DEPARTAMENTO DE ELETRÓNICA, TELECOMUNICAÇÕES
E INFORMÁTICA

MESTRADO INTEGRADO EM ENG. DE COMPUTADORES E TELEMÁTICA

ANO 2020/2021

DESEMPENHO E DIMENSIONAMENTO DE REDES

ASSIGNMENT GUIDE NO. 2

BLOCKING PERFORMANCE OF VIDEO-STREAMING SERVICES

1. Preamble

The aim of this assignment is to assess the blocking performance of video-streaming services. In its simplest form, these services are provided by a single server, as illustrated in Figure 1. The server has a catalogue of video items (movies or series episodes), each one with a given duration, to be selected by the service subscribers. Video items can be available in one or more video formats, depending on possible different types of subscribers.

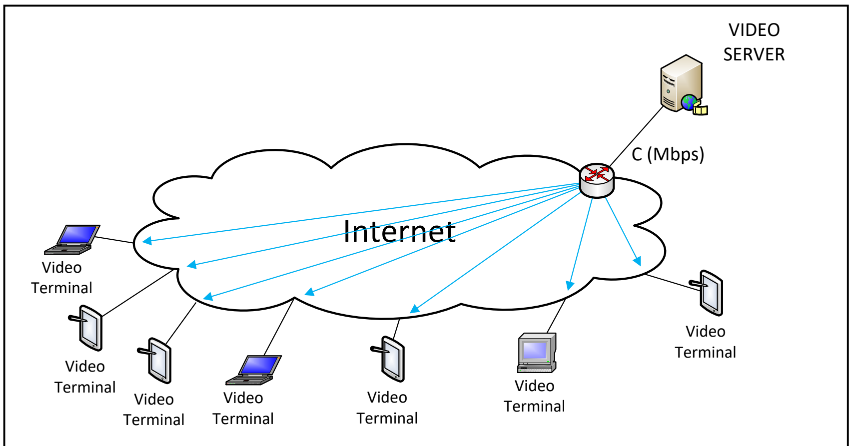
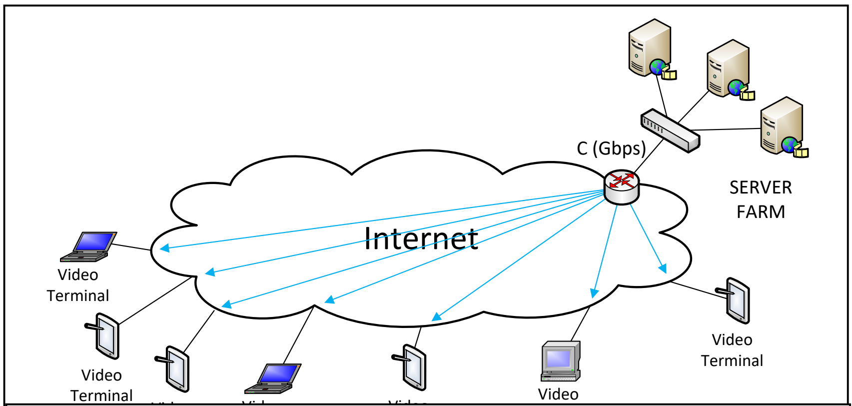


Figure 1

In alternative, the video-streaming service is provided by a server farm, located on a single Data Centre (DC), as illustrated in Figure 2. The advantages of using a server farm are (i) to scale the service to a larger number of subscribers and (ii) to make the service robust to server failures.



Finally, on its most general case, the service is provided by multiple server farms hosted on different DC sites, as illustrated in Figure 3. The advantages of using multiple server farms are (i) to further scale the services to an even larger number of subscribers, (ii) to decrease the average routing distance (and, consequently, round-trip-time delay) between subscriber and server locations and (iii) to make the service robust not only to server failures but also to DC site failures. In this case, the number of server farms and, for each farm, its location, its number of servers and its Internet connection capacity is a layout problem which typically involves some sort of optimization.

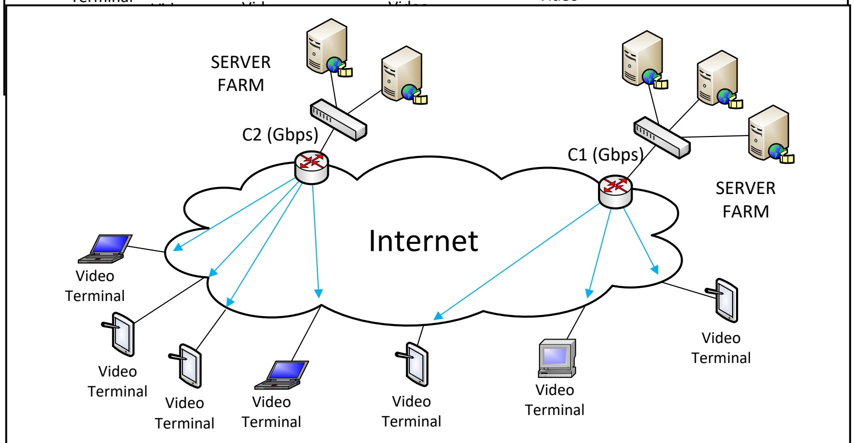


Figure 3

Assignment Guide No. 2

Assignment Description

Implement the following tasks using MATLAB to obtain the requested numerical solutions and conclusions. At the end, submit a report with the answers to the questions of the tasks requested for reporting including the numerical results, the MATLAB codes duly explained and the requested conclusions.

Task 1

In this task, consider a video-streaming service provided by a single server, as presented in Figure 1, whose Internet connection has a capacity of *C* (in Mbps). Consider that the server provides a set of movies on a single video format, each movie requires a throughput of *M* (in Mbps) and all movies have the same probability of being requested. The server has a catalogue of 2814 items (between movies and series episodes) with an average movie duration of 86.3 minutes and the duration (in minutes) of each item is in file *movies.txt*.

When a movie is requested by a subscriber, it starts being transmitted by the server if the resulting total throughput is within the Internet connection capacity; otherwise, the request is blocked. Consider that arrival of movie requests is a Poisson process with an average rate of λ (in requests/hour). Appendix A provides a MATLAB function named simulator1 which implements an event driven simulator for the video-streaming service based on a single server and providing movies with a single video format. The input parameters of simulator1 are:

- λ – movie request rate (in requests/hour)
- *C* – Internet connection capacity (in Mbps)
- *M* – throughput of each movie (in Mbps)
- *R* – number of movie requests to stop the simulation

The performance parameters estimated by simulator1 are:

- *b* – blocking probability (percentage of movie requests that are blocked)
- *o* – average occupation of the Internet connection (in Mbps)

The stopping criterion is the time instant of the arrival of the movie request number *R*. Simulator1 considers:

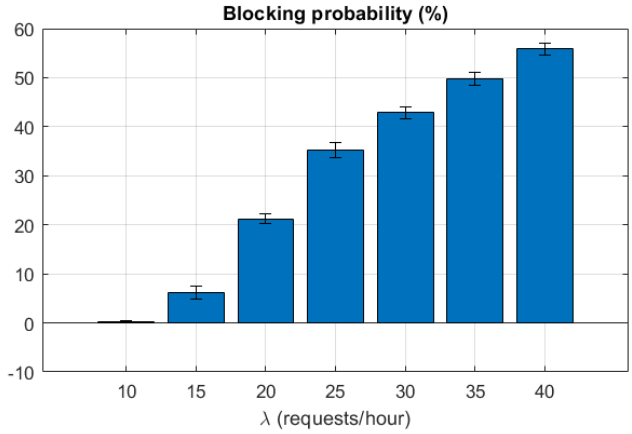
- events: ARRIVAL (the time instant of a movie request) and DEPARTURE (the time instant of a movie termination);
- state variable: STATE (total throughput of the movies in transmission);
- statistical counters: OCCUPATION (the integral of connection occupation up to the current time instant), REQUESTS (number of movie requests up to current time instant) and BLOCKED (number of blocked requests up to the current time instant).

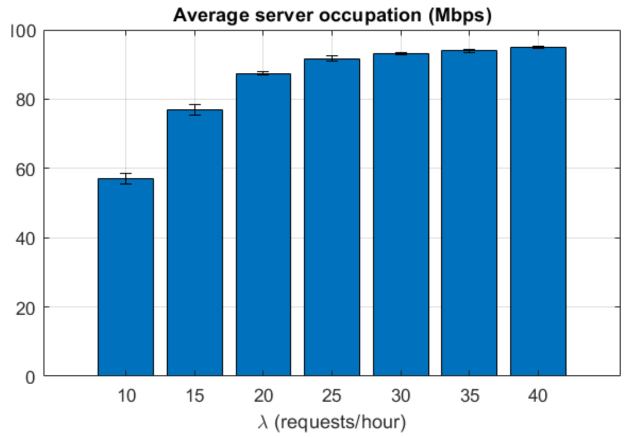
1.a. Develop a MATLAB script to run simulator1 10 times with a stopping criterion of *R* = 500 at each run and to compute the estimated values and the 90% confidence intervals of both performance parameters (see Module3, slide 22, of theoretical classes) when λ = 20 requests/hour, *C* = 100 Mbps and *M* = 4 Mbps. Answer (recall that these are simulation results):

```
Blocking probability (%) = 2.1360e+01 +- 1.45e+00

Average occupation (Mbps)= 8.6975e+01 +- 8.59e-01
```

1.b. Develop a MATLAB script to run simulator1 10 times with a stopping criterion of R = 500 and to compute the estimated values and the 90% confidence intervals of both performance parameters when λ = 10, 15, 20, 25, 30, 35 and 40 requests/hour, C = 100 Mbps and M = 4 Mbps. Present the results and the confidence intervals in bar charts with error bars[1]. Answer:

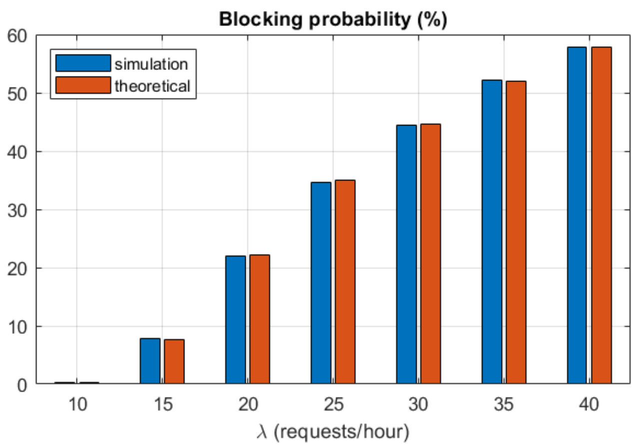


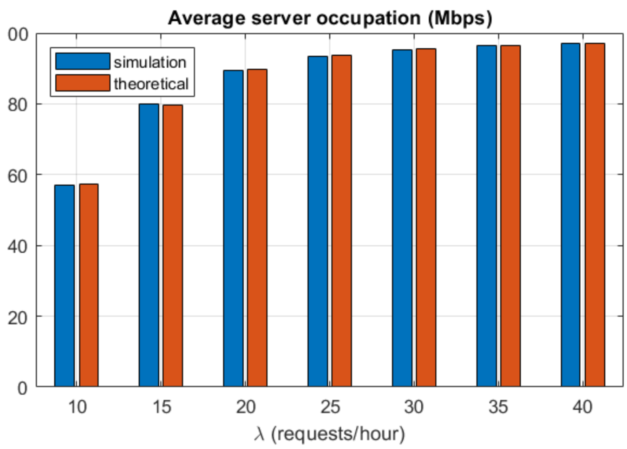


1.c. Repeat question **1.b** but consider now a stopping criterion of R = 5000 (10 times higher than the previous value). Present again the results in bar charts with error bars. Compare the obtained confidence intervals of these and the last results and take conclusions.

1.d. Run simulator1 10 times with a stopping criterion of R = 5000 to compute the estimated values and the 90% confidence intervals of both performance parameters when λ = 100, 150, 200, 250, 300, 350 and 400 requests/hour, C = 1000 Mbps and M = 4 Mbps. Present again the results in bar charts with error bars. With these results and the results of previous question **1.c**, take conclusions on the impact of the input parameters on the performance parameters.

1.e. Assume that the duration of the movies is an exponential distributed random variable with the same average duration of the items catalogue, i.e., $1/\mu$ = 86.3 minutes. In this case, the system can be modelled by an *M*/*M*/*m*/*m* queuing system. Determine the analytical values of the blocking probability and the average connection occupation (see Appendix B) for the cases considered in question **1.c**. Present both theoretical and simulation results in bar charts. Is the *M*/*M*/*m*/*m* queuing system a good approximation of the simulated system? Answer:





[1] https://www.mathworks.com/help/matlab/creating_plots/bar-chart-with-error-bars.html

1.f. Repeat question **1.e** now for the cases considered in question **1.d**. Present both theoretical and simulation results in bar charts. Is the *M*/*M*/*m*/*m* queuing system a good approximation of the simulated system in these cases?

Task 2 – for reporting (evaluation weight = 50%)

In this second task, consider a video-streaming service provided by one server farm, as presented in Figure 2, with n servers where each server has an interface of S Mbps and assuming that the Internet connection of the server farm is $C = n \times S$ Mbps. Consider also that the server farm provides movies on 2 possible video formats: HD with a throughput of 5 Mbps and 4K with a throughput of 25 Mbps. Consider the same catalogue of items as in the previous task. All catalogue items are available in both formats in all servers and, again, all items have the same probability of being requested.

Consider a front-office system that assigns movie requests to servers using a load balancing strategy (i.e., each request is assigned to the least loaded server) and implements admission control with a resource reservation of W (in Mbps) for 4K movies (i.e., HD movies cannot occupy more than $C - W$ Mbps). In more detail, the admission control is as follows:

- When a 4K movie is requested, it starts being transmitted by the least loaded server if it has at least 25 Mbps of unused capacity; otherwise, the request is blocked.
- When a HD movie is 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; otherwise, the request is blocked.

Consider that all movie requests are a Poisson process with an average rate of λ (in requests/hour) and that p (in percentage) of the requests are for movies of 4K format. Develop a MATLAB function, with the name simulator2, implementing an event driven simulator for this case to estimate the blocking probability of movie requests of each format (see Appendix C).

In the following questions of this task, consider 3 server farm configurations with the same Internet connection capacity of $C = 1000$ Mbps:

Configuration 1: the server farm is composed by $n = 10$ servers and each server has a network interface of $S = 100$ Mbps;

Configuration 2: the server farm is composed by $n = 4$ servers and each server has a network interface of $S = 250$ Mbps;

Configuration 3: the server farm is composed by $n = 1$ server with a network interface of $S = 1000$ Mbps.

2.a. Develop a MATLAB script to run 10 times simulator2 with a stopping criterion of $R = 10000$ and to compute the estimated values and the 90% confidence intervals of both blocking probabilities. Consider *Configuration 1* for $\lambda = 100$, 120, 140, 160, 180 and 200 requests/hour, $p = 20\%$, and a resource reservation $W = 0$ Mbps. Present the results and the confidence intervals in bar charts with error bars. Analyse the results and take conclusions on (i) the impact of the arriving rate λ of the movie requests in the blocking probability of each movie format and (ii) if the stopping criterion value is large enough or should be larger.

2.b. Repeat the simulations requested in question **2.a** but now considering *Configurations 2* and *3* (consider the same values of all other parameters). Present the bar charts of these results together with the previous results on a single figure for the blocking probability of HD movie requests and another single figure for the blocking probabilities of 4K movie requests (for clarity, do not include the error bars). Analyse the results and take conclusions on the impact of the 3 server farm configurations in the blocking probability of each movie format.

2.c. Repeat the simulations for all 3 server farm configurations but now considering a resource reservation $W = 400$ Mbps. Again, present the bar charts of these results on a single figure for the blocking probability of HD movie requests and another single figure for the blocking probabilities of 4K movie requests (and do not include the error bars). Compare these results with the previous results and take conclusions on the impact of the resource reservation in the blocking probability of each movie format for the 3 server farm configurations.

2.d. Repeat the previous question **2.c** but now considering a resource reservation $W = 600$ Mbps and present the results in the same way as before. Compare these results with the previous results and take conclusions on the impact of the different resource reservation values in the blocking probability of each movie format for the 3 server farm configurations.

2.e. Consider a video-service company with 100000 (one hundred thousand) subscribers, among which 24% are golden subscribers and 76% are regular subscribers (i.e., a golden subscriber is provided with movies in 4K format while a regular subscriber is provided with movies in HD format). Both types of subscribers are expected to request, on average, 1 movie per day.

The company aims to have a robust server farm solution such that the worst blocking probability among the two types of subscribers is not higher than 0.1% when all servers are working and 1% when one server fails. Consider that each server has an interface of $S = 10$ Gbps (= 10000 Mbps). Determine by simulation (using a stopping criterion of $R = 100000$) the minimum number of required servers and a proper reservation value W to be set in the front-office of the service to meet the required blocking performance.

Task 3 – for reporting (evaluation weight = 50%)

In this third task, consider a video-streaming service provided by multiple server farms (see Figure 3 in the Preamble) where, again, movies are available in the 2 previous possible video formats (HD with a throughput of 5 Mbps and 4K with a throughput of 25 Mbps). Consider the same catalogue of items as in the previous tasks and that all catalogue items are available in both formats in all servers. Consider in all server farms that each server has an interface of 1000 Mbps.

Consider that the Internet part that covers the subscribers of the service is given by Figure 4 (in the next page), which specifies the different types of Autonomous Systems (ASs) and how they are connected (the list of pairs of connected ASs is provided as a matrix G in MATLAB format in Appendix D). The total number of Tier-2 ASs is 10 and the total number of Tier-3 ASs is 25.

Besides determining the total number of servers required in all server farms, the video-streaming service provider also needs (i) to identify the ASs where the different server farms must be connected and (ii) to decide how many servers must be put in operation on each server farm. Consider that only Tier-2 and Tier-3 ASs provide the Internet access service and, therefore, can connect the server farms to the Internet.

The cost of the Internet connection of a server farm is 12 when the connection is provided by a Tier-2 AS and 8 when the connection is provided by a Tier-3 AS. The video-streaming service provider assumes that it can reach 5000 subscribers on each Tier-2 AS and 2500 subscribers on each Tier-3 AS with average requests of 1 movie per day per subscriber and 30% of requests for movies of 4K format.

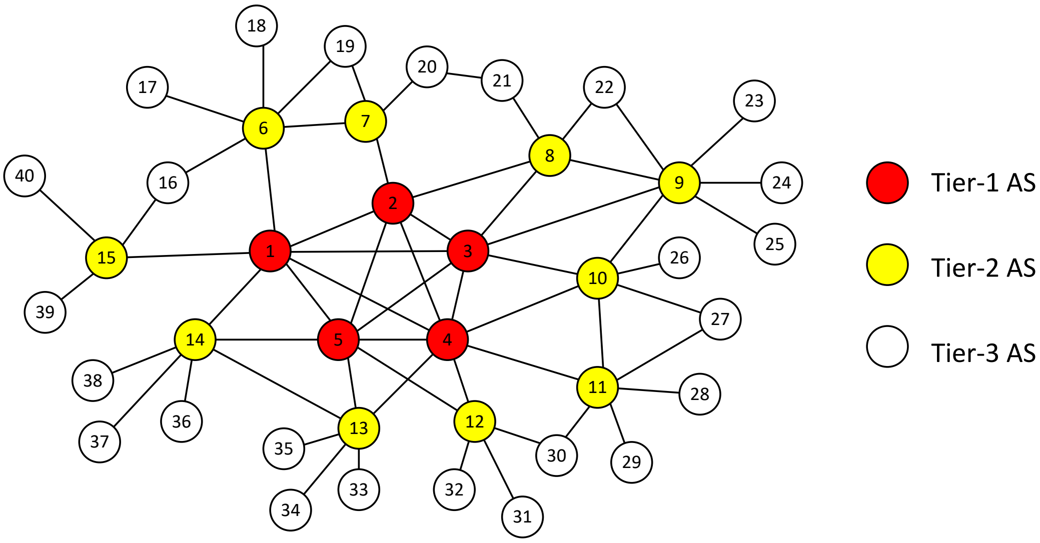


Figure 4

3.a. Compute the ASs to connect the server farms such that the total cost of the Internet connections is minimized (see Appendix E). The solution must guarantee that the shortest path from any Tier-2 or Tier3 AS to the closest server farm has no more than 1 intermediate AS. What is the total Internet connections cost of the solution? How many server farms are required and in which ASs they must be connected to? Analyse the selected ASs and take conclusions.

3.b. Use simulator2 to determine the total number of required servers and the appropriate reservation *W* to provide a blocking probability of at most 1% for movie requests of both formats. Present the simulation results with confidence intervals guaranteeing that both intervals are fully below the required blocking probability value. What was the required stopping criterion value of the simulations?

3.c. To define the final solution (i.e., how many servers are put in operation on each server farm), split the total number of servers determined in **3.b** by the ASs determined in **3.a** in a proportion as close as possible to the number of subscribers that are closer to each server farm. How many servers are installed on each server farm? Analyse the final solution and take conclusions.

Appendix A – Proposed MATLAB function for Simulator 1

To run the function:

```
lambda= 20;
C= 100;
M= 4;
R= 500;
fname= 'movies.txt';
[b o]= simulator1(lambda,C,M,R,fname);
```

Function:

```
function [b o]= simulator1(lambda,C,M,R,fname)
    %lambda = request arrival rate (in requests per hour)
    %C=       Internet connection capacity (in Mbps)
    %M=       throughput of each movie (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

    %Events definition:
    ARRIVAL= 0;         %movie request
    DEPARTURE= 1;       %termination of a movie transmission
    %State variables initialization:
    STATE= 0;
    %Statistical counters initialization:
    OCCUPATION= 0;
    REQUESTS= 0;
    BLOCKED= 0;
    %Simulation Clock and initial List of Events:
    Clock= 0;
    EventList= [ARRIVAL exprnd(invlambda)];

    while REQUESTS < R
        event= EventList(1,1);
        Previous_Clock= Clock;
        Clock= EventList(1,2);
        EventList(1,:)= [];
        OCCUPATION= OCCUPATION + STATE*(Clock-Previous_Clock);
        if event == ARRIVAL
            EventList= [EventList; ARRIVAL Clock+exprnd(invlambda)];
            REQUESTS= REQUESTS+1;
            if STATE + M <= C
                STATE= STATE+M;
                EventList= [EventList; DEPARTURE Clock+invmiu(randi(Nmovies))];
            else
                BLOCKED= BLOCKED+1;
            end
        else
            STATE= STATE-M;
        end
        EventList= sortrows(EventList,2);
    end
    b= 100*BLOCKED/REQUESTS;     % blocking probability in %
    o= OCCUPATION/Clock;         % average occupation in Mbps
end
```

Appendix B – *M*/*M*/*m*/*m* queuing system

Consider an *M*/*M*/*m*/*m* queuing system with a capacity *N* and an offered load of $\rho = \lambda/\mu$ Erlangs.

Blocking probability: the probability of a request being blocked is given by the ErlangB formula:

$$E(\rho, N) = \frac{\frac{\rho^N}{N!}}{\sum_{n=0}^{N} \frac{\rho^n}{n!}}$$

The straightforward MATAB implementation is:

```
numerator= ro^N/factorial(N);
denominator= 0;
for n= 0:N
    denominator= denominator + ro^n/factorial(n);
end
p= numerator/denominator
```

This implementation has two problems. First, for large values of *N*, it causes overflow (for example, `factorial(200)` results in infinite). Second, it is inefficient because it requires a large number of elementary mathematical operations. An efficient way to compute the ErlangB formula is as follows. If we divide both terms of the division by its numerator, we get:

$$E(\rho, N) = \frac{\frac{\rho^N}{N!}}{\sum_{n=0}^{N} \frac{\rho^n}{n!}} = \frac{1}{\sum_{n=0}^{N} \left(\frac{N!}{\rho^N} \times \frac{\rho^n}{n!}\right)}$$

$$E(\rho, N) = \frac{1}{\frac{N \times (N-1) \times \ldots \times 2 \times 1}{\rho^N} + \frac{N \times (N-1) \times \ldots \times 2}{\rho^{N-1}} + \cdots + \frac{N \times (N-1)}{\rho^2} + \frac{N}{\rho} + 1}$$

We define the sequence $a(n)$, with $n = N+1, N, N-1, \ldots, 2, 1$, in the following way:

$a(N+1) = 1$

$a(n) = a(n+1) \times n / \rho$, for $n = N, N-1, \ldots, 2, 1$

Then, if we sum all $a(n)$ values and inverse the result, we obtain the ErlangB $E(\rho, N)$ value. In MATLAB, this method can be implemented as:

```
a= 1;
p= 1;
for n= N:-1:1
    a= a*n/ro;
    p= p + a;
end
p= 1/p
```

Average System Occupation: the average system occupation is given by:

$$L(\rho, N) = \frac{\sum_{i=1}^{N} \frac{\rho^i}{(i-1)!}}{\sum_{n=0}^{N} \frac{\rho^n}{n!}}$$

The straightforward MATLAB implementation is:

```
numerator= 0;
for i=1:N
    numerator= numerator + ro^i/factorial(i-1);
end
denominator= 0;
for n=0:N
    denominator= denominator + ro^n/factorial(n);
end
o= numerator/denominator
```

This implementation has the same problems as previously described for the ErlangB formula. To compute the average system occupation in an efficient way, we reformulate the expression as:

$$L(\rho, N) = \frac{\sum_{i=1}^{N} \frac{\rho^i}{(i-1)!}}{\sum_{n=0}^{N} \frac{\rho^n}{n!}} = \frac{\frac{\rho^N}{N!} \times \sum_{i=1}^{N} \left(\frac{N!}{\rho^N} \times \frac{\rho^i}{(i-1)!} \right)}{\sum_{n=0}^{N} \frac{\rho^n}{n!}} = E(\rho, N) \times \sum_{i=1}^{N} \left(\frac{N!}{\rho^N} \times \frac{\rho^i}{(i-1)!} \right)$$

$$L(\rho, N) = \frac{\frac{N \times (N-1) \times \ldots \times 2 \times 1}{\rho^{N-1}} + \frac{N \times (N-1) \times \ldots \times 2}{\rho^{N-2}} + \cdots + \frac{N \times (N-1)}{\rho} + N}{\frac{N \times (N-1) \times \ldots \times 2 \times 1}{\rho^N} + \frac{N \times (N-1) \times \ldots \times 2}{\rho^{N-1}} + \cdots + \frac{N \times (N-1)}{\rho^2} + \frac{N}{\rho} + 1}$$

If we define the sequence $a(n)$, with $n = N, N, N - 1, \ldots, 2, 1$, in the following way:

$a(N) = N$

$a(n) = a(n+1) \times n / \rho$, for $n = N - 1, \ldots, 2, 1$

and we sum all $a(n)$ values, we obtain the numerator of $L(\rho, N)$. The denominator is obtained in the same way as for the ErlangB formula $E(\rho, N)$. In MATLAB, this method can be implemented as:

```
a= N;
numerator= a;
for i= N-1:-1:1
    a= a*i/ro;
    numerator= numerator + a;
end
a= 1;
denominator= a;
for i= N:-1:1
    a= a*i/ro;
    denominator= denominator + a;
end
o= numerator/denominator
```

Appendix C – Specification of Simulator 2

Develop a MATLAB function, with the name simulator2, implementing an event driven simulator for the service architecture based on one server farm with *n* servers where each server has an interface capacity of *S* Mbps. Consider that the server farm provides movies on 2 possible video formats: HD with a throughput of 5 Mbps and 4K with a throughput of 25 Mbps. As starting point, use the MATLAB function of simulator1 (see Appendix A). The input parameters of simulator2 must be:

λ – movies request rate (in requests/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 – number of movie requests to stop simulation
fname – file name with the duration (in minutes) of the items

The performance parameters estimated by simulator2 must be:

b_{HD} – blocking probability of HD movie requests
b_{4K} – blocking probability of 4K movie requests

The stopping criteria must be the time instant of the arrival of the movie request number *R*.

In the simulator development, consider (as a suggestion) the following events (include the information of server *i* as a third column in the Event List):

ARRIVAL - time instant of a request
DEPARTURE_HD(*i*) - time instant of a HD movie termination on server *i* (*i* = 1,…,*n*)
DEPARTURE_4K(*i*) - time instant of a 4K movie termination on server *i* (*i* = 1,…,*n*)

Consider the following state variables:

STATE(*i*) - total throughput of the movies in transmission by server *i* (*i* = 1,…,*n*)
STATE_HD - total throughput of HD movies in transmission

Consider the following statistical counters:

NARRIVALS - total number of movie requests up to current time instant
REQUESTS_HD - number of HD movie requests up to current time instant
REQUESTS_4K - number of 4K movie requests up to current time instant
BLOCKED_HD - number of blocked HD movie requests up to current time instant
BLOCKED_4K - number of blocked 4K movie requests up to current time instant

Appendix D – List of pairs of connected ASs

```
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];
```

Appendix E – Solving the server farm location problem using ILP (Integer Linear Programming)

We have a set of Autonomous Systems (ASs) and we aim to select a subset of ASs to connect one server farm on each selected AS. The solution must guarantee that in the network of ASs, there is a path between each Tier-2 and Tier-3 AS and its closest server farm with no more than one intermediate AS. Consider the following notation:

$n_1, n_2, \ldots$ – IDs of Tier-2 and Tier-3 ASs where server farms can be connected to;
c_i – cost of Internet connection to AS i, with $i = n_1, n_2, \ldots$;
$I(j)$ – set of Tier-2 and Tier-3 AS IDs such that there is a shortest path from AS j to each AS $i \in I(j)$ with at most one intermediate AS.

In the first step, the sets of nodes $I(j)$, one for each node j, must be computed. Consider the matrix G defining the AS pairs (i, j) with a direct connection, as provided in the previous Appendix D. To compute each set $I(j)$ for each node j, you need to compute the shortest path from node j to each of the other nodes and check how many intermediate nodes are in the shortest path. Then, the nodes whose shortest path has either 0 or 1 intermediate node are included in the set $I(j)$. To know how you can compute shortest paths in MATLAB from matrix G (provided in the previous Appendix D), please check https://www.mathworks.com/help/matlab/ref/graph.shortestpath.html.

Then, you need to solve the optimization problem defined as an Integer Linear Programming (ILP) model. Consider the following variables:

x_i – binary variable, with $i = n_1, n_2, \ldots$, that when is equal to 1 means that AS i must have a connected server farm.

The ILP model defining the optimization problem is as follows:

Minimize $\sum_{i=n_1,n_2,\ldots}(c_i x_i)$ (1)

Subject to:

$\sum_{i \in I(j)} x_i \geq 1 \quad , j = n_1, n_2, \ldots$ (2)

$x_i \in \{0,1\} \quad , i = n_1, n_2, \ldots$ (3)

The objective function (1) is the minimization of the total Internet connection costs of the ASs with connected server farms. Constraints (2) guarantee that each AS j has at least one server farm connected to one AS $i \in I(j)$, i.e., one server farm whose shortest path has at most one intermediate AS. Constraints (3) define all variables as binary ones.

Any set of values assigned to variables x_i compliant with constraints (2–3) defines a feasible solution. A set of assigned values that provides the minimum value for the objective function (1) is an optimal solution: the variables x_i set to 1 define the selected ASs to be connected by server farms and the value of the objective function (1) is the minimum possible cost of the Internet connections.

To solve the ILP model, develop a MATLAB script to write an ASCII file with the ILP model and following the LP format. The LP format is illustrated in the following example:

```
Minimize
 + 2 u1_1 + 3 u1_2 - 2.0 u1_3 + 5.2 u1_4 + 4.3 u2_1 + u2_2 - u2_3 + 2.5 u2_4
Subject To
```

```
\ restricao 1
 + u1_1 + 3 u1_2 - u2_1 - u2_2 = 0
\ restricao 2
 - 4.3 u1_1 + 3.5 u2_1 + ff1 - ff2 >= 3.54
\ restricao 3
 + 5 u1_4 - 3 u2_4 + 4.54 ff1 <= 0
Binary
 ff1
 ff2
General
 u1_1
 u1_2
 u1_3
 u1_4
End
```

- Variables can be named anything provided that the name does not exceed 255 characters, all of which must be alphanumeric (a-z, A-Z, 0-9) or one of the symbols ! " # $ % & ( ) , . ; ? @ _ ‘ ’ { } ~. A variable name cannot begin with a number or a period.
- In the line after `Minimize` (or `Maximize`), specify the objective function.
- In the lines after `Subject To`, specify the problem constraints.
- Constraints must be in canonical format, i.e., the variables before ‘=’ , ‘<=’ or ‘>=’ and the constant afterwards.
- Anything that follows a backslash (\) in a line is a comment and is ignored until a return is encountered (blank lines are also ignored).
- In the lines after `Binary`, list all binary variables and in the lines after `General`, list all integer (non-binary) variables. All variables not listed in these fields are assumed to be real variables.

With the ASCII file defining the optimization problem in LP format, you can use any standard solver to solve the problem. There are some public sites that enable to solve ILP problems in the cloud. As a suggestion, use Gurobi, as provided in https://neos-server.org/neos/solvers/index.html.