Departamento de Eletrónica, Telecomunicações
e Informática

Mestrado Integrado em Eng. de Computadores e Telemática

Ano 2017/2018

Desempenho e Dimensionamento de Redes

Assignment Guide No. 3

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. At each point in time, the server has a catalogue of video items, 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 the targeted types of subscribers and/or the revenue strategy of the company.

In alternative, the service is provided by a server farm, located on a single Datacentre, 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 Datacentres, as illustrated in Figure 3. The advantages of using multiple server farms are (i) to scale the services to even larger number of subscribers, (ii) to lower 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 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.

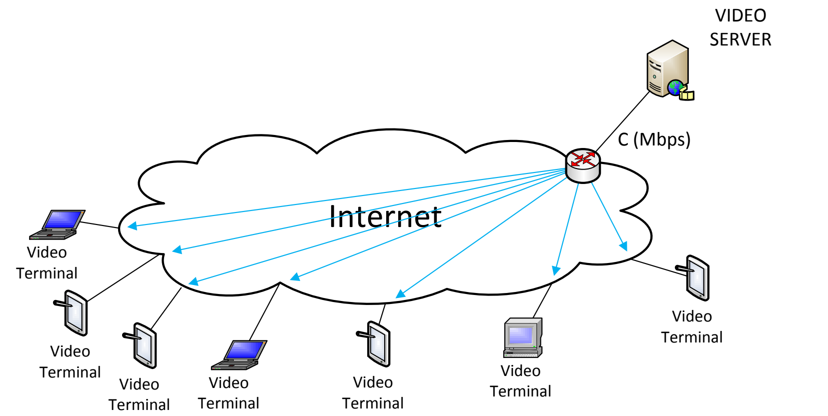


Figure 1

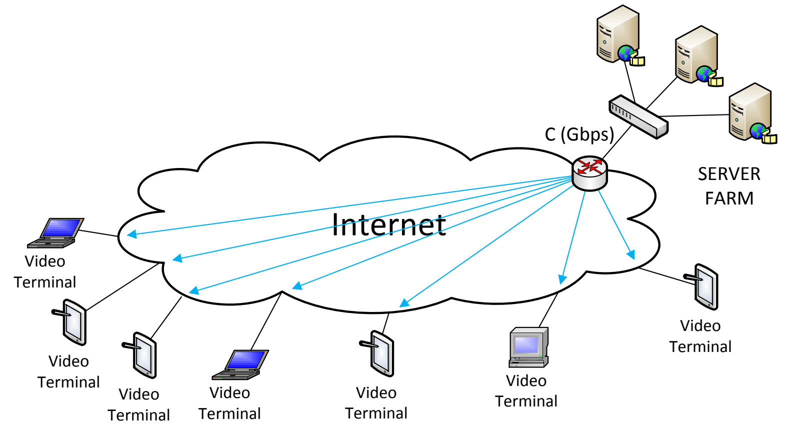


Figure 2

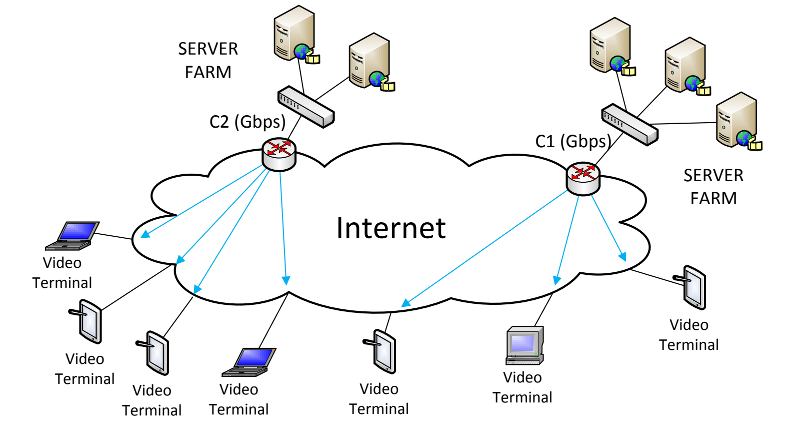


Figure 3

2. First Part of the Assignment

In the first part of the assignment, 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 movies on a single video format and each movie has a throughput of M (in Mbps). The server has a catalogue of 2814 items (between movies and series episodes) with an average movie duration of 86.3 minutes. 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 movie requests are a Poisson process with an average rate of λ (in requests/hour).

Appendix B provides a MATLAB function named simulator1, implementing 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: LOAD (the integral of connection occupation up to the current time instant), NARRIVALS (number of movie requests) and BLOCKED (number of blocked requests).

a) Assume that the movies duration 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 A) for the cases defined in *Table 1*. Analyse these results and take conclusions on the impact of the different input parameters on the two performance parameters.

b) Develop a MATLAB script to run simulator1 10 times with a stopping criterion of $R = 50000$ and to compute the estimated values and the 90% confidence intervals of both performance parameters (confidence intervals in the form $a \pm b$). For all cases defined in *Table 1*, determine by simulation the estimated values and the 90% confidence intervals of both performance parameters. Compare the confidence intervals with the analytical values obtained in **a)**. Is the $M/M/m/m$ queuing system a good approximation of the simulated system? Why?

Table 1

Case	λ (requests/hour)	C (Mbps)	M (Mbps)	Blocking Probability (%)	Average Connection Occupation (Mbps)
A	10	100	4		
B	20	100	4		
C	30	100	4		
D	40	100	4		
E	10	100	10		
F	20	100	10		
G	30	100	10		
H	40	100	10		
I	100	1000	4		
J	200	1000	4		
K	300	1000	4		
L	400	1000	4		
M	100	1000	10		
N	200	1000	10		
O	300	1000	10		
P	400	1000	10		

3. Second Part of the Assignment

In this second part of the assignment, consider a video-streaming service provided by one server farm, as presented in Figure 2, with S servers where each server has an interface of 100 Mbps (assume that the Internet connection of the server farm is $C = S \times 100$ Mbps). Consider also that the server farm provides movies on 2 possible video formats: HD format whose throughput is 4 Mbps and 4K format whose throughput is 10 Mbps. Consider the same catalogue of items as in the first part of the assignment. All catalogue items are available in both formats in all servers.

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 10 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 4 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% of the requests are for movies of 4K format.

Develop a MATLAB function named simulator2, implementing an event driven simulator for this case to estimate the blocking probability of movies of each format (see Appendix C). Then, develop a MATLAB script to run simulator2 40 times with a stopping criterion of R = 50000 and to compute the estimated values and the 90% confidence intervals of the two blocking probability parameters.

a) For all cases defined in *Table 2*, determine by simulation the estimated values and the 90% confidence intervals of the two blocking probability parameters. Analyse these results and take conclusions on the impact of the different input parameters on the two blocking probabilities.

Table 2

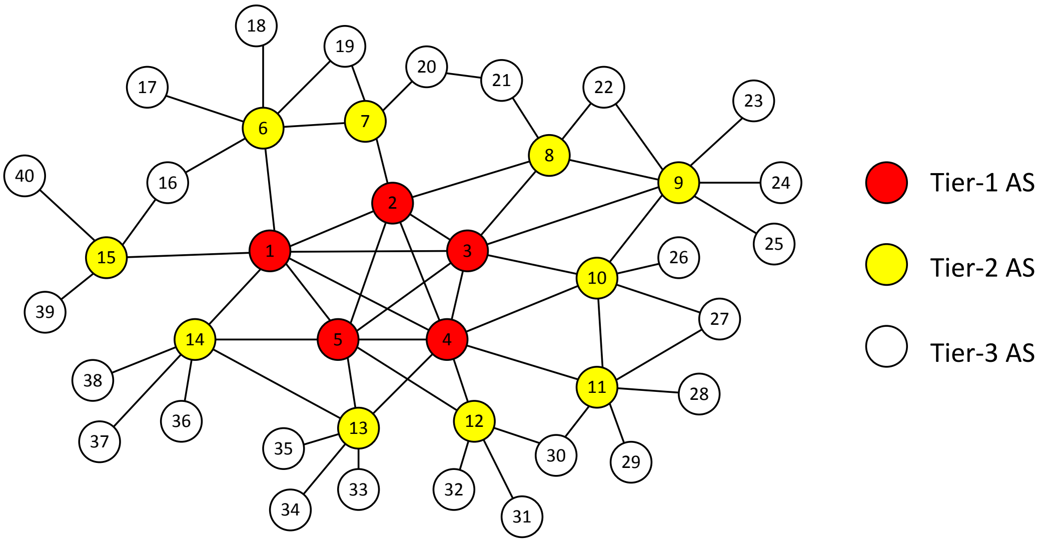
Case	λ (requests/hour)	S (No. servers)	W (Mbps)	p (%)	HD Blocking Probability (%)	4K Blocking Probability (%)
A	10	1	0	20%		
B	10	1	30	20%		
C	10	1	60	20%		
D	10	1	0	40%		
E	10	1	30	40%		
F	10	1	60	40%		
G	30	3	0	20%		
H	30	3	120	20%		
I	30	3	180	20%		
J	30	3	0	40%		
K	30	3	120	40%		
L	30	3	180	40%		

b) Consider that the video-service company aims to reach 20000 subscribers and 30% of 4K requests (it is expected that each subscriber requests, on average, 1 movie per week). The company aims to have a robust solution such that the worst blocking probability between the two formats must be not higher than 0.1% when all servers are working and 1% when one server fails. Determine by simulation how many servers are required. Determine also an adequate reservation value W to be set in the front-office.

4. Third Part of the Assignment

In the third part of the assignment, consider the service provided by multiple server farms, as presented in Figure 3, where, again, movies are available in the 2 previous possible video formats (HD and 4K). Consider the same catalogue of items as previous. All catalogue items are available in both formats in all servers. In all farms, each server has an interface of 100 Mbps.

Consider that the Internet part that covers the company targeted subscribers is given by the following figure, which specifies the different types of Autonomous Systems (ASs) and how they are connected (the list of pairs of connected ASs is provided in MATLAB format in Appendix D):



Besides determining the total number of servers, the company also needs (i) to identify the ASs where the different server farms must be connected and (ii) to decide how many servers must be placed on each farm. Only Tier-2 of Tier-3 ASs provide the Internet access service.

The average OPEX costs (Operational Costs) of a server farm is 10 when it is connected to a Tier-2 AS and 8 when it is connected to a Tier-3 AS. The company assumes that it can reach 3000 subscribers on each Tier-2 AS and 1500 subscribers on each Tier-3 AS with average requests of 2 movies per week per customer in the prime time and 30% of requests for movies of 4K format.

a) Select a minimum cost set of ASs to install the server farms (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 OPEX cost of the solution? How many server farms are required and in which ASs they must be connected to?

b) Use simulator2 to determine the total number of required servers and the appropriate reservation *W* to provide a blocking probability around 1% for movie requests of both formats.

c) To define the final solution, split the total number of servers determined in **b)** by the locations determined in **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 location?

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

Blocking probability:

Consider an *M*/*M*/*m*/*m* queuing system with a capacity N and an offered load of ρ Erlangs. The ErlangB formula $E(\rho, N)$:

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

gives the probability of a new request being blocked. In MATLAB, the straightforward 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 larger values of *N* and *ρ*, it causes overflow. Second, it is inefficient because it requires a large number of elementary mathematical operations. An efficient way to compute ErlangB formula is as follows. If we divide both terms of the division by its numerator, we reformulate ErlangB formula in the following way:

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

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

Now, if we define the sequence *a*(*n*), with *n* = *N*+1, *N*, *N* – 1, …, 2, 1, in the following way:

$a(N+1) = 1$

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

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

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

Average System Occupation:

On the other hand, the average system occupation is given by:

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

In MATLAB, the straightforward implementation of the average system occupation 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 implement 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 with the following code:

```
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 B – Proposed MATLAB function for Simulator 1

```matlab
function [b o]= simulator1(lambda,C,M,R)
    %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

    invlambda=60/lambda; %average time between requests (in minutes)
    invmiu= load('movies.txt');  %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:
    LOAD= 0;
    NARRIVALS= 0;
    BLOCKED= 0;
    %Simulation Clock and initial List of Events:
    Clock= 0;
    EventList= [ARRIVAL exprnd(invlambda)];

    while NARRIVALS < R
        event= EventList(1,1);
        Previous_Clock= Clock;
        Clock= EventList(1,2);
        EventList(1,:)= [];
        LOAD= LOAD + STATE*(Clock-Previous_Clock);
        if event == ARRIVAL
            EventList= [EventList; ARRIVAL Clock+exprnd(invlambda)];
            NARRIVALS= NARRIVALS+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/NARRIVALS; % blocking probability in %
    o= LOAD/Clock;            % average connection occupation in Mbps
end
```

Appendix C – Specification of Simulator 2

Develop a MATLAB function named simulator2, implementing an event driven simulator for the service architecture based on one server farm with S servers, providing movies in 2 video formats and with a resource reservation of W for 4K movies. As starting point, use the simulator1 MATLAB function proposed in Appendix B. The input parameters of simulator2 must be:

λ – movies request rate (in requests/hour)
p – percentage of requests for 4K movies (in %)
S – number of servers (each server with a capacity of 100 Mbps)
W – resource reservation for high-definition movies (in Mbps)
M_{HD} – throughput of movies in HD format (4 Mbps)
M_{4K} – throughput of movies in 4K format (10 Mbps)
R – number of movie requests to stop simulation

The performance parameters estimated by simulator2 must be:

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

The stopping criteria must be the time instant of the arrival of the movie request number R. In the simulator development, consider the following events:

ARRIVAL_HD - time instant of a HD movie request
ARRIVAL_4K - time instant of a 4K movie request
DEPARTURE_HD(i) - time instant of a HD movie termination on server i (i = 1,…,S)
DEPARTURE_4K(i) - time instant of a 4K movie termination on server i (i = 1,…,S)

Consider the following state variables:

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

Consider the following statistical counters:

NARRIVALS - total number of movie requests
NARRIVALS_HD - number of HD movie requests
NARRIVALS_4K - number of 4K movie requests
BLOCKED_HD - number of blocked HD movie requests
BLOCKED_4K - number of blocked 4K movie requests

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 at least one server farm with no more than one intermediate AS. Consider the following notation:

n – number of Tier-2 and Tier-3 ASs where server farms can be connected to;
c_i – OPEX cost of connecting a server farm to AS i, with $1 \leq i \leq n$;
$I(j)$ – set of Tier-2 and Tier-3 ASs such that there is a shortest path between AS j and each AS $i \in I(j)$ with at most one intermediate AS.

Consider the list *G* of AS pairs (i, j), as provided in the previous Appendix D. To compute each set $I(j)$, use the following labelling algorithm:

- Start by assigning label 0 to AS *j* and label –1 to all other ASs.
- For a = 0:1 do:
    - for each AS pair $(i, j) \in G$, if one AS has label a and the other AS has label –1, assign label a+1 to the AS that has label –1
- At the end: (i) the shortest path between AS *j* and each AS with label 1 has no intermediate ASs and (ii) the shortest path between AS *j* and each AS with label 2 has one intermediate AS.
- The set $I(j)$ is composed by all Tier-2 and Tier-3 ASs whose label at the end of the algorithm is not negative.

Consider the following variables:

x_i – binary variable, with $1 \leq i \leq n$, that when is equal to 1 means that AS *i* must be connected to one server farm;
y_{ji} – binary variable, with $1 \leq j \leq n$ and $i \in I(j)$, that when is equal to 1 means that AS j is associated with AS i.

The Integer Linear Programming (ILP) model defining the optimization problem is defined as:

Minimize $\sum_{i=1}^{n} c_i x_i$ (1)

Subject to:

$\sum_{i \in I(j)} y_{ji} = 1 \quad , j = 1 \dots n$ (2)

$y_{ji} \leq x_i \quad , j = 1 \dots n, i \in I(j)$ (3)

$x_i \in \{0,1\} \quad , i = 1 \dots n$ (4)

$y_{ji} \in \{0,1\} \quad , j = 1 \dots n, i \in I(j)$ (5)

The objective function (1) is the minimization of the OPEX costs of the selected server farms. Constraints (2) guarantee that each AS j is associated with one AS $i \in I(j)$ while constraints (3) guarantee that an associated AS $i \in I(j)$ must have one server farm connected (when y_{ji} is 1, x_i must be also equal to 1 in constraints (3)). Therefore, constraints (2–3) guarantee that each AS *j* has always one server farm whose shortest path has at most one intermediate AS. Constraints (4–5) are the constraints that define all variables as binary ones.

Any set of values assigned to variables x_i and y_{ji} compliant with constraints (2–5) 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 (1) is the minimum possible OPEX cost.

To solve the ILP model, develop a code to write an ASCII file using 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 (\) 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.