Open and virtualized networks - Network Exercises - Part 10

Network Exercise

The aim of these exercises is to introduce in the Network abstraction the concept of traffic matrix and the request of a specific bitrate (traffic grooming size) for a connection request. This exercises can be part of the material for the final exam questions. You are strongly encouraged to find yourself a solution to the presented problems.

1. Modify the Network class such that the number of channels available in each line is given once as an attribute `Nch` and set accordingly throughout the software. Set `Nch` to 20.

2. In your main program, perform a Monte-Carlo simulation on the traffic matrix with N_{MC} realizations. Keep the traffic matrix to uniform with constant request_rate for each node pair. Randomize instead the order in which the connection requests between each couple of nodes of the traffic matrix are issued. (You can use the shuffle() function to randomize the node pairs). For each Monte-Carlo realization start with the empty network and load it with a different node pair sequence of connection requests. Start with $N_{MC} = 100$. Assume `shannon` as transceiver technology.

3. Calculate the average traffic in the network and the average bitrate per lightpath. Obtain these results as the average of the traffic and the average bitrate obtained in each different Monte-Carlo realization.

4. Calculate also the average congestion of the network lines. The congestion of a line is defined as the percentage of allocated channel to the total available channel `Nch`. Calculate the congestion for each line of the Network and average the result over all the Monte-Carlo iterations. Plot the average congestion at the end of the Monte-Carlo simulation as an histogram with the lines on the X-axis and its average congestion on the Y-axis

5. set a rate_request multiplier on the traffic matrix. It is an integer number multiplying the values of the traffic matrix in order to increase the number of lightpath that needs to be allocated for each connection request between a node pair. Tune this parameter so that the lightpaths requests are enough to saturate the network. Tune also the N_{MC}, by increasing its value to 200, 300, 500 and so on until the average rate per lightpath converges to an almost constant value. Look what happens to the congestion as you increase the rate multiplier.

6. Look at the congestion plots and identify the most congested line. Write a method in the **Network** class called **upgrade_line()** which takes a line object (or its string description) and perform an upgrade of the equipment

by decreasing by 3 dB the noise figure of the amplifiers only for that line. Redo the Monte-Carlo simulation on the upgraded network and check for improvements in congestion and overal network capacity.