ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ & ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
Τομέας Επικοινωνιών, Ηλεκτρονικής & Συστημάτων Πληροφορικής
Εργαστήριο Διαχείρησης και Βέλτιστου Σχεδιασμού Δικτύων Τηλεματικής – NETMODE

Πεγειώτη Νάταλυ
Α.Μ: 03117707

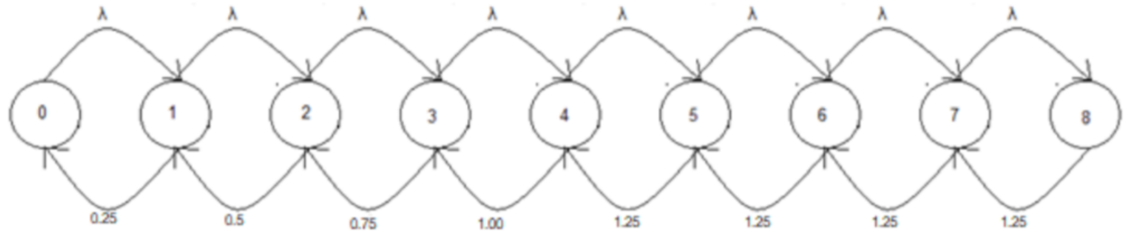
Συστήματα Αναμονής (Queuing Systems)

<u>4η Ομάδα Ασκήσεων</u>

Σύστημα M/M/N/K (call center)

Πρόκειται για την μελέτη ενός συστήματος call center με 5 εξυπηρετητές και μέγιστη χωρητικότητα 8 πελάτες.
Οι αφίξεις στο σύστημα ακολουθούν την κατανομή Poisson με μέσο ρυθμό αφίξεων λ πελάτες/min, ενώ οι εξυπηρετήσεις ακολουθούν την εκθετική κατανομή με μέση διάρκεια 4 min/πελάτη.

(1) Για το σύστημα M/M/5/8 προκύπτει το ακόλουθο διάγραμμα ρυθμών μεταβάσεων:



(2) *Ο κώδικας που χρησιμοποιήθηκε παρατίθεται στο αρχείο qsmm58.m (βλ. Παράρτημα Κώδικα)*

(α) λ = 1/4 πελάτες/min

```
Ergodic probability of state 0 is 0.36782
Ergodic probability of state 1 is 0.36782
Ergodic probability of state 2 is 0.18391
Ergodic probability of state 3 is 0.061303
Ergodic probability of state 4 is 0.015326
Ergodic probability of state 5 is 0.0030652
Ergodic probability of state 6 is 0.00061303
Ergodic probability of state 7 is 0.00012261
Ergodic probability of state 8 is 2.4521e-005
```

(β) λ = 1 πελάτης/min

```
Ergodic probability of state 0 is 0.0168
Ergodic probability of state 1 is 0.0672
Ergodic probability of state 2 is 0.1344
Ergodic probability of state 3 is 0.1792
Ergodic probability of state 4 is 0.1792
Ergodic probability of state 5 is 0.14336
Ergodic probability of state 6 is 0.11469
Ergodic probability of state 7 is 0.091751
Ergodic probability of state 8 is 0.0734
```

(3) Η πιθανότητα ένας πελάτης να χρειαστεί να παραμείνει στην ουρά του συστήματος πριν εξυπηρετηθεί υπολογίζεται ως εξής:

P[Αναμονής] = P[5] + P[6] + P[7]+P[8]

Για λ = ¼ πελάτες/min
P[Αναμονής] = 0.0038253, όπως υπολογίζεται από τις τιμές που φαίνονται στο προηγούμενο ερώτημα.
Η συνάρτηση erlangc δίνει P[Αναμονής] = 0.0038314

Για λ = 1 πελάτης/min
P[Αναμονής] = 0.423201, όπως υπολογίζεται από τις τιμές που φαίνονται στο προηγούμενο ερώτημα.
Η συνάρτηση erlangc δίνει P[Αναμονής] = 0.55441

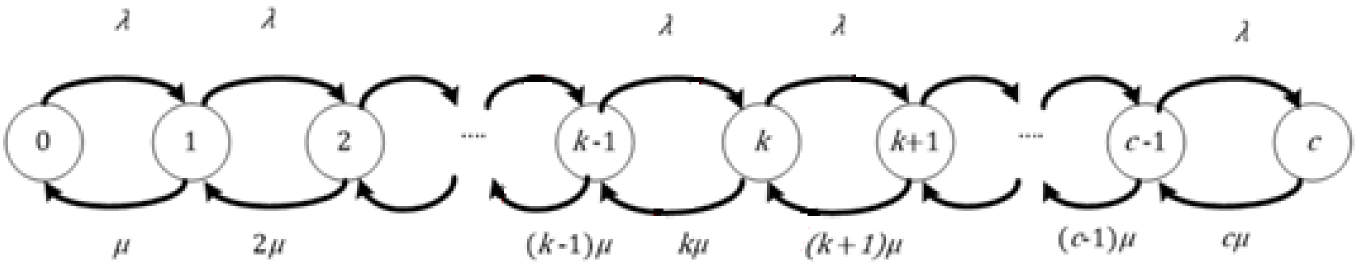
Παρατηρούμε, πως αν και η συνάρτηση erlangc αφορά συστήματα άπειρης χωρητικότητας, δίνει εξαιρετικά ακριβή αποτελέσματα (με πολύ μικρή απόκλιση) και για συστήματα πεπερασμένης χωρητικότητας.
Παρατηρείται βέβαια ότι αυτό δεν ισχύει για μεγάλα φορτία ρ, καθώς γίνεται απόρριψη με μεγαλύτερη πιθανότητα και αυτό μειώνει την αναμονή κάποιου μελλοντικού πελάτη.

Ανάλυση και σχεδιασμός τηλεφωνικού κέντρου

Ο κώδικας που χρησιμοποιήθηκε παρατίθεται στο αρχείο tilefwniko_kentro.m (βλ. Παράρτημα Κώδικα)

Πρόκειται για την ανάλυση και τον σχεδιασμό ενός τηλεφωνικού κέντρου, που μοντελοποιείται ως μια ουρά M/M/c/c.
Οι αφίξεις στο σύστημα ακολουθούν την κατανομή Poisson με ομοιόμορφο μέσο ρυθμό λ και οι εξυπηρετήσεις ακολουθούν εκθετική κατανομή με ομοιόμορφο μέσο ρυθμό μ.

(1) Για το σύστημα M/M/c/c προκύπτει το ακόλουθο διάγραμμα ρυθμών μεταβάσεων:



Παίρνοντας τις εξισώσεις ισορροπίας του τυχαίου κόμβου k προκύπτει:

$$k\mu P_k=\lambda P_{k-1} \rightarrow P_k=\frac{\lambda}{\mu k}P_{k-1}=\frac{1}{k}\rho P_{k-1}=\left(\frac{\rho^k}{k!}\right)P_0$$

Από τη συνθήκη κανονικοποίησης $P_0+P_1+\ldots\ldots+P_c=1$ προκύπτει $P_0=\frac{1}{\sum_{k=0}^{c}\frac{\rho^k}{k!}}$

Από τις δύο πιο πάνω σχέσεις υπολογίζεται $P_c = P[blocking] = B(\rho, c) \times \frac{1}{\sum_{k=0}^{c} \frac{\rho^k}{k!}}$

Ο μέσος ρυθμός απωλειών δίνεται από τη σχέση:

$$\lambda - \gamma = \lambda \times P[blocking] = \frac{\rho^c}{c!} \times \frac{\lambda}{\sum_{k=0}^{c} \frac{\rho^k}{k!}}$$

(2) Ισχύει ότι: $P_c = P[blocking] = B(\rho, c) \times \frac{1}{\sum_{k=0}^{c} \frac{\rho^k}{k!}} = \frac{1}{\sum_{k=0}^{c} \frac{c!\rho^k}{\rho^c k!}} = \frac{1}{\sum_{k=0}^{c} \frac{c!}{\rho^{c-k} k!}}$

Για c+1:

$$B(\rho, c+1) = \frac{1}{\sum_{k=0}^{c+1} \frac{(c+1)!}{\rho^{c+1-k} k!}} = \frac{1}{1 + \sum_{k=0}^{c} \frac{(c+1)!}{\rho^{c+1-k} k!}} = \frac{1}{1 + \frac{c+1}{\rho} \sum_{k=0}^{c} \frac{c!}{\rho^{c-k} k!}} = \frac{1}{1 + \frac{c+1}{\rho} \times \frac{1}{B(\rho, c)}} = \frac{\rho B(\rho, c)}{\rho B(\rho, c) + c + 1}$$

Θέτοντας n = c+1 προκύπτει η ζητούμενη σχέση:

$$B(\rho, n) = \frac{\rho B(\rho, n-1)}{\rho B(\rho, n-1) + n}$$

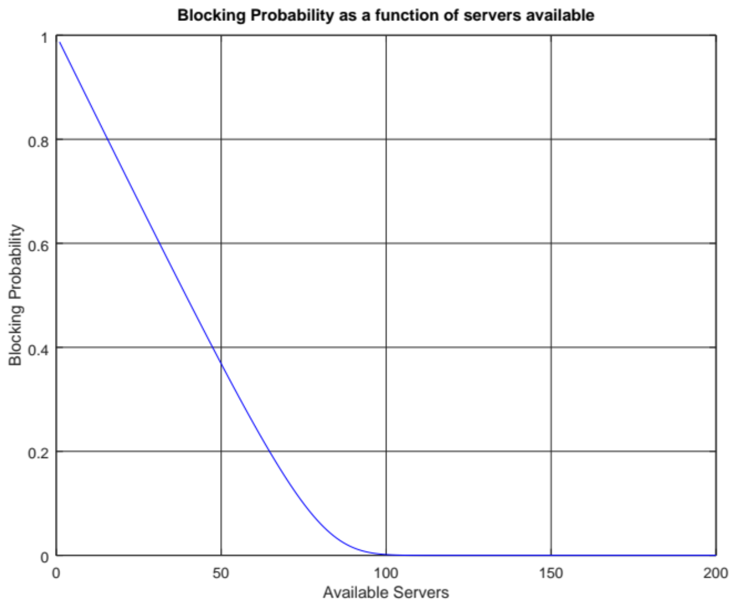
Από την προηγούμενη σχέση προκύπτει και η “βάση” της αναδρομής $B(\rho, 0) = \frac{\rho^0}{0!} \times \frac{1}{\sum_{k=0}^{0} \frac{\rho^k}{k!}} = 1$

(3) Παρατηρούμε πως η erlangb_interactive δίνει όντως τη σωστή τιμή 0.024524, ενώ η erlangb_factorial δίνει τιμή NaN.
Αυτό προφανώς συμβαίνει γιατί στην προσπάθεια υπολογισμού της τελικής τιμής η συνάρτηση erlangb_factorial προσπαθεί να υπολογίσει πολύ μεγάλες τιμές, (όπως το 1024! στην περίπτωση μας), με αποτέλεσμα να δημιουργείται υπερχείλιση και αδυναμία υπολογισμού κάποιου αποτελέσματος.

(4)

(α) ρ = λμ = 2360 = 0.38333 erlangs
Το σύστημα μπορεί να μοντελοπιηθεί ως εξής:
λ = 1 κλήση/ώρα με διάρκεια 1/μ = 23/60 ώρες.
Συνεπώς η συνολική ένταση του φορτίου για όλες τις γραμμές θα είναι
200ρ = 76.66667 erlangs.

(β) Το διάγραμμα της πιθανότητας απόρριψης πελάτη από το σύστημα ως προς τον αριθμό των τηλεφωνικών γραμμών σχεδιάζεται με τη βοήθεια της Octave και παρουσιάζεται ακολούθως:



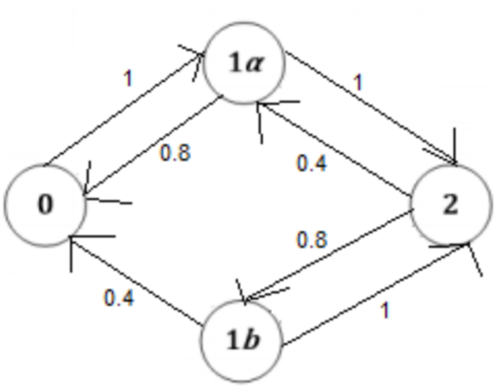
(γ) Ο κατάλληλος αριθμός τηλεφωνικών γραμμών ώστε η πιθανότητα απόρριψης τηλεφωνικής κλήσης να είναι μικρότερη από 1% υπολογίστηκε επίοσης στην Octave.

```
We want 93 lines for a probability of 0.008368 which is less than 1%.
The probability for 92 lines is 0.010236
```

Σύστημα εξυπηρέτησης με δύο ανόμοιους εξυπηρετητές

Πρόκειται για ένα σύστημα εξυπηρέτησης που αποτελείται από δύο εξυπηρετητές 1 και 2 χωρίς δυνατότητα αναμονής πελατών. Όταν και οι δύο εξυπηρετητές είναι διαθέσιμοι, ένας πελάτης δρομολογείται πάντα στον εξυπηρετητή 1. Σε περίπτωση που ο εξυπηρετητής 1 δεν είναι διαθέσιμος, ένας νέος πελάτης δρομολογείται στον εξυπηρετητή 2. Σε περίπτωση που και οι δύο εξυπηρετητές δεν είναι διαθέσιμοι, ένας νέος πελάτης απορρίπτεται χωρίς επανάληψη προσπάθειας εξυπηρέτησης. Οι αφίξεις πελατών στο σύστημα ακολουθούν την κατανομή Poisson με μέσο ρυθμό λ = 1 πελάτες/sec. Οι εξυπηρετήσεις είναι εκθετικά κατανεμημένες με μέσους χρόνους εξυπηρέτησης $1/\mu_1 = 1.25$ sec και $1/\mu_2 = 2.5$ sec αντίστοιχα.

(1) Το διάγραμμα ρυθμών μεταβάσεων του συστήματος είναι το ακόλουθο:



Εξισώσεις Ισορροπίας:
$1 \times P_0 = 0.8 \times P_{1a} + 0.4 \times P_{1b}$
$1.8 \times P_{1a} = 0.4 \times P_2 + 1 \times P_0$
$1.4 \times P_{1b} = 0.8 \times P_2$
και η συνθήκη κανονικοποίησης:
$P_0 + P_{1a} + P_{1b} + P_2 = 1$

Λύνοντας τις εξισώσεις υπολογίζονται οι εργοδικές πιθανότητες του συστήματος:
$P_0 = 24.951\%$, $P_{1a} = 21.442\%$, $P_{1b} = 19.493\%$, $P_2 = 34.113\%$

Η πιθανότητα απόρριψης πελάτη από το σύστημα είναι: $P[\text{blocking}] = P_N = P_2 = 34.113\%$

Ο μέσος αριθμός των πελατών στο σύστημα δίνεται από τον τύπο
$E[n(t)] = 1 \times (0.21442 + 0.19493) + 2 \times 0.34113 = 1.09161$ πελάτες

(2) Τα κενά του προγράμματος συμπληρώνονται ως ακολούθως:

```
threshold_1a = lambda/(lambda+m1);
threshold_1b = lambda/(lambda+m2);
threshold_2_first = lambda/(lambda+m1+m2);
threshold_2_second = (lambda+m1)/(lambda+m1+m2);
```

Κριτήριο σύγκλισης της προσομοίωσης είναι ο μέσος αριθμός πελατών στο σύστημα, για δύο διαδοχικές μεταβάσεις, να μην έχει απόκλιση μεγαλύτερη του 0.00001.

Οι πιθανότητες που υπολογίζονται από το πρόγραμμα φαίνονται πιο κάτω και είναι σύμφωνες με τις αντίστοιχες πιθανότητες που υπολογίστηκαν στο προηγούμενο ερώτημα.

```
The ergodic probabilities of the system are (P0, P1a, P1b, P2):
     0.24719     0.20964     0.19913     0.34403
The blocking probabilityof the system is:
0.34403
The average number of clients in the system is:
1.0968
```

<u>Παράρτημα Κώδικα</u>

qsmm58.m

```
clc;
clear all;
close all;

lambda = 1/4;
m = 1/4;
states = 0:1:8;
initial_sate = [1, 0, 0, 0, 0, 0, 0, 0 , 0];   # initial state of the system

genniseis = [lambda, lambda, lambda, lambda, lambda, lambda, lambda, lambda];
thanatoi = [m, 2*m, 3*m, 4*m, 5*m, 5*m, 5*m, 5*m];

metavatikos = ctmcbd(genniseis, thanatoi);


ergodic_prob = ctmc(metavatikos);

format short g;
for i = 1:1:9
  message = cstrcat ("Ergodic probability of state ", num2str(i-1), " is ", num2str(ergodic_prob(i)));
  disp(message);
endfor

P_waiting = erlangc(lambda/m, 5);
disp(cstrcat("The possibility of every server being busy (for an infinite system) is ",
num2str(P_waiting)));
```

 # erlangb_factorial

```
function pblock = erlangb_factorial(rho, c)
  if ( nargin != 2 )
    print_usage();
  endif
  sum = 0;
  for k = 1:1:c+1
    sum = sum + (rho**(k-1))/(factorial(k-1));
  endfor

  pblock = (rho**c)/((factorial(c))*sum);
endfunction
```

erlangb_iterative

```
function pblock = erlangb_iterative(rho, c)
  if ( nargin != 2 )
    print_usage();
  endif
  results = zeros(1, c + 1);
  results(1) = 1;
  for i = 2:1:c+1
    results(i) = rho*results(i-1) / (rho*results(i-1) + i-1);
  endfor
  pblock = results(c + 1);
endfunction
```

tilefwniko_kentro.m

```
clc;
clear all;
close all;

blocking_prob = zeros(1, 200);
rho = 76.66667;
c = 1:1:200;
```

 # erlangb_factorial

```
function pblock = erlangb_factorial(rho, c)
  if ( nargin != 2 )
    print_usage();
  endif
  sum = 0;
  for k = 1:1:c+1
    sum = sum + (rho**(k-1))/(factorial(k-1));
  endfor

  pblock = (rho**c)/((factorial(c))*sum);
endfunction
```

 # erlangb_iterative

```
function pblock = erlangb_iterative(rho, c)
  if ( nargin != 2 )
    print_usage();
  endif
  results = zeros(1, c + 1);
  results(1) = 1;
  for i = 2:1:c+1
    results(i) = rho*results(i-1) / (rho*results(i-1) + i-1);
  endfor
  pblock = results(c + 1);
endfunction
#calculating the blocking probabilities
for i = 1:1:200;
  blocking_prob(i) = erlangb_iterative(rho, i);
endfor

#finding the first probability less than 1%
for i = 1:1:200
  if blocking_prob(i) < 0.01
    this_value = i;
    break;
  endif
endfor
```

```
disp(cstrcat("We want ", num2str(i), " lines for a probability of ", num2str(blocking_prob(i)), " which is less than 1%. " ));
disp(cstrcat("The probability for ", num2str(i-1), " lines is ", num2str(blocking_prob(i-1))));

figure(1);
plot(c, blocking_prob, '');
grid on;
title("Blocking Probability as a function of servers available");
xlabel("Available Servers");
ylabel("Blocking Probability");
```