Relatório do Guião nº 1 de Desempenho e Dimensionamento de Redes

Task 4

4.a. Determine the probability of the link being in the normal state and in the interference state.

Para a resolução desta alínea, utilizamos a fórmula das probabilidades limite de processos de nascimento e morte.

$$\pi_0 = \frac{1}{1 + \sum_{i=1}^{\infty} \frac{\lambda_0 \lambda_1 \cdots \lambda_{i-1}}{\mu_1 \mu_2 \cdots \mu_i}} \qquad \pi_n = \frac{\lambda_0 \lambda_1 \cdots \lambda_{n-1}}{\mu_1 \mu_2 \cdots \mu_n \left(1 + \sum_{i=1}^{\infty} \frac{\lambda_0 \lambda_1 \cdots \lambda_{i-1}}{\mu_1 \mu_2 \cdots \mu_i}\right)} = \frac{\lambda_0 \lambda_1 \cdots \lambda_{n-1}}{\mu_1 \mu_2 \cdots \mu_n} \cdot \pi_0$$

Começamos por calcular o somatório seguindo a fórmula e em seguida calculamos a probabilidade de o link estar em cada um dos 5 estados.
Por fim, para calcular a probabilidade de o link estar no estado normal, somamos as probabilidades de cada um dos 3 estados normais (p1, p2 e p3).
Em relação à probabilidade de estar no estado de interferência, somamos as probabilidades de cada um dos 2 estados de interferência (p4 e p5).

```
somatorio = 1 + 8/600 + 8/600 * 5/200 + 8/600 * 5/200 * 2/50 + 8/600 * 5/200 * 2/50 * 1/5;

p1 = 1 / somatorio;
p2 = 8/600 / somatorio;
p3 = 8/600 * 5/200 / somatorio;
p4 = 8/600 * 5/200 * 2/50 / somatorio;
p5 = 8/600 * 5/200 * 2/50 * 1/5 / somatorio;

p_normal = p1 + p2 + p3;
p_interf = p4 + p5;

fprintf('Probability of the link being in the normal state = %.7f\n', p_normal)
fprintf('Probability of the link being in the interference state = %.7f\n', p_interf)
```

Resultado obtido na consola:

```
Probability of the link being in the normal state = 0.9999842
Probability of the link being in the interference state = 0.0000158
```

4.b. Determine the average ber of the link when it is in the normal state and when it is in the interference state.

Nesta alínea, calculamos a média de *bit error rate* através da soma da multiplicação da probabilidade de cada estado pelo seu *bit error rate.* No final dividimos pela soma da sua probabilidade devido à soma das suas probabilidades ser menor que 1.

```
somatorio = 1 + 8/600 + 8/600 * 5/200 + 8/600 * 5/200 * 2/50 + 8/600 * 5/200 * 2/50 * 1/5;

p1 = 1 / somatorio;
p2 = 8/600 / somatorio;
p3 = 8/600 * 5/200 / somatorio;
p4 = 8/600 * 5/200 * 2/50 / somatorio;
p5 = 8/600 * 5/200 * 2/50 * 1/5 / somatorio;

mediaint = (p4*10^-3 + p5*10^-2) / (p4+p5);
medianor = (p1*10^-6 + p2*10^-5 + p3*10^-4) / (p1+p2+p3);

fprintf('Average ber in normal state = %.2e\n', medianor)
fprintf('Average ber in interference state = %.2e\n', mediaint)
```

Resultado obtido na consola:

```
Average ber in normal state = 1.15e-06
Average ber in interference state = 2.50e-03
```

4.c. Consider that a data frame of size B (in Bytes) sent by one of the stations is received with errors by the other station. Draw a plot of the probability of the link being in the normal state as a function of the packet size (from 64 Bytes up to 200 Bytes). Analyze and justify the results.

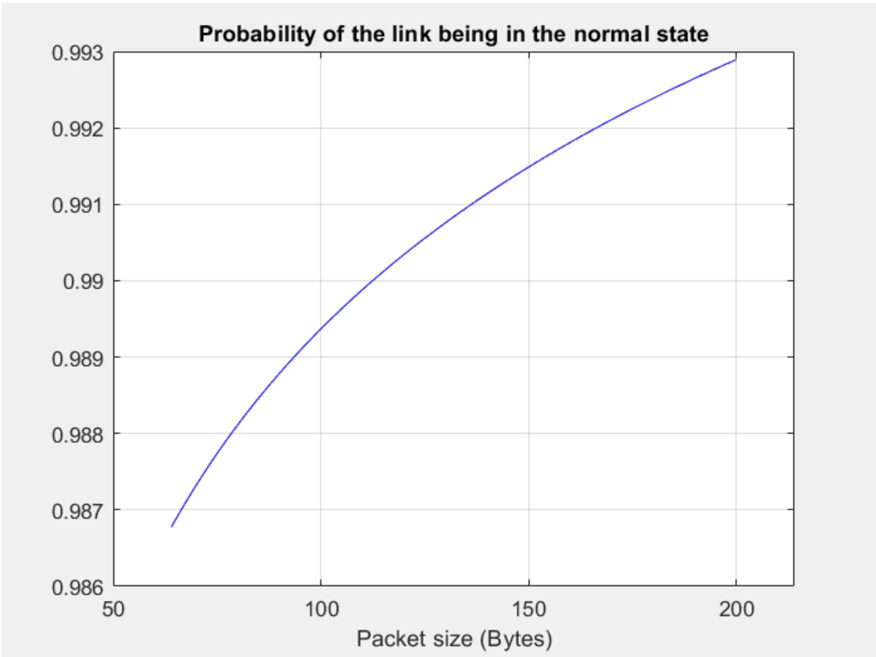
Começamos por calcular a probabilidade de erro de cada um dos 5 estados. Em seguida calculamos através da Regra de Bayes a probabilidade de o link estar no estado 1, 2 ou 3, sabendo que foi recebido com erro.

$$P(F_j \mid E) = \frac{P(EF_j)}{P(E)} = \frac{P(E \mid F_j)P(F_j)}{P(E)} = \frac{P(E \mid F_j)P(F_j)}{\sum_{i=1}^{n} P(E \mid F_i)P(F_i)}$$

Por fim, ao somarmos essas 3 probabilidades obtemos o resultado final (probabilidade de estar no estado normal sabendo que o link chegou com erros).

```
h = linspace(64*8, 200*8);

prob_erro_1 = (1 - (1 - 10^-6).^h);
prob_erro_2 = (1 - (1 - 10^-5).^h);
prob_erro_3 = (1 - (1 - 10^-4).^h);
prob_erro_4 = (1 - (1 - 10^-3).^h);
prob_erro_5 = (1 - (1 - 10^-2).^h);

prob_total_1 = prob_erro_1 * p1 ./ (prob_erro_1 * p1 + prob_erro_2 * p2 + prob_erro_3 *
p3 + prob_erro_4 * p4 + prob_erro_5 * p5);
prob_total_2 = prob_erro_2 * p2 ./ (prob_erro_1 * p1 + prob_erro_2 * p2 + prob_erro_3 *
p3 + prob_erro_4 * p4 + prob_erro_5 * p5);
prob_total_3 = prob_erro_3 * p3 ./ (prob_erro_1 * p1 + prob_erro_2 * p2 + prob_erro_3 *
p3 + prob_erro_4 * p4 + prob_erro_5 * p5);

prob_total = prob_total_1 + prob_total_2 + prob_total_3;

plot( h/8, prob_total, 'b-')
xlim([50, 214])

grid on
xlabel('Packet size (Bytes)')
title('Probability of the link being in the normal state')
```

Perante os resultados obtidos e sabendo que o *data frame* foi recebido com erros, observamos que, quanto maior for o *packet size*, maior a probabilidade de o link estar no estado normal. A probabilidade de estar no estado normal é bastante superior à de estar no estado de interferência. Perante este facto, o aumento da probabilidade de erro no estado normal aumenta com o número de *bytes* que é enviado, tal como no estado de interferência, sendo mais significativo ao ponto de aumentar a probabilidade do estado normal à medida que o número de *bytes* aumenta.

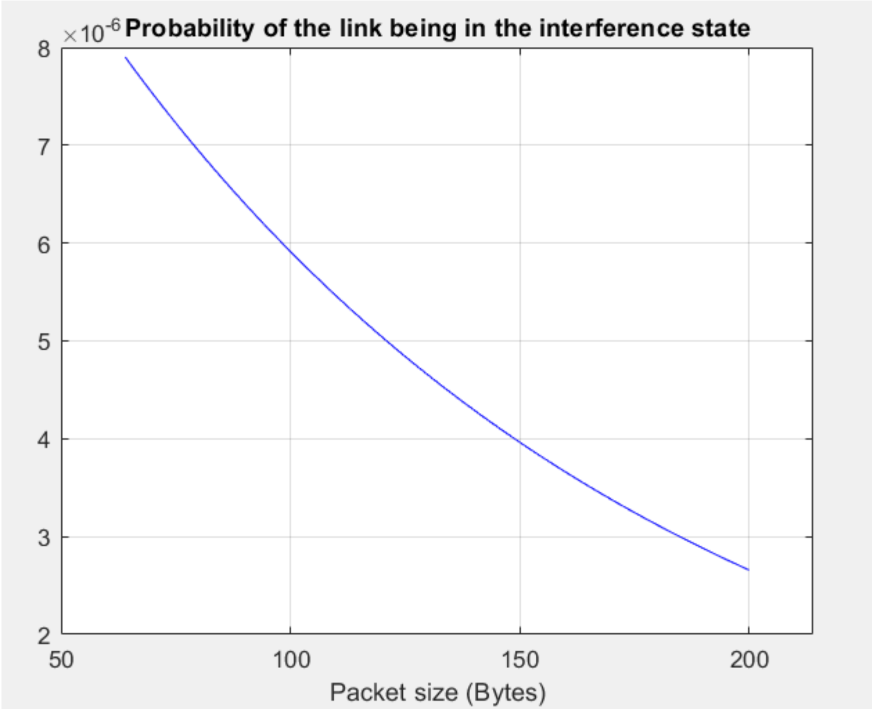


4.d. Consider now that a data frame of size B (in Bytes) sent by one of the stations is received without errors by the other station. Draw a plot of the probability of the link being in the interference state as a function of the packet size (from 64 Bytes up to 200 Bytes). Analyze and justify the results.

Para a resolução desta alínea começamos por calcular a probabilidade de o frame ser recebido sem erros em cada um dos estados, e depois, mais uma vez aplicamos a regra de Bayes para obter a probabilidade de o *data frame* estar no estado p4 e p5, tendo sido recebido sem erros.
Finalmente, somando as 2 probabilidades obtivemos a probabilidade de o *data frame* estar no estado de interferência, sendo recebido sem erros.

```
h = linspace(64*8, 200*8);

prob_n_erro_1 = (1 - 10^-6).^h;
prob_n_erro_2 = (1 - 10^-5).^h;
prob_n_erro_3 = (1 - 10^-4).^h;
prob_n_erro_4 = (1 - 10^-3).^h;
prob_n_erro_5 = (1 - 10^-2).^h;


prob_total_4 = prob_n_erro_4 * p4 ./ (prob_n_erro_1 * p1 + prob_n_erro_2 * p2 +
prob_n_erro_3 * p3 + prob_n_erro_4 * p4 + prob_n_erro_5 * p5);
prob_total_5 = prob_n_erro_5 * p5 ./ (prob_n_erro_1 * p1 + prob_n_erro_2 * p2 +
prob_n_erro_3 * p3 + prob_n_erro_4 * p4 + prob_n_erro_5 * p5);

prob_total = prob_total_4 + prob_total_5;

plot( h/8, prob_total, 'b-')
xlim([50, 214])

grid on
xlabel('Packet size (Bytes)')
title('Probability of the link being in the interference state')
```

Relativamente aos resultados obtidos, podemos concluir que à medida que mais pacotes vão sendo recebidos, a probabilidade de os mesmos estarem no estado de interferência, sabendo que foram recebidos sem erros vai diminuindo. Isto deve-se ao facto de a probabilidade de todos os pacotes serem recebidos sem erros no estado de interferência descer drasticamente à medida que aumentam o número de pacotes enviados, enquanto que no estado normal têm uma descida muito menos acentuada. Isto leva a que a probabilidade de estar no estado de interferência sendo o pacote recebido sem erros diminua à medida que o tamanho do pacote aumenta.



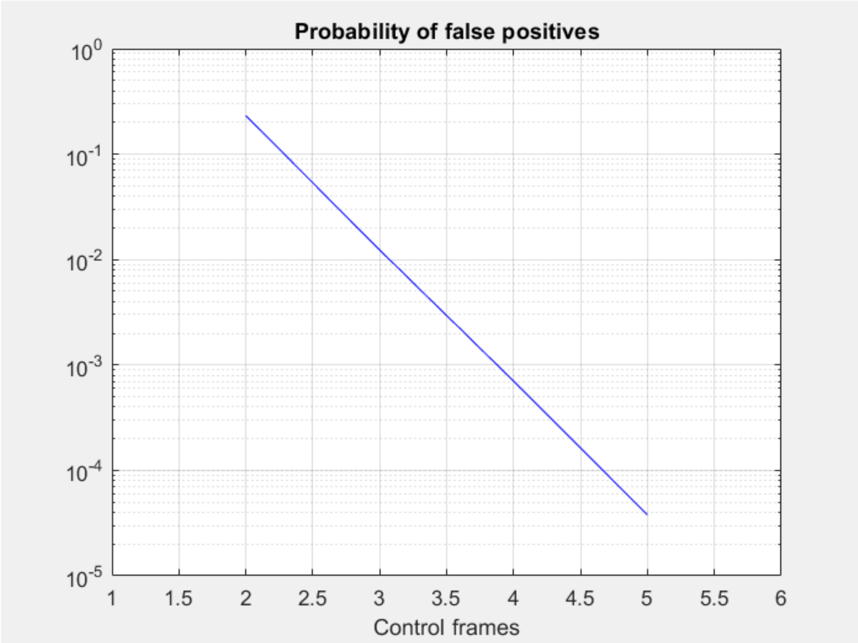
Task 5

5.a. Draw a plot using a logarithmic scale for the Y-axis (use the MATLAB function semilogy) with <u>the probability of false positives</u> for n = 2, 3, 4 and 5. Analyze and justify the results.

Neste exercício, aproveitamos o raciocínio da task 4, onde já tínhamos calculado a probabilidade de 1 *frame* possuir erros, apenas tivemos que adaptar para que fossem vários frames e que tivessem apenas 64 bytes.
Para isso, elevamos a probabilidade de erros por *frame* ao número de *frames* existentes (2 a 5). Após obtermos essa probabilidade para cada estado, voltamos a aplicar a regra de Bayes para os estados normais, e no final efetuamos a sua soma.

```
b = 64*8;
h = 2:5;

prob_erro_1 = (1 - (1 - 10^-6) ^ b) .^ h;
prob_erro_2 = (1 - (1 - 10^-5) ^ b) .^ h;
prob_erro_3 = (1 - (1 - 10^-4) ^ b) .^ h;
prob_erro_4 = (1 - (1 - 10^-3) ^ b) .^ h;
prob_erro_5 = (1 - (1 - 10^-2) ^ b) .^ h;

somatorio = 1 + 8/600 + 8/600 * 5/200 + 8/600 * 5/200 * 2/50 + 8/600 * 5/200 * 2/50 * 1/5;

p1 = 1 / somatorio;
p2 = 8/600 / somatorio;
p3 = 8/600 * 5/200 / somatorio;
p4 = 8/600 * 5/200 * 2/50 / somatorio;
p5 = 8/600 * 5/200 * 2/50 * 1/5 / somatorio;

prob_total_1 = prob_erro_1 * p1 ./ (prob_erro_1 * p1 + prob_erro_2 * p2 + prob_erro_3 *
p3 + prob_erro_4 * p4 + prob_erro_5 * p5);
prob_total_2 = prob_erro_2 * p2 ./ (prob_erro_1 * p1 + prob_erro_2 * p2 + prob_erro_3 *
p3 + prob_erro_4 * p4 + prob_erro_5 * p5);
prob_total_3 = prob_erro_3 * p3 ./ (prob_erro_1 * p1 + prob_erro_2 * p2 + prob_erro_3 *
p3 + prob_erro_4 * p4 + prob_erro_5 * p5);

prob_total = prob_total_1 + prob_total_2 + prob_total_3

semilogy( h, prob_total, 'b-')

grid on
xlabel('Control frames')
title('Probability of false positives')
xlim([1, 6])
```

Analisando os resultados, percebemos que a probabilidade de falsos positivos desce drasticamente com o aumento do número de *frames*. Isto deve-se ao facto de a probabilidade de todos os frames terem erros no estado normal descer muito em relação à probabilidade no estado de interferência que desce muito menos à medida que o número de *frames* aumenta.



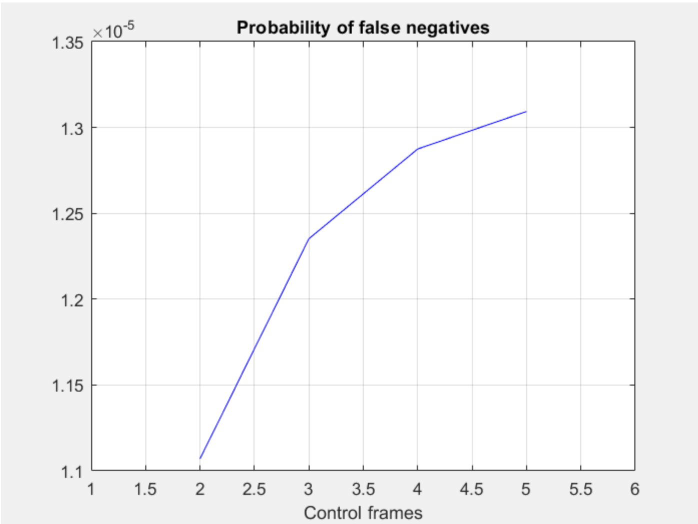
5.b. Draw a plot with the probability of false negatives for n = 2, 3, 4 and 5. Analyze and justify the results.

Para esta alínea, tivemos de calcular a probabilidade de em cada estado haver pelo menos um pacote sem erros (que é subtrair a probabilidade de o frame não ter erros ao total). Para finalizar aplicamos a regra de Bayes aos 2 estados de interferência e somamos o resultado.

```
b = 64*8;
h = 2:5;

prob_erro_1 = (1 - (1 - 10^-6) ^ b) .^ h;
prob_erro_2 = (1 - (1 - 10^-5) ^ b) .^ h;
prob_erro_3 = (1 - (1 - 10^-4) ^ b) .^ h;
prob_erro_4 = (1 - (1 - 10^-3) ^ b) .^ h;
prob_erro_5 = (1 - (1 - 10^-2) ^ b) .^ h;

prob_n_erros_1 = 1 - prob_erro_1;
prob_n_erros_2 = 1 - prob_erro_2;
prob_n_erros_3 = 1 - prob_erro_3;
prob_n_erros_4 = 1 - prob_erro_4;
prob_n_erros_5 = 1 - prob_erro_5;

somatorio = 1 + 8/600 + 8/600 * 5/200 + 8/600 * 5/200 * 2/50 + 8/600 * 5/200 * 2/50 * 1/5;

p1 = 1 / somatorio;
p2 = 8/600 / somatorio;
p3 = 8/600 * 5/200 / somatorio;
p4 = 8/600 * 5/200 * 2/50 / somatorio;
p5 = 8/600 * 5/200 * 2/50 * 1/5 / somatorio;
```

```
prob_total_4 = prob_n_erros_4 * p4 ./ (prob_n_erros_1 * p1 + prob_n_erros_2 * p2 +
prob_n_erros_3 * p3 + prob_n_erros_4 * p4 + prob_n_erros_5 * p5);
prob_total_5 = prob_n_erros_5 * p5 ./ (prob_n_erros_1 * p1 + prob_n_erros_2 * p2 +
prob_n_erros_3 * p3 + prob_n_erros_4 * p4 + prob_n_erros_5 * p5);

prob_total = prob_total_4 + prob_total_5


plot( h, prob_total, 'b-')
grid on
xlabel('Control frames')
title('Probability of false negatives')
xlim([1, 6])
```

Perante o gráfico obtido podemos concluir que a probabilidade de haver falsos negativos aumenta, ainda que pouco, consoante o número de frames enviados. Isto resulta do facto de, à medida que mais frames são enviados a probabilidade de pelo menos 1 *frame* ser recebido sem erros aumentar de forma mais significativa nos estados de interferência do que nos estados normais. Enquanto que nos estados normais essa probabilidade é sempre muito próxima de 1, nos estados de interferência com, por exemplo, 2 *frames* é muito baixa em comparação com os estados normais (chegando a ser abaixo de 0.1 para o estado p5). Com o aumento do número de *frames*, a probabilidade quase não sobe nos estados normais, pois já se encontrava muito próxima de 1, e sobe consideravelmente nos estados de interferência, levando a que a probabilidade de falsos positivos suba, pois sabendo que pelo menos 1 dos frames recebidos não tem erros, a probabilidade dos estados de interferência sobe consideravelmente.



5.c. Assume that the probabilities of false positives and false negatives are equally important in the accuracy of the interference detection system. From the plots obtained in **5.a** and **5.b**, determine the best value of n to be used by the system. Justify your answer.

Considerando que as probabilidades de falsos positivos e falsos negativos são igualmente importantes, podemos concluir que o melhor valor de *n* a usar é 5.
Comparando os valores obtidos na alínea **5.a** com os da alínea **5.b**, notamos que os valores da segunda alínea são menores do que todos os da primeira alínea. Logo, devemos definir o melhor *n* baseado nos resultados da primeira alínea. Concluímos então que o melhor valor para *n* é 5, pois apresenta a menor taxa de erro.

Pedro Gonçalves 88859
Pedro Silva 89228