Tarefa #2

Performance de Bloqueio de Serviços de Streaming de Vídeo

Martim Neves, 88904
Gabriel Saudade, 89304

23 de Maio 2021

Universidade de Aveiro
UC: Desempenho e Dimensionamento de Redes
Professor: Amaro Fernandes de Sousa

Task 2

Simulador 2

Código Matlab

```
function [bHD b4K]= simulator2(lambda,p,n,S,W,R,fname)
    %lambda = request arrival rate (in requests per hour)
    %p=       percentage of requests for 4K movies (%)
    %n=       number of servers
    %S=       interface capacity of each server (Mbps)
    %W=       resource reservation for 4K movies (Mbps)
    %R=       stop simulation on ARRIVAL no. R
    %fname=   filename with the duration of each movie

    invlambda=60/lambda;      %average time between
        requests (in minutes)
    invmiu= load(fname);      %duration (in minutes) of
        each movie
    Nmovies= length(invmiu); % number of movies

    %Events definition:
    ARRIVAL= 0;               %movie request
    DEPARTURE_HD= 1;          %termination of a HD movie
        transmission
    DEPARTURE_4K= 2;          %termination of a 4K movie
        transmission
    %State variables initialization:
    STATE= zeros(1,n);
    STATE_HD=0;
    %Statistical counters initialization:
    REQUESTS_HD=0;
    REQUESTS_4K=0;
    BLOCKED_HD=0;
    BLOCKED_4K=0;
    %Simulation Clock and initial List of Events:
    Clock= 0;
    EventList= [ARRIVAL exprnd(invlambda) 0];
    C = n*S;
    while (REQUESTS_HD + REQUESTS_4K) < R
        event= EventList(1,1);
        Clock= EventList(1,2);
        old=EventList(1,3);
        EventList(1,:)= [];
        if event == ARRIVAL
```

```
36              EventList= [EventList; ARRIVAL Clock+exprnd(
                    invlambda) 0];
37              [val,pos]=min(STATE);
38              if (rand * 100) < p
39                  REQUESTS_4K= REQUESTS_4K + 1;
40                  if val + 25 <= S
41                      STATE(pos) = STATE(pos) + 25;
42                      EventList= [EventList; DEPARTURE_4K
                            Clock+invmiu(randi(Nmovies)) pos];
43                  else
44                      BLOCKED_4K = BLOCKED_4K + 1;
45                  end
46              else
47                  REQUESTS_HD= REQUESTS_HD + 1;
48                  if STATE_HD + 5 <= (C-W) && val + 5 <= S
49                      STATE_HD = STATE_HD + 5;
50                      STATE(pos) = STATE(pos) + 5;
51                      EventList= [EventList; DEPARTURE_HD
                            Clock+invmiu(randi(Nmovies)) pos];
52                  else
53                      BLOCKED_HD = BLOCKED_HD + 1;
54                  end
55              end
56          elseif event == DEPARTURE_4K
57              STATE(old) = STATE(old) - 25;
58          else
59              STATE(old) = STATE(old) - 5;
60              STATE_HD = STATE_HD - 5;
61          end
62          EventList= sortrows(EventList,2);
63      end
64      bHD= 100*BLOCKED_HD/REQUESTS_HD;      % blocking
            probability in %
65      b4K= 100*BLOCKED_4K/REQUESTS_4K;
66  end
```

Análise do Código

Na construção do simulador 2 foram inicialmente declaradas variáveis para a definição de cada evento (arrival, departureHD e departure4K), cada uma delas inicializada com um valor diferente para que seja possível que cada uma represente um estado diferente. Foram também inicializadas variáveis que permitem saber a capacidade já usada e livre de cada servidor, assim como o throughput

total de filmes HD. Adicionalmente foram declaradas variáveis para contabilizar o número de pedidos e bloqueios de cada tipo de filme, assim como uma EventList, para saber o tipo de evento, e a variável C, que representa a capacidade de conexão à Internet da server farm, sendo neste caso calculada pela multiplicação entre a capacidade de cada servidor e o número de servidores constituintes dessa server farm.

Em seguida é feito um ciclo while, para garantir que o programa corre enquanto o número total de filmes pedidos é menor que o critério de paragem, é verificado qual é o evento que ocorre e são realizadas ações em função de cada evento. Caso o evento seja a chegada de um pedido de filme, é preciso encontrar o servidor menos carregado, para que este possa aceitar o pedido, caso tenha capacidade para tal. Depois de encontrado o servidor menos carregado, é gerado um número aleatório para simular a probabilidade de o pedido ser de um filme 4K. Caso o número aleatório seja menor que a probabilidade de o filme pedido ser 4K, assume-se que o filme pedido é de facto 4K, a variável de contagem de filmes 4K é incrementada em uma unidade e é feita uma verificação para saber se o servidor encontrado anteriormente tem capacidade para atender o pedido. Caso tenha, isto é, caso a capacidade já utilizada mais 25 (throughput de um filme 4K), seja menor ou igual que a capacidade da interface do servidor, é somado o valor 25 à capacidade já utilizada desse servidor e é atualizada a EventList. Caso o servidor não tenha capacidade para atender o pedido, é incrementada em uma unidade a variável que contabiliza o número de pedidos 4K bloqueados.

No caso de o valor aleatório gerado anteriormente ser maior ou igual que a probabilidade de o pedido ser um filme 4K, assume-se que o pedido é um filme HD e, como tal, é incrementada em uma unidade a variável que contabiliza o número de pedidos de filmes HD. Depois disso, e caso sejam satisfeitas as condições de o servidor ter capacidade suficiente para atender o novo pedido e garantir que o atendimento do pedido respeita a reserva de recursos para filmes 4K, isto é, depois do atendimento deste novo pedido a server farm ainda tem pelo menos W MegaBytes disponíveis para atender pedidos de filmes 4K, a variável que contabiliza o throughput de filmes HD é incrementada em cinco unidades (throughput de um filme HD), é adicionado o valor 5 à capacidade já utilizada desse servidor e é atualizada a EventList. Caso as condições referidas anteriormente não sejam satisfeitas, é incrementada em uma unidade a variável que contabiliza o número de pedidos HD bloqueados.

Caso o evento seja uma departure4K, isto é, um filme 4K tenha chegado ao fim, é libertada no servidor a capacidade que estava alocada a esse filme, que, neste caso, seria 25. O último caso possível é uma departureHD, que indica que terminou um filme HD. Neste caso, é libertada no servidor e na variável que contabiliza o throughput de filmes HD a capacidade que estava alocada a

esse filme, que neste caso é 5. Após ser processado cada caso, e no fim de cada iteração, é organizada a EventList.

Finalmente, são calculadas as probabilidades de bloqueio de cada tipo de filme, sendo estas obtidas através da divisão entre o número de filmes bloqueados de cada tipo e número total de filmes do tipo correspondente que foram pedidos.

Task 2.a - usando a configuração 1 e W=0, determinar os intervalos de 90% de confiança para ambas as probabilidades de bloqueio

Código Matlab

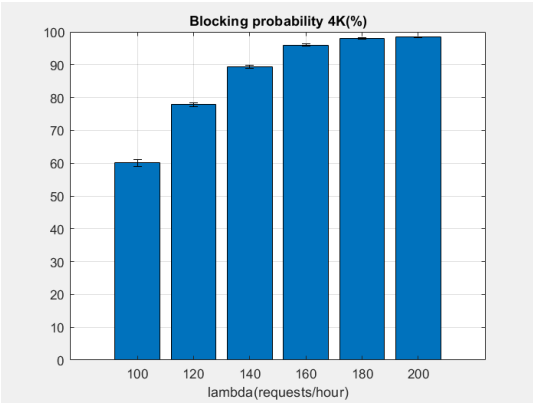
```
R=10000;
n=10;
S=100;
W=0;
p=20;
lambda=[100,120,140,160,180,200];
fname='movies.txt';
for i=1:length(lambda)
    for j=1:10
        [bHD(j) b4K(j)] = simulator2(lambda(i),p,n,S,W,R,
            fname);
    end
    alfa= 0.1; %90% confidence interval%
    media4k(i) = mean(b4K);
    term4k(i) = norminv(1-alfa/2)*sqrt(var(b4K)/10);
    mediahd(i) = mean(bHD);
    termhd(i) = norminv(1-alfa/2)*sqrt(var(bHD)/10);
end
figure(1)
bar(lambda,media4k)
hold on
errb=errorbar(lambda,media4k,term4k,term4k);
errb.Color=[0 0 0];
errb.LineStyle = 'none';
hold off
grid on
title('Blocking probability 4K(%)')
xlabel('lambda(requests/hour)')
ylim([0 100])

figure(2)
bar(lambda,mediahd)
```

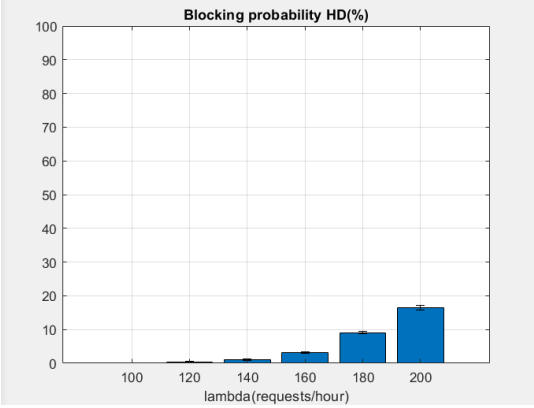
```
hold on
erro=errorbar(lambda,mediahd,termhd,termhd);
erro.Color=[0 0 0];
erro.LineStyle = 'none';
hold off
grid on
title('Blocking probability HD(%)')
xlabel('lambda(requests/hour)')
ylim([0 100])
prob1=pErr1+pErr2+pErr3;
```

Análise do Código

As variáveis declaradas no início do programa correspondem aos valores usados na configuração 1. A seguir às declarações iniciais, é feito um ciclo for para percorrer todos os valores de lambda e outro ciclo for para fazer as 10 corridas pedidas no enunciado. Em seguida, é chamado o simulador 2 e os valores de retorno são guardados para poderem ser usados no cálculo do intervalo de confiança de ambas as probabilidades de bloqueio. Por fim, consoante os valores calculados, são desenhados os gráficos de barras e as respetivas barras de erro para cada valor de probabilidade, em função do valor de lambda.

Resultados





Análises e Justificações

Através dos resultados obtidos, podemos concluir que, quanto maior for o número de pedidos por hora, maior é a probabilidade de bloqueio, tanto nos filmes HD, como nos filmes 4K. Porém, é previsível que, dado o mesmo número de pedidos por hora, a probabilidade de bloqueio nos filmes 4K seja superior à probabilidade de bloqueio dos filmes HD, pois a taxa de transferência de filmes 4K é 5 vezes superior à de filmes HD.

Relativamente ao valor do critério de paragem R, trata-se de um valor alto o suficiente, pois a probabilidade de erro já é um valor muito menor comparativamente ao valor da probabilidade de bloqueio, logo não há necessidade de ter uma resolução de amostragem superior, já que esta dá valores suficientemente bons.

Task 2.b - repetir a simulação anterior mas para as configurações 2 e 3 e apresentar todos os resultados no mesmo gráfico de barras

Código Matlab

```
n2=4;
S2=250;
n3=1;
S3=1000;
for i=1:length(lambda)
    for j=1:10
        [bHD2(j) b4K2(j)] = simulator2(lambda(i),p,n2,S2,
            W,R,fname);
    end
    alfa= 0.1; %90% confidence interval%
    media4k2(i) = mean(b4K2);
    term4k2(i) = norminv(1-alfa/2)*sqrt(var(b4K2)/10);
    mediahd2(i) = mean(bHD2);
    termhd2(i) = norminv(1-alfa/2)*sqrt(var(bHD2)/10);
end
for i=1:length(lambda)
    for j=1:10
        [bHD3(j) b4K3(j)] = simulator2(lambda(i),p,n3,S3,
            W,R,fname);
    end
    media4k3(i) = mean(b4K3);
    mediahd3(i) = mean(bHD3);
end
figure(3)
```

```
23 bar(lambda,[media4k;media4k2;media4k3])
24 grid on
25 title('Blocking probability 4K(%)')
26 xlabel('lambda(requests/hour)')
27 ylim([0 100])
28
29 figure(4)
30 bar(lambda,[mediahd;mediahd2;mediahd3])
31 grid on
32 title('Blocking probability HD(%)')
33 xlabel('lambda(requests/hour)')
34 ylim([0 100])
```

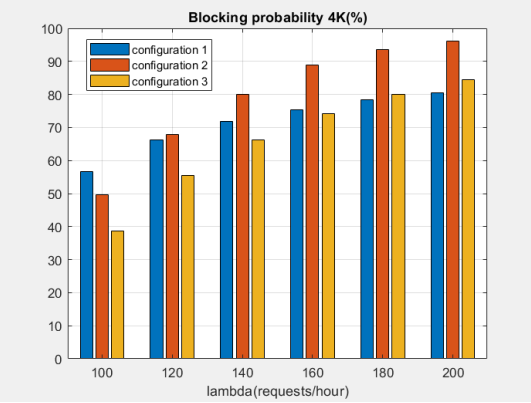
Análise do Código

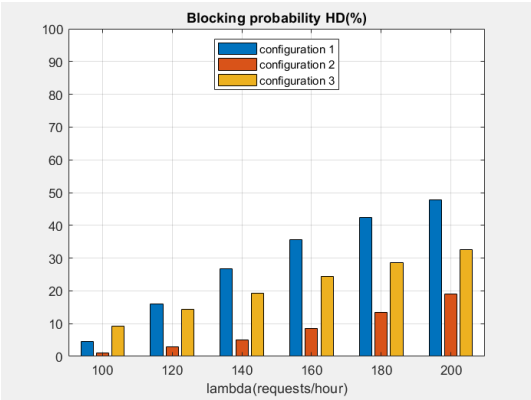
Inicialmente são declaradas as variáveis que têm um valor diferente para cada configuração. Seguidamente, são feitos dois ciclos for, um para fazer as 10 corridas, e outro para percorrer todos os valores de lambda. Dentro desses ciclos é invocado o simulador 2, que vai retornar os valores das probabilidades de bloqueio de cada tipo de filme. Com esses valores de retorno é calculado o intervalo de confiança para a configuração 2. O próximo passo é repetir o processo descrito anteriormente, mas desta vez para a configuração 3. Por fim, e aproveitando o resultado da alínea anterior, juntamente com os resultados obtidos nesta alínea, são desenhados os gráficos de barras correspondentes à probabilidade de bloqueio de cada tipo de filme, dependendo da configuração, e de acordo com os valores de lambda.

Resultados

Análises e Justificações

Relativamente aos filmes 4k, após a obtenção de resultados podemos concluir que é preferível ter um menor número de servidores com capacidade superior por cada *server fam* a uma *server fam* com vários servidores e uma capacidade de interface inferior. Isto deve-se ao facto de que sempre que é requisitado um novo filme, faz-se a verificação do estado de ocupação do servidor e este tem de ter pelo menos 25Mbps de capacidade não utilizada para que o pedido não seja bloqueado. Assim, com interfaces de rede com baixa largura de banda, a





probabilidade de esta se encontrar próxima da sua capacidade total é superior e, portanto, a probabilidade de bloqueio também é maior.

Relativamente aos filmes HD podemos concluir que, dado que o estado de bloqueio de filmes 4k para a configuração 2 é superior, existem mais recursos disponíveis para transmitir filmes HD e, geralmente, estes podem explorar a capacidade total de taxa de transferência dos servidores dado que necessitam unicamente 5Mbps para serem transmitidos.

Sendo assim, é visível que, para ambas as situações, e no caso de não haver reserva de recursos, é preferível ter a configuração 3 (apenas um servidor com maior uma interface com capacidade superior) pois, no geral, esta configuração apresenta menores probabilidades de bloqueio para ambos os tipos de filme que as outras configurações.

Task 2.c - repetir a simulação anterior para todas as configurações e W=400 e apresentar todos os resultados no mesmo gráfico de barras

Código Matlab

```
1 W=400;
2 for i=1:length(lambda)
3     for j=1:10
4         [bHD(j) b4K(j)] = simulator2(lambda(i),p,n,S,W,R,
              fname);
5         [bHD2(j) b4K2(j)] = simulator2(lambda(i),p,n2,S2,
              W,R,fname);
6         [bHD3(j) b4K3(j)] = simulator2(lambda(i),p,n3,S3,
              W,R,fname);
```

```
7       end
8       media4k(i) = mean(b4K);
9       mediahd(i) = mean(bHD);
10      media4k2(i) = mean(b4K2);
11      mediahd2(i) = mean(bHD2);
12      media4k3(i) = mean(b4K3);
13      mediahd3(i) = mean(bHD3);
14  end
15  figure(5)
16  bar(lambda,[media4k;media4k2;media4k3])
17  grid on
18  title('Blocking probability 4K(%)')
19  xlabel('lambda(requests/hour)')
20  ylim([0 100])
21
22  figure(6)
23  bar(lambda,[mediahd;mediahd2;mediahd3])
24  grid on
25  title('Blocking probability HD(%)')
26  xlabel('lambda(requests/hour)')
27  ylim([0 100])
```

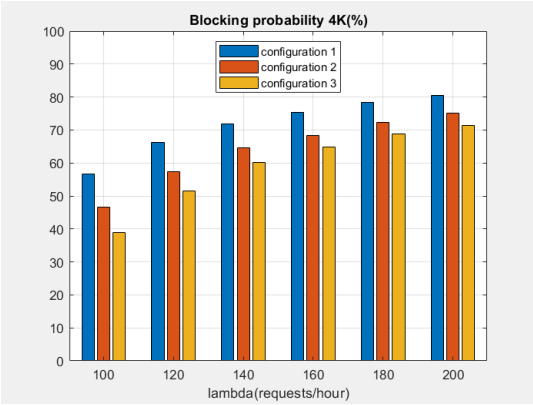
Análise do Código

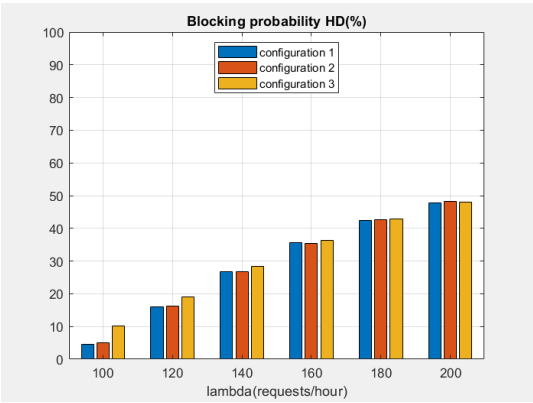
O código deste alínea difere do código desenvolvido na alínea anterior em apenas dois pontos: o valor da variável W, que passa a ser 400, e o facto de que todas as chamadas ao simulador 2 e cálculos das probabilidades de bloqueio foram feitos dentro de apenas dois ciclos for, por uma questão de conveniência e organização.

Resultados

Análises e Justificações

Relativamente aos resultados obtidos para o estado de bloqueio dos filmes 4k, podemos concluir que estes diminuíram substancialmente, comparativamente aos resultados obtidos na alínea anterior, pois neste caso foi imposta a condição de reserva de recursos e, portanto, existe sempre uma taxa de transferência disponível para o seu *streaming*.





Relativamente aos resultados obtidos para o estado de bloqueio de filmes HD, podemos concluir que estes aumentaram significativamente relativamente à alínea anterior, pois neste caso os filmes HD entram em estado de bloqueio caso o total de taxa de transferência seja superior à diferença entre a capacidade da *server farm* e o total da reserva de recursos para filmes 4k ($W - C = 1000 - 400 = 600Mbps$).

Posto isto, e para o caso em que temos uma reserva de recursos de 400 MegaBytes, facilmente chegamos à conclusão que, mais uma vez, a melhor configuração possível para este cenário é a configuração 3 pois, no geral, apresenta menores probabilidades de bloqueio para ambos os filmes, como se pode ver nos gráficos de barras acima apresentados.

Task 2.d - repetir a alínea anterior para W=600

Código Matlab

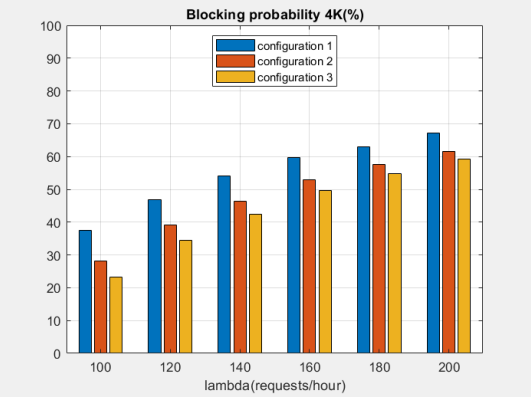
```
W1=600;
for i=1:length(lambda)
    for j=1:10
        [bHD(j) b4K(j)] = simulator2(lambda(i),p,n,S,W1,R
            ,fname);
        [bHD2(j) b4K2(j)] = simulator2(lambda(i),p,n2,S2,
            W1,R,fname);
        [bHD3(j) b4K3(j)] = simulator2(lambda(i),p,n3,S3,
            W1,R,fname);
    end
    media4k(i) = mean(b4K);
    mediahd(i) = mean(bHD);
    media4k2(i) = mean(b4K2);
    mediahd2(i) = mean(bHD2);
```

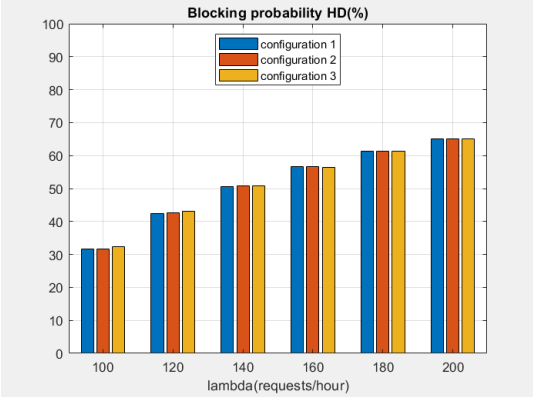
```
    media4k3(i) = mean(b4K3);
    mediahd3(i) = mean(bHD3);
end
figure(7)
bar(lambda,[media4k;media4k2;media4k3])
grid on
title('Blocking probability 4K(%)')
xlabel('lambda(requests/hour)')
ylim([0 100])

figure(8)
bar(lambda,[mediahd;mediahd2;mediahd3])
grid on
title('Blocking probability HD(%)')
xlabel('lambda(requests/hour)')
ylim([0 100])
```

Análise do Código

Mais uma vez, o código desenvolvido nesta alínea apenas difere do código da alínea anterior num único sítio, sendo esse sítio o valor da variável W, que passa agora a ser 600.

Resultados





Análises e Justificações

Com os resultados obtidos, é possível concluir que, comparativamente com o exercício anterior, e para um valor de reserva de recursos superior, a probabilidade de bloqueio de filmes 4K diminui e a probabilidade de bloqueio de filmes HD aumenta, ficando ambas sensivelmente equiparadas, o que significa que este é um bom valor de reserva de recursos. A mesma explicação é aplicável neste exercício.

Portanto, dado todas as varáveis impostas pela simulação (o número de servidores, a capacidade de cada interface, o número de pedidos por hora e a reserva de recursos), podemos mais uma vez concluir que, para este caso, a melhor solução a adotar seria a configuração 3 pois, novamente, é a configuração que, no geral, apresenta menores probabilidades de bloqueio para ambos os tipos de filme, como é visível nos gráficos de barras acima apresentados.

Task 2.e - considerando os dados do problema encontrar, por simulação, valores de n e W que correspondam aos valores de probabilidade pedidos

Código Matlab

```
Re = 100000;
Se = 10000;
pe = 24;
lambdae = 100000/24;
ne = 7;
We = 36500;
for j=1:10
    [bHD(j) b4K(j)] = simulator2(lambdae,pe,ne,Se,We,Re,
        fname);
end
media4k = mean(b4K);
mediahd = mean(bHD);

fprintf('n = %d W = %d\n',ne,We)
fprintf('blocking probability 4K = %.4e\n',media4k)
fprintf('blocking probability HD = %.10e\n',mediahd)
```

Análise do Código

Tal como anteriormente, são declaradas as variáveis com os valores dados no enunciado, é feito um ciclo for para as 10 corridas, é chamado o simulador 2 e calculados os valores das probabilidades e são apresentados os resultados sob a forma de prints. No entanto, os valores de n (número de servidores) e W (reserva de recursos para filmes 4K) não são dados e devem ser encontrados por simulação. Deste modo, deve ser corrido o programa até serem encontrados valores para n e W que satisfaçam os valores de probabilidade pedidos no enunciado.

Resultados

```
n = 7 W = 36500
blocking probability 4K = 0.0000e+00
blocking probability HD = 0.0000000000e+00
n = 6 W = 36500
blocking probability 4K = 6.0715e-01
blocking probability HD = 8.8184466535e-02
```

Análises e Justificações

Após várias simulações, podemos concluir que os primeiros valores que satisfazem o critério de a probabilidade de bloqueio ser inferior a 0.1% são: $n = 7$ e $w = 36500$.

Passando à justificação destes valores, para $n = 7$ e $w = 0$ já conseguimos ter um valor inferior a 0.1% e estes valores também satisfazem a condição seguinte de, em caso de falha de um servidor, a probabilidade de bloqueio ser inferior a 1%, isto é, para $n = 6$ e $w = 0$, a probabilidade de bloqueio também é inferior a 1%. Significa isto que foi encontrado o valor correto para n.

No entanto, quando existe falha de um servidor, estes valores levam a probabilidades de bloqueio muito diferentes para os diferentes tipos de filme. Deste modo, é necessário encontrar um valor para a reserva de recursos que, quando exista falha de um servidor, origine probabilidades de bloqueio semelhantes, sendo assim considerado um bom valor para a reserva de recursos. Após várias simulações, o valor para a reserva de recursos que melhor se ajusta a esta situação

é $w = 36500$, como se pode ser na figura apresentada acima, onde temos $n = 6$.
.

Task 3

Task 3.a - de acordo com o grafo fornecido, calcular quantas server famrs são precisas e a que AS devem estar ligadas, de modo a que o caminho mais curto entre um AS Tier-2 ou 3 e a server farm mais próxima tenha, no máximo, apenas um AS intermédio

Código Matlab

```
Gini= [ 1 2
 1 3
 1 4
 1 5
 1 6
 1 14
 1 15
 2 3
 2 4
 2 5
 2 7
 2 8
 3 4
 3 5
 3 8
 3 9
 3 10
 4 5
 4 10
 4 11
 4 12
 4 13
 5 12
 5 13
 5 14
 6 7
 6 16
 6 17
 6 18
 6 19
 7 19
 7 20
 8 9
 8 21
 8 22
```

```
 9 10
 9 22
 9 23
 9 24
 9 25
 10 11
 10 26
 10 27
 11 27
 11 28
 11 29
 11 30
 12 30
 12 31
 12 32
 13 14
 13 33
 13 34
 13 35
 14 36
 14 37
 14 38
 15 16
 15 39
 15 40
 20 21];
s = Gini(:,1);
t = Gini(:,2);
G = graph(s,t);

fid = fopen('exemplo.lp','wt');
fprintf(fid,'Minimize\n');
for i=6:40
    if i <= 15
        fprintf(fid,' + %f x%d',12,i);
    else
        fprintf(fid,' + %f x%d',8,i);
    end
end
fprintf(fid,'\nSubject To\n');
for j=6:40
    for i=6:40
        if length(shortestpath(G,i,j)) <= 3
            fprintf(fid,' + x%d',i);
        end
    end
```

```
82      fprintf(fid,' >= 1\n');
83  end
84  fprintf(fid,'Binary\n');
85  for i=6:40
86      fprintf(fid,' x%d\n',i);
87  end
88  fprintf(fid,'End\n');
89  fclose(fid);
```

Análise do Código

Primeiramente são declarados, em forma de matriz, os AS com ligação entre si. De seguida é passada para a variável s a primeira coluna da matriz anterior e para a variável t a segunda coluna, representando assim as arestas do grafo como um par de vértices. Essas duas variáveis, s e t, vão depois ser usadas para construir o grafo que representa as ligações entre AS. O próximo passo é criar e abrir um ficheiro para escrita e colocar nele o problema de programação linear inteira com o auxílio de ciclos for. Todos os ciclos for começam em 6 pois é o primeiro AS Tier-2 e vão até 40 pois é o último AS tier-3.

Inicialmente é escrito "Minimize" no ficheiro pois neste problema de otimização queremos minimizar o custo total das ligações à Internet. Depois, dentro do primeiro ciclo for, é associado a cada AS o respetivo custo, 12 para AS Tier-2 (que vão de 6 a 15, daí a restrição no if) e 8 para AS Tier-3, de modo a formar a expressão a minimizar e essa expressão é escrita no ficheiro. Seguidamente, é escrito "Subject" dentro do ficheiro, seguido das restrições que, no contexto deste problema, é o caminho mais curto entre um AS Tier-2 ou Tier-3 e a server farm mais próxima ter, no máximo, um AS intermédio. Por fim, os números dos AS são declarados como variáveis binárias para apenas tomarem os valores 0 e 1, sendo que AS com valor 1 significa que têm a si associada uma server farm.

Finalmente, o problema programação linear inteira presente no ficheiro é resolvido com recurso a um solver online.

Resultados

```
********** Begin .sol file *************

# Objective value = 48
x6 0
x7 0
x8 0
x9 1
x10 0
x11 0
x12 0
x13 1
x14 0
x15 0
x16 1
x17 0
x18 0
x19 0
x20 0
x21 1
x22 0
x23 0
x24 0
x25 0
x26 0
x27 0
x28 0
x29 0
x30 1
x31 0
x32 0
x33 0
x34 0
x35 0
x36 0
x37 0
x38 0
x39 0
x40 0

********** End .sol file ***************
```

Análises e Justificações

Após a obtenção dos resultados, podemos concluir que o número de *server farms* necessárias são 5 e encontram-se nas AS 9, 13, 16, 21 e 30. Dado o esquema da figura 4, que representa a cobertura da Internet, estes AS encontram-se uniformemente distribuídas pela rede, tendo fácil acesso aos restantes AS e, consequentemente, conseguindo cobrir facilmente toda a rede. O custo total de ligação à Internet é 48 (24+24) pois a conexão é feita através de 2 AS Tier-2 (AS 9 e 13, 12+12=24) e 3 AS Tier-3 (AS 16, 21 e 30, 8+8+8=24).

Task 3.b - determinar, por simulação, o número mínimo de servidores necessários e o valor de reserva de recursos correspondentes, de maneira a que nenhuma probabilidade de bloqueio seja superior a 1%

Código Matlab

```
R = 100000;
S = 1000;
p = 30;
lambda = ((5000*10)+(2500*25))/24;
n = 76;
W = 52000;
fname='movies.txt';
for j=1:10
    [bHD(j) b4K(j)] = simulator2(lambda,p,n,S,W,R,fname);
end
alfa= 0.1; %90% confidence interval%
media4k = mean(b4K);
term4k = norminv(1-alfa/2)*sqrt(var(b4K)/10);
mediahd = mean(bHD);
termhd = norminv(1-alfa/2)*sqrt(var(bHD)/10);

fprintf('n = %d W = %d\n',n,W)
fprintf('blocking probability 4K = %.4e\n',media4k+term4k)
fprintf('blocking probability HD = %.10e\n',mediahd+termhd)
```

Análise do Código

À semelhança da alínea e) da Task 2, este problema pede que sejam encontrados os valores de n e W, de modo a que as probabilidades de bloqueio não ultrapassem o limite estabelecido. Para tal, são declaradas as variáveis dadas no enunciado, é feito um ciclo for para efetuar 10 corridas, é chamado o simulador 2 e são calculados os valores das probabilidades de bloqueio e os intervalos de confiança. Também à semelhança da alínea e) da Task 2, os valores de n e W têm que ser obtidos por simulação e, para tal, deve ser corrido o programa várias vezes até serem encontrados valores que satisfaçam os requisitos do enunciado. No entanto, e contrastando com o que foi feito na alínea e) da Task 2, desta vez foi preciso garantir que ambos os intervalos, ou seja, a soma da probabilidade de bloqueio e o intervalo de confiança, são inferiores à probabilidade pedida.

Resultados

```
n = 76 W = 52000
blocking probability 4K = 9.3542e-01
blocking probability HD = 3.1473291710e-01
```

Análises e Justificações

Após várias simulações, chegámos à conclusão que os valores que satisfazem a condição da probabilidade de bloqueio ser inferior a 1% são: $n = 76$ e $w = 5200$. Para o mesmo valor de n e para valores de reserva de recursos de filmes 4k inferiores, a probabilidade de bloqueio dos filmes 4k torna-se superior a 1% e, portanto, não satisfaz as condições necessárias da simulação.

Para o valor de $n = 77$ e $w = 0$, a probabilidade de bloqueio é muito baixa, no entanto, possivelmente plausível, dado que a probabilidade de bloqueio mínima necessária para ser inferior a 1% é muito perto desse valor. No entanto, como queremos o número mínimo de servidores, a conclusão a que chegámos foi que esse número tem que ser 76, com um valor de reserva de recursos adequado que faz com que as probabilidades de bloqueio sejam sensivelmente iguais.

Dado um maior número de servidores, o critério de paragem usado para a simulação foi 100000, pois o intervalo de confiança é muito inferior à probabilidade de bloqueio imposta pela simulação.

Task 3.c - distribuir o número total de servidores pelas server farms, numa proporção o mais próxima possível do número de subscritores mais próximo de cada server farm

Resultados

Tal como indicado na alínea a) desta Task, foi determinado que devem existir 5 server farms, estando estas conectadas aos AS 9, 13, 16, 21 e 30. Sendo assim, a distribuição que consideramos ser a melhor é alocar 15 servidores às server farms dos AS 9, 16 e 30, perfazendo assim 45 servidores, alocar 17 servidores à server farm do AS 13 e alocar os restantes 14 servidores à server farm do AS 30.

Análises e Justificações

Visto que existem 5 server farms, 10 AS Tier-2 e cada AS que tem a si ligada uma server farm consegue abranger, pelo menos, dois AS Tier-2, optámos por fazer com que cada AS com server farm fique encarregue de dois AS Tier-2, de maneira a que cada server farm tenha, para já, o mesmo número de subscritores. Desta forma, a server farm associada ao AS9 abrange os AS 9 e 10, a server farm do AS13 abrange os AS 13 e 14, a server farm do AS16 abrange os AS 6 e 15, a server farm do AS21 abrange os AS 7 e 8 e a server farm do AS30 abrange os AS 11 e 12.

Passando agora à distribuição dos AS Tier-3 pelas server farms, uma vez que cada AS Tier-2 tem ligação a aproximadamente o mesmo número de AS Tier-3, decidimos que cada AS Tier-3 iria ser abrangido pela server farm que abrange o AS Tier-2 com o qual tem ligação direta. Assim sendo, a server farm do AS9 abrange os AS 23 27, a server farm do AS13 abrange os AS 33 e 38, a server farm do AS16 abrange os AS 16 a 18, 39 e 40, a server farm do AS21 abrange os AS 19 a 22 e a server farm do AS30 abrange os AS 28 e 32.

Com esta distribuição de servidores pelas server farms, e tendo em conta que existe um total de 112500 subscritores (5000*10 + 2500*25), as servers farms dos AS 9, 16 e 30 ficam encarregues de 22500 subscritores cada uma (correspondente a 20% do total de subscritores), a server farm do AS13 fica encarregue de 25000 subscritores (22,2% do total) e a server farm do AS21 fica encarregue de 20000 subscritores (17,8% do total). Assim sendo, as server farms dos AS 9, 16 e 30 ficam com 15 servidores cada uma (20% dos 76 servidores disponíveis), a server farm do AS13 fica com 17 servidores (22,2% de 76) e a server farm do AS21 fica com 14 servidores (17,8% de 76).