

NetXPTO - NetPlanner

14 de Novembro 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	9
4.1.3	Dimensioning using Heuristics	12
4.1.4	Analysis and comparison of results	12
4.2	Transparent with 1+1 Protection	13
4.2.1	Physical Network Topology	13
4.2.2	Dimensioning using ILP	15
4.2.3	Dimensioning using Heuristics	17
4.2.4	Analysis and comparison of results	17
4.3	Translucent with 1+1 Protection	18
4.3.1	Physical Network Topology	18
4.3.2	Dimensioning using ILP	19
4.3.3	Dimensioning using Heuristics	22
4.3.4	Analysis and comparison of results	22
5	Attachments	23
5.1	Installing Net2Plan and its main options	23
5.1.1	Net2Plan download and installation	23
5.1.2	Net2Plan Options and installing solvers	24
5.2	Net2Plan Tools	26

<i>Conteúdo</i>	2
5.2.1 Creating Traffic Matrices	26
5.2.2 Creating the Network topologies	29
5.2.3 Routing and Grooming	36
5.2.4 Reports	37
5.3 Results	40
5.3.1 Opaque with 1+1 protection	41
5.3.2 Transparent with 1+1 protection	43
5.4 Simulations	46
5.5 Implementing new algorithms on Net2Plan	47
5.6 Developing new Reports	50

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

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

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 his own branch with his/her name.

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

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 transportation is that it eliminates the build up of physical deficiencies and allows for total flexibility in terms of change and cleanliness. It can thus improve the usability of optical channels by providing grooming at each node.

4.1.1 Physical Network Topology

4.1.1.1 Reference Network

As we can see in the figure, 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.

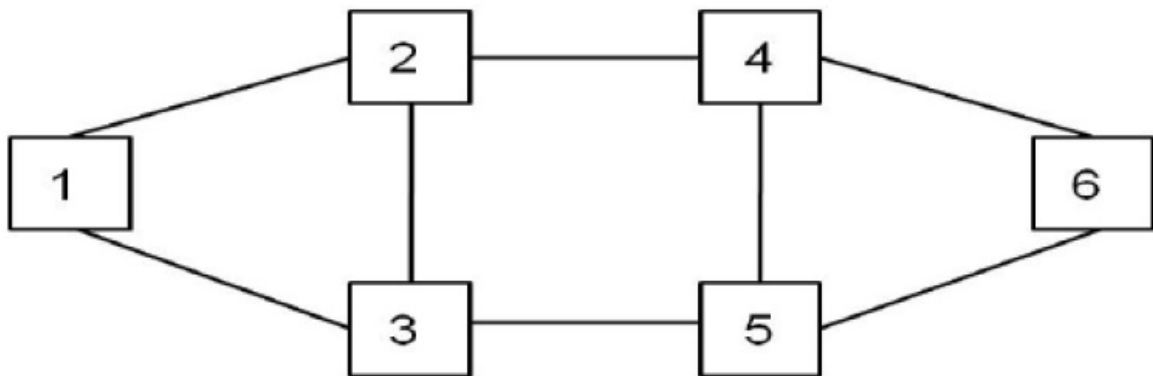


Figura 4.1: Physical Topology of the Reference Network.

As we can see from table 4.1, to do all the calculations necessary for this project, let us know the value of the traffic used. This value is defined depending on the scenario used, as we can see:

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

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

Constant	Description	Value
N	Number of Nodes	6
L	Number of Bidirectional Links	8
$\langle \delta \rangle$	Node out-degree	2,667
$\langle \text{len} \rangle$	Mean Link Length (km)	500
$\langle h \rangle$	Mean Number of Hops,for Working Paths	1,533
$\langle h' \rangle$	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). The way the nodes are arranged geographically can be seen from the following figure.

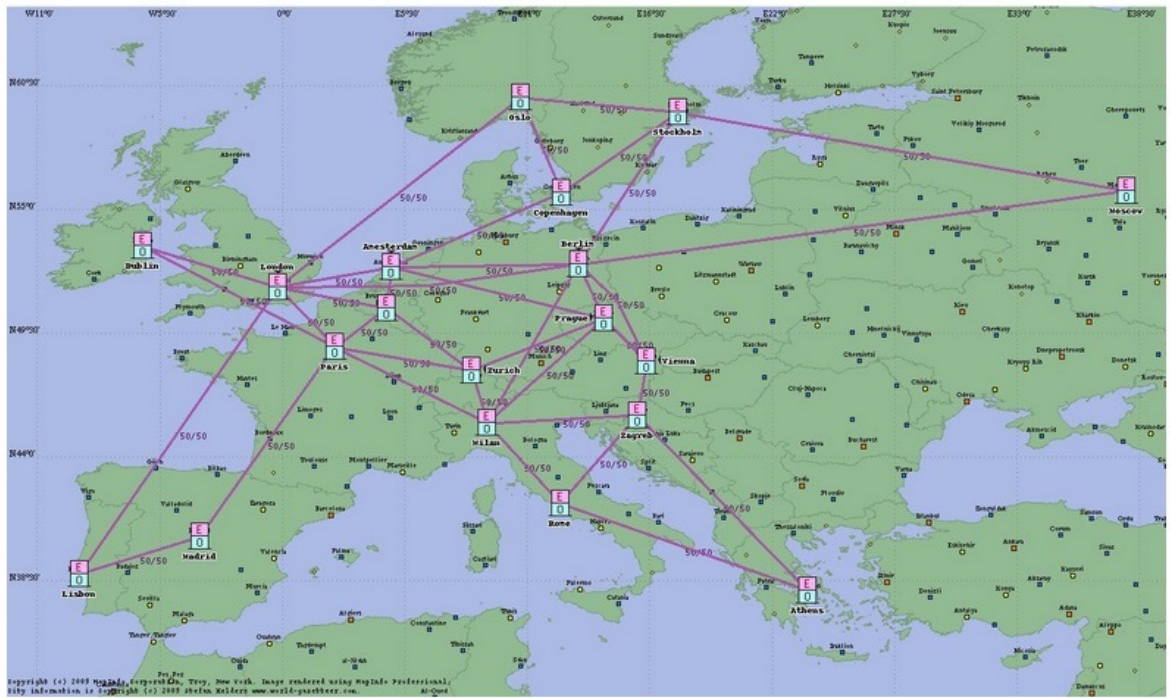


Figura 4.2: Physical Topology of the Realistic Network.

Again, to make all the necessary calculations, only the value of the traffic used is missing. This value is set depending on the scenario used, as we can see:

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

The table 4.2 shows the values of the variables associated with this network.

Constant	Description	Value
N	Number of Nodes	19
L	Number of Bidirectional Links	37
$\langle \delta \rangle$	Node out-degree	3,89
$\langle \text{len} \rangle$	Mean Link Length (km)	753,76
$\langle h \rangle$	Mean Number of Hops,for Working Paths	2,3
$\langle h' \rangle$	Mean Number of Hops,for Backup Paths	3,2

Tabela 4.2: Table of realistic network values

4.1.2 Dimensioning using ILP

4.1.2.1 ILP models

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).

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

subject to

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

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

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

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

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

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

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

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 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.

The value of the CAPEX of the network will be calculated based on the costs of the equipment present in the figure 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

Figura 4.3: 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.

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

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.

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

- $C_{exc} \rightarrow$ 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.

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

- $C_L \rightarrow$ 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.

Using the values calculated through MatLab as well as the values indicated in table 4.1 or table 4.2 (depending on the scenario used) and figure 4.3 we can finally calculate the CAPEX value using equations 4.10 and 4.11 for the various situations mentioned.

Low Traffic scenario:

$$C_L = 24\,336\,000\text{€}$$

$$C_N = C_{exc} = 5\,860\,000\text{€}$$

$$CAPEX = 24\,336\,000 + 5\,860\,000 = 30\,196\,000\text{€}$$

High Traffic scenario:

$$C_L = 191\,336\,000\text{€}$$

$$C_N = C_{exc} = 48\,260\,000\text{€}$$

$$CAPEX = 191\,336\,000 + 48\,260\,000 = 239\,596\,000\text{€}$$

4.1.3 Dimensioning using Heuristics

4.1.3.1 Heuristics Models

4.1.3.2 Heuristics Results

4.1.4 Analysis and comparison of results

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

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

4.2.1.1 Reference Network

As we can see in the figure, 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.

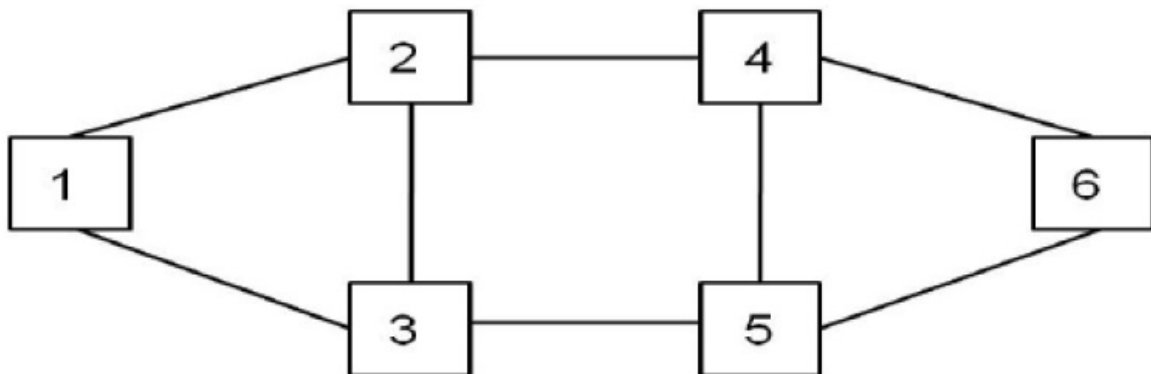


Figura 4.4: Physical Topology of the Reference Network.

As we can see from table 4.3, to do all the calculations necessary for this project, let us know the value of the traffic used. This value is defined depending on the scenario used, as we can see:

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

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

Constant	Description	Value
N	Number of Nodes	6
L	Number of Bidirectional Links	8
$\langle \delta \rangle$	Node out-degree	2,667
$\langle \text{len} \rangle$	Mean Link Length (km)	500
$\langle h \rangle$	Mean Number of Hops,for Working Paths	1,533
$\langle h' \rangle$	Mean Number of Hops,for Backup Paths	2,467

Tabela 4.3: 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.

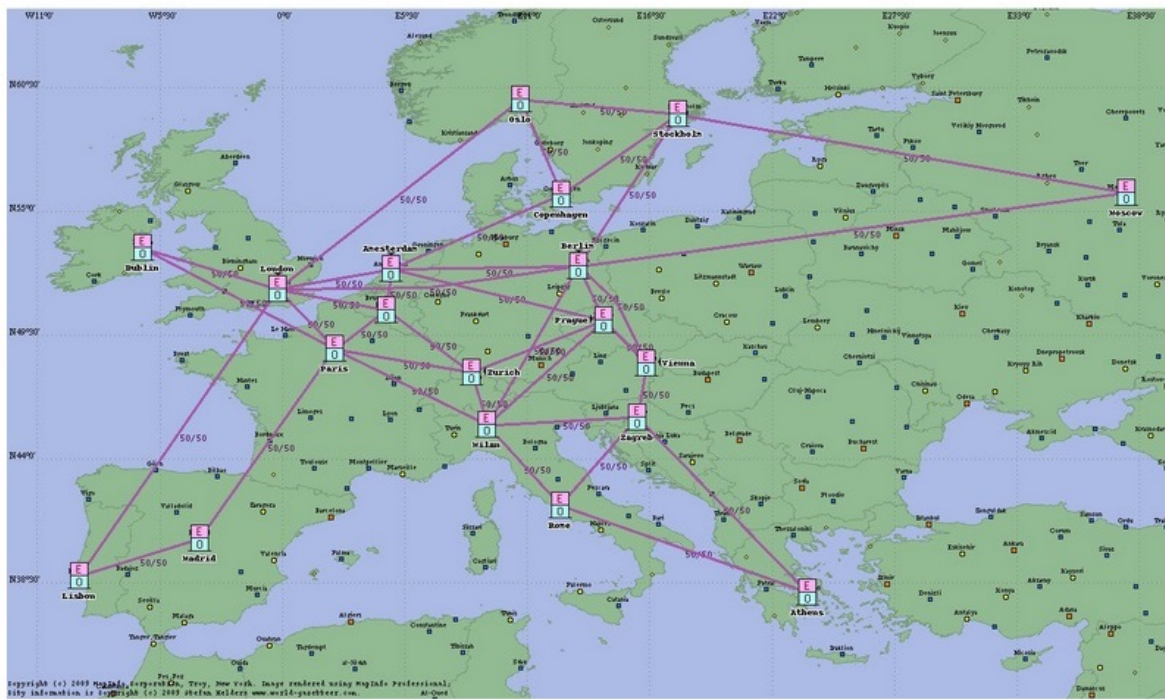


Figura 4.5: Physical Topology of the Realistic Network.

Again, to make all the necessary calculations, only the value of the traffic used is missing. This value is set depending on the scenario used, as we can see:

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

The table 4.4 shows the values of the variables associated with this network.

Constant	Description	Value
N	Number of Nodes	19
L	Number of Bidirectional Links	37
$\langle \delta \rangle$	Node out-degree	3,89
$\langle \text{len} \rangle$	Mean Link Length (km)	753,76
$\langle h \rangle$	Mean Number of Hops,for Working Paths	2,3
$\langle h' \rangle$	Mean Number of Hops,for Backup Paths	3,2

Tabela 4.4: Table of realistic network values

4.2.2 Dimensioning using ILP

4.2.2.1 ILP Models

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.

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

subject to

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

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

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

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

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

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

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.

The value of the CAPEX of the network will be calculated based on the costs of the equipment present in the figure 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

Figura 4.6: 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.

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

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 4.21 and the electrical cost by the equation 4.20.

$$C_{exc} = (\gamma_{e0} \times N) + \gamma_{e1} \times (2 \times T_1) \quad (4.20)$$

$$C_{oxc} = (\gamma_{o0} \times N) + \gamma_{o1} \times (P_{LINE} + P_{ADD}) \quad (4.21)$$

- $C_{oxc} \rightarrow$ 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

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

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

To perform the calculations using the implementation of the models described in section 4.2.2.1 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.

Using the values calculated through MatLab as well as the values indicated in table 4.3 or table 4.4 (depending on the scenario used) and figure 4.6 we can finally calculate the CAPEX value using equations 4.20, 4.21 and 4.22 for the various situations mentioned.

Low Traffic scenario:

$$\begin{aligned} C_L &= 44\,336\,000\text{€} \\ C_N &= C_{oxc} + C_{exc} = 2\,515\,000\text{€} \\ CAPEX &= 44\,336\,000 + 2\,515\,000 = 46\,851\,000\text{€} \end{aligned}$$

High Traffic scenario:

$$\begin{aligned} C_L &= 391\,336\,000\text{€} \\ C_N &= C_{oxc} + C_{exc} = 21\,445\,000\text{€} \\ CAPEX &= 391\,336\,000 + 21\,445\,000 = 412\,781\,000\text{€} \end{aligned}$$

4.2.3 Dimensioning using Heuristics

4.2.3.1 Heuristics Models

4.2.3.2 Heuristics Results

4.2.4 Analysis and comparison of results

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

As we can see in the figure, 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.

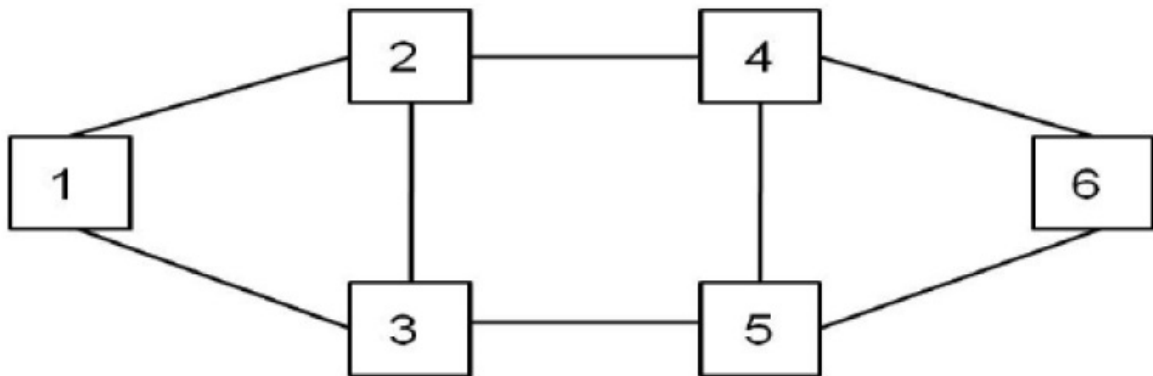


Figura 4.7: Physical Topology of the Reference Network.

The following table shows 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 4.5: Table of reference network values

As we can see from table 4.5, to do all the calculations necessary for this project, let us know the value of the traffic used. This value is defined depending on the scenario used, as we can see:

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

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.



Figura 4.8: Physical Topology of the Realistic Network.

Again, to make all the necessary calculations, only the value of the traffic used is missing. This value is set depending on the scenario used, as we can see:

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

4.3.2 Dimensioning using ILP

4.3.2.1 ILP models

$$\text{minimize} \quad \sum_{(p,k,b)} l[p,k,b] \quad (4.23)$$

The table 4.6 shows the values of the variables associated with this network.

Constant	Description	Value
N	Number of Nodes	19
L	Number of Bidirectional Links	37
$\langle \delta \rangle$	Node out-degree	3,89
$\langle \text{len} \rangle$	Mean Link Length (km)	753,76
$\langle h \rangle$	Mean Number of Hops,for Working Paths	2,3
$\langle h' \rangle$	Mean Number of Hops,for Backup Paths	3,2

Tabela 4.6: Table of realistic network values

subject to

$$\sum_{k \neq s} L[s, d, o, s, k] == \text{traffic}[o, s, d] \quad \forall s, d, o : s < d \quad (4.24)$$

$$\sum_{k \neq p, k \neq s} L[s, d, o, p, k] == \sum_{k \in \{p, d\}} L[s, d, o, k, p] \quad \forall s, d, p, o : s < d : p \neq s : p \neq d \quad (4.25)$$

$$\sum_{k \neq d} L[s, d, o, k, d] == \text{traffic}[o, s, d] \quad \forall s, d, o : s < d \quad (4.26)$$

$$\sum_{(s, d, o) : s < d} (\text{granularities}[o] \times L[s, d, o, p, k]) \Leftarrow \sum_b BD[b] \times l[p, k, b] \quad \forall p, k \quad (4.27)$$

$$L[s, d, o, p, k] \geq 0; \quad \forall s, d, p, k, o : s < d \quad (4.28)$$

$$\sum_{j \neq p} y[p, k, p, j, b] \equiv l[p, k, b] \quad \forall k, p, b \quad (4.29)$$

$$\sum_{j \neq i, j \neq p} y[p, k, i, j, b] \equiv \sum_{j \neq i, j \neq k} y[p, k, j, i, b] \quad \forall k, p, i, b : i \neq k : i \neq p \quad (4.30)$$

$$\sum_{j \neq k} y[p, k, j, k, b] \equiv l[p, k, b] \quad \forall k, p, b \quad (4.31)$$

$$\sum_{(p, k, b)} (y[p, k, i, j, b] + y[p, k, j, i, b]) \leq 80 \times C[i, j] \quad \forall i, j : i < j \quad (4.32)$$

$$y[p, k, i, j, b] \geq 0 \quad \forall p, k, i, j, b \quad (4.33)$$

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 figure 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

Figura 4.9: 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.

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

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:

$$C_L = \text{€}$$

$$C_N = \text{€}$$

$$CAPEX = \text{€}$$

High Traffic scenario:

$$C_L = \text{€}$$

$$C_N = \text{€}$$

$$CAPEX = \text{€}$$

4.3.3 Dimensioning using Heuristics

4.3.3.1 Heuristics Models

4.3.3.2 Heuristics Results

4.3.4 Analysis and comparison of results

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.

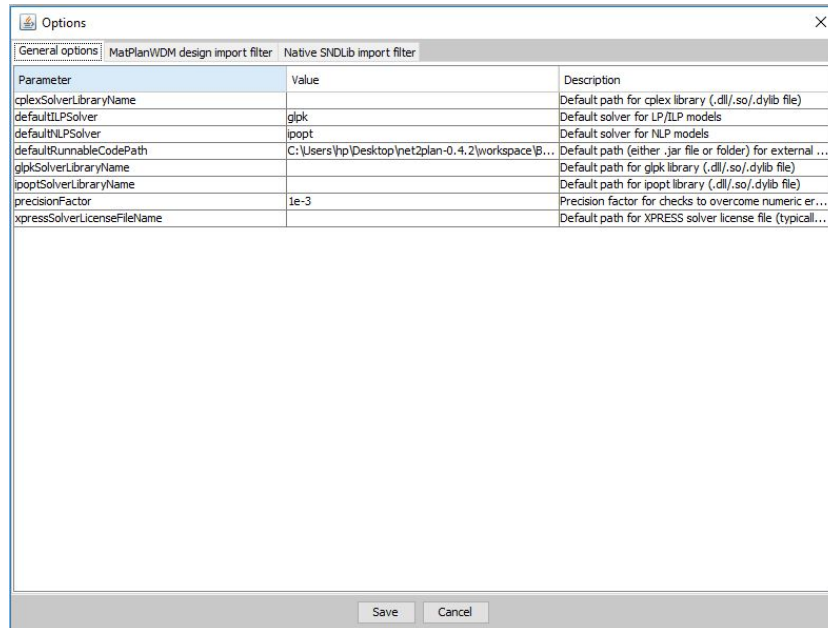


Figura 5.2: Net2Plan General Options

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.

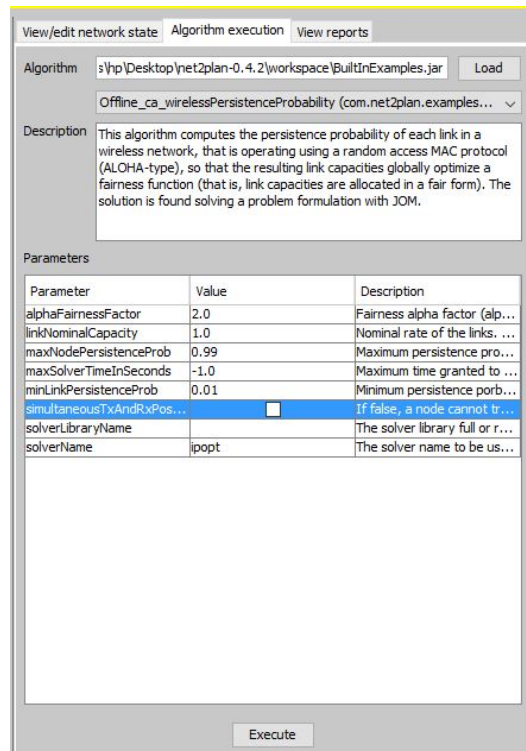


Figura 5.3: Net2Plan Algorithm with *ipopt* solver

5.2 Net2Plan Tools

This section will describe in some detail the tools presented in Net2Plan as a network planner, most notably how to create 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 → 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 a 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.

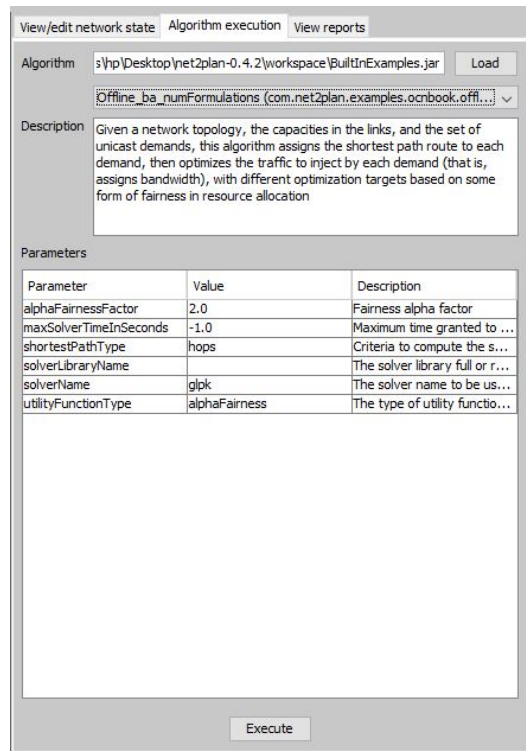
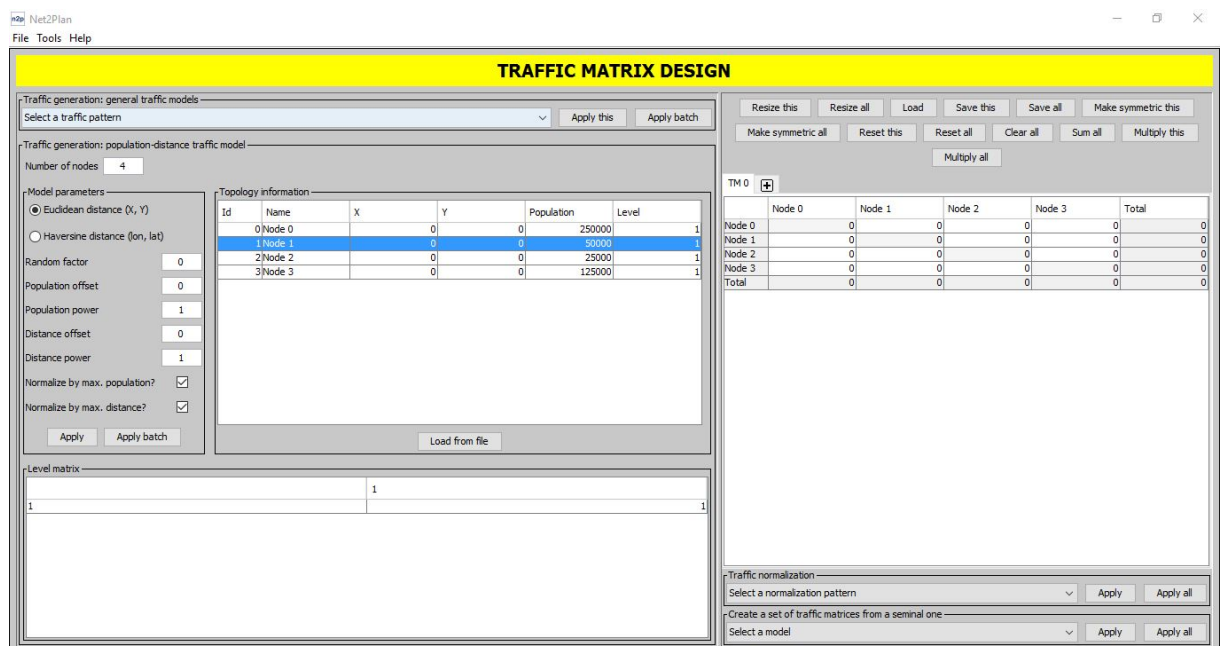
Figura 5.4: Net2Plan Algorithm with *glpk* solver

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 than 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.

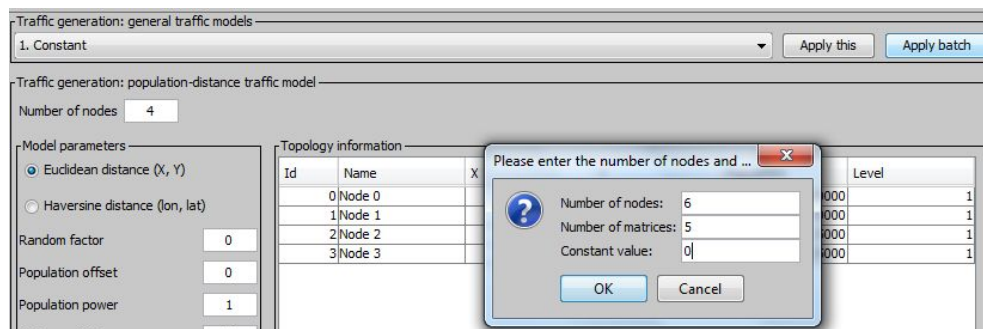


Figura 5.6: Net2Plan example on creating a batch of matrices

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.

	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.

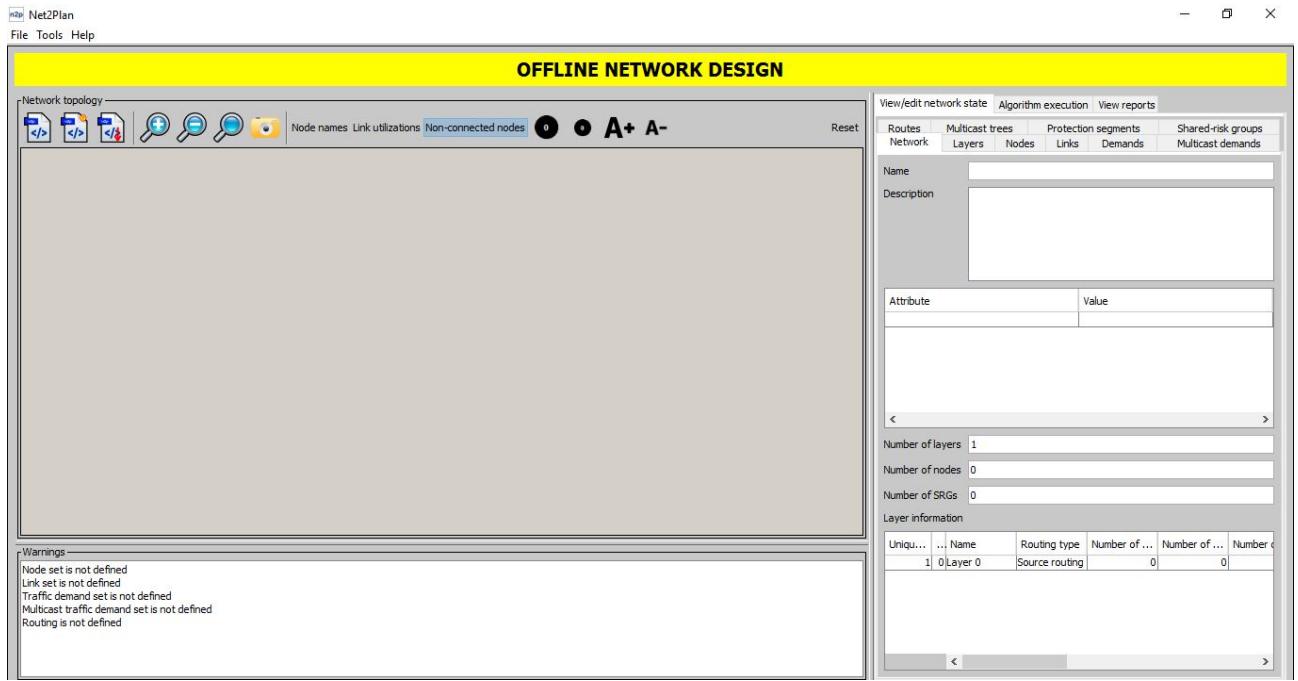


Figura 5.8: Net2Plan Offline Network Design

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 were 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 still 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.

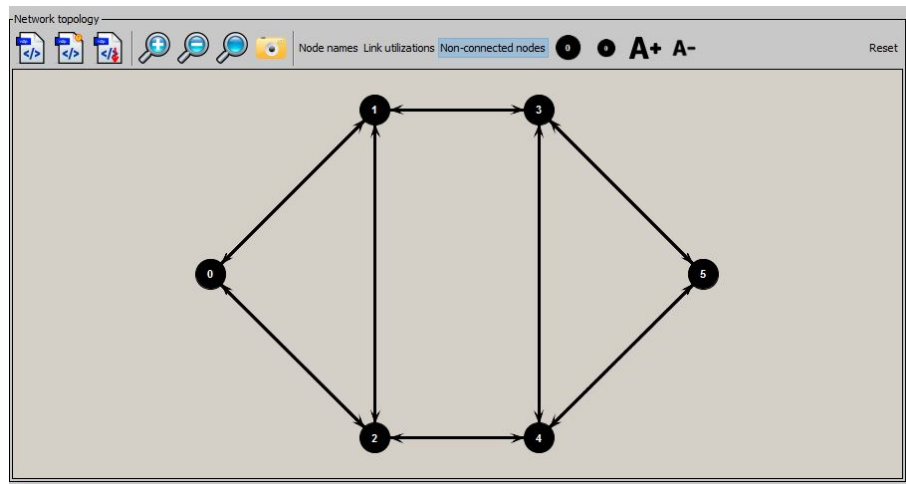


Figura 5.9: Net2Plan Network Example

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.

View/edit network state						
Algorithm execution						
View reports						
Routes	Multicast trees	Protection segments		Shared-risk groups		
Network	Layers	Nodes	Links	Demands	Multicast demands	
Unique ide...	Index	Show/Hide	Name	State	xCoord	
2	0	<input checked="" type="checkbox"/>	Node 0	<input checked="" type="checkbox"/>		
3	1	<input checked="" type="checkbox"/>	Node 1	<input checked="" type="checkbox"/>		
4	2	<input checked="" type="checkbox"/>	Node 2	<input checked="" type="checkbox"/>		
5	3	<input checked="" type="checkbox"/>	Node 3	<input checked="" type="checkbox"/>		
6	4	<input checked="" type="checkbox"/>	Node 4	<input checked="" type="checkbox"/>		
7	5	<input checked="" type="checkbox"/>	Node 5	<input checked="" type="checkbox"/>		

(a) a

View/edit network state						
Algorithm execution						
View reports						
Routes	Multicast trees	Protection segments		Shared-risk groups		
Network	Layers	Nodes	Links	Demands	Multicast demands	
Unique ide...	Index	Show/Hide	Origin node	Destinatio...	State	
8	0	<input checked="" type="checkbox"/>	0 (Node 0)	1 (Node 1)	<input checked="" type="checkbox"/>	
9	1	<input checked="" type="checkbox"/>	1 (Node 1)	0 (Node 0)	<input checked="" type="checkbox"/>	
10	2	<input checked="" type="checkbox"/>	0 (Node 0)	2 (Node 2)	<input checked="" type="checkbox"/>	
11	3	<input checked="" type="checkbox"/>	2 (Node 2)	0 (Node 0)	<input checked="" type="checkbox"/>	
12	4	<input checked="" type="checkbox"/>	1 (Node 1)	2 (Node 2)	<input checked="" type="checkbox"/>	
13	5	<input checked="" type="checkbox"/>	2 (Node 2)	1 (Node 1)	<input checked="" type="checkbox"/>	
14	6	<input checked="" type="checkbox"/>	1 (Node 1)	3 (Node 3)	<input checked="" type="checkbox"/>	
15	7	<input checked="" type="checkbox"/>	3 (Node 3)	1 (Node 1)	<input checked="" type="checkbox"/>	
16	8	<input checked="" type="checkbox"/>	2 (Node 2)	4 (Node 4)	<input checked="" type="checkbox"/>	
17	9	<input checked="" type="checkbox"/>	4 (Node 4)	2 (Node 2)	<input checked="" type="checkbox"/>	
18	10	<input checked="" type="checkbox"/>	3 (Node 3)	4 (Node 4)	<input checked="" type="checkbox"/>	
19	11	<input checked="" type="checkbox"/>	4 (Node 4)	3 (Node 3)	<input checked="" type="checkbox"/>	
20	12	<input checked="" type="checkbox"/>	3 (Node 3)	5 (Node 5)	<input checked="" type="checkbox"/>	
21	13	<input checked="" type="checkbox"/>	5 (Node 5)	3 (Node 3)	<input checked="" type="checkbox"/>	
22	14	<input checked="" type="checkbox"/>	4 (Node 4)	5 (Node 5)	<input checked="" type="checkbox"/>	
23	15	<input checked="" type="checkbox"/>	5 (Node 5)	4 (Node 4)	<input checked="" type="checkbox"/>	

(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.

Parameter	Value	Description
trafficMatrix1	C:/net2plan-0.4.2/works...	Path to ODU0 traffic matrix
trafficMatrix2	C:/net2plan-0.4.2/works...	Path to ODU1 traffic matrix
trafficMatrix3	C:/net2plan-0.4.2/works...	Path to ODU2 traffic matrix
trafficMatrix4	C:/net2plan-0.4.2/works...	Path to ODU3 traffic matrix
trafficMatrix5	C:/net2plan-0.4.2/works...	Path to ODU4 traffic matrix
trafficMatrix6	C:/net2plan-0.4.2/works...	Path to file with new dem...

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 where 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.

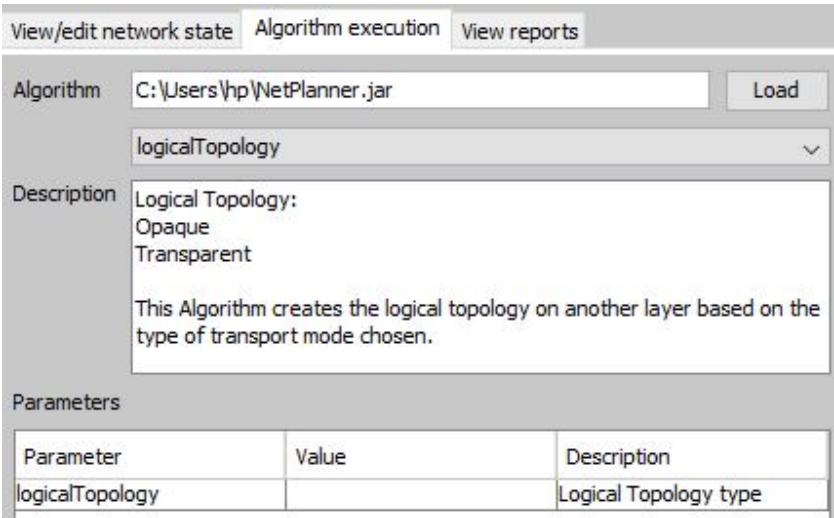


Figura 5.12: Net2Plan Logical Topology Algorithm

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.

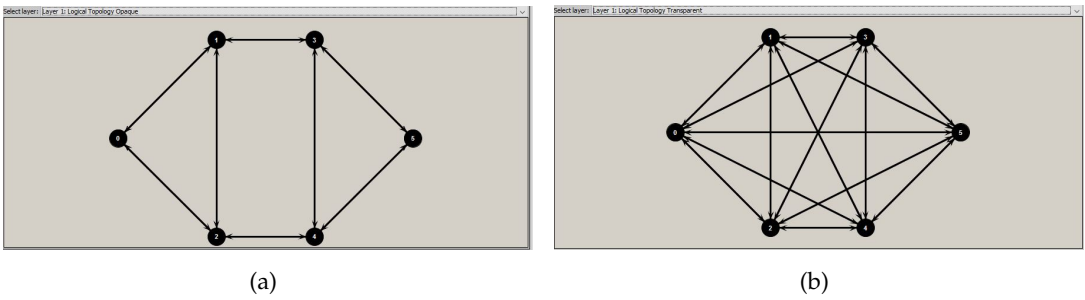


Figura 5.13: Logical Topology: a) Opaque; b) Transparent;

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 created based on 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.

Parameter	Value	Description
shortestPathType	hops	Each demand is routed ac...
wavelengthCapacity	80	ODU0 Capacity per Wave...

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.

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

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.

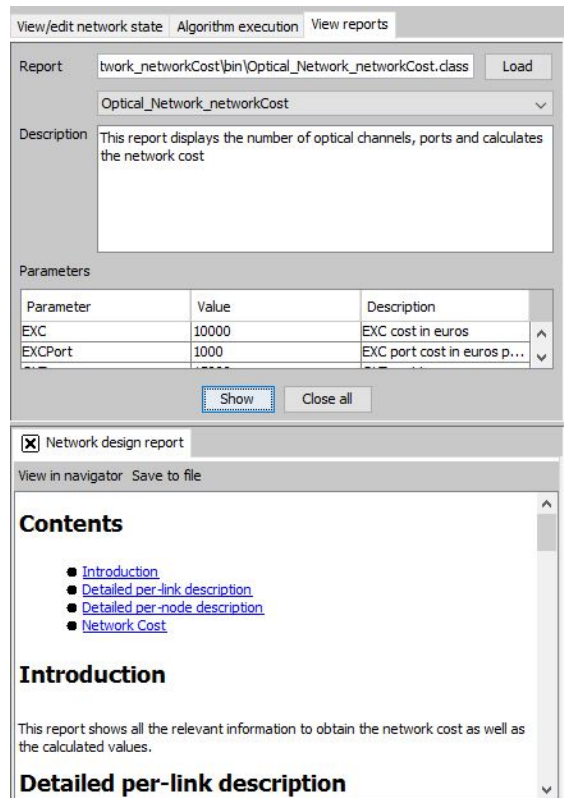


Figura 5.15: Network Cost Report

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.

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

- $C_{exc} \rightarrow$ 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.

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

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

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

- $C_{oxc} \rightarrow$ 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.

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

- $C_L \rightarrow$ 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

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

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).

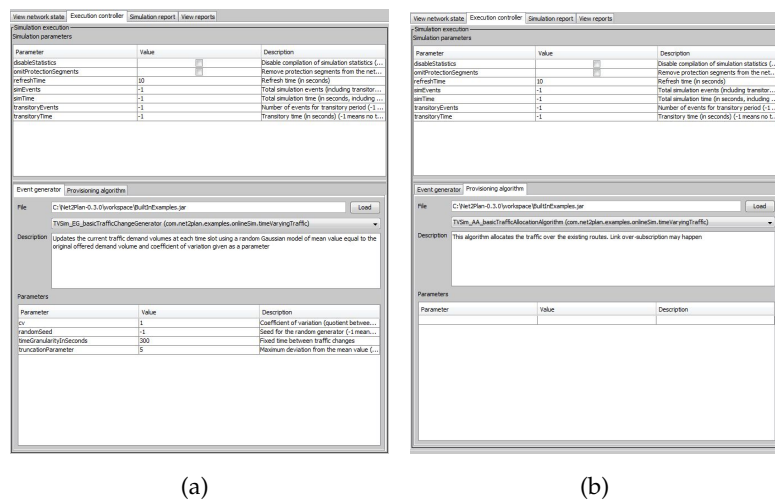


Figura 5.23: a) Net2Plan Event generator ; b) Net2Plan Provisioning algorithm

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.

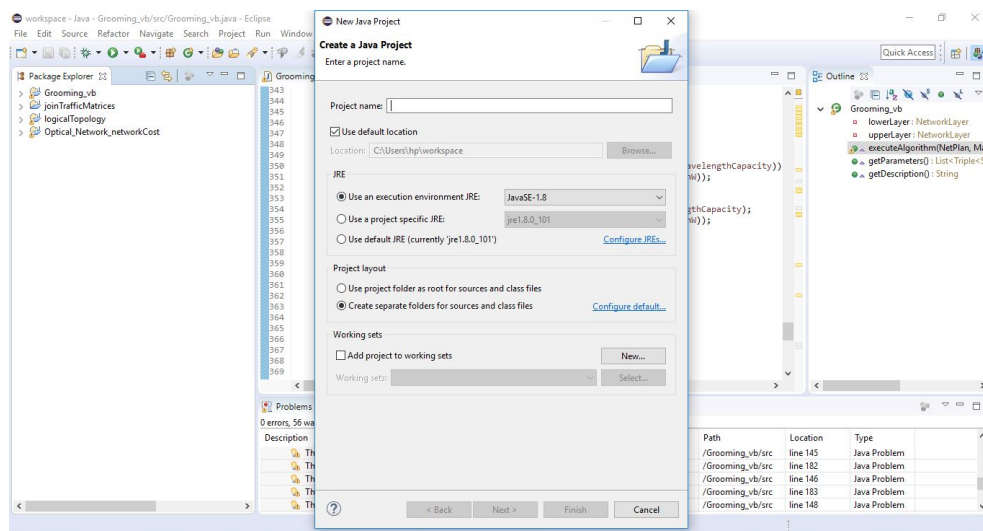


Figura 5.24: Eclipse new project

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".

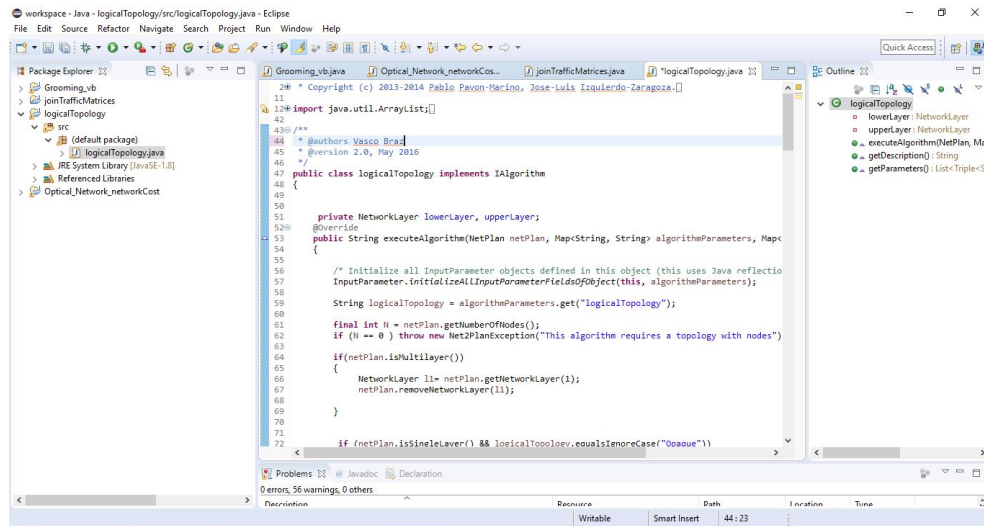
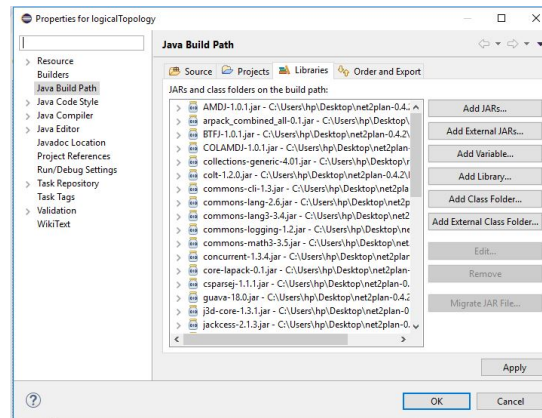


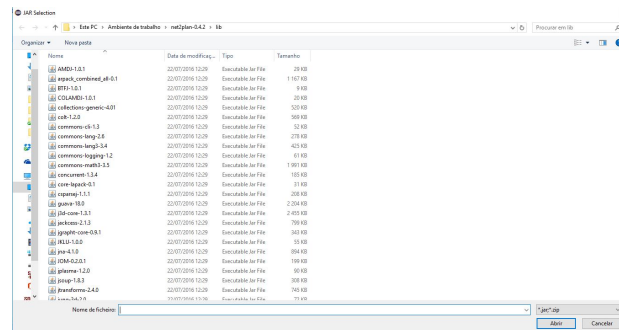
Figura 5.25: Eclipse new project with source file

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 → 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.

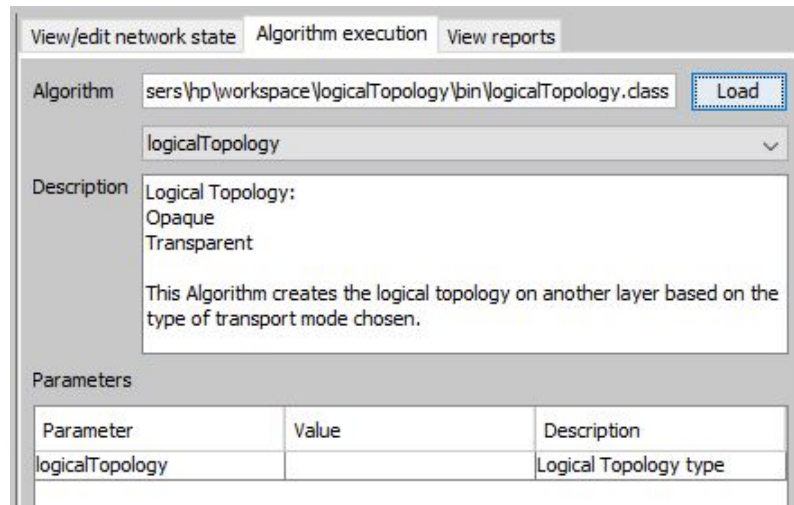


Figura 5.27: Net2Plan new algorithm

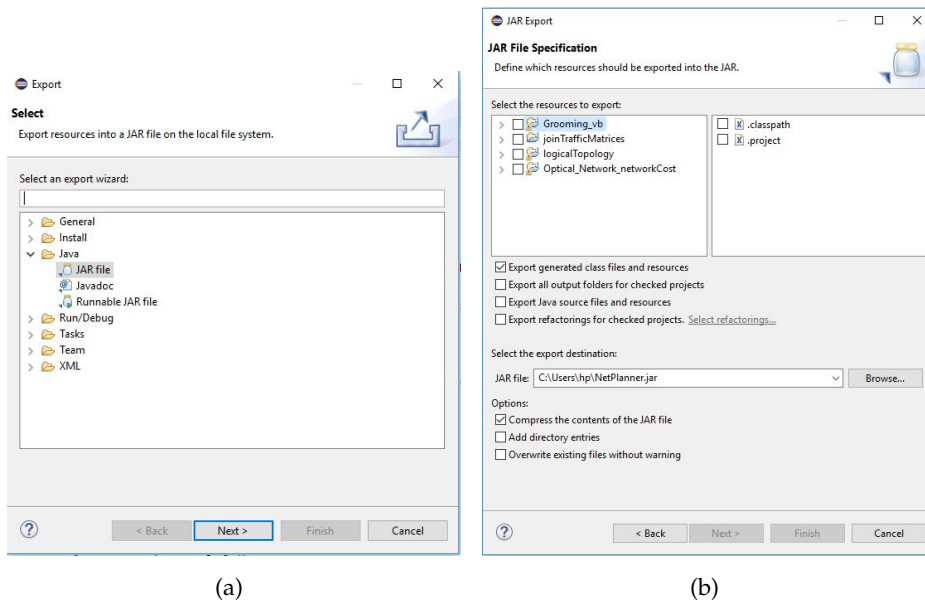


Figura 5.28: a) Eclipse export ; b) Projects to export into a .jar file

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 "Notepad++" 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.

```

1 <html>
2 <head><title>Network design report</title></head>
3 <body>
4
5 <h1>Contents</h1>
6
7 <ul>
8 <li><a href="#sectionIntroduction">Introduction</a></li>
9 <li><a href="#sectionlinkInformation">Detailed per-link description</a></li>
10 <li><a href="#sectionNodeInformation">Detailed per-node description</a></li>
11 <li><a href="#sectioncostInformation">Network Cost</a></li>
12 </ul>
13
14 <a name="sectionIntroduction"><h1>Introduction</h1></a>
15
16 <p> This report shows all the relevant information to obtain the network cost as well as the calculated values.</p>
17
18 <a name="sectionlinkInformation"><h1>Detailed per-link description</h1></a>
19
20 <center>
21 #linkInformationTable#
22 </center>
23
24 <a name="sectionNodeInformation"><h1>Detailed per-node description</h1></a>
25
26 <center>
27 #nodeInformationTable#
28 </center>
29
30 <a name="sectioncostInformation"><h1>Network Cost</h1></a>
31
32 <center>
33 #costInformationTable#
34 </center>
35
36 </body>
37

```

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.

