ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΣΙΤΩΝ
ΡΟΗ Δ - ΣΥΣΤΗΜΑΤΑ ΑΝΑΜΟΝΗΣ (QUEUING SYSTEMS)

ΓΕΩΡΓΑΚΟΠΟΥΛΟΣ ΓΕΩΡΓΙΟΣ 03120827

ΑΝΑΦΟΡΑ 5ΗΣ ΕΡΓΑΣΤΗΡΙΑΚΗΣ ΑΣΚΗΣΗΣ

ΑΣΚΗΣΗ 1:
ΔΙΚΤΥΟ ΜΕ ΕΝΑΛΛΑΚΤΙΚΗ ΔΡΟΜΟΛΟΓΗΣΗ

Ισχύει ότι:
λ = 10*10^3 πακέτα/sec (10Kpps)
Μεσο μήκος πακέτου = 128bytes
C1 = 15 Mbps
C2 = 12 Mbps

ΕΡΩΤΗΜΑ 1:
Οι απαραίτητες παραδοχές ώστε οι σύνδεσμοι (γραμμές) να μπορούν να μοντελοποιηθούν σαν M/M/1 ουρές είναι σε κάθε γραμμή:

a. Η εισερχόμενη ροή πελατών είναι διαδικασία Poisson.
b. Ο μέσος ρυθμός εξυπηρέτησης να ακολουθά εκθετική κατανομή με ρυθμό μ.

Αναλύουμε κάθε γραμμη ξεχωριστά γνωρίζοντας ότι η συνολική ροή εισερχόμενη ροή πελατών ειναι διαδικασία Poisson με ρυθμό λ και διασπαται τυχαία σε 2 διαδικασίες Poisson με ρυθμούς αλ και (1-α)λ

- Άθροιση – Διάσπαση διαδικασιών Poisson
  - Άθροιση (aggregation) ανεξαρτήτων ροών Poisson λ_1, λ_2: Poisson με μέσο ρυθμό $\lambda = \lambda_1 + \lambda_2$
  - Τυχαία Διάσπαση (random split, routing) ροής Poisson μέσου ρυθμού λ με πιθανότητες $p,\ q = 1 - p$:
    Παράγει διαδικασίες Poisson με ρυθμούς $p\lambda, (1-p)\lambda$

(απο διαφάνειες μαθήματος)

1. Γραμμή 1:

Ο μέσος ρυθμός αφιξης είναι $\lambda_1 = \alpha \cdot 10\, Kpps$
Ο μέσος ρυθμός εξυπηρέτησης είναι

$$\mu_1 = \frac{c_1 \frac{bits}{sec}}{128 \cdot 8 \frac{bits}{packet}} = \frac{(15 \cdot 10^6)}{128 \cdot 8} = 14648.44 = 14.65 \frac{Kpakcets}{sec}$$

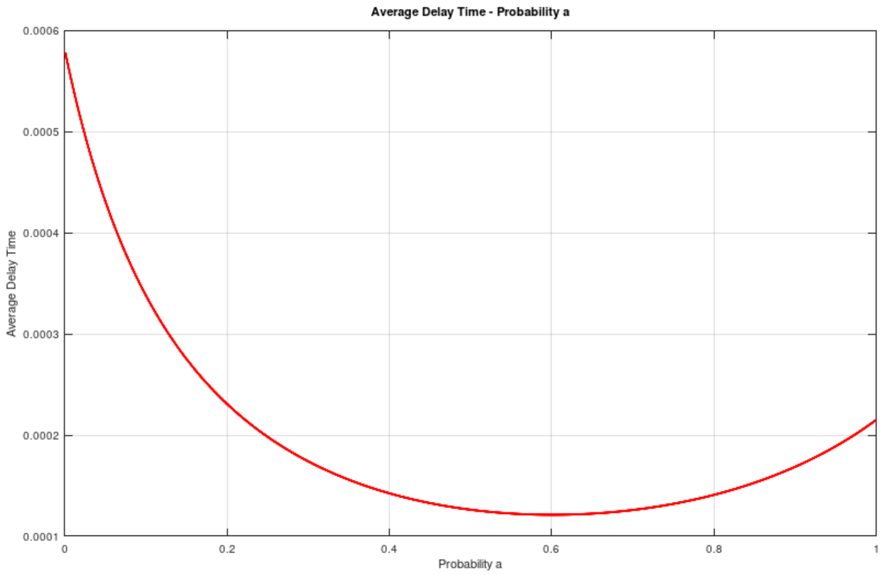
2. Γραμμή 2:

Ο μέσος ρυθμός αφιξης είναι $\lambda_2 = (1 - \alpha) \cdot 10\, Kpps$
Ο μέσος ρυθμός εξυπηρέτησης είναι

$$\mu_2 = \frac{c_2 \frac{bits}{sec}}{128 \cdot 8 \frac{bits}{packet}} = \frac{(12 \cdot 10^6)}{128 \cdot 8} = 11718.75 = 11.72 \frac{Kpakcets}{sec}$$

<u>ΕΡΩΤΗΜΑ 2:</u>

Ισχύει ότι E[n] = E[n1] + E[n2]. Επιπλέον ο μέσος χρόνος καθυστέρησης E[T] υπολογίζεται, με βάση τον νόμο οτυ Little, διαιρώντας τον μέσο αριθμό των πελατών E[n] με το συνολικό ρυθμό λ.
Χρησιμοποιώντας το octave, και συγκεκριμένα την συνάρτηση qsmm1, φτιάχνουμε το διάγραμα του μεσου χρόνου καθυστέρησης ένος τυχαίου πακέτου συναρτήσει του α.



Επίσης, χρησιμοποιωντας το octave υπολογίζουμε την τιμή του α που ελαχιστοποιεί το E[T].

```
Command Window
Minimum value of a: 0.601
Minimum total delay: 0.00012121
>>
```

Ο κώδικας που χρησιμοποιήσαμε για το ερώτημα αυτό είναι ο εξής:

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

% Initialization
a = 0.001:0.001:0.999;
lambda = 10000; % l = 10*10^3
lambda1 = 10000*a;
lambda2 = 10000*(1-a);
mu1 = 14648;
mu2 = 11718;

total_delay = zeros(size(a));
```

```
for i = 1:length(a)
    [U1, R1, Q1, X1, P1] = qsmm1(lambda1(i), mu1);
    [U2, R2, Q2, X2, P2] = qsmm1(lambda2(i), mu2);
    total_requests = Q1 + Q2;
    total_delay(i) = total_requests / lambda;
end
#https://octave.sourceforge.io/queueing/function/qsmm1.html
#U = Server utilization
#S = Server response time
#Q = Average number of requests in the system
#X = Server throughput. If the system is ergodic (mu > lambda), we always have X = lambda
#P = Steady-state probability that there are no requests in the system.
[min_delay, min_index] = min(total_delay);
min_a = a(min_index);

figure(1);
plot(a, total_delay, 'r', 'linewidth', 2);
title('Average Delay Time - Probability a')
xlabel('Probability a');
ylabel('Average Delay Time');
grid on;

disp(['Minimum value of a: ', num2str(min_a)]);
disp(['Minimum total delay: ', num2str(min_delay)]);
```

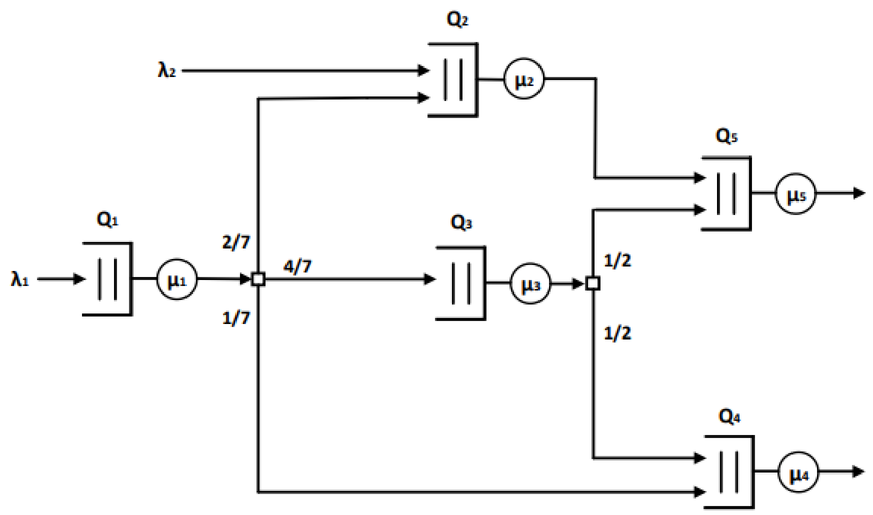
<u>ΑΣΚΗΣΗ 2:</u>
<u>ΑΝΟΙΚΤΟ ΔΙΚΤΥΟ ΟΥΡΩΝ ΑΝΑΜΟΝΗΣ</u>

Όλες οι αφίξεις ακολουθούν την κατανομή Poisson με παραμέτρους:

$$\lambda_i\,,\, i = 1{,}2$$

και οι εξυπηρετήσεις είναι εκθετικά κατανεμημένες με ρυθμούς:

$$\mu_i\,,\, i\ =\ 1{,}2{,}3{,}4{,}5$$



<u>ΕΡΩΤΗΜΑ 1:</u>

Οι απαραίτητες παραδοχές ώστε το παραπάνω δίκτυο να μπορεί να μελετηθεί ως ένα ανοιχτό δίκτυο με το θεώρημα Jackson είναι:

1. Κάθε ουρα Qi να αποτελεί δικτυακό κόμβο εξυπηρετησης κορμού με εκθετικό ρυθμο εξυπηρετησης μi
2. Οι αφίξεις πελατών από εξωτερικές πηγές να είναι συνδεδεμένες σε καποιον κόμβο Qs και προς εξωτερικούς προορισμούς σε καποιον κόμβο Qd. Στην δικιά μας περιπτωση οι εχουμε 2 κομβους Qs τους Q1 και Q2 καθώς και 2 κομβους Qd, τους Q4 και Q5

3. Η εσωτερική δρομολόγηση γίνεται με τυχαίο τρόπο και η πιθανότητα δρομολόγησης από κόμβο Qi προς κόμβο Qj είναι: rij. Στην δικιά μας περίπτωση ισχύει ότι r12=2/7, r13=4/7, r14=1/7, r35=r34=½.
4. Οι ροές που διαπερνούν τον κόμβο Qj έχουν συνολικό μέσο ρυθμό: $\lambda_j = \gamma_j + \sum_{i=1,i\neq j}^{M} r_{ij} \cdot \lambda_i$
5. Οι χρόνοι εξυπηρέτησης των πελατών στις ουρές εξαρτώνται από την κατανομή του κάθε εξυπηρετητή (ιδιότητα έλλειψης μνήμης)

Από Διαφανεις Μαθήματος:

- Ανοικτό δίκτυο M ***δικτυακών κόμβων εξυπηρέτησης κορμού*** (ουρών αναμονής) $\mathbf{Q}_i$, $i = 1,2,\dots,M$ με εκθετικούς ρυθμούς εξυπηρέτησης μ_i
- Αφίξεις πελατών (πακέτων) από εξωτερικές πηγές (sources) άμεσα συνδεμένες στον δικτυακό κόμβο κορμού $\mathbf{Q}_s$προς εξωτερικούς προορισμούς (destinations) άμεσα συνδεμένους στον δικτυακό κόμβο κορμού $\mathbf{Q}_d$:
  Ανεξάρτητες ροές Poisson μέσου ρυθμού γ_{sd} όπου $s, d \in \{1,2,\dots,M\}$
  Συνολική εξωγενής ροή Poisson σε $\mathbf{Q}_s$: $\gamma_s = \sum_{d=1,d\neq s}^{M} \gamma_{sd}$, $s, d \in \{1,2,\dots,M\}$
- Εσωτερική ***δρομολόγηση*** (routing) με τυχαίο τρόπο και πιθανότητα δρομολόγησης πελάτη από τον κόμβο κορμού (ουρά) $\mathbf{Q}_i$ στον κόμβο $\mathbf{Q}_j$: r_{ij}
- Τότε τον κόμβο εξυπηρέτησης $\mathbf{Q}_j$ διαπερνούν ροές με συνολικό μέσο ρυθμό

$$\lambda_j = \gamma_j + \sum_{i=1,i\neq j}^{M} r_{ij}\lambda_i , \qquad j = 1,2,\dots,M$$

- Οι χρόνοι εξυπηρετήσεις πελατών όπως διαπερνούν το δίκτυο δεν διατηρούν την τιμή τους (έλλειψη μνήμης) αλλά αποκτούν χρόνο εξυπηρέτησης ανάλογα με την κατανομή του κάθε εξυπηρετητή (***Kleinrock's Independence Assumption***, επαληθευμένη με προσομοιώσεις σε δίκτυα με όχι απλοϊκή τοπολογία)

ΕΡΩΤΗΜΑ 2:

Θα χρησιμοποιήσουμε το θεώρημα Burke. Συμφωνα με το θεώρημα αυτό: Η έξοδος πελατών από ουρά M/M/1 ακολουθεί κατανομή Poisson με ρυθμό τον ρυθμό εισόδου λ.
Επομένως

$$\lambda_{o\lambda 1} = \lambda_1$$
$$\lambda_{o\lambda 2} = \lambda_2 + r_{12} \cdot \lambda_1$$
$$\lambda_{o\lambda 3} = r_{13} \cdot \lambda_1$$
$$\lambda_{o\lambda 4} = r_{14} \cdot \lambda_1 + r_{34} \cdot \lambda_{o\lambda 3} = r_{14} \cdot \lambda_1 + r_{34} \cdot r_{13} \cdot \lambda_1$$

$$\lambda_{o\lambda 5} = \lambda_{o\lambda 2} + r_{35}\lambda_{o\lambda 3} = \lambda_2 + r_{12} \cdot \lambda_1 + r_{35} \cdot r_{13} \cdot \lambda_1$$

Ισχύει ότι ρ = λ/μ. Άρα:

$$\rho_1 = \frac{\lambda_1}{\mu_1}$$
$$\rho_2 = \frac{\lambda_2 + r_{12}\lambda_1}{\mu_2} = \frac{\lambda_2 + \frac{2}{7}\lambda_1}{\mu_2}$$
$$\rho_3 = \frac{r_{13}\lambda_1}{\mu_3} = \frac{\frac{4}{7}\lambda_1}{\mu_3}$$
$$\rho_4 = \frac{r_{14}\lambda_1 + r_{34} \cdot r_{13} \cdot \lambda_1}{\mu_4} = \frac{\frac{3}{7}\lambda_1}{\mu_4}$$
$$\rho_5 = \frac{\lambda_2 + r_{12} \cdot \lambda_1 + r_{35} \cdot r_{13} \cdot \lambda_1}{\mu_5} = \frac{\frac{4}{7}\lambda_1 + \lambda_2}{\mu_5}$$

Χρησιμοποιούμε το Octave για να υλοποιησουμε τη συνάρτηση intensities.
Η συναρτηση αυτή θα υπολογίζει τις τιμές ρi , i = 1, 2, 3, 4, 5 .Θα έχει σαν ορίσματα τις παραμέτρους λi , i = 1, 2 και μi , i = 1, 2, 3, 4, 5 και θα επιστρέφει:

1. Τις τιμές ρi
2. Την τιμή 1 αν το σύστημα είναι εργοδικό ή 0, εάν παραβιάζεται η συνθήκη της εργοδικότητας σε κάποια ουρά.

Ο κώδικας που χρησιμοποιήσαμε για το ερώτημα αυτό είναι ο εξής:

```
clc;
clear all;
close all;
pkg load queueing
#Ερώτημα 2
function [r1, r2, r3, r4, r5, return_value] = intesities(lambda1, lambda2, mu1, mu2, mu3, mu4, mu5)
  r1 = (lambda1/mu1);
  r2 = ((lambda2+(2/7)*lambda1)/mu2);
  r3 = ((4/7)*lambda1/mu3);
  r4 = ((3/7)*lambda1/mu4);
  r5 = (((4/7)*lambda1+lambda2)/mu5);
  if((r1<1) || (r2<1) || (r3<1) || (r4<1) || (r5<1))
    return_value = 1;
  else
    return_value = 0;
  endif
endfunction
```

ΕΡΩΤΗΜΑ 3:

Χρησιμοποιούμε το Octave για να υλοποιησουμε τη συνάρτηση mean_clients.
Η συναρτηση αυτή δέχεται ως ορίσματα τις παραμέτρους τις τιμές λi , i = 1,2 και μi, i=1,2,3,4,5 και θα επιστρέφει ένα διάνυσμα με τους μέσους αριθμούς πελατών των Qi, i=1,2,3,4,5
Ο τύπος που χρησιμοποιείται για κάθε ουρά είναι:

$$\mathrm{E}[n] = \frac{\rho}{1-\rho}$$

Ο κώδικας που χρησιμοποιήσαμε για το ερώτημα αυτό είναι ο εξής:

```
#Ερώτημα 3
function [Q1, Q2, Q3, Q4, Q5] = mean_clients(lambda1, lambda2, mu1, mu2, mu3, mu4, mu5)
  [r1, r2, r3, r4, r5, e] = intesities(lambda1, lambda2, mu1, mu2, mu3, mu4, mu5);
  Q1 = r1/(1-r1);
  Q2 = r2/(1-r2);
  Q3 = r3/(1-r3);
  Q4 = r4/(1-r4);
  Q5 = r5/(1-r5);
endfunction
```

ΕΡΩΤΗΜΑ 4:

Για τις τιμές των παραμέτρων λ1 = 4, λ2 = 1, μ1 = 6, μ2 = 5, μ3 = 8, μ4 = 7, μ5 = 6 θα υπολογίσουμε χρησιμοποιώντας τις προηγούμενες συναρτήσεις (α) την ένταση του φορτίου που δέχεται η κάθε ουρά και (β) το μέσο χρόνο καθυστέρησης ενός πελάτη από άκρο σε άκρο του δικτύου.
Για τον μέσο χρόνο καθυστέρησης ενός πελάτη από άκρο σε άκρο ισχύει ότι:

$$\mathrm{E}[\mathrm{T}] = \frac{\mathrm{E}[n]}{\gamma}$$

Οπου γ = λ1+λ2 και E[n] = Q1+Q2+Q3+Q4+Q5
Προκύπτουν τα εξής αποτελέσματα:

```
Command Window
r1 = 0.666667
r2 = 0.428571
r3 = 0.285714
r4 = 0.244898
r5 = 0.547619
Average Delay Time = 0.936970
>>
```

Ο κώδικας που χρησιμοποιήσαμε για το ερώτημα αυτό είναι ο εξής:

```
#Ερώτημα 4
lambda1 = 4;
lambda2 = 1;
mu1 = 6;
mu2 = 5;
mu3 = 8;
mu4 = 7;
mu5 = 6;
[r1,r2,r3,r4,r5,e]=intesities(lambda1, lambda2, mu1, mu2, mu3, mu4, mu5);
fprintf('r1 = %f\n', r1);
fprintf('r2 = %f\n', r2);
fprintf('r3 = %f\n', r3);
fprintf('r4 = %f\n', r4);
fprintf('r5 = %f\n', r5);
[Q1, Q2, Q3, Q4, Q5] = mean_clients(lambda1, lambda2, mu1, mu2, mu3, mu4, mu5);
fprintf('Average Delay Time = %f\n', (Q1+Q2+Q3+Q4+Q5)/(lambda1+lambda2));
```

ΕΡΩΤΗΜΑ 5:

Στενωπός (bottleneck) του δικτύου είναι η οποία έχει την μεγαλύτερη ροή φορτίου. Συνεπώς είναι η ουρά 1.
Η μέγιστη τιμή του λ1 ώστε η ουρά αυτή να παραμείνει εργοδική προκύπτει ως εξής:

$$\rho_1 = \frac{\lambda_1}{\mu_1} < 1 \Rightarrow \lambda_1 < \mu_1$$

Άρα ως οριακή τιμή παίρνουμε την τιμή

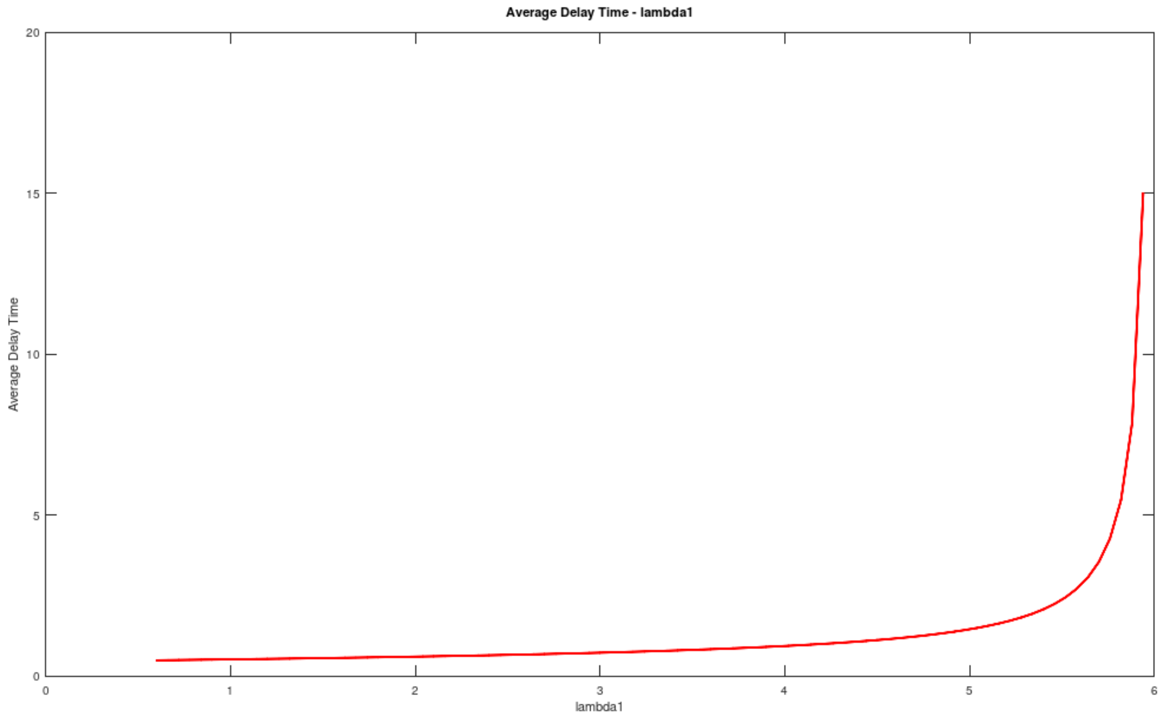
$$\lambda_1 = \mu_1 = 6\ \frac{\pi\varepsilon\lambda\alpha\tau\varepsilon\varsigma}{sec}$$

ΕΡΩΤΗΜΑ 6:

Το λ1 ανήκει στο διάστημα:

$$[6 \cdot 0.1,\ 6 \cdot 0.99] = [0.6\ ,\ 5.94]$$

Μέσω του octave υλοποιούμε το διάγραμμα του μέσου χρόνου καθυστέρησης ενός πελάτη από άκρο σε άκρο του δικτύου για κάθε τιμή του λ1



Όπως παρατηρούμε για λ1=4 πελάτες/sec προκύπτει (περίπου) η τιμή 1, όπως υπολογίσαμε στο ερώτημα 4.

Ο κώδικας που χρησιμοποιήσαμε για το ερώτημα αυτό είναι ο εξής:

```
#Ερώτημα 6
maxlambda1 = 6;
for i = 1:1:90;
  lambda1 = (0.1*maxlambda1)+(i-1)*0.01*maxlambda1;
  [Q1, Q2, Q3, Q4, Q5] = mean_clients(lambda1, lambda2, mu1, mu2, mu3, mu4, mu5);
  delay_time(i)= (Q1+Q2+Q3+Q4+Q5)/(lambda1+lambda2);
endfor

lambda1 = (0.1*maxlambda1):(0.01*maxlambda1):(0.99*maxlambda1);
figure(1);
plot(lambda1, delay_time,"r","linewidth",2);
title("Average Delay Time - lambda1")
xlabel("lambda1");
ylabel("Average Delay Time");
```