Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών
Συστήματα Αναμονής - Ροή Δ - 6ο Εξάμηνο
5η Εργαστηριακή Άσκηση
Θωμάς Πετρόπουλος - el18915

Δίκτυο με εναλλακτική δρομολόγηση

(1) Για να μοντελοποιηθούν οι σύνδεσμοι ως M/M/1 ουρές θα πρέπει να ισχύει:

1) Η ροή λ να είναι μια κατανομή Poisson.
2) Τα μεγέθη των πακέτων να είναι κατανεμημένα εκθετικά.

Έχουμε ότι : $\lambda 1 = \alpha\lambda$ και $\lambda 2=(1-\alpha)\lambda$ και $\mu 1=C1/E(L)$, $\mu 2=C2/E(L)$.
Όπου C1= 15*10^6 , C2=12*10^6 και E(L) = 128*8.
Επομένως : $\mu 1$=14648,43 Hz και $\mu 2$=11718,75 Hz
Και προκύπτει ότι : $\rho 1=0{,}682\alpha <1$ και $\rho 2=0{,}682(1-\alpha) <1$.

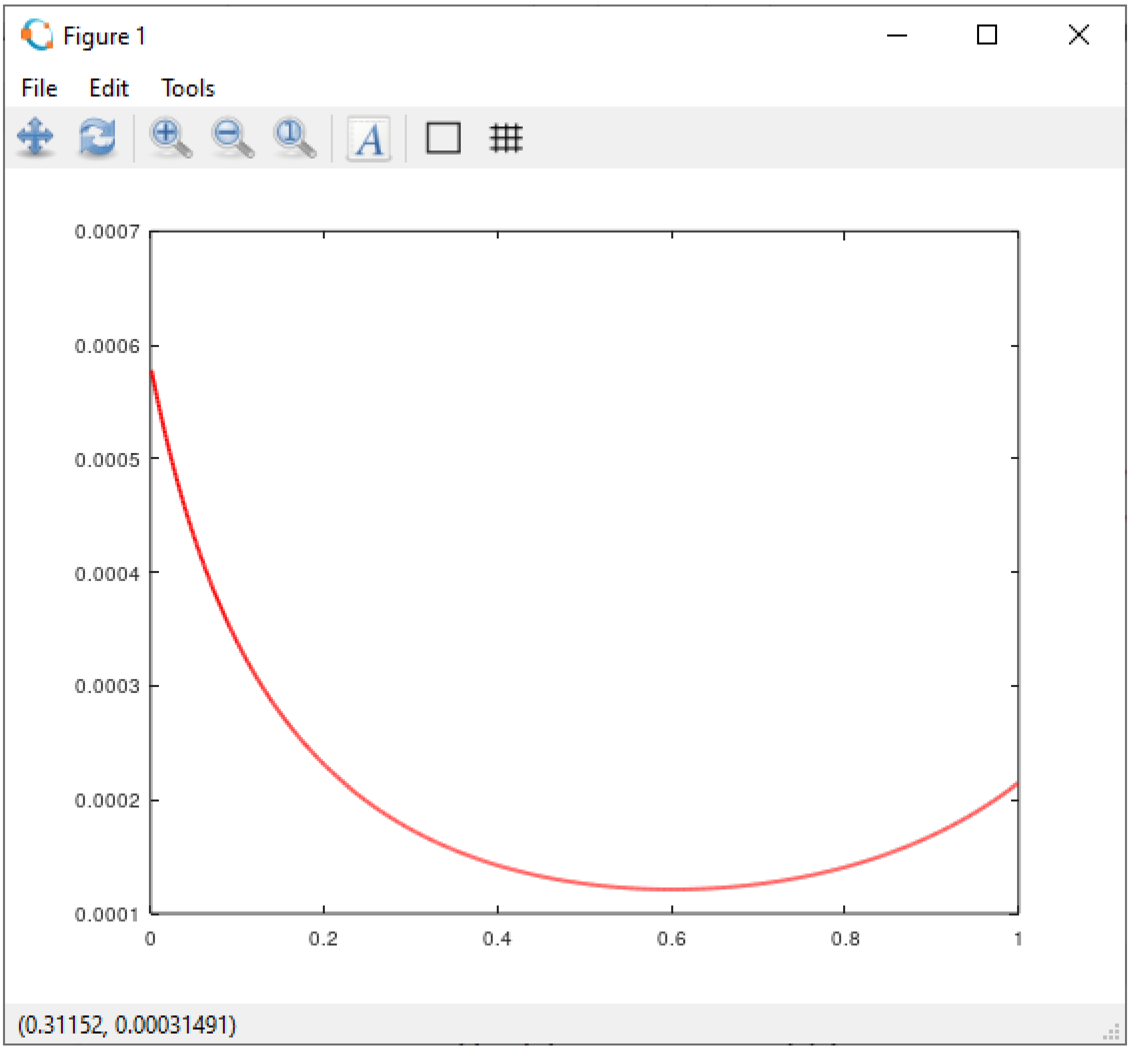
(2) Από το θεώρημα του Little ξέρουμε ότι :
$E(T)=E(n)/\gamma = E(n)/\lambda$ στο σύστημά μας.
Και το E(n) είναι το άθροισμα των δύο κλάδων.

Ο κώδικας σε Octave είναι ο παρακάτω:

5h.m

```
pkg load queueing
clc;
clear all;
close all;
a= 0.001:0.001:0.999;
lamda=10000;
mu1=(15*10^6)/(128*8);
mu2=(12*10^6)/(128*8);
lamda1=a.*lamda;
lamda2=(1-a).*lamda;
[U1 R1 Q1 X1 P1]= qsmm1(lamda1,mu1);
[U2 R2 Q2 X2 P2]= qsmm1(lamda2,mu2);
R=a.*R1 + (1-a).*R2;
figure(1);
plot(a,R,'r',"linewidth",2);
[minR,position] = min(R);
display(minR);
display(position);
clc;
clear all;
close all;
exit
```

Το διάγραμμα είναι το εξής :



Και ο ελάχιστος χρόνος καθυστέρησης E(T)= 0,00012120s επιτυγχάνεται για α= 0,601.

<u>Ανοιχτό δίκτυο ουρών αναμονής</u>

(1) Σύμφωνα με το θεώρημα Jackson , οι απαραίτητες παραδοχές είναι:

1) Οι αφίξεις να ακολουθούν ανεξάρτητη κατανομή Poisson.
2) Οι θάνατοι να ακολουθούν ανεξάρτητη εκθετική κατανομή.
3) Στο σύστημα κάθε διαφοροποίηση να είναι στοχαστική.
4) Οι χρόνοι εξυπηρέτησης να είναι χωρίς μνήμη , ούτως ώστε να ισχύει ο νόμος του Kleinrock.

(2) $Q1= \lambda 1/\mu 1$.

$Q2= (\lambda 2 + 2/7\lambda 1)/\mu 2$

$Q3= 4\lambda 1/7\mu 3$

$Q4= 3\lambda 1/7\mu 4$

$Q5= (4/7\lambda 1 + \lambda 2)/\mu 5$

Ο κώδικας σε Octave είναι ο εξής:

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

(3)

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

(4)

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

(5)
Παρατηρούμε ότι η Q1 είναι η στενωπός , αφού δέχεται το μεγαλύτερο φορτίο.
Επομώνς λ1,max = 1*μ1=6.

(6)
Ο κώδικας είναι :

```
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);
saveas (1, "figure2.png");
```

Και το γράφημα που προκύπτει :