**DEPARTAMENTO DE ELETRÓNICA, TELECOMUNICAÇÕES
E INFORMÁTICA**

**MESTRADO EM ENGENHARIA DE COMPUTADORES E TELEMÁTICA**

ANO 2022/2023

# MODELAÇÃO E DESEMPENHO DE REDES E SERVIÇOS

PRACTICAL GUIDE

Task 1

Consider a multiple choice test such that each question has n multiple answers and only one is correct. Assume that the student has studied a percentage p (with $0\% \leq p \leq 100\%$) of the test content. When a question addresses the content the student has studied, he selects the right answer with 100% of probability. Otherwise, the student always selects randomly one of the n answers with a uniform distribution.

1.a. When $p = 60\%$ and $n = 4$, determine the probability of the student to select the right answer. <u>Answer: 70%</u>

1.b. When $p = 70\%$ and $n = 5$, determine the probability of the student to known the answer when he selects the right answer. <u>Answer: 92.1%</u>

1.c. Draw a plot with the same look as the plot below with the probability of the student to select the right answer as a function of the probability p (consider the number of multiple answers $n = 3, 4$ and 5). What do you conclude from these results? <u>Answer:</u>
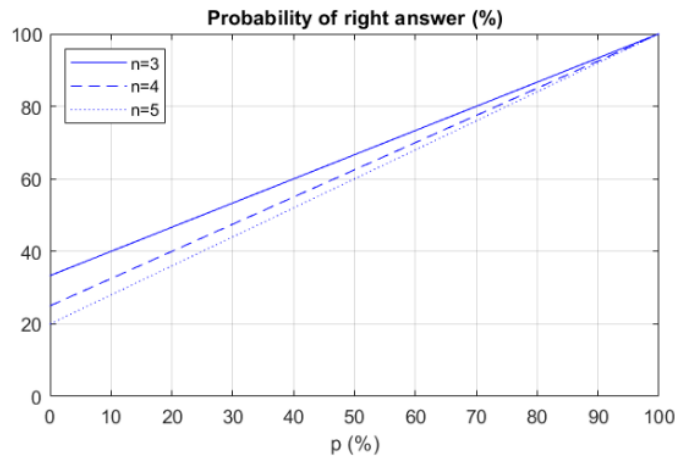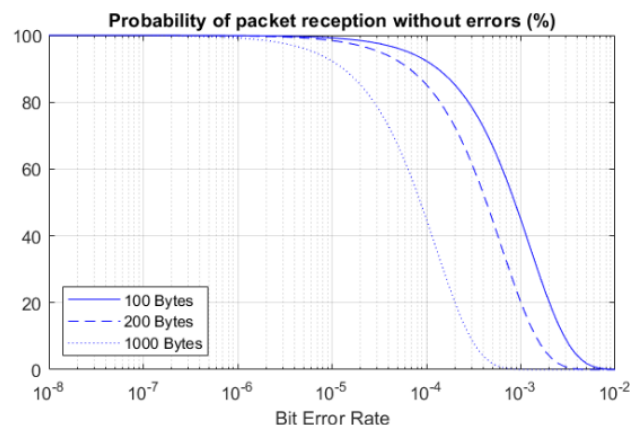

1.d. Draw a plot with the same look as the plot below with the probability of the student to know the answer when he selects the right answer as a function of the probability p (consider $n = 3, 4$ and 5). What do you conclude from these results? <u>Answer:</u>
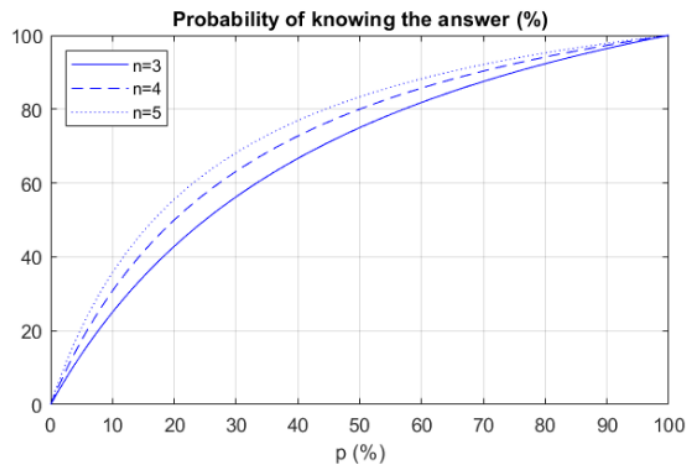
Task 2

Consider a wireless link between multiple stations for data communications with a bit error rate (*ber*) of p. Assume that transmission errors in the different bits of a data frame are statistically independent (i.e., the number of errors of a data packet is a binomial random variable).

2.a. Determine the probability of a data frame of 100 Bytes to be received without errors when $p = 10^{-2}$. <u>Answer: 0.0322%</u>

2.b. Determine the probability of a data frame of 1000 Bytes to be received with exactly one error when $p = 10^{-3}$. <u>Answer: 0.2676%</u>

2.c. Determine the probability of a data frame of 200 Bytes to be received with one or more errors when $p = 10^{-4}$. <u>Answer: 14.7863%</u>

2.d. Draw a plot using a logarithmic scale for the X-axis (use the MATLAB function `semilogx`) with the same look as the plot below with the probability of a data frame (of size 100 Bytes, 200 Bytes or 1000 Bytes) being received without errors as a function of the *ber* (from $p = 10^{-8}$ up to $p = 10^{-2}$). What do you conclude from these results? <u>Answer:</u>


2.e. Draw a plot using a logarithmic scale for the Y-axis (use the MATLAB function `semilogy`) with the same look as the plot below with the probability of a data frame being received without errors (for $p = 10^{-4}$, 10^{-3} and 10^{-2}) as a function of the packet size (all integer values from 64 Bytes up to 1518 Bytes). What do you conclude from these results? <u>Answer:</u>
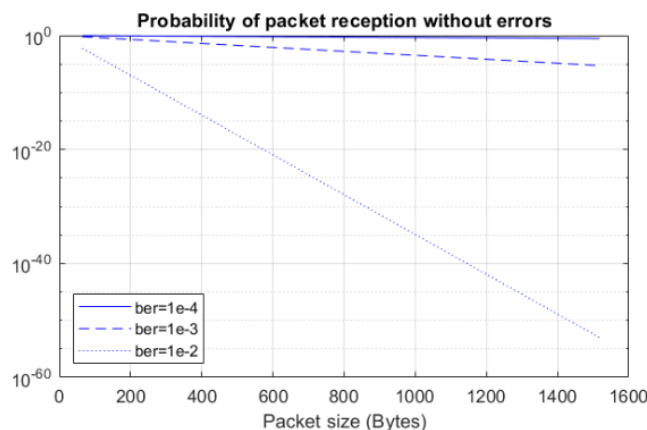
Task 3

Consider a wireless link between multiple stations for data communications. The bit error rate (*ber*) introduced by the wireless link (due to the variation of the propagation and interference factors along with time) is approximately given by the following Markov chain:
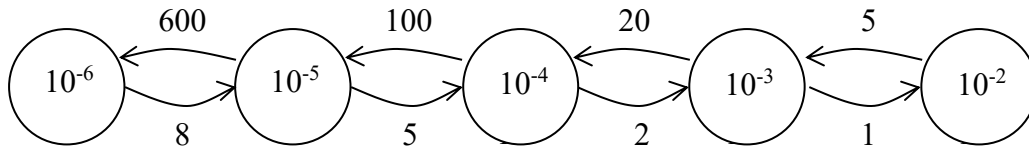
where the state transition rates are in number of transitions per hour. Consider that the link is in an interference state when its *ber* is at least 10^{-3} and in a normal state, otherwise. Assume that all stations detect with a probability of 100% when the data frames sent by the other stations are received with errors. Determine:

3.a. the probability of the link being in each of the five states; <u>answer:</u>
9.86×10^{-1} (10^{-6}), 1.31×10^{-2} (10^{-5}), 6.57×10^{-4} (10^{-4}), 6.57×10^{-5} (10^{-3}), 1.31×10^{-5} (10^{-2})

3.b. the average percentage of time the link is in each of the five states; <u>answer:</u>
9.86×10^{-1} (10^{-6}), 1.31×10^{-2} (10^{-5}), 6.57×10^{-4} (10^{-4}), 6.57×10^{-5} (10^{-3}), 1.31×10^{-5} (10^{-2})

3.c. the average *ber* of the link; <u>answer: 1.38×10^{-6}</u>

3.d. the average time duration (in minutes) that the link stays in each of the five states;
<u>answer: 7.5 (10^{-6}), 0.10 (10^{-5}), 0.59 (10^{-4}), 2.86 (10^{-3}), 12.0 (10^{-2})</u>

3.e. the probability of the link being in the normal state and in interference state; <u>answer:</u>
<u>0.999921 (normal), 7.89×10^{-5} (interference)</u>

3.f. the average *ber* of the link when it is in the normal state and when it is in the interference state; <u>answer: 1.18×10^{-6} (normal), 2.50×10^{-3} (interference)</u>

3.g. considering a data frame of size *B* (in Bytes) sent by one source station to a destination station, draw a plot with the same look as the plot below of the probability of the packet being received by the destination station <u>with at least one error</u> as a function of the packet size (from 64 Bytes up to 1500 Bytes); analyze and justify the results; <u>answer:</u>
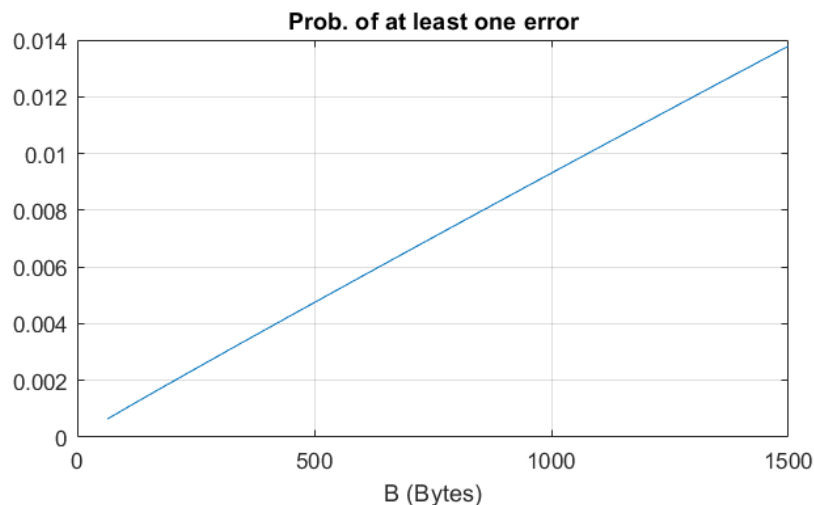
3.h. considering that a data frame of size B (in Bytes) sent by one source station is received <u>with at least one error</u> by the destination station, draw a plot with the same look as the plot below of the probability of the link being <u>in the normal state</u> as a function of the packet size (from 64 Bytes up to 1500 Bytes); analyze and justify the results; <u>answer:</u>

3.i. considering that a data frame of size B (in Bytes) sent by one source station is received <u>without errors</u> by the destination station, draw a plot with the same look as the plot below (use the MATLAB function `semilogy`) of the probability of the link being <u>in the interference state</u> as a function of the packet size (from 64 Bytes up to 1500 Bytes); analyze and justify the results; <u>answer:</u>
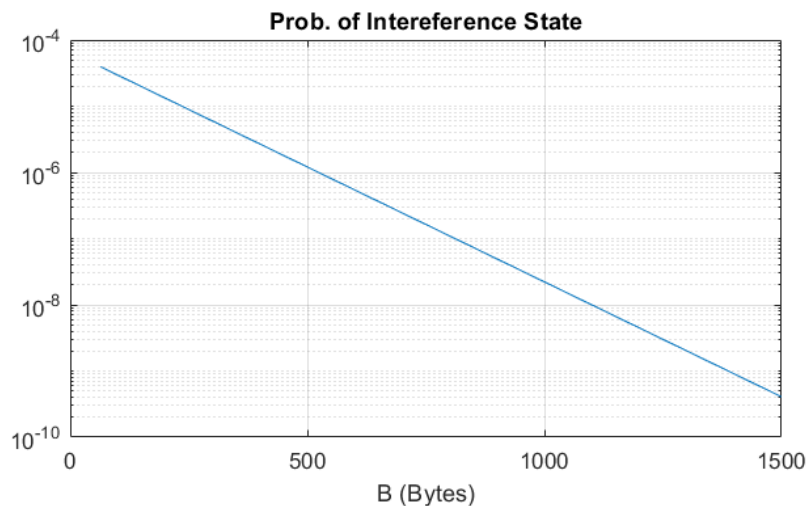
Task 4

Consider an ideal link (i.e., with a *ber* = 0) from one router to another router with a capacity of C Mbps (1 Mbps = 10^6 bps) for IP communications. The link has a propagation delay of 10 μs (1 μs = 10^{-6} seconds). There is a very large queue at the output port of the link. The IP packet flow supported by the link is characterized by:

(i) the packet arrivals are a Poisson process with rate λ pps (packets per second)
(ii) the size of each IP packet is between 64 and 1518 bytes (the size includes the overhead of the Layer 2 protocol) with the probabilities: 19% for 64 bytes, 23% for 110 bytes, 17% for 1518 bytes and an equal probability for all other values (i.e., from 65 to 109 and from 111 to 1517).

Consider that λ = 1000 pps and C = 10 Mbps. Determine:

4.a. the average packet size (in Bytes) and the average packet transmission time of the IP flow; answer: 620.02 Bytes, 4.96×10^{-4} seconds

4.b. the average throughput (in Mbps) of the IP flow; answer: 4.96 Mbps

4.c. the capacity of the link, in pps; answer: 2016.06 pps

4.d. the average packet queuing delay and packet system delay of the IP flow (the system delay is the queuing delay + transmission time + propagation delays); answer: queuing – 4.60×10^{-4} seconds, system – 9.66×10^{-4} seconds

4.e. for C = 10 Mbps, draw a plot with the same look as the plot below with the average system delay as a function of the packet arrival rate λ (from λ = 100 pps up to λ = 2000 pps); analyze and justify the results;

4.f. for C = 10, 20 and 100 Mbps, draw a plot with the same look as the plot below with the average system delay as a function of the packet arrival rate λ (from λ = 100 pps up to λ = 2000 pps when C = 10, from λ = 200 pps up to λ = 4000 pps when C = 20 and from λ = 1000 pps up to λ = 20000 pps when C = 100); the x-axis should indicate the value of λ as a percentage of the capacity of the link, in pps (determined in **4.c.**); analyze and justify the results.
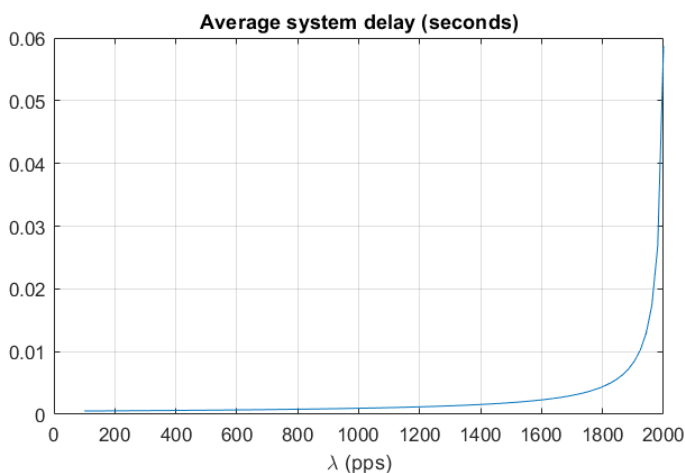
Answer to **4.e.**
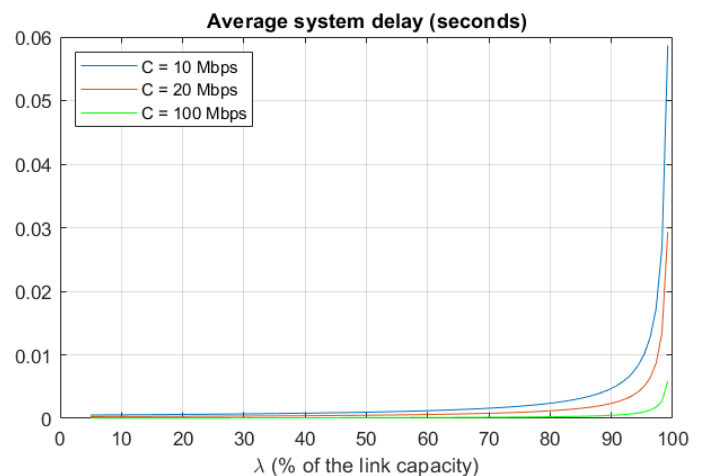
Answer to **4.f.**

Task 5

Consider the event driven simulator, implemented in the provided MATLAB function *Simulator1*, to estimate the performance of a point-to-point IP link between a company router and its ISP (Internet Service Provider). The simulator only considers the downstream direction, *i.e.*, from ISP to the company (usually, the direction with highest traffic load).

Simulator1 considers a link of *C* (in Mbps) and a queue of size *f* (in Bytes) with a FIFO (First-In-First-Out) queueing discipline. The packet flow submitted to the link is characterized by: (i) an exponentially distributed time between packet arrivals with average $1/\lambda$ and (ii) a random packet size between 64 and 1518 bytes with the probabilities: 19% for 64 bytes, 23% for 110 bytes, 17% for 1518 bytes and an equal probability for all other values (i.e., from 65 to 109 and from 111 to 1517).

Input parameters of *Simulator1*:
  λ – packet rate, in packets per second (pps)
  C   – link capacity, in Mbps
  f   – queue size, in Bytes
  P   – total number of transmitted packets of a simulation run

Performance parameters estimated by *Simulator1*:
  PL   – Packet Loss (%)
  APD   – Average Packet Delay (milliseconds)
  MPD   – Maximum Packet Delay (milliseconds)
  TT   – Transmitted Throughput (Mbps)

Stopping criterion of *Simulator1*:

  ➤ Time instant when the link ends the transmission of the Pth packet (*P* is one the of input parameter); in *Simulator1*, the queued packets at the end of the simulation do not count for the performance estimation.

Simulator1 is based on the following variables:

Events: ARRIVAL (the arrival of a packet) and DEPARTURE (the transmission end of a packet).

State variables: STATE (binary variable indicating if the link is free or busy with the transmission of a packet), QUEUEOCCUPATION (occupation of the queue, in number of bytes, with the queued packets) and QUEUE (matrix with a variable number of rows and 2 columns where each column has the size and the arriving time instant of each packet in the queue).

Statistical counters: TOTALPACKETS (number of packets arrived to the system), LOSTPACKETS (number of packets dropped due to buffer overflow), TRANSMITTEDPACKETS (number of transmitted packets), TRANSMITTEDBYTES (sum of the bytes of the transmitted packets), DELAYS (sum of the delays of the transmitted packets), MAXDELAY (maximum delay among all transmitted packets).

Based on the statistical counters, the performance parameters are estimated at the end as:

PL = 100 × LOSTPACKETS / TOTALPACKETS
APD = 1000 × DELAYS / TRANSMITTEDPACKETS
MPD = 1000 × MAXDELAY
TT = 10^{-6} × TRANSMITTEDBYTES × 8 / total simulated time

5.a. Develop a MATLAB script to run *Simulator1* 10 times with a stopping criterion of $P = 10000$ at each run and to compute the estimated values and the 90% confidence intervals of all performance parameters when $\lambda = 1800$ pps, $C = 10$ Mbps and $f = 1.000.000$ Bytes (~1 MByte). <u>Results (recall that these are simulation results)</u>:

```
PacketLoss (%)        = 0.00e+00 +- 0.00e+00
Av. Packet Delay (ms) = 4.67e+00 +- 3.07e-01
Max. Packet Delay (ms) = 2.54e+01 +- 2.90e+00
Throughput (Mbps)     = 8.94e+00 +- 6.68e-02
```

5.b. Repeat the previous experiment but now run *Simulator1* 100 times. Compare these results with the previous ones and take conclusions. <u>Results (recall that these are simulation results)</u>:

```
PacketLoss (%)        = 0.00e+00 +- 0.00e+00
Av. Packet Delay (ms) = 4.32e+00 +- 1.01e-01
Max. Packet Delay (ms) = 2.27e+01 +- 7.71e-01
Throughput (Mbps)     = 8.93e+00 +- 1.82e-02
```

5.c. Repeat the experiment **5.b** but now consider $f = 10.000$ Bytes (~10 KBytes) Justify the differences between these results and the results of experiment **5.b**. <u>Results (recall that these are simulation results)</u>:

```
PacketLoss (%)        = 1.01e+00 +- 3.56e-02
Av. Packet Delay (ms) = 2.95e+00 +- 2.39e-02
Max. Packet Delay (ms) = 8.99e+00 +- 1.90e-02
Throughput (Mbps)     = 8.75e+00 +- 1.60e-02
```

5.d. Repeat the experiment **5.c** but now consider $f = 2.000$ Bytes (~2 KBytes) Justify the differences between these results and the results of experiments **5.b** and **5.c**. <u>Results (recall that these are simulation results)</u>:

```
PacketLoss (%)        = 1.04e+01 +- 5.17e-02
Av. Packet Delay (ms) = 9.54e-01 +- 1.59e-03
Max. Packet Delay (ms) = 2.71e+00 +- 7.85e-03
Throughput (Mbps)     = 7.12e+00 +- 1.02e-02
```

5.e. Consider that the system is modelled by a $M/G/1$ queueing model[1]. Determine the theoretical values of the packet loss, average packet delay and total throughput using the $M/G/1$ model for the parameters considered in experiments **5.a** and **5.b**. Compare these values with the simulation results of experiments **5.a** and **5.b** and take conclusions. <u>Results</u>:

```
Packet Loss (%)       = 0.0000
Av. Packet Delay (ms) = 4.3883
Throughput (Mbps)     = 8.9283
```

---

[1] In Module 1 of the theoretical slides, check the $M/G/1$ queueing model (slide 53) and the resolution of Example 8 (slides 58-60).

Task 6

In the previous Task 5, it was assumed that the link between the company router and its ISP (Internet Service Provider) does not introduce transmission/propagation errors (the typical case of a wired access network). In this task, assume that the link is provided by a wireless access (for example, through a 4G/5G mobile network). In this case, when a packet reaches the company router with at least one error, the packet is discarded (recall the FCS field of IEEE 802 frames).

6.a. Develop *Simulator2* by changing the provided *Simulator1* in order to consider that the link introduces a BER (Bit Error Rate) given by *b*. The input parameters of *Simulator2* should be all the input parameters of *Simulator1* plus parameter *b*. The performance parameters estimated by *Simulator2* should be the same as the ones of *Simulator1*. The stopping criterion of *Simulator2* should be time instant when the link ends the transmission of the Pth packet without errors (i.e., the packets with errors do not count for the stopping criterion)

6.b. Develop a MATLAB script to run *Simulator2* 100 times with a stopping criterion of *P* = 10000 at each run and to compute the estimated values and the 90% confidence intervals of all performance parameters when λ = 1800 pps, *C* = 10 Mbps, *f* = 1.000.000 Bytes (~1 MByte) and *b* = 10^{-6}. Compare these results with the results of **5.b** and take conclusions. <u>Results (recall that these are simulation results):</u>

```
PacketLoss (%)        = 4.91e-01 +- 1.22e-02
Av. Packet Delay (ms)  = 4.34e+00 +- 1.23e-01
Max. Packet Delay (ms) = 2.24e+01 +- 8.72e-01
Throughput (Mbps)      = 8.85e+00 +- 2.03e-02
```

6.c. Repeat the experiment **6.b** but now consider *f* = 10.000 Bytes (~10 KBytes) Justify the differences between these results and the results of experiment **5.c**. <u>Results (recall that these are simulation results):</u>

```
PacketLoss (%)        = 1.49e+00 +- 3.68e-02
Av. Packet Delay (ms)  = 2.95e+00 +- 2.05e-02
Max. Packet Delay (ms) = 8.97e+00 +- 1.76e-02
Throughput (Mbps)      = 8.68e+00 +- 1.45e-02
```

6.d. Repeat the experiment **6.c** but now consider *f* = 2.000 Bytes (~2 KBytes) Justify the differences between these results and the results of experiment **5.d**. <u>Results (recall that these are simulation results):</u>

```
PacketLoss (%)        = 1.08e+01 +- 5.83e-02
Av. Packet Delay (ms)  = 9.50e-01 +- 1.89e-03
Max. Packet Delay (ms) = 2.70e+00 +- 7.42e-03
Throughput (Mbps)      = 7.04e+00 +- 1.10e-02
```

Task 7

In Task 5, it was assumed that the link between the company router and its ISP supports a flow of data packets. In this task, consider that, besides the flow of data packets, the link also supports *n* VoIP (Voice over IP) packet flows. Each VoIP flow generates packets with size uniformly distributed between 110 Bytes and 130 Bytes, and the time between packet arrivals is uniformly distributed between 16 milliseconds and 24 milliseconds[2].

7.a. Develop *Simulator3* by changing the provided *Simulator1* in order to consider the *n* additional VoIP packet flows[3]. Consider that packets of all flows (data and VoIP) are queued on a single queue served with a FIFO discipline. The input parameters of *Simulator3* should be all the input parameters of *Simulator1* plus parameter *n*. The performance parameters estimated by *Simulator3* should be:

> PL_{data} − Packet Loss of data packets (%)
> PL_{VoIP} − Packet Loss of VoIP packets (%)
> APD_{data} − Average Delay of data packets (milliseconds)
> APD_{VoIP} − Average Delay of VoIP packets (milliseconds)
> MPD_{data} − Maximum Delay of data packets (milliseconds)
> MPD_{VoIP} − Maximum Delay of VoIP packets (milliseconds)
> TT − Transmitted Throughput (data + VoIP) (Mbps)

7.b. Develop a MATLAB script to run *Simulator3* 100 times with a stopping criterion of *P* = 10000 at each run and to compute the estimated values and the 90% confidence intervals of all performance parameters when $\lambda = 1800$ pps, *C* = 10 Mbps, *f* = 1.000.000 Bytes (~1 MByte) and *n* = 20. Results (recall that these are simulation results):

```
PacketLoss of data (%)       = 0.00e+00 +- 0.00e+00

PacketLoss of VoIP (%)       = 0.00e+00 +- 0.00e+00

Av. Packet Delay of data (ms) = 2.03e+01 +- 1.94e+00

Av. Packet Delay of VoIP (ms) = 1.99e+01 +- 1.95e+00

Max. Packet Delay of data (ms) = 5.34e+01 +- 3.50e+00

Max. Packet Delay of VoIP (ms) = 5.30e+01 +- 3.49e+00

Throughput (Mbps)            = 9.82e+00 +- 1.77e-02
```

7.c. Repeat the experiment **7.b** but now consider *f* = 10.000 Bytes (~10 KBytes) Justify the differences between these results and the results of experiment **7.b**. Results (recall that these are simulation results):

```
PacketLoss of data (%)       = 2.62e+00 +- 8.52e-02

PacketLoss of VoIP (%)       = 3.63e-01 +- 2.27e-02

Av. Packet Delay of data (ms) = 3.99e+00 +- 3.91e-02

Av. Packet Delay of VoIP (ms) = 3.67e+00 +- 4.03e-02
```

---

[2] These values are based on G.726 standard, at 32 kbps, assuming a packet size of ± 10 Bytes with a uniform distribution around its average size and a jitter of ± 4 milliseconds with a uniform distribution around the average inter arrival time of 20 milliseconds.

[3] In the implementation of *Simulator3*, the time instant of the first packet arrival of each VoIP flow must be randomly generated with a uniform distribution between 0 and 20 milliseconds.

```
Max. Packet Delay of data (ms) = 9.01e+00 +- 1.27e-02
Max. Packet Delay of VoIP (ms) = 8.78e+00 +- 1.87e-02
Throughput (Mbps)              = 9.42e+00 +- 1.60e-02
```

7.d. Repeat the experiment **7.c** but now consider f = 2.000 Bytes (~2 KBytes) Justify the differences between these results and the results of experiment **7.b** and **7.c**. Results (recall that these are simulation results):

```
PacketLoss of data (%)         = 1.26e+01 +- 7.96e-02
PacketLoss of VoIP (%)         = 1.97e+00 +- 4.20e-02
Av. Packet Delay of data (ms)  = 9.88e-01 +- 2.26e-03
Av. Packet Delay of VoIP (ms)  = 7.41e-01 +- 2.60e-03
Max. Packet Delay of data (ms) = 2.71e+00 +- 7.36e-03
Max. Packet Delay of VoIP (ms) = 2.59e+00 +- 1.14e-02
Throughput (Mbps)              = 7.74e+00 +- 1.23e-02
```

7.e. Develop *Simulator4* by changing *Simulator3* so that VoIP packets are given higher priority than data packets in the queue. Develop a MATLAB script to run *Simulator4* 100 times with a stopping criterion of P = 10000 at each run and to compute the estimated values and the 90% confidence intervals of all performance parameters when λ = 1800 pps, C = 10 Mbps, f = 1.000.000 Bytes (~1 MByte) and n = 20. Compare these results with the results of **7.b** and take conclusions. Results (recall that these are simulation results):

```
PacketLoss of data (%)         = 0.00e+00 +- 0.00e+00
PacketLoss of VoIP (%)         = 0.00e+00 +- 0.00e+00
Av. Packet Delay of data (ms)  = 2.35e+01 +- 2.34e+00
Av. Packet Delay of VoIP (ms)  = 5.59e-01 +- 1.39e-03
Max. Packet Delay of data (ms) = 5.95e+01 +- 3.76e+00
Max. Packet Delay of VoIP (ms) = 1.38e+00 +- 4.31e-03
Throughput (Mbps)              = 9.83e+00 +- 2.01e-02
```

7.f. Repeat the experiment **7.e** but now consider f = 10.000 Bytes (~10 KBytes). Compare these results with the results of **7.c** and take conclusions. Results (recall that these are simulation results):

```
PacketLoss of data (%)         = 2.66e+00 +- 6.99e-02
PacketLoss of VoIP (%)         = 3.45e-01 +- 2.47e-02
Av. Packet Delay of data (ms)  = 4.36e+00 +- 3.71e-02
Av. Packet Delay of VoIP (ms)  = 5.34e-01 +- 1.61e-03
Max. Packet Delay of data (ms) = 9.96e+00 +- 2.58e-02
Max. Packet Delay of VoIP (ms) = 1.38e+00 +- 4.83e-03
Throughput (Mbps)              = 9.43e+00 +- 1.52e-02
```

7.g. Repeat the experiment **7.f** but now consider f = 2.000 Bytes (~2 KBytes). Compare these results with the results of **7.d** and take conclusions. Results (recall that these are simulation results):

```
PacketLoss of data (%)       = 1.27e+01 +- 6.47e-02
PacketLoss of VoIP (%)       = 1.63e+00 +- 4.18e-02
Av. Packet Delay of data (ms) = 1.04e+00 +- 2.24e-03
Av. Packet Delay of VoIP (ms) = 4.34e-01 +- 1.14e-03
Max. Packet Delay of data (ms) = 2.94e+00 +- 1.67e-02
Max. Packet Delay of VoIP (ms) = 1.38e+00 +- 5.71e-03
Throughput (Mbps)            = 7.73e+00 +- 1.15e-02
```

Task 8

Consider the MPLS (Multi-Protocol Label Switching) network of an ISP (Internet Service Provider) with a topology defined over a rectangle with 600 Km by 300 Km as follows:
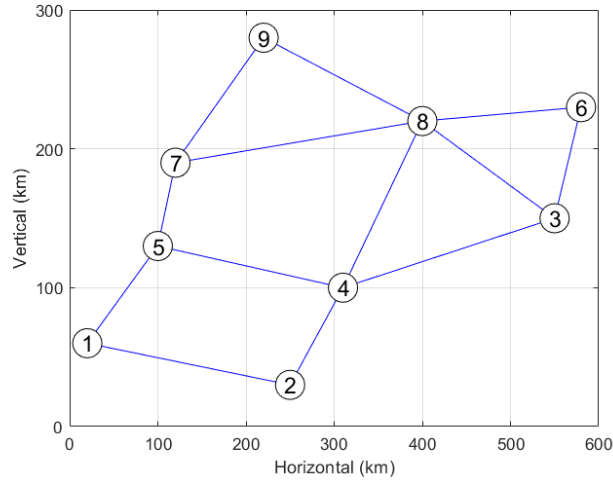
Consider that all links of the network have a capacity of 10 Gbps:

- The coordinates of the nodes are provided by matrix *Nodes* with the number of rows equal to the number of nodes and 2 columns (column 1 with the horizontal coordinate and column 2 with the vertical coordinate).
- The list of links is provided by matrix *Links* with the number of rows equal to the number of links and 2 columns (with the end nodes of each link).
- The capacity of the links is provided by square matrix C, where $C(i,j)$ is either the capacity of arc (i,j) if it exists or is 0 if arc (i,j) does not exist.
- The length of the links is provided by the square matrix L, where $L(i,j)$ is either the length (in Km) of arc (i,j) if it exists or is $+\infty$ if arc (i,j) does not exist.

The network supports a unicast service with the following traffic flows (throughput values b_t and $\underline{b}_t$ in Gbps):

t	o_t	d_t	b_t	$\underline{b}_t$
1	1	3	1.0	1.0
2	1	4	0.7	0.5
3	2	7	3.4	2.5
4	3	4	2.4	2.1
5	4	9	2.0	1.4
6	5	6	1.2	1.5
7	5	8	2.1	2.7
8	5	9	2.6	1.9

The provided matrix T contains the information of all traffic flows: row t defines traffic flow t with origin node o_t given by $T(t,1)$, destination node d_t given by $T(k,2)$, average throughput from origin to destination b_k (in Gbps) given by $T(k,3)$ and average throughput from destination to origin $\underline{b}_t$ (in Gbps) given by $T(t,4)$.

To load the input matrices, run on your MATLAB script: `load('InputData.mat')`

Based on the provided MATLAB codes, develop MATLAB scripts to execute the following tasks.

8.a. Determine the shortest path provided by the network to each traffic flow. Present the length and the sequence of nodes of each path. <u>Results:</u>

```
Flow 1 (1 -> 3): length = 578, Path =  1  5  4  3
Flow 2 (1 -> 4): length = 328, Path =  1  5  4
Flow 3 (2 -> 7): length = 382, Path =  2  4  5  7
Flow 4 (3 -> 4): length = 250, Path =  3  4
Flow 5 (4 -> 9): length = 350, Path =  4  8  9
Flow 6 (5 -> 6): length = 540, Path =  5  7  8  6
Flow 7 (5 -> 8): length = 355, Path =  5  7  8
Flow 8 (5 -> 9): length = 208, Path =  5  7  9
```

8.b. Determine the link loads of all links when all traffic flows are routed through the shortest path provided by the network. Determine also the worst link load. <u>Results:</u>

```
Worst link load = 9.30
{1-2}:  0.00  0.00
{1-5}:  1.70  1.50
{2-4}:  3.40  2.50
{3-4}:  3.40  3.10
{3-6}:  0.00  0.00
{3-8}:  0.00  0.00
{4-5}:  4.90  4.20
{4-8}:  2.00  1.40
{5-7}:  9.30  8.60
{6-8}:  1.50  1.20
{7-8}:  3.30  4.20
{7-9}:  2.60  1.90
{8-9}:  2.00  1.40
```

8.c. Determine the k=4 shortest paths provided by the network for traffic flow 1. Present the length and the sequence of nodes of each path. <u>Results:</u>

```
Path 1 =  1  5  4  3  (length= 578)
Path 2 =  1  2  4  3  (length = 584)
Path 3 =  1  5  7  8  3  (length = 637)
Path 4 =  1  5  4  8  3  (length = 654)
```

8.d. Consider the determination of a symmetrical single routing path solution with minimum worst link load. Run the provided optimization algorithm based on the random strategy. Consider a runtime limit of 5 seconds and all possible routing paths for each flow. Display the assigned routing paths, the resulting link loads, and the performance parameters of the algorithm. Compare the worst link load values of this solution and the solution of **8.b**.

8.e. Repeat task **8.d** but now considering only 6 candidate routing paths for each flow. Compare these results with the ones obtained in task **8.d**.

8.f. Use the provided optimization algorithm (based on the random strategy) to develop an optimization algorithm based on a greedy randomized strategy.

8.g. Run the optimization algorithm based on the greedy randomized strategy (developed in **8.f**) with a runtime limit of 5 seconds and considering all possible routing paths for each flow. Compare these results with the ones obtained in tasks **8.d** and **8.e**.

8.h. Repeat task **8.g** but now considering only 6 candidate routing paths for each flow. Compare these results with the ones obtained in tasks **8.d**, **8.e** and **8.g**.
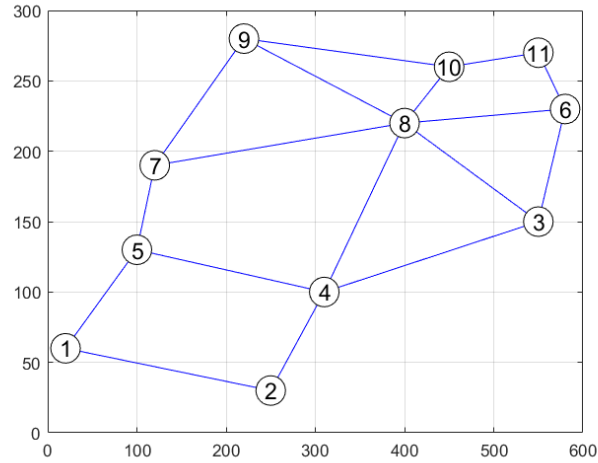
Task 9

Consider the MPLS (Multi-Protocol Label Switching) network of an ISP (Internet Service Provider) with a topology defined over a rectangle with 600 Km by 300 Km as follows:

Consider that all links of the network have a capacity of 10 Gbps. The network supports a unicast service with the following traffic flows (throughput values b_t and $\underline{b}_t$ in Gbps):

t	o_t	d_t	b_t	$\underline{b}_t$
1	1	3	1.0	1.0
2	1	4	0.7	0.5
3	2	6	1.2	2.5
4	2	7	3.4	2.5
5	3	4	2.4	2.1
6	3	5	1.6	1.4
7	3	11	2.7	1.5
8	4	6	2.3	1.8
9	4	9	2.0	1.4
10	5	6	1.2	1.5
11	5	8	2.1	2.7
12	5	10	2.6	1.9
13	6	7	3.2	2.1
14	6	11	1.2	0.8
15	7	11	1.8	1.1

All input data is provided with the same type of matrices as defined in the previous Task 8. To load these input matrices, run on your MATLAB script: `load('InputData2.mat')`

Consider the optimization problem of computing a symmetrical single routing path solution with minimum worst link load. Consider the following algorithmic versions:

Algorithm A: Random algorithm (provided in the MATLAB codes)

Algorithm B: Greedy Randomized algorithm (requested in task **8.f**)

Algorithm C: Multi Start Hill Climbing with initial Random solutions

Algorithm D: Multi Start Hill Climbing with initial Greedy Randomized solutions

9.a. Start by developing Algorithm C (using Algorithm A as starting point) and Algorithm D (using Algorithm B as starting point). Add to all algorithms the appropriate code to register the running time at which each algorithm obtains its best solution.

9.b. Determine the number of routing paths provided by the network to each traffic flow. Register the minimum and maximum number of routing paths and the corresponding traffic flows. <u>Results:</u>

```
Minimum no. of paths= 18
    Flow  2 (1 -> 4)
    Flow 11 (5 -> 8)
Maximum no. of paths= 43
    Flow  3 (2 -> 6)
```

9.c. Run each of the 4 algorithms (A, B, C and D) considering a runtime limit of 5 seconds and all possible routing paths for each flow. Register the worst link load (**W**) of the best obtained solution, the total number of solutions (**No. sol**), the average worst link load value (**Av. W**) among all solutions and the running time (**time**) at which each algorithm has obtained its best solution. Analyze the results and take conclusions. <u>Results (recall that these algorithms include randomness):</u>

```
Random algorithm (all possible paths):

    W = 9.90 Gbps,  No. sol = 828365,  Av. W = 17.75,  time = 2.28 sec

Greedy randomized (all possible paths):

    W = 6.70 Gbps,  No. sol =   3486,  Av. W =  7.89,  time = 0.12 sec

Multi start hill climbing with random (all possible paths):

    W = 6.90 Gbps,  No. sol =    227,  Av. W =  8.74,  time = 2.05 sec

Multi start hill climbing with greedy randomized (all possible paths):

    W = 6.70 Gbps,  No. sol =    850,  Av. W =  7.48,  time = 0.41 sec
```

9.d. Run each of the 4 algorithms considering a runtime limit of 5 seconds and the 6 shortest routing paths for each flow. Register the worst link load (**W**) of the best obtained solution, the total number of solutions (**No. sol**), the average worst link load value (**Av. W**) among all solutions and the running time (**time**) at which each algorithm has obtained its best solution. Analyze these results, compare with the previous results obtained in **9.c** and take conclusions. <u>Results (recall that these algorithms include randomness):</u>

```
Random algorithm (6 paths per flow):

    W = 7.20 Gbps,  No. sol = 889027,  Av. W = 12.82,  time = 0.42 sec

Greedy randomized algorithm (6 paths per flow):

    W = 6.70 Gbps,  No. sol =  16837,  Av. W =  7.90,  time = 0.09 sec

Multi start hill climbing with random (6 paths per flow):

    W = 6.70 Gbps,  No. sol =   2107,  Av. W =  7.93,  time = 0.12 sec

Multi start hill climbing with greedy randomized (6 paths per flow):

    W = 6.70 Gbps,  No. sol =   4699,  Av. W =  7.46,  time = 0.07 sec
```

Task 10

Consider again the MPLS (Multi-Protocol Label Switching) network and the set of traffic flows as specified in Task 9. Recall that the length of all links is provided by the square matrix *L*.

Assume that all routers are of very high availability (i.e., their availability is 1.0). To compute the availability of each link as a square matrix *A* assuming the model considered in *J.-P. Vasseur, M. Pickavet and P. Demeester, "Network Recovery: Protection and Restoration of Optical, SONET-SDH, IP, and MPLS", Elsevier (2004)*, use the following MATLAB code:

```
MTTR= 24:
MTBF= (450*365*24)./L;
A= MTBF./(MTBF + MTTR);
A= A + eye(size(A));
A(isnan(A))= 0;
```

Based on the provided MATLAB code, develop MATLAB scripts to compute the following tasks.

10.a. For each flow, compute the most available path (and its availability value). <u>Results:</u>

```
Flow  1: Availability= 0.9964894 - Path=   1   5   4   3
Flow  2: Availability= 0.9980061 - Path=   1   5   4
Flow  3: Availability= 0.9973444 - Path=   2   4   3   6
Flow  4: Availability= 0.9976781 - Path=   2   4   5   7
Flow  5: Availability= 0.9984802 - Path=   3   4
Flow  6: Availability= 0.9971628 - Path=   3   4   5
Flow  7: Availability= 0.9991178 - Path=   3   6  11
Flow  8: Availability= 0.9979334 - Path=   4   3   6
Flow  9: Availability= 0.9978725 - Path=   4   8   9
Flow 10: Availability= 0.9967200 - Path=   5   7   8   6
Flow 11: Availability= 0.9978426 - Path=   5   7   8
Flow 12: Availability= 0.9974236 - Path=   5   7   8  10
Flow 13: Availability= 0.9971326 - Path=   6   8   7
Flow 14: Availability= 0.9996653 - Path=   6  11
Flow 15: Availability= 0.9971991 - Path=   7   8  10  11
```

10.b. Determine the average availability per flow when each flow is routed by the most available path. <u>Result:</u>

```
Average availability= 0.9977379
```

10.c. Assume that the most available path computed in the previous request is the first routing path of each flow. For each flow, compute a second routing path (if exists) given by the most available path which is link disjoint with the previously computed first routing path. Determine the availability of the routing solution of each flow.

```
Flow  1: Availability= 0.9999859 - Path1=   1   5   4   3
                                   Path2=   1   2   4   8   3
Flow  2: Availability= 0.9999960 - Path1=   1   5   4
                                   Path2=   1   2   4
Flow  3: Availability= 0.9999857 - Path1=   2   4   3   6
                                   Path2=   2   1   5   7   8   6
```

```
Flow  4: Availability= 0.9976781 - Path1=   2   4   5   7
                                    Path2=
Flow  5: Availability= 0.9999970 - Path1=   3   4
                                    Path2=   3   8   4
Flow  6: Availability= 0.9999909 - Path1=   3   4   5
                                    Path2=   3   8   7   5
Flow  7: Availability= 0.9999981 - Path1=   3   6  11
                                    Path2=   3   8  10  11
Flow  8: Availability= 0.9999957 - Path1=   4   3   6
                                    Path2=   4   8   6
Flow  9: Availability= 0.9999945 - Path1=   4   8   9
                                    Path2=   4   5   7   9
Flow 10: Availability= 0.9999889 - Path1=   5   7   8   6
                                    Path2=   5   4   3   6
Flow 11: Availability= 0.9999951 - Path1=   5   7   8
                                    Path2=   5   4   8
Flow 12: Availability= 0.9999888 - Path1=   5   7   8  10
                                    Path2=   5   4   3   6  11  10
Flow 13: Availability= 0.9999907 - Path1=   6   8   7
                                    Path2=   6  11  10   9   7
Flow 14: Availability= 0.9999993 - Path1=   6  11
                                    Path2=   6   8  10  11
Flow 15: Availability= 0.9999902 - Path1=   7   8  10  11
                                    Path2=   7   9   8   6  11
```

10.d. Determine the average availability per flow when each flow is routed by the previous routing solution. Compare this value with the value obtained in **10.b** and take conclusions. <u>Result:</u>

```
Average availability= 0.9998383
```

10.e. Compute how much bandwidth is required on each link to support all flows with 1+1 protection when each flow is routed by the routing solution of **10.c**. Compute also the worst link load and the total bandwidth required among all links. <u>Results:</u>

```
Worst link load = 14.6 Gbps
Total bandwidth among all links = 280.5 Gbps
{ 1- 2}:      4.20  2.70
{ 1- 5}:      2.90  4.00
{ 2- 4}:      6.30  6.50
{ 3- 4}:     12.70 11.80
{ 3- 6}:     10.00  9.20
{ 3- 8}:      7.70  6.00
{ 4- 5}:     14.60 12.90
{ 4- 8}:      9.50  9.30
{ 5- 7}:     13.90 14.10
{ 6- 8}:     11.30  9.40
{ 6-11}:     11.50  7.40
{ 7- 8}:     12.40 14.50
{ 7- 9}:      5.90  5.70
{ 8- 9}:      3.10  3.20
{ 8-10}:      8.30  5.30
{ 9-10}:      2.10  3.20
{10-11}:      9.70  9.20
```

10.f. Compute how much bandwidth is required on each link to support all flows with 1:1 protection when each flow is routed by the routing solution of **10.c**. Compute also the worst link load (what does it mean?) and the total bandwidth required among all links. Compare these results with the results of **8.e** and take conclusions. <u>Results:</u>

```
Worst link load = 14.5 Gbps
Total bandwidth among all links = 247.8 Gbps
{ 1- 2}:      3.50  2.20
{ 1- 5}:      2.90  4.00
{ 2- 4}:      6.30  6.50
{ 3- 4}:     12.70 11.80
{ 3- 6}:     10.00  9.20
{ 3- 8}:      5.00  4.50
{ 4- 5}:     12.60 11.50
{ 4- 8}:      7.40  6.60
{ 5- 7}:     11.90 12.70
{ 6- 8}:      9.00  6.80
{ 6-11}:     11.50  7.40
{ 7- 8}:     12.40 14.50
{ 7- 9}:      3.90  4.30
{ 8- 9}:      3.10  3.20
{ 8-10}:      8.30  5.30
{ 9-10}:      2.10  3.20
{10-11}:      5.70  5.80
```

Task 11

Consider again the MPLS (Multi-Protocol Label Switching) network and the set of traffic flows as specified in Task 9 (and used in Task 10). Recall that the length (in km) of all links is provided by the square matrix *L*. Consider that each link of the network can be set in sleeping mode if it does not support traffic in any of its directions. Assume that the energy consumption of each link:

- is equal to 10 (5 of each of the 2 network interfaces) plus 10% of the link length (in km) when the link is active;
- is equal to 0.5 when the link is in sleeping mode.

Consider the optimization problem of computing a symmetrical single routing path for each flow such that the energy consumption of the network is minimized while a given worst link load fraction α (with $0 < \alpha \le 1$) is guaranteed.

11.a. Develop a Multi Start Hill Climbing algorithm with initial Greedy Randomized solutions to solve this optimization problem (use algorithm D of Task 9 as starting point).

11.b. Run the algorithm considering $\alpha = 1$, a runtime limit of 15 seconds and all possible routing paths for each flow. Register the energy consumption (**E**), the worst link load (**W**), the total number of solutions (**No. sol**), the running time (**time**) at which the algorithm has obtained the best solution and the list of links in sleeping mode of the best solution. Results (recall that these algorithms include randomness):

```
E= 277.70    W = 9.90 Gbps    No. sols = 2959   time = 0.31
List of links in sleeping mode: {1,2} {3,8} {6,8} {7,8} {9,10}
```

11.c. Run the algorithm considering $\alpha = 1$, a runtime limit of 15 seconds and the 6 shortest routing paths for each flow. Register the same parameters as before. Compare these results with the results obtained in **11.b** and take conclusions. Results (recall that these algorithms include randomness):

```
E= 277.70    W = 9.90 Gbps    No. sols = 16414  time = 0.11
List of links in sleeping mode: {1,2} {3,8} {6,8} {7,8} {9,10}
```

11.d. Run the algorithm considering $\alpha = 0.8$, a runtime limit of 15 seconds and all possible routing paths for each flow. Register the same parameters as before. Results (recall that these algorithms include randomness):

```
E= 313.00    W = 7.80 Gbps    No. sols = 1644   time = 3.37
List of links in sleeping mode: {1,5} {3,8} {7,9} {9,10} {10,11}
```

11.e. Run the algorithm considering $\alpha = 0.8$, a runtime limit of 15 seconds and the 6 shortest routing paths for each flow. Register the same parameters as before. Compare these results with the results obtained in **11.b**, **11.c** and **11.d** and take conclusions. Results (recall that these algorithms include randomness):

```
E= 313.00    W = 7.70 Gbps    No. sols = 10537  time = 0.15
List of links in sleeping mode: {1,5} {3,8} {7,9} {9,10} {10,11}
```

Task 12

Consider again the MPLS (Multi-Protocol Label Switching) network specified in Task 9 (and used in Tasks 10 and 11). Recall that the length (in km) of all links is provided by the square matrix *L*.

Consider that the propagation delay of each link of the network in each direction is given by the speed of light on fibers $v = 2 \times 10^5$ km/sec. Start by computing a matrix *D* with the propagation delay of each direction on each link as $D = L / v$.

The network is supporting an anycast service with clients attached in all network nodes. The servers are hosted in Data Centers (DCs) located in different network nodes (i.e., the anycast nodes). The client flows are routed to the closest anycast node (in terms of delay) and through the routing path providing the shortest delay.

The average download throughput b_i (in Gbps) and upload throughput $\underline{b}_i$ (in Gbps) generated by the clients at each network node *i* is given by:

i	1	2	3	4	5	6	7	8	9	10	11
b_i	0.7	1.5	2.1	1.0	1.3	2.7	2.2	0.8	1.7	1.9	2.8
$\underline{b}_i$	0.1	0.2	0.3	0.1	0.1	0.4	0.3	0.1	0.2	0.3	0.4

Develop MATLAB scripts to compute the following tasks.

12.a. Consider that the service is provided by two DCs, one connected to network node 3 and the other connected to network node 5 (i.e., the anycast nodes are node 3 and node 5). Compute the worst round-trip delay (**W**) and the average round trip delay (**A**) among all clients, presenting both values in milliseconds. Compute also the network node whose clients suffer the worst round-trip delay. Results:

```
Anycast nodes= 3  5
Node with the worst round-trip delay: 2
W = 3.14 ms        A = 1.42 ms
```

12.b. Consider again that the anycast nodes are node 3 and node 5. Determine the link loads of all links and the worst link load. Results:

```
Anycast nodes= 3  5
Worst link load = 5.50 Gbps
{1-2}:        0.00  0.00
{1-5}:        0.70  0.10
{2-4}:        1.50  0.20
{3-4}:        0.00  0.00
{3-6}:        0.80  5.50
{3-8}:        0.40  2.70
{4-5}:        2.50  0.30
{4-8}:        0.00  0.00
{5-7}:        0.50  3.90
{6-8}:        0.00  0.00
{6-11}:       0.40  2.80
{7-8}:        0.00  0.00
```

```
{7-9}:        0.20  1.70
{8-9}:        0.00  0.00
{8-10}:       0.30  1.90
{9-10}:       0.00  0.00
{10-11}:      0.00  0.00
```

12.c. Consider that you can freely select the nodes to connect the two DCs running the service. Try all possible combinations of 2 nodes and select the one that minimizes the worst round-trip delay. Indicate the two selected nodes, the worst round-trip delay, and the worst link load of the best solution. Compare these results with the results obtained for the anycast nodes 3 and 5 and take conclusions. <u>Results:</u>

```
Best Anycast nodes= 1  8

Worst round-trip delay = 2.37 ms

Worst link load = 4.70 Gbps
```

12.d. Again, consider that you can freely select the nodes to connect the two DCs running the service. Try all possible combinations of 2 nodes and select the one that minimizes the worst link load. Indicate the two selected nodes, the worst round-trip delay, and the worst link load of the best solution. Compare these results with all previous results and take conclusions. <u>Results:</u>

```
Best Anycast nodes= 6  8

Worst round-trip delay = 4.66 ms

Worst link load = 4.20 Gbps
```