

Mini Projeto 1

Universidade de Aveiro

Mariana Pinto, Raquel Pinto



Mini Projeto 1

Modelação e Desempenho de Redes e Serviços
Departamento de Eletrónica, Telecomunicações e
Informática

Universidade de Aveiro

Mariana Pinto, Raquel Pinto
(84792) mariana17@ua.pt, (92948) raq.milh@ua.pt

Delivery Date: 14th December 2021

Conteúdo

1	Task 1	1
1.a	Task 1.a - Enunciado	1
1.a.1	Código	1
1.a.2	Resultados e Conclusões	2
1.b	Task 1.b - Enunciado	5
1.b.1	Código	5
1.b.2	Resultados e Conclusões	6
1.c	Task 1.c - Enunciado	9
1.c.1	Código	9
1.c.2	Resultados e Conclusões	10
1.d	Task 1.d - Enunciado	12
1.d.1	Código	12
1.d.2	Resultados e Conclusões	13
1.e	Task 1.e - Enunciado	14
1.e.1	Código	14
1.e.2	Resultados e Conclusões	21
2	Task 2	24
2.a	Task 2.a - Enunciado	24
2.a.1	Código	24
2.a.2	Resultados e Conclusões	26
2.b	Task 2.b - Enunciado	28
2.b.1	Código	28
2.b.2	Resultados e Conclusões	30
2.c	Task 2.c - Enunciado	32
2.c.1	Código	32
2.c.2	Resultados e Conclusões	33
2.d	Task 2.d - Enunciado	35
2.d.1	Código	35
2.d.2	Resultados e Conclusões	38
2.e	Task 2.e - Enunciado	41
2.e.1	Código	41
2.e.2	Resultados e Conclusões	44
2.f	Task 2.f - Enunciado	47

2.f.1	Código	47
2.f.2	Resultados e Conclusões	50
3	Contribuições dos autores	53

Capítulo 1

Task 1

Neste exercício, considerou-se o Simulador1 usado na Task 5 nos trabalhos da aula prática.

1.a Task 1.a - Enunciado

Consider the case of $C = 10$ Mbps and $f = 1.000.000$ Bytes. Run Simulator1 50 times with a stopping criterion of $P = 10000$ each run and compute the estimated values and the 90% confidence intervals of the average delay performance parameter when $\lambda = 400, 800, 1200, 1600$ and 2000 pps. Present the average packet delay results in bar charts with the confidence intervals in error bars. Justify the results and take conclusions concerning the impact of the packet rate in the obtained average packet delay.

1.a.1 Código

Tendo sido sugerido que se corresse 50 iterações do simulador 1 para quatro números diferentes de taxas de pacotes ($\lambda = 400, 800, 1200, 1600$ and 2000 pps), com $C = 10$ Mbps, $f = 1.000.000$ Bytes, $P = 10000$ e com intervalos de confiança de 90% de desempenho de atraso médio dos pacotes, criou-se um script que obedecesse aos valores anteriormente descritos.

Criou-se uma matriz ***lambdaArray*** com uma linha e quatro colunas, correndo para cada valor da matriz a simulação 50 vezes, guardando os valores de cada simulação para, a partir desses, calcular o tempo de atraso médio dos pacotes de dados para cada valor diferente de λ .

O código gerado para a resolução do exercício é o seguinte:

```
1 fprintf( 'Task 1 - Alinea A\n' );  
2 P = 10000;
```

```

3 lambdaArray = [400,800,1200,1600,2000];
4 C = 10;
5 f = 1000000;
6 for index=1:numel(lambdaArray)
7     N = 50;
8     alfa=0.1;
9     PL_lst = zeros(1,N);
10    APD_lst = zeros(1,N);
11    MPD_lst = zeros(1,N);
12    TT_lst = zeros(1,N);
13    for i = 1:N
14        [PL_lst(i),APD_lst(i),MPD_lst(i),TT_lst(i)] =
            Simulator1(lambdaArray(index),C,f,P);
15    end
16    fprintf('Valor %d de labmda:\n',lambdaArray(index));
17    % Calculate Average Packet Delay
18    APD = mean(APD_lst);
19    APD_conf = norminv(1-alfa/2)*sqrt(var(APD_lst)/N);
20    fprintf('Av. Packet Delay (ms)= %.2e +-%.2e\n',APD,
        APD_conf);
21    APD_Results(index) = APD;
22    APD_Erro(index) = APD_conf;
23 end
24 figure(1);
25 bar(lambdaArray,APD_Results);
26 grid on
27 xlabel("Packet Rate (pps)");
28 ylabel("Packet Delay");
29 title(["Average Packet Delay - Alinea A"]);
30 hold on
31 er = errorbar(lambdaArray,APD_Results,APD_Erro,APD_Erro);
32 er.Color = [0 0 0];
33 er.LineStyle = 'none';
34 hold off

```

1.a.2 Resultados e Conclusões

Perante este programa, é de esperar que com o aumento da taxa de pacotes o tempo médio de espera dos pacotes de dados aumente. A fila de espera vai enchendo, fazendo com que os pacotes esperem mais tempo para serem atendidos.

Como é possível ver na Figura 1.1, com o aumento da taxa de pacotes, o tempo médio de espera aumenta, como argumentado anteriormente.

```
Task 1 - Alinea A
Valor 400 de labmda:
Av. Packet Delay (ms)= 6.13e-01 +-1.87e-03
Valor 800 de labmda:
Av. Packet Delay (ms)= 8.05e-01 +-3.36e-03
Valor 1200 de labmda:
Av. Packet Delay (ms)= 1.18e+00 +-7.99e-03
Valor 1600 de labmda:
Av. Packet Delay (ms)= 2.30e+00 +-4.14e-02
Valor 2000 de labmda:
Av. Packet Delay (ms)= 2.89e+01 +-3.96e+00
```

Figura 1.1: Resultados numéricos da alínea A.

Partindo do Gráfico 1.2 gerado a partir do último script, é possível observar que a taxa de pacotes influencia o tempo médio de espera dos pacotes. Quanto maior o valor da taxa de pacotes, maior o número de pacotes na fila de espera para serem transmitidos e, como consequência, o tempo de espera de cada pacote aumenta.

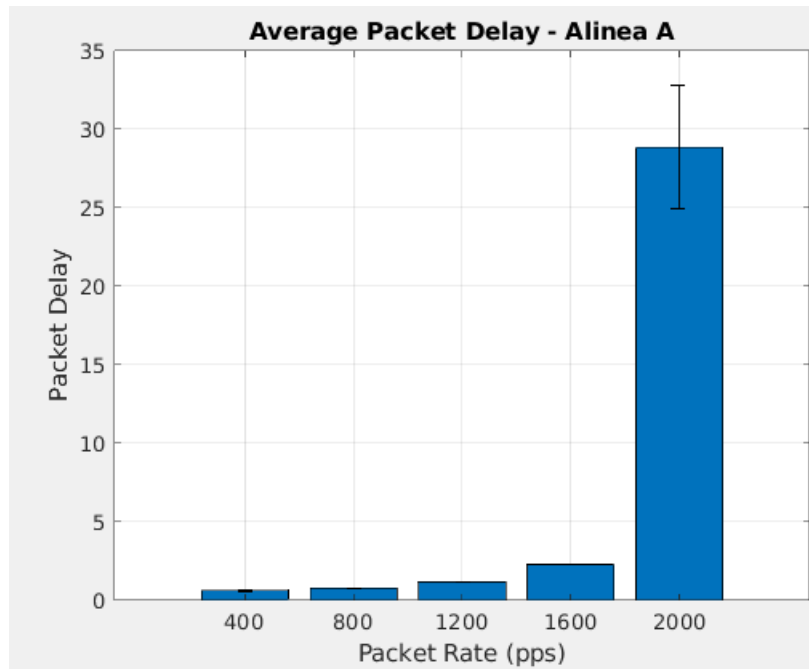


Figura 1.2: Resultados gráficos da alínea A.

É possível observar uma diferença mais acentuada entre valores de taxa de pacotes de 1600 e 2000 pps, onde o atraso é 12x maior para valores de 2000 pps, quando comparado com 1600 pps.

Concluindo, a partir de valores aproximadamente maiores de 1660 pps, o valor médio do atraso de pacotes é muito acentuado, pelo que não faz sentido escolher valores de taxa de pacotes maiores que 1600 pps, quando o objetivo é obter baixos tempos de espera para os pacotes.

1.b Task 1.b - Enunciado

Consider the case of $\lambda = 1800$ pps and $C = 10$ Mbps. Run Simulator1 50 times with a stopping criterion of $P = 10000$ each run and compute the estimated values and the 90% confidence intervals of the average delay and packet loss performance parameters when $f = 100.000, 20.000, 10.000$ and 2.000 Bytes. Present the average packet delay results in one figure and the average packet loss results in another figure (in both cases, in bar charts with the confidence intervals in error bars). Justify the results and take conclusions concerning the impact of the queue size in the obtained average packet delay and average packet loss.

1.b.1 Código

Tendo sido sugerido que se corresse 50 iterações do simulador 1 para quatro números diferentes de queue size ($f = 100.000, 20.000, 10.000$ and 2.000 bytes), com $C = 10$ Mbps, $\lambda = 1800$ pps, $P = 10000$ e com intervalos de confiança de 90% de desempenho de atraso médio dos pacotes, criou-se um script que obedecesse aos valores anteriormente descritos. .

Criou-se uma matriz *farray* com uma linha e quatro colunas, correndo para cada valor da matriz a simulação 50 vezes, guardando os valores de cada simulação para, a partir desses, calcular o tempo médio de atraso dos pacotes de dados e o número médio de pacotes perdido para cada valor diferente de f .

O código gerado para a resolução do exercício é o seguinte:

```
1 fprintf('Task 1 - Alinea B\n');
2 P = 10000;
3 lambda = 1800;
4 C = 10;
5 farray = [100000, 20000, 10000, 2000];
6 for index=1:numel(farray)
7     N = 50;
8     alfa=0.1;
9     PL_lst = zeros(1,N);
10    APD_lst = zeros(1,N);
11    MPD_lst = zeros(1,N);
12    TT_lst = zeros(1,N);
13    for i = 1:N
14        [PL_lst(i),APD_lst(i),MPD_lst(i),TT_lst(i)] =
            Simulator1(lambda,C,farray(index),P);
15    end
16    fprintf('Valor %d de f :\n',farray(index));
17    % Calculate Average Packet Delay
```

```

18     APD = mean(APD_lst);
19     APD_conf = norminv(1-alfa/2)*sqrt(var(APD_lst)/N);
20     fprintf('Av. Packet Delay (ms)= %.2e +-%.2e\n',APD,
        APD_conf);
21     APD_Resultsb(index) = APD;
22     APD_Errob(index) = APD_conf;
23     % Calculate Average Packet Loss
24     PL = mean(PL_lst);
25     PL_conf = norminv(1-alfa/2)*sqrt(var(PL_lst)/N);
26     fprintf('Av. Packet Loss (%%)= %.2e +-%.2e\n',PL,
        PL_conf);
27     PL_Resultsb(index)= PL;
28     PL_Errob(index)= PL_conf;
29 end
30 figure(1);
31 bar(farray,APD_Resultsb);
32 grid on
33 xlabel("Queue size (Bytes)");
34 ylim([0 11])
35 ylabel("Packet Delay");
36 title(["Average Packet Delay - Alinea B"]);
37 hold on
38 er = errorbar(farray,APD_Resultsb,APD_Errob,APD_Errob);
39 er.Color = [0 0 0];
40 er.LineStyle = 'none';
41 hold off
42 figure(2);
43 bar(farray,PL_Resultsb);
44 grid on
45 xlabel("Queue size (Bytes)");
46 ylim([0 11])
47 ylabel("Packet Lost");
48 title(["Average Packet Lost - Alinea B"]);
49 hold on
50 er = errorbar(farray,PL_Resultsb,PL_Errob,PL_Errob);
51 er.Color = [0 0 0];
52 er.LineStyle = 'none';
53 hold off

```

1.b.2 Resultados e Conclusões

Perante este programa, é de esperar que com o aumento do tamanho da fila, o tempo médio de espera dos pacotes de dados aumente e a média de pacotes perdidos diminua. A fila de espera vai crescendo, fazendo com que os pacotes esperem mais para serem atendidos ao existirem mais pacotes na fila de espera,

ao mesmo tempo que há diminuição na perda de pacotes, pois a fila tem mais capacidade, como consequência do aumento do tamanho da fila.

Como é possível ver na Figura 1.3, com o aumento do tamanho da fila, o tempo médio de espera aumenta e a média de pacotes perdida diminui, como argumentado anteriormente.

```
Task 1 - Alinea B
Valor 100000 de f :
Av. Packet Delay (ms)= 4.28e+00 +-1.56e-01
Av. Packet Loss (%)= 0.00e+00 +-0.00e+00
Valor 20000 de f :
Av. Packet Delay (ms)= 4.01e+00 +-8.79e-02
Av. Packet Loss (%)= 1.38e-01 +-2.13e-02
Valor 10000 de f :
Av. Packet Delay (ms)= 2.96e+00 +-3.52e-02
Av. Packet Loss (%)= 1.00e+00 +-4.87e-02
Valor 2000 de f :
Av. Packet Delay (ms)= 9.55e-01 +-2.48e-03
Av. Packet Loss (%)= 1.05e+01 +-7.87e-02
```

Figura 1.3: Resultados numéricos da alínea B.

Partindo do Gráfico 1.4 gerado a partir do último script, é possível observar-se que o tamanho da fila influencia o tempo médio de espera dos pacotes. Quanto maior o tamanho da fila, maior o número de pacotes na fila de espera para serem transmitidos e, como consequência, o tempo de espera de cada pacote aumenta. Observando o Gráfico 1.5, conclui-se que o tamanho da fila influencia a média de pacotes perdidos. Quanto maior o tamanho da fila, maior a capacidade da mesma e, como tal, o número de pacotes perdidos diminui.

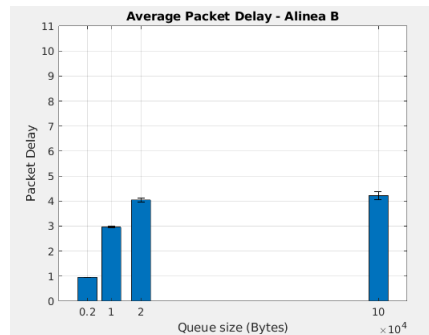


Figura 1.4: Resultados gráficos para a média de atraso de pacotes.

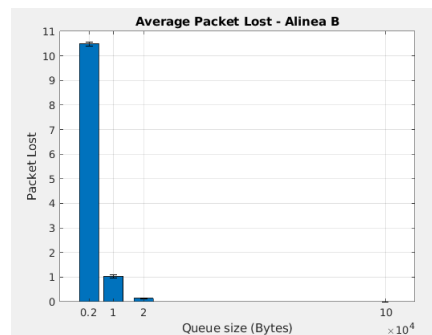


Figura 1.5: Resultados gráficos para a média de perda de pacotes.

Enquanto o tamanho da fila não influencia tão acentuadamente o atraso médio dos pacotes, o mesmo não se poderá confirmar para a média de pacotes perdidos. Para o valor mais baixo de tamanho da fila de 2000 bytes, o valor

aumenta aproximadamente 10x quando comparado com os restantes. No caso de o tamanho da fila ser 100000 bytes, o valor de perda é nulo.

Posto isto, é necessário atender aos requisitos do problema e decidir qual dos valores é mais fulcral para a resolução do mesmo, atendendo às suas necessidades. Se a perda de pacotes não for importante, deve-se optar por um valor de tamanho de fila baixo de forma a diminuir os valores de espera. Se por outro lado, o atraso nos pacotes não for importante, deve-se optar por um valor de tamanho de fila alto, de forma a diminuir os valores de pacotes perdidos. Ainda considerando que ambos são importantes, deve-se decidir um meio termo entre os valores, de forma a não prejudicar nenhum dos requisitos.

1.c Task 1.c - Enunciado

Consider the case of $\lambda = 1800$ pps and $f = 1.000.000$ Bytes. Run Simulator1 50 times with a stopping criterion of $P = 10000$ at each run and compute the estimated values and the 90% confidence intervals of the average delay performance parameter when $C = 10, 20, 30$ and 40 Mbps. Present the average packet delay results in bar charts with the confidence intervals in error bars. Justify the results and take conclusions concerning the impact of the link capacity in the obtained average packet delay.

1.c.1 Código

Tendo sido sugerido que se corresse 50 iterações do simulador 1 para quatro números diferentes de largura de banda de canais ($C = 10, 20, 30$ and 40 Mbps), com $f = 1.000.000$ Bytes, $\lambda = 1800$ pps, $P = 10000$ e com intervalos de confiança de 90% de desempenho de atraso médio dos pacotes, criou-se um script que obedecesse aos valores anteriormente descritos.

Criou-se uma matriz **Carray** com uma linha e quatro colunas, correndo para cada valor da matriz a simulação 50 vezes, guardando os valores de cada simulação para, a partir desses, calculou-se o tempo médio de atraso dos pacotes de dados para cada valor diferente de C .

O código gerado para a resolução do exercício é o seguinte:

```
1 fprintf( 'Task 1 – Alinea C\n' );
2 P = 10000;
3 lambda = 1800;
4 Carray = [10, 20, 30, 40];
5 f = 1000000;
6 for index=1:numel( Carray )
7     N = 50;
8     alfa=0.1;
9     PL_lst = zeros(1,N);
10    APD_lst = zeros(1,N);
11    MPD_lst = zeros(1,N);
12    TT_lst = zeros(1,N);
13    for i = 1:N
14        [PL_lst(i),APD_lst(i),MPD_lst(i),TT_lst(i)] =
            Simulator1( lambda, Carray(index), f, P );
15    end
16    fprintf( 'Valor %d de C:\n', Carray(index) );
17    % Calculate Average Packet Delay
18    APD = mean( APD_lst );
19    APD_conf = norminv(1-alfa/2)*sqrt( var( APD_lst )/N );
```

```

20     fprintf('Av. Packet Delay (ms)= %.2e +-%.2e\n',APD,
            APD_conf);
21     APD_Resultsc(index) = APD;
22     APD_Erroc(index) = APD_conf;
23 end
24 figure(1);
25 bar(Carray,APD_Resultsc);
26 grid on
27 xlabel("Link bandwidth (Mbps)");
28 ylabel("Packet Delay");
29 title(["Average Packet Delay"]);
30 hold on
31 er = errorbar(Carray,APD_Resultsc,APD_Erroc,APD_Erroc);
32 er.Color = [0 0 0];
33 er.LineStyle = 'none';
34 hold off

```

1.c.2 Resultados e Conclusões

Perante este programa, é de esperar que com o aumento da largura de banda de canais o tempo médio de espera dos pacotes de dados diminua. A largura de banda vai crescendo, fazendo com que mais pacotes sejam transmitidos ao mesmo tempo, resultando numa diminuição da fila de espera e, como consequência, uma diminuição do atraso médio dos pacotes.

Como é possível ver na Figura 1.6, com o aumento da largura de banda, o tempo médio de espera diminui, como argumentado anteriormente.

```

Task 1 - Alinea C
Valor 10 de C:
Av. Packet Delay (ms)= 4.27e+00 +-1.24e-01
Valor 20 de C:
Av. Packet Delay (ms)= 4.37e-01 +-2.21e-03
Valor 30 de C:
Av. Packet Delay (ms)= 2.31e-01 +-8.07e-04
Valor 40 de C:
Av. Packet Delay (ms)= 1.58e-01 +-5.50e-04

```

Figura 1.6: Resultados gráficos da alínea C.

Partindo do Gráfico 1.7 gerado a partir do último script, é possível observar que a largura de banda do canal influencia o tempo médio de espera dos pacotes. Quanto maior a largura de banda do canal, menor o número de pacotes na fila de espera para serem transmitidos e, como consequência, o tempo de espera de cada pacote diminui.

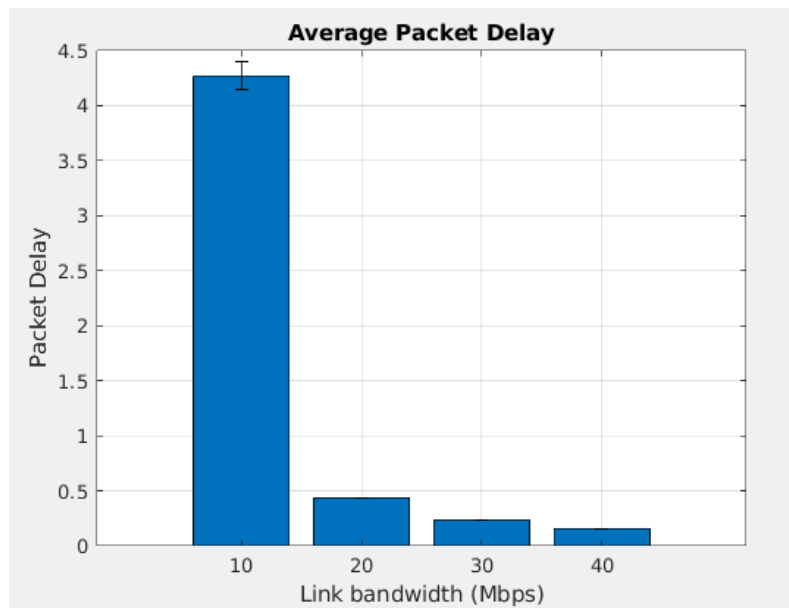


Figura 1.7: Resultados gráficos da alínea C.

Concluindo, o valor mais baixo de largura de banda de 10 Mbps aumenta o valor do atraso em 9 vezes, quando comparado com o tamanho de 20 Mbps. Neste caso, não faz sentido escolher valores aproximadamente menores que 20 Mbps, pois o atraso de pacotes para esses mesmos valores é muito significativo.

1.d Task 1.d - Enunciado

Consider that the system is modelled by a M/G/1 queueing model. Determine the theoretical values of the average packet delay using the M/G/1 model for all cases of 1.c. Compare the theoretical values with the simulation results of experiments 1.c and take conclusions.

1.d.1 Código

Tendo sido sugerido que se corresse 50 iterações do simulador 1 para quatro números diferentes de largura de banda de canais ($C = 10, 20, 30$ and 40 Mbps), usando o modelo M/G/1, criou-se um script que obedecesse aos valores anteriormente descritos.

Criou-se uma matriz *Carray* com uma linha e quatro colunas, correndo para cada valor da matriz o script e, a partir desses, calculou-se o tempo médio de atraso dos pacotes de dados para cada valor de C . Os valores usados para o modelo foram os lecionados na disciplina.

O código gerado para a resolução do exercício é o seguinte:

```
1 fprintf('Task 1 - Alinea D\n');
2 lambda = 1800;
3 Carray = [10*10^6, 20*10^6, 30*10^6, 40*10^6];
4 packetSize = 64:1518;
5 prob = zeros(1,1518);
6 prob(packetSize) = (1 - 0.19 - 0.23 - 0.17) / (length(
    packetSize)-3);
7 prob(64) = 0.19;
8 prob(110) = 0.23;
9 prob(1518) = 0.17;
10 for index=1:numel(Carray)
11     Spacket = packetSize.*8./Carray(index);
12     Spacket2= Spacket.^2;
13     E = sum(prob(packetSize).*Spacket);
14     E_2 = sum(prob(packetSize).*Spacket2);
15     avgPacketSize = sum(prob(packetSize).*packetSize);
16     AvPacketDelay = (((lambda.*E_2) ./ (2*(1-lambda.*E)))
        +E)*10^3; %converter em ms
17     fprintf('o valor de C: %.2e \n',Carray(index));
18     fprintf('AvPacketDelay= %.2e \n',AvPacketDelay);
19 end
```


1.d.2 Resultados e Conclusões

Perante este programa, é de esperar que com o aumento da largura de banda de canais, o tempo médio de espera dos pacotes de dados diminua. A largura de banda vai crescendo, fazendo com que mais pacotes sejam transmitidos ao mesmo tempo, resultando numa diminuição do número de pacotes na fila de espera e, como consequência, uma diminuição do atraso médio dos pacotes.

Como é possível ver na Figura 1.6, com o aumento da largura de banda, o tempo médio de espera diminui, como argumentado anteriormente.

```
Task 1 - Alínea D
o valor de C: 1.00e+07
AvPacketDelay= 4.39e+00
o valor de C: 2.00e+07
AvPacketDelay= 4.36e-01
o valor de C: 3.00e+07
AvPacketDelay= 2.31e-01
o valor de C: 4.00e+07
AvPacketDelay= 1.58e-01
```

Figura 1.8: Resultados numéricos da alínea D.

Comparando os resultados com a alínea anterior, é possível verificar que os valores teóricos e os da simulação são congruentes e, como tal, as conclusões anteriormente tiradas estão corretas.

1.e Task 1.e - Enunciado

Develop a new version of Simulator1 to estimate 3 additional performance parameters: the average packet delay of the packets of size 64, 110 and 1518 Bytes, respectively. Consider the case of $\lambda = 1800$ pps and $f = 1.000.000$. Run the new version of Simulator1 50 times with a stopping criterion of $P = 10000$ at each run and compute the estimated values and the 90% confidence intervals of the 3 new average delay performance parameters when $C = 10, 20, 30$ and 40 Mbps. Present the average packet delay results in bar charts with the confidence intervals in error bars. Justify these results and the differences between them and the results of 1.c. Take conclusions concerning the impact of the link capacity in the obtained average packet delay of packets with different sizes.

1.e.1 Código

Para desenvolver a nova versão do Simulador 1 que estimasse os parâmetros de performance da média de atrasos de pacotes de tamanho de 64, 110 e 1518 bytes desenvolveu-se um novo simulador 1. Este pode ser observado a seguir.

```
1 function [PL , APD,APD_64,APD_110,APD_1518 , MPD , TT] =  
    Simulator1New(lambda,C,f,P)  
2 %Events:  
3 ARRIVAL= 0;          % Arrival of a packet  
4 DEPARTURE= 1;        % Departure of a packet  
5 %State variables:  
6 STATE = 0;           % 0 – connection free; 1 – connection  
    bysy  
7 QUEUEOCCUPATION= 0; % Occupation of the queue (in Bytes)  
8 QUEUE= [];           % Size and arriving time instant of  
    each packet in the queue  
9 %Statistical Counters:  
10 TOTALPACKETS= 0;     % No. of packets arrived to the  
    system  
11 LOSTPACKETS= 0;       % No. of packets dropped due to  
    buffer overflow  
12 TRANSMITTEDPACKETS= 0; % No. of transmitted packets  
13 TRANSMITTEDPACKETS_64= 0; % No. of transmitted packets  
    with 64 packetSize  
14 TRANSMITTEDPACKETS_110= 0; % No. of transmitted packets  
    with 110 packetSize  
15 TRANSMITTEDPACKETS_1518= 0; % No. of transmitted packets  
    with 1518 packetSize
```

```

16 TRANSMITTEDBYTES= 0; % Sum of the Bytes of transmitted
    packets
17 DELAYS= 0; % Sum of the delays of transmitted
    packets
18 DELAYS_64= 0; % Sum of the delays of transmitted
    packets with 64 packetSize
19 DELAYS_110= 0; % Sum of the delays of transmitted
    packets with 110 packetSize
20 DELAYS_1518= 0; % Sum of the delays of transmitted
    packets with 1518 packetSize
21 MAXDELAY= 0; % Maximum delay among all
    transmitted packets
22 % Initializing the simulation clock:
    Clock= 0;
23 % Initializing the List of Events with the first ARRIVAL:
    tmp= Clock + exprnd(1/lambda);
24 EventList = [ARRIVAL, tmp, GeneratePacketSize(), tmp];
25 %Simulation loop:
26 while TRANSMITTEDPACKETS<P % Stopping
    criterium
27     EventList= sortrows(EventList,2); % Order
        EventList by time
28     Event= EventList(1,1); % Get first
        event and
29     Clock= EventList(1,2); % and
        PacketSize= EventList(1,3); % associated
        ArrivalInstant= EventList(1,4); % parameters.
        EventList(1,:)= []; % Eliminate
        first event
30 switch Event
31     case ARRIVAL % If first event
        is an ARRIVAL
32         TOTALPACKETS= TOTALPACKETS+1;
33         tmp= Clock + exprnd(1/lambda);
34         EventList = [EventList; ARRIVAL, tmp,
            GeneratePacketSize(), tmp];
35         if PacketSize==64
36             if STATE==0
37                 STATE= 1;
38                 EventList = [EventList; DEPARTURE,
                    Clock + 8*PacketSize/(C*10^6),
                    PacketSize, Clock];
39             else
40                 if QUEUEOCCUPATION + PacketSize <= f
41                     QUEUE= [QUEUE; PacketSize, Clock
42 ];

```

```

47         QUEUEOCCUPATION= QUEUEOCCUPATION
           + PacketSize;
48     else
49         LOSTPACKETS= LOSTPACKETS + 1;
50     end
51 end
52 elseif PacketSize==110
53     if STATE==0
54         STATE= 1;
55         EventList = [ EventList; DEPARTURE,
           Clock + 8*PacketSize/(C*10^6),
           PacketSize , Clock];
56     else
57         if QUEUEOCCUPATION + PacketSize <= f
58             QUEUE= [QUEUE; PacketSize , Clock
           ];
59             QUEUEOCCUPATION= QUEUEOCCUPATION
           + PacketSize;
60         else
61             LOSTPACKETS= LOSTPACKETS + 1;
62         end
63     end
64 elseif PacketSize==1518
65     if STATE==0
66         STATE= 1;
67         EventList = [ EventList; DEPARTURE,
           Clock + 8*PacketSize/(C*10^6),
           PacketSize , Clock];
68     else
69         if QUEUEOCCUPATION + PacketSize <= f
70             QUEUE= [QUEUE; PacketSize , Clock
           ];
71             QUEUEOCCUPATION= QUEUEOCCUPATION
           + PacketSize;
72         else
73             LOSTPACKETS= LOSTPACKETS + 1;
74         end
75     end
76 end
77 case DEPARTURE % If first
78     event is a DEPARTURE
79     if PacketSize==64
80         TRANSMITTEDPACKETS_64=
           TRANSMITTEDPACKETS_64 + 1;
           DELAYS_64= DELAYS_64 + (Clock -
           ArrivalInstant);

```

```

81         elseif PacketSize==110
82             TRANSMITTEDPACKETS_110=
83                 TRANSMITTEDPACKETS_110 + 1;
84             DELAYS_110= DELAYS_110 + (Clock -
85                 ArrivalInstant);
86         elseif PacketSize==1518
87             TRANSMITTEDPACKETS_1518=
88                 TRANSMITTEDPACKETS_1518 + 1;
89             DELAYS_1518= DELAYS_1518 + (Clock -
90                 ArrivalInstant);
91         end
92         TRANSMITTEDBYTES= TRANSMITTEDBYTES +
93             PacketSize;
94         if PacketSize==64
95             DELAYS64= DELAYS + (Clock -
96                 ArrivalInstant);
97             if Clock - ArrivalInstant > MAXDELAY
98                 MAXDELAY= Clock - ArrivalInstant;
99             end
100             TRANSMITTEDPACKETS= TRANSMITTEDPACKETS +
101                 1;
102             if QUEUEOCCUPATION > 0
103                 EventList = [ EventList; DEPARTURE,
104                     Clock + 8*QUEUE(1,1)/(C*10^6),
105                     QUEUE(1,1), QUEUE(1,2) ];
106                 QUEUEOCCUPATION= QUEUEOCCUPATION -
107                     QUEUE(1,1);
108                 QUEUE(1,:) = [];
109             else
110                 STATE= 0;
111             end
112         elseif PacketSize == 110
113             DELAYS110= DELAYS + (Clock -
114                 ArrivalInstant);
115             if Clock - ArrivalInstant > MAXDELAY
116                 MAXDELAY= Clock - ArrivalInstant;
117             end
118             TRANSMITTEDPACKETS= TRANSMITTEDPACKETS +
119                 1;
120             if QUEUEOCCUPATION > 0
121                 EventList = [ EventList; DEPARTURE,
122                     Clock + 8*QUEUE(1,1)/(C*10^6),
123                     QUEUE(1,1), QUEUE(1,2) ];
124                 QUEUEOCCUPATION= QUEUEOCCUPATION -
125                     QUEUE(1,1);
126                 QUEUE(1,:) = [];

```

```

112         else
113             STATE= 0;
114         end
115     elseif PacketSize == 1518
116         DELAYS1158= DELAYS + (Clock -
117             ArrivalInstant);
118         if Clock - ArrivalInstant > MAXDELAY
119             MAXDELAY= Clock - ArrivalInstant;
120         end
121         TRANSMITTEDPACKETS= TRANSMITTEDPACKETS +
122             1;
123         if QUEUEOCCUPATION > 0
124             EventList = [ EventList; DEPARTURE,
125                 Clock + 8*QUEUE(1,1)/(C*10^6),
126                 QUEUE(1,1), QUEUE(1,2) ];
127             QUEUEOCCUPATION= QUEUEOCCUPATION -
128                 QUEUE(1,1);
129             QUEUE(1,:) = [];
130         else
131             STATE= 0;
132         end
133     end
134 end
135
136 %Performance parameters determination:
137 PL= 100*LOSTPACKETS/TOTALPACKETS; % in %
138 APD= 1000*DELAYS/TRANSMITTEDPACKETS; % in milliseconds
139 APD_64= 1000*DELAYS_64/TRANSMITTEDPACKETS_64;
140 APD_110= 1000*DELAYS_110/TRANSMITTEDPACKETS_110;
141 APD_1518= 1000*DELAYS_1518/TRANSMITTEDPACKETS_1518;
142 MPD= 1000*MAXDELAY; % in milliseconds
143 TT= 10^(-6)*TRANSMITTEDBYTES*8/Clock; % in Mbps
144 end
145 function out= GeneratePacketSize()
146     aux= rand();
147     aux2= [65:109 111:1517];
148     if aux <= 0.19
149         out= 64;
150     elseif aux <= 0.19 + 0.23
151         out= 110;
152     elseif aux <= 0.19 + 0.23 + 0.17
153         out= 1518;
154     else
155         out = aux2(randi(length(aux2)));
156     end
157 end

```

Tendo sido sugerido que se corresse 50 iterações do simulador 1 para quatro números diferentes de largura de banda de canais ($C = 10, 20, 30$ and 40 Mbps), com $f = 1.000.000$ Bytes, $\lambda = 1800$ pps, $P = 10000$ e com intervalos de confiança de 90% de desempenho de atraso médio dos pacotes, criou-se um script que obedecesse aos valores anteriormente descritos.

Criou-se uma matriz **Carray** com uma linha e quatro colunas, correndo para cada valor da matriz a simulação 50 vezes, guardando os valores de cada simulação para, a partir desses, calculou-se o tempo de atraso médio dos pacotes de dados para cada valor diferente de C .

O código gerado para a resolução do exercício é o seguinte:

```

1 fprintf('Task 1 - Alinea E\n');
2 P = 10000;
3 lambda = 1800;
4 Carray = [10, 20, 30, 40];
5 f = 1000000;
6 for index=1:numel(Carray)
7     N = 50;
8     alfa=0.1;
9     PL_lst = zeros(1,N);
10    APD_lst = zeros(1,N);
11    APD_lst_64 = zeros(1,N);
12    APD_lst_110 = zeros(1,N);
13    APD_lst_1518 = zeros(1,N);
14    MPD_lst = zeros(1,N);
15    TT_lst = zeros(1,N);
16    for i = 1:N
17        [PL_lst(i),APD_lst(i),APD_lst_64(i),APD_lst_110(i),
18            APD_lst_1518(i),MPD_lst(i),TT_lst(i)] =
19            Simulator1New(lambda,Carray(index),f,P);
20    end
21    fprintf('Valor %d de C:\n',Carray(index));
22    % Calculate Average Packet Delay
23    % pode dar jeito para comparar os resultados por isso
24    % nao apaguei
25    %APD = mean(APD_lst);
26    %APD_conf = norminv(1-alfa/2)*sqrt(var(APD_lst)/N);
27    %fprintf('Av. Packet Delay (ms)= %.2e +-%.2e\n',APD,
28        APD_conf);
29    APD_64 = mean(APD_lst_64);
30    APD_conf_64 = norminv(1-alfa/2)*sqrt(var(APD_lst_64)/
31        N);
32    fprintf('Av. Packet Delay with size 64 (ms)= %.2e
33        +-%.2e\n',APD_64,APD_conf_64);
34    APD_110 = mean(APD_lst_110);
35    APD_conf_110 = norminv(1-alfa/2)*sqrt(var(APD_lst_110

```

```

    )/N);
30     fprintf('Av. Packet Delay with size 110 (ms)= %.2e
        +-%.2e\n',APD_110,APD_conf_110);
31     APD_1518 = mean(APD_lst_1518);
32     APD_conf_1518 = norminv(1-alfa/2)*sqrt(var(
        APD_lst_1518)/N);
33     fprintf('Av. Packet Delay with size 1518 (ms)= %.2e
        +-%.2e\n',APD_1518,APD_conf_1518);
34     APD_Resultsc_64(index) = APD_64;
35     APD_Erroc_64(index) = APD_conf_64;
36     APD_Resultsc_110(index) = APD_110;
37     APD_Erroc_110(index) = APD_conf_110;
38     APD_Resultsc_1518(index) = APD_1518;
39     APD_Erroc_1518(index) = APD_conf_1518;
40 end
41 figure(1);
42 bar(Carray,APD_Resultsc_64);
43 grid on
44 xlabel("Link bandwidth (Mbps)");
45 ylim([0 6])
46 ylabel("Packet Delay");
47 title(["Average Packet Delay Size 64 - Alinea E"]);
48 hold on
49 er = errorbar(Carray,APD_Resultsc_64,APD_Erroc_64,
        APD_Erroc_64);
50 er.Color = [0 0 0];
51 er.LineStyle = 'none';
52 hold off
53 figure(2);
54 bar(Carray,APD_Resultsc_110);
55 grid on
56 xlabel("Link bandwidth (Mbps)");
57 ylim([0 6])
58 ylabel("Packet Delay");
59 title(["Average Packet Delay Size 110 - Alinea E"]);
60 hold on
61 er = errorbar(Carray,APD_Resultsc_110,APD_Erroc_110,
        APD_Erroc_110);
62 er.Color = [0 0 0];
63 er.LineStyle = 'none';
64 hold off
65 figure(3);
66 bar(Carray,APD_Resultsc_1518);
67 grid on
68 xlabel("Link bandwidth (Mbps)");
69 ylim([0 6])

```



```

70 ylabel("Packet Delay");
71 title(["Average Packet Delay Size 1518 - Alinea E"]);
72 hold on
73 er = errorbar(Carray,APD_Resultsc_1518,APD_Erroc_1518,
              APD_Erroc_1518);
74 er.Color = [0 0 0];
75 er.LineStyle = 'none';
76 hold off

```

1.e.2 Resultados e Conclusões

Perante este programa, é de esperar que com o aumento da largura de banda de canais, o tempo médio de espera dos pacotes de dados diminua. Contudo, com o aumento do tamanho dos pacotes, para a mesma largura de banda, o valor do tempo de espera aumenta visto que cada pacote ocupa mais largura de banda com o aumento do seu tamanho.

Como é possível verificar na Figura 1.9, à medida que a largura de banda vai crescendo, fazendo com que mais pacotes sejam transmitidos ao mesmo tempo e resultando numa diminuição da fila de espera que, como consequência, leva a uma diminuição do atraso médio dos pacotes, o aumento do tamanho dos pacotes faz com que haja um aumento do atraso dos mesmos.

```

Task 1 - Alinea E
Valor 10 de C:
Av. Packet Delay with size 64 (ms)= 4.50e-01 +-4.98e-03
Av. Packet Delay with size 110 (ms)= 4.85e-01 +-4.05e-03
Av. Packet Delay with size 1518 (ms)= 1.61e+00 +-5.96e-03
Valor 20 de C:
Av. Packet Delay with size 64 (ms)= 9.71e-02 +-9.42e-04
Av. Packet Delay with size 110 (ms)= 1.16e-01 +-8.66e-04
Av. Packet Delay with size 1518 (ms)= 6.80e-01 +-1.34e-03
Valor 30 de C:
Av. Packet Delay with size 64 (ms)= 4.66e-02 +-4.73e-04
Av. Packet Delay with size 110 (ms)= 5.91e-02 +-4.00e-04
Av. Packet Delay with size 1518 (ms)= 4.35e-01 +-4.16e-04
Valor 40 de C:
Av. Packet Delay with size 64 (ms)= 2.89e-02 +-2.79e-04
Av. Packet Delay with size 110 (ms)= 3.80e-02 +-2.47e-04
Av. Packet Delay with size 1518 (ms)= 3.20e-01 +-2.48e-04

```

Figura 1.9: Resultados numéricos da alínea E.

Partindo dos Gráficos 1.10, 1.11 e 1.12 gerados a partir do último script, é possível observar que tanto a largura de banda do canal e o tamanho dos pacotes influenciam o tempo médio de espera dos pacotes. Quanto maior a largura de banda do canal, menor o número de pacotes na fila de espera para serem transmitidos e, como consequência, o tempo de espera de cada pacote diminui.

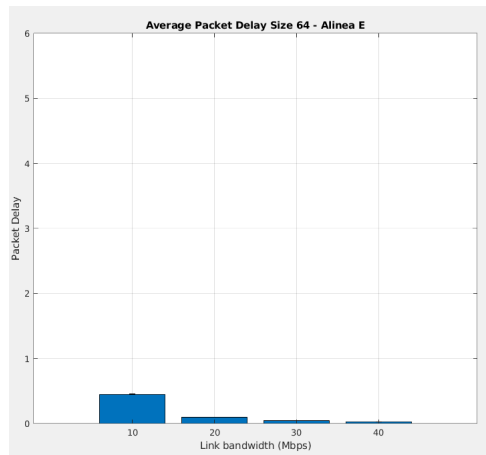


Figura 1.10: Resultados gráficos da alínea E para pacotes com tamanho de 64 bytes.

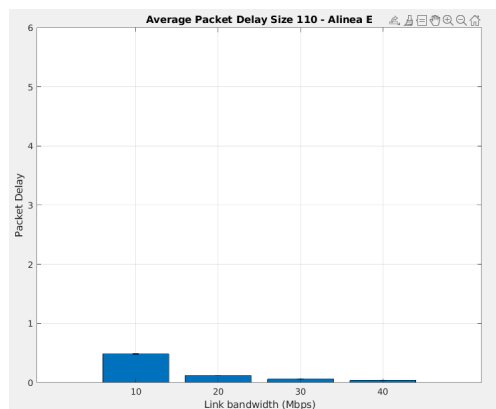


Figura 1.11: Resultados gráficos da alínea E para pacotes com tamanho de 110 bytes.

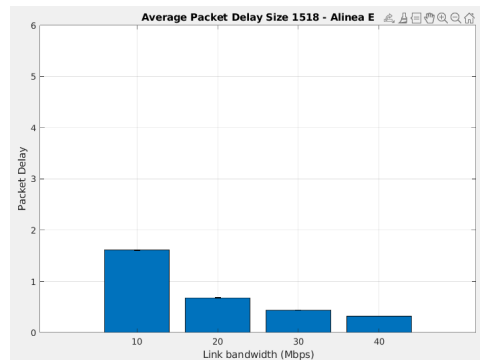


Figura 1.12: Resultados gráficos da alínea E para pacotes com tamanho de 1518 bytes.

Comparando os resultados com a alínea C, verificamos que a diferença dos resultados encontra-se no tamanho dos pacotes usados. Enquanto a alínea C usava qualquer valor de pacotes entre 64 e 1518, a alínea E usa apenas pacotes com tamanhos de 64, 110 e 1518 bytes. Como os pacotes para os últimos valores descritos tem probabilidade diferente dos restantes, ao analisarmos apenas os valores pedidos o tempo médio de atraso irá ser diferente.

Capítulo 2

Task 2

Neste exercício considerou-se os Simuladores 3 e 4 usado na Task 7 nos trabalhos da aula prática.

2.a Task 2.a - Enunciado

Consider the case of $\lambda = 1500$ pps, $C = 10$ Mbps and $f = 1.000.000$ Bytes. Run Simulator3 50 times with a stopping criterion of $P = 10000$ each run and compute the estimated values and the 90% confidence intervals of the average delay performance parameter of data packets and VoIP packets when $n = 10, 20, 30$ and 40 VoIP flows. Present the average data packet delay results in one figure and the average VoIP packet delay results in another figure (in both cases, in bar charts with the confidence intervals in error bars). Justify the results and take conclusions concerning the impact of the number of VoIP flows in the obtained average packet delay of each service when both services (data and VoIP) are statistically multiplexed in a single FIFO queue.

2.a.1 Código

Foi pedido que se corresse 50 iterações do simulador 3 para quatro números diferentes de fluxos de pacotes VoIP ($n = 10, 20, 30$ e 40), sendo que $\lambda = 1500$ pps, $C = 10$ Mbps, $f = 1.000.000$ Bytes, $P = 10000$ e com intervalos de confiança de 90% de desempenho de atraso médio dos pacotes de dados.

Para tal, criou-se uma matriz **n** com uma linha e quatro colunas. Assim, para cada valor desta matriz correu-se a simulação cinquenta vezes, guardando os valores de cada simulação. Calculou-se o tempo médio de espera dos pacotes

de dados e do tipo VoIP e os seus intervalos de erro associado a esse tempo médio.

A partir daqui apresentou-se o tempo de atraso médio dos pacotes de dados e do tipo VoIP em gráficos de barras com os intervalos de confiança nas barras de erro.

O código gerado para a resolução do exercício é o seguinte:

```

1 fprintf('Task 2 – Alinea A\n');
2 P = 10000;
3 lambda = 1500;
4 C = 10;
5 f = 1000000;
6 n = [10,20,30,40];
7 for index=1:numel(n)
8     N = 50;
9     alfa=0.1; % 90% confidence intervals
10    PL_lst_data = zeros(1,N);
11    APD_lst_data = zeros(1,N);
12    MPD_lst_data = zeros(1,N);
13    PL_lst_VoIP = zeros(1,N);
14    APD_lst_VoIP = zeros(1,N);
15    MPD_lst_VoIP = zeros(1,N);
16    TT_lst = zeros(1,N);
17    for i = 1:N
18        [PL_lst_data(i),PL_lst_VoIP(i),APD_lst_data(i),
19            APD_lst_VoIP(i),MPD_lst_data(i),MPD_lst_VoIP(i),
20            TT_lst(i)] = Simulator3(lambda,C,f,P,n(index));
21    end
22    fprintf('Valor %d de n for VoIP:\n',n(index));
23    % Calculate Average Data Packet Delay
24    APD_data = mean(APD_lst_data);
25    APD_conf_data = norminv(1-alfa/2)*sqrt(var(
26        APD_lst_data)/N);
27    fprintf('Data Av. Packet Delay (ms)= %.2e +-%.2e\n',
28        APD_data,APD_conf_data);
29    % Calculate Average VoIP Packet Delay
30    APD_VoIP = mean(APD_lst_VoIP);
31    APD_conf_VoIP = norminv(1-alfa/2)*sqrt(var(
32        APD_lst_VoIP)/N);
33    fprintf('VoIP Av. Packet Delay (ms)= %.2e +-%.2e\n',
34        APD_VoIP,APD_conf_VoIP);
35    APD_Results_data(index) = APD_data;
36    APD_Erro_data(index) = APD_conf_data;
37    APD_Results_VoIP(index) = APD_VoIP;
38    APD_Erro_VoIP(index) = APD_conf_VoIP;

```

```

33 end
34 % figure for Average Data Packet Delay
35 figure(1);
36 bar(n,APD_Results_data);
37 title("Data Av. Packet Delay (ms) – Alinea A");
38 grid on
39 xlabel("Number of VoIP packets flows");
40 ylim([0 7])
41 ylabel("Data Paket Delay");
42 hold on
43 er1 = errorbar(n,APD_Results_data,APD_Erro_data,
44               APD_Erro_data);
45 er1.Color = [0 0 0];
46 er1.LineStyle = 'none';
47 hold off
48 % figure for Average VoIP Packet Delay
49 figure(2);
50 bar(n,APD_Results_VoIP);
51 grid on
52 title("VoIP Av. Packet Delay (ms) – Alinea A");
53 xlabel("Number of VoIP packets flows");
54 ylim([0 7])
55 ylabel("VoIP Paket Delay");
56 hold on
57 er2 = errorbar(n,APD_Results_VoIP,APD_Erro_VoIP,
58               APD_Erro_VoIP);
59 er2.Color = [0 0 0];
60 er2.LineStyle = 'none';
61 hold off

```

2.a.2 Resultados e Conclusões

Perante este programa, é de esperar que com o aumento do número de fluxos de pacotes VoIP o tempo médio de espera, quer dos pacotes de dados, quer dos pacotes VoIP, aumente. A fila de espera vai enchendo e quanto mais cheia estiver, mais tempo os pacotes vão esperar para serem atendidos.

Como é possível ver na Figura 2.1, com o aumento do número de fluxos de pacotes VoIP o tempo médio de espera aumenta, quer nos pacotes de dados, quer nos pacotes VoIP, como dito anteriormente.

```

Task 2 - Alínea A
Valor 10 de n for VoIP:
Data Av. Packet Delay (ms)= 2.19e+00 +-3.39e-02
VoIP Av. Packet Delay (ms)= 1.77e+00 +-3.21e-02
Valor 20 de n for VoIP:
Data Av. Packet Delay (ms)= 2.69e+00 +-7.65e-02
VoIP Av. Packet Delay (ms)= 2.26e+00 +-7.22e-02
Valor 30 de n for VoIP:
Data Av. Packet Delay (ms)= 3.62e+00 +-1.64e-01
VoIP Av. Packet Delay (ms)= 3.17e+00 +-1.52e-01
Valor 40 de n for VoIP:
Data Av. Packet Delay (ms)= 6.03e+00 +-6.03e-01
VoIP Av. Packet Delay (ms)= 5.61e+00 +-5.97e-01

```

Figura 2.1: Resultados numéricos da alínea A.

A partir dos Gráficos 2.2 e 2.3, é possível observar que quando ambos os serviços (dados e VoIP) são estatisticamente multiplexados numa única fila FIFO, o número de fluxo de pacotes VoIP influencia o tempo médio de espera, quer dos pacotes de dados, quer dos pacotes VoIP. Quanto maior for o número de fluxo de pacotes VoIP, existirá mais pacotes à espera (na fila de espera) para serem transmitidos, ou seja, a fila fica cada vez mais cheia com o aumento do fluxo de pacotes VoIP, fazendo com que o tempo de espera de cada pacote aumente.

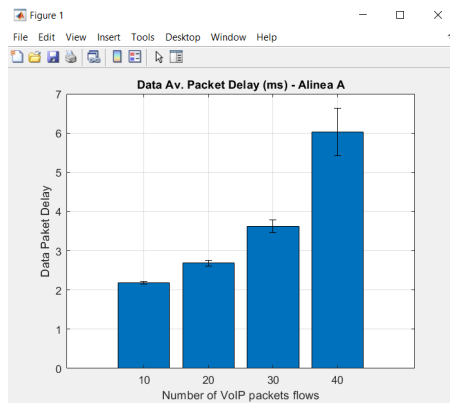


Figura 2.2: Tempo médio de espera de pacotes de dados.

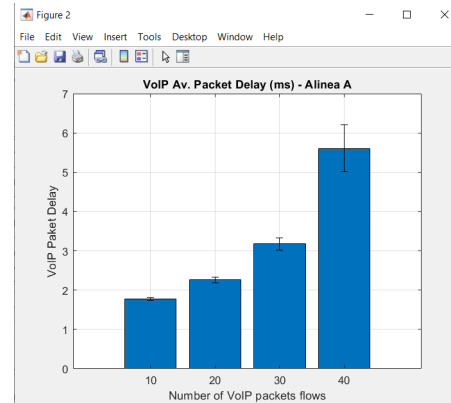


Figura 2.3: Tempo médio de espera de pacotes VoIP.

Concluindo, e comparando ambos os gráficos, não existe diferença significativa nos atrasos de ambos os pacotes para os valores obtidos. Se o objetivo for minimizar o tempo de atraso de ambos os pacotes, quanto menor o valor de fluxo de pacotes, menor o tempo de atraso, tanto para pacotes de dados, como para pacotes VoIP.

2.b Task 2.b - Enunciado

Repeat experiment 2.a but now with Simulator4. Justify these results and the differences between them and the results of 2.a. Take conclusions concerning the impact of the number of VoIP flows in the obtained average packet delay of each service when VoIP service is supported with a priority which is higher than the data service.

2.b.1 Código

Foi pedido que se corresse 50 iterações do simulador 4 para quatro números diferentes de fluxos de pacotes VoIP ($n = 10, 20, 30$ e 40), sendo que $\lambda = 1500$ pps, $C = 10$ Mbps, $f = 1.000.000$ Bytes, $P = 10000$ e com intervalos de confiança de 90% de desempenho de atraso médio dos pacotes de dados.

Para tal, criou-se uma matriz \mathbf{n} com uma linha e quatro colunas. Assim, para cada valor desta matriz correu-se a simulação cinquenta vezes, guardando os valores de cada simulação. Calculou-se o tempo médio de espera dos pacotes de dados e do tipo VoIP e os seus intervalos de erro associado a esse tempo médio.

A partir daqui, apresentou-se o tempo de atraso médio dos pacotes de dados e do tipo VoIP em gráficos de barras com os intervalos de confiança nas barras de erro.

O código gerado para a resolução do exercício é o seguinte:

```
1 fprintf('Task 2 – Alinea B\n');
2 P = 10000;
3 lambda = 1500;
4 C = 10;
5 f = 1000000;
6 n = [10,20,30,40];
7 for index=1:numel(n)
8     N = 50;
9     alfa=0.1; % 90% confidence intervals
10    PL_lst_data = zeros(1,N);
11    APD_lst_data = zeros(1,N);
12    MPD_lst_data = zeros(1,N);
13    PL_lst_VoIP = zeros(1,N);
14    APD_lst_VoIP = zeros(1,N);
15    MPD_lst_VoIP = zeros(1,N);
16    TT_lst = zeros(1,N);
17    for i = 1:N
18        [PL_lst_data(i),PL_lst_VoIP(i),APD_lst_data(i),
            APD_lst_VoIP(i),MPD_lst_data(i),MPD_lst_VoIP(i),
            TT_lst(i)] = Simulator4(lambda,C,f,P,n(index));
```



```

19     end
20     fprintf('Valor %d de n for VoIP:\n',n(index));
21     % Calculate Average Data Packet Delay
22     APD_data = mean(APD_lst_data);
23     APD_conf_data = norminv(1-alfa/2)*sqrt(var(
        APD_lst_data)/N);
24     fprintf('Data Av. Packet Delay (ms)= %.2e +-%.2e\n',
        APD_data,APD_conf_data);
25     % Calculate Average VoIP Packet Delay
26     APD_VoIP = mean(APD_lst_VoIP);
27     APD_conf_VoIP = norminv(1-alfa/2)*sqrt(var(
        APD_lst_VoIP)/N);
28     fprintf('VoIP Av. Packet Delay (ms)= %.2e +-%.2e\n',
        APD_VoIP,APD_conf_VoIP);
29     APD_Results_data(index) = APD_data;
30     APD_Erro_data(index) = APD_conf_data;
31     APD_Results_VoIP(index) = APD_VoIP;
32     APD_Erro_VoIP(index) = APD_conf_VoIP;
33 end
34 % figure for Average Data Packet Delay
35 figure(1);
36 bar(n,APD_Results_data);
37 title("Data Av. Packet Delay (ms) - Alinea B");
38 grid on
39 xlabel("Number of VoIP packets flows");
40 ylim([0 8])
41 ylabel("Data Paket Delay");
42 hold on
43 er1 = errorbar(n,APD_Results_data,APD_Erro_data,
        APD_Erro_data);
44 er1.Color = [0 0 0];
45 er1.LineStyle = 'none';
46 hold off
47 % figure Average VoIP Packet Delay
48 figure(2);
49 bar(n,APD_Results_VoIP);
50 grid on
51 title("VoIP Av. Packet Delay (ms) - Alinea B");
52 xlabel("Number of VoIP packets flows");
53 ylim([0 8])
54 ylabel("VoIP Paket Delay");
55 hold on
56 er2 = errorbar(n,APD_Results_VoIP,APD_Erro_VoIP,
        APD_Erro_VoIP);
57 er2.Color = [0 0 0];
58 er2.LineStyle = 'none';

```

2.b.2 Resultados e Conclusões

Observando a simulação 4, pode-se notar uma diferença entre esta e a simulação 3. Na simulação 4, os pacotes do tipo VoIP na fila de espera têm prioridade em relação aos pacotes de dados, por isso é de esperar que o tempo médio de espera dos pacotes do tipo VoIP diminua e o tempo médio de espera dos pacotes de dados aumente, quando comparado com a alínea anterior. Também se poderá esperar que o aumento do número de fluxos de pacotes VoIP não influencie tanto o tempo médio de espera dos pacotes VoIP como influenciará o tempo médio de espera dos pacotes de dados, devido ao facto dos pacotes VoIP serem prioritários em relação aos de dados.

Como é possível ver na Figura 2.4, ao usar o simulador 4 em vez do 3, observou-se que o tempo médio de espera nos pacotes de dados é sempre maior do que nos pacotes VoIP. Com o aumento do número de fluxos de pacotes VoIP, o tempo médio de espera dos pacotes de dados aumentou bastante mais do que o tempo médio de espera dos pacotes VoIP, ou seja, a diferença no tempo médio de espera com o aumento do número de fluxos de pacotes VoIP notou-se mais no tempo médio de espera dos pacotes de dados, como dito anteriormente.

```
Task 2 - Alinea B
Valor 10 de n for VoIP:
Data Av. Packet Delay (ms)= 2.24e+00 +-3.86e-02
VoIP Av. Packet Delay (ms)= 4.63e-01 +-2.47e-03
Valor 20 de n for VoIP:
Data Av. Packet Delay (ms)= 2.93e+00 +-7.80e-02
VoIP Av. Packet Delay (ms)= 4.84e-01 +-1.92e-03
Valor 30 de n for VoIP:
Data Av. Packet Delay (ms)= 4.18e+00 +-2.06e-01
VoIP Av. Packet Delay (ms)= 5.06e-01 +-2.36e-03
Valor 40 de n for VoIP:
Data Av. Packet Delay (ms)= 7.08e+00 +-5.63e-01
VoIP Av. Packet Delay (ms)= 5.37e-01 +-2.69e-03
```

Figura 2.4: Resultados numéricos da alínea B.

A partir dos Gráficos 2.5 e 2.6, é possível observar que quando ambos os serviços (dados e VoIP) são estatisticamente multiplexados numa única fila FIFO onde existe prioridade dos pacotes VoIP em relação aos pacotes de dados, o número de fluxo de pacotes VoIP influencia o tempo médio de espera, quer dos pacotes de dados, quer dos pacotes VoIP. Quanto maior for o número de fluxo de pacotes VoIP, existirá mais pacotes na fila de espera para serem transmitidos, ou seja, a fila fica cada vez mais cheia com o aumento do fluxo de pacotes VoIP, fazendo com que o tempo de espera de cada pacote aumente.

Para além disto, pode-se observar que o aumento do número de fluxo de pacotes VoIP influencia mais no tempo médio de espera nos pacotes de dados, do que nos pacotes VoIP (praticamente constante). Isto acontece porque os pacotes

VoIP têm mais prioridade que os pacotes de dados, ou seja, os pacotes VoIP são sempre atendidos primeiros do que os pacotes de dados (quando chegam são colocados à frente de todos os pacotes dados que houver na fila), por isso um pacote VoIP da fila só tem que esperar que os pacotes VoIP a sua frente sejam transmitidos. Por outro lado, os pacotes de dados têm que esperar que não exista nenhum pacote do tipo VoIP na fila de espera para ser transmitido.

Perante isto, sabe-se que com o aumento do número de fluxos de pacotes VoIP, é normal que o tempo médio de espera dos pacotes de dados aumente, pois existe mais pacotes VoIP a serem transmitidos primeiro que os pacotes de dados (mais pacotes VoIP a passar a frente na fila de espera dos pacotes de dados).

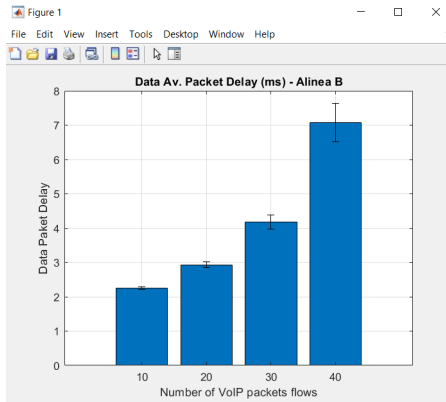


Figura 2.5: Tempo médio de espera de pacotes do tipo Data.

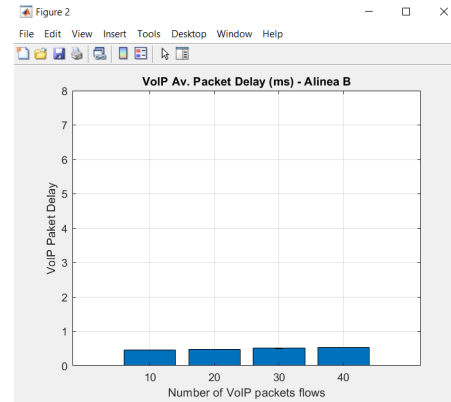


Figura 2.6: Tempo médio de espera de pacotes do tipo VoIP.

Concluindo, e comparando todos os gráficos, apesar de existir diferença significativa nos atrasos de ambos os pacotes para os valores obtidos, os valores aumentam com o aumento do fluxo de pacotes. Se o objetivo for minimizar o tempo de atraso de ambos os pacotes, como ambos aumentam com o aumento do fluxo dos pacotes VoIP, apesar de ser dito no enunciado que os pacotes VoIP têm prioridade sobre os pacotes de dados, o mais indicado seria o menor valor possível para o fluxo de pacotes em questão, pois ambos crescem com o aumento do fluxo.

2.c Task 2.c - Enunciado

Consider that the system is modelled by a M/G/1 queueing model with priorities. Determine the theoretical values of the average data packet delay and average VoIP packet delay using the M/G/1 model for all cases of 2.b. Compare the theoretical values with the simulation results of experiments 2.b and take conclusions.

2.c.1 Código

Tendo sido sugerido que se corresse 50 iterações do simulador 4 para quatro números diferentes de fluxos de pacotes VoIP ($n = 10, 20, 30$ e 40), usando o modelo M/G/1, criou-se um script que obedecesse aos valores anteriormente descritos.

Criou-se uma matriz \mathbf{n} com uma linha e quatro colunas, correndo para cada valor da matriz o script e, a partir desses, calculou-se o tempo médio de espera dos pacotes de dados e dos pacotes do tipo VoIP. Os valores usados para o modelo foram os lecionados na disciplina.

O código gerado para a resolução do exercício é o seguinte:

```
1
2 fprintf('Task 2 - Alinea C\n');
3
4 lambda = 1500; %pacotes data
5
6 C = 10*10^6;
7 packetSize = 64:1518;
8 prob = zeros(1,1518);
9 prob(packetSize) = (1 - 0.19 - 0.23 - 0.17) / (length(
    packetSize)-3);
10 prob(64) = 0.19;
11 prob(110) = 0.23;
12 prob(1518) = 0.17;
13
14 t=0.02; % cada pacote chega de 20ms em 20 ms
15 %lambda para os pacotes de VoIP
16 n = [10/t 20/t 30/t 40/t];
17
18 Spacket = (packetSize.*8)./C;
19 Spacket2= Spacket.^2;
20 E = sum(prob(packetSize).*Spacket);
21 E_2 = sum(prob(packetSize).*Spacket2);
22 avgPacketSize = sum(prob(packetSize).*packetSize);
23
24 p_data = lambda*E;
```

```

25
26 packetSizeVoip = 110:130;
27 avgPacketSizeVoip = sum(packetSizeVoip)/length(
    packetSizeVoip);
28 E_Voip = 1/(C/(avgPacketSizeVoip*8));
29 E_2_Voip = 2/((C/(avgPacketSizeVoip*8))^2);
30
31 for index=1:numel(n)
32
33     p_VoIP = n(index)*E_Voip;
34     w_Voip = (((n(index)*E_2_Voip+lambda*E_2) / (2*(1 -
        p_VoIP)) ) + E_Voip) * 10^3; % valor em ms
35     w_data = (((n(index)*E_2_Voip+lambda*E_2) / (2*(1 -
        p_VoIP)*(1-p_VoIP-p_data)) ) +E) * 10^3; % valor
        em ms
36     fprintf('o valor de n: %.2e \n',n(index));
37     fprintf('AvDataPacketDelay= %.2e \n',w_data);
38     fprintf('AvVoIPPacketDelay= %.2e \n',w_Voip);
39     fprintf('\n');
40 end

```

2.c.2 Resultados e Conclusões

Como podemos observar na Figura 2.7, os resultados obtidos são muito próximos com os obtidos na alínea anterior, pois nestes dois exercícios os pacotes do tipo VoIP têm prioridade sobre os pacotes de dados. Assim é de esperar que o tempo médio de espera dos pacotes VoIP seja menor que o tempo médio de espera dos pacotes de dados.

Com o aumento do número de fluxos de pacotes VoIP, o tempo médio de espera dos pacotes de dados aumentou bastante mais do que o tempo médio de espera dos pacotes VoIP, ou seja, a diferença no tempo médio de espera com o aumento do número de fluxos de pacotes VoIP notou-se mais no tempo médio de espera dos pacotes de dados.

```
Task 2 - Alinea C
o valor de n: 5.00e+02
AvDataPacketDelay= 2.27e+00
AvVoIPPacketDelay= 4.66e-01

o valor de n: 1.00e+03
AvDataPacketDelay= 2.96e+00
AvVoIPPacketDelay= 4.91e-01

o valor de n: 1.50e+03
AvDataPacketDelay= 4.27e+00
AvVoIPPacketDelay= 5.18e-01

o valor de n: 2.00e+03
AvDataPacketDelay= 7.58e+00
AvVoIPPacketDelay= 5.49e-01
```

Figura 2.7: Resultados numéricos da alínea C.

Concluindo, e comparando os resultados teóricos com os de simulação, os resultados são congruentes e provam a veracidade dos mesmos.

2.d Task 2.d - Enunciado

Consider the case of $\lambda = 1500$ pps, $C = 10$ Mbps and $f = 10.000$ Bytes. Run Simulator3 50 times with a stopping criterion of $P = 10000$ each run and compute the estimated values and the 90% confidence intervals of the average delay and packet loss performance parameters of data packets and VoIP packets when $n = 10, 20, 30$ and 40 VoIP flows. Present the results of each of the 4 performance parameters (average data packet delay, average VoIP packet delay, data packet loss and VoIP packet loss) in different figures (in all cases, in bar charts with the confidence intervals in error bars). Justify the results and take conclusions concerning the impact of the number of VoIP flows in the obtained average packet delay and packet loss of each service when both services (data and VoIP) are statistically multiplexed in a single FIFO queue of small size.

2.d.1 Código

Foi pedido que se corresse 50 iterações do simulador 3 para quatro números diferentes de fluxos de pacotes VoIP ($n = 10, 20, 30$ e 40), sendo que $\lambda = 1500$ pps, $C = 10$ Mbps, $f = 10.000$ Bytes, $P = 10000$ e com intervalos de confiança de 90% de desempenho de atraso médio dos pacotes de dados.

Para tal, criou-se uma matriz n com uma linha e quatro colunas. Assim, para cada valor desta matriz correu-se a simulação cinquenta vezes, guardando os valores de cada simulação. Calculou-se o tempo médio de espera dos pacotes de dados e do tipo VoIP e os seus intervalos de erro associado a esse tempo médio.

A partir daqui, apresentou-se o tempo de atraso médio e a perda de pacotes dos pacotes de dados e do tipo VoIP em gráficos de barras com os intervalos de confiança nas barras de erro.

O código gerado para a resolução do exercício é o seguinte:

```
1 fprintf('Task 2 - Alinea D\n');
2 P = 10000;
3 lambda = 1500;
4 C = 10;
5 f = 10000;
6 n = [10,20,30,40];
7 for index=1:numel(n)
8     N = 50;
9     alfa=0.1; % 90% confidence intervals
10    PL_lst_data = zeros(1,N);
11    APD_lst_data = zeros(1,N);
12    MPD_lst_data = zeros(1,N);
```

```

13     PL_lst_VoIP = zeros(1,N);
14     APD_lst_VoIP = zeros(1,N);
15     MPD_lst_VoIP = zeros(1,N);
16     TT_lst = zeros(1,N);
17     for i = 1:N
18         [PL_lst_data(i),PL_lst_VoIP(i),APD_lst_data(i),
            APD_lst_VoIP(i),MPD_lst_data(i),MPD_lst_VoIP(i)
            ),TT_lst(i)] = Simulator3(lambda,C,f,P,n(index
            ));
19     end
20     fprintf('Valor %d de n for VoIP:\n',n(index));
21     % Calculate Data Packet Loss
22     PL_data = mean(PL_lst_data);
23     PL_conf_data = norminv(1-alfa/2)*sqrt(var(PL_lst_data
        )/N);
24     fprintf('PacketLoss_Data (%%) = %.2e +- %.2e\n',
        PL_data,PL_conf_data)
25     % Calculate VoIP Packet Loss
26     PL_VoIP = mean(PL_lst_VoIP);
27     PL_conf_VoIP = norminv(1-alfa/2)*sqrt(var(PL_lst_VoIP
        )/N);
28     fprintf('PacketLoss_VoIP (%%) = %.2e +- %.2e\n',
        PL_VoIP,PL_conf_VoIP)
29     % Calculate Average Data Packet Delay
30     APD_data = mean(APD_lst_data);
31     APD_conf_data = norminv(1-alfa/2)*sqrt(var(
        APD_lst_data)/N);
32     fprintf('Data Av. Packet Delay (ms)= %.2e +-%.2e\n',
        APD_data,APD_conf_data);
33     % Calculate Average VoIP Packet Delay
34     APD_VoIP = mean(APD_lst_VoIP);
35     APD_conf_VoIP = norminv(1-alfa/2)*sqrt(var(
        APD_lst_VoIP)/N);
36     fprintf('VoIP Av. Packet Delay (ms)= %.2e +-%.2e\n',
        APD_VoIP,APD_conf_VoIP);
37     PL_Results_data(index) = PL_data;
38     PL_Erro_data(index) = PL_conf_data;
39     PL_Results_VoIP(index) = PL_VoIP;
40     PL_Erro_VoIP(index) = PL_conf_VoIP;
41     APD_Results_data(index) = APD_data;
42     APD_Erro_data(index) = APD_conf_data;
43     APD_Results_VoIP(index) = APD_VoIP;
44     APD_Erro_VoIP(index) = APD_conf_VoIP;
45     end
46     % figure for Calculate Data Packet Loss
47     figure(1);

```



```

48 bar(n, PL_Results_data);
49 title("Data Packet Loss (%) – Alinea D");
50 grid on
51 xlabel("Number of VoIP packets flows");
52 ylim([0 1.6])
53 ylabel("Data Paket Delay");
54 hold on
55 er1 = errorbar(n, PL_Results_data, PL_Erro_data,
    PL_Erro_data);
56 er1.Color = [0 0 0];
57 er1.LineStyle = 'none';
58 hold off
59 % figure for Calculate VoIP Packet Loss
60 figure(2);
61 bar(n, PL_Results_VoIP);
62 title("VoIP Packet Loss (%) – Alinea D");
63 grid on
64 xlabel("Number of VoIP packets flows");
65 ylim([0 1.6])
66 ylabel("VoIP Paket Delay");
67 hold on
68 er1 = errorbar(n, PL_Results_VoIP, PL_Erro_VoIP,
    PL_Erro_VoIP);
69 er1.Color = [0 0 0];
70 er1.LineStyle = 'none';
71 hold off
72 % figure for Average Data Packet Delay
73 figure(3);
74 bar(n, APD_Results_data);
75 title("Data Av. Packet Delay (ms) – Alinea D");
76 grid on
77 xlabel("Number of VoIP packets flows");
78 ylim([0 3.5])
79 ylabel("Data Paket Delay");
80 hold on
81 er1 = errorbar(n, APD_Results_data, APD_Erro_data,
    APD_Erro_data);
82 er1.Color = [0 0 0];
83 er1.LineStyle = 'none';
84 hold off
85 % figure for Average VoIP Packet Delay
86 figure(4);
87 bar(n, APD_Results_VoIP);
88 title("VoIP Av. Packet Delay (ms) – Alinea D");
89 grid on
90 xlabel("Number of VoIP packets flows");

```

```

91 ylim([0 3.5])
92 ylabel("VoIP Paket Delay");
93 hold on
94 er2 = errorbar(n,APD_Results_VoIP,APD_Erro_VoIP,
    APD_Erro_VoIP);
95 er2.Color = [0 0 0];
96 er2.LineStyle = 'none';
97 hold off

```

2.d.2 Resultados e Conclusões

Perante este programa é de esperar que com o aumento do número de fluxos de pacotes VoIP o tempo médio de espera, quer dos pacotes de dados, quer dos pacotes VoIP, aumente. A fila de espera vai enchendo e quanto mais cheia estiver, mais tempo os pacotes vão esperar para serem atendidos. Para além disto, como a fila de espera é mais pequena ($f = 10.000$ Bytes) também é esperado que com o aumento do número de pacotes esta fila ficará cada vez mais cheia. Atingindo o limite da capacidade da fila fará com que os pacotes que chegarem entretanto sejam descartados, ou seja, são pacotes perdidos.

Como é possível ver na Figura 2.8, com o aumento do número de fluxos de pacotes VoIP, o tempo médio de espera aumenta, quer nos pacotes de dados, quer nos pacotes VoIP, e o número de pacotes perdidos também aumenta, como dito anteriormente.

É possível observar que o número de pacotes perdidos do tipo VoIP é sempre menor que nos pacotes de dados, pois estes são em menor número.

```

Task 2 - Alinea D
Valor 10 de n for VoIP:
PacketLoss_Data (%) = 2.54e-01 +- 2.67e-02
PacketLoss_VoIP (%) = 2.63e-02 +- 9.30e-03
Data Av. Packet Delay (ms)= 2.02e+00 +-2.74e-02
VoIP Av. Packet Delay (ms)= 1.62e+00 +-2.79e-02
Valor 20 de n for VoIP:
PacketLoss_Data (%) = 4.13e-01 +- 3.34e-02
PacketLoss_VoIP (%) = 5.14e-02 +- 9.90e-03
Data Av. Packet Delay (ms)= 2.34e+00 +-3.33e-02
VoIP Av. Packet Delay (ms)= 1.94e+00 +-3.07e-02
Valor 30 de n for VoIP:
PacketLoss_Data (%) = 7.11e-01 +- 6.21e-02
PacketLoss_VoIP (%) = 9.31e-02 +- 1.77e-02
Data Av. Packet Delay (ms)= 2.65e+00 +-4.04e-02
VoIP Av. Packet Delay (ms)= 2.28e+00 +-3.83e-02
Valor 40 de n for VoIP:
PacketLoss_Data (%) = 1.42e+00 +- 9.23e-02
PacketLoss_VoIP (%) = 1.74e-01 +- 1.88e-02
Data Av. Packet Delay (ms)= 3.25e+00 +-5.58e-02
VoIP Av. Packet Delay (ms)= 2.89e+00 +-5.78e-02

```

Figura 2.8: Resultados numéricos da alínea D.

A partir dos Gráficos 2.9 e 2.10, é possível observar que quando ambos os serviços (dados e VoIP) são estatisticamente multiplexados numa única fila FIFO de tamanho pequeno, o número de fluxo de pacotes VoIP influencia o tempo médio de espera quer dos pacotes de dados quer dos pacotes VoIP. Quanto maior for o número de fluxo de pacotes VoIP, existirá mais pacotes à espera (na fila de espera) para serem transmitidos, ou seja, a fila fica cada vez mais cheia com o aumento do fluxo de pacotes VoIP, fazendo com que o tempo de espera de cada pacote aumente. É de notar que agora o tempo médio de espera, quer dos pacotes VoIP, quer de dados, é mais pequeno que na alínea 2.a, pois como a fila de espera é mais pequena, esta terá sempre menos pacotes para atender que na fila da alínea 2.a, logo cada pacote que entrar na fila terá que esperar menos do que se a fila fosse grande.

A partir dos Gráficos 2.11 e 2.12, é possível ver o que acontece quando a fila fica cheia. Como referido anteriormente, a fila de espera é pequena por isso enche rapidamente. Quando esta estiver cheia e chegar mais um pacote, este é descartado, ou seja, torna-se um pacote perdido.

Com o aumento do número de fluxo de pacotes VoIP, existirá mais pacotes a entrar no sistema o que faz com que após a fila de espera encher o resto dos pacotes são descartados. Existindo mais pacotes, existem mais pacotes perdidos porque a fila de espera não aumentou de tamanho.

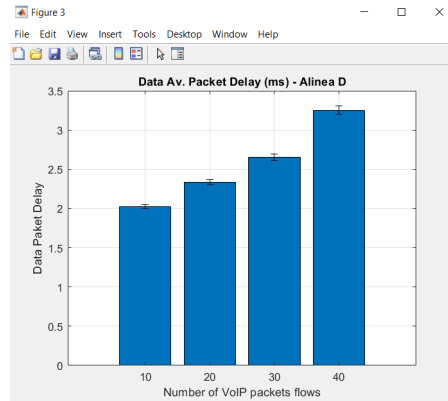


Figura 2.9: Tempo médio de espera de pacotes do tipo Data.

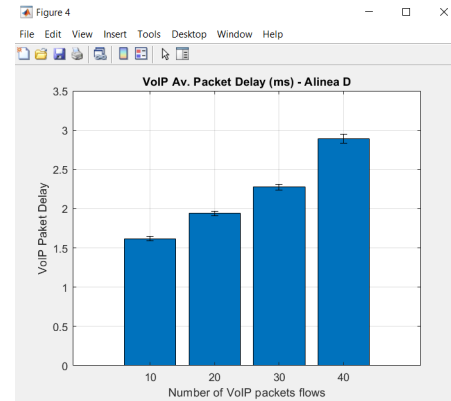


Figura 2.10: Tempo médio de espera de pacotes do tipo VoIP.

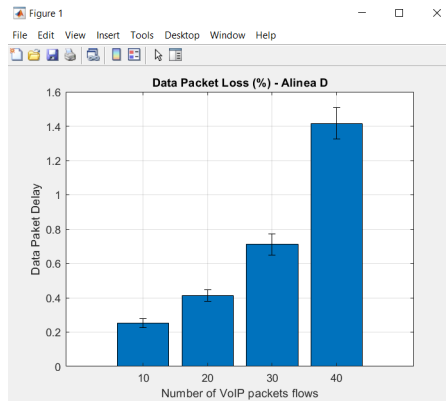


Figura 2.11: Percentagem de pacotes do tipo Data perdidos.

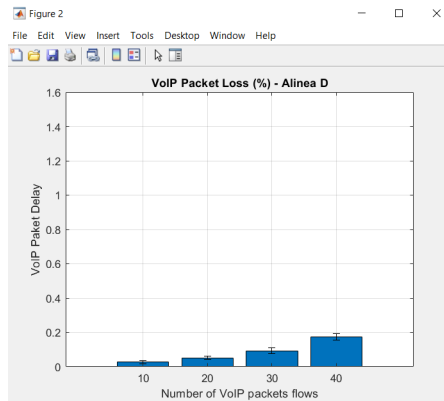


Figura 2.12: Percentagem de pacotes do tipo VoIP perdidos.

Concluindo, e comparando todos gráficos, apesar de existir diferença significativa nos atrasos de ambos os pacotes para os valores obtidos, os valores aumentam com o aumento do fluxo de pacotes. O mesmo se verifica para a percentagem de pacotes perdidos. Se o objetivo for minimizar o tempo de atraso e a percentagem de pacotes perdidos de ambos os pacotes, como ambos aumentam com o fluxo dos pacotes VoIP, o mais indicado seria o menor valor possível para o fluxo de pacotes em questão, pois ambos crescem com o aumento do fluxo.

2.e Task 2.e - Enunciado

Repeat experiment 2.d but now with Simulator4. Justify these results and the differences between them and the results of 2.d. Take conclusions concerning the impact of the number of VoIP flows in the obtained average packet delay and packet loss of each service when VoIP service is supported with a priority which is higher than the data service and the queue is of small size.

2.e.1 Código

Foi pedido que se corresse 50 iterações do simulador 4 para quatro números diferentes de fluxos de pacotes VoIP ($n = 10, 20, 30$ e 40), sendo que $\lambda = 1500$ pps, $C = 10$ Mbps, $f = 10.000$ Bytes, $P = 10000$ e com intervalos de confiança de 90% de desempenho de atraso médio dos pacotes de dados.

Para tal, criou-se uma matriz \mathbf{n} com uma linha e quatro colunas. Assim, para cada valor desta matriz correu-se a simulação cinquenta vezes, guardando os valores de cada simulação. Calculou-se o tempo médio de espera dos pacotes de dados e do tipo VoIP e os seus intervalos de erro associado a esse tempo médio.

A partir daqui, apresentou-se o tempo de atraso médio e a perda de pacotes dos pacotes de dados e do tipo VoIP em gráficos de barras com os intervalos de confiança nas barras de erro.

O código gerado para a resolução do exercício é o seguinte:

```
1 fprintf('Task 2 - Alinea E\n');
2 P = 10000;
3 lambda = 1500;
4 C = 10;
5 f = 10000;
6 n = [10,20,30,40];
7 for index=1:numel(n)
8     N = 50;
9     alfa=0.1; % 90% confidence intervals
10    PL_lst_data = zeros(1,N);
11    APD_lst_data = zeros(1,N);
12    MPD_lst_data = zeros(1,N);
13    PL_lst_VoIP = zeros(1,N);
14    APD_lst_VoIP = zeros(1,N);
15    MPD_lst_VoIP = zeros(1,N);
16    TT_lst = zeros(1,N);
17    for i = 1:N
18        [PL_lst_data(i),PL_lst_VoIP(i),APD_lst_data(i),
            APD_lst_VoIP(i),MPD_lst_data(i),MPD_lst_VoIP(i),
            TT_lst(i)] = Simulator4(lambda,C,f,P,n(index
```

```

    ));
19     end
20     fprintf('Valor %d de n for VoIP:\n',n(index));
21     % Calculate Data Packet Loss
22     PL_data = mean(PL_lst_data);
23     PL_conf_data = norminv(1-alfa/2)*sqrt(var(PL_lst_data
24         )/N);
25     fprintf('PacketLoss_Data (%%) = %.2e +- %.2e\n',
26         PL_data,PL_conf_data)
27     % Calculate VoIP Packet Loss
28     PL_VoIP = mean(PL_lst_VoIP);
29     PL_conf_VoIP = norminv(1-alfa/2)*sqrt(var(PL_lst_VoIP
30         )/N);
31     fprintf('PacketLoss_VoIP (%%) = %.2e +- %.2e\n',
32         PL_VoIP,PL_conf_VoIP)
33     % Calculate Average Data Packet Delay
34     APD_data = mean(APD_lst_data);
35     APD_conf_data = norminv(1-alfa/2)*sqrt(var(
36         APD_lst_data)/N);
37     fprintf('Data Av. Packet Delay (ms)= %.2e +-%.2e\n',
38         APD_data,APD_conf_data);
39     % Calculate Average VoIP Packet Delay
40     APD_VoIP = mean(APD_lst_VoIP);
41     APD_conf_VoIP = norminv(1-alfa/2)*sqrt(var(
42         APD_lst_VoIP)/N);
43     fprintf('VoIP Av. Packet Delay (ms)= %.2e +-%.2e\n',
44         APD_VoIP,APD_conf_VoIP);
45     PL_Results_data(index) = PL_data;
46     PL_Erro_data(index) = PL_conf_data;
47     PL_Results_VoIP(index) = PL_VoIP;
48     PL_Erro_VoIP(index) = PL_conf_VoIP;
49     APD_Results_data(index) = APD_data;
50     APD_Erro_data(index) = APD_conf_data;
51     APD_Results_VoIP(index) = APD_VoIP;
52     APD_Erro_VoIP(index) = APD_conf_VoIP;
53     end
54     % figure for Calculate Data Packet Loss
55     figure(1);
56     bar(n,PL_Results_data);
57     title("Data Packet Loss (%) - Alinea E");
58     grid on
59     xlabel("Number of VoIP packets flows");
60     ylim([0 1.7])
61     ylabel("Data Paket Delay");
62     hold on
63     er1 = errorbar(n,PL_Results_data,PL_Erro_data,

```

```

        PL_Erro_data);
56 er1.Color = [0 0 0];
57 er1.LineStyle = 'none';
58 hold off
59 % figure for Calculate VoIP Packet Loss
60 figure(2);
61 bar(n,PL_Results_VoIP);
62 title("VoIP Packet Loss (%) – Alinea E");
63 grid on
64 xlabel("Number of VoIP packets flows");
65 ylim([0 1.7])
66 ylabel("VoIP Paket Delay");
67 hold on
68 er1 = errorbar(n,PL_Results_VoIP,PL_Erro_VoIP,
        PL_Erro_VoIP);
69 er1.Color = [0 0 0];
70 er1.LineStyle = 'none';
71 hold off
72 % figure for Average Data Packet Delay
73 figure(3);
74 bar(n,APD_Results_data);
75 title("Data Av. Packet Delay (ms) – Alinea E");
76 grid on
77 xlabel("Number of VoIP packets flows");
78 ylim([0 4.3])
79 ylabel("Data Paket Delay");
80 hold on
81 er1 = errorbar(n,APD_Results_data,APD_Erro_data,
        APD_Erro_data);
82 er1.Color = [0 0 0];
83 er1.LineStyle = 'none';
84 hold off
85 % figure for Average VoIP Packet Delay
86 figure(4);
87 bar(n,APD_Results_VoIP);
88 title("VoIP Av. Packet Delay (ms) – Alinea E");
89 grid on
90 xlabel("Number of VoIP packets flows");
91 ylim([0 4.3])
92 ylabel("VoIP Paket Delay");
93 hold on
94 er2 = errorbar(n,APD_Results_VoIP,APD_Erro_VoIP,
        APD_Erro_VoIP);
95 er2.Color = [0 0 0];
96 er2.LineStyle = 'none';
97 hold off

```

2.e.2 Resultados e Conclusões

Neste exercício, foi usada a simulação 4. Como dito anteriormente, nesta simulação os pacotes do tipo VoIP na fila de espera têm prioridade em relação aos pacotes de dados, por isso é de esperar que o tempo médio de espera dos pacotes do tipo VoIP diminua e o tempo médio de espera dos pacotes de dados aumente, quando comparado com a alínea anterior.

Com o aumento do número de fluxos de pacotes VoIP, o tempo médio de espera quer dos pacotes de dados, quer dos pacotes VoIP, aumenta. A fila de espera vai enchendo e quanto mais cheia estiver, mais tempo os pacotes vão esperar para serem atendidos. O aumento do número de fluxos de pacotes VoIP não influencia tanto o tempo médio de espera dos pacotes VoIP como influenciará o tempo médio de espera dos pacotes de dados, devido ao facto dos pacotes VoIP serem prioritários em relação aos de dados.

Para além disto, como a fila de espera é mais pequena ($f = 10.000$ Bytes), também é esperado que com o aumento do número de pacotes esta fila ficará cada vez mais cheia. Atingindo o limite da capacidade da fila fará com que os pacotes que chegarem entretanto sejam descartados, ou seja, são pacotes perdidos.

Como é possível ver na Figura 2.13, ao usar o simulador 4 em vez do 3 observou-se que o tempo médio de espera nos pacotes de dados é sempre maior do que nos pacotes VoIP. Com o aumento do número de fluxos de pacotes VoIP, o tempo médio de espera dos pacotes de dados é muito maior, enquanto que o tempo médio de espera dos pacotes VoIP diminuiu, comparando com a alínea anterior. O tempo médio de espera nos pacotes de dados permanece sempre mais ou menos constante, ou seja, a diferença no tempo médio de espera com o aumento do número de fluxos de pacotes VoIP notou-se mais no tempo médio de espera dos pacotes de dados, como dito anteriormente. É possível observar que o número de pacotes perdidos do tipo VoIP é sempre menor que nos pacotes de dados, pois estes são em menor número.


```

Task 2 - Alínea E
Valor 10 de n for VoIP:
PacketLoss_Data (%) = 2.24e-01 +- 2.62e-02
PacketLoss_VoIP (%) = 2.39e-02 +- 7.96e-03
Data Av. Packet Delay (ms)= 2.07e+00 +-2.82e-02
VoIP Av. Packet Delay (ms)= 4.59e-01 +-2.48e-03
Valor 20 de n for VoIP:
PacketLoss_Data (%) = 4.18e-01 +- 3.08e-02
PacketLoss_VoIP (%) = 5.04e-02 +- 9.21e-03
Data Av. Packet Delay (ms)= 2.54e+00 +-3.58e-02
VoIP Av. Packet Delay (ms)= 4.81e-01 +-2.07e-03
Valor 30 de n for VoIP:
PacketLoss_Data (%) = 8.24e-01 +- 6.30e-02
PacketLoss_VoIP (%) = 1.05e-01 +- 1.62e-02
Data Av. Packet Delay (ms)= 3.10e+00 +-5.15e-02
VoIP Av. Packet Delay (ms)= 5.01e-01 +-2.31e-03
Valor 40 de n for VoIP:
PacketLoss_Data (%) = 1.46e+00 +- 9.62e-02
PacketLoss_VoIP (%) = 1.82e-01 +- 2.15e-02
Data Av. Packet Delay (ms)= 3.92e+00 +-7.19e-02
VoIP Av. Packet Delay (ms)= 5.22e-01 +-2.44e-03

```

Figura 2.13: Resultados numéricos da alínea E.

A partir dos Gráficos 2.14 e 2.15, é possível observar que, quando ambos os serviços (dados e VoIP) são estatisticamente multiplexados numa única fila FIFO de tamanho pequeno e onde existe prioridade dos pacotes VoIP em relação aos pacotes de dados, o número de fluxo de pacotes VoIP influencia o tempo médio de espera, quer dos pacotes de dados, quer dos pacotes VoIP. Quanto maior for o número de fluxo de pacotes VoIP, existirá mais pacotes na fila de espera para serem transmitidos, ou seja, a fila fica cada vez mais cheia com o aumento do fluxo de pacotes VoIP, fazendo com que o tempo de espera de cada pacote aumente. Este aumento não é igual para os dois tipos de pacotes, influenciando mais o tempo médio de espera nos pacotes de dados do que nos pacotes VoIP (praticamente constante).

Isto acontece porque os pacotes VoIP têm mais prioridade que os pacotes de dados, ou seja, os pacotes VoIP são sempre atendidos primeiros do que os pacotes de dados (quando chegam são colocados à frente de todos os pacotes dados que houver na fila), por isso, um pacote VoIP da fila só tem que esperar que os pacotes VoIP à sua frente sejam transmitidos. Por outro lado, os pacotes de dados têm que esperar que não exista nenhum pacote do tipo VoIP na fila de espera para ser transmitido.

A partir dos Gráficos 2.16 e 2.17, é possível ver o que acontece quando a fila fica cheia. Como referido anteriormente, a fila de espera é pequena por isso enche rapidamente. Quando esta estiver cheia e chegar mais um pacote, este é descartado, ou seja, torna-se um pacote perdido.

Com o aumento do número de fluxo de pacotes VoIP, existirá mais pacotes a entrar no sistema o que faz com que após a fila de espera encher o resto dos pacotes são descartados. Existindo mais pacotes, existem mais pacotes perdidos.

O facto dos pacotes VoIP terem mais prioridade que os pacotes de dados não influenciou a perda de pacotes, porque a fila de espera não aumentou de

tamanho. O que mudou foi a prioridade de transmissão da fila.

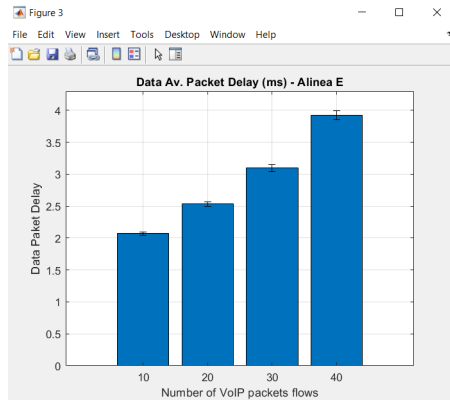


Figura 2.14: Tempo médio de espera de pacotes do tipo Data.

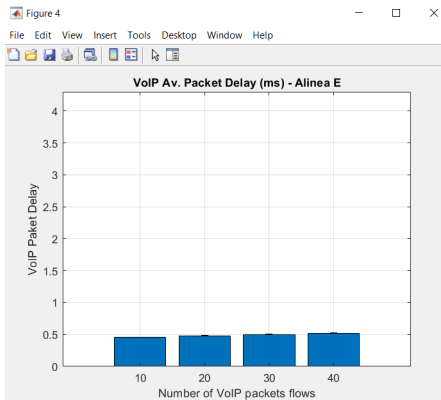


Figura 2.15: Tempo médio de espera de pacotes do tipo VoIP.

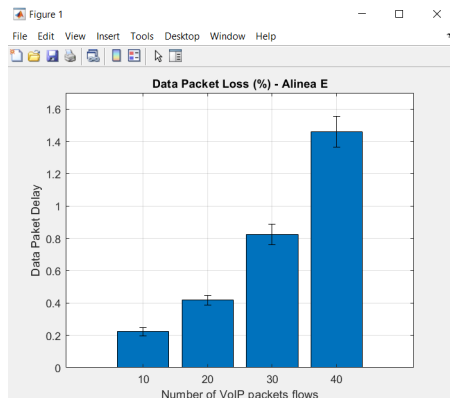


Figura 2.16: Percentagem de pacotes do tipo Data perdidos.

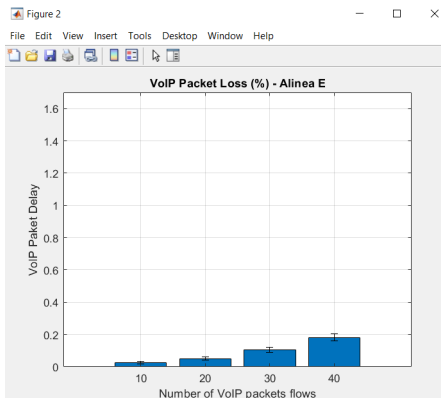


Figura 2.17: Percentagem de pacotes do tipo VoIP perdidos.

Concluindo, e comparando ambos os gráficos, apesar de existir diferença significativa nos tempos médios de atrasos e nas percentagens de perdas de pacotes de ambos para os valores obtidos, os valores aumentam com o aumento do fluxo de pacotes. Se o objetivo for minimizar o tempo médio de atraso e a percentagem de perdas de ambos os pacotes, como ambos aumentam com o aumento do fluxo dos pacotes VoIP, apesar de ser dito no enunciado que os pacotes VoIP têm prioridade sobre os pacotes de dados, o mais indicado seria o menor valor possível para o fluxo de pacotes em questão, pois ambos crescem com o aumento do fluxo.

2.f Task 2.f - Enunciado

Develop a new version of Simulator4 to consider that VoIP packets are always accepted in the queue (if there is enough space) but data packets are accepted in the queue only if the total queue occupation does not become higher than 90% (a simplified version of WRED – Weighted Random Early Discard). Repeat experiment 2.e but now with the new version of Simulator4. Justify these results and the differences between them and the results of 2.e. Take conclusions concerning the impact of the number of VoIP flows in the obtained average packet delay and packet loss of each service when (i) VoIP service is supported with a priority which is higher than the data service and (ii) the packet acceptance in the queue is differentiated.

2.f.1 Código

Foi pedido para desenvolvermos um novo simulador 4, em que se considera que os pacotes VoIP são sempre aceites na fila (se houver espaço suficiente), mas os pacotes de dados só são aceites na fila se a ocupação total da fila não se for superior a 90%. Para isso, como se pode observar na Figura 2.18, alterou-se a linha 86 do simulador. O tamanho da fila depois de se colocar o novo pacote vai ser a soma entre tamanho da fila já ocupado e o tamanho do pacote que acabou de chegar. Isto não pode ser superior a 90% da ocupação total da fila ($f \cdot 0.9$).

```
74 case ARRIVAL % If first event is an ARRIVAL
75 TOTALPACKETS = TOTALPACKETS +1;
76 if packetType == 0 % Data
77 TOTALPACKETS_data= TOTALPACKETS_data+1;
78 tmp= Clock + exprnd(1/lambda);
79 EventList = [EventList; ARRIVAL, tmp, GeneratePacketSize_Data(), tmp,packetType];
80 if STATE==0
81 STATE= 1;
82 EventList = [EventList; DEPARTURE, Clock + 8*PacketSize/(C*10^6), PacketSize, Clock,packetType];
83 else
84 % Data packets are accepted in the queue only if the total
85 % queue occupation does not become higher than 90%
86 if QUEUEOCCUPATION + PacketSize <= (f*0.9)
87 QUEUE= [QUEUE;PacketSize , Clock,packetType];
88 QUEUEOCCUPATION= QUEUEOCCUPATION + PacketSize;
89 else
90 LOSTPACKETS_data= LOSTPACKETS_data + 1;
91 end
92 end
93 else %VoIP
94 TOTALPACKETS_VoIP= TOTALPACKETS_VoIP+1;
```

Figura 2.18: Nova versão do simulador 4.

O código gerado para a resolução do exercício é o seguinte:

```
1 fprintf('Task 2 – Alinea F\n');
```

```

2 P = 10000;
3 lambda = 1500;
4 C = 10;
5 f = 10000;
6 n = [10,20,30,40];
7 for index=1:numel(n)
8     N = 50;
9     alfa=0.1; % 90% confidence intervals
10    PL_lst_data = zeros(1,N);
11    APD_lst_data = zeros(1,N);
12    MPD_lst_data = zeros(1,N);
13    PL_lst_VoIP = zeros(1,N);
14    APD_lst_VoIP = zeros(1,N);
15    MPD_lst_VoIP = zeros(1,N);
16    TT_lst = zeros(1,N);
17    for i = 1:N
18        [PL_lst_data(i),PL_lst_VoIP(i),APD_lst_data(i),
19         APD_lst_VoIP(i),MPD_lst_data(i),MPD_lst_VoIP(i)
20         ),TT_lst(i)] = Simulator4_f(lambda,C,f,P,n(
21         index));
22    end
23    fprintf('Valor %d de n for VoIP:\n',n(index));
24    % Calculate Data Packet Loss
25    PL_data = mean(PL_lst_data);
26    PL_conf_data = norminv(1-alfa/2)*sqrt(var(PL_lst_data
27    )/N);
28    fprintf('PacketLoss_Data (%%) = %.2e +- %.2e\n',
29    PL_data,PL_conf_data);
30    % Calculate VoIP Packet Loss
31    PL_VoIP = mean(PL_lst_VoIP);
32    PL_conf_VoIP = norminv(1-alfa/2)*sqrt(var(PL_lst_VoIP
33    )/N);
34    fprintf('PacketLoss_VoIP (%%) = %.2e +- %.2e\n',
35    PL_VoIP,PL_conf_VoIP);
36    % Calculate Average Data Packet Delay
37    APD_data = mean(APD_lst_data);
38    APD_conf_data = norminv(1-alfa/2)*sqrt(var(
39    APD_lst_data)/N);
40    fprintf('Data Av. Packet Delay (ms)= %.2e +-%.2e\n',
41    APD_data,APD_conf_data);
42    % Calculate Average VoIP Packet Delay
43    APD_VoIP = mean(APD_lst_VoIP);
44    APD_conf_VoIP = norminv(1-alfa/2)*sqrt(var(
45    APD_lst_VoIP)/N);
46    fprintf('VoIP Av. Packet Delay (ms)= %.2e +-%.2e\n',
47    APD_VoIP,APD_conf_VoIP);

```

```

37     PL_Results_data(index) = PL_data;
38     PL_Erro_data(index) = PL_conf_data;
39     PL_Results_VoIP(index) = PL_VoIP;
40     PL_Erro_VoIP(index) = PL_conf_VoIP;
41     APD_Results_data(index) = APD_data;
42     APD_Erro_data(index) = APD_conf_data;
43     APD_Results_VoIP(index) = APD_VoIP;
44     APD_Erro_VoIP(index) = APD_conf_VoIP;
45 end
46 % figure for Calculate Data Packet Loss
47 figure(1);
48 bar(n,PL_Results_data);
49 title("Data Packet Loss (%) – Alinea F");
50 grid on
51 xlabel("Number of VoIP packets flows");
52 ylim([0 2])
53 ylabel("Data Paket Delay");
54 hold on
55 er1 = errorbar(n,PL_Results_data,PL_Erro_data,
56     PL_Erro_data);
57 er1.Color = [0 0 0];
58 er1.LineStyle = 'none';
59 hold off
60 % figure for Calculate VoIP Packet Loss
61 figure(2);
62 bar(n,PL_Results_VoIP);
63 title("VoIP Packet Loss (%) – Alinea F");
64 grid on
65 xlabel("Number of VoIP packets flows");
66 ylim([0 2])
67 ylabel("VoIP Paket Delay");
68 hold on
69 er1 = errorbar(n,PL_Results_VoIP,PL_Erro_VoIP,
70     PL_Erro_VoIP);
71 er1.Color = [0 0 0];
72 er1.LineStyle = 'none';
73 hold off
74 % figure for Average Data Packet Delay
75 figure(3);
76 bar(n,APD_Results_data);
77 title("Data Av. Packet Delay (ms) – Alinea F");
78 grid on
79 xlabel("Number of VoIP packets flows");
80 ylim([0 4])
81 ylabel("Data Paket Delay");
82 hold on

```

```

81 er1 = errorbar(n,APD_Results_data,APD_Erro_data,
      APD_Erro_data);
82 er1.Color = [0 0 0];
83 er1.LineStyle = 'none';
84 hold off
85 % figure for Average VoIP Packet Delay
86 figure(4);
87 bar(n,APD_Results_VoIP);
88 title("VoIP Av. Packet Delay (ms) - Alinea F");
89 grid on
90 xlabel("Number of VoIP packets flows");
91 ylim([0 4])
92 ylabel("VoIP Paket Delay");
93 hold on
94 er2 = errorbar(n,APD_Results_VoIP,APD_Erro_VoIP,
      APD_Erro_VoIP);
95 er2.Color = [0 0 0];
96 er2.LineStyle = 'none';
97 hold off

```

2.f.2 Resultados e Conclusões

Neste exercício, foi usada uma nova versão da simulação 4 onde os pacotes do tipo VoIP na fila de espera têm prioridade em relação aos pacotes de dados e para além disso considera-se que os pacotes VoIP são sempre aceites na fila (se houver espaço suficiente) mas os pacotes de dados só são aceites na fila se a ocupação total da fila não se for superior a 90%.

Assim é de esperar que o tempo médio de espera dos pacotes de dados e do tipo VoIP não se altere quando comparado com a alínea anterior, pois o que se alterou no simulador novo foi uma condição para a entrada na fila de espera e não uma condição para pacotes que estão dentro desta. Onde se vai verificar uma mudança será no número de perda de pacotes, pois quando a fila de espera estiver 90% ocupada os pacotes data são descartados, entrando só pacotes VoIP para a fila de espera.

Como é possível ver através da Figura 2.19 ou através dos Gráficos 2.20 e 2.21, o tempo médio dos pacotes de dados e do VoIP não se alterou quando comparado com a alínea anterior. Mas nos Gráficos 2.22 e 2.23, é possível observar que a perda de pacotes VoIP é nula e a perda de pacotes de dados aumentou relativamente a alínea anterior. Isto ocorreu porque se alterou no simulador novo uma condição para a entrada na fila de espera e não uma condição para pacotes que estão dentro desta. Isto faz com que quando a fila de espera estiver 90% ocupada os pacotes data são descartados, entrando só pacotes VoIP para a fila de espera.

```

Task 2 - Alinea F
Valor 10 de n for VoIP:
PacketLoss_Data (%) = 3.90e-01 +- 3.40e-02
PacketLoss_VoIP (%) = 0.00e+00 +- 0.00e+00
Data Av. Packet Delay (ms)= 2.04e+00 +-2.40e-02
VoIP Av. Packet Delay (ms)= 4.57e-01 +-2.02e-03
Valor 20 de n for VoIP:
PacketLoss_Data (%) = 6.20e-01 +- 5.15e-02
PacketLoss_VoIP (%) = 0.00e+00 +- 0.00e+00
Data Av. Packet Delay (ms)= 2.42e+00 +-4.01e-02
VoIP Av. Packet Delay (ms)= 4.78e-01 +-2.58e-03
Valor 30 de n for VoIP:
PacketLoss_Data (%) = 9.39e-01 +- 7.34e-02
PacketLoss_VoIP (%) = 0.00e+00 +- 0.00e+00
Data Av. Packet Delay (ms)= 2.84e+00 +-4.02e-02
VoIP Av. Packet Delay (ms)= 4.96e-01 +-2.35e-03
Valor 40 de n for VoIP:
PacketLoss_Data (%) = 1.69e+00 +- 7.09e-02
PacketLoss_VoIP (%) = 0.00e+00 +- 0.00e+00
Data Av. Packet Delay (ms)= 3.58e+00 +-5.17e-02
VoIP Av. Packet Delay (ms)= 5.20e-01 +-1.79e-03

```

Figura 2.19: Resultados numéricos da alínea F.

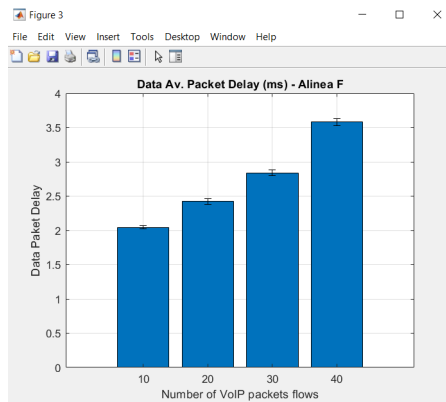


Figura 2.20: Tempo médio de espera de pacotes do tipo Data.

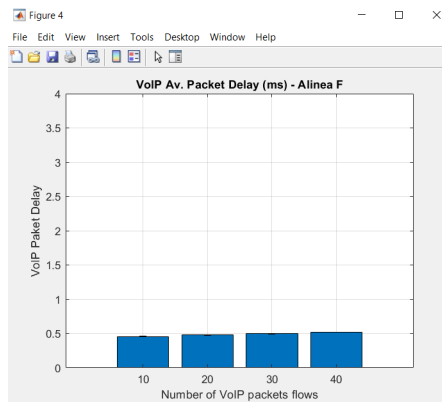


Figura 2.21: Tempo médio de espera de pacotes do tipo VoIP.

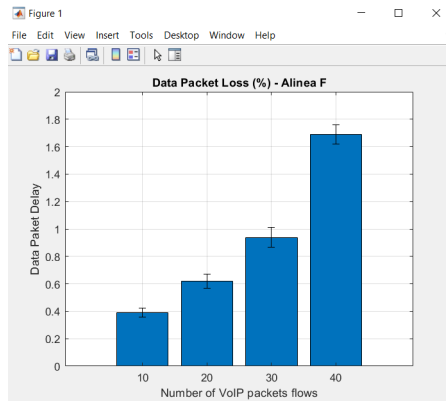


Figura 2.22: Percentagem de pacotes do tipo Data perdidos.

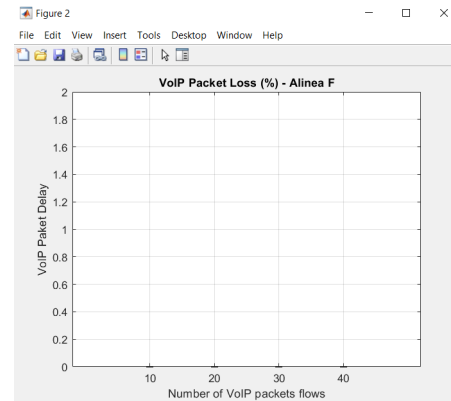


Figura 2.23: Percentagem de pacotes do tipo VoIP perdidos.

Concluindo, e comparando ambos os gráficos, apesar de existir uma diferença significativa no tempo médio de atraso e na percentagem de perda de ambos os pacotes para os valores obtidos, os valores aumentam com o aumento do fluxo de pacotes, excepto no caso da percentagem de pacotes VoIP perdidos (sempre nulo), pois estes são prioritários. Se o objetivo for minimizar o tempo médio de atraso de ambos os pacotes, como ambos aumentam com o aumento do fluxo dos pacotes VoIP, como é dito no enunciado que os pacotes VoIP têm prioridade sobre os pacotes de dados, o mais indicado seria o menor valor possível para o fluxo de pacotes em questão, pois ambos crescem com o aumento do fluxo.

Capítulo 3

Contribuições dos autores

Mariana Pinto - 50%
Raquel Pinto - 50%