



**DEPARTAMENTO DE ELETRÓNICA, TELECOMUNICAÇÕES  
E INFORMÁTICA**

**MESTRADO EM ENGENHARIA DE COMPUTADORES E TELEMÁTICA**

**ANO 2025/2026**

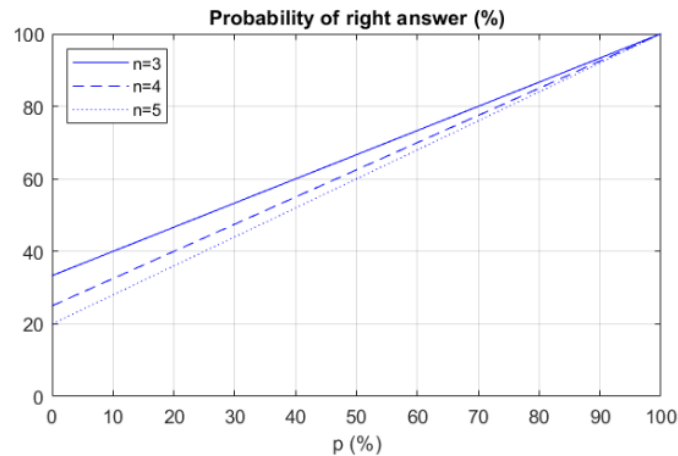
**MODELAÇÃO E DESEMPENHO DE REDES E SERVIÇOS**

**PRACTICAL GUIDE**

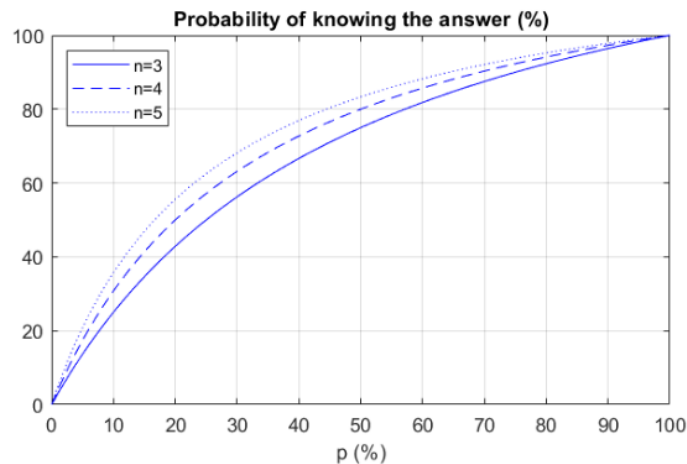
## Task 1

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 that 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. Answer: 70%
- 1.b. When  $p = 70\%$  and  $n = 5$ , determine the probability of the student to know the answer when he selects the right answer. Answer: 92.1%
- 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? Answer:



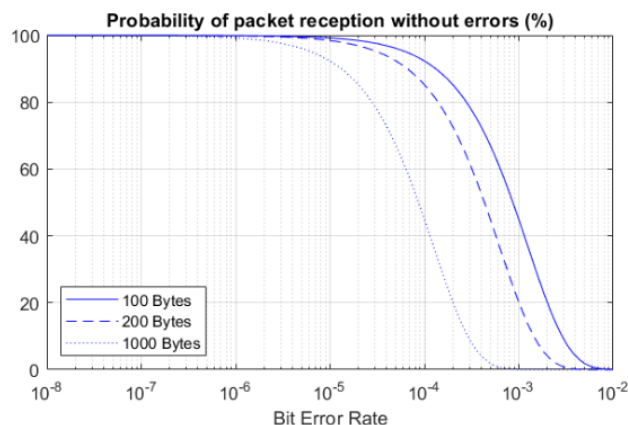
- 1.d. Draw a plot (with the same look as the plot below) with the probability of the student to know the answer when he selects the right answer as a function of the probability  $p$  (consider  $n = 3, 4$  and  $5$ ). What do you conclude from these results? Answer:



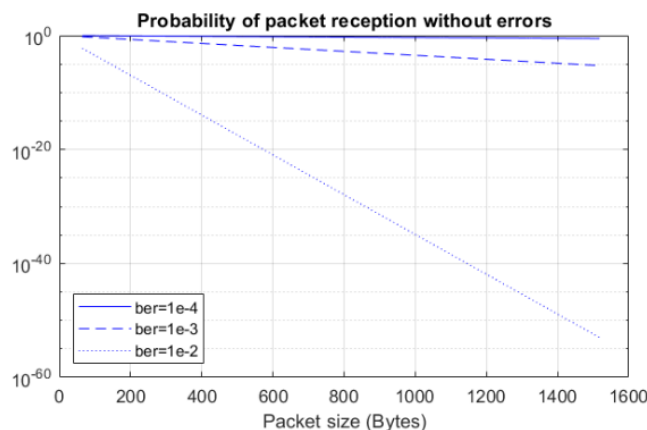
## Task 2

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  $p = 10^{-2}$ . Answer: 0.0322%
- 2.b. Determine the probability of a data frame of 1000 Bytes to be received with exactly one error when  $p = 10^{-3}$ . Answer: 0.2676%
- 2.c. Determine the probability of a data frame of 200 Bytes to be received with one or more errors when  $p = 10^{-4}$ . Answer: 14.7863%
- 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? Answer:

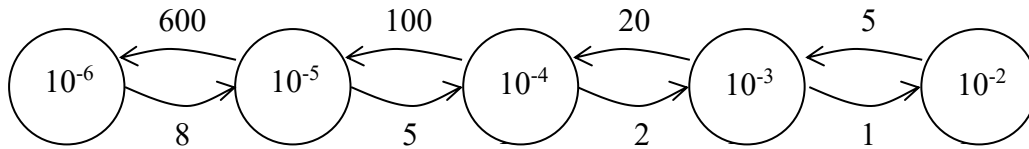


- 2.e. Draw a plot using a logarithmic scale for the Y-axis (use the MATLAB function `semilogy`) with the same look as the plot below with the probability of a data frame being received without errors (for  $p = 10^{-4}$ ,  $10^{-3}$  and  $10^{-2}$ ) as a function of the packet size (all integer values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



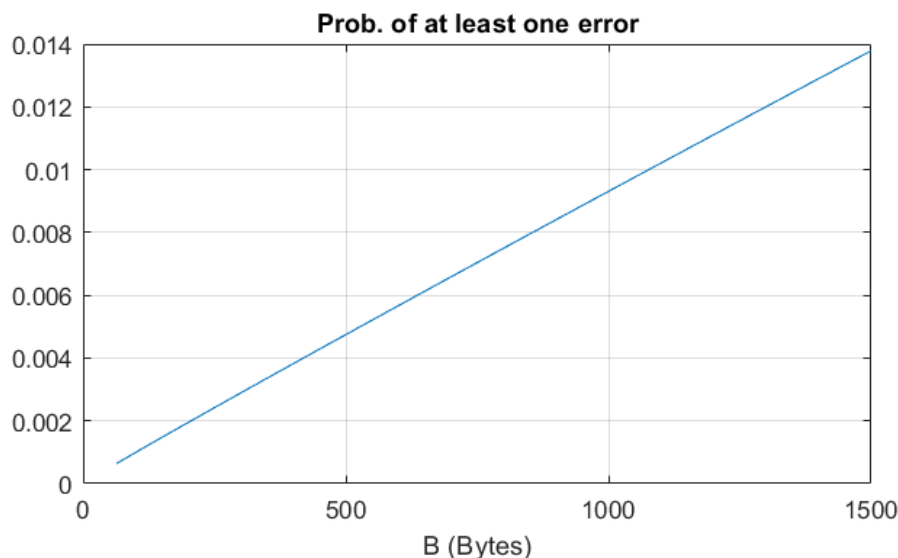
### Task 3

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:

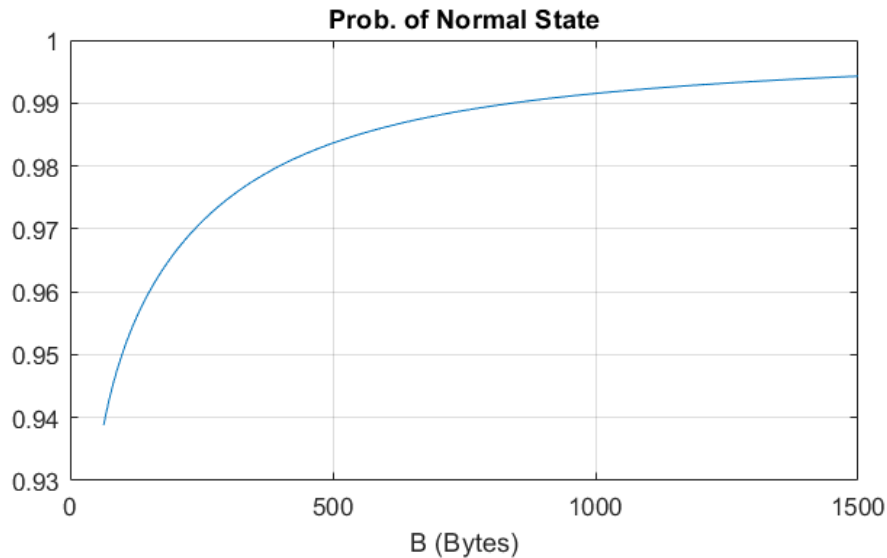


where the state transition rates are in number of transitions per hour. Consider that the link is in an interference state when its *ber* is at least  $10^{-3}$  and in a normal state, otherwise. Assume that all stations detect with a probability of 100% when the data frames sent by the other stations are received with errors. Determine:

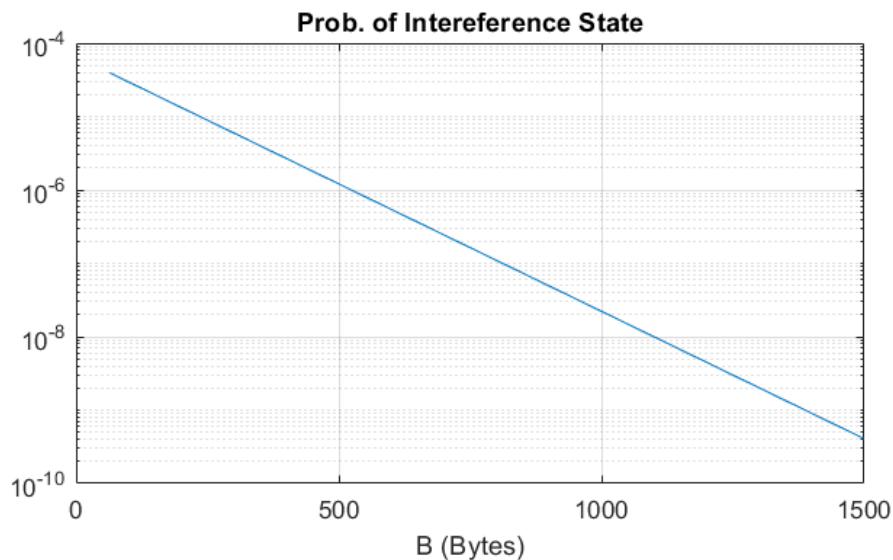
- 3.a. the probability of the link being in each of the five states; answer:  
 $9.86 \times 10^{-1}$  ( $10^{-6}$ ),  $1.31 \times 10^{-2}$  ( $10^{-5}$ ),  $6.57 \times 10^{-4}$  ( $10^{-4}$ ),  $6.57 \times 10^{-5}$  ( $10^{-3}$ ),  $1.31 \times 10^{-5}$  ( $10^{-2}$ )
- 3.b. the average percentage of time the link is in each of the five states; answer:  
 $9.86 \times 10^{-1}$  ( $10^{-6}$ ),  $1.31 \times 10^{-2}$  ( $10^{-5}$ ),  $6.57 \times 10^{-4}$  ( $10^{-4}$ ),  $6.57 \times 10^{-5}$  ( $10^{-3}$ ),  $1.31 \times 10^{-5}$  ( $10^{-2}$ )
- 3.c. the average *ber* of the link; answer:  $1.38 \times 10^{-6}$
- 3.d. the average holding time (in minutes) of the link in each of the five states; answer: 7.5 min ( $10^{-6}$ ), 0.10 min ( $10^{-5}$ ), 0.59 min ( $10^{-4}$ ), 2.86 min ( $10^{-3}$ ), 12.0 min ( $10^{-2}$ )
- 3.e. the probability of the link being in the normal state and in interference state; answer:  
 $0.999921$  (normal),  $7.89 \times 10^{-5}$  (interference)
- 3.f. the average *ber* of the link when it is in the normal state and when it is in the interference state; answer:  $1.18 \times 10^{-6}$  (normal),  $2.50 \times 10^{-3}$  (interference)
- 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 with at least one error as a function of the packet size (from 64 Bytes up to 1500 Bytes); analyze and justify the results; answer:



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



- 3.i. considering that a data frame of size  $B$  (in Bytes) sent by one source station is received without errors 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 in the interference state as a function of the packet size (from 64 Bytes up to 1500 Bytes); analyze and justify the results; answer:



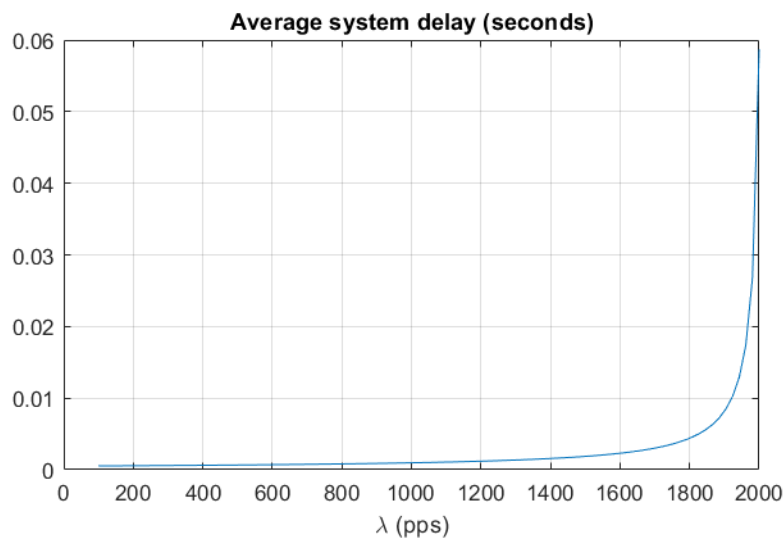
## Task 4

Consider an ideal link (i.e., with a  $ber = 0$ ) from one router to another router with a capacity of  $C$  Mbps ( $1 \text{ Mbps} = 10^6 \text{ bps}$ ) for IP communications. The link has a propagation delay of  $10 \mu\text{s}$  ( $1 \mu\text{s} = 10^{-6}$  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  $\lambda$  pps (packets/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  $\lambda = 1000$  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,  $4.96 \times 10^{-4}$  seconds
- 4.b. the average throughput (in Mbps) of the IP flow; answer: 4.96 Mbps
- 4.c. the capacity of the link, in packets/second; answer: 2016.06 pps
- 4.d. the average packet queuing delay and average packet system delay of the IP flow (the system delay is the queuing delay + transmission time + propagation delay) using the  $M/G/1$  queuing model; answer: queuing –  $4.60 \times 10^{-4}$  seconds, system –  $9.66 \times 10^{-4}$  seconds
- 4.e. draw a plot with the same look as the plot below with the average packet system delay as a function of the packet arrival rate  $\lambda$  (from  $\lambda = 100$  pps up to  $\lambda = 2000$  pps); analyze the results and take conclusions.



## 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 capacity  $C$  (in Mbps) and a queue of size  $f$  (in Bytes) with a FIFO (First-In-First-Out) scheduling 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*:

- $\lambda$  – 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 finishes the transmission of the  $P^{\text{th}}$  packet ( $P$  is one of the input parameter); in *Simulator1*, the queued packets at the end of the simulation do not count for the performance estimation.

*Simulator1* makes use of 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), TRANSPACKETS (number of transmitted packets), TRANSBYTES (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 of the simulation as:

$$PL = 100 \times \text{LOSTPACKETS} / \text{TOTALPACKETS}$$

$$APD = 1000 \times \text{DELAYS} / \text{TRANSPACKETS}$$

$$MPD = 1000 \times \text{MAXDELAY}$$

$$TT = 10^{-6} \times \text{TRANSBYTES} \times 8 / \text{total simulated time}$$

- 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). Results (recall that these are simulation results):

```
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 now running *Simulator1* 100 times. Compare these results with the previous ones and take conclusions. Results (recall that these are simulation results):

```
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.** Consider system modelled by an  $M/G/1$  queueing model<sup>1</sup>. 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. Results:

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

- 5.d.** Repeat experiment **5.b** now considering  $f = 10.000$  Bytes (~10 KBytes) Justify the differences between these results and the results of experiment **5.b**. Results (recall that these are simulation results):

```
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.e.** Repeat experiment **5.b** now considering  $f = 2.000$  Bytes (~2 KBytes) Justify the differences between these results and the results of experiments **5.b** and **5.d**. Results (recall that these are simulation results):

```
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
```

<sup>1</sup> 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 does not introduce errors (the typical case of a wired access network). In this Task 6, assume that the link is provided by a wireless access network (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). So, packets can be lost for two reasons: (i) they can be dropped in the input queue, or (ii) they can be discarded after being transmitted.

**6.a.** Develop *Simulator2* by changing the provided *Simulator1* to consider that the link introduces a BER (Bit Error Rate) given by  $b$ . The input parameters of *Simulator2* must be all the input parameters of *Simulator1* plus parameter  $b$ . The performance parameters estimated by *Simulator2* must be the same as the ones of *Simulator1*. The stopping criterion of *Simulator2* must be the time instant when the link finishes the transmission of the  $P^{\text{th}}$  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  $\lambda = 1800$  pps,  $C = 10$  Mbps,  $f = 1.000.000$  Bytes and  $b = 10^{-6}$ . Compare these results with the results of **5.b** and take conclusions. Results (recall that these are simulation results):

```
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** now considering  $f = 10.000$  Bytes. Justify the differences between these results and the results of **5.d**. Results (recall that these are simulation results):

```
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 experiment **6.b** now considering  $f = 2.000$  Bytes. Justify the differences between these results and the results of experiment **5.e**. Results (recall that these are simulation results):

```
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<sup>2</sup>.

- 7.a.** Develop *Simulator3* by changing the provided *Simulator1* to consider the  $n$  additional VoIP packet flows<sup>3</sup>. Consider that packets of all flows (data and VoIP) are queued on a single queue served with a FIFO scheduling 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** now considering  $f = 10.000$  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
  
```

<sup>2</sup> These values are based on G.726 standard, at 32 kbps, assuming a packet size of  $\pm 10$  Bytes with a uniform distribution around its average size and a jitter of  $\pm 4$  milliseconds with a uniform distribution around the average inter arrival time of 20 milliseconds.

<sup>3</sup> 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.b** now considering  $f = 2.000$  Bytes (~2 KBytes) Justify the differences between these results and the results of experiments **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  $\lambda = 1800$  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** now considering  $f = 10.000$  Bytes (~10 KBytes). Compare these results with the results of **7.c** and take conclusions. Results (recall that these are simulation results):

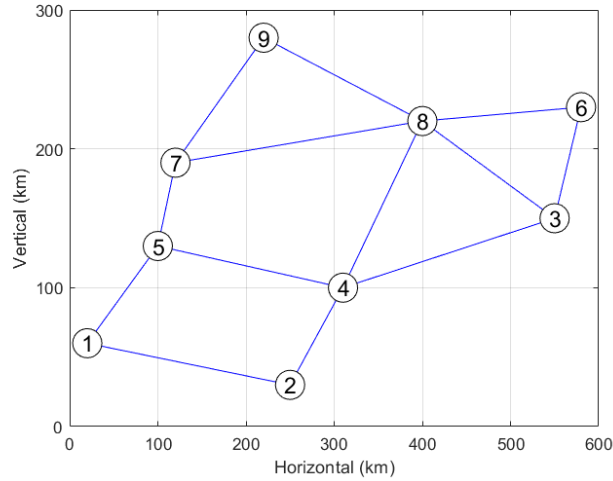
PacketLoss of data (%) = 2.66e+00 +- 6.99e-02  
 PacketLoss of VoIP (%) = 3.45e-01 +- 2.47e-02  
 Av. Packet Delay of data (ms) = 4.36e+00 +- 3.71e-02  
 Av. Packet Delay of VoIP (ms) = 5.34e-01 +- 1.61e-03  
 Max. Packet Delay of data (ms) = 9.96e+00 +- 2.58e-02  
 Max. Packet Delay of VoIP (ms) = 1.38e+00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

**7.g.** Repeat the experiment **7.e** now considering  $f = 2.000$  Bytes ( $\sim 2$  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 MATLAB codes provided, 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. Results:

```
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 worst link load and the link loads of all links when all traffic flows are routed through the shortest path provided by the network. load. Results:

```
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
```

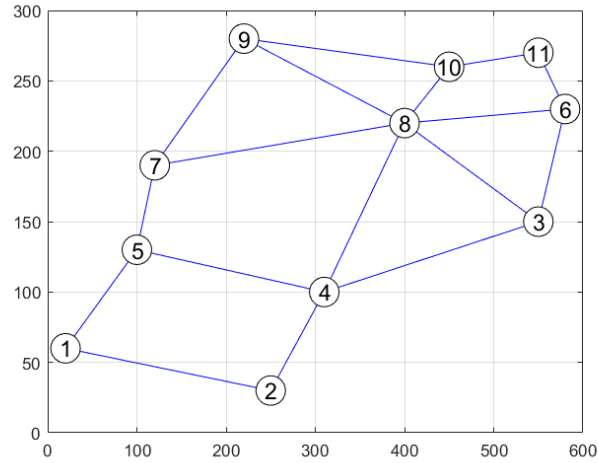
- 8.c.** Determine the  $k = 4$  shortest paths provided by the network for traffic flow 1. Present the length and the sequence of nodes of each path. Results:

```
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. Present 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 the 6 shortest routing paths as candidate paths for each flow. Compare the worst link load values of this solution and the solution of **8.d**.
- 8.f.** Change the provided optimization algorithm (based on the random strategy) to be based on the 1<sup>st</sup> alternative of a greedy randomized strategy (see slide 16 of Module 4).
- 8.g.** Run the optimization algorithm 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 the 6 shortest paths as candidate 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 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 (developed 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.** 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. Results:

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

- 9.b.** Develop Algorithm C (using Algorithm A as starting point) and Algorithm D (using Algorithm B as starting point).

- 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. Results (recall that these algorithms include randomness):

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 as candidate 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. Results (recall that these algorithms include randomness):

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. Assume that the capacity of all links is now ten times larger, i.e., 100 Gbps. Recall that the length (in Km) 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). Moreover, the availability of each link is given by 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)* with the Mean Time to Repair (MTTR) parameter of 24 hours and the Cable Cut (CC) parameter of 450 Km. Based on the matrix  $L$  with the length of all links (in Km), the availability of each link can be computed as a square matrix with the following MATLAB code:

```
MTTR= 24;
CC= 450;
MTBF= (CC*365*24) ./L;
A= MTBF ./ (MTBF + MTTR);
A(isnan(A))= 1;
```

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). Results:

```
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 computed in 10.a. Result:

```
Average availability= 0.9977379
```

**10.c.** Assume that the most available path (computed in 10.a) of each flow is its first routing path. 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. Is there always a second routing path for all flows? Why? Results:

## Practical Guide

```

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.  
Result:

Average availability= 0.9998383

**10.e.** Compute how much bandwidth capacity 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 bandwidth capacity required among all links and the total bandwidth required in all links. Results:

```

Worst bandwidth capacity = 14.6 Gbps
Total bandwidth capacity on 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 capacity 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 bandwidth capacity required among all links and the total bandwidth required in all links. Compare these results with the results of **10.e** and take conclusions. Results:

```
Worst bandwidth capacity = 14.5 Gbps
Total bandwidth capacity on 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 network and the set of traffic flows as specified in Task 9 (and used in Task 10). Remember 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 20.0 (10.0 of each of the 2 network interfaces) plus 10% of the link length (in km) when the link is active,
- is equal to 1.0 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  $\alpha$  (with  $0 < \alpha \leq 1$ ) is guaranteed.

- 11.a.** Determine the total link energy consumption (**E**) and the worst link load (**W**) of the network and when all traffic flows are routed through the shortest path provided by the network. What do you conclude? Results:

$E = 452.40$        $W = 10.40$  Gbps

- 11.b.** 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.c.** Run the algorithm considering  $\alpha = 1$ , a runtime limit of 30 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):

$E = 391.00$        $W = 9.90$  Gbps      No. sols = 5895      time = 12.88

List of links in sleeping mode: {1,5} {3,8} {6,8} {7,8} {8,9} {8,10}

- 11.d.** Run the algorithm considering  $\alpha = 1$ , a runtime limit of 30 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.c** and take conclusions. Results (recall that these algorithms include randomness):

$E = 400.20$        $W = 9.90$  Gbps      No. sols = 32711      time = 0.34

List of links in sleeping mode: {1,2} {3,8} {6,8} {7,8} {9,10}

- 11.e.** Run the algorithm considering  $\alpha = 0.7$ , a runtime limit of 30 seconds and all possible routing paths for each flow. Register the same parameters as before. Results (recall that these algorithms include randomness):

$E = 492.00$        $W = 6.90$  Gbps      No. sols = 313      time = 3.91

List of links in sleeping mode: {1,2} {9,10} {10,11}

- 11.f.** Run the algorithm considering  $\alpha = 0.7$ , a runtime limit of 30 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.c**, **11.d** and **11.e** and take conclusions. Results (recall that these algorithms include randomness):

$E = 492.00$        $W = 6.90$  Gbps      No. sols = 1383      time = 0.46

List of links in sleeping mode: {1,2} {9,10} {10,11}