άθροισμα 2 κατανομών Poisson έχουμε .
3. Έχουμε το εξής διάγραμμα μεταβάσεων:



1. Θέλουμε να επαληθεύσουμε τη σχέση

* Για έχουμε .

Από την υπόθεση παίρνουμε

* Για έχουμε

Από την υπόθεση παίρνουμε:

* Για έχουμε

Από την υπόθεση:

1. Έχουμε , όπου , αφού λόγω άπειρης χωρητικότητας δεν έχουμε απόρριψη πελατών. Έχουμε επίσης, και τελικά .

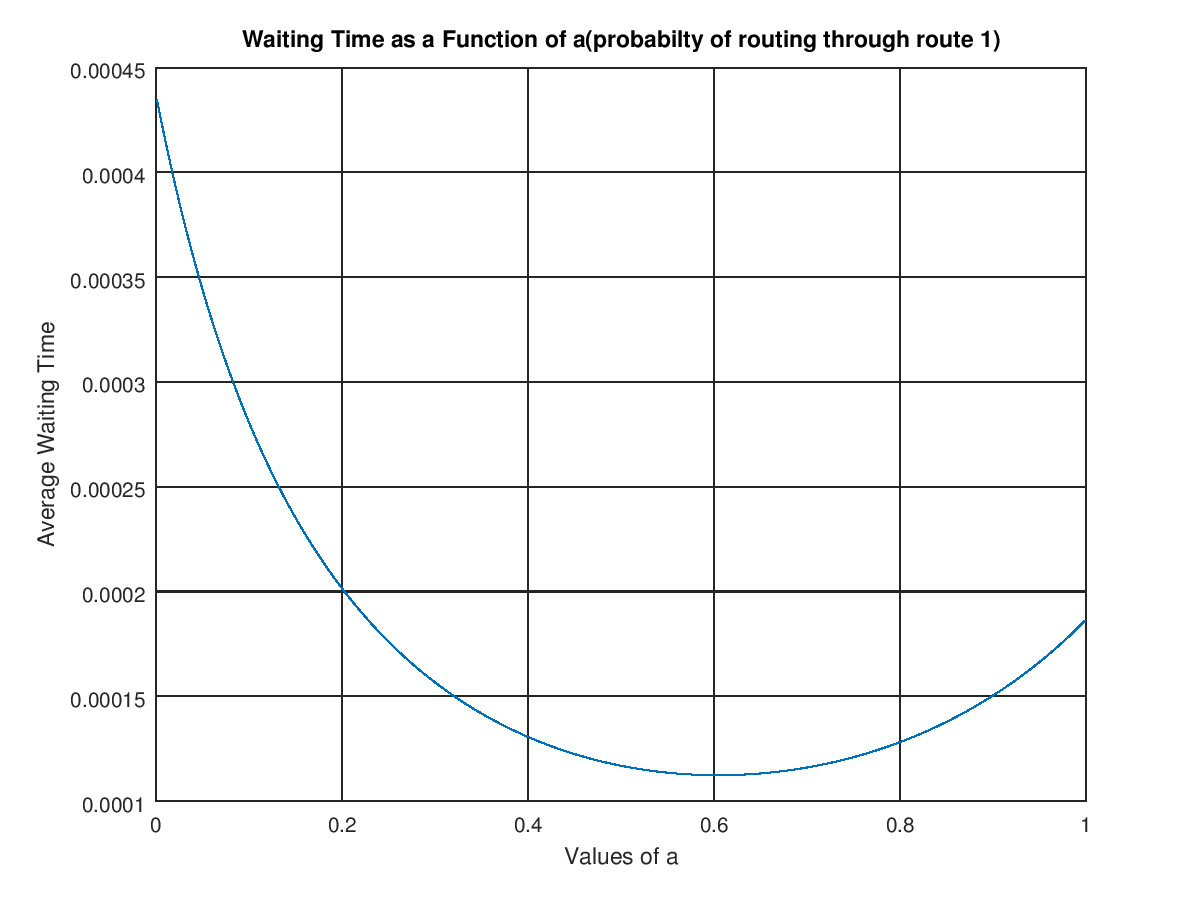
**Δίκτυο με Εναλλακτική Δρομολόγηση**

1. Για να μοντελοποιηθούν οι γραμμές του συστήματος σαν ουρές Μ/Μ/1 θα πρέπει να κάνουμε τις εξής παραδοχές:

* Κάθε γραμμή θα μοντελοποιηθεί σαν μια ουρά με ρυθμό εξυπηρέτησης εκθετικό με ανεξάρτητο χρόνο εξυπηρέτησης και με συνολικό ρυθμό εμφάνισης πελατών Poisson` που θα ισούται με το άθροισμα των εξωτερικών πηγών που εισέρχονται στον κόμβο και τυχούσες εισόδους από εξόδους των άλλων ουρών.
* Σε περιπτώσεις εντός του δικτύου, που υπάρχουν περισσότεροι από 1 δρόμοι για δρομολόγηση του πελάτη η επιλογή του δρόμου γίνεται τυχαία.
* Οι χρόνοι εξυπηρέτησης πελατών έχουν έλλειψη μνήμης.
* Άπειρες ουρές FIFO, χωρίς απώλειες.

1. Αναλύουμε το σύστημα σε 2 ουρές όπως φαίνεται στους πιο κάτω υπολογισμούς

Βάσει του τελευταίου και με χρήση του Octave παίρνουμε το εξής:



Ενώ για την ελάχιστη καθυστέρηση έχουμε:



**Ανοικτό Δίκτυο Ουρών Αναμονής**

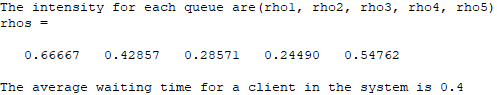
1. Για να μελετηθεί το δίκτυο με χρήση του θεωρήματος Jackson θα πρέπει να κάνουμε τις εξής παραδοχές:

* Κάθε ουρά έχει ανεξάρτητο χρόνο εξυπηρέτησης μέσου ρυθμού και με μέσο ρυθμό εμφάνισης πελατών που θα ισούται με το άθροισμα των εξωτερικών πηγών που εισέρχονται στον κόμβο και τυχούσες εισόδους από εξόδους των άλλων ουρών.
* Σε περιπτώσεις εντός του δικτύου, που υπάρχουν περισσότεροι από 1 δρόμοι για δρομολόγηση του πελάτη η επιλογή του δρόμου γίνεται τυχαία.
* Οι χρόνοι εξυπηρέτησης πελατών έχουν έλλειψη μνήμης.
* Άπειρες ουρές FIFO χωρίς απώλειες.

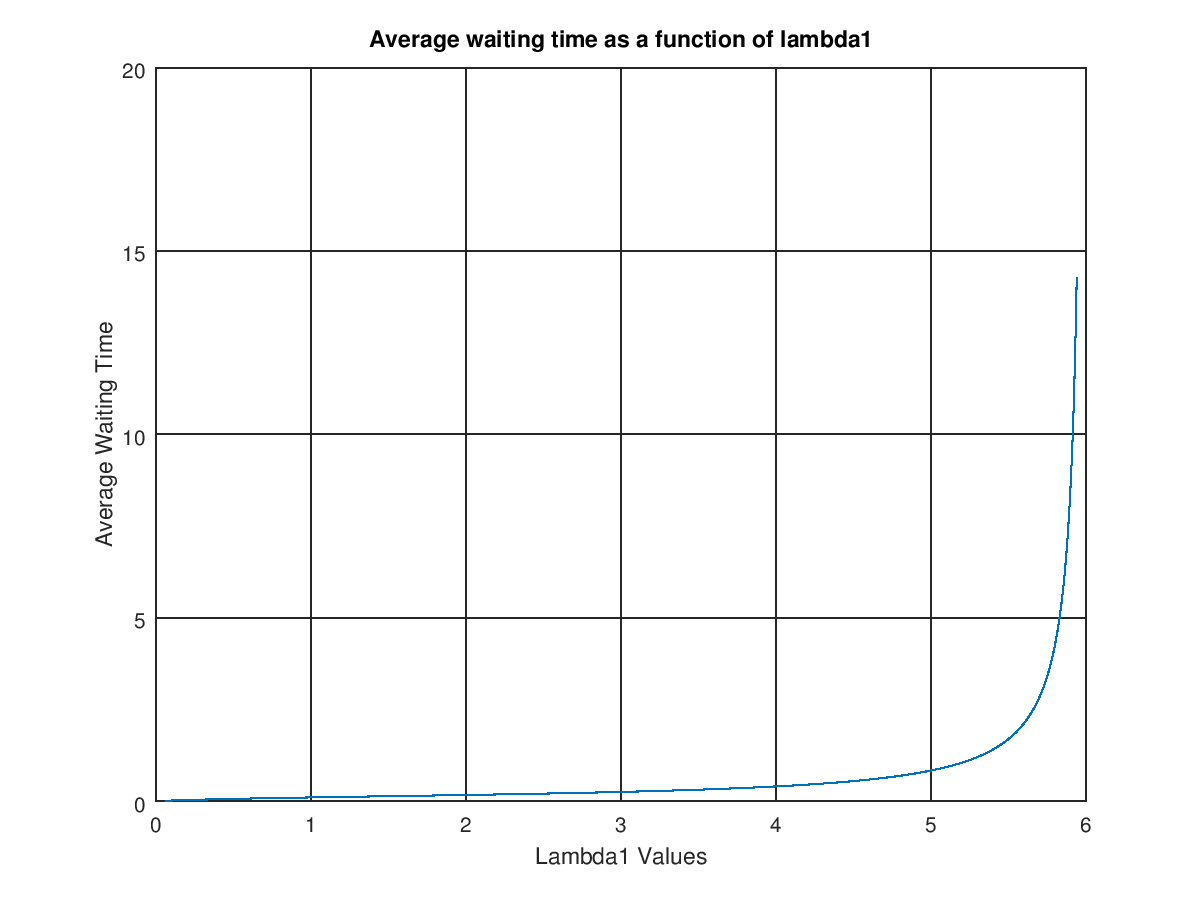
1. Έχουμε για κάθε ουρά:

* Q1:
* Q2:
* Q3:
* Q4:
* Q2:

1. Με χρήση των συναρτήσεων που γράψαμε παίρνουμε την εξής απάντηση από την Octave:



1. Ως στενωπός ουρά χαρακτηρίζεται η ουρά με το μεγαλύτερο φορτίο στο σύστημα, καθώς αυτή είναι εν γένει και η ουρά που θέτει τα «όρια» στις δυνατότητες του συστήματος. Όπως βλέπουμε από τα αποτελέσματα του ερωτήματος (4) αυτή είναι η Q1. Για να είναι το σύστημα εργοδικό θα πρέπει και συνεπώς
2. Παίρνουμε την εξής γραφική παράσταση από την Octave:



Όλοι οι κώδικες που χρησιμοποιήθηκαν, καθώς και οι ζητούμενες συναρτήσεις των ερωτημάτων (2) και (3) του τελευταίου μέρους βρίσκονται στο ακόλουθο παράρτημα.

**ΠΑΡΑΡΤΗΜΑ**

**routing.m:**

clc;

clear all;

close all;

a = 0.001:0.001:0.999;

average\_clients1 = (0.65104 \* a) ./ (1 - 0.65104 \* a);

average\_clients2 = (0.81380 \* (1 - a)) ./ (1 - (1 - a) \* 0.81380);

average\_waiting = (average\_clients1 + average\_clients2) ./ (10\*\*4);

min\_time = min(average\_waiting);

for i = 1:1:999;

if min\_time == average\_waiting(i)

best\_prob = a(i);

break;

endif

endfor

disp(cstrcat("The minimum waiting time is ", num2str(min\_time), " found for probability a = ", num2str(best\_prob)));

figure(1);

plot(a, average\_waiting);

grid on;

title("Waiting Time as a Function of a(probabilty of routing through route 1)");

xlabel("Values of a");

ylabel("Average Waiting Time");

**intensities.m:**

function [rho1 rho2 rho3 rho4 rho5 ergodic] = intensities(lambda1, lambda2, mu1, mu2, mu3, mu4, mu5)

rho1 = lambda1 / mu1;

rho2 = ((2 / 7) \* lambda1 + lambda2) / mu2;

rho3 = (4 / 7) \* lambda1 / mu3;

rho4 = (3 / 7) \* lambda1 / mu4;

rho5 = ((4 / 7) \* lambda1 + lambda2) / mu5;

if rho1 < 1 && rho2 < 1 && rho3 < 1 && rho4 < 1 && rho5 < 1

ergodic = true;

else

ergodic = false;

endif

endfunction

**mean\_clients.m:**

function [clients1 clients2 clients3 clients4 clients5] = mean\_clients(lambda1, lambda2, mu1, mu2, mu3, mu4, mu5)

rho1 = lambda1 / mu1;

rho2 = ((2 / 7) \* lambda1 + lambda2) / mu2;

rho3 = (4 / 7) \* lambda1 / mu3;

rho4 = (3 / 7) \* lambda1 / mu4;

rho5 = ((4 / 7) \* lambda1 + lambda2) / mu5;

clients1 = rho1 / (1 - rho1);

clients2 = rho2 / (1 - rho2);

clients3 = rho3 / (1 - rho3);

clients4 = rho4 / (1 - rho4);

clients5 = rho5 / (1 - rho5);

endfunction

**queue\_example.m:**

clc;

clear all;

close all;

lambda1 = 4;

lambda2 = 1;

mu1 = 6;

mu2 = 5;

mu3 = 8;

mu4 = 7;

mu5 = 6;

rhos = zeros(1,5);

[rhos(1) rhos(2) rhos(3) rhos(4) rhos(5)] = intensities (lambda1, lambda2, mu1, mu2, mu3, mu4, mu5);

clients = mean\_clients(lambda1, lambda2, mu1, mu2, mu3, mu4, mu5);

total\_waiting\_time = sum(clients) / (lambda1 + lambda2);

disp("The intensity for each queue are(rho1, rho2, rho3, rho4, rho5)");

display(rhos);

disp(cstrcat("The average waiting time for a client in the system is ", num2str(total\_waiting\_time)));

**queue\_example\_continued.m:**

clc;

clear all;

close all;

lambda1 = 0.06: 0.006: 5.94;

lambda2 = 1;

mu1 = 6;

mu2 = 5;

mu3 = 8;

mu4 = 7;

mu5 = 6;

for i = 1:1:981

clients = mean\_clients(lambda1(i), lambda2, mu1, mu2, mu3, mu4, mu5);

waiting\_times(i) = sum(clients) / (lambda1(i) + lambda2);

endfor

figure(1);

plot(lambda1, waiting\_times);

grid on;

title("Average waiting time as a function of lambda1");

xlabel("Lambda1 Values");

ylabel("Average Waiting Time");