universidade de aveiro
Departamento de Eletrónica, Telecomunicações e Informática

Desempenho e Dimensionamento de Redes

Trabalho Prático 1

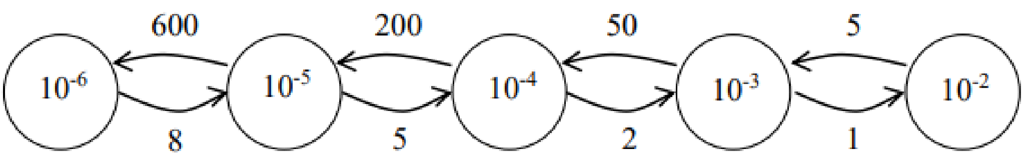
2020/2021

P1

Alunos:

Rodrigo Santos, 89180
Alexey Kononov, 89227

- **Task 4:**



a)

```
%4a
P1 = 1 / (1 + 8/600 + 8/600*5/200 + 8/600*5/200*2/50 + 8/600*5/200*2/50*1/5);

P2 = (8/600) / (1 + 8/600 + 8/600*5/200 + 8/600*5/200*2/50 + 8/600*5/200*2/50*1/5);


P3 = (8/600*5/200) / (1 + 8/600 + 8/600*5/200 + 8/600*5/200*2/50 + 8/600*5/200*2/50*1/5);


P4 = (8/600*5/200*2/50) / (1 + 8/600 + 8/600*5/200 + 8/600*5/200*2/50 + 8/600*5/200*2/50*1/5);


P5 = (8/600*5/200*2/50*1/5) / (1 + 8/600 + 8/600*5/200 + 8/600*5/200*2/50 + 8/600*5/200*2/50*1/5);

P_normal = (P1 + P2 + P3);
P_interf = P4 + P5;
```

Começamos por calcular a probabilidade de estar em cada um dos 4 estados, usando as fórmulas :

$$\pi_0 = \frac{1}{1 + \sum_{i=1}^{\infty} \frac{\lambda_0 \lambda_1 \cdots \lambda_{i-1}}{\mu_1 \mu_2 \cdots \mu_i}}$$

e

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

Somamos as probabilidades de estar nos primeiros três estados, em que o ber é inferior a 10-3, para calcular a probabilidade de estar no estado normal.
A soma dos outros 2 estados deu a probabilidade de estar no estado de interferência.

Após cálculo dos valores temos:

- Probabilidade de estar no estado normal (*P_normal*) = 0.9999754
- Probabilidade de estar no estado de interferência (*P_interf*) = 0,0000158

b)

```
%4b

Av_ber_normal = (P1*1e-6 + P2*1e-5 + P3*1e-4)/P_normal
Av_ber_interf = (P4*1e-3 + P5*1e-2)/P_interf
```

Após cálculo dos valores temos:

- ber médio no estado normal (*Av_ber_normal*) = 1.1509e-06
- ber médio no estado de interferência (*Av_ber_interf*) = 2.5e-3

c)

Para calcular a probabilidade de estar no estado normal sabendo que o pacote foi recebido com erros:

P(N | E) = P(S1 | E) + P(S2 | E) + P(S3 | E)

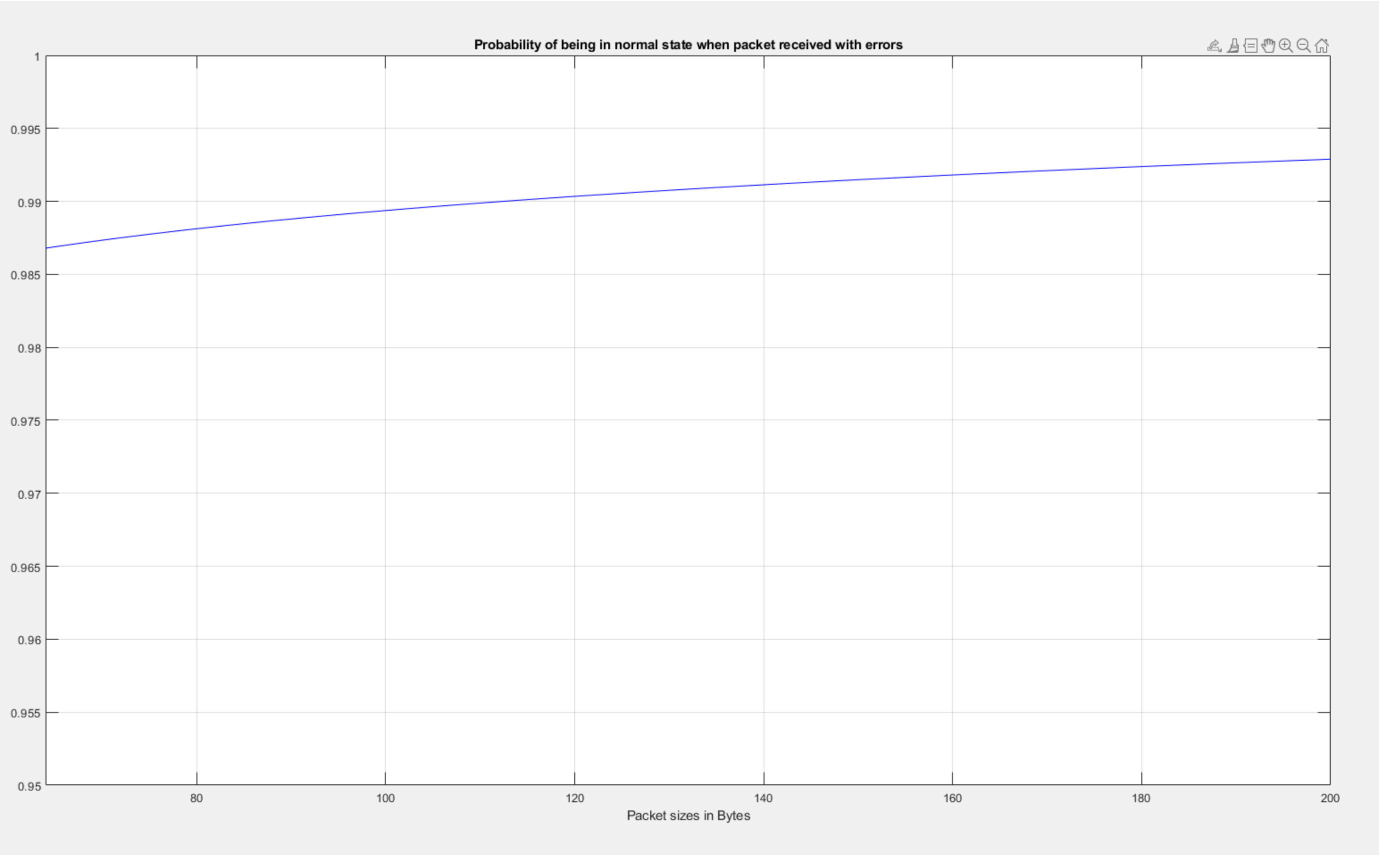
que é dado, utilizando a regra de Bayes, por:

P(N | E) = (P(S1) * P(E | S1) + P(S2) * P(E | S2) + P(S3) * P(E | S3)) / (P(S1) * P(E | S1) + P(S2) * P(E | S2) + P(S3) * P(E | S3) + P(S4) * P(E | S4) + P(S5) * P(E | S5))

```
%4c
states_prob = [0.9865020 0.0131534 0.0003288 0.0000132 0.0000026];
ber = [1e-6 1e-5 1e-4 1e-3 1e-2];
packet_Bytes = 64:200;

numerador = 0;
for i=1:3
    numerador = numerador + states_prob(i) * (1 - (1-ber(i)) .^ (packet_Bytes*8));
end
denominador = numerador;

for i=4:5
    denominador = denominador + states_prob(i) * (1 - (1-ber(i)) .^ (packet_Bytes*8));
end
y = numerador ./ denominador;

figure(1)
plot(packet_Bytes , y, 'b' )
title("Probability of being in normal state when packet received with errors")
xlabel("Packet sizes in Bytes")
grid on
axis([64 200 0.95 1])
```

Criamos um vetor com as probabilidades de cada estado - *states_prob*, outro vetor com os valores de ber de cada estado - *ber* e o vetor *packet_Bytes* com o número de bits dos pacotes com tamanho de 64 a 200 Bytes.
Para calcular P(Erros | EstadoN) foi usada a fórmula: 1 - probabilidade binomial de não ter erros, ou seja, 1 - $\binom{n}{i} p^i (1-p)^{n-i}$, em que n=bytes*8, p = ber, i=0 (0 erros).
Os erros em diferentes bits são estatisticamente independentes. O número de erros num pacote é uma variável aleatória binomial em que a probabilidade de sucesso é o BER e o número de experiências de Bernoulli é o número de bits do pacote.

Efetuados os cálculos, fazendo o seu plot chegamos ao seguinte resultado da probabilidade em função do tamanho do pacote:



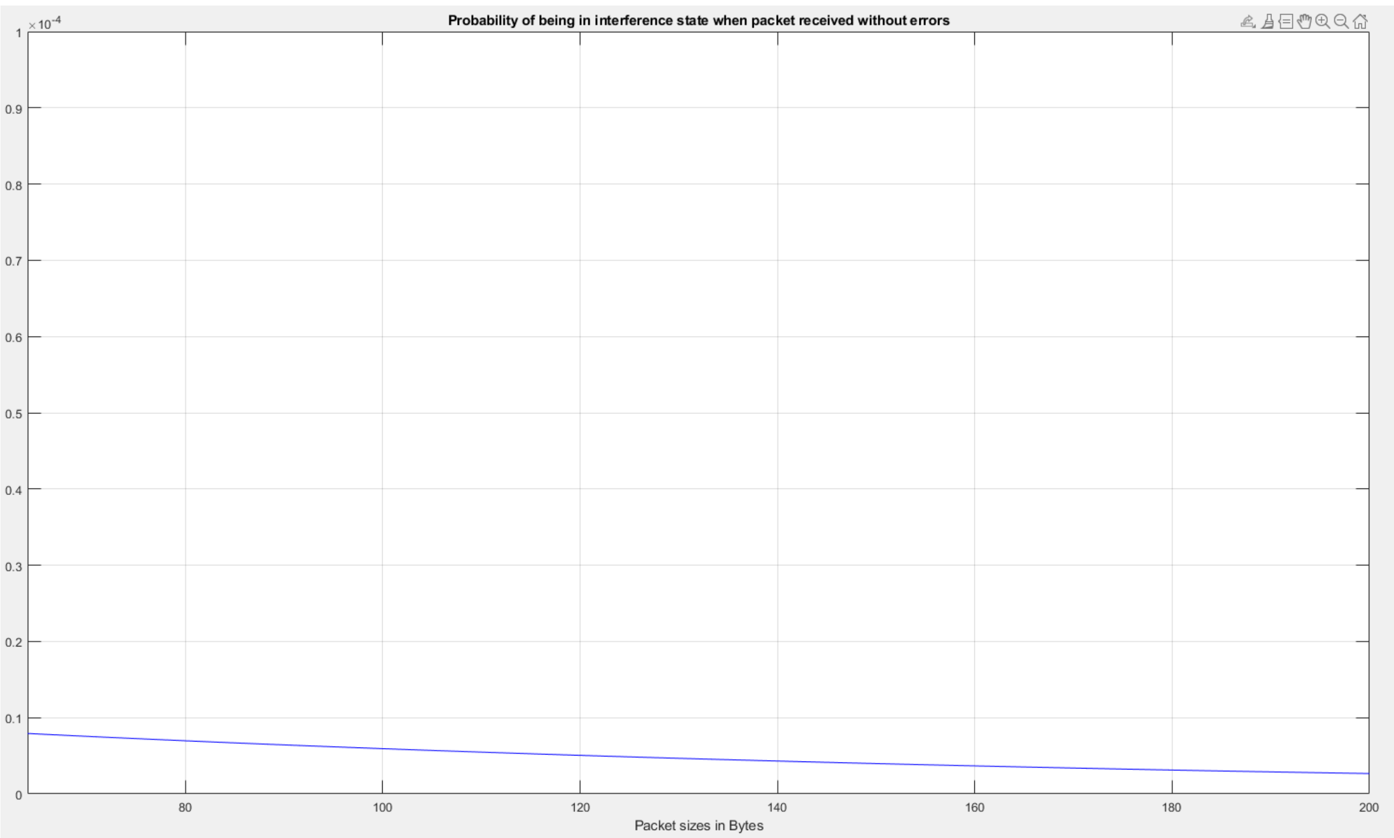
Da análise do gráfico verificamos que com o crescimento do tamanho dos pacotes a probabilidade de a rede se encontrar no estado normal quando um pacote é recebido com erros é ligeiramente maior, algo que é natural, dado que há medida que são enviados mais bits, há mais bits com a probabilidade de estarem errados.

d)

Para calcular a probabilidade de estar no estado de interferência sabendo que o pacote foi recebido sem erros, segue-se um procedimento semelhante ao procedimento da alínea anterior:

```
%4d
states_prob = [0.9865020 0.0131534 0.0003288 0.0000132 0.0000026];
ber = [1e-6 1e-5 1e-4 1e-3 1e-2];
packet_Bytes = 64:200;

numerador = 0;
for i=4:5
    numerador = numerador + states_prob(i) * ((1-ber(i)) .^ (packet_Bytes*8));
end
denominador = numerador;

for i=1:3
    denominador = denominador + states_prob(i) * ((1-ber(i)) .^ (packet_Bytes*8));
end
y = numerador ./ denominador;

figure(2)
plot(packet_Bytes , y, 'b' )
title("Probability of being in interference state when packet received without errors")
xlabel("Packet sizes in Bytes")
grid on
axis([64 200 0 0.0001])
```

Após realizados os cálculos, fazendo o *plot* destes, chegamos ao seguinte resultado da probabilidade em função do tamanho do pacote:



Da análise do gráfico verificamos que com o crescimento do tamanho dos pacotes, a probabilidade de a rede se encontrar no estado de interferência quando um pacote é recebido sem erros é ligeiramente menor e muito próxima de 0, isto é explicado pelo facto de que as taxas de transferência de estados com ber mais alto (interferência) para os estados com ber mais baixo (normal) são muito superiores. Como os pacotes têm mais bits, é mais provável ter estes serem recebidos com erros e sabendo que isto não se verifica, ou seja os pacotes foram recebidos sem erros, a probabilidade de estar num estado de interferência onde o bit error rate é maior, é menor.

- ## Task 5:

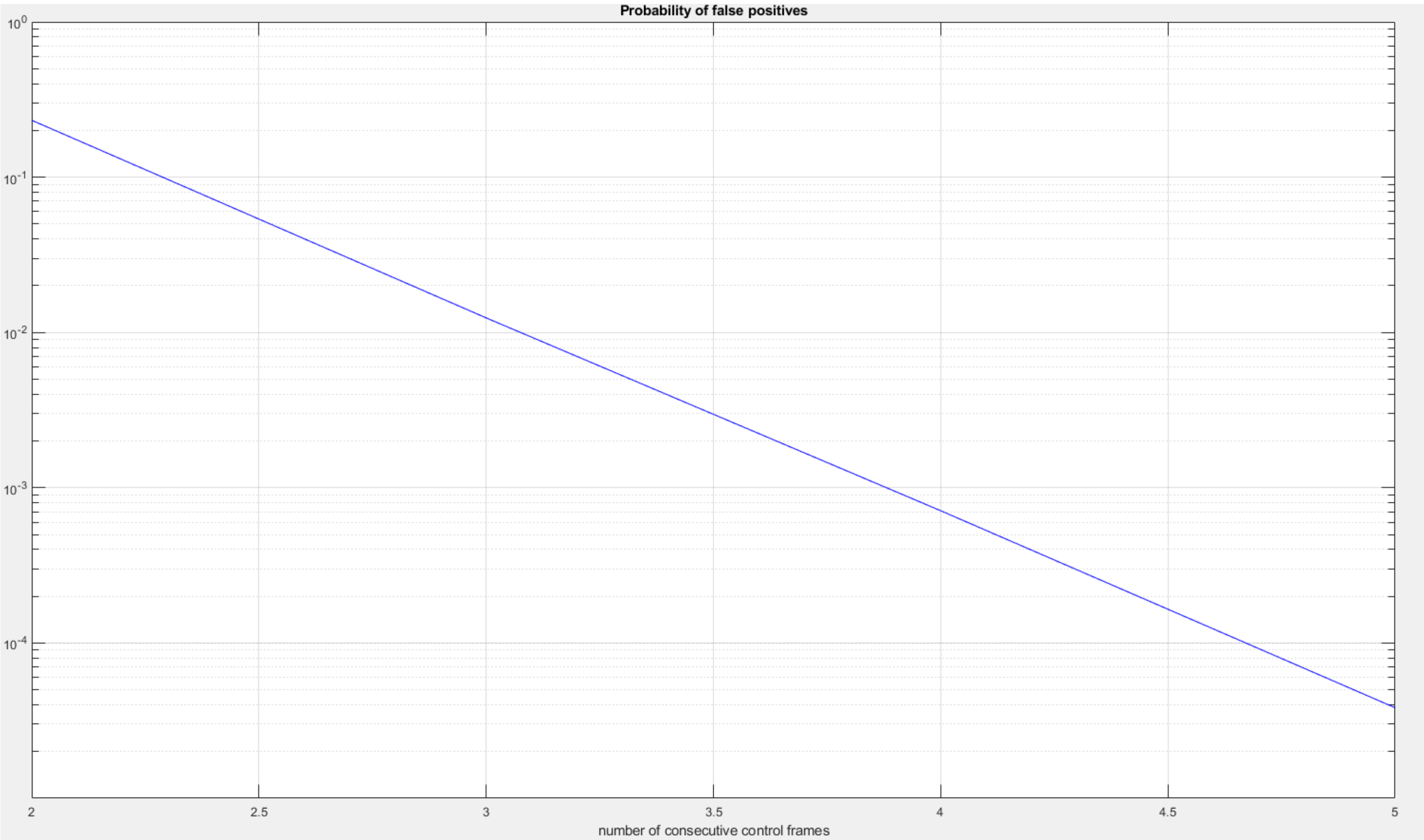
a)

```
% 5a
states_prob = [0.9865020 0.0131534 0.0003288 0.0000132 0.0000026];
ber = [1e-6 1e-5 1e-4 1e-3 1e-2];
packet_size = 64*8;
n = 2:5;
normal_state_with_errors = zeros(size(1:3));
for i=1:3
    normal_state_with_errors(i) = 1 - ( 1*(ber(i))^0 * (1-(ber(i)))^(packet_size-0) );
end

interf_state_errors = zeros(size(4:5));
for i=4:5
    interf_state_errors(i) = 1 - ( 1*(ber(i))^0 * (1-(ber(i)))^(packet_size-0) );
end

prob_false_positive = zeros(size(2:5));
for i = n
    prob6 = normal_state_with_errors(1) ^ i;
    prob5 = normal_state_with_errors(2) ^ i;
    prob4 = normal_state_with_errors(3) ^ i;
    prob3 = interf_state_errors(4)^ i;
    prob2 = interf_state_errors(5)^ i;
    den = (prob6 * states_prob(1) + prob5 * states_prob(2) + prob4 * states_prob(3) + prob3 * states_prob(4) + prob2 * states_prob(5));
    prob_false_positive(i-1) = (prob6 * states_prob(1) + prob5 * states_prob(2) + prob4 * states_prob(3)) / den;
end

figure(1)
semilogy(n,prob_false_positive, 'b')
title("Probability of false positives")
xlabel("number of consecutive control frames")
grid on
axis([2 5 1e-5 1])
```

Para calcular a probabilidade de falsos positivos, inicialmente começamos por calcular a probabilidade de os control frames chegarem com erros, para cada estado da cadeia de Markov, armazenando esses valores nas listas *normal_state_with_errors* e *interf_state_errors.*
Usamos a fórmula de probabilidade binomial como no exercício anterior.

De seguida, calculamos a probabilidade de *n* control frames seguidos serem recebidos com erros elevando a probabilidade ao número de frames. Por fim é calculado o valor da probabilidade de falsos positivos que é equivalente a P(E | N).

Colocando os resultados obtidos num gráfico obtemos o seguinte resultado:



Da observação do gráfico resultante, podemos concluir que quanto maior for o número de control frames, menor é a probabilidade de falsos positivos na rede, sendo que esta decresce aproximadamente com um fator de 10 vezes, a cada *control frame* adicionado.

b)

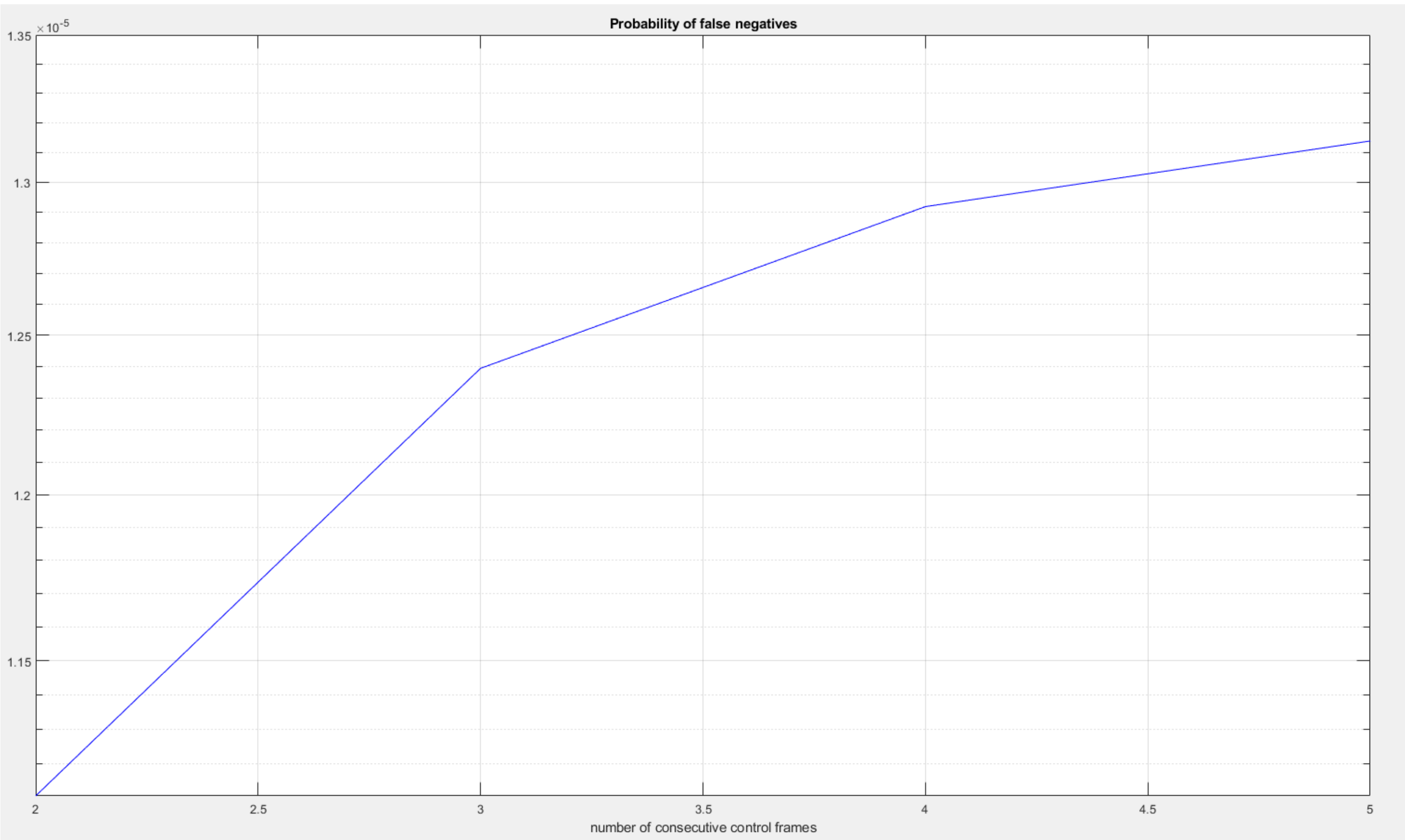
Nesta alínea seguimos o raciocínio semelhante ao da alínea anterior, começando por calcular a probabilidade de os control frames chegarem a cada estado sem erros e armazenar as probabilidades correspondentes a cada estado (normal e interferência) em vetores respetivos *normal_state_no_errors* e *interf_state_no_errors*.

Depois calculamos a probabilidade de cada estado receber pelo menos um control frame sem erro.

Depois usando a regra de Bayes calculamos a probabilidade de falsos negativos que é P(~Erros | Interferência) .

```matlab
% 5b
states_prob = [0.9865020 0.0131534 0.0003288 0.0000132 0.0000026];
ber = [1e-6 1e-5 1e-4 1e-3 1e-2];
packet_size = 64*8;
n = 2:5;
normal_state_no_errors = zeros(size(1:3));
for i=1:3
    normal_state_no_errors(i) = ( 1*(ber(i))^0 * (1-(ber(i)))^(packet_size-0) );
end

interf_state_no_errors = zeros(size(4:5));
for i=4:5
    interf_state_no_errors(i) = ( 1*(ber(i))^0 * (1-(ber(i)))^(packet_size-0) );
end

prob_false_negative = zeros(size(2:5));
for i = n
    prob6 = 1- ((1- normal_state_no_errors(1)) ^ i);
    prob5 = 1- ((1- normal_state_no_errors(2)) ^ i);
    prob4 = 1- ((1- normal_state_no_errors(3)) ^ i);
    prob3 = 1- ((1- interf_state_no_errors(4)) ^ i);
    prob2 = 1- ((1- interf_state_no_errors(5)) ^ i);
    den = (prob6 * states_prob(1) + prob5 * states_prob(2) + prob4 * states_prob(3) + prob3 * states_prob(4) + prob2 * states_prob(5));
    prob_false_negative(i-1) = (prob3 * states_prob(4) + prob2 * states_prob(5)) / den;
end

figure(2)
semilogy(n,prob_false_negative, 'b')
ylim([0,0.0000135])
title("Probability of false negatives")
xlabel("number of consecutive control frames")
grid on
```

Colocando os resultados obtidos num gráfico obtemos o seguinte resultado:



Após observação do gráfico concluímos que quanto maior for o número de frames de controlo tanto maior a probabilidade de falsos negativos. Como a frequência de pacotes sem erros é maior, o aumento do número de frames de controlo aumenta bastante a probabilidade de receber um pacote sem erros e concluir erradamente estar no estado normal.

c)

Tendo em consideração os resultados obtidos e a análise dos gráficos das alíneas anteriores, o melhor valor de *n* relativo ao número de control frames utilizados é 5, uma vez que a probabilidade de falsos positivos diminui uma ordem de grandeza a cada incremento de *n,* logo quanto maior for *n,* mais fiável será a rede, e a ordem de grandeza da probabilidade de falsos negativos é para os valores de *n* de 2 a 5*,* sempre 10^-5, algo que é bastante baixo e não altera significativamente com o aumento de control frames. Assim, podemos concluir que quanto maior for o número de control frames mais preciso será o sistema de detecção de interferência e o desempenho da rede.