

# D.D.S.R

---

## **Relatório nº4 - Routing in packet switched and circuit switcher networks**

**Curso:**METI

**Turno:** 3ª feira 15:00 - 16:30)

**Grupo:** 8

**Trabalho realizado por:**

Ruben Condesso, nº 81969

André Mendes, nº78079

## Exercício 1:

A rede em causa neste exercício, pode ser ilustrada pela seguinte figura:

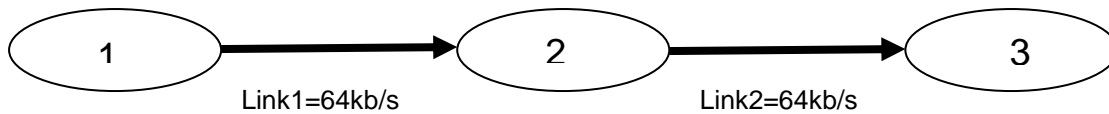


Figura 1: Topologia da rede

Cada *flow*, chega com pacotes de tamanho exponencialmente distribuídos, de acordo com um processo de *Poisson*, cujo tamanho médio de cada pacote é 1000 *bits*. O atraso médio de cada pacote, segundo a aproximação de *Kleinrock*, é dada por:  $W = \frac{1}{\frac{c}{L} + \lambda} + \frac{1}{\frac{c}{L} + \lambda}$ . Calculou-se para 4 valores distintos de  $\rho$  (0.05;0.5;0.95;0.975).

De seguida, usando o *pnet* simulou-se o atraso médio de cada pacote, com o intuito de comparar com o valor dado pela aproximação de *Kleinrock*, desta forma, foram usados os mesmos valores de  $\rho$ . Dado os valores de  $\rho$ , e sabendo que  $\rho = \lambda / \mu$ ;  $\mu = \frac{c}{L_p}$ , os valores de  $\lambda$  considerados foram os seguintes: 3.2; 32; 60.8; 62.4. Posto isto, obtivemos os seguintes resultados:

```
>> ex1(100, 0.05)
For lambda =
    3.2000

Kleinrock Average Delay
    0.0329

Para ro =
    0.0500

The average packet delay through pnet simulator:
    0.0331

-----
The average packet delay through the kleinrock approximation:
    0.0329
```

Figura 2: Comparação do resultado de kleinrock com o resultado obtido usando o pnet para:100 simulações;  $\rho=0.05$ ;

```
>> ex1(100, 0.5)
For lambda =
    32

Kleinrock Average Delay
    0.0625

Para ro =
    0.5000

The average packet delay through pnet simulator:
    0.0622

-----
The average packet delay through the kleinrock approximation:
    0.0625
```

Figura 3: Comparação do resultado de kleinrock com o resultado obtido usando o pnet para:100 simulações;  $\rho=0.5$ ;

```
>> ex1(100, 0.95)
For lambda =
    60.8000
```

```
Kleinrock Average Delay
    0.6250
```

```
Para ro =
    0.9500
```

```
The average packet delay through pnet simulator:
    0.5479
```

```
-----
The average packet delay through the kleinrock approximation:
    0.6250
```

```
>> ex1(100, 0.975)
For lambda =
    62.4000
```

```
Kleinrock Average Delay
    1.2500
```

```
Para ro =
    0.9750
```

```
The average packet delay through pnet simulator:
    0.7906
```

```
-----
The average packet delay through the kleinrock approximation:
    1.2500
```

Figura 4: Comparação do resultado de kleinrock com o resultado obtido usando o pnet para:100 simulações;  $\rho=0.95$ ;

Figura 5: Comparação do resultado de kleinrock com o resultado obtido usando o pnet para:100 simulações;  $\rho=0.975$ ;

Pelas figuras ilustradas acima, podemos ver como varia a diferença entre o valor obtido usando o *pnet simulator*, comparativamente com o valor calculado segundo a aproximação de *Kleinrock*. Podemos concluir que à medida que o  $\rho$  vai aumentando, estes dois valores vão ficando cada vez mais dispersos um do outro, contudo para  $\rho=0.05$  e  $\rho=0.5$ , há uma grande aproximação de resultados. Tal já não acontece com  $\rho=0.95$  e  $\rho=0.975$ .

Podemos então estimar que para  $0 < \rho < 0.5$ , a aproximação de *Kleinrock* é adequada à situação em causa.

## Exercício 2:

Podemos esquematizar a rede deste exercício da seguinte forma:

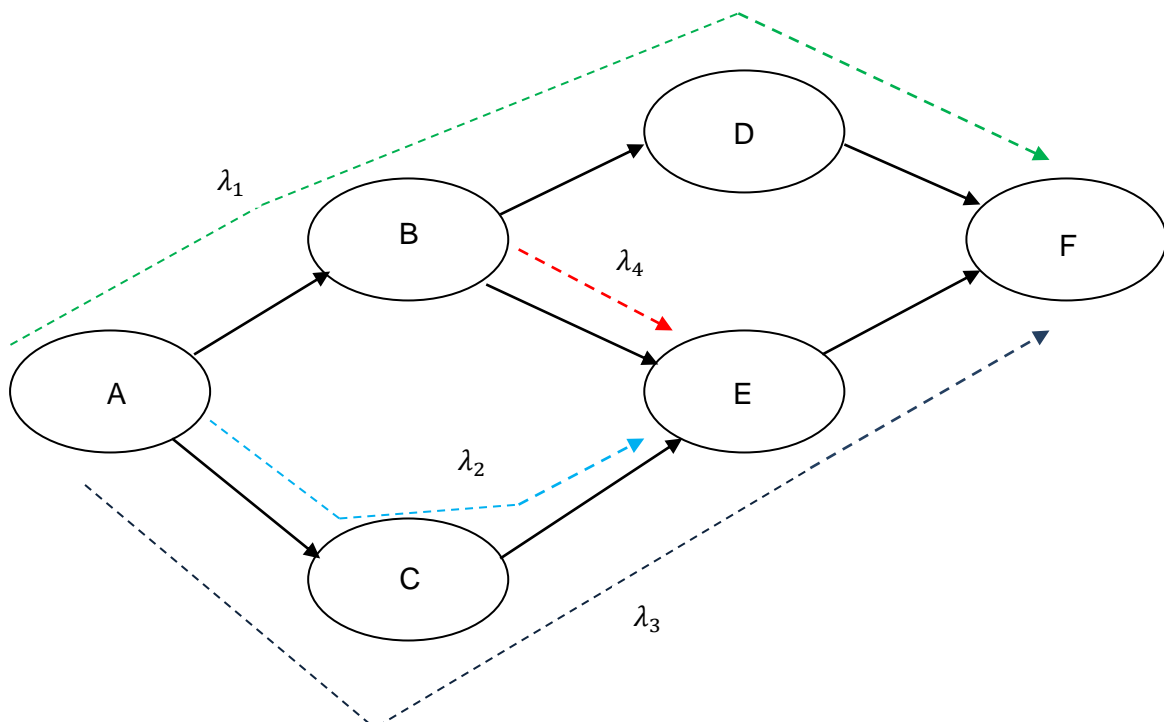


Figura 6: Topologia da rede do exercício 2, com o trajeto de cada *flow* existente;

**(2.1)** Numa ligação ponto-a-ponto, segundo a *Kleinrock approximation*, o atraso médio de um pacote é dado por:  $W = \frac{1}{\mu - \lambda}$ ; e tendo em conta que  $\mu = \frac{C}{L_p} = \frac{256k}{1000} = 256$ , calculamos o atraso médio de um pacote para cada fluxo  $i$  existente na rede:

- Para o fluxo 1:  $W_1 = \frac{1}{\mu - \lambda_1} + \frac{1}{\mu - \lambda_1} + \frac{1}{\mu - \lambda_1} = \frac{3}{256 - 215} = 0.07317$ ;
- Para o fluxo 2:  $W_2 = \frac{1}{\mu - (\lambda_2 + \lambda_3)} + \frac{1}{\mu - (\lambda_2 + \lambda_3)} = \frac{2}{256 - (64 + 128)} = 0.03125$ ;
- Para o fluxo 3:  $W_3 = W_2 + \frac{1}{\mu - \lambda_3} = 0.03125 + \frac{1}{256 - 128} = 0.03906$ ;
- Para o fluxo 4:  $W_4 = 1/(\mu - \lambda_4) = \frac{1}{256 - 128} = 0.00781$ ;

O atraso médio dos pacotes da rede é dada por:  $W = W_1 \left( \frac{\lambda_1}{\sum_{i=1}^4 \lambda_i} \right) + W_2 \left( \frac{\lambda_2}{\sum_{i=1}^4 \lambda_i} \right) + W_3 \left( \frac{\lambda_3}{\sum_{i=1}^4 \lambda_i} \right) + W_4 \left( \frac{\lambda_4}{\sum_{i=1}^4 \lambda_i} \right) = 0.07317 \left( \frac{215}{535} \right) + 0.03125 \left( \frac{64}{535} \right) + 0.03901 \left( \frac{128}{535} \right) + 0.00781 \left( \frac{128}{535} \right) = 0.044194$ .

**(2.2)** Usando o *pnet simulator*, chegamos aos seguintes resultados:

```
>> ex2(10)
The average packet delay of flow1 through pnet simulator:
0.0559

The average packet delay of flow2 through pnet simulator:
0.0275

The average packet delay of flow3 through pnet simulator:
0.0351

The average packet delay of flow4 through pnet simulator:
0.0078

The average packet delay of the network through pnet simulator:
0.0360
```

Figura 7: Resultado obtido usando o pnet para cada *flow*, para 10 simulações;

Podemos verificar pela figura 6 que existe uma boa aproximação entre os valores calculados, segundo *Kleinrock simulation*, com os valores gerados usando o *pnet simulator*. Sendo assim, podemos confirmar que a *Kleinrock simulation* é um método adequado para calcular o atraso médio dos pacotes, considerando a topologia da rede em causa.

**(2.3)** Considerando agora que o *flow1* sofreu uma bifurcação, o mesmo passará a desenrolar-se da seguinte forma:

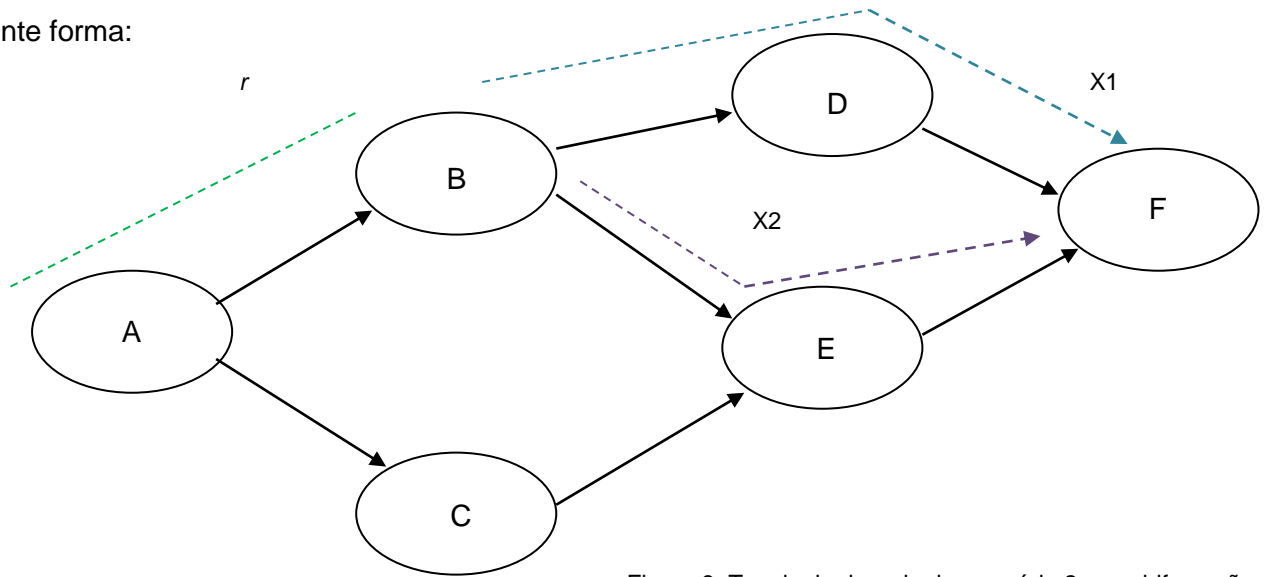


Figura 8: Topologia da rede do exercício 2, com bifurcação no *flow1*;

Tendo em conta que todas as ligações têm igual capacidade, considerando uma bifurcação ótima, podemos admitir o seguinte:

- $128 + X2 = X1$ ;
- $r = 215 = X1 + X2$ ,  $X1 \text{ \& } X2 > 0$ ;
- $\Rightarrow X1 = 171.5$ ;  $X2 = 43.5$ ;

Posto isto, a topologia da rede considerando uma bifurcação ótima será a seguinte:

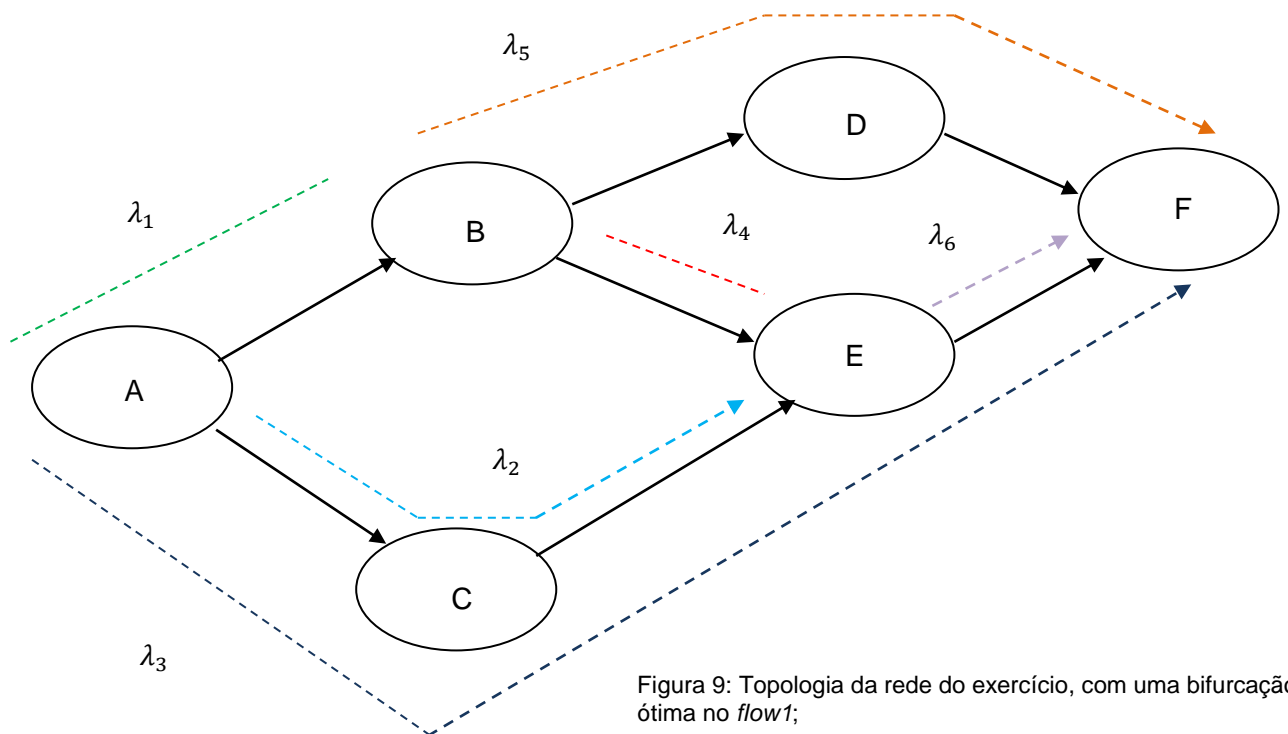


Figura 9: Topologia da rede do exercício, com uma bifurcação ótima no *flow1*;

Com  $\lambda_1 = 215k$ ;  $\lambda_2 = 64k$ ;  $\lambda_3 = 128k$ ;  $\lambda_4 = 171.5k$ ;  $\lambda_5 = 171.5k$ ;  $\lambda_6 = 43.5k$ ;

O atraso médio no sistema foi calculado efetuando a média dos atrasos médios dos fluxos considerados, onde obtivemos os seguintes resultados:

```
>> ex2_2(10)
The average packet delay of flow1 through pnet simulator:
0.0232

The average packet delay of flow2 through pnet simulator:
0.0278

The average packet delay of flow3 through pnet simulator:
0.0394

The average packet delay of flow4 through pnet simulator:
0.0119

The average packet delay of flow5 through pnet simulator:
0.0141

The average packet delay of flow6 through pnet simulator:
0.0129

The average packet delay of the network through pnet simulator:
0.0216
```

Figura 10: Resultado obtido usando o pnet para cada *flow*, usando bifurcação no *flow1*, para:10 simulações;

A partir da figura 10, e em comparação com a figura 7, podemos concluir que o atraso médio dos pacotes é mais baixa onde houve bifurcação, como já seria expectável. O atraso médio do *flow1*, no caso onde houve bifurcação é cerca de metade comparativamente com o caso onde não existiu, o que também era esperado sendo que o *flow1* sofreu uma divisão por duas rotas distintas, no *router B*.

O *flow2* e o *flow3* não são afetados pela bifurcação do *flow1*, e logo podemos observar que o atraso médio dos pacotes, para os respetivo *flows*, é praticamente idêntico nos dois casos. Relativamente, aos valores do *flow4* e do *flow5* da figura 7, e aos valores do *flow4*, *flow5* e do *flow6* da figura 10, e tendo em conta que estes *flows* correspondem à mesma zona da mesma topologia, podemos verificar que na figura 10 (onde há bifurcação) existe um maior equilíbrio no que diz respeito ao atraso médio dos pacotes para cada *flow*, considerando, está claro, que a soma do atraso médio dos pacotes nessa zona da topologia é, aproximadamente, igual nos dois casos.

Considerando a *Kleinrock simulation*, podemos concluir que existe uma maior aproximação de valores para o caso onde não existe bifurcação, sendo que comparativamente ao caso onde existe bifurcação, o valor do *Kleinrock simulation* é cerca do dobro.

### Exercício 3:

A topologia da rede referente a este exercício é a seguinte:

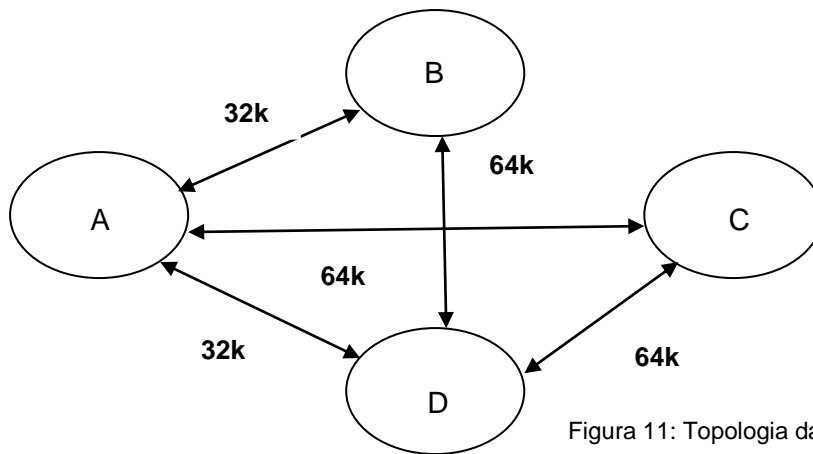


Figura 11: Topologia da rede do exercício 3;

**(3.1)** Para o cálculo das equações de balanço do processo em causa, construi-se o seguinte diagrama *birth-death* que representa a ocupação da rede:

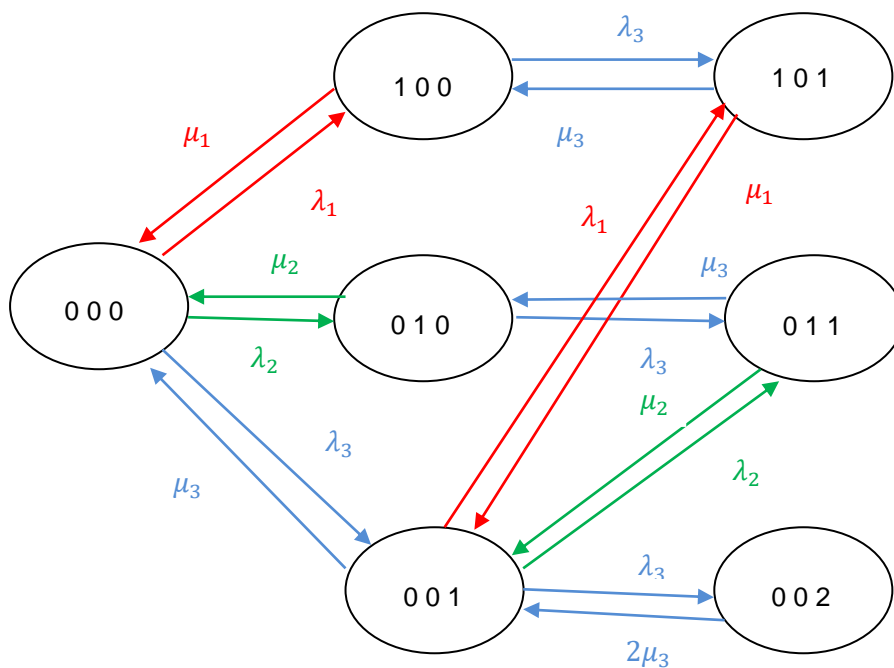


Figura 12 : Diagrama *birth-death* que representa a ocupação da rede em causa

Com o auxílio do diagrama, podemos afirmar que as equações de balanço para este processo são:

$$\Pi_{000} + \Pi_{100} + \Pi_{101} + \Pi_{010} + \Pi_{011} + \Pi_{001} + \Pi_{002} = 1;$$

$$\Pi_{000} * (\lambda_1 + \lambda_2 + \lambda_3) = \Pi_{100} * \mu_1 + \Pi_{010} * \mu_2 + \Pi_{001} * \mu_3;$$

$$\Pi_{100} * (\lambda_1 + \mu_1) = \Pi_{101} * \mu_3 + \Pi_{000} * \lambda_1;$$

$$\Pi_{010} * (\lambda_3 + \mu_3) = \Pi_{000} * \lambda_2 + \Pi_{011} * \mu_3;$$

$$\Pi_{001} * (\lambda_1 + \lambda_2 + \lambda_3 + \mu_3) = \Pi_{000} * \lambda_3 + \Pi_{101} * \mu_1 + \Pi_{002} * 2\mu_3;$$

$$\Pi_{101} * (\mu_1 + \mu_3) = \Pi_{100} * \lambda_3 + \Pi_{001} * \lambda_1;$$

$$\Pi_{011} * (\mu_2 + \mu_3) = \Pi_{001} * \lambda_2 + \Pi_{010} * \lambda_3;$$

$$\Pi_{002} * (2\mu_3) = \Pi_{001} * \lambda_3;$$

Recorrendo ao *Matlab*, para resolver as equações apresentadas, chegamos aos seguintes valores:

$$\Pi_{000} = 0.009;$$

$$\Pi_{100} = 0.0271;$$

$$\Pi_{010} = 0.0271;$$

$$\Pi_{001} = 0.0814;$$

$$\Pi_{101} = 0.2443;$$

$$\Pi_{011} = 0.2443;$$

$$\Pi_{002} = 0.3665;$$

Relativamente às probabilidades de bloqueio do *flow1*, *flow2* e *flow3*, são dadas pelas seguintes equações:

$$Flow1 : \Pi_{100} + \Pi_{101} + \Pi_{010} + \Pi_{011} + \Pi_{002};$$

$$Flow2 : \Pi_{100} + \Pi_{101} + \Pi_{010} + \Pi_{011} + \Pi_{002};$$

$$Flow3 : \Pi_{011} + \Pi_{101} + \Pi_{002};$$

Posto isto, chegamos aos seguintes resultados:

$$Flow1 : 0.9095; Flow2 : 0.9095; Flow3 : 0.8552;$$

Recorrendo ao *Matnet*, de forma a confirmar os resultados apresentados acima, obtivemos os seguintes resultados:



# RELATÓRIO

Probabilidade de bloqueio da classe1 :

0.9095

Tempo de cálculo :

0 h : 0 min : 0.41019 s

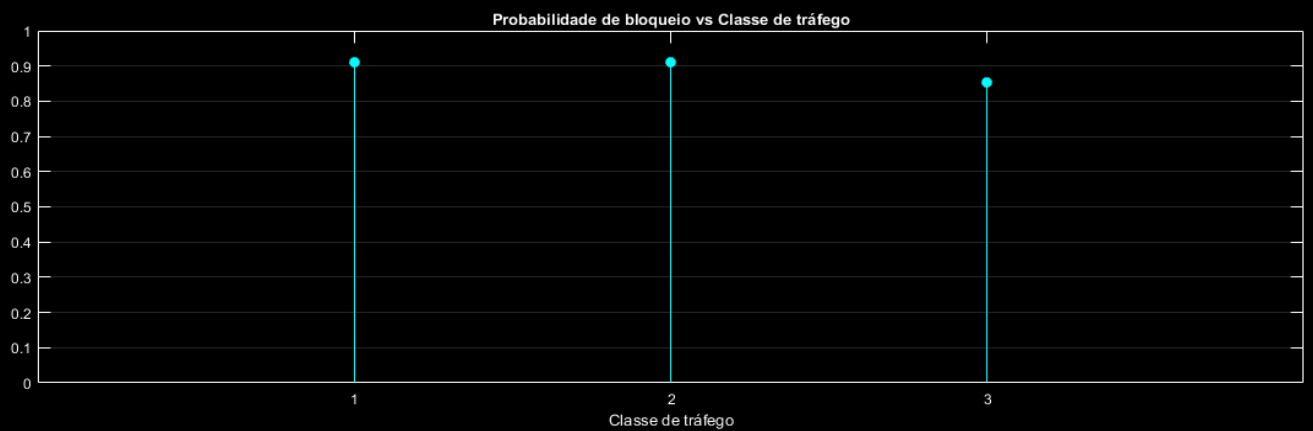
Selecionar classe

Classe 1

Número de estados :

7

## REPRESENTAÇÃO GRÁFICA



# RELATÓRIO

Probabilidade de bloqueio da classe2 :

0.9095

Tempo de cálculo :

0 h : 0 min : 0.41019 s

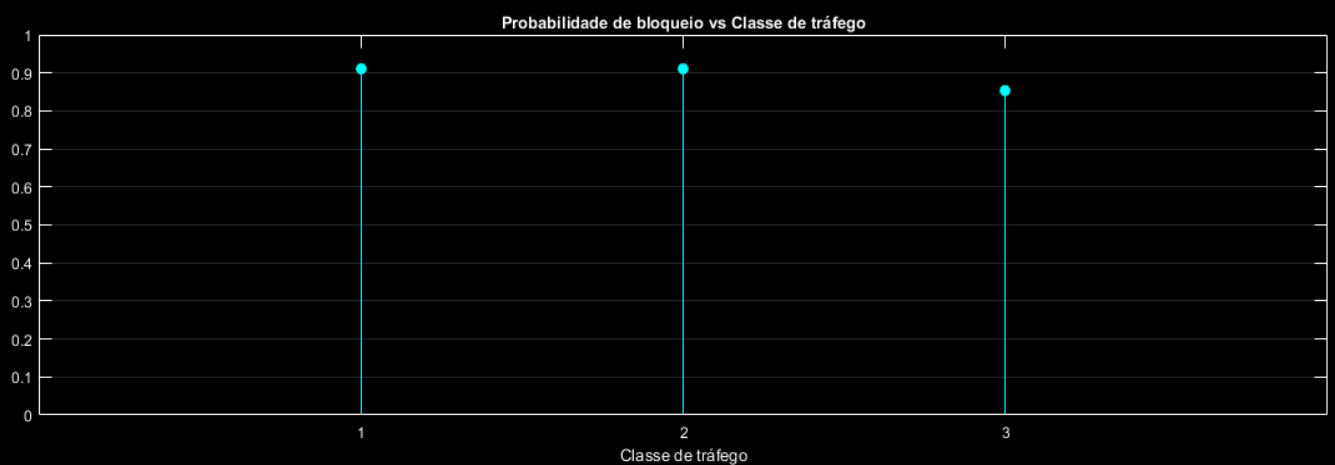
Selecionar classe

Classe 2

Número de estados :

7

## REPRESENTAÇÃO GRÁFICA



# RELATÓRIO

Probabilidade de bloqueio da classe3 :

0.8552

Tempo de cálculo :

0 h : 0 min : 0.41019 s

Selecionar classe

Classe 3

Número de estados :

7

## REPRESENTAÇÃO GRÁFICA

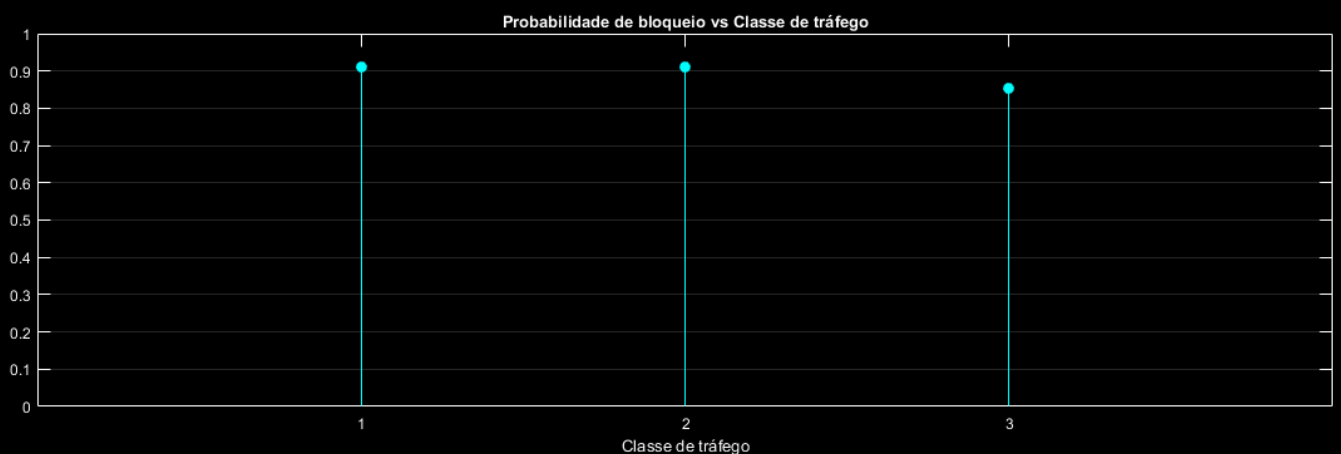
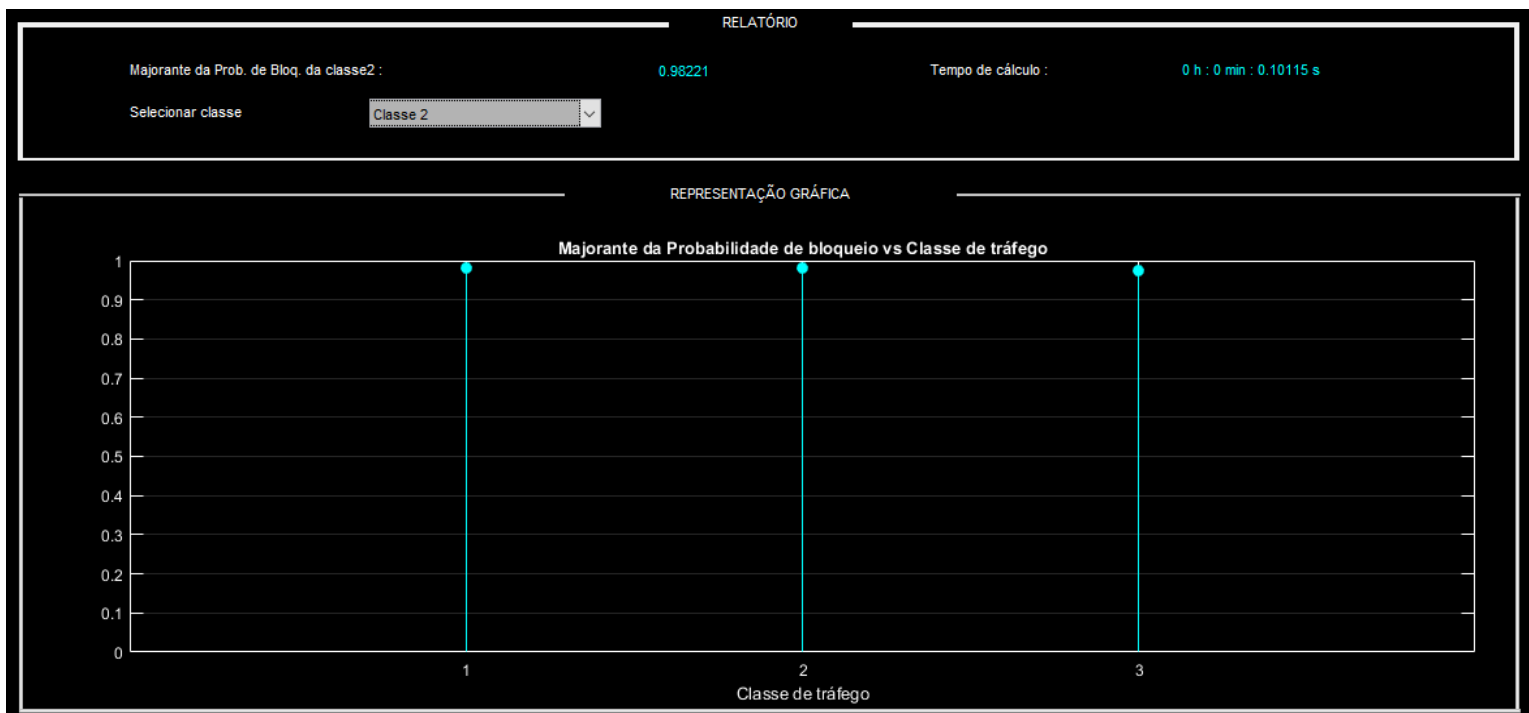


Figura 13: Probabilidades de bloqueio para cada *flow*, segundo o *matnet*

Podemos ver pela figura 13 ilustradas acima, que a probabilidade de bloqueio calculada para cada *flow*, usando o *matnet*, coincide com os valores teóricos calculados anteriormente: *flow1*: 0.9095; *flow2*: 0.9095; *flow3*: 0.8552.

**(3.2)** Voltando a recorrer o programa *Matnet*, a aproximação majorante da probabilidade de bloqueio relativamente aos 3 *flows* existentes é a seguinte:



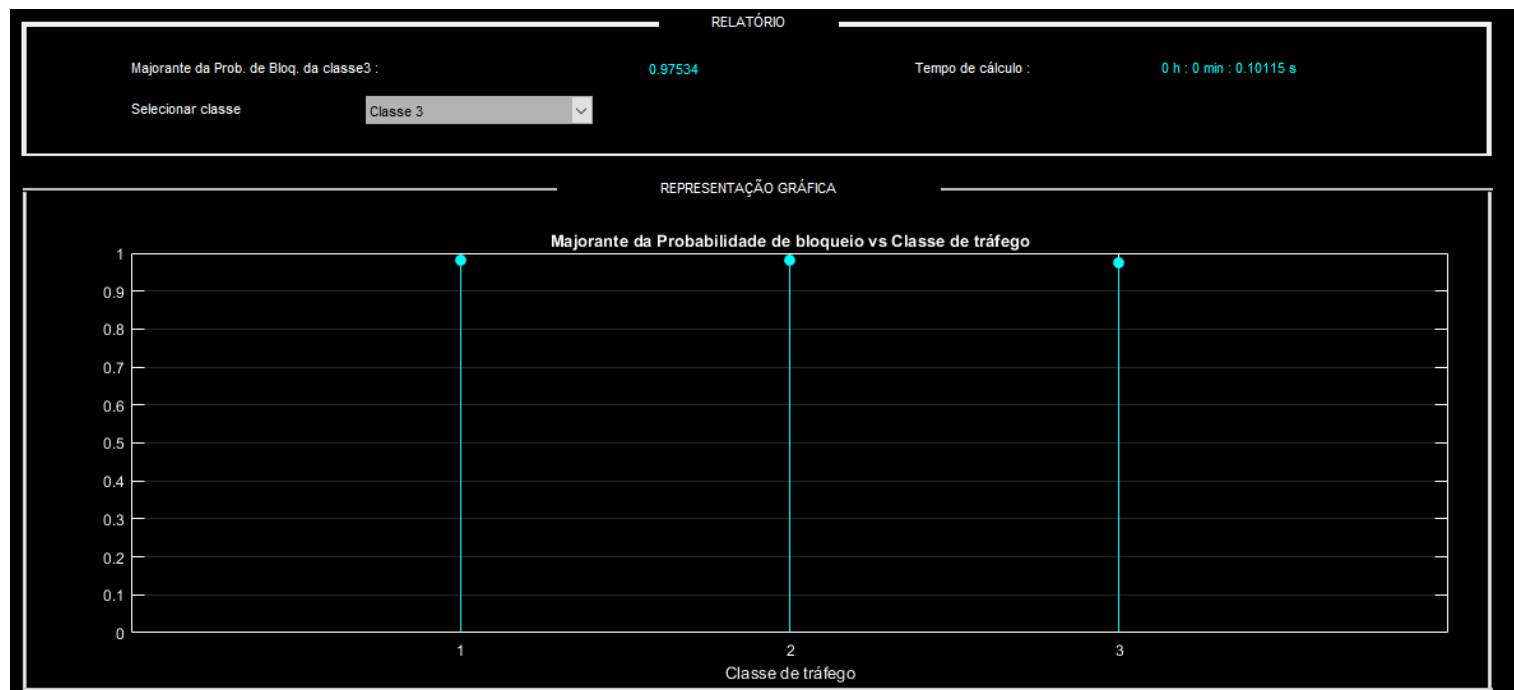


Figura 14: Aproximação do majorante das probabilidades de bloqueio para cada *flow*, segundo o *matnet*

Podemos verificar que os valores gerados pelo *matnet* foram 0.98221; 0.98221 e 0.97534, para o *flow1*, *flow2* e *flow3*, respetivamente. Relativamente aos valores da aproximação da carga reduzida da probabilidade de bloqueio de cada *flow*, obtivemos o seguinte:





Figura 15: Cálculo aproximado das probabilidade de bloqueio para cada *flow*, segundo o *matnet*

A aproximação do majorante exprime o pior caso possível, referente à probabilidade de bloqueio de cada *flow*, considerando os mesmos como acontecimentos independentes o bloqueio de cada ligação. Daí ser expectável que os valores apresentados sejam superiores aos valores exatos. A aproximação de carga reduzida não considera que haja uma independência entre cada ligação, no bloqueio de um *flow* considerado, logo os valores apresentados aproximar-se-ão dos valores exatos, comparativamente com a aproximação do majorante.

**(3.3)** De forma a confirmar os resultados ilustrados acima, agora utilizando o *cnet*, os resultados obtidos foram os seguintes:

```
>> ex3(100)
The blocking probability of flow1 through cnet simulator:
0.9071

The blocking probability of flow2 through cnet simulator:
0.9072

The blocking probability of flow3 through cnet simulator:
0.8510
```

Figura 16: Probabilidade de bloqueio de cada *flow*, segundo o *cnet*, para N=100 simulações

Podemos concluir que os valores gerados, segundo o *cnet*, aproximam-se bastante dos valores exatos calculados anteriormente, com o valor do *flow1* ser praticamente igual ao do *flow2*, como já seria de esperar, sendo que os dois *flows* são idênticos, e com o *flow3* a ter um valor mais baixo que os restantes, como também já era espectável, devido às conclusões feitas nas alíneas anteriores.

**(3.4)** Nesta alínea, é-nos pedido a solução de *routing* que minimize a probabilidade de bloqueio das chamadas, em cada *flow* existente da rede. Chegamos à seguinte solução, que é ilustrada pela figura 17:

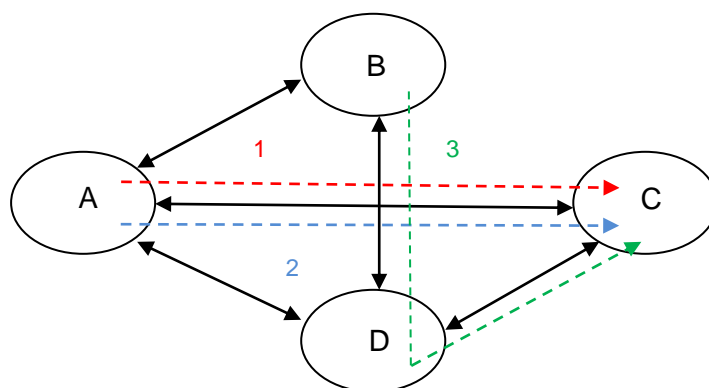


Figura 17: Topologia da rede do exercício 3, com solução de *routing* que minimize a probabilidade de bloqueio das chamadas

Considerando o enunciado do exercício em causa e que as chamadas que ocorrem nos 3 *flows* têm a mesma duração, e que chegam com probabilidades de chegada diferentes (excepto no *flow1* e no *flow2*); sabendo que o nó C tem apenas duas ligações para os vizinhos (nó B e nó D), então na melhor das hipóteses, dois *flows* terão de partilhar a mesma ligação.

Será uma vantagem colocar os *flows* que têm uma menor probabilidade de chegada de uma chamada na mesma ligação, desta forma, a melhor forma é os *flows1* e *flows2* partilharem a mesma ligação, ou seja, irem pelo caminho direto para o nó C. O *flow3*, que por sinal é o que contém maior probabilidade de chegada de uma chamada, irá ter um caminho independente dos outros *flows*, ou seja, vai estar isolado, e irá, igualmente, ter o menor caminho para o nó C (passando pelo nó D).

Por fim, irão ser apenas utilizadas as ligações de maior capacidade, neste caso concreto serão as de 64kb/s, que irá contribuir também para a redução da probabilidade de bloqueio. Recorrendo agora ao matnet, obtivemos os seguintes resultados:





Figura 18: Probabilidade de bloqueio de cada *flow*, para a solução de *routing*, segundo o *matnet*

Podemos verificar que agora a probabilidades de bloqueio de cada *flow* são 0.72;0.72 e 0.80198, para o *flow1*, *flow2* e *flow3*, respetivamente. Concluimos assim, que segundos estes valores, esta solução de *routing* é melhor que a original, dado que os valores agora são mais baixos, com a probabilidade do *flow1* e *flow2* a baixar bastante, como seria de esperar.

## Código em anexo:

Nota: O código em anexo apresentado representa as funções criadas de raiz ou as funções que sofreram alterações, comparativamente com o código fornecido pelo professor.

### (exercício 1)

```
function f = kleinrockApproximation(ro)
    C1=64000;          %velocidade de ligacao do link1
    C2=64000;          %velocidade de ligacao link2
    Lp= 1000;          %tamanho medio de cada pacote

    u= C1/Lp;
    lambda= ro*u;

    kleinrockAvgDellay=(1/ (C1/Lp -lambda))+(1/ (C2/Lp-lambda));

    disp('For lambda = ');
    disp(lambda);
    disp('Kleinrock Average Dellay');
    disp(kleinrockAvgDellay);

    f=kleinrockAvgDellay;

end

function parameters

    global LinkCapacities;
    global Flows;
    global endTime;
    global Topology

    Topology=[0 1 0;
              0 0 2;
              0 0 0];

    %Define here the capacity of each link
    LinkCapacities=[64000 64000 64000]; %In bits/sec

    %Define here the flows. Flows is a cell array that stores for each flow the
    %mean inter-arrival time (in seconds), the mean packet length (in bits) and
    %the route
    Flows={1/3.2,1000,[1];
           1/3.2,1000,[2]};

    %Define here the simulation end time, function of the maximum mean
    %interarrival time
    endTime=1000*(1/16);

function [ f ] = ex1(N, ro)

    media_atraso=0;
    aux=0;

    for i=1:N
        vector_atraso=pnet();
        aux=vector_atraso(1)+vector_atraso(2);
        media_atraso=(media_atraso+aux);
    end

    media_atraso=media_atraso/N;
```



```

valor_teorico=kleinrockApproximation(ro);

disp('Para ro = ');
disp(ro);
disp('The average packet delay through pnet simulator: ');
disp(media_atraso);
disp('-----');
disp('The average packet delay through the kleinrock approximation: ');
disp(valor_teorico);

end

```

## (exercício 2.2)

```

function parameters

global LinkCapacities;
global Flows;
global endTime;
global Topology

Topology=[0 1 2 0 0 0;
          0 0 0 3 4 0;
          0 0 0 0 5 0;
          0 0 0 0 0 6;
          0 0 0 0 0 7;
          0 0 0 0 0 0;];

%Define here the capacity of each link
LinkCapacities=[256000 256000 256000 256000 256000 256000]; %In bits/sec

Flows={1/215,1000,[1 3 6];
       1/64,1000,[2 5];
       1/128,1000,[2 5 7];
       1/128,1000,[4];};

endTime=1000*(1/16);

function [ f ] = ex2(N)

media_atrasoTotal=0;
media_atrasoFlow1=0;
media_atrasoFlow2=0;
media_atrasoFlow3=0;
media_atrasoFlow4=0;

lambda1=215;
lambda2=64;
lambda3=128;
lambda4=lambda3;
lambdaTotal=lambda1+lambda2+lambda3+lambda4;

auxTotal=0;
for i=1:N
    vector_atraso=pnet();

    media_atrasoFlow1=media_atrasoFlow1+vector_atraso(1);
    media_atrasoFlow2=media_atrasoFlow2+vector_atraso(2);
    media_atrasoFlow3=media_atrasoFlow3+vector_atraso(3);
    media_atrasoFlow4=media_atrasoFlow4+vector_atraso(4);

end

```

```

media_atrasoFlow1=media_atrasoFlow1/N;
media_atrasoFlow2=media_atrasoFlow2/N;
media_atrasoFlow3=media_atrasoFlow3/N;
media_atrasoFlow4=media_atrasoFlow4/N;

```

```

media_atrasoTotal=(lambda1/lambdaTotal)*media_atrasoFlow1+(lambda2/lambdaTotal)*media_atrasoFlow2+(lambda3/lambdaTotal)*media_atrasoFlow3+(lambda4/lambdaTotal)*media_atrasoFlow4;

```

```

disp('The average packet delay of flow1 through pnet simulator: ');
disp(media_atrasoFlow1);

disp('The average packet delay of flow2 through pnet simulator: ');
disp(media_atrasoFlow2);

disp('The average packet delay of flow3 through pnet simulator: ');
disp(media_atrasoFlow3);

disp('The average packet delay of flow4 through pnet simulator: ');
disp(media_atrasoFlow4);

disp('The average packet delay of the network through pnet simulator: ');
disp(media_atrasoTotal);

end

```

### (exercício 2.3)

```

function parameters

```

```

global LinkCapacities;
global Flows;
global endTime;
global Topology

```

```

Topology=[0 1 2 0 0 0;
          0 0 0 3 4 0;
          0 0 0 0 5 0;
          0 0 0 0 0 6;
          0 0 0 0 0 7;
          0 0 0 0 0 0;];

```

```

LinkCapacities=[256000 256000 256000 256000 256000 256000 256000];

```

```

Flows={1/215,1000,[1];
       1/64,1000,[2 5];
       1/128,1000,[2 5 7];
       1/171.5,1000,[4];
       1/171.5,1000,[3 6];
       1/43.5,1000,[7]};

```

```

endTime=1000*(1/16);

```

```

function [ f ] = ex2_2(N)

```

```

media_atrasoTotal=0;
media_atrasoFlow1=0;
media_atrasoFlow2=0;
media_atrasoFlow3=0;
media_atrasoFlow4=0;
media_atrasoFlow5=0;
media_atrasoFlow6=0;

```

```

lambda1=215;

```

```

lambda2=64;
lambda3=128;
lambda4=171.5;
lambda5=171.5;
lambda6=43.4;

lambdaTotal=lambda1+lambda2+lambda3+lambda4+lambda5+lambda6;

auxTotal=0;
for i=1:N
    vector_atraso=pnet();

    media_atrasoFlow1=media_atrasoFlow1+vector_atraso(1);
    media_atrasoFlow2=media_atrasoFlow2+vector_atraso(2);
    media_atrasoFlow3=media_atrasoFlow3+vector_atraso(3);
    media_atrasoFlow4=media_atrasoFlow4+vector_atraso(4);
    media_atrasoFlow5=media_atrasoFlow4+vector_atraso(5);
    media_atrasoFlow6=media_atrasoFlow4+vector_atraso(6);

end

media_atrasoFlow1=media_atrasoFlow1/N;
media_atrasoFlow2=media_atrasoFlow2/N;
media_atrasoFlow3=media_atrasoFlow3/N;
media_atrasoFlow4=media_atrasoFlow4/N;
media_atrasoFlow5=media_atrasoFlow5/N;
media_atrasoFlow6=media_atrasoFlow6/N;

media_atrasoTotal=(media_atrasoFlow1+media_atrasoFlow2+media_atrasoFlow3+media_atrasoFlow4+
media_atrasoFlow5+media_atrasoFlow6)/6;

disp('The average packet delay of flow1 through pnet simulator: ');
disp(media_atrasoFlow1);

disp('The average packet delay of flow2 through pnet simulator: ');
disp(media_atrasoFlow2);

disp('The average packet delay of flow3 through pnet simulator: ');
disp(media_atrasoFlow3);

disp('The average packet delay of flow4 through pnet simulator: ');
disp(media_atrasoFlow4);

disp('The average packet delay of flow5 through pnet simulator: ');
disp(media_atrasoFlow5);

disp('The average packet delay of flow6 through pnet simulator: ');
disp(media_atrasoFlow6);

disp('The average packet delay of the network through pnet simulator: ');
disp(media_atrasoTotal);

end

```

### (exercício 3.3)

```

function parameters

global Topology;
global Capacities;
global Flows;
global endTime;

```

```
Topology=[0 0 1 0;  
          0 0 2 0;  
          0 0 0 3;  
          0 0 0 0];
```

```
Capacities=[32000 64000 64000]; %In bits/sec
```

```
Flows={3,1,32000,[1 3];  
       3,1,32000,[1 3];  
       3,3,32000,[2 3];};
```

```
endTime=1000*(0.1);
```

```
function [ f ] = ex3(N)
```

```
media_blockFlow1=0;  
media_blockFlow2=0;  
media_blockFlow3=0;
```

```
for i=1:N  
    vector_atraso=cnet();  
  
    media_blockFlow1=media_blockFlow1+vector_atraso(1);  
    media_blockFlow2=media_blockFlow2+vector_atraso(2);  
    media_blockFlow3=media_blockFlow3+vector_atraso(3);
```

```
end
```

```
media_blockFlow1=media_blockFlow1/N;  
media_blockFlow2=media_blockFlow2/N;  
media_blockFlow3=media_blockFlow3/N;
```

```
disp('The blocking probability of flow1 through cnet simulator: ');  
disp(media_blockFlow1);
```

```
disp('The blocking probability of flow2 through cnet simulator: ');  
disp(media_blockFlow2);
```

```
disp('The blocking probability of flow3 through cnet simulator: ');  
disp(media_blockFlow3);
```

```
end
```