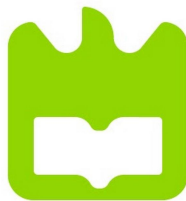


PERFORMANCE EVALUATION OF POINT-TO-POINT LINKS SUPPORTING PACKET SERVICES

Universidade de Aveiro

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

Relatório



António Domingues 89007 & Henrique Silva 88857

Professor Amaro de Sousa

01 de Dezembro de 2021

Índice

Tarefa 1	3
1.a	3
1.b	5
1.c	8
1.d	10
1.e	12
 Tarefa 2	 15
2.a	15
2.b	17
2.c	18
2.d	20
2.e	22
2.f	24

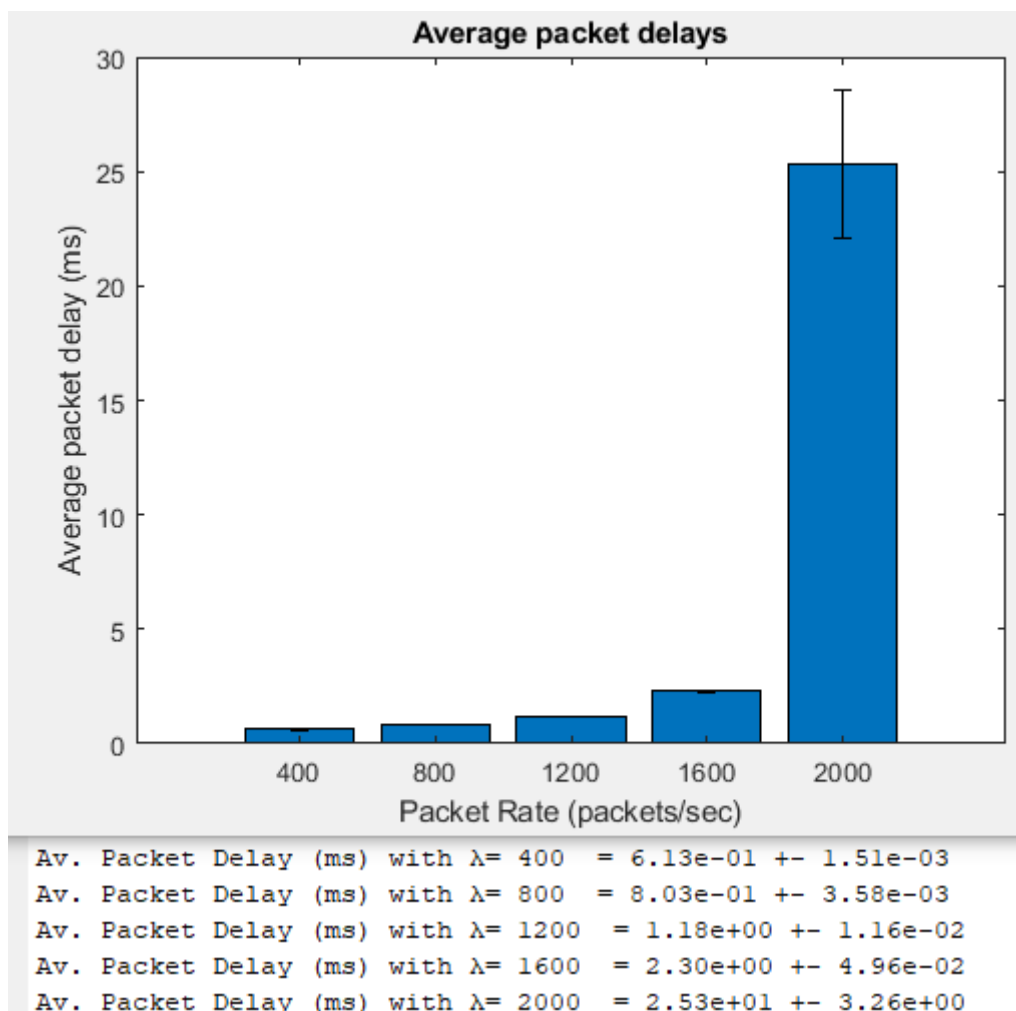
Tarefa 1

1.a

Nesta primeira tarefa, é utilizado o *Simulator1*, previamente fornecido nas aulas, logo, como concluído, este modela uma ligação de um **sistema M/G/1**.

Nesta primeira alínea, faz-se apenas variar a taxa de pacotes (λ) que chegam ao sistema. A capacidade do link, o tamanho da fila e o número de pacotes (critério de paragem), são sempre os mesmos para os vários valores de λ .

Com isto, após correr o código matlab para os vários valores de *lambda*, obteve-se o seguinte resultado:



Após uma breve análise do gráfico obtido, pode dizer-se de imediato que quanto maior for a taxa de chegada (λ), maior vai ser o atraso médio de pacotes.

Como o atraso médio no sistema é dado por: $W = \frac{(\lambda E[S^2])}{2(1-\lambda E[S])} + E[S]$ o seu valor vai crescer de forma exponencial. Assim, pode observar-se o mesmo nos resultados obtidos. Para um valor de $\lambda=400$, $\lambda=800$, $\lambda=1200$, os valores do atraso vão crescendo pouco a pouco, até que chega a um determinado valor λ , em que o valor do atraso médio, vai aumentar significativamente.

Como a as características do sistema são sempre as mesmas, para os vários λ , e a taxa de chegada de pacotes aumenta, chegamos a um ponto, em que o sistema não é capaz de processar em tempo útil (a capacidade do link já não é suficiente), todos os pacotes que lhe vão chegando, daí existir um aumento do atraso médio de pacotes. Outro fator a acrescentar, é o fato do número de pacotes que está na fila, ser também, exponencialmente maior. Como o sistema possui uma fila sem prioridades e do tipo *FIFO* (First In, First Out), conclui-se que para valores de λ maiores, um pacote que é inserido por último no fim da fila, irá demorar uma quantidade superior de tempo até que seja atendido.

Refletindo sobre os intervalos de erro, quando estes forem menores, pode afirmar-se que temos bons e viáveis resultados de simulação.

```
lambda = [400 800 1200 1600 2000];
C=10;
f=1000000;
P=10000;
N=50; %Run Simulator1 50 times
per1 = zeros(5,N); % 5 arrays de 50 posições
per2 = zeros(5,N); %average packet delay storage
per3 = zeros(5,N);
per4 = zeros(5,N);
for i = 1 : 5
    for it = 1 : N
        [per1(i,it), per2(i,it), per3(i,it), per4(i,it)] = Simulator1(lambda(i),C,f,P);
    end
end
alfa = 0.1; % 90% confidence interval
term1=zeros(1,5); %para armazenar os diversos erros
avgpacketDelay= zeros(1,5);
errHigh= zeros(1,5);
errLow = zeros(1,5);
for i = 1 : 5
    avgpacketDelay(i) = mean(per2(i, 1:N)); %média dos average packet delays para os vários lambda
    term1(i) = norminv(1-alfa/2)*sqrt(var(per2(i, 1:N))/N); %cálculo do erro
    fprintf('Av. Packet Delay (ms) with λ= %d = %.2e +- %.2e\n',lambda(i), avgpacketDelay(i),term1(i))
    errHigh(i) = term1(i);
    errLow(i) = -(term1(i));
```

O código matlab criado para esta alínea, é semelhante ao utilizado na aula, com a diferença que este armazena 5 valores de atraso médio (para os vários λ).

1.b

Na alínea b) da tarefa 1, os parâmetros de entrada para o simulador 1 são os seguintes:

```
C = 10; %MBbps
f = [100000 20000 10000 2000]; %tamanho da fila
P = 10000; %pacotes (stop criterium)
lambda = 1800; %pps
```

De seguida foram inicializadas as matrizes onde são armazenados os valores retornados pelas simulações. Apesar de haver interesse apenas no packet loss e no packet delay, foram criadas 4 matrizes pois o simulador retorna sempre os 4 parâmetros de performance. As matrizes mean foram criadas para armazenar os valores médios das simulações e as matrizes term para os intervalos de confiança.

```
PLmatriz = zeros(size,N);
APDmatriz = zeros(size,N);
MPDmatriz = zeros(size,N);
TTmatriz = zeros(size,N);
PLmean = zeros(1,size);
APDmean = zeros(1,size);
PLterm = zeros(1,size);
APDterm = zeros(1,size);
```

Após a inicialização das variáveis necessárias, foram criados 2 ciclos for. Um para percorrer o array de valores de f e o outro para chamar o simulador 1 50 vezes para cada valor de f.

```
for j = 1:size
    for i = 1:N
        [PLmatriz(j,i), APDmatriz(j,i), MPDmatriz(j,i), TTmatriz(j,i)] = Simulator1(lambda,C,f(j),P);
    end

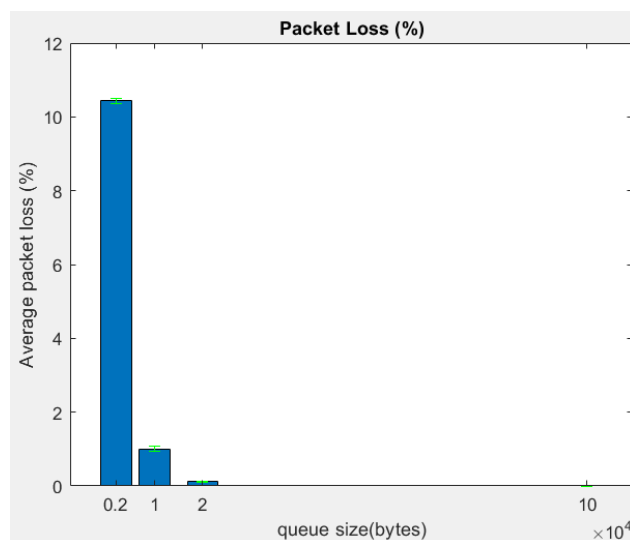
    PLmean(j) = mean(PLmatriz(j,:));
    APDmean(j) = mean(APDmatriz(j,:));
    PLterm(j) = norminv(1-alfa/2)*sqrt(var(PLmatriz(j,:))/N);
    APDterm(j) = norminv(1-alfa/2)*sqrt(var(APDmatriz(j,:))/N);
end
```

Por fim foram criados os gráficos, um para o packet loss e outro para o packet delay.

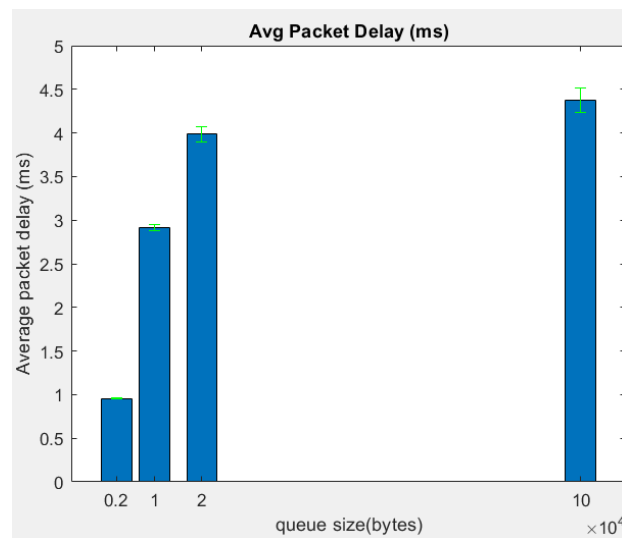
```
%grafico do packet loss
figure(1)
bar(f, PLmean)
hold on
er = errorbar(f, PLmean, PLterm);
er.Color = [0 1 0];
er.LineStyle = 'none';
title('Packet Loss (%)');
hold off

%grafico do packet delay
figure(2)
bar(f, APDmean)
hold on
er = errorbar(f, APDmean, APDterm);
er.Color = [0 1 0];
er.LineStyle = 'none';
title('Avg Packet Delay (ms)');
hold off
```

No gráfico do packet loss, é notório um decréscimo bastante acentuado da percentagem de pacotes perdidos. Esta descida é inversa ao tamanho da fila, isto é, à medida que o tamanho da fila cresce, a perda de pacotes diminui. Se o tamanho da fila for pequeno ($f=2000$) é possível observar que a perda de pacotes é superior a 10%. Ou seja, 10% dos pacotes recebidos são perdidos, pois a fila não tem capacidade para os reter. No entanto, se o tamanho da fila for aumentado para os 10000 pacotes, a perda de pacotes sofre uma queda abrupta para cerca de 1%. Para uma fila de tamanho muito grande ($f=100000$), os pacotes perdidos são muito poucos ou nenhum. Isto acontece pois a fila tem capacidade para reter muito mais pacotes, evitando assim que estes sejam perdidos.

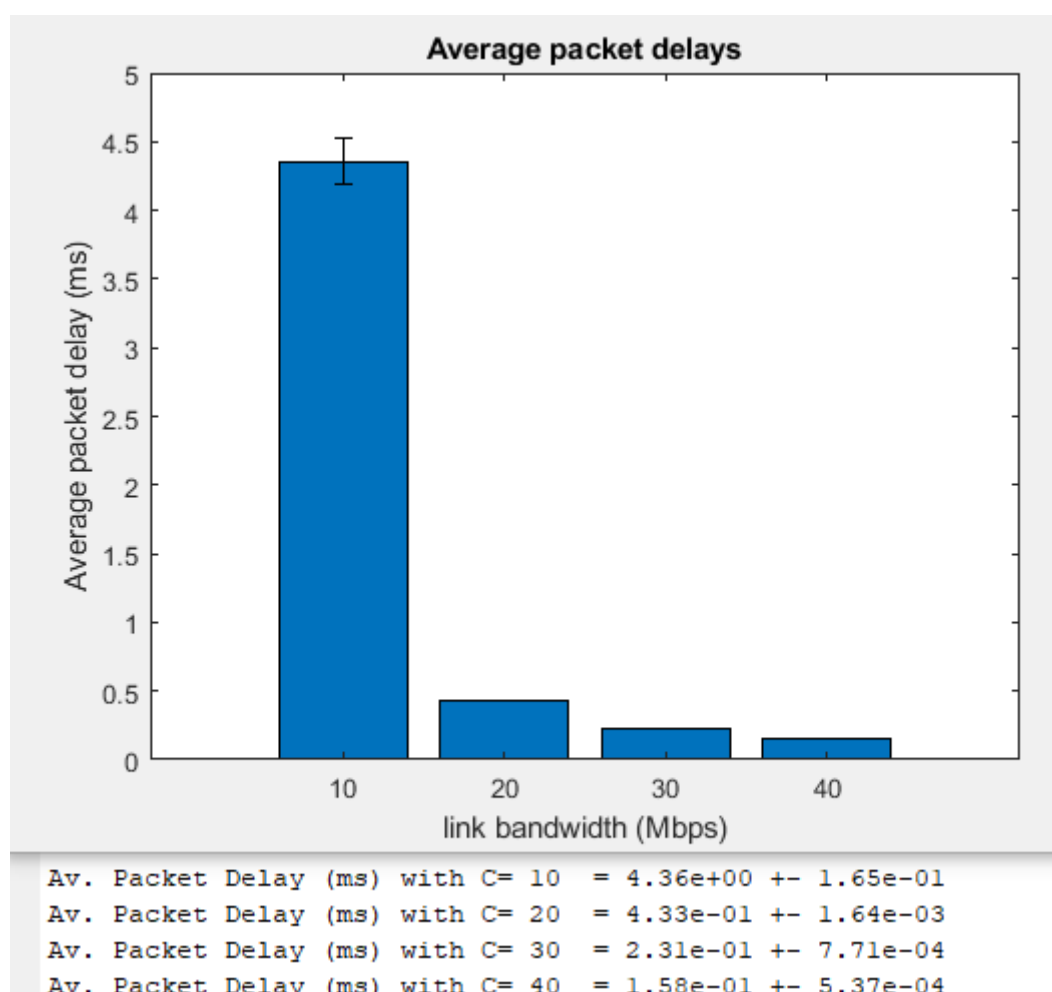


No gráfico do average packet delay o comportamento é diferente do gráfico anterior. Isto porque o average packet delay aumenta consoante o aumento da fila. Por exemplo, para uma fila de 2000 bytes o atraso médio é de aproximadamente 1 ms. No entanto, se a capacidade da fila for aumentada para 10000 bytes o atraso médio já sobe para aproximadamente 3 ms. Isto acontece porque uma fila de maior capacidade implica mais pacotes a serem transmitidos, o que quer dizer que o tempo de espera passa a ser maior. Isto provoca um aumento do tempo médio de atraso. Por outro lado, se a fila for de menor capacidade o número de pacotes atendidos é mais reduzido, visto que a probabilidade de serem perdidos é mais elevada, e portanto o atraso médio diminui.



1.c

Antes de analisar qualquer resultado proveniente do simulador, foi feita uma análise prévia. Se a taxa de chegada de pacotes é sempre a mesma, a fila é considerada infinita, e o número total de pacotes também se mantém, é esperado um decréscimo de atraso médio de pacotes com o aumento da capacidade do link. A capacidade da ligação, traduz a velocidade com que o sistema é capaz de atender cada pacote que a ele chega. Se a capacidade for maior, significa que é capaz de processar um maior número de bits/s, do que se a sua capacidade for menor.



Analisando os resultados, estes vão de encontro ao que fora dito anteriormente. Para uma ligação com capacidade C=10 Mbps obteve-se um atraso médio de pacotes igual a aproximadamente 4.25ms. Na segunda simulação, onde C=20 Mbps, o atraso médio, reduz cerca de 10 vezes. Nos seguintes valores de C, o atraso médio reduz para

aproximadamente metade, em relação à ligação anterior. É ainda de notar que o erro, também vai diminuindo ao longo do tempo da mesma maneira que o *Avg. packet delay*.

Deste modo, o atraso médio de pacotes, decresce exponencialmente à medida que a capacidade da ligação aumenta (acontecimento contrário em relação ao exercício a).

```
lambda = 1800;
C=[10 20 30 40];
f=1000000;
P=10000;
N=50; %Run Simulator1 50 times
per1 = zeros(4,N); % 5 arrays de 50 posições
per2 = zeros(4,N); %average packet delay storage
per3 = zeros(4,N);
per4 = zeros(4,N);
for i = 1 : 4
    for it = 1 : N
        [per1(i,it), per2(i,it), per3(i,it), per4(i,it)] = Simulator1(lambda,C(i),f,P);
    end
end
alfa = 0.1;%confidence interval
term1=zeros(1,4);
avgpacketDelay= zeros(1,4); %storage of avg packet delay values
errHigh= zeros(1,4);
errLow = zeros(1,4);
for i = 1 : 4
    avgpacketDelay(i) = mean(per2(i, 1:N));
    term1(i) = norminv(1-alfa/2)*sqrt(var(per2(i, 1:N))/N);
    fprintf('Av. Packet Delay (ms) with C= %d = %.2e +- %.2e\n',C(i), avgpacketDelay(i),term1(i))
    errHigh(i) = term1(i);
    errLow(i) = -(term1(i));
end
```

À semelhança do exercício anterior, este é o código utilizado para a simulação do exercício 1.c. Desta vez é feita a chamada ao *simulator1*, as 50 vezes, para os vários valores de largura de banda, armazenando todos os valores de atraso médio de pacotes. De seguida é feita a sua média, assim como o cálculo do erro.

1.d

Nesta alínea foi novamente pedido o cálculo do atraso médio dos pacotes, no entanto com valores teóricos. Primeiramente foram inicializadas as variáveis necessárias.

```
lambda = 1800; %pps
packetSize = [65:1:109 111:1:1517]; %tamanho que os pacotes podem ter, para alem de 64, 110 e 1518
C = [40000000 30000000 20000000 10000000]; %valores de C
size = length(C); %numero de valores de C
mg1 = zeros(1, size); %matriz para guardar os valores teoricos de acordo com o valor de C
```

De seguida foi criado um ciclo for para percorrer o array de valores de C. Dentro do ciclo são calculados os valores de S para os vários tamanhos de pacotes. São ainda calculados os valores de $E[S]$ e $E[S^2]$ e do atraso médio dos pacotes. Cada valor de atraso médio é guardado no array mg1 na posição correspondente ao seu valor de capacidade do link.

```
for j = 1:size %percorre cada posicao de C

    Sresto2 = 0; %quadrado do remaining (para usar no calculo do ES2)

    S64 = (64*8)/C(j);
    S110 = (110*8)/C(j);
    S1518 = (1518*8)/C(j);
    Sresto = (mean(packetSize)*8)/C(j);
    for i = [65:109 111:1517]
        Sresto2 = Sresto2 + ((i*8)/C(j))^2; %valor total do S do restantes packetSize
    end

    Sresto2 = Sresto2/length(packetSize); %valor medio do S dos restantes packetSize

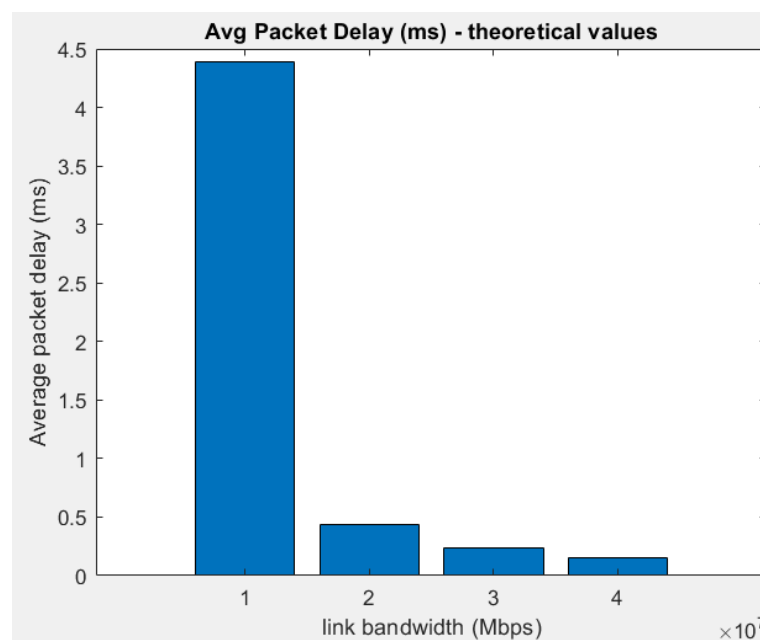
    ES = 0.19 * S64 + 0.23 * S110 + 0.17 * S1518 + (1 - 0.19 - 0.23 - 0.17) * Sresto;
    ES2 = 0.19 * S64^2 + 0.23 * S110^2 + 0.17 * S1518^2 + (1 - 0.19 - 0.23 - 0.17) * Sresto2;
    mg1(j) = (((lambda*ES2) / (2*(1-(lambda*ES)))) + ES)*1000; %APD em ms

end
```

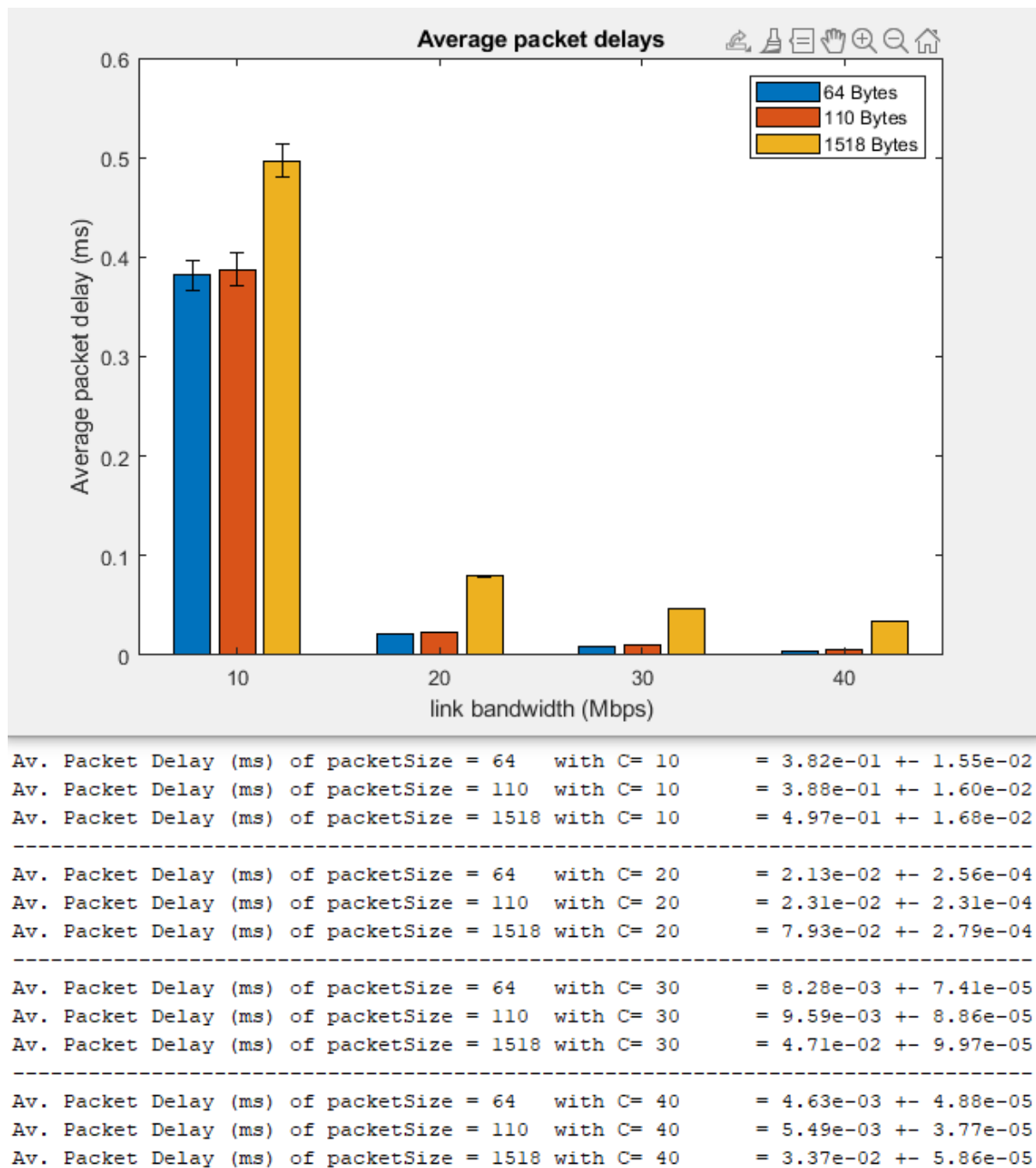
Por fim foi implementado o código para impressão do gráfico correspondente aos valores teóricos do atraso médio dos pacotes.

```
disp(mg1)
%grafico do packet loss
figure(1)
bar(C, mg1)
hold on
title('Avg Packet Delay (ms) - theoretical values');
hold off
```

A evolução do gráfico teórico é bastante similar à evolução do gráfico referente à simulação. Esta evolução caracteriza-se por um decréscimo muito acentuado do atraso médio dos pacotes à medida que a capacidade do link aumenta. À semelhança dos valores apresentados na simulação, se a capacidade do link for mais reduzida ($C=10\text{Mbps}$) o atraso médio dos pacotes é de aproximadamente $4,5\text{ms}$. Assim que a capacidade do link aumenta ($C=20\text{Mbps}$) o atraso médio reduz logo para cerca de $0,5\text{ms}$. Isto acontece pois uma maior capacidade do link é sinónimo de maior velocidade no atendimento dos pacotes, o que implica uma redução no tempo de espera e por consequência uma redução também no atraso médio dos pacotes.



1.e



Visto que, neste exemplo, o objetivo é calcular o atraso médio de diversos pacotes para diferentes valores de C, podemos afirmar que o atraso vai ser menor à medida que a largura de banda do canal aumenta (como se concluiu na alínea c). Confrontando os

valores de atraso médio de pacotes com a alínea c, estes também vão diminuir exponencialmente à medida que a largura de banda do canal aumenta.

Ao efetuar uma análise a cada tamanho de pacote, verifica-se que os pacotes com maior tamanho (número de *Bytes* mais elevado) vão sofrer um maior atraso. Se os pacotes são maiores, o tempo necessário para serem transmitidos, por sua vez, também terá de ser maior comparando com pacotes de tamanho inferior. Como já referido, à medida que a capacidade do canal (*C*) aumenta, o atraso médio de pacotes diminui exponencialmente. Mas ao efetuar uma análise entre os diferentes tamanhos de pacotes (para o mesmo *C*), estes são sempre menores se o tamanho do pacote for o menor. A proporção do atraso entre os vários tamanhos, **não** é sempre proporcional para os vários valores de largura de banda do link. Tendo em conta no caso onde $C=10$, o atraso dos pacotes de 64 e 110 Bytes é cerca de $3/4$ do atraso de pacotes de tamanho 1518 Bytes. Para um $C=20$, o atraso dos dois pacotes de tamanho menor é cerca de $1/4$ do atraso dos pacotes de 1518 Bytes.

É também importante realçar que, não justifica aumentar o tamanho de *C* a partir de um determinado valor, pois a diminuição do valor do atraso médio de pacotes começa a ser insignificante por muito que se aumente a capacidade do link.

```
case DEPARTURE                                % If first event is a DEPARTURE
    TRANSMITTEDBYTES= TRANSMITTEDBYTES + PacketSize;
    %caso o pacote tenha 64 Bytes
    if(packetSIZE == 64)
        delays(1) = delays(1)+(Clock - ArrivalInstant);
        transmittedPackets(1) = transmittedPackets(1)+ 1;
    elseif (packetSIZE == 110)    %caso o pacote tenha 110 Bytes
        delays(2) = delays(2)+(Clock - ArrivalInstant);
        transmittedPackets(2) = transmittedPackets(2)+ 1;
    elseif (packetSIZE == 1518)    %caso o pacote tenha 1518 Bytes
        delays(3) = delays(3)+(Clock - ArrivalInstant);
        transmittedPackets(3) = transmittedPackets(3)+ 1;
    else
        DELAYS= DELAYS + (Clock - ArrivalInstant);
    end
```

Para se obter o atraso médio dos tamanhos de pacotes pedidos, primeiro foi preciso obter o tamanho do pacote que é tratado em cada DEPARTURE. Para tal foi necessário fazer 1º o seguinte: `packetSIZE = EventList(1,3);` %obter o tamanho do pacote

De seguida, se o pacote em causa possuir o tamanho que é pedido, determina-se o delay desse pacote, e incrementa-se o *transmittedPackets* desse tamanho de pacotes, para no fim se conseguir obter o atraso médio.

```
per1 = zeros(4,N); % 4 arrays de N posições
per2 = zeros(4,N); %average packet delay storage
per3 = zeros(4,N);
per4 = zeros(4,N);
per5 = zeros(4,N); %average packet delay for packet size = 64
per6 = zeros(4,N); %average packet delay for packet size = 110
per7 = zeros(4,N); %average packet delay for packet size = 1518
for i = 1 : 4
    for it = 1 : N
        [per1(i,it), per2(i,it), per3(i,it), per4(i,it), per5(i,it), per6(i,it), per7(i,it)] = Simulator1e(lambda,C(i),f,P);
    end
end
alfa = 0.1;
term1=zeros(1,4);
term2=zeros(1,4);
term3=zeros(1,4);
term4=zeros(1,4);
avgpaketDelay= zeros(1,4);
avgpaketDelay64= zeros(1,4);
avgpaketDelay110= zeros(1,4);
avgpaketDelay1518= zeros(1,4);

error= [];
for i = 1 : 4

    avgpaketDelay64(i) = mean(per5(i, 1:N));
    term2(i) = norminv(1-alfa/2)*sqrt(var(per5(i, 1:N))/N);
    fprintf('Av. Packet Delay (ms) of packetSize = 64 with C= %d = %.2e +- %.2e\n',C(i), avgpaketDelay64(i),term2(i))
```

Optou-se por agrupar os valores, para melhor leitura e comparação de valores entre os vários tamanhos de pacotes.

```
y = [];

%determining y values (average packet delay per packet size)
for i =1:4
    y = [y; avgpaketDelay64(i) avgpaketDelay110(i) avgpaketDelay1518(i)];
end
%creating bar graph
b = bar(C,y, 'grouped');
hold on
%determining error
[ngroups,nbars] = size(y);
x = nan(nbars, ngroups);

for i = 1:nbars
    x(i,:) = b(i).XEndPoints;
end

errorbar(x',y,error,'k','linestyle','none');
title('Average packet delays')
ylabel('Average packet delay (ms)')
xlabel('link bandwidth (Mbps)')
legend('64 Bytes', '110 Bytes', '1518 Bytes', Location='northeast')
hold off
```

Tarefa 2

2.a

Na alínea a) da tarefa 2 é usado o `simulator3` para descobrir os valores de simulação correspondentes ao atraso médio de pacotes do tipo data e o atraso médio de pacotes do tipo VoIP. Para isso foram definidos os valores dos parâmetros de entrada da função:

```
C = 10; %link capacity Mbps
f = 1000000; %queue size (bytes)
P = 10000; %stopping criterium
n = [40 30 20 10]; %numero de pacotes VoIP
alfa = 0.1; %confidence intervals
N = 50; %numero de vezes que o simulator3 e chamado
size = length(n); %numero de valores de n
```

De seguida foram definidas as matrizes onde ficam armazenados os valores das diversas simulações:

```
PLmatriz = zeros(size,N); %data
PLVoIPmatriz = zeros(size,N); %voip
APDmatriz = zeros(size,N);
APDVoIPmatriz = zeros(size,N);
MPDmatriz = zeros(size,N);
MPDVoIPmatriz = zeros(size,N);
TTmatriz = zeros(size,N);

APDmean = zeros(1,size);
APDVoIPmean = zeros(1,size);
APDterm = zeros(1,size);
APDVoIPTerm = zeros(1,size);
```

Após isso foi criado um ciclo for que para cada valor de n chama o `simulator3` 50 vezes e guarda os valores da simulação nas respetivas matrizes.

```
]for j = 1:size %para cada valor de n
]   for i = 1:N %chamamos o simulator3 50vezes para cada valor de n
]       [PLmatriz(j,i), PLVoIPmatriz(j,i), APDmatriz(j,i), APDVoIPmatriz(j,i), MPDmatriz(j,i), MPDVoIPmatriz(j,i), TTmatriz(j,i)] = Simulator3(lambda,C,
-   end

APDmean(j) = mean(APDmatriz(j,:)); %media dos valores de cada valor de n (n(0) fica no APDmean(0), n(1) fica no APDmean(1)...
APDVoIPmean(j) = mean(APDVoIPmatriz(j,:));
APDterm(j) = norminv(1-alfa/2)*sqrt(var(APDmatriz(j,:))/N);
APDVoIPTerm(j) = norminv(1-alfa/2)*sqrt(var(APDVoIPmatriz(j,:))/N);

-end
```

Por fim, foi implementado o código para a impressão dos gráficos correspondentes.

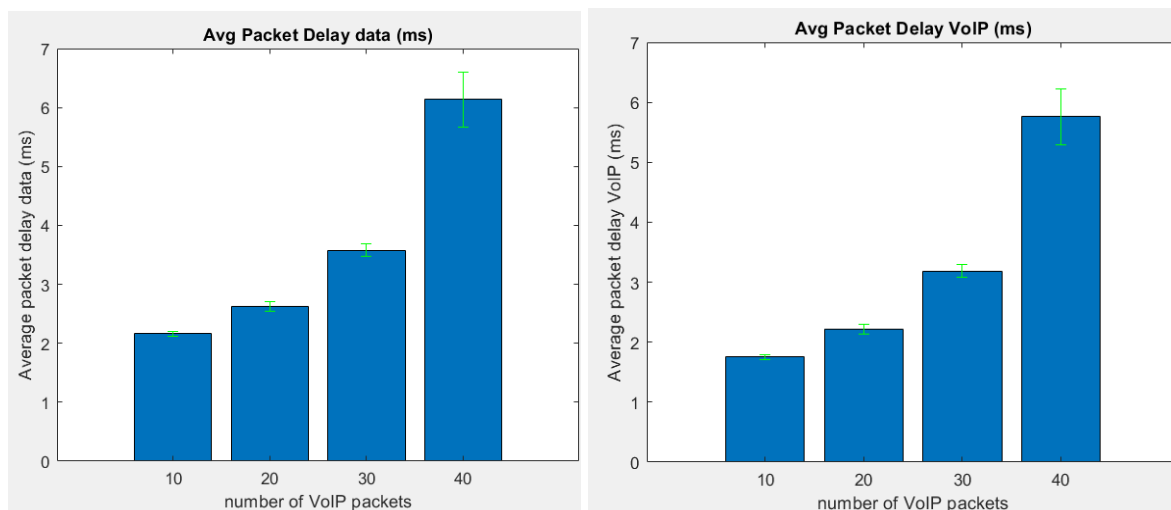
```

figure(1)
bar(n, APDmean)
ylabel('Average packet delay data (ms)')
xlabel('number of VoIP packets')
hold on
er = errorbar(n, APDmean, APDterm);
er.Color = [0 1 0];
er.LineStyle = 'none';
title('Avg Packet Delay data (ms)');
hold off

figure(2)
bar(n, APDVoIPmean)
ylabel('Average packet delay VoIP (ms)')
xlabel('number of VoIP packets')
hold on
er = errorbar(n, APDVoIPmean, APDVoIPterm);
er.Color = [0 1 0];
er.LineStyle = 'none';
title('Avg Packet Delay VoIP (ms)');
hold off

```

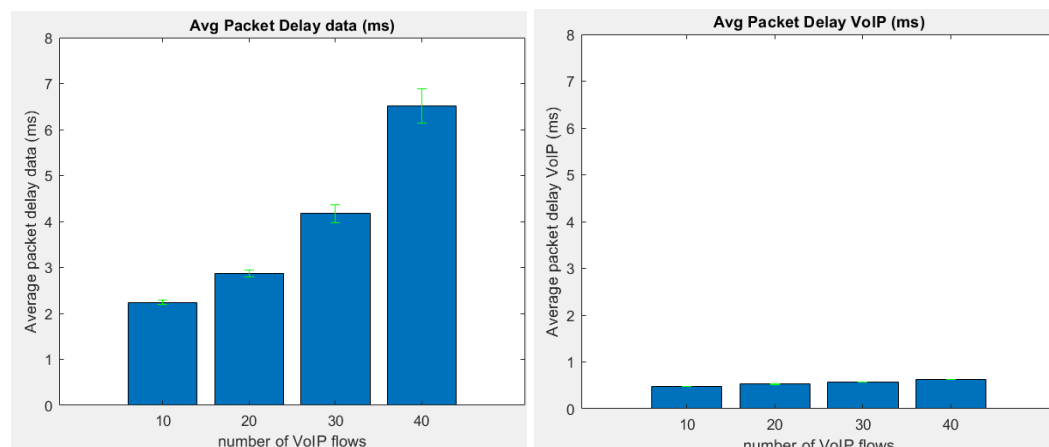
Os gráficos do atraso médio dos pacotes para data e VoIP são bastantes semelhantes. Esta semelhança ocorre tanto nos valores como na sua evolução. É perceptível que o aumento do número de pacotes VoIP provoca o aumento do atraso médio, tanto dos pacotes de data como dos pacotes VoIP. Isto acontece pois quanto mais pacotes são recebidos, mais cheia a fila fica e mais tempo os pacotes demoram a ser atendidos, o que provoca um aumento do seu tempo médio de atraso. Como o sistema em questão não tem prioridades, a chegada de pacotes VoIP afeta de igual forma os pacotes de data e os pacotes VoIP, pois a fila de espera é a mesma para ambos e o atendimento é feito por ordem de chegada (FIFO). Por estes motivos é que o comportamento de ambos os gráficos é bastante semelhante.



2.b

Para a realização da alínea b) o código usado é igual à alínea anterior, com exceção da chamada do `simulator4` em vez do `simulator3`. Quanto aos resultados obtidos, são bastante diferentes. No `simulator4` é dada prioridade aos pacotes VoIP, o que significa que sempre que um pacote VoIP chega à fila pode passar à frente todos os pacotes do tipo data. A consequência disto é que o tempo de atendimento dos pacotes do tipo data vai ser muito superior ao tempo de atendimento dos pacotes VoIP.

Nos gráficos é possível observar que o tempo de atendimento aumenta conforme o número de pacotes VoIP existentes. No entanto, este aumento é muito mais acentuado nos pacotes do tipo data e os valores são muito superiores relativamente aos dos pacotes do tipo VoIP. Esta subida mais acentuada ocorre pois os pacotes do tipo data têm que esperar que os pacotes data à sua frente sejam atendidos e ainda têm que esperar que todos os pacotes do tipo VoIP que vão chegando sejam também atendidos. Já os pacotes do tipo VoIP apenas têm que esperar que os pacotes VoIP à sua frente sejam atendidos visto que passam todos os pacotes do tipo data à frente.



2.c

Neste exercício são pedidos os mesmos parâmetros da alínea anterior, mas neste caso é pedido que sejam calculados os valores teóricos do atraso médio dos pacotes. Para tal começaram por ser inicializadas as matrizes onde ficam guardados os valores teóricos correspondentes a cada valor de n .

```
n = [40 30 20 10];
size = length(n);
WVoIPms = zeros(1,size); %atraso VoIP em ms
WDatams = zeros(1,size); %atraso Data em ms
```

De seguida foi criado um ciclo for que para cada valor de n calcula o atraso médio dos pacotes VoIP e o atraso dos pacotes Data. Os cálculos e fórmulas aplicadas são as demonstradas no slide 19 do módulo 3. Começam por ser calculados os λ de cada fluxo de pacotes, bem como os respetivos μ . Depois são calculados os $E[S]$ e $E[S^2]$, e por fim é determinado o valor dos W , valor esse correspondente ao atraso de cada tipo de pacote.

```
for i = 1:size
    lambdaData = 1500;
    lambdaVoIP = 50*n(i); % 1/0.02
    avgVoIPPacketSize = 120; %media de 110 a 130, todos os tamanhos tem a mesma prob
    avgDataPacketSize = 620; %valor calculado no ex4 do guia01
    miuVoIP = 10e6/(120*8); %pps
    miuData = 10e6/(620*8); %pps
    ESVoIP = 1/miuVoIP; %seg
    ESData = 1/miuData; %seg
    ES2VoIP = 2/(miuVoIP^2); %seg2
    ES2Data = 2/(miuData^2); %seg2
    roVoIP = lambdaVoIP*ESVoIP;
    roData = lambdaData*ESData;
    WVoIP = (((lambdaVoIP*ES2VoIP)+(lambdaData*ES2Data))/(2*(1-roVoIP))) + ESVoIP;
    WData = (((lambdaVoIP*ES2VoIP)+(lambdaData*ES2Data))/(2*(1-roVoIP)*(1-roVoIP-roData))) + ESData;
    WVoIPms(i) = WVoIP*1000;
    WDatams(i) = WData*1000;
end
```

Por fim foi implementado o código que permite imprimir os gráficos pedidos.

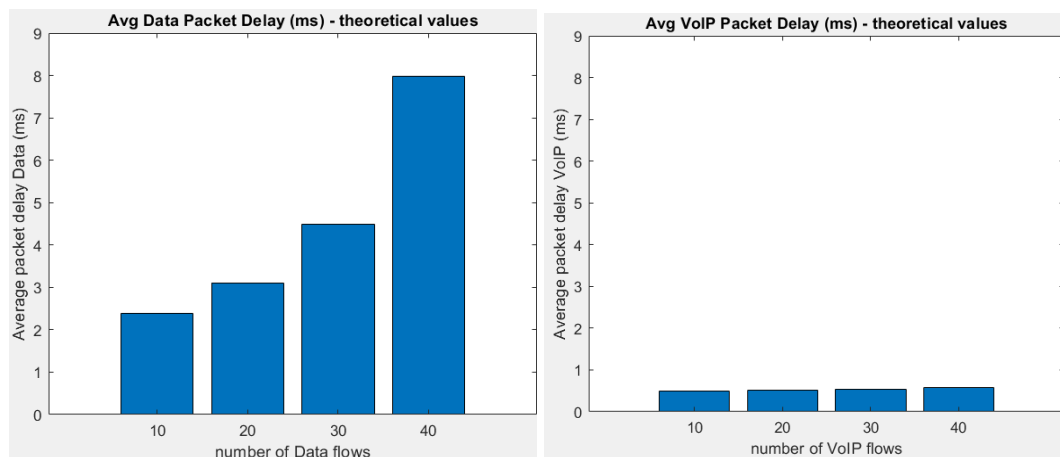
```

%grafico do packet delay VoIP
figure(1)
bar(n, WVoIPms)
ylabel('Average packet delay VoIP (ms)')
xlabel('number of VoIP flows')
ylim([0 8])
hold on
title('Avg VoIP Packet Delay (ms) - theoretical values');
hold off

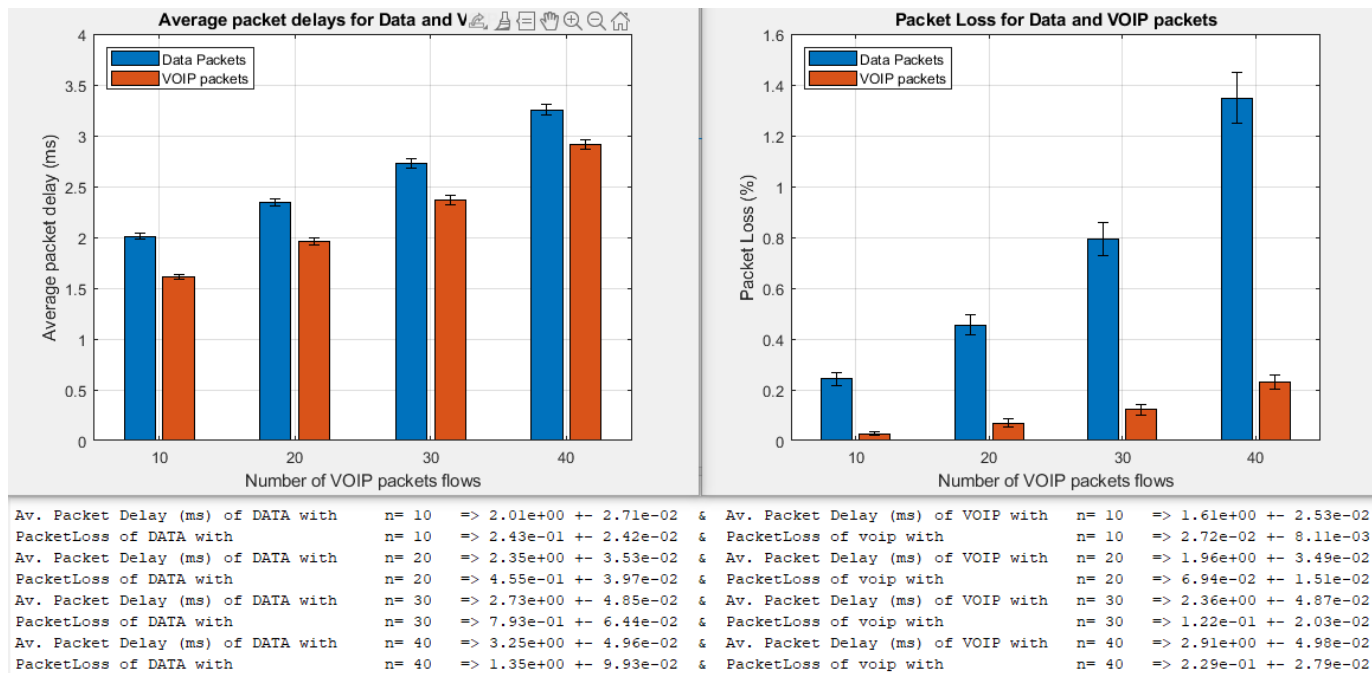
%grafico do packet delay Data
figure(2)
bar(n, WDatams)
ylabel('Average packet delay Data (ms)')
xlabel('number of Data flows')
ylim([0 8])
hold on
title('Avg Data Packet Delay (ms) - theoretical values');
hold off

```

Quanto aos resultados obtidos para os valores teóricos, tal como esperado, têm um comportamento semelhante aos resultados da simulação. Os valores do atraso dos pacotes do tipo VoIP são muito semelhantes aos valores da simulação e a evolução do gráfico é a mesma, pelas razões já referidas na alínea anterior. Quanto aos valores obtidos no gráfico de atraso dos pacotes Data, são também eles muito próximos dos valores da simulação, com exceção da coluna correspondente a 40 fluxos VoIP. Neste caso o valor é consideravelmente superior, passando dos cerca de 7 milissegundos na simulação para os mais de 8 milissegundos no valor teórico.



2.d



Fazendo a análise da imagem da esquerda, pode concluir-se que os pacotes de *voip* sofrem um atraso menor em relação aos pacotes de *dados*. Visto que todas as características do sistema são iguais para os diferentes valores de *n*, o motivo que leva os pacotes de *voip* a sofrer um menor atraso, é o facto do tamanho destes ser menor que o tamanho dos pacotes de *dados*. Como efetuado e analisado nas aulas, os pacotes de dados, têm tamanho médio entre 64 e 1518 Bytes, enquanto que os pacotes de *voip* possuem tamanho uniformemente distribuído entre 110 e 130 Bytes. Como visto na alínea 1.e, pacotes com tamanhos superiores, sofrem atrasos superiores em relação a pacotes de tamanho inferiores. Visto também que existem muitos mais fluxos *voip*, ao ser colocado um pacote na fila do tipo "*dados*", este vai ficar em último, seguido de inúmeros pacotes de *voip*, o que tem consequência no tempo de permanência (do pacote dados) na fila até ser atendido, daí o seu atraso também ser maior. Se o número de fluxos de pacotes *voip* aumenta, o atraso de pacotes do tipo dados e *voip*, também vão aumentar, pois temos cada vez mais pacotes na fila de espera. Assim o tempo que o pacote está na fila vai ser maior, o que tem impacto, quer no pacote de dados, quer no pacote de *voip*, no aumento do atraso médio de pacotes.

Tirando conclusões sobre a perda de pacotes, que acontece na imagem da direita. À medida que se aumenta o número de fluxos de pacotes *voip* (aumento do *n*), o número total de pacotes, na fila, também vai aumentar. Visto que esta tem um tamanho finito, vão existir

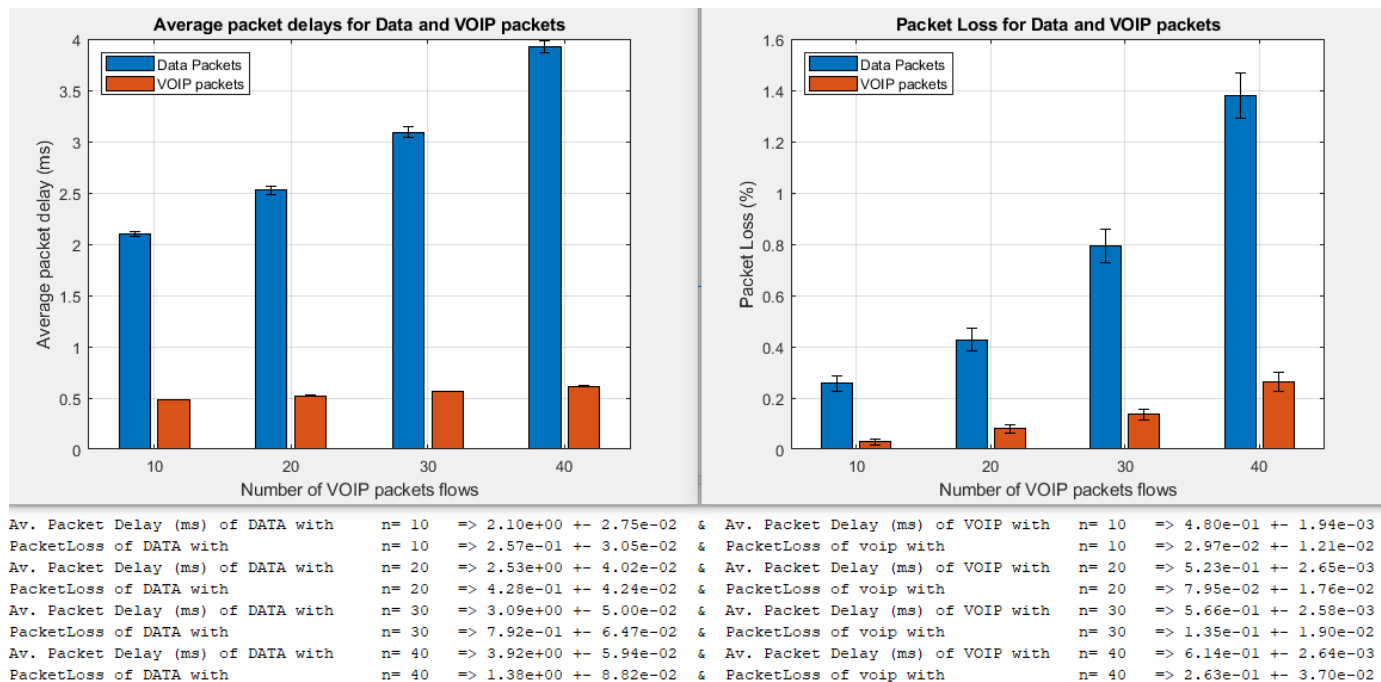
casos em que determinados pacotes já não cabem na fila, daí a taxa de packet loss crescer com o aumento de n . A taxa de packet loss cresce exponencialmente, pois quanto mais fluxos de voip existirem, mais pacotes são inseridos e existem na fila de espera. Existe ainda o caso de a fila de espera estar perto da sua capacidade máxima e não ser possível inserir um pacote de dados na mesma (visto que o tamanho do pacote de dados pode ser bastante maior do que o tamanho do pacote de voip e pode não sobrar espaço na fila para o inserir), inserindo por sua vez um pacote de *voip*, uma vez que o mesmo poderá já caber, incrementando por sua vez, o valor de packet loss para o pacote do tipo de dados.

Em relação a perdas de pacotes, visto que a fila é pequena, vão existir sempre perdas de pacotes. Uma vez que o fluxo de pacotes aumenta, a taxa de perda de pacotes também aumenta. Isto acontece tanto para pacotes de dados como voip.

```
per1 = zeros(4,N);
per2 = zeros(4,N);
per3 = zeros(4,N);
per4 = zeros(4,N);
per5 = zeros(4,N);
per6 = zeros(4,N);
per7 = zeros(4,N);
lambda= 1500;
C=10;
f=10000;
P=10000;
n=[10 20 30 40];
for i =1:4
    for it = 1 : N
        [per1(i,it), per2(i,it), per3(i,it), per4(i,it), per5(i,it), per6(i,it), per7(i,it))] = Simulator3(lambda,C,f,P,n(i));
    end
end
alfa = 0.1;
term1=zeros(1,4);
term2=zeros(1,4);
term3=zeros(1,4);
term4=zeros(1,4);
avgpacketDelayData= zeros(1,4);
avgpacketDelayVoip= zeros(1,4);
avgPacketLossData= zeros(1,4);
avgPacketLossVOIP= zeros(1,4);
y = []; %storing average packet delay values
y1 = []; %storing packet loss values
errorAvgDelay = []; %storing avg packet delay errors
```

Nesta alínea optou-se por expor os valores de atrasos médios de pacotes de pacotes dados e voip, no mesmo gráfico para se conseguir uma melhor observação e comparação dos valores. O mesmo acontece para o gráfico do packet loss

2.e



Neste exercício irá ser feita primeiramente uma análise dos valores obtidos e só depois é que se fará uma comparação com a alínea 2d.

A diferença deste simulador, é que o mesmo possui uma fila com prioridades, ou seja, cada vez que existirem pacotes voip na fila, estes são sempre atendidos em primeiro lugar. Para a imagem da esquerda, não interessa quantos fluxos de pacotes voip existem, pois os pacotes voip são sempre atendidos primeiro que os pacotes de dados. Com isto o atraso médio de pacotes do tipo voip é quase constante. Não o é pelo facto de estarem a chegar cada vez mais pacotes (à medida que o n aumenta) e com isto a fila de espera aumenta de tamanho, o que vai ter consequência no tempo de espera na fila, aumentando assim ligeiramente o atraso médio de pacotes. Como é de esperar, o atraso médio de pacotes para os dados, vai ser elevado, pois sempre que existirem pacotes na fila do tipo dados e voip, a mesma irá atender em primeiro lugar os voip. Por estas razões, um pacote de dados passa uma quantidade de tempo na fila de espera, muito superior em relação a pacotes de voip(atraso médio de dados aumenta). Com o aumento de fluxos voip, tem-se mais pacotes voip na fila, o que implica que necessite de mais tempo para processar esses pacotes, e só depois atender os pacotes de dados. Deste modo o atraso médio de pacotes de dados cresce à medida que o número de fluxos voip também aumenta.

Quanto a perdas de pacotes, vão sempre existir perdas, visto que a fila é pequena. Esta taxa, aumenta tanto nos pacotes de dados como nos pacotes de voip, uma vez que temos mais pacotes à medida que o n aumenta. A fila vai estar cada vez mais cheia para o aumento de fluxo de voip, o que faz com que sejam descartados mais pacotes com este aumento. A taxa de perda de pacotes, vai ser igual à alínea anterior, pois só existe perda de pacotes, quando a fila não tem espaço para os receber. Visto que a inserção de pacotes na fila é igual tanto no *simulator3* como no *simulator4*, a prioridade dos pacotes voip sobre os dados, não influencia em nada a perda de pacotes. Influencia sim a ordem pelos quais são atendidos.

Fazendo a comparação de atraso médio de pacotes para com a alínea 2d. Como a fila do *simulator4* dá prioridade aos pacotes voip, estes têm menos atraso em relação aos voip do *simulator3*. Com isto o aumento do atraso dos pacotes de dados vai aumentar na fila com prioridades, pelo facto de estes pacotes permanecerem uma maior quantidade de tempo na fila.

Quanto a intervalos de erro, quando estes forem menores, pode afirmar-se que temos bons e viáveis resultados de simulação, o que acontece nos casos de atraso médio de pacotes. Quanto ao caso de *packet loss*, os intervalos de confiança já são relativamente maiores, o que indica que os resultados são menos precisos.

O código matlab para este exercício é semelhante ao do exercício anterior, só que desta vez é chamado o *simulator4* para obtenção dos valores

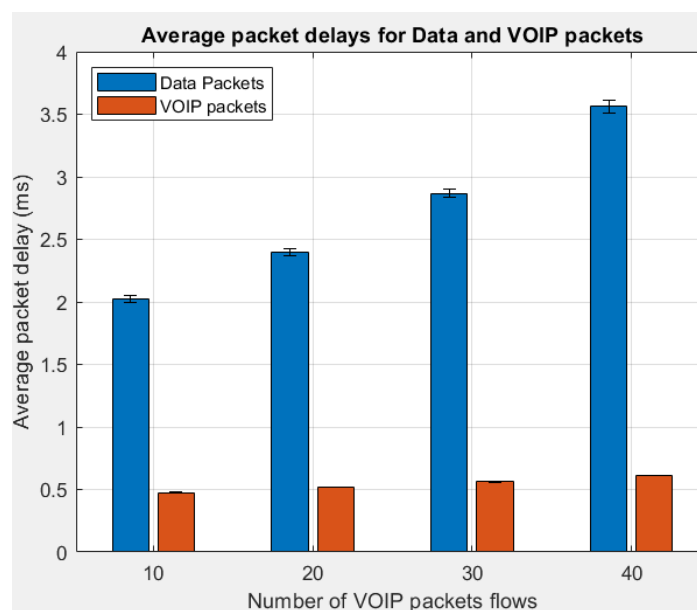
2.f

O código usado para este exercício é o mesmo que o anterior, mas em vez de ser chamado o simulador4 é chamada a nova versão, simulador 4f. A diferença da nova versão do simulador para a anterior é que os pacotes data apenas são aceites na fila, se a capacidade desta não exceder os 90%. Para tal, na condição de aceitação dos pacotes data foi feita a seguinte alteração:

```
if QUEUEOCCUPATION + PacketSize <= 0.9*f %se a queue mais o tamanho do pacote for menor que 90% da capacidade o pacote é aceite
    QUEUE= [QUEUE;PacketSize , Clock, DADOS];
    QUEUEOCCUPATION= QUEUEOCCUPATION + PacketSize;
else
    LOSTPACKETS= LOSTPACKETS + 1; %caso contrario o pacote e descartado
end
```

No que toca aos resultados obtidos, são visíveis algumas diferenças em relação à alínea anterior. Relativamente ao atraso médio dos pacotes, o comportamento do gráfico é bastante semelhante, com o atraso dos pacotes do tipo data a crescer consoante o aumento do número de pacotes VoIP. Já os pacotes VoIP continuam a ter uma variação muito pequena dos seus valores de atraso.

Contudo, apesar de a evolução do gráfico ser bastante semelhante à da alínea anterior, os valores de atraso dos pacotes do tipo data são menores. Isto acontece pois a limitação dos 90% de capacidade provoca que menos pacotes sejam atendidos, o que aumenta a velocidade de atendimento e reduz o atraso médio dos pacotes do tipo data. Os pacotes VoIP mantêm os mesmos valores pois a redução do número de pacotes do tipo data não afeta o seu tempo de espera, visto que têm prioridade e são sempre atendidos primeiro.



Quanto aos valores relativos à percentagem de perda de pacotes sofrem também algumas alterações. A percentagem de pacotes data que são perdidos aumenta ligeiramente, devido à limitação dos 90% de capacidade. Isto significa que sensivelmente 10% mais pacotes deste tipo serão perdidos, o que se reflete num aumento dos valores em relação à alínea anterior. Já os valores de perda dos pacotes do tipo VoIP sofrem uma redução ainda maior. Isto acontece pois a limitação de entrada de pacotes data na fila de espera deixa mais espaço para os pacotes VoIP. Isso significa que estes têm uma maior probabilidade de serem aceites na fila, reduzindo assim o número de pacotes VoIP que são perdidos, chegando mesmo a zero em alguns casos.

