ΣΥΣΤΗΜΑΤΑ ΑΝΑΜΟΝΗΣ

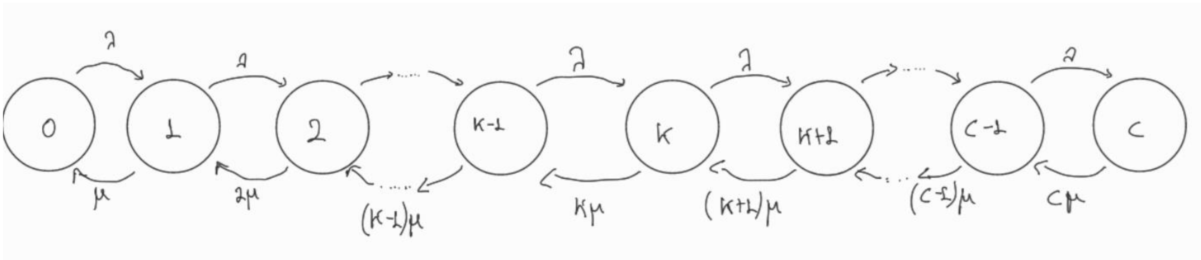
ΝΤΟΝΤΟΡΟΣ ΗΛΙΑΣ

el19206

ΑΣΚΗΣΗ 1η

Ερώτημα 1ο

Το διάγραμμα μεταβάσεων του συστήματος είναι το παρακάτω :



Από τις εξισώσεις ισορροπίας έχουμε :

$$\begin{cases} P_k - \left(\dfrac{\lambda}{k\mu}\right) \cdot P_{k-1}, k - 1,2,\ldots,c \\ P_0 + P_1 + \cdots P_{c-1} + P_c = 1 \end{cases}$$

Από την πρώτη αναδρομικά έχουμε :

$$P_k = \left(\frac{\lambda}{k\mu}\right) \cdot P_{k-1} = \left(\frac{\lambda}{k\mu}\right) \cdot \left[\left(\frac{\lambda}{(k-1)\mu}\right) \cdot P_{k-2}\right] = \frac{\lambda^2}{k(k-1)\mu^2} P_{k-2} = \cdots = \frac{\lambda^k}{k!\,\mu^k} P_0$$

Με αντικατάσταση έχουμε :

$$P_0 + \frac{\lambda}{\mu} P_0 + \cdots + \frac{\lambda^{c-1}}{(c-1)!\,\mu^{c-1}} P_0 + \frac{\lambda^c}{c!\,\mu^c} P_0 = 1 \Rightarrow$$

$$P_0 \left[\frac{\lambda^0}{0!\,\mu^0} + \frac{\lambda^1}{1!\,\mu^1} + \cdots + \frac{\lambda^{c-1}}{(c-1)!\,\mu^{c-1}} + \frac{\lambda^c}{c!\,\mu^c}\right] = 1 \Rightarrow$$

$$P_0 \sum_{k=0}^{c} \frac{\rho^k}{k!} = 1 \Rightarrow$$

$$P_0 = \frac{1}{\sum_{k=0}^{c} \frac{\rho^k}{k!}}$$

Η πιθανότητα απόρριψης πελάτη είναι η πιθανότητα το σύστημα να βρίσκεται στην κατάσταση k :

$$P_k = P_0 \frac{\rho^k}{k!} = \frac{\frac{\rho^k}{k!}}{\sum_{k=0}^{c} \frac{\rho^k}{k!}}$$

Για k = c προκύπτει ο τυπος Erlang-B :

$$P_{\text{blocking}} = B(\rho, c) = \frac{\frac{\rho^c}{c!}}{\sum_{k=0}^{c} \frac{\rho^k}{k!}}$$

Για να υπολογίσουμε τον μέσο αριθμό απωλειών πελατών πολλαπλασιάζουμε την πιθανότητα απόρριψης με τον ρυθμό άφιξης πελατών :

$$\lambda - \gamma = \lambda - \lambda\left(1 - P_{\text{blocking}}\right) = \lambda * P_{\text{blocking}} = \lambda * \frac{\frac{\rho^c}{c!}}{\sum_{k=0}^{c} \frac{\rho^k}{k!}}$$

O κώδικας στο octave για την υλοποίηση της συνάρτησης είναι :

```
function p = erlangb_factorial (r,c)
  s = 0;
  for k = 0:1:c
    s = s + (power(r,k)/factorial(k));
  endfor
  p = (power(r,c)/factorial(c))/s;
endfunction
```

και τα αποτελέσματα είναι ίδια με την έτοιμη συνάρτηση του octave :

```
Erlangb_factorial(9,9) =
0.2243
Erlangb(9,9) =
0.2243
```

Ερώτημα 2ο

Η συνάρτηση στο octave είναι :

```
function p = erlangb_iterative (r,c)
  p = 1;
  for i=0:1:c
    p = ((r*p)/((r*p)+i));
  endfor
endfunction
```

και τα αποτελέσματα που βγάζει είναι ίδια με το παραπάνω :

```
1.  Erlangb_iterative(9,9) =
2.  0.2243
```

Ερώτημα 3ο

Τα αποτελέσματα που παίρνουμε είναι :

```
1.  Erlangb_factorial(1024,1024) =
2.  NaN
3.  Erlangb_iterative(1024,1024) =
4.  0.024524
```

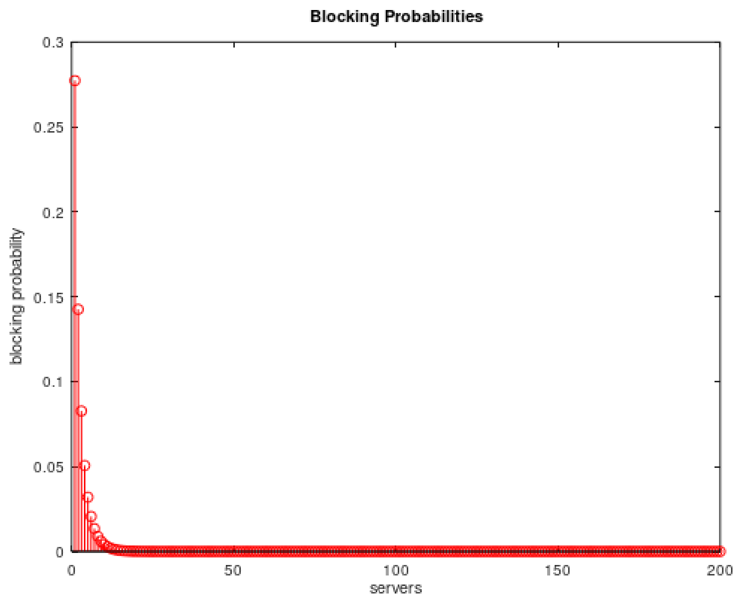
Η erlangb_factorial προσπαθεί να υπολογίσει το 1024! το οποίο είναι υπερβολικά μεγάλος αριθμός οπότε δεν μπορεί να βρει τελικό αποτέλεσμα.

Ερώτημα 4ο

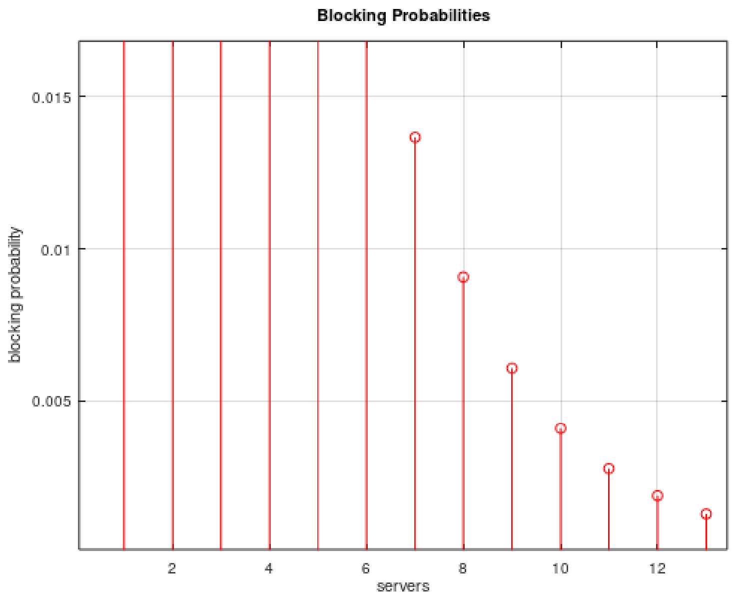
a) Έχοντας ως πρότυπο τον πιο απαιτητικό χρήστη τότε η συνολική κυκλοφοριακή ένταση του δικτύου είναι :

$$\rho = \frac{200 \cdot 23}{60} \approx 76{,}667 \text{ Erlangs}$$

b) Το διάγραμμα πιθανότητα απόρριψης πελάτη ως προς το αριθμό τηλεφωνικών γραμμών είναι :



c) Αν μεγεθύνουμε το παραπάνω διάγραμμα βλέπουμε ότι για να έχουμε πιθανότητα απόρριψης μικρότερη από 1% πρέπει ο αριθμός τηλεφωνικών γραμμών να είναι μεγαλύτερος η ίσος με 8.



Ο κώδικας για ολόκληρη την άσκηση 1 :

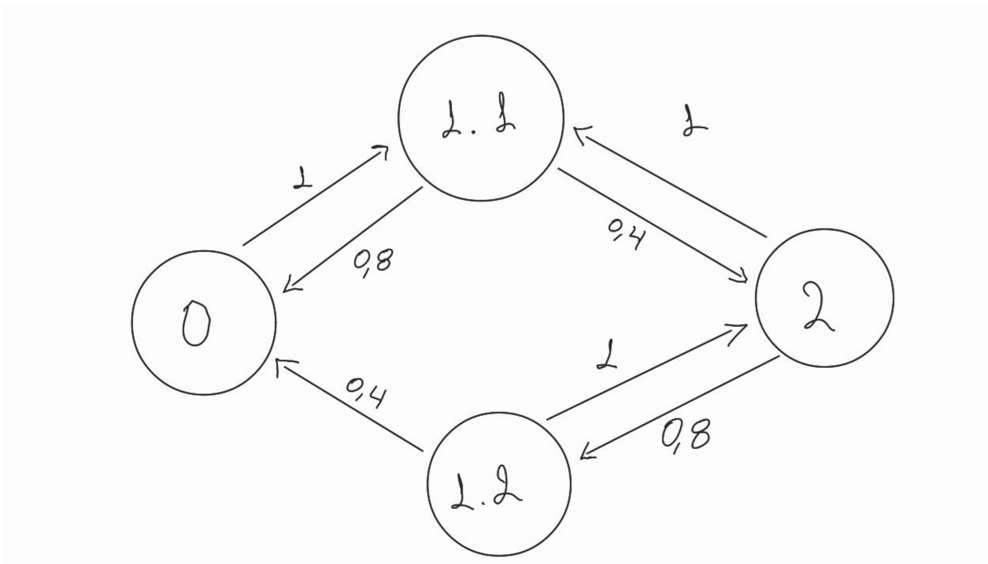
```
# Exercise 1
clc;
clear all;
close all;
pkg load queueing;

% r = lamda / mu
% c: number of servers

function p = erlangb_factorial (r,c)
  s = 0;
  for k = 0:1:c
    s = s + (power(r,k)/factorial(k));
  endfor
  p = (power(r,c)/factorial(c))/s;
endfunction


function p = erlangb_iterative (r,c)
  p = 1;
  for i=0:1:c
    p = ((r*p)/((r*p)+i));
  endfor
endfunction

display("Erlangb_factorial(9,9) =");
disp(erlangb_factorial(9,9));

display("Erlangb(9,9) =");
disp(erlangb(9,9));

```

```
display("Erlangb_iterative(9,9) =");
disp(erlangb_iterative(9,9));

display("Erlangb_factorial(1024,1024) =");
disp(erlangb_factorial(1024,1024));

display("Erlangb_iterative(1024,1024) =");
disp(erlangb_iterative(1024,1024));

P = zeros(0,200);

for i = 1:1:200
  P(i) = erlangb_iterative (i*(23/60),i)
endfor

figure(1);
stem(P,'b',"linewidth",0.4);
title("Blocking Probabilities")
xlabel("Servers");
ylabel("Blocking probability");
```

ΑΣΚΗΣΗ 2η

Ερώτημα 1ο

Το διάγραμμα ρυθμών μεταβάσεων του συστήματος στην κατάσταση ισορροπίας είναι :



a) Για το σύστημα μας ισχύουν οι παρακάτω εξισώσεις :

$$\begin{cases} \lambda \cdot P_0 = \mu_1 P_{11} + \mu_2 \cdot P_{12} \\ (\lambda + \mu_1) \cdot P_{11} = p \cdot \lambda \cdot P_0 + \mu_2 \cdot P_2 \\ (\lambda + \mu_2) \cdot P_{12} = (1-p) \cdot \lambda \cdot P_0 + \mu_1 \cdot P_2 \\ P_0 + P_{11} + P_{12} + P_2 = 1 \end{cases}$$

Αντικαθιστώντας τις τιμές : $\mu 1 = 0.8, \mu 2 = 0.4, \lambda = 1 \; και \; p = 1$ προκύπτουν οι εργοδικές πιθανότητες :

$P0 = 0.24951$
$P11 = 0.21442$
$P12 = 0.19493$
$P2 = 0.34113$

b) Η πιθανότητα απόρριψης πελάτη από το σύστημα είναι :
$P2 = 0.34113$

c) Ο μεσος αριθμός πελατών στο σύστημα είναι :

$$\sum_{k=0}^{2} \kappa * P(k) = 0 \cdot \mathrm{P0} + 1 \cdot \mathrm{P11} + 1 \cdot \mathrm{P12} + 2 \cdot \mathrm{P2} = 1.0917$$

Ερώτημα 2ο

a) Τα thresholds που χρησιμοποιήσαμε είναι :

```
1. threshold_0 = lamda/1;
2. threshold_1a = lamda/(lamda+m1);
3. threshold_1b = lamda/(lamda+m2);
4. threshold_2_first = lamda/(lamda+m2+m1);
5. threshold_2_second = (m1+lamda)/(lamda+m2+m1);
```

<u>threshold 0</u>: Από την κατάσταση 0 είναι δυνατή να πραγματοποιηθεί μόνο άφιξη επομένως θα ισούται με 1.

<u>threshold 1a</u>: Από την κατάσταση 1_1 μπορούμε να έχουμε είτε αναχώρηση είτε άφιξη και να μεταβούμε στην κατάσταση 0 με ρυθμό μ1 είτε στην κατάσταση 2 με ρυθμό λ αντίστοιχα. Επομένως ο παρονομαστής θα είναι λ + μ1 και ο αριθμητής λ.

<u>threshold 1b</u>: Από την κατάσταση 1_2 μπορούμε να έχουμε είτε αναχώρηση είτε άφιξη και να μεταβούμε στην κατάσταση 0 με ρυθμό μ2 είτε στην κατάσταση 2 με ρυθμό λ αντίστοιχα. Επομένως ο παρονομαστής θα είναι λ + μ2 και ο αριθμητής λ.

<u>threshold 2 first</u>: Από την κατάσταση 2 μπορούμε να έχουμε άφιξη και απόρριψη του πελάτη με ρυθμό λ, επομένως ο αριθμητής του threshold_2_first θα είναι λ.

<u>threshold 2 second</u>: Από την κατάσταση 2 μπορούμε να έχουμε αναχώρηση και να μεταβούμε στην κατάσταση 1_2 με ρυθμό μ1, επομένως ο αριθμητής του threshold_2_second θα είναι λ+μ1.

Για τα δυο τελευταία μπορούμε να έχουμε αναχώρηση και να πάμε στην κατάσταση 1_1 με ρυθμό μ2, άρα ο παρονομαστής θα είναι λ + μ1 + μ2

b) Το κριτήριο σύγκλισης είναι η διαφορά μεταξύ δυο διαδοχικών μέσων αριθμών πελατών να είναι κάτω από 0,001%

c) Οι πιθανότητες που υπολογίζει η προσομοίωση είναι :

```
0.2517
0.2168
0.1939
0.3375
```

Οι οποίες είναι ίσες με τις τιμές που υπολογίσαμε παραπάνω αλλά υπάρχουν μικρές αποκλίσεις

Ο κώδικας για ολόκληρη την άσκηση 2 :

```
# Exercise 2

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

lamda = 1;
m1 = 0.8;
m2 = 0.4;

threshold_0 = lamda/1;
threshold_1a = lamda/(lamda+m1);
threshold_1b = lamda/(lamda+m2);
threshold_2_first = lamda/(lamda+m2+m1);
threshold_2_second = (m1+lamda)/(lamda+m2+m1);

current_state = 0;
arrivals = zeros(1,4);
total_arrivals = 0;
maximum_state_capacity = 2;
previous_mean_delay = 0;
delay_counter = 0;
time = 0;

while 1 > 0
  time = time + 1;

  if mod(time,1000) == 0
    for i=1:1:4
      P(i) = arrivals(i)/total_arrivals;
    endfor

    delay_counter = delay_counter + 1;
```

```

    mean_delay = 0*P(1) + 1*P(2) + 1*P(3) + 2*P(4);

    delay_table(delay_counter) = mean_delay;

    if abs(mean_delay - previous_mean_delay) < 0.00001
       break;
    endif
    previous_mean_delay = mean_delay;
  endif

  random_number = rand(1);

  if current_state == 0
    if random_number < threshold_0
      current_state = 1;
      arrivals(1) = arrivals(1) + 1;
      total_arrivals = total_arrivals + 1;
    endif
  elseif current_state == 1
    if random_number < threshold_1a
      current_state = 3;
      arrivals(2) = arrivals(2) + 1;
      total_arrivals = total_arrivals + 1;
    else
      current_state = 0;
    endif
  elseif current_state == 2
    if random_number < threshold_1b
      current_state = 3;
      arrivals(3) = arrivals(3) + 1;
      total_arrivals = total_arrivals + 1;
    else
      current_state = 0;
    endif
  else
      if random_number < threshold_2_first
        arrivals(4) = arrivals(4) + 1;
        total_arrivals = total_arrivals + 1;
      elseif random_number < threshold_2_second
        current_state = 2;
      else
        current_state = 1;
      endif
   endif

endwhile

display(P(1));
display(P(2));
display(P(3));
display(P(4));
```