universidade
de aveiro

Relatório do Guião nº 3 de Desempenho e Dimensionamento de Redes

Task 3

3.a. Consider the connection characterized by C = 10 Mbps, f = 10^7 Bytes (~10 MBytes) and BER b = 0. Run Simulator2 10 times with a stopping criterion of P = 10000 at each run to compute the estimated values and the 90% confidence intervals of all performance parameters when = 1500, 1600, 1700, 1800, 1900 and 2000 pps. Present the results and the confidence intervals in bar charts with error bars4.

3.b. Repeat the previous experiment 3.a now running the simulator 40 times for each case. Again, present the results and the confidence intervals in bar charts with error bars. Justify the obtained results for the different performance parameters. Take also conclusions of the impact of the number of simulation runs on the obtained confidence intervals of the different cases.

```
clear all

P = 10000; % Stopping criterion
lambda = 1500:100:2000; % Packet rate (packets/sec)
alfa = 0.1; % Confidence intervals
N = 10; % number of simulations
C = 10; % link bandwidth (Mbps)
f = 10000000; % queue size (Bytes)
b = 0; % bit error rate

for it = 1:6 % iterate through lambda values
   for i= 1:N
      [PL(i)  APD(i)  MPD(i)  TT(i)] = simulator2(lambda(it),C,f,P,b);
   end
      % average of the values obtained
      media_PL(it) = mean(PL);
      media_APD(it) = mean(APD);
      media_MPD(it) = mean(MPD);
      media_TT(it) = mean(TT);
      % error
      term_PL(it) = norminv(1-alfa/2)*sqrt(var(PL)/N);
      term_APD(it) = norminv(1-alfa/2)*sqrt(var(APD)/N);
      term_MPD(it) = norminv(1-alfa/2)*sqrt(var(MPD)/N);
      term_TT(it) = norminv(1-alfa/2)*sqrt(var(TT)/N);
end

% APD
```

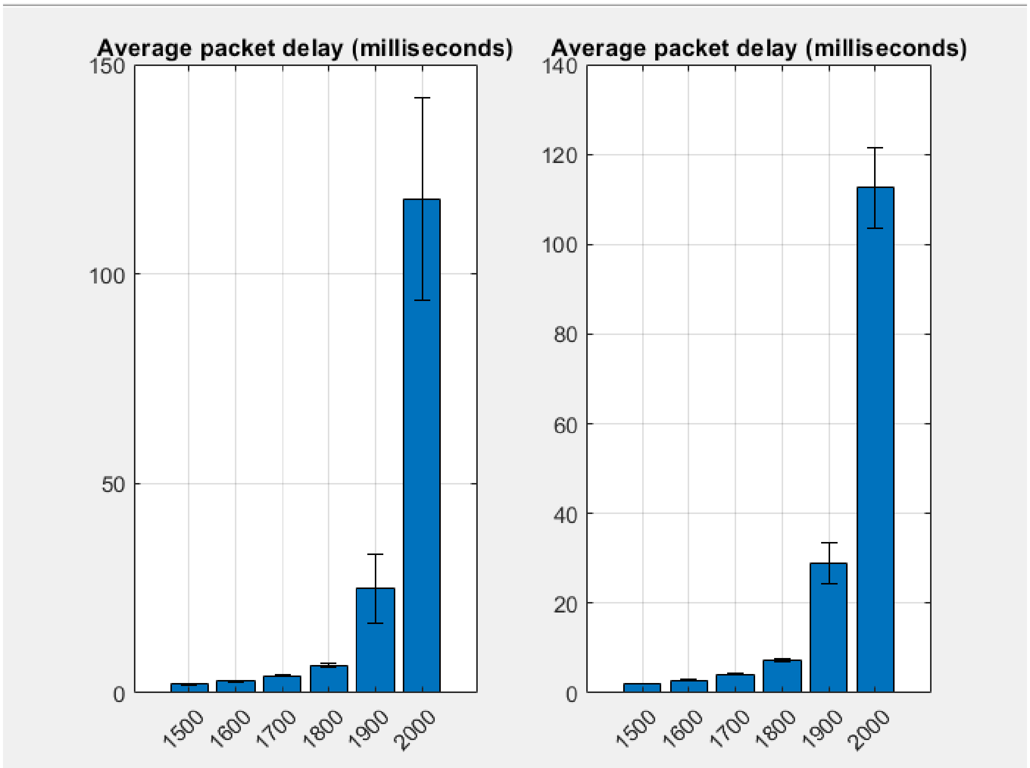
```
figure(2)
subplot(1,2,1)
bar(lambda, media_APD)
title('Average packet delay (milliseconds)')
grid on
hold on

% Error bar
er = errorbar(lambda, media_APD, term_APD, term_APD);
er.Color = [0 0 0];
er.LineStyle = 'none';
hold off

% MPD
figure(3)
subplot(1,2,1)
bar(lambda, media_MPD)
title('Maximum packet delay (milliseconds)')
grid on
hold on

% Error bar
er = errorbar(lambda, media_MPD, term_MPD, term_MPD);
er.Color = [0 0 0];
er.LineStyle = 'none';
hold off

% TT
figure(4)
subplot(1,2,1)
bar(lambda, media_TT)
title('Transmitted throughput (Mbps)')
ylim([0 11])
grid on
hold on

% Error bar
er = errorbar(lambda, media_TT, term_TT, term_TT);
er.Color = [0 0 0];
er.LineStyle = 'none';
hold off

clear all

P = 10000; % Stopping criterion
lambda = 1500:100:2000; % Packet rate (packets/sec)
alfa = 0.1; % Confidence intervals
N = 40; % number of simulations
C = 10; % link bandwidth (Mbps)
f = 10000000; % queue size (Bytes)
b = 0; % bit error rate
```

```
for it = 1:6 % iterate through lambda values
    for i= 1:N
        [PL(i)  APD(i)  MPD(i)  TT(i)] = simulator2(lambda(it),C,f,P,b);
    end
        % average of the values obtained
        media_PL(it) = mean(PL);
        media_APD(it) = mean(APD);
        media_MPD(it) = mean(MPD);
        media_TT(it) = mean(TT);

        term_PL(it) = norminv(1-alfa/2)*sqrt(var(PL)/N);
        term_APD(it) = norminv(1-alfa/2)*sqrt(var(APD)/N);
        term_MPD(it) = norminv(1-alfa/2)*sqrt(var(MPD)/N);
        term_TT(it) = norminv(1-alfa/2)*sqrt(var(TT)/N);
end

% APD
figure(2)

subplot(1,2,2)
bar(lambda, media_APD)
title('Average packet delay (milliseconds)')
grid on
hold on

% Error bar
er = errorbar(lambda, media_APD, term_APD, term_APD);
er.Color = [0 0 0];
er.LineStyle = 'none';
hold off

% MPD
figure(3)
subplot(1,2,2)
bar(lambda, media_MPD)
title('Maximum packet delay (milliseconds)')
grid on
hold on

% Error bar
er = errorbar(lambda, media_MPD, term_MPD, term_MPD);
er.Color = [0 0 0];
er.LineStyle = 'none';
hold off

% TT
figure(4)
subplot(1,2,2)
bar(lambda, media_TT)
title('Transmitted throughput (Mbps)')
ylim([0 11])
grid on
hold on
```

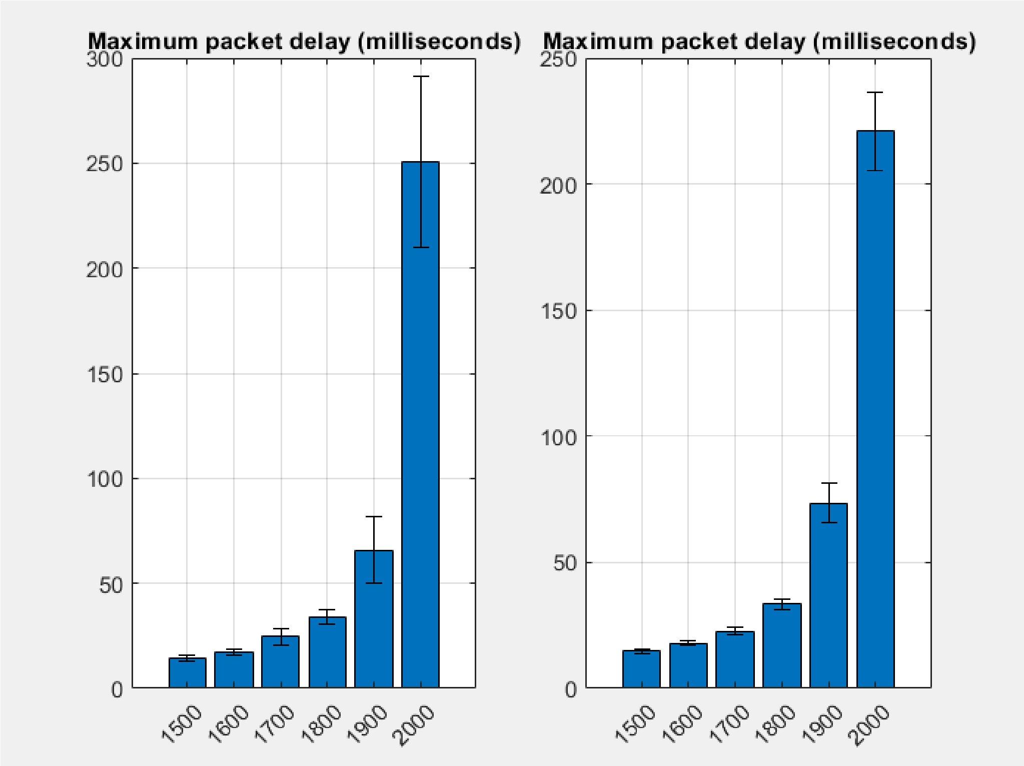
```
% Error bar
er = errorbar(lambda, media_TT, term_TT, term_TT);
er.Color = [0 0 0];
er.LineStyle = 'none';
hold off
```

Para a resolução destas 2 alíneas começamos por realizar a simulação 10 vezes com os valores dados no enunciado e representar os valores no gráfico da esquerda, depois realizamos a mesma simulação 40 vezes e representamos os gráficos do lado direito para tornar mais fácil a comparação das 2 alíneas.

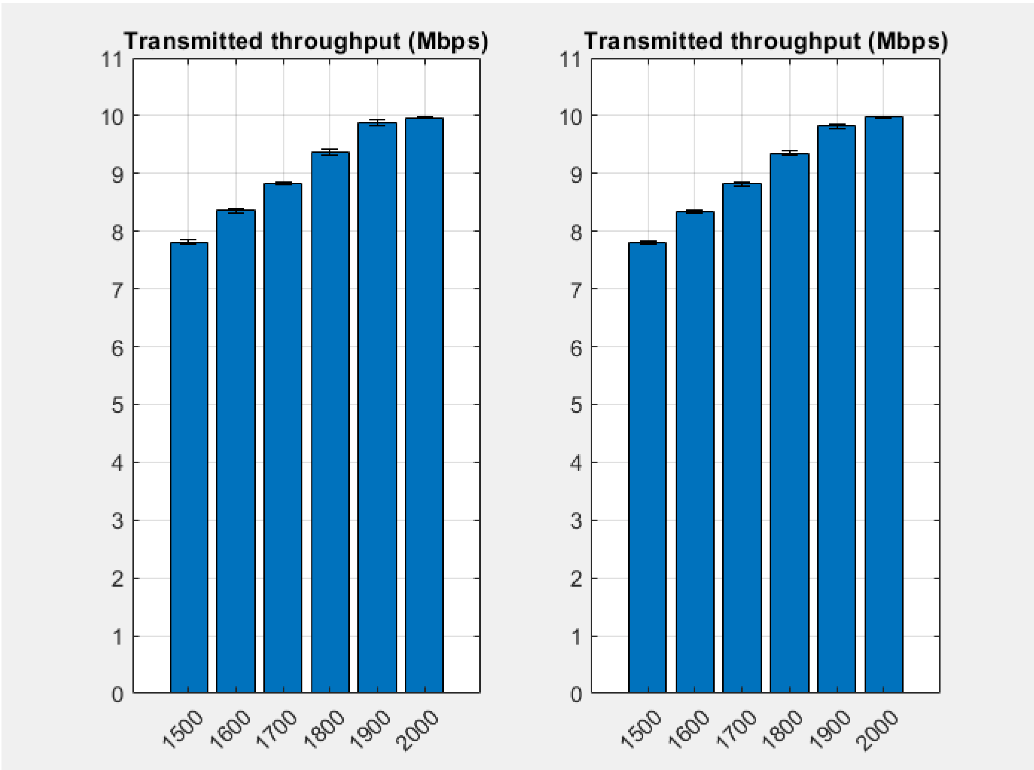
Obtivemos então os seguintes gráficos:



Alínea a) Alínea b)



Alínea a) Alínea b)



Alínea a) Alínea b)

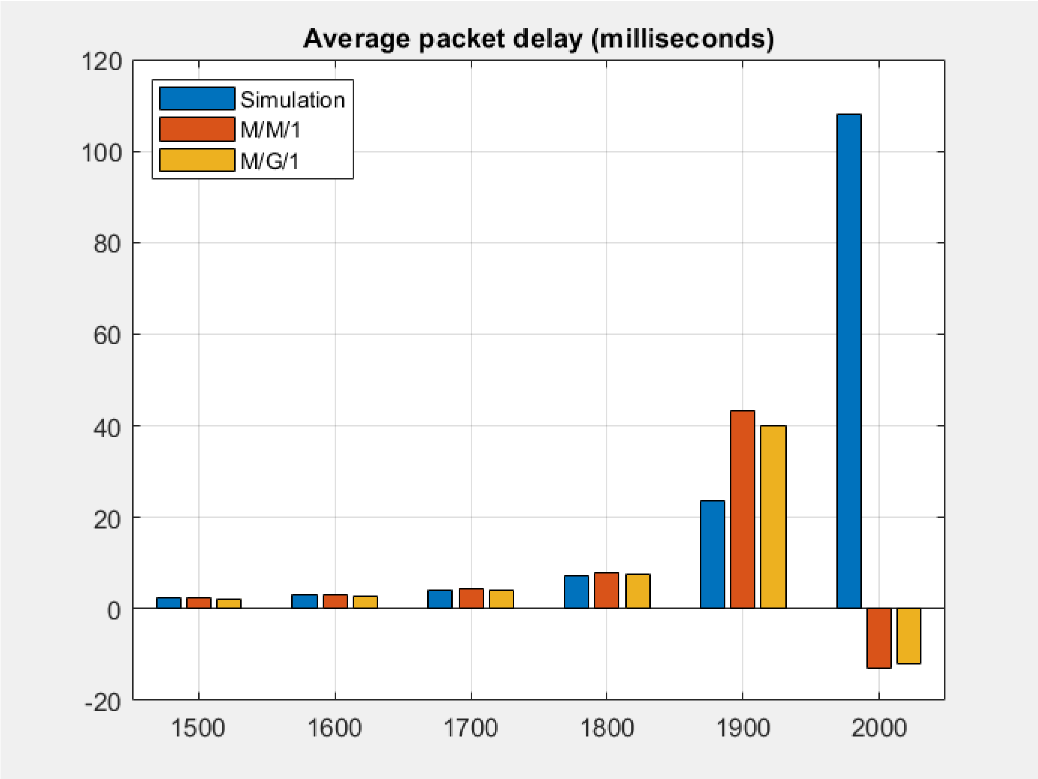
Perante estes gráficos podemos concluir que os resultados são bastante semelhantes, tanto na alínea (a) (10 simulações) como na alínea (b) (40 simulações). Salientamos que embora os resultados sejam quase iguais, é evidente uma diferença bastante grande no erro.
Esta diferença era expectável visto que quanto mais simulações se fazem, mais precisos são os resultados e portanto menor será o erro.

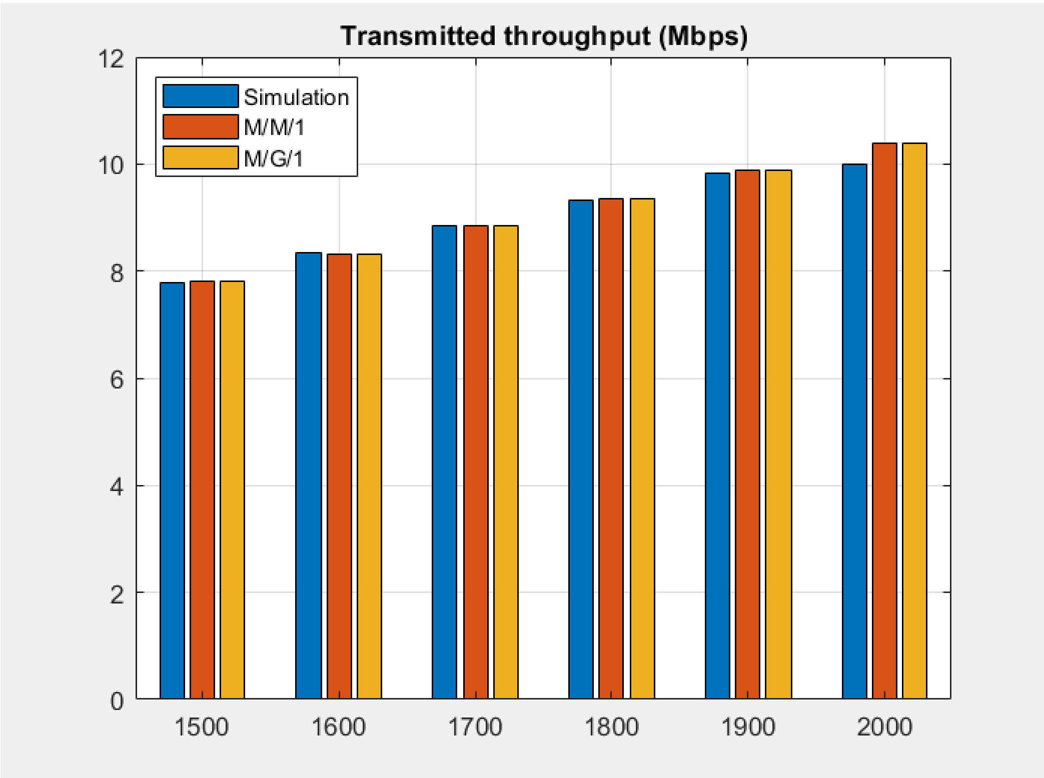
3.c. For each of the previous cases, compute the average packet delay and the total throughput of the connection considering that it is modelled by a M/M/1 system and by a M/G/1 system. Present both theoretical values and the simulation results of 3.b (without confidence intervals) in a bar chart for the average packet delay and another bar chart for the total throughput. Take conclusions on how close each theoretical model is from the simulation results of the different cases.

```
clear all

P = 10000; % Stopping criterion
lambda = 1500:100:2000; % Packet rate (packets/sec)
alfa = 0.1; % Confidence intervals
N = 40; % number of simulations
C = 10; % link bandwidth (Mbps)
f = 10000000; % queue size (Bytes)
b = 0; % bit error rate

for it = 1:6
   for i= 1:N
      [PL(i)  APD(i)  MPD(i)  TT(i)] = simulator2(lambda(it),C,f,P,b);
   end
      media_APD(it) = mean(APD);
      media_TT(it) = mean(TT);
end

media = mean([65:109, 111:1517]);
C = 10e6;      % link bandwidth (bytes)

% Average packet size (bits)
B = 0.16 * 64 + 0.25 * 110 + 0.2 * 1518 + 0.39 * media;
B = B * 8;
miu = (C / B);
a = [65:109, 111:1517];
p_each = (1 - (0.16 + 0.25 + 0.2)) / length(a);
E_S = (B / C);
E_S_2 = 0.16 * (64 * 8 / C)^2 + 0.25 * (110 * 8 / C)^2 + 0.2 * (1518 * 8 / C)^2;
```

```
for i = 1:length(a)
    E_S_2 = E_S_2 + p_each * (a(i) * 8 / C)^2;
end


for it = 1:6
    % M/M/1
    W1(it) = 1 / (miu - lambda(it)) * 1000;
    TT1(it) = lambda(it) * B / 1e6;
    % M/G/1
    W2(it) = ((lambda(it) * E_S_2) / (2 * (1 - lambda(it) * E_S)) + E_S) * 1000;
    TT2(it) = lambda(it) * B / 1e6;
end

% PL
figure(1)
bar(lambda, [media_APD; W1; W2])
title('Average packet delay (milliseconds)')
legend('Simulation', 'M/M/1', 'M/G/1', 'location', 'northwest')
ylim([-20 120])
grid on

% APL
figure(2)
bar(lambda, [media_TT; TT1; TT2])
title('Transmitted throughput (Mbps)')
legend('Simulation', 'M/M/1', 'M/G/1', 'location', 'northwest')
grid on
```

Obtivemos então os seguintes gráficos:





Ao analisarmos os gráficos, podemos observar valores de *Transmitted Throughput* muito próximos da simulação. Quanto aos valores de *Average Packet Delay*, verificamos uma subida repentina quando os valores de lambda se aproximam de 2000, pois o tamanho da fila de espera é aproximadamente 1920. No caso dos módulos M/M/1 e M/G/1, o atraso até dá negativo para valores de lambda iguais a 2000 pps, pois estes módulos não estão preparados para casos em que o número de pacotes por segundo é superior ao tamanho da fila de espera. Como podemos ver pela fórmula abaixo:

$$W_Q = W - \frac{1}{\mu} = \frac{\lambda}{\mu(\mu - \lambda)}$$

Fórmula para o atraso médio na fila de espera no sistema M/M/1

Como podemos ver, no denominador, a subtração de μ por λ dará um valor negativo.

3.d. Consider the packet rate = 1800 pps and the connection characterized by C = 10 Mbps and BER b = 0. Run Simulator2 40 times with a stopping criterion of P = 10000 at each run to compute the estimated values and the 90% confidence intervals of all performance parameters when f = 2500, 5000, 7500, 10000, 12500, 15000, 17500 and 20000 Bytes. Present the results and the confidence intervals in bar charts with error bars. Justify the obtained results.

3.f. Repeat the previous experiment 3.d considering now that the connection has a BER of b = 10^-5 . Present the results and the confidence intervals in bar charts with error bars. Justify the obtained results comparing them with the simulation results of experiment 3.d.

```
clear all

C = 10; % link bandwidth (Mbps)
f = 2500:2500:20000; % queue size (Bytes)
b = 0; % bit error rate
P = 10000; % Stopping criterion
alfa = 0.1;    % Confidence intervals
lambda = 1800;
N = 40;

for it = 1:8
   for i= 1:N
      [PL(i)  APD(i)  MPD(i)  TT(i)] = simulator2(lambda, C, f(it), P, b);
   end
      media_PL(it) = mean(PL);
      media_APD(it) = mean(APD);
      media_MPD(it) = mean(MPD);
      media_TT(it) = mean(TT);
```

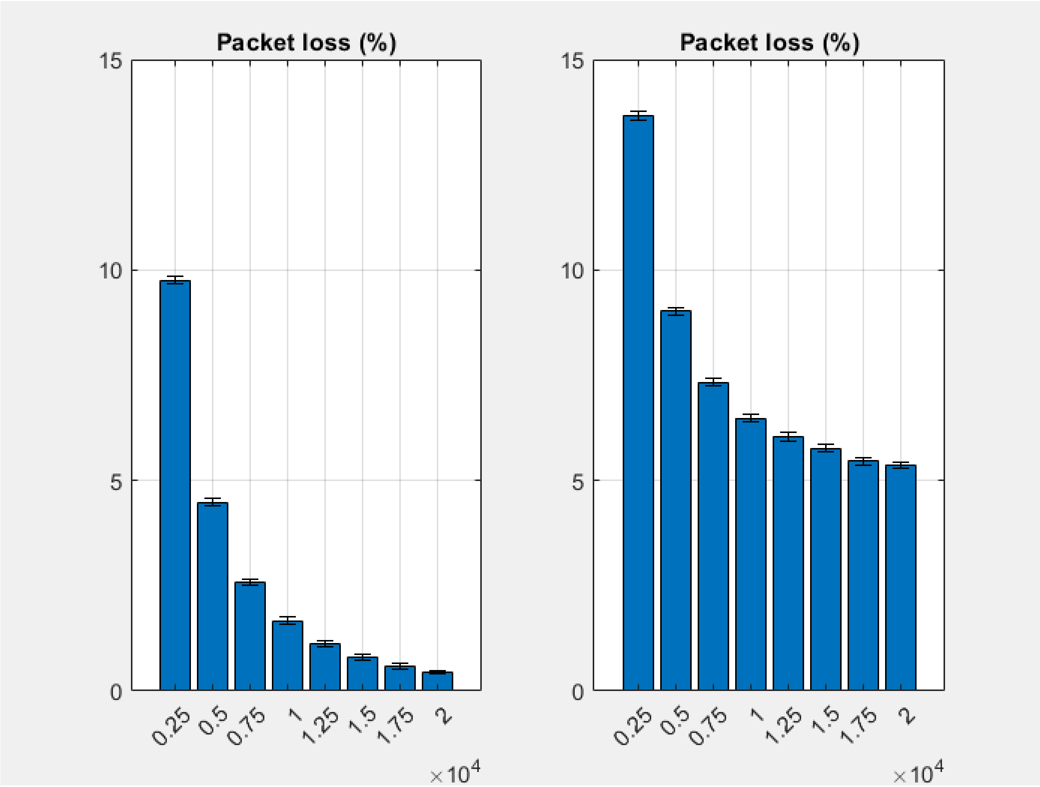
```
	term_PL(it) = norminv(1-alfa/2)*sqrt(var(PL)/N);
	term_APD(it) = norminv(1-alfa/2)*sqrt(var(APD)/N);
	term_MPD(it) = norminv(1-alfa/2)*sqrt(var(MPD)/N);
	term_TT(it) = norminv(1-alfa/2)*sqrt(var(TT)/N);
end


% PL
figure(1)
subplot(1,2,1)
bar(f, media_PL)
title('Packet loss (%)')
ylim([0 100])
grid on
hold on

% Error bar
er = errorbar(f, media_PL, term_PL, term_PL);
er.Color = [0 0 0];
er.LineStyle = 'none';
hold off

% APD
figure(2)
subplot(1,2,1)
bar(f, media_APD)
title('Average packet delay (milliseconds)')
grid on
hold on

% Error bar
er = errorbar(f, media_APD, term_APD, term_APD);
er.Color = [0 0 0];
er.LineStyle = 'none';
hold off

% MPD
figure(3)
subplot(1,2,1)
bar(f, media_MPD)
title('Maximum packet delay (milliseconds)')
grid on
hold on

% Error bar
er = errorbar(f, media_MPD, term_MPD, term_MPD);
er.Color = [0 0 0];
er.LineStyle = 'none';
hold off

% TT
figure(4)
subplot(1,2,1)
```

```
bar(f, media_TT)
title('Transmitted throughput (Mbps)')
ylim([6 10])
grid on
hold on

% Error bar
er = errorbar(f, media_TT, term_TT, term_TT);
er.Color = [0 0 0];
er.LineStyle = 'none';
hold off

clear all

C = 10;
f = 2500:2500:20000; % queue size (Bytes)
b = 1e-5; % bit error rate
P = 10000; % Stopping criterion
alfa = 0.1;     % 90% confidence interval
lambda = 1800;
N = 40;

for it = 1:8
   for i= 1:N
      [PL(i)  APD(i)  MPD(i)  TT(i)] = simulator2(lambda, C, f(it), P, b);
   end
      media_PL(it) = mean(PL);
      media_APD(it) = mean(APD);
      media_MPD(it) = mean(MPD);
      media_TT(it) = mean(TT);

      term_PL(it) = norminv(1-alfa/2)*sqrt(var(PL)/N);
      term_APD(it) = norminv(1-alfa/2)*sqrt(var(APD)/N);
      term_MPD(it) = norminv(1-alfa/2)*sqrt(var(MPD)/N);
      term_TT(it) = norminv(1-alfa/2)*sqrt(var(TT)/N);
end


% PL
figure(1)
subplot(1,2,2)
bar(f, media_PL)
title('Packet loss (%)')
ylim([0 100])
grid on
hold on

% Error bar
er = errorbar(f, media_PL, term_PL, term_PL);
er.Color = [0 0 0];
er.LineStyle = 'none';
hold off
```

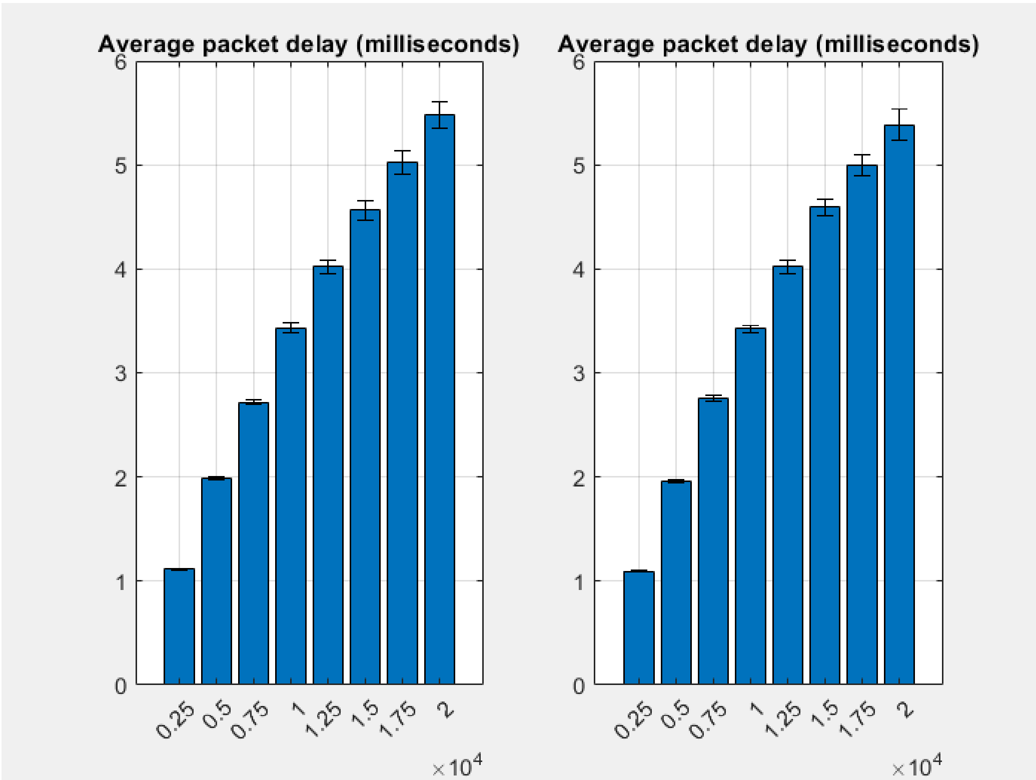
```
% APD
figure(2)
subplot(1,2,2)
bar(f, media_APD)
title('Average packet delay (milliseconds)')
grid on
hold on

% Error bar
er = errorbar(f, media_APD, term_APD, term_APD);
er.Color = [0 0 0];
er.LineStyle = 'none';
hold off

% MPD
figure(3)
subplot(1,2,2)
bar(f, media_MPD)
title('Maximum packet delay (milliseconds)')
grid on
hold on

% Error bar
er = errorbar(f, media_MPD, term_MPD, term_MPD);
er.Color = [0 0 0];
er.LineStyle = 'none';
hold off

% TT
figure(4)
subplot(1,2,2)
bar(f, media_TT)
title('Transmitted throughput (Mbps)')
ylim([6 10])
grid on
hold on

% Error bar
er = errorbar(f, media_TT, term_TT, term_TT);
er.Color = [0 0 0];
er.LineStyle = 'none';
hold off
```

Para resolver estas 2 alíneas, começamos por fazer a simulação 40 vezes com os parâmetros pedidos na alínea d) e colocamos os gráficos do lado esquerdo, depois realizamos a simulação mais 40 vezes, com o ber igual a 10^-5 como pedido na alínea f).

Obtivemos então os seguintes gráficos:

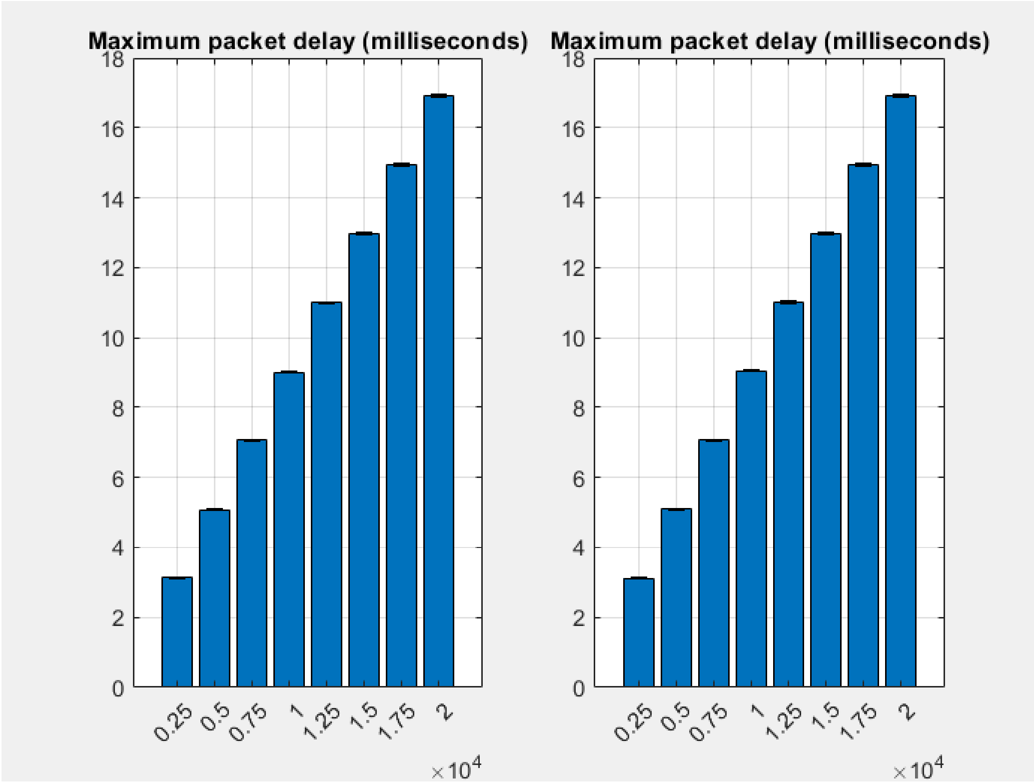


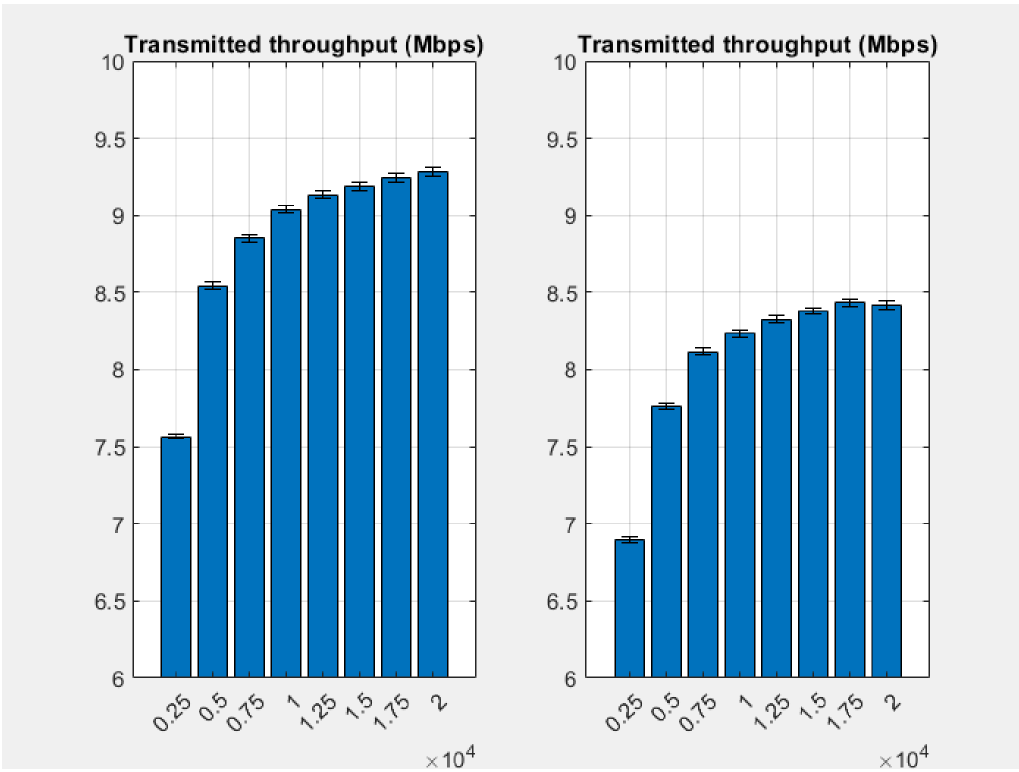
Packet loss (%)
15
10
5
0
0.25
0.5
0.75
1
1.25
1.5
1.75
2
×10^4
Packet loss (%)
15
10
5
0
0.25
0.5
0.75
1
1.25
1.5
1.75
2
×10^4



Average packet delay (milliseconds)
6
5
4
3
2
1
0
0.25
0.5
0.75
1
1.25
1.5
1.75
2
×10^4
Average packet delay (milliseconds)
6
5
4
3
2
1
0
0.25
0.5
0.75
1
1.25
1.5
1.75
2
×10^4

universidade
de aveiro





Perante estes resultados podemos observar que a percentagem de pacotes que são perdidos (*packet loss*) é superior com um bit error rate superior, isto é o esperado visto que, se o número de *bit errors* é superior, o número de pacotes perdidos também será maior. Podemos concluir também que o *transmitted throughput* é

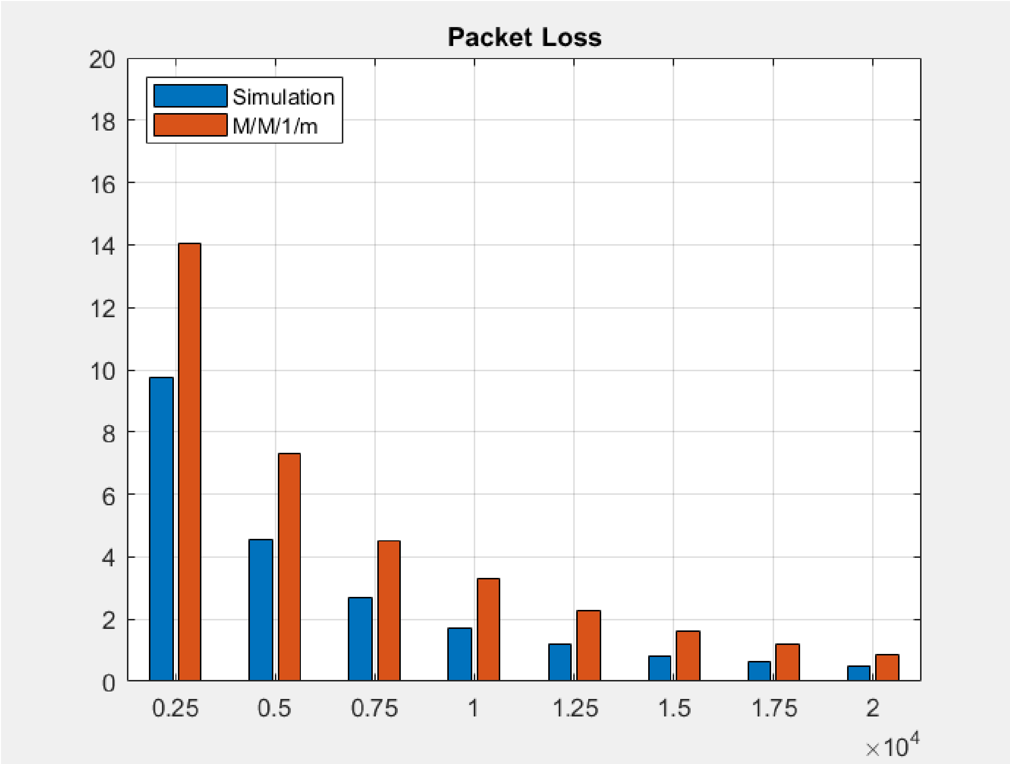
menor com um BER superior, pois é inversamente proporcional ao *packet loss,* dado que se há maior percentagem de um pacote ser perdido, serão transmitidos menos pacotes.
Quanto ao *average packet delay* e *maximum packet delay,* os resultados são independentes do BER pois os pacotes em fila de espera para serem processados são os mesmos visto que o BER só influencia após os pacotes saírem da fila de espera.

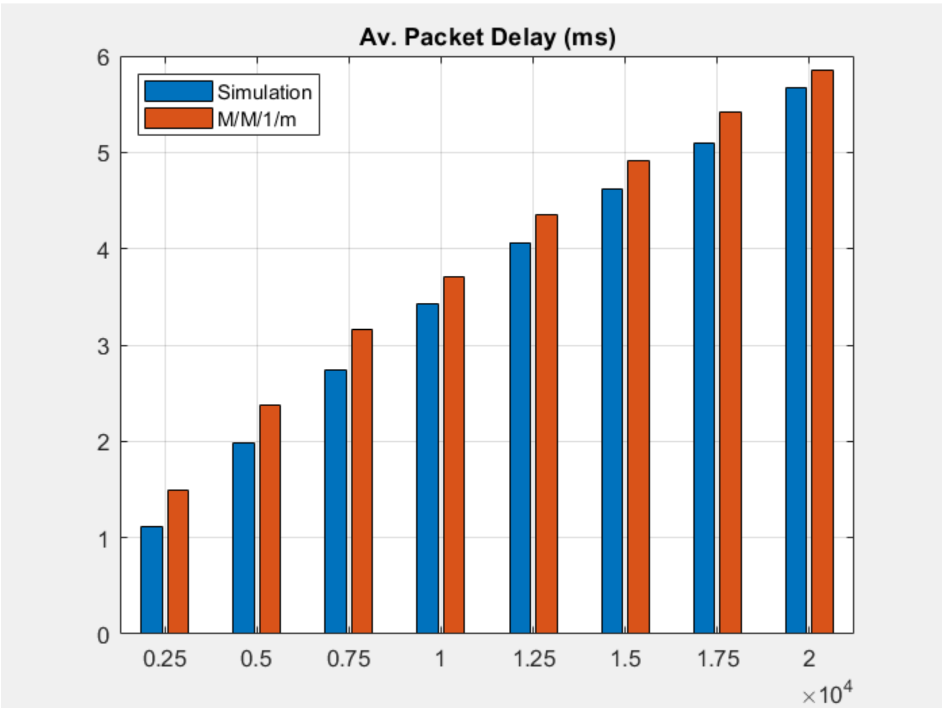
3.e. For each of the previous cases of experiment 3.d, compute the packet loss, the average packet delay and the total throughput of the connection considering that it is modelled by a M/M/1/m. Present both theoretical values and the simulation results of 3.d (without confidence intervals) in a bar chart for the packet loss, another bar chart for the average packet delay and a third bar chart for the total throughput. Take conclusions on how close the theoretical model is from the simulation results of the different cases.

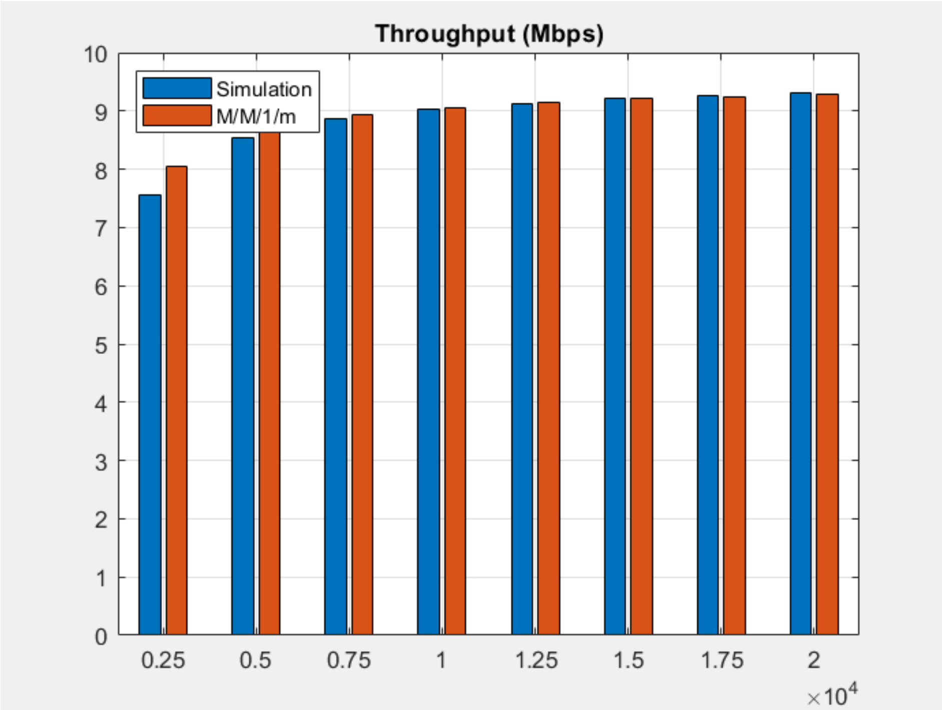
```
clear all

C = 10; % link bandwidth (Mbps)
f = 2500:2500:20000; % queue size (Bytes)
b = 0; % bit error rate
P = 10000; % Stopping criterion
alfa = 0.1;     % Confidence intervals (90%)
lambda = 1800;
N = 40;

for it = 1:8
   for i= 1:N
      [PL(i)  APD(i)  MPD(i)  TT(i)] = simulator2(lambda, C, f(it), P, b);
   end
      media_PL(it) = mean(PL);
      media_APD(it) = mean(APD);
      media_TT(it) = mean(TT);
end

C=10 * 1000000; % link bandwidth (bytes)
media = mean([65:109, 111:1517]);
% Tamanho médio por pacote (em bits)
B = 0.16 * 64 + 0.25 * 110 + 0.2 * 1518 + 0.39 * media;
B = B * 8;
miu = (C / B);  % Sempre que o miu é superior ao lambda o modelo é valido
                % porque tem espera infinita
f = 2500:2500:20000;

for i = 1:8
   m = f(i) * 8 / B + 1;
   m = round(m);
```

```
    somatorio = 0;
    for j = 0:m
        somatorio = somatorio + ((lambda / miu)^j);
    end
    miu_m(i) = 100 * ((lambda/miu)^m) / somatorio

    somatorio_2 = 0;
    somatorio_3 = 0;
    for k = 0:m
        somatorio_2 = somatorio_2 + (k * ((lambda/miu)^k));
        somatorio_3 = somatorio_3 + ((lambda/miu)^k);
    end

    L = somatorio_2/somatorio_3
    W(i) = L / (lambda * (1 - miu_m(i)/100)) * 1000;
    TT2(i) = (lambda * B / 1e6) * (1-miu_m(i)/100);
end

% PL
figure(1)
bar(f, [media_PL; miu_m])
title('Packet Loss')
legend('Simulation', 'M/M/1/m', 'location', 'northwest')
ylim([0 20])
grid on

% APL
figure(2)
bar(f, [media_APD; W])
title('Av. Packet Delay (ms)')
legend('Simulation', 'M/M/1/m', 'location', 'northwest')
grid on

% TT
figure(3)
bar(f, [media_TT; TT2])
title('Throughput (Mbps)')
legend('Simulation', 'M/M/1/m', 'location', 'northwest')
grid on
```

Obtivemos então os seguintes gráficos:


Packet Loss
Simulation
M/M/1/m
20
18
16
14
12
10
8
6
4
2
0
0.25
0.5
0.75
1
1.25
1.5
1.75
2
$\times 10^4$


Av. Packet Delay (ms)
Simulation
M/M/1/m
6
5
4
3
2
1
0
0.25
0.5
0.75
1
1.25
1.5
1.75
2
$\times 10^4$



Como podemos ver, o valor dado pelo sistema M/M/1/m é sempre um pouco superior ao valor da simulação.

3.g. Repeat experiment 3.b considering now that the connection has a BER of b = 10^-5 . Present the results and the confidence intervals in bar charts with error bars. Justify the obtained results comparing them with the simulation results of experiment 3.b.

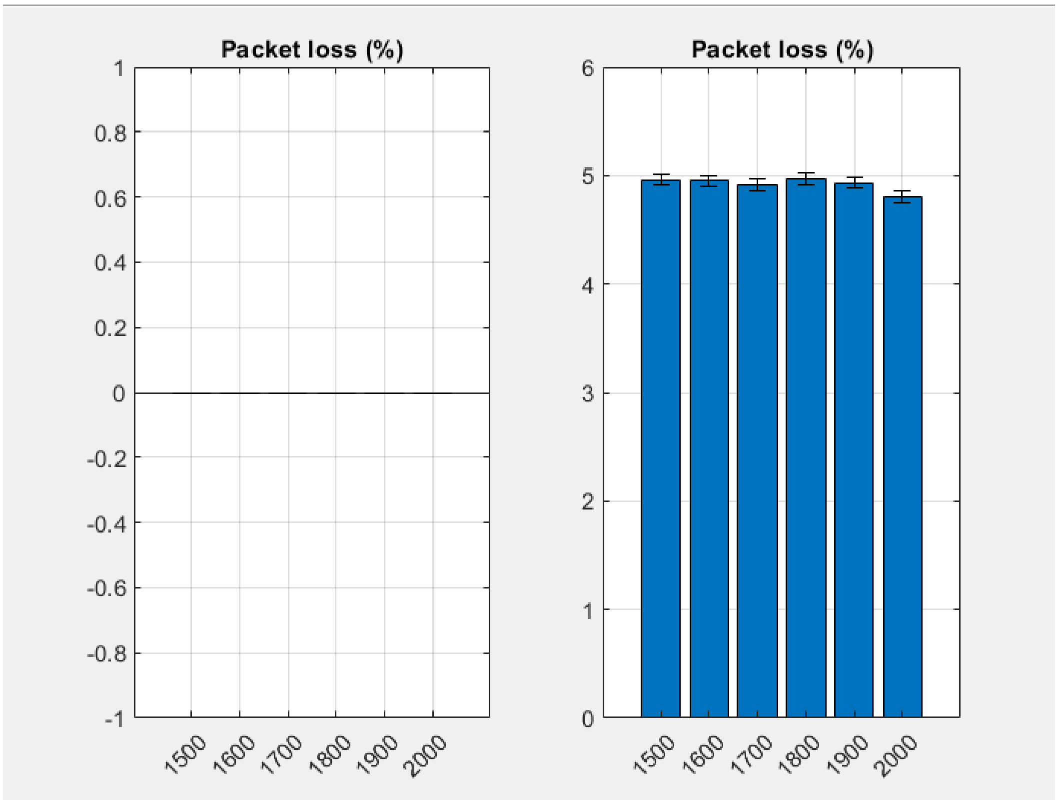
```
clear all

P = 10000; % Stopping criterion
lambda = 1500:100:2000;
alfa = 0.1; % Confidence intervals (90%)
N = 40; % number of simulations
C = 10; % link bandwidth (Mbps)
f = 10000000; % queue size (Bytes)
b = 0; % bit error rate

for it = 1:6
   for i= 1:N
      [PL(i)  APD(i)  MPD(i)  TT(i)] = simulator2(lambda(it),C,f,P,b);
   end
      media_PL(it) = mean(PL);
      media_APD(it) = mean(APD);
      media_MPD(it) = mean(MPD);
      media_TT(it) = mean(TT);
```

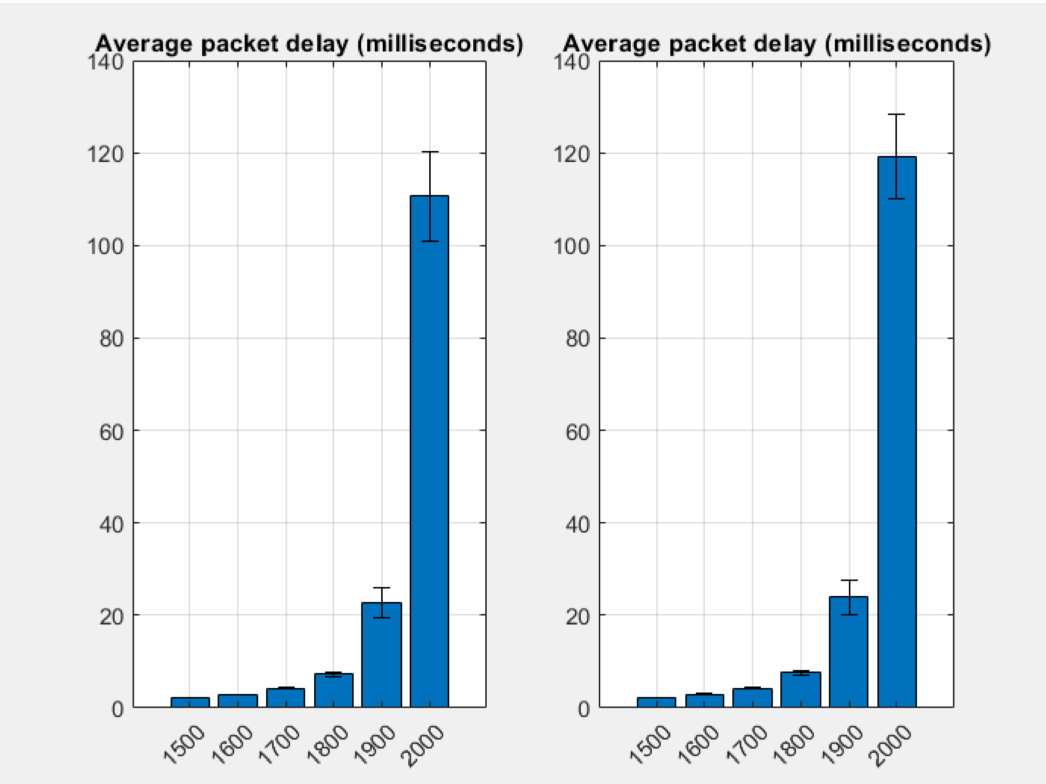
```
        term_PL(it) = norminv(1-alfa/2)*sqrt(var(PL)/N);
        term_APD(it) = norminv(1-alfa/2)*sqrt(var(APD)/N);
        term_MPD(it) = norminv(1-alfa/2)*sqrt(var(MPD)/N);
        term_TT(it) = norminv(1-alfa/2)*sqrt(var(TT)/N);
end

% PL
figure(1)
subplot(1,2,1)
bar(lambda, media_PL)
title('Packet loss (%)')
grid on
hold on

% Error bar
er = errorbar(lambda, media_PL, term_PL, term_PL);
er.Color = [0 0 0];
er.LineStyle = 'none';
hold off

% APD
figure(2)

subplot(1,2,1)
bar(lambda, media_APD)
title('Average packet delay (milliseconds)')
grid on
hold on

% Error bar
er = errorbar(lambda, media_APD, term_APD, term_APD);
er.Color = [0 0 0];
er.LineStyle = 'none';
hold off

% MPD
figure(3)
subplot(1,2,1)
bar(lambda, media_MPD)
title('Maximum packet delay (milliseconds)')
grid on
hold on

% Error bar
er = errorbar(lambda, media_MPD, term_MPD, term_MPD);
er.Color = [0 0 0];
er.LineStyle = 'none';
hold off

% TT
figure(4)
subplot(1,2,1)
bar(lambda, media_TT)
```

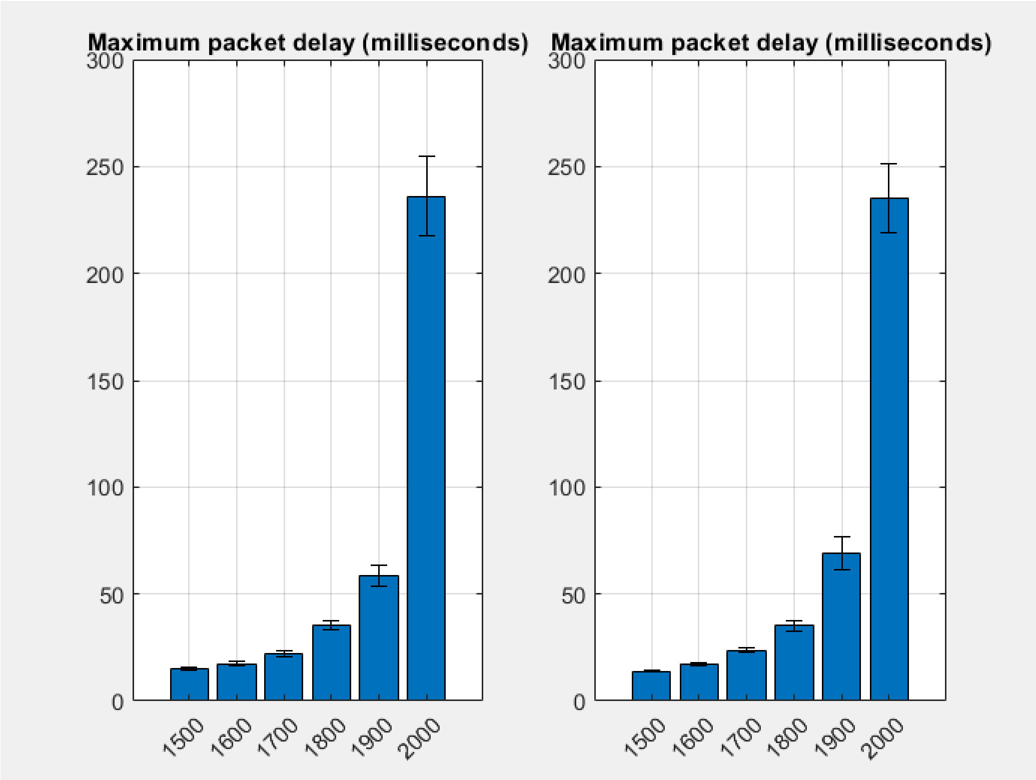
```
title('Transmitted throughput (Mbps)')
grid on
hold on

% Error bar
er = errorbar(lambda, media_TT, term_TT, term_TT);
er.Color = [0 0 0];
er.LineStyle = 'none';
hold off

clear all

P = 10000; % Stopping criterion
lambda = 1500:100:2000;
alfa = 0.1; % Confidence intervals (90%)
N = 40; % number of simulations
C = 10; % link bandwidth (Mbps)
f = 10000000; % queue size (Bytes)
b = 10e-6; % bit error rate

for it = 1:6
   for i= 1:N
      [PL(i)  APD(i)  MPD(i)  TT(i)] = simulator2(lambda(it),C,f,P,b);
   end
      media_PL(it) = mean(PL);
      media_APD(it) = mean(APD);
      media_MPD(it) = mean(MPD);
      media_TT(it) = mean(TT);

      term_PL(it) = norminv(1-alfa/2)*sqrt(var(PL)/N);
      term_APD(it) = norminv(1-alfa/2)*sqrt(var(APD)/N);
      term_MPD(it) = norminv(1-alfa/2)*sqrt(var(MPD)/N);
      term_TT(it) = norminv(1-alfa/2)*sqrt(var(TT)/N);
end

% PL
figure(1)
subplot(1,2,2)
bar(lambda, media_PL)
title('Packet loss (%)')
grid on
hold on

% Error bar
er = errorbar(lambda, media_PL, term_PL, term_PL);
er.Color = [0 0 0];
er.LineStyle = 'none';
hold off

% APD
figure(2)

subplot(1,2,2)
```

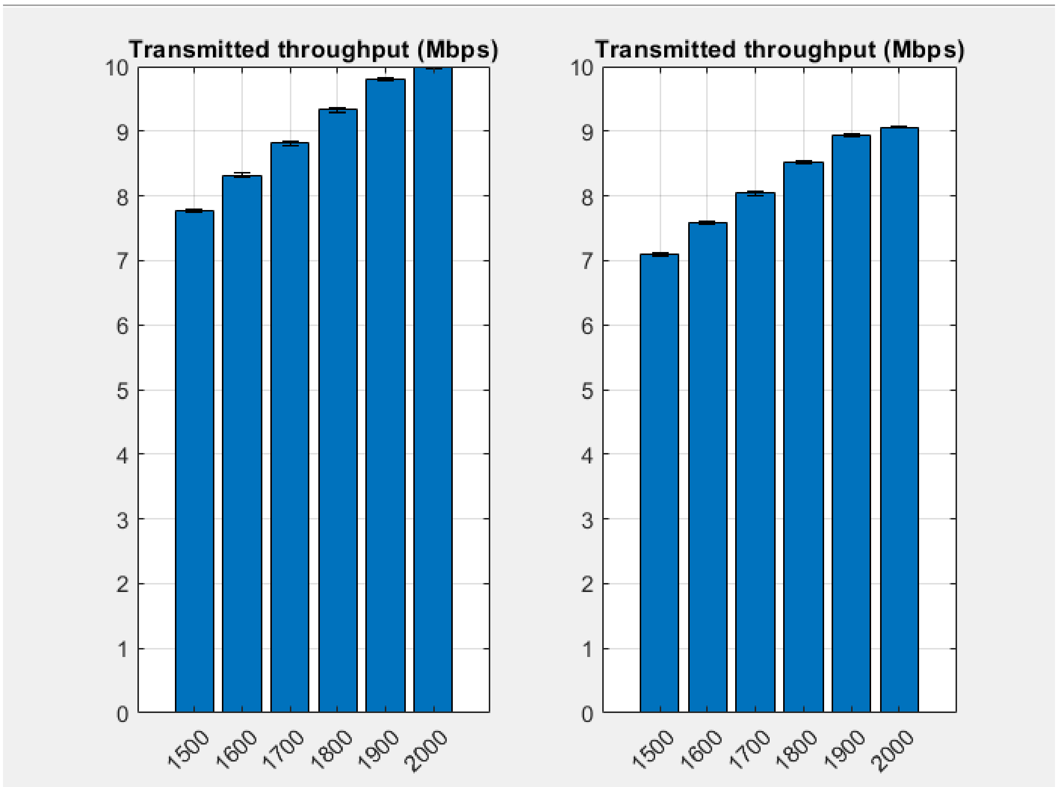
```
bar(lambda, media_APD)
title('Average packet delay (milliseconds)')
grid on
hold on

% Error bar
er = errorbar(lambda, media_APD, term_APD, term_APD);
er.Color = [0 0 0];
er.LineStyle = 'none';
hold off

% MPD
figure(3)
subplot(1,2,2)
bar(lambda, media_MPD)
title('Maximum packet delay (milliseconds)')
grid on
hold on

% Error bar
er = errorbar(lambda, media_MPD, term_MPD, term_MPD);
er.Color = [0 0 0];
er.LineStyle = 'none';
hold off

% TT
figure(4)
subplot(1,2,2)
bar(lambda, media_TT)
title('Transmitted throughput (Mbps)')
grid on
hold on

% Error bar
er = errorbar(lambda, media_TT, term_TT, term_TT);
er.Color = [0 0 0];
er.LineStyle = 'none';
hold off
```

Obtivemos então os seguintes gráficos:


Packet loss (%)
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
1500
1600
1700
1800
1900
2000
Packet loss (%)
6
5
4
3
2
1
0
1500
1600
1700
1800
1900
2000


Average packet delay (milliseconds)
140
120
100
80
60
40
20
0
1500
1600
1700
1800
1900
2000
Average packet delay (milliseconds)
140
120
100
80
60
40
20
0
1500
1600
1700
1800
1900
2000





Como podemos ver, os resultados foram similares em ambas as simulações, com exceção no *packet loss*. Isto deve-se ao facto de na alínea b) não existir *bit error rate*, enquanto que na alínea g) tem *bit error rate*.

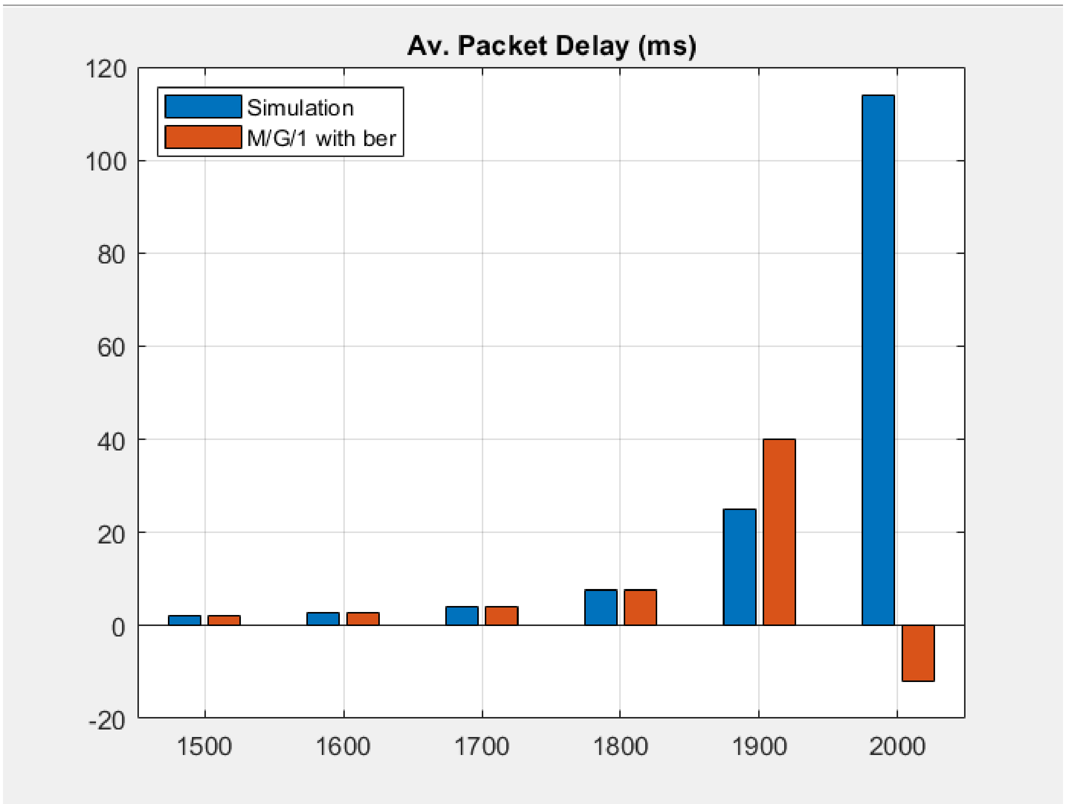
3.h. For each of the previous cases of experiment 3.g, compute the average packet delay and the total throughput of the connection considering that it is modelled by a M/G/1 system. Present both theoretical values and the simulation results of 3.g (without confidence intervals) in a bar chart for the average packet delay and another bar chart for the total throughput. Take conclusions on how close the theoretical model is from the simulation results of the different cases.

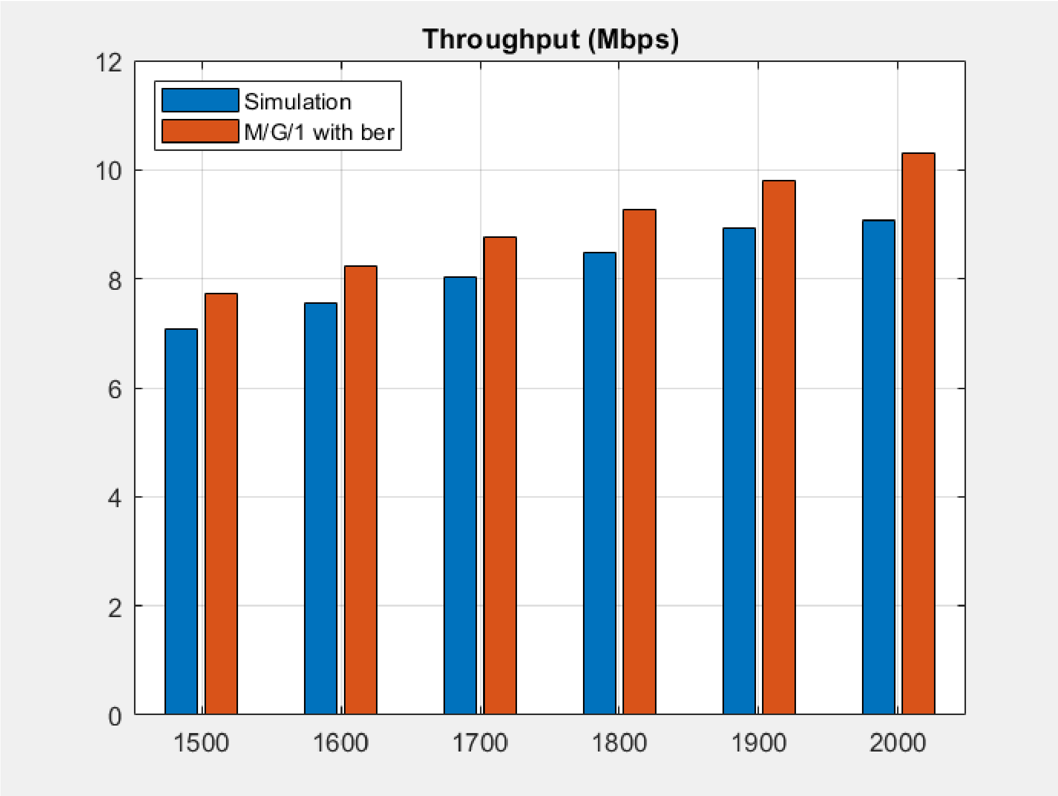
```
clear all

P = 10000; % Stopping criterion
lambda = 1500:100:2000;
alfa = 0.1;
N = 40; % number of simulations
C = 10;
f = 10000000;
b = 10e-6;

for it = 1:6
    for i= 1:N
        [PL(i)  APD(i)  MPD(i)  TT(i)] = simulator2(lambda(it),C,f,P,b);
    end
        media_APD(it) = mean(APD);
        media_TT(it) = mean(TT);
end

somatorio = 0;
C = 10e6;       % Capacidade da ligação
a = [65:109, 111:1517];
p_each = (1 - (0.16 + 0.25 + 0.2)) / length(a);

%average packet delay

e_s = 0;
e_s2 = 0;
for i = 64:1518
    if i == 64
        e_s = e_s + 0.16 * (8 * i / C);
        e_s2 = e_s2 + 0.16 * (8 * i / C)^2;
    elseif i == 110
        e_s = e_s + 0.25 * (8 * i / C);
        e_s2 = e_s2 + 0.25 * (8 * i / C)^2;
    elseif i == 1518
        e_s = e_s + 0.20 * (8 * i / C);
        e_s2 = e_s2 + 0.20 * (8 * i / C)^2;
    else
        e_s = e_s + p_each * (8 * i / C);
        e_s2 = e_s2 + p_each * (8 * i / C)^2;
    end
end
for i = 1:6
```

```
   wq(i) = (lambda(i) * e_s2) / (2 * (1 - lambda(i) * e_s));
end


num = [0,0,0,0,0,0];
den = [0,0,0,0,0,0];
for it = 1:6
   for i = 64:1518
      wi = wq(it) + (8 * i / C);
      if i == 64
         num(it) = num(it) + 0.16 * ((1 - 1e-5)^(8*i)) * wi;
         den(it) = den(it) + 0.16 * ((1 - 1e-5)^(8*i));
      elseif i == 110
         num(it) = num(it) + 0.25 * ((1 - 1e-5)^(8*i)) * wi;
         den(it) = den(it) + 0.25 * ((1 - 1e-5)^(8*i));
      elseif i == 1518
         num(it) = num(it) + 0.20 * ((1 - 1e-5)^(8*i)) * wi;
         den(it) = den(it) + 0.20 * ((1 - 1e-5)^(8*i));
      else
         num(it) = num(it) + p_each * ((1 - 1e-5)^(8*i)) * wi;
         den(it) = den(it) + p_each * ((1 - 1e-5)^(8*i));
      end
   end
   APD2(it) = num(it)/den(it) * 1000; % multiplica -se por 1000 para transformar em
milissegundos
end

%total throughput
TT2 = [0,0,0,0,0,0];
for it = 1:6
   for i = 64:1518
      if i == 64
         TT2(it) = TT2(it) + 0.16 * (1 - 1e-5)^(8*i) * lambda(it) * (8 * i);
      elseif i == 110
         TT2(it) = TT2(it) + 0.25 * (1 - 1e-5)^(8*i) * lambda(it) * (8 * i);
      elseif i == 1518
         TT2(it) = TT2(it) + 0.20 * (1 - 1e-5)^(8*i) * lambda(it) * (8 * i);
      else
         TT2(it) = TT2(it) + p_each * (1 - 1e-5)^(8*i) * lambda(it) * (8 * i);
      end
   end
end

% APL
figure(1)
bar(lambda, [media_APD; APD2])
title('Av. Packet Delay (ms)')
legend('Simulation', 'M/G/1 with ber', 'location', 'northwest')
grid on

% TT
figure(2)
bar(lambda, [media_TT; TT2/1e6])
```

```
title('Throughput (Mbps)')
legend('Simulation', 'M/G/1 with ber', 'location', 'northwest')
grid on
```

Obtivemos então os seguintes gráficos:





Quanto aos valores de *Average Packet Delay*, verificamos uma subida repentina quando os valores de lambda se aproximam de 2000, pois o tamanho da fila de espera é aproximadamente 1920. No caso do sistema *M/G/1 with ber*, o atraso até dá negativo para valores de lambda iguais a 2000 pps, pois estes módulos não estão preparados para casos em que o número de pacotes por segundo é superior ao tamanho da fila de espera.

Quanto ao *transmitted throughput*, podemos observar que o seu valor é maior para o sistema *M/G/1 with ber*.

Task 4

Simulador 3

```
function [PL , APD , MPD , TT] = simulator3(lambda,C,f,P,b)
% INPUT PARAMETERS:
%  lambda - packet rate (packets/sec)
%  C      - link bandwidth (Mbps)
%  f      - queue size (Bytes)
%  P      - number of packets (stopping criterium)
%  b      - bit error rate
% OUTPUT PARAMETERS:
%  PL   - packet loss (%)
%  APD  - average packet delay (milliseconds)
```

universidade de aveiro

```
%  MPD  - maximum packet delay (milliseconds)
%  TT   - transmitted throughput (Mbps)

% Probabilidade de cada estado
somatorio=1 + 10/5 + 10/5 * 5/10;
p1=1 / somatorio;  %probabilidade estado 1
p2=10/5 / somatorio; %probabilidade estado 2
p3=10/5 * 5/10 / somatorio; %probabilidade estado 3

%Tempo de permanência em cada estado
t=1/10; %t é igual em qualquer estado; t1 = 1/10, t2 = 1/(5+5), t3 = 1/10

%Events:
ARRIVAL= 0;       % Arrival of a packet
DEPARTURE= 1;     % Departure of a packet
TRANSITION = 2;   % The transition of a state in the packet arriving Markov chain

%State variables:
STATE = 0;          % 0 - connection free; 1 - connection bysy
QUEUEOCCUPATION= 0; % Occupation of the queue (in Bytes)
QUEUE= [];          % Size and arriving time instant of each packet in the queue
FLOWSTATE = 0;      % (integer variable that can be 1, 2 or 3 indicating which is the
% current state of the packet arriving Markov chain
lambda_vector = [0.5 * lambda, lambda, 1.5 * lambda];

%Statistical Counters:
TOTALPACKETS= 0;       % No. of packets arrived to the system
LOSTPACKETS= 0;        % No. of packets dropped due to buffer overflow
TRANSMITTEDPACKETS= 0; % No. of transmitted packets
TRANSMITTEDBYTES= 0;   % Sum of the Bytes of transmitted packets
DELAYS= 0;             % Sum of the delays of transmitted packets
MAXDELAY= 0;           % Maximum delay among all transmitted packets

%Auxiliary variables:
% Initializing the simulation clock:
Clock= 0;

x=rand;                          % random value to know the markov chain state
if x<=p1                % caso esteja no estado 1
   FLOWSTATE=1;
   EventList = [ARRIVAL, Clock + exprnd(1/(lambda/2)), GeneratePacketSize(), 0];
elseif x<=p1+p2    % caso esteja no estado 2
   FLOWSTATE=2;
   EventList = [ARRIVAL, Clock + exprnd(1/lambda), GeneratePacketSize(), 0];
else                    % caso esteja no estado 3
   FLOWSTATE=3;
   EventList = [ARRIVAL, Clock + exprnd(1/(lambda*3/2)), GeneratePacketSize(), 0];
end
% Initializing the List of Events with the first ARRIVAL:
EventList = [EventList; TRANSITION, Clock + exprnd(t), 0, 0];

%Similation loop:
while TRANSMITTEDPACKETS<P              % Stopping criterium
```

```
    EventList= sortrows(EventList,2);    % Order EventList by time
    Event= EventList(1,1);               % Get first event and
    Clock= EventList(1,2);               %   and
    PacketSize= EventList(1,3);          %   associated
    ArrivalInstant= EventList(1,4);      %   parameters.
    EventList(1,:)= [];                  % Eliminate first event


    switch Event
       case ARRIVAL                      % If first event is an ARRIVAL
          TOTALPACKETS= TOTALPACKETS+1;
          EventList = [EventList; ARRIVAL, Clock + exprnd(1/lambda_vector(FLOWSTATE)),
GeneratePacketSize(), 0];
          if STATE==0
             STATE= 1;
             EventList = [EventList; DEPARTURE, Clock + 8*PacketSize/(C*10^6),
PacketSize, Clock];
          else
             if QUEUEOCCUPATION + PacketSize <= f
                QUEUE= [QUEUE;PacketSize , Clock];
                QUEUEOCCUPATION= QUEUEOCCUPATION + PacketSize;
             else
                LOSTPACKETS= LOSTPACKETS + 1;
             end
          end
       case DEPARTURE                    % If first event is a DEPARTURE
          P_error = (1-b)^(8*PacketSize);
          x=rand();
          if x<P_error
             TRANSMITTEDBYTES= TRANSMITTEDBYTES + PacketSize;
             DELAYS= DELAYS + (Clock - ArrivalInstant);
             if Clock - ArrivalInstant > MAXDELAY
                MAXDELAY= Clock - ArrivalInstant;
             end
             TRANSMITTEDPACKETS= TRANSMITTEDPACKETS + 1;
          else
             LOSTPACKETS= LOSTPACKETS + 1;
          end
          if QUEUEOCCUPATION > 0
             EventList = [EventList; DEPARTURE, Clock + 8*QUEUE(1,1)/(C*10^6),
QUEUE(1,1), QUEUE(1,2)];
             QUEUEOCCUPATION= QUEUEOCCUPATION - QUEUE(1,1);
             QUEUE(1,:)= [];
          else
             STATE= 0;
          end
       case TRANSITION
          EventList = [EventList; TRANSITION, Clock + exprnd(t), 0, 0];
          if FLOWSTATE~=2 % se o flowstate for diferente de 2, o próximo flowstate será
% obrigatoriamente 2, pois a única transição possível do flowstate 1 e 3 é para o 2
             FLOWSTATE=2;
          else
             if rand > 0.5 % probabilidade de transição será (5 / (5+5))  para ambos os
```

```
%estados
                FLOWSTATE = 3;
            else
                FLOWSTATE = 1;
            end
        end

    end
end

% Performance parameters determination:
PL= 100*LOSTPACKETS/TOTALPACKETS;       % in %
APD= 1000*DELAYS/TRANSMITTEDPACKETS;    % in milliseconds
MPD= 1000*MAXDELAY;                     % in milliseconds
TT= 10^(-6)*TRANSMITTEDBYTES*8/Clock;   % in Mbps

end

function out= GeneratePacketSize()
    aux= rand();
    aux2= [65:109 111:1517];
    if aux <= 0.16
        out= 64;
    elseif aux <= 0.16 + 0.25
        out= 110;
    elseif aux <= 0.16 + 0.25 + 0.2
        out= 1518;
    else
        out = aux2(randi(length(aux2)));
    end
end
```

4.a. Run Simulator3 10 times with a stopping criterion of P = 100000 at each run to estimate all performance parameters and their 90% confidence intervals when = 1800 pps, C = 10 Mbps, f = 10^6 Bytes (~1 MByte) and the BER is b = 0.

```
clear all

P = 100000; % Stopping criterion
lambda = 1800;
alfa = 0.1;
N = 10; % number of simulations
C = 10;
f = 1e6;
b = 0;

for i = 1:N
    [PL(i) APD(i) MPD(i) TT(i)] = simulator3(lambda,C,f,P,b);
end
```

```
media_PL = mean(PL);
media_APD = mean(APD);
media_MPD = mean(MPD);
media_TT = mean(TT);

term_PL = norminv(1-alfa/2)*sqrt(var(PL)/N);
term_APD = norminv(1-alfa/2)*sqrt(var(APD)/N);
term_MPD = norminv(1-alfa/2)*sqrt(var(MPD)/N);
term_TT = norminv(1-alfa/2)*sqrt(var(TT)/N);

fprintf('PacketLoss (%%) = %.2e +- %.2e\n', media_PL, term_PL);
fprintf('Av. Packet Delay (ms) = %.2e +- %.2e\n', media_APD, term_APD);
fprintf('Max. Packet Delay (ms) = %.2e +- %.2e\n', media_MPD, term_MPD);
fprintf('Throughput (Mbps) = %.2e +- %.2e\n', media_TT, term_TT);
```

```
PacketLoss (%) = 1.61e-02 +- 2.03e-02
Av. Packet Delay (ms) = 1.34e+02 +- 3.28e+01
Max. Packet Delay (ms) = 5.15e+02 +- 8.93e+01
Throughput (Mbps) = 9.36e+00 +- 1.12e-01
```

Pelos resultados obtidos, temos uma perceção de que o simulador 3 está correto.

4.b. Run Simulator3 10 times with a stopping criterion of P = 100000 at each run to estimate all performance parameters and their 90% confidence intervals when = 1800 pps, C = 10 Mbps, f = 10^4 Bytes (~10 KBytes) and the BER is b = 10^-5 .

```
clear all

P = 100000; % Stopping criterion
lambda = 1800;
alfa = 0.1;
N = 10; % number of simulations
C = 10;
f = 1e4;
b = 1e-6;

for i = 1:N
   [PL(i) APD(i) MPD(i) TT(i)] = simulator3(lambda,C,f,P,b);
end

media_PL = mean(PL);
media_APD = mean(APD);
media_MPD = mean(MPD);
media_TT = mean(TT);
```

```
term_PL = norminv(1-alfa/2)*sqrt(var(PL)/N);
term_APD = norminv(1-alfa/2)*sqrt(var(APD)/N);
term_MPD = norminv(1-alfa/2)*sqrt(var(MPD)/N);
term_TT = norminv(1-alfa/2)*sqrt(var(TT)/N);

fprintf('PacketLoss (%%) = %.2e +- %.2e\n', media_PL, term_PL);
fprintf('Av. Packet Delay (ms) = %.2e +- %.2e\n', media_APD, term_APD);
fprintf('Max. Packet Delay (ms) = %.2e +- %.2e\n', media_MPD, term_MPD);
fprintf('Throughput (Mbps) = %.2e +- %.2e\n', media_TT, term_TT);
```

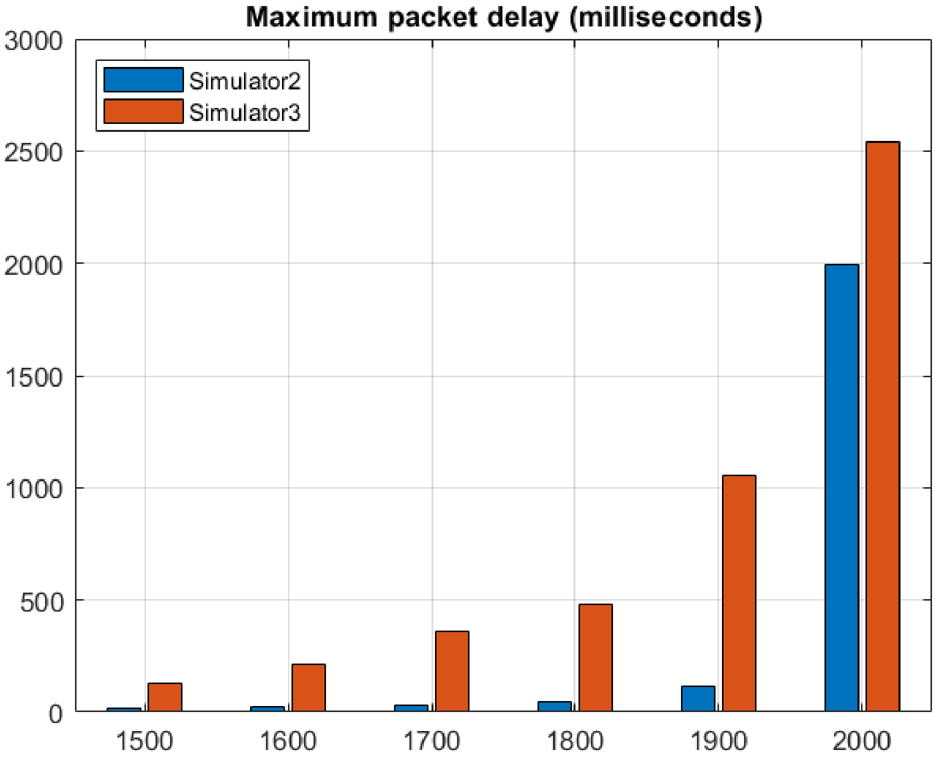
```
PacketLoss (%) = 6.80e+00 +- 1.51e-01
Av. Packet Delay (ms) = 4.25e+00 +- 4.24e-02
Max. Packet Delay (ms) = 9.18e+00 +- 7.80e-03
Throughput (Mbps) = 8.18e+00 +- 5.99e-02
```

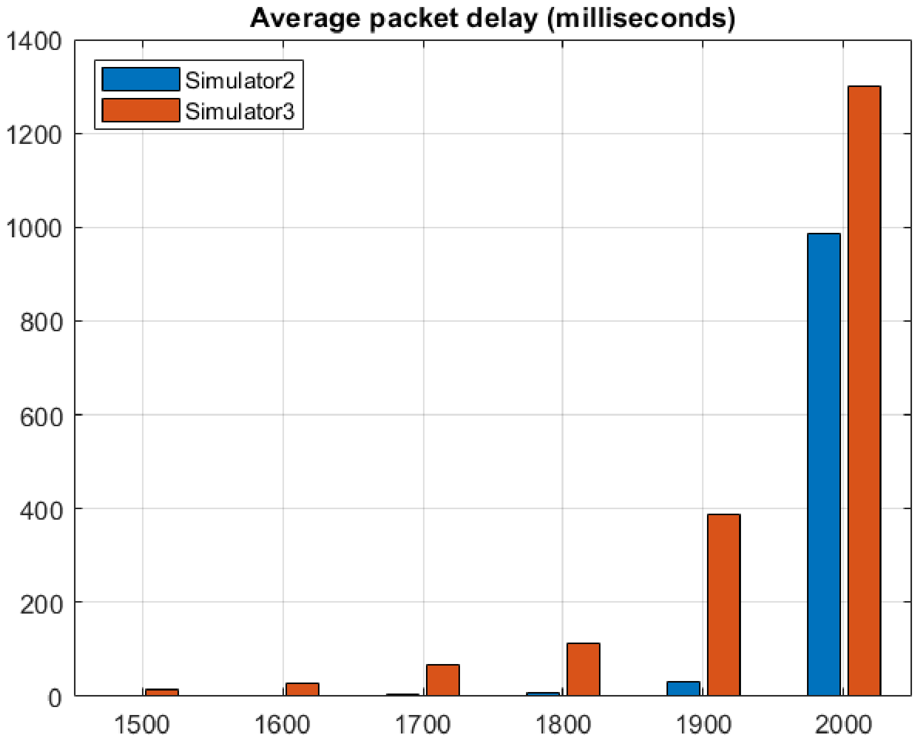
4.c. Consider the connection characterized by C = 10 Mbps, f = 10^7 Bytes (~10 MBytes) and BER b = 0. Run both Simulator2 and Simulator3, 10 times each, with a stopping criterion of P = 100000 at each run to estimate all performance parameters when λ = 1500, 1600, 1700, 1800, 1900 and 2000 pps. Present the results of both simulators in a single bar chart for each performance parameter. Take conclusions on the impact of both packet arriving models on the different performance parameters and justify the obtained results.

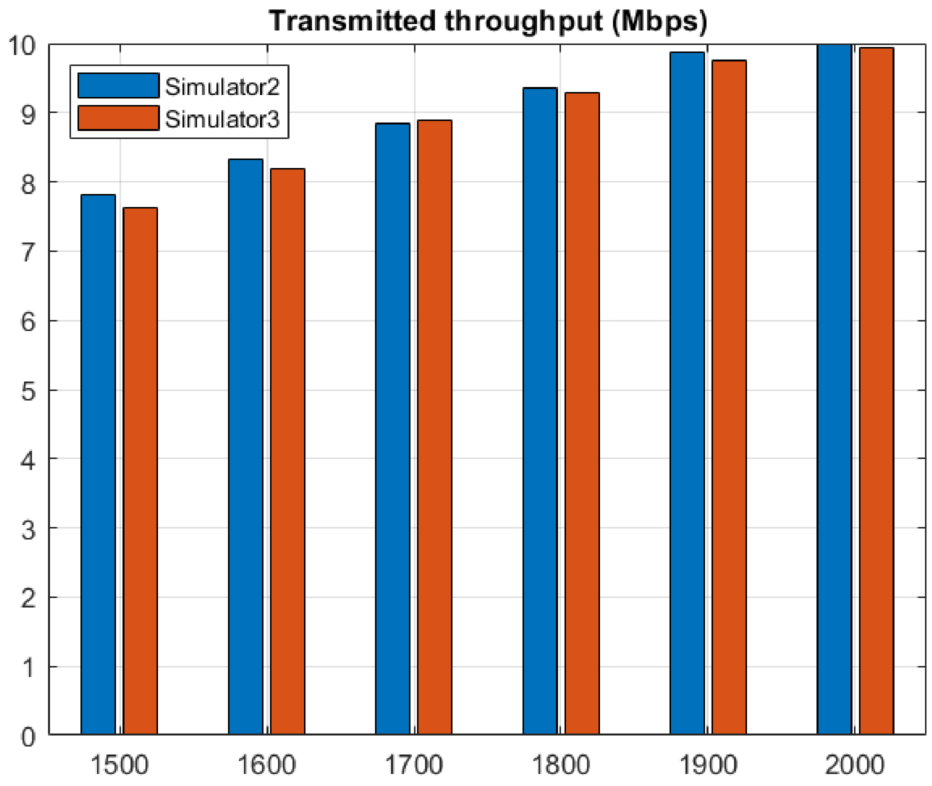
```
clear all

P = 100000; % Stopping criterion
lambda = 1500:100:2000;
alfa = 0.1;
N = 10; % number of simulations
C = 10;
f = 10000000;
b = 0;

for it = 1:6
   for i= 1:N
      [PL(i)  APD(i)  MPD(i)  TT(i)] = simulator2(lambda(it),C,f,P,b);
      [PL2(i)  APD2(i)  MPD2(i)  TT2(i)] = simulator3(lambda(it),C,f,P,b);
   end
      media_PL(it) = mean(PL);
      media_APD(it) = mean(APD);
      media_MPD(it) = mean(MPD);
      media_TT(it) = mean(TT);

      media_PL2(it) = mean(PL2);
      media_APD2(it) = mean(APD2);
      media_MPD2(it) = mean(MPD2);
```

```
    media_TT2(it) = mean(TT2);

end

% PL
figure(1)
bar(lambda, [media_PL; media_PL2])
title('Packet loss (%)')
legend('Simulator2', 'Simulator3', 'location', 'northwest')
grid on

% APD
figure(2)
bar(lambda, [media_APD; media_APD2])
title('Average packet delay (milliseconds)')
legend('Simulator2', 'Simulator3', 'location', 'northwest')
grid on

% MPD
figure(3)
bar(lambda, [media_MPD; media_MPD2])
title('Maximum packet delay (milliseconds)')
legend('Simulator2', 'Simulator3', 'location', 'northwest')
grid on

% TT
figure(4)
bar(lambda, [media_TT; media_TT2])
title('Transmitted throughput (Mbps)')
legend('Simulator2', 'Simulator3', 'location', 'northwest')
grid on
```

Obtivemos então os seguintes gráficos:


Maximum packet delay (milliseconds)
3000
2500
2000
1500
1000
500
0
Simulator2
Simulator3
1500
1600
1700
1800
1900
2000


Average packet delay (milliseconds)
1400
1200
1000
800
600
400
200
0
Simulator2
Simulator3
1500
1600
1700
1800
1900
2000



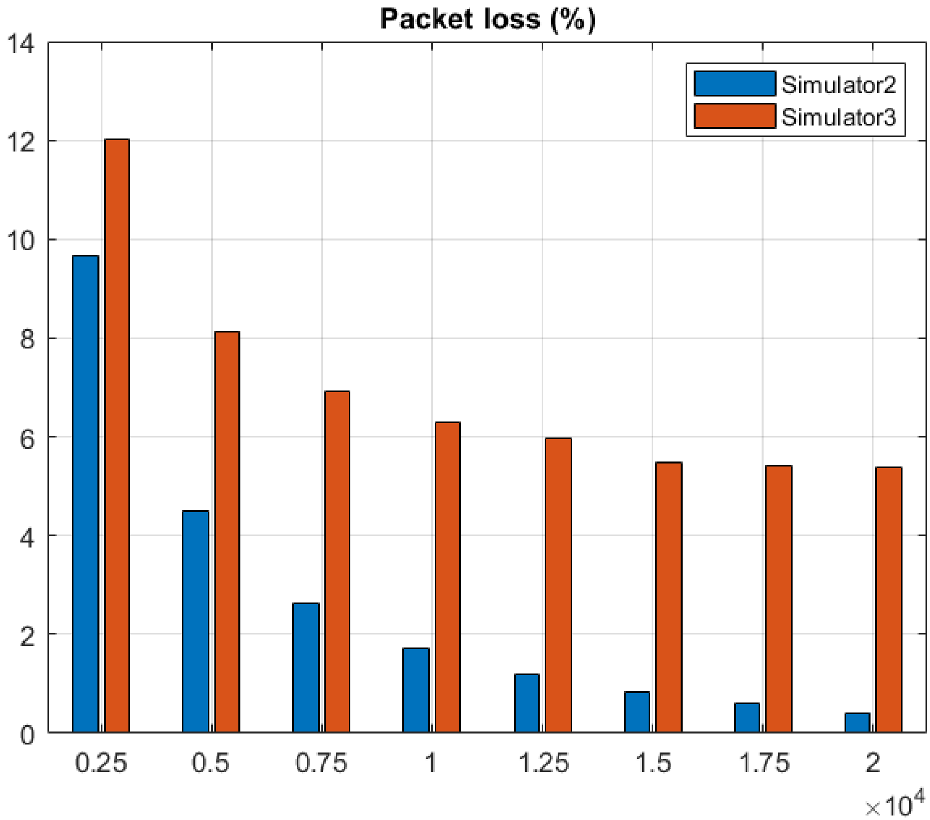
Para o *maximum packet delay* e para o *average packet delay*, o valor obtido no simulador 3 é maior, pois há mais estados no simulador, o que vai atrasar o processamento dos pacotes. Enquanto no simulador 2 apenas temos o *ARRIVAL* e o *DEPARTURE*, no simulador 3 temos o *ARRIVAL*, *TRANSITION* e o *DEPARTURE*. Como o *maximum delay* apenas é calculado no *DEPARTURE*, no simulador 3 demorará mais tempo a chegar ao *DEPARTURE*.
O que foi descrito anteriormente vai afetar o *transmitted throughput,* levando a que este seja menor no simulador 3, pois o atraso será maior, afetando o número de pacotes processados por unidade de tempo.

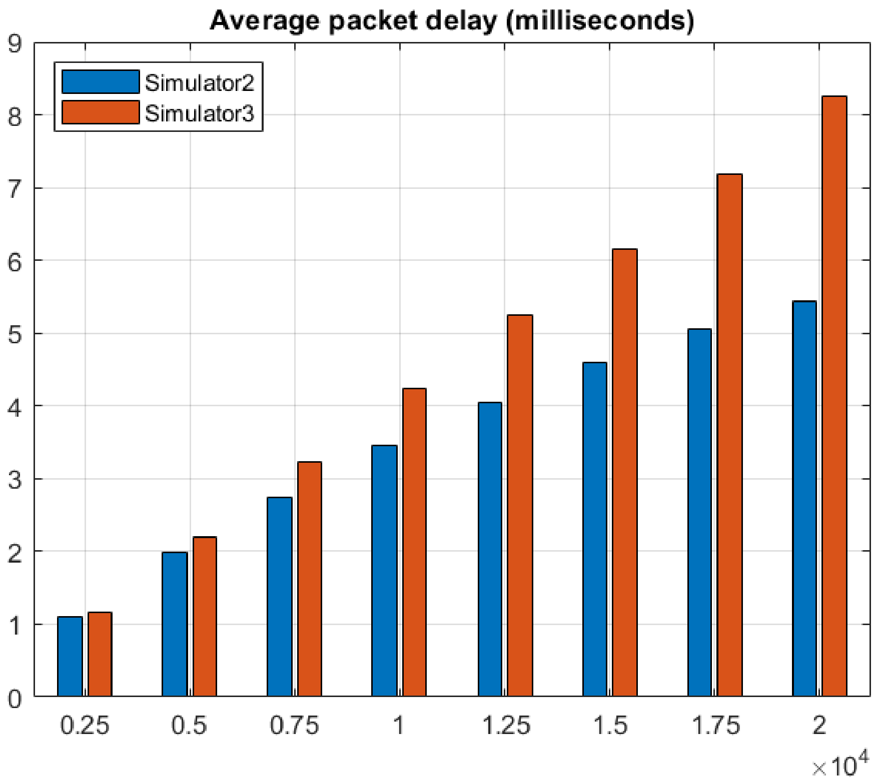
4.d. Consider the packet rate = 1800 pps and the connection characterized by C = 10 Mbps and BER b = 0. Run both Simulator2 and Simulator3, 10 times each, with a stopping criterion of P = 100000 at each run to estimate all performance parameters when f = 2500, 5000, 7500, 10000, 12500, 15000, 17500 and 20000 Bytes. Present the results of both simulators in a single bar chart for each performance parameter Take conclusions on the impact of both packet arriving models on the different performance parameters and justify the obtained results.

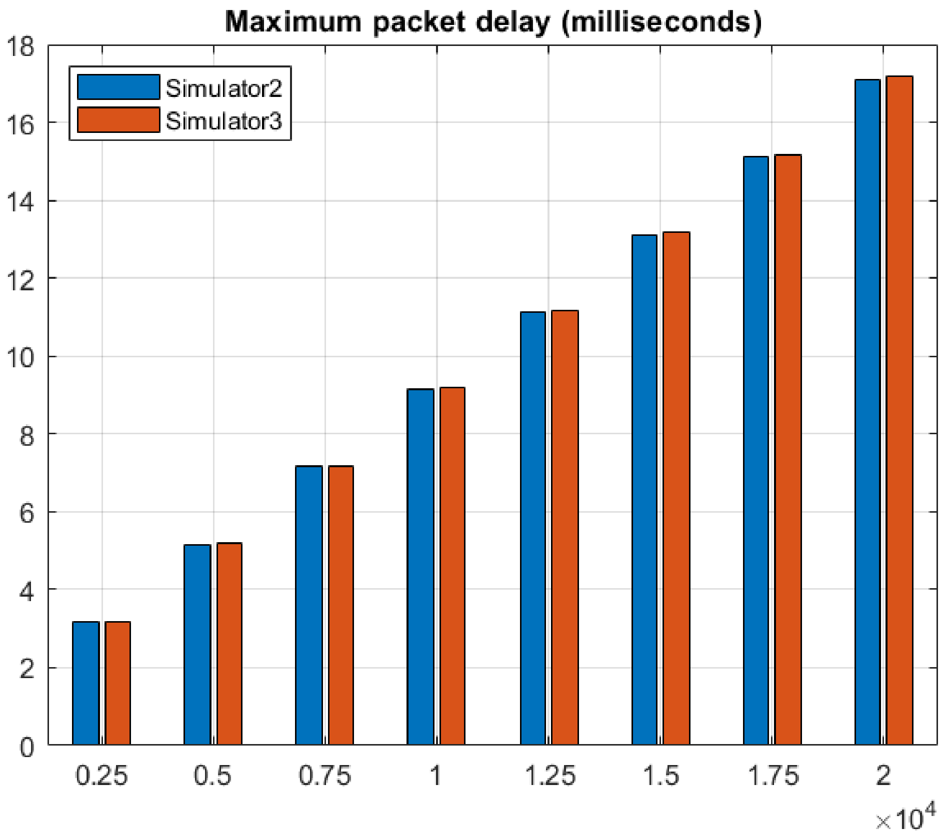
```
clear all
```

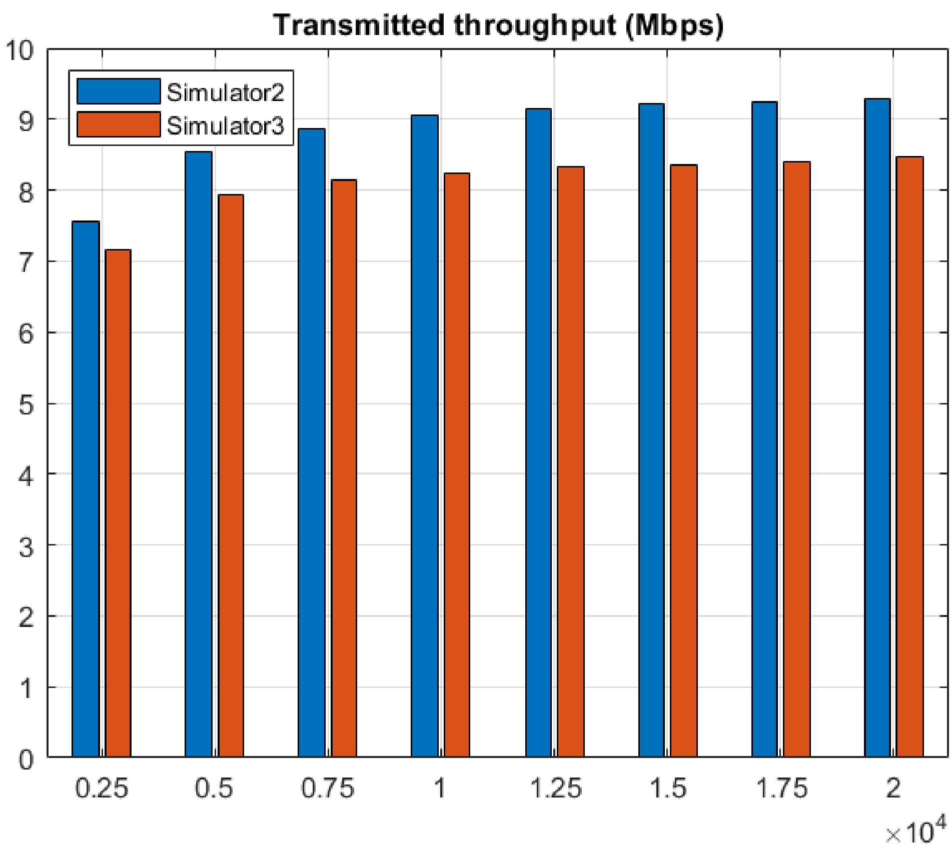
```
C = 10;
f = 2500:2500:20000;
b = 0;
P = 100000;
alfa = 0.1;     % 90% confidence interval
lambda = 1800;
N = 10;

for it = 1:8
   for i= 1:N
      [PL(i)  APD(i)  MPD(i)  TT(i)] = simulator2(lambda,C,f(it),P,b);
      [PL2(i)  APD2(i)  MPD2(i)  TT2(i)] = simulator3(lambda,C,f(it),P,b);
   end
      media_PL(it) = mean(PL);
      media_APD(it) = mean(APD);
      media_MPD(it) = mean(MPD);
      media_TT(it) = mean(TT);

      media_PL2(it) = mean(PL2);
      media_APD2(it) = mean(APD2);
      media_MPD2(it) = mean(MPD2);
      media_TT2(it) = mean(TT2);
end


% PL
figure(1)
bar(f, [media_PL; media_PL2])
title('Packet loss (%)')
legend('Simulator2', 'Simulator3', 'location', 'northeast')
grid on

% APD
figure(2)
bar(f, [media_APD; media_APD2])
title('Average packet delay (milliseconds)')
legend('Simulator2', 'Simulator3', 'location', 'northwest')
grid on

% MPD
figure(3)
bar(f, [media_MPD; media_MPD2])
title('Maximum packet delay (milliseconds)')
legend('Simulator2', 'Simulator3', 'location', 'northwest')
grid on

% TT
figure(4)
bar(f, [media_TT; media_TT2])
title('Transmitted throughput (Mbps)')
legend('Simulator2', 'Simulator3', 'location', 'northwest')
grid on
```

Obtivemos então os seguintes gráficos:






Maximum packet delay (milliseconds)
Simulator2
Simulator3
18
16
14
12
10
8
6
4
2
0
0.25
0.5
0.75
1
1.25
1.5
1.75
2
×10^4


Transmitted throughput (Mbps)
Simulator2
Simulator3
10
9
8
7
6
5
4
3
2
1
0
0.25
0.5
0.75
1
1.25
1.5
1.75
2
×10^4

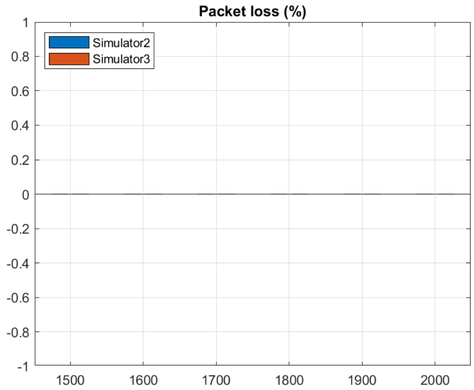
Perante estes resultados podemos concluir que o packet loss já não é insignificante, pois o tamanho da fila de espera é bastante mais pequeno, o que significa que a fila de espera fica cheia relativamente rápido e portanto os pacotes que chegam quando a fila de espera está cheia são perdidos. Daí que quanto maior a fila de espera, menor a probabilidade de os pacotes serem perdidos, como podemos ver no gráfico.
Outro aspeto importante é relativo ao Maximum packet delay, que é semelhante nas 2 simulações, isto deve-se ao facto de que o Maxdelay é calculado através da diferença entre o Clock e o ArrivalInstant. Visto que a fila de espera é bastante reduzida, os pacotes não ficam quase tempo nenhum na fila, pois ou entram ou são perdidos. Isto faz com que as transições sejam insignificantes no cálculo do Maximum packet delay.
Quanto ao average packet delay e ao transmitted throughput, aplica-se o motivo da alínea anterior.

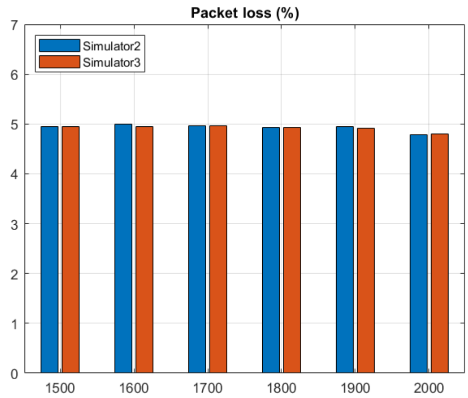
4.e. Repeat experiment 4.c considering now that the connection has a BER of b = 10^-5 . Justify the obtained results comparing them with the simulation results of experiment 4.c and take conclusions on the impact of both packet arriving models on the different performance parameters.

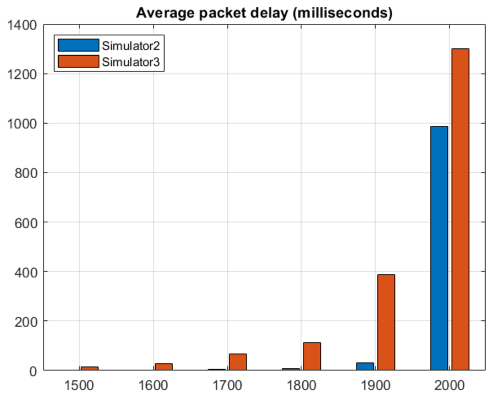
```
clear all

P = 100000; % Stopping criterion
lambda = 1500:100:2000;
alfa = 0.1;
N = 10; % number of simulations
C = 10;
f = 10000000;
b = 1e-5;

for it = 1:6
   for i= 1:N
      [PL(i)  APD(i)  MPD(i)  TT(i)] = simulator2(lambda(it),C,f,P,b);
      [PL2(i)  APD2(i)  MPD2(i)  TT2(i)] = simulator3(lambda(it),C,f,P,b);
   end
      media_PL(it) = mean(PL);
      media_APD(it) = mean(APD);
      media_MPD(it) = mean(MPD);
      media_TT(it) = mean(TT);

      media_PL2(it) = mean(PL2);
      media_APD2(it) = mean(APD2);
      media_MPD2(it) = mean(MPD2);
      media_TT2(it) = mean(TT2);
```

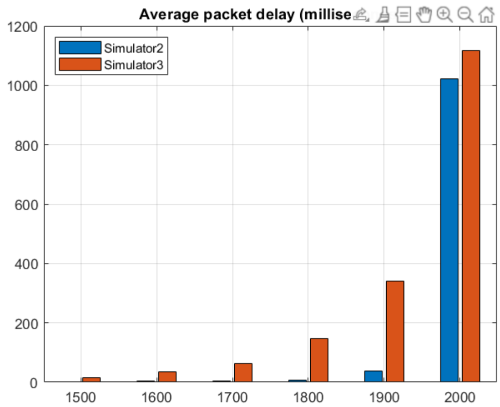
```
end

% PL
figure(1)
bar(lambda, [media_PL; media_PL2])
title('Packet loss (%)')
legend('Simulator2', 'Simulator3', 'location', 'northwest')
grid on

% APD
figure(2)
bar(lambda, [media_APD; media_APD2])
title('Average packet delay (milliseconds)')
legend('Simulator2', 'Simulator3', 'location', 'northwest')
grid on

% MPD
figure(3)
bar(lambda, [media_MPD; media_MPD2])
title('Maximum packet delay (milliseconds)')
legend('Simulator2', 'Simulator3', 'location', 'northwest')
grid on

% TT
figure(4)
bar(lambda, [media_TT; media_TT2])
title('Transmitted throughput (Mbps)')
legend('Simulator2', 'Simulator3', 'location', 'northwest')
grid on
```

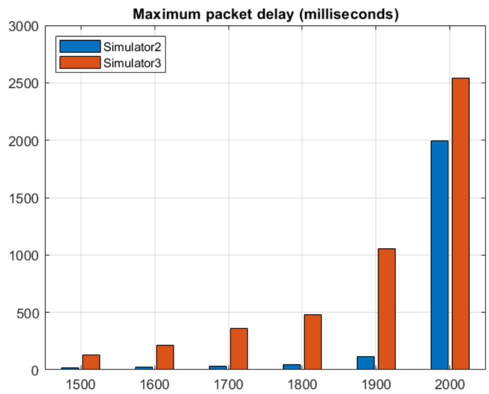
Obtivemos então os seguintes gráficos:

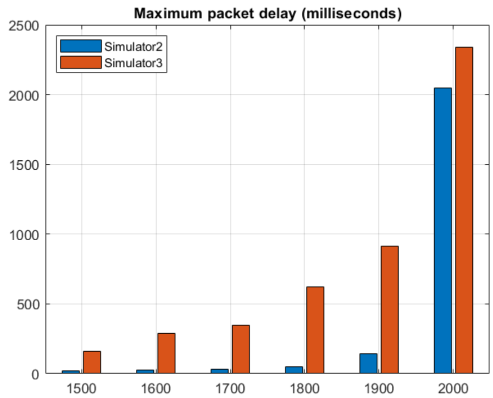


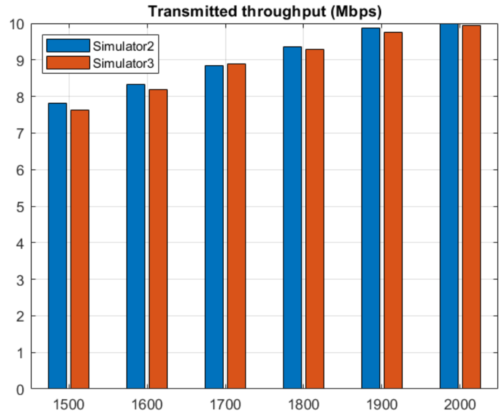


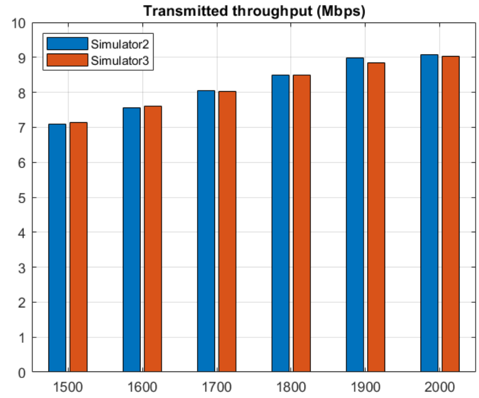












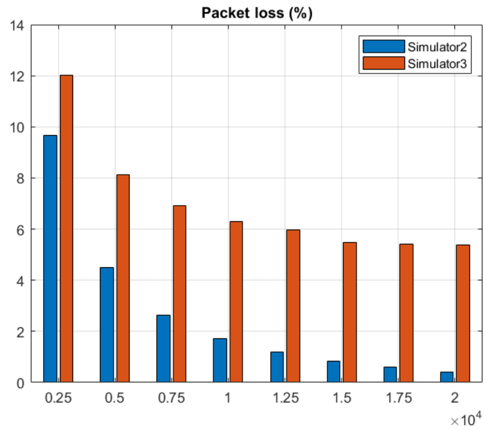
Nas imagens podemos ver os resultados da experiência 4.c do lado esquerdo, e da experiência 4.e do lado direito.
Nestes resultados podemos verificar que o packet loss é muito superior nesta experiência do que na experiência 4.c, pois na experiência 4.c não havia *bit error rate*.
Quanto ao *transmitted throughput*, será menor nesta experiência pois o número de pacotes processados será menor devido ao número de pacotes perdidos devido ao *bit error rate*.
O *bit error rate* não influencia os *maximum* e *average packet delay*.

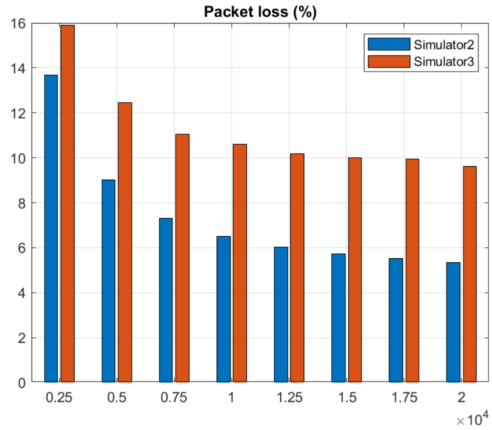
4.f. Repeat experiment 4.d considering now that the connection has a BER of b = 10^-5 . Justify the obtained results comparing them with the simulation results of experiment 4.d and take conclusions on the impact of both packet arriving models on the different performance parameters.

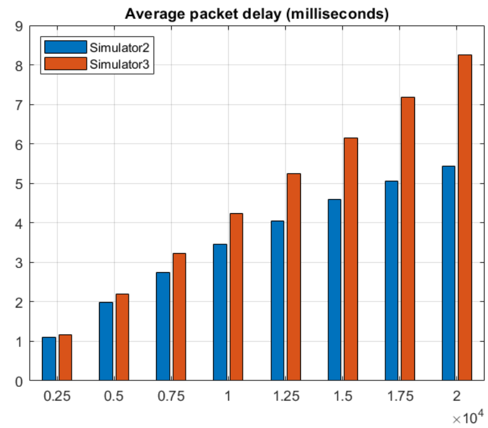
```
clear all

C = 10;
f = 2500:2500:20000;
b = 1e-5;
P = 100000;
alfa = 0.1;     % 90% confidence interval
lambda = 1800;
N = 10;

for it = 1:8
   for i= 1:N
      [PL(i)  APD(i)  MPD(i)  TT(i)] = simulator2(lambda,C,f(it),P,b);
      [PL2(i)  APD2(i)  MPD2(i)  TT2(i)] = simulator3(lambda,C,f(it),P,b);
   end
```

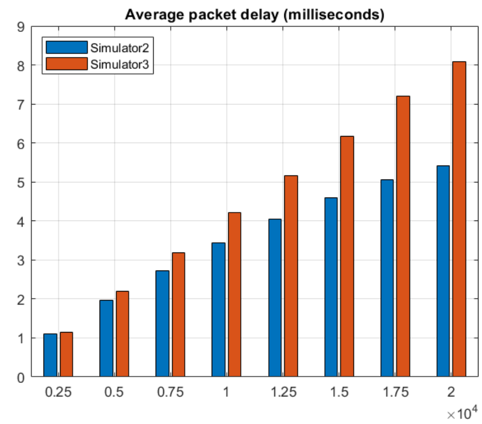
```
	media_PL(it) = mean(PL);
	media_APD(it) = mean(APD);
	media_MPD(it) = mean(MPD);
	media_TT(it) = mean(TT);

	media_PL2(it) = mean(PL2);
	media_APD2(it) = mean(APD2);
	media_MPD2(it) = mean(MPD2);
	media_TT2(it) = mean(TT2);
end


% PL
figure(1)
bar(f, [media_PL; media_PL2])
title('Packet loss (%)')
legend('Simulator2', 'Simulator3', 'location', 'northeast')
grid on

% APD
figure(2)
bar(f, [media_APD; media_APD2])
title('Average packet delay (milliseconds)')
legend('Simulator2', 'Simulator3', 'location', 'northwest')
grid on

% MPD
figure(3)
bar(f, [media_MPD; media_MPD2])
title('Maximum packet delay (milliseconds)')
legend('Simulator2', 'Simulator3', 'location', 'northwest')
grid on

% TT
figure(4)
bar(f, [media_TT; media_TT2])
title('Transmitted throughput (Mbps)')
legend('Simulator2', 'Simulator3', 'location', 'northwest')
grid on
```

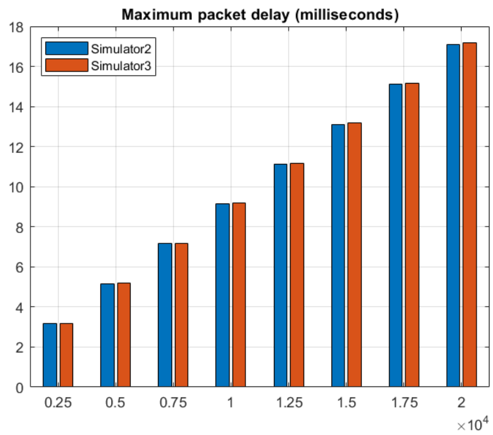
Obtivemos então os seguintes gráficos:

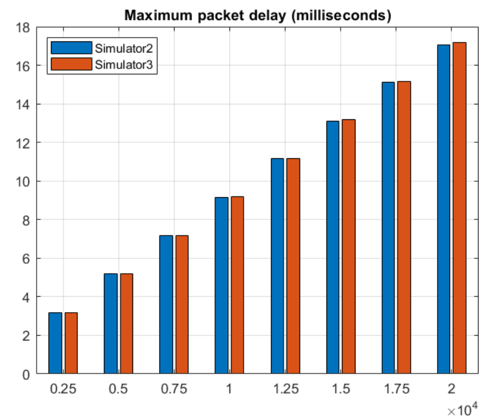


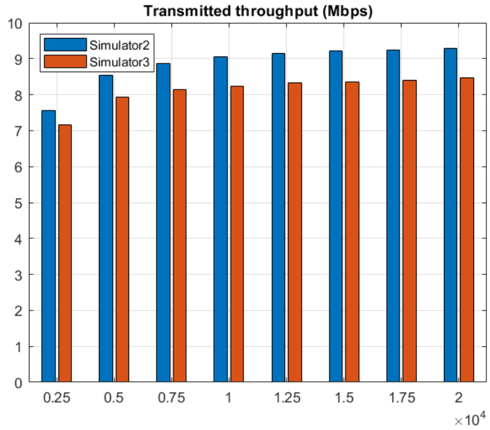


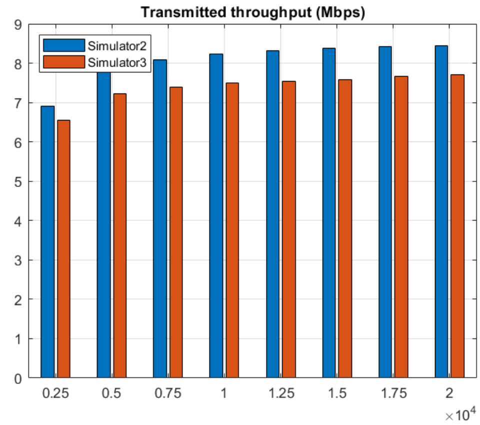












Nas imagens podemos ver os resultados da experiência 4.d do lado esquerdo, e da experiência 4.f do lado direito.

Nesta última alínea podemos observar que, tal como na anterior, o *packet loss* é maior com um *bit error rate* não nulo pois agora os pacotes têm uma probabilidade de serem perdidos, mesmo depois de entrarem na fila de espera.

Relativamente ao *transmitted throughput*, é também menor nesta experiência pois o número de pacotes processados será menor devido ao número de pacotes perdidos, tendo em conta o *bit error rate*.

Pedro Gonçalves 88859
Pedro Silva 89228