NetXPTO - NetPlanner

6 de Dezembro de 2017

Conteúdo

1 Introduction **3**

2 Simulator Structure **4**

2.1 System 4

2.2 Blocks 4

2.3 Signals 4

3 Development Cycle **5**

4 Case Studies **6**

4.1 Opaque with 1+1 Protection 7

4.1.1 Physical Network Topology 7

4.1.2 Dimensioning using ILP 15

4.1.3 Dimensioning using Heuristics 22

4.1.4 Comparative Analysis 22

4.2 Transparent with 1+1 Protection 23

4.2.1 Physical Network Topology 23

4.2.2 Dimensioning using ILP 27

4.2.3 Dimensioning using Heuristics 34

4.2.4 Comparative Analysis 34

4.3 Translucent with 1+1 Protection 35

4.3.1 Physical Network Topology 35

4.3.2 Dimensioning using ILP 39

4.3.3 Dimensioning using Analytical solution 41

4.3.4 Dimensioning using Heuristics 42

4.3.5 Comparative Analysis 42

5 Appendices **43**

5.1 Installing Net2Plan and its main options 44

5.1.1 Net2Plan download and installation 44

5.1.2 Net2Plan Options and installing solvers 44

5.2 Net2Plan Tools . . . 47
5.2.1 Creating Traffic Matrices . . . 47
5.2.2 Creating the Network topologies . . . 50
5.2.3 Routing and Grooming . . . 57
5.2.4 Reports . . . 58
5.3 Results . . . 61
5.3.1 Opaque with 1+1 protection . . . 62
5.3.2 Transparent with 1+1 protection . . . 64
5.4 Simulations . . . 67
5.5 Implementing new algorithms on Net2Plan . . . 68
5.6 Developing new Reports . . . 71
5.7 Installing LPSOLVE for using in MatLab . . . 73
5.8 Opaque with 1+1 protection appendices . . . 74
5.8.1 Script using MatLab . . . 74
5.8.2 ILP using LPSolve . . . 79
5.8.3 Results using LPSolve . . . 96
5.9 Transparent with 1+1 protection appendices . . . 97
5.9.1 Script using MatLab . . . 97
5.9.2 ILP using LPSolve . . . 99
5.9.3 Results using LPSolve . . . 107
5.10 Translucent with 1+1 protection appendices . . . 108
5.10.1 Script using MatLab . . . 108
5.10.2 ILP using LPSolve . . . 108
5.10.3 Results using LPSolve . . . 108

6 Master Dissertations **109**
Vasco Braz, Dimensioning and Optimization of Node Architecture in Optical Transport Networks, University of Aveiro, 2016 . . . 110

Capítulo 1

Introduction

LinkPlanner is devoted to the simulation of point-to-point links.

Capítulo 2

Simulator Structure

LinkPlanner is a signals open-source simulator.

The major entity is the system.

A system comprises a set of blocks.

The blocks interact with each other through signals.

2.1 System

2.2 Blocks

2.3 Signals

List of available signals:

- Signal

Capítulo 3

Development Cycle

The NetXPTO-LinkPlanner has been developed by several people using git as a version control system. The NetXPTO-LinkPlanner repository is located in the GitHub site http://github.com/netxpto/linkplanner. The more updated functional version of the software is in the branch master. Master should be considered a functional beta version of the software. Periodically new releases are delivered from the master branch under the branch name Release<Year><Month><Day>. The integration of the work of all people is performed by Armando Nolasco Pinto in the branch Develop. Each developer has is how branch with his/her name.

Capítulo 4

Case Studies

4.1 Opaque with 1+1 Protection

In this case study we focus on the opaque case with 1 + 1 protection. The opaque transport mode performs OEO (optical-electric-optical) conversions on each intermediate node from the source to the destination node. One advantage of this mode of transport is that it eliminates accumulation of physical impairments, and allows optimum grooming by performing grooming at each node.

4.1.1 Physical Network Topology

Student Name	:	Tiago Esteves (October 03, 2017 -)
Goal	:	Implement the dimensioning of optical networks in the opaque transport mode.

4.1.1.1 Reference Network

In the figure below we ca see that our reference network consists of 6 nodes and 8 Bidirectional links. The average length of the links was chosen so that the following calculations are more simplistic, for this was created a matrix of distances between the respective nodes. Finally, ODU's matrices were also created to be able to determine the total traffic used in each scenario.

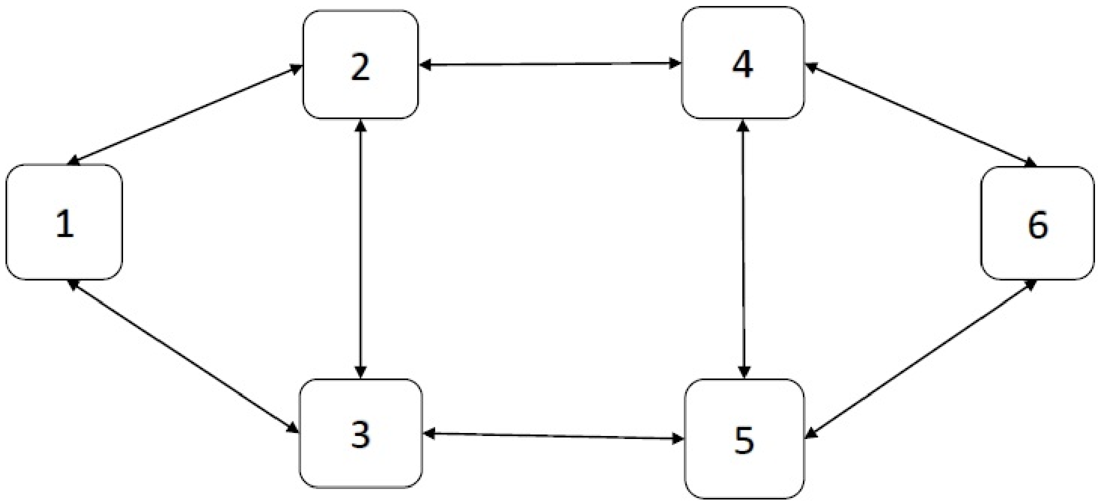


Figura 4.1: Physical topology of the reference network.

The distance matrix is the same for the two scenarios but the ODU's matrices are not. In this way only the matrices for the case of low traffic are elucidated, being that in the case of a high traffic it is only necessary to multiply these matrices by the value 10.

$$Dist = \begin{bmatrix} 0 & 500 & 500 & 0 & 0 & 0 \\ 500 & 0 & 400 & 500 & 0 & 0 \\ 500 & 400 & 0 & 0 & 500 & 0 \\ 0 & 500 & 0 & 0 & 600 & 450 \\ 0 & 0 & 500 & 600 & 0 & 550 \\ 0 & 0 & 0 & 450 & 550 & 0 \end{bmatrix}$$

$$ODU0 = \begin{bmatrix} 0 & 5 & 1 & 3 & 1 & 3 \\ 5 & 0 & 0 & 1 & 5 & 0 \\ 1 & 0 & 0 & 1 & 4 & 1 \\ 3 & 1 & 1 & 0 & 1 & 0 \\ 1 & 5 & 4 & 1 & 0 & 3 \\ 3 & 0 & 1 & 1 & 3 & 0 \end{bmatrix} \quad ODU1 = \begin{bmatrix} 0 & 2 & 4 & 2 & 0 & 5 \\ 2 & 0 & 0 & 3 & 1 & 1 \\ 4 & 0 & 0 & 1 & 1 & 0 \\ 3 & 3 & 1 & 0 & 1 & 3 \\ 0 & 1 & 1 & 1 & 0 & 1 \\ 5 & 1 & 0 & 3 & 1 & 0 \end{bmatrix} \quad ODU2 = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

$$ODU3 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \quad ODU4 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \end{bmatrix}$$

The values indicated in the distance matrix, referred to below, are expressed in kilometers (Km) and as it couldn't be otherwise, this matrix is symmetric. In relation to the traffic matrices each ODU, referred previously, has its respective value being that the ODU0 corresponds to 1.25 Gbits/s, ODU1 to 2.5 Gbits/s, ODU2 to 10 Gbits/s, ODU3 to 40 Gbits/s and finally the ODU4 corresponds to 100 Gbits/s. As we can see these matrices are bidirectional because they are symmetric matrices and as such the traffic sent in one direction must be the same traffic sent in the opposite direction.

Through these ODU's we can calculate total network traffic for the low traffic scenario:

T_1^0 = 60x1.25 = 75 Gbits/s $\quad$ T_1^1 = 50x2.5 = 125 Gbits/s $\quad$ T_1^2 = 16x10 = 160 Gbits/s

T_1^3 = 6x40 = 240 Gbits/s $\quad$ T_1^4 = 4x100 = 400 Gbits/s

T_1 = 75 + 125 + 160 + 240 + 400 = 1000 Gbits/s $\quad$ T = 1000/2 = 0.5 Tbits/s

Where the variable T_1^x represents the unidirectional traffic of the ODUx, for example, T_1^0 represents the unidirectional traffic of the ODU0 and T_1^1 represents the unidirectional traffic of the ODU1. The variable T_1 represents the total of unidirectional traffic that is injected into the network and finally the variable T represents the total of bidirectional traffic.

We can thus conclude that the total traffic for the two scenarios is as follows:

- Low Traffic: **0.5 TBits/s**
- High Traffic: **5 TBits/s**

Finally for this project has to take into consideration the table 4.1 because in it we can see the values of the variables associated with this network.

Constant	Description	Value
N	Number of nodes	6
L	Number of bidirectional links	8
<δ>	Node out-degree	2.667
<len>	Mean link length (km)	500
<h>	Mean number of hops for working paths	1.533
<h'>	Mean number of hops for backup paths	2.467

Tabela 4.1: Table of reference network values

4.1.1.2 Realistic Network

The real network chosen for this work is the EON (European Optical Network).

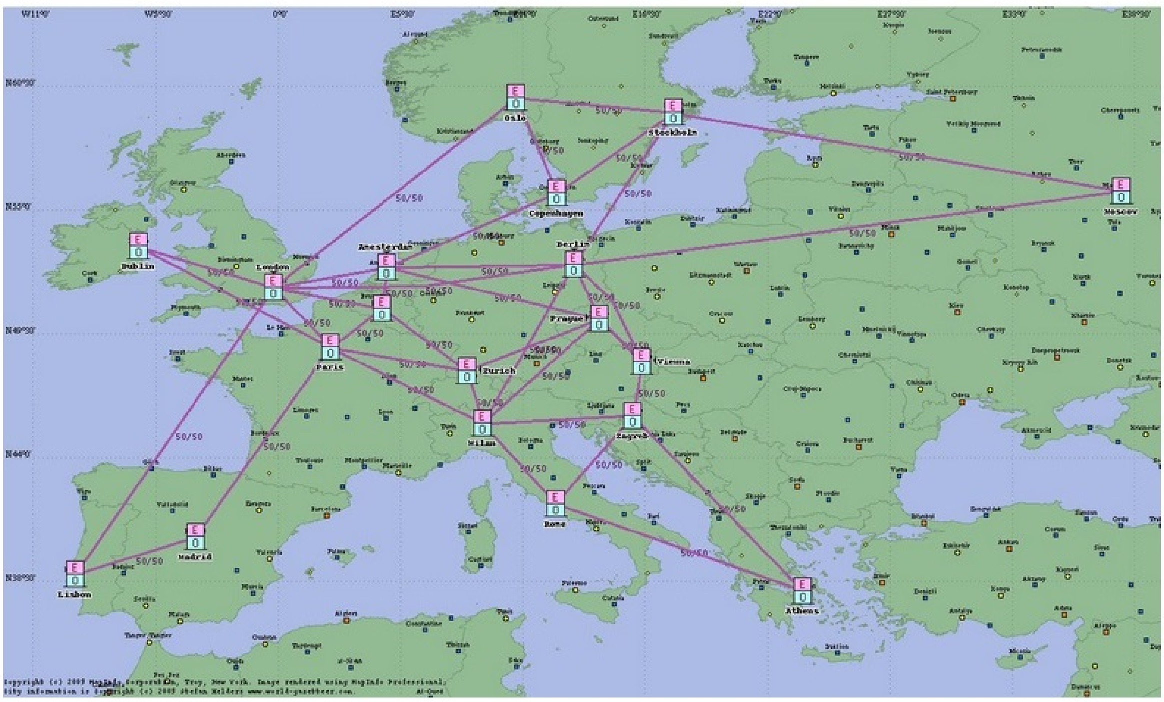


Figura 4.2: Physical topology of the realistic network.

Through the previous figure we can see how nodes are organized geographically and the distance matrix created on the next page is constructed based on real distances between them. For a better understanding of the distances matrix the table 4.2 was created to assign to each city a number of a node in the network.

City	Node
Oslo	1
Stockholm	2
Moscow	3
Copenhagen	4
Berlin	5
Prague	6
Vienna	7
Zagreb	8
Athens	9
Rome	10
Milan	11
Zurich	12
Brussels	13
Amesterdan	14
London	15
Dublin	16
Paris	17
Madrid	18
Lisbon	19

Tabela 4.2: Table of city and respective node

In this real case we have once again to take into consideration the table 4.3 because it is through it that we can see the values of the variables associated with this network.

Constant	Description	Value
N	Number of nodes	19
L	Number of bidirectional links	37
$<\delta>$	Node out-degree	3.89
<len>	Mean link length (km)	753.76
<h>	Mean number of hops for working paths	2.3
<h'>	Mean number of hops for backup paths	3.2

Tabela 4.3: Table of realistic network values

The values indicated in the distance matrix, referred to below, are expressed in kilometers (Km). For this network we must also create matrices of ODU's to determine the total traffic used in each scenario.

$$Dist = \begin{pmatrix}
0 & 417 & 0 & 484 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1155 & 0 & 0 & 0 & 0 \\
417 & 0 & 1228 & 523 & 811 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1228 & 0 & 0 & 1611 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
484 & 523 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 622 & 0 & 0 & 0 & 0 & 0 \\
0 & 811 & 1611 & 0 & 0 & 281 & 524 & 0 & 0 & 0 & 843 & 0 & 0 & 577 & 933 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 281 & 0 & 251 & 0 & 0 & 0 & 646 & 527 & 0 & 712 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 524 & 251 & 0 & 268 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 268 & 0 & 1081 & 518 & 530 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1081 & 0 & 1052 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 518 & 1052 & 0 & 477 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 843 & 646 & 0 & 530 & 0 & 477 & 0 & 219 & 0 & 0 & 0 & 0 & 640 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 646 & 0 & 0 & 0 & 0 & 219 & 0 & 493 & 0 & 0 & 0 & 488 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 493 & 0 & 173 & 321 & 0 & 264 & 0 & 0 \\
0 & 0 & 0 & 622 & 577 & 712 & 0 & 0 & 0 & 0 & 0 & 0 & 173 & 0 & 358 & 0 & 0 & 0 & 0 \\
1155 & 0 & 0 & 0 & 933 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 321 & 358 & 0 & 464 & 344 & 0 & 1587 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 464 & 0 & 782 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 640 & 488 & 264 & 0 & 344 & 782 & 0 & 1054 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1054 & 0 & 503 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1587 & 0 & 0 & 503 & 0
\end{pmatrix}$$

$$ODU0 = \begin{bmatrix}
0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}$$

$$ODU1 = \begin{bmatrix}
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}$$

$$
ODU2 = \begin{bmatrix}
0 & 1 & 1 & 2 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 2 & 2 & 2 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 2 & 0 & 2 & 2 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
2 & 2 & 2 & 0 & 2 & 2 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
2 & 2 & 2 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 2 & 2 & 2 & 0 & 0 & 2 & 0 & 2 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 2 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
$$

$$
ODU3 = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0
\end{bmatrix}
$$

$$ODU4 = \begin{bmatrix}
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0
\end{bmatrix}$$

In the traffic matrices each ODU, referred previously, has its respective value being that the ODU0 corresponds to 1.25 Gbits/s, ODU1 to 2.5 Gbits/s, ODU2 to 10 Gbits/s, ODU3 to 40 Gbits/s and finally the ODU4 corresponds to 100 Gbits/s. As we can see these matrices are bidirectional because they are symmetric matrices and as such the traffic sent in one direction must be the same traffic sent in the opposite direction.

Through these ODU's we can calculate total network traffic for the low traffic scenario:

T_1^0 = 240x1.25 = 300 Gbits/s T_1^1 = 200x2.5 = 500 Gbits/s T_1^2 = 64x10 = 640 Gbits/s

T_1^3 = 24x40 = 960 Gbits/s T_1^4 = 16x100 = 1600 Gbits/s

T_1 = 300 + 500 + 640 + 960 + 1600 = 4000 Gbits/s T = 4000/2 = 2 Tbits/s

Where the variable T_1^x represents the unidirectional traffic of the ODUx, for example, T_1^3 represents the unidirectional traffic of the ODU3 and T_1^4 represents the unidirectional traffic of the ODU4. The variable T_1 represents the total of unidirectional traffic that is injected into the network and finally the variable T represents the total of bidirectional traffic.

Again, we can thus conclude that the total traffic for the two scenarios is as follows:

- Low Traffic: **2 TBits/s**
- High Traffic: **20 TBits/s**

4.1.2 Dimensioning using ILP

Student Name	:	Tiago Esteves (October 03, 2017 -)
Goal	:	Implement the dimensioning of optical networks in the opaque transport mode.

4.1.2.1 ILP models

For a better understanding of the functions and variables used in the ILP, a table 4.4 will be created with all the variables and their description.

Description of notation used in the objective function	
i	index for start node of a physical link
j	index for end node of a physical link
o	index for node that is origin of a demand
d	index for node that is destination of a demand
(i,j)	physical link between the nodes i and j
(o,d)	demand between the nodes o and d
c	index for bit rate of the client signal
C	set of the client signal
f_{ij}^{od}	binary variable indicating if link between the nodes i and j is used in the path between nodes o and d
W_{ij}	number of optical channels between the nodes i and j
B_c	client signals granularities (1.25, 2.5, 10, 40, 100)
D_{odc}	client demands with bit rate c between nodes o and d
G	network topology in form of adjacency matrix

Tabela 4.4: Table with description of variables

The objective function of following ILP is a minimization of the sum of two variables: total number of flows crossing link (i,j) for all demand pairs (o,d) and total number of optical channels in each link (i,j).

$$minimize \sum_{(i,j)} \sum_{(o,d)} f_{ij}^{od} + \sum_{(i,j)} W_{ij} \tag{4.1}$$

$$subject\ to$$

$$\sum_{j \backslash \{o\}} f_{ij}^{od} = 2 \qquad \forall (o,d) : o < d, \forall i : i = o \tag{4.2}$$

$$\sum_{j \backslash \{o\}} f_{ij}^{od} = \sum_{j \backslash \{d\}} f_{ji}^{od} \qquad \forall (o,d) : o < d, \forall i : i \neq o, d \tag{4.3}$$

$$\sum_{j \backslash \{d\}} f_{ji}^{od} = 2 \qquad \forall (o,d) : o < d, \forall i : i = d \tag{4.4}$$

$$\sum_{(o,d):o<d} \left(f_{ij}^{od} + f_{ji}^{od} \right) + \sum_{c \in C} (B\,(c)\,D_{cod} \leq 100 W_{ij} G_{ij} \qquad \forall (i,j) : i < j \tag{4.5}$$

$$W_{ij} \leq 80 \qquad \forall (i,j) : i < j \tag{4.6}$$

$$f_{ij}^{od}, f_{ji}^{od} \in \{0, 2\} \qquad \forall (i,j) : i < j, \forall (o,d) : o < d \tag{4.7}$$

$$W_{ij} \in \mathbb{N} \qquad \forall (i,j) : i < j \tag{4.8}$$

The objective function, to be minimized, is the expression 4.1. The flow conservation constraints are 4.2, 4.3 and 4.4. First constraint ensures that, for all demand pairs (o,d), it routes two flows of traffic for all bidirectional links (i,j) when j is not equal to the origin of the demand. Equation 4.4 is based on the same idea of 4.1, however applied in reverse direction. Assuming bidirectional traffic, so the number of flows in both directions of the link is the same 4.3. The inequality 4.5 is considered grooming constraint, so it means the total client traffic flows can not be greater than the capacity of optical channels on all links. Another important constraint 4.6 is the capacity of the optical channels which must be less or equal to 100 Gb/s or 80 ODU0. The number of flows per demand can be zero if there are no traffic demands or two if considering working and protection traffic 4.7. The last constraint 4.8 is just needed to ensure the number optical of channels is a positive integer values greater than zero.

4.1.2.2 ILP Results

In this initial phase the results will be presented using ILP to calculate the CAPEX of the reference network and the realistic network. The value of the CAPEX of the network will be calculated based on the costs of the equipment present in the table below.

Equipment	Cost
OLT without transponders	15000 €
Transponder	5000 €/Gb
Optical Amplifier	4000 €
EXC	10000 €
OXC	20000 €
EXC Port	1000 €/Gb/s
OXC Port	2500 €/porto

Tabela 4.5: Table with costs

In addition to the equipment costs we will also use the parameter "span", which in this case will have a value of 100, because this value is used to calculate the number of optical amplifiers required in the network using Equation 4.9.

$$N^R = \sum_{l=1}^{L} \left(\left\lceil \frac{len_l}{span} \right\rceil - 1 \right) \tag{4.9}$$

The other parameters of this equation are:

- $N^R \rightarrow$ Total number of regenerators/amplifiers
- $len_l \rightarrow$ Length of link l
- $span \rightarrow$ Distance between amplifiers

To know the value of CAPEX it is necessary to know the value of the cost of the links and the cost of the nodes. To calculate the cost of the nodes, the sum of the costs of the optical and electrical node is made. For this case the value of the optical cost is zero only needing to know the electric cost of the nodes that is given by equation 4.10.

$$C_{exc} = (\gamma_{e0} \times N) + \gamma_{e1} \times \big(T_1 + \big(2 \times w^0 \times \tau\big)\big) \tag{4.10}$$

- $C_{exc} \rightarrow$ Electrical ports cost
- $\gamma_{e0} \rightarrow$ EXC cost in euros
- $N \rightarrow$ Number of nodes
- $\gamma_{e1} \rightarrow$ EXC port cost in euros
- $T_1 \rightarrow$ Total unidirectional traffic
- $w^0 \rightarrow$ Total number of optical channels
- $\tau \rightarrow$ Traffic per port

To calculate the cost of the Links we will use the equation 4.11.

$$C_L = \left(2 \times \gamma_0^{OLT} \times L\right) + \left(2 \times \gamma_1^{OLT} \times \tau \times W\right) + \left(N^R \times c^R\right) \tag{4.11}$$

- $C_L \rightarrow$ Links cost
- $\gamma_0^{OLT} \rightarrow$ OLT cost in euros
- $L \rightarrow$ Number of unidirectional links
- $\gamma_1^{OLT} \rightarrow$ Transponder cost in euros
- $W \rightarrow$ Total number of optical channels
- $N^R \rightarrow$ Total number of optical amplifiers
- $c^R \rightarrow$ Optical amplifiers cost in euros

To perform the calculations using the implementation of the models described in section 4.1.2.1 it is necessary to use a mathematical software tool. For this we will use MATLAB which is ideal for dealing with linear programming problems and can call the LPsolve through an external interface.

Scenario 1: Test Network Low Traffic
In this scenario we used the table 4.1. In the table 4.6 we can see the values calculated through MatLab and using the values indicated in table 4.5 we can finally calculate the CAPEX value.

Number of optical channels	Value
in the link (1,2)	2
in the link (1,3)	2
in the link (2,3)	4
in the link (2,4)	3
in the link (3,5)	3
in the link (4,5)	3
in the link (4,6)	3
in the link (5,6)	3

Tabela 4.6: Table with results

Using equation 4.11 :
C_L = (2 * 15 000 * 8) + (2 * 5 000 * 100 * 23) + (24 * 4 000)
C_L = **23 336 000€**

Using equation 4.10 :
C_{exc} = (6 * 10 000) + 1 000 * (1 000 + (2 * 23 * 100))
C_N = C_{exc} = **5 660 000€**

$CAPEX$ = 23 336 000 + 5 660 000 = **28 996 000€**

Scenario 2: Test Network High Traffic
In this scenario we used again the table 4.1. In the table 4.7 we can see the values calculated through MatLab and using the values indicated in table 4.5 we can finally calculate the CAPEX value.

Number of optical channels	Value
in the link (1,2)	12
in the link (1,3)	12
in the link (2,3)	33
in the link (2,4)	28
in the link (3,5)	28
in the link (4,5)	26
in the link (4,6)	30
in the link (5,6)	30

Tabela 4.7: Table with results

Using equation 4.11 :
C_L = (2 * 15 000 * 8) + (2 * 5 000 * 100 * 199) + (24 * 4 000)
C_L = **199 336 000€**

Using equation 4.10 :
C_{exc} = (6 * 10 000) + 1 000 * (1 000 + (2 * 199 * 100))
C_N = C_{exc} = **40 860 000€**

$CAPEX$ = 199 336 000 + 40 860 000 = **240 196 000€**

Scenario 3: Realistic Network Low Traffic
In this scenario we used the table 4.3. In the table 4.6 we can see the values calculated through MatLab and using the values indicated in table 4.21 we can finally calculate the CAPEX value.

Number of optical channels	Value
in the link (1,2)	
in the link (1,3)	
in the link (2,3)	
in the link (2,4)	
in the link (3,5)	
in the link (4,5)	
in the link (4,6)	
in the link (5,6)	

Tabela 4.8: Table with results

Using equation 4.11 :
C_L = (2 * 15 000 * 37) + (2 * 5 000 * 100 *) + (24 * 4 000)
C_L = €

Using equation 4.10 :
C_{exc} = (19 * 10 000) + 1 000 * (1 000 + (2 * * 100))
C_N = C_{exc} = €

$CAPEX$ = + = €

Scenario 4: Realistic Network High Traffic
In this scenario we used again the table 4.3. In the table 4.9 we can see the values calculated through MatLab and using the values indicated in table 4.5 we can finally calculate the CAPEX value.

Number of optical channels	Value
in the link (1,2)	
in the link (1,3)	
in the link (2,3)	
in the link (2,4)	
in the link (3,5)	
in the link (4,5)	
in the link (4,6)	
in the link (5,6)	

Tabela 4.9: Table with results

Using equation 4.11 :
C_L = (2 * 15 000 * 37) + (2 * 5 000 * 100 *) + (24 * 4 000)

C_L = €

Using equation 4.10 :
C_{exc} = (19 * 10 000) + 1 000 * (1 000 + (2 * 100 *))
C_N = C_{exc} = €

$CAPEX$ = + = €

4.1.3 Dimensioning using Heuristics

4.1.3.1 Heuristics Models

4.1.3.2 Heuristics Results

4.1.4 Comparative Analysis

4.2 Transparent with 1+1 Protection

In this case study we focus on the transparent case with 1 + 1 protection. In this mode of transport, the information travels in a route defined through optical channels between origin and destination nodes always in the optical domain and, consequently, physical topology and logical topology are different. An advantage of this mode of transport is the possibility of transporting express traffic. An disadvantage is that the capacity utilization of the optical channels is worse than in the opaque mode of transport due to grooming only customer signs with the same endpoints.

4.2.1 Physical Network Topology

Student Name	:	Tiago Esteves (October 03, 2017 -)
Goal	:	Implement the dimensioning of optical networks in the transparent transport mode.

4.2.1.1 Reference Network

In the figure below we ca see that our reference network consists of 6 nodes and 8 Bidirectional links. The matrix of distances between the respective nodes and the ODU's matrices are the same as those reported in the opaque transport mode.

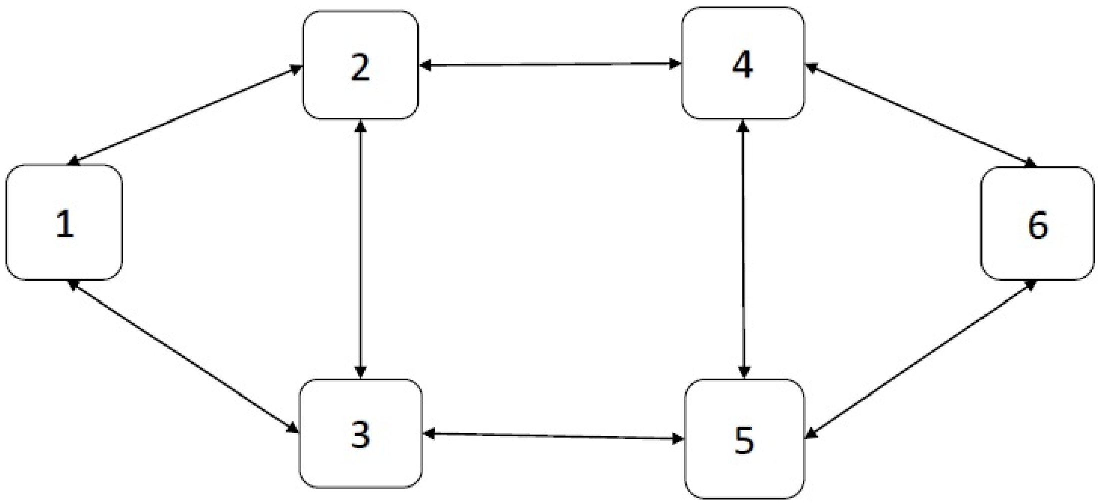


Figura 4.3: Physical Topology of the Reference Network.

The distance matrix is the same for the two scenarios but the ODU's matrices are not. In this case only the matrices for the case of high traffic are elucidated, being that in the case of a low traffic it is only necessary to divide these matrices by the value 10.

$$Dist = \begin{bmatrix} 0 & 500 & 500 & 0 & 0 & 0 \\ 500 & 0 & 400 & 500 & 0 & 0 \\ 500 & 400 & 0 & 0 & 500 & 0 \\ 0 & 500 & 0 & 0 & 600 & 450 \\ 0 & 0 & 500 & 600 & 0 & 550 \\ 0 & 0 & 0 & 450 & 550 & 0 \end{bmatrix} \quad ODU0 = \begin{bmatrix} 0 & 50 & 10 & 30 & 10 & 30 \\ 50 & 0 & 0 & 10 & 50 & 0 \\ 10 & 0 & 0 & 10 & 40 & 10 \\ 30 & 10 & 10 & 0 & 10 & 0 \\ 10 & 50 & 40 & 10 & 0 & 30 \\ 30 & 0 & 10 & 10 & 30 & 0 \end{bmatrix}$$

$$ODU1 = \begin{bmatrix} 0 & 20 & 40 & 20 & 0 & 50 \\ 20 & 0 & 0 & 30 & 10 & 10 \\ 40 & 0 & 0 & 10 & 10 & 0 \\ 30 & 30 & 10 & 0 & 10 & 30 \\ 0 & 10 & 10 & 10 & 0 & 10 \\ 50 & 10 & 0 & 30 & 10 & 0 \end{bmatrix} \quad ODU2 = \begin{bmatrix} 0 & 10 & 10 & 10 & 0 & 0 \\ 10 & 0 & 0 & 0 & 10 & 0 \\ 10 & 0 & 0 & 10 & 10 & 0 \\ 10 & 0 & 10 & 0 & 10 & 0 \\ 0 & 10 & 10 & 10 & 0 & 10 \\ 0 & 0 & 0 & 0 & 10 & 0 \end{bmatrix}$$

$$ODU3 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 10 & 0 & 0 & 10 \\ 0 & 10 & 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 10 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 & 0 & 0 \end{bmatrix} \quad ODU4 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 10 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 10 \\ 0 & 10 & 0 & 0 & 10 & 0 \end{bmatrix}$$

The values indicated in the distance matrix, referred to below, are expressed in kilometers (Km) and as it couldn't be otherwise, this matrix is symmetric. In relation to the traffic matrices each ODU, referred previously, has its respective value being that the ODU0 corresponds to 1.25 Gbits/s, ODU1 to 2.5 Gbits/s, ODU2 to 10 Gbits/s, ODU3 to 40 Gbits/s and finally the ODU4 corresponds to 100 Gbits/s. As we can see these matrices are bidirectional because they are symmetric matrices and as such the traffic sent in one direction must be the same traffic sent in the opposite direction.

Through these ODU's we can calculate total network traffic for the low traffic scenario:

T_1^0 = 600x1.25 = 750 Gbits/s T_1^1 = 500x2.5 = 1250 Gbits/s T_1^2 = 160x10 = 1600 Gbits/s

T_1^3 = 60x40 = 2400 Gbits/s T_1^4 = 40x100 = 4000 Gbits/s

T_{1total} = 750 + 1250 + 1600 + 2400 + 4000 = 10000 Gbits/s T_{total} = 10000/2 = 5 Tbits/s

We can thus conclude that the total traffic for the two scenarios is as follows:

- Low Traffic: **0.5 TBits/s**
- High Traffic: **5 TBits/s**

Finally for this project has to take into consideration the table 4.10 because in it we can see the values of the variables associated with this network.

Constant	Description	Value
N	Number of nodes	6
L	Number of bidirectional links	8
$<\delta>$	Node out-degree	2.667
<len>	Mean link length (km)	500
<h>	Mean number of hops for working paths	1.533
<h'>	Mean number of hops for backup paths	2.467

Tabela 4.10: Table of reference network values

4.2.1.2 Realistic Network

The real network chosen for this work is the EON (European Optical Network). The way the nodes are arranged geographically can be seen from the following figure and the matrix of distances created in the next page is constructed based on real distances. In this case just ODU's matrices are created to be able to determine the total traffic used in each scenario.

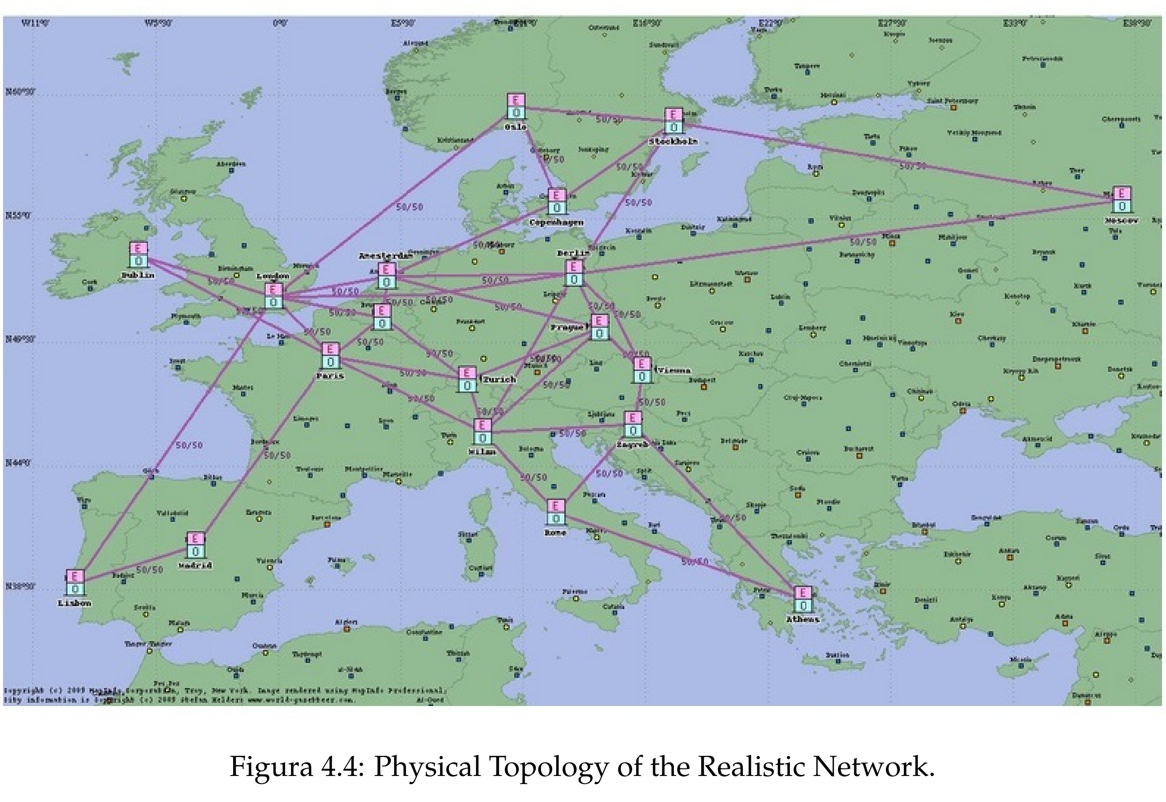


Figura 4.4: Physical Topology of the Realistic Network.

The table 4.11 shows the values of the variables associated with this network.

Again, through the ODU's we can calculate the total traffic for both scenarios being them:

Constant	Description	Value
N	Number of nodes	19
L	Number of bidirectional links	37
$<\delta>$	Node out-degree	3.89
<len>	Mean link length (km)	753.76
<h>	Mean number of hops for working paths	2.3
<h′>	Mean number of hops for backup paths	3.2

Tabela 4.11: Table of realistic network values

- Low Traffic: **2 TBits/s**
- High Traffic: **20 TBits/s**

Dist =

	Oslo	*Stockholm*	*Moscow*	*Copenhagen*	*Berlin*	*Prague*	*Vienna*	*Zagreb*	*Athens*	*Rome*	*Mil*
Oslo	0	500	500	0	0	0	0	500	500	0	0
Stockholm	0	500	500	0	0	0	0	500	500	0	0
Moscow	0	500	500	0	0	0	0	500	500	0	0
Copenhagen	0	500	500	0	0	0	0	500	500	0	0
Berlin	0	500	500	0	0	0	0	500	500	0	0
Prague	0	500	500	0	0	0	0	500	500	0	0
Vienna	0	500	500	0	0	0	0	500	500	0	0
Zagreb	0	500	500	0	0	0	0	500	500	0	0
Athens	0	500	500	0	0	0	0	500	500	0	0
Rome	0	500	500	0	0	0	0	500	500	0	0
Milan	0	500	500	0	0	0	0	500	500	0	0
Zurich	0	500	500	0	0	0	0	500	500	0	0
Brussels	0	500	500	0	0	0	0	500	500	0	0
Amesterdan	0	500	500	0	0	0	0	500	500	0	0
London	0	500	500	0	0	0	0	500	500	0	0
Dublin	0	500	500	0	0	0	0	500	500	0	0
Paris	0	500	500	0	0	0	0	500	500	0	0
Madrid	0	500	500	0	0	0	0	500	500	0	0
Lisbon	0	500	500	0	0	0	0	500	500	0	0

4.2.2 Dimensioning using ILP

Student Name	:	Tiago Esteves (October 03, 2017 -)
Goal	:	Implement the dimensioning of optical networks in the transparent transport mode.

4.2.2.1 ILP Models

Again, for a better understanding of the functions and variables used in the ILP, a table 4.12 will be created with all the variables and their description.

Description of notation used in the objective function	
i	index for start node of a physical link
j	index for end node of a physical link
o	index for node that is origin of a demand
d	index for node that is destination of a demand
(i,j)	physical link between the nodes i and j
(o,d)	demand between the nodes o and d
f_{ij}^{od}	Number of 100 Gbit/s optical channels (number of flows) between the link i and j for all demand pairs between o and d
W_{od}	number of optical channels between the nodes o and d
G	Network topology in form of adjacency matrix

Tabela 4.12: Table with description of variables

The optimization model suggested for transparent transport mode with dedicated path protection intends to minimize the total number of flows crossing link (i, j) for all demand pairs (o, d). The mathematical model described below also minimizes the total number of optical channels between each demand end nodes W_{od}, instead of minimizing the number of optical link-by-link channels as in the previous model.

$$minimize \sum_{(i,j)} \sum_{(o,d)} f_{ij}^{od} + \sum_{(o,d)} W_{od} \tag{4.12}$$

$subject\ to$

$$\sum_{j \backslash \{o\}} f_{ij}^{od} = 2 \qquad \forall (o,d) : o < d, \forall i : i = o \tag{4.13}$$

$$\sum_{j \backslash \{o\}} f_{ij}^{od} = \sum_{j \backslash \{d\}} f_{ji}^{od} \qquad \forall (o,d) : o < d, \forall i : i \neq o, d \tag{4.14}$$

$$\sum_{j\backslash\{d\}} f_{ji}^{od} = 2 \qquad \forall(o,d) : o < d, \forall i : i = d \tag{4.15}$$

$$\sum_{(o,d):o<d} \left(f_{ij}^{od} + f_{ji}^{od}\right) W_{od} \leq 80G_{ij} \qquad \forall(i,j) : i < j \tag{4.16}$$

$$f_{ij}^{od}, f_{ji}^{od} \in \{0, 2\} \qquad \forall(i,j) : i < j, \forall(o,d) : o < d \tag{4.17}$$

$$W_{od} \in \mathbb{N} \qquad \forall(o,d) : o < d \tag{4.18}$$

The objective function, to be minimized, is the expression 4.12. The flow conservation is performed by equations 4.13, 4.14 and 4.15 and share the same mathematical description of opaque model. The inequality 4.16 answers capacity constraint problem. Then, total flows times the traffic of the demands must be less or equal to the capacity of network links. The grooming of this model can be done before routing since the traffic is aggregated just for demands between the same nodes, thus not depending on the routes. Last two constraints define the total number of flows must be zero if there is no demand, or two for a demand with traffic protection, and the number of optical channels must be a counting number.

4.2.2.2 ILP Results

In this initial phase the results will be presented using ILP to calculate the CAPEX of the reference network and the realistic network. The value of the CAPEX of the network will be calculated based on the costs of the equipment present in the table below.

Equipment	Cost
OLT without transponders	15000 €
Transponder	5000 €/Gb
Optical Amplifier	4000 €
EXC	10000 €
OXC	20000 €
EXC Port	1000 €/Gb/s
OXC Port	2500 €/porto

Tabela 4.13: Table with costs

In addition to the equipment costs we will also use the parameter "span", which in this case will have a value of 100, because this value is used to calculate the number of optical amplifiers required in the network using Equation 4.19.

$$N^R = \sum_{l=1}^{L} \left(\left\lceil \frac{len_l}{span} \right\rceil - 1 \right) \tag{4.19}$$

To know the value of CAPEX it is necessary to know the value of the cost of the links and the cost of the nodes.

To calculate the cost of the nodes, the sum of the costs of the optical and electrical node is made. For this case the optical cost is given by equation 4.20 and the electrical cost by the equation 4.21.

$$C_{oxc} = (\gamma_{o0} \times N) + \gamma_{o1} \times (P_{LINE} + P_{ADD}) \tag{4.20}$$

- $C_{oxc} \rightarrow$ Optical Ports Cost
- $\gamma_{o0} \rightarrow$ OXC cost in Euros
- $\gamma_{o1} \rightarrow$ OXC port cost in Euros
- $P_{TRIB} \rightarrow$ Number of tributary ports
- $P_{ADD} \rightarrow$ Number of adding ports

$$C_{exc} = (\gamma_{e0} \times N) + \gamma_{e1} \times (2 \times T_1) \tag{4.21}$$

To calculate the cost of the Links we will use the equation 4.22.

$$C_L = \left(2 \times \gamma_0^{OLT} \times L\right) + \left(2 \times \gamma_1^{OLT} \times \tau \times W\right) + \left(N^R \times c^R\right) \tag{4.22}$$

To perform the calculations using the implementation of the models described in section 4.1.2.1 it is necessary to use a mathematical software tool. For this we will use MATLAB which is ideal for dealing with linear programming problems and can call the LPsolve through an external interface.

Scenario 1: Test Network Low Traffic
In this scenario we used the table 4.10. In the table 4.14 we can see the values calculated through MatLab and using the values indicated in table 4.13 we can finally calculate the CAPEX value.

Number of optical channels	Value	ADD PORTS	Value	LINE PORTS	Value
in the link (1,2)	3	Node 1	5	Node 1	5
in the link (1,3)	2	Node 2	6	Node 2	12
in the link (2,3)	4	Node 3	5	Node 3	11
in the link (2,4)	5	Node 4	5	Node 4	11
in the link (3,5)	5	Node 5	6	Node 5	10
in the link (4,5)	2	Node 6	7	Node 6	7
in the link (4,6)	4				
in the link (5,6)	3				

Tabela 4.14: Table with results

Using equation 4.22 :
C_L = (2 * 15 000 * 8) + (2 * 5 000 * 100 * 28) + (24 * 4 000)
C_L = **28 336 000 €**

Using equation 4.21 :
C_{exc} = (6 * 10 000) + 1 000 * (2 * 1 000)
C_{exc} = **2 060 000€**

Using equation 4.20 :
C_{oxc} = (6 * 10 000) + 1 000 * (34 + 56)
C_{oxc} = **150 000 €**
C_N = C_{oxc} + C_{exc} = **2 210 000 €**

$CAPEX$ = 28 336 000 + 2 210 000 = **30 546 000 €**

Scenario 2: Test Network High Traffic
In this scenario we used again the table 4.10 In the table 4.15 we can see the values calculated through MatLab and using the values indicated in table 4.13 we can finally calculate the CAPEX value.

Number of optical channels	Value	ADD PORTS	Value	LINE PORTS	Value
in the link (1,2)	3	Node 1	5	Node 1	5
in the link (1,3)	2	Node 2	6	Node 2	12
in the link (2,3)	4	Node 3	5	Node 3	11
in the link (2,4)	5	Node 4	5	Node 4	11
in the link (3,5)	5	Node 5	6	Node 5	10
in the link (4,5)	2	Node 6	7	Node 6	7
in the link (4,6)	4				
in the link (5,6)	3				

Tabela 4.15: Table with results

Using equation 4.22 :
C_L = (2 * 15 000 * 8) + (2 * 5 000 * 100 *) + (24 * 4 000)
C_L = **€**

Using equation 4.21 :
C_{exc} = (6 * 10 000) + 1 000 * (2 * 2 000)
C_{exc} = **4 060 000 €**

Using equation 4.20 :
C_{oxc} = (6 * 10 000) + 1 000 * (+)
C_{oxc} = **€**
C_N = C_{oxc} + C_{exc} = **€**

$CAPEX$ = + = **€**

Scenario 3: Realistic Network Low Traffic
In this scenario we used the table 4.11 In the table 4.16 we can see the values calculated through MatLab and using the values indicated in table 4.13 we can finally calculate the CAPEX value.

Number of optical channels	Value	ADD PORTS	Value	LINE PORTS	Value
in the link (1,2)	3	Node 1	5	Node 1	5
in the link (1,3)	2	Node 2	6	Node 2	12
in the link (2,3)	4	Node 3	5	Node 3	11
in the link (2,4)	5	Node 4	5	Node 4	11
in the link (3,5)	5	Node 5	6	Node 5	10
in the link (4,5)	2	Node 6	7	Node 6	7
in the link (4,6)	4				
in the link (5,6)	3				

Tabela 4.16: Table with results

Using equation 4.22 :
C_L = (2 * 15 000 * 37) + (2 * 5 000 * 100 *) + (24 * 4 000)
C_L = **€**

Using equation 4.21 :
C_{exc} = (19 * 10 000) + 1 000 * (2 * 4 000)
C_{exc} = **8 190 000€**

Using equation 4.20 :

C_{oxc} = (19 * 10 000) + 1 000 * (+)
C_{oxc} = **€**
C_N = C_{oxc} + C_{exc} = **€**

$CAPEX$ = + = **€**

Scenario 4: Realistic Network High Traffic
In this scenario we used again the table 4.11 In the table 4.17 we can see the values calculated through MatLab and using the values indicated in table 4.13 we can finally calculate the CAPEX value.

Number of optical channels	Value
in the link (1,2)	3
in the link (1,3)	2
in the link (2,3)	4
in the link (2,4)	5
in the link (3,5)	5
in the link (4,5)	2
in the link (4,6)	4
in the link (5,6)	3

ADD PORTS	Value
Node 1	5
Node 2	6
Node 3	5
Node 4	5
Node 5	6
Node 6	7

LINE PORTS	Value
Node 1	5
Node 2	12
Node 3	11
Node 4	11
Node 5	10
Node 6	7

Tabela 4.17: Table with results

Using equation 4.22 :
C_L = (2 * 15 000 * 37) + (2 * 5 000 * 100 *) + (24 * 4 000)
C_L = **€**

Using equation 4.21 :
C_{exc} = (19 * 10 000) + 1 000 * (2 * 40 000)
C_{exc} = **80 190 000€**

Using equation 4.20 :
C_{oxc} = (19 * 10 000) + 1 000 * (+)
C_{oxc} = **€**

$C_N = C_{oxc} + C_{exc} =$ €

$$CAPEX = + = \text{€}$$

4.2.3 Dimensioning using Heuristics

4.2.3.1 Heuristics Models

4.2.3.2 Heuristics Results

4.2.4 Comparative Analysis

Student Name	:	Tiago Esteves
Starting Date	:	October 03, 2017
Goal	:	Implement the dimensioning of optical networks in the translucent transport mode.

4.3 Translucent with 1+1 Protection

In this case study we focus on the translucent case with 1 + 1 protection.

4.3.1 Physical Network Topology

4.3.1.1 Reference Network

In the figure below we ca see that our reference network consists of 6 nodes and 8 Bidirectional links. The average length of the links was chosen so that the following calculations are more simplistic, for this was created a matrix of distances between the respective nodes. Finally, ODU's matrices were also created to be able to determine the total traffic used in each scenario.

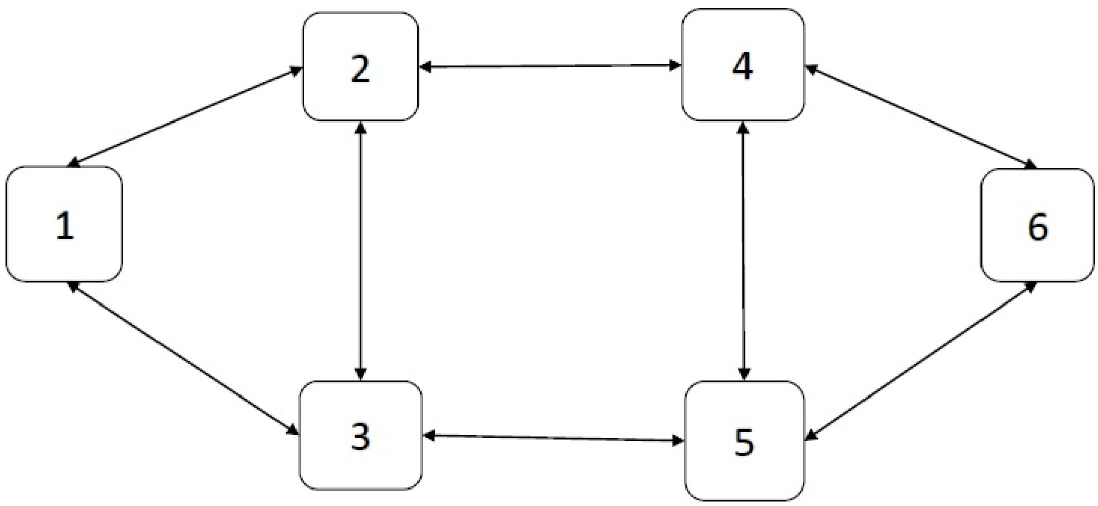


Figura 4.5: Physical Topology of the Reference Network.

The distance matrix is the same for the two scenarios but the ODU's matrices are not. In this way only the matrices for the case of low traffic are elucidated, being that in the case of a high traffic it is only necessary to multiply these matrices by the value 10.

$$Dist = \begin{bmatrix} 0 & 500 & 500 & 0 & 0 & 0 \\ 500 & 0 & 400 & 500 & 0 & 0 \\ 500 & 400 & 0 & 0 & 500 & 0 \\ 0 & 500 & 0 & 0 & 600 & 450 \\ 0 & 0 & 500 & 600 & 0 & 550 \\ 0 & 0 & 0 & 450 & 550 & 0 \end{bmatrix}$$

$$ODU0 = \begin{bmatrix} 0 & 5 & 1 & 3 & 1 & 3 \\ 5 & 0 & 0 & 1 & 5 & 0 \\ 1 & 0 & 0 & 1 & 4 & 1 \\ 3 & 1 & 1 & 0 & 1 & 0 \\ 1 & 5 & 4 & 1 & 0 & 3 \\ 3 & 0 & 1 & 1 & 3 & 0 \end{bmatrix} \quad ODU1 = \begin{bmatrix} 0 & 2 & 4 & 2 & 0 & 5 \\ 2 & 0 & 0 & 3 & 1 & 1 \\ 4 & 0 & 0 & 1 & 1 & 0 \\ 3 & 3 & 1 & 0 & 1 & 3 \\ 0 & 1 & 1 & 1 & 0 & 1 \\ 5 & 1 & 0 & 3 & 1 & 0 \end{bmatrix} \quad ODU2 = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

$$ODU3 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \quad ODU4 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Through these ODU's we can calculate total network traffic for the low traffic scenario:
T_1^0 = 60x1.25 = 75 Gbits/s T_1^1 = 50x2.5 = 125 Gbits/s T_1^2 = 16x10 = 160 Gbits/s
T_1^3 = 6x40 = 240 Gbits/s T_1^4 = 4x100 = 400 Gbits/s
T_{1total} = 75 + 125 + 160 + 240 + 400 = 1000 Gbits/s T_{total} = 1000/2 = 0.5 Tbits/s

We can thus conclude that the total traffic for the two scenarios is as follows:

- Low Traffic: **0.5 TBits/s**
- High Traffic: **5 TBits/s**

Finally for this project has to take into consideration the table 4.18 because in it we can see the values of the variables associated with this network.

Constant	Description	Value
N	Number of nodes	6
L	Number of bidirectional links	8
<δ>	Node out-degree	2.667
<len>	Mean link length (km)	500
<h>	Mean number of hops for working paths	1.533
<h'>	Mean number of hops for backup paths	2.467

Tabela 4.18: Table of reference network values

4.3.1.2 Realistic Network

The real network chosen for this work is the EON (European Optical Network). The way the nodes are arranged geographically can be seen from the following figure and the matrix of distances created in the next page is constructed based on real distances. In this case just ODU's matrices are created to be able to determine the total traffic used in each scenario.

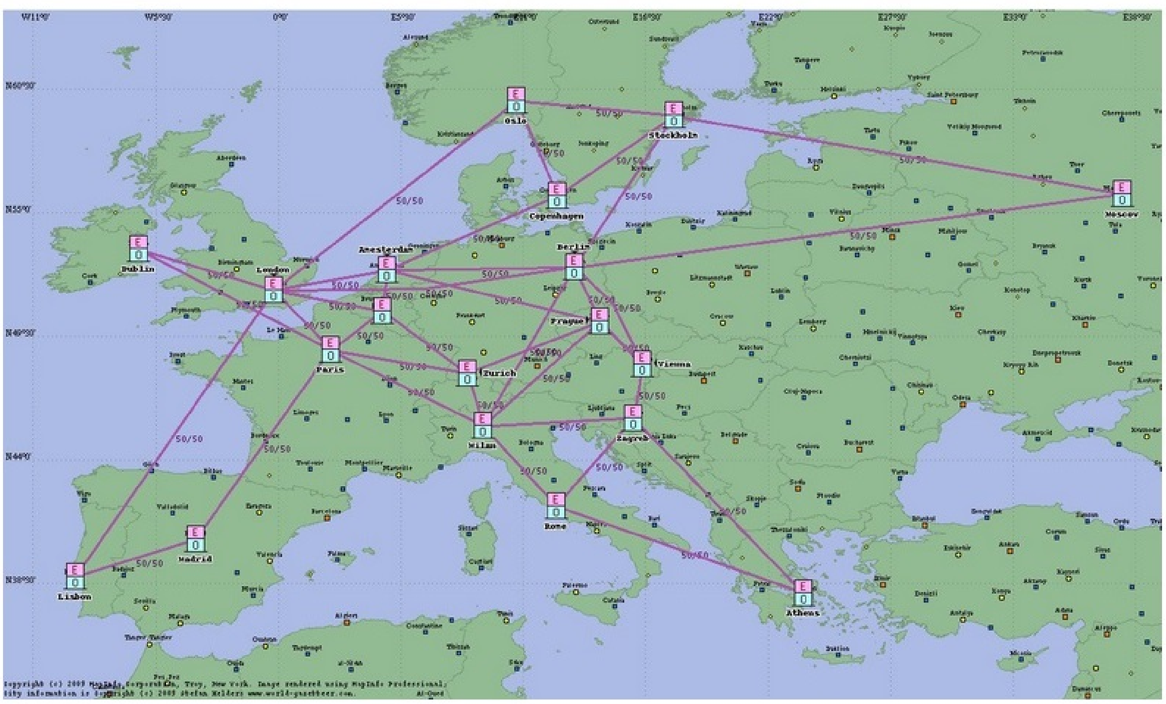


Figura 4.6: Physical Topology of the Realistic Network.

The table 4.19 shows the values of the variables associated with this network.

Constant	Description	Value
N	Number of nodes	19
L	Number of bidirectional links	37
$<\delta>$	Node out-degree	3.89
<len>	Mean link length (km)	753.76
<h>	Mean number of hops for working paths	2.3
<h'>	Mean number of hops for backup paths	3.2

Tabela 4.19: Table of realistic network values

Again, through the ODU's we can calculate the total traffic for both scenarios being them:

- Low Traffic: **2 TBits/s**
- High Traffic: **20 TBits/s**

Dist =

	Oslo	Stockholm	Moscow	Copenhagen	Berlin	Prague	Vienna	Zagreb	Athens	Rome	Mil
Oslo	0	500	500	0	0	0	0	500	500	0	0
Stockholm	0	500	500	0	0	0	0	500	500	0	0
Moscow	0	500	500	0	0	0	0	500	500	0	0
Copenhagen	0	500	500	0	0	0	0	500	500	0	0
Berlin	0	500	500	0	0	0	0	500	500	0	0
Prague	0	500	500	0	0	0	0	500	500	0	0
Vienna	0	500	500	0	0	0	0	500	500	0	0
Zagreb	0	500	500	0	0	0	0	500	500	0	0
Athens	0	500	500	0	0	0	0	500	500	0	0
Rome	0	500	500	0	0	0	0	500	500	0	0
Milan	0	500	500	0	0	0	0	500	500	0	0
Zurich	0	500	500	0	0	0	0	500	500	0	0
Brussels	0	500	500	0	0	0	0	500	500	0	0
Amesterdan	0	500	500	0	0	0	0	500	500	0	0
London	0	500	500	0	0	0	0	500	500	0	0
Dublin	0	500	500	0	0	0	0	500	500	0	0
Paris	0	500	500	0	0	0	0	500	500	0	0
Madrid	0	500	500	0	0	0	0	500	500	0	0
Lisbon	0	500	500	0	0	0	0	500	500	0	0

4.3.2 Dimensioning using ILP

4.3.2.1 ILP models

Again, for a better understanding of the functions and variables used in the ILP, a table 4.20 will be created with all the variables and their description.

Description of notation used in the objective function	
i	index for start node of a physical link
j	index for end node of a physical link
o	index for node that is origin of a demand
d	index for node that is destination of a demand
(i,j)	physical link between the nodes i and j
(o,d)	demand between the nodes o and d
c	Client traffic Type (1 to 5)
L_{ij}^{od}	Number of ODU-o low speed signals from node o to node d employing lightpath (i,j)
f_{ij}^{od}	Number of 100 Gbit/s optical channels (number of flows) between the link i and j for all demand pairs between o and d
W_{od}	Number of lightpath channels between the nodes o and d
B	Client signals granularities (1.25, 2.5, 10, 40, 100)
D_{od}	Client traffic demands between the nodes o and d
G	Network topology in form of adjacency matrix
BD	Bandwidth

Tabela 4.20: Table with description of variables

$$minimize \qquad \sum_{(o,d)} W_{od} \tag{4.23}$$

$subject\ to$

$$\sum_{j\backslash\{o\}} L_{ij}^{od} = D_{odc} \qquad \forall(o,d) : o < d \tag{4.24}$$

$$\sum_{j\backslash\{o\}} L_{ji}^{od} = \sum_{j\backslash\{d\}} L_{ji}^{od} \qquad \forall(o,d) : o < d, \forall i : i \neq o, d \tag{4.25}$$

$$\sum_{j\backslash\{d\}} L_{ij}^{od} = D_{odc} \qquad \forall(o,d) : o < d \tag{4.26}$$

$$\sum_{(o,d):o<d} \left(B(c) \times L_{ij}^{od} \right) \leq \sum BD \times W_{ij} \qquad \forall (i,j) \tag{4.27}$$

$$L_{ij}^{od} \geq 0; \qquad \forall (i,j), \forall (o,d) : o < d \tag{4.28}$$

$$\sum_{j \setminus \{o\}} f_{ji}^{od} = W_{od} \qquad \forall (o,d) : o < d \tag{4.29}$$

$$\sum_{j \setminus \{o\}} f_{ji}^{od} = \sum_{j \setminus \{d\}} f_{ji}^{od} \qquad \forall (o,d) : o < d, \forall i : i \neq o, d \tag{4.30}$$

$$\sum_{j \setminus \{d\}} f_{ji}^{od} = W_{od} \qquad \forall (o,d) : o < d \tag{4.31}$$

$$\sum_{(o,d):o<d} \left(f_{ij}^{od} + f_{ji}^{od} \right) \leq 80 G_{ij} \qquad \forall (i,j) : i < j \tag{4.32}$$

$$f_{ij}^{od} \geq 0 \qquad \forall (i,j) \forall (o,d) \tag{4.33}$$

4.3.2.2 ILP Results

In this initial phase the results will be presented using ILP to calculate the CAPEX of the reference network.

The value of the CAPEX of the network will be calculated based on the costs of the equipment present in the table below.

Equipment	Cost
OLT without transponders	15000 €
Transponder	5000 €/Gb
Optical Amplifier	4000 €
EXC	10000 €
OXC	20000 €
EXC Port	1000 €/Gb/s
OXC Port	2500 €/porto

Tabela 4.21: Table with costs

In addition to the equipment costs, we will also use the parameter "span", which in this case will have a value of 100. Because this value is used to calculate the number of optical amplifiers required in the network using Equation 4.34.

$$N^R = \sum_{l=1}^{L} \left(\left\lceil \frac{len_l}{span} \right\rceil - 1 \right) \tag{4.34}$$

To know the value of CAPEX it is necessary to know the value of the cost of the links and the cost of the nodes.

To calculate the cost of the nodes, the sum of the costs of the optical and electrical node is made.

To calculate the cost of the Links we will use the equation .

Finally we will calculate the CAPEX values for the various situations mentioned.

Low Traffic scenario:

C_L = €
C_N = €
$CAPEX$ = €

High Traffic scenario:

C_L = €
C_N = €
$CAPEX$ = €

4.3.3 Dimensioning using Analytical solution

4.3.3.1 Analytical Models

4.3.3.2 Analytical Results

4.3.4 Dimensioning using Heuristics

4.3.4.1 Heuristics Models

4.3.4.2 Heuristics Results

4.3.5 Comparative Analysis

Capítulo 5

Appendices

5.1 Installing Net2Plan and its main options

This first section will describe how to install Net2Plan and some of the solvers usable by it as well as the main program preferences available.

5.1.1 Net2Plan download and installation

Before downloading Net2Plan, the first step is verifying if the computer has the necessary Java Runtime Environment, it is recommended the latest release (Version 8). This can be download from the java website at https://java.com/en/download/. The Java Runtime Environment is necessary as Net2Plan was coded in Java.

Having installed the Java Environment it is now possible to install Net2Plan. The download is available on its website at http://net2plan.com/download.php. The files just need to be extracted and the program can be run without an installation by just double clicking on the file "Net2Plan.jar". The latest Net2Plan version available at the time this report was revised is 0.4.2 from July 22nd, 2016

Figura 5.1: Net2Plan Opening Menu

5.1.2 Net2Plan Options and installing solvers

To access the main Net2Plan options click "File → Options". In this window the global parameters for simulations can be changed if needed. For example, an important option to note in this tab is the parameter "defaultRunnableCodePath", whose value should be the path to the jar file containing NetPlanner algorithms. As will be explained further on, Net2Plan is an open source tool and as such, new algorithms can be implemented and the default path can be changed to the path where those will be available instead of loading them manually each time Net2Plan is opened. The remaining parameters are related to solver options, which are the default external solvers used and also the path in which the "$.dll$", "$.so$", "$.dylib$" files of each solver are available. By default there is no path for each solver but in this case it was already changed to where the solvers were installed.

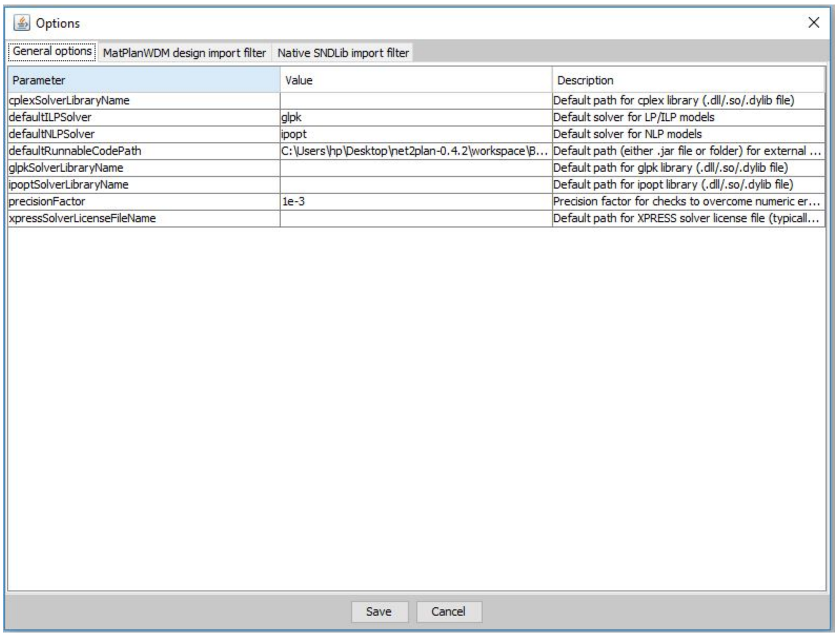


Figura 5.2: Net2Plan General Options

These external solvers are not extracted along with Net2Plan and as such, need to be downloaded if needed for the algorithms to be used. As "cplex"is a paid application, only the other two solvers will be shown as the process is similar.

The "IPOPT"solver can be downloaded from http://www.coin-or.org/download/source/Ipopt/. There are various choices available to download but for this case the *.dll* is the main file needed. An example of an algorithm which uses this solver is shown on Figure 5.3. Note that the "solverLibraryName"has the path shown earlier on the "Solver options"tab, this would have to be added manually if not introduced into the main options. The other free solver also used by some Net2Plan is "glpk", this one can be downloaded from http://sourceforge.net/projects/winglpk/?source=typ_redirect. An example is shown on Figure 5.4. Again note the path shows up as in the options.

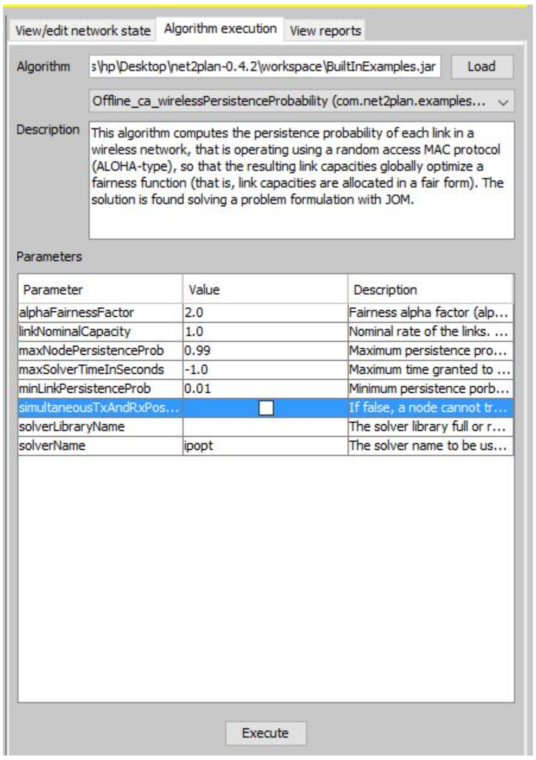


Figura 5.3: Net2Plan Algorithm with *ipopt* solver

5.2 Net2Plan Tools

This section will describe in some detail the tools presented in Net2Plan as a network planner, most notably how to created a traffic matrix, design a network and some of the simulation options available.

5.2.1 Creating Traffic Matrices

To start creating a traffic matrix in Net2Plan go to "Tools $\rightarrow$ Traffic matrix design"or press $Alt + 2$. The traffic matrix menu is shown on Figure 5.5.

On the top left side a traffic pattern can be chosen for one matrix or several if used the "Apply batch"option.

- "Constant"has two parameters the number of nodes and a constant value. This creates an uniform matrix with the number of nodes chosen and traffic equal to the value selected.
- "Uniform (0,10)"has the number of nodes and the option of being symmetric as the parameters. The matrix then has the number of nodes introduced and an amount of traffic chosen randomly between 0 and 10 which can be symmetric or not depending on the choice done.

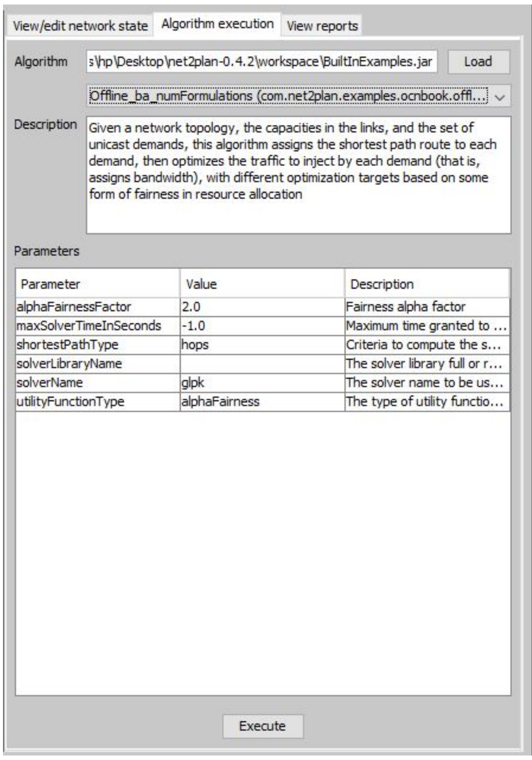


Figura 5.4: Net2Plan Algorithm with $glpk$ solver

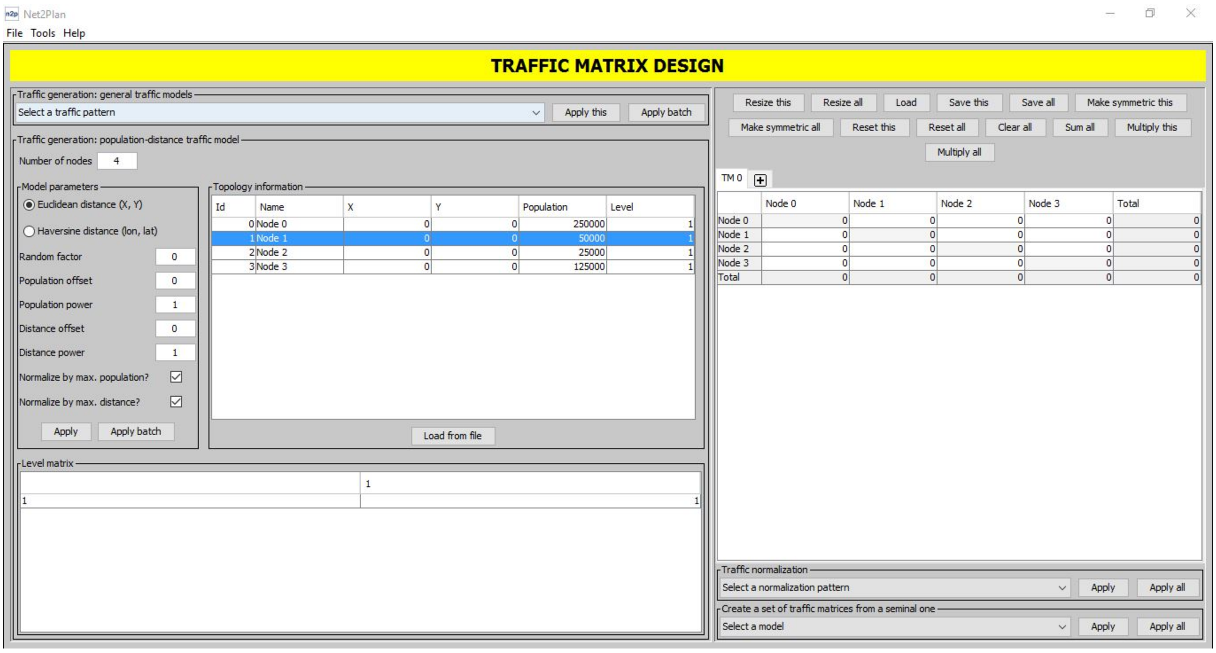


Figura 5.5: Net2Plan Traffic Matrix Design

- "Uniform (0,100)"is very similar to the other uniform option whereas in this case the traffic values are chosen randomly between 0 and 100.

- "50% Uniform (0,100) & 50% Uniform (0,10)"and "25% Uniform (0,100) & 75% Uniform (0,10)"are as expected a mixture of the previous two options.
- "Gravity model"in this option a number of nodes is chosen as well as the amount of traffic both generated and received by each node. The sum of the traffic generated by all the nodes needs to be equal to the sum of the traffic received by them.

Below the traffic pattern options, an existing model can be loaded and additional parameters defined such as Population and Node Level.

On the right side a traffic matrix can be created manually by defining the number of nodes on "resize this"and the amount of traffic can be typed on each demand. The other options above the matrix are self explanatory, for example, "multiply this"multiplies all the traffic by a constant number chosen. A point to note is that most options has an "all"choice as it is possible to have more then one matrix created.

Below the matrix part are two further options available for the matrices, the first one is the option to select a normalization pattern such as "Total normalization"where a total number of traffic can be chosen for the network and the demands are adapted to it accordingly. The other option is to create a set of matrices based on the designed one.

Figure 5.6 shows how to create batch of matrices with constant traffic.

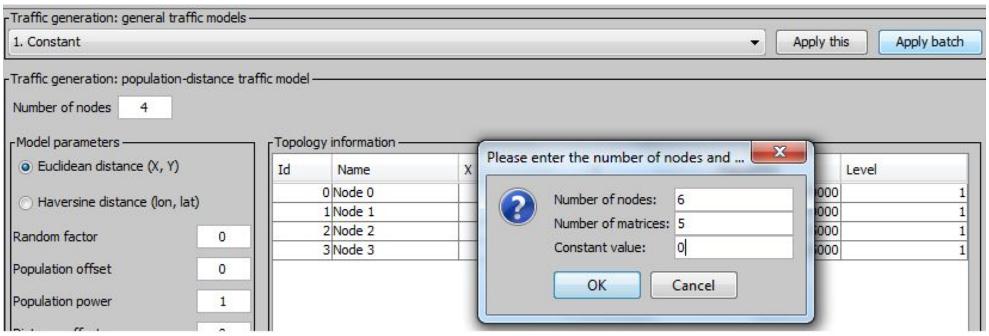
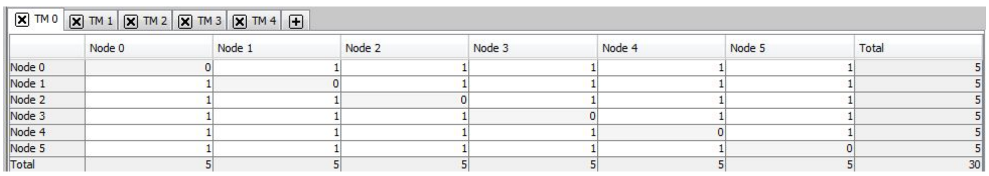


Figura 5.6: Net2Plan example on creating a batch of matrices

Using this option, 5 traffic matrices for a 6 node network were created all with a constant value of 1 as can be seen on figure 5.7 that shows the first matrix of the batch.



TM 0 | TM 1 | TM 2 | TM 3 | TM 4

	Node 0	Node 1	Node 2	Node 3	Node 4	Node 5	Total
Node 0	0	1	1	1	1	1	5
Node 1	1	0	1	1	1	1	5
Node 2	1	1	0	1	1	1	5
Node 3	1	1	1	0	1	1	5
Node 4	1	1	1	1	0	1	5
Node 5	1	1	1	1	1	0	5
Total	5	5	5	5	5	5	30

Figura 5.7: Net2Plan Traffic Matrix Example

This example demonstrates how several different types of traffic can be introduced for a network by creating different matrices for each. These can then be saved individually and will further on be used as traffic matrices for ODU's 0 through 4.

5.2.2 Creating the Network topologies

To start with the Network creation tools in Net2Plan go to "Tools → Offline network design"or press $Alt + 1$. The network design menu is shown on Figure 5.8.

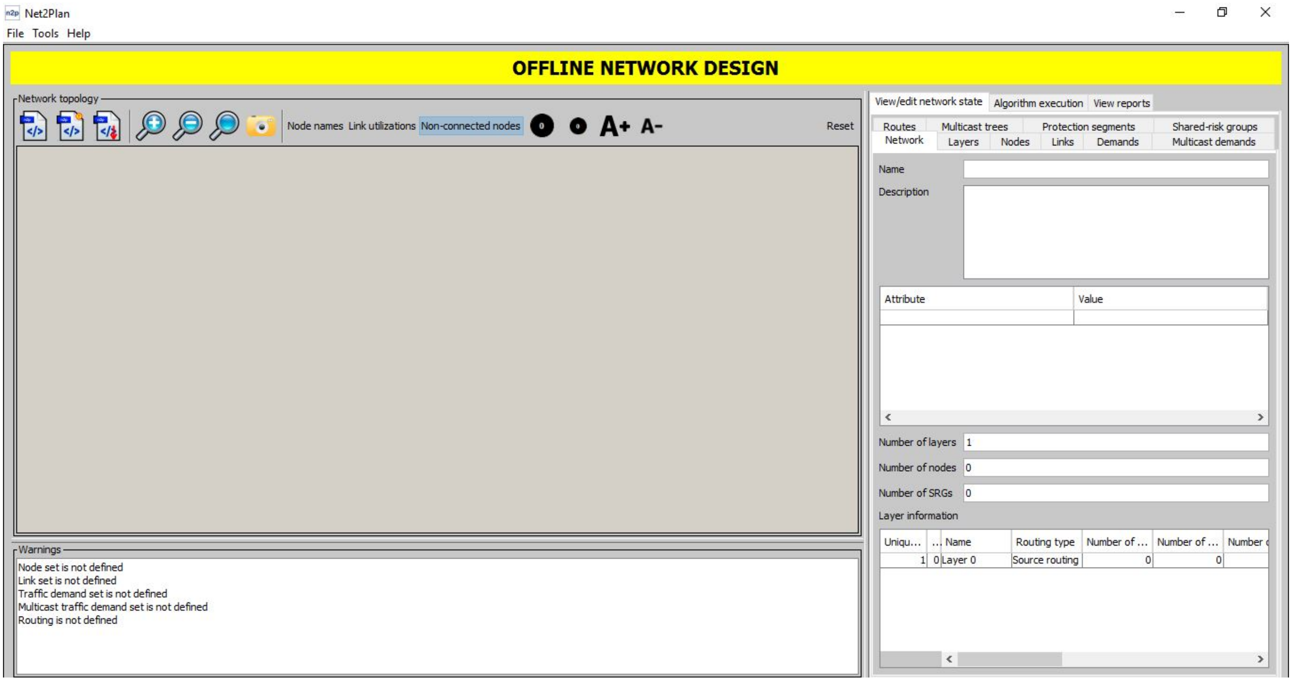


Figura 5.8: Net2Plan Offline Network Design

On the left side, the network topology part has the option to load an existing design and demand set or a new one can be created. To start creating a new network, first nodes have to be introduced by right clicking on the grey area and choosing "Add node here". Links between nodes are created by holding a click on the origin node and dragging until the destination node, holding shift before releasing the click creates 2 links, one in each direction. Another option to create links is to right click on an existing node and choosing the desired create a link option. Nodes can be moved by holding control and dragging them into the desired position.

Below the network topology is the "Warnings"box where the parts missing from having a functional network are displayed. For example if the nodes and links where already created it should say "traffic demand set is not defined"and "Routing is not defined"as these were still not introduced.

The whole right side of the network design menu are the parameters separated into various tabs which will be explored further on in this document. Besides these tabs, there is also the tab for Algorithm execution where the network is modified based on built algorithms, for example a routing algorithm and the View reports tab where information on the network can be displayed from built in reports.

Figure 5.9 demonstrates an example of the 6 node and 16 links network created using the tools explained above. As can be seen on the image at the warning tab, this network sill has several steps left to become a fully functional network. The link capacity will be defined based on the routing algorithm chosen and the demand set will be loaded based on the matrices created.

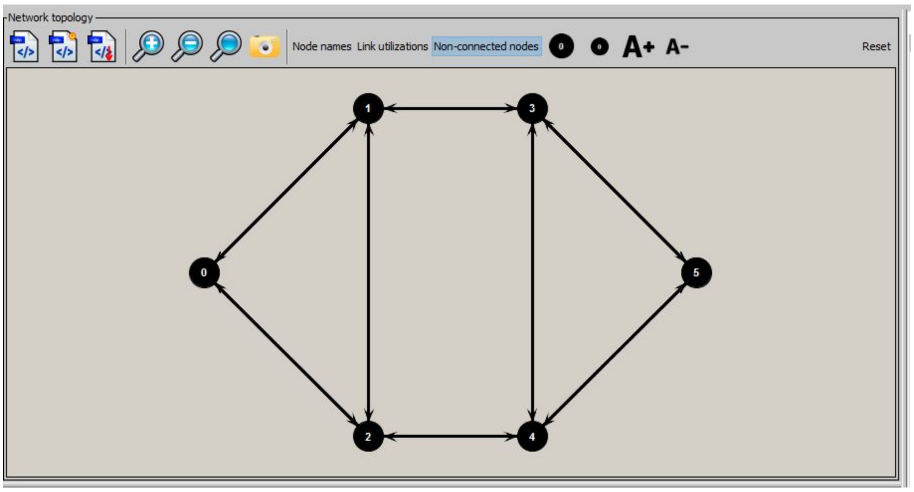
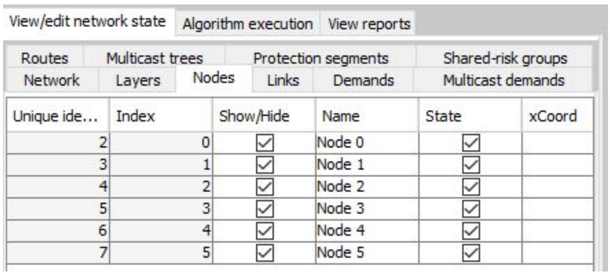


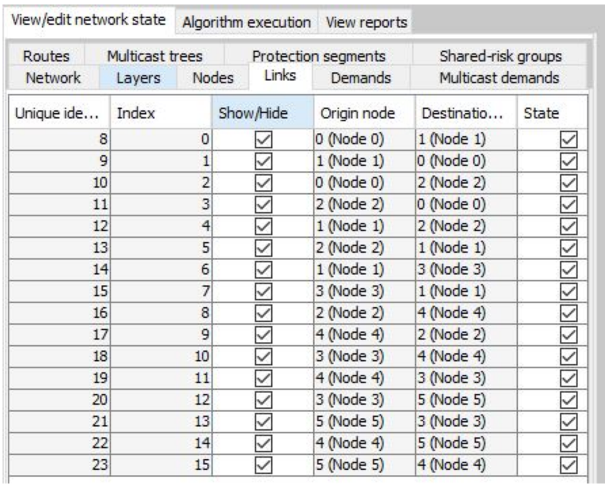
Figura 5.9: Net2Plan Network Example

The links and nodes parameters created for the network can be visualized and modified as seen on Figures 5.10(a) and 5.10(b) displaying the tabs for each case.



Unique ide...	Index	Show/Hide	Name	State	xCoord
2	0	☑	Node 0	☑	
3	1	☑	Node 1	☑	
4	2	☑	Node 2	☑	
5	3	☑	Node 3	☑	
6	4	☑	Node 4	☑	
7	5	☑	Node 5	☑	

(a) a



Unique ide...	Index	Show/Hide	Origin node	Destinatio...	State
8	0	☑	0 (Node 0)	1 (Node 1)	☑
9	1	☑	1 (Node 1)	0 (Node 0)	☑
10	2	☑	0 (Node 0)	2 (Node 2)	☑
11	3	☑	2 (Node 2)	0 (Node 0)	☑
12	4	☑	1 (Node 1)	2 (Node 2)	☑
13	5	☑	2 (Node 2)	1 (Node 1)	☑
14	6	☑	1 (Node 1)	3 (Node 3)	☑
15	7	☑	3 (Node 3)	1 (Node 1)	☑
16	8	☑	2 (Node 2)	4 (Node 4)	☑
17	9	☑	4 (Node 4)	2 (Node 2)	☑
18	10	☑	3 (Node 3)	4 (Node 4)	☑
19	11	☑	4 (Node 4)	3 (Node 3)	☑
20	12	☑	3 (Node 3)	5 (Node 5)	☑
21	13	☑	5 (Node 5)	3 (Node 3)	☑
22	14	☑	4 (Node 4)	5 (Node 5)	☑
23	15	☑	5 (Node 5)	4 (Node 4)	☑

(b) b

Figura 5.10: Network a) Nodes tab ; b) Links tab

On the Nodes tab most of the parameters are still 0 as there is no traffic on the network but there are three parameters that can be changed here. A node name can be set and both x and y coordinates can be defined as a more thorough alternative to define the node position.

On the links tab, again most is at 0 at this moment while the parameters that can be manually set are the link capacity, at 0 until defined and the link length which was set to the same value in every link.

Having the basic physical topology created, the next step is to load the demand set into the network. In the case where there are multiple traffic matrices an algorithm was developed to aggregate these in order for it to be possible to load all demands. For traffic matrices with ODU signals, an algorithm called "joinTrafficMatrices"can aggregate the different ODUs and convert them to ODU0 in order to have all the traffic in the same units. Besides converting the different ones to ODU0 it also creates an attribute in each demand indicating the type of signal before converting. This attribute can be seen on the demands tab after loading the resulting demand list. Figure 5.11 shows the algorithm to be used.

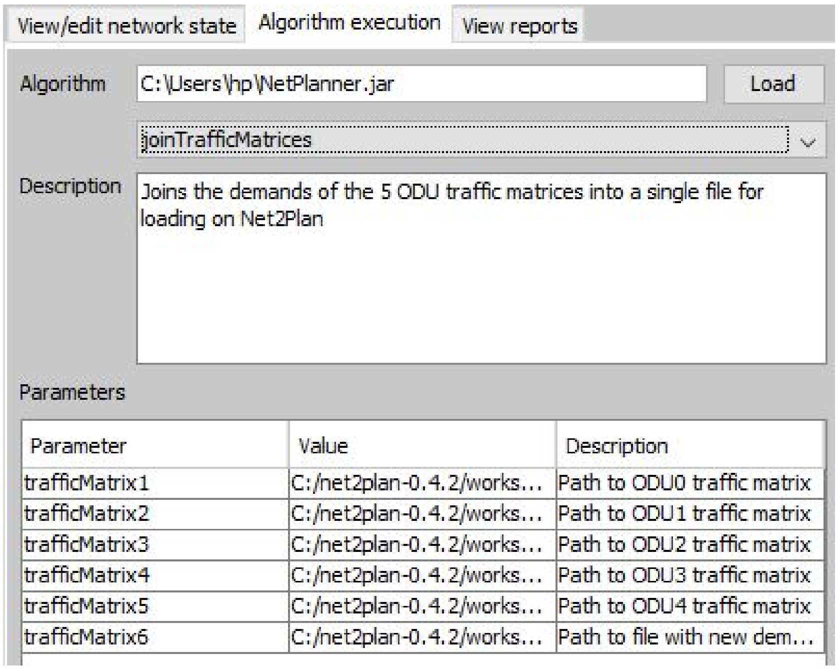


Figura 5.11: joinTrafficMatrices Algorithm

As can be seen on Figure 5.11 there are 6 user defined parameters, the first five are the paths for the traffic matrices to be aggregated in order, as said in the description. The last parameter is the resulting demand list that can then be loaded into the network.

The paths are by default defined considering Net2Plan is on C: and the matrices are in the default directory were they are saved. Lastly, the name of the files are in order ODU0.n2p through ODU4.n2p. All the path and file names can be changed to where the matrices are saved taking into account that just the order of the ODUs needs to be kept due to the conversion to ODU0 units.

To load the resulting demands into the created network the second icon on top of the network topology called "Load a demand traffic set"is used. After this, the warning tab changes from "Traffic demand set not defined"to "Traffic losses: Not all the traffic is being carried". This new warning indicates that the demand are in the network but as the routes have not yet been defined the traffic is not being transported.

In the demands tab, all the traffic that was created will be displayed in order of ODU type. For this case as all matrices were unitary and uniform, there are thirty demands with offered traffic 1 which is the ODU0 matrix and then consecutively groups of 30 demands (6 nodes) with offered traffic based on the ODU type (5 matrices). For example, an ODU1 is equivalent to two ODU0 so these demands have 2 in offered traffic and an attribute called ODU with value 1.

Before going into the network routing, the network transport mode needs to be defined by creating a logical topology. An algorithm was developed that creates a new layer consisting on this topology depending on the transport mode chosen. This algorithm can be seen on Figure 5.12.

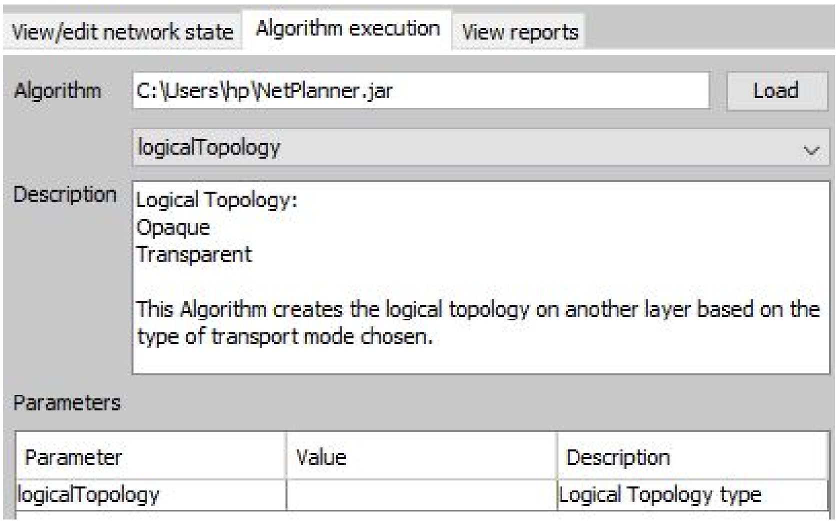
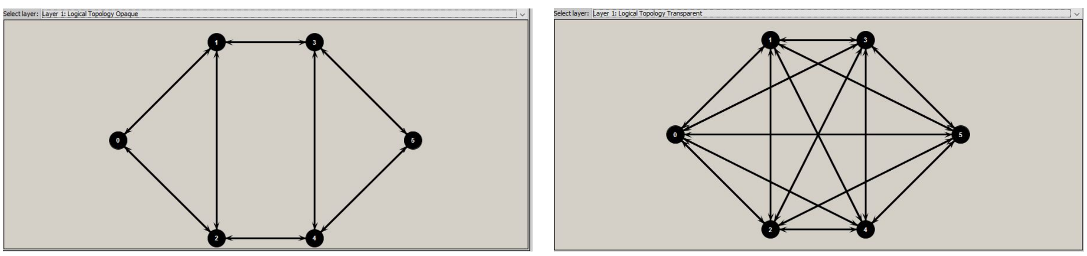


Figura 5.12: Net2Plan Logical Topology Algorithm

There are two user defined parameters on this algorithm. The "logicalTopology"parameter defines the type of transport mode, Opaque or Transparent.

Besides creating this new Layer, the algorithm also copies the demands to that layer and defines the logical links based on the length of the physical ones. Figures 5.13(a), 5.13(b) demonstrate the resulting logical topologies for each transport mode.



(a) (b)

Figura 5.13: Logical Topology: a) Opaque; b) Transparent;

As can be seen on the logical topologies, for an Opaque transport mode the traffic goes through an OEO conversion at every node and as such the logical topology is the same as the physical one.

In the Transparent mode, there are no regeneration in intermediate nodes and as such the logical topology shows that the traffic between nodes flows directly without grooming with signals from another source.

5.2.3 Routing and Grooming

In this section, different routing and grooming options will be discussed for both a network without protection and using a 1+1 protection scheme (dedicated path protection).

The routing will be done based on a shortest path algorithm where the routes for each demand are are created based one either the shortest number of hops needed to reach the destination node or by shortest distance in km. The option can be chosen as a user defined parameter on the algorithm as can be seen on Figure 5.14. This algorithm does the routing in both the logical and physical topologies based on the transport mode chosen and makes sure routes are bidirectional meaning the route from node o to d should be the opposite direction of node d to o as there could be different routes with the shortest path that are not using the same path.

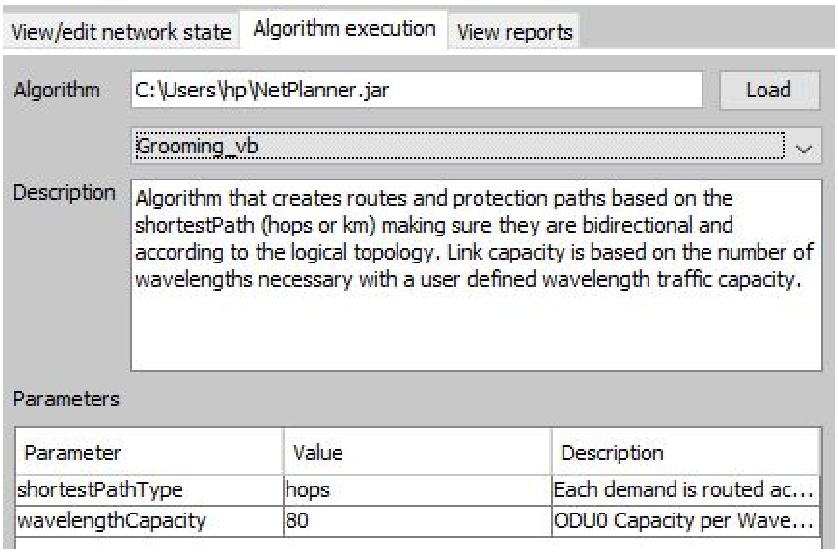


Figura 5.14: Net2Plan Grooming shortest Path Algorithm

Besides the metric through which the shortest path is calculated, the other available parameter defines the amount of ODU0s each wavelength is capable of carrying. By default it is set for 80 ODU0s as it is equal to an ODU4 or 100 Gbit/s.

The protection segments similarly to the routes have their own tab where information on their path, route it protects and such can be observed.

5.2.4 Reports

As looking separately at each tab to obtain information for different parts of the network is a slow process and does not show some important metrics, Net2Plan allows for the creation of reports where in a similar way to algorithms they can be adjusted to display the information needed, these can also be seen in html format for an easier read. In this section, the report developed will be demonstrated.

A very important aspect in network planning that is not present natively in Net2Plan is a Network Cost report. To fulfil this gap, a report was created to obtain the network Capex based on user defined equipment costs present on Table 5.1.

Equipment	Costs
OLT	15000€
Transponder	5000€/GB
Optical Amplifier	4000€
EXC	10000€
OXC	20000€
EXC Port	1000€/GB/s
OXC Port	2500€/port

Tabela 5.1: Equipment Costs

These Equipment costs are introduced into a report as user defined parameters as can be seen on Figure 5.15.

Besides the equipment costs, this report also has the parameter "span". The value of this variable is used to calculate the number of optical amplifiers needed in the network using Equation 5.1.

$$N^R = \sum_{l=1}^{L} \left(\left\lceil \frac{len_l}{span} \right\rceil - 1 \right) \tag{5.1}$$

The other parameters of this equation being:

- $N^R \rightarrow$ Total number of regenerators/amplifiers
- $len_l \rightarrow$ Length of link l
- $span \rightarrow$ Distance between amplifiers

By running the report three main categories are presented to the user.

The first category displayed by the report is the Detailed per-link description. In here the number of optical channels or wavelengths is displayed for each link based on the grooming algorithm used. The numbers displayed are based on the physical topology and represent all the wavelengths that will be needed to transport the network traffic. Using this information it is possible to obtain the average and total number of optical channels on the network.

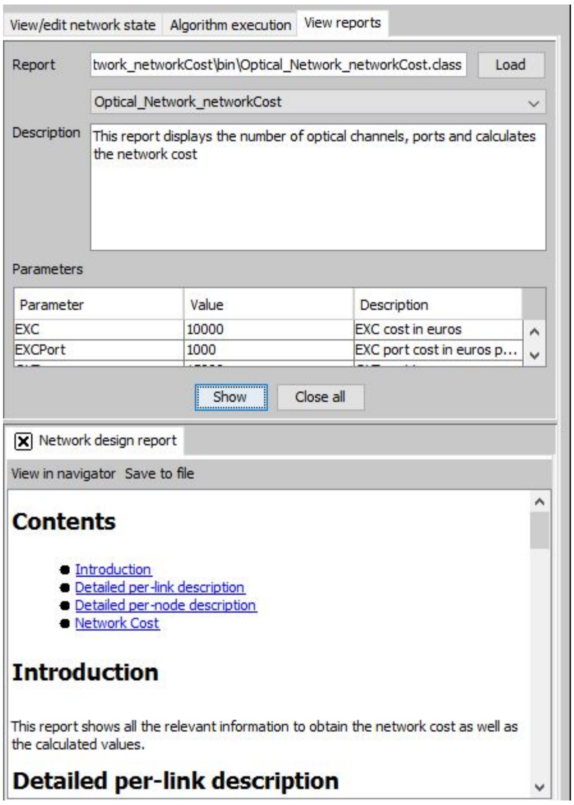


Figura 5.15: Network Cost Report

Besides the number of wavelengths, this section also indicates the amount of amplifiers necessary in each link.

The second category is the Detailed per-node description. This section displays a table indicating how many ports are needed of each type for every node. The number of tributary ports obtained in each node is the sum of all traffic originating from that node or ending on it depending if its the input or output ports divided by the amount of traffic each optical channel can carry. This number also depends on the links through which traffic will be routed, for example, if 40 ODU0s are transmitted into 2 separate links only one wavelengths could carry it but as they are going through different routes then 2 wavelengths will be used resulting in also a need for 2 tributary ports.

The number of line ports is obtained by adding the total amount of optical channels in the links that use that specific node as origin or destination.

Finally the total number of ports is as expected the sum of all the tributary ports with the line ones. With this information the average and the total number of ports in the network can be obtained which will later be used in calculating the network cost.

Having the node and link information available, the network cost can then be calculated as displayed on the third category of the report. The Node electrical cost is obtained with Equation 5.2 for a Transparent Network.

$$C_{exc} = (\gamma_{e0} \times N) + (\gamma_{e1} \times \tau \times 2 \times P_{TRIB}) \tag{5.2}$$

- $C_{exc} \rightarrow$ Electrical Ports Cost
- $\gamma_{e0} \rightarrow$ EXC cost in Euros
- $N \rightarrow$ Number of Nodes
- $\gamma_{e1} \rightarrow$ EXC port cost in Euros per GB/s
- $\tau \rightarrow$ Traffic supported by optical channel
- $P_{TRIB} \rightarrow$ Number of tributary ports

The cost values can be obtained from Table 5.1, the number of nodes is a known value when designing a network, the traffic supported by optical channel is defined by the grooming algorithm or by dividing the link capacity by its amount of optical channels and the number of tributary ports was obtained on the previous section of the report.

For an Opaque network, the electrical nodes cost is similar as displayed in Equation 5.3.

$$C_{exc} = (\gamma_{e0} \times N) + (\gamma_{e1} \times \tau\,(P_{LINE} + P_{TRIB})) \tag{5.3}$$

The node optical cost on the other hand, can be calculated for a Transparent network using Equation 5.4.

$$C_{oxc} = (\gamma_{o0} \times N) + \gamma_{o1} \times (P_{LINE} + P_{TRIB}) \tag{5.4}$$

- $C_{oxc} \rightarrow$ Optical Ports Cost
- $\gamma_{o0} \rightarrow$ OXC cost in Euros
- $N \rightarrow$ Number of Nodes
- $\gamma_{o1} \rightarrow$ OXC port cost in Euros
- $P_{TRIB} \rightarrow$ Number of tributary ports
- $P_{LINE} \rightarrow$ Number of line ports

As for the electrical ports, the cost values were previously defined in Table 5.1 and as such, only the number of ports is needed. These value were obtained on the second part of the report (Detailed per-Node description).

For an Opaque network, the node optical cost is 0 as the ports are all electrical.

The Node Total Cost is as expected the sum of both the optical and electrical node costs.

The rest of the network cost is from the links. This cost is obtained with Equation 5.5.

$$C_L = \left(\gamma_0^{OLT} \times L\right) + \left(\gamma_1^{OLT} \times \tau \times W\right) + \left(N^R \times c^R\right) \tag{5.5}$$

- $C_L \rightarrow$ Links Cost
- $\gamma_0^{OLT} \rightarrow$ OLT cost in Euros
- $L \rightarrow$ Number of unidirectional Links
- $\gamma_1^{OLT} \rightarrow$ Transponder cost in Euros
- $\tau \rightarrow$ Traffic per port
- $W \rightarrow$ Total number of optical channels
- $N^R \rightarrow$ Total number of optical amplifiers
- $c^R \rightarrow$ Optical amplifiers cost in Euros

As in previous equations, the costs are all available in Table 5.1. The total number of optical channels can be obtained by summing the wavelengths in each link on the Detailed per-Link description section. The number of optical amplifiers was calculated previously with Equation 5.1.

The middle part of the equation: $\gamma_1^{OLT} \times \tau \times W$ refers to the Transponders cost while the rest is the "Fiber"and the "OLT"cost. Lastly the total network cost can be obtained by adding the Links cost with the Nodes cost.

5.3 Results

This section will display the results obtained using the algorithms and reports previously explained for a network with an Opaque transport mode and for one with Transparent.

5.3.1 Opaque with 1+1 protection

The results will be displayed only in the logical topology as in an opaque network it is the same as the physical one. Using the algorithm presented on figure 5.14 the routes and protection segments are created as well as the grooming.

There is not a second algorithm type for wavelengths reduction due to the fact that, that algorithm chooses the best path based on the shortest or disjointed path which in this case both need to be used one for work and one for protection. As such, is difficult to reduce in any instance the shortest path because of the algorithm performance.

The traffic matrix for the reference 6 node network, used for demonstration is shown below.

$$\begin{bmatrix} 0 & 17 & 17 & 15 & 1 & 13 \\ 17 & 0 & 32 & 7 & 15 & 114 \\ 17 & 32 & 0 & 11 & 46 & 1 \\ 15 & 7 & 11 & 0 & 11 & 7 \\ 1 & 15 & 46 & 11 & 0 & 93 \\ 13 & 114 & 1 & 7 & 93 & 0 \end{bmatrix}$$

Figura 5.16

The amount of traffic that needs to be reserved in each link is as was to be expected a lot higher due to the need to reserve double the amount and in more links. The same happens in terms of wavelengths.

The number of wavelengths can again be seen on the links section of the "networkCost"report as well as the amplifiers needed on Figure 5.17.

Detailed per-link description

Node Pair	Wavelengths forward	Wavelengths backward	Amplifiers forward	Amplifiers backward
Node 0 «-» Node 1	10	10	1	1
Node 0 «-» Node 2	10	10	1	1
Node 1 «-» Node 2	14	14	1	1
Node 1 «-» Node 3	14	14	1	1
Node 2 «-» Node 4	14	14	1	1
Node 3 «-» Node 4	14	14	1	1
Node 3 «-» Node 5	10	10	1	1
Node 4 «-» Node 5	10	10	1	1

Figura 5.17: Links for Opaque Network with 1+1 Protection

The conclusions to take from these results are the same as was previously discussed as the number of amplifiers does not change and the wavelengths are the ones shown on the line matrices.

As for the nodes in the network Figure 5.18 shows the ports needed.

Detailed per-node description

Name	Trib ports in	Trib ports out	Line Ports in	Line Ports out	Total Ports in	Total Ports out
Node 0	9	9	20	20	29	29
Node 1	9	9	38	38	47	47
Node 2	9	9	38	38	47	47
Node 3	10	10	38	38	48	48
Node 4	10	10	38	38	48	48
Node 5	9	9	20	20	29	29
Total	56	56	192	192	248	248

Figura 5.18: Nodes for Opaque Network with 1+1 Protection

Again, the difference for the case without protection is only on the number of line ports as this value is based on the number of wavelengths going in or out of that node.

Comparing the number of ports obtained here with the network with a transparent transport mode, the amount is lower for the opaque network due to the reduced number of wavelengths required to route the traffic.

Lastly the total network cost is on Figure 5.19.

Network Cost

<table>
<tr><th colspan="2">Category</th><th>Cost</th><th>Total</th></tr>
<tr><td rowspan="3">Link Cost</td><td>OLT</td><td>240,000</td><td rowspan="3">96,304,000</td></tr>
<tr><td>Transponders</td><td>96,000,000</td></tr>
<tr><td>Amplifiers</td><td>64,000</td></tr>
<tr><td rowspan="2">Node Cost</td><td>Electrical</td><td>24,860,000</td><td rowspan="2">24,860,000</td></tr>
<tr><td>Optical</td><td>0</td></tr>
<tr><td colspan="3">Total Network Cost</td><td>121,164,000</td></tr>
</table>

Figura 5.19: Network Cost for Opaque Network with 1+1 Protection

The increase in cost is as described on the transparent network just based on the additional number of wavelengths required which translates in also more trunk ports needed. As noted above in the amount of ports, the cost is also lower in this instance when compared to the transparent network due to the cheaper cost in transponders and optical ports.

5.3.2 Transparent with 1+1 protection

For a network with a transparent transport mode, the routing as was explained before, is done using a shortest path algorithm since there are no traffic grooming between different node pairs. For this instance as there is also a 1+1 protection scheme in place, the algorithm needs to not only create the routes but also a protection segment for each route. This segment is the shortest disjoint path of the route created.

Comparing the results obtained here with the previous example, it can be seen that the amount of traffic and wavelengths is significantly higher. It is in both cases, double the amount of before since the same quantity needs to be reserved for protection.

The conclusions that can be taken from the physical topology are as explained before, the huge number of wavelengths is related to the needed for double the amount of traffic where this extra will go through even more links.

For the logical topology the Average second shortest path number of hops is 1 since as for the shortest path, it is considered that there are always direct links between nodes in a transparent network. As for the physical topology, this value is not so obvious as it has to be calculated based on the second shortest path between each node pair.

These differences for the transparent network with protection segments can also be seen on the information provided in the "networkCost"report. Figure 5.20 shows the results for the links in the physical topology.

Detailed per-link description

Node Pair	Wavelengths forward	Wavelengths backward	Amplifiers forward	Amplifiers backward
Node 0 «-» Node 1	12	12	1	1
Node 0 «-» Node 2	12	12	1	1
Node 1 «-» Node 2	18	18	1	1
Node 1 «-» Node 3	18	18	1	1
Node 2 «-» Node 4	18	18	1	1
Node 3 «-» Node 4	18	18	1	1
Node 3 «-» Node 5	12	12	1	1
Node 4 «-» Node 5	12	12	1	1

Figura 5.20: Links for Transparent Network with 1+1 Protection

It can be seen that as expected the number of amplifiers is the same due to the link lengths remaining constant but the number of wavelengths are higher due to having a grooming scheme worst with this topology.

The results in terms of ports per node are shown below.

Detailed per-node description

Name	Trib ports in	Trib ports out	Line Ports in	Line Ports out	Total Ports in	Total Ports out
Node 0	10	10	24	24	34	34
Node 1	10	10	48	48	58	58
Node 2	10	10	48	48	58	58
Node 3	10	10	48	48	58	58
Node 4	10	10	48	48	58	58
Node 5	10	10	24	24	34	34
Total	60	60	240	240	300	300

Figura 5.21: Nodes for Transparent Network with 1+1 Protection

The number of tributary ports remain the same but the number of line ports increase based on the higher number of wavelengths needed in the network.

Lastly, the total network cost is shown on Figure 5.22.

Network Cost

Category		Cost	Total
Link Cost	OLT	240,000	120,304,000
	Transponders	120,000,000	
	Amplifiers	64,000	
Node Cost	Electrical	12,060,000	12,930,000
	Optical	870,000	
Total Network Cost			133,234,000

Figura 5.22: Network Cost for Transparent Network with 1+1 Protection

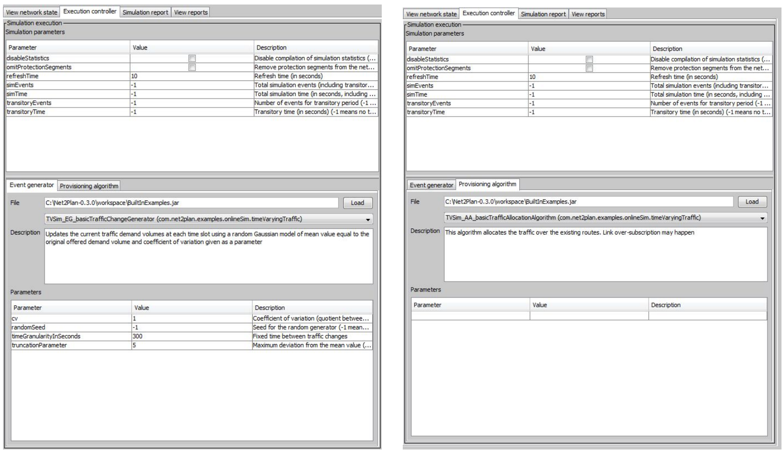
The results obtained for the network Cost confirm those obtained in the previous categories in this report. The OLT and amplifiers cost does not change as the number of links and amplifiers remains the same. Similarly, the electrical ports cost is also the same as the amount of ADD/DROP ports remains the same.

The differences are in the Transponders cost in the links and the Optical cost in the nodes. These as expected, cost more based on the increased number of them needed in the network to have a 1+1 protection scheme in a transparent transport mode network.

5.4 Simulations

To access the Simulations window go to "Tools → Online Simulation"or press $Alt + 3$. The simulations menu is very similar to the one available for network design with the notable difference that in this instance the network needs to have already been saved with every definition done as all the tabs described earlier are only available here for viewing.

Using the already built network with the demand set introduced as well as routing and protection segments, an example of a Time-varying simulation is demonstrated. The main parameters to be chosen on this simulation are the "Event generator"and the "Provisioning algorithm", displayed on Figures 5.23(a) and 5.23(b).



(a) (b)

Figura 5.23: a) Net2Plan Event generator ; b) Net2Plan Provisioning algorithm

The Event generator shown creates a time varying simulation by updating the network traffic based on the chosen parameters while the allocation algorithm in this case only allocates this traffic into the available routes. Besides these options it is also possible to change the main simulation parameters which are displayed on the top half.

Having defined all the simulation parameters and the other necessary options, the simulation can be started by just pressing "run"below the network topology at the lower left side. The "simulation controller"will update automatically based on the time defined at the simulation parameters or it can be paused for an update on the results.

5.5 Implementing new algorithms on Net2Plan

This section will demonstrate some of the possibilities provided by Net2Plan as an open source tool. By creating new algorithms or reports it is possible to adapt this program for most necessities in terms of network planning.

There are already several built-in algorithms present in Net2Plan but as it is impossible to have an algorithm built for every specific necessity it is possible for each user to build new ones or modify existing ones to fulfil what needs to be done.

As everything in Net2Plan was built in Java, the program "Eclipse"that can be downloaded from `https://eclipse.org/downloads/` was chosen as the best option for coding. All the .java files from the available algorithms in Net2Plan can be downloaded from its website and introduced into "Eclipse"to create a class.

When opening Eclipse, the first choice is to define the work directory in which all the projects will be created. Having defined the workspace, Figure 5.24 demonstrates the window for creating new projects in Eclipse, this can be accessed by going into "File → New → Java Project". In this window, only the name needs to be defined and then finish.

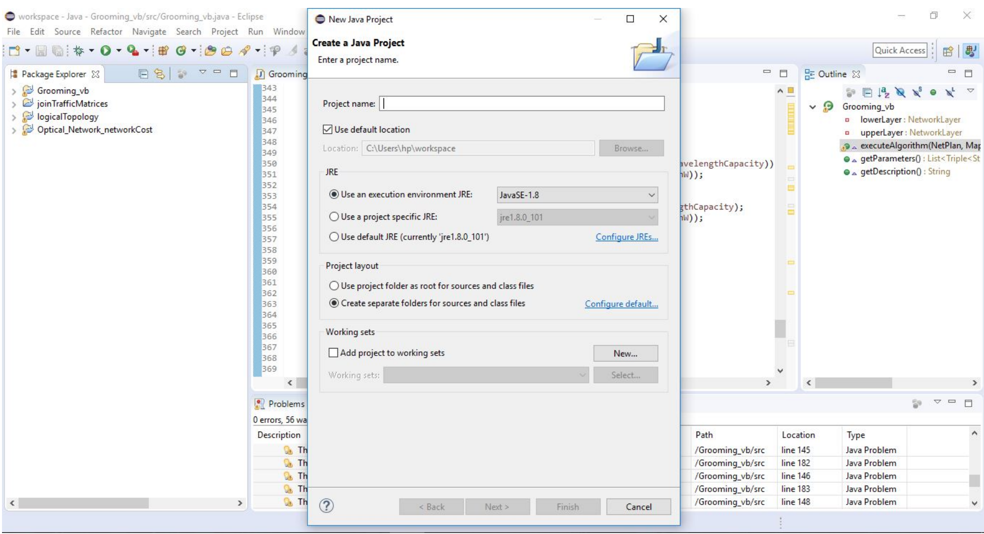


Figura 5.24: Eclipse new project

Having created a new project, a "src"directory should be available where the .java should be located. As a starting point, an existing algorithm should be used as a template and then modified to do its necessary purpose. Figure 5.25 shows a newly created project called "logicalTopology".

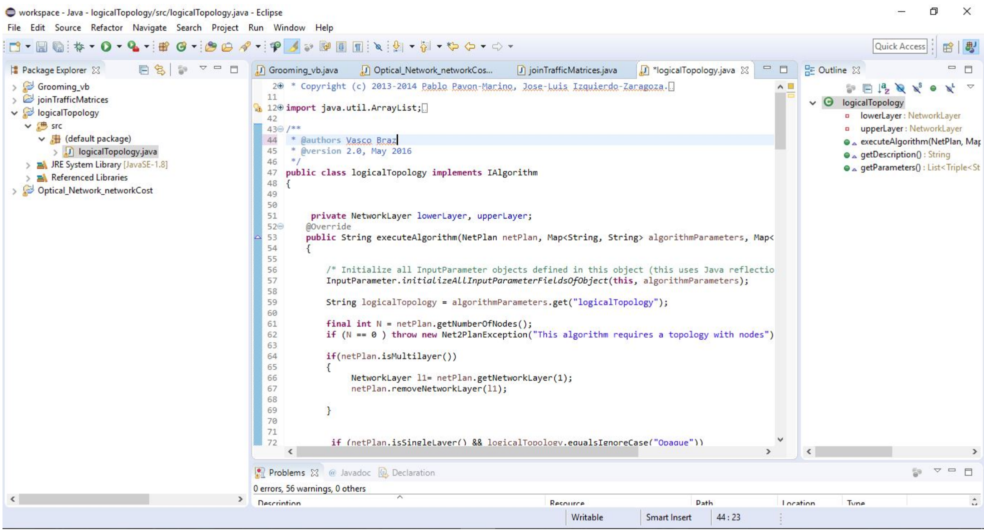
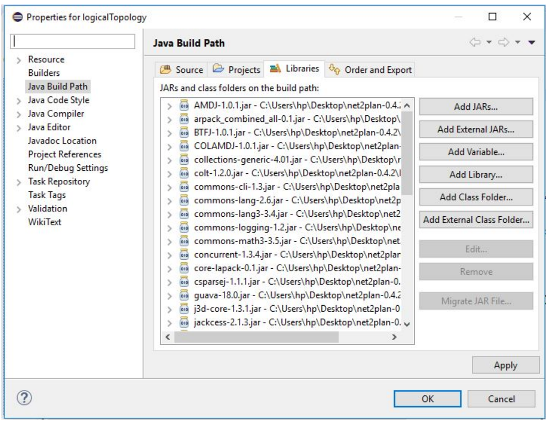


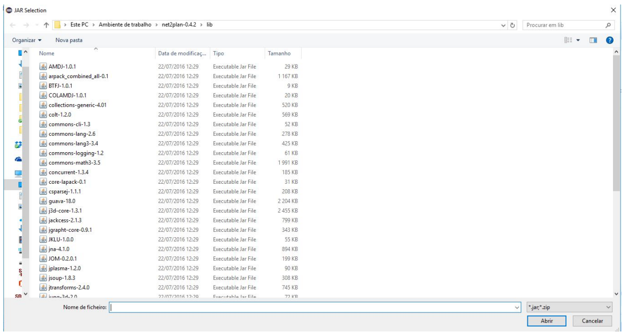
Figura 5.25: Eclipse new project with source file

To add the library files to a project, right click on it and choose "Build Path → Configure Build Path ...". On the window that appears, press "Add External Jars..."and include all the files in the Net2Plan "lib"directory as shown on Figures 5.26(a) and 5.26(b).

To further illustrate how these modifications to algorithms work, the project created above using an existing code as a template was modified to create a new algorithm which creates the logical topology of a network in another layer.



(a)



(b)

Figura 5.26: a) Eclipse Java Configure Build Path ; b) Net2Plan library files

The code created is shown on Figure 5.24. By saving this project on Eclipse a .class file is created on the bin directory of the project which can be loaded on Net2Plan. On the "Algorithm execution"tab at the "Offline network design", the "BuiltInExamples.jar"is loaded as the default location for algorithms and as it is a .jar file all the available ones that came with Net2Plan are integrated into it. To get the newly created algorithm available, press "Load"and find the .class file created in Eclipse as shown on Figure 5.27.

As was said before and can be seen on the "Description", this algorithm creates the network logical topology as was explained on section 5.2.2.

Algorithms developed on Eclipse can be exported into a .jar file so on Net2Plan this file can be loaded and all the algorithms developed are shown in a list in the same manner as the ones that came with the Net2Plan installation. The export option can be accessed by going into File $\rightarrow$ Export, and the menu are shown in Figures 5.28(a) and 5.28(b).

By default only the .class files are exported along with the necessary libraries so that the algorithms can be loaded on Net2Plan. There is however an option to also export the .java files so that if needed the ones who will use the code also have access to it if they need to change it.

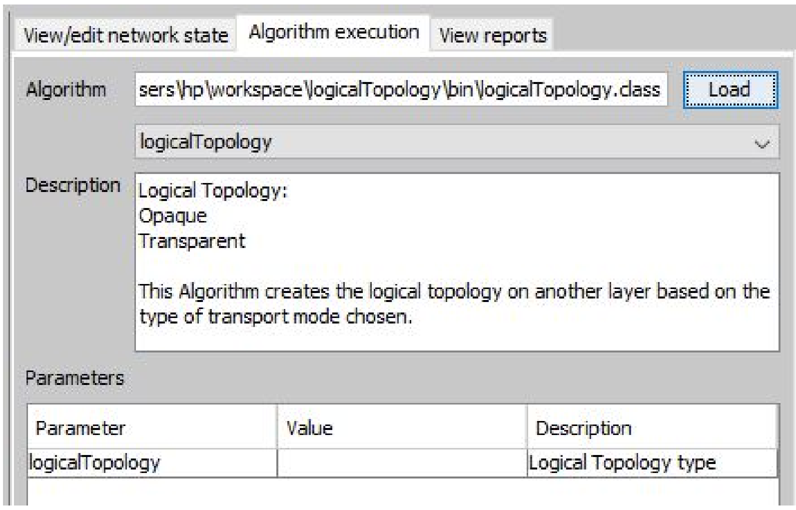
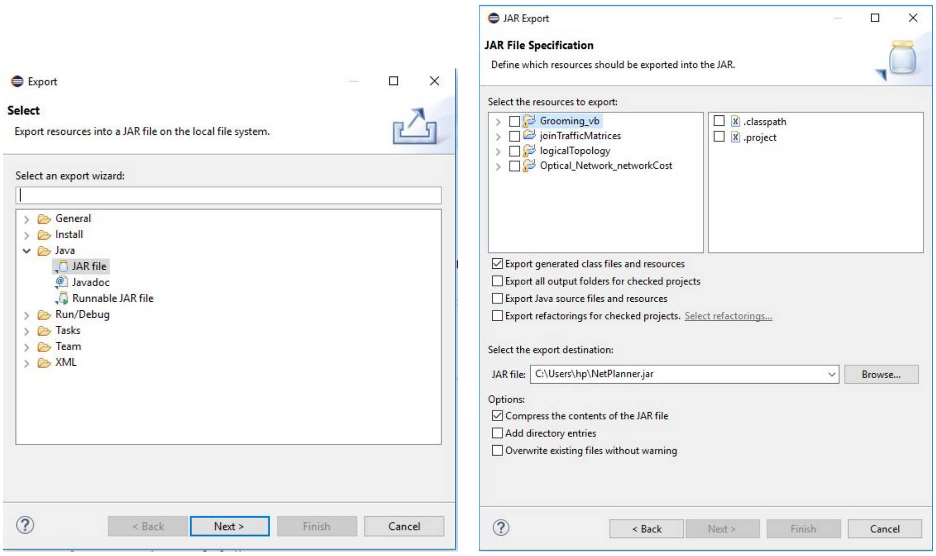


Figura 5.27: Net2Plan new algorithm



(a) (b)

Figura 5.28: a) Eclipse export ; b) Projects to export into a .jar file

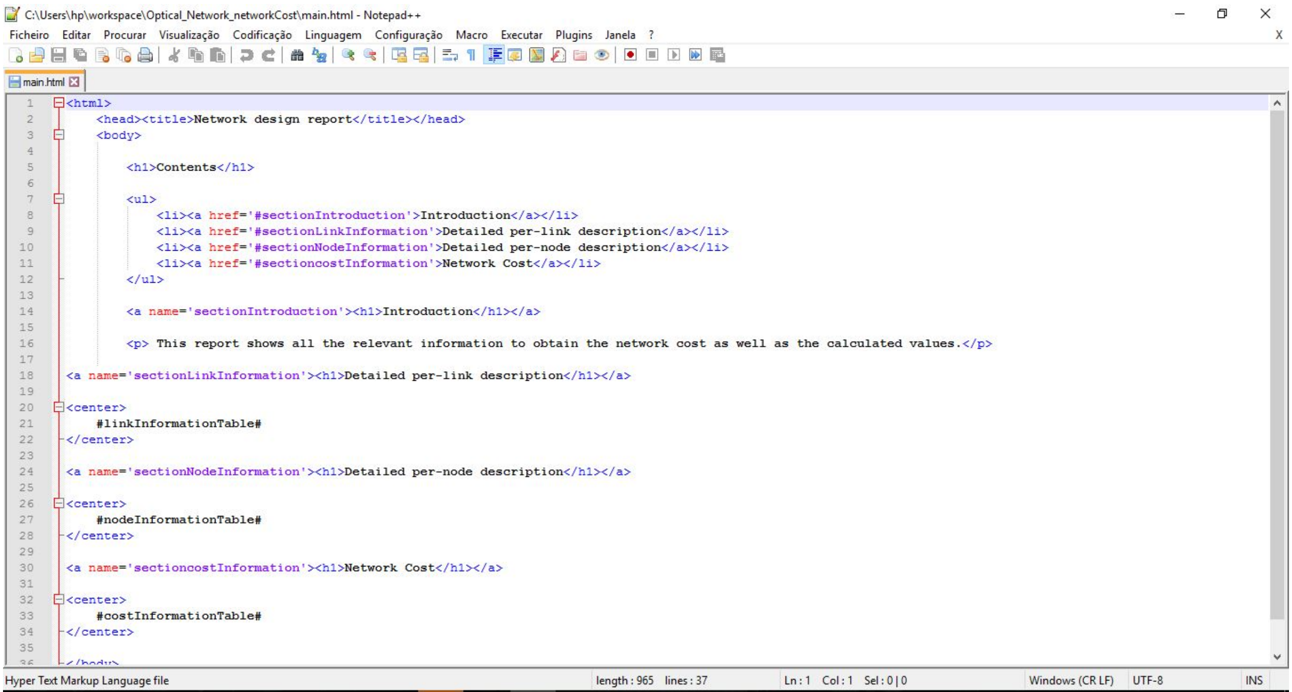
5.6 Developing new Reports

Similarly to the way algorithms can be modified or new ones created, also reports can be done using almost the same steps. For the following examples, the "Optical_Network_networkcost"is being used as a basis for modifying or creating new reports.

An important point to note as the main difference as to when modifying algorithms, is that in this case not only are the Net2Plan libraries needed but also the extra files summoned by the report. These files can be found opening the "BuiltInExamples.jar"file in the Net2Plan directory on the corresponding report.

For the report being used there is an .html file called "main"which is where the information to be displayed in html form is described as well as several image files that are displayed in the report. As such, if the modifications to be done in the reports are to be shown in html format the "main.html"file needs to be modified in order to adapt to these changes.

The tables themselves are created in eclipse as Java code but the html file needs to be opened for example with "Notepadd++"to change some its code as the tables are being appended into the html. Figure 5.29 shows the modified html that is used in the Optical_Network_networkcost.




```html
<html>
    <head><title>Network design report</title></head>
    <body>

        <h1>Contents</h1>

        <ul>
            <li><a href='#sectionIntroduction'>Introduction</a></li>
            <li><a href='#sectionLinkInformation'>Detailed per-link description</a></li>
            <li><a href='#sectionNodeInformation'>Detailed per-node description</a></li>
            <li><a href='#sectioncostInformation'>Network Cost</a></li>
        </ul>

        <a name='sectionIntroduction'><h1>Introduction</h1></a>

        <p> This report shows all the relevant information to obtain the network cost as well as the calculated values.</p>

<a name='sectionLinkInformation'><h1>Detailed per-link description</h1></a>

<center>
    #linkInformationTable#
</center>

<a name='sectionNodeInformation'><h1>Detailed per-node description</h1></a>

<center>
    #nodeInformationTable#
</center>

<a name='sectioncostInformation'><h1>Network Cost</h1></a>

<center>
    #costInformationTable#
</center>

</body>
```

Figura 5.29: html file for Network Cost report

As can be seen, this is a simple example of an html file since there are only hyper links created to link the contents index to the tables. Other options could be added as for example, hyper links to each of the network costs with the formula describing its calculations by adding the necessary information in this file. These extra options are present on more complex reports such as the "Report_networkDesign"where the images used are equations showcasing how some of the calculations are done.

5.7 Installing LPSOLVE for using in MatLab

Student Name	:	Tiago Esteves (November 28, 2017 - December 05, 2017)
Goal	:	Help other to install lpsolve for using in MatLab.

In this section will describe how to install lpsolve and how it can be used through matlab. For this it is necessary to follow the following steps:

1. Install lpsolve in your computer
2. Install lpsolve matlab extensions
3. Install the library

Step 1:
The first thing to do is to install lpsolve using the execute file "lp-solve-5.5.2.5-IDE-Setup"that can be found in GitHub through this link https://github.com/netxpto/NetPlanner/tree/Develop/software/lpsolve. The installation is quite simple and contains few steps for its execution.
In case there is any doubt or question you can always use the lpsolve reference guide in link: http://lpsolve.sourceforge.net/5.5/

Step 2:
In this step it is necessary to go to GitHub again and download with the name "lp-solve-5.5.2.0-MATLAB-exe-win64"in link https://github.com/netxpto/NetPlanner/tree/Develop/software/lpsolve and extract all the files. Then you need to put the **mxlpsolve.mexw64** and **mxlpsolve.m** files in the same folder as the .m files.
In case there is any doubt or question you can always use this guide for help: thereisanydoubtorquestionyoucanalwaysusethelpsolvereferenceguideinlink:

Setp 3:
Finally, once again in GitHub through the link https://github.com/netxpto/NetPlanner/tree/Develop/software/lpsolve we can find the last folder to get the necessary library, thus downloading the folder "lp-solve-5.5.2.0-dev-win64"and then include in the Windows PATH environment.

5.8 Opaque with 1+1 protection appendices

Student Name	:	Tiago Esteves (November 28, 2017 -)
Goal	:	Implement the dimensioning of optical networks in the opaque transport mode.

5.8.1 Script using MatLab

5.8.1.1 Scenario 1

RESULTS: Test Network

Scenario: Opaque Low Traffic

LINKS

Number of optical channels in the link (1,2): 2
Number of optical channels in the link (1,3): 2
Number of optical channels in the link (2,3): 4
Number of optical channels in the link (2,4): 3
Number of optical channels in the link (3,5): 3
Number of optical channels in the link (4,5): 3
Number of optical channels in the link (4,6): 3
Number of optical channels in the link (5,6): 3

NODES

Tributary Ports =

Node	ODU0	ODU1	ODU2	ODU3	ODU4
1	13	13	3	0	0
2	11	7	2	2	1
3	7	6	3	2	0
4	7	10	3	0	0
5	14	4	4	1	1
6	8	10	1	1	2

Line Ports =

Node	ODU4
1	4
2	9
3	9
4	9
5	9
6	6

PATHS
Link (1,2)————————————
Path between node (1,2)
Path between node (1,3)
Path between node (1,4)
Path between node (1,5)
Path between node (1,6)
Path between node (2,3)
Link (1,3)————————————
Path between node (1,2)
Path between node (1,3)
Path between node (1,4)
Path between node (1,5)
Path between node (1,6)
Path between node (2,3)
Link (2,3)————————————
Path between node (1,2)
Path between node (1,3)
Path between node (2,3)
Path between node (2,4)
Path between node (2,5)
Path between node (2,6)
Path between node (3,4)
Path between node (3,5)
Path between node (3,6)
Link (2,4)————————————
Path between node (1,4)
Path between node (1,5)
Path between node (1,6)
Path between node (2,4)
Path between node (2,5)
Path between node (2,6)
Path between node (3,4)

Path between node (3,5)
Path between node (3,6)
Link (3,5)————————————
Path between node (1,4)
Path between node (1,5)
Path between node (1,6)
Path between node (2,4)
Path between node (2,5)
Path between node (2,6)
Path between node (3,4)
Path between node (3,5)
Path between node (3,6)
Link (4,5)————————————
Path between node (1,4)
Path between node (1,5)
Path between node (2,4)
Path between node (2,5)
Path between node (3,4)
Path between node (3,5)
Path between node (4,5)
Path between node (4,6)
Path between node (5,6)
Link (4,6)————————————
Path between node (1,6)
Path between node (2,6)
Path between node (3,6)
Path between node (4,5)
Path between node (4,6)
Path between node (5,6)
Link (5,6)————————————
Path between node (1,6)
Path between node (2,6)
Path between node (3,6)
Path between node (4,5)
Path between node (4,6)
Path between node (5,6)

5.8.1.2 Scenario 2

RESULTS: Test Network

Scenario: Opaque High Traffic

LINKS

Number of optical channels in the link (1,2): 12
Number of optical channels in the link (1,3): 12
Number of optical channels in the link (2,3): 33
Number of optical channels in the link (2,4): 28
Number of optical channels in the link (3,5): 28
Number of optical channels in the link (4,5): 26
Number of optical channels in the link (4,6): 30
Number of optical channels in the link (5,6): 30

NODES

Tributary Ports =

Node	ODU0	ODU1	ODU2	ODU3	ODU4
1	130	130	30	0	0
2	110	70	20	20	10
3	70	60	30	20	0
4	70	100	30	0	0
5	140	40	40	10	10
6	80	100	10	10	20

Line Ports =

Node	ODU4
1	24
2	73
3	73
4	84
5	84
6	60

PATHS
Link (1,2)——————————
Path between node (1,2)
Path between node (1,3)
Path between node (1,4)

Path between node (1,5)
Path between node (1,6)
Path between node (2,3)
Link (1,3)————————————
Path between node (1,2)
Path between node (1,3)
Path between node (1,4)
Path between node (1,5)
Path between node (1,6)
Path between node (2,3)
Link (2,3)————————————
Path between node (1,2)
Path between node (1,3)
Path between node (2,3)
Path between node (2,4)
Path between node (2,5)
Path between node (2,6)
Path between node (3,4)
Path between node (3,5)
Path between node (3,6)
Link (2,4)————————————
Path between node (1,4)
Path between node (1,5)
Path between node (1,6)
Path between node (2,4)
Path between node (2,5)
Path between node (2,6)
Path between node (3,4)
Path between node (3,5)
Path between node (3,6)
Link (3,5)————————————
Path between node (1,4)
Path between node (1,5)
Path between node (1,6)
Path between node (2,4)
Path between node (2,5)
Path between node (2,6)
Path between node (3,4)
Path between node (3,5)
Path between node (3,6)
Link (4,5)————————————
Path between node (1,4)

Path between node (1,5)
Path between node (2,4)
Path between node (2,5)
Path between node (3,4)
Path between node (3,5)
Path between node (4,5)
Path between node (4,6)
Path between node (5,6)
Link (4,6)——————————
Path between node (1,6)
Path between node (2,6)
Path between node (3,6)
Path between node (4,5)
Path between node (4,6)
Path between node (5,6)
Link (5,6)——————————
Path between node (1,6)
Path between node (2,6)
Path between node (3,6)
Path between node (4,5)
Path between node (4,6)
Path between node (5,6)

5.8.1.3 Scenario 3

5.8.1.4 Scenario 4

5.8.2 ILP using LPSolve

5.8.2.1 Scenario 1

/* Objective function */
min: +C1 +C2 +C3 +C4 +C5 +C6 +C7 +C8 +C9 +C10 +C11 +C12 +C13 +C14 +C15 +C16 +C17 +C18 +C19 +C20 +C21 +C22 +C23 +C24 +C25 +C26 +C27 +C28 +C29 +C30 +C31 +C32 +C33 +C34 +C35 +C36 +C37 +C38 +C39 +C40 +C41 +C42 +C43 +C44 +C45 +C46 +C47 +C48 +C49 +C50 +C51 +C52 +C53 +C54 +C55 +C56 +C57 +C58 +C59 +C60 +C61 +C62 +C63 +C64 +C65 +C66 +C67 +C68 +C69 +C70 +C71 +C72 +C73 +C74 +C75 +C76 +C77 +C78 +C79 +C80 +C81 +C82 +C83 +C84 +C85 +C86 +C87 +C88 +C89 +C90 +C91 +C92 +C93 +C94 +C95 +C96 +C97 +C98 +C99 +C100 +C101 +C102 +C103 +C104 +C105 +C106 +C107 +C108 +C109 +C110 +C111 +C112 +C113 +C114 +C115 +C116 +C117 +C118 +C119 +C120 +C121 +C122 +C123 +C124 +C125 +C126 +C127 +C128 +C129 +C130 +C131 +C132 +C133 +C134 +C135 +C136 +C137 +C138 +C139 +C140 +C141 +C142 +C143 +C144 +C145 +C146 +C147 +C148 +C149 +C150 +C151 +C152 +C153 +C154 +C155 +C156 +C157 +C158 +C159 +C160 +C161 +C162

+C163 +C164 +C165 +C166 +C167 +C168 +C169 +C170 +C171 +C172 +C173 +C174 +C175
+C176 +C177 +C178 +C179 +C180 +C181 +C182 +C183 +C184 +C185 +C186 +C187 +C188
+C189 +C190 +C191 +C192 +C193 +C194 +C195 +C196 +C197 +C198 +C199 +C200 +C201
+C202 +C203 +C204 +C205 +C206 +C207 +C208 +C209 +C210 +C211 +C212 +C213 +C214
+C215 +C216 +C217 +C218 +C219 +C220 +C221 +C222 +C223 +C224 +C225 +C226 +C227
+C228 +C229 +C230 +C231 +C232 +C233 +C234 +C235 +C236 +C237 +C238 +C239 +C240
+C241 +C242 +C243 +C244 +C245 +C246 +C247 +C248 +C249 +C250 +C251 +C252 +C253
+C254 +C255 +C256 +C257 +C258 +C259 +C260 +C261 +C262 +C263 +C264 +C265 +C266
+C267 +C268 +C269 +C270 +C271 +C272 +C273 +C274 +C275 +C276 +C277 +C278 +C279
+C280 +C281 +C282 +C283 +C284 +C285 +C286 +C287 +C288 +C289 +C290 +C291 +C292
+C293 +C294 +C295 +C296 +C297 +C298 +C299 +C300 +C301 +C302 +C303 +C304 +C305
+C306 +C307 +C308 +C309 +C310 +C311 +C312 +C313 +C314 +C315 +C316 +C317 +C318
+C319 +C320 +C321 +C322 +C323 +C324 +C325 +C326 +C327 +C328 +C329 +C330 +C331
+C332 +C333 +C334 +C335 +C336 +C337 +C338 +C339 +C340 +C341 +C342 +C343 +C344
+C345 +C346 +C347 +C348 +C349 +C350 +C351 +C352 +C353 +C354 +C355 +C356 +C357
+C358 +C359 +C360 +C361 +C362 +C363 +C364 +C365 +C366 +C367 +C368 +C369 +C370
+C371 +C372 +C373 +C374 +C375 +C376 +C377 +C378 +C379 +C380 +C381 +C382 +C383
+C384 +C385 +C386 +C387 +C388 +C389 +C390 +C391 +C392 +C393 +C394 +C395 +C396
+C397 +C398 +C399 +C400 +C401 +C402 +C403 +C404 +C405 +C406 +C407 +C408 +C409
+C410 +C411 +C412 +C413 +C414 +C415 +C416 +C417 +C418 +C419 +C420 +C421 +C422
+C423 +C424 +C425 +C426 +C427 +C428 +C429 +C430 +C431 +C432 +C433 +C434 +C435
+C436 +C437 +C438 +C439 +C440 +C441 +C442 +C443 +C444 +C445 +C446 +C447 +C448
+C449 +C450 +C451 +C452 +C453 +C454 +C455 +C456 +C457 +C458 +C459 +C460 +C461
+C462 +C463 +C464 +C465 +C466 +C467 +C468 +C469 +C470 +C471 +C472 +C473 +C474
+C475 +C476 +C477 +C478 +C479 +C480 +C481 +C482 +C483 +C484 +C485 +C486 +C487
+C488 +C489 +C490 +C491 +C492 +C493 +C494 +C495 +C496 +C497 +C498 +C499 +C500
+C501 +C502 +C503 +C504 +C505 +C506 +C507 +C508 +C509 +C510 +C511 +C512 +C513
+C514 +C515 +C516 +C517 +C518 +C519 +C520 +C521 +C522 +C523 +C524 +C525 +C526
+C527 +C528 +C529 +C530 +C531 +C532 +C533 +C534 +C535 +C536 +C537 +C538 +C539
+C540 +C541 +C542 +C543 +C544 +C545 +C546 +C547 +C548 +C549 +C550 +C551 +C552
+C553 +C554 +C555 +C556 +C557 +C558 +C559 +C560 +C561 +C562 +C563 +C564 +C565
+C566 +C567 +C568 +C569 +C570 +C571 +C572 +C573 +C574 +C575 +C576 +C577 +C578
+C579 +C580 +C581 +C582 +C583 +C584 +C585 +C586 +C587 +C588 +C589 +C590 +C591
+C592 +C593 +C594 +C595 +C596 +C597 +C598 +C599 +C600 +C601 +C602 +C603 +C604
+C605 +C606 +C607 +C608 +C609 +C610 +C611 +C612 +C613 +C614 +C615 +C616 +C617
+C618 +C619 +C620 +C621 +C622 +C623 +C624 +C625 +C626 +C627 +C628 +C629 +C630
+C631 +C632 +C633 +C634 +C635 +C636 +C637 +C638 +C639 +C640 +C641 +C642 +C643
+C644 +C645 +C646 +C647 +C648 +C649 +C650 +C651 +C652 +C653 +C654 +C655 +C656
+C657 +C658 +C659 +C660 +C661 +C662 +C663 +C664 +C665 +C666 +C667 +C668 +C669
+C670 +C671 +C672 +C673 +C674 +C675 +C676 +C677 +C678 +C679 +C680 +C681 +C682
+C683 +C684 +C685 +C686 +C687 +C688 +C689 +C690 +C691 +C692 +C693 +C694 +C695
+C696 +C697 +C698 +C699 +C700 +C701 +C702 +C703 +C704 +C705 +C706 +C707 +C708

+C709 +C710 +C711 +C712 +C713 +C714 +C715 +C716 +C717 +C718 +C719 +C720 +C721 +C722 +C723 +C724 +C725 +C726 +C727 +C728 +C729 +C730 +C731 +C732 +C733 +C734 +C735 +C736 +C737 +C738 +C739 +C740 +C741 +C742 +C743 +C744 +C745 +C746 +C747 +C748 +C749 +C750 +C751 +C752 +C753 +C754 +C755 +C756 +C757 +C758 +C759 +C760 +C761 +C762 +C763 +C764 +C765 +C766 +C767 +C768 +C769 +C770 +C771 +C772 +C773 +C774 +C775 +C776 +C777 +C778 +C779 +C780 +C781 +C782 +C783 +C784 +C785 +C786 +C787 +C788 +C789 +C790 +C791 +C792 +C793 +C794 +C795 +C796 +C797 +C798 +C799 +C800 +C801 +C802 +C803 +C804 +C805 +C806 +C807 +C808 +C809 +C810 +C811 +C812 +C813 +C814 +C815 +C816 +C817 +C818 +C819 +C820 +C821 +C822 +C823 +C824 +C825 +C826 +C827 +C828 +C829 +C830 +C831 +C832 +C833 +C834 +C835 +C836 +C837 +C838 +C839 +C840 +C841 +C842 +C843 +C844 +C845 +C846 +C847 +C848 +C849 +C850 +C851 +C852 +C853 +C854 +C855 +C856 +C857 +C858 +C859 +C860 +C861 +C862 +C863 +C864 +C865 +C866 +C867 +C868 +C869 +C870 +C871 +C872 +C873 +C874 +C875 +C876 +C877 +C878 +C879 +C880 +C881 +C882 +C883 +C884 +C885 +C886 +C887 +C888 +C889 +C890 +C891 +C892 +C893 +C894 +C895 +C896 +C897 +C898 +C899 +C900 +C901 +C902 +C903 +C904 +C905 +C906 +C907 +C908 +C909 +C910 +C911 +C912 +C913 +C914 +C915;

/* Constraints */
+C1 +C2 = 2; +C1 +C12 +C17 = 2; -C2 +C12 +C14 -C23 = 0; +C17 +C19 +C20 -C24 -C29 = 0; -C14 -C19 +C23 +C24 +C25 -C30 = 0; -C20 -C25 +C29 +C30 = 0; +C31 +C32 = 2; -C31 +C37 +C38 -C47 = 0; +C32 +C37 +C53 = 2; -C38 +C47 +C49 +C50 -C54 -C59 = 0; -C49 +C53 +C54 +C55 -C60 = 0; -C50 -C55 +C59 +C60 = 0; +C61 +C62 = 2; -C61 +C67 +C68 -C72 = 0; -C62 -C67 +C72 +C74 -C83 = 0; +C68 +C84 +C89 = 2; -C74 +C83 +C84 +C85 -C90 = 0; -C85 +C89 +C90 = 0; +C91 +C92 = 2; -C91 +C97 +C98 -C102 -C107 = 0; -C92 -C97 +C102 +C104 = 0; -C98 +C107 +C109 +C110 -C119 = 0; +C104 +C109 +C120 = 2; -C110 +C119 +C120 = 0; +C121 +C122 = 2; -C121 +C127 +C128 -C132 -C137 = 0; -C122 -C127 +C132 +C134 -C143 = 0; -C128 +C137 +C139 +C140 -C144 = 0; -C134 -C139 +C143 +C144 +C145 = 0; +C140 +C145 = 2; +C152 -C156 = 0; +C156 +C157 +C158 = 2; +C152 +C157 +C173 = 2; -C158 +C169 +C170 -C174 -C179 = 0; -C169 +C173 +C174 +C175 -C180 = 0; -C170 -C175 +C179 +C180 = 0; +C182 -C186 -C191 = 0; +C186 +C187 +C188 = 2; -C182 -C187 +C191 +C194 -C203 = 0; +C188 +C204 +C209 = 2; -C194 +C203 +C204 +C205 -C210 = 0; -C205 +C209 +C210 = 0; +C212 -C216 -C221 = 0; +C216 +C217 +C218 = 2; -C212 -C217 +C221 +C224 = 0; -C218 +C229 +C230 -C239 = 0; +C224 +C229 +C240 = 2; -C230 +C239 +C240 = 0; +C242 -C246 -C251 = 0; +C246 +C247 +C248 = 2; -C242 -C247 +C251 +C254 -C263 = 0; -C248 +C259 +C260 -C264 = 0; -C254 -C259 +C263 +C264 +C265 = 0; +C260 +C265 = 2; +C271 -C276 -C281 = 0; -C271 +C276 +C278 -C282 = 0; +C281 +C282 +C284 = 2; +C278 +C294 +C299 = 2; -C284 +C294 +C295 -C300 = 0; -C295 +C299 +C300 = 0; +C301 -C306 -C311 = 0; -C301 +C306 +C308 -C312 -C317 = 0; +C311 +C312 +C314 = 2; -C308 +C317 +C319 +C320 -C329 = 0; +C314 +C319 +C330 = 2; -C320 +C329 +C330 = 0; +C331 -C336 -C341 = 0; -C331 +C336 +C338 -C342 -C347 = 0; +C341 +C342 +C344 = 2; -C338 +C347 +C349 +C350 -C354 = 0; -C344 -C349 +C354 +C355 = 0; +C350 +C355 = 2; +C361 +C362 -C366 -C371 = 0; -C361 +C366 +C367 -C372 -C377 = 0; -C362 -C367 +C371 +C372 +C374 = 0; +C377 +C379 +C380 = 2; +C374 +C379 +C390 = 2; -C380 +C390 = 0; +C391

+C392 -C396 -C401 = 0; -C391 +C396 +C397 -C402 -C407 = 0; -C392 -C397 +C401 +C402 +C404 -C413 = 0; +C407 +C409 +C410 = 2; -C404 -C409 +C413 +C415 = 0; +C410 +C415 = 2; +C421 +C422 -C426 -C431 = 0; -C421 +C426 +C427 +C428 -C432 -C437 = 0; -C422 -C427 +C431 +C432 -C443 = 0; -C428 +C437 +C440 -C444 = 0; +C443 +C444 +C445 = 2; +C440 +C445 = 2; +21.25 C1 +21.25 C6 +21.25 C31 +21.25 C36 +18.75 C61 +18.75 C66 +1.25 C91 +1.25 C96 +16.25 C121 +16.25 C126 +40 C151 +40 C156 +8.75 C181 +8.75 C186 +18.75 C211 +18.75 C216 +142.5 C241 +142.5 C246 +13.75 C271 +13.75 C276 +57.5 C301 +57.5 C306 +1.25 C331 +1.25 C336 +13.75 C361 +13.75 C366 +8.75 C391 +8.75 C396 +116.25 C421 +116.25 C426 -100 C901 <= 0; +21.25 C2 +21.25 C11 +21.25 C32 +21.25 C41 +18.75 C62 +18.75 C71 +1.25 C92 +1.25 C101 +16.25 C122 +16.25 C131 +40 C152 +40 C161 +8.75 C182 +8.75 C191 +18.75 C212 +18.75 C221 +142.5 C242 +142.5 C251 +13.75 C272 +13.75 C281 +57.5 C302 +57.5 C311 +1.25 C332 +1.25 C341 +13.75 C362 +13.75 C371 +8.75 C392 +8.75 C401 +116.25 C422 +116.25 C431 -100 C902 <= 0; +21.25 C3 +21.25 C16 +21.25 C33 +21.25 C46 +18.75 C63 +18.75 C76 +1.25 C93 +1.25 C106 +16.25 C123 +16.25 C136 +40 C153 +40 C166 +8.75 C183 +8.75 C196 +18.75 C213 +18.75 C226 +142.5 C243 +142.5 C256 +13.75 C273 +13.75 C286 +57.5 C303 +57.5 C316 +1.25 C333 +1.25 C346 +13.75 C363 +13.75 C376 +8.75 C393 +8.75 C406 +116.25 C423 +116.25 C436 <= 0; +21.25 C4 +21.25 C21 +21.25 C34 +21.25 C51 +18.75 C64 +18.75 C81 +1.25 C94 +1.25 C111 +16.25 C124 +16.25 C141 +40 C154 +40 C171 +8.75 C184 +8.75 C201 +18.75 C214 +18.75 C231 +142.5 C244 +142.5 C261 +13.75 C274 +13.75 C291 +57.5 C304 +57.5 C321 +1.25 C334 +1.25 C351 +13.75 C364 +13.75 C381 +8.75 C394 +8.75 C411 +116.25 C424 +116.25 C441 <= 0; +21.25 C5 +21.25 C26 +21.25 C35 +21.25 C56 +18.75 C65 +18.75 C86 +1.25 C95 +1.25 C116 +16.25 C125 +16.25 C146 +40 C155 +40 C176 +8.75 C185 +8.75 C206 +18.75 C215 +18.75 C236 +142.5 C245 +142.5 C266 +13.75 C275 +13.75 C296 +57.5 C305 +57.5 C326 +1.25 C335 +1.25 C356 +13.75 C365 +13.75 C386 +8.75 C395 +8.75 C416 +116.25 C425 +116.25 C446 <= 0; +21.25 C7 +21.25 C12 +21.25 C37 +21.25 C42 +18.75 C67 +18.75 C72 +1.25 C97 +1.25 C102 +16.25 C127 +16.25 C132 +40 C157 +40 C162 +8.75 C187 +8.75 C192 +18.75 C217 +18.75 C222 +142.5 C247 +142.5 C252 +13.75 C277 +13.75 C282 +57.5 C307 +57.5 C312 +1.25 C337 +1.25 C342 +13.75 C367 +13.75 C372 +8.75 C397 +8.75 C402 +116.25 C427 +116.25 C432 -100 C906 <= 0; +21.25 C8 +21.25 C17 +21.25 C38 +21.25 C47 +18.75 C68 +18.75 C77 +1.25 C98 +1.25 C107 +16.25 C128 +16.25 C137 +40 C158 +40 C167 +8.75 C188 +8.75 C197 +18.75 C218 +18.75 C227 +142.5 C248 +142.5 C257 +13.75 C278 +13.75 C287 +57.5 C308 +57.5 C317 +1.25 C338 +1.25 C347 +13.75 C368 +13.75 C377 +8.75 C398 +8.75 C407 +116.25 C428 +116.25 C437 -100 C907 <= 0; +21.25 C9 +21.25 C22 +21.25 C39 +21.25 C52 +18.75 C69 +18.75 C82 +1.25 C99 +1.25 C112 +16.25 C129 +16.25 C142 +40 C159 +40 C172 +8.75 C189 +8.75 C202 +18.75 C219 +18.75 C232 +142.5 C249 +142.5 C262 +13.75 C279 +13.75 C292 +57.5 C309 +57.5 C322 +1.25 C339 +1.25 C352 +13.75 C369 +13.75 C382 +8.75 C399 +8.75 C412 +116.25 C429 +116.25 C442 <= 0; +21.25 C10 +21.25 C27 +21.25 C40 +21.25 C57 +18.75 C70 +18.75 C87 +1.25 C100 +1.25 C117 +16.25 C130 +16.25 C147 +40 C160 +40 C177 +8.75 C190 +8.75 C207 +18.75 C220 +18.75 C237 +142.5 C250 +142.5 C267 +13.75 C280 +13.75 C297 +57.5 C310 +57.5 C327 +1.25 C340 +1.25 C357 +13.75 C370 +13.75 C387 +8.75 C400 +8.75 C417 +116.25 C430 +116.25 C447 <= 0; +21.25 C13 +21.25 C18 +21.25 C43 +21.25 C48 +18.75 C73 +18.75 C78 +1.25 C103 +1.25 C108 +16.25 C133 +16.25 C138 +40 C163 +40 C168 +8.75 C193 +8.75 C198 +18.75 C223 +18.75

C228 +142.5 C253 +142.5 C258 +13.75 C283 +13.75 C288 +57.5 C313 +57.5 C318 +1.25 C343 +1.25 C348 +13.75 C373 +13.75 C378 +8.75 C403 +8.75 C408 +116.25 C433 +116.25 C438 <= 0; +21.25 C14 +21.25 C23 +21.25 C44 +21.25 C53 +18.75 C74 +18.75 C83 +1.25 C104 +1.25 C113 +16.25 C134 +16.25 C143 +40 C164 +40 C173 +8.75 C194 +8.75 C203 +18.75 C224 +18.75 C233 +142.5 C254 +142.5 C263 +13.75 C284 +13.75 C293 +57.5 C314 +57.5 C323 +1.25 C344 +1.25 C353 +13.75 C374 +13.75 C383 +8.75 C404 +8.75 C413 +116.25 C434 +116.25 C443 -100 C911 <= 0; +21.25 C15 +21.25 C28 +21.25 C45 +21.25 C58 +18.75 C75 +18.75 C88 +1.25 C105 +1.25 C118 +16.25 C135 +16.25 C148 +40 C165 +40 C178 +8.75 C195 +8.75 C208 +18.75 C225 +18.75 C238 +142.5 C255 +142.5 C268 +13.75 C285 +13.75 C298 +57.5 C315 +57.5 C328 +1.25 C345 +1.25 C358 +13.75 C375 +13.75 C388 +8.75 C405 +8.75 C418 +116.25 C435 +116.25 C448 <= 0; +21.25 C19 +21.25 C24 +21.25 C49 +21.25 C54 +18.75 C79 +18.75 C84 +1.25 C109 +1.25 C114 +16.25 C139 +16.25 C144 +40 C169 +40 C174 +8.75 C199 +8.75 C204 +18.75 C229 +18.75 C234 +142.5 C259 +142.5 C264 +13.75 C289 +13.75 C294 +57.5 C319 +57.5 C324 +1.25 C349 +1.25 C354 +13.75 C379 +13.75 C384 +8.75 C409 +8.75 C414 +116.25 C439 +116.25 C444 -100 C913 <= 0; +21.25 C20 +21.25 C29 +21.25 C50 +21.25 C59 +18.75 C80 +18.75 C89 +1.25 C110 +1.25 C119 +16.25 C140 +16.25 C149 +40 C170 +40 C179 +8.75 C200 +8.75 C209 +18.75 C230 +18.75 C239 +142.5 C260 +142.5 C269 +13.75 C290 +13.75 C299 +57.5 C320 +57.5 C329 +1.25 C350 +1.25 C359 +13.75 C380 +13.75 C389 +8.75 C410 +8.75 C419 +116.25 C440 +116.25 C449 -100 C914 <= 0; +21.25 C25 +21.25 C30 +21.25 C55 +21.25 C60 +18.75 C85 +18.75 C90 +1.25 C115 +1.25 C120 +16.25 C145 +16.25 C150 +40 C175 +40 C180 +8.75 C205 +8.75 C210 +18.75 C235 +18.75 C240 +142.5 C265 +142.5 C270 +13.75 C295 +13.75 C300 +57.5 C325 +57.5 C330 +1.25 C355 +1.25 C360 +13.75 C385 +13.75 C390 +8.75 C415 +8.75 C420 +116.25 C445 +116.25 C450 -100 C915 <= 0;
R106: +C901 <= 80; R107: +C902 <= 80; R108: +C903 <= 80; R109: +C904 <= 80; R110: +C905 <= 80; R111: +C906 <= 80; R112: +C907 <= 80; R113: +C908 <= 80; R114: +C909 <= 80; R115: +C910 <= 80; R116: +C911 <= 80; R117: +C912 <= 80; R118: +C913 <= 80; R119: +C914 <= 80; R120: +C915 <= 80;

/* Variable bounds */
C1 <= 1; C2 <= 1; C3 <= 1; C4 <= 1; C5 <= 1; C6 <= 1; C7 <= 1; C8 <= 1; C9 <= 1; C10 <= 1; C11 <= 1; C12 <= 1; C13 <= 1; C14 <= 1; C15 <= 1; C16 <= 1; C17 <= 1; C18 <= 1; C19 <= 1; C20 <= 1; C21 <= 1; C22 <= 1; C23 <= 1; C24 <= 1; C25 <= 1; C26 <= 1; C27 <= 1; C28 <= 1; C29 <= 1; C30 <= 1; C31 <= 1; C32 <= 1; C33 <= 1; C34 <= 1; C35 <= 1; C36 <= 1; C37 <= 1; C38 <= 1; C39 <= 1; C40 <= 1; C41 <= 1; C42 <= 1; C43 <= 1; C44 <= 1; C45 <= 1; C46 <= 1; C47 <= 1; C48 <= 1; C49 <= 1; C50 <= 1; C51 <= 1; C52 <= 1; C53 <= 1; C54 <= 1; C55 <= 1; C56 <= 1; C57 <= 1; C58 <= 1; C59 <= 1; C60 <= 1; C61 <= 1; C62 <= 1; C63 <= 1; C64 <= 1; C65 <= 1; C66 <= 1; C67 <= 1; C68 <= 1; C69 <= 1; C70 <= 1; C71 <= 1; C72 <= 1; C73 <= 1; C74 <= 1; C75 <= 1; C76 <= 1; C77 <= 1; C78 <= 1; C79 <= 1; C80 <= 1; C81 <= 1; C82 <= 1; C83 <= 1; C84 <= 1; C85 <= 1; C86 <= 1; C87 <= 1; C88 <= 1; C89 <= 1; C90 <= 1; C91 <= 1; C92 <= 1; C93 <= 1; C94 <= 1; C95 <= 1; C96 <= 1; C97 <= 1; C98 <= 1; C99 <= 1; C100 <= 1; C101 <= 1; C102 <= 1; C103 <= 1; C104 <= 1; C105 <= 1; C106 <= 1; C107 <= 1; C108 <= 1; C109 <= 1; C110 <= 1; C111 <= 1; C112 <= 1; C113 <= 1; C114 <= 1; C115 <= 1; C116 <= 1;

C117 <= 1; C118 <= 1; C119 <= 1; C120 <= 1; C121 <= 1; C122 <= 1; C123 <= 1; C124 <= 1;
C125 <= 1; C126 <= 1; C127 <= 1; C128 <= 1; C129 <= 1; C130 <= 1; C131 <= 1; C132 <= 1;
C133 <= 1; C134 <= 1; C135 <= 1; C136 <= 1; C137 <= 1; C138 <= 1; C139 <= 1; C140 <= 1;
C141 <= 1; C142 <= 1; C143 <= 1; C144 <= 1; C145 <= 1; C146 <= 1; C147 <= 1; C148 <= 1;
C149 <= 1; C150 <= 1; C151 <= 1; C152 <= 1; C153 <= 1; C154 <= 1; C155 <= 1; C156 <= 1;
C157 <= 1; C158 <= 1; C159 <= 1; C160 <= 1; C161 <= 1; C162 <= 1; C163 <= 1; C164 <= 1;
C165 <= 1; C166 <= 1; C167 <= 1; C168 <= 1; C169 <= 1; C170 <= 1; C171 <= 1; C172 <= 1;
C173 <= 1; C174 <= 1; C175 <= 1; C176 <= 1; C177 <= 1; C178 <= 1; C179 <= 1; C180 <= 1;
C181 <= 1; C182 <= 1; C183 <= 1; C184 <= 1; C185 <= 1; C186 <= 1; C187 <= 1; C188 <= 1;
C189 <= 1; C190 <= 1; C191 <= 1; C192 <= 1; C193 <= 1; C194 <= 1; C195 <= 1; C196 <= 1;
C197 <= 1; C198 <= 1; C199 <= 1; C200 <= 1; C201 <= 1; C202 <= 1; C203 <= 1; C204 <= 1;
C205 <= 1; C206 <= 1; C207 <= 1; C208 <= 1; C209 <= 1; C210 <= 1; C211 <= 1; C212 <= 1;
C213 <= 1; C214 <= 1; C215 <= 1; C216 <= 1; C217 <= 1; C218 <= 1; C219 <= 1; C220 <= 1;
C221 <= 1; C222 <= 1; C223 <= 1; C224 <= 1; C225 <= 1; C226 <= 1; C227 <= 1; C228 <= 1;
C229 <= 1; C230 <= 1; C231 <= 1; C232 <= 1; C233 <= 1; C234 <= 1; C235 <= 1; C236 <= 1;
C237 <= 1; C238 <= 1; C239 <= 1; C240 <= 1; C241 <= 1; C242 <= 1; C243 <= 1; C244 <= 1;
C245 <= 1; C246 <= 1; C247 <= 1; C248 <= 1; C249 <= 1; C250 <= 1; C251 <= 1; C252 <= 1;
C253 <= 1; C254 <= 1; C255 <= 1; C256 <= 1; C257 <= 1; C258 <= 1; C259 <= 1; C260 <= 1;
C261 <= 1; C262 <= 1; C263 <= 1; C264 <= 1; C265 <= 1; C266 <= 1; C267 <= 1; C268 <= 1;
C269 <= 1; C270 <= 1; C271 <= 1; C272 <= 1; C273 <= 1; C274 <= 1; C275 <= 1; C276 <= 1;
C277 <= 1; C278 <= 1; C279 <= 1; C280 <= 1; C281 <= 1; C282 <= 1; C283 <= 1; C284 <= 1;
C285 <= 1; C286 <= 1; C287 <= 1; C288 <= 1; C289 <= 1; C290 <= 1; C291 <= 1; C292 <= 1;
C293 <= 1; C294 <= 1; C295 <= 1; C296 <= 1; C297 <= 1; C298 <= 1; C299 <= 1; C300 <= 1;
C301 <= 1; C302 <= 1; C303 <= 1; C304 <= 1; C305 <= 1; C306 <= 1; C307 <= 1; C308 <= 1;
C309 <= 1; C310 <= 1; C311 <= 1; C312 <= 1; C313 <= 1; C314 <= 1; C315 <= 1; C316 <= 1;
C317 <= 1; C318 <= 1; C319 <= 1; C320 <= 1; C321 <= 1; C322 <= 1; C323 <= 1; C324 <= 1;
C325 <= 1; C326 <= 1; C327 <= 1; C328 <= 1; C329 <= 1; C330 <= 1; C331 <= 1; C332 <= 1;
C333 <= 1; C334 <= 1; C335 <= 1; C336 <= 1; C337 <= 1; C338 <= 1; C339 <= 1; C340 <= 1;
C341 <= 1; C342 <= 1; C343 <= 1; C344 <= 1; C345 <= 1; C346 <= 1; C347 <= 1; C348 <= 1;
C349 <= 1; C350 <= 1; C351 <= 1; C352 <= 1; C353 <= 1; C354 <= 1; C355 <= 1; C356 <= 1;
C357 <= 1; C358 <= 1; C359 <= 1; C360 <= 1; C361 <= 1; C362 <= 1; C363 <= 1; C364 <= 1;
C365 <= 1; C366 <= 1; C367 <= 1; C368 <= 1; C369 <= 1; C370 <= 1; C371 <= 1; C372 <= 1;
C373 <= 1; C374 <= 1; C375 <= 1; C376 <= 1; C377 <= 1; C378 <= 1; C379 <= 1; C380 <= 1;
C381 <= 1; C382 <= 1; C383 <= 1; C384 <= 1; C385 <= 1; C386 <= 1; C387 <= 1; C388 <= 1;
C389 <= 1; C390 <= 1; C391 <= 1; C392 <= 1; C393 <= 1; C394 <= 1; C395 <= 1; C396 <= 1;
C397 <= 1; C398 <= 1; C399 <= 1; C400 <= 1; C401 <= 1; C402 <= 1; C403 <= 1; C404 <= 1;
C405 <= 1; C406 <= 1; C407 <= 1; C408 <= 1; C409 <= 1; C410 <= 1; C411 <= 1; C412 <= 1;
C413 <= 1; C414 <= 1; C415 <= 1; C416 <= 1; C417 <= 1; C418 <= 1; C419 <= 1; C420 <= 1;
C421 <= 1; C422 <= 1; C423 <= 1; C424 <= 1; C425 <= 1; C426 <= 1; C427 <= 1; C428 <= 1;
C429 <= 1; C430 <= 1; C431 <= 1; C432 <= 1; C433 <= 1; C434 <= 1; C435 <= 1; C436 <= 1;
C437 <= 1; C438 <= 1; C439 <= 1; C440 <= 1; C441 <= 1; C442 <= 1; C443 <= 1; C444 <= 1;
C445 <= 1; C446 <= 1; C447 <= 1; C448 <= 1; C449 <= 1; C450 <= 1; C451 <= 1; C452 <= 1;

C453 <= 1; C454 <= 1; C455 <= 1; C456 <= 1; C457 <= 1; C458 <= 1; C459 <= 1; C460 <= 1;
C461 <= 1; C462 <= 1; C463 <= 1; C464 <= 1; C465 <= 1; C466 <= 1; C467 <= 1; C468 <= 1;
C469 <= 1; C470 <= 1; C471 <= 1; C472 <= 1; C473 <= 1; C474 <= 1; C475 <= 1; C476 <= 1;
C477 <= 1; C478 <= 1; C479 <= 1; C480 <= 1; C481 <= 1; C482 <= 1; C483 <= 1; C484 <= 1;
C485 <= 1; C486 <= 1; C487 <= 1; C488 <= 1; C489 <= 1; C490 <= 1; C491 <= 1; C492 <= 1;
C493 <= 1; C494 <= 1; C495 <= 1; C496 <= 1; C497 <= 1; C498 <= 1; C499 <= 1; C500 <= 1;
C501 <= 1; C502 <= 1; C503 <= 1; C504 <= 1; C505 <= 1; C506 <= 1; C507 <= 1; C508 <= 1;
C509 <= 1; C510 <= 1; C511 <= 1; C512 <= 1; C513 <= 1; C514 <= 1; C515 <= 1; C516 <= 1;
C517 <= 1; C518 <= 1; C519 <= 1; C520 <= 1; C521 <= 1; C522 <= 1; C523 <= 1; C524 <= 1;
C525 <= 1; C526 <= 1; C527 <= 1; C528 <= 1; C529 <= 1; C530 <= 1; C531 <= 1; C532 <= 1;
C533 <= 1; C534 <= 1; C535 <= 1; C536 <= 1; C537 <= 1; C538 <= 1; C539 <= 1; C540 <= 1;
C541 <= 1; C542 <= 1; C543 <= 1; C544 <= 1; C545 <= 1; C546 <= 1; C547 <= 1; C548 <= 1;
C549 <= 1; C550 <= 1; C551 <= 1; C552 <= 1; C553 <= 1; C554 <= 1; C555 <= 1; C556 <= 1;
C557 <= 1; C558 <= 1; C559 <= 1; C560 <= 1; C561 <= 1; C562 <= 1; C563 <= 1; C564 <= 1;
C565 <= 1; C566 <= 1; C567 <= 1; C568 <= 1; C569 <= 1; C570 <= 1; C571 <= 1; C572 <= 1;
C573 <= 1; C574 <= 1; C575 <= 1; C576 <= 1; C577 <= 1; C578 <= 1; C579 <= 1; C580 <= 1;
C581 <= 1; C582 <= 1; C583 <= 1; C584 <= 1; C585 <= 1; C586 <= 1; C587 <= 1; C588 <= 1;
C589 <= 1; C590 <= 1; C591 <= 1; C592 <= 1; C593 <= 1; C594 <= 1; C595 <= 1; C596 <= 1;
C597 <= 1; C598 <= 1; C599 <= 1; C600 <= 1; C601 <= 1; C602 <= 1; C603 <= 1; C604 <= 1;
C605 <= 1; C606 <= 1; C607 <= 1; C608 <= 1; C609 <= 1; C610 <= 1; C611 <= 1; C612 <= 1;
C613 <= 1; C614 <= 1; C615 <= 1; C616 <= 1; C617 <= 1; C618 <= 1; C619 <= 1; C620 <= 1;
C621 <= 1; C622 <= 1; C623 <= 1; C624 <= 1; C625 <= 1; C626 <= 1; C627 <= 1; C628 <= 1;
C629 <= 1; C630 <= 1; C631 <= 1; C632 <= 1; C633 <= 1; C634 <= 1; C635 <= 1; C636 <= 1;
C637 <= 1; C638 <= 1; C639 <= 1; C640 <= 1; C641 <= 1; C642 <= 1; C643 <= 1; C644 <= 1;
C645 <= 1; C646 <= 1; C647 <= 1; C648 <= 1; C649 <= 1; C650 <= 1; C651 <= 1; C652 <= 1;
C653 <= 1; C654 <= 1; C655 <= 1; C656 <= 1; C657 <= 1; C658 <= 1; C659 <= 1; C660 <= 1;
C661 <= 1; C662 <= 1; C663 <= 1; C664 <= 1; C665 <= 1; C666 <= 1; C667 <= 1; C668 <= 1;
C669 <= 1; C670 <= 1; C671 <= 1; C672 <= 1; C673 <= 1; C674 <= 1; C675 <= 1; C676 <= 1;
C677 <= 1; C678 <= 1; C679 <= 1; C680 <= 1; C681 <= 1; C682 <= 1; C683 <= 1; C684 <= 1;
C685 <= 1; C686 <= 1; C687 <= 1; C688 <= 1; C689 <= 1; C690 <= 1; C691 <= 1; C692 <= 1;
C693 <= 1; C694 <= 1; C695 <= 1; C696 <= 1; C697 <= 1; C698 <= 1; C699 <= 1; C700 <= 1;
C701 <= 1; C702 <= 1; C703 <= 1; C704 <= 1; C705 <= 1; C706 <= 1; C707 <= 1; C708 <= 1;
C709 <= 1; C710 <= 1; C711 <= 1; C712 <= 1; C713 <= 1; C714 <= 1; C715 <= 1; C716 <= 1;
C717 <= 1; C718 <= 1; C719 <= 1; C720 <= 1; C721 <= 1; C722 <= 1; C723 <= 1; C724 <= 1;
C725 <= 1; C726 <= 1; C727 <= 1; C728 <= 1; C729 <= 1; C730 <= 1; C731 <= 1; C732 <= 1;
C733 <= 1; C734 <= 1; C735 <= 1; C736 <= 1; C737 <= 1; C738 <= 1; C739 <= 1; C740 <= 1;
C741 <= 1; C742 <= 1; C743 <= 1; C744 <= 1; C745 <= 1; C746 <= 1; C747 <= 1; C748 <= 1;
C749 <= 1; C750 <= 1; C751 <= 1; C752 <= 1; C753 <= 1; C754 <= 1; C755 <= 1; C756 <= 1;
C757 <= 1; C758 <= 1; C759 <= 1; C760 <= 1; C761 <= 1; C762 <= 1; C763 <= 1; C764 <= 1;
C765 <= 1; C766 <= 1; C767 <= 1; C768 <= 1; C769 <= 1; C770 <= 1; C771 <= 1; C772 <= 1;
C773 <= 1; C774 <= 1; C775 <= 1; C776 <= 1; C777 <= 1; C778 <= 1; C779 <= 1; C780 <= 1;
C781 <= 1; C782 <= 1; C783 <= 1; C784 <= 1; C785 <= 1; C786 <= 1; C787 <= 1; C788 <= 1;

C789 <= 1; C790 <= 1; C791 <= 1; C792 <= 1; C793 <= 1; C794 <= 1; C795 <= 1; C796 <= 1; C797 <= 1; C798 <= 1; C799 <= 1; C800 <= 1; C801 <= 1; C802 <= 1; C803 <= 1; C804 <= 1; C805 <= 1; C806 <= 1; C807 <= 1; C808 <= 1; C809 <= 1; C810 <= 1; C811 <= 1; C812 <= 1; C813 <= 1; C814 <= 1; C815 <= 1; C816 <= 1; C817 <= 1; C818 <= 1; C819 <= 1; C820 <= 1; C821 <= 1; C822 <= 1; C823 <= 1; C824 <= 1; C825 <= 1; C826 <= 1; C827 <= 1; C828 <= 1; C829 <= 1; C830 <= 1; C831 <= 1; C832 <= 1; C833 <= 1; C834 <= 1; C835 <= 1; C836 <= 1; C837 <= 1; C838 <= 1; C839 <= 1; C840 <= 1; C841 <= 1; C842 <= 1; C843 <= 1; C844 <= 1; C845 <= 1; C846 <= 1; C847 <= 1; C848 <= 1; C849 <= 1; C850 <= 1; C851 <= 1; C852 <= 1; C853 <= 1; C854 <= 1; C855 <= 1; C856 <= 1; C857 <= 1; C858 <= 1; C859 <= 1; C860 <= 1; C861 <= 1; C862 <= 1; C863 <= 1; C864 <= 1; C865 <= 1; C866 <= 1; C867 <= 1; C868 <= 1; C869 <= 1; C870 <= 1; C871 <= 1; C872 <= 1; C873 <= 1; C874 <= 1; C875 <= 1; C876 <= 1; C877 <= 1; C878 <= 1; C879 <= 1; C880 <= 1; C881 <= 1; C882 <= 1; C883 <= 1; C884 <= 1; C885 <= 1; C886 <= 1; C887 <= 1; C888 <= 1; C889 <= 1; C890 <= 1; C891 <= 1; C892 <= 1; C893 <= 1; C894 <= 1; C895 <= 1; C896 <= 1; C897 <= 1; C898 <= 1; C899 <= 1; C900 <= 1;

/* Integer definitions */
int C1, C2, C3, C4, C5, C6, C7, C8, C9, C10, C11, C12, C13, C14, C15, C16, C17, C18, C19, C20, C21, C22, C23, C24, C25, C26, C27, C28, C29, C30, C31, C32, C33, C34, C35, C36, C37, C38, C39, C40, C41, C42, C43, C44, C45, C46, C47, C48, C49, C50, C51, C52, C53, C54, C55, C56, C57, C58, C59, C60, C61, C62, C63, C64, C65, C66, C67, C68, C69, C70, C71, C72, C73, C74, C75, C76, C77, C78, C79, C80, C81, C82, C83, C84, C85, C86, C87, C88, C89, C90, C91, C92, C93, C94, C95, C96, C97, C98, C99, C100, C101, C102, C103, C104, C105, C106, C107, C108, C109, C110, C111, C112, C113, C114, C115, C116, C117, C118, C119, C120, C121, C122, C123, C124, C125, C126, C127, C128, C129, C130, C131, C132, C133, C134, C135, C136, C137, C138, C139, C140, C141, C142, C143, C144, C145, C146, C147, C148, C149, C150, C151, C152, C153, C154, C155, C156, C157, C158, C159, C160, C161, C162, C163, C164, C165, C166, C167, C168, C169, C170, C171, C172, C173, C174, C175, C176, C177, C178, C179, C180, C181, C182, C183, C184, C185, C186, C187, C188, C189, C190, C191, C192, C193, C194, C195, C196, C197, C198, C199, C200, C201, C202, C203, C204, C205, C206, C207, C208, C209, C210, C211, C212, C213, C214, C215, C216, C217, C218, C219, C220, C221, C222, C223, C224, C225, C226, C227, C228, C229, C230, C231, C232, C233, C234, C235, C236, C237, C238, C239, C240, C241, C242, C243, C244, C245, C246, C247, C248, C249, C250, C251, C252, C253, C254, C255, C256, C257, C258, C259, C260, C261, C262, C263, C264, C265, C266, C267, C268, C269, C270, C271, C272, C273, C274, C275, C276, C277, C278, C279, C280, C281, C282, C283, C284, C285, C286, C287, C288, C289, C290, C291, C292, C293, C294, C295, C296, C297, C298, C299, C300, C301, C302, C303, C304, C305, C306, C307, C308, C309, C310, C311, C312, C313, C314 ,C315, C316, C317, C318, C319, C320, C321, C322, C323, C324, C325, C326, C327, C328, C329, C330, C331, C332, C333, C334, C335, C336, C337, C338, C339, C340, C341, C342, C343, C344, C345, C346, C347, C348, C349, C350, C351, C352, C353, C354, C355, C356, C357, C358, C359, C360, C361, C362, C363, C364, C365, C366, C367, C368, C369, C370, C371, C372, C373, C374, C375, C376, C377, C378, C379, C380, C381, C382, C383, C384, C385, C386, C387, C388, C389, C390, C391, C392, C393, C394, C395, C396, C397, C398, C399, C400, C401, C402, C403, C404, C405, C406, C407, C408,

C409, C410, C411, C412, C413, C414, C415, C416, C417, C418, C419, C420, C421, C422, C423, C424, C425, C426, C427, C428, C429, C430, C431, C432, C433, C434, C435, C436, C437, C438, C439, C440, C441, C442, C443, C444, C445, C446, C447, C448, C449, C450, C451, C452, C453, C454, C455, C456, C457, C458, C459, C460, C461, C462, C463, C464, C465, C466, C467, C468, C469, C470, C471, C472, C473, C474, C475, C476, C477, C478, C479, C480, C481, C482, C483, C484, C485, C486, C487, C488, C489, C490, C491, C492, C493, C494, C495, C496, C497, C498, C499, C500, C501, C502, C503, C504, C505, C506, C507, C508, C509, C510, C511, C512, C513, C514, C515, C516, C517, C518, C519, C520, C521, C522, C523, C524, C525, C526, C527, C528, C529, C530, C531, C532, C533, C534, C535, C536, C537, C538, C539, C540, C541, C542, C543, C544, C545, C546, C547, C548, C549, C550, C551, C552, C553, C554, C555, C556, C557, C558, C559, C560, C561, C562, C563, C564, C565, C566, C567, C568, C569, C570, C571, C572, C573, C574, C575, C576, C577, C578, C579, C580, C581, C582, C583, C584, C585, C586, C587, C588, C589, C590, C591, C592, C593, C594, C595, C596, C597, C598, C599, C600, C601, C602, C603, C604, C605, C606, C607, C608, C609, C610, C611, C612, C613, C614, C615, C616, C617, C618, C619, C620, C621, C622, C623, C624, C625, C626, C627, C628, C629, C630, C631, C632, C633, C634, C635, C636, C637, C638, C639, C640, C641, C642, C643, C644, C645, C646, C647, C648, C649, C650, C651, C652, C653, C654, C655, C656, C657, C658, C659, C660, C661, C662, C663, C664, C665, C666, C667, C668, C669, C670, C671, C672, C673, C674, C675, C676, C677, C678, C679, C680, C681, C682, C683, C684, C685, C686, C687, C688, C689, C690, C691, C692, C693, C694, C695, C696, C697, C698, C699, C700, C701, C702, C703, C704, C705, C706, C707, C708, C709, C710, C711, C712, C713, C714, C715, C716, C717, C718, C719, C720, C721, C722, C723, C724, C725, C726, C727, C728, C729, C730, C731, C732, C733, C734, C735, C736, C737, C738, C739, C740, C741, C742, C743, C744, C745, C746, C747, C748, C749, C750, C751, C752, C753, C754, C755, C756, C757, C758, C759, C760, C761, C762, C763, C764, C765, C766, C767, C768, C769, C770, C771, C772, C773, C774, C775, C776, C777, C778, C779, C780, C781, C782, C783, C784, C785, C786, C787, C788, C789, C790, C791, C792, C793, C794, C795, C796, C797, C798, C799, C800, C801, C802, C803, C804, C805, C806, C807, C808, C809, C810, C811, C812, C813 ,C814, C815, C816, C817, C818, C819, C820, C821, C822, C823, C824, C825, C826, C827, C828, C829, C830, C831, C832, C833, C834, C835, C836, C837, C838, C839, C840, C841, C842, C843, C844, C845, C846, C847, C848, C849, C850, C851, C852, C853, C854, C855, C856, C857, C858, C859, C860, C861, C862, C863, C864, C865, C866, C867, C868, C869, C870, C871, C872, C873, C874, C875, C876, C877, C878, C879, C880, C881, C882, C883, C884, C885, C886, C887, C888, C889, C890, C891, C892, C893, C894, C895, C896, C897, C898, C899, C900, C901, C902, C903, C904, C905, C906, C907, C908, C909, C910, C911, C912, C913, C914, C915;

5.8.2.2 Scenario 2

/* Objective function */
min: +C1 +C2 +C3 +C4 +C5 +C6 +C7 +C8 +C9 +C10 +C11 +C12 +C13 +C14 +C15 +C16 +C17 +C18 +C19 +C20 +C21 +C22 +C23 +C24 +C25 +C26 +C27 +C28 +C29 +C30 +C31 +C32 +C33 +C34 +C35 +C36 +C37 +C38 +C39 +C40 +C41 +C42 +C43 +C44 +C45 +C46 +C47 +C48 +C49 +C50 +C51 +C52 +C53 +C54 +C55 +C56 +C57 +C58 +C59 +C60 +C61 +C62 +C63 +C64

+C65 +C66 +C67 +C68 +C69 +C70 +C71 +C72 +C73 +C74 +C75 +C76 +C77 +C78 +C79 +C80 +C81 +C82 +C83 +C84 +C85 +C86 +C87 +C88 +C89 +C90 +C91 +C92 +C93 +C94 +C95 +C96 +C97 +C98 +C99 +C100 +C101 +C102 +C103 +C104 +C105 +C106 +C107 +C108 +C109 +C110 +C111 +C112 +C113 +C114 +C115 +C116 +C117 +C118 +C119 +C120 +C121 +C122 +C123 +C124 +C125 +C126 +C127 +C128 +C129 +C130 +C131 +C132 +C133 +C134 +C135 +C136 +C137 +C138 +C139 +C140 +C141 +C142 +C143 +C144 +C145 +C146 +C147 +C148 +C149 +C150 +C151 +C152 +C153 +C154 +C155 +C156 +C157 +C158 +C159 +C160 +C161 +C162 +C163 +C164 +C165 +C166 +C167 +C168 +C169 +C170 +C171 +C172 +C173 +C174 +C175 +C176 +C177 +C178 +C179 +C180 +C181 +C182 +C183 +C184 +C185 +C186 +C187 +C188 +C189 +C190 +C191 +C192 +C193 +C194 +C195 +C196 +C197 +C198 +C199 +C200 +C201 +C202 +C203 +C204 +C205 +C206 +C207 +C208 +C209 +C210 +C211 +C212 +C213 +C214 +C215 +C216 +C217 +C218 +C219 +C220 +C221 +C222 +C223 +C224 +C225 +C226 +C227 +C228 +C229 +C230 +C231 +C232 +C233 +C234 +C235 +C236 +C237 +C238 +C239 +C240 +C241 +C242 +C243 +C244 +C245 +C246 +C247 +C248 +C249 +C250 +C251 +C252 +C253 +C254 +C255 +C256 +C257 +C258 +C259 +C260 +C261 +C262 +C263 +C264 +C265 +C266 +C267 +C268 +C269 +C270 +C271 +C272 +C273 +C274 +C275 +C276 +C277 +C278 +C279 +C280 +C281 +C282 +C283 +C284 +C285 +C286 +C287 +C288 +C289 +C290 +C291 +C292 +C293 +C294 +C295 +C296 +C297 +C298 +C299 +C300 +C301 +C302 +C303 +C304 +C305 +C306 +C307 +C308 +C309 +C310 +C311 +C312 +C313 +C314 +C315 +C316 +C317 +C318 +C319 +C320 +C321 +C322 +C323 +C324 +C325 +C326 +C327 +C328 +C329 +C330 +C331 +C332 +C333 +C334 +C335 +C336 +C337 +C338 +C339 +C340 +C341 +C342 +C343 +C344 +C345 +C346 +C347 +C348 +C349 +C350 +C351 +C352 +C353 +C354 +C355 +C356 +C357 +C358 +C359 +C360 +C361 +C362 +C363 +C364 +C365 +C366 +C367 +C368 +C369 +C370 +C371 +C372 +C373 +C374 +C375 +C376 +C377 +C378 +C379 +C380 +C381 +C382 +C383 +C384 +C385 +C386 +C387 +C388 +C389 +C390 +C391 +C392 +C393 +C394 +C395 +C396 +C397 +C398 +C399 +C400 +C401 +C402 +C403 +C404 +C405 +C406 +C407 +C408 +C409 +C410 +C411 +C412 +C413 +C414 +C415 +C416 +C417 +C418 +C419 +C420 +C421 +C422 +C423 +C424 +C425 +C426 +C427 +C428 +C429 +C430 +C431 +C432 +C433 +C434 +C435 +C436 +C437 +C438 +C439 +C440 +C441 +C442 +C443 +C444 +C445 +C446 +C447 +C448 +C449 +C450 +C451 +C452 +C453 +C454 +C455 +C456 +C457 +C458 +C459 +C460 +C461 +C462 +C463 +C464 +C465 +C466 +C467 +C468 +C469 +C470 +C471 +C472 +C473 +C474 +C475 +C476 +C477 +C478 +C479 +C480 +C481 +C482 +C483 +C484 +C485 +C486 +C487 +C488 +C489 +C490 +C491 +C492 +C493 +C494 +C495 +C496 +C497 +C498 +C499 +C500 +C501 +C502 +C503 +C504 +C505 +C506 +C507 +C508 +C509 +C510 +C511 +C512 +C513 +C514 +C515 +C516 +C517 +C518 +C519 +C520 +C521 +C522 +C523 +C524 +C525 +C526 +C527 +C528 +C529 +C530 +C531 +C532 +C533 +C534 +C535 +C536 +C537 +C538 +C539 +C540 +C541 +C542 +C543 +C544 +C545 +C546 +C547 +C548 +C549 +C550 +C551 +C552 +C553 +C554 +C555 +C556 +C557 +C558 +C559 +C560 +C561 +C562 +C563 +C564 +C565 +C566 +C567 +C568 +C569 +C570 +C571 +C572 +C573 +C574 +C575 +C576 +C577 +C578 +C579 +C580 +C581 +C582 +C583 +C584 +C585 +C586 +C587 +C588 +C589 +C590 +C591 +C592 +C593 +C594 +C595 +C596 +C597 +C598 +C599 +C600 +C601 +C602 +C603 +C604 +C605 +C606 +C607 +C608 +C609 +C610 +C611 +C612 +C613 +C614 +C615 +C616 +C617

+C618 +C619 +C620 +C621 +C622 +C623 +C624 +C625 +C626 +C627 +C628 +C629 +C630 +C631 +C632 +C633 +C634 +C635 +C636 +C637 +C638 +C639 +C640 +C641 +C642 +C643 +C644 +C645 +C646 +C647 +C648 +C649 +C650 +C651 +C652 +C653 +C654 +C655 +C656 +C657 +C658 +C659 +C660 +C661 +C662 +C663 +C664 +C665 +C666 +C667 +C668 +C669 +C670 +C671 +C672 +C673 +C674 +C675 +C676 +C677 +C678 +C679 +C680 +C681 +C682 +C683 +C684 +C685 +C686 +C687 +C688 +C689 +C690 +C691 +C692 +C693 +C694 +C695 +C696 +C697 +C698 +C699 +C700 +C701 +C702 +C703 +C704 +C705 +C706 +C707 +C708 +C709 +C710 +C711 +C712 +C713 +C714 +C715 +C716 +C717 +C718 +C719 +C720 +C721 +C722 +C723 +C724 +C725 +C726 +C727 +C728 +C729 +C730 +C731 +C732 +C733 +C734 +C735 +C736 +C737 +C738 +C739 +C740 +C741 +C742 +C743 +C744 +C745 +C746 +C747 +C748 +C749 +C750 +C751 +C752 +C753 +C754 +C755 +C756 +C757 +C758 +C759 +C760 +C761 +C762 +C763 +C764 +C765 +C766 +C767 +C768 +C769 +C770 +C771 +C772 +C773 +C774 +C775 +C776 +C777 +C778 +C779 +C780 +C781 +C782 +C783 +C784 +C785 +C786 +C787 +C788 +C789 +C790 +C791 +C792 +C793 +C794 +C795 +C796 +C797 +C798 +C799 +C800 +C801 +C802 +C803 +C804 +C805 +C806 +C807 +C808 +C809 +C810 +C811 +C812 +C813 +C814 +C815 +C816 +C817 +C818 +C819 +C820 +C821 +C822 +C823 +C824 +C825 +C826 +C827 +C828 +C829 +C830 +C831 +C832 +C833 +C834 +C835 +C836 +C837 +C838 +C839 +C840 +C841 +C842 +C843 +C844 +C845 +C846 +C847 +C848 +C849 +C850 +C851 +C852 +C853 +C854 +C855 +C856 +C857 +C858 +C859 +C860 +C861 +C862 +C863 +C864 +C865 +C866 +C867 +C868 +C869 +C870 +C871 +C872 +C873 +C874 +C875 +C876 +C877 +C878 +C879 +C880 +C881 +C882 +C883 +C884 +C885 +C886 +C887 +C888 +C889 +C890 +C891 +C892 +C893 +C894 +C895 +C896 +C897 +C898 +C899 +C900 +C901 +C902 +C903 +C904 +C905 +C906 +C907 +C908 +C909 +C910 +C911 +C912 +C913 +C914 +C915;

/* Constraints */
+C1 +C2 = 2; +C1 +C12 +C17 = 2; -C2 +C12 +C14 -C23 = 0; +C17 +C19 +C20 -C24 -C29 = 0; -C14 -C19 +C23 +C24 +C25 -C30 = 0; -C20 -C25 +C29 +C30 = 0; +C31 +C32 = 2; -C31 +C37 +C38 -C47 = 0; +C32 +C37 +C53 = 2; -C38 +C47 +C49 +C50 -C54 -C59 = 0; -C49 +C53 +C54 +C55 -C60 = 0; -C50 -C55 +C59 +C60 = 0; +C61 +C62 = 2; -C61 +C67 +C68 -C72 = 0; -C62 -C67 +C72 +C74 -C83 = 0; +C68 +C84 +C89 = 2; -C74 +C83 +C84 +C85 -C90 = 0; -C85 +C89 +C90 = 0; +C91 +C92 = 2; -C91 +C97 +C98 -C102 -C107 = 0; -C92 -C97 +C102 +C104 = 0; -C98 +C107 +C109 +C110 -C119 = 0; +C104 +C109 +C120 = 2; -C110 +C119 +C120 = 0; +C121 +C122 = 2; -C121 +C127 +C128 -C132 -C137 = 0; -C122 -C127 +C132 +C134 -C143 = 0; -C128 +C137 +C139 +C140 -C144 = 0; -C134 -C139 +C143 +C144 +C145 = 0; +C140 +C145 = 2; +C152 -C156 = 0; +C156 +C157 +C158 = 2; +C152 +C157 +C173 = 2; -C158 +C169 +C170 -C174 -C179 = 0; -C169 +C173 +C174 +C175 -C180 = 0; -C170 -C175 +C179 +C180 = 0; +C182 -C186 -C191 = 0; +C186 +C187 +C188 = 2; -C182 -C187 +C191 +C194 -C203 = 0; +C188 +C204 +C209 = 2; -C194 +C203 +C204 +C205 -C210 = 0; -C205 +C209 +C210 = 0; +C212 -C216 -C221 = 0; +C216 +C217 +C218 = 2; -C212 -C217 +C221 +C224 = 0; -C218 +C229 +C230 -C239 = 0; +C224 +C229 +C240 = 2; -C230 +C239 +C240 = 0; +C242 -C246 -C251 = 0; +C246 +C247 +C248 = 2; -C242 -C247 +C251 +C254 -C263 = 0; -C248 +C259 +C260 -C264 = 0; -C254 -C259 +C263 +C264 +C265 = 0; +C260 +C265 = 2; +C271 -C276 -C281 = 0; -C271 +C276 +C278

-C282 = 0; +C281 +C282 +C284 = 2; +C278 +C294 +C299 = 2; -C284 +C294 +C295 -C300 = 0; -C295 +C299 +C300 = 0; +C301 -C306 -C311 = 0; -C301 +C306 +C308 -C312 -C317 = 0; +C311 +C312 +C314 = 2; -C308 +C317 +C319 +C320 -C329 = 0; +C314 +C319 +C330 = 2; -C320 +C329 +C330 = 0; +C331 -C336 -C341 = 0; -C331 +C336 +C338 -C342 -C347 = 0; +C341 +C342 +C344 = 2; -C338 +C347 +C349 +C350 -C354 = 0; -C344 -C349 +C354 +C355 = 0; +C350 +C355 = 2; +C361 +C362 -C366 -C371 = 0; -C361 +C366 +C367 -C372 -C377 = 0; -C362 -C367 +C371 +C372 +C374 = 0; +C377 +C379 +C380 = 2; +C374 +C379 +C390 = 2; -C380 +C390 = 0; +C391 +C392 -C396 -C401 = 0; -C391 +C396 +C397 -C402 -C407 = 0; -C392 -C397 +C401 +C402 +C404 -C413 = 0; +C407 +C409 +C410 = 2; -C404 -C409 +C413 +C415 = 0; +C410 +C415 = 2; +C421 +C422 -C426 -C431 = 0; -C421 +C426 +C427 +C428 -C432 -C437 = 0; -C422 -C427 +C431 +C432 -C443 = 0; -C428 +C437 +C440 -C444 = 0; +C443 +C444 +C445 = 2; +C440 +C445 = 2; +212.5 C1 +212.5 C6 +212.5 C31 +212.5 C36 +187.5 C61 +187.5 C66 +12.5 C91 +12.5 C96 +162.5 C121 +162.5 C126 +400 C151 +400 C156 +87.5 C181 +87.5 C186 +187.5 C211 +187.5 C216 +1425 C241 +1425 C246 +137.5 C271 +137.5 C276 +575 C301 +575 C306 +12.5 C331 +12.5 C336 +137.5 C361 +137.5 C366 +87.5 C391 +87.5 C396 +1162.5 C421 +1162.5 C426 -100 C901 <= 0; +212.5 C2 +212.5 C11 +212.5 C32 +212.5 C41 +187.5 C62 +187.5 C71 +12.5 C92 +12.5 C101 +162.5 C122 +162.5 C131 +400 C152 +400 C161 +87.5 C182 +87.5 C191 +187.5 C212 +187.5 C221 +1425 C242 +1425 C251 +137.5 C272 +137.5 C281 +575 C302 +575 C311 +12.5 C332 +12.5 C341 +137.5 C362 +137.5 C371 +87.5 C392 +87.5 C401 +1162.5 C422 +1162.5 C431 -100 C902 <= 0; +212.5 C3 +212.5 C16 +212.5 C33 +212.5 C46 +187.5 C63 +187.5 C76 +12.5 C93 +12.5 C106 +162.5 C123 +162.5 C136 +400 C153 +400 C166 +87.5 C183 +87.5 C196 +187.5 C213 +187.5 C226 +1425 C243 +1425 C256 +137.5 C273 +137.5 C286 +575 C303 +575 C316 +12.5 C333 +12.5 C346 +137.5 C363 +137.5 C376 +87.5 C393 +87.5 C406 +1162.5 C423 +1162.5 C436 <= 0; +212.5 C4 +212.5 C21 +212.5 C34 +212.5 C51 +187.5 C64 +187.5 C81 +12.5 C94 +12.5 C111 +162.5 C124 +162.5 C141 +400 C154 +400 C171 +87.5 C184 +87.5 C201 +187.5 C214 +187.5 C231 +1425 C244 +1425 C261 +137.5 C274 +137.5 C291 +575 C304 +575 C321 +12.5 C334 +12.5 C351 +137.5 C364 +137.5 C381 +87.5 C394 +87.5 C411 +1162.5 C424 +1162.5 C441 <= 0; +212.5 C5 +212.5 C26 +212.5 C35 +212.5 C56 +187.5 C65 +187.5 C86 +12.5 C95 +12.5 C116 +162.5 C125 +162.5 C146 +400 C155 +400 C176 +87.5 C185 +87.5 C206 +187.5 C215 +187.5 C236 +1425 C245 +1425 C266 +137.5 C275 +137.5 C296 +575 C305 +575 C326 +12.5 C335 +12.5 C356 +137.5 C365 +137.5 C386 +87.5 C395 +87.5 C416 +1162.5 C425 +1162.5 C446 <= 0; +212.5 C7 +212.5 C12 +212.5 C37 +212.5 C42 +187.5 C67 +187.5 C72 +12.5 C97 +12.5 C102 +162.5 C127 +162.5 C132 +400 C157 +400 C162 +87.5 C187 +87.5 C192 +187.5 C217 +187.5 C222 +1425 C247 +1425 C252 +137.5 C277 +137.5 C282 +575 C307 +575 C312 +12.5 C337 +12.5 C342 +137.5 C367 +137.5 C372 +87.5 C397 +87.5 C402 +1162.5 C427 +1162.5 C432 -100 C906 <= 0; +212.5 C8 +212.5 C17 +212.5 C38 +212.5 C47 +187.5 C68 +187.5 C77 +12.5 C98 +12.5 C107 +162.5 C128 +162.5 C137 +400 C158 +400 C167 +87.5 C188 +87.5 C197 +187.5 C218 +187.5 C227 +1425 C248 +1425 C257 +137.5 C278 +137.5 C287 +575 C308 +575 C317 +12.5 C338 +12.5 C347 +137.5 C368 +137.5 C377 +87.5 C398 +87.5 C407 +1162.5 C428 +1162.5 C437 -100 C907 <= 0; +212.5 C9 +212.5 C22 +212.5 C39 +212.5 C52 +187.5 C69 +187.5 C82 +12.5 C99 +12.5 C112 +162.5 C129 +162.5 C142 +400 C159 +400 C172 +87.5 C189 +87.5 C202 +187.5 C219 +187.5 C232 +1425 C249 +1425 C262 +137.5 C279 +137.5 C292 +575 C309

+575 C322 +12.5 C339 +12.5 C352 +137.5 C369 +137.5 C382 +87.5 C399 +87.5 C412 +1162.5 C429 +1162.5 C442 <= 0; +212.5 C10 +212.5 C27 +212.5 C40 +212.5 C57 +187.5 C70 +187.5 C87 +12.5 C100 +12.5 C117 +162.5 C130 +162.5 C147 +400 C160 +400 C177 +87.5 C190 +87.5 C207 +187.5 C220 +187.5 C237 +1425 C250 +1425 C267 +137.5 C280 +137.5 C297 +575 C310 +575 C327 +12.5 C340 +12.5 C357 +137.5 C370 +137.5 C387 +87.5 C400 +87.5 C417 +1162.5 C430 +1162.5 C447 <= 0; +212.5 C13 +212.5 C18 +212.5 C43 +212.5 C48 +187.5 C73 +187.5 C78 +12.5 C103 +12.5 C108 +162.5 C133 +162.5 C138 +400 C163 +400 C168 +87.5 C193 +87.5 C198 +187.5 C223 +187.5 C228 +1425 C253 +1425 C258 +137.5 C283 +137.5 C288 +575 C313 +575 C318 +12.5 C343 +12.5 C348 +137.5 C373 +137.5 C378 +87.5 C403 +87.5 C408 +1162.5 C433 +1162.5 C438 <= 0; +212.5 C14 +212.5 C23 +212.5 C44 +212.5 C53 +187.5 C74 +187.5 C83 +12.5 C104 +12.5 C113 +162.5 C134 +162.5 C143 +400 C164 +400 C173 +87.5 C194 +87.5 C203 +187.5 C224 +187.5 C233 +1425 C254 +1425 C263 +137.5 C284 +137.5 C293 +575 C314 +575 C323 +12.5 C344 +12.5 C353 +137.5 C374 +137.5 C383 +87.5 C404 +87.5 C413 +1162.5 C434 +1162.5 C443 -100 C911 <= 0; +212.5 C15 +212.5 C28 +212.5 C45 +212.5 C58 +187.5 C75 +187.5 C88 +12.5 C105 +12.5 C118 +162.5 C135 +162.5 C148 +400 C165 +400 C178 +87.5 C195 +87.5 C208 +187.5 C225 +187.5 C238 +1425 C255 +1425 C268 +137.5 C285 +137.5 C298 +575 C315 +575 C328 +12.5 C345 +12.5 C358 +137.5 C375 +137.5 C388 +87.5 C405 +87.5 C418 +1162.5 C435 +1162.5 C448 <= 0; +212.5 C19 +212.5 C24 +212.5 C49 +212.5 C54 +187.5 C79 +187.5 C84 +12.5 C109 +12.5 C114 +162.5 C139 +162.5 C144 +400 C169 +400 C174 +87.5 C199 +87.5 C204 +187.5 C229 +187.5 C234 +1425 C259 +1425 C264 +137.5 C289 +137.5 C294 +575 C319 +575 C324 +12.5 C349 +12.5 C354 +137.5 C379 +137.5 C384 +87.5 C409 +87.5 C414 +1162.5 C439 +1162.5 C444 -100 C913 <= 0; +212.5 C20 +212.5 C29 +212.5 C50 +212.5 C59 +187.5 C80 +187.5 C89 +12.5 C110 +12.5 C119 +162.5 C140 +162.5 C149 +400 C170 +400 C179 +87.5 C200 +87.5 C209 +187.5 C230 +187.5 C239 +1425 C260 +1425 C269 +137.5 C290 +137.5 C299 +575 C320 +575 C329 +12.5 C350 +12.5 C359 +137.5 C380 +137.5 C389 +87.5 C410 +87.5 C419 +1162.5 C440 +1162.5 C449 -100 C914 <= 0; +212.5 C25 +212.5 C30 +212.5 C55 +212.5 C60 +187.5 C85 +187.5 C90 +12.5 C115 +12.5 C120 +162.5 C145 +162.5 C150 +400 C175 +400 C180 +87.5 C205 +87.5 C210 +187.5 C235 +187.5 C240 +1425 C265 +1425 C270 +137.5 C295 +137.5 C300 +575 C325 +575 C330 +12.5 C355 +12.5 C360 +137.5 C385 +137.5 C390 +87.5 C415 +87.5 C420 +1162.5 C445 +1162.5 C450 -100 C915 <= 0; R106: +C901 <= 80; R107: +C902 <= 80; R108: +C903 <= 80; R109: +C904 <= 80; R110: +C905 <= 80; R111: +C906 <= 80; R112: +C907 <= 80; R113: +C908 <= 80; R114: +C909 <= 80; R115: +C910 <= 80; R116: +C911 <= 80; R117: +C912 <= 80; R118: +C913 <= 80; R119: +C914 <= 80; R120: +C915 <= 80;

/* Variable bounds */
C1 <= 1; C2 <= 1; C3 <= 1; C4 <= 1; C5 <= 1; C6 <= 1; C7 <= 1; C8 <= 1; C9 <= 1; C10 <= 1; C11 <= 1; C12 <= 1; C13 <= 1; C14 <= 1; C15 <= 1; C16 <= 1; C17 <= 1; C18 <= 1; C19 <= 1; C20 <= 1; C21 <= 1; C22 <= 1; C23 <= 1; C24 <= 1; C25 <= 1; C26 <= 1; C27 <= 1; C28 <= 1; C29 <= 1; C30 <= 1; C31 <= 1; C32 <= 1; C33 <= 1; C34 <= 1; C35 <= 1; C36 <= 1; C37 <= 1; C38 <= 1; C39 <= 1; C40 <= 1; C41 <= 1; C42 <= 1; C43 <= 1; C44 <= 1; C45 <= 1; C46 <= 1; C47 <= 1; C48 <= 1; C49 <= 1; C50 <= 1; C51 <= 1; C52 <= 1; C53 <= 1; C54 <= 1; C55 <= 1; C56 <= 1; C57 <= 1; C58 <= 1; C59 <= 1; C60 <= 1; C61 <= 1; C62 <= 1; C63 <= 1; C64 <= 1;

C65 <= 1; C66 <= 1; C67 <= 1; C68 <= 1; C69 <= 1; C70 <= 1; C71 <= 1; C72 <= 1; C73 <= 1; C74 <= 1; C75 <= 1; C76 <= 1; C77 <= 1; C78 <= 1; C79 <= 1; C80 <= 1; C81 <= 1; C82 <= 1; C83 <= 1; C84 <= 1; C85 <= 1; C86 <= 1; C87 <= 1; C88 <= 1; C89 <= 1; C90 <= 1; C91 <= 1; C92 <= 1; C93 <= 1; C94 <= 1; C95 <= 1; C96 <= 1; C97 <= 1; C98 <= 1; C99 <= 1; C100 <= 1; C101 <= 1; C102 <= 1; C103 <= 1; C104 <= 1; C105 <= 1; C106 <= 1; C107 <= 1; C108 <= 1; C109 <= 1; C110 <= 1; C111 <= 1; C112 <= 1; C113 <= 1; C114 <= 1; C115 <= 1; C116 <= 1; C117 <= 1; C118 <= 1; C119 <= 1; C120 <= 1; C121 <= 1; C122 <= 1; C123 <= 1; C124 <= 1; C125 <= 1; C126 <= 1; C127 <= 1; C128 <= 1; C129 <= 1; C130 <= 1; C131 <= 1; C132 <= 1; C133 <= 1; C134 <= 1; C135 <= 1; C136 <= 1; C137 <= 1; C138 <= 1; C139 <= 1; C140 <= 1; C141 <= 1; C142 <= 1; C143 <= 1; C144 <= 1; C145 <= 1; C146 <= 1; C147 <= 1; C148 <= 1; C149 <= 1; C150 <= 1; C151 <= 1; C152 <= 1; C153 <= 1; C154 <= 1; C155 <= 1; C156 <= 1; C157 <= 1; C158 <= 1; C159 <= 1; C160 <= 1; C161 <= 1; C162 <= 1; C163 <= 1; C164 <= 1; C165 <= 1; C166 <= 1; C167 <= 1; C168 <= 1; C169 <= 1; C170 <= 1; C171 <= 1; C172 <= 1; C173 <= 1; C174 <= 1; C175 <= 1; C176 <= 1; C177 <= 1; C178 <= 1; C179 <= 1; C180 <= 1; C181 <= 1; C182 <= 1; C183 <= 1; C184 <= 1; C185 <= 1; C186 <= 1; C187 <= 1; C188 <= 1; C189 <= 1; C190 <= 1; C191 <= 1; C192 <= 1; C193 <= 1; C194 <= 1; C195 <= 1; C196 <= 1; C197 <= 1; C198 <= 1; C199 <= 1; C200 <= 1; C201 <= 1; C202 <= 1; C203 <= 1; C204 <= 1; C205 <= 1; C206 <= 1; C207 <= 1; C208 <= 1; C209 <= 1; C210 <= 1; C211 <= 1; C212 <= 1; C213 <= 1; C214 <= 1; C215 <= 1; C216 <= 1; C217 <= 1; C218 <= 1; C219 <= 1; C220 <= 1; C221 <= 1; C222 <= 1; C223 <= 1; C224 <= 1; C225 <= 1; C226 <= 1; C227 <= 1; C228 <= 1; C229 <= 1; C230 <= 1; C231 <= 1; C232 <= 1; C233 <= 1; C234 <= 1; C235 <= 1; C236 <= 1; C237 <= 1; C238 <= 1; C239 <= 1; C240 <= 1; C241 <= 1; C242 <= 1; C243 <= 1; C244 <= 1; C245 <= 1; C246 <= 1; C247 <= 1; C248 <= 1; C249 <= 1; C250 <= 1; C251 <= 1; C252 <= 1; C253 <= 1; C254 <= 1; C255 <= 1; C256 <= 1; C257 <= 1; C258 <= 1; C259 <= 1; C260 <= 1; C261 <= 1; C262 <= 1; C263 <= 1; C264 <= 1; C265 <= 1; C266 <= 1; C267 <= 1; C268 <= 1; C269 <= 1; C270 <= 1; C271 <= 1; C272 <= 1; C273 <= 1; C274 <= 1; C275 <= 1; C276 <= 1; C277 <= 1; C278 <= 1; C279 <= 1; C280 <= 1; C281 <= 1; C282 <= 1; C283 <= 1; C284 <= 1; C285 <= 1; C286 <= 1; C287 <= 1; C288 <= 1; C289 <= 1; C290 <= 1; C291 <= 1; C292 <= 1; C293 <= 1; C294 <= 1; C295 <= 1; C296 <= 1; C297 <= 1; C298 <= 1; C299 <= 1; C300 <= 1; C301 <= 1; C302 <= 1; C303 <= 1; C304 <= 1; C305 <= 1; C306 <= 1; C307 <= 1; C308 <= 1; C309 <= 1; C310 <= 1; C311 <= 1; C312 <= 1; C313 <= 1; C314 <= 1; C315 <= 1; C316 <= 1; C317 <= 1; C318 <= 1; C319 <= 1; C320 <= 1; C321 <= 1; C322 <= 1; C323 <= 1; C324 <= 1; C325 <= 1; C326 <= 1; C327 <= 1; C328 <= 1; C329 <= 1; C330 <= 1; C331 <= 1; C332 <= 1; C333 <= 1; C334 <= 1; C335 <= 1; C336 <= 1; C337 <= 1; C338 <= 1; C339 <= 1; C340 <= 1; C341 <= 1; C342 <= 1; C343 <= 1; C344 <= 1; C345 <= 1; C346 <= 1; C347 <= 1; C348 <= 1; C349 <= 1; C350 <= 1; C351 <= 1; C352 <= 1; C353 <= 1; C354 <= 1; C355 <= 1; C356 <= 1; C357 <= 1; C358 <= 1; C359 <= 1; C360 <= 1; C361 <= 1; C362 <= 1; C363 <= 1; C364 <= 1; C365 <= 1; C366 <= 1; C367 <= 1; C368 <= 1; C369 <= 1; C370 <= 1; C371 <= 1; C372 <= 1; C373 <= 1; C374 <= 1; C375 <= 1; C376 <= 1; C377 <= 1; C378 <= 1; C379 <= 1; C380 <= 1; C381 <= 1; C382 <= 1; C383 <= 1; C384 <= 1; C385 <= 1; C386 <= 1; C387 <= 1; C388 <= 1; C389 <= 1; C390 <= 1; C391 <= 1; C392 <= 1; C393 <= 1; C394 <= 1; C395 <= 1; C396 <= 1; C397 <= 1; C398 <= 1; C399 <= 1; C400 <= 1; C401 <= 1; C402 <= 1; C403 <= 1; C404 <= 1;

C405 <= 1; C406 <= 1; C407 <= 1; C408 <= 1; C409 <= 1; C410 <= 1; C411 <= 1; C412 <= 1; C413 <= 1; C414 <= 1; C415 <= 1; C416 <= 1; C417 <= 1; C418 <= 1; C419 <= 1; C420 <= 1; C421 <= 1; C422 <= 1; C423 <= 1; C424 <= 1; C425 <= 1; C426 <= 1; C427 <= 1; C428 <= 1; C429 <= 1; C430 <= 1; C431 <= 1; C432 <= 1; C433 <= 1; C434 <= 1; C435 <= 1; C436 <= 1; C437 <= 1; C438 <= 1; C439 <= 1; C440 <= 1; C441 <= 1; C442 <= 1; C443 <= 1; C444 <= 1; C445 <= 1; C446 <= 1; C447 <= 1; C448 <= 1; C449 <= 1; C450 <= 1; C451 <= 1; C452 <= 1; C453 <= 1; C454 <= 1; C455 <= 1; C456 <= 1; C457 <= 1; C458 <= 1; C459 <= 1; C460 <= 1; C461 <= 1; C462 <= 1; C463 <= 1; C464 <= 1; C465 <= 1; C466 <= 1; C467 <= 1; C468 <= 1; C469 <= 1; C470 <= 1; C471 <= 1; C472 <= 1; C473 <= 1; C474 <= 1; C475 <= 1; C476 <= 1; C477 <= 1; C478 <= 1; C479 <= 1; C480 <= 1; C481 <= 1; C482 <= 1; C483 <= 1; C484 <= 1; C485 <= 1; C486 <= 1; C487 <= 1; C488 <= 1; C489 <= 1; C490 <= 1; C491 <= 1; C492 <= 1; C493 <= 1; C494 <= 1; C495 <= 1; C496 <= 1; C497 <= 1; C498 <= 1; C499 <= 1; C500 <= 1; C501 <= 1; C502 <= 1; C503 <= 1; C504 <= 1; C505 <= 1; C506 <= 1; C507 <= 1; C508 <= 1; C509 <= 1; C510 <= 1; C511 <= 1; C512 <= 1; C513 <= 1; C514 <= 1; C515 <= 1; C516 <= 1; C517 <= 1; C518 <= 1; C519 <= 1; C520 <= 1; C521 <= 1; C522 <= 1; C523 <= 1; C524 <= 1; C525 <= 1; C526 <= 1; C527 <= 1; C528 <= 1; C529 <= 1; C530 <= 1; C531 <= 1; C532 <= 1; C533 <= 1; C534 <= 1; C535 <= 1; C536 <= 1; C537 <= 1; C538 <= 1; C539 <= 1; C540 <= 1; C541 <= 1; C542 <= 1; C543 <= 1; C544 <= 1; C545 <= 1; C546 <= 1; C547 <= 1; C548 <= 1; C549 <= 1; C550 <= 1; C551 <= 1; C552 <= 1; C553 <= 1; C554 <= 1; C555 <= 1; C556 <= 1; C557 <= 1; C558 <= 1; C559 <= 1; C560 <= 1; C561 <= 1; C562 <= 1; C563 <= 1; C564 <= 1; C565 <= 1; C566 <= 1; C567 <= 1; C568 <= 1; C569 <= 1; C570 <= 1; C571 <= 1; C572 <= 1; C573 <= 1; C574 <= 1; C575 <= 1; C576 <= 1; C577 <= 1; C578 <= 1; C579 <= 1; C580 <= 1; C581 <= 1; C582 <= 1; C583 <= 1; C584 <= 1; C585 <= 1; C586 <= 1; C587 <= 1; C588 <= 1; C589 <= 1; C590 <= 1; C591 <= 1; C592 <= 1; C593 <= 1; C594 <= 1; C595 <= 1; C596 <= 1; C597 <= 1; C598 <= 1; C599 <= 1; C600 <= 1; C601 <= 1; C602 <= 1; C603 <= 1; C604 <= 1; C605 <= 1; C606 <= 1; C607 <= 1; C608 <= 1; C609 <= 1; C610 <= 1; C611 <= 1; C612 <= 1; C613 <= 1; C614 <= 1; C615 <= 1; C616 <= 1; C617 <= 1; C618 <= 1; C619 <= 1; C620 <= 1; C621 <= 1; C622 <= 1; C623 <= 1; C624 <= 1; C625 <= 1; C626 <= 1; C627 <= 1; C628 <= 1; C629 <= 1; C630 <= 1; C631 <= 1; C632 <= 1; C633 <= 1; C634 <= 1; C635 <= 1; C636 <= 1; C637 <= 1; C638 <= 1; C639 <= 1; C640 <= 1; C641 <= 1; C642 <= 1; C643 <= 1; C644 <= 1; C645 <= 1; C646 <= 1; C647 <= 1; C648 <= 1; C649 <= 1; C650 <= 1; C651 <= 1; C652 <= 1; C653 <= 1; C654 <= 1; C655 <= 1; C656 <= 1; C657 <= 1; C658 <= 1; C659 <= 1; C660 <= 1; C661 <= 1; C662 <= 1; C663 <= 1; C664 <= 1; C665 <= 1; C666 <= 1; C667 <= 1; C668 <= 1; C669 <= 1; C670 <= 1; C671 <= 1; C672 <= 1; C673 <= 1; C674 <= 1; C675 <= 1; C676 <= 1; C677 <= 1; C678 <= 1; C679 <= 1; C680 <= 1; C681 <= 1; C682 <= 1; C683 <= 1; C684 <= 1; C685 <= 1; C686 <= 1; C687 <= 1; C688 <= 1; C689 <= 1; C690 <= 1; C691 <= 1; C692 <= 1; C693 <= 1; C694 <= 1; C695 <= 1; C696 <= 1; C697 <= 1; C698 <= 1; C699 <= 1; C700 <= 1; C701 <= 1; C702 <= 1; C703 <= 1; C704 <= 1; C705 <= 1; C706 <= 1; C707 <= 1; C708 <= 1; C709 <= 1; C710 <= 1; C711 <= 1; C712 <= 1; C713 <= 1; C714 <= 1; C715 <= 1; C716 <= 1; C717 <= 1; C718 <= 1; C719 <= 1; C720 <= 1; C721 <= 1; C722 <= 1; C723 <= 1; C724 <= 1; C725 <= 1; C726 <= 1; C727 <= 1; C728 <= 1; C729 <= 1; C730 <= 1; C731 <= 1; C732 <= 1; C733 <= 1; C734 <= 1; C735 <= 1; C736 <= 1; C737 <= 1; C738 <= 1; C739 <= 1; C740 <= 1;

C741 <= 1; C742 <= 1; C743 <= 1; C744 <= 1; C745 <= 1; C746 <= 1; C747 <= 1; C748 <= 1; C749 <= 1; C750 <= 1; C751 <= 1; C752 <= 1; C753 <= 1; C754 <= 1; C755 <= 1; C756 <= 1; C757 <= 1; C758 <= 1; C759 <= 1; C760 <= 1; C761 <= 1; C762 <= 1; C763 <= 1; C764 <= 1; C765 <= 1; C766 <= 1; C767 <= 1; C768 <= 1; C769 <= 1; C770 <= 1; C771 <= 1; C772 <= 1; C773 <= 1; C774 <= 1; C775 <= 1; C776 <= 1; C777 <= 1; C778 <= 1; C779 <= 1; C780 <= 1; C781 <= 1; C782 <= 1; C783 <= 1; C784 <= 1; C785 <= 1; C786 <= 1; C787 <= 1; C788 <= 1; C789 <= 1; C790 <= 1; C791 <= 1; C792 <= 1; C793 <= 1; C794 <= 1; C795 <= 1; C796 <= 1; C797 <= 1; C798 <= 1; C799 <= 1; C800 <= 1; C801 <= 1; C802 <= 1; C803 <= 1; C804 <= 1; C805 <= 1; C806 <= 1; C807 <= 1; C808 <= 1; C809 <= 1; C810 <= 1; C811 <= 1; C812 <= 1; C813 <= 1; C814 <= 1; C815 <= 1; C816 <= 1; C817 <= 1; C818 <= 1; C819 <= 1; C820 <= 1; C821 <= 1; C822 <= 1; C823 <= 1; C824 <= 1; C825 <= 1; C826 <= 1; C827 <= 1; C828 <= 1; C829 <= 1; C830 <= 1; C831 <= 1; C832 <= 1; C833 <= 1; C834 <= 1; C835 <= 1; C836 <= 1; C837 <= 1; C838 <= 1; C839 <= 1; C840 <= 1; C841 <= 1; C842 <= 1; C843 <= 1; C844 <= 1; C845 <= 1; C846 <= 1; C847 <= 1; C848 <= 1; C849 <= 1; C850 <= 1; C851 <= 1; C852 <= 1; C853 <= 1; C854 <= 1; C855 <= 1; C856 <= 1; C857 <= 1; C858 <= 1; C859 <= 1; C860 <= 1; C861 <= 1; C862 <= 1; C863 <= 1; C864 <= 1; C865 <= 1; C866 <= 1; C867 <= 1; C868 <= 1; C869 <= 1; C870 <= 1; C871 <= 1; C872 <= 1; C873 <= 1; C874 <= 1; C875 <= 1; C876 <= 1; C877 <= 1; C878 <= 1; C879 <= 1; C880 <= 1; C881 <= 1; C882 <= 1; C883 <= 1; C884 <= 1; C885 <= 1; C886 <= 1; C887 <= 1; C888 <= 1; C889 <= 1; C890 <= 1; C891 <= 1; C892 <= 1; C893 <= 1; C894 <= 1; C895 <= 1; C896 <= 1; C897 <= 1; C898 <= 1; C899 <= 1; C900 <= 1;

/* Integer definitions */
int C1, C2, C3, C4, C5, C6, C7, C8, C9, C10, C11, C12, C13, C14, C15, C16, C17, C18, C19, C20, C21, C22, C23, C24, C25, C26, C27, C28, C29, C30, C31, C32, C33, C34, C35, C36, C37, C38, C39, C40, C41, C42, C43, C44, C45, C46, C47, C48, C49, C50, C51, C52, C53, C54, C55, C56, C57, C58, C59, C60, C61, C62, C63, C64, C65, C66, C67, C68, C69, C70, C71, C72, C73, C74, C75, C76, C77, C78, C79, C80, C81, C82, C83, C84, C85, C86, C87, C88, C89, C90, C91, C92, C93, C94, C95, C96, C97, C98, C99, C100, C101, C102, C103, C104, C105, C106, C107, C108, C109, C110, C111, C112, C113, C114, C115, C116, C117, C118, C119, C120, C121, C122, C123, C124, C125, C126, C127, C128, C129, C130, C131, C132, C133, C134, C135, C136, C137, C138, C139, C140, C141, C142, C143, C144, C145, C146, C147, C148, C149, C150, C151, C152, C153, C154, C155, C156, C157, C158, C159, C160, C161, C162, C163, C164, C165, C166, C167, C168, C169, C170, C171, C172, C173, C174, C175, C176, C177, C178, C179, C180, C181, C182, C183, C184, C185, C186, C187, C188, C189, C190, C191, C192, C193, C194, C195, C196, C197, C198, C199, C200, C201, C202, C203, C204, C205, C206, C207, C208, C209, C210, C211, C212, C213, C214, C215, C216, C217, C218, C219, C220, C221, C222, C223, C224, C225, C226, C227, C228, C229, C230, C231, C232, C233, C234, C235, C236, C237, C238, C239, C240, C241, C242, C243, C244, C245, C246, C247, C248, C249, C250, C251, C252, C253, C254, C255, C256, C257, C258, C259, C260, C261, C262, C263, C264, C265, C266, C267, C268, C269, C270, C271, C272, C273, C274, C275, C276, C277, C278, C279, C280, C281, C282, C283, C284, C285, C286, C287, C288, C289, C290, C291, C292, C293, C294, C295, C296, C297, C298, C299, C300, C301, C302, C303, C304, C305, C306, C307, C308, C309, C310, C311, C312, C313, C314 ,C315, C316, C317, C318,

C319, C320, C321, C322, C323, C324, C325, C326, C327, C328, C329, C330, C331, C332, C333, C334, C335, C336, C337, C338, C339, C340, C341, C342, C343, C344, C345, C346, C347, C348, C349, C350, C351, C352, C353, C354, C355, C356, C357, C358, C359, C360, C361, C362, C363, C364, C365, C366, C367, C368, C369, C370, C371, C372, C373, C374, C375, C376, C377, C378, C379, C380, C381, C382, C383, C384, C385, C386, C387, C388, C389, C390, C391, C392, C393, C394, C395, C396, C397, C398, C399, C400, C401, C402, C403, C404, C405, C406, C407, C408, C409, C410, C411, C412, C413, C414, C415, C416, C417, C418, C419, C420, C421, C422, C423, C424, C425, C426, C427, C428, C429, C430, C431, C432, C433, C434, C435, C436, C437, C438, C439, C440, C441, C442, C443, C444, C445, C446, C447, C448, C449, C450, C451, C452, C453, C454, C455, C456, C457, C458, C459, C460, C461, C462, C463, C464, C465, C466, C467, C468, C469, C470, C471, C472, C473, C474, C475, C476, C477, C478, C479, C480, C481, C482, C483, C484, C485, C486, C487, C488, C489, C490, C491, C492, C493, C494, C495, C496, C497, C498, C499, C500, C501, C502, C503, C504, C505, C506, C507, C508, C509, C510, C511, C512, C513, C514, C515, C516, C517, C518, C519, C520, C521, C522, C523, C524, C525, C526, C527, C528, C529, C530, C531, C532, C533, C534, C535, C536, C537, C538, C539, C540, C541, C542, C543, C544, C545, C546, C547, C548, C549, C550, C551, C552, C553, C554, C555, C556, C557, C558, C559, C560, C561, C562, C563, C564, C565, C566, C567, C568, C569, C570, C571, C572, C573, C574, C575, C576, C577, C578, C579, C580, C581, C582, C583, C584, C585, C586, C587, C588, C589, C590, C591, C592, C593, C594, C595, C596, C597, C598, C599, C600, C601, C602, C603, C604, C605, C606, C607, C608, C609, C610, C611, C612, C613, C614, C615, C616, C617, C618, C619, C620, C621, C622, C623, C624, C625, C626, C627, C628, C629, C630, C631, C632, C633, C634, C635, C636, C637, C638, C639, C640, C641, C642, C643, C644, C645, C646, C647, C648, C649, C650, C651, C652, C653, C654, C655, C656, C657, C658, C659, C660, C661, C662, C663, C664, C665, C666, C667, C668, C669, C670, C671, C672, C673, C674, C675, C676, C677, C678, C679, C680, C681, C682, C683, C684, C685, C686, C687, C688, C689, C690, C691, C692, C693, C694, C695, C696, C697, C698, C699, C700, C701, C702, C703, C704, C705, C706, C707, C708, C709, C710, C711, C712, C713, C714, C715, C716, C717, C718, C719, C720, C721, C722, C723, C724, C725, C726, C727, C728, C729, C730, C731, C732, C733, C734, C735, C736, C737, C738, C739, C740, C741, C742, C743, C744, C745, C746, C747, C748, C749, C750, C751, C752, C753, C754, C755, C756, C757, C758, C759, C760, C761, C762, C763, C764, C765, C766, C767, C768, C769, C770, C771, C772, C773, C774, C775, C776, C777, C778, C779, C780, C781, C782, C783, C784, C785, C786, C787, C788, C789, C790, C791, C792, C793, C794, C795, C796, C797, C798, C799, C800, C801, C802, C803, C804, C805, C806, C807, C808, C809, C810, C811, C812, C813 ,C814, C815, C816, C817, C818, C819, C820, C821, C822, C823, C824, C825, C826, C827, C828, C829, C830, C831, C832, C833, C834, C835, C836, C837, C838, C839, C840, C841, C842, C843, C844, C845, C846, C847, C848, C849, C850, C851, C852, C853, C854, C855, C856, C857, C858, C859, C860, C861, C862, C863, C864, C865, C866, C867, C868, C869, C870, C871, C872, C873, C874, C875, C876, C877, C878, C879, C880, C881, C882, C883, C884, C885, C886, C887, C888, C889, C890, C891, C892, C893, C894, C895, C896, C897, C898, C899, C900, C901, C902, C903, C904, C905, C906, C907, C908, C909, C910, C911, C912, C913, C914, C915;

5.8.3 Results using LPSolve

5.9 Transparent with 1+1 protection appendices

Student Name	:	Tiago Esteves (November 28, 2017 -)
Goal	:	Implement the dimensioning of optical networks in the transparent transport mode.

5.9.1 Script using MatLab

5.9.1.1 Scenario 1

RESULTS: Reference Network

Scenario: Transparent Low Traffic

LINKS

Number of optical channels in link (1,2): 3
Number of optical channels in link (1,3): 2
Number of optical channels in link (2,3): 4
Number of optical channels in link (2,4): 5
Number of optical channels in link (3,5): 5
Number of optical channels in link (4,5): 2
Number of optical channels in link (4,6): 4
Number of optical channels in link (5,6): 3

NODES

Tributary Ports =

Node	ODU0	ODU1	ODU2	ODU3	ODU4
1	13	13	3	0	0
2	11	7	2	2	1
3	7	6	3	2	0
4	7	10	3	0	0
5	14	4	4	1	1
6	8	10	1	1	2

Line Ports =

Node	ODU4
1	5
2	6
3	5
4	5
5	6
6	7

Through Ports =

Node	ODU4
1	5
2	12
3	11
4	11
5	10
6	7

PATHS
Link (1,2)——————————
Path between node (1,2)
Path between node (1,4)
Path between node (1,6)
Link (1,3)——————————
Path between node (1,3)
Path between node (1,5)
Link (2,3)——————————
Path between node (2,3)
Path between node (2,5)
Path between node (2,6)
Path between node (3,4)
Link (2,4)——————————
Path between node (1,4)
Path between node (1,6)
Path between node (2,4)
Path between node (2,6)
Path between node (3,4)
Link (3,5)——————————
Path between node (1,5)
Path between node (2,5)
Path between node (2,6)

Path between node (3,5)
Path between node (3,6)
Link (4,5)——————————
Path between node (4,5)
Path between node (5,6)
Link (4,6)——————————
Path between node (1,6)
Path between node (2,6)
Path between node (4,6)
Path between node (5,6)
Link (5,6)——————————
Path between node (2,6)
Path between node (3,6)
Path between node (5,6)

5.9.1.2 Scenario 2

5.9.1.3 Scenario 3

5.9.1.4 Scenario 4

5.9.2 ILP using LPSolve

5.9.2.1 Scenario 1

/* Objective function */
min: +C1 +C2 +C3 +C4 +C5 +C6 +C7 +C8 +C9 +C10 +C11 +C12 +C13 +C14 +C15 +C16 +C17 +C18 +C19 +C20 +C21 +C22 +C23 +C24 +C25 +C26 +C27 +C28 +C29 +C30 +C31 +C32 +C33 +C34 +C35 +C36 +C37 +C38 +C39 +C40 +C41 +C42 +C43 +C44 +C45 +C46 +C47 +C48 +C49 +C50 +C51 +C52 +C53 +C54 +C55 +C56 +C57 +C58 +C59 +C60 +C61 +C62 +C63 +C64 +C65 +C66 +C67 +C68 +C69 +C70 +C71 +C72 +C73 +C74 +C75 +C76 +C77 +C78 +C79 +C80 +C81 +C82 +C83 +C84 +C85 +C86 +C87 +C88 +C89 +C90 +C91 +C92 +C93 +C94 +C95 +C96 +C97 +C98 +C99 +C100 +C101 +C102 +C103 +C104 +C105 +C106 +C107 +C108 +C109 +C110 +C111 +C112 +C113 +C114 +C115 +C116 +C117 +C118 +C119 +C120 +C121 +C122 +C123 +C124 +C125 +C126 +C127 +C128 +C129 +C130 +C131 +C132 +C133 +C134 +C135 +C136 +C137 +C138 +C139 +C140 +C141 +C142 +C143 +C144 +C145 +C146 +C147 +C148 +C149 +C150 +C151 +C152 +C153 +C154 +C155 +C156 +C157 +C158 +C159 +C160 +C161 +C162 +C163 +C164 +C165 +C166 +C167 +C168 +C169 +C170 +C171 +C172 +C173 +C174 +C175 +C176 +C177 +C178 +C179 +C180 +C181 +C182 +C183 +C184 +C185 +C186 +C187 +C188 +C189 +C190 +C191 +C192 +C193 +C194 +C195 +C196 +C197 +C198 +C199 +C200 +C201 +C202 +C203 +C204 +C205 +C206 +C207 +C208 +C209 +C210 +C211 +C212 +C213 +C214 +C215 +C216 +C217 +C218 +C219 +C220 +C221 +C222 +C223 +C224 +C225 +C226 +C227 +C228 +C229 +C230 +C231 +C232 +C233 +C234 +C235 +C236 +C237 +C238 +C239 +C240 +C241 +C242 +C243 +C244 +C245 +C246 +C247 +C248 +C249 +C250 +C251 +C252 +C253

+C254 +C255 +C256 +C257 +C258 +C259 +C260 +C261 +C262 +C263 +C264 +C265 +C266 +C267 +C268 +C269 +C270 +C271 +C272 +C273 +C274 +C275 +C276 +C277 +C278 +C279 +C280 +C281 +C282 +C283 +C284 +C285 +C286 +C287 +C288 +C289 +C290 +C291 +C292 +C293 +C294 +C295 +C296 +C297 +C298 +C299 +C300 +C301 +C302 +C303 +C304 +C305 +C306 +C307 +C308 +C309 +C310 +C311 +C312 +C313 +C314 +C315 +C316 +C317 +C318 +C319 +C320 +C321 +C322 +C323 +C324 +C325 +C326 +C327 +C328 +C329 +C330 +C331 +C332 +C333 +C334 +C335 +C336 +C337 +C338 +C339 +C340 +C341 +C342 +C343 +C344 +C345 +C346 +C347 +C348 +C349 +C350 +C351 +C352 +C353 +C354 +C355 +C356 +C357 +C358 +C359 +C360 +C361 +C362 +C363 +C364 +C365 +C366 +C367 +C368 +C369 +C370 +C371 +C372 +C373 +C374 +C375 +C376 +C377 +C378 +C379 +C380 +C381 +C382 +C383 +C384 +C385 +C386 +C387 +C388 +C389 +C390 +C391 +C392 +C393 +C394 +C395 +C396 +C397 +C398 +C399 +C400 +C401 +C402 +C403 +C404 +C405 +C406 +C407 +C408 +C409 +C410 +C411 +C412 +C413 +C414 +C415 +C416 +C417 +C418 +C419 +C420 +C421 +C422 +C423 +C424 +C425 +C426 +C427 +C428 +C429 +C430 +C431 +C432 +C433 +C434 +C435 +C436 +C437 +C438 +C439 +C440 +C441 +C442 +C443 +C444 +C445 +C446 +C447 +C448 +C449 +C450 +C451 +C452 +C453 +C454 +C455 +C456 +C457 +C458 +C459 +C460 +C461 +C462 +C463 +C464 +C465 +C466 +C467 +C468 +C469 +C470 +C471 +C472 +C473 +C474 +C475 +C476 +C477 +C478 +C479 +C480 +C481 +C482 +C483 +C484 +C485 +C486 +C487 +C488 +C489 +C490 +C491 +C492 +C493 +C494 +C495 +C496 +C497 +C498 +C499 +C500 +C501 +C502 +C503 +C504 +C505 +C506 +C507 +C508 +C509 +C510 +C511 +C512 +C513 +C514 +C515 +C516 +C517 +C518 +C519 +C520 +C521 +C522 +C523 +C524 +C525 +C526 +C527 +C528 +C529 +C530 +C531 +C532 +C533 +C534 +C535 +C536 +C537 +C538 +C539 +C540 +C541 +C542 +C543 +C544 +C545 +C546 +C547 +C548 +C549 +C550 +C551 +C552 +C553 +C554 +C555 +C556 +C557 +C558 +C559 +C560 +C561 +C562 +C563 +C564 +C565 +C566 +C567 +C568 +C569 +C570 +C571 +C572 +C573 +C574 +C575 +C576 +C577 +C578 +C579 +C580 +C581 +C582 +C583 +C584 +C585 +C586 +C587 +C588 +C589 +C590 +C591 +C592 +C593 +C594 +C595 +C596 +C597 +C598 +C599 +C600 +C601 +C602 +C603 +C604 +C605 +C606 +C607 +C608 +C609 +C610 +C611 +C612 +C613 +C614 +C615 +C616 +C617 +C618 +C619 +C620 +C621 +C622 +C623 +C624 +C625 +C626 +C627 +C628 +C629 +C630 +C631 +C632 +C633 +C634 +C635 +C636 +C637 +C638 +C639 +C640 +C641 +C642 +C643 +C644 +C645 +C646 +C647 +C648 +C649 +C650 +C651 +C652 +C653 +C654 +C655 +C656 +C657 +C658 +C659 +C660 +C661 +C662 +C663 +C664 +C665 +C666 +C667 +C668 +C669 +C670 +C671 +C672 +C673 +C674 +C675 +C676 +C677 +C678 +C679 +C680 +C681 +C682 +C683 +C684 +C685 +C686 +C687 +C688 +C689 +C690 +C691 +C692 +C693 +C694 +C695 +C696 +C697 +C698 +C699 +C700 +C701 +C702 +C703 +C704 +C705 +C706 +C707 +C708 +C709 +C710 +C711 +C712 +C713 +C714 +C715 +C716 +C717 +C718 +C719 +C720 +C721 +C722 +C723 +C724 +C725 +C726 +C727 +C728 +C729 +C730 +C731 +C732 +C733 +C734 +C735 +C736 +C737 +C738 +C739 +C740 +C741 +C742 +C743 +C744 +C745 +C746 +C747 +C748 +C749 +C750 +C751 +C752 +C753 +C754 +C755 +C756 +C757 +C758 +C759 +C760 +C761 +C762 +C763 +C764 +C765 +C766 +C767 +C768 +C769 +C770 +C771 +C772 +C773 +C774 +C775 +C776 +C777 +C778 +C779 +C780 +C781 +C782 +C783 +C784 +C785 +C786 +C787 +C788 +C789 +C790 +C791 +C792 +C793 +C794 +C795 +C796 +C797 +C798 +C799

+C800 +C801 +C802 +C803 +C804 +C805 +C806 +C807 +C808 +C809 +C810 +C811 +C812 +C813 +C814 +C815 +C816 +C817 +C818 +C819 +C820 +C821 +C822 +C823 +C824 +C825 +C826 +C827 +C828 +C829 +C830 +C831 +C832 +C833 +C834 +C835 +C836 +C837 +C838 +C839 +C840 +C841 +C842 +C843 +C844 +C845 +C846 +C847 +C848 +C849 +C850 +C851 +C852 +C853 +C854 +C855 +C856 +C857 +C858 +C859 +C860 +C861 +C862 +C863 +C864 +C865 +C866 +C867 +C868 +C869 +C870 +C871 +C872 +C873 +C874 +C875 +C876 +C877 +C878 +C879 +C880 +C881 +C882 +C883 +C884 +C885 +C886 +C887 +C888 +C889 +C890 +C891 +C892 +C893 +C894 +C895 +C896 +C897 +C898 +C899 +C900 +C901 +C902 +C903 +C904 +C905 +C906 +C907 +C908 +C909 +C910 +C911 +C912 +C913 +C914 +C915 +C916 +C917 +C918 +C919 +C920 +C921 +C922 +C923 +C924 +C925 +C926 +C927 +C928 +C929 +C930;

/* Constraints */
R1: +100 C901 >= 21.25; R2: +100 C902 >= 21.25; R3: +100 C903 >= 18.75; R4: +100 C904 >= 1.25; R5: +100 C905 >= 16.25; R6: +100 C906 >= 40; R7: +100 C907 >= 8.75; R8: +100 C908 >= 18.75; R9: +100 C909 >= 142.5; R10: +100 C910 >= 13.75; R11: +100 C911 >= 57.5; R12: +100 C912 >= 1.25; R13: +100 C913 >= 13.75; R14: +100 C914 >= 8.75; R15: +100 C915 >= 116.25; +C1 +C2 -C901 = 0; +C1 +C12 +C17 -C901 = 0; -C2 +C12 +C14 -C23 = 0; +C17 +C19 +C20 -C24 -C29 = 0; -C14 -C19 +C23 +C24 +C25 -C30 = 0; -C20 -C25 +C29 +C30 = 0; +C31 +C32 -C902 = 0; -C31 +C37 +C38 -C47 = 0; +C32 +C37 +C53 -C902 = 0; -C38 +C47 +C49 +C50 -C54 -C59 = 0; -C49 +C53 +C54 +C55 -C60 = 0; -C50 -C55 +C59 +C60 = 0; +C61 +C62 -C903 = 0; -C61 +C67 +C68 -C72 = 0; -C62 -C67 +C72 +C74 -C83 = 0; +C68 +C84 +C89 -C903 = 0; -C74 +C83 +C84 +C85 -C90 = 0; -C85 +C89 +C90 = 0; +C91 +C92 -C904 = 0; -C91 +C97 +C98 -C102 -C107 = 0; -C92 -C97 +C102 +C104 = 0; -C98 +C107 +C109 +C110 -C119 = 0; +C104 +C109 +C120 -C904 = 0; -C110 +C119 +C120 = 0; +C121 +C122 -C905 = 0; -C121 +C127 +C128 -C132 -C137 = 0; -C122 -C127 +C132 +C134 -C143 = 0; -C128 +C137 +C139 +C140 -C144 = 0; -C134 -C139 +C143 +C144 +C145 = 0; +C140 +C145 -C905 = 0; +C152 -C156 = 0; +C156 +C157 +C158 -C906 = 0; +C152 +C157 +C173 -C906 = 0; -C158 +C169 +C170 -C174 -C179 = 0; -C169 +C173 +C174 +C175 -C180 = 0; -C170 -C175 +C179 +C180 = 0; +C182 -C186 -C191 = 0; +C186 +C187 +C188 -C907 = 0; -C182 -C187 +C191 +C194 -C203 = 0; +C188 +C204 +C209 -C907 = 0; -C194 +C203 +C204 +C205 -C210 = 0; -C205 +C209 +C210 = 0; +C212 -C216 -C221 = 0; +C216 +C217 +C218 -C908 = 0; -C212 -C217 +C221 +C224 = 0; -C218 +C229 +C230 -C239 = 0; +C224 +C229 +C240 -C908 = 0; -C230 +C239 +C240 = 0; +C242 -C246 -C251 = 0; +C246 +C247 +C248 -C909 = 0; -C242 -C247 +C251 +C254 -C263 = 0; -C248 +C259 +C260 -C264 = 0; -C254 -C259 +C263 +C264 +C265 = 0; +C260 +C265 -C909 = 0; +C271 -C276 -C281 = 0; -C271 +C276 +C278 -C282 = 0; +C281 +C282 +C284 -C910 = 0; +C278 +C294 +C299 -C910 = 0; -C284 +C294 +C295 -C300 = 0; -C295 +C299 +C300 = 0; +C301 -C306 -C311 = 0; -C301 +C306 +C308 -C312 -C317 = 0; +C311 +C312 +C314 -C911 = 0; -C308 +C317 +C319 +C320 -C329 = 0; +C314 +C319 +C330 -C911 = 0; -C320 +C329 +C330 = 0; +C331 -C336 -C341 = 0; -C331 +C336 +C338 -C342 -C347 = 0; +C341 +C342 +C344 -C912 = 0; -C338 +C347 +C349 +C350 -C354 = 0; -C344 -C349 +C354 +C355 = 0; +C350 +C355 -C912 = 0; +C361 +C362 -C366 -C371 = 0; -C361 +C366 +C367 -C372 -C377 = 0; -C362 -C367 +C371 +C372 +C374 = 0; +C377 +C379 +C380 -C913 = 0; +C374 +C379

+C390 -C913 = 0; -C380 +C390 = 0; +C391 +C392 -C396 -C401 = 0; -C391 +C396 +C397 -C402 -C407 = 0; -C392 -C397 +C401 +C402 +C404 -C413 = 0; +C407 +C409 +C410 -C914 = 0; -C404 -C409 +C413 +C415 = 0; +C410 +C415 -C914 = 0; +C421 +C422 -C426 -C431 = 0; -C421 +C426 +C427 +C428 -C432 -C437 = 0; -C422 -C427 +C431 +C432 -C443 = 0; -C428 +C437 +C440 -C444 = 0; +C443 +C444 +C445 -C915 = 0; +C440 +C445 -C915 = 0; +C451 +C452 -C916 = 0; +C451 +C462 +C467 -C916 = 0; -C452 +C462 +C464 -C473 = 0; +C467 +C469 +C470 -C474 -C479 = 0; -C464 -C469 +C473 +C474 +C475 -C480 = 0; -C470 -C475 +C479 +C480 = 0; +C481 +C482 -C917 = 0; -C481 +C487 +C488 -C497 = 0; +C482 +C487 +C503 -C917 = 0; -C488 +C497 +C499 +C500 -C504 -C509 = 0; -C499 +C503 +C504 +C505 -C510 = 0; -C500 -C505 +C509 +C510 = 0; +C511 +C512 -C918 = 0; -C511 +C517 +C518 -C522 = 0; -C512 -C517 +C522 +C524 -C533 = 0; +C518 +C534 +C539 -C918 = 0; -C524 +C533 +C534 +C535 -C540 = 0; -C535 +C539 +C540 = 0; +C541 +C542 -C919 = 0; -C541 +C547 +C548 -C552 -C557 = 0; -C542 -C547 +C552 +C554 = 0; -C548 +C557 +C559 +C560 -C569 = 0; +C554 +C559 +C570 -C919 = 0; -C560 +C569 +C570 = 0; +C571 +C572 -C920 = 0; -C571 +C577 +C578 -C582 -C587 = 0; -C572 -C577 +C582 +C584 -C593 = 0; -C578 +C587 +C589 +C590 -C594 = 0; -C584 -C589 +C593 +C594 +C595 = 0; +C590 +C595 -C920 = 0; +C602 -C606 = 0; +C606 +C607 +C608 -C921 = 0; +C602 +C607 +C623 -C921 = 0; -C608 +C619 +C620 -C624 -C629 = 0; -C619 +C623 +C624 +C625 -C630 = 0; -C620 -C625 +C629 +C630 = 0; +C632 -C636 -C641 = 0; +C636 +C637 +C638 -C922 = 0; -C632 -C637 +C641 +C644 -C653 = 0; +C638 +C654 +C659 -C922 = 0; -C644 +C653 +C654 +C655 -C660 = 0; -C655 +C659 +C660 = 0; +C662 -C666 -C671 = 0; +C666 +C667 +C668 -C923 = 0; -C662 -C667 +C671 +C674 = 0; -C668 +C679 +C680 -C689 = 0; +C674 +C679 +C690 -C923 = 0; -C680 +C689 +C690 = 0; +C692 -C696 -C701 = 0; +C696 +C697 +C698 -C924 = 0; -C692 -C697 +C701 +C704 -C713 = 0; -C698 +C709 +C710 -C714 = 0; -C704 -C709 +C713 +C714 +C715 = 0; +C710 +C715 -C924 = 0; +C721 -C726 -C731 = 0; -C721 +C726 +C728 -C732 = 0; +C731 +C732 +C734 -C925 = 0; +C728 +C744 +C749 -C925 = 0; -C734 +C744 +C745 -C750 = 0; -C745 +C749 +C750 = 0; +C751 -C756 -C761 = 0; -C751 +C756 +C758 -C762 -C767 = 0; +C761 +C762 +C764 -C926 = 0; -C758 +C767 +C769 +C770 -C779 = 0; +C764 +C769 +C780 -C926 = 0; -C770 +C779 +C780 = 0; +C781 -C786 -C791 = 0; -C781 +C786 +C788 -C792 -C797 = 0; +C791 +C792 +C794 -C927 = 0; -C788 +C797 +C799 +C800 -C804 = 0; -C794 -C799 +C804 +C805 = 0; +C800 +C805 -C927 = 0; +C811 +C812 -C816 -C821 = 0; -C811 +C816 +C817 -C822 -C827 = 0; -C812 -C817 +C821 +C822 +C824 = 0; +C827 +C829 +C830 -C928 = 0; +C824 +C829 +C840 -C928 = 0; -C830 +C840 = 0; +C841 +C842 -C846 -C851 = 0; -C841 +C846 +C847 -C852 -C857 = 0; -C842 -C847 +C851 +C852 +C854 -C863 = 0; +C857 +C859 +C860 -C929 = 0; -C854 -C859 +C863 +C865 = 0; +C860 +C865 -C929 = 0; +C871 +C872 -C876 -C881 = 0; -C871 +C876 +C877 +C878 -C882 -C887 = 0; -C872 -C877 +C881 +C882 -C893 = 0; -C878 +C887 +C890 -C894 = 0; +C893 +C894 +C895 -C930 = 0; +C890 +C895 -C930 = 0; +C1 +C451 <= 1; +C2 +C452 <= 1; +C3 +C453 <= 1; +C4 +C454 <= 1; +C5 +C455 <= 1; +C7 +C457 <= 1; +C8 +C458 <= 1; +C9 +C459 <= 1; +C10 +C460 <= 1; +C13 +C463 <= 1; +C14 +C464 <= 1; +C15 +C465 <= 1; +C19 +C469 <= 1; +C20 +C470 <= 1; +C25 +C475 <= 1; +C31 +C481 <= 1; +C32 +C482 <= 1; +C33 +C483 <= 1; +C34 +C484 <= 1; +C35 +C485 <= 1; +C37 +C487 <= 1; +C38 +C488 <= 1; +C39 +C489 <= 1; +C40 +C490 <= 1; +C43 +C493 <= 1; +C44 +C494 <= 1; +C45 +C495 <= 1; +C49 +C499 <= 1; +C50 +C500 <= 1; +C55 +C505 <= 1; +C61 +C511 <= 1; +C62 +C512 <= 1;

+C63 +C513 <= 1; +C64 +C514 <= 1; +C65 +C515 <= 1; +C67 +C517 <= 1; +C68 +C518 <= 1; +C69 +C519 <= 1; +C70 +C520 <= 1; +C73 +C523 <= 1; +C74 +C524 <= 1; +C75 +C525 <= 1; +C79 +C529 <= 1; +C80 +C530 <= 1; +C85 +C535 <= 1; +C91 +C541 <= 1; +C92 +C542 <= 1; +C93 +C543 <= 1; +C94 +C544 <= 1; +C95 +C545 <= 1; +C97 +C547 <= 1; +C98 +C548 <= 1; +C99 +C549 <= 1; +C100 +C550 <= 1; +C103 +C553 <= 1; +C104 +C554 <= 1; +C105 +C555 <= 1; +C109 +C559 <= 1; +C110 +C560 <= 1; +C115 +C565 <= 1; +C121 +C571 <= 1; +C122 +C572 <= 1; +C123 +C573 <= 1; +C124 +C574 <= 1; +C125 +C575 <= 1; +C127 +C577 <= 1; +C128 +C578 <= 1; +C129 +C579 <= 1; +C130 +C580 <= 1; +C133 +C583 <= 1; +C134 +C584 <= 1; +C135 +C585 <= 1; +C139 +C589 <= 1; +C140 +C590 <= 1; +C145 +C595 <= 1; +C151 +C601 <= 1; +C152 +C602 <= 1; +C153 +C603 <= 1; +C154 +C604 <= 1; +C155 +C605 <= 1; +C157 +C607 <= 1; +C158 +C608 <= 1; +C159 +C609 <= 1; +C160 +C610 <= 1; +C163 +C613 <= 1; +C164 +C614 <= 1; +C165 +C615 <= 1; +C169 +C619 <= 1; +C170 +C620 <= 1; +C175 +C625 <= 1; +C181 +C631 <= 1; +C182 +C632 <= 1; +C183 +C633 <= 1; +C184 +C634 <= 1; +C185 +C635 <= 1; +C187 +C637 <= 1; +C188 +C638 <= 1; +C189 +C639 <= 1; +C190 +C640 <= 1; +C193 +C643 <= 1; +C194 +C644 <= 1; +C195 +C645 <= 1; +C199 +C649 <= 1; +C200 +C650 <= 1; +C205 +C655 <= 1; +C211 +C661 <= 1; +C212 +C662 <= 1; +C213 +C663 <= 1; +C214 +C664 <= 1; +C215 +C665 <= 1; +C217 +C667 <= 1; +C218 +C668 <= 1; +C219 +C669 <= 1; +C220 +C670 <= 1; +C223 +C673 <= 1; +C224 +C674 <= 1; +C225 +C675 <= 1; +C229 +C679 <= 1; +C230 +C680 <= 1; +C235 +C685 <= 1; +C241 +C691 <= 1; +C242 +C692 <= 1; +C243 +C693 <= 1; +C244 +C694 <= 1; +C245 +C695 <= 1; +C247 +C697 <= 1; +C248 +C698 <= 1; +C249 +C699 <= 1; +C250 +C700 <= 1; +C253 +C703 <= 1; +C254 +C704 <= 1; +C255 +C705 <= 1; +C259 +C709 <= 1; +C260 +C710 <= 1; +C265 +C715 <= 1; +C271 +C721 <= 1; +C272 +C722 <= 1; +C273 +C723 <= 1; +C274 +C724 <= 1; +C275 +C725 <= 1; +C277 +C727 <= 1; +C278 +C728 <= 1; +C279 +C729 <= 1; +C280 +C730 <= 1; +C283 +C733 <= 1; +C284 +C734 <= 1; +C285 +C735 <= 1; +C289 +C739 <= 1; +C290 +C740 <= 1; +C295 +C745 <= 1; +C301 +C751 <= 1; +C302 +C752 <= 1; +C303 +C753 <= 1; +C304 +C754 <= 1; +C305 +C755 <= 1; +C307 +C757 <= 1; +C308 +C758 <= 1; +C309 +C759 <= 1; +C310 +C760 <= 1; +C313 +C763 <= 1; +C314 +C764 <= 1; +C315 +C765 <= 1; +C319 +C769 <= 1; +C320 +C770 <= 1; +C325 +C775 <= 1; +C331 +C781 <= 1; +C332 +C782 <= 1; +C333 +C783 <= 1; +C334 +C784 <= 1; +C335 +C785 <= 1; +C337 +C787 <= 1; +C338 +C788 <= 1; +C339 +C789 <= 1; +C340 +C790 <= 1; +C343 +C793 <= 1; +C344 +C794 <= 1; +C345 +C795 <= 1; +C349 +C799 <= 1; +C350 +C800 <= 1; +C355 +C805 <= 1; +C361 +C811 <= 1; +C362 +C812 <= 1; +C363 +C813 <= 1; +C364 +C814 <= 1; +C365 +C815 <= 1; +C367 +C817 <= 1; +C368 +C818 <= 1; +C369 +C819 <= 1; +C370 +C820 <= 1; +C373 +C823 <= 1; +C374 +C824 <= 1; +C375 +C825 <= 1; +C379 +C829 <= 1; +C380 +C830 <= 1; +C385 +C835 <= 1; +C391 +C841 <= 1; +C392 +C842 <= 1; +C393 +C843 <= 1; +C394 +C844 <= 1; +C395 +C845 <= 1; +C397 +C847 <= 1; +C398 +C848 <= 1; +C399 +C849 <= 1; +C400 +C850 <= 1; +C403 +C853 <= 1; +C404 +C854 <= 1; +C405 +C855 <= 1; +C409 +C859 <= 1; +C410 +C860 <= 1; +C415 +C865 <= 1; +C421 +C871 <= 1; +C422 +C872 <= 1; +C423 +C873 <= 1; +C424 +C874 <= 1; +C425 +C875 <= 1; +C427 +C877 <= 1; +C428 +C878 <= 1; +C429 +C879 <= 1; +C430 +C880 <= 1; +C433 +C883 <= 1; +C434 +C884 <= 1; +C435 +C885 <= 1; +C439 +C889 <= 1; +C440 +C890 <= 1; +C445 +C895 <= 1; +C1 +C6 +C31 +C36 +C61 +C66 +C91 +C96 +C121 +C126 +C151 +C156 +C181 +C186

+C211 +C216 +C241 +C246 +C271 +C276 +C301 +C306 +C331 +C336 +C361 +C366 +C391 +C396 +C421 +C426 +C451 +C456 +C481 +C486 +C511 +C516 +C541 +C546 +C571 +C576 +C601 +C606 +C631 +C636 +C661 +C666 +C691 +C696 +C721 +C726 +C751 +C756 +C781 +C786 +C811 +C816 +C841 +C846 +C871 +C876 <= 80; +C2 +C11 +C32 +C41 +C62 +C71 +C92 +C101 +C122 +C131 +C152 +C161 +C182 +C191 +C212 +C221 +C242 +C251 +C272 +C281 +C302 +C311 +C332 +C341 +C362 +C371 +C392 +C401 +C422 +C431 +C452 +C461 +C482 +C491 +C512 +C521 +C542 +C551 +C572 +C581 +C602 +C611 +C632 +C641 +C662 +C671 +C692 +C701 +C722 +C731 +C752 +C761 +C782 +C791 +C812 +C821 +C842 +C851 +C872 +C881 <= 80; +C3 +C16 +C33 +C46 +C63 +C76 +C93 +C106 +C123 +C136 +C153 +C166 +C183 +C196 +C213 +C226 +C243 +C256 +C273 +C286 +C303 +C316 +C333 +C346 +C363 +C376 +C393 +C406 +C423 +C436 +C453 +C466 +C483 +C496 +C513 +C526 +C543 +C556 +C573 +C586 +C603 +C616 +C633 +C646 +C663 +C676 +C693 +C706 +C723 +C736 +C753 +C766 +C783 +C796 +C813 +C826 +C843 +C856 +C873 +C886 <= 0; +C4 +C21 +C34 +C51 +C64 +C81 +C94 +C111 +C124 +C141 +C154 +C171 +C184 +C201 +C214 +C231 +C244 +C261 +C274 +C291 +C304 +C321 +C334 +C351 +C364 +C381 +C394 +C411 +C424 +C441 +C454 +C471 +C484 +C501 +C514 +C531 +C544 +C561 +C574 +C591 +C604 +C621 +C634 +C651 +C664 +C681 +C694 +C711 +C724 +C741 +C754 +C771 +C784 +C801 +C814 +C831 +C844 +C861 +C874 +C891 <= 0; +C5 +C26 +C35 +C56 +C65 +C86 +C95 +C116 +C125 +C146 +C155 +C176 +C185 +C206 +C215 +C236 +C245 +C266 +C275 +C296 +C305 +C326 +C335 +C356 +C365 +C386 +C395 +C416 +C425 +C446 +C455 +C476 +C485 +C506 +C515 +C536 +C545 +C566 +C575 +C596 +C605 +C626 +C635 +C656 +C665 +C686 +C695 +C716 +C725 +C746 +C755 +C776 +C785 +C806 +C815 +C836 +C845 +C866 +C875 +C896 <= 0; +C7 +C12 +C37 +C42 +C67 +C72 +C97 +C102 +C127 +C132 +C157 +C162 +C187 +C192 +C217 +C222 +C247 +C252 +C277 +C282 +C307 +C312 +C337 +C342 +C367 +C372 +C397 +C402 +C427 +C432 +C457 +C462 +C487 +C492 +C517 +C522 +C547 +C552 +C577 +C582 +C607 +C612 +C637 +C642 +C667 +C672 +C697 +C702 +C727 +C732 +C757 +C762 +C787 +C792 +C817 +C822 +C847 +C852 +C877 +C882 <= 80; +C8 +C17 +C38 +C47 +C68 +C77 +C98 +C107 +C128 +C137 +C158 +C167 +C188 +C197 +C218 +C227 +C248 +C257 +C278 +C287 +C308 +C317 +C338 +C347 +C368 +C377 +C398 +C407 +C428 +C437 +C458 +C467 +C488 +C497 +C518 +C527 +C548 +C557 +C578 +C587 +C608 +C617 +C638 +C647 +C668 +C677 +C698 +C707 +C728 +C737 +C758 +C767 +C788 +C797 +C818 +C827 +C848 +C857 +C878 +C887 <= 80; +C9 +C22 +C39 +C52 +C69 +C82 +C99 +C112 +C129 +C142 +C159 +C172 +C189 +C202 +C219 +C232 +C249 +C262 +C279 +C292 +C309 +C322 +C339 +C352 +C369 +C382 +C399 +C412 +C429 +C442 +C459 +C472 +C489 +C502 +C519 +C532 +C549 +C562 +C579 +C592 +C609 +C622 +C639 +C652 +C669 +C682 +C699 +C712 +C729 +C742 +C759 +C772 +C789 +C802 +C819 +C832 +C849 +C862 +C879 +C892 <= 0; +C10 +C27 +C40 +C57 +C70 +C87 +C100 +C117 +C130 +C147 +C160 +C177 +C190 +C207 +C220 +C237 +C250 +C267 +C280 +C297 +C310 +C327 +C340 +C357 +C370 +C387 +C400 +C417 +C430 +C447 +C460 +C477 +C490 +C507 +C520 +C537 +C550 +C567 +C580 +C597 +C610 +C627 +C640 +C657 +C670 +C687 +C700 +C717 +C730 +C747 +C760 +C777 +C790 +C807 +C820 +C837 +C850 +C867 +C880 +C897 <= 0; +C13 +C18 +C43 +C48 +C73 +C78 +C103 +C108 +C133 +C138 +C163 +C168 +C193 +C198 +C223 +C228 +C253 +C258 +C283 +C288 +C313 +C318 +C343

+C348 +C373 +C378 +C403 +C408 +C433 +C438 +C463 +C468 +C493 +C498 +C523 +C528 +C553 +C558 +C583 +C588 +C613 +C618 +C643 +C648 +C673 +C678 +C703 +C708 +C733 +C738 +C763 +C768 +C793 +C798 +C823 +C828 +C853 +C858 +C883 +C888 <= 0; +C14 +C23 +C44 +C53 +C74 +C83 +C104 +C113 +C134 +C143 +C164 +C173 +C194 +C203 +C224 +C233 +C254 +C263 +C284 +C293 +C314 +C323 +C344 +C353 +C374 +C383 +C404 +C413 +C434 +C443 +C464 +C473 +C494 +C503 +C524 +C533 +C554 +C563 +C584 +C593 +C614 +C623 +C644 +C653 +C674 +C683 +C704 +C713 +C734 +C743 +C764 +C773 +C794 +C803 +C824 +C833 +C854 +C863 +C884 +C893 <= 80; +C15 +C28 +C45 +C58 +C75 +C88 +C105 +C118 +C135 +C148 +C165 +C178 +C195 +C208 +C225 +C238 +C255 +C268 +C285 +C298 +C315 +C328 +C345 +C358 +C375 +C388 +C405 +C418 +C435 +C448 +C465 +C478 +C495 +C508 +C525 +C538 +C555 +C568 +C585 +C598 +C615 +C628 +C645 +C658 +C675 +C688 +C705 +C718 +C735 +C748 +C765 +C778 +C795 +C808 +C825 +C838 +C855 +C868 +C885 +C898 <= 0; +C19 +C24 +C49 +C54 +C79 +C84 +C109 +C114 +C139 +C144 +C169 +C174 +C199 +C204 +C229 +C234 +C259 +C264 +C289 +C294 +C319 +C324 +C349 +C354 +C379 +C384 +C409 +C414 +C439 +C444 +C469 +C474 +C499 +C504 +C529 +C534 +C559 +C564 +C589 +C594 +C619 +C624 +C649 +C654 +C679 +C684 +C709 +C714 +C739 +C744 +C769 +C774 +C799 +C804 +C829 +C834 +C859 +C864 +C889 +C894 <= 80; +C20 +C29 +C50 +C59 +C80 +C89 +C110 +C119 +C140 +C149 +C170 +C179 +C200 +C209 +C230 +C239 +C260 +C269 +C290 +C299 +C320 +C329 +C350 +C359 +C380 +C389 +C410 +C419 +C440 +C449 +C470 +C479 +C500 +C509 +C530 +C539 +C560 +C569 +C590 +C599 +C620 +C629 +C650 +C659 +C680 +C689 +C710 +C719 +C740 +C749 +C770 +C779 +C800 +C809 +C830 +C839 +C860 +C869 +C890 +C899 <= 80; +C25 +C30 +C55 +C60 +C85 +C90 +C115 +C120 +C145 +C150 +C175 +C180 +C205 +C210 +C235 +C240 +C265 +C270 +C295 +C300 +C325 +C330 +C355 +C360 +C385 +C390 +C415 +C420 +C445 +C450 +C475 +C480 +C505 +C510 +C535 +C540 +C565 +C570 +C595 +C600 +C625 +C630 +C655 +C660 +C685 +C690 +C715 +C720 +C745 +C750 +C775 +C780 +C805 +C810 +C835 +C840 +C865 +C870 +C895 +C900 <= 80;

/* Integer definitions */
int C1, C2, C3, C4, C5, C6, C7, C8, C9, C10, C11, C12, C13, C14, C15, C16, C17, C18, C19, C20, C21, C22, C23, C24, C25, C26, C27, C28, C29, C30, C31, C32, C33, C34, C35, C36, C37, C38, C39, C40, C41, C42, C43, C44, C45, C46, C47, C48, C49, C50, C51, C52, C53, C54, C55, C56, C57, C58, C59, C60, C61, C62, C63, C64, C65, C66, C67, C68, C69, C70, C71, C72, C73, C74, C75, C76, C77, C78, C79, C80, C81, C82, C83, C84, C85, C86, C87, C88, C89, C90, C91, C92, C93, C94, C95, C96, C97, C98, C99, C100, C101, C102, C103, C104, C105, C106, C107, C108, C109, C110, C111, C112, C113, C114, C115, C116, C117, C118, C119, C120, C121, C122, C123, C124, C125, C126, C127, C128, C129, C130, C131, C132, C133, C134, C135, C136, C137, C138, C139, C140, C141, C142, C143, C144, C145, C146, C147, C148, C149, C150, C151, C152, C153, C154, C155, C156, C157, C158, C159, C160, C161, C162, C163, C164, C165, C166, C167, C168, C169, C170, C171, C172, C173, C174, C175, C176, C177, C178, C179, C180, C181, C182, C183, C184, C185, C186, C187, C188, C189, C190, C191, C192, C193, C194, C195, C196, C197, C198, C199, C200, C201, C202, C203, C204, C205, C206, C207, C208, C209, C210, C211, C212, C213, C214, C215, C216, C217, C218, C219, C220, C221, C222, C223, C224, C225, C226, C227, C228,

C229, C230, C231, C232, C233, C234, C235, C236, C237, C238, C239, C240, C241, C242, C243, C244, C245, C246, C247, C248, C249, C250, C251, C252, C253, C254, C255, C256, C257, C258, C259, C260, C261, C262, C263, C264, C265, C266, C267, C268, C269, C270, C271, C272, C273, C274, C275, C276, C277, C278, C279, C280, C281, C282, C283, C284, C285, C286, C287, C288, C289, C290, C291, C292, C293, C294, C295, C296, C297, C298, C299, C300, C301, C302, C303, C304, C305, C306, C307, C308, C309, C310, C311, C312, C313, C314 ,C315, C316, C317, C318, C319, C320, C321, C322, C323, C324, C325, C326, C327, C328, C329, C330, C331, C332, C333, C334, C335, C336, C337, C338, C339, C340, C341, C342, C343, C344, C345, C346, C347, C348, C349, C350, C351, C352, C353, C354, C355, C356, C357, C358, C359, C360, C361, C362, C363, C364, C365, C366, C367, C368, C369, C370, C371, C372, C373, C374, C375, C376, C377, C378, C379, C380, C381, C382, C383, C384, C385, C386, C387, C388, C389, C390, C391, C392, C393, C394, C395, C396, C397, C398, C399, C400, C401, C402, C403, C404, C405, C406, C407, C408, C409, C410, C411, C412, C413, C414, C415, C416, C417, C418, C419, C420, C421, C422, C423, C424, C425, C426, C427, C428, C429, C430, C431, C432, C433, C434, C435, C436, C437, C438, C439, C440, C441, C442, C443, C444, C445, C446, C447, C448, C449, C450, C451, C452, C453, C454, C455, C456, C457, C458, C459, C460, C461, C462, C463, C464, C465, C466, C467, C468, C469, C470, C471, C472, C473, C474, C475, C476, C477, C478, C479, C480, C481, C482, C483, C484, C485, C486, C487, C488, C489, C490, C491, C492, C493, C494, C495, C496, C497, C498, C499, C500, C501, C502, C503, C504, C505, C506, C507, C508, C509, C510, C511, C512, C513, C514, C515, C516, C517, C518, C519, C520, C521, C522, C523, C524, C525, C526, C527, C528, C529, C530, C531, C532, C533, C534, C535, C536, C537, C538, C539, C540, C541, C542, C543, C544, C545, C546, C547, C548, C549, C550, C551, C552, C553, C554, C555, C556, C557, C558, C559, C560, C561, C562, C563, C564, C565, C566, C567, C568, C569, C570, C571, C572, C573, C574, C575, C576, C577, C578, C579, C580, C581, C582, C583, C584, C585, C586, C587, C588, C589, C590, C591, C592, C593, C594, C595, C596, C597, C598, C599, C600, C601, C602, C603, C604, C605, C606, C607, C608, C609, C610, C611, C612, C613, C614, C615, C616, C617, C618, C619, C620, C621, C622, C623, C624, C625, C626, C627, C628, C629, C630, C631, C632, C633, C634, C635, C636, C637, C638, C639, C640, C641, C642, C643, C644, C645, C646, C647, C648, C649, C650, C651, C652, C653, C654, C655, C656, C657, C658, C659, C660, C661, C662, C663, C664, C665, C666, C667, C668, C669, C670, C671, C672, C673, C674, C675, C676, C677, C678, C679, C680, C681, C682, C683, C684, C685, C686, C687, C688, C689, C690, C691, C692, C693, C694, C695, C696, C697, C698, C699, C700, C701, C702, C703, C704, C705, C706, C707, C708, C709, C710, C711, C712, C713, C714, C715, C716, C717, C718, C719, C720, C721, C722, C723, C724, C725, C726, C727, C728, C729, C730, C731, C732, C733, C734, C735, C736, C737, C738, C739, C740, C741, C742, C743, C744, C745, C746, C747, C748, C749, C750, C751, C752, C753, C754, C755, C756, C757, C758, C759, C760, C761, C762, C763, C764, C765, C766, C767, C768, C769, C770, C771, C772, C773, C774, C775, C776, C777, C778, C779, C780, C781, C782, C783, C784, C785, C786, C787, C788, C789, C790, C791, C792, C793, C794, C795, C796, C797, C798, C799, C800, C801, C802, C803, C804, C805, C806, C807, C808, C809, C810, C811, C812, C813 ,C814, C815, C816, C817, C818, C819, C820, C821, C822, C823, C824, C825, C826, C827, C828, C829, C830, C831, C832, C833, C834, C835, C836, C837, C838, C839, C840, C841, C842, C843, C844, C845, C846, C847, C848, C849, C850, C851, C852, C853, C854, C855, C856, C857, C858, C859, C860, C861, C862, C863, C864, C865, C866, C867, C868, C869, C870, C871, C872, C873, C874, C875, C876, C877, C878, C879, C880, C881, C882, C883, C884, C885, C886, C887, C888, C889, C890, C891, C892, C893, C894, C895, C896, C897, C898, C899, C900, C901, C902, C903, C904, C905, C906, C907, C908, C909, C910, C911, C912, C913, C914, C915, C916, C917, C918, C919, C920, C921, C922, C923, C924, C925, C926, C927, C928, C929, C930;

5.9.3 Results using LPSolve

5.10 Translucent with 1+1 protection appendices

Student Name	:	Tiago Esteves (November 28, 2017 -)
Goal	:	Implement the dimensioning of optical networks in the translucent transport mode.

5.10.1 Script using MatLab

5.10.1.1 Scenario 1

5.10.1.2 Scenario 2

5.10.1.3 Scenario 3

5.10.1.4 Scenario 4

5.10.2 ILP using LPSolve

5.10.3 Results using LPSolve

Capítulo 6

Master Dissertations

- Vasco Braz, Dimensioning and Optimization of Node Architecture in Optical Transport Networks, University of Aveiro, 2016

Universidade de Aveiro Departamento de Electrónica, Telecomunicações e Informática,
2016

Vasco Rafael Brites dos Santos Braz

Dimensionamento e Optimização da Arquitetura dos Nós em Redes de Trasporte Óticas

Dimensioning and Optimization of Node Architecture in Optical Transport Networks

Universidade de Aveiro
2016
Departamento de
Electrónica, Telecomunicações e Informática,

Vasco Rafael Brites dos Santos Braz

Dimensionamento e Optimização da Arquitetura dos Nós em Redes de Trasporte Óticas

Dimensioning and Optimization of Node Architecture in Optical Transport Networks

"Anyone who has never made a mistake has never tried anything new."

— Albert Einstein

Universidade de Aveiro Departamento de Electrónica, Telecomunicações e Informática,
2016

Vasco Rafael Brites dos Santos Braz

Dimensionamento e Optimização da Arquitetura dos Nós em Redes de Trasporte Óticas

Dimensioning and Optimization of Node Architecture in Optical Transport Networks

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Eletrónica e Telecomunicações, realizada sob a orientação científica do Dr. Armando Humberto Moreira Nolasco Pinto, Professor Associado do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro, coorientação empresarial do Dr. Rui Manuel Dias Morais, doutorado em Engenharia Eletrotécnica pela Universidade de Aveiro, coordenador de atividades de investigação em optimização de redes na Coriant Portugal.

o júri / the jury

presidente / president — **Mário José Neves de Lima**
Professor Auxiliar da Universidade de Aveiro

vogais / examiners committee — **Maria do Carmo Raposo de Medeiros**
Professora Associada da Universidade de Coimbra

Armando Humberto Moreira Nolasco Pinto
Professor Associado da Universidade de Aveiro

agradecimentos / acknowledgements

Em primeiro lugar agradecer a orientação rigorosa e sempre disponível do Prof. Doutor Armando Pinto e a coorientação eficaz do Doutor Rui Morais.

Agradeço também ao projeto UID/EEA/50008/2013 (ação SoftTransceiver - Software Defined Transceiver for Flexible Optical Networks) pelo apoio prestado.

Queria agradecer especialmente à minha mãe que tanto fez por este momento.

Resumo

Nesta dissertação é apresentada uma introdução às redes de transporte óticas multicamada. Foram caracterizados os dois elementos principais da rede: nós e ligações. Ao nível das ligações foi feita uma abordagem baseada nos seus elementos físicos principais. Ao nível dos nós foram tidos em consideração o tráfego de cliente (baixo débito) e o tráfego de linha (alto débito), bem como os componentes necessários para os transportar. A forma como o tráfego de cliente é agregado e o encaminhamento do mesmo na rede, exigem a elaboração de uma arquitetura que minimize os recursos necessários. A necessidade de otimizar este processo de dimensionamento da rede levou à construção e validação de métodos de agregação de tráfego e encaminhamento baseados em topologias lógicas da rede. Assim, proponho nesta dissertação algoritmos de agregação e encaminhamento aplicados a um software livre, previamente validados por modelos de programação linear baseados em restrições e funções objectivo adequadas à topologia pretendida. A apresentação detalhada dos resultados considerando o CAPEX, bem como a sua análise são considerados na dissertação. Por fim, são apresentadas conclusões e sugerido o trabalho científico que ainda pode ser realizado neste âmbito.

Abstract

In this dissertation an introduction is presented to the multilayer optical transport networks. The two main elements of the network were characterized: nodes and links. Regarding the connections it was made a shallower approach based on its key physical elements. In terms of nodes client traffic (low bandwith) and the line traffic (high bandwith) were considered as well as the components necessary to carry them. The way the client traffic is aggregated and its forwarding in the same network requires an architecture which makes use of the minimum resources. The need of optimizing this network design process led to the construction and validation of traffic aggregation methods and routing based on logical network topologies. I therefore propose in this dissertation routing and grooming algorithms applied to a open source software, previously validated by linear programming models based on constraints and objective functions suitable to the desired topology. A detailed presentation of the results considering the CAPEX and its analysis are also taken into account. Finally, conclusions are presented and the scientific work that can still be done in this area is suggested.

Contents

Contents **i**

List of Figures **iii**

List of Tables **v**

List of Acronyms **vii**

1 Introduction **1**
1.1 Motivation . . . 2
1.2 Objectives . . . 3
1.3 Thesis outline . . . 4

2 Network Design and Dimensioning **5**
2.1 Optical Networks . . . 6
2.1.1 Links Arquitecture . . . 6
2.1.2 Nodes Arquitecture . . . 7
2.1.2.1 Grooming . . . 7
2.1.2.2 Switching . . . 8
2.1.2.2.1 Electrical Switching . . . 8
2.1.2.2.2 Optical Switching . . . 9
2.1.3 Physical Topology . . . 10
2.1.4 Logical Topology . . . 11
2.2 Transport modes . . . 11
2.2.1 Opaque . . . 11
2.2.2 Transparent . . . 12
2.2.3 Translucent . . . 12
2.3 Network Survivability . . . 14
2.4 Chapter summary . . . 15
References . . . 15

3 Integer Linear Programming Models **17**
3.1 Network protection topological design problem . . . 18
3.2 ILP models . . . 19
3.2.1 Opaque with 1+1 Protection . . . 19
3.2.2 Transparent with 1+1 Protection . . . 21
3.3 LPSolve and Matlab implementation . . . 22

3.4 ILP results . . . 23
3.5 Chapter Summary . . . 25
References . . . 25

4 Heuristics **27**
4.1 Heuristic routing . . . 28
4.2 Heuristic grooming . . . 30
4.3 Net2Plan Aproach . . . 31
4.3.1 Join Traffic Matrices Alghoritm . . . 32
4.3.2 Logical Topology Alghoritm . . . 33
4.3.3 Grooming and Routing Alghoritm . . . 34
4.3.4 Reports . . . 35
4.4 Heuristic results validation . . . 36
4.5 Chapter Summary . . . 38

5 Case Study **39**
5.1 NSFNET . . . 40
5.2 Network Cost . . . 43
5.3 Opaque topology results . . . 45
5.4 Transparent topology results . . . 46
5.5 Chapter Summary . . . 47

6 Conclusions and future directions **48**
6.1 Conclusions . . . 48
6.2 Future directions . . . 49

References **50**

Appendices **53**

List of Figures

1.1 Cisco forecast for ip traffic growth [1] 1

2.1 Bidirectional transmission system architecture: optical fiber and inline optical amplifiers [2] 6
2.2 Node schematic: port, module, slot, shelf and rack [2] 7
2.3 Grooming of OTN Signals [2] 8
2.4 Scheme of a node for opaque transport mode network 9
2.5 Scheme of a node using electrical an optical switching 9
2.6 Nodes position and connection established by links located according to the real optical network 10
2.7 Opaque transport mode with a link-by-link grooming scheme 11
2.8 Logical topology for previous physical topology with each node connected to each other (Transparent Transport Mode) 12
2.9 Translucent logical topologies 13
2.10 Network protection scheme 14

3.1 Traffic demand path from node ”o” to node ”d” 18
3.2 Example of two nodes in a network sending traffic through disjoint links . . . 20
3.3 LPSolve integrated development interface 22
3.4 ILP solution for ODU0 client traffic 23
3.5 ILP solution for ODU1 client traffic 24
3.6 ILP solution for ODU2 client traffic 24
3.7 ILP solution for ODU3 client traffic 24
3.8 ILP solution changing client traffic type 25

4.1 Heuristic routing algorithm code flow 28
4.2 Heuristic routing algorithm code flow 29
4.3 Heuristic grooming algorithm code flow 30
4.4 Net2Plan user interface for traffic matrices creation 31
4.5 Net2Plan user interface for reference network topological design 32
4.6 Net2Plan join traffic matrices alghoritm input parameters 33
4.7 Net2Plan logical topology for reference network physical topology based on a) Opaque transport mode b)Transparent transport mode 33
4.8 Net2Plan route using *grooming dedicated* alghoritm 34
4.9 Reference network link information using random traffic matrices 35
4.10 Heuristic solution for ODU0 client traffic 36
4.11 Heuristic solution for ODU1 client traffic 36

4.12 Heuristic solution for ODU2 client traffic 37
4.13 Heuristic solution for ODU3 client traffic 37
4.14 Heuristic comparative results for heterogeneous traffic 37

5.1 NSF Network topology [3]. 40
5.2 Net2Plan user interface for NSFNET topological network design 41
5.3 Net2Plan logical topology for NSFNET physical topology based on a) Opaque transport mode b)Transparent transport mode 42
5.4 Protection traffic route between Seattle and Palo Alto 42
5.5 Network cost inputs 45
5.6 Opaque ports per node 45
5.7 Opaque cost given by Net2Plan 46
5.8 Transparent ports per node 46
5.9 Transparent cost given by Net2Plan 47

List of Tables

2.1 Transport mode comparation 13

3.1 Inputs and variables of opaque and transparent transport modes 19

List of Acronyms

API	aplication user interface
EDFA	erbium doped fiber amplifier
EXC	electrical cross connect
CAPEX	capital expenditures
IDE	integrated development interface
ILP	integer linear programming
ITU-T	international telecommunication union - telecommunication standardization
IP	Internet protocol
MPLS-TP	multi protocol label switching - transport profile
M2M	machine-to-machine
OTN	optical transport network
OEO	optica-electrical-optical
OPEX	operational expenditures
OXC	optical cross connect
OTN	optical transport network
TCO	total cost of ownership
SDH	synchronous digital hierarchy
WAM	wide area network
WDM	wavelength division multiplexing

Chapter 1

Introduction

Systems based on Machine-to-Machine (M2M) communication, Voice over Internet Protocol (VOIP) services, online gaming and data storage are considered penetration services. These services have led to an overhead on existing networks caused by a huge number of users [4][5][6]. As it increases the amount of traffic and the transmission rates required, telecommunication agents involved are undergoing a pressure to develop the network technology in their processes, in order to increase the bandwidth.[7]

Technological development has allowed lower prices for network components, however, these prices are similar to the major operators. The costs with the setup of the infrastructure, CAPEX, compromises physical network components, buildings where they are installed and software needed for network operation and management. Currently, almost all traffic is transported through optical networks, so the operators are very interested in reducing the cost per transported bit as much as possible without compromising the quality of service. At the same time, there are entities concerned about the ecological footprint because of the uncontrolled increase in energy reserved for telecommunications. Therefore, it is inevitable to create or enhance methods and equipment compatible with networks already realized.So, to lead the market competition, those operators are making an effort to optimize their networks, using the minimum resources with the minimum power consumption. Hence, the outcome of reducing power consumption is the diminish of operational costs (OPEX) and a crucial step to save the planet. The result of summing capital expenditures and operational expenditures is the total cost of ownership (TCO).[8]

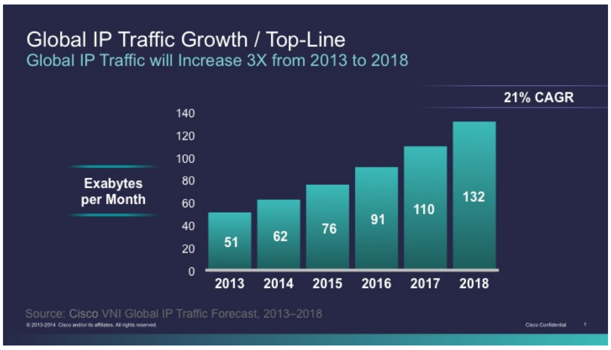


Figure 1.1: Cisco forecast for ip traffic growth [1]

1.1 Motivation

Scientific researchers in the academic and business scope are joining efforts to improve network planning processes. In an optical network there are inevitable problems that occur such as, excessive traffic in one certain node or link and physical damage that may happen in the components of the network, this type of problems can block the flow of traffic and cause a huge loss of information if the network is not properly designed. The consequences of a failure of a single fiber can be catastrophic and may affect the medical emergency number, banking operations, or many other important services, that's why the study of protection and survivability methods are so crucial. In this context the operators are obliged to invest in more equipment and new strategies that ensure the protection and survivability of the network [9].

Nonetheless, it would be extremely difficult to make a fast and scalable planning to an optical network by hand, so it is used a network planning tool which is purposeful for systems vendors and network operators. It is used in the various stages of the telecommunications business, in the budgeting stage a planning tool offers a cost-efficient solution has a enormous benefit in a competing environment. However, the needs of people in terms of telecommunications are constantly changing, so in the operation stage, the planning tool can be used to re-optimize the available resources, making possible additional cost savings to network operators. Many companies as Coriant, Cisco and Ericson operating in the telecommunications area, developed their own optical network planners [2]. However, it is evident that these tools are not publicly available to perform comparative studies of the obtained solutions for corporative reasons.

Therefore, in this dissertation, it is used an open source software (Net2Plan) as a platform to achieve a heuristic solution.Heuristic algorithms tend to be relatively fast, scalable, and suitable for huge networks. The optimal solutions can be obtained through ILP models. However, scalability limitations may arise depending on the computational resources available and the network scale. Nevertheless, the type of solutions obtained and the models themselves can give an insight in key and structural aspects of the problem and are crucial to verify the quality of heuristic algorithms.These dimensioning tools, process input parameters and generate outputs regarding optical network constraints, in order to obtain a solution focused on cost efficiency.

1.2 Objectives

This dissertation emerges from the collaboration between University of Aveiro, Instituto de Telecomunicações Aveiro, and Coriant Portugal due to the interest of network operators in saving some resources inherent to the planning and operation of their networks and also contribute with knowledge to the scientific community of optical networks. To achieve the main objectives of this dissertation, the following steps should be fulfilled:

1. Develop ILP models for opaque and transparent networks using dedicated path protection.
2. Get a solution for optical networks with protection dimensioning through heuristic algorithms applied in Net2Plan (open source platform).
3. Compare and validate heuristics based results with the ILP based results.
4. Apply heuristic method to a realistic network.

1.3 Thesis outline

This dissertation is organized in 6 chapters. Chapter 2 is a state-of-art of optical networks topological design, dimensioning and dedicated protection. The basic concepts and fundamentals are defined as well as the physical and logical topology. In terms of physical topology, basic optical network elements are described in that chapter. The relation between logical topology and transport modes is presented in detail. Chapter 3 starts by a reference network dimensioning problem analysis and an integer linear programming approach. It begins with a problem definition, and the set of variables and inputs used. Hence, the remaining of the chapter is devoted to propose dimensioning models for optical networks based on opaque and transparent transport modes and the consequent results. In Chapter 4, a heuristic approach is suggested and described. Thus, an open source software was chosen to implement this solution to be accessible to students and modifiable for educational purposes. An overview of the software used to create the algorithms on the platform used (Net2Plan). The pseudo code behind the routing and grooming is also contained in this chapter. The solution obtained through integer linear programming is considered the optimal solution and served to evaluate the quality of the solution based on heuristics. The optical channels needed for a specified client traffic matrix as a function of the algorithms runtime or as a function of different client traffic type are some of the results presented on the results chapter. The results of this two approaches are in Chapter 5. In this chapter and after validating the heuristics, they are applied to a real network and the results will be analyzed. The last step is the conclusions 6 and suggestions for future research directions.

Chapter 2

Network Design and Dimensioning

Internet is becoming one of the most powerful platforms allowing entertainment and productivity. Some services, as cloud storage or online social networks, have been implemented over Internet and they require a suitable network design for huge traffic requirements and must be able to deal, efficiently, with time-varying traffic [10]. Thus, it is mandatory to perform the best arrangement for physical resources, in order to simplify future improvements and ensure a reliable and efficient network. So, it is necessary to consider an overview of this chapter, starting from the optical reference network approach and the definition of main elements (links and nodes). As described on section 2.1.3, node mapping and links connecting them are considered physical topology.

Optical links establish the connection between two adjacent nodes, ensuring the transmission of optical signals between them. Nodes can perform six main functions: encapsulation; electrical switching; deterministic or statistical multiplexing (grooming); wavelength assignment; optical switching; and optical multiplexing. This chapter is more focused on grooming tasks. Up to now, the majority of the functions performed at the nodes are only capable of being carried out in the electrical domain. However, an optical signal may suffer optical-electrical-optical (OEO) conversions in intermediate nodes. The consequence of this clear optical channels inside a wide area network(WAN) is a virtual topology as known as optical or logical topology 2.1.4. According to all-optical fragments in a single lightpath, three transport modes will emerge: opaque, transparent and translucent.

In section 2.2, each transport mode is detailed, clarified and exemplified. An efficient topological network design makes optical networks more and more robust and reliable, but not infallible. An alternative for network working paths should be considered, regarding the limit of failure recovery time established by standardized protocols. Throughout the dissertation, only dedicated protection implementation strategies are considered 2.3. Thus, considering protection schemes and working paths for each traffic demand, it is possible to suggest an approach for node dimensioning, including optical and electrical components and support structures. The consequence of a well designed network is the minimization of node components and,as a result, lower capital expenditures (CAPEX) [11].

2.1 Optical Networks

Optical technologies are widely used in telecommunication networks, and currently they constitute the central physical network elements in most parts of the world, thanks to their high speed, large capacity, and other attractive characteristics. Optical networks can be classified as core, metro and access networks. The backbone infrastructure of telecommunication networks are core networks, which interconnects large cities (as network nodes), and spans nationwide, continental, and even intercontinental distances connected by links.

2.1.1 Links Arquitecture

Links are basically the connection between two adjacent nodes. In transport networks links are physical point-to-point connections ensured by transmission systems. The transmission system starts and ends in adjacent nodes, i.e. in directly connected nodes. Signals are transmitted through a pair of fibers required bidirectional communication.

Optical signal is sent through optical fiber and this propagation causes signal deterioration and produce linear and non-linear effects. Loss coefficient imposed by optical fibers is a physical parameter described on each different fiber datasheet. In order to compensate attenuation, transmission systems contain optical amplifiers spaced by an expected distance (span) enough to increase signal power to allow a reliable signal detection at the receivers.

In figure 2.1 each colored line corresponds to an optical channel which is associated a wavelength. The WDM signal requires WDM transmission systems in the ends of the link.

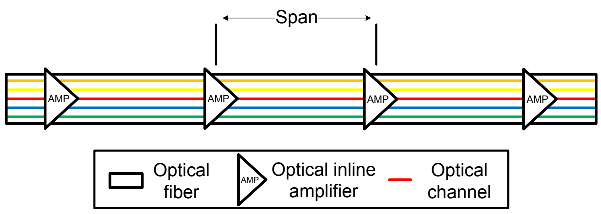


Figure 2.1: Bidirectional transmission system architecture: optical fiber and inline optical amplifiers [2]

Depending on the bandwidth used, WDM signals suffer different attenuation. Although, all channels are simultaneously amplified. An amplifier acts over the band selected, for example, EDFA (Erbium Doped Fiber Amplifier) is the most deployed fiber amplifier as its amplification window coincides with the third transmission window (1530nm –1570nm) of silica-based optical fiber [12].

2.1.2 Nodes Arquitecture

The more significant operations that the optical signal goes through in the network occur in nodes. Node operations require a lot of hardware (e.g. processors, modules, control modules, short-reach and long-reach transceivers), therefore nodes are generally considered the most expensive element of an optical transport network. In optical networks, nodes are composed of three essential structures: modules, shelves and rack. The modules can contain multiple ports. Shelves are the place where different modules are assembled and these shelves are contained within a larger structure, capable of providing provide sufficient power for each shelf. In high-capacity multilayer transport nodes, depending on network operation requirements in optical and electrical layer, different type of modules are used. Modules comprise electrical and optical components in order to perform well-defined functionalities as encapsulation, grooming and wavelength assignment. Connection between modules are established through front panel and backplane. Usually, front panel connections are mainly ensured by transceivers located in the ports. Depending on the optical signal travel distance, long-reach or short-reach transmitters must be used [13].

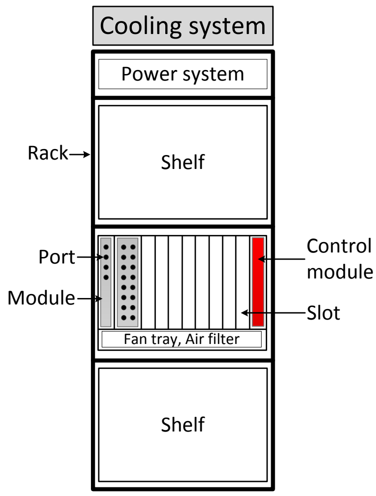


Figure 2.2: Node schematic: port, module, slot, shelf and rack [2]

2.1.2.1 Grooming

Traffic grooming refers to techniques used to combine low-speed traffic streams onto high-speed wavelengths in order to minimize the networkwide cost in terms of line terminating equipment and/or electronic switching. Network planners need to define which mixture of client signals should be aggregated as well as in which node will be grooming capable components, in order to improve the dimensioning of a multilayer node. Dimensioning models are closely related to the grooming operation as it determines the number of ports and wavelengths required. These models rely on ILP or heuristics, which require complete information about the network topology and traffic demands. In order to minimize the number of wavelengths used and, consequently, power consumption, network planners started with the aggregation of SDH/SONET traffic. The optimization of multi-layer networks has been focused on the two layer IP-over-WDM, or the three layer IP-over-OTN-over-WDM architectures. All optimization models of this dissertation are focused on OTN technology defined in ITU-T recommendation G.709 however the approaches and models are flexible to support other technologies such as SDH/SONET or Ethernet. The set of all client bit rates is denoted by C = $\{c : c \in \{1.25, 2.5, 10, 40, 100\}\}$ corresponding to an ODU1, ODU2, ODU3 and ODU4, these lower bit rate client signals, defined in Gbps, are then groomed to form a higher bit rate line signal, l. The set of line bit rates, defined in Gbps, will be denoted by L = $\{l : l \in \{2.5, 10, 40, 100\}\}$, corresponding to an OTU1, OTU2, OTU3 and OTU4, respectively. According to the ITU-T recommendation G.709, a single higher bit rate signal can be composed by a mix of different lower bit rate signals, then various lower bit rate signals, with different bit rates can be groomed into a single higher bit rate signal [14, 2].

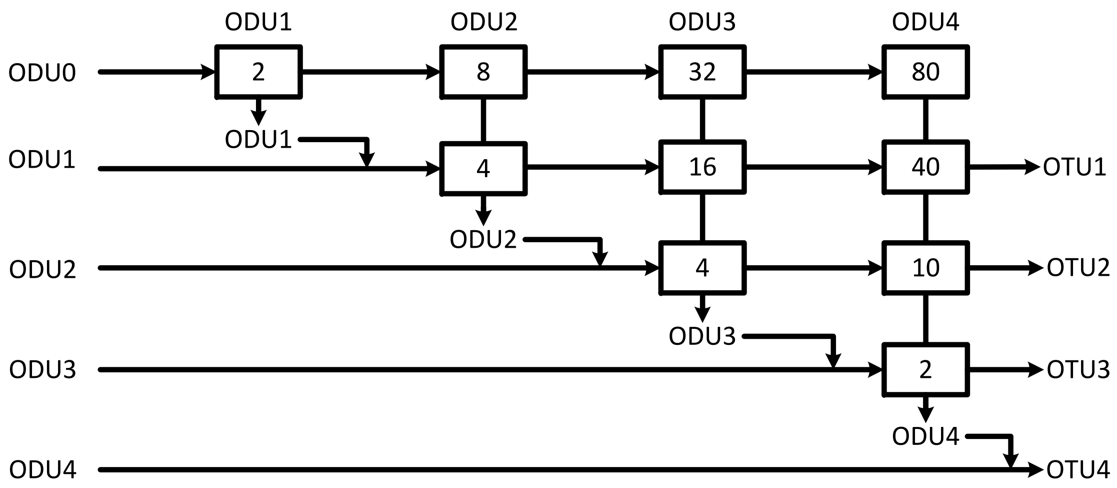


Figure 2.3: Grooming of OTN Signals [2]

As depicted in figure 2.3, the grooming of eight ODU0 client signals result in an OTU2 line signal. However, two ODU0 client signals mixed with three ODU1 is another possibility to form an OTU2 signal. The same reasoning is extensible to 100 Gbps line signals (OTU4) which are the line signals used in optimization models proposed in the next chapters.

2.1.2.2 Switching

In transport networks, currently, encapsulation, electrical switching and electrical grooming are usually realized using OTN. Regarding the original protocol, client traffic is encapsulated into ODU (optical data units) to be groomed and switched. The OTN technology is defined as circuit switching and depending on the layer where the switching occurs, two main types of switching were defined: electrical and optical.

2.1.2.2.1 Electrical Switching The use of circuit switched networks can introduce a waste of bandwidth due to the constant bit rate imposed by the OTN technology when the network becomes more and more packet dominated. Then, the Internet Engineering Task Force (IETF) defined the multi-protocol label switching - transport profile (MPLS-TP) as a customization of MPLS, in order to allow the possibility of packet switching transport networks. The MPLS-TP technology distinguishes the units of data through labels which are manipulated to provide an efficient bandwidth utilization. Then, there are different configurations for an electrical switch, but in general electrical switches are capable of receiving the traffic from an input interface, processing the traffic and switching to the appropriate output interface. The transparent transport mode does not perform switching in electrical domain. Then, the figure 2.4 shows an electrical switching capable node [15].

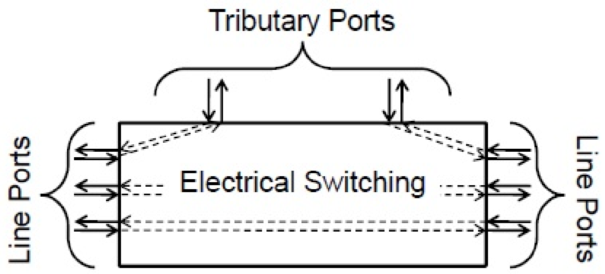


Figure 2.4: Scheme of a node for opaque transport mode network

Tributary ports are add/drop ports. Add ports are responsible for inserting traffic in the node from the access network and, inversely, the drop ports are used to extract traffic from the node. The line ports are used to make the connection between a node and the links of the core network through a higher bit rate signal (line signal).

2.1.2.2.2 Optical Switching In transparent and translucent networks, the optical layer has an important role in terms of switching. If a wavelength reaches a node in a transparent or translucent network, it can de switched to electrical domain to drop traffic information or can be correctly routed to the appropriate output port. In opaque networks there is no optical switching and the electrical layer is only used for wavelength multiplexing to perform WDM signals [16].

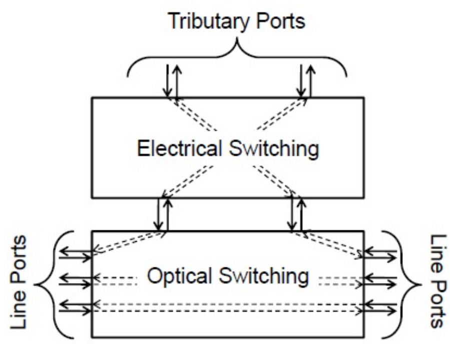


Figure 2.5: Scheme of a node using electrical an optical switching

2.1.3 Physical Topology

Physical topology can be seen as a layout of a real optical network, the placement of an optical network components, i.e., nodes disposition and connection conceded by links. Hence, an optical network simulation can be advantageous to analyse this approach. Thus, can be helpful to consider a simple six node core optical network connected by links, so physical topology design:

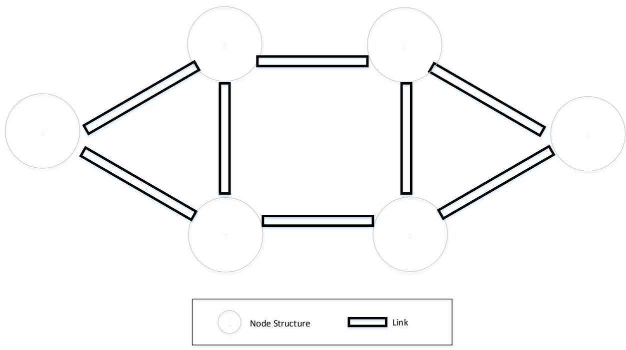


Figure 2.6: Nodes position and connection established by links located according to the real optical network

As depicted in Figure 4.4, the main elements are nodes and links. In this specific case study the number of nodes (N) and the number of links (L) are 6 and 8 respectively. The degree of a node (δ_i) is given by the number of direct connection it has to other nodes.

2.1.4 Logical Topology

Optical or logical topology for networks is an approach that defines how components are connected. In the case of optical networks, logical topology is associated with the optical path segments in a single lightpath. Thus, each node may be optical directly connected to each other, or only optical connected to adjacent nodes or optical connected to suitable nodes. Therefore, these shorter optical paths along the route imposed by logical topology lead to a situation of three transport modes: Opaque, Transparent and Translucent. During this dissertation the focus will be on opaque and transparent models.

2.2 Transport modes

Although the logical topologies of optical networks share the same physical links, the traffic is carried differently. Depending on the number of conversions from an optical signal to the electric domain different transport modes will appear.

2.2.1 Opaque

Opaque transport mode performs OEO(optical-electrical-optical) conversions at every intermediate node since origin to destination node. Thus, signal can be regenerated to provide signal quality conceded by electronic signal processing. In terms of topology, logical and physical topologies are the same, so each traffic route in logical topology corresponds to the link-by-link path imposed by optical fibers between each intermediate nodes until destination [17].

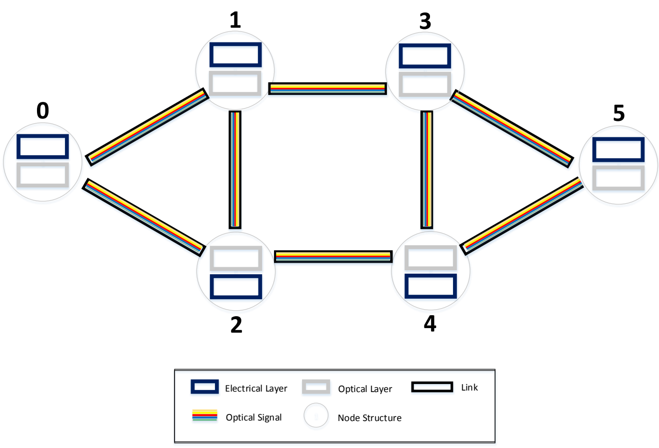


Figure 2.7: Opaque transport mode with a link-by-link grooming scheme

As a deduction, at all consecutive nodes until egress node, line signal was subjected to OEO conversions. An advantage of this transport mode is that it eliminates accumulation of physical impairments and allows full flexibility in terms of switching and grooming. Thus, it can improve capacity utilization of optical channels by providing traffic grooming at every node.

2.2.2 Transparent

In transparent transport mode, the information travels in a defined route through optical channels between origin and destination nodes(i.e. lightpaths) always in optical domain, and consequently physical topology and logical topology are different 2.8. Generally this procedure includes the conservation of the same wavelength on all the links which compose the route, as known as, wavelength continuity constraint. Grooming at intermediate nodes is not possible due to all-optical path applied end-to-end. Grooming scheme is single-hop because only client signals with the same destination and origin can be groomed in the same wavelength. Thus, figure 2.8 elucidate what is the optical topology for transparent transport mode from physical topology considered above [18].

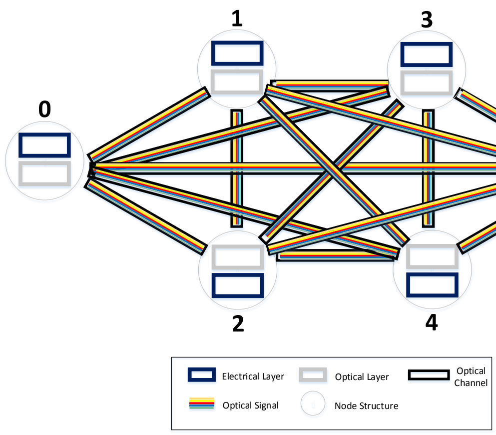


Figure 2.8: Logical topology for previous physical topology with each node connected to each other (Transparent Transport Mode)

Multicolor lines,obviously, assume the role of the optical signal. An advantage of this transport mode is the possibility for transportation of express traffic. Nevertheless, the quality of the optical signals degrade as they traverse the optical components along the route limiting the maximum optical reach of the signal. Another disadvantage is that the capacity utilization of optical channels is worse than in the opaque transport mode due to the grooming only in client signals with the same end-points.

2.2.3 Translucent

Translucent transport mode is a combination of previous transport modes. Thus, for the same physical topology there are several logical topologies. Within a single traffic route, the optical signal can cross some nodes regardless OEO conversions. However, to recover signal from all optical impairments and to optimize traffic grooming OEO conversions are needed. So, for a defined route there is some all-optical segments separated by regeneration, switching and grooming capable nodes. Then, using the same physical topology design as used in previous transport modes, figure 2.9 shows two possible examples of logical topologies.

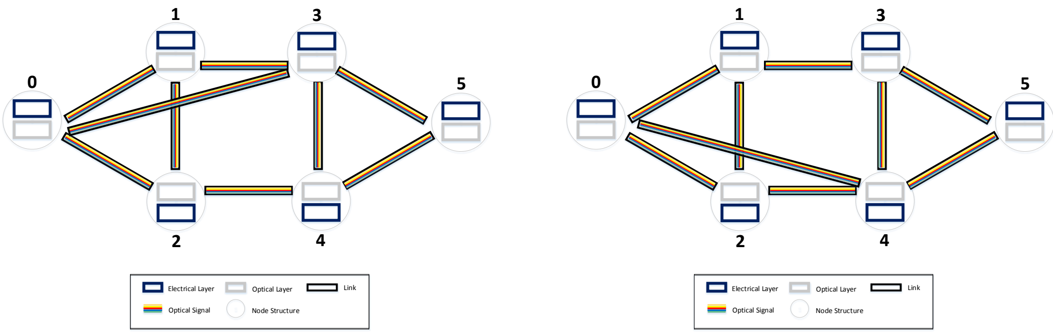


Figure 2.9: Translucent logical topologies

In translucent transport mode a multi-hop grooming scheme is employed, so different source and destination nodes can share the same lightpath until reach some intermediate node common to both. Thus, in all network nodes, lightpaths carrying local or client traffic that need to be switched a different to a different lightpath are sent to electrical domain, while through traffic lightpaths are kept in optical domain.

Transport mode	**Opaque**	**Transparent**	**Translucent**
Grooming			
Link-by-link	✓		
Single-Hop		✓	
Multi-Hop			✓
Intermediate nodes			
Electrical Switching	✓		✓
Optical switching		✓	✓
Grooming	✓		✓
Wavelength assignment	✓		✓
WDM multiplexing	✓	✓	✓
End nodes			
Encapsulation	✓	✓	✓
Electrical switching	✓	✓	✓
Grooming	✓	✓	✓
Wavelength assignment	✓	✓	✓
WDM multiplexing	✓	✓	✓

Table 2.1: Transport mode comparation

2.3 Network Survivability

Dedicated or shared protection are two main types of survivability applied for protection at any layer. Shared protection requires less spare capacity, employing protection reserved capacity for multiple working paths. A network planned for shared protection are feasible if the working paths that share the protection capacity have no links or intermediate nodes in common. This process is typically more vulnerable to multiple failures and slower to recover from a failure .

In dedicated protection backup resources are needed even before the failure occurrence. Depending on backup paths state, there will be two categories of dedicated path protection. In 1+1 mode, the backup path available is active, so both independent paths are transmitting at the same time and the choice of which is the best is the responsibility of the destination node. In opposition, the 1:1 dedicated protection activates the backup path, just in case of a failure on the primary path. After network restore, routes involved on previous failure may return to primary path (revertive mode) or proceed in backup path (non-revertive mode). In the following chapters will be proposed ways to implement dedicated path protection [19, 20].

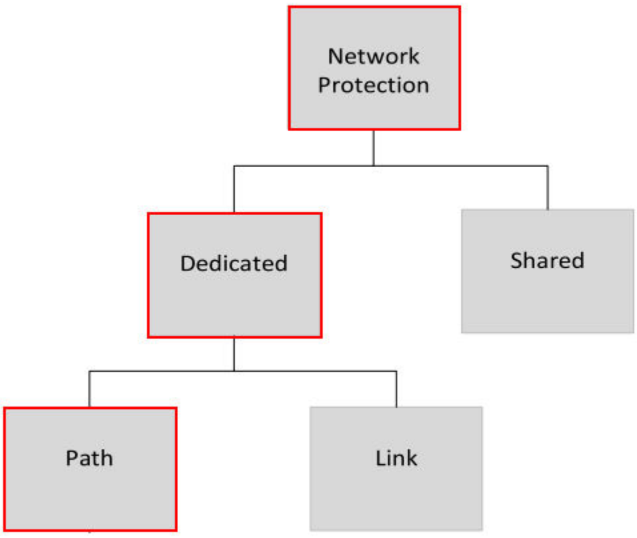


Figure 2.10: Network protection scheme

2.4 Chapter summary

In terms of optical networking design, there are two basic concepts, physical and logical topologies. The physical topology is the physical component of the network and how they are arranged, while the logical topology is how they work in terms of connection between the elements. Different types of logical topology in the same physical topology may have advantages or disadvantages desired by the network designer.

Logical topologies are related to how the optical signal is propagated. The signal can cross the nodes without being converted to the electrical domain (transparent), thus saving on the electrical components, and still be converted to the electrical domain and aggregate with other signals that cross the same next link, optimizing the bandwidth of the optical channels and saving on the optical components. It can even be a mixture of the other two topologies combining the advantages of both. In network planning it is extremely important to consider protecting traffic from failures. The protection can be shared, thus saving on the resources assigned for protection, or can be dedicated by assigning the resources before the failure occurs, preventing any failure in nodes or links of the working path. Then, based on the characteristics necessary for the network, different logical topologies may be more useful.

Chapter 3

Integer Linear Programming Models

Study of problems in which it is intended to minimize or maximize a function subject to a set of constraints is called optimization. Optimization models have a specific number of variables and constraints dependent on the problem to be solved. In telecommunications, ILP models are used to design networks describing real components and their capacities through a set of linear equations and inequalities. Despite the quality of the solutions obtained through the ILP models, depending on the number of variables and computational resources, the results of the ILP models can take days, months or even years [21]. The focus of the current chapter is to propose and describe two optimization models, based on opaque and transparent transport mode with protection. In section 3.1, is described a reference network protection topological design problem, including the definition of variables and inputs of the model and problem description. In section 3.2.1, it is proposed in detail the constraints of the opaque model with dedicated path protection as well as the objective function of the dimensioning problem. In accordance, the next section 3.2.2 is where the model of opaque transport mode will take place. In section 3.3, it is a brief approach to how an ILP model can be implemented through the LPsolve package. In section 3.4, will be the results of the models considering homogeneous and heterogeneous traffic while varying the number of demands. Then, the last section 3.5 contains a succinct summary of the chapter with the most important concepts.

3.1 Network protection topological design problem

The development of an ILP model requires first the definition of the inputs, outputs and variables of the problem in a mathematical representation. Then, let's consider a network consisting of a set of nodes, $V = \{1, \ldots, n\}$, and a set of bidirectional links connecting nodes, $L = \{\{i,j\} : i, j \in V, i < j\}$. Network node connections are in the form of an adjacency matrix called $G_{i,j}$. The client traffic demands from each origin node to the destination are also in matrix form: $D_{od} = \{[o, d] : o, d \in V, o < d\}$. For each type of client traffic, it must be created a bidirectional demand matrix D_{od}. The result of the sum of this matrices is a 3-dimensional matrix D_{cod},depending on client traffic type in Gb/s, $c = \{1.25, 2.5, 10, 40, 100\}$. Thereby, it will be considered the reference network below as an example of a smaller network where ILP approach will be tested.

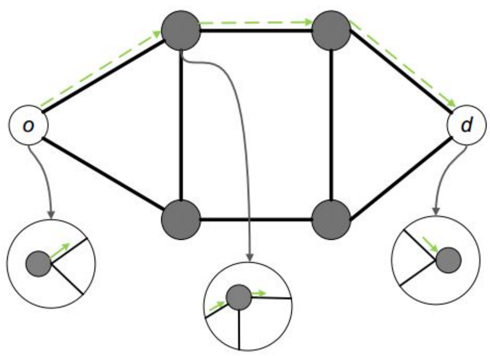


Figure 3.1: Traffic demand path from node "o" to node "d"

The traffic demands go through one or more links to reach their destination and it is assumed that the path between node o and d is the same as the path between node d and o. Therefore, considering dedicated path protection, the integer variable f_{ij}^{od} indicates how many flows starting at o and ending in d cross the link (i, j). Thus, to be consistent, f_{ji}^{od} indicates if the link (i, j) is used in the reverse direction between nodes (o, d). A demand of traffic requires a reserved capacity in each transmission system depending on the type of client traffic (c).For instance, if there are four traffic demands at 40 Gb/s, so a bandwidth of 160 Gb/s will have to be reserved. Otherwise, in this dissertation will be considered line ports capable of transmitting at 100Gb/s. Each optical signal is composed by several optical channels and each optical channel has a corresponding wavelength. Then, variable W_{ij} is an integer variable indicating the number of 100 Gb/s optical channels between nodes i and j. Consequently, almost the same model may be applied for transparent transport mode, but this time using optical channels (W_{od}) between end nodes instead of using optical channels link-by-link along the path. It is important to highlight that to reduce the computing time will be used bidirectional traffic matrices, i.e, $D_{cod} = D_{cod}^T$.

	Opaque	**Transparent**
Inputs	$G_{i,j}$ D_{cod}	G_{ij} d
Variables	f_{ij}^{od} f_{ji}^{od} W_{ij}	t_{ij}^{od} W_{od}

Table 3.1: Inputs and variables of opaque and transparent transport modes

3.2 ILP models

Integer linear program approaches proposed below, follow the standard form. All decision variable values that satisfy the set of constraints are called admissible solutions, in the cases that the solutions set is finite and the decision variables are restricted to integer values, it is called Integer Linear Programming. The optimal solution to the problem will be the values of an admissible solution(s) that obtain the higher or lower objective function value. The main goal is then to determine the non-negative values of the decision variables, such that all the linear equations or inequalities are satisfied and the value of the objective function is maximized or minimized. The objective function is introduced by the keyword *"minimize"* or *"maximize"*, and the set of the constraints are introduced by the expression *"subject to"*. Therefore, in order to solve the optimization models, it is required a software as MATLAB to effect a solution. There are also optimization software packages such as IBM ILOG CPLEX Optimization Studio, GLPK or Gurobi Optimizer to simplify the optimization problem description. In this dissertation, LPsolve was the package used to solve optimization problems. Then, ILP models proposed along this dissertation will be described using this structure [22].

3.2.1 Opaque with 1+1 Protection

The objective function of following ILP is a minimization of the sum of two variables: total number of flows crossing link (i, j) for all demand pairs (o, d) and total number of optical channels in each link (i, j).

$$minimize \quad \sum_{(i,j)} \sum_{(o,d)} f_{ij}^{od} + \sum_{(i,j)} W_{ij} \tag{3.1}$$

$$subject\ to$$

$$\sum_{j \backslash \{o\}} f_{ij}^{od} = 2 \qquad \forall (o,d) : o < d \ , \forall i : i = o \tag{3.2}$$

$$\sum_{j \backslash \{o\}} f_{ij}^{od} = \sum_{j \backslash \{d\}} f_{ji}^{od} \qquad \forall (o,d) : o < d \ , \forall i : i \neq o, d \tag{3.3}$$

$$\sum_{j \backslash \{d\}} f_{ji}^{od} = 2 \qquad \forall (o,d) : o < d \ , \forall i : i = d \tag{3.4}$$

$$\sum_{(o,d):o<d} (f_{ij}^{od} + f_{ji}^{od}) + \sum_{c \in C} (B(c) D_{cod} \leq 100 W_{ij} G_{ij} \qquad \forall (i,j) : i < j \tag{3.5}$$

$$W_{ij} \leq 80 \qquad \forall (i,j) : i < j \tag{3.6}$$

$$f_{ij}^{od}, f_{ji}^{od} \in \{0, 2\} \qquad \forall (i,j) : i < j, \forall (o,d) : o < d \tag{3.7}$$

$$W_{ij} \in \mathbb{N} \qquad \forall (i,j) : i < j \tag{3.8}$$

The objective function, to be minimized, is the expression(3.1). The flow conservation constraints are (3.2), (3.3) and (3.4). First constraint ensures that, for all demand pairs (o,d), it routes two flows of traffic for all bidirectional links (i,j) when "j" is not equal to the origin of the demand. Equation (3.4) is based on the same idea of (3.1), however applied in reverse direction. Assuming bidirectional traffic, so the number of flows in both directions of the link is the same (3.3). The inequality (3.5) is considered grooming constraint, so it means the total client traffic flows can not be greater than the capacity of optical channels on all links. Another important constraint (3.6) is the capacity of the optical channels which must be less or equal to 100 Gb/s or 80 ODU0. The number of flows per demand can be zero if there are no traffic demands or two if considering working and protection traffic (3.7). The last constraint is just needed to ensure the number optical of channels is a positive integer values greater than zero.

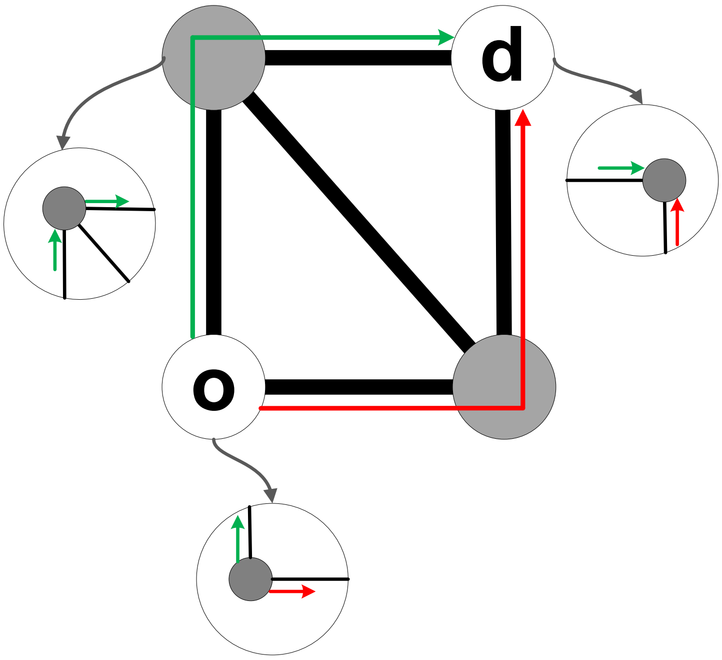


Figure 3.2: Example of two nodes in a network sending traffic through disjoint links

Constraints (3.2) and (3.4), was proposed to ensure that the flow used for protection should be different from working path. Figure 3.2 shows a signal being sent by disjoint paths as supposed in dedicated path protection. From client point of view, there is just a single flow from the origin to the destination, but as shown in the detailed view of the figure above, it's possible to verify that there is a flow arriving and another starting at each intermediate node.

3.2.2 Transparent with 1+1 Protection

The optimization model suggested for transparent transport mode with dedicated path protection intends to minimize the total number of flows crossing link (i, j) for all demand pairs (o, d). The mathematical model described below also minimizes the total number of optical channels between each demand end nodes W_{od}, instead of minimizing the number of optical link-by-link channels as in the previous model.

$$minimize \quad \sum_{(i,j)} \sum_{(o,d)} f_{ij}^{od} + \sum_{(o,d)} W_{od} \tag{3.9}$$

$$subject\ to$$

$$\sum_{j \backslash \{o\}} f_{ij}^{od} = 2 \qquad \forall (o,d) : o < d, \forall i : i = o \tag{3.10}$$

$$\sum_{j \backslash \{o\}} f_{ij}^{od} = \sum_{j \backslash \{d\}} f_{ji}^{od} \qquad \forall (o,d) : o < d\ , \forall i : i \neq o, d \tag{3.11}$$

$$\sum_{j \backslash \{d\}} f_{ji}^{od} = 2 \qquad \forall (o,d) : o < d, \forall i : i = d \tag{3.12}$$

$$\sum_{(o,d):o<d} (f_{ij}^{od} + f_{ji}^{od}) x W_{od} \leq 80 G_{ij} \qquad \forall (i,j) : i < j \tag{3.13}$$

$$f_{ij}^{od}, f_{ji}^{od} \in \{0, 2\} \qquad \forall (i,j) : i < j, \forall (o,d) : o < d \tag{3.14}$$

$$W_{od} \in \mathbb{N} \qquad \forall (o,d) : o < d \tag{3.15}$$

The objective function, to be minimized, is the expression(3.9). The flow conservation is performed by equations (3.10), (3.11) and (3.12) and share the same mathematical description of opaque model. The inequality (3.13) answers capacity constraint problem. Then, total flows times the traffic of the demands must be less or equal to the capacity of network links. The grooming of this model can be done before routing since the traffic is aggregated just for demands between the same nodes, thus not depending on the routes. Last two constraints define the total number of flows must be zero if there is no demand, or two for a demand with traffic protection, and the number of optical channels must be a counting number.

3.3 LPSolve and Matlab implementation

Implementation of the models described in section 3.2 requires the use of mathematical software tools. Software chosen was MATLAB, since it is ideally suited to handle linear programming problems. The use of this powerful software allows declaration and definition of variables and constraints in an iterative and simplified procedure. Variables are written and stored into vector or matrices sorted by indices known by the developer and dependent on the network size. This computing resource can call LPsolve through an external interface or MEX-function. LPsolve is used to resolve linear programming problems when declared according to a specific structure, in this case, achieved through MATLAB. LPsolve is an optimization software package that reuses other free software components to make an integrated development environment (IDE). The use of this IDE allows solving ILPs in a user-friendly graphical interface, thus making it easier to debug problems. The control of LPsolve is provided by mxlpsolve driver which is allowed to operate with data sent through an application programming interface (API).

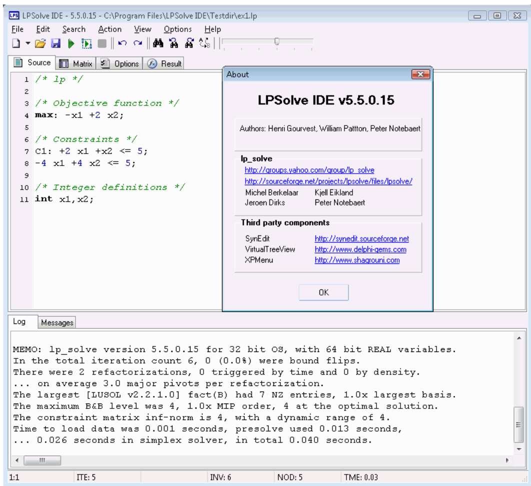


Figure 3.3: LPSolve integrated development interface

Optimization models presented in section 3.2.1 and 3.2.2 should be written on programming format imposed by mxlpsolve driver, then the following steps have to be followed [23]:

1. Create and initialize the new model with total number of variables.
2. Set the coefficients of the objective function.
3. Add the constraint vector with the coefficients and independent terms.
4. Set variable types.
5. Write ILP to an .lp file
6. Solve the model
7. Returns the value of the objective function
8. Returns the value of the variables

The first step is to declare the model and send the total number of problem variables to the solver. Thereafter, it is needed to send the objective function coefficients using the appropriate API function. Then, the driver is called iteratively sending as arguments the name of the function to add a constraint, the correct indexes of the variables vector, the corresponding equality or inequality and the independent terms. It makes it possible to define all constraints of the problem. Regarding problem description, the last step is to set variables type according to the mathematical model. Hence, in order to simplify the visualization, it is advisable to write the model to a file that can be opened in LPsolve IDE. Finally, it is necessary to instruct the solver to resolve the problem and to return the results.

3.4 ILP results

The presentation and analysis of the results of the model 3.2.1 and 3.2.2 are consummated during this section. The models will be tested with homogeneous and heterogeneous traffic, changing the number of demands or fluctuating the client traffic type (lower to higher bandwith). The results for opaque and transparent models are shown using, respectively, blue and green bars. All results were obtained considering the reference network. The model was not applied to a real network because the execution time would be protracted.

Then, starting with homogeneous client traffic from the lowest to the highest bandwidth, the following results were obtained:

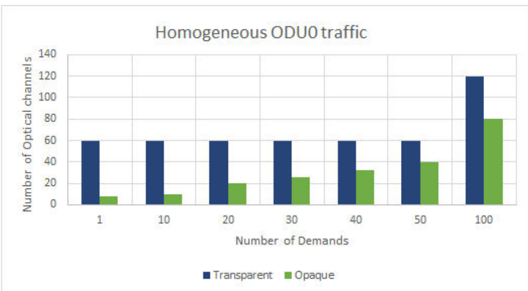


Figure 3.4: ILP solution for ODU0 client traffic

Considering a low number of demands the result of transparent model is exaggeratedly high. When there is only one traffic demand, sixty optical channels are required, since the network has six nodes and there is a demand from each node to the other five and the same number of demands for protection.

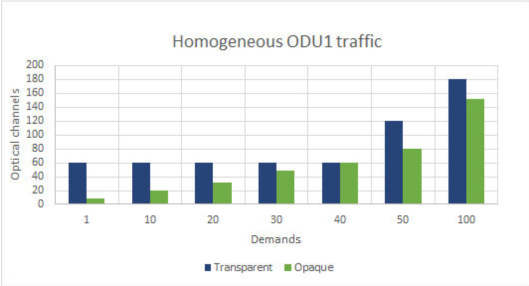


Figure 3.5: ILP solution for ODU1 client traffic

The analysis of results for ODU0 (1.25 Gb/s) client traffic is almost the same as for ODU1 traffic (2.5 Gb/s). A result to be highlighted is the number of optical channels for 40 demands of traffic. If there are 40 demands at 2.5Gb/s, then it performs 100Gb/s which is the capacity of each optical channel. So, traffic aggregation is impossible because all optical channels are completely occupied.

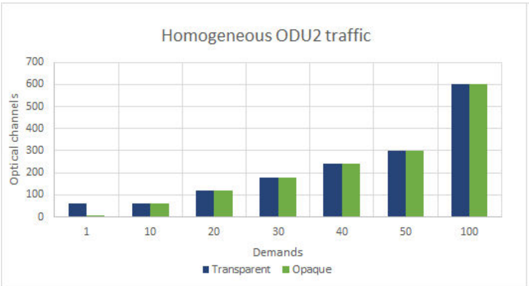


Figure 3.6: ILP solution for ODU2 client traffic

Both ILP models has approximately the same performance. As expected, the results sending one demand of traffic ODU2 (10Gb/s) are much better using opaque transport mode. When the number of demands is a multiple of 10, traffic between each node pair is a multiple of 100 (optical channel capacity), so link-by-link grooming has the same result as grooming only in end nodes.

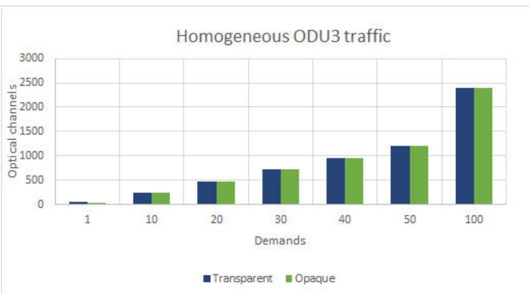


Figure 3.7: ILP solution for ODU3 client traffic

Figure 3.7 shows the behaviour of the algorithm for ODU3 (40Gb/s) traffic. The reasons why opaque and transparent models have practically the same results are mentioned above. It is important to note that 100 demands of ODU3 traffic are approximately 4Tb/s of traffic. When the number of optical channels required in a network is the same for both models, transparent is cheaper than opaque since some electrical components are not needed at intermediate nodes and the propagation of optical signals are faster. Wilfully, results for demands at 100Gb/s (ODU4) are not necessary since the demands has the same bandwidth as optical channels capacity, then there is no traffic aggregation.

After analysing and detailing the results for each client traffic and changing the number of demands, it is necessary to see the results from another point of view.

Figure 3.8: ILP solution changing client traffic type

Then, setting number of demands constant and changing the type of traffic, the number of optical channels has a big leap for client traffic based on demands of ODU3 and ODU4 traffic. Transparent model has 8 times more optical channels for signals with the lowest bandwidth.

3.5 Chapter Summary

In sum, this chapter was dedicated to the introduction of linear programming concept. In order to understand the meaning of mathematical models, all models used were described in the first section. The description of proposed integer linear problems in mathematical form as well as the use of optimization packages and their specific syntax are essential for the developer. Finally, the results prove the efficiency of opaque model for lower bandwidth traffic and a small difference when client traffic has higher bandwidth. In cases where the number of optical channels is lower, it will be beneficial to use an opaque network, however when the number of optical channels is similar to the transparent network, transparent network is cheaper and the signals take less time to reach their destinations.

Chapter 4

Heuristics

Heuristic approaches are techniques for solving a problem, faster than optimal methods and able to find an acceptable solution. Heuristics can be applied in several problems and are widely used in computer science and telecommunication networks [24, 25]. There are some commonly used heuristics such as genetic algorithm or simulated annealing. When networks are too large, the ILP models can be very slow to obtain the solution. In this chapter, algorithms are proposed based on heuristics and their implementation in a networks design software. The software was chosen for being open source and, consequently, easy to employ in academic projects. The grooming and routing algorithms presented in this chapter involve a graphical user interface developed for Net2Plan tool. Thus, section 4.1 is an overview of the routing algorithm, starting by trying to find the protection and work paths used by grooming algorithm. Aggregation algorithm is proposed in section 4.2. Routing and grooming algorithms are the core of the code behind the graphical user interface and focus in all the steps required for an offline simulation. In section 4.3 there is a description of how the program works, the developers involved in the program and a brief approach to the interface from creating traffic matrices and network topology (inputs) until obtaining the number of necessary ports of the designed network. Concerning publication of the results, node features and most expensive components are depicted on the last subsection. In section 4.4, the algorithms proposed in previous sections are tested and compared with those of the previous chapter. Finally, the last section is a brief summary of the principal terms to retain.

4.1 Heuristic routing

Regarding the heuristic approach, an algorithm for routing and grooming was implemented, which will be explained in three steps. The pseudo code will be shown in a flowchart using rounded rectangle shapes to symbolize the beginning and the end of the program, the parallelogram shape indicates a point where there is input to or output from the program, the diamond shape symbolizing a decision point, the rectangle shape indicating the assignment of a value to a variable, constant or parameter, and the hexagon shows the beginning of a repetition.

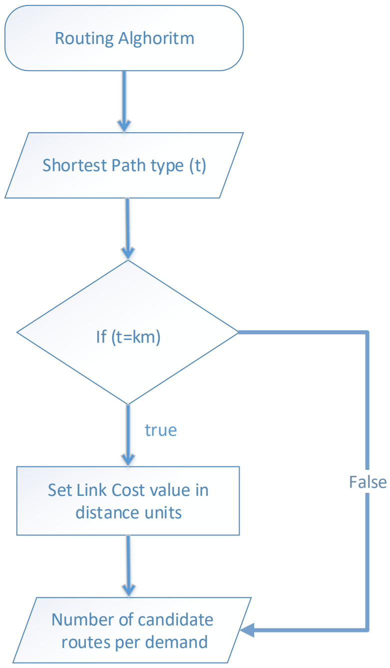


Figure 4.1: Heuristic routing algorithm code flow

Routing algorithm begins with the definition of the input parameter. The variable "t" is an input variable, which is defined by the user. The previous variable can assume two possible values: "hops" or "km". If the units of distance are the number kilometres, then it is necessary to assign a virtual cost to the links that will be used, in order to decrease the computing time of the algorithm used to find the shortest path. Through the Dijkstra algorithm the paths sorted by distance units are added to a list and ordered by node pair. The number of candidate routes per node pair depends on the user, and can be defined through graphical interface setting the corresponding variable. The performance of the algorithm can be compromised if the number of routes or the network topology is too large.

Once the list of candidate routes has been filled, it is necessary to find the primary and protection paths for each demand. So, for each demand will run the code below.

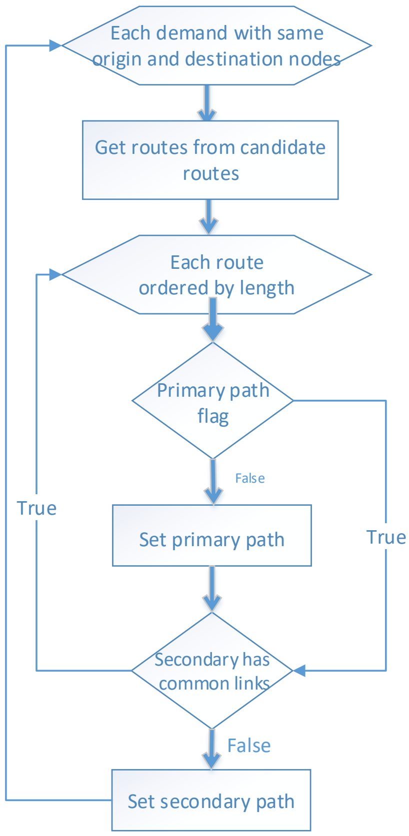


Figure 4.2: Heuristic routing algorithm code flow

The second step of routing algorithm begins with the search of candidate routes for the demand. The routes are analysed one by one, starting from the smallest to the largest. Then, the first (shortest) is the primary path and this must be set if the flag is disabled. If the primary path flag is enabled, the program flow jumps to the decision point to verify if the route does not have links in common with the first path. Then, if it is verified, the protection path is the current route. On the other hand, if the route has at least a common link, the next step is to try another route. The routing ends when the protection path of the last demand was found.

4.2 Heuristic grooming

After traffic routing decision, the consequent step is traffic aggregation. The algorithm begins by getting the routes (primary and protection path) already saved for each node pair. The input parameter of this algorithm is the type of grooming desired, which is represented in the flowchart as "g".

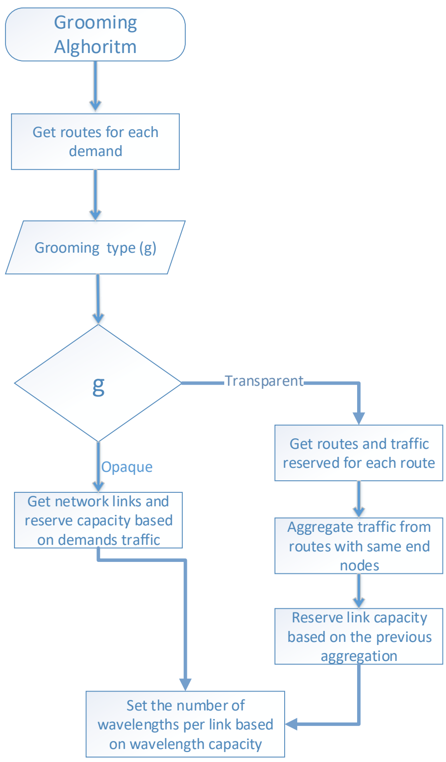


Figure 4.3: Heuristic grooming algorithm code flow

The consequence of the proposed type of grooming is the branch of flowchart. In Opaque form, the procedure starts by obtaining all links and the capacity based on traffic per demand. The sum of traffic that crosses the same link gives the information of carried traffic. After traffic aggregation, dividing the total carried traffic by the wavelength capacity it is possible to get the number of wavelengths (line ports) in each node. Although, if logical topology is Transparent the plan of action is different. The branch starts by getting reserved traffic capacity in each path. Then, using routes from different demands with common end nodes it aggregates the traffic of these routes. After this step, knowing the links that constitute the path, it is reserved the capacity required in each link. In order to know the number of wavelengths, the reserved traffic capacity in each link needs to be divided by wavelength capacity as in the previous grooming type. Obviously, these flowcharts are simplified schemes which express as much as possible the complexity of hundreds of lines of code.

4.3 Net2Plan Aproach

The Net2Plan tool began in 2011, with the aim of being used as a material for educational in Universidad Politcnica de Cartagena. At this time, there are already many simulators available for network planning as, for example, SIMTON, OPNET, NIST Merlin or TONetS [2]. Up to now, the lastest version of Net2Plan tool is version 0.4.0. All alghoritms were developed in JAVA language, and the specific alghoritms presented in this dissertation were developed using Eclipse for JAVA developers IDE. The first step to start the simulation is the creation of traffic matrices.

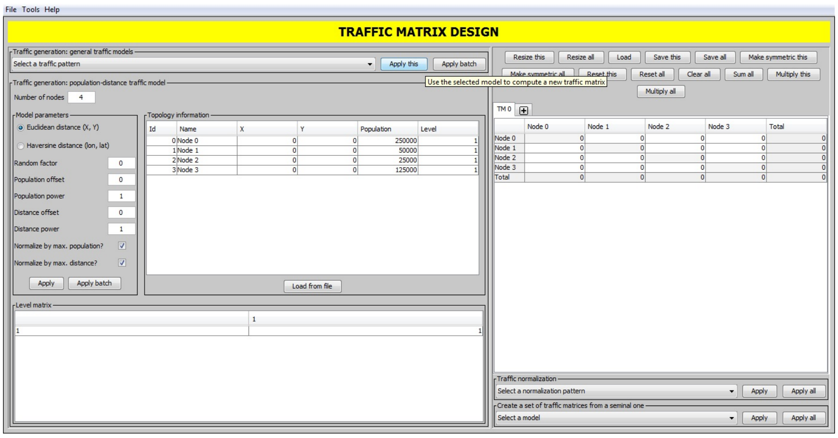


Figure 4.4: Net2Plan user interface for traffic matrices creation

The matrices created can be defined as a matrix with constant traffic, so the amount of traffic between all node pairs are the same. Another possibility is to choose randomly the traffic for each demand and there is also an option to choose automatically the number of nodes and the traffic amount, or define them manually as parameters. The results presented in this dissertation are obtained defining matrices manually to ensure that the ILPS and heuristics have the same inputs. After the matrices creation for each type of client bit rate (ODU0, ODU1...) is needed a network topological design.

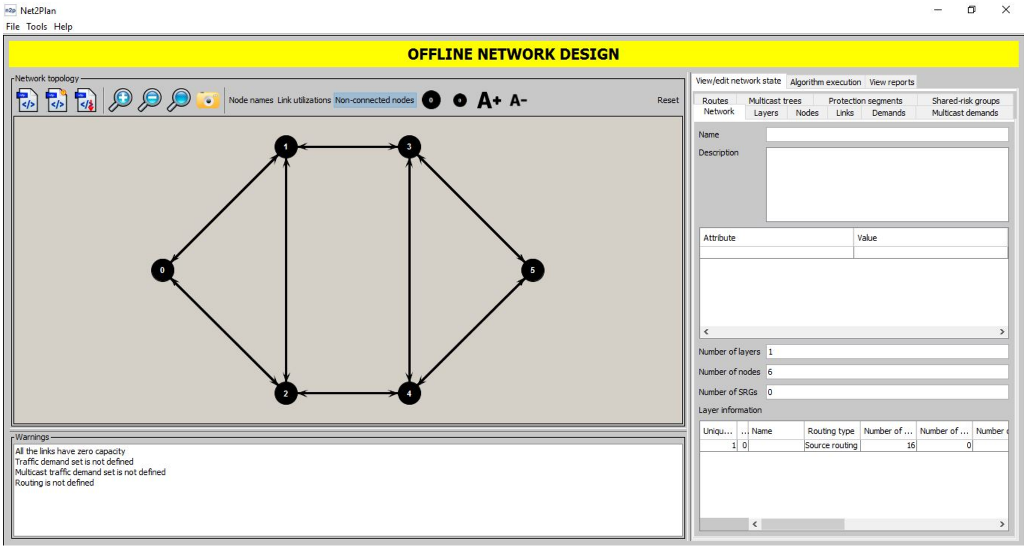


Figure 4.5: Net2Plan user interface for reference network topological design

In figure 5.2 it is shown how the real network suggested in chapter 3 is designed in Net2Plan. Each black narrow symbolizes a bidirectional link. Once created the design of the network and the traffic matrices defined all inputs are set. Thus, the network is ready for routing and grooming.

4.3.1 Join Traffic Matrices Alghoritm

Having the basic physical topology created, the next step is to load the demand set into the network. In the case where there are multiple traffic matrices an algorithm was developed to aggregate these, in order for it to be possible to load all demands. In order to deal with different client traffic in OTN units, it was developed the *"joinTrafficMatrices"* algorithm to convert all traffic matrices to ODU0. Besides converting the different ones to ODU0 it also creates an attribute in each demand indicating the type of signal before converting.

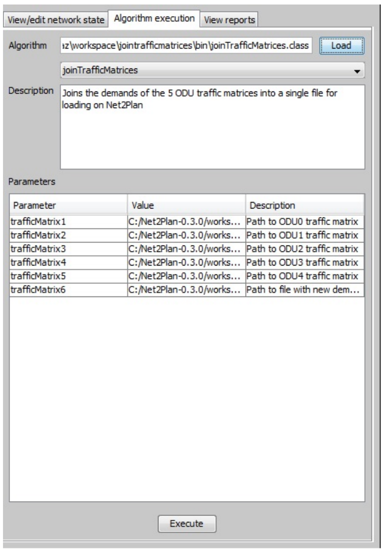


Figure 4.6: Net2Plan join traffic matrices alghoritm input parameters

As shown in figure 4.6 above, using five different paths for the client traffic matrices in ODU0, ODU1, ODU2, ODU3 and ODU4 as inputs and the last path to save the resultant demand set, is possible to create a matrix with total traffic regarding the original client traffic type. Thus, to set all network design inputs is just needed to load the demand traffic set to the network topology previously designed.

4.3.2 Logical Topology Alghoritm

The *"logicalTopology"* algorithm accepts only one input parameter and it creates a new layer depending on the transport mode chosen. This step is needed before network routing and grooming, because these two operations depend on the logical topology of the network. Besides creating this new Layer, the algorithm also copies the demands to that layer and defines the logical links based on the length of the physical ones.

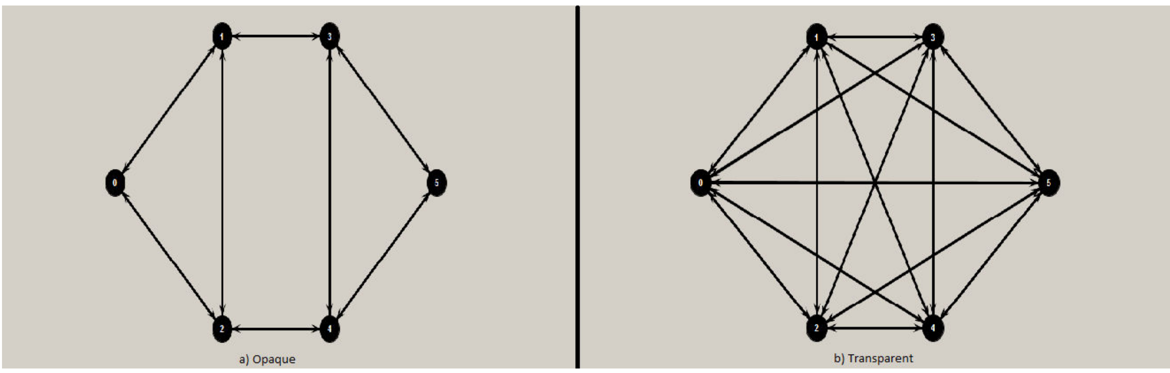


Figure 4.7: Net2Plan logical topology for reference network physical topology based on a) Opaque transport mode b)Transparent transport mode

As depicted in figure 5.3, an Opaque transport mode the traffic goes through an OEO conversion at every node and as such the logical topology is the same as the physical one. In the Transparent mode, there are no regeneration in intermediate nodes, so the logical topology shows that the traffic between nodes flows directly without grooming with signals from another source. Thus, in optical layer, for opaque transport mode the topology of the network are equal to the physic one and for transparent transport mode is a full mesh network topology.

4.3.3 Grooming and Routing Alghoritm

In this section, it is presented an algorithm to perform grooming an routing with dedicated path protection scheme. This algorithms are based on reducing the number of wavelengths needed for the network to carry all traffic. One way to minimize the number of wavelengths is minimizing the number of links for each path between all node pairs. The consequence of the reduction of the number of links is to decrease the number of necessary transmission systems leading to a lower cost of nodes.

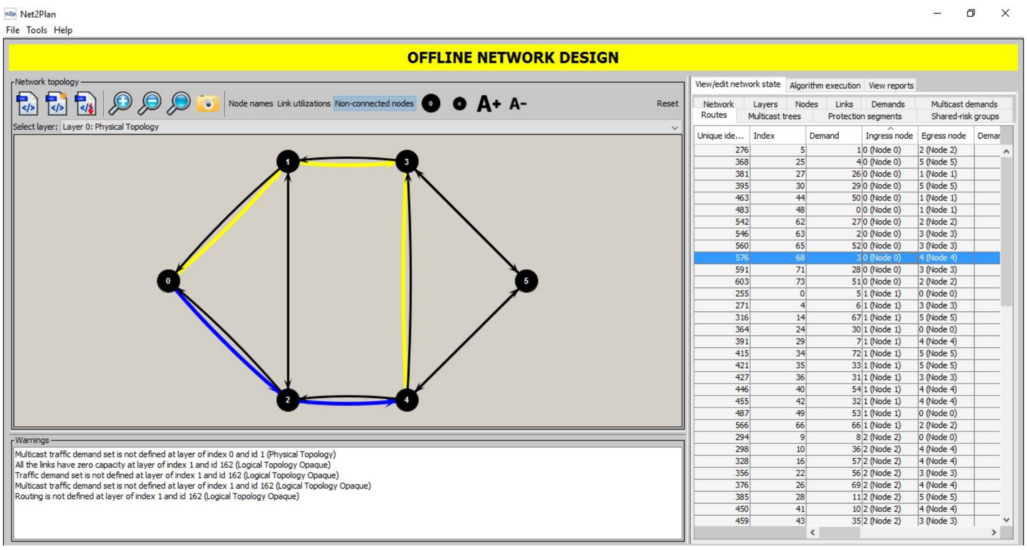


Figure 4.8: Net2Plan route using *grooming dedicated* alghoritm

Thus, this approach is done based on a shortest path algorithm where the routes for each demand are created based on the shortest number of hops needed to reach the destination node.The protection path for each demand, as shown in figure 4.8, is created using the same methodology, so the protection path is the second shortest loop disjoint path. The routes are made through the use of existing classes in Net2Plan, and the shortest path is computed using Dijkstra algorithm. The blue lines symbolize the working path and the protection segments are depicted in yellow. Grooming is different for the modes described above. Although, it is implemented in the same physical topology , i.e. the first layer in the program. The inputs, defined by network designer, of this algorithm are the number of routes computed per demand in order to find two disjoint shortest loopless paths and the wavelength capacity for each transmission system in ODU0.

4.3.4 Reports

The display of results was also a concern in the development of algorithms, a report was created in HTML language that can be written to a file or open through web browsers. The *networkCost* algorithm, which has as inputs the elements with higher cost. The most expensive elements are optical ports and equipments (OLT, OXC and OXC ports, Transponders and amplifiers) and electrical ports and equipments (EXCs, EXC ports).

The number of optical channels is already calculated for each link in subsection 4.3.3 and, consequently, the number of transmission systems. Thus, in order to make a characterization of the links is required the number of optical amplifiers. The result, for instance, of link information in Net2Plan report for the reference network is:

Node Pair	Wavelengths forward	Wavelengths backward	Amplifiers forward	Amplifiers backward
Node 0 -» Node 1	2	2	5	5
Node 1 -» Node 0	2	2	5	5
Node 0 -» Node 2	2	2	5	5
Node 2 -» Node 0	2	2	5	5
Node 1 -» Node 2	4	4	3	3
Node 2 -» Node 1	4	4	3	3
Node 1 -» Node 3	3	3	3	3
Node 3 -» Node 1	3	3	3	3
Node 2 -» Node 4	3	3	3	3
Node 4 -» Node 2	3	3	3	3
Node 3 -» Node 4	3	3	3	3
Node 4 -» Node 3	3	3	3	3
Node 3 -» Node 5	3	3	5	5
Node 5 -» Node 3	3	3	5	5
Node 4 -» Node 5	3	3	5	5
Node 5 -» Node 4	3	3	5	5

Figure 4.9: Reference network link information using random traffic matrices

More information can be obtained on the reporting algorithms that will be detailed in next chapter.

4.4 Heuristic results validation

Heuristics are approximate results so their results when not compared to a solution taken as correct may be misleading. Thus, the algorithms were tested using the same inputs as ILP models and the results are presented through line graphs and compared with previously obtained bar graphs. The blue line is relative to the results for transparent transport mode and the yellow line for the opaque transport mode.

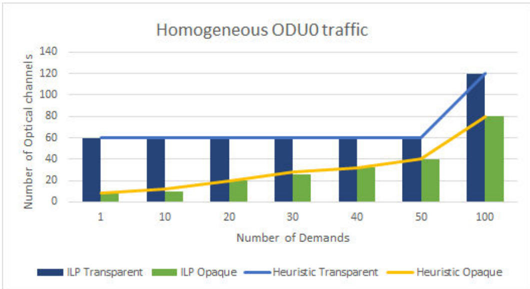


Figure 4.10: Heuristic solution for ODU0 client traffic

The figure above shows that even when grooming is done for small amounts of low-bandwidth traffic, the results are close. In the results of transparent transport mode, the line has exactly the same points as the graph which means that the traffic was aggregated through the same method. On the other hand, for opaque transport mode the results are slightly different, but as the number of demands increases the relative error suffers a huge decrease.

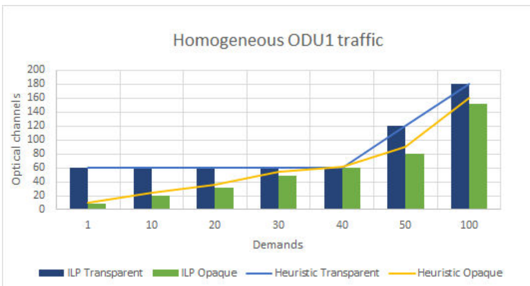


Figure 4.11: Heuristic solution for ODU1 client traffic

From the results for ODU1 traffic the same considerations are correct. Then, the comparative results are the same for transparent model and have small differences for opaque. The results for 40 demands are the same for all models and transport modes, because 40 demands requires 100Gb/s of bandwidth which is the total capacity of each optical channel, then there is no possible traffic aggregation.

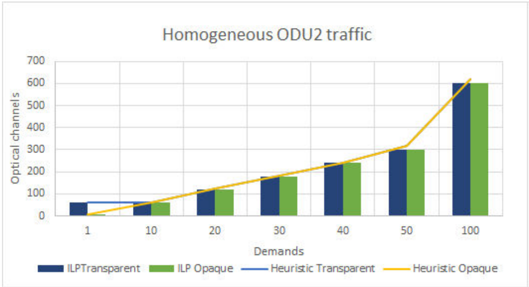


Figure 4.12: Heuristic solution for ODU2 client traffic

When demands at 10Gb/s are used the differences between ilps and heuristics are minimal and they are only visible when the number of demands is not a multiple of 10. When one demand is sent the opaque model is much more advantageous in terms of grooming for both approaches.

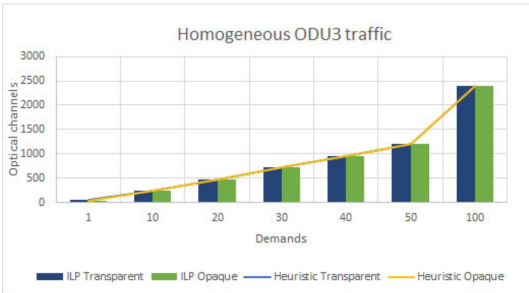


Figure 4.13: Heuristic solution for ODU3 client traffic

Traffic at 40Gb/s (ODU3) is high bandwidth traffic, therefore the aggregation in channels in 100Gb/s can not be done in different ways, so the results are approximately the same for both transport modes and methods. Sending 100 demands of traffic, it has 2500 optical channels which is a huge number and a consequence of the flow of 4Tb/s between each node pair.

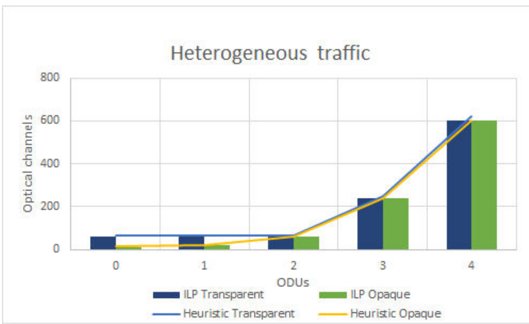


Figure 4.14: Heuristic comparative results for heterogeneous traffic

The last validation results were obtained applying heuristic methods in sex node reference network and changing client traffic type. Relative error between ILP solution and heuristic is less than 0.1. The number of optical channels for high bandwidth traffic is accordingly high.

4.5 Chapter Summary

In sum, the implementation of heuristics involves not only the creation of the algorithms but also their implementation in an open source software with graphical user interface called Net2Plan. This customized software has four algorithms developed in the context of this dissertation. The first is used for conversion of traffic matrices. The visualization of the network logical topology is performed by the second algorithm proposed. Routing and grooming algorithm is the core of the program. The code flow and graphical user interface of this algorithm can be seen in this chapter. The last customization made on net2plan concerns the presentation of results that is done in the reports section. Then, regarding the validation of the heuristic methods, in the last section is shown the comparative results between ILPs and Heuristics. The results of heuristics are in line with ILP results for homogeneous and heterogeneous traffic and using the six node reference network.

Chapter 5

Case Study

A network dimensioning software is used in the design of real networks. The cost of a network depends on the network size and capacity, but this huge investment from network stakeholders re-enforces the importance of network planning. There are networks created for research purposes and installed in the field. Then, to get results from a real network, it will be studied a network from United States known as NFSNET. The first section of this chapter is a network characterization in terms of physical network elements regarding their geographic location and the traffic matrix generated. In section 5.2 is presented an approach for network cost, considering the cost of links and the cost of nodes. This section has the equations used as the basis of network cost reports in Net2Plan. This equations perform network cost using as input the most expensive network components cost defined by the designer. In section 5.3, the results in terms of tributary and line ports and network cost of opaque transport mode are shown. The results of the transparent transport mode for the same matrix of traffic are described in the next section (5.4). The chapter ends, as in previous chapters, with a brief overview of all important information and results of the chapter.

5.1 NSFNET

After validating the quality of the algorithms implemented in Net2Plan, let's consider a realistic network. NSFNET refers to a program sponsored by National Science Foundation to support and foment advanced networking among United States research and education institutions. The way the nodes are geographically arranged can be seen from the figure 5.1.

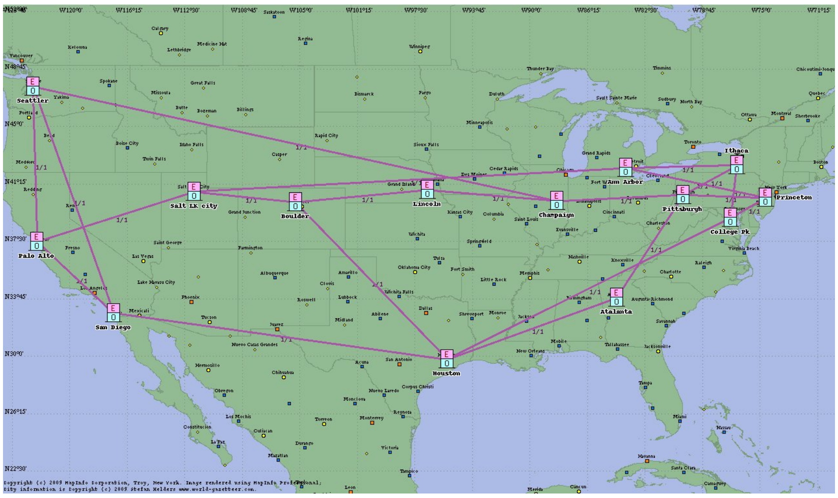


Figure 5.1: NSF Network topology [3].

The network represented above is composed of fourteen nodes and twenty one bidirectional links. The result of dividing the number of links per node by the number of nodes is the average degree of the nodes in this network. All nodes have less than four and more than two links, so major nodal degree is 4 and minor is 2. Then, there are no isolated nodes (node with only one connection) and it's important because the dedicated path protections requires disjoint paths, so each node must have at least two links. The maximum number of connections per node is also important because the larger it is, the fewer will be the number of hops suffered by each demand from the origin to the destination. Obviously, the number of links per node could be higher, but it would compromise the price to be paid for the network.

Constant	**Description**	**Value**
N	Number of,Nodes	14
E	Number of Bidirectional Links	21
$<\delta>$	Node out-degree (max,min, avg)	4,2,3.0
$<h>$	Mean Number of Hops,for Working Paths	2.14
$<h'>$	Mean Number of Hops,for Backup Paths	3.60
$<s>$	Mean Link Length (km)	1086

To begin the analysis of traffic routing and aggregation it is necessary to consider the matrix of demands (D):

$$D = \begin{bmatrix} 0 & 12 & 34 & 15 & 12 & 54 & 12 & 12 & 43 & 12 & 12 & 23 & 12 & 11 \\ 12 & 0 & 60 & 12 & 35 & 12 & 23 & 20 & 12 & 36 & 12 & 23 & 33 & 13 \\ 34 & 60 & 0 & 15 & 12 & 12 & 12 & 18 & 14 & 12 & 21 & 23 & 12 & 12 \\ 15 & 12 & 15 & 0 & 21 & 18 & 12 & 12 & 43 & 12 & 12 & 23 & 12 & 11 \\ 12 & 35 & 12 & 21 & 0 & 12 & 12 & 12 & 29 & 12 & 12 & 26 & 12 & 15 \\ 54 & 12 & 12 & 18 & 12 & 0 & 42 & 30 & 12 & 12 & 21 & 12 & 30 & 84 \\ 12 & 23 & 12 & 12 & 12 & 42 & 0 & 30 & 48 & 12 & 12 & 14 & 9 & 54 \\ 12 & 20 & 18 & 30 & 12 & 30 & 30 & 0 & 12 & 48 & 12 & 60 & 30 & 72 \\ 43 & 12 & 14 & 25 & 29 & 12 & 48 & 12 & 0 & 12 & 12 & 54 & 12 & 66 \\ 12 & 12 & 21 & 12 & 12 & 21 & 12 & 12 & 12 & 12 & 0 & 12 & 12 & 12 \\ 23 & 23 & 24 & 12 & 26 & 12 & 14 & 60 & 54 & 12 & 12 & 0 & 12 & 12 \\ 12 & 33 & 12 & 19 & 12 & 30 & 9 & 30 & 12 & 36 & 12 & 12 & 0 & 11 \\ 11 & 13 & 12 & 11 & 15 & 84 & 54 & 72 & 66 & 12 & 12 & 12 & 11 & 0 \end{bmatrix}$$

This traffic matrix was generated randomly through Net2Plan. The total amount of traffic in the NFSNET network considering this demands matrix is 5Tb/s. It should be noted that the total number of columns and rows is equal to the number of nodes and the main diagonal of the matrix is composed of zeros since it does not make sense for a node to send traffic to itself. Once traffic has been defined, the next step is to upload the shape of the network to the dimensioning software.

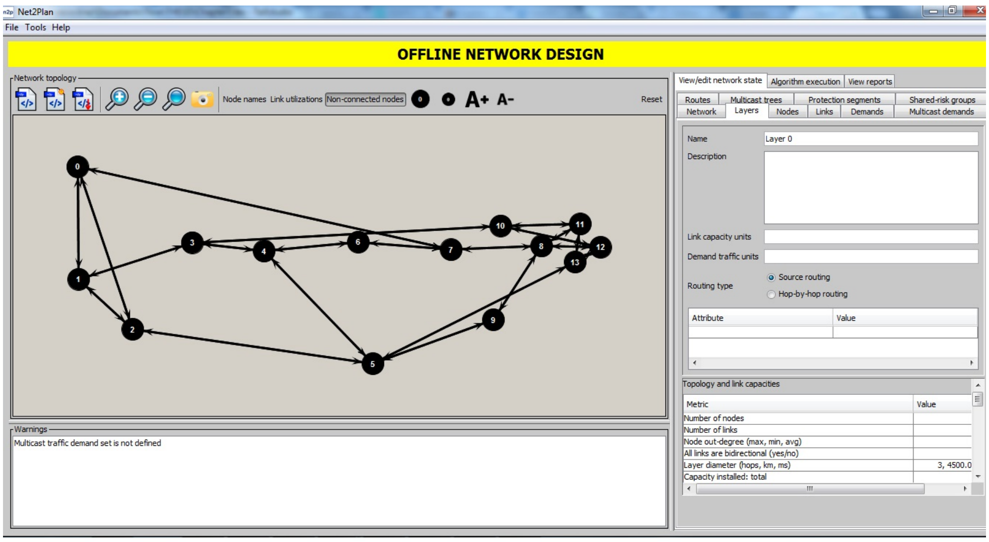


Figure 5.2: Net2Plan user interface for NSFNET topological network design

In the logical layer the network aspect for the Opaque and Transparent topology will be as follows:

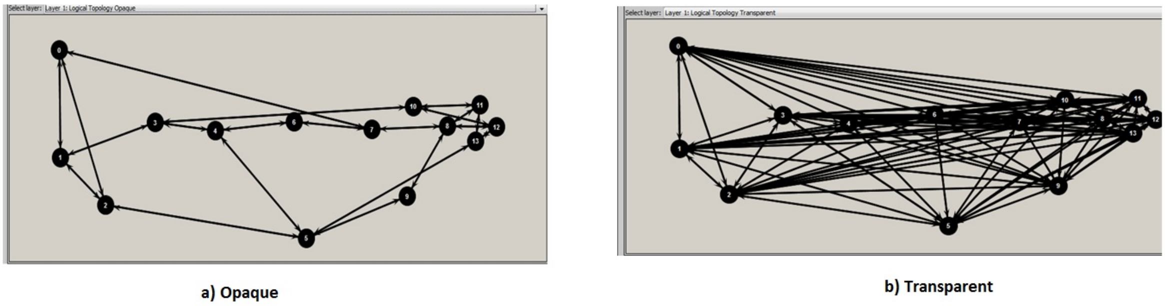
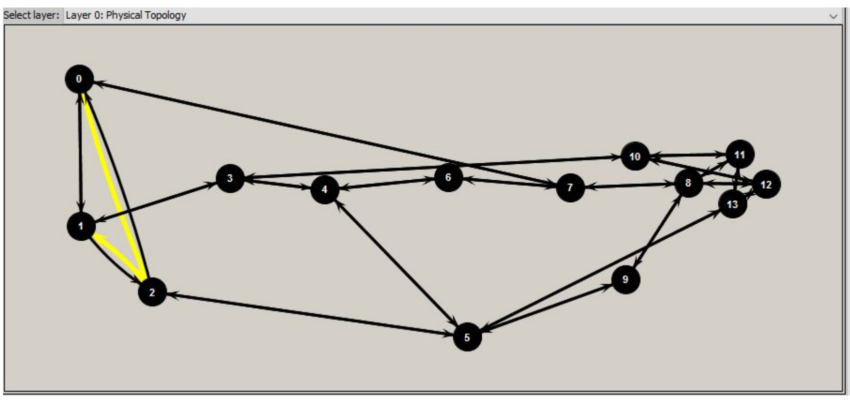


Figure 5.3: Net2Plan logical topology for NSFNET physical topology based on a) Opaque transport mode b)Transparent transport mode

All twenty one network links have the source and destination node attributes corresponding to the cities they belong, for instance, the link between node 0 and 2 has de attribute origin node equals to Seattle and destination San Diego. Obviously, to be consistent all nodes have the length attribute corresponding to the distance in kilometres they cross in the real context. Although there are 182 demands corresponding to traffic requests from each of the fourteen nodes to all the other thirteen, only 91 routes are needed because the path from "o" to "d" is assumed to be the same as "d" to". The computation of the routes obeys the same reasoning.



Unique ide...	Index	Origin node	Destinatio...
2285	0	0 (Seattle (...	1 (Palo Alto ...
2286	1	0 (Seattle (...	2 (San Dieg...
2287	2	0 (Seattle (...	3 (Salt Lake ...
2288	3	0 (Seattle (...	4 (Boulder (...
2289	4	0 (Seattle (...	5 (Houston (...
2290	5	0 (Seattle (...	6 (Lincoln (N...
2291	6	0 (Seattle (...	7 (Urbana-C...
2292	7	0 (Seattle (...	8 (Pittsburg...
2293	8	0 (Seattle (...	9 (Atlanta (...
2294	9	0 (Seattle (...	10 (Ann Arb...
2295	10	0 (Seattle (...	11 (Ithaca (...
2296	11	0 (Seattle (...	12 (Princeto...
2297	12	0 (Seattle (...	13 (College ...
2298	13	1 (Palo Alto ...	0 (Seattle (...
2299	14	1 (Palo Alto ...	2 (San Dieg...
2300	15	1 (Palo Alto ...	3 (Salt Lake ...
2301	16	1 (Palo Alto ...	4 (Boulder (...
2302	17	1 (Palo Alto ...	5 (Houston (...
2303	18	1 (Palo Alto ...	6 (Lincoln (N...
2304	19	1 (Palo Alto ...	7 (Urbana-C...
2305	20	1 (Palo Alto ...	8 (Pittsburg...
2306	21	1 (Palo Alto ...	9 (Atlanta (...
2307	22	1 (Palo Alto ...	10 (Ann Arb...
2308	23	1 (Palo Alto ...	11 (Ithaca (...

Figure 5.4: Protection traffic route between Seattle and Palo Alto

5.2 Network Cost

Regarding network cost, the approach will be based on three major factors: cost of the electrical part of the nodes, cost of the optical part of the nodes and cost of the links. Then, besides the equipment costs, this report also has the parameter *span*. The value of this variable is used to calculate the number of optical amplifiers needed in the network:

$$N^R = \sum_{l=1}^{L} \left(\left\lceil \frac{len_l}{span} \right\rceil - 1 \right) \tag{5.1}$$

- $N^R \rightarrow$ Total number of regenerators/amplifiers
- $len_l \rightarrow$ Length of link
- $span \rightarrow$ Distance between amplifiers

Thus, the number of regenerators or amplifiers is calculated based on the approximation (5.1). The equation means that the number of regenerators is given the division of the size of each link by the desired spacing between amplifiers minus one. The reason for the subtraction is to prevent the amplifier closer to the destination node from being unnecessary [26].

$$C_L = \left(\gamma_0^{OLT} \times L\right) + \left(\gamma_1^{OLT} \times \tau \times W\right) + \left(N^R \times c^R\right) \tag{5.2}$$

- $C_L \rightarrow$ Links Cost
- $\gamma_0^{OLT} \rightarrow$ OLT cost in Euros
- $L \rightarrow$ Number of unidirectional Links
- $\gamma_1^{OLT} \rightarrow$ Transponder cost in Euros
- $\tau \rightarrow$ Traffic per port
- $W \rightarrow$ Total number of optical channels
- $N^R \rightarrow$ Total number of optical amplifiers
- $c^R \rightarrow$ Optical amplifiers cost in Euros

The cost of links includes the transmission systems (OLT - Optical line termination) for each link plus a parcel depending on the optical channels and bandwidth summing with the number of amplifiers and their cost. After analysing how to calculate the cost of links, will be considered the approximation to the cost of nodes, starting with the electrical layer.

$$C_{exc} = (\gamma_{e0} \times N) + (\gamma_{e1} \times \tau \times 2 \times P_{TRIB}) \tag{5.3}$$

- $C_{exc} \rightarrow$ Electrical Equipment Cost
- $\gamma_{e0} \rightarrow$ EXC cost in Euros
- $N \rightarrow$ Number of Nodes
- $\gamma_{e1} \rightarrow$ EXC port cost in Euros per Gb/s
- $\tau \rightarrow$ Traffic supported by optical channel
- $P_{TRIB} \rightarrow$ Number of tributary ports

The equation (5.3) suggest the price of an electrical node based on the price of the electrical cross connect(electrical switching) for each node. In addition, there is a price that depends on the traffic that arrives at that node and consequently the number of ports. Then, the cost depends on the ports costs per Gb/s, the traffic per optical channel and 2 times the number of tributary ports because it's needed one for input client traffic and one for output at the node point of view. All of this equipment is used in opaque topology.

$$C_{oxc} = (\gamma_{o0} \times N) + \gamma_{o1} \times (P_{LINE} + P_{TRIB}) \tag{5.4}$$

- $C_{oxc} \rightarrow$ Optical Equipment Cost
- $\gamma_{o0} \rightarrow$ OXC cost in Euros
- $N \rightarrow$ Number of Nodes
- $\gamma_{o1} \rightarrow$ OXC port cost in Euros
- $P_{TRIB} \rightarrow$ Number of tributary ports
- $P_{LINE} \rightarrow$ Number of line ports

The equation (5.4) suggest the price of an optical node based on the price of the optical cross connect(optical switching) for each node. In addiction, there is a price depending on the cost of ports times the sum of tributary ports and the ports used to send an optical channel (line ports). Lastly, the price of node components and amplifiers span which will be the inputs of network cost are shown in the image below. It's difficult to know the real prices of network components, so the prices are just an approximation.

Parameter	Value	Description
EXC	10000	EXC cost in euros
EXCPort	1000	EXC port cost in euros pe...
OLT	15000	OLT cost in euros
OXC	20000	OXC cost in euros
OXCPort	2500	OXC port cost in euros
Transponder	5000	Transponder cost in euros
opticalAmplifier	4000	Optical amplifier cost in e...
span	100	Separation between rege...

Figure 5.5: Network cost inputs

5.3 Opaque topology results

The consequences of the network designing can be seen in Net2Plan reports. As previous chapters the results for opaque transport mode are obtained first. So, using the traffic matrix proposed in the first section, the results in terms of tributary and line ports are:

Name	Trib ports in	Trib ports out	Line Ports in	Line Ports out	Total Ports in	Total Ports out
Seattle (WA)	5	5	21	21	26	26
Palo Alto (CA)	5	5	17	17	22	22
San Diego (CA)	5	5	19	19	24	24
Salt Lake City (UT)	4	4	22	22	26	26
Boulder (CO)	4	4	27	27	31	31
Houston (TX)	6	6	34	34	40	40
Lincoln (NE)	5	5	17	17	22	22
Urbana-Champaign (IL)	6	6	27	27	33	33
Pittsburgh (PA)	6	6	33	33	39	39
Atlanta (GA)	4	4	16	16	20	20
Ann Arbor (MI)	5	5	17	17	22	22
Ithaca (NY)	5	5	16	16	21	21
Princeton (NJ)	4	4	20	20	24	24
College Park (MD)	7	7	20	20	27	27
Total	71	71	306	306	377	377

Figure 5.6: Opaque ports per node

The number of ports per node for opaque transport mode. Tributary ports are the ports used for input traffic in a node from access networks. For instance, Seattle node requires 10 tributary ports because of bidirectional traffic. To send line signals is required line ports, two ports are required per line signal because signals considered are bidirectional also. The table above is given by reports algorithm and is the basis to calculate network cost.

Category		Cost	Total
Link Cost	OLT	630,000	155,278,000
	Transponders	153,000,000	
	Amplifiers	1,648,000	
Node Cost	Electrical	37,840,000	37,840,000
	Optical	0	
Total Network Cost			193,118,000

Figure 5.7: Opaque cost given by Net2Plan

Figure 5.7 contains the table of the network cost for NFSNET and the input traffic of 5Tb/s. The important information, besides total cost, is the cost of optical switches which is equals to zero. The switching is performed in electrical domain, so the node cost is given by the electrical switching equipments. The number of transponders is proportional to the number of optical channels and the most expensive components. The total network cost in euros is 193,118,000 for the equipments price defined in previous chapter.

5.4 Transparent topology results

The results for transparent transport mode in NFSNET are shown in the current section. The number of input and output ports for each node and the network CAPEX given by the network design software are crucial information for network planning. Then, the results for the same traffic matrix are:

Name	Trib ports in	Trib ports out	Line Ports in	Line Ports out	Total Ports in	Total Ports out
Seattle (WA)	13	13	66	66	79	79
Palo Alto (CA)	13	13	64	64	77	77
San Diego (CA)	13	13	63	63	76	76
Salt Lake City (UT)	13	13	88	88	101	101
Boulder (CO)	13	13	96	96	109	109
Houston (TX)	14	14	120	120	134	134
Lincoln (NE)	13	13	53	53	66	66
Urbana-Champaign (IL)	13	13	87	87	100	100
Pittsburgh (PA)	13	13	103	103	116	116
Atlanta (GA)	13	13	53	53	66	66
Ann Arbor (MI)	13	13	68	68	81	81
Ithaca (NY)	13	13	55	55	68	68
Princeton (NJ)	13	13	72	72	85	85
College Park (MD)	14	14	70	70	84	84
Total	184	184	1058	1058	1242	1242

Figure 5.8: Transparent ports per node

An important information is the total number of ports. In opaque transport mode, the total number of ports were 377 and for transparent transport mode are 1242. Obviously, it will affect the cost of the network. The huge number of line ports is related with the grooming. The grooming link-by-link is more efficient than grooming in end nodes.Then, it is important to see the impact of the number of ports on the total cost of the network.

Category		Cost	Total
Link Cost	OLT	630,000	531,278,000
	Transponders	529,000,000	
	Amplifiers	1,648,000	
Node Cost	Electrical	36,940,000	40,325,000
	Optical	3,385,000	
Total Network Cost			571,603,000

Figure 5.9: Transparent cost given by Net2Plan

Unlike the previous model, this transport mode requires optical components in nodes. The total network cost in euros is 571,603,000 and comparing with opaque transport mode which is less than half of this value it shows how grooming is important. The number of transponders is almost five times more, because of the number of optical channels. The number of optical amplifiers for transparent and opaque transport modes is the same, because the physical links as the same length. Transparent transport mode requires less electrical switches than opaque transport mode, but the price to pay for less efficient grooming is great.

5.5 Chapter Summary

In summary, heuristic methods are crucial for network planning, since they can be applied in real networks. Despite there are several networks used for researching and academic studies, NFSNET was the option taken for testing in this work. This network from the United States of America has fourteen nodes and twenty one links, making it more generic and useful for testing. Applying both transport modes and the algorithms proposed on previous chapter, it is possible to obtain an estimation of the network ports per node and cost. It is also possible to conclude that, in general, opaque networks are cheaper than transparent networks, but this is not true when the number of optical channels is equal in both transport modes. The signal propagation is faster in transparent transport mode, then depending on the network requirements, the most efficient model may vary.

Chapter 6

Conclusions and future directions

6.1 Conclusions

Through this dissertation the optical network design problem was studied having dedicated path protection insight. The same physical topology may have several logical topologies, choosing for further study opaque and transparent types. For each logical topology approach an ILP model was developed. These models contain a set of constraints used to minimize or maximize an objective function, in order to find an optimal solution. Both ILP models proposed were implemented using a mathematical software and an open source solver, and both of them were tested in a reference network, with the comparison results being shown and analysed during chapter 3. Having ILP results as optimal solutions for the dimensioning problem, heuristic approaches were then developed to find similar solutions for the same problems.

The main advantage of using heuristics is the save in processing time. Then, in chapter 4, a set of algorithms capable of being implemented in an open source software with graphical user interface was proposed. The results obtained through the algorithms were compared with those obtained by ILP models to verify their quality. The relative error was not considerably high, so the algorithms could be a good solution for the real network design problem. By the end, in chapter 5, the heuristic solution was applied to a real network which is used primarily for academic and research purposes. The results of this test, in a more real environment, revealed some advantages in terms of cost for the opaque network, since grooming is more efficient. However, from the results of the previous chapters, it is known that when the number of channels is equal in both models it is advantageous to choose the transparent model, as it is cheaper to implement and is able to achieve a faster signal transmission. As it can be verified in the results section of the chapter 5, optical network has a huge CAPEX so it is important to invest in the improvement of the dimensioning tools to minimize the required resources.

6.2 Future directions

Through this dissertation it was clearly shown the importance of network planning in a world with an increasing number of digital devices requiring faster and infallible interconnections. While data is exponentially growing and its transport means everything, this dissertation gives way to a new exploration work which should be stopped here. There is always space for improvement especially when optimization is involved. Some of the most important topics are presented below:

1. Heuristics and ILP tested in this dissertation with transparent and opaque networks can be applied in the future to different transport types, making it possible to evaluate advantages and disadvantages of choosing translucent transport.

2. Regarding protection methods, there are also several options that should be taken into account affecting directly the network planning. The usage of shared protection or dedicated link protection may also be considered and tested. The combination of transport type and protection type create a whole new range of options which might come up as a better solution for some specific situation.

3. The customization and the inclusion of new features in Net2Plan would ease the testing process and retrieve results quicker.

4. A GIT repository was already created during this dissertation. There the existing code was released, as well as a document, shown in appendices, which explains in detail how to generate algorithms to be used in Net2Plan. Giving continuity to this repository also composes an important part of the future work, ensuring code organization and a well-documented investigation.

References

[1] Cisco, "Cisco vni forecast and methodology, 2015-2020.".

[2] R. M. D. Morais, *Planning and Dimensioning of Multilayer Optical Transport Networks*. PhD thesis, Universidade de Aveiro, 2015.

[3] http://www.av.it.pt/anp/on/refnet2.html, "Nfsnet.".

[4] H. T. Cruz, "Quality of service parameters modeling of voice over IP traffic: An overview," in *Proceedings of the International Conference on Intelligent Information Processing, Security and Advanced Communication*, IPAC '15, (New York, NY, USA), pp. 2:1–2:1, ACM, 2015.

[5] J. Färber, "Network game traffic modelling," in *Proceedings of the 1st Workshop on Network and System Support for Games*, NetGames '02, (New York, NY, USA), pp. 53–57, ACM, 2002.

[6] A. Asensio, M. Ruiz, and L. Velasco, "Requirements to support cloud, video and 5g services on the telecom cloud," in *Proc. 21st European Conf. Networks and Optical Communications (NOC)*, pp. 64–69, June 2016.

[7] "Cisco visual networking index: Forecast and methodology, 20132018," tech. rep., CISCO, 2014.

[8] C. G. Gruber, "Capex and opex in aggregation and core networks," in *Proc. Conf. Optical Fiber Communication - incudes post deadline papers OFC 2009*, pp. 1–3, Mar. 2009.

[9] M. Aibin and K. Walkowiak, "Adaptive survivability algorithm for path protection with various traffic classes in elastic optical networks," in *Proc. 7th Int. Workshop Reliable Networks Design and Modeling (RNDM)*, pp. 56–62, Oct. 2015.

[10] B. Gunawardena and P. Saengudomlert, "Dimensioning and expansion algorithm for WDM networks under traffic growth," in *Proc. Telecommunications and Information Technology 2009 6th Int. Conf. Electrical Engineering/Electronics, Computer*, vol. 02, pp. 892–895, May 2009.

[11] A. Jarray, B. Jaumard, and A. C. Houle, "Reducing the capex and opex costs of optical backbone networks," in *Proc. IEEE Int. Conf. Communications*, pp. 1–5, May 2010.

[12] Y. K. Huang, S. Zhang, E. Ip, T. Wang, Y. Aono, and T. Tajima, "Single carrier 400g WDM transmission over 2,424-Km edfa-amplified link using 100GBe-grade analog components and 4-PAm to 16-QAM conversion," in *Proc. OptoElectronics and Communication Conf. and Australian Conf. Optical Fibre Technology*, pp. 368–370, July 2014.

[13] M. Schiano, A. Percelsi, and M. Quagliotti, “Flexible node architectures for metro networks [invited],” *IEEE/OSA Journal of Optical Communications and Networking*, vol. 7, pp. B131–B140, Dec. 2015.

[14] T. Kataoka, M. Tomizawa, and Y. Tada, “Network system architecture based on 43 Gbit/s otn itu-t standard,” in *Proc. 28TH European Conf. Optical Communication*, vol. 3, pp. 1–2, Sept. 2002.

[15] E. A. Doumith and M. Gagnaire, “Impact of traffic predictability on WDM exc/oxc network performance,” *IEEE Journal on Selected Areas in Communications*, vol. 25, pp. 895–904, June 2007.

[16] Y. Ji, J. Zhang, Y. Zhao, H. Li, Q. Yang, C. Ge, Q. Xiong, D. Xue, J. Yu, and S. Qiu, “All optical switching networks with energy-efficient technologies from components level to network level,” *IEEE Journal on Selected Areas in Communications*, vol. 32, pp. 1600–1614, Aug. 2014.

[17] A. Autenrieth, A. K. Tilwankar, C. M. Machuca, and J. P. Elbers, “Power consumption analysis of opaque and transparent optical core networks,” in *Proc. 13th Int. Conf. Transparent Optical Networks*, pp. 1–5, June 2011.

[18] H. Abdalla, H. A. F. Crispim, E. T. L. Pastor, A. J. M. Soares, and L. A. Bermudez, “Optical transparent IP/WDM network simulation,” in *Proc. SBMO/IEEE MTT-S Int. Conf. Microwave and Optoelectronics*, pp. 19–23, July 2005.

[19] R. Antosik, “Protection and restoration in optical networks,” in *Proc. (Cat. No.00EX408) 2000 2nd Int Conf. Transparent Optical Networks*, pp. 13–, 2000.

[20] S. Kamamura, T. Takeda, T. Miyamura, Y. Uematsu, and K. Shiomoto, “Sticky 1 +1 path protection method by dynamic disjoint path discovery,” in *Proc. 15th Int. Conf. Optical Network Design and Modeling - ONDM 2011*, pp. 1–6, Feb. 2011.

[21] L. Gyarmati, T. Cinkler, and G. Sallai, “Srlg-disjoint multi-path protection: When lp meets ILP,” in *Proc. Networks 2008 - The 13th Int. Telecommunications Network Strategy and Planning Symp*, vol. Supplement, pp. 1–10, Sept. 2008.

[22] K. Walkowiak, M. Kucharzak, P. Kope, and A. Kasprzak, “ILP model and algorithms for restoration of anycast flows in elastic optical networks,” in *Proc. 6th Int. Workshop Reliable Networks Design and Modeling (RNDM)*, pp. 102–108, Nov. 2014.

[23] J. W. Park, C. K. Hwang, and Y. W. Lee, “New ILP formulations for multicast routing in sparse-splitting optical networks,” in *Proc. 8th international conference network and service management (cnsm) and 2012 workshop systems virtualiztion management (svm)*, pp. 238–241, Oct. 2012.

[24] E. E. Bedeer, H. M. H. Shalaby, E. S. A. El-Badawy, and S. A. Khamis, “A heuristic method of logical topology design in WDM optical networks,” in *Proc. National Radio Science Conf*, pp. 1–11, Mar. 2008.

[25] F. E. El-Khamy, M. Nasr, H. M. H. Shalaby, and H. T. Mouftah, “The performance for heuristic algorithms for virtual topology design in all-optical WDM networks,” in *Proc. 11th Int. Conf. Transparent Optical Networks*, pp. 1–4, June 2009.

[26] A. Pinto, “Aulas de redes opticas 2015-2016.”.

Appendices

NetPlanner

University of Aveiro

October 26, 2016

Contents

1 Installing Net2Plan and its main options **3**
1.1 Net2Plan download and installation 3
1.2 Net2Plan Options and installing solvers 3

2 Net2Plan Tools **6**
2.1 Creating Traffic Matrices 6
2.2 Creating the Network topologies 8
2.3 Routing and Grooming 12
2.4 Reports 13

3 Results **16**
3.1 Opaque with 1+1 protection 16
3.2 Transparent with 1+1 protection 18

4 Simulations **20**

5 Implementing new algorithms on Net2Plan **21**

6 Developing new Reports **24**

1 Installing Net2Plan and its main options

This first section will describe how to install Net2Plan and some of the solvers usable by it as well as the main program preferences available.

1.1 Net2Plan download and installation

Before downloading Net2Plan, the first step is verifying if the computer has the necessary Java Runtime Environment, it is recommended the latest release (Version 8). This can be download from the java website at `https://java.com/en/download/`. The Java Runtime Environment is necessary as Net2Plan was coded in Java.

Having installed the Java Environment it is now possible to install Net2Plan. The download is available on its website at `http://net2plan.com/download.php`. The files just need to be extracted and the program can be run without an installation by just double clicking on the file "Net2Plan.jar". The latest Net2Plan version available at the time this report was revised is 0.4.2 from July 22nd, 2016

Figure 1: Net2Plan Opening Menu

1.2 Net2Plan Options and installing solvers

To access the main Net2Plan options click "File → Options". In this window the global parameters for simulations can be changed if needed. For example, an important option to note in this tab is the parameter "defaultRunnableCodePath", whose value should be the path to the jar file containing NetPlanner algorithms. As will be explained further on, Net2Plan is an open source tool and as such, new algorithms can be implemented and the default path can be changed to the path where those will be available instead of loading them manually each time Net2Plan is opened. The remaining parameters are related to solver options, which are the default external solvers used and also the path in which the ".*dll*", ".*so*", ".*dylib*" files of each solver are available. By default there is no path for each solver but in this case it was already changed to where the solvers were installed.

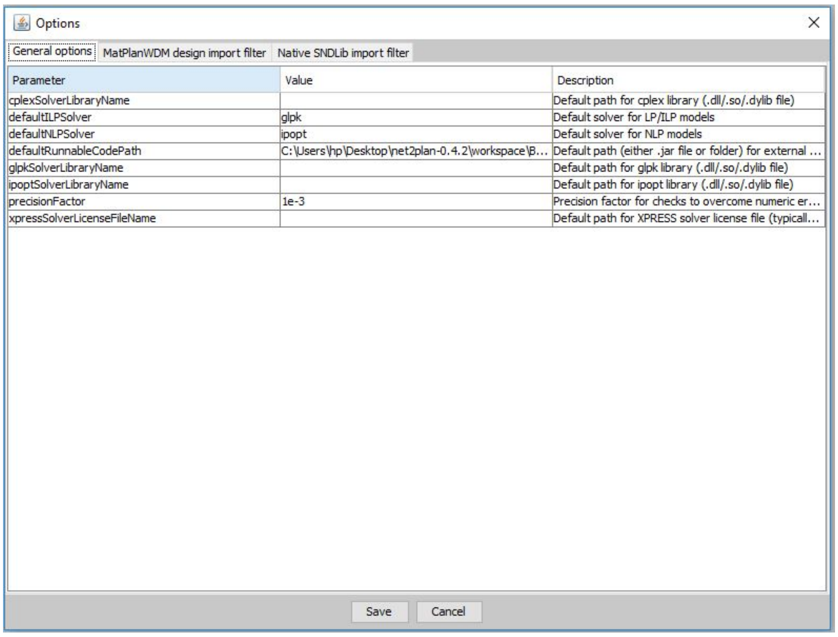


Figure 2: Net2Plan General Options

These external solvers are not extracted along with Net2Plan and as such, need to be downloaded if needed for the algorithms to be used. As ”cplex” is a paid application, only the other two solvers will be shown as the process is similar.

The ”IPOPT” solver can be downloaded from http://www.coin-or.org/download/source/Ipopt/. There are various choices available to download but for this case the *.dll* is the main file needed. An example of an algorithm which uses this solver is shown on Figure 3. Note that the ”solverLibraryName” has the path shown earlier on the ”Solver options” tab, this would have to be added manually if not introduced into the main options.

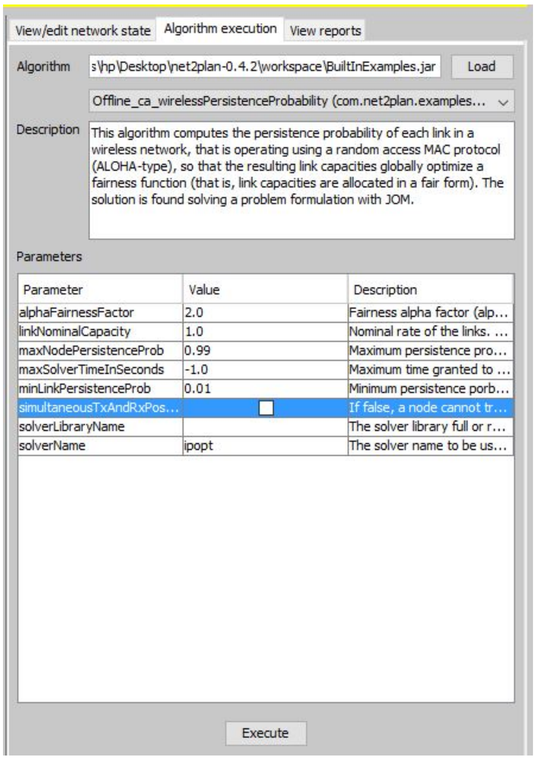


Figure 3: Net2Plan Algorithm with *ipopt* solver

The other free solver also used by some Net2Plan is ”glpk”, this one can be downloaded from http://sourceforge.net/projects/winglpk/?source=typ_redirect. An example is shown on Figure 4. Again note the path shows up as in the options.

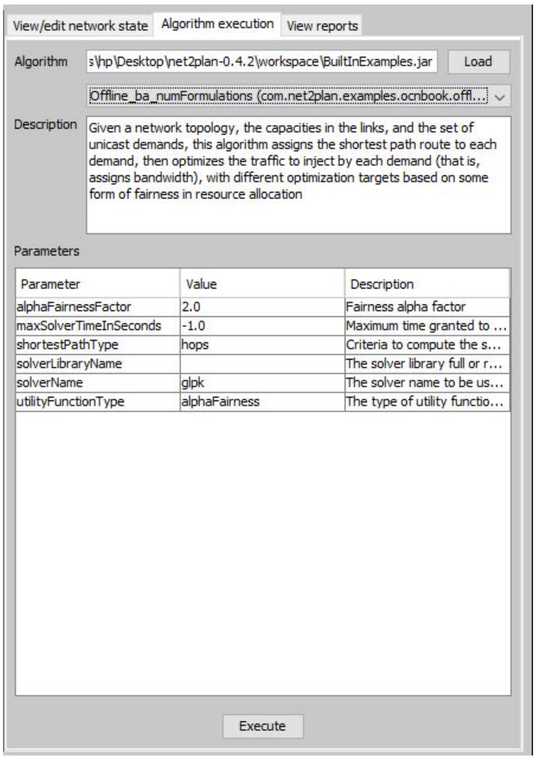


Figure 4: Net2Plan Algorithm with *glpk* solver

2 Net2Plan Tools

This section will describe in some detail the tools presented in Net2Plan as a network planner, most notably how to created a traffic matrix, design a network and some of the simulation options available.

2.1 Creating Traffic Matrices

To start creating a traffic matrix in Net2Plan go to ”Tools $\rightarrow$ Traffic matrix design” or press $Alt + 2$. The traffic matrix menu is shown on Figure 5.

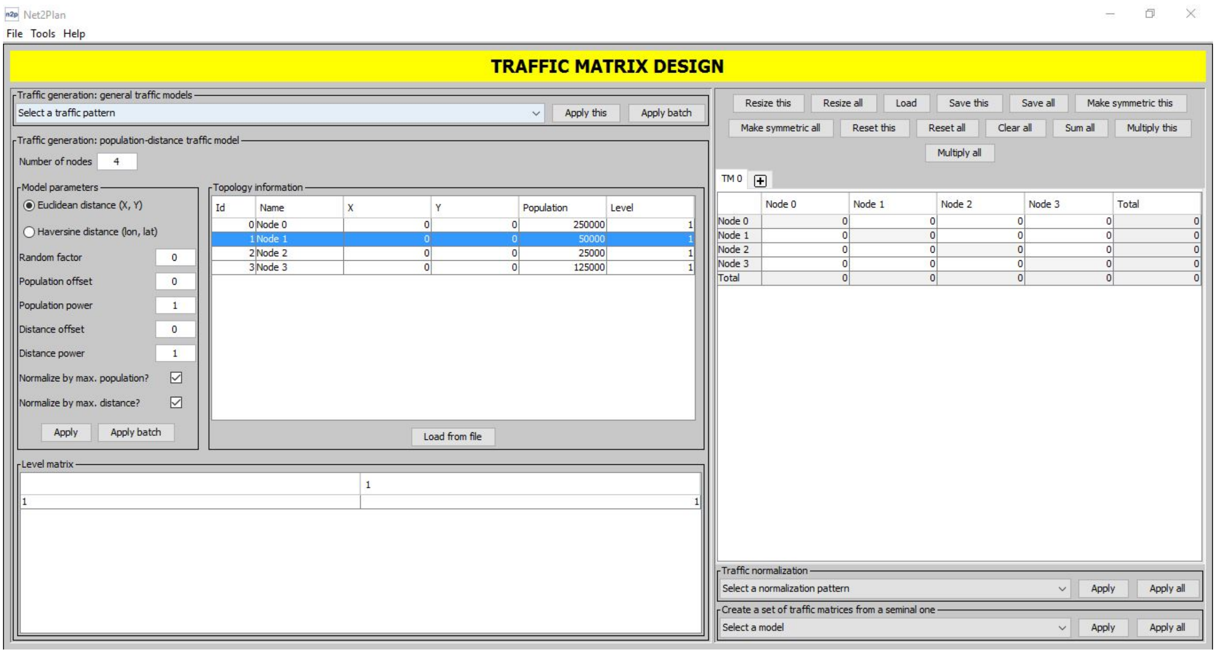


Figure 5: Net2Plan Traffic Matrix Design

On the top left side a traffic pattern can be chosen for one matrix or several if used the ”Apply batch” option.

- ”Constant” has two parameters the number of nodes and a constant value. This creates an uniform matrix with the number of nodes chosen and traffic equal to the value selected.
- ”Uniform (0,10)” has the number of nodes and the option of being symmetric as the parameters. The matrix then has the number of nodes introduced and an amount of traffic chosen randomly between 0 and 10 which can be symmetric or not depending on the choice done.
- ”Uniform (0,100)” is very similar to the other uniform option whereas in this case the traffic values are chosen randomly between 0 and 100.
- ”50% Uniform (0,100) & 50% Uniform (0,10)” and ”25% Uniform (0,100) & 75% Uniform (0,10)” are as expected a mixture of the previous two options.
- ”Gravity model” in this option a number of nodes is chosen as well as the amount of traffic both generated and received by each node. The sum of the traffic generated by all the nodes needs to be equal to the sum of the traffic received by them.

Below the traffic pattern options, an existing model can be loaded and additional parameters defined such as Population and Node Level.

On the right side a traffic matrix can be created manually by defining the number of nodes on ”resize this” and the amount of traffic can be typed on each demand. The other options above the matrix are self explanatory, for example, "multiply this" multiplies all the traffic by a constant number chosen. A point to note is that most options has an "all" choice as it is possible to have more then one matrix created.

Below the matrix part are two further options available for the matrices, the first one is the option to select a normalization pattern such as "Total normalization" where a total number of traffic can be chosen for the network and the demands are adapted to it accordingly. The other option is to create a set of matrices based on the designed one.

Figure 6 shows how to create batch of matrices with constant traffic.

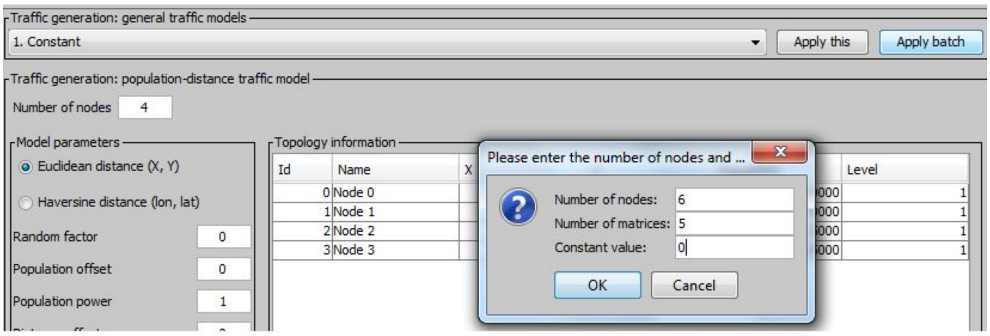
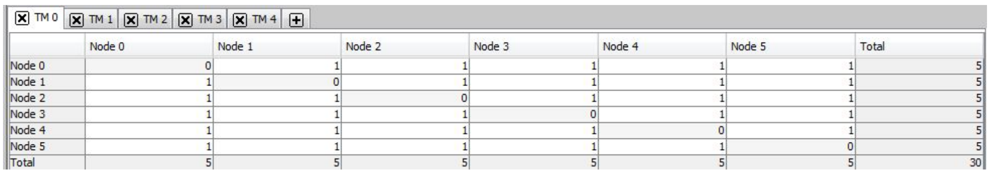


Figure 6: Net2Plan example on creating a batch of matrices

Using this option, 5 traffic matrices for a 6 node network were created all with a constant value of 1 as can be seen on figure 7 that shows the first matrix of the batch.



TM 0 | TM 1 | TM 2 | TM 3 | TM 4

	Node 0	Node 1	Node 2	Node 3	Node 4	Node 5	Total
Node 0	0	1	1	1	1	1	5
Node 1	1	0	1	1	1	1	5
Node 2	1	1	0	1	1	1	5
Node 3	1	1	1	0	1	1	5
Node 4	1	1	1	1	0	1	5
Node 5	1	1	1	1	1	0	5
Total	5	5	5	5	5	5	30

Figure 7: Net2Plan Traffic Matrix Example

This example demonstrates how several different types of traffic can be introduced for a network by creating different matrices for each. These can then be saved individually and will further on be used as traffic matrices for ODU's 0 through 4.

2.2 Creating the Network topologies

To start with the Network creation tools in Net2Plan go to ”Tools → Offline network design” or press $Alt + 1$. The network design menu is shown on Figure 8.

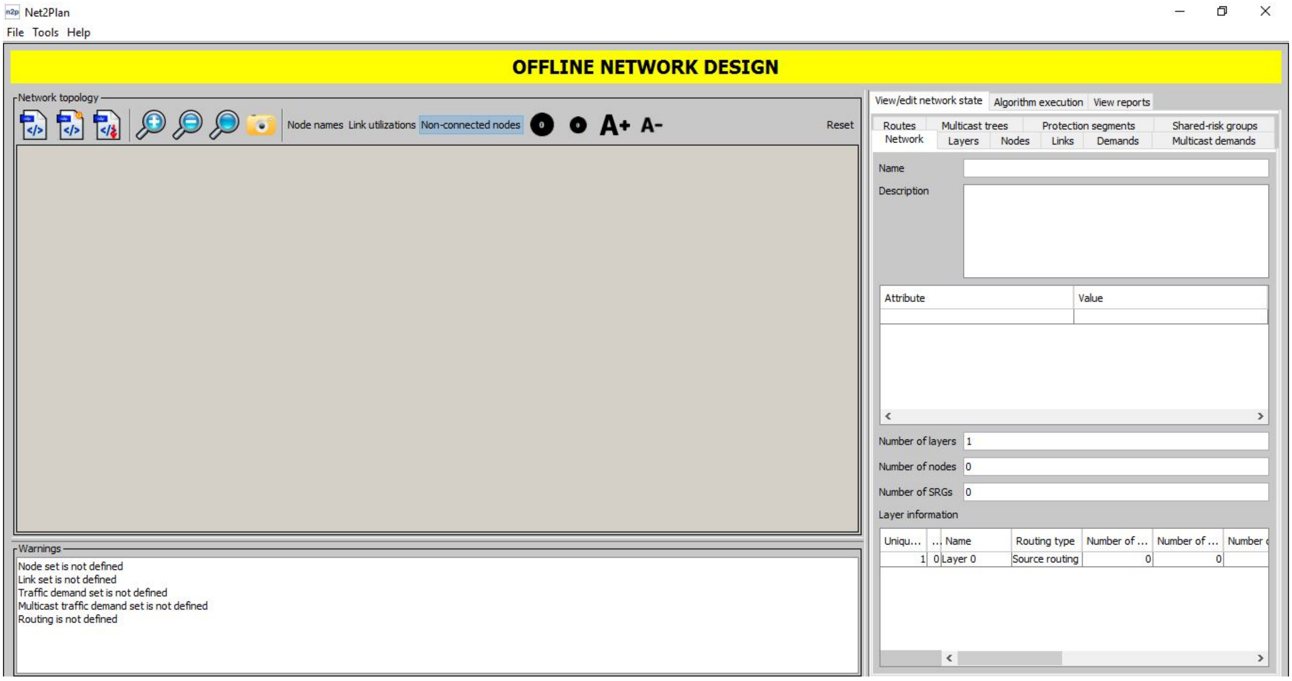


Figure 8: Net2Plan Offline Network Design

On the left side, the network topology part has the option to load an existing design and demand set or a new one can be created. To start creating a new network, first nodes have to be introduced by right clicking on the grey area and choosing ”Add node here”. Links between nodes are created by holding a click on the origin node and dragging until the destination node, holding shift before releasing the click creates 2 links, one in each direction. Another option to create links is to right click on an existing node and choosing the desired create a link option. Nodes can be moved by holding control and dragging them into the desired position.

Below the network topology is the ”Warnings” box where the parts missing from having a functional network are displayed. For example if the nodes and links where already created it should say ”traffic demand set is not defined” and ”Routing is not defined” as these were still not introduced.

The whole right side of the network design menu are the parameters separated into various tabs which will be explored further on in this document. Besides these tabs, there is also the tab for Algorithm execution where the network is modified based on built algorithms, for example a routing algorithm and the View reports tab where information on the network can be displayed from built in reports.

Figure 9 demonstrates an example of the 6 node and 16 links network created using the tools explained above. As can be seen on the image at the warning tab, this network sill has several steps left to become a fully functional network. The link capacity will be defined based on the routing algorithm chosen and the demand set will be loaded based on the matrices created.

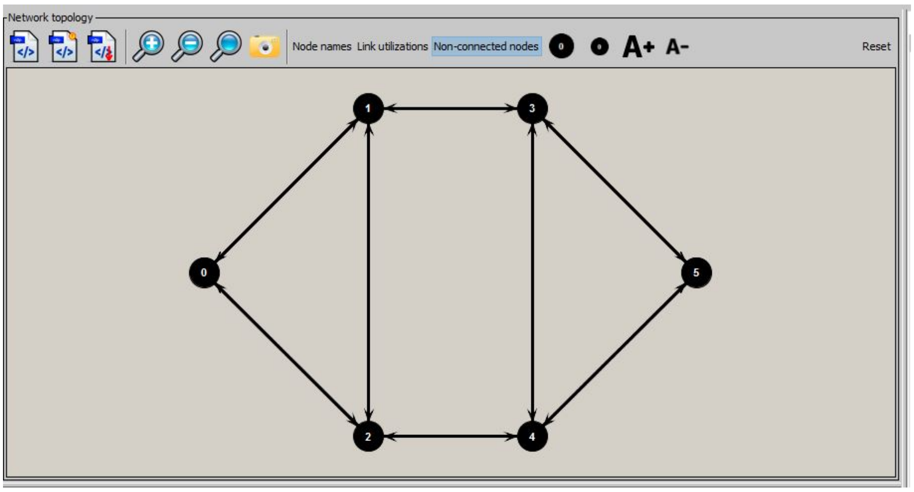
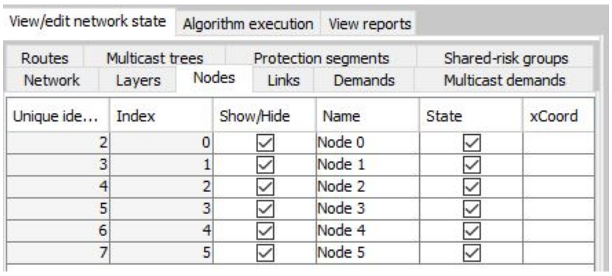


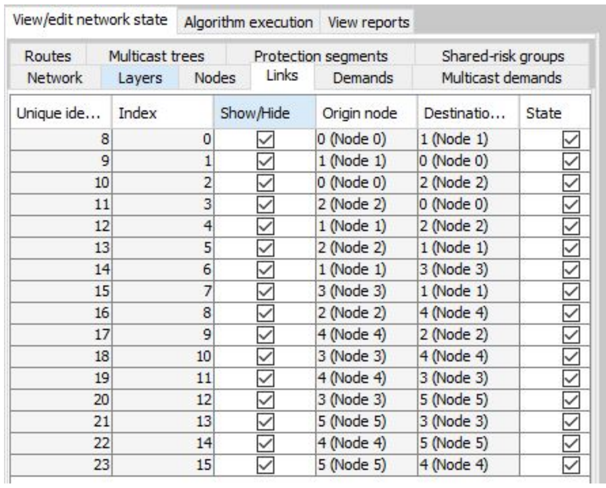
Figure 9: Net2Plan Network Example

The links and nodes parameters created for the network can be visualized and modified as seen on Figures 10(a) and 10(b) displaying the tabs for each case.



Unique ide...	Index	Show/Hide	Name	State	xCoord
2	0	☑	Node 0	☑	
3	1	☑	Node 1	☑	
4	2	☑	Node 2	☑	
5	3	☑	Node 3	☑	
6	4	☑	Node 4	☑	
7	5	☑	Node 5	☑	

(a)



Unique ide...	Index	Show/Hide	Origin node	Destinatio...	State
8	0	☑	0 (Node 0)	1 (Node 1)	☑
9	1	☑	1 (Node 1)	0 (Node 0)	☑
10	2	☑	0 (Node 0)	2 (Node 2)	☑
11	3	☑	2 (Node 2)	0 (Node 0)	☑
12	4	☑	1 (Node 1)	2 (Node 2)	☑
13	5	☑	2 (Node 2)	1 (Node 1)	☑
14	6	☑	1 (Node 1)	3 (Node 3)	☑
15	7	☑	3 (Node 3)	1 (Node 1)	☑
16	8	☑	2 (Node 2)	4 (Node 4)	☑
17	9	☑	4 (Node 4)	2 (Node 2)	☑
18	10	☑	3 (Node 3)	4 (Node 4)	☑
19	11	☑	4 (Node 4)	3 (Node 3)	☑
20	12	☑	3 (Node 3)	5 (Node 5)	☑
21	13	☑	5 (Node 5)	3 (Node 3)	☑
22	14	☑	4 (Node 4)	5 (Node 5)	☑
23	15	☑	5 (Node 5)	4 (Node 4)	☑

(b)

Figure 10: Network a) Nodes tab ; b) Links tab

On the Nodes tab most of the parameters are still 0 as there is no traffic on the network but there are three parameters that can be changed here. A node name can be set and both x and y coordinates can be defined as a more thorough alternative to define the node position.

On the links tab, again most is at 0 at this moment while the parameters that can be manually set are the link capacity, at 0 until defined and the link length which was set to the same value in every link.

Having the basic physical topology created, the next step is to load the demand set into the network. In the case where there are multiple traffic matrices an algorithm was developed to aggregate these in order for it to be possible to load all demands. For traffic matrices with ODU signals, an algorithm called "joinTrafficMatrices" can aggregate the different ODUs and convert them to ODU0 in order to have all the traffic in the same units. Besides converting the different ones to ODU0 it also creates an attribute in each demand indicating the type of signal before converting. This attribute can be seen on the demands tab after loading the resulting demand list. Figure 11 shows the algorithm to be used.

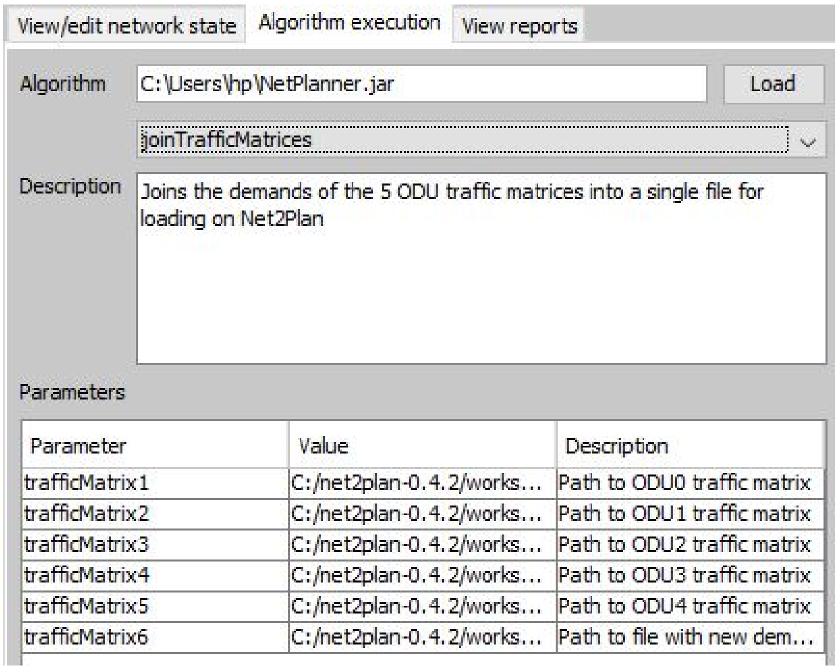


Figure 11: joinTrafficMatrices Algorithm

As can be seen on Figure 11 there are 6 user defined parameters, the first five are the paths for the traffic matrices to be aggregated in order, as said in the description. The last parameter is the resulting demand list that can then be loaded into the network.

The paths are by default defined considering Net2Plan is on C: and the matrices are in the default directory were they are saved. Lastly, the name of the files are in order ODU0.n2p through ODU4.n2p. All the path and file names can be changed to where the matrices are saved taking into account that just the order of the ODUs needs to be kept due to the conversion to ODU0 units.

To load the resulting demands into the created network the second icon on top of the network topology called ”Load a demand traffic set” is used. After this, the warning tab changes from ”Traffic demand set not defined” to ”Traffic losses: Not all the traffic is being carried”. This new warning indicates that the demand are in the network but as the routes have not yet been defined the traffic is not being transported.

In the demands tab, all the traffic that was created will be displayed in order of ODU type. For this case as all matrices were unitary and uniform, there are thirty demands with offered traffic 1 which is the ODU0 matrix and then consecutively groups of 30 demands (6 nodes) with offered traffic based on the ODU type (5 matrices). For example, an ODU1 is equivalent to two ODU0 so these demands have 2 in offered traffic and an attribute called ODU with value 1.

Before going into the network routing, the network transport mode needs to be defined by creating a logical topology. An algorithm was developed that creates a new layer consisting on this topology depending on the transport mode chosen. This algorithm can be seen on Figure 12.

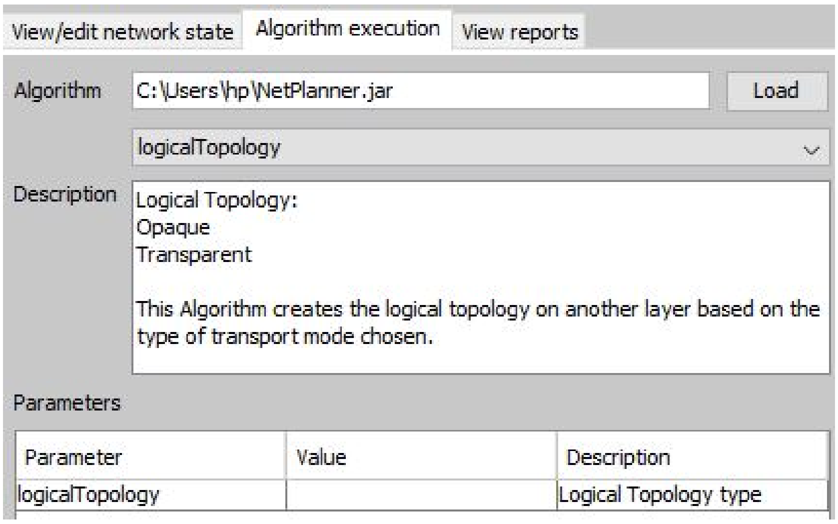
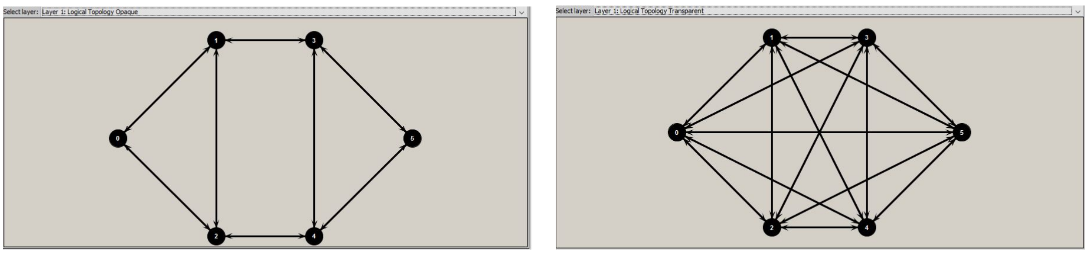


Figure 12: Net2Plan Logical Topology Algorithm

There are two user defined parameters on this algorithm. The ”logicalTopology” parameter defines the type of transport mode, Opaque or Transparent.

Besides creating this new Layer, the algorithm also copies the demands to that layer and defines the logical links based on the length of the physical ones. Figures 13(a), 13(b) demonstrate the resulting logical topologies for each transport mode.



(a) (b)

Figure 13: Logical Topology: a) Opaque; b) Transparent;

As can be seen on the logical topologies, for an Opaque transport mode the traffic goes through an OEO conversion at every node and as such the logical topology is the same as the physical one.

In the Transparent mode, there are no regeneration in intermediate nodes and as such the logical topology shows that the traffic between nodes flows directly without grooming with signals from another source.

2.3 Routing and Grooming

In this section, different routing and grooming options will be discussed for both a network without protection and using a 1+1 protection scheme (dedicated path protection).

The routing will be done based on a shortest path algorithm where the routes for each demand are are created based one either the shortest number of hops needed to reach the destination node or by shortest distance in km. The option can be chosen as a user defined parameter on the algorithm as can be seen on Figure 14. This algorithm does the routing in both the logical and physical topologies based on the transport mode chosen and makes sure routes are bidirectional meaning the route from node *o* to *d* should be the opposite direction of node *d* to *o* as there could be different routes with the shortest path that are not using the same path.

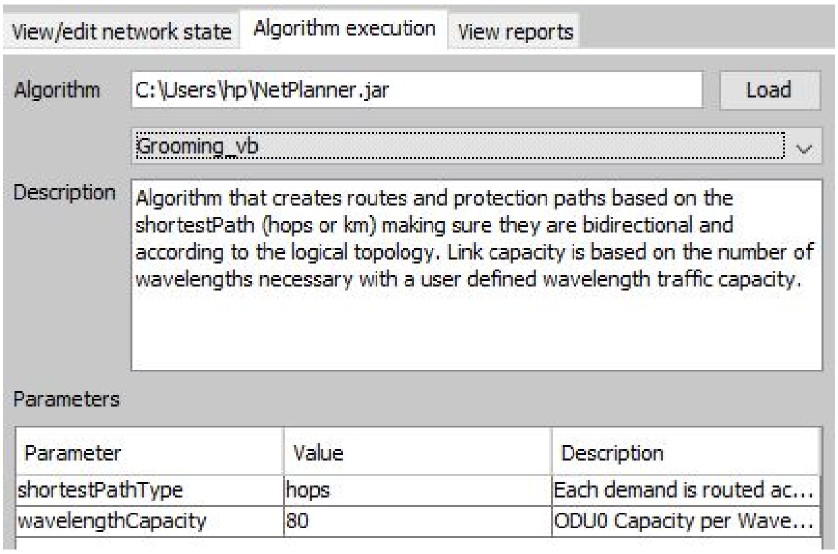


Figure 14: Net2Plan Grooming shortest Path Algorithm

Besides the metric through which the shortest path is calculated, the other available parameter defines the amount of ODU0s each wavelength is capable of carrying. By default it is set for 80 ODU0s as it is equal to an ODU4 or 100 Gbit/s.

The protection segments similarly to the routes have their own tab where information on their path, route it protects and such can be observed.

2.4 Reports

As looking separately at each tab to obtain information for different parts of the network is a slow process and does not show some important metrics, Net2Plan allows for the creation of reports where in a similar way to algorithms they can be adjusted to display the information needed, these can also be seen in html format for an easier read. In this section, the report developed will be demonstrated.

A very important aspect in network planning that is not present natively in Net2Plan is a Network Cost report. To fulfil this gap, a report was created to obtain the network Capex based on user defined equipment costs present on Table 1.

Equipment	Costs
OLT	15000€
Transponder	5000€/GB
Optical Amplifier	4000€
EXC	10000€
OXC	20000€
EXC Port	1000€/GB/s
OXC Port	2500€/port

Table 1: Equipment Costs

These Equipment costs are introduced into a report as user defined parameters as can be seen on Figure 15.

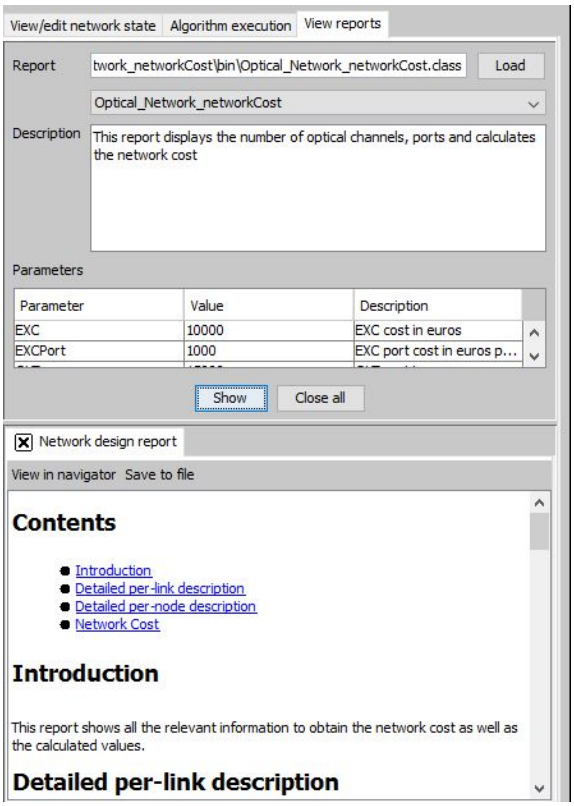


Figure 15: Network Cost Report

Besides the equipment costs, this report also has the parameter ”span”. The value of this variable is used to calculate the number of optical amplifiers needed in the network using Equation 1.

$$N^R = \sum_{l=1}^{L} \left(\left\lceil \frac{len_l}{span} \right\rceil - 1 \right) \tag{1}$$

The other parameters of this equation being:

- $N^R \rightarrow$ Total number of regenerators/amplifiers
- $len_l \rightarrow$ Length of link l
- $span \rightarrow$ Distance between amplifiers

By running the report three main categories are presented to the user.

The first category displayed by the report is the Detailed per-link description. In here the number of optical channels or wavelengths is displayed for each link based on the grooming algorithm used. The numbers displayed are based on the physical topology and represent all the wavelengths that will be needed to transport the network traffic. Using this information it is possible to obtain the average and total number of optical channels on the network.

Besides the number of wavelengths, this section also indicates the amount of amplifiers necessary in each link.

The second category is the Detailed per-node description. This section displays a table indicating how many ports are needed of each type for every node. The number of tributary ports obtained in each node is the sum of all traffic originating from that node or ending on it depending if its the input or output ports divided by the amount of traffic each optical channel can carry. This number also depends on the links through which traffic will be routed, for example, if 40 ODU0s are transmitted into 2 separate links only one wavelengths could carry it but as they are going through different routes then 2 wavelengths will be used resulting in also a need for 2 tributary ports.

The number of line ports is obtained by adding the total amount of optical channels in the links that use that specific node as origin or destination.

Finally the total number of ports is as expected the sum of all the tributary ports with the line ones. With this information the average and the total number of ports in the network can be obtained which will later be used in calculating the network cost.

Having the node and link information available, the network cost can then be calculated as displayed on the third category of the report. The Node electrical cost is obtained with Equation 2 for a Transparent Network.

$$C_{exc} = (\gamma_{e0} \times N) + (\gamma_{e1} \times \tau \times 2 \times P_{TRIB}) \tag{2}$$

- $C_{exc} \rightarrow$ Electrical Ports Cost
- $\gamma_{e0} \rightarrow$ EXC cost in Euros
- $N \rightarrow$ Number of Nodes
- $\gamma_{e1} \rightarrow$ EXC port cost in Euros per GB/s
- $\tau \rightarrow$ Traffic supported by optical channel

- $P_{TRIB} \rightarrow$ Number of tributary ports

The cost values can be obtained from Table 1, the number of nodes is a known value when designing a network, the traffic supported by optical channel is defined by the grooming algorithm or by dividing the link capacity by its amount of optical channels and the number of tributary ports was obtained on the previous section of the report.

For an Opaque network, the electrical nodes cost is similar as displayed in Equation 3.

$$C_{exc} = (\gamma_{e0} \times N) + (\gamma_{e1} \times \tau (P_{LINE} + P_{TRIB})) \tag{3}$$

The node optical cost on the other hand, can be calculated for a Transparent network using Equation 4.

$$C_{oxc} = (\gamma_{o0} \times N) + \gamma_{o1} \times (P_{LINE} + P_{TRIB}) \tag{4}$$

- $C_{oxc} \rightarrow$ Optical Ports Cost
- $\gamma_{o0} \rightarrow$ OXC cost in Euros
- $N \rightarrow$ Number of Nodes
- $\gamma_{o1} \rightarrow$ OXC port cost in Euros
- $P_{TRIB} \rightarrow$ Number of tributary ports
- $P_{LINE} \rightarrow$ Number of line ports

As for the electrical ports, the cost values were previously defined in Table 1 and as such, only the number of ports is needed. These value were obtained on the second part of the report (Detailed per-Node description).

For an Opaque network, the node optical cost is 0 as the ports are all electrical.

The Node Total Cost is as expected the sum of both the optical and electrical node costs.

The rest of the network cost is from the links. This cost is obtained with Equation 5.

$$C_L = \left(\gamma_0^{OLT} \times L\right) + \left(\gamma_1^{OLT} \times \tau \times W\right) + \left(N^R \times c^R\right) \tag{5}$$

- $C_L \rightarrow$ Links Cost
- $\gamma_0^{OLT} \rightarrow$ OLT cost in Euros
- $L \rightarrow$ Number of unidirectional Links
- $\gamma_1^{OLT} \rightarrow$ Transponder cost in Euros
- $\tau \rightarrow$ Traffic per port
- $W \rightarrow$ Total number of optical channels

- $N^R \rightarrow$ Total number of optical amplifiers
- $c^R \rightarrow$ Optical amplifiers cost in Euros

As in previous equations, the costs are all available in Table 1. The total number of optical channels can be obtained by summing the wavelengths in each link on the Detailed per-Link description section. The number of optical amplifiers was calculated previously with Equation 1.

The middle part of the equation: $\gamma_1^{OLT} \times \tau \times W$ refers to the Transponders cost while the rest is the "Fiber" and the "OLT" cost. Lastly the total network cost can be obtained by adding the Links cost with the Nodes cost.

3 Results

This section will display the results obtained using the algorithms and reports previously explained for a network with an Opaque transport mode and for one with Transparent.

3.1 Opaque with 1+1 protection

The results will be displayed only in the logical topology as in an opaque network it is the same as the physical one. Using the algorithm presented on figure 14 the routes and protection segments are created as well as the grooming.

There is not a second algorithm type for wavelengths reduction due to the fact that, that algorithm chooses the best path based on the shortest or disjointed path which in this case both need to be used one for work and one for protection. As such, is difficult to reduce in any instance the shortest path because of the algorithm performance.

The traffic matrix for the reference 6 node network, used for demonstration is shown below.

$$\begin{bmatrix} 0 & 17 & 17 & 15 & 1 & 13 \\ 17 & 0 & 32 & 7 & 15 & 114 \\ 17 & 32 & 0 & 11 & 46 & 1 \\ 15 & 7 & 11 & 0 & 11 & 7 \\ 1 & 15 & 46 & 11 & 0 & 93 \\ 13 & 114 & 1 & 7 & 93 & 0 \end{bmatrix}$$

Figure 16:

The amount of traffic that needs to be reserved in each link is as was to be expected a lot higher due to the need to reserve double the amount and in more links. The same happens in terms of wavelengths.

The number of wavelengths can again be seen on the links section of the "networkCost" report as well as the amplifiers needed on Figure 17.

The conclusions to take from these results are the same as was previously discussed as the number of amplifiers does not change and the wavelengths are the ones shown on the line matrices.

As for the nodes in the network Figure 18 shows the ports needed.

Detailed per-link description

Node Pair	Wavelengths forward	Wavelengths backward	Amplifiers forward	Amplifiers backward
Node 0 «-» Node 1	10	10	1	1
Node 0 «-» Node 2	10	10	1	1
Node 1 «-» Node 2	14	14	1	1
Node 1 «-» Node 3	14	14	1	1
Node 2 «-» Node 4	14	14	1	1
Node 3 «-» Node 4	14	14	1	1
Node 3 «-» Node 5	10	10	1	1
Node 4 «-» Node 5	10	10	1	1

Figure 17: Links for Opaque Network with 1+1 Protection

Detailed per-node description

Name	Trib ports in	Trib ports out	Line Ports in	Line Ports out	Total Ports in	Total Ports out
Node 0	9	9	20	20	29	29
Node 1	9	9	38	38	47	47
Node 2	9	9	38	38	47	47
Node 3	10	10	38	38	48	48
Node 4	10	10	38	38	48	48
Node 5	9	9	20	20	29	29
Total	56	56	192	192	248	248

Figure 18: Nodes for Opaque Network with 1+1 Protection

Again, the difference for the case without protection is only on the number of line ports as this value is based on the number of wavelengths going in or out of that node.

Comparing the number of ports obtained here with the network with a transparent transport mode, the amount is lower for the opaque network due to the reduced number of wavelengths required to route the traffic.

Lastly the total network cost is on Figure 19.

Network Cost

Category		Cost	Total
Link Cost	**OLT**	240,000	96,304,000
	Transponders	96,000,000	
	Amplifiers	64,000	
Node Cost	**Electrical**	24,860,000	24,860,000
	Optical	0	
Total Network Cost			121,164,000

Figure 19: Network Cost for Opaque Network with 1+1 Protection

The increase in cost is as described on the transparent network just based on the additional number of wavelengths required which translates in also more trunk ports needed. As noted above in the amount of ports, the cost is also lower in this instance when compared to the transparent network due to the cheaper cost in transponders and optical ports.

3.2 Transparent with 1+1 protection

For a network with a transparent transport mode, the routing as was explained before, is done using a shortest path algorithm since there are no traffic grooming between different node pairs. For this instance as there is also a 1+1 protection scheme in place, the algorithm needs to not only create the routes but also a protection segment for each route. This segment is the shortest disjoint path of the route created.

Comparing the results obtained here with the previous example, it can be seen that the amount of traffic and wavelengths is significantly higher. It is in both cases, double the amount of before since the same quantity needs to be reserved for protection.

The conclusions that can be taken from the physical topology are as explained before, the huge number of wavelengths is related to the needed for double the amount of traffic where this extra will go through even more links.

For the logical topology the Average second shortest path number of hops is 1 since as for the shortest path, it is considered that there are always direct links between nodes in a transparent network. As for the physical topology, this value is not so obvious as it has to be calculated based on the second shortest path between each node pair.

These differences for the transparent network with protection segments can also be seen on the information provided in the ”networkCost” report. Figure 20 shows the results for the links in the physical topology.

Detailed per-link description

Node Pair	Wavelengths forward	Wavelengths backward	Amplifiers forward	Amplifiers backward
Node 0 «-» Node 1	12	12	1	1
Node 0 «-» Node 2	12	12	1	1
Node 1 «-» Node 2	18	18	1	1
Node 1 «-» Node 3	18	18	1	1
Node 2 «-» Node 4	18	18	1	1
Node 3 «-» Node 4	18	18	1	1
Node 3 «-» Node 5	12	12	1	1
Node 4 «-» Node 5	12	12	1	1

Figure 20: Links for Transparent Network with 1+1 Protection

It can be seen that as expected the number of amplifiers is the same due to the link lengths remaining constant but the number of wavelengths are higher due to having a grooming scheme worst with this topology.

The results in terms of ports per node are shown below.

Detailed per-node description

Name	Trib ports in	Trib ports out	Line Ports in	Line Ports out	Total Ports in	Total Ports out
Node 0	10	10	24	24	34	34
Node 1	10	10	48	48	58	58
Node 2	10	10	48	48	58	58
Node 3	10	10	48	48	58	58
Node 4	10	10	48	48	58	58
Node 5	10	10	24	24	34	34
Total	60	60	240	240	300	300

Figure 21: Nodes for Transparent Network with 1+1 Protection

The number of tributary ports remain the same but the number of line ports increase based on the higher number of wavelengths needed in the network.

Lastly, the total network cost is shown on Figure 22.

Network Cost

Category		Cost	Total
Link Cost	OLT	240,000	120,304,000
	Transponders	120,000,000	
	Amplifiers	64,000	
Node Cost	Electrical	12,060,000	12,930,000
	Optical	870,000	
Total Network Cost			133,234,000

Figure 22: Network Cost for Transparent Network with 1+1 Protection

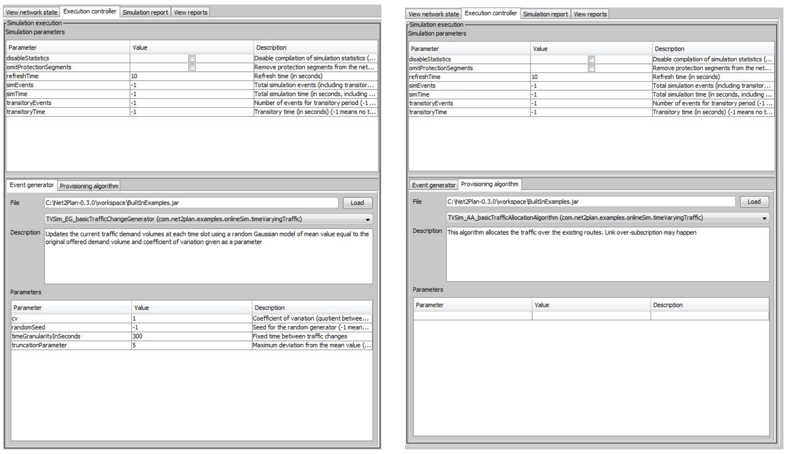
The results obtained for the network Cost confirm those obtained in the previous categories in this report. The OLT and amplifiers cost does not change as the number of links and amplifiers remains the same. Similarly, the electrical ports cost is also the same as the amount of ADD/DROP ports remains the same.

The differences are in the Transponders cost in the links and the Optical cost in the nodes. These as expected, cost more based on the increased number of them needed in the network to have a 1+1 protection scheme in a transparent transport mode network.

4 Simulations

To access the Simulations window go to ”Tools → Online Simulation” or press $Alt + 3$. The simulations menu is very similar to the one available for network design with the notable difference that in this instance the network needs to have already been saved with every definition done as all the tabs described earlier are only available here for viewing.

Using the already built network with the demand set introduced as well as routing and protection segments, an example of a Time-varying simulation is demonstrated. The main parameters to be chosen on this simulation are the ”Event generator” and the ”Provisioning algorithm”, displayed on Figures 23(a) and 23(b).



(a) (b)

Figure 23: a) Net2Plan Event generator ; b) Net2Plan Provisioning algorithm

The Event generator shown creates a time varying simulation by updating the network traffic based on the chosen parameters while the allocation algorithm in this case only allocates this traffic into the available routes. Besides these options it is also possible to change the main simulation parameters which are displayed on the top half.

Having defined all the simulation parameters and the other necessary options, the simulation can be started by just pressing ”run” below the network topology at the lower left side. The ”simulation controller” will update automatically based on the time defined at the simulation parameters or it can be paused for an update on the results.

5 Implementing new algorithms on Net2Plan

This section will demonstrate some of the possibilities provided by Net2Plan as an open source tool. By creating new algorithms or reports it is possible to adapt this program for most necessities in terms of network planning.

There are already several built-in algorithms present in Net2Plan but as it is impossible to have an algorithm built for every specific necessity it is possible for each user to build new ones or modify existing ones to fulfil what needs to be done.

As everything in Net2Plan was built in Java, the program ”Eclipse” that can be downloaded from https://eclipse.org/downloads/ was chosen as the best option for coding. All the .java files from the available algorithms in Net2Plan can be downloaded from its website and introduced into ”Eclipse” to create a class.

When opening Eclipse, the first choice is to define the work directory in which all the projects will be created. Having defined the workspace, Figure 24 demonstrates the window for creating new projects in Eclipse, this can be accessed by going into ”File → New → Java Project”. In this window, only the name needs to be defined and then finish.

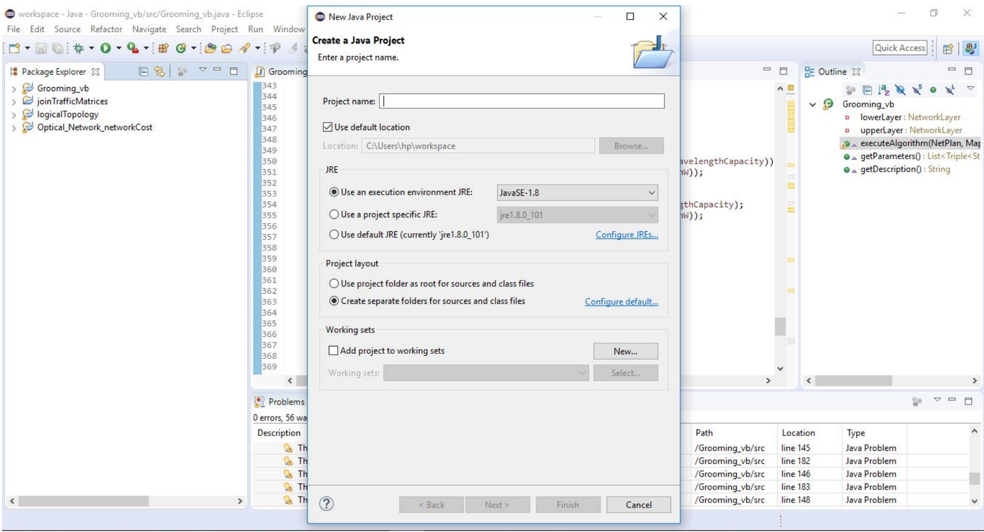


Figure 24: Eclipse new project

Having created a new project, a "src" directory should be available where the .java should be located. As a starting point, an existing algorithm should be used as a template and then modified to do its necessary purpose. Figure 25 shows a newly created project called "logicalTopology".

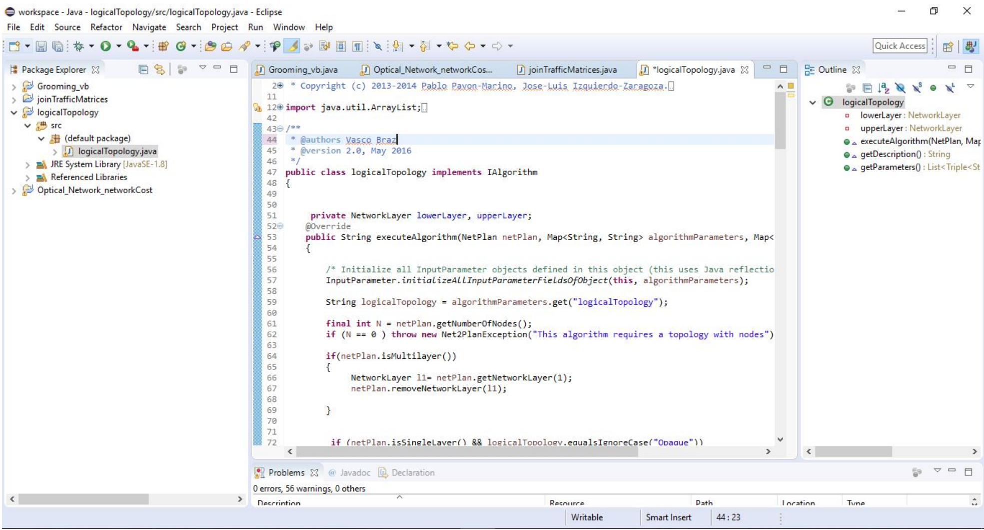
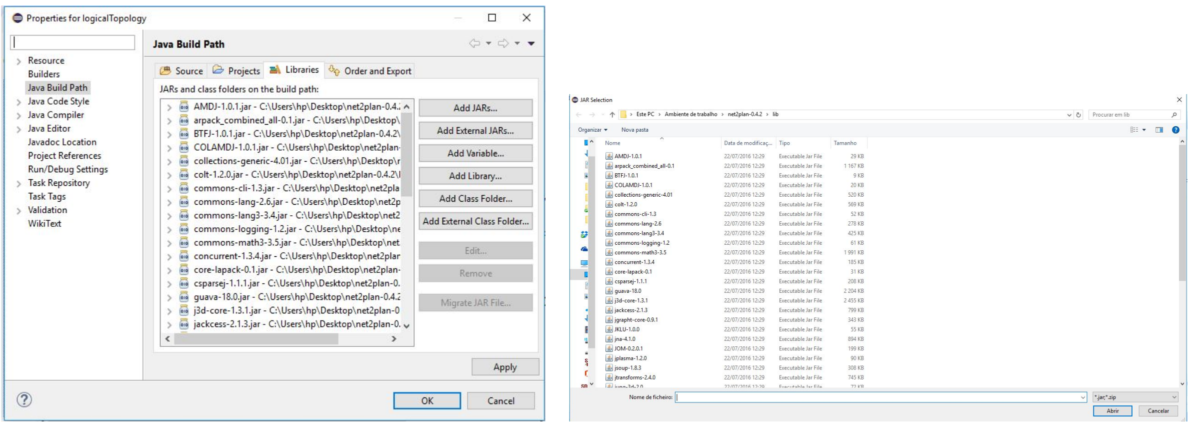


Figure 25: Eclipse new project with source file

To add the library files to a project, right click on it and choose "Build Path → Configure Build Path ..." . On the window that appears, press "Add External Jars..." and include all the files in the Net2Plan "lib" directory as shown on Figures 26(a) and 26(b).



(a) (b)

Figure 26: a) Eclipse Java Configure Build Path ; b) Net2Plan library files

To further illustrate how these modifications to algorithms work, the project created above using an existing code as a template was modified to create a new algorithm which creates the logical topology of a network in another layer.

The code created is shown on Figure 24. By saving this project on Eclipse a .class file is created on the bin directory of the project which can be loaded on Net2Plan. On the "Algorithm execution" tab at the "Offline network design", the "BuiltInExamples.jar" is loaded as the default location for algorithms and as it is a .jar file all the available ones that came with Net2Plan are integrated into it. To get the newly created algorithm available, press "Load" and find the .class file created in Eclipse as shown on Figure 27.

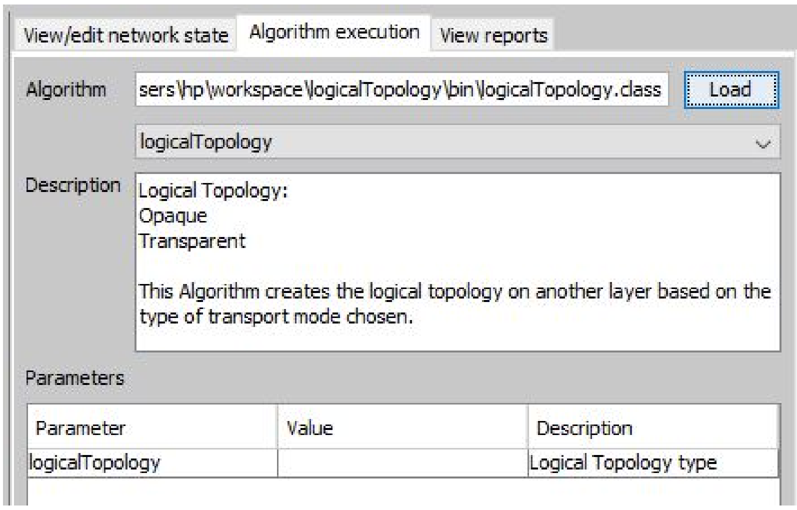
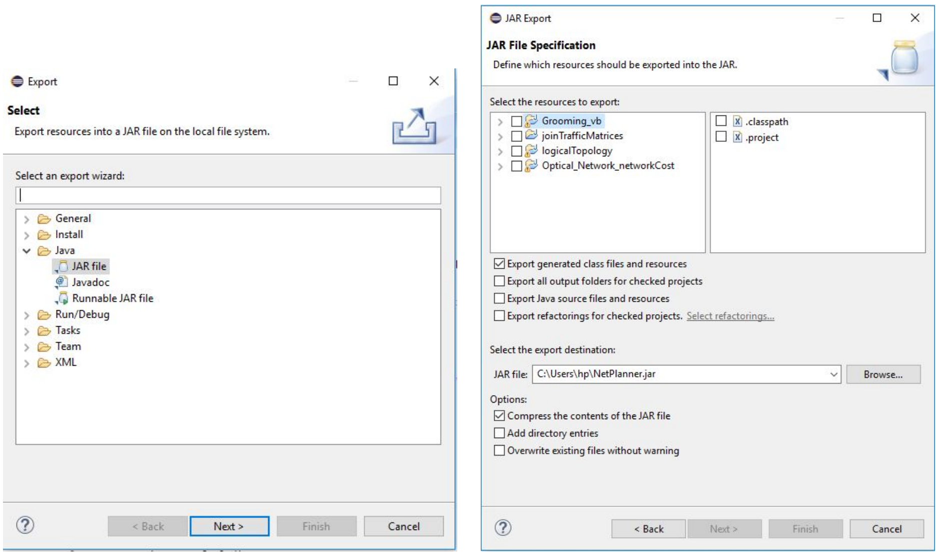


Figure 27: Net2Plan new algorithm

As was said before and can be seen on the "Description", this algorithm creates the network logical topology as was explained on section 2.2.

Algorithms developed on Eclipse can be exported into a .jar file so on Net2Plan this file can be loaded and all the algorithms developed are shown in a list in the same manner as the ones that came with the Net2Plan installation. The export option can be accessed by going into File → Export, and the menu are shown in Figures 28(a) and 28(b).



(a) (b)

Figure 28: a) Eclipse export ; b) Projects to export into a .jar file

By default only the .class files are exported along with the necessary libraries so that the algorithms can be loaded on Net2Plan. There is however an option to also export the .java files so that if needed the ones who will use the code also have access to it if they need to change it.

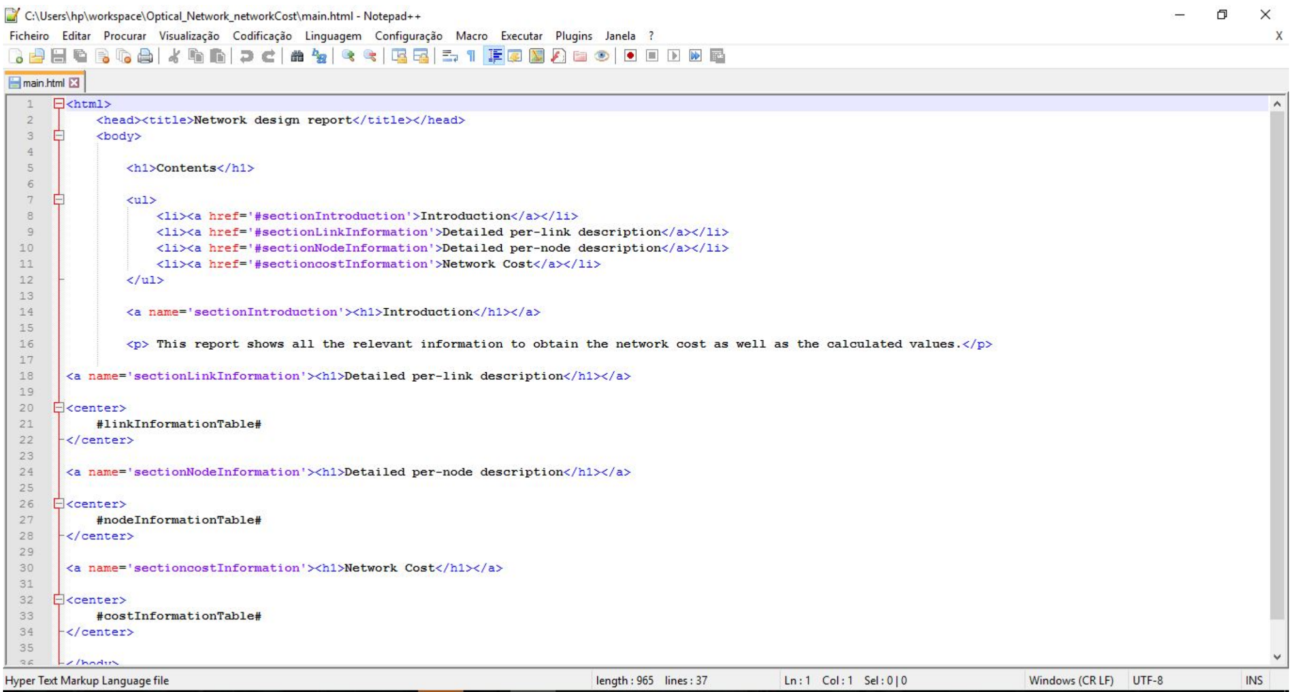
6 Developing new Reports

Similarly to the way algorithms can be modified or new ones created, also reports can be done using almost the same steps. For the following examples, the ”Optical_Network_networkcost” is being used as a basis for modifying or creating new reports.

An important point to note as the main difference as to when modifying algorithms, is that in this case not only are the Net2Plan libraries needed but also the extra files summoned by the report. These files can be found opening the ”BuiltInExamples.jar” file in the Net2Plan directory on the corresponding report.

For the report being used there is an .html file called ”main” which is where the information to be displayed in html form is described as well as several image files that are displayed in the report. As such, if the modifications to be done in the reports are to be shown in html format the ”main.html” file needs to be modified in order to adapt to these changes.

The tables themselves are created in eclipse as Java code but the html file needs to be opened for example with ”Notepadd++” to change some its code as the tables are being appended into the html. Figure 29 shows the modified html that is used in the Optical_Network_networkcost.




```
<html>
    <head><title>Network design report</title></head>
    <body>

        <h1>Contents</h1>

        <ul>
            <li><a href='#sectionIntroduction'>Introduction</a></li>
            <li><a href='#sectionLinkInformation'>Detailed per-link description</a></li>
            <li><a href='#sectionNodeInformation'>Detailed per-node description</a></li>
            <li><a href='#sectioncostInformation'>Network Cost</a></li>
        </ul>

        <a name='sectionIntroduction'><h1>Introduction</h1></a>

        <p> This report shows all the relevant information to obtain the network cost as well as the calculated values.</p>

<a name='sectionLinkInformation'><h1>Detailed per-link description</h1></a>

<center>
    #linkInformationTable#
</center>

<a name='sectionNodeInformation'><h1>Detailed per-node description</h1></a>

<center>
    #nodeInformationTable#
</center>

<a name='sectioncostInformation'><h1>Network Cost</h1></a>

<center>
    #costInformationTable#
</center>

</body>
```

Figure 29: html file for Network Cost report

As can be seen, this is a simple example of an html file since there are only hyper links created to link the contents index to the tables. Other options could be added as for example, hyper links to each of the network costs with the formula describing its calculations by adding the necessary information in this file. These extra options are present on more complex reports such as the ”Report_networkDesign” where the images used are equations showcasing how some of the calculations are done.