DUBLIN CITY UNIVERSITY

ELECTRONIC AND COMPUTER ENGINEERING

EE517 - Network Analysis And Dimensioning

Assignment 2

Author

Michael Lenehan michael.lenehan4@mail.dcu.ie

Student Number: 15410402

08/04/2020

Declaration

I declare that this material, which I now submit for assessment, is entirely my own work and has not been taken from the work of others, save and to the extent that such work has been cited and acknowledged within the text of my work. I understand that plagiarism, collusion, and copying are grave and serious offences in the university and accept the penalties that would be imposed should I engage in plagiarism, collusion or copying. I have read and understood the Assignment Regulations set out in the module documentation. I have identified and included the source of all facts, ideas, opinions, and viewpoints of others in the assignment references. Direct quotations from books, journal articles, internet sources, module text, or any other source whatsoever are acknowledged and the source cited are identified in the assignment references. This assignment, or any part of it, has not been previously submitted by me or any other person for assessment on this or any other course of study.

I have read and understood the DCU Academic Integrity and Plagiarism at `https://www4.dcu.ie/sites/default/files/policy/1%20-%20integrity_and_plagiarism_ovpaa_v3.pdf` and IEEE referencing guidelines found at `https://loop.dcu.ie/mod/url/view.php?id=448779`.

Signed: ____________________ Date: xx/xx/20xx

Michael Lenehan

Contents

1 Question 1

Given:

$$
\begin{aligned}
\text{Poisson Distribution: } p(k) &= \frac{\lambda^k e^{-\lambda}}{k!} \\
\text{Probability Generating Function: } G(z) &\triangleq E[z^k] = \sum_{\forall k \in \Omega} z^k px(k) \\
\text{nth Central Moment: } \mu_n &= E[(X - E[X])^2] \\
G(z) = \sum_{k=0}^{\infty} z^k \frac{\lambda^k e^{-\lambda}}{k!} &= e^{-\lambda} \sum_{k=0}^{\infty} \frac{(z\lambda)^k}{k!} \\
e^{-\lambda} e^{z\lambda} &= e^{(z-1)\lambda} \\
\text{ith Moment: } \overline{X^i} &= \frac{d}{dz}(z\frac{d}{dz})^{i-1} G(z)\Big|_{z=1} \\
\text{First Moment: } \overline{X} = \frac{d}{dz} G(z)\Big|_{z=1} &= \frac{d}{dz} e^{(z-1)\lambda}\Big|_{z=1} \\
&= \lambda \\
\text{Second Moment: } \overline{X^2} &= \frac{d}{dz} z \frac{d}{dz} G(z)\Big|_{z=1} \\
&= \frac{d}{dz} z(\frac{d}{dz} e^{(z-1)\lambda})\Big|_{z=1} \\
&= \lambda^2 + \lambda \\
\therefore \text{ Third Moment: } \overline{X^3} &= \frac{d}{dz}(z\frac{d}{dz})^2 G(z)\Big|_{z=1} \\
&= \frac{d}{dz} z(\frac{d}{dz} \lambda e^{(z-1)\lambda}) \\
&= \lambda^3 + 3\lambda^2 + \lambda \\
\text{Third Central Moment: } \mu_3 &= E[(X - E[X])^3] \\
&= E[X^3] - 3E[X^2]E[X] + 3E[x]E^2[X] - E^3[X] \\
&= \lambda^3 + 3(\lambda^2 + \lambda) - 3(\lambda^2 + \lambda)(\lambda) + 3(\lambda)(\lambda)^2 - \lambda^3 \\
&= \lambda^3 + 3\lambda^3 - 3\lambda^3 - \lambda^3 + 3\lambda^2 - 3\lambda^2 + \lambda \\
&= \lambda.
\end{aligned}
$$

2 Question 2

2.1 Part 1

This process is a Markov chain, as it meets the definition of a Markoc chain, having both a discrete state space and is in discrete time.

The process is ergodic, as its properties are not time dependent. The mean value is achieved through a sufficiently long sample (i.e. a large number of coin flips).

2.2 Part 2

The following table shows the Transition Probability Matrix for the process.

Table 1: Transition Probability Matrix

0	1	2	3	4	...
0	0.5	0.5	0	0	...
0	0	0.5	0.5	0	...
0	0	0	0.5	0.5	...
0	0	0	0	0.5	0.5

2.3 Part 3

$$
\begin{aligned}
\pi^{(n)} &= \pi^{(n-1)}P \\
\text{Given: } n &= 3 \\
\pi^{0} &= 1 \\
\pi^{(1)} &= \pi^{(0)}P = 1(0.5) \\
&= 0.5 \\
\pi^{(2)} &= \pi^{(1)}P = 0.5(0.5) \\
&= 0.25 \\
\pi^{(3)} &= \pi^{(2)}P = 0.25(0.5) \\
&= 0.125 \\
\therefore P(X = 4|n = 3) &= 1 - 0.125 \\
&= \frac{7}{8}.
\end{aligned}
$$

3 Question 3

$$\tau.$$

4 Question 4

Distribution of the Waiting Time:

$$F_W(T) = 1 - \rho e^{-\mu(1-\rho)t}$$

Probability of Delay > t

$$Pr[W > t] = 1 - F_W(t) = \rho e^{-\mu(1-\rho)t}$$

Given:

$$t = 0.001$$

$$\lambda = 1000$$

$$\mu = \text{Link Capacity}/\text{Mean Packet Length (bits)}$$

$$\begin{aligned}
\therefore Pr[W > t] &= \frac{\lambda}{\mu} e^{i-\mu(1-\lambda/\mu)t} \\
&= \frac{\lambda * (700 * 8)}{\text{Link Capacity}} e^{(-\text{Link Capacity}/(700*8)+\lambda)(0.001)} \\
&= \frac{5.6x10^6}{\text{Link Capacity}} e^{(-\frac{0.001*\text{Link Capacity}}{5600}+1)}.
\end{aligned}$$

Using trial-and-error, a value of 23Mbps ($23x10^6$) gives an approximate probability value of 0.01089.

5 Question 5

Table 2: Blocking Probabilities (As given by Erlang-B Chart)

Number of Channels (W)	Blocking Probability (B)	Initialisation Cost (1.2 x W)	Blocking Cost (3.1 x 10 x B)	Total Cost (IC + BC)
4	0.6467	4.8	20.047	24.847
8	0.3383	9.6	10.488	20.088
12	0.1197	14.4	3.712	18.119
16	0.0223	19.2	0.6914	19.891
20	0.0019	24	0.0589	24.058

The following plot shows the total overall costs for between zero and 25 channels.

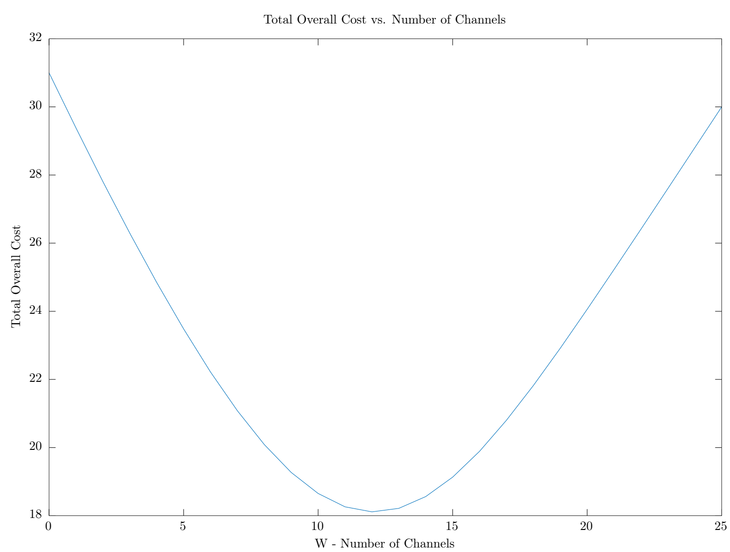

Figure 1: Plot of the Total Overall Cost within the given system

Therefore, the number of channels which minimizes the total overall cost is 12.

The above answers were obtained using the Erlang-B Iterative Formula, programmed using GNU-Octave. This code is shown below.

```
x = 0;
Ex = 1;
A = 10;
B = zeros(0,25);
IC = zeros(0,25);
BC = zeros(0,25);
Total = zeros(0,25);
f1 = fopen('erlang.txt', 'w');
f2 = fopen('costs.txt', 'w');
while(x<26)
        Ex = (A*Ex)/(x+A*Ex);
        fprintf(f1, "W: %d \t B: %d\n", x, Ex);
        x++;
        B(x) = Ex;
        IC(x) = 1.2*(x-1);
        BC(x) = 3.1*A*Ex;
        Total(x) = IC(x)+BC(x);
        fprintf(f2, "Installation: %d \t Blocking: %d \t
            Total: %d\n", IC(x),
        BC(x), Total(x));
endwhile
h=figure();
plot(0:1:25, Total);
xlabel("W - Number of Channels");
ylabel("Total Overall Cost");
title("Total Overall Cost vs. Number of Channels");
print(h, "Q5.pdf", "-dpdflatexstandalone")
system("pdflatex Q5.pdf")
system("pdftoppm Q5.pdf Q5 -png");
```

```
pause();
```

6 Question 6

6.1 Part 1

The following figure shows the block diagram of traffic flows/overflows within the system. The A values (A_{micA}, A_{micB}, A_{micC}, A_{mac}, represent the "Offered Load" to the cells (Micro Cell A, Micro Cell B, Micro Cell C, Macro Cell) respectively. N_{mic} represents the number of channels in the micro cells, while N_{mac} represents the number of channels in the macro cell. The "Offered Load" for the Overflow is represented by A^*, with the number of channels represented by $N_{mac} + N^*$.

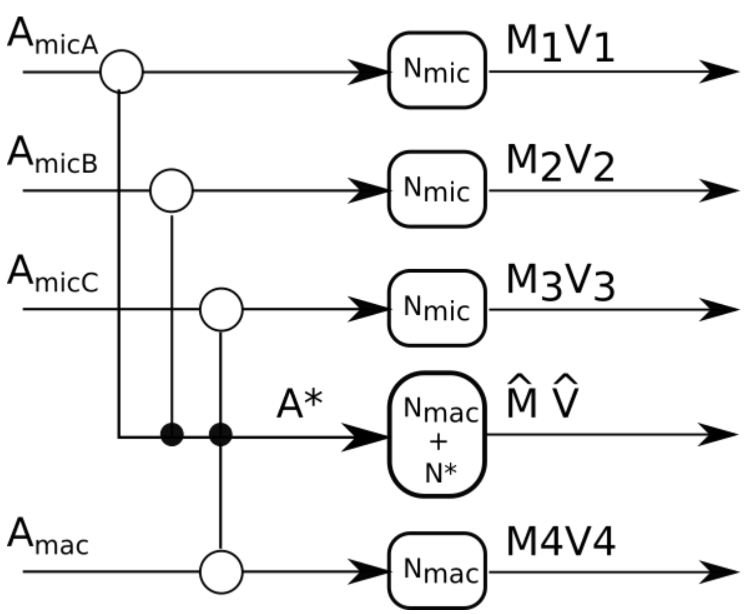

Figure 2: Traffic Flows/Overflows Block Diagram

6.2 Part 2

$$
\begin{aligned}
\text{Given: } A_{micA} &= 20 \\
A_{micB} &= 30 \\
A_{micC} &= 25 \\
A_{mac} &= 5 \\
B &\leq 0.0 \\
\hat{M}_1 &= A_{micA}E(N_{mic}, A_{micA}) \\
\hat{V}_1 &= \hat{M}_1(1 - \hat{M}_1 + A_{micA}/(N_{mic} + 1 - A_{micA} + \hat{M}_1)) \\
\hat{M}_2 &= A_{micB}E(N_{mic}, A_{micB}) \\
\hat{V}_2 &= \hat{M}_2(1 - \hat{M}_2 + A_{micB}/(N_{mic} + 1 - A_{micB} + \hat{M}_2)) \\
\hat{M}_3 &= A_{micC}E(N_{mic}, A_{micC}) \\
\hat{V}_3 &= \hat{M}_3(1 - \hat{M}_3 + A_{micC}/(N_{mic} + 1 - A_{micC} + \hat{M}_3)) \\
\hat{M}_4 &= A_{mac}E(N_{mac}, A_{mac}) \\
\hat{V}_4 &= \hat{M}_4(1 - \hat{M}_4 + A_{mac}/(N_{mac} + 1 - A_{mac} + \hat{M}_4)) \\
M &= \hat{M}_1 + \hat{M}_2 + \hat{M}_3 + \hat{M}_4 \\
V &= \hat{M}_1 + \hat{M}_2 + \hat{M}_3 + \hat{M}_4
\end{aligned}
$$

Rapps Approximations:

$$
\begin{aligned}
A^* &\approx V + 3Z(Z - 1) \\
N^* &\approx A^*(M + Z)/(M + Z - 1) - M - 1 \\
\text{Where: } Z &= V/M
\end{aligned}
$$

Final Blocking Probability:

$$
B = E(A^*, N^* + N_{mac}).
$$

6.3 Part 3

The minimum cost of the system is 102 cost units. This cost corresponds to 24 Micro Cell Channels at 1 cost unit per channel, and 78 Macro Cell Channels at 3 cost units per channel.

The figure below shows the Total System cost versus the number of Micro Cell channels. The cost decreases to 102 up to 24 Micro Cell Channels, and increases past this value.

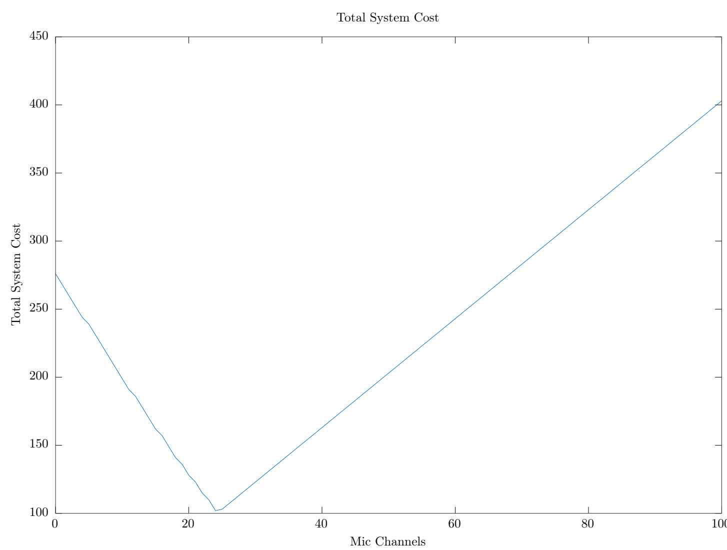

Figure 3: Total System Cost vs. Micro Cell Channels

The following octave code shows the GNU-Octave implementation of the presented problem. In order to calculate the minimum cost, the calculations described in Part 2 are executed for increasing values of Micro Cell and Macro Cell Channels. The blocking probability of each micro cell is calculated using the loops current micro cell channel number and each cells respective offered load. These blocking probabilities can be used to find the mean and variance for each cell.

A nested loop increments through the number of macro cell channels, calculating the blocking probability, mean, and variance of the directly accessed macro cell. The offered load and number of channels for the overflow channel is given by using Rapps approximations on the total means and variances for the micro and macro cells.

Using the offered load and number of channels for the overflow channel from Rapps approximations, the overall blocking probability can be calculated. This loop executes until the blocking probability drops below the required QoS value of 1

For all calculations of blocking probabilities, the iterative method for Erlang-B is used.

```
AmicA = 20;
AmicB = 30;
AmicC = 25;
Amac = 5;
B = 0.01;
Nmic=0;
Nmac=0;
MacCost = 3;
MicCost = 1;
TotalCost = zeros(0,1000)
cnt = 1;

f1 = fopen('nVals.txt', 'w');
f2 = fopen('costs.txt', 'w');

while(Nmic <= 100)
        E1 = E2 = E3 = 1;
        Ex = 1;
        x=0;
        i=0;
        tmp = Nmac;
        while(x <= Nmic)
```

```
        E1 = (AmicA*E1)/(x+AmicA*E1);
        E2 = (AmicB*E2)/(x+AmicB*E2);
        E3 = (AmicC*E3)/(x+AmicC*E3);
        x++;
endwhile

M1 = AmicA*E1;
V1 = M1*(1-M1+1*(AmicA/(Nmic+1-AmicA+M1)));

M2 = AmicB*E2;
V2 = M2*(1-M2+1*(AmicB/(Nmic+1-AmicB+M2)));

M3 = AmicC*E3;
V3 = M3*(1-M3+1*(AmicC/(Nmic+1-AmicC+M3)));

while(Ex > B)
        E4 = 1;
        while(i <= Nmac)
                E4 = (Amac*E4)/(i+Amac*E4);
                M4 = Amac*E4;
                V4 = M4*(1-M4+1*(Amac/(i+1-Amac+
                    M4)));

                M = M1+M2+M3+M4;
                V = V1+V2+V3+V4;
                Z = V/M;
                Astar = V+3*Z*(Z-1);
                Nstar = Astar*((M+Z)/(M+Z-1))-M
                    -1;
                Ex = (Astar*Ex)/(i+Nstar+Astar*Ex
```

```
                            );
                        i++;
                endwhile
                Nmac++;
        endwhile
        fprintf(f1, "Nmic: %d \t Nmac: %d \t Astar: %d \t
            Ex: %d\n", Nmic, Nmac,
        Astar, Ex);
        TotalCost(cnt) = (MacCost*Nmac) + (MicCost*Nmic);
        fprintf(f2, "Mic Cost: %d \t Mac Cost: %d \t
            Total: %d\n", MicCost*Nmic,
        MacCost*Nmac, TotalCost(cnt));
        cnt++;
        Nmac = tmp;
        Nmic++;
        Nmac++;
endwhile

h = figure();
plot(0:1:100, TotalCost);
xlabel("Mic Channels");
ylabel("Total System Cost");
title("Total System Cost");
print(h, "Q6.pdf", "-dpdflatexstandalone");
system("pdflatex Q6.pdf")
system("pdftoppm Q6.pdf Q6 -png");
pause();
```