Open and virtualized networks - Network Exercises - Part 9

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 **Connection** class in order to support data-rate requests. Add an attribute **rate_request** to the class Connection where you store the requested rate. *Example*: instead of just requesting the connection between A and B and accept the available rate, request a connection between A and B with a requested minimum bitrate.

2. Modify the **stream()** method of the class **Network** so that it marks a connection request as blocked if the requested rate cannot be satisfied. Modify the lightpath allocation method so that if a path is found but the available bitrate is less than the requested rate, it searches for other lightpath so that the sum of their capacity is at least equal to the requested. If the requested rate is reached one or more lightpaths are allocated for that connection request, otherwise it is marked as blocked. Use the **bitrate** attributes of class **Connection** to store the sum of the bitrates of the lighpath allocated to that connection. Modify the **calculate_capacity()** method to perform the sum.

3. In your main script, modify the way your software generates the connection requests. Instead of generating a sequence of N_{MC} random requests, generate them accordingly to a **uniform traffic matrix**.

   A traffic matrix T is defined as a matrix with a row and a column for each node of the network. Each element $T_{i,j}$ represents the bitrate request in Gbps between the nodes i, j. If $T_{i,j} = 0$ then no connection request is issued between i, j. If $T_{i,j} = \text{Inf}$, then it is intended as pure connection request where whatever available bitrate from the lightpath is accepted.

   Assume, for now, a **uniform** distribution: all node pairs requests always the same bitrate of 100, 200 or 400 Gbps. Use the pandas **DataFrame** to generate the traffic matrix (use the **values()** method from the DataFrame to get a numpy matrix).

4. Plot a 3D histogram of the requested rate. Plot also an histogram of the allocated rates and compare with the requested ones for all the three transceiver types. Put on the X, Y and Z axis the source nodes, destination nodes and bitrate, respectively. Also calculate the total capacity allocated into the network. You can play with the fiber and amplifier parameters

to degrade the quality of transmission and increase the number of blocked requests.