

Blocking Performance of Video-Streaming Services

20 de maio 2021

Todos os blocos de código, estão devidamente comentados para uma melhor análise de código.

Exercício 2

2a.

Simulator2

```
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 (in %)
    %n      = number of servers
    %S      = interface capacity of each server (in Mbps)
    %W      = resource reservation for 4K movies (in 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
    C = n*S;                  % Internet connection capacity (in Mbps)
    throughput_HD = 5;        % throughput of HD format
    throughput_4K = 25;        % throughput of 4K format

    %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:
    NARRIVALS = 0;
    REQUESTS_HD = 0;
```

```

REQUESTS_4K = 0;
BLOCKED_HD = 0;
BLOCKED_4K = 0;
%Simulation initial List of Events:
%%[Lower, i] = min(STATE);
EventList= [ARRIVAL exprnd(invlambda) 0]; %% 0

while NARRIVALS < R
    event= EventList(1,1);
    Clock= EventList(1,2);
    OldIdx = EventList(1,3);
    EventList(1,:)= [];
    if event == ARRIVAL
        [Lower, i] = min(STATE);
        EventList= [EventList; ARRIVAL Clock+exprnd(invlambda) 0];
        NARRIVALS= NARRIVALS+1;

        if (rand(1) > p/100)
            % if HD movie requested, it starts being transmitted by the least .
            % loaded server if it has
            % at least 5 Mbps of unused capacity and the total throughput of HD
            % movies does not become
            % higher than C - W Mbps;
            REQUESTS_HD = REQUESTS_HD+1;
            if STATE_HD + throughput_HD <= C-W && Lower <= S - throughput_HD
                STATE_HD = STATE_HD + throughput_HD;
                STATE(i) = STATE(i)+ throughput_HD;
                EventList= [EventList; DEPARTURE_HD
Clock+invmtiu(randi(Nmovies)) i];
            else
                % otherwise, the request is blocked
                BLOCKED_HD = BLOCKED_HD+1;
            end
        else
            % if 4k movie is requested, it starts being transmitted by the
            % least loaded server if it has
            % at least 25 Mbps of unused capacity;
            REQUESTS_4K = REQUESTS_4K+1;
            if Lower <= S - throughput_4K % state(i) lower
                STATE(i) = STATE(i) + throughput_4K;
                EventList= [EventList; DEPARTURE_4K
Clock+invmtiu(randi(Nmovies)) i];
            else
                % otherwise, the request is blocked.
                BLOCKED_4K = BLOCKED_4K+1;
            end
        end
    elseif event == DEPARTURE_HD

```

```

        STATE_HD = STATE_HD - throughput_HD;
        % total throughput of the movies in transmission by server i (
        STATE(OldIdx) = STATE(OldIdx) - throughput_HD;
    else
        STATE(OldIdx) = STATE(OldIdx) - throughput_4K;
    end
    EventList= sortrows(EventList,2);
end
bHD = 100 * BLOCKED_HD / REQUESTS_HD;    % blocking probability of HD in %
b4K = 100 * BLOCKED_4K / REQUESTS_4K;    % blocking probability of 4K in %
end

```

2a.

```

N = 10;                % number of simulations
lambdas = 100:20:200; % request arrival rate (in requests per hour)
p = 20;                % percentage of requests for 4K movies (in %)
R = 10000;             % stop simulation on ARRIVAL no. R
W = 0;                 % resource reservation for 4K movies (in Mbps)
n = 10;                % number of servers
S = 100;               % interface capacity of each server (in Mbps)
alfa = 0.1;            % 90 confidence interval%
%C = 1000;             % internet connect of server farm
fname = 'movies.txt'; % nome do ficheiro

% 100, 120, 140, 160, 180, 200
% arrays para guardar os valores que vao ser calculados
blockingHD = zeros(1, 6);
blocking4K = zeros(1, 6);
saveTermHD = zeros(1, 6);
saveTerm4K = zeros(1, 6);

for lambda = 1:6
    results_HD = zeros(1,N); %vector with N simulation results
    results_4K = zeros(1,N); %vector with N simulation results

    % iterar 10 vezes, para correr o simulator2 10 vezes
    for it = 1:10
        [results_HD(it), results_4K(it)] =
simulator2(lambdas(lambda),p,n,S,W,R,fname);
    end

    % calcular a media dos resultados dos filmes hd
    blockingHD(lambda) = mean(results_HD);
    saveTermHD(lambda) = norminv(1-alfa/2)*sqrt(var(results_HD)/ N);

```

```

    % calcular a media dos resultados dos filmes 4k
    blocking4K(lambda) = mean(results_4K);
    saveTerm4K(lambda) = norminv(1-alfa/2)*sqrt(var(results_4K)/ N);
end

figure(1)

bar(lambdas,blockingHD)

hold on

er = errorbar(lambdas,blockingHD(1,:) , -saveTermHD(1,:),saveTermHD(1,:));
er.Color = [0 0 0];
er.LineStyle = 'none';

hold off
ylim([0 100])
xlabel('\lambda (requests/hour)')
title('Blocking probability HD (%)')

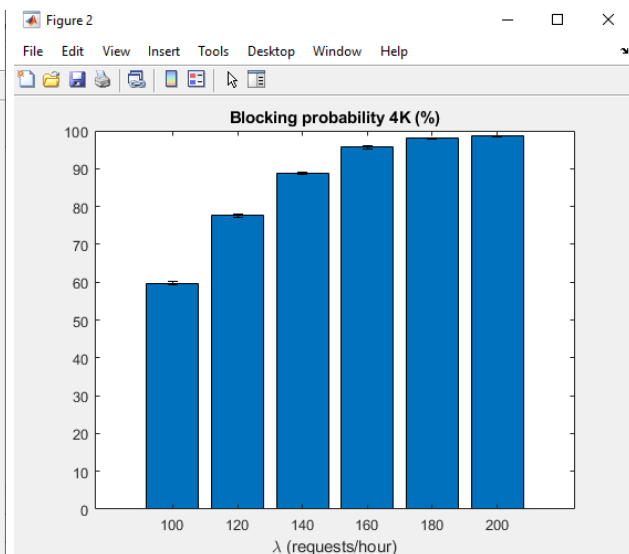
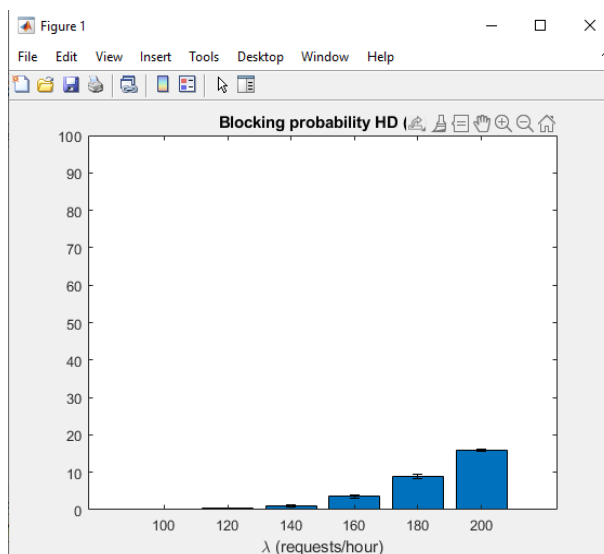
figure(2)
bar(lambdas,blocking4K(1,:))

hold on

er = errorbar(lambdas,blocking4K(1,:), -saveTerm4K(1,:),saveTerm4K(1,:));
er.Color = [0 0 0];
er.LineStyle = 'none';

hold off
xlabel('\lambda (requests/hour)')
title('Blocking probability 4K (%)')

```



Ao analisar os respectivos gráficos relativos à probabilidade de bloqueio, tanto em vídeos 4k, como em vídeos HD, podemos observar, que quanto maior forem as taxas de pedidos de filmes por hora (100 a 200) mais carga o server farm vai ter de operar. É de esperar também, que as probabilidades de bloqueio relativamente ao videos HD sejam bastantes inferiores às dos videos 4k, visto que quando é pedido um vídeo HD, ele necessita de pelo menos 5 Mbps de capacidade livre, ao contrário dos vídeos 4k que necessitam de um valor de 25 Mbps para não ser bloqueado.

Analisando o número de pedidos de filmes para parar a simulação, as grandezas de erro são bastante inferiores às grandezas das probabilidades de bloqueio, logo o critério de paragem já é auto suficiente.

2b.

```
N = 10; % number of simulations
lambdas = 100:20:200; % request arrival rate (in requests per hour)
p = 20; % percentage of requests for 4K movies (in %)
R = 10000; % stop simulation on ARRIVAL no. R
W = 0; % resource reservation for 4K movies (in Mbps)
n = [10,4,1]; % number of servers
S = [100,250,1000]; % interface capacity of each server (in Mbps)
alfa = 0.1; % 90 confidence interval%
%C = 1000; % internet connect of server farm
fname = 'movies.txt'; % nome do ficheiro

blockingHD = zeros(3, 6);
blocking4K = zeros(3, 6);
saveTermHD = zeros(3, 6);
saveTerm4K = zeros(3, 6);

it = 1;

for lambdas = 100:20:200
    for configuration = 1 : length(n)
        results = zeros(2,N);

        for it2=1:N
            [results(1,it2), results(2,it2)] =
simulator2(lambdas,p,n(configuration),S(configuration),W,R,fname);
```

```

end

media_blockingHD = mean(results(1,:));
media_blocking4K = mean(results(2,:));

termHD = norminv(1-alfa/2)*sqrt(var(results(1,:))/N);
term4K = norminv(1-alfa/2)*sqrt(var(results(2,:))/N);

blockingHD(configuration, it) = media_blockingHD
saveTermHD(configuration, it) = termHD
blocking4K(configuration, it) = media_blocking4K
saveTerm4K(configuration, it) = term4K

end
it = it + 1;

end
xgraph = 100:20:200;
figure;

bar(xgraph,blockingHD)
ylim([0 100])
grid on;

hold on

legend('Configuration1','Configuration2', 'Configuration3','Location','northwest');

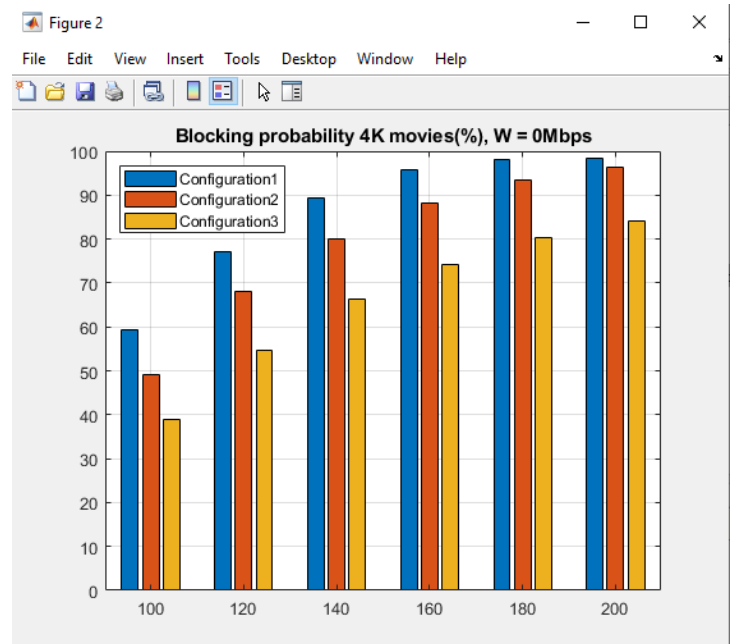
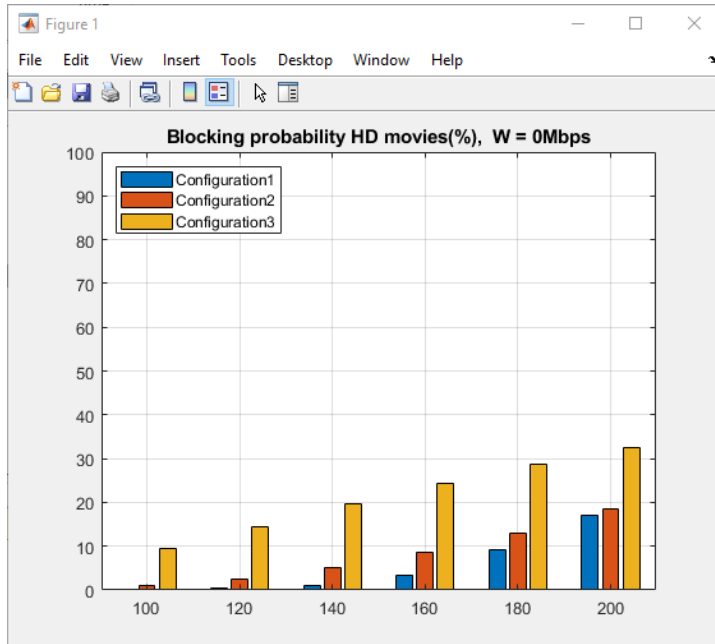
title('Blocking probability HD movies(%), W = 0Mbps');
hold off

figure;
bar(xgraph,blocking4K)
grid on;
hold on

legend('Configuration1','Configuration2', 'Configuration3','Location','northwest');

title('Blocking probability 4K movies(%), W = 0Mbps');
hold off

```



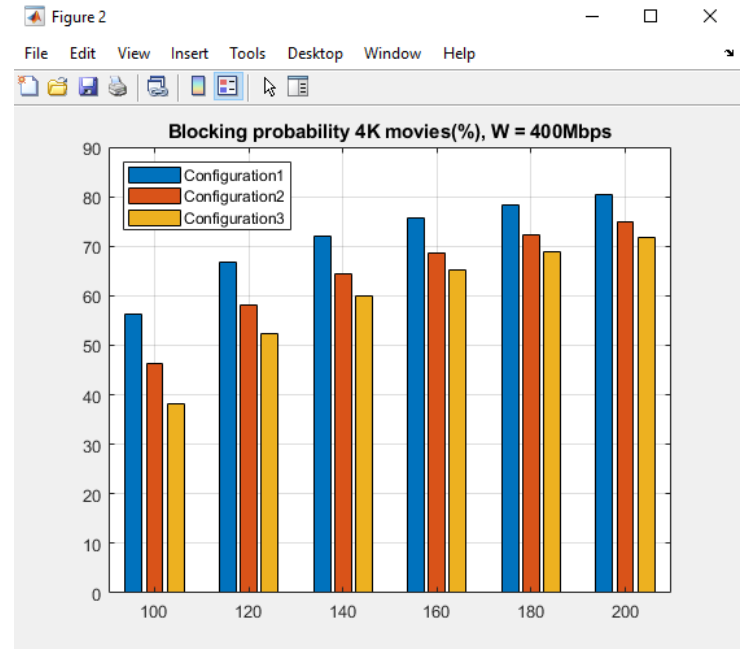
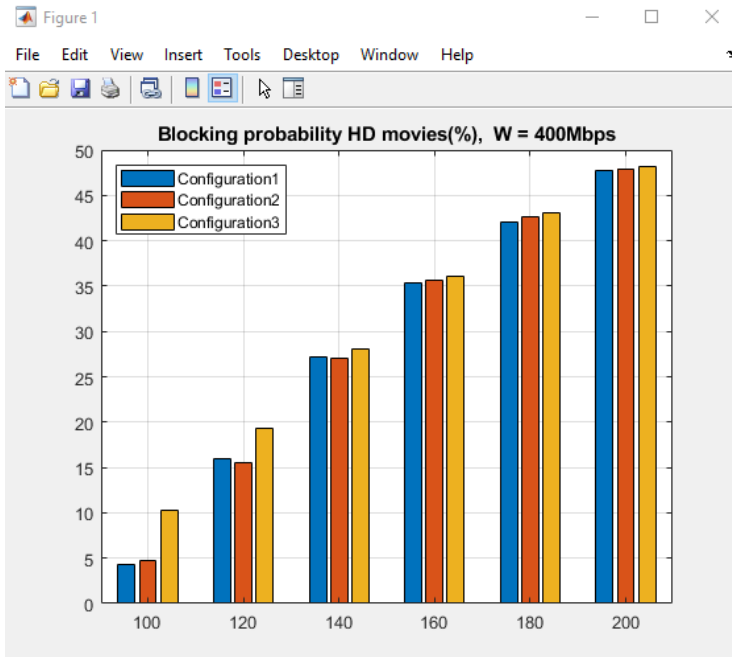
Neste exercício, calculamos as probabilidades de bloqueio para os ambos tipos de vídeos, para as três configurações pedidas, que diferem no número de servidores e também nos Mbps da network interface.

Ao analisar os gráficos, conseguimos reparar que para os vídeos no formato 4K, a probabilidade de bloqueio é menor na configuração 3, onde existe um menor número de servidores com uma network interface de valor maior, que as restantes configurações.

À medida que são requisitados novos filmes, e esses filmes forem do tipo de video 4K, sabemos que necessitam de um valor de 25 Mbps para não ser bloqueado. Existindo baixas larguras de banda, vai ser superior a probabilidade de esta estar mais próxima da sua capacidade total, logo essa probabilidade de bloqueio vai ser maior.

Analisando o gráfico para a probabilidade de bloqueio nos videos de formato HD, conseguimos observar que existe uma probabilidade maior de bloqueio com a configuração 3, onde existe um menor número de servidores e uma maior interface de rede.

2c.

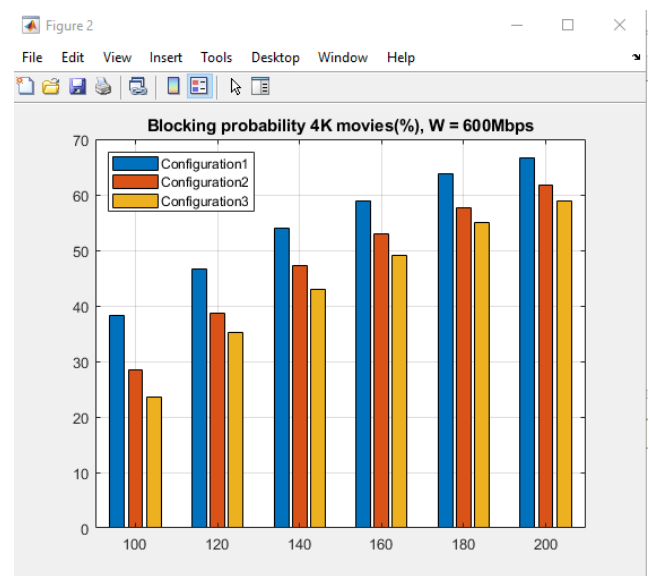
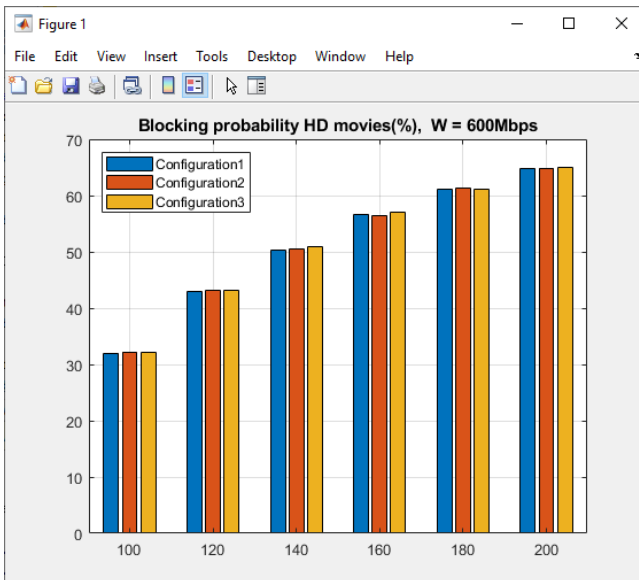


Neste exercício foi alterado o valor de reserva de recursos do valor de 0 para 400.

Analisando o valor destes gráficos, com os gráficos relativos ao exercício anterior, conseguimos observar que relativamente aos vídeos 4K, os valores da probabilidade de bloqueio, são ligeiramente mais baixo com o valor de W diferente pois existe uma reserva de recursos.

Analisando o valor destes gráficos, com os gráficos relativos ao exercício anterior, conseguimos observar que relativamente ao vídeos HD, os valores da probabilidade de bloqueio aumentaram significativamente. Como a reserva de recursos é de 400, e os vídeos HD entram em bloqueio quando a taxa de transferência é superior a $C - W$, com $C = 1000$ e $W = 400$, logo 600Mbps, observamos que a melhor configuração é a 3 pois tem uma menor probabilidade de bloqueio.

2d.



Neste exercício, relacionando com o exercício anterior, foi alterado o valor de reserva de recursos de 400 para 600. Analisando os gráficos, observamos que no caso dos vídeos serem do formato HD, os valores do bloqueio aumentam ligeiramente, quando aos valores de bloqueio dos vídeos com formato 4K, podemos observar que diminuíram um pouco.

Como a reserva de recursos é de 600, e os vídeos HD entram em bloqueio quando a taxa de transferência é superior a $C - W$, com $C = 1000$ e $W = 600$, logo 400Mbps, observamos que a melhor configuração é a 3 pois tem uma menor probabilidade de bloqueio.

2e.

```
lambda = 100000/24;
p = 24;
n = 6;
S = 10000;
W = 36500;
R = 100000;
fname='movies.txt';
blockingHD = zeros(1,10);
blocking4K = zeros(1,10);

for i=1:10
    [blockingHD(i),blocking4K(i)] = simulator2(lambda,p,n,S,W,R,fname);
end

mediaHD = mean(blockingHD)
media4K = mean(blocking4K)
```

Era necessário correr várias simulações (10 de cada vez) para tentar atingir valores de 'n' e 'W' que resultassem nos valores de probabilidade pedidos.

Para n=6 e W=36500 as probabilidades podem ser:

mediaHD = 0.0714

media4K = 0.5069

Para n=7 e W=36500 as probabilidades podem ser:

mediaHD = 0

media4K = 0

Para n=7 e W=0, conseguimos obter menos de 0.1% de probabilidade de bloqueio. Em caso de falha de um servidor (isto é, n=6), continuamos a ter menos de 1%. Está encontrado o valor de 'n'.

Temos de ter em conta que, quanto um servidor falha, estes valores de 'n' e 'W' geram probabilidades de bloqueio bastante diferentes para filmes HD e 4K. Depois de várias simulações, assumimos que W=36500 é o valor de reserva de recursos que melhor reduz essa diferença.

Exercício 3

3a.

```
G= [ 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 = G(:,1);
t = G(:,2);
D = graph(s,t);
plot(D)
%P = shortestpath(D,39,32)

I = zeros(40);
C = zeros(1,40);

for j = 6:1:40
    for i = 6:1:40
        P = shortestpath(D,j,i);
        if (length(P) <= 3)
            % pertence a I(j)
            I(j,i) = 1;
        end
    end
    if (j < 16)
        C(j) = 12;
    else
        C(j) = 8;
    end
end

fid = fopen('3a.lp','wt');

fprintf(fid,'Minimize\n');
for i = 6:1:40
    fprintf(fid,' + %f x%d',C(i),i);
end

```

```

fprintf(fid, '\nSubject To\n');
for j = 6:1:40
    for i = 6:1:40
        if (I(j,i) == 1)
            fprintf(fid, ' + x%d', i);
        end
    end
    fprintf(fid, ' >= 1\n');
end

fprintf(fid, 'Binary\n');
for i=6:40
    fprintf(fid, ' x%d\n', i);
end

fprintf(fid, 'End\n');

fclose(fid);

```

‘G’ representa as ligações entre os vários AS’s (‘s’ é a sua primeira coluna e ‘t’ é a segunda). Com essa informação, criamos um grafo ‘D’, com todas as ligações.

De seguida, iteramos cada nó (ou AS, do 6 ao 40, isto é, Tiers 2 e 3), e em cada iteração ‘j’, voltamos a iterar todos os nós ‘i’, calculando o “shortest path” (caminho mais curto) de ‘j’ para ‘i’. Se o comprimento desse caminho for inferior a quatro, isto é, se dois AS’s tiverem um ou menos AS’s intermédios entre eles, então vamos dizer que a relação entre ‘j’ e ‘i’ é importante, e guardamos essa informação numa matriz ‘I’ 40 por 40, em que $I(j,i)$ está a 1 se essa relação é importante, caso contrário está a 0. Guardamos ainda, na iteração ‘j’, o custo da ligação desse nó/AS, num vetor C.

De seguida iniciamos a escrita (num ficheiro) do problema de programação linear inteira.

Após analisar, verifica-se que o custo das conexões internet é de 48 e que o número de server farms necessários são 5 e, estavam presentes nos AS número 9, 13, 16, 21 e 30.

O custo total de ligação à internet é 48. Sabendo que o custo da conexão da internet de um server farm é 12 quando existe uma conexão com um server farm de Tier 2, e 8 quando existe uma conexão com um server farm de Tier 3. Existem então 2 server farms de Tier 2 e 3 server farms 3 de Tier 3, existindo então um custo de 48.

3b.

Nesta alínea utilizamos o bloco de código do exercício 2e) onde alteramos alguns parâmetros, o λ utilizado é o número de subscritores multiplicado pelo número de AS de Tier 2 e Tier3. Então esse valor será $(5000 * 10) + (2500 * 25) / 24$, dividido por 24 pois existe em média de um pedido de filme por dia (24 horas). Foi utilizado também $S = 21000$ e $p = 30$;

Após vários testes e várias simulações, o número total de servidores necessários e a reserva W para fornecer uma probabilidade de bloqueio de no máximo 1%, para quando existem solicitações de ambos os formatos de vídeo são 76 e 52000 respectivamente.

```
R = 50000;
S = 1000;
p = 30;
lambda = ((5000 * 10) + (2500 * 25)) / 24;
fname = 'movies.txt';
n = 76;
W = 0;
alfa = 0.1;
blockingHD = zeros(1,10);
blocking4K = zeros(1,10);
for i=1:10
    [blockingHD(i),blocking4K(i)] = simulator2(lambda,p,n,S,W,R,fname);
end

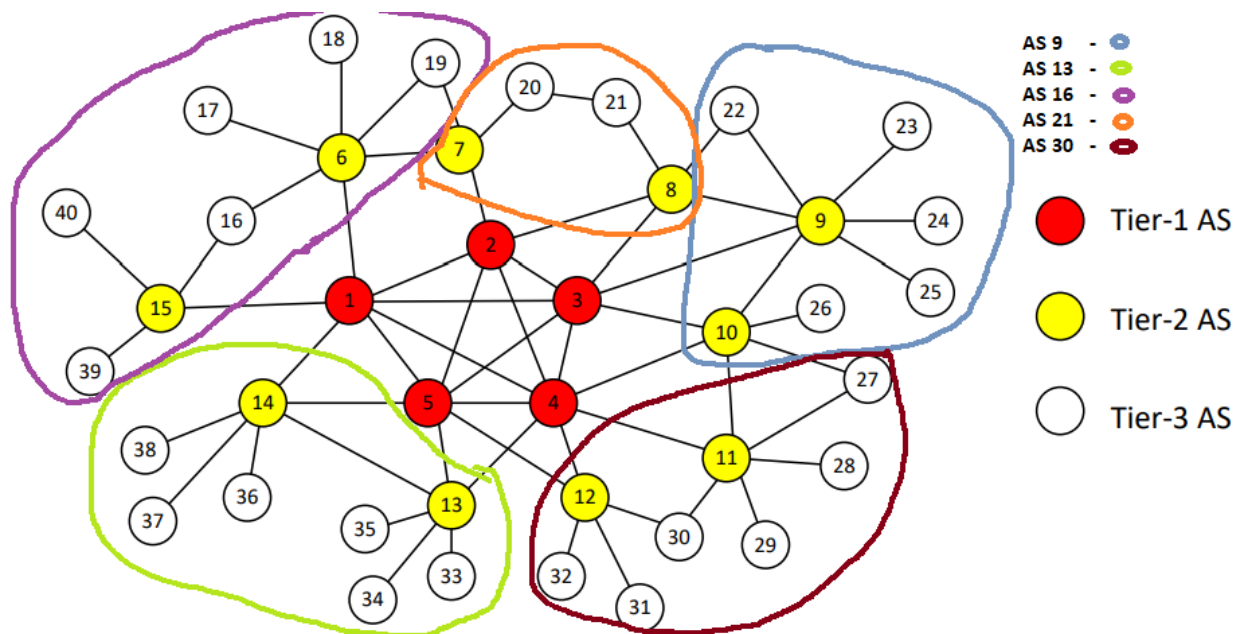
mediaHD = mean(blockingHD)
media4K = mean(blocking4K)
termHD = norminv(1-alfa/2)*sqrt(var(results(1,:))/N);
term4k = norminv(1-alfa/2)*sqrt(var(results(1,:))/N);
fprintf('Blocking Probability HD = %.2e +-.2e\n',mediaHD, termHD)
fprintf('Blocking Probability 4K = %.2e +-.2e\n',media4K, term4k)
```

Foi necessário um $R = 50000$ para também conseguir fornecer uma probabilidade de bloqueio de no máximo 1%.

3c.

O serviço de vídeo-streaming consegue alcançar um número de 5000 subscritores em cada AS de Tier2 e 2500 subscritores em cada AS de Tier3.

Sabendo que é necessário saber quantos servers estão em operação em cada server farm, foi então dividido o número total de server que foram determinados na alinea b) (76 servers), por os AS determinados na alinea a) (9, 13, 16, 21, 30). Sabendo que a solução calculada na alinea a) deve garantir o caminho mais curto entre qualquer AS de Tier2 ou Tier3 para o server farm mais próximo não pode ter mais de 1 intermediário. Logo para a nossa solução, calculamos os AS mais próximos de cada AS com server farms e também o seu respetivo número de clientes, podemos ver a divisão na seguinte imagem.



De seguida para cada AS é calculado o número de clientes e posteriormente o número de servidores.

	Número de clientes	Número de servidores
AS9	$5000*2 + 5*2500 = 22500$	$76 * (22500/112300) = 15.2$
AS13	$5000*2 + 6*2500 = 25000$	$76 * (25000/112300) = 16.9$
AS16	$5000*2 + 6*2500 = 25000$	$76 * (25000/112300) = 16.9$
AS21	$5000*2 + 2*2500 = 15000$	$76 * (15000/112300) = 10.1$
AS30	$5000*2 + 6*2500 = 25000$	$76 * (25000/112300) = 16.9$

Arredondando o valor dos servidores, conseguimos observar que o número de servidores é suficiente pois, iguala o número de servidores de 76.