

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:

3

2

1

Link2=64kb/s

Link1=64kb/s

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: . Calculou-se para 4 valores distintos de ρ (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 ρ. Dado os valores de ρ, e sabendo que ;, os valores de λ considerados foram os seguintes: 3.2; 32; 60.8; 62.4. Posto isto, obtivemos os seguintes resultados:

# 

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

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

# 

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

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

# 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 ρ vai aumentado, estes dois valores vão ficando cada vez mais dispersos um do outro, contudo para ρ=0.05 e ρ=0.5, há uma grande aproximação de resultados. Tal já não acontece com ρ=0.95 e ρ=0.975.

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

# Exercício 2:

# Podemos esquematizar a rede deste exercício da seguinte forma:

D

F

B

E

A

C

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: ; e tendo em conta que calculamos o atraso médio de um pacote para cada fluxo i existente na rede:

# *Para o fluxo 1*:

# *Para o fluxo 2*:

# *Para o fluxo 3*:

# *Para o fluxo 4*:

# O atraso médio dos pacotes da rede é dada por:

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

# 

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:

X1

*r*

D

F

B

X2

A

E

C

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 & X2 > 0;

# ⇒ X1 = 171.5; X2 = 43.5;

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

# 

D

F

B

E

A

C

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

Com

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

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:

**32k**

B

**64k**

A

C

**64k**

**64k**

**32k**

D

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:

1 0 1

1 0 0

0 1 1

0 1 0

0 0 0

0 0 2

0 0 1

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

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

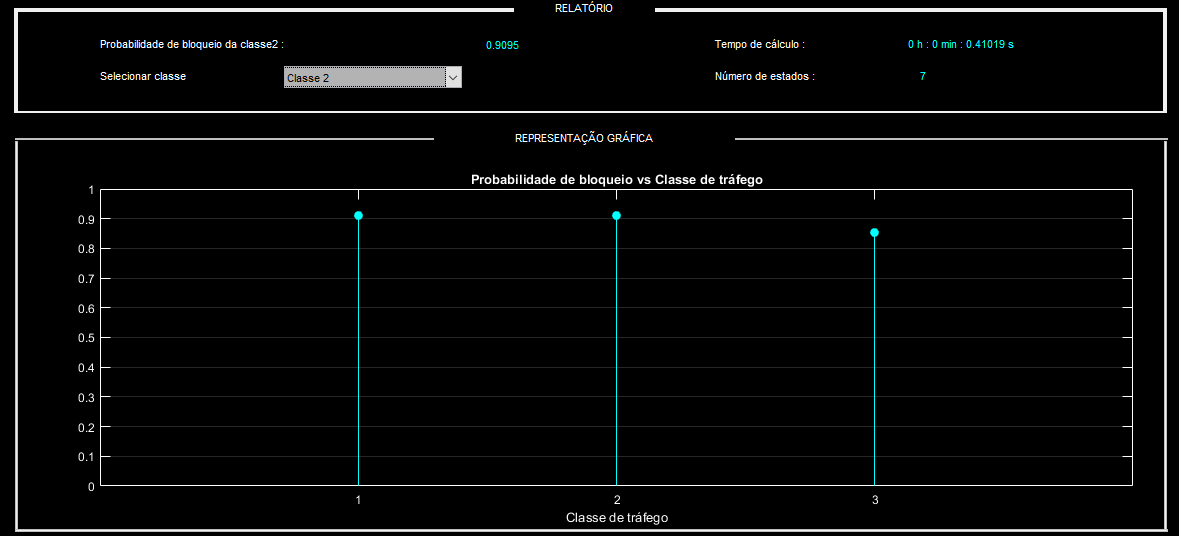
Recorrendo ao *Matlab*, para resolver as equações apresentadas, chegamos aos seguintes valores:

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

Posto isto, chegamos aos seguintes resultados:

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





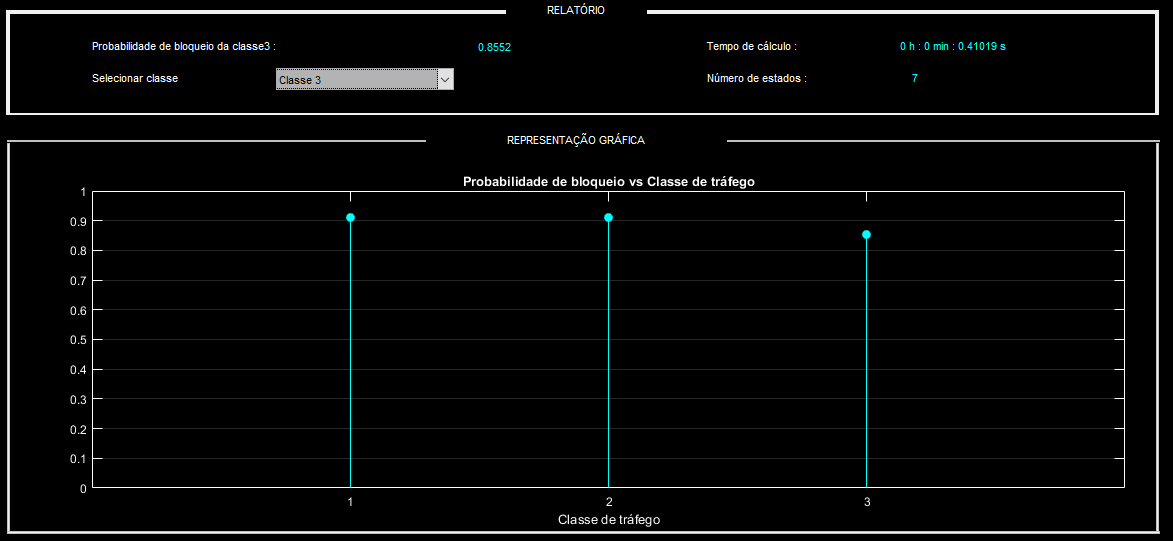
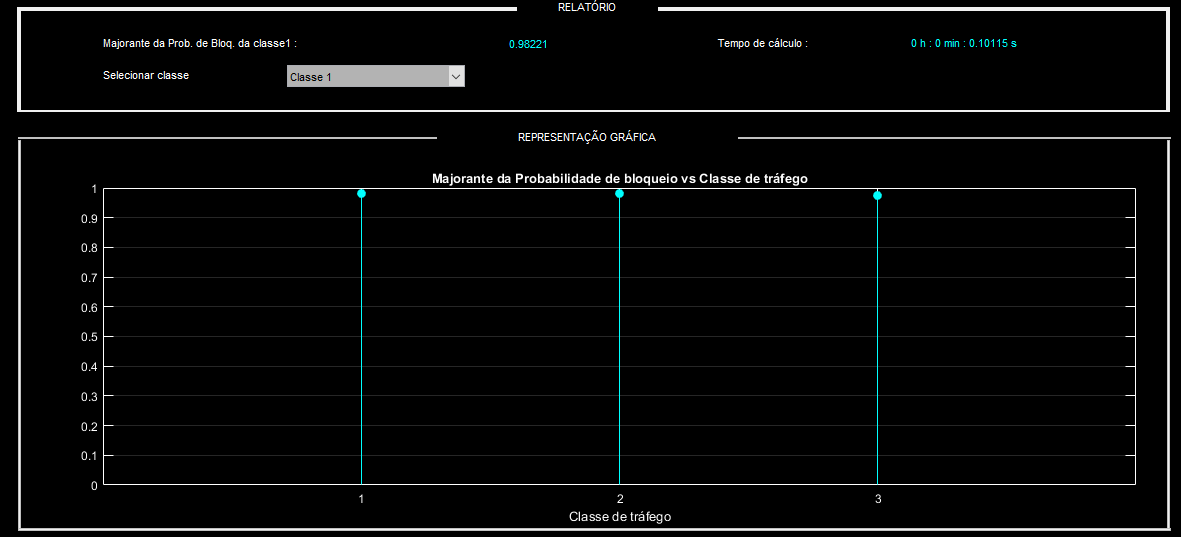
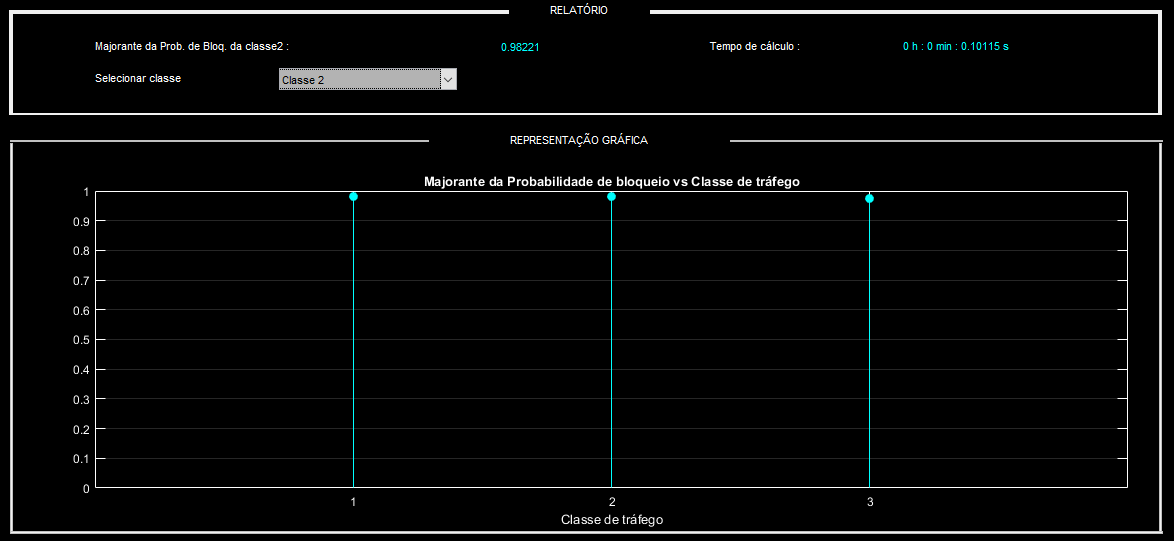


Figura 13: Probabilidades de bloqueio para cada *flow*, segundo o *matnet*

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:



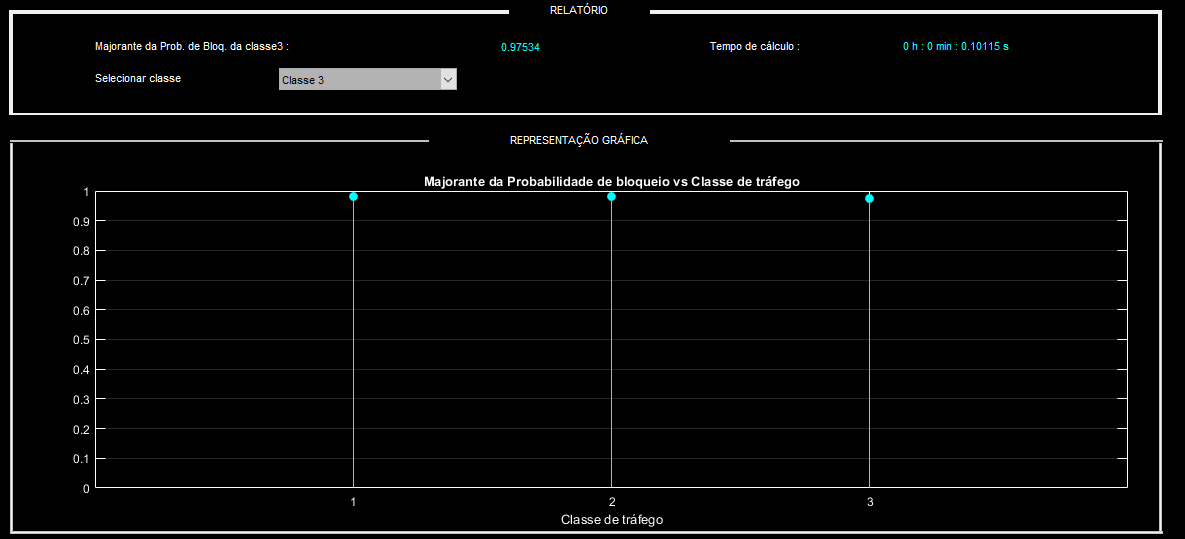


Figura 14: Aproximação do majorante das probabilidades de bloqueio para cada *flow*, segundo o *matnet*

 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:

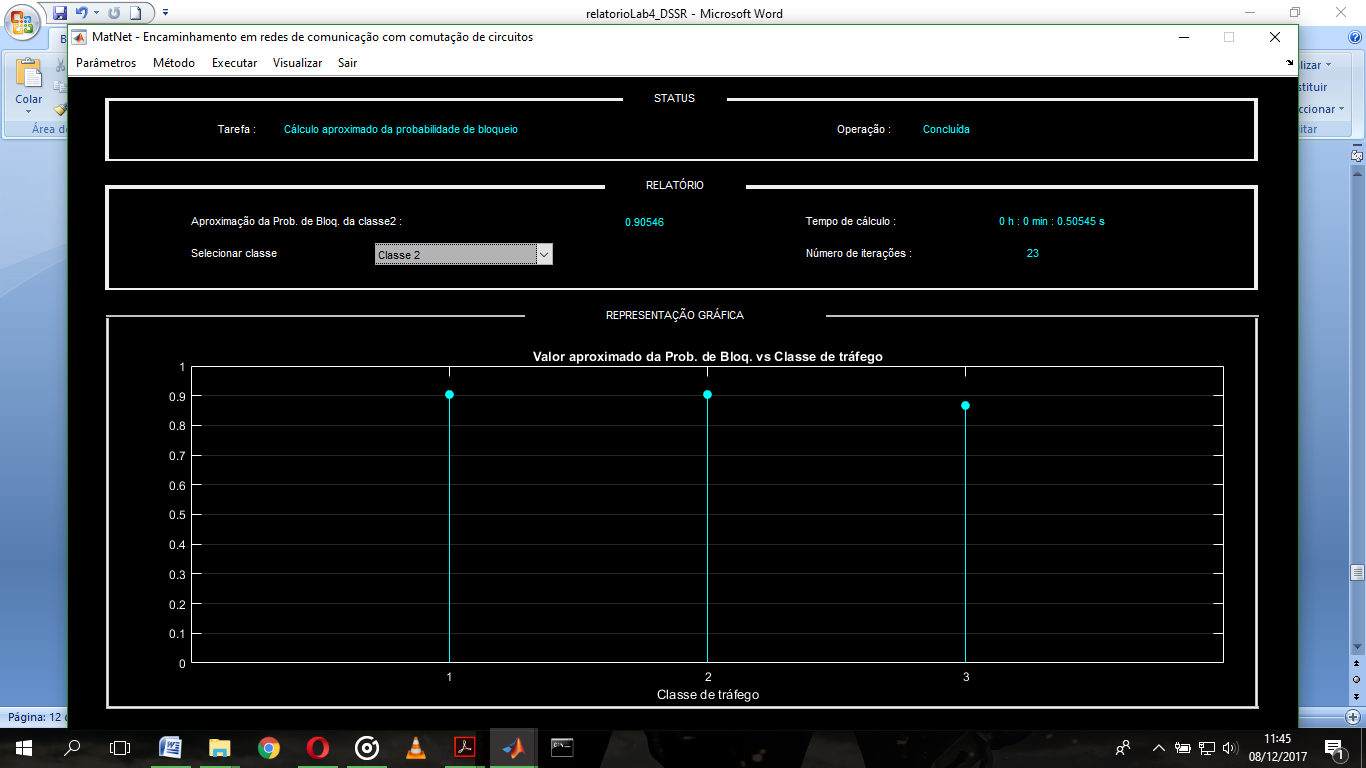
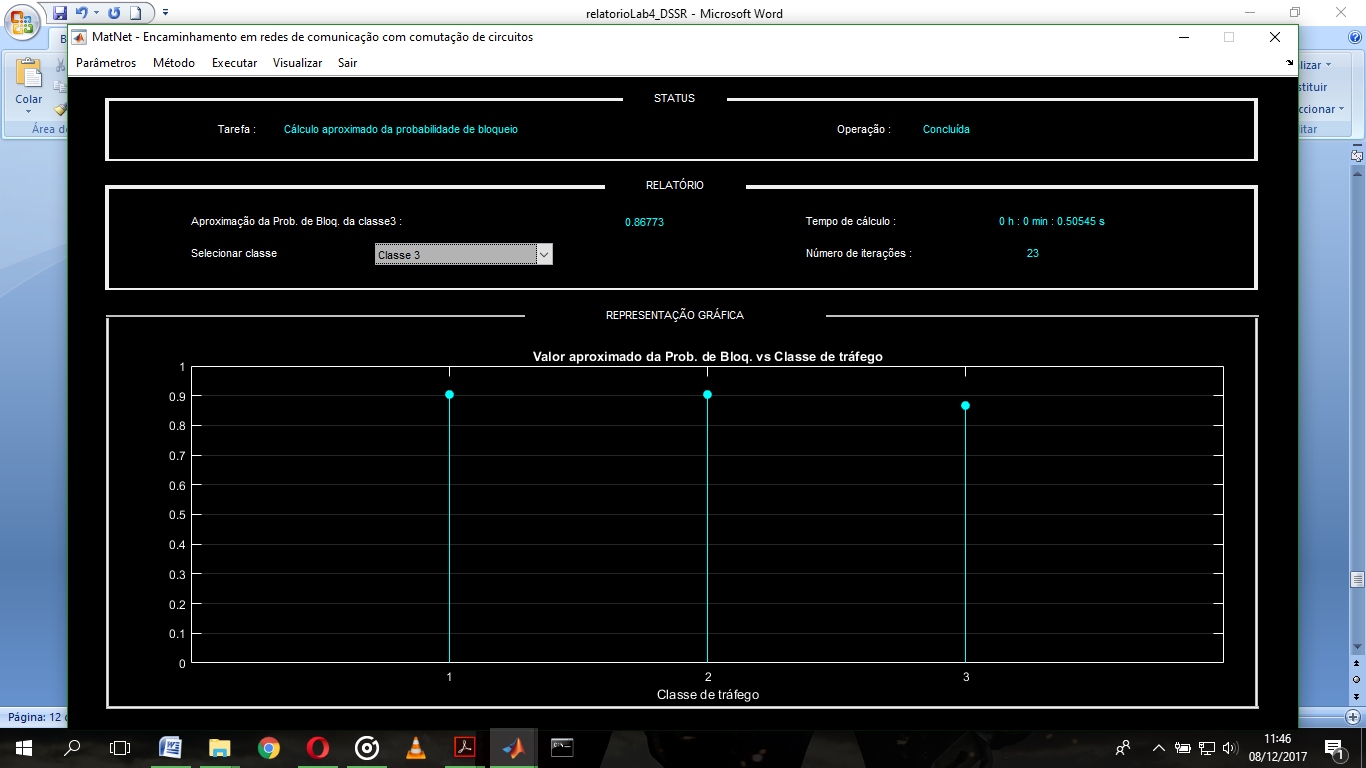


Figura 15: Cálculo aproximado das probabilidade de bloqueio para cada *flow*, segundo o *matnet*

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.

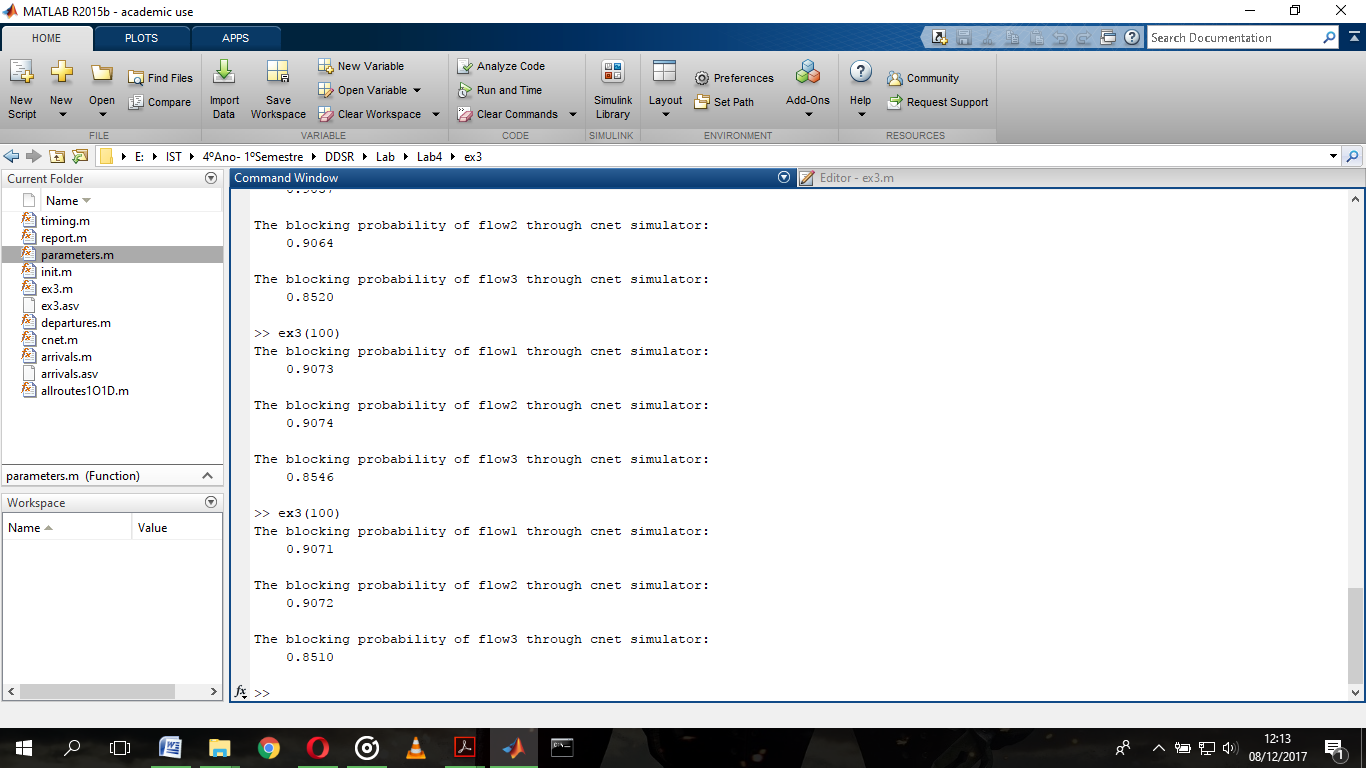
 ***(3.3)*** De forma a confirmar os resultados ilustrados acima, agora utilizando o *cnet*, os resultados obtidos foram os seguintes:

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:

B

3

1

C

A

2

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

D

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 contêm 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:



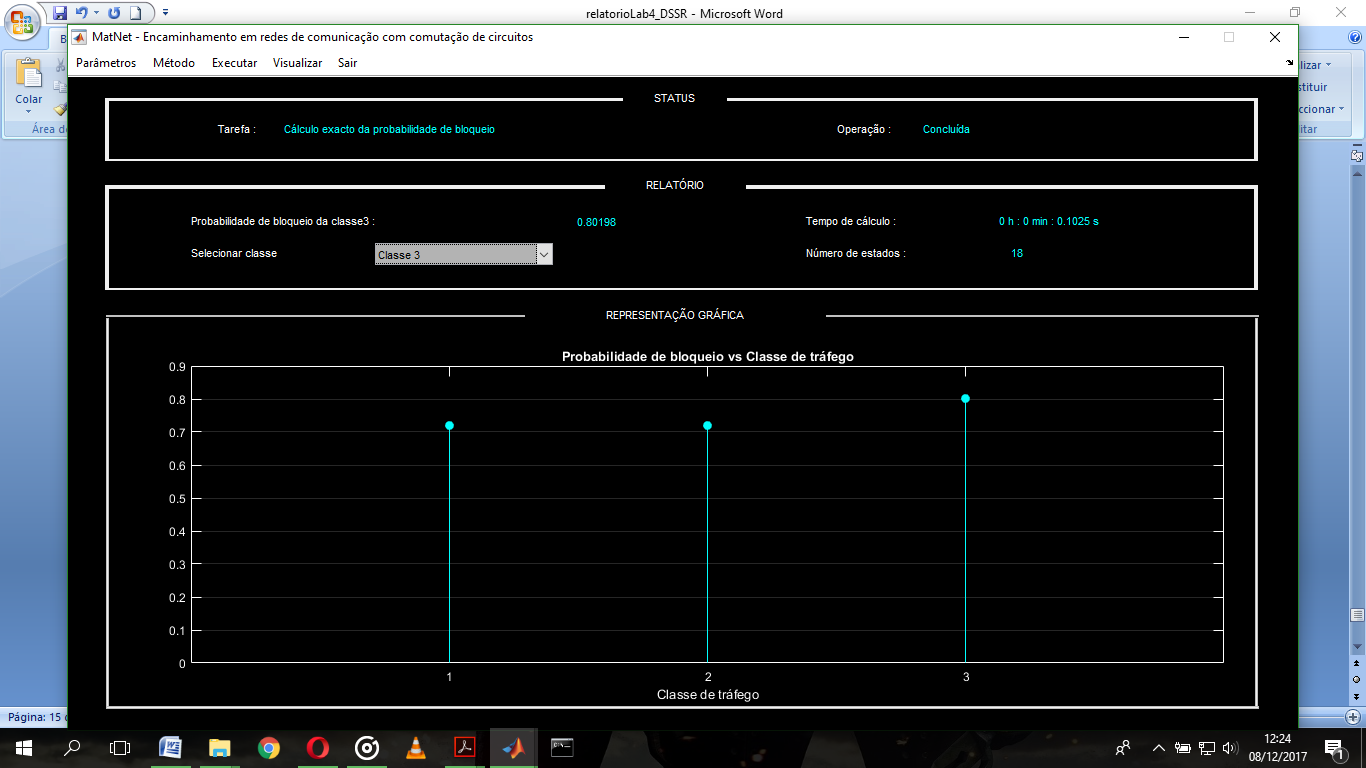


Figura 18: Probabilidade de bloqueio de cada *flow*, para a solução de *routing*, segundo o *matnet*

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. Concluímos 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 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\_atras oFlow2+(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