Open and virtualized networks - Network Exercises - Part 11

Network Exercise

Considering the last version of the software that you have developed until exercise set n.10. You should haver now a software perfoming a monte-carlo analysis on a network of choide, described with a json file with certain physical parameters described mainly in the network and line classes. In the following points, some open analysis suggestions are given, which will be discussed during the final exam.

1. Use the **rate_multiplier** parameter to progressively load the network. Set the channel number to 10 and N_{MC} to 50 (or less) in order to speed up the simulations adn transceiver to shannon technology. What do you observe on the total allocated traffic, the average rate per lightpath and the congestion as the network load increases with the **rate_multiplier**?

2. From the previous point, fix the rate multiplier to the full network load. What happens to the total allocatec traffic and the congestion if you increase the number of channels to 15?

3. Consider the last point of exercise 10 where you have been asked to write the **upgrade_line** method by improving a specific line noise figure by 3 dB. What you observe on the total allocated traffic, the average rate per lightpath and the congestion, if instead of upgrading the most congested line you upgrade one of the least used? Repeat the two upgrade cases at two different network loads ("full" and "not full" states). How you explain the results that you get?

4. Repeat the previous point by degrading the noise figure by 3 dB (i.e, noise figure 3 dB larger). What you observe? Why?

5. We have seen the results on a network made of SMF as fiber type, with dispersion (`beta`) equal to 21.27e-27 [s/(Hz m)]. What happens to the total traffic and the average lightpath bitrate if you assume instead a network made up of LEAF fiber only (`beta` equal to 6.58e-27 [s/(Hz m)])? Why?

6. Modify the nodes.json file and construct your own network topology bigger with respect to the previous one. Construct it by adding 3 further nodes and connect them as you want. HINT: don't make the new network fully connected (each node is not connected to every other node) and try to keep the average length of the lines connecting the new nodes not shorter of that in the last network json provided. Repeat the previous points on the new network and make your comments.