Open Optical Networks
Lab 10

December 16, 2020

These exercise sets cover some aspects you will find useful for the final exam software development. 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.

Errata Corrige

- **LAB 8**: The text of the exercises has been loaded newly with the correct formulae and input data.
- **LAB 9**:
  - $\alpha_{dB} =$ 0.2 dB/km;
  - $|\beta_2| =$ 2.13 e−26 (m $\text{Hz}^2)^{-1}$;
  - $\gamma =$ 1.27e-3 $(\text{m W})^{-1}$;
  - $R_s = 32$ GHz
  - df = 50 GHz

  Remember that: $\alpha = \frac{\alpha_{dB}}{20\, log_{10}(e)}$ and $L_{eff} = \frac{1}{2\alpha}$;
  Define in the class **Line** the method **nli_generation** that evaluate the total amount generated by the nonlinear interface noise using the formula (in linear units):

$$NLI = Pch^3\, \eta_{nli} N_{span} B_n$$

  where B_n is the noise bandwidth (12.5 GHz) and N_{span} is the number of fiber span within the considered line.

Exercises

1. Modify the method **noise_generation** within the class **Line** including the methods for the computation of the ASE and the NLI and removing the old formula.

2. Create a new method **optimized_launch_power** in the class **Line** for the determination of the optimal launch power.
   Hint: exploit the formula of the Local Optimization Global Optimization (LOGO) strategy.

3. Modify the method **propagate** of the class **Node** in order to set for each line the optimal launch power.

4. Modify the method **calculate_bit_rate** of the class **Network** to have as input a light-path instead of the path in order to retrieve the specific symbol rate R_s.

5. Run the main that evaluates the distribution of the SNR on a list of 100 randomly chosen connections for the three newly provided network description json files and plot the histogram of the accepted connections bit rates calculating the overall average. Also calculate the total capacity allocated into the network. Compare the three results obtained for the three different transceiver strategies.

6. In your main script, modify the way your code generates the connection requests. Instead of generating a sequence of N 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 bit rate request in Gbps between the nodes i, j. If $T_{i,j} = 0$, there is no connection request between i, j. If $T_{i,j} = \text{Inf}$, all the possible achievable traffic is allocated. Add in the class **Network** a method that creates and manages the connections given a traffic matrix. This method chooses a random pair of source-destination nodes with a non zero request $T_{i,j}$. After the connection streaming, the allocated traffic has to be subtracted to the given traffic matrix. Assume a **uniform** distribution: all node pair requests always the same bit rate of 100 * M Gbps, where M is an increasing integer number (1, 2, 3, ...).

7. Run again the main script of point 5 increasing M until the network saturation and graphically report the results.