



## **Mini-Project #1**

### Performance Evaluation of Point-to-Point Links Supporting Packet Services

Martim Neves, 88904

João Simões, 88930

7 de novembro de 2022

Departamento de Eletrónica, Telecomunicações e Informática

**Modelação e Desempenho de Redes e Serviços**

Professor: Amaro Fernandes de Sousa

# Contents

<b>Task 1</b>	<b>3</b>
Task 1.a. . . . .	3
Task 1.b. . . . .	6
Task 1.c. . . . .	9
<b>Task 2</b>	<b>11</b>
Simulator3A . . . . .	11
Task 2.a. . . . .	13
Task 2.b. . . . .	17
Task 2.c. . . . .	21
<b>Task 3</b>	<b>23</b>
Task 3.a. . . . .	23
Task 3.b. . . . .	28
Simulator4c . . . . .	32
Task 3.c. . . . .	33
Simulator4d . . . . .	37
Task 3.d. . . . .	39

# Task 1

## Task 1.a.

### Código MATLAB

```
1 lambda = [1500, 1600, 1700, 1800, 1900];
2 C=10;
3 f=1000000;
4 P=100000;
5 b=10^-6;
6 N=20;
7 alfa=0.1;
8 PL = zeros(1,N);
9 APD = zeros(1,N);
10 MPD = zeros(1,N);
11 TT= zeros(1,N);
12
13 for j=1:length(lambda)
14     for i=1:N
15         [PL(i), APD(i), MPD(i), TT(i)] = Simulator2(
16             lambda(j),C,f,P,b);
17     end
18     avgPL(j) = sum(PL)/N;
19     termPL(j) = norminv(1-alfa/2)*sqrt(var(PL)/N);
20
21     avgAPD(j) = sum(APD)/N;
22     termAPD(j) = norminv(1-alfa/2)*sqrt(var(APD)/N);
23 end
24
25 figure(1)
26 bar(lambda,avgAPD)
27 hold on
28 erro=errorbar(lambda,avgAPD,termAPD,termAPD);
29 erro.Color=[0 0 0];
```

```

30 erro.LineStyle = 'none';
31 hold off
32 grid on
33 xlabel('lambda values')
34 ylabel('milliseconds')
35 title('Average Data Packet Delay (milliseconds) ')
36
37 figure(2)
38 bar(lambda, avgPL)
39 hold on
40 erro=errorbar(lambda, avgPL, termPL, termPL);
41 erro.Color=[0 0 0];
42 erro.LineStyle = 'none';
43 hold off
44 grid on
45 xlabel('lambda values')
46 ylabel('%')
47 title('Average Data Packet Loss (%) ')

```

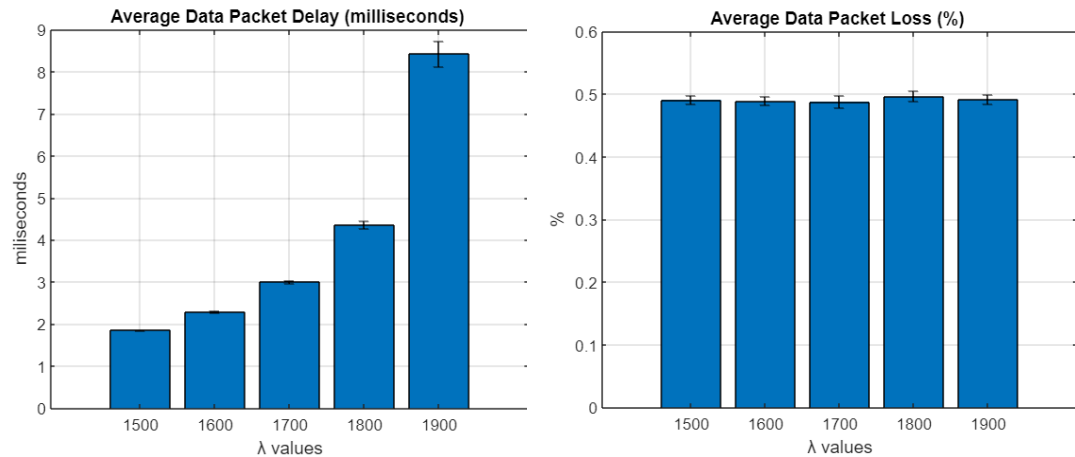
## Análise do Código

Inicialmente são definidos os valores para os parâmetros de entrada do simulador, valores estes dados no enunciado. Em seguida são criadas matrizes de zeros com dimensão igual ao número de corridas, de modo a guardar os valores provenientes do simulador. Este processo é repetido em todas as alíneas, pelo que apenas será explicado nesta alínea.

De seguida são feitas dois ciclos for, um para percorrer todos os valores de lambda e outro para realizar as corridas pedidas, e é invocado o Simulator2 com os respetivos parâmetros de entrada, sendo os valores de saída guardados nas respetivas matrizes.

Por fim, são usados os resultados guardados nas matrizes para desenhar os gráficos de barras e as respetivas barras de erro.

## Resultados



## Análises e Justificações

Uma vez que os vários valores de lambda ( $\lambda$ ) representam a taxa de pacotes de dados enviados (em pacotes por segundo), pressupunha-se que à medida que este valor aumenta-se, também o atraso médio de chegada desses pacotes aumentasse. O primeiro gráfico apresentado em cima comprova esta relação de proporcionalidade direta.

Quanto às percentagens da perda média de pacotes, podemos concluir que não variam significativamente com o aumento da taxa de pacotes enviados, permanecendo aproximadamente em 0.5%.

## Task 1.b.

### Código MATLAB

```
1 lambda = [1500, 1600, 1700, 1800, 1900];
2 C=10;
3 f=1000000;
4 P=100000;
5 b=10^-4;      % Changed from 10^-6 to 10^-4
6 N=20;
7 alfa=0.1;
8 PL = zeros(1,N);
9 APD = zeros(1,N);
10 MPD = zeros(1,N);
11 TT= zeros(1,N);
12
13 for j=1:length(lambda)
14     for i=1:N
15         [PL(i), APD(i), MPD(i), TT(i)] = Simulator2(
16             lambda(j),C,f,P,b);
17     end
18     avgPL(j) = sum(PL)/N;
19     termPL(j) = norminv(1-alfa/2)*sqrt(var(PL)/N);
20
21     avgAPD(j) = sum(APD)/N;
22     termAPD(j) = norminv(1-alfa/2)*sqrt(var(APD)/N);
23 end
24
25 figure(3)
26 bar(lambda,avgAPD)
27 hold on
28 erro=errorbar(lambda,avgAPD,termAPD,termAPD);
29 erro.Color=[0 0 0];
30 erro.LineStyle = 'none';
31 hold off
32 grid on
33 xlabel('lambda values')
34 ylabel('milliseconds')
35 title('Average Data Packet Delay (milliseconds) ')
36
37 figure(4)
38 bar(lambda,avgPL)
39 hold on
40 erro=errorbar(lambda,avgPL,termPL,termPL);
41 erro.Color=[0 0 0];
```

```

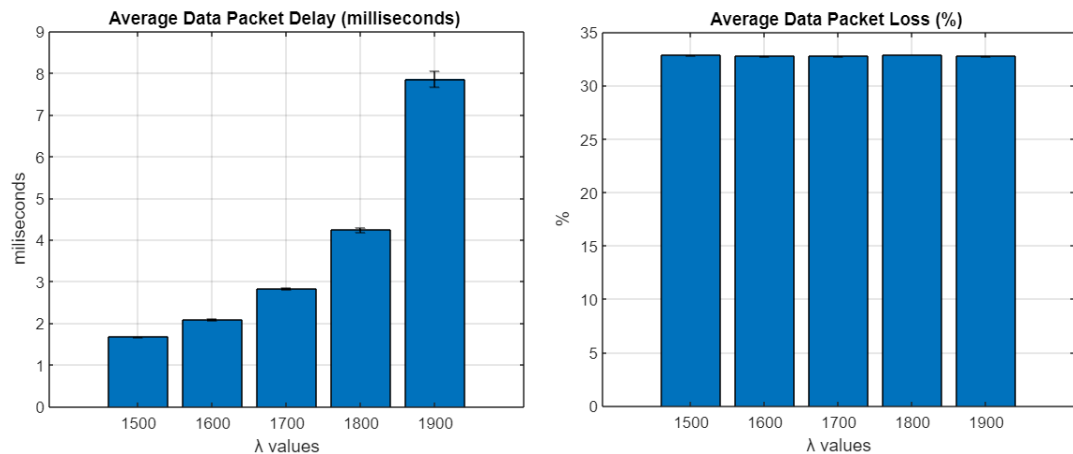
42 erro.LineStyle = 'none';
43 hold off
44 grid on
45 xlabel('lambda values')
46 ylabel('%')
47 title('Average Data Packet Loss (%) ')

```

## Análise do Código

O código para a resolução desta alínea é igual ao da alínea anterior, com a exceção de que desta vez o valor do Bit Error Rate (BER) é  $10^{-4}$ .

## Resultados



## Análises e Justificações

De forma semelhante ao exercício anterior, à medida que o valor da taxa de pacotes de dados enviados aumenta, também o atraso de chegada desses pacotes aumenta, como se pode comprovar pelo primeiro gráfico apresentado acima. Como o único parâmetro que se alterou foi o BER, então os valores do atraso médio da chegada dos pacotes não sofreram grandes alterações do exercício anterior para este.

Quanto às percentagens da perda média de pacotes, podemos concluir pela observação do segundo gráfico que, tal como no exercício anterior, também não variam significativamente com o aumento da taxa de pacotes enviados, mas desta vez permanecendo num valor de, aproximadamente, 33%. Este abrupto aumento na percentagem, comparativamente aos 0.5% do exercício anterior, deve-se à alteração do valor do BER de  $10^{-6}$  para  $10^{-4}$ . Isto porque, desta forma, a taxa de erro aumentou 100 vezes, desencadeando uma maior probabilidade de acontecerem erros nos pacotes de dados durante a simulação.



## Task 1.c.

### Código MATLAB

```
1 aux1=[65:109 111:1517];
2 aux3=length(aux1);
3 aux4=(1-(0.19+0.23+0.17))/aux3;
4 ber1=10^-6;
5 ber2=10^-4;
6 pl1=zeros(1,aux3+3);
7 pl2=zeros(1,aux3+3);
8 for i = 1:length(pl1)
9     if i == 1
10         pl1(i) = 0.19*(100*(1-(nchoosek((i+64-1)*8,0)*
11             ber1^(0)*(1-ber1)^((i+64-1)*8-0))));
12         pl2(i) = 0.19*(100*(1-(nchoosek((i+64-1)*8,0)*
13             ber2^(0)*(1-ber2)^((i+64-1)*8-0))));
14     elseif i == 110-64+1
15         pl1(i) = 0.23*(100*(1-(nchoosek((i+64-1)*8,0)*
16             ber1^(0)*(1-ber1)^((i+64-1)*8-0))));
17         pl2(i) = 0.23*(100*(1-(nchoosek((i+64-1)*8,0)*
18             ber2^(0)*(1-ber2)^((i+64-1)*8-0))));
19     elseif i == 1518-64+1
20         pl1(i) = 0.17*(100*(1-(nchoosek((i+64-1)*8,0)*
21             ber1^(0)*(1-ber1)^((i+64-1)*8-0))));
22         pl2(i) = 0.17*(100*(1-(nchoosek((i+64-1)*8,0)*
23             ber2^(0)*(1-ber2)^((i+64-1)*8-0))));
24     else
25         pl1(i) = aux4*(100*(1-(nchoosek((i+64-1)*8,0)*
26             ber1^(0)*(1-ber1)^((i+64-1)*8-0))));
27         pl2(i) = aux4*(100*(1-(nchoosek((i+64-1)*8,0)*
28             ber2^(0)*(1-ber2)^((i+64-1)*8-0))));
29     end
30 end
31 apdpl1 = sum(pl1);
32 apdpl2 = sum(pl2);
33 fprintf(" Theoretical average packet loss with BER = 10^-6
34     -> %.3f %%\n", ap11);
35 fprintf(" Theoretical average packet loss with BER = 10^-4
36     -> %.3f %%\n", ap12);
```

## Análise do Código

Inicialmente foi um vetor com todos os tamanhos de pacotes equiprováveis e foi calculada essa mesma probabilidade. Em seguida foram declarados os BER e criadas duas matrizes com tamanho igual ao número de tamanhos que os pacotes podem ter, de modo a armazenar a perda de pacotes para cada tamanho. O próximo passo foi fazer um ciclo for para percorrer todos os tamanhos e, consoante o tamanho do pacote, calcular a probabilidade de o pacote ter pelo menos um erro devido ao BER (usando a função de probabilidade para variáveis aleatórias binomiais) e multiplicar essa probabilidade pela respetiva probabilidade do tamanho do pacote. Este processo foi feito duas vezes para cada condição de modo a ter em conta ambos os BER. Foi calculada a probabilidade de o pacote ter pelo menos um erro pois é essa a condição para que o pacote seja perdido, ou seja, ao calcular a probabilidade de um pacote de um tamanho específico ter um ou mais erros, estamos a calcular a média da perda de pacotes para um pacote com esse tamanho.

Por fim, foi feita a soma das médias de perdas de pacote para cada tamanho e para cada BER e foram apresentados os resultados.

## Resultados

```
Theoretical average packet loss with BER = 10^-6 -> 0.494 %  
Theoretical average packet loss with BER = 10^-4 -> 32.828 %
```

## Análises e Justificações

Verifica-se assim que, tanto para o Bit Error Rate com o valor  $10^{-6}$  como para com o valor  $10^{-4}$ , os valores percentuais da perda média de pacotes de dados do sistema, obtidos na simulação prática (exercícios 1.a. e 1.b.), coincidem exatamente com os valores teóricos calculados neste exercício.

# Task 2

## Simulator3A

### Código MATLAB

```
1 case DEPARTURE % If first event is a DEPARTURE
2     if type == 0 % Data Packet
3         if rand() < (1-b)^(PacketSize*8)
4             TRANSMITTEDBYTES= TRANSMITTEDBYTES +
5                 PacketSize;
6             DELAYS= DELAYS + (Clock - ArrivalInstant);
7             if Clock - ArrivalInstant > MAXDELAY
8                 MAXDELAY= Clock - ArrivalInstant;
9             end
10            TRANSMITTEDPACKETS= TRANSMITTEDPACKETS + 1;
11        else
12            LOSTPACKETS = LOSTPACKETS + 1;
13        end
14        if QUEUEOCCUPATION > 0
15            EventList = [EventList; DEPARTURE, Clock + 8*
16                QUEUE(1,1)/(C*10^6), QUEUE(1,1), QUEUE
17                (1,2), QUEUE(1,3)];
18            QUEUEOCCUPATION= QUEUEOCCUPATION - QUEUE(1,1)
19            ;
20            QUEUE(1,:)= [];
21        else
22            STATE= 0;
23        end
24    elseif type == 1 % VoIP Packet
25        if rand() < (1-b)^(PacketSize*8)
26            TRANSMITTEDBYTESV= TRANSMITTEDBYTESV +
27                PacketSize;
28            DELAYSV= DELAYSV + (Clock - ArrivalInstant);
29            if Clock - ArrivalInstant > MAXDELAYV
30                MAXDELAYV= Clock - ArrivalInstant;
```

```

26         end
27         TRANSMITTEDPACKETSV= TRANSMITTEDPACKETSV + 1;
28     else
29         LOSTPACKETSV = LOSTPACKETSV + 1;
30     end

```

## Análise do Código

Para construir o Simulator3A usámos como base o Simulator3 e fizemos pequenas alterações. Dado um Bit Error Rate (BER)  $b$ , é gerado um número aleatório usando a função `rand()` e, caso o evento seja uma Departure, esse número aleatório é comparado com a probabilidade de um pacote ter erros, probabilidade essa calculada usando a função de probabilidade para variáveis aleatórias geométricas. Se o valor aleatório for menor, significa que não houve erros e o pacote foi transmitido com sucesso, sendo incrementada a variável que conta o número de pacotes transmitidos. Caso contrário, significa que houve erros e, como tal, o pacote foi perdido, pelo que é incrementada a variável que conta o número de pacotes perdidos. Um raciocínio idêntico foi aplicado para os pacotes VoIP.

## Task 2.a.

### Código MATLAB

```
1 lambda = 1500;
2 c=10;
3 f=1000000;
4 b=10^-5;
5 P=100000;
6 N=20;
7 n=[10,20,30,40];
8 alfa=0.1;
9 PLdata = zeros(1,N);
10 APDdata = zeros(1,N);
11 PLvoip = zeros(1,N);
12 APDvoip = zeros(1,N);
13 for i=1:length(n)
14
15     for j=1:N
16         [PLdata(j), APDdata(j), MPDdata(j), TT(j), PLvoip
            (j), APDvoip(j), MPDvoip(j)] = Simulator3A(
                lambda,c,f,P,n(i),b);
17     end
18
19     avgPLdata(i) = mean(PLdata);
20     termPLdata(i) = norminv(1-alfa/2)*sqrt(var(PLdata)/N)
        ;
21
22     avgAPDdata(i) = mean(APDdata);
23     termAPDdata(i) = norminv(1-alfa/2)*sqrt(var(APDdata)/
        N);
24
25     avgPLvoip(i) = mean(PLvoip);
26     termPLvoip(i) = norminv(1-alfa/2)*sqrt(var(PLvoip)/N)
        ;
27
28     avgAPDvoip(i) = mean(APDvoip);
29     termAPDvoip(i) = norminv(1-alfa/2)*sqrt(var(APDvoip)/
        N);
30 end
31
32 figure(1)
33 bar(n,avgAPDdata)
34 hold on
35 erro=errorbar(n,avgAPDdata,termAPDdata,termAPDdata);
36 erro.Color=[0 0 0];
```

```

37 erro.LineStyle = 'none';
38 hold off
39 grid on
40 xlabel('n values')
41 ylabel('milliseconds')
42 title('Average Data Packet Delay (milliseconds)')
43
44 figure(2)
45 bar(n, avgPLdata)
46 hold on
47 erro=errorbar(n, avgPLdata, termPLdata, termPLdata);
48 erro.Color=[0 0 0];
49 erro.LineStyle = 'none';
50 hold off
51 grid on
52 xlabel('n values')
53 ylabel('%')
54 title('Average Data Packet Loss (%)')
55
56 figure(3)
57 bar(n, avgAPDvoip)
58 hold on
59 erro=errorbar(n, avgAPDvoip, termAPDvoip, termAPDvoip);
60 erro.Color=[0 0 0];
61 erro.LineStyle = 'none';
62 hold off
63 grid on
64 xlabel('n values')
65 ylabel('milliseconds')
66 title('Average VoIP Packet Delay (milliseconds)')
67
68 figure(4)
69 bar(n, avgPLvoip)
70 hold on
71 erro=errorbar(n, avgPLvoip, termPLvoip, termPLvoip);
72 erro.Color=[0 0 0];
73 erro.LineStyle = 'none';
74 hold off
75 grid on
76 xlabel('n values')
77 ylabel('%')
78 title('Average VoIP Packet Loss (%)')

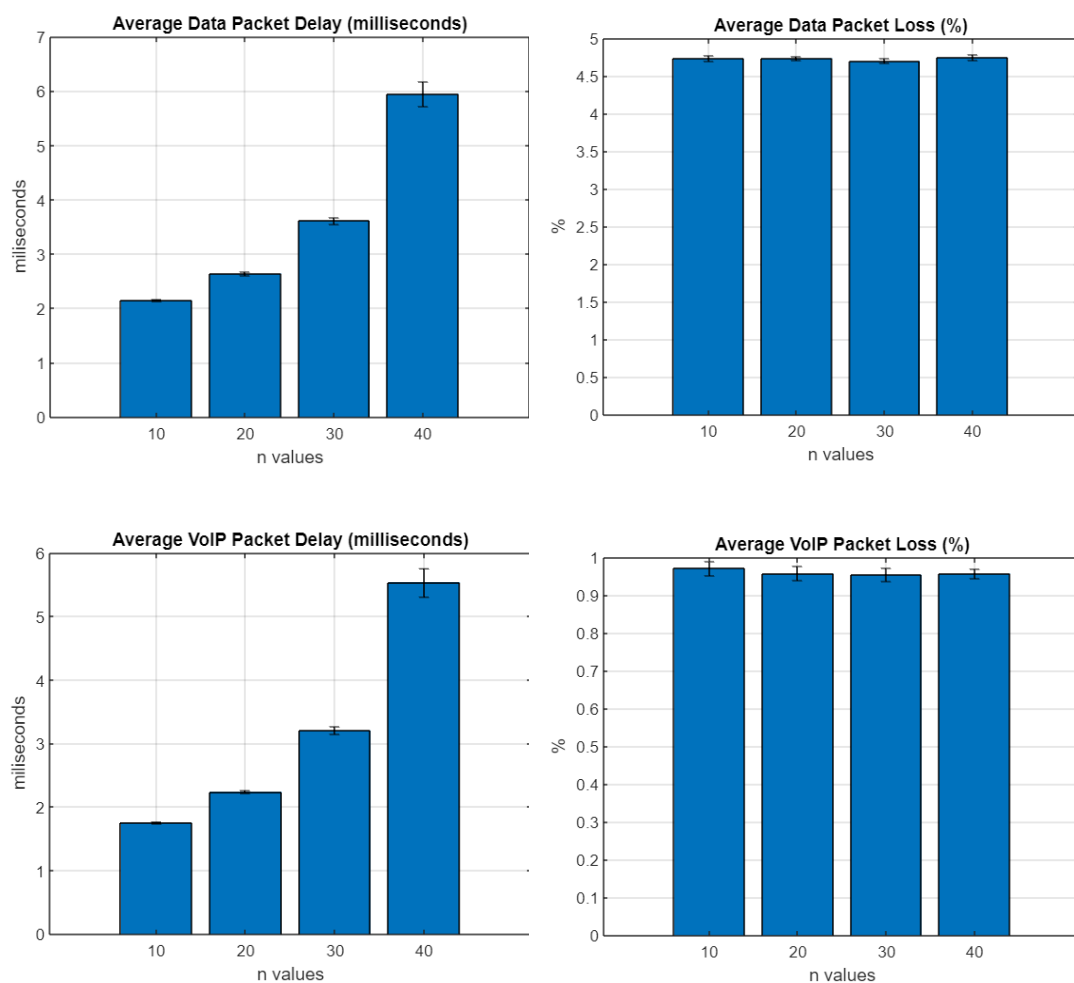
```

## Análise do Código

Inicialmente são feitos dois ciclos for, um para percorrer os vários valores de  $n$  e outro para executar o número de corridas pretendido. Dentro desses ciclos é invocado o Simulator3A e os valores de retorno são armazenados nas matrizes anteriormente criadas. Depois disso são calculadas as métricas de desempenho e o processo é repetido para todos os valores de  $n$ .

Por fim são feitos gráficos de barras, com as respectivas barras de erro, para mostrar o atraso e perda de pacotes médios para cada valor de  $n$ .

## Resultados



## Análises e Justificações

O Simulator3A introduz um Bit Error Rate (BER), o que faz com que haja a possibilidade de alguns pacotes serem recebidos com erros, sendo esses pacotes considerados como perdidos. Assim sendo, e tendo em conta que o tamanho médio dos pacotes de dados (cerca de 620 Bytes) é superior ao tamanho médio dos pacotes VoIP (120 Bytes), a probabilidade de os pacotes de dados, por serem maiores, serem recebidos com erros é superior à probabilidade de os pacotes VoIP serem recebidos com erros, o que resulta numa maior perda média de pacotes de dados, quando comparando com a perda média de pacotes VoIP, tal como podemos ver nos gráficos. No entanto, podemos também ver que o número de fluxos de pacotes VoIP não tem qualquer influência na perda média de pacotes, uma vez os fatores que afetam a perda de pacotes são apenas o BER e o tamanho do pacote.

Quanto ao atraso médio, através da análise dos gráficos é possível ver que aumenta, tanto para pacotes de dados como para pacotes VoIP, conforme aumenta o número de fluxos de pacotes VoIP. Isto deve-se ao facto de que, quanto maior for o número de pacotes no sistema, maior é a probabilidade de a fila de espera estar cheia e o link estar ocupado, o que faz com que os pacotes não sejam atendidos imediatamente, provocando um aumento no atraso médio. Além disso, podemos também ver que, para o mesmo valor de fluxo, o atraso é maior nos pacotes de dados. Esta última observação leva-nos a concluir que, quanto maior for o pacote, maior vai ser o atraso no seu atendimento.



## Task 2.b.

### Código MATLAB

```
1 fb=10000;
2 for i=1:length(n)
3     for j=1:N
4         [PLdata(j), APDdata(j), MPDdata(j), TT(j), PLvoip
           (j), APDvoip(j), MPDvoip(j)] = Simulator3A(
           lambda,c,fb,P,n(i),b);
5     end
6
7     avgPLdata(i) = mean(PLdata);
8     termPLdata(i) = norminv(1-alfa/2)*sqrt(var(PLdata)/N)
9     ;
10
11    avgAPDdata(i) = mean(APDdata);
12    termAPDdata(i) = norminv(1-alfa/2)*sqrt(var(APDdata)/
13    N);
14
15    avgPLvoip(i) = mean(PLvoip);
16    termPLvoip(i) = norminv(1-alfa/2)*sqrt(var(PLvoip)/N)
17    ;
18
19    avgAPDvoip(i) = mean(APDvoip);
20    termAPDvoip(i) = norminv(1-alfa/2)*sqrt(var(APDvoip)/
21    N);
22 end
23
24 figure(5)
25 bar(n,avgAPDdata)
26 hold on
27 erro=errorbar(n,avgAPDdata,termAPDdata,termAPDdata);
28 erro.Color=[0 0 0];
29 erro.LineStyle = 'none';
30 hold off
31 grid on
32 xlabel('n values')
33 ylabel('milliseconds')
34 title('Average Data Packet Delay (milliseconds)')
35
36 figure(6)
37 bar(n,avgPLdata)
38 hold on
39 erro=errorbar(n,avgPLdata,termPLdata,termPLdata);
40 erro.Color=[0 0 0];
```

```

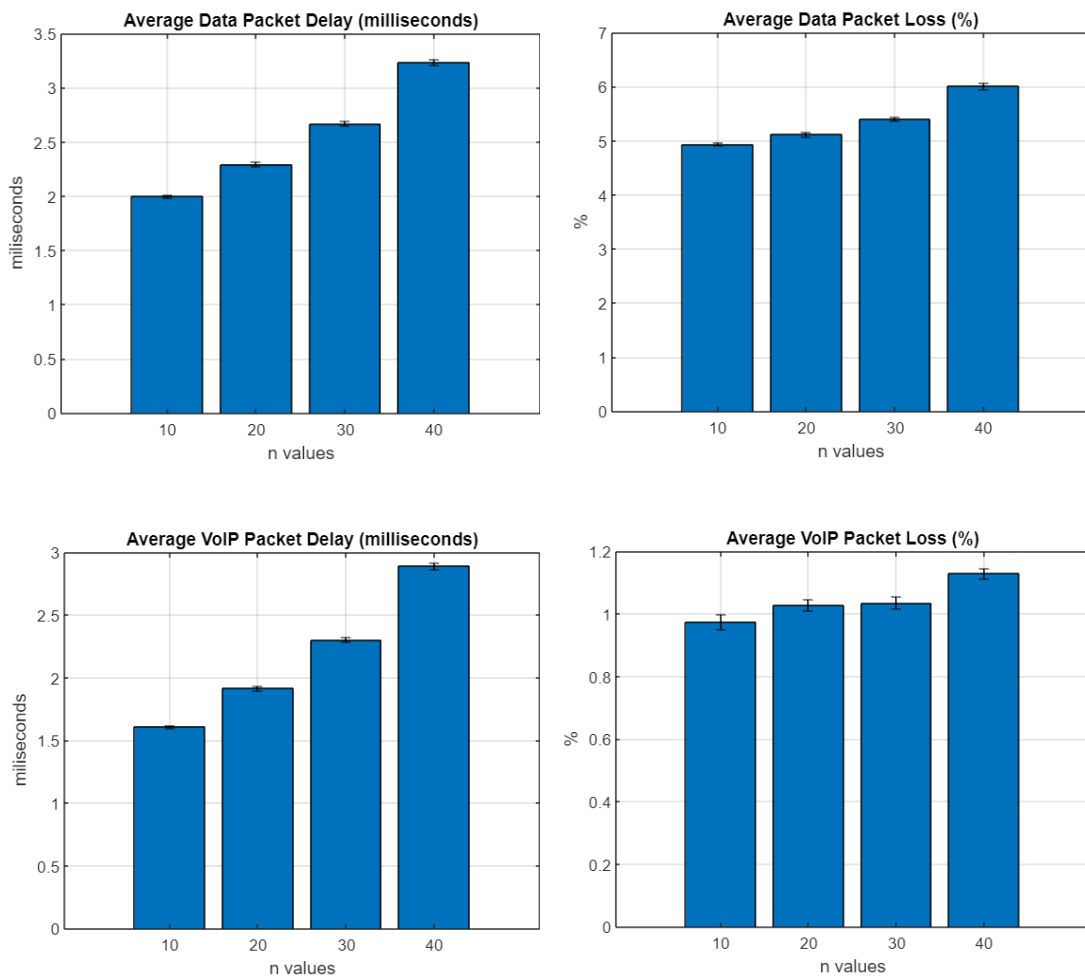
37 erro.LineStyle = 'none';
38 hold off
39 grid on
40 xlabel('n values')
41 ylabel('%')
42 title('Average Data Packet Loss (%)')
43
44 figure(7)
45 bar(n, avgAPDvoip)
46 hold on
47 erro=errorbar(n, avgAPDvoip, termAPDvoip, termAPDvoip);
48 erro.Color=[0 0 0];
49 erro.LineStyle = 'none';
50 hold off
51 grid on
52 xlabel('n values')
53 ylabel('milliseconds')
54 title('Average VoIP Packet Delay (milliseconds)')
55
56 figure(8)
57 bar(n, avgPLvoip)
58 hold on
59 erro=errorbar(n, avgPLvoip, termPLvoip, termPLvoip);
60 erro.Color=[0 0 0];
61 erro.LineStyle = 'none';
62 hold off
63 grid on
64 xlabel('n values')
65 ylabel('%')
66 title('Average VoIP Packet Loss (%)')

```

## Análise do Código

Usando como parâmetros de entrada os valores definidos na alínea anterior, o tamanho da fila é mudado para 10000 e são repetidos os ciclos for para percorrer os vários valores de n e para o número pedido de corridas, são calculadas as métricas de performance e são feitos os gráficos de barras com as respectivas barras de erro,

## Resultados



## Análises e Justificações

Comparando o atraso médio para ambos os tipos de pacote com os resultados obtidos na alínea anterior, é possível concluir que uma diminuição do tamanho da fila faz com que o atraso médio diminua. Isto acontece pois, sendo a fila mais pequena, não é capaz de aceitar tantos pacotes e, portanto, o tempo que os pacotes passam dentro da fila de espera antes de serem atendidos é menor, visto terem menos pacotes à sua frente na fila.

No entanto, com a diminuição do tamanho da fila, verifica-se um aumento da perda média de pacotes pois, se a fila tem uma capacidade menor, significa que há um maior número de pacotes que chega e é descartado por não haver espaço para a sua aceitação na fila, devido a esta já se encontrar cheia.

Concluindo, uma fila maior leva a maiores atrasos mas menores perdas de pacotes, enquanto que uma fila menor leva a menores atrasos mas maiores perdas de pacotes. Assim sendo, o tamanho da fila deve ser ajustado consoante as necessidades e o cenário onde é usada. Por exemplo, na transferência de um ficheiro, é crucial que não haja perdas de pacotes e, como tal, é conveniente ter uma fila maior. Já no caso de tráfego VoIP, é mais importante que não haja atrasos, pelo que uma fila pequena é mais apropriada.

## Task 2.c.

### Código MATLAB

```
1 aux1=[65:109 111:1517];
2 aux3=length(aux1);
3 aux4=(1-(0.19+0.23+0.17))/aux3;
4 ber=10^-5;
5 pld=zeros(1,aux3+3);
6 for i = 1:length(pld)
7     if i == 1
8         pld(i) = 0.19*(100*(1-(nchoosek((i+64-1)*8,0)*ber
9             ^((i+64-1)*8-0)))));
10    elseif i == 110-64+1
11        pld(i) = 0.23*(100*(1-(nchoosek((i+64-1)*8,0)*ber
12            ^((i+64-1)*8-0)))));
13    elseif i == 1518-64+1
14        pld(i) = 0.17*(100*(1-(nchoosek((i+64-1)*8,0)*ber
15            ^((i+64-1)*8-0)))));
16    else
17        pld(i) = aux4*(100*(1-(nchoosek((i+64-1)*8,0)*ber
18            ^((i+64-1)*8-0)))));
19    end
20 end
21 aux2=[110:130];
22 aux5=length(aux2);
23 aux6=1/aux5;
24 plv=zeros(1,aux5);
25 for j = 1:length(plv)
26     plv(j) = aux6*(100*(1-(nchoosek((j+110-1)*8,0)*ber
27         ^((j+110-1)*8-0)))));
28 end
29 apld = sum(pld);
30 aplv = sum(plv);
31
32 fprintf("Theoretical average packet loss of data packets
33     with BER = 10^-5 -> %.3f %%\n", apld);
34 fprintf("Theoretical average packet loss of VoIP packets
35     with BER = 10^-5 -> %.3f %%\n", aplv);
```

## Análise do Código

Para os pacotes de dados, foi criado inicialmente um vetor com todos os tamanhos de pacotes equiprováveis e foi calculada essa mesma probabilidade. Em seguida foi declarado o BER e criada uma matriz com tamanho igual ao número de tamanhos que os pacotes podem ter, de modo a armazenar a perda de pacotes para cada tamanho. O próximo passo foi fazer um ciclo for para percorrer todos os tamanhos e, consoante o tamanho do pacote, calcular a probabilidade de o pacote ter pelo menos um erro devido ao BER (usando a função de probabilidade para variáveis aleatórias binomiais) e multiplicar essa probabilidade pela respetiva probabilidade do tamanho do pacote. Foi calculada a probabilidade de o pacote ter pelo menos um erro pois é essa a condição para que o pacote seja perdido, ou seja, ao calcular a probabilidade de um pacote de um tamanho específico ter um ou mais erros, estamos a calcular a média da perda de pacotes para um pacote com esse tamanho.

Para os pacotes VoIP foi aplicado um raciocínio idêntico, foi criado um vetor com todos os tamanhos de pacote possíveis, foi calculada a probabilidade de cada tamanho e foi criada uma matriz para armazenar a perda de pacotes de todos os tamanhos. Em seguida foi feito um ciclo for para calcular a probabilidade de o pacote ter pelo menos um erro devido ao BER (mais uma vez, usando a função de probabilidade para variáveis aleatórias binomiais) e foi multiplicado esse resultado pela probabilidade do tamanho do pacote. Por fim, foi feita a soma das médias para a perda de ambos os tipos de pacotes e foi apresentado o resultado.

## Resultados

```
Theoretical average packet loss of data packets with BER = 10^-5 -> 4.737 %  
Theoretical average packet loss of VoIP packets with BER = 10^-5 -> 0.955 %
```

## Análises e Justificações

Quando comparamos os resultados teóricos com os valores das simulações, é possível ver que os resultados de ambas as simulações se aproximam muito do resultado teórico. No entanto, os resultados obtidos na alínea 2a, em que a fila era maior, são mais próximos que os resultados obtidos na alínea 2b, onde a fila era menor. Isto deve-se ao facto de a fórmula teórica apenas ter em conta o BER e o tamanho dos pacotes e não considerar o tamanho da fila. Assim sendo, podemos concluir que a fórmula teórica para o cálculo da perda média de pacotes assume uma fila infinita e assim, sendo a fila na alínea 2a maior que na alínea 2b, os resultados vão ser mais próximos pois o tamanho da fila também é mais próximo.

# Task 3

## Task 3.a.

### Código MATLAB

```
1 lambda = 1500;
2 c=10;
3 f=10000;
4 P=100000;
5 N=20;
6 n=[10,20,30,40];
7 alfa=0.1;
8 PLdata = zeros(1,N);
9 APDdata = zeros(1,N);
10 PLvoip = zeros(1,N);
11 APDvoip = zeros(1,N);
12
13 for i=1:length(n)
14     for j=1:N
15         [PLdata(j), APDdata(j), MPDdata(j), TT(j), PLvoip
16             (j), APDvoip(j), MPDvoip(j)] = Simulator3(
17             lambda,c,f,P,n(i));
18     end
19     avgPLdata(i) = mean(PLdata);
20     termPLdata(i) = norminv(1-alfa/2)*sqrt(var(PLdata)/N)
21     ;
22     avgAPDdata(i) = mean(APDdata);
23     termAPDdata(i) = norminv(1-alfa/2)*sqrt(var(APDdata)/
24         N);
25     avgPLvoip(i) = mean(PLvoip);
26     termPLvoip(i) = norminv(1-alfa/2)*sqrt(var(PLvoip)/N)
27     ;
```

```

26
27     avgAPDvoip(i) = mean(APDvoip);
28     termAPDvoip(i) = norminv(1-alfa/2)*sqrt(var(APDvoip)/
29         N);
30
31 end
32
33 figure(1)
34 bar(n,avgAPDdata)
35 hold on
36 erro=errorbar(n,avgAPDdata,termAPDdata,termAPDdata);
37 erro.Color=[0 0 0];
38 erro.LineStyle = 'none';
39 hold off
40 grid on
41 xlabel('n values')
42 ylabel('milliseconds')
43 title('Average Data Packet Delay (milliseconds)')
44
45 figure(2)
46 bar(n,avgPLdata)
47 hold on
48 erro=errorbar(n,avgPLdata,termPLdata,termPLdata);
49 erro.Color=[0 0 0];
50 erro.LineStyle = 'none';
51 hold off
52 grid on
53 xlabel('n values')
54 ylabel('%')
55 title('Average Data Packet Loss (%)')
56
57 figure(3)
58 bar(n,avgAPDvoip)
59 hold on
60 erro=errorbar(n,avgAPDvoip,termAPDvoip,termAPDvoip);
61 erro.Color=[0 0 0];
62 erro.LineStyle = 'none';
63 hold off
64 grid on
65 xlabel('n values')
66 ylabel('milliseconds')
67 title('Average VoIP Packet Delay (milliseconds)')
68
69 figure(4)
70 bar(n,avgPLvoip)
71 hold on
72 erro=errorbar(n,avgPLvoip,termPLvoip,termPLvoip);

```

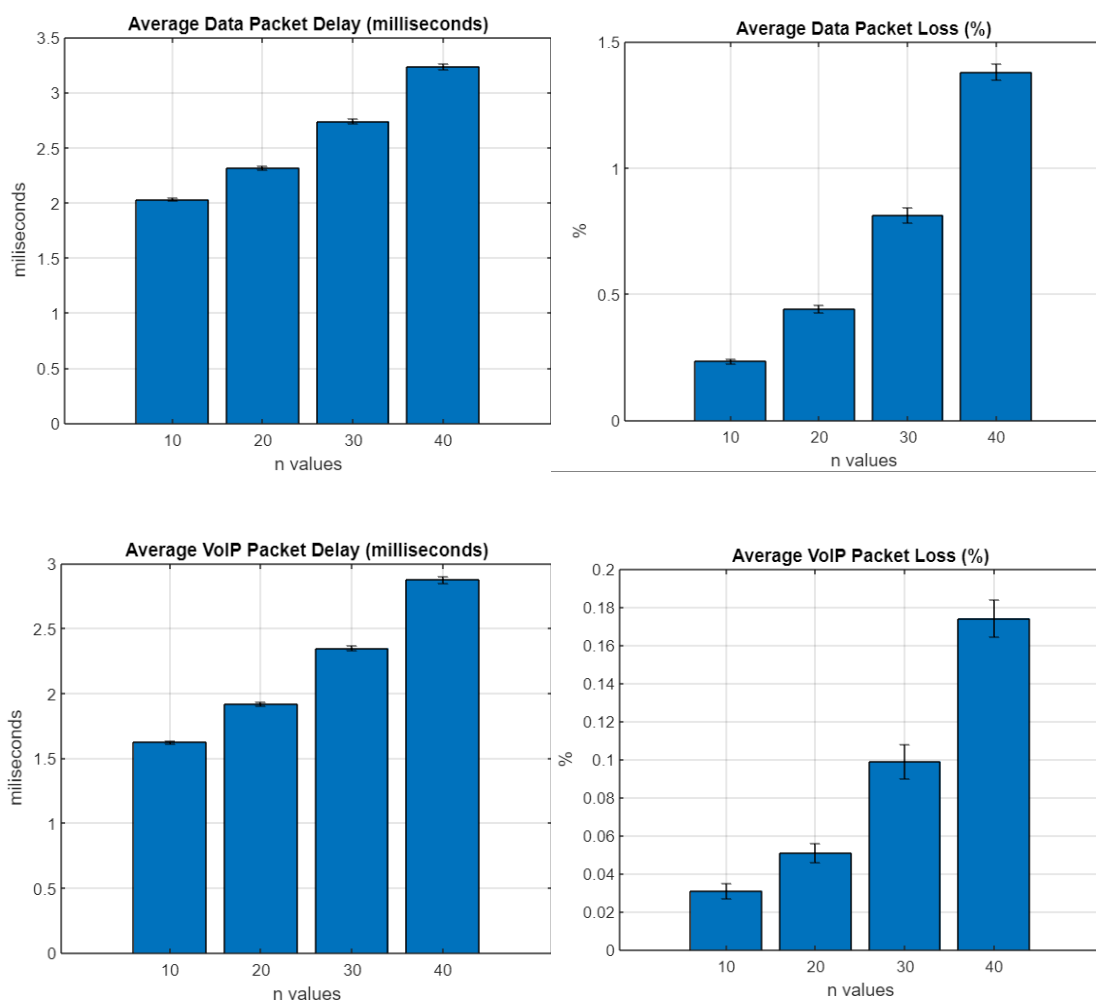


```
71 erro.Color=[0 0 0];
72 erro.LineStyle = 'none';
73 hold off
74 grid on
75 xlabel('n values')
76 ylabel('%')
77 title('Average VoIP Packet Loss (%)');
```

## Análise do Código

Inicialmente são criadas matrizes de zeros para armazenar os valores provenientes da simulação e em seguida são criados dois ciclos for para percorrer os vários valores de n e fazer o número de corridas pedido. Dentro dos ciclos for é chamado o Simulator3 com os respectivos parâmetros e os valores de saída vão sendo guardados nas respectivas matrizes. Por fim são desenhados os gráficos pedidos com as respectivas barras de erro.

## Resultados



## Análises e Justificações

Através da observação dos gráficos é possível ver que, à medida que aumenta o número de fluxos de pacotes VoIP, aumentam também o atraso médio e a perda de pacotes para ambos os tipos de dados, quando ambos são multiplexados na mesma fila. Tal como explicado anteriormente, o aumento no atraso médio deve-se ao facto de haver mais pacotes na fila de espera, o que faz com que os

últimos pacotes a chegar demorem mais a ser atendidos pois têm mais pacotes à sua frente na fila. Por sua vez, o aumento na perda de pacotes provém do facto de que, com um aumento do número de pacotes no sistema, a fila de espera enche mais rapidamente, fazendo com que pacotes que cheguem à fila quando esta já está cheia sejam descartados e, consequentemente, perdidos.

Quanto às diferenças entre dados e VoIP para o mesmo valor de fluxo de pacotes, a maior taxa de perda de pacotes do tipo de dados pode ser explicada pelo facto de os pacotes terem um tamanho maior, tal como foi explicado anteriormente. Quanto ao atraso médio, apesar de a diferença ser pequena, o maior atraso nos pacotes de dados comprova a hipótese colocada anteriormente, de que quanto maior for o tamanho do pacote, maior o atraso no seu atendimento.

### Task 3.b.

#### Código MATLAB

```
1 for i=1:length(n)
2     for j=1:N
3         [PLdata(j), APDdata(j), MPDdata(j), TT(j), PLvoip
4             (j), APDvoip(j), MPDvoip(j)] = Simulator4(
5                 lambda,c,f,P,n(i));
6     end
7     avgPLdata(i) = mean(PLdata);
8     termPLdata(i) = norminv(1-alfa/2)*sqrt(var(PLdata)/N)
9         ;
10    avgAPDdata(i) = mean(APDdata);
11    termAPDdata(i) = norminv(1-alfa/2)*sqrt(var(APDdata)/
12        N);
13    avgPLvoip(i) = mean(PLvoip);
14    termPLvoip(i) = norminv(1-alfa/2)*sqrt(var(PLvoip)/N)
15        ;
16    avgAPDvoip(i) = mean(APDvoip);
17    termAPDvoip(i) = norminv(1-alfa/2)*sqrt(var(APDvoip)/
18        N);
19 end
20 figure(5)
21 bar(n,avgAPDdata)
22 hold on
23 erro=errorbar(n,avgAPDdata,termAPDdata,termAPDdata);
24 erro.Color=[0 0 0];
25 erro.LineStyle = 'none';
26 hold off
27 grid on
28 xlabel('n values')
29 ylabel('milliseconds')
30 title('Average Data Packet Delay (milliseconds)')
31 figure(6)
32 bar(n,avgPLdata)
33 hold on
34 erro=errorbar(n,avgPLdata,termPLdata,termPLdata);
35 erro.Color=[0 0 0];
36 erro.LineStyle = 'none';
```

```

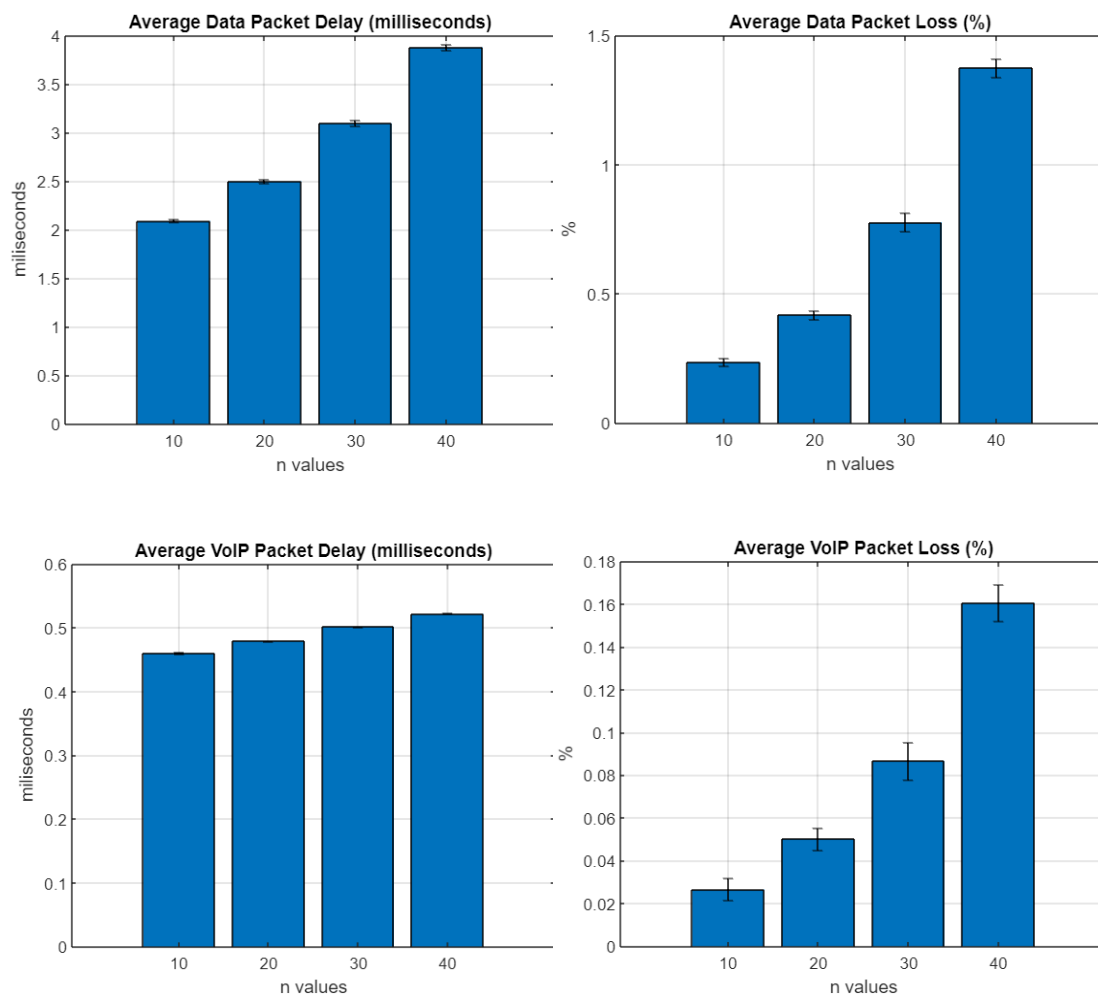
37 hold off
38 grid on
39 xlabel('n values')
40 ylabel('%')
41 title('Average Data Packet Loss (%)')
42
43 figure(7)
44 bar(n, avgAPDvoip)
45 hold on
46 erro=errorbar(n, avgAPDvoip, termAPDvoip, termAPDvoip);
47 erro.Color=[0 0 0];
48 erro.LineStyle = 'none';
49 hold off
50 grid on
51 xlabel('n values')
52 ylabel('milliseconds')
53 title('Average VoIP Packet Delay (milliseconds)')
54
55 figure(8)
56 bar(n, avgPLvoip)
57 hold on
58 erro=errorbar(n, avgPLvoip, termPLvoip, termPLvoip);
59 erro.Color=[0 0 0];
60 erro.LineStyle = 'none';
61 hold off
62 grid on
63 xlabel('n values')
64 ylabel('%')
65 title('Average VoIP Packet Loss (%)')

```

## Análise do Código

O código para a resolução desta alínea é igual ao da alínea anterior, com a exceção de que desta vez é invocado o Simulator4.

## Resultados



## Análises e Justificações

Como podemos ver nos gráficos dos pacotes de dados, a perda média de pacotes manteve-se inalterada em relação ao exercício anterior, no entanto, o atrasado médio aumentou ligeiramente. Isto acontece pois o Simulator4 dá prioridade aos pacotes VoIP, o que significa que, enquanto houver pacotes VoIP na fila para serem atendidos, os pacotes de dados vão continuar na fila à espera

da sua vez, o que aumenta o tempo de atendimento. No entanto, isso não se traduz num aumento da perda de pacotes, uma vez que estes continuam a ser aceites na fila.

Já no caso dos pacotes VoIP, podemos observar que, à semelhança da alínea anterior, a perda de pacotes não ultrapassa os 0.2%. No entanto, há uma diminuição drástica no atraso de atendimento dos pacotes VoIP pois, tal como já foi explicado, o Simulator4 dá prioridade a estes pacotes. Enquanto houver pacotes VoIP na fila para serem atendidos, estes vão ser sempre atendidos primeiro que os pacotes de dados, diminuindo assim o tempo médio de atendimento.

## Simulator4c

### Código MATLAB

```
1  case ARRIVAL                                % If first event is an
   ARRIVAL
2      if type == 0 % Data packet
3          TOTALPACKETS= TOTALPACKETS+1;
4          tmp= Clock + exprnd(1/lambda);
5          EventList = [EventList; ARRIVAL, tmp,
                        GeneratePacketSize(), tmp, 0];
6      if STATE==0
7          STATE= 1;
8          EventList = [EventList; DEPARTURE,
                        Clock + 8*PacketSize/(C*10^6),
                        PacketSize, Clock, 0];
9      else
10         % Queue occupation must not become
11         higher than 90%
12         if ((QUEUEOCCUPATION+PacketSize)/f)
13             <= 0.9
14             QUEUE= [QUEUE; PacketSize, Clock,
15                     0];
16             QUEUEOCCUPATION= QUEUEOCCUPATION
17                             + PacketSize;
18         else
19             LOSTPACKETS= LOSTPACKETS + 1;
20         end
21     end
22 end
```

### Análise do Código

Tendo como base o Simulator4, e tendo como condição os pacotes de dados serem aceites apenas se a ocupação da fila for menor que noventa por cento, a única alteração necessária foi na condição de aceitação dos pacotes de dados na fila. Assim sendo, em vez de os pacotes de dados serem aceites caso a ocupação seja menor ou igual que o tamanho da fila, passam a ser aceites apenas se o resultado da divisão da ocupação da fila pelo seu tamanho for menor ou igual a 0.9, sendo que os pacotes VoIP não sofrem alteração, sendo aceites sempre que há espaço na fila.



### Task 3.c.

#### Código MATLAB

```
1 for i=1:length(n)
2     for j=1:N
3         [PLdata(j), APDdata(j), MPDdata(j), TT(j), PLvoip
4             (j), APDvoip(j), MPDvoip(j)] = Simulator4c(
5                 lambda,c,f,P,n(i));
6     end
7     avgPLdata(i) = mean(PLdata);
8     termPLdata(i) = norminv(1-alfa/2)*sqrt(var(PLdata)/N)
9         ;
10    avgAPDdata(i) = mean(APDdata);
11    termAPDdata(i) = norminv(1-alfa/2)*sqrt(var(APDdata)/
12        N);
13    avgPLvoip(i) = mean(PLvoip);
14    termPLvoip(i) = norminv(1-alfa/2)*sqrt(var(PLvoip)/N)
15        ;
16    avgAPDvoip(i) = mean(APDvoip);
17    termAPDvoip(i) = norminv(1-alfa/2)*sqrt(var(APDvoip)/
18        N);
19 end
20 figure(9)
21 bar(n,avgAPDdata)
22 hold on
23 erro=errorbar(n,avgAPDdata,termAPDdata,termAPDdata);
24 erro.Color=[0 0 0];
25 erro.LineStyle = 'none';
26 hold off
27 grid on
28 xlabel('n values')
29 ylabel('milliseconds')
30 title('Average Data Packet Delay (milliseconds)')
31 figure(10)
32 bar(n,avgPLdata)
33 hold on
34 erro=errorbar(n,avgPLdata,termPLdata,termPLdata);
35 erro.Color=[0 0 0];
36 erro.LineStyle = 'none';
```

```

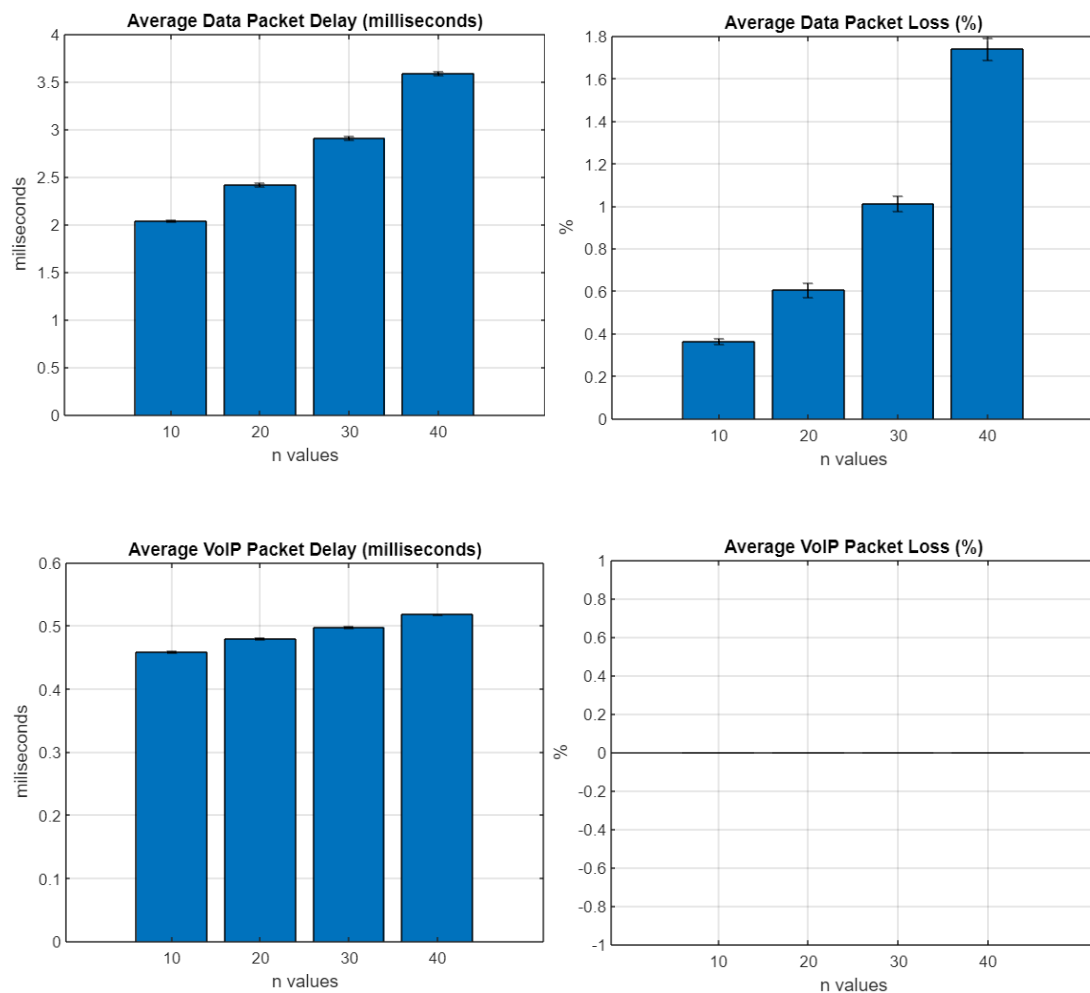
37 hold off
38 grid on
39 xlabel('n values')
40 ylabel('%')
41 title('Average Data Packet Loss (%)')
42
43 figure(11)
44 bar(n, avgAPDvoip)
45 hold on
46 erro=errorbar(n, avgAPDvoip, termAPDvoip, termAPDvoip);
47 erro.Color=[0 0 0];
48 erro.LineStyle = 'none';
49 hold off
50 grid on
51 xlabel('n values')
52 ylabel('milliseconds')
53 title('Average VoIP Packet Delay (milliseconds)')
54
55 figure(12)
56 bar(n, avgPLvoip)
57 hold on
58 erro=errorbar(n, avgPLvoip, termPLvoip, termPLvoip);
59 erro.Color=[0 0 0];
60 erro.LineStyle = 'none';
61 hold off
62 grid on
63 xlabel('n values')
64 ylabel('%')
65 title('Average VoIP Packet Loss (%)')

```

## Análise do Código

Novamente, o código para a resolução desta alínea é igual ao da alínea anterior, com a exceção de que desta vez é invocado o Simulator4c.

## Resultados



## Análises e Justificações

Olhando para os gráficos é possível observar um aumento na perda de pacotes de dados e uma diminuição drástica na perda de pacotes VoIP quando comparados com a alínea anterior, sendo que os tempos de atendimento para ambos os tipos de pacotes se mantiveram inalterados, sendo que ambas as métricas aumentam com o aumento do número de fluxos.

A perda de pacotes VoIP é nula pois, para além de estes terem prioridade em relação aos pacotes de dados, a versão do simulador usada neste alínea faz com que os pacotes de VoIP sejam sempre aceites desde que haja espaço na fila, ao passo que os pacotes de dados apenas são aceites caso a ocupação da fila não exceda os 90%. Efetivamente, isto faz com que os pacotes de dados e VoIP partilhem 90% da fila e os restantes 10% são reservados exclusivamente para pacotes VoIP o que, aliado ao facto de estes terem prioridade, faz com que a sua perda desça para 0%.

Já o aumento na perda de pacotes de dados pode ser justificado pelo facto de haver menos 10% de espaço para pacotes de dados, o que faz com que haja menos pacotes deste tipo na fila e os restantes pacotes de dados que seriam aceites normalmente, são assim descartados, aumentando a perda média de pacotes de dados.

## Simulator4d

### Código MATLAB

```
1 elseif type == 1 % VoIP packet
2     TOTALPACKETSV= TOTALPACKETSV+1;
3     tmp= Clock + unifrnd(0.016,0.024);
4     EventList = [EventList; ARRIVAL, tmp,
5         randi([110,130]), tmp, 1];
6     if STATE==0
7         STATE= 1;
8         EventList = [EventList; DEPARTURE,
9             Clock + 8*PacketSize/(C*10^6),
10             PacketSize, Clock, 1];
11     else
12         % Queue occupation must not become
13         % higher than 90%
14         if ((QUEUEOCCUPATION+PacketSize)/f)
15             <= 0.9
16             QUEUE= [QUEUE; PacketSize , Clock ,
17                 1];
18             %Sort Queue by type (Data or VoIP
19             %)
20             % VoIP vao ter prioridade sobre os de
21             % data
22             QUEUE= sortrows(QUEUE, 3, "
23                 descend");
24             QUEUEOCCUPATION= QUEUEOCCUPATION
25                 + PacketSize;
26         else
27             LOSTPACKETSV= LOSTPACKETSV + 1;
28         end
29     end
30 end
31 end
```

## **Análise do Código**

Pegando de novo no Simulator4, a única alteração introduzida para o desenvolvimento deste simulador foi igual à alteração introduzida para a realização do Simulator4c mas desta vez para os pacotes VoIP. Assim sendo, os pacotes VoIP passam a ser aceites apenas se o resultado da divisão da ocupação da fila pelo seu tamanho for menor ou igual a 0.9, sendo que os pacotes de dados não sofrem alteração, sendo aceites sempre que há espaço na fila.

### Task 3.d.

#### Código MATLAB

```
1 for i=1:length(n)
2     for j=1:N
3         [PLdata(j), APDdata(j), MPDdata(j), TT(j), PLvoip
4             (j), APDvoip(j), MPDvoip(j)] = Simulator4d(
5                 lambda,c,f,P,n(i));
6     end
7     avgPLdata(i) = mean(PLdata);
8     termPLdata(i) = norminv(1-alfa/2)*sqrt(var(PLdata)/N)
9         ;
10    avgAPDdata(i) = mean(APDdata);
11    termAPDdata(i) = norminv(1-alfa/2)*sqrt(var(APDdata)/
12        N);
13    avgPLvoip(i) = mean(PLvoip);
14    termPLvoip(i) = norminv(1-alfa/2)*sqrt(var(PLvoip)/N)
15        ;
16    avgAPDvoip(i) = mean(APDvoip);
17    termAPDvoip(i) = norminv(1-alfa/2)*sqrt(var(APDvoip)/
18        N);
19 end
20 figure(13)
21 bar(n,avgAPDdata)
22 hold on
23 erro=errorbar(n,avgAPDdata,termAPDdata,termAPDdata);
24 erro.Color=[0 0 0];
25 erro.LineStyle = 'none';
26 hold off
27 grid on
28 xlabel('n values')
29 ylabel('milliseconds')
30 title('Average Data Packet Delay (milliseconds)')
31 figure(14)
32 bar(n,avgPLdata)
33 hold on
34 erro=errorbar(n,avgPLdata,termPLdata,termPLdata);
35 erro.Color=[0 0 0];
36 erro.LineStyle = 'none';
```

```

37 hold off
38 grid on
39 xlabel('n values')
40 ylabel('%')
41 title('Average Data Packet Loss (%)')
42
43 figure(15)
44 bar(n, avgAPDvoip)
45 hold on
46 erro=errorbar(n, avgAPDvoip, termAPDvoip, termAPDvoip);
47 erro.Color=[0 0 0];
48 erro.LineStyle = 'none';
49 hold off
50 grid on
51 xlabel('n values')
52 ylabel('milliseconds')
53 title('Average VoIP Packet Delay (milliseconds)')
54
55 figure(16)
56 bar(n, avgPLvoip)
57 hold on
58 erro=errorbar(n, avgPLvoip, termPLvoip, termPLvoip);
59 erro.Color=[0 0 0];
60 erro.LineStyle = 'none';
61 hold off
62 grid on
63 xlabel('n values')
64 ylabel('%')
65 title('Average VoIP Packet Loss (%)')

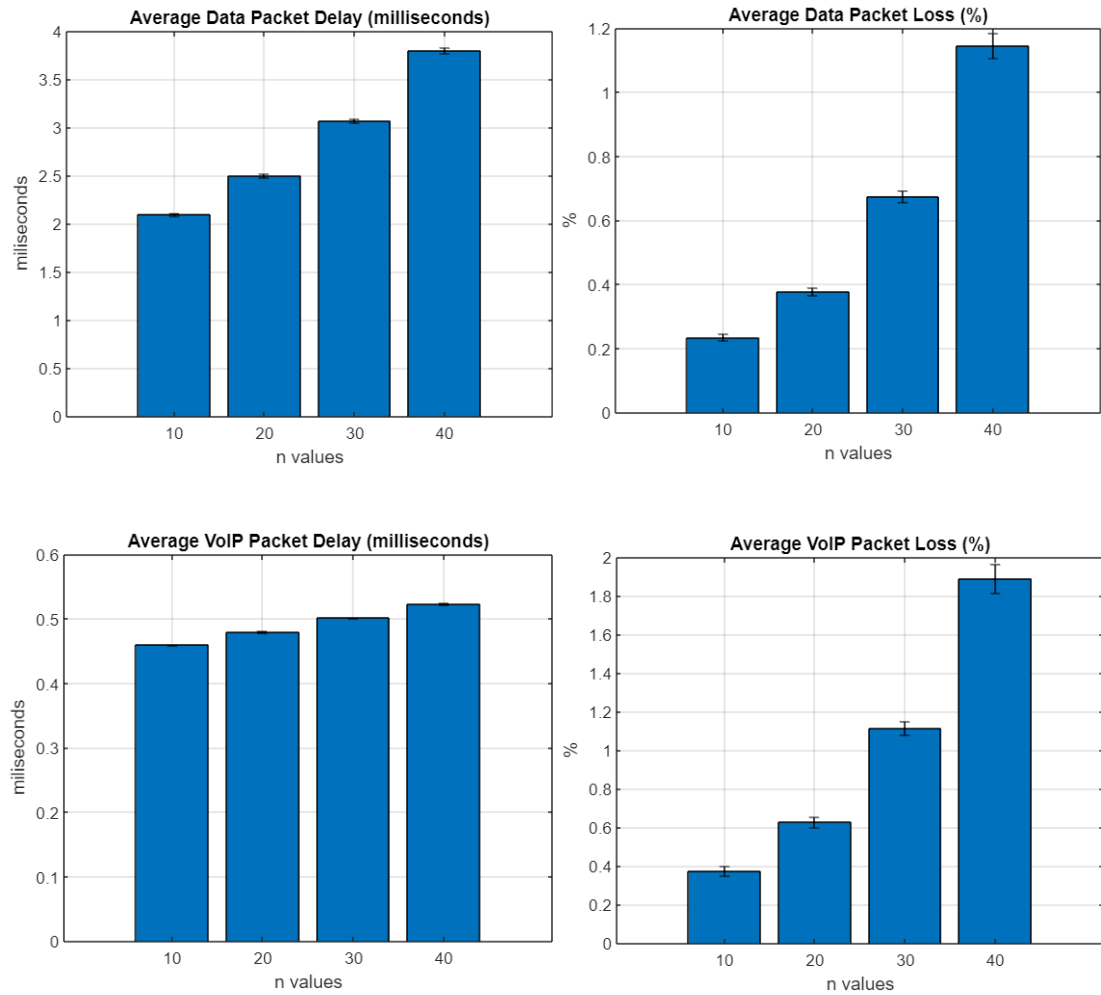
```

## Análise do Código

Mais uma vez, o código para a resolução desta alínea é igual ao da alínea anterior, com a exceção de que desta vez é invocado o Simulator4d.



## Resultados



## Análises e Justificações

Mais uma vez é possível observar que os valores de atraso e perda de pacotes aumentam com o aumento do número de fluxos. Neste caso, os valores de atraso para ambos os tipos de pacotes mantêm-se inalterado, sendo os valores dessas métricas iguais aos obtidos nas alíneas 3b e 3c.

Já os valores para a perda de pacotes apresentam ligeiras diferenças, diferenças essas que podem ser justificadas pela condição introduzida no simulador utilizado para esta alínea. Desta vez, os pacotes de dados são sempre aceites desde que haja espaço na fila e são os pacotes VoIP que apenas são aceites caso a ocupação da fila não ultrapasse os 90%. Deste modo, a perda de pacotes de dados diminui, tanto em relação à alínea 3b, como à 3c, uma vez que, tal como na alínea anterior, os pacotes de dados e VoIP partilham 90% da fila, desta vez são os pacotes de dados que têm os restantes 10% reservados para si. Assim sendo, visto que é possível haver mais pacotes de dados na fila do que antes, a quantidade de pacotes descartados deste tipo diminui, o que diminui também a perda média de pacotes.

Pelo contrário, a perda de pacotes VoIP aumentou comparativamente com as duas alíneas anteriores pois, apesar de continuarem a ter prioridade no atendimento, existe menos 10% de espaço para pacotes VoIP, o que faz com que o número de pacotes deste tipo na fila seja menor e os que estariam normalmente na fila, são assim descartados, aumentando a perda média de pacotes.