

ΣΥΣΤΗΜΑΤΑ ΑΝΑΜΟΝΗΣ

5Η ΟΜΑΔΑ ΑΣΚΗΣΕΩΝ

Όνοματεπώνυμο: Δημήτριος Βασιλείου

A.M: el19830

Εξάμηνο: 6ο

Σχολή: ΗΜΜΥ

Δίκτυο με εναλλακτική δρομολόγηση:

1). Για να μπορούν οι γραμμές να μοντελοποιηθούν σαν M/M/1 ουρές θεωρούμε αρχικά η εισερχόμενη ροή πελατών στο σύστημα να είναι Poisson με μέσο ρυθμό λ . Όπως γνωρίζουμε η διάσπαση μιας διαδικασίας Poisson οδηγεί σε δύο νέες διαδικασίες Poisson και αφού γνωρίζουμε το ποσοστό διάσπασης των πακέτων πάνω στις 2 γραμμές, έχουμε δύο νέες διαδικασίες Poisson, μία με ρυθμό

$\lambda_1 = \alpha \cdot \lambda$ πάνω στην γραμμή 1, και μία με ρυθμό $\lambda_2 = (1 - \alpha) \cdot \lambda$ πάνω στην γραμμή 2. Επίσης, θεωρούμε ότι οι γραμμές 1, 2 έχουν μέσο ρυθμό εξυπηρέτησης εκθετικά κατανεμημένο με μέσο ρυθμό

μ_1, μ_2 αντίστοιχα.

Έχοντας θεωρήσει τις παραπάνω παραδοχές, θα υπολογίσουμε τα $\lambda_1, \lambda_2, \mu_1, \mu_2$.

- Για την γραμμή 1, έχουμε

$$\mu_1 = \frac{c_1}{128 \cdot 8} = \frac{15 \cdot 10^6}{1024} \approx 14.65 \cdot 10^3 \text{ πακέτα/sec} \quad \text{και} \quad \lambda_1 = \alpha \cdot 10 \cdot 10^3 \text{ πακέτα/sec}$$

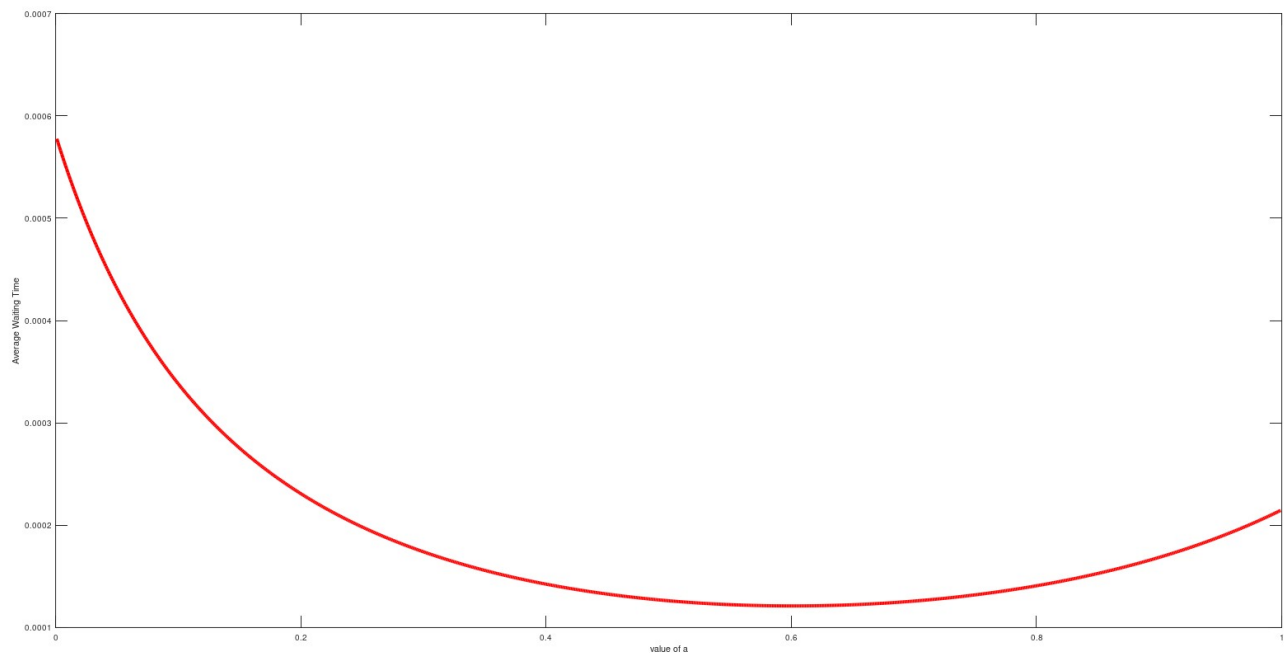
- Για την γραμμή 2 έχουμε

$$\mu_2 = \frac{c_2}{128 \cdot 8} = \frac{12 \cdot 10^6}{1024} \approx 11.72 \cdot 10^3 \text{ πακέτα/sec} \quad \text{και} \quad \lambda_2 = (1 - \alpha) \cdot 10 \cdot 10^3 \text{ πακέτα/sec}$$

2). Ο συνολικός μέσος χρόνος καθυστέρησης ενός πακέτου στο σύστημα ισούται με το άθροισμα των μέσων αριθμών πελατών σε κάθε γραμμή διαιρεμένο με την συνολική εισερχόμενη ροή. Δηλαδή θα είναι:

$$E[T] = \frac{E[n]}{\lambda} = \frac{E[n_1] + E[n_2]}{\lambda}.$$

Στη συνέχεια παραθέτουμε το διάγραμμα του μέσου χρόνου καθυστέρησης $E[T]$ συναρτήσει του a για ένα τυχαίο πακέτο στο σύστημα:



Χρησιμοποιήσαμε την συνάρτηση `qsmm1` του πακέτου `queueing` για να προσομοιώσουμε σαν M/M/1 ουρές τις δύο γραμμές. Ο κώδικας που χρησιμοποιήθηκε είναι ο ακόλουθος:

```
pkg load queueing;
clc;
clear all;

a = 0.001:0.001:0.999;
l = 10000;
l1 = a*l;
l2 = (1-a)*l;
m1 = 14650;
m2 = 11720;

[U1, R1, Q1, X1, p1] = qsmm1(l1, m1);
[U2, R2, Q2, X2, p2] = qsmm1(l2, m2);

averageNumberOfClients = Q1 + Q2;
averageWaitTime = averageNumberOfClients / l;

figure(1);
plot(a, averageWaitTime, "r", "linewidth", 2);
xlabel("value of a");
ylabel("Average Waiting Time");
```

Στη συνέχεια υπολογίζουμε με την βοήθεια του Octave την ελάχιστη τιμή του μέσου χρόνου καθυστέρησης καθώς και την τιμή του α για την οποία έχουμε ελάχιστο χρόνο:

```
minimumTime = 1.2118e-04  
0.6010
```

Ο κώδικας που χρησιμοποιήθηκε είναι ο ακόλουθος:

```
minimumTime = min(averageWaitTime);  
display(minimumTime);  
  
mina = find(averageWaitTime == minimumTime);  
display(mina*0.001);
```

Η συνάρτηση find, μας γυρνάει το κατά σειρά α για το οποίο επιτυγχάνεται η ελάχιστη τιμή του μέσου χρόνου καθυστέρησης. Επειδή το 1ο α για εμάς είναι 0.001 και αυξάνεται με το βήμα 0.001, πρέπει να πολλαπλασιάσουμε αυτή την τιμή με 0.001 για να πάρουμε την πραγματική τιμή του α .

Ανοιχτό δίκτυο ουρών αναμονής:

1). Για να μπορέσει το δίκτυο να μελετηθεί ως ένα ανοιχτό δίκτυο με το θεώρημα Jackson πρέπει να ισχύουν οι ακόλουθες παραδοχές:

- Πρέπει να έχουμε 5 δικτυακούς κόμβους εξυπηρέτησης κορμού όπου κάθε ουρά Q_i αποτελεί τέτοιο κόμβο με εκθετικά κατανεμημένο ρυθμό εξυπηρέτησης μ_i , όπου $i = 1, 2, 3, 4, 5$.
- Οι δικτυακοί κόμβοι εξυπηρέτησης Q_1, Q_2 , βλέπουμε ότι δέχονται άμεσα ροές από εξωτερικές πηγές με γ_1, γ_2 , αντίστοιχα. Οι ροές αυτές προορίζονται για εξωτερικούς προορισμούς άμεσα συνδεδεμένους με τους κόμβους Q_4, Q_5 . Οι ροές μεταξύ των κόμβων είναι ανεξάρτητες ροές Poisson μέσου ρυθμού γ_{sd} , όπου $s, d \in \{1, 2, \dots, 5\}$. Τέλος, η συνολική εξωγενής ροή Poisson

για κάθε κόμβο Q_s είναι $\gamma_s = \sum_{d=1, d \neq s}^M \gamma_{sd}, s, d \in 1, 2, \dots, 5$.

- Η εσωτερική δρομολόγηση γίνεται με τυχαίο τρόπο και η πιθανότητα δρομολόγησης πελάτη από τον κόμβο κορμού(ουρά) Q_i στο κόμβο Q_j είναι r_{ij} . Έτσι, θα έχουμε $r_{12} = 2/7, r_{13} = 4/7, r_{14} = 1/7, r_{35} = 1/2, r_{34} = 1/2$.
- Για έναν κόμβο εξυπηρέτησης Q_j , τον διαπερνούν ροές με συνολικό μέσο ρυθμό

$$\lambda_j = \gamma_j + \sum_{i=1, i \neq j}^M r_{ij} \lambda_i, j=1,2,3,4,5.$$

- Οι χρόνοι εξυπηρέτησης πελατών όπως διαπερνούν το δίκτυο δεν διατηρούν την τιμή τους(έλλειψη μνήμης) αλλά αποκτούν χρόνο εξυπηρέτησης ανάλογα με την κατανομή του κάθε εξυπηρετητή.

2). Για την ένταση φορτίου που δέχεται η κάθε ουρά του δικτύου γνωρίζουμε ότι ισχύει $\rho_i = \frac{\lambda_i}{\mu_i}$ με $i = 1, 2, 3, 4, 5$. Για να βρούμε τα λ_i για κάθε ουρά θα βασιστούμε στην 4η παραδοχή που κάναμε παραπάνω. Συνεπώς θα έχουμε:

$$\rho_1 = \frac{\lambda_1}{\mu_1}$$

$$\rho_2 = \frac{\lambda_2 + r_{12} \cdot \lambda_1}{\mu_2} = \frac{\lambda_2 + \frac{2}{7} \cdot \lambda_1}{\mu_2}$$

$$\rho_3 = \frac{r_{13} \cdot \lambda_1}{\mu_3} = \frac{\frac{4}{7} \cdot \lambda_1}{\mu_3}$$

$$\rho_4 = \frac{r_{14} \cdot \lambda_1 + r_{13} \cdot r_{34} \cdot \lambda_1}{\mu_4} = \frac{\frac{1}{7} \cdot \lambda_1 + \frac{4}{7} \cdot \left(\frac{1}{2}\right) \cdot \lambda_1}{\mu_4} = \frac{\frac{3}{7} \cdot \lambda_1}{\mu_4}$$

$$\rho_5 = \frac{\lambda_1 \cdot r_{12} + \lambda_1 \cdot r_{13} \cdot r_{35} + \lambda_2}{\mu_5} = \frac{\frac{2}{7} \cdot \lambda_1 + \frac{2}{7} \cdot \lambda_1 + \lambda_2}{\mu_5} = \frac{\frac{4}{7} \cdot \lambda_1 + \lambda_2}{\mu_5}$$

Θα υλοποιήσουμε την συνάρτηση **intensities** με το Octave. Σημειώνουμε πως για να είναι το σύστημά μας εργοδικό θα πρέπει $\rho_i < 1$ για $i = 1, 2, 3, 4, 5$. Ο κώδικας που χρησιμοποιήσαμε είναι ο ακόλουθος:

```
pkg load queueing
clc;
clear all;

function [erg, r1, r2, r3, r4, r5] = intensities (l1, l2, m1, m2, m3, m4, m5)
    r1 = (l1/m1);
    r2 = (l2 + (2/7)*l1) / m2;
    r3 = ((4/7)*l1) / m3;
    r4 = ((3/7)*l1) / m4;
    r5 = ((4/7)*l1 + l2) / m5;
    printf("r1 = %d\n", r1);
    printf("r2 = %d\n", r2);
    printf("r3 = %d\n", r3);
    printf("r4 = %d\n", r4);
    printf("r5 = %d\n", r5);

    if r1<1 && r2<1 && r2<1 && r4<1 && r5<1
        erg = 1;
    else
        erg = 0;
    endif
    intensities = erg;
endfunction
```

3). Για κάθε ουρά γνωρίζουμε ότι ο μέσος αριθμός πελατών ισούται με $\frac{\rho_i}{1-\rho_i}$. Βάσει αυτού θα υλοποιήσουμε την συνάρτηση **mean_clients**. Ο κώδικας που χρησιμοποιήσαμε είναι ο ακόλουθος:

```
function [meanQ1, meanQ2, meanQ3, meanQ4, meanQ5] = mean_clients (l1, l2, m1, m2, m3, m4, m5)
    [erg, r1, r2, r3, r4, r5] = intensities(l1, l2, m1, m2, m3, m4, m5)
    meanQ1 = r1/(1-r1);
    meanQ2 = r2/(1-r2);
    meanQ3 = r3/(1-r3);
    meanQ4 = r4/(1-r4);
    meanQ5 = r5/(1-r5);
endfunction
```

4). Για τον υπολογισμό της έντασης φορτίου που δέχεται κάθε ουρά θα χρησιμοποιήσουμε την συνάρτηση intensities. Για τον μέσο χρόνο καθυστέρησης από άκρο σε άκρο του δικτύου θα χρησιμοποιήσουμε την συνάρτηση **mean_clients** και θα βασιστούμε στον νόμο του Little $E[n] = E[T] / \gamma$ και στο γεγονός ότι $\gamma = \lambda_1 + \lambda_2$, καθώς σύμφωνα με το θεώρημα Jackson γ είναι ο συνολικός μέσος ρυθμός πελατών που εισέρχονται από εξωτερικές πηγές. Αρχικά τυπώνουμε την ένταση φορτίου κάθε ουράς:

```
r1 = 0.666667
r2 = 0.428571
r3 = 0.285714
r4 = 0.244898
r5 = 0.547619
```

Ο κώδικας που χρησιμοποιήσαμε είναι ο ακόλουθος:

```
l1 = 4;
l2 = 1;
m1 = 6;
m2 = 5;
m3 = 8;
m4 = 7;
m5 = 6;
[erg, r1, r2, r3, r4, r5] = intensities(l1, l2, m1, m2, m3, m4, m5);
```

Στην συνέχεια τυπώνουμε το μέσο χρόνο καθυστέρησης ενός πελάτη από άκρο σε άκρο του δικτύου:

```
E(T) = 0.93697
```

Ο κώδικας που χρησιμοποιήσαμε είναι ο ακόλουθος:

```
[meanQ1, meanQ2, meanQ3, meanQ4, meanQ5] = mean_clients(l1, l2, m1, m2, m3, m4, m5);
meanTime = (meanQ1 + meanQ2 + meanQ3 + meanQ4 + meanQ5) / (l1 + l2);
printf("E(T) = %d\n", meanTime);
```

5).

Στενωπός ουρά του δικτύου είναι αυτή που έχει την μεγαλύτερη ένταση, η οποία είναι η ουρά Q1 με $\rho_1 = 0.67$. Για να παραμένει το σύστημα εργοδικό πρέπει να ισχύει $\rho_1 < 1$, οπότε για $\rho_1 = 1$ θα έχουμε $\lambda_1 = 0.67 * 7 = 4.02$

6).