Συστήματα Αναμονής

(Queuing Systems)

5η Εργαστηριακή Άσκηση

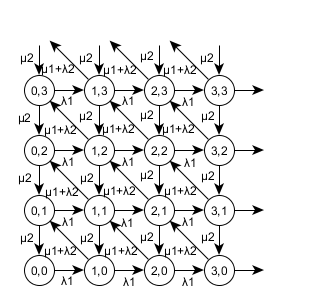
# Λεούσης Σάββας

# Α.Μ.: 03114945

Δίκτυο δύο εκθετικών ουρών εν σειρά

1. Οι παραδοχές που απαιτούνται ώστε να έχουν οι εργοδικές πιθανότητες του συστήματος τη μορφή γινομένου είναι να έχουμε:
   * Ένα ανοικτό δίκτυο δικτυακών κόμβων εξυπηρέτησης κορμού (ουρών αναμονής) με εκθετικούς ρυθμούς εξυπηρέτησης
   * Αφίξεις πελατών από εξωτερικές πηγές άμεσα συνδεδεμένες στον δικτυακό κόμβο κορμού προς εξωτερικούς προορισμούς άμεσα συνδεδεμένους στον δικτυακό κόμβο κορμού : Ανεξάρτητες ροές Poisson μέσου ρυθμού όπου
   * Εσωτερική δρομολόγηση με τυχαίο τρόπο και πιθανότητα δρομολόγησης πελάτη από τον κόμβο κορμού στον κόμβο
   * Έστω πελάτες της ροής διακινούνται μέσα από τον κόμβο κορμού ή αλλιώς τότε τον κόμβο εξυπηρέτησης διαπερνούν ροές με συνολικό μέσο ρυθμό
   * Οι χρόνοι εξυπηρέτησης πελατών όπως διαπερνούν το δίκτυο δεν διατηρούν την τιμή τους αλλά αποκτούν χρόνο εκθυπηρέτησης ανάλογα με την κατανομή του κάθε εξυπηρετητή
2. 1η ουρά:

2η ουρά:



1. Είναι

* Για
* Για
* Για

1. Είναι

# Δίκτυο με εναλλακτική δρομολόγηση

# Πρέπει:

# Οι χρόνοι εξυπηρέτησης να μην διατήρουν τα μεγέθη τους όταν προωθούνται μεταξύ συστημάτων εξυπηρέτησης. Οι χρόνοι εξυπηρέτησης ανατίθενται σε κάθε σύστημα σαν ανεξάρτητες εκθετικές τυχαίες μεταβλητές.

# Οι εργοδικές πιθανότητες έχουν μορφή γινομένου.

# Το ζητούμενο διάγραμμα είναι το παρακάτω:

# 

# Με τη βοήθεια του Octave, η βέλτιστη τιμή που ελαχιστοποιεί το E(T) και ο ελάχιστος χρόνος καθυστέρησης E(T) είναι τα παρακάτω:

# 

# Ανοιχτό δίκτυο ουρών αναμονής

1. Οι παραδοχές που απαιτούνται ώστε το δίκτυο να μπορεί να μελετηθεί ως ένα ανοιχτό δίκτυο με το θεώρημα Jackson είναι να έχουμε:
   1. Ένα ανοικτό δίκτυο δικτυακών κόμβων εξυπηρέτησης κορμού (ουρών αναμονής) με εκθετικούς ρυθμούς εξυπηρέτησης
   2. Αφίξεις πελατών από εξωτερικές πηγές άμεσα συνδεδεμένες στον δικτυακό κόμβο κορμού προς εξωτερικούς προορισμούς άμεσα συνδεδεμένους στον δικτυακό κόμβο κορμού : Ανεξάρτητες ροές Poisson μέσου ρυθμού όπου
   3. Εσωτερική δρομολόγηση με τυχαίο τρόπο και πιθανότητα δρομολόγησης πελάτη από τον κόμβο κορμού στον κόμβο
   4. Έστω πελάτες της ροής διακινούνται μέσα από τον κόμβο κορμού ή αλλιώς τότε τον κόμβο εξυπηρέτησης διαπερνούν ροές με συνολικό μέσο ρυθμό
   5. Οι χρόνοι εξυπηρέτησης πελατών όπως διαπερνούν το δίκτυο δεν διατηρούν την τιμή τους αλλά αποκτούν χρόνο εκθυπηρέτησης ανάλογα με την κατανομή του κάθε εξυπηρετητή

# Η ένταση του φορτίου που δέχεται κάθε ουρά του δικτύου είναι:

# 1η ουρά:

# 2η ουρά:

# 3η ουρά:

# 4η ουρά:

# 5η ουρά:

# Η ζητούμενη συνάρτηση intensities είναι η παρακάτω:

# function [rho ergodicity] = intesities(lambda, mu)

# rho(1)=lambda(1)/mu(1);

# rho(2)=(lambda(2)+(2/7)\*lambda\_1(1))/mu(2);

# rho(3)=((4/7)\*lambda(1))/mu(3);

# rho(4)=((3/7)\*lambda(1))/mu(4);

# rho(5)=(lambda(2)+(4/7)\*lambda(1))/mu(5);

# ergodicity=1;

# for i=1:5

# if rho(i)>1

# ergodicity=0;

# endif

# endfor

# #display(rho);

# endfunction

# Η ζητούμενη συνάρτηση mean\_clients είναι η παρακάτω:

# function clients = mean\_clients(lambda, mu)

# [rho erg] = intesities(lambda,mu);

# if erg == 1

# for i=1:5

# clients(i)=rho(i)/(1-rho(i));

# endfor

# else

# for i=1:5

# clients(i)=0;

# endfor

# endif

# endfunction

# Οι εντάσεις των φορτίων και ο μέσος χρόνος καθυστέρησης από άκρο σε άκρο είναι τα παρακάτω:

# 

# Στενωπός του δικτύου είναι η ουρά 1, και το καλύτερο είναι το παρακάτω:

# 

# Το ζητούμενο διάγραμμα είναι το παρακάτω:

# 

# Παράρτημα (κώδικας Lab5.m)

clc;

clear all;

close all;

##### NETWORK WITH ALTERNATIVE ROUTING #####

#2#

lambda = 10\*10^3;

b\_1= 1875000;

b\_2= 1500000;

mu\_1=b\_1/128;

mu\_2=b\_2/128;

i=1;

best\_a=0;

lowest\_E=9999999;

for a=0.001:0.001:0.999;

E(i)=(1/(mu\_1-a\*lambda))\*a + (1/(mu\_2-(1-a)\*lambda))\*(1-a);

if E(i)<lowest\_E;

lowest\_E = E(i);

best\_a=a;

endif

i++;

endfor

#figure(1);

#plot(0.001:0.001:0.999,E);

#xlabel("Probability a");

#ylabel("Average Waiting Time");

#title("Average Waiting Time");

#display(lowest\_E);

#display(best\_a);

##### OPEN QUEUING SYSTEM NETWORK #####

#2#

function [rho ergodicity] = intensities(lambda, mu)

rho(1)=lambda(1)/mu(1);

rho(2)=(lambda(2)+(2/7)\*lambda(1))/mu(2);

rho(3)=((4/7)\*lambda(1))/mu(3);

rho(4)=((3/7)\*lambda(1))/mu(4);

rho(5)=(lambda(2)+(4/7)\*lambda(1))/mu(5);

ergodicity=1;

for i=1:5

if rho(i)>1

ergodicity=0;

endif

endfor

display(rho);

endfunction

#3#

function clients = mean\_clients(lambda, mu)

[rho erg] = intensities(lambda,mu);

if erg == 1

for i=1:5

clients(i)=rho(i)/(1-rho(i));

endfor

else

for i=1:5

clients(i)=0;

endfor

endif

endfunction

#4#

lambda = [4 1];

mu = [6 5 8 7 6];

[rho erg] = intensities(lambda,mu);

mean\_clients\_ = mean\_clients(lambda,mu);

Average\_Time = (mean\_clients\_(1)+mean\_clients\_(2)+mean\_clients\_(3)+mean\_clients\_(4)+mean\_clients\_(5))/(lambda(1)+lambda(2));

display(Average\_Time);

#5#

mu = [6 5 8 7 6];

best\_lambda=99;

lambda\_1 = 4;

while 1

lambda=[lambda\_1 1];

[rho erg]=intensities(lambda,mu);

if erg == 0

best\_lambda = lambda\_1;

break;

endif

lambda\_1 = lambda\_1 + 0.01;

endwhile

display(best\_lambda);

#6#

mu = [6 5 8 7 6];

a = 1;

mean\_clients\_ = zeros(1, 5);

for i = 0.1:0.01:0.99

list=[best\_lambda\*i 1];

mean\_clients\_ = mean\_clients(list,mu);

Average\_Time\_(a) = (mean\_clients\_(1)+mean\_clients\_(2)+mean\_clients\_(3)+mean\_clients\_(4)+mean\_clients\_(5))/(lambda(1)+lambda(2));

a++;

endfor

figure(2)

plot(0.1:0.01:0.99,Average\_Time\_);

title("Average Waiting Time");

xlabel("Percentage of lambda 1");

ylabel("Average Waiting Time");