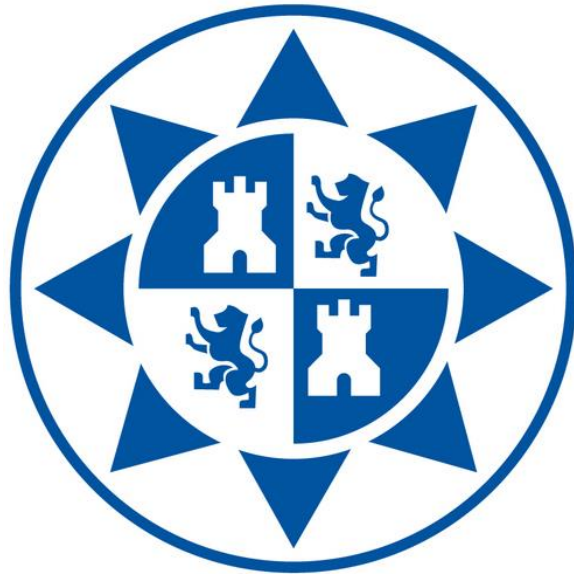


ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA DE
TELECOMUNICACIÓN

UNIVERSIDAD POLITÉCNICA DE CARTAGENA



Trabajo Fin de Máster

*HERRAMIENTA DE ESTIMACIÓN DE EFECTOS NO LINEALES
EN ENLACES DE FIBRA ÓPTICA*



AUTOR: Elena Martín Seoane

DIRECTOR: Pablo Pavón Mariño

Octubre 2018



Autor	Elena Martín Seoane
E-mail del autor	elena.mseoane@edu.upct.es
Director	Pablo Pavón Mariño
E-mail del director	pablo.pavon@upct.es
Título del TFM	Herramienta de estimación de efectos no lineales en enlaces de fibra óptica
Título en inglés	A computation tool for estimating non-linear impairments in optical fibres
Descriptorios	WDM, Gaussian Noise Model, non-linear impairments, optical systems
Resumen	<p>Las modulaciones coherentes para transmisión en fibra óptica están implicando grandes avances en los alcances y tasas. Estas modulaciones son capaces de compensar ciertos efectos con procesado en el receptor, como la dispersión cromática o la dispersión de polarización. Los efectos no lineales son el nuevo cuello de botella. Recientemente, el modelo de ruido gaussiano se ha propuesto para estimación de la degradación de la señal a ruido causada por estos efectos no lineales. Posteriormente, la herramienta GNPY desarrollada en el marco del proyecto TIP (Telecom Infra Project) por el Politecnico di Torino, ha ofrecido una implementación del mismo. Este proyecto se centra en la creación de una interfaz desde Net2Plan con la herramienta GNPY para el cálculo de las penalizaciones de la OSNR según este modelo, y en la realización de una implementación propia integrada en Net2Plan, accesible al usuario a través de informe de calidad de línea WDM.</p>
Titulación	Máster Universitario en Ingeniería de Telecomunicación
Departamento	Tecnologías de la Información y las Comunicaciones
Fecha de presentación	Octubre - 2018

ÍNDICE

1	Introducción	7
2	Tecnologías utilizadas	9
2.1	Net2Plan.....	9
2.2	GNpy.....	10
3	Desarrollo	13
3.1	Estudio de los parámetros que caracterizan la fibra óptica.....	13
3.1.1	Atributos de la fibra.....	14
3.1.1.1	Características del core o núcleo	14
3.1.1.2	Características del revestimiento.....	14
3.1.1.3	Características del modo de campo electromagnético.....	14
3.1.1.4	Área efectiva [A_{eff}]	15
3.1.1.5	Apertura numérica [NA]	15
3.1.1.6	Longitud de onda de corte	16
3.1.1.7	Pérdida por doblaje.....	16
3.1.1.8	Dispersión de velocidad de grupo	17
3.1.1.9	Dispersión cromática	17
3.1.2	Atributos de cable.....	18
3.1.2.1	Atenuación.....	18
3.1.2.2	Dispersión del modo de polarización (PMD).....	18
3.1.3	Atributos de enlace.....	19
3.1.3.1	Atenuación.....	19
3.1.3.2	Coefficiente no lineal.....	19
3.2	Estudio de la herramienta GNpy e implementación del modelo de cálculo de ruido Gaussiano en Net2Plan.....	20
3.2.1	Estudio de GNpy y su empleo en Net2Plan	20
3.2.1.1	Estructura de GNpy	21
3.2.1.2	Arquitectura del proyecto	27
3.2.1.3	Archivo Report_WDM_lineEngineering_TIP.java.....	28
3.2.2	Implementación de los cálculos del Gaussian Noise Model en el Report Line Engineering de Net2Plan.....	31
3.2.2.1	Archivo OpticalImparimentUtils.java	32

3.2.2.2	Archivo Report_WDM_lineEngineering_GNModel.java	33
4	Generación de informe y resultados.....	37
4.1	Resultados para el report basado en GNpy	37
4.2	Resultados para el report Line Engineering de Net2Plan	39
5	Manual de usuario	41
5.1	Uso del informe Net2Plan-TIP-OpticalImpairments-OLE-Report.....	41
5.1.1	Prerrequisitos.....	41
5.1.2	Instrucciones de instalación	42
5.1.3	Breve guía para la interfaz gráfica de usuario de Net2Plan	43
5.2	Uso del informe Line Engineering de Net2Plan	45
6	Conclusiones	47
7	Bibliografía.....	49

1 INTRODUCCIÓN

Una de las mejores tecnologías de transporte de información es la fibra óptica, gracias a su alta velocidad y bajas pérdidas. Sin embargo, su coste es elevado, por lo que se hacen necesarios numerosos estudios de caracterización y desempeño antes de desplegar este tipo de redes.

A lo largo de las últimas décadas se han desarrollado nuevas modulaciones de transmisión para transmitir el mayor número de señales en diferentes canales a través de una misma fibra óptica, aprovechando al máximo su ancho de banda disponible. Esto permite ahorrar costes y optimizar los equipos de transmisión y recepción, sin embargo, provoca una mayor complejidad a la hora de diseñar los equipos y el software asociado.

La multiplexación de canales se puede realizar en el dominio del tiempo o la frecuencia, que en el campo de las comunicaciones ópticas se denominan Optical Time Division Multiplexing (OTDM) y Wavelength Division Multiplexing (WDM) respectivamente. Sin embargo, la división en tiempo suele ser empleada para sistemas eléctricos en lugar de ópticos. OTDM necesita del uso de fuentes ópticas que sean capaces de emitir trenes de pulsos cortos con un ratio de 40 GHz. Además, debido a la longitud de los pulsos, la distancia de transmisión se ve muy limitada porque la dispersión en ellos es mayor y a grandes longitudes de fibra la señal se haría irrecuperable.

Debido a estas dificultades, la técnica utilizada en sistemas de transmisión ópticos es WDM, y será la que sirva de soporte a todo lo desarrollado en este trabajo.

WDM se corresponde con un sistema en el que múltiples portadoras a diferentes longitudes de onda se modulan independientemente y se transmiten por la misma fibra para después ser demultiplexadas en el receptor en sus diferentes canales individuales. La multiplexación por división de longitudes de onda permite explotar todo el potencial de ancho de banda ofrecido por la fibra, operando con cientos de canales en el mismo enlace si la distancia entre canal se reduce a menos de 100 GHz.

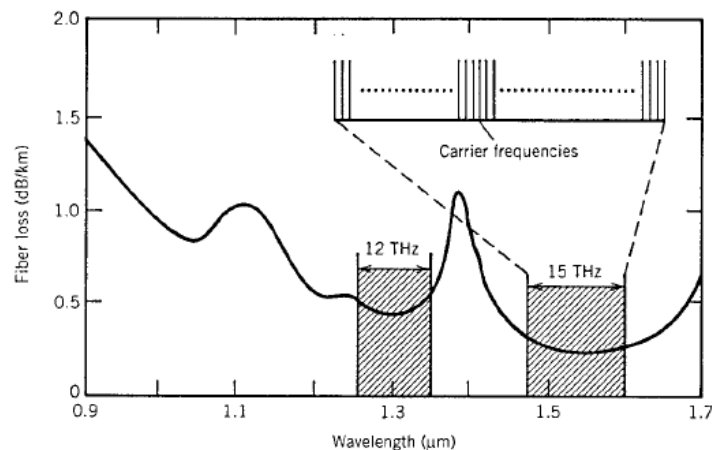


Fig. 1. Pérdidas de transmisión en fibras respecto a su longitud de onda. G. P. Agrawal, "Fiber-Optic Communication Systems" [1]

WDM permite la transmisión en dos bandas en torno a los 1300 nm y los 1550 nm. La banda situada entre ellos presenta un pico de pérdidas debido a los iones OH residuales en el silicio que hace imposible su utilización. Sin embargo, si se emplean fibras de las denominadas *low water peak* que eliminan este problema, el ancho de banda a utilizar por estos sistemas puede llegar a los 30 Tbps.

Los desarrollos en sistemas WDM y equipos ópticos hacen posible compensar la mayoría de efectos nocivos hasta convertirlos en casi despreciables. En la actualidad, los efectos no lineales son los más complejos de prever y modelar, por lo que se han convertido en el nuevo objeto de estudio de los investigadores. Recientemente se han desarrollado complejos modelos matemáticos en un esfuerzo por caracterizar las pérdidas no lineales y degradación de las señales en enlaces de fibra óptica.

GNpy es una herramienta que realiza estos cálculos desarrollada por el Politecnico di Torino y por ello será el objeto principal del estudio de este proyecto, que se centrará en la creación de una interfaz en Net2Plan para el uso de GNpy, calculando de esta forma las penalizaciones que afectarán a la OSNR.

Por otra parte, se desarrollará también una implementación de los cálculos integrada en Net2Plan a través de un report ejecutable por el usuario sobre la calidad de enlaces WDM.

Los parámetros de entrada de éste se corresponderán con la topología en estudio y las características físicas de las fibras ópticas. El objetivo final será la obtención de unas tablas resumen con los resultados de desempeño y determinar si se adecuarán al funcionamiento de los equipos de la red.

Las fases previstas para este trabajo son las que se enumeran a continuación:

- 1) Estudio previo de los parámetros ópticos de la fibra óptica.
- 2) Estudio de la herramienta GNpy e implementación del modelo de cálculo Gaussian-Noise model en Net2Plan.
- 3) Generación del informe y validación de los resultados.

2 TECNOLOGÍAS UTILIZADAS

En primer lugar se detallarán las tecnologías empleadas durante el desarrollo de este proyecto, así como los motivos de su elección y su función en el mismo.

2.1 NET2PLAN

Net2Plan es una herramienta gratuita y de software libre de diseño y planificación de redes desarrollada por el Grupo de Investigación de Redes de Telecomunicación de la Universidad Politécnica de Cartagena. Desde su primera versión en 2011 ha incorporado nuevas funciones como la posibilidad de estimación de tráfico y desempeño de redes de fibra óptica en capa física y WDM, que será clave en el desarrollo de este proyecto.

Proporciona una interfaz gráfica de usuario para crear topologías y puede ser empleada para la planificación, optimización y evaluación de redes de comunicaciones, además de permitir al usuario desarrollar sus propios algoritmos e informes de resultados, ejecutándolos sobre las redes simuladas.

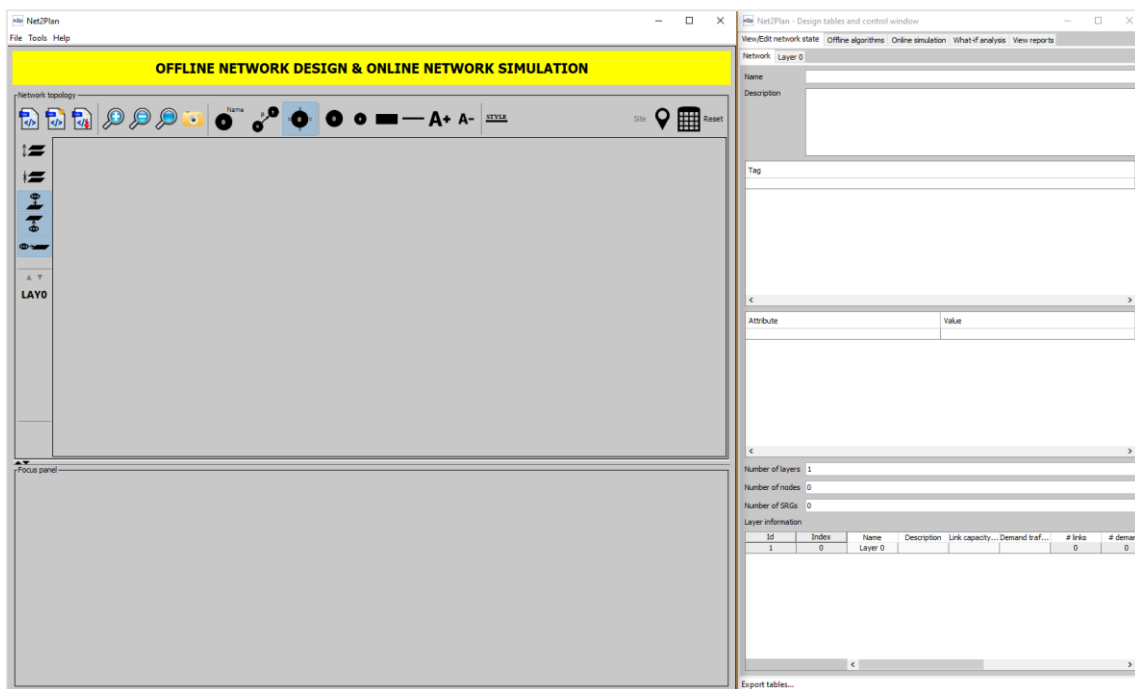


Fig. 2. Herramienta de diseño de redes Net2Plan.

Net2Plan permite el diseño de redes de comunicaciones ópticas como las necesarias para el testeo de los cálculos de efectos no lineales que serán estudiados a lo largo de este proyecto, por lo que es la herramienta ideal en la que desarrollar el trabajo.

Entre los ejemplos que proporciona Net2Plan se puede encontrar un informe para redes WDM que calcula y evalúa el desempeño de la red, proporcionando parámetros como la relación señal a ruido o la dispersión cromática en los extremos de los enlaces. Este será el informe que se modificará durante la última parte del proyecto.

Además, ha sido escogida también por las librerías de código que ofrece y el conocimiento previo de la misma, dado que ha sido utilizada en numerosas ocasiones durante el Grado en Ingeniería Telemática y el Máster en Ingeniería de Telecomunicaciones cursados en la UPCT.

2.2 GNPY

Investigadores del Politécnico di Torino han descrito un modelo de ruido Gaussiano para ajustar las pérdidas y variaciones no lineales que se producen en la fibra y que hasta el momento no tenían una forma apropiada de prever ni, por tanto, evitar.

Para desarrollar una herramienta que implemente estos cálculos, se unieron al Telecom InfraProject (TIP), una comunidad colaborativa de investigación en telecomunicaciones. Comenzó en 2016 con la intención de acelerar la innovación tecnológica y ampliar los conocimientos de la academia además de llevarlos a un terreno práctico que la industria sea capaz de aprovechar.

Consta de una junta liderada por representantes de las mayores empresas del sector de las telecomunicaciones, además de comités técnicos de expertos que ayudan a dirigir el foco de las investigaciones llevadas a cabo en todo el proyecto. Además de estos dos grupos, el TIP permite involucrarse a más de 500 organizaciones de miembros a diferentes niveles, incluyendo tanto operadoras, startups, desarrolladores y otras entidades que puedan aportar algo a cualquiera de sus trabajos, reuniéndose en una convención anual en la que todos presentan sus avances, el TIP Summit.

Como parte de su estructura organizativa y para desarrollar tantos proyectos simultáneos como resulte posible, se dividen en áreas de desarrollo y grupos de trabajo. Sus grupos de proyectos se dividen en las siguientes categorías, dándole una estructura de red:

- **Grupos de capa de acceso.** Trabajan en mejorar soluciones de infraestructura de red, tratando de eliminar algunos de los obstáculos que dificultan las conexiones con el usuario. Entre ellos se incluyen los grupos de Edge Computing, Power and Connectivity y System Integration and Site Optimization, además de varios dedicados a las RAN (Radio Access Network).
- **Grupos de capa de distribución.** Se centran en la escalabilidad, convergencia y configuración de la capa de distribución, tanto cableada como wireless, dado el desarrollo exponencial que está teniendo el tráfico

de red. Entre ellos se encuentran los grupos mmWave Networks y el Open Optical & Packet Transport.

- **Grupos de la capa core y de management.** Tratan de simplificar la arquitectura del core de una red para disminuir los costes de despliegue, así como mejorar la eficiencia y flexibilidad de la misma. Incluyen a los grupos de Artificial Intelligence & Applied Machine Learning, End to End Network Slicing (E2E-NS) y People & Process.

El proyecto GNpy se encuadra dentro del Open Optical & Packet Transport (OOPT) mencionado anteriormente. Éste es un grupo que trabaja en la definición de tecnologías y dispositivos *abiertas*, es decir, sin licenciar, para las capas IP y óptica de una red. Está orientado a la ingeniería de red, desarrollando tanto dispositivos de acceso IP, transponders ópticos, herramientas de simulación y planificación de redes y librerías de código abierto para desarrollar proyectos propios.

Los aspectos relevantes a la realización de este Trabajo se concentran en la capa física y las redes ópticas. El grupo define una arquitectura de transporte basada en Wavelength Division Multiplexing, la *Dense-WDM*. Los sistemas DWDM incluyen sistemas de control y de línea abiertos, manejo y administración de transponders y tecnologías de encaminamiento de paquetes.

El OOPT se subdivide en cinco grupos diferentes, cada uno orientado a un aspecto diferente de los mencionados anteriormente.

- *Disaggregated Cell Site Gateways (DCSG)*
- *Physical Simulation Environment (PSE)*
- *Open Line Systems (OLS)*
- *Common API (C-API)*
- *Disaggregated Transponder Chips (DTC)*

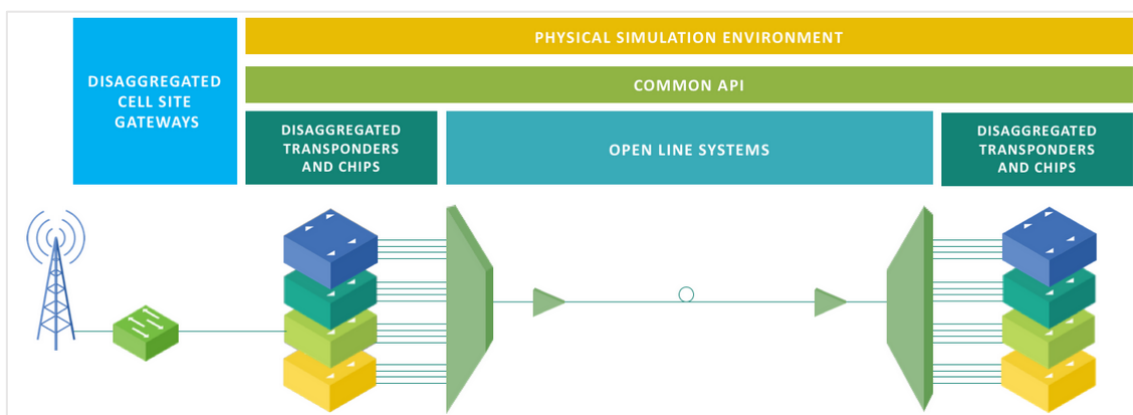


Fig. 3. Esquema de la organización por grupos en el OOPT group. [2]

En el marco del Telecom InfraProject, el proyecto Open Optical Packet Transport (OOPT) del grupo Physical Layer Simulation Environment (PSE), que involucra a contribuyentes tanto del Politecnico di Torino como diversas empresas dedicadas al despliegue de redes y fabricación de equipos como son Orange, Facebook, Microsoft, Cisco o Juniper Networks, se ha desarrollado una librería para evaluar el desempeño de la capa física de una red de comunicaciones ópticas, empleando un modelo de ruido Gaussiano.

Este proyecto representa la intención de la comunidad científica y empresarial de aunar esfuerzos en el camino hacia una optimización completa en redes multicapa.

Su objetivo principal es obtener un entorno de simulación que permita definir modelos de los dispositivos ópticos de una red, sus funciones, parámetros e interconexiones, independientemente del fabricante de cada uno. El porcentaje de exactitud variará con la complejidad de la red, por lo que su intención es validar la herramienta con medidas obtenidas de redes reales.

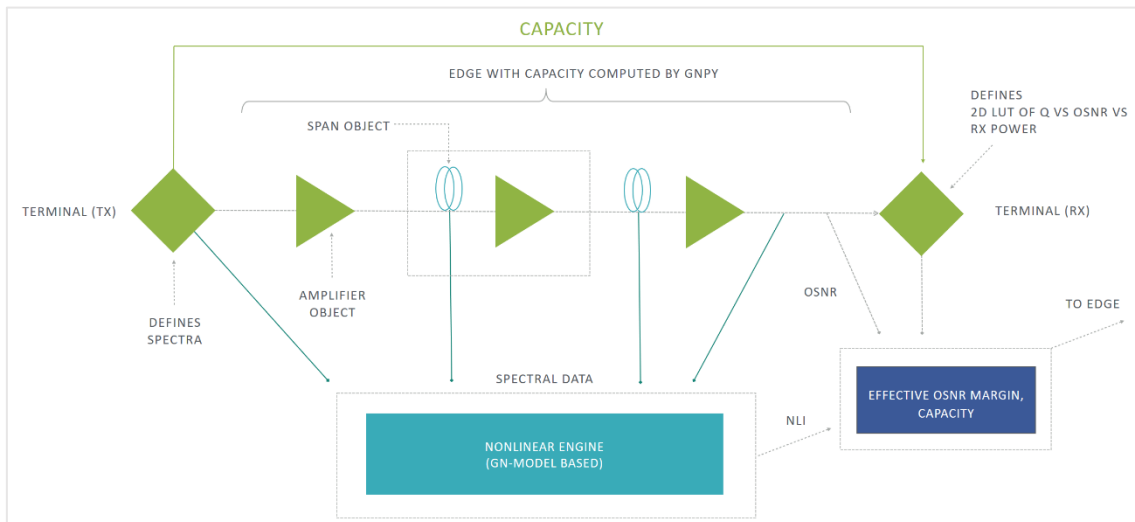


Fig. 4. GNpy link and calculations description representation [2]

Esta librería de código abierto permitirá representar enlaces como los descritos en la Fig. 4 y obtener los cálculos relativos, no sólo a las pérdidas no-lineales del modelo de ruido Gaussiano desarrollado por el Politecnico di Torino, sino los datos para la OSNR lineal y no lineal de salida al final del enlace, permitiendo determinar si se cumplirán o no los parámetros de calidad requeridos para la correcta recuperación e interpretación de la señal óptica.

3 DESARROLLO

3.1 ESTUDIO DE LOS PARÁMETROS QUE CARACTERIZAN LA FIBRA ÓPTICA

El objetivo principal de cualquier canal de comunicaciones es transportar una señal desde el transmisor hasta el receptor con las menores pérdidas posibles. Por ello las fibras ópticas son los sistemas de transmisión más empleados en la actualidad dadas las bajas pérdidas que producen estos materiales. Sin embargo, no sólo las pérdidas en potencia son importantes en las transmisiones, las dispersiones que se producen en la señal debidas a la propagación son uno de los graves problemas de diseño, pues pueden provocar la total deformación de la señal y convertirla en irrecuperable por el receptor.

Las fibras ópticas transmiten de manera habitual múltiples canales simultáneamente; si bien hay diversas maneras de implementar esto, la más empleada en la actualidad y en la que se apoya este trabajo es la basada en división por longitud de onda, *Wavelength Division Multiplexing* (WDM), en la que el ancho de banda que proporciona la fibra se divide en canales, cada uno operando a una frecuencia diferente. La señal óptica a una frecuencia determinada se verá afectada por las potencias de otras longitudes de onda que se emitan en la misma fibra, especialmente porque el campo está encerrado en el núcleo durante largas distancias. La potencia en un determinado canal produce efectos nocivos sobre los canales adyacentes, perturbando la linealidad de la potencia de transmisión y produciendo los denominados *efectos no-lineales*, que son el objeto principal de estudio en este proyecto.

Se hace imprescindible, por tanto, un estudio sobre los parámetros que caracterizan las fibras ópticas de forma que sea posible comprender y prever los resultados de una transmisión. La Unión Internacional de las Telecomunicaciones los clasifica en los siguientes grupos.

- **Atributos de la fibra** son aquellos independientes de los procesos de cableado e instalación. Es decir, aquellos determinados por las características físicas del diseño del propio cable de fibra y debidas al proceso de fabricación.
- **Atributos de cable** son aquellos determinados por el proceso de cableado. Entre ellos se encuentran la atenuación y el coeficiente de PMD.
- **Atributos de enlace** son característicos de las concatenaciones de varios spans de fibra y están afectados por los conectores, la forma de instalación y los empalmes.

3.1.1 Atributos de la fibra

Entre los atributos debidos al diseño, material y fabricación de una fibra encontramos las siguientes características.

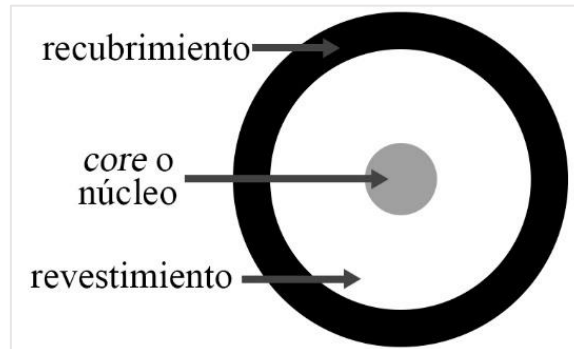


Fig. 5. Esquemático de la sección transversal de la fibra.

3.1.1.1 Características del core o núcleo

Especifican el diámetro (medido en micrómetros) y su no-circularidad (medida en porcentaje respecto a un círculo perfecto), además de describir el centro del círculo en el que la intensidad de campo próximo es constante, el del formado por el círculo de revestimiento y el error entre ambos.

3.1.1.2 Características del revestimiento

Es la zona inmediatamente exterior al núcleo y se caracteriza por parámetros similares a los del núcleo tales como su diámetro, centro, y su no-circularidad en porcentaje respecto a un círculo perfecto. Se considera centro del revestimiento el punto central de la circunferencia exterior definida por el revestimiento; y, el diámetro, la distancia entre ambos expresada en micrómetros.

Se define también como extensión de las características del núcleo y del revestimiento el parámetro de error de concentricidad, que representa la distancia entre el centro del núcleo y el del revestimiento, en micrómetros.

3.1.1.3 Características del modo de campo electromagnético

Esta característica se considera únicamente en fibras monomodo y se refiere a la intensidad de la distribución del campo electromagnético en la sección transversal de las

mismas. Este parámetro se denomina diámetro del modo de campo (MFD), se mide en micrómetros y es función de la longitud de onda, el diámetro del núcleo y de la diferencia entre los índices de refracción del núcleo y el revestimiento.

El MFD es importante para determinar la resistencia a las pérdidas por ser doblado e influye, aunque de manera menos significativa, en las pérdidas producidas por el empalme de dos tramos de fibra.

3.1.1.4 Área efectiva [A_{eff}]

La densidad de potencia en los núcleos de las fibras viene dada por la potencia de la señal dividida por el área en que se distribuye, pero esta área puede variar.

La potencia se concentra en el centro de la fibra y se va disipando conforme se acerca al revestimiento, especialmente en el caso de las fibras monomodo hasta llegar a anularse por completo. Es por ello que se introduce el concepto de *área efectiva* (μm^2), siendo ésta la que realmente debe aplicarse en los modelos de cálculo puesto que es donde se concentra la potencia óptica.

$$A_{eff} = \frac{2\pi \left(\int_0^\infty |E_a(r)|^2 r dr \right)^2}{\int_0^\infty |E_a(r)|^4 r dr} = \frac{2\pi \left(\int_0^\infty I(r) r dr \right)^2}{\int_0^\infty I^2(r) r dr}$$

En la definición de A_{eff} , medida en micrómetros cuadrados, $E_a(r)$ representa la amplitud e $I(r)$ la intensidad de campo próximo del modo fundamental con un radio r en el eje de la fibra.

Los efectos no-lineales dependen de la intensidad del campo electromagnético en el centro de la fibra, pero en la práctica normalmente nos encontramos con que las mediciones realizadas son para el conjunto total de potencia a la entrada y salida de la fibra, impidiendo comparar los modelos teóricos con los datos reales.

3.1.1.5 Apertura numérica [NA]

Se define como el seno del mayor ángulo de incidencia que puede tener un rayo para que se produzca la reflexión total en el núcleo y puede determinarse midiendo el ángulo de divergencia del cono de luz emitido cuando todos los modos de la fibra están excitados.

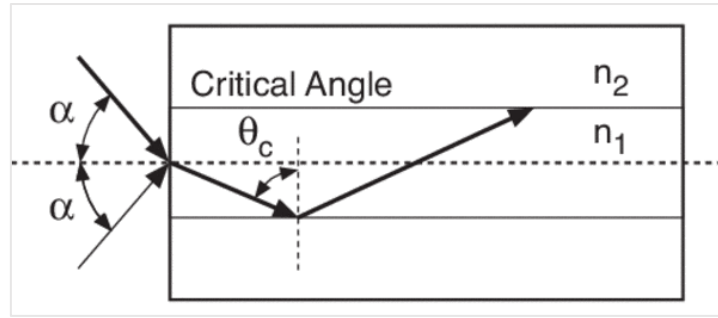


Fig. 6. Diagrama representativo de la apertura numérica [3].

Para calcularla se toma como valor el máximo teórico de apertura numérica, que se deriva de los índices de refracción del núcleo y del revestimiento, y su valor efectivo suele ser un 5% menor a este máximo teórico.

$$NA_{tmax} = \sin \alpha = \sqrt{(n_{core}^2 - n_{clad}^2)}$$

3.1.1.6 Longitud de onda de corte

La longitud de onda de corte, medida en nanómetros, es aquella que marca el límite en el que una fibra se comporta como monomodo. A longitudes de onda inferiores pueden pasar diferentes modos de propagación, mientras que a longitudes superiores a la de corte únicamente el modo fundamental puede propagarse.

Este parámetro es función de diversos factores como la longitud, las condiciones en que se encuentra el cable, es decir, si ha sido doblado, tensado, etc., por lo que se distingue la longitud de onda de corte en la fibra como aquella medida en una determinada configuración: longitud de 2 metros, un bucle de 14 cm de radio sujeto sin tensión y el resto del cable recto. En estas condiciones se puede calcular con la siguiente fórmula:

$$\lambda_c = 2\pi \frac{r_{core}}{V_c} NA_{tmax} = 2\pi \frac{r_{core}}{V_c} \sqrt{(n_{core}^2 - n_{clad}^2)}$$

Donde r_{core} es el radio del núcleo, V_c es el parámetro de frecuencia normalizada de la fibra con un valor de 2.405 y n_{core} y n_{clad} son los índices de refracción del núcleo y del revestimiento respectivamente.

3.1.1.7 Pérdida por doblaje

Se producen por los efectos de doblar la fibra y varían también en función de la longitud de onda y el radio y número de vueltas con las que se doble. Suelen medirse en dB por vuelta (dB/turn).

3.1.1.8 Dispersión de velocidad de grupo

Se define el retardo de grupo $\tau(t)$ como el tiempo requerido para que un pulso completo de luz atraviese una unidad de longitud de fibra y depende de la longitud de onda.

Se le suele denominar GVD (*Group Velocity Dispersion*) o parámetro β_2 y determina cuánto se deformará a lo ancho un pulso al propagarse por la fibra. Este parámetro está muy relacionado con la dispersión cromática y la dispersión del modo de polarización.

3.1.1.9 Dispersión cromática

El índice de refracción varía con la longitud de onda, por tanto, diferentes longitudes de onda se propagan a diferentes velocidades por la fibra; esto es lo que se conoce como dispersión cromática. Está causada por las diferencias en los retardos entre el grupo de velocidades de las diferentes longitudes de onda emitidas y, como consecuencia, produce un ensanchamiento de los impulsos transmitidos.

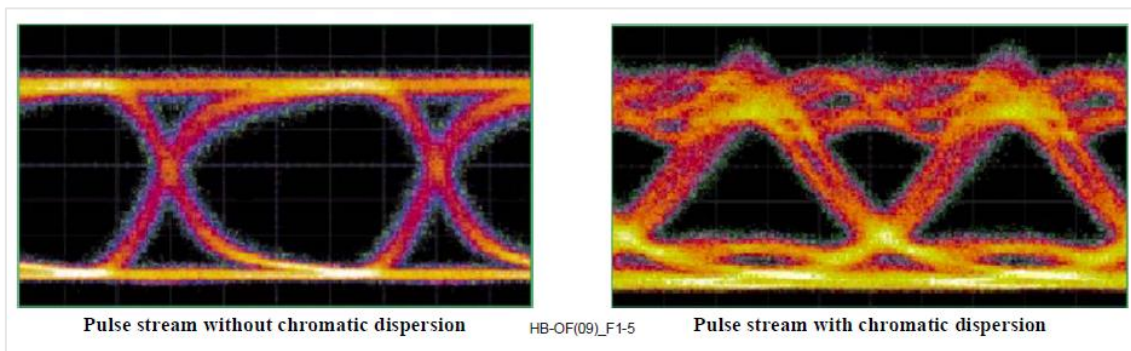


Fig. 7. "The effect of the chromatic dispersion: pulse broadening and eye closing" [4]

El coeficiente de dispersión cromática se expresa en ps/(nm*km) y representa el retardo diferencial de una emisión de un ancho de 1 nm que atraviesa 1 km de fibra. Se forma con la contribución de la dispersión debida al índice de refracción del material, que es la más importante excepto en la banda de 1300 nm, y la producida por la guía-onda.

$$D(\lambda) = \frac{d\tau}{d\lambda} = -\frac{2\pi c}{\lambda^2} \beta_2$$

A él está relacionado el concepto de longitud de onda de dispersión cero, siendo aquel punto de frecuencia en el cual la dispersión cromática es nula.

Generalmente en las características de una fibra se especifican las dispersiones cromáticas máximas y mínimas para unas longitudes de onda máximas y mínimas determinadas.

3.1.2 Atributos de cable

Estos atributos se encuentran determinados más directamente por el proceso de cableado, es decir, dependen de la disposición de las fibras.

3.1.2.1 Atenuación

La propagación de la luz en una fibra decrece exponencialmente con la longitud debido a las pérdidas de absorción y de *scattering*. La atenuación define la degradación de potencia de salida de una fibra por unidad de longitud y suele expresarse en dB/km.

$$A(\lambda) = 10 \log \frac{P_{out}}{P_{in}} \text{ (dB)}$$

$$\alpha(\lambda) = \frac{A(\lambda)}{L} \text{ (dB/unidad de longitud)}$$

Siendo P_{out} y P_{in} las potencias de salida y entrada respectivamente.

La atenuación de potencia en una fibra depende directamente de la longitud de onda a la que se esté transmitiendo. En las fibras ópticas convencionales, se puede apreciar claramente un pico de atenuación en torno a los 1390 nm, sin embargo, se han desarrollado unas fibras denominadas *low water peak* que evitan este fenómeno y tienen una atenuación menor a las convencionales, permitiendo usar un rango de longitudes de onda mucho más amplio, teniendo su mínimo de atenuación en la región de los 1550 nm.

3.1.2.2 Dispersión del modo de polarización (PMD)

La dispersión del modo de polarización (ps/km^{1/2}) es debida principalmente a la dispersión de los materiales y el modo de polarización, estando íntimamente relacionada con el retardo de diferencial de grupo (DGD), aquel entre dos modos de polarización ortogonales.

En las fibras monomodo solo se admiten dos polarizaciones ortogonales que deberían ser idénticas, pero debido a las variaciones en la circularidad y concentricidad del núcleo y el revestimiento, una viaja más rápida que la otra, causando la dispersión del pulso.

Si bien el DGD varía de manera aleatoria con el tiempo y la longitud de onda, la PMD presenta un comportamiento estadístico medio que hace posible su compensación.

El coeficiente de dispersión del modo de polarización es el valor normalizado con la longitud de la fibra, medido en picosegundos.

3.1.3 Atributos de enlace

Por último, encontramos los atributos de enlace, que incluyen aquellos que se ven afectados por las características del enlace, siendo una suma de los atributos de las secciones 3.1.1 y 3.1.2 junto con otros factores como la fusión de tramos de fibra o el número y longitud de spans que forman el enlace. Entre ellos se incluyen algunos ya mencionados como la dispersión cromática, el retardo diferencial de grupo y los detallados a continuación.

3.1.3.1 Atenuación

En la sección 3.1.2.1 se ha desarrollado cómo se atenúa la señal dentro de un tramo de fibra, sin embargo, un enlace raramente está compuesto de un único tramo, por lo que la atenuación a nivel de enlace viene dada por la siguiente expresión:

$$A = \alpha L + \alpha_s x + \alpha_c y$$

Donde α es la atenuación referenciada en el apartado 3.1.2.1, L es la longitud del enlace, α_s son las pérdidas por empalme, con x el número de empalmes de tramos de fibra en el enlace y α_c las pérdidas por los conectores, siendo y el número de conectores totales del enlace.

3.1.3.2 Coeficiente no lineal

Es el que modula las pérdidas y efectos no-lineales en las fibras ópticas y está íntimamente relacionado con el área efectiva, puesto que ésta influye en gran manera en estos productos indeseados.

$$n_2/A_{eff}$$

Los efectos no lineales comienzan a aparecer conforme se incrementa la potencia de transmisión por la fibra óptica para alcanzar mayores distancias y grandes anchos de banda. De esta forma los efectos no-lineales han de ser considerados en redes con grandes distancias sin regeneración y en sistemas de alta capacidad que introduzcan grandes potencias de transmisión y bit rates elevados.

Dada la tendencia de los efectos no-lineales a aparecer cuando se transmiten cantidades elevadas de potencia, intuitivamente podría decidirse reducir las potencias de transmisión y así evitarlos. Sin embargo, esto provocaría una degradación de la OSNR, que establece el límite inferior de la potencia de transmisión. Además, debe tenerse en cuenta que reducir la potencia conllevaría un incremento del número de amplificadores y regeneradores necesarios para atravesar las mismas distancias, y ellos inducen sus propias pérdidas en la señal, además de incrementar el coste del diseño.

Estos efectos se pueden catalogar normalmente en los siguientes grupos.

- Efectos de scattering. Comprende tanto el scattering Brillouin como el scattering de Raman estimulado.
- Efectos relacionados con el efecto Kerr, es decir, aquellos que se relacionan con los índices de refracción de los materiales del núcleo y revestimiento y sus dependencias. Entre ellos se encuentran la self-phase modulation de los sistemas monocanal o la cross-phase modulation y four wave mixing en los sistemas multicanales.

3.2 ESTUDIO DE LA HERRAMIENTA GNPy E IMPLEMENTACIÓN DEL MODELO DE CÁLCULO DE RUIDO GAUSSIANO EN NET2PLAN

La herramienta de cálculos GNpy desarrolla el modelo de ruido Gaussiano en una aplicación escrita en Python. Para este trabajo primeramente nos centraremos en comprender esta herramienta y utilizarla desde Net2Plan para después incorporar la forma de realizar los cálculos en un report propio de Net2Plan.

3.2.1 Estudio de GNpy y su empleo en Net2Plan

Para poder utilizar GNpy primero se debe comprender su funcionamiento y cómo toma los datos de entrada, así como saber analizar los datos de salida.

GNpy es una herramienta que, bajo las asunciones de que los enlaces de fibra están “descompensados”, es decir, no introducen compensaciones de dispersión cromática y los enlaces emplean WDM y sistemas coherentes, deduce que el desempeño de la red se verá limitado por el ruido de amplified spontaneous emissions (ASE) generado por los amplificadores y por los efectos no lineales, que incluyen diversos factores modelados con un sistema de ruido Gaussiano.

$$\text{SNR} = \frac{L_F P_{ch}}{P_{ASE} + P_{NLI}} = L_F \left(\frac{1}{\text{SNR}_{LIN}} + \frac{1}{\text{SNR}_{NL}} \right)^{-1}$$

Donde P_{ch} es la potencia del canal, P_{ASE} y P_{NLI} las potencias de ruido ASE y no-lineal respectivamente y L_F un parámetro de penalty que simboliza las pérdidas debidas a los filtros de la red óptica.

3.2.1.1 Estructura de GNpy

La herramienta GNpy consta de varias carpetas y ficheros que se interconectan entre sí con la siguiente estructura.

Nombre	Fecha de modifica...	Tipo	Tamaño
__pycache__	19/10/2017 11:45	Carpeta de archivos	
configuration	27/06/2018 20:15	Carpeta de archivos	
input	27/06/2018 20:15	Carpeta de archivos	
output	04/08/2017 9:03	Carpeta de archivos	
__init__	04/08/2017 9:03	Archivo PY	1 KB
__main__	04/08/2017 9:03	Archivo PY	3 KB
__main_ole__	04/08/2017 9:03	Archivo PY	3 KB
cli	04/08/2017 9:03	Archivo PY	1 KB
gnpy	19/10/2017 10:24	Archivo PY	42 KB

Fig. 8. Estructura principal de ficheros de GNpy.

Por una parte, en la carpeta principal tenemos los archivos encargados de realizar los cálculos del programa. “__main_ole__” es el encargado de iniciar la ejecución y llamar a “gnpy”, recoger los datos de salida e imprimir las gráficas del ruido no lineal que se generan.

Como se observa en la Fig. 8, además de archivos sueltos existen también tres carpetas que le dan estructura a los ficheros: “configuration”, “input” y “output”.

Los scripts “__init__”, “__main__” y “cli” no serán utilizados y sirven simplemente como ejemplos y para iniciar los cálculos desde consola.

3.2.1.1.1 Directorio “configuration”

En él encontramos los parámetros correspondientes a la configuración y descripción de la red, divididos en tres ficheros.

Nombre	Fecha de modifica...	Tipo	Tamaño
__pycache__	02/11/2017 13:05	Carpeta de archivos	
__init__	04/08/2017 9:03	Archivo PY	1 KB
fiber_parameters	26/06/2018 9:41	Archivo PY	1 KB
general_parameters	27/06/2018 13:01	Archivo PY	1 KB
link_description	27/06/2018 12:48	Archivo PY	5 KB

Fig. 9. Estructura de la carpeta "configuration" de GNpy.

FIBER_PARAMETERS.PY

Contiene un diccionario Python con los posibles tipos de fibras a encontrar en la red y sus parámetros asociados.

Un diccionario Python tiene una sintaxis muy similar a lo que podrían ser objetos en JSON.

```

1      # coding=utf-8
2      fibers = {
3          'SMF': {
4              'reference_frequency': 193.5,
5              'alpha': 0.2,
6              'alpha_1st': 0.0,
7              'beta_2': 21.27,
8              'n_2': 2.5E-20,
9              'a_eff': 77.77
10         },
11         'NZDF': {
12             'reference_frequency': 193.5,
13             'alpha': 0.22,
14             'alpha_1st': 0.0,
15             'beta_2': 21.0,
16             'n_2': 2.5E-20,
17             'a_eff': 70.0
18         }
19     }

```

Fig. 10. Ejemplo de fichero "fiber_parameters.py".

Entre los parámetros a describir de cada fibra encontramos su frecuencia de referencia, es decir, la banda a la que trabaja (en este caso, la C); su atenuación (α) y pendiente de atenuación (α_{1st}); y los parámetros necesarios para los cálculos de las pérdidas no-lineales: su área efectiva (A_{eff}), su coeficiente no-lineal (n_2) y la dispersión de velocidad de grupo (β_2).

LINK_DESCRIPTION.PY

Contiene una representación de la descripción del enlace. Se detalla una variable "link" en la que se incluyen grupos descriptores de cada objeto.

Cada uno de los objetos debe contener los parámetros “comp_cat”, que describe el tipo del componente; “comp_id”, un identificador único; y los atributos propios de cada tipo de objeto.

- En el caso de que sea un span de fibra, deberá contener el “fiber_type”, siendo éste siempre uno de los descritos en el fichero “fiber_parameters.py” y la longitud del span en cuestión.
- En el caso de un amplificador, los atributos añadidos serán la frecuencia de operación, la ganancia a la que trabaja y su variación, así como su figura de ruido.

```

1  # coding=utf-8
2  link = [
3  {
4      'comp_cat': 'fiber',
5      'comp_id': '1',
6      'fiber_type': 'SMF',
7      'length': 100.0
8  },
9  {
10     'comp_cat': 'OA',
11     'comp_id': '2',
12     'ref_freq': 193.5,
13     'gain': 20.0,
14     'gain_tlt': 0.0,
15     'noise_figure': 5.0
16  },
17  {
18     'comp_cat': 'fiber',
19     'comp_id': '3',
20     'fiber_type': 'SMF',
21     'length': 100.0
22  }
23  ]

```

Fig. 11. Ejemplo de archivo "link_description.py".

GENERAL_PARAMETERS.PY

Este archivo contiene parámetros que describen el sistema de comunicación en un objeto denominado “sys_param”, con los atributos de frecuencia de inicio de la grid de WDM y número de slots de 6.25 GHz disponibles en ella.

Además, contiene otro objeto llamado “control_param” que incluye atributos para la ejecución de GNpy.

- ‘save_each_comp’: si se desea que se guarden los datos del espectro para la salida de cada componente.
- ‘is_linear’: calcular sólo los datos de la OSNR lineal.

- ‘is_analytic’: si se debe usar la fórmula analítica o no. La fórmula analítica es ligeramente menos precisa pero mucho más rápida de calcular.
- ‘kind_interp’: el tipo de interpolación que se utilizará para la doble integral si no se emplea el método analítico.
- ‘points_not_interp’: si se emplea la doble integral, indica el número de puntos que serán interpolados.
- ‘th_fwm’: el umbral de eficiencia para los cálculos no analíticos.
- ‘n_points’ y ‘n_points_min’: número de puntos en los que será evaluada la doble integral en las regiones de alta y baja eficiencia respectivamente.
- ‘n_cores’: número de cores a emplear para los cálculos. Esta opción continúa en desarrollo y por el momento solo trabaja con uno.

```

1  # coding=utf-8
2  sys_param = {
3      'f0': 191.55,
4      'ns': 704
5  }
6  control_param = {
7      'save_each_comp': True,
8      'is_linear': False,
9      'is_analytic': True,
10     'points_not_interp': 2,
11     'kind_interp': 'linear',
12     'th_fwm': 50.0,
13     'n_points': 500,
14     'n_points_min': 4,
15     'n_cores': 1
16 }

```

Fig. 12. Ejemplo de archivo "general_parameters.py".

3.2.1.1.2 Directorio "input"

Contiene un único archivo denominado "spectrum_in.py" que describe la señal transmitida en el primer punto del enlace.

Los parámetros que encontramos son:

- ‘laser_position’: una lista de valores booleanos en los que se describe si un determinado láser de transmisión estará encendido y emitiendo o apagado.
- Parámetros de la señal:
 - ‘p_ch’: la potencia de un canal, en Watios.
 - ‘p_nli’: la potencia de ruido no lineal inicial, en Watios.

- ‘p_ase’: la potencia de emisión de ruido de amplificación espontánea, en Watos.
- ‘b_ch’: el ancho de banda de un canal WDM a -3 dB.
- ‘roll_off’: el parámetro para el filtro root-raised-cosine, que tiene un valor típico de 0.15.

```

1  # coding=utf-8
2  spectrum = {
3      'laser_position': [1, 1, 1, 1],
4      'signals': [{
5          'b_ch': 0.032,
6          'roll_off': 0.15,
7          'p_ch': 0.001,
8          'p_nli': 0.0,
9          'p_ase': 0.0
10     }]
11 }

```

Fig. 13. Ejemplo de archivo "spectrum_in.py".

3.2.1.1.3 Archivos “__main_ole__” y “gnpy”

Son los encargados de obtener todos los datos de entrada de sus ficheros correspondientes desde los directorios de “configuration” e “input”, tratarlos y realizar los cálculos de desempeño de red, guardando los resultados obtenidos en el directorio de “output”.

El script que pone en marcha todos los cálculos es el Optical Link Emulator (OLE) main y realiza las siguientes funciones:

1. Lee el archivo de descripción del enlace.
2. Lee el espectro de entrada.
3. Llama a gnpy.py para cada uno de los elementos de la descripción del enlace de manera sucesiva, leyendo sus parámetros de salida y pasándoselos como entrada del siguiente componente.
4. Al final de la emulación del enlace, genera los archivos del espectro de salida.

El archivo “gnpy.py”, por otra parte, es la librería que contiene los métodos para realizar los cálculos lineales y no lineales de cada uno de los componentes.

Entre ellos se encuentran implementaciones del modelo matemático de integral doble que calcula los efectos no lineales con gran precisión pero de manera muy costosa y lenta, y una implementación más sencilla basada en la simplificación de unas fórmulas analíticas desarrolladas por el Politecnico di Torino que realizan los cálculos mucho más

rápidamente perdiendo muy poca precisión. Estos serán los cálculos que serán trasladados a Net2Plan en las secciones posteriores de este trabajo.

```

raised_cosine_comb
fwm_eff
get_freqarray
GN_integral
compute_psi
analytic_formula
gn_analytic
get_f_computed_interp
interpolate_in_range
gn_model
compute_gain_profile
compute_ase_noise
compute_edfa_profile
compute_attenuation_profile
passive_component
optical_amplifier
fiber
get_frequencies_wdm
get_spectrum_param
change_component_ref
compute_and_save_osnr
ole

```

Fig. 14. Lista de métodos de "gnpy.py".

3.2.1.1.4 Directorio "output"

En él se guardarán los datos de cada ejecución que realicemos en GNpy en una carpeta individual con la fecha y hora a la que se realizó.

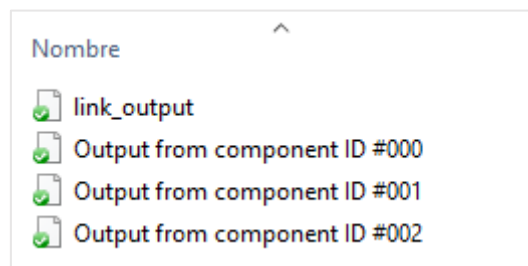


Fig. 15. Ejemplo de contenido de un directorio de salida de GNpy.

Todos los archivos de salida generados tendrán la misma estructura:

- OSNR lineal para el canal central.
- OSNR no-lineal para el canal central.
- Un array con las OSNR lineales para cada uno de los canales.
- Un array con las OSNR no-lineales para cada uno de los canales.

- Una variable “spectrum” que contendrá en primer lugar una lista de los láseres que están transmitiendo en el enlace y un array en el que cada posición contendrá los valores para un canal determinado de todas las variables analizadas: potencia del canal, potencia de ruido, etc.

```

1 # Output parameters. The values of OSNR are evaluated in the -3 dB channel band
2
3 osnr_lin_central_channel_db = 15.911605923
4
5 osnr_nl_central_channel_db = 13.4645122741
6
7 osnr_lin_db = [ 15.93415574  15.93302547  15.93189549  15.9307658  15.9296364
8 15.9285073  15.9273785  15.92624998  15.92512176  15.92399383
9 15.9228662  15.92173886  15.92061181  15.91948505  15.91835858
10 15.91723241  15.91610653  15.91498094  15.91385564  15.91273064
11 15.91160592  15.9104815  15.90935737  15.90823353  15.90710998
12 15.90598671  15.90486375  15.90374107  15.90261868  15.90149658
13 15.90037477  15.89925325  15.89813202  15.89701108  15.89589043
14 15.89477006  15.89364999  15.89253021  15.89141071  15.8902915
15 15.88917258]
16
17 osnr_nl_db = [ 14.08638375  13.86930034  13.76899484  13.70527078  13.659443
18 13.62424309  13.59610098  13.57300507  13.55371407  13.53741105
19 13.52353222  13.51167439  13.50154133  13.49291117  13.48561554
20 13.47952592  13.47454445  13.47059758  13.46763174  13.46561043
21 13.46451227  13.46433001  13.46507009  13.46675304  13.4694145
22 13.47310707  13.47790326  13.48389973  13.49122361  13.5000416
23 13.51057354  13.52311308  13.53806008  13.55597378  13.57766478
24 13.60436491  13.63807072  13.68232601  13.74434611  13.84268037
25 14.0569361 ]
26
27 spectrum = {'laser_position': [0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, ... ], |
28 'signals': [{'b_ch': 0.032, 'roll_off': 0.15, 'p_ch': 0.00063095734448019331,
29 'p_nli': 8.5331198415325372e-06, 'p_ase': 1.6091051529261068e-05},
30 {'b_ch': 0.032, 'roll_off': 0.15, 'p_ch': 0.00063095734448019331,
31 'p_nli': 9.791059322962751e-06, 'p_ase': 1.6095239862179033e-05},
32 ...

```

Fig. 16. Ejemplo de un fichero "link_output" de GNpy.

3.2.1.2 Arquitectura del proyecto

El flujo de esta parte del proyecto será la siguiente: el usuario podrá describir la red apoyándose en la interfaz gráfica de usuario de Net2Plan, donde tiene disponible librerías y modelos de todos los dispositivos que necesitará para describir el enlace. Una vez tenga el esquema a analizar, ejecutará un script que será el encargado de llamar a GNpy, quien realizará todos los cálculos y devolverá los resultados. Una vez terminado el proceso, Net2Plan los leerá y presentará al usuario de forma ordenada en un informe HTML que podrá ser visualizado en un navegador.

Los archivos de entrada y salida con los que se comunicarán Net2Plan y GNpy son los descritos en las secciones 3.2.1.1.1 (configuration), 3.2.1.1.2 (input) y 3.2.1.1.4 (output). Los correspondientes a las entradas serán escritos por Net2Plan a través del report que se ha desarrollado como objetivo de este trabajo y los que representan las

salidas de GNpy serán escritos por el mismo y leídos desde el report de Net2Plan, que organizará los datos de manera comprensible y los presentará en el HTML.

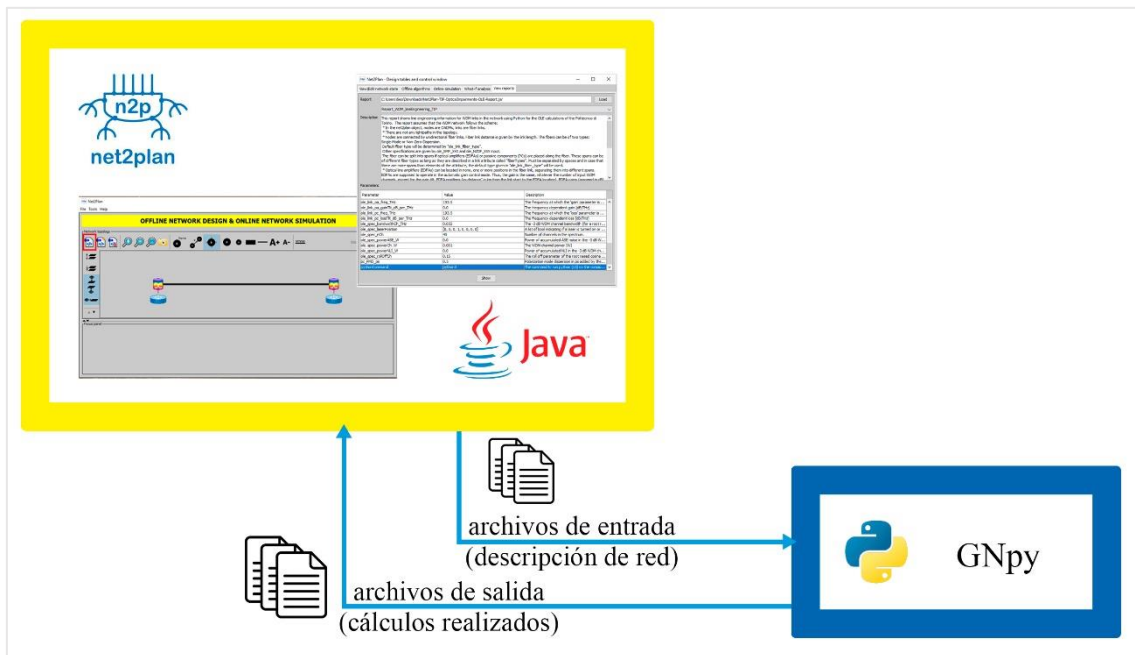


Fig. 17. Arquitectura del proyecto Net2Plan+GNpy.

3.2.1.3 Archivo *Report_WDM_lineEngineering_TIP.java*

Es el informe mencionado en la sección anterior y tiene la estructura típica de cualquier report en Net2Plan. Implementa la interfaz `IReport` que cuenta con los métodos:

- *executeReport* (*NetPlan netPlan*, *Map <String, String> reportParam*, *Map<String, String> net2planParameters*). Es el encargado de llamar a los métodos apropiados y orquestar toda la ejecución. Devuelve un `String` en HTML que Net2Plan interpretará en su visor interno.
- *getTitle*(*).* Devuelve el título del informe.
- *getDescription*(*).* Devuelve la descripción del informe.
- *getParameters*(*).* Devuelve una lista con los parámetros de entrada, donde cada uno viene representado por un `Triple <Nombre, Valor, Descripción>`.

Además de los métodos propios de la interfaz, se ha seguido la estructura de desarrollo de los informes de ejemplo de Net2Plan y la cadena de caracteres que contiene toda la información HTML preparada se obtiene a través de un método individual: *printReport*, que toma como argumentos un mapa con los elementos de cada enlace y otro con los datos calculados para cada uno de ellos.

```

● ▲ executeReport(NetPlan, Map<String, String>, Map<String, String>) : String
● ▲ getDescription() : String
● ▲ getParameters() : List<Triple<String, String, String>>
● ▲ getTitle() : String
■ printReport(Map<Link, LinkedList<Quadruple<Double, String, Double, String>>>, Map<Link, LinkedList<Pair<double[], double[]>>>) : String
■ getElementPositionsList(List<Link>) : LinkedList<Quadruple<Double, String, Double, String>>
■ computeImpairments(LinkedList<Quadruple<Double, String, Double, String>>) : LinkedList<Pair<double[], double[]>>
■ updateOSNRAfterEDFA(double, double, double, double) : double
■ s dB2Linear(double) : double
■ s linear2dB(double) : double
● getMostRecentOutputFile() : String
■ writeGNPYInputFiles(LinkedList<Quadruple<Double, String, Double, String>>) : boolean
■ writeGNPYFiberFile() : boolean
■ writeGNPYGeneralFile() : boolean
■ writeGNPYSpectrumFile() : boolean
■ writeGNPYLinkFile(LinkedList<Quadruple<Double, String, Double, String>>) : boolean
■ readOutputFile(List<String>) : Quintuple<Pair<String, Double>, Pair<String, Double>, Pair<String[], Double[]>, Pair<String[], Double[]>, Double[]>
■ getCentralChannelFinalSpecs(Link) : Quadruple<String, String, String, String>
■ getPerChannelFinalSpecs(Link) : Quadruple<String[], String[], String[], Double[]>
■ getPerChannelComponentSpecs(Link, int) : Quadruple<String[], String[], String[], Double[]>

```

Fig. 18. Lista de métodos en *Report_WDM_lineEngineering_TIP.java*

En cuanto al resto del código, dado que GNpy está pensado para calcular un único enlace a la vez, *executeReport* va realizando la siguiente rutina para cada uno de los enlaces de la red:

1. En primer lugar llama al método *writeGNPYInputFiles* que recibe una lista de los elementos del enlace y se encarga de llamar a los métodos de escritura individuales de cada fichero, guardándolos automáticamente en la carpeta indicada como parámetro de entrada al algoritmo, que contendrá la versión ejecutable de GNpy.
2. Una vez ha escrito todos, llama a la ejecución de GNpy a través de la consola de comandos.
3. Guarda la ruta de cada uno de los archivos de salida para su correspondiente enlace, de forma que luego sea capaz de leerlos apropiadamente.
4. Calcula con el método ya implementado en Net2Plan el desempeño de la red para luego poder comparar ambas ejecuciones.
5. Por último, llama a *printReport*.

Este último método es quien se encargará de leer todos los datos en el orden correcto, modelar los datos y generar el HTML a presentar al usuario.

PARÁMETROS DE ENTRADA

Respecto a los parámetros de entrada del report, se ha escrito la tabla que se muestra a continuación y que sigue las siguientes convenciones. El resto de parámetros de descripción de la red se tomarán de la manera habitual desde el diseño realizado en Net2Plan.

- Los parámetros relativos a la ejecución de GNpy vendrán precedidos de “ole_” en su nombre. Estos han sido ya detallados en las secciones 3.2.1.1.1 y 3.2.1.1.2. Según su archivo de escritura contendrán “gen_” para los

parámetros generales de GNpy de “general_parameters.py”, “link_” si van en el fichero de descripción del enlace y “spec_” si pertenecen al de descripción del espectro.

- Los parámetros que describen las fibras vendrán precedidos de “fiber_XXX”, siendo XXX el tipo de fibra, ya sea monomodo (SMF), o Non-Zero Dispersion-Shifted (NZDF) y también han sido descritos en la sección 3.2.1.1.1.
- Parámetros de descripción de los elementos: para fibra (no incluidos entre los de GNpy) serán precedidos de “fiber_”; amplificadores ópticos, “edfa_” y componentes pasivos, “pc_”; todos detallando las características del modelo. Los valores por defecto son los valores típicos de las hojas de catálogo.

Nombre	Valor	Descripción
ole_GNPYbasePath	/home/user/tip /gnpy- feature_polito 1	Home de la distribución de GNPY en la máquina local
ole_NZDF_Aeff_um2 ole_SMF_Aeff_um2	70.0 77.77	A_eff en GNPY
ole_NZDF_alpha1st_dB_per_k _km_per_THz ole_SMF_alpha1st_dB_per_ _km_per_THz	0.0 0.0	Alpha_1st en GNpy
ole_NZDF_alpha_dB_per_k m ole_SMF_alpha_dB_per_k m	0.22 0.2	Alpha en GNpy
ole_NZDF_beta2_ps2_per_k m ole_SMF_beta2_ps2_per_k m	21.0 21.27	Beta_2 en GNpy
ole_NZDF_n2_m2_per_W ole_SMF_n2_m2_per_W	2.5E-20 2.5E-20	N2 en GNpy
ole_NZDF_refFreq_THz ole_SMF_refFreq_THz	193.5 193.5	Frecuencia a la que están evaluados los parámetros de la fibra
ole_gen_f0_THz	192.075	Descripción en ref. 3.2.1.1.1
ole_gen_isAnalytic	true	Descripción en ref. 3.2.1.1.1
ole_gen_kindInterp	linear	Descripción en ref. 3.2.1.1.1
ole_gen_nPointsIntegral	500	Descripción en ref. 3.2.1.1.1
ole_gen_nPointsMinIntegral	4	Descripción en ref. 3.2.1.1.1
ole_gen_ns	800	Descripción en ref. 3.2.1.1.1
ole_gen_pointsNotInterp	2	Descripción en ref. 3.2.1.1.1
ole_gen_thFWM	50.0	Descripción en ref. 3.2.1.1.1

ole_link_fiber_type	SMF	Un string detallando el tipo de fibra por defecto a utilizar en los enlaces. Este parámetro puede ser sobrescrito con los atributos de Net2Plan.
ole_link_oa_freq_THz	193.5	Descripción en ref. 3.2.1.1.1
ole_link_oa_gainTlt_dB_per_THz	0.0	Descripción en ref. 3.2.1.1.1
ole_link_pc_freq_THz	193.5	Descripción en ref. 3.2.1.1.1
ole_link_pc_lossTlt_dB_per_THz	0.0	Descripción en ref. 3.2.1.1.1
ole_spec_bandwidthCh_THz	0.032	Descripción en ref. 3.2.1.1.2
ole_spec_laserPosition	[0, 0, 0, 1, 0, 0, 0, 0]	Descripción en ref. 3.2.1.1.2
ole_spec_nCh	40	Descripción en ref. 3.2.1.1.2
ole_spec_powerASE_W	0.0	Descripción en ref. 3.2.1.1.2
ole_spec_powerCh_W	0.001	Descripción en ref. 3.2.1.1.2
ole_spec_powerNLI_W	0.0	Descripción en ref. 3.2.1.1.2
ole_spec_rolloffCh	0.15	Descripción en ref. 3.2.1.1.2
pc_PMD_ps	0.5	PMD en ps añadida por un componente pasivo
pythonCommand	python3	Comando para ejecutar python (v3) en la máquina local. Normalmente es 'python3' o 'python'.
edfa_PMD_ps	0.5	PMD en ps añadida por un EDFA
fiber_PMD_ps_per_sqrroot_km	0.4	PMD de la fibra (PMD_Q link factor)

3.2.2 Implementación de los cálculos del Gaussian Noise Model en el Report Line Engineering de Net2Plan

Al alcanzar esta sección ya contamos con una implementación funcional de los cálculos de OSNR tal como los describe GNpy, sin embargo, al realizarse a través de OLE en python con la estructura descrita en la sección 3.2.1.2, el tiempo de cómputo es muy elevado, y existe la restricción de tener funcionando en la máquina local una versión 3 de Python y otros prerequisites que serán detallados en la sección 5.1.1. Además, existe el inconveniente de generar numerosos archivos de escritura y lectura que ocupan espacio y retrasan la obtención de los resultados, por lo que no es nada eficiente.

Gracias al conocimiento adquirido durante el desarrollo de las fases anteriores de este proyecto, se decidió continuar e implementar los cálculos del modelo de ruido Gaussiano para la OSNR no lineal directamente en Net2Plan, de esta forma, se solucionan todas las desventajas descritas anteriormente.

Al decidir esto se determinó la necesidad de realizar una clase *OpticalImpairmentUtils.java* que incluya todos los métodos comunes para el tipo de cálculos de las redes ópticas y tenerlos de esta forma centralizados y documentados.

3.2.2.1 Archivo *OpticalImparimentUtils.java*

En este archivo se han implementado los métodos y constantes de la Fig. 19.



```

OpticalImpairmentUtils
  SF constant_h : double
  SF constant_c : double
  SF stSpectrum_powerPerChannel_W : String
  SF stSpectrum_nliNoisePowerg_W : String
  SF stSpectrum_aseNoisePower_W : String
  SF stSpectrum_bandwidthPerChannel_THz : String
  SF stFiber_alpha_dB_per_km : String
  SF stFiber_alpha1st_dB_per_km_per_THz : String
  SF stFiber_beta2_ps2_per_km : String
  SF stFiber_effectiveArea_um2 : String
  SF stFiber_n2Coeff_m2_per_W : String
  C OpticalImpairmentUtils()
  S getOSNR(Map<String, double[]>) : Triple<double[], double[], double[]>
  S computeNLIfiber(double, double, double, double, double, double, double, double[], double[], double, double[]) : double[]
  S getAttenuationCompLinear(double, double, double, double, double[]) : double[]
  S computeImpairments(List<Quadruple<Double, String, Double, String>>, Map<String, double[]>, Map<String, Map<String, Double>>, dc
  S getSpectrumAfterOADM(double, double[], double[], double[], double, double, double, double[], double[]) : Triple<double[], double[], double[]>
  S asinh(double) : double
  S dB2linear(double) : double
  S linear2dB(double) : double
  S linear2dB(double[]) : double[]
  S dB2linear(double[]) : double[]
  
```

Fig. 19. Contenido de la clase *OpticalImparimentUtils.java*

Como se puede comprobar de la lista anterior, se incluyen tanto métodos útiles para la conversión de lineal a dB (valores únicos y en arrays), como otros más importantes desarrollados a continuación:

- *Public getOSNR*. Recibe como entrada un mapa con el espectro de la red y devuelve un triple con la OSNR por canal lineal, no-lineal y la total.
- *Private computeNLIfiber*. Obtiene los cálculos no-lineales de un tramo de fibra. Toma como parámetros todos los relativos a la descripción del tipo de fibra y devuelve un array con la densidad espectral de potencia de ruido no lineal para cada canal.
- *Private getAttenuationCompLinear*. Obtiene la componente lineal de la atenuación y toma como parámetros los datos de la fibra.
- *Public computeImpairments*. Calcula la OSNR, potencia y PMD del enlace descrito por los elementos pasados como argumento empleando los cálculos de la fórmula GN para los no lineales.

- *Private getSpectrumAfterOADM*. Obtiene los datos del espectro a la salida de un OADM.

3.2.2.2 Archivo *Report_WDM_lineEngineering_GNModel.java*

Para escribir este archivo se ha empleado el *Report_WDM_lineEngineering* ya existente en *Net2Plan*, que se ha actualizado e incorporado las modificaciones necesarias para realizar los cálculos de la OSRN no-lineal con el modelo GN.

Una vez podemos usar los métodos escritos en la clase de la sección anterior, el desarrollo de esta es casi inmediato.

```

executeReport(NetPlan, Map<String, String>, Map<String, String>) : String
getDescription() : String
getParameters() : List<Triple<String, String, String>>
getTitle() : String
getFiberSpecsMap() : Map<String, Map<String, Double>>
initializeSpectrum() : Map<String, double[]>
getLaserPositions(boolean[]) : Boolean[]
getBasebandFrequency(boolean[]) : double[]
computeWarningMessages(List<Quadruple<Double, String, Double, String>>, List<Quadruple<Map<String, double[]>, Double, Map<String, double[]>, Double>>>) : List<String>
getElementPositionsListPerLightpath(List<Link>) : List<Quadruple<Double, String, Double, String>>
printReport(Map<Link, List<Quadruple<Double, String, Double, String>>>, Map<Link, List<Quadruple<Map<String, double[]>, Double, Map<String, double[]>, Double>>>, Map<

```

Fig. 20. Métodos del *Report_WDM_lineEngineering_GNModel.java*

Al igual que lo descrito en la sección 3.2.1.3, implementa la interfaz *IReport* de la misma forma, exceptuando el método *executeReport* que en esta ocasión no tiene que escribir ficheros ni llamar a OLE desde la consola de python, sino que llama al método de *computeImpairments* de la clase *OpticalImpairmentUtils* que se encargará de obtener todos los datos de desempeño de los enlaces llamando a los métodos privados necesarios de dicha clase.

Además, tal como puede observarse en la lista de la Fig. 20, hay algunos métodos privados que son utilizados para tratar los datos.

- *getFiberSpecsMap*. Comprueba que los parámetros de entrada relativos a las fibras hayan sido introducidos correctamente y los guarda en un mapa de clave el nombre del tipo de fibra y como valor un mapa <parámetro, valor>.
- *initializeSpectrum*. Inicializa los parámetros del espectro al comienzo del enlace, puesto que el método de *OpticalImpairmentUtils* necesita esta entrada.
- *getLaserPositions*. Obtiene el estado de los láseres que transmiten para cada canal.
- *getBasebandFrequency*. Basándose en lo anterior, obtiene las frecuencias de trabajo de cada canal.

- *computeWarningMessages*. Dados los datos de desempeño calculados en *OpticalImpairmentUtils*, determina qué valores no cumplen con las especificaciones del diseño y genera todos los warnings asociados.
- *printReport*. Obtiene la cadena de caracteres HTML que será interpretada por el navegador.

PARÁMETROS DE ENTRADA

Respecto a los parámetros de entrada del report, se ha escrito la tabla que se muestra a continuación y que sigue las siguientes convenciones. El resto de parámetros de descripción de la red se tomarán de la manera habitual desde el diseño realizado en Net2Plan.

- Los parámetros relativos a la ejecución de GNpy vendrán precedidos de “gn_” en su nombre. Estos han sido ya detallados en las secciones 3.2.1.1.1 y 3.2.1.1.2.
- Los parámetros que describen las fibras vendrán precedidos de “fiber_” y “fibers_” en el caso de ser genéricos para varios tipos (se escribirán en orden y separados por espacios), y también han sido descritos en la sección 3.2.1.1.1.
- Parámetros de descripción de los elementos: para transponders, precedidos de “tp_”, OADM, con “oadm_”; amplificadores ópticos, “edfa_” y componentes pasivos, “pc_”; todos detallando las características del modelo. Los valores por defecto son los valores típicos de las hojas de catálogo.
- Por último, los relativos a las penalties de OSNR para los cálculos de Net2Plan, todos precedidos de “osrnPenalty_”.

Nombre	Valor	Descripción
gn_gen_f0_THz	192.075	f0 en GNpy
gn_gen_ns	800	ns en GNpy
gn_spec_nCh	16	nCh en GNpy
gn_spec_laserPosition	false false true false false	laserPosition en GNpy
gn_spec_bandwidthCh_THz	0.032	b_ch en GNpy
fiber_PMD_ps_per_sqrt_km	0.4	PMD por km ^{0.5} de fibra (PMD_Q link factor)
fiber_default_type	SMF	Tipo de fibra a emplear por defecto si no se indica otra cosa en el atributo 'fiberTypes' de Net2Plan. Debe ser uno de los valores de 'fibers_types'.
fibers_alpha_dB_per_km	0.2 0.22	Alpha en GNpy [dB/km]

fibers_alpha1st_dB_per_km_per_THz	0 0	Alpha_1st en GNpy
fibers_beta2_ps2_per_km	21.27 21	Beta_2 en GNpy
fibers_n2_m2_per_W	2.5E-20 2.5E-20	N2 en GNpy [m ² /W]
fibers_Aeff_um2	77.77 70	A_eff en GNpy [um ²]
fibers_types	SMF NZDF	Nombres de los tipos de fibra descritos en fibers_XXX parámetros. Deben estar siempre en el mismo orden.
fibers_numberOfFiberTypes	2	Número de tipos de fibra descritos.
tp_minOSNR_dB	7.0	OSNR mínima en recepción
tp_minWavelength_nm	1529.55	Longitud de onda mínima utilizable
tp_maxWavelength_nm	1561.84	Longitud de onda máxima utilizable
tp_pmdTolerance_ps	10.0	Tolerancia a la PMD (ps) en recepción
tp_inputPowerSensitivityMin_dBm	-20.0	Sensibilidad de potencia de entrada en recepción
tp_inputPowerSensitivityMax_dBm	-8.0	Máxima potencia de entrada en recepción.
oadm_outputPowerPerChannel_W	0.001	Potencia de salida por canal del OADM
oadm_perChannelMinInputPower_dBm	-19.0	Potencia mínima por canal a la entrada del OADM
oadm_perChannelMaxInputPower_dBm	1000.0	Potencia máxima por canal a la entrada del OADM
oadm_muxDemuxPMD_ps	0.5	PMD de mux/demux del OADMs.
oadm_preAmplifierPMD_ps	0.5	PMD de OADM preamplifier
oadm_boosterPMD_ps	0.5	PMD de OADM booster amplifier
oadm_addChannelNoiseFactor_dB	6.0	Noise factor para canales añadidos
oadm_dropChannelNoiseFactor_dB	6.0	Noise factor para drop channels
oadm_expressChannelNoiseFactor_dB	10.0	Noise factor para express channels
edfa_minWavelength_nm	1530.0	Longitud de onda mínima para el EDFA
edfa_maxWavelength_nm	1563.0	Longitud de onda máxima para el EDFA
edfa_minInputPower_dBm	-29.0	Potencia de entrada mínima al EDFA
edfa_maxInputPower_dBm	2.0	Potencia de entrada máxima al EDFA
edfa_minOutputPower_dBm	-6.0	Potencia de salida mínima al EDFA
edfa_maxOutputPower_dBm	19.0	Potencia de salida máxima al EDFA
edfa_minGain_dB	17.0	Ganancia mínima al EDFA
edfa_maxGain_dB	23.0	Ganancia máxima al EDFA
edfa_PMD_ps	0.5	PMD añadida por el EDFA
edfa_noiseFactorMaximumGain_dB	5.0	Noise factor del EDFA a la máxima ganancia (para el resto se estiman interpolando)

edfa_noiseFactorMinimumGain_dB	5.0	Noise factor de EDFA a la mínima ganancia (para el resto se estiman interpolando)
pc_PMD_ps	0.5	PMD del PC
osnrPenalty_nonLinear_dB	2.0	OSNR penalty por los efectos no lineales
osnrPenalty_PMD_dB	0.5	OSNR penalty por la PMD (asumiendo que esté dentro de los límites)
osnrPenalty_PDL_dB	0.3	OSNR penalty por las pérdidas de dispersión de polarización
osnrPenalty_transmitterChirp_dB	0.5	OSNR penalty por transmitter chirp
osnrPenalty_OADMCrosstalk_dB	0.8	OSNR penalty por crosstalk en los OADMs
osnrPenalty_unassignedMargin_dB	3.0	OSNR penalty por efectos no contemplados (ej. aleatorios, envejecimiento de equipos...)

4 GENERACIÓN DE INFORME Y RESULTADOS.

Gracias al trabajo realizado podemos obtener los datos de desempeño de los enlaces de fibra empleando Net2Plan de manera cómoda y sencilla.

4.1 RESULTADOS PARA EL REPORT BASADO EN GNPY

Siguiendo las instrucciones del Manual de Usuario que se detalla en la sección 5.1 para el informe que emplea OLE y GNpy como base, obtenemos los resultados que se muestran a continuación.

En el archivo n2p proporcionado en el paquete de descarga del informe para comprobar el ejemplo, tenemos un enlace WDM como el descrito en la Fig. 21 en el que tenemos una serie de spans separados por amplificadores situados cada 100 km, con figuras de ruido de 5 dB y ganancias de 20 dB. Además, consta también de dos componentes pasivos en los extremos que inducen unas pérdidas de 0.5 dB cada uno.

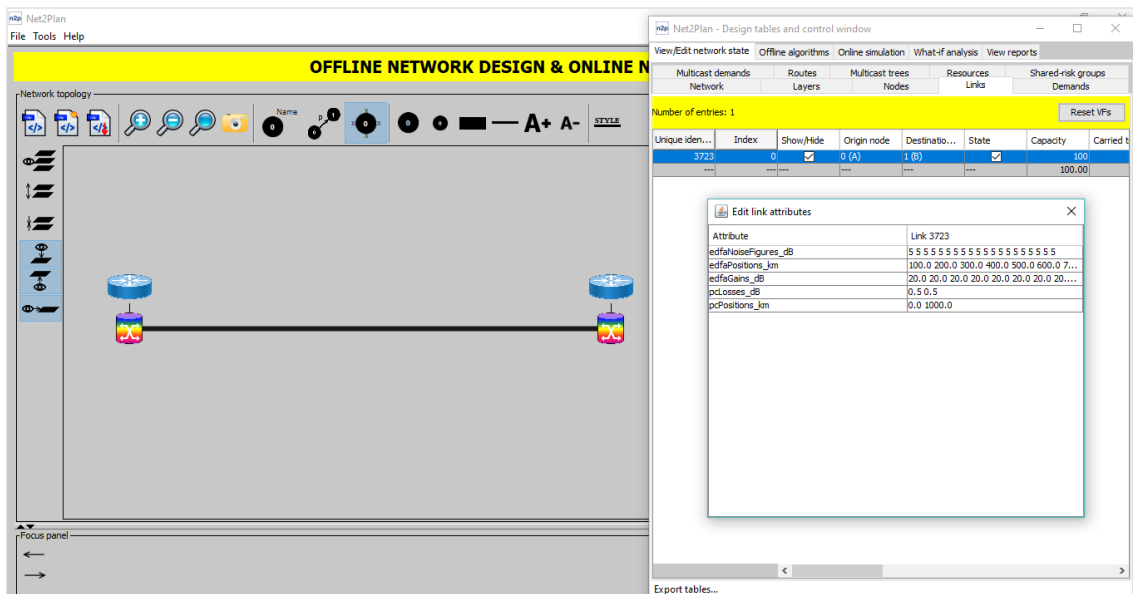


Fig. 21. Ejemplo en Net2Plan para el report basado en GNpy.

Mantenemos la mayoría de parámetros de entrada por defecto, cambiando únicamente los relativos a nuestra instalación de Python para que GNpy sea capaz de encontrar su carpeta y ejecutarse correctamente.

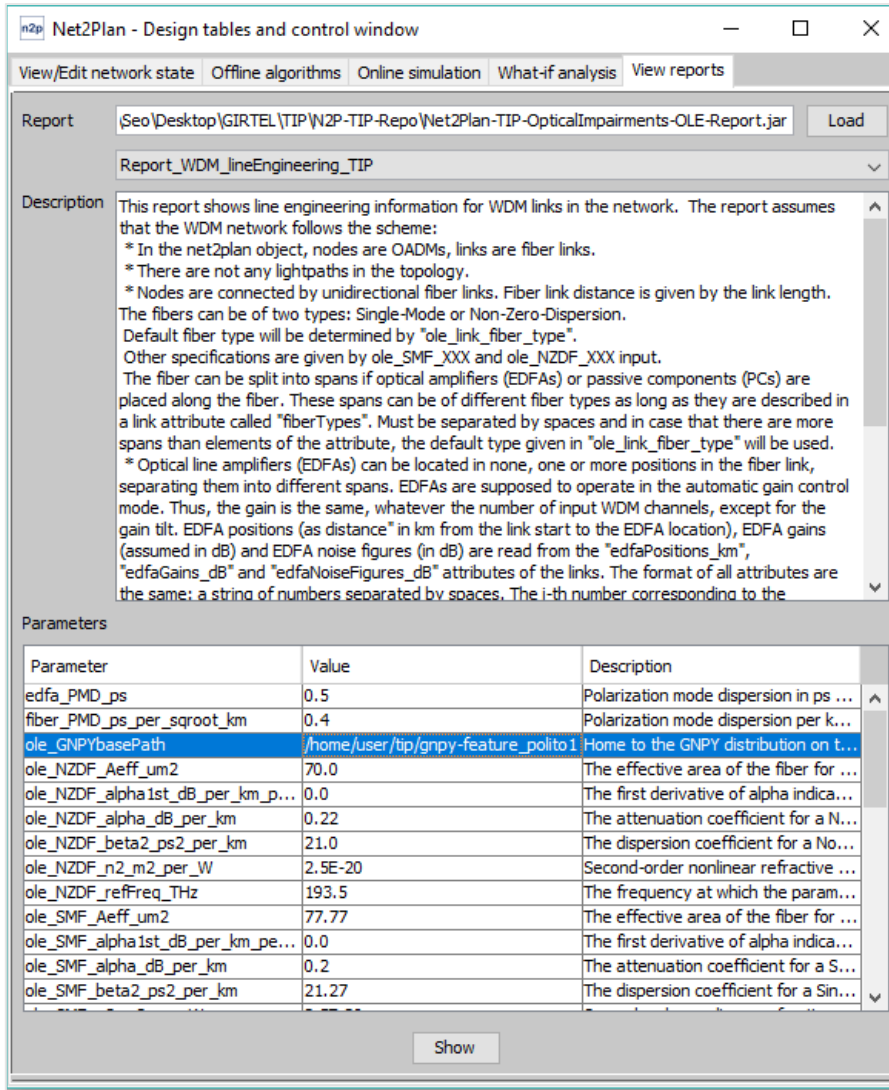


Fig. 22. Parámetros para la ejecución del report basado en GNpy.

Al ejecutar vemos que se genera un informe que pasamos a vista en el navegador.

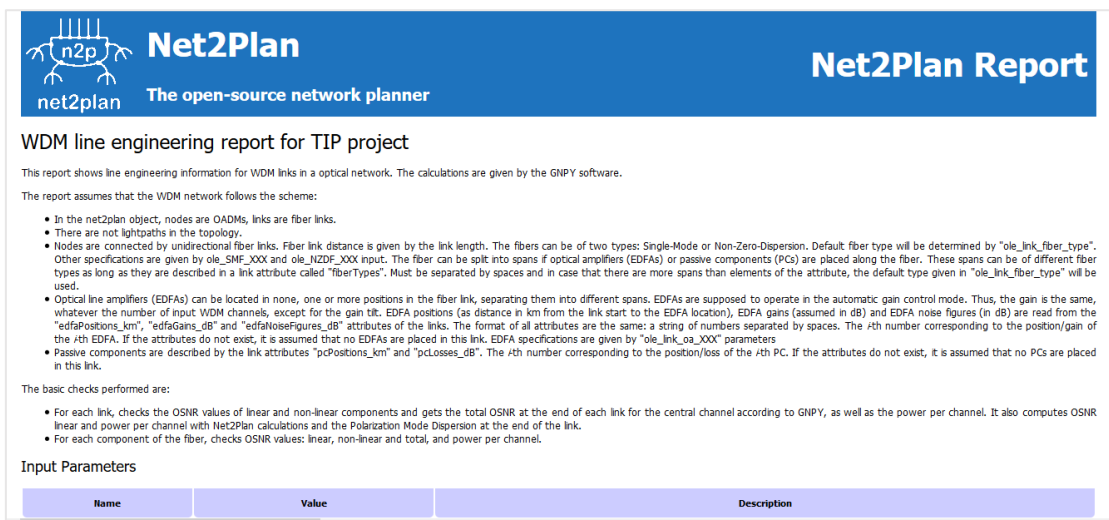


Fig. 23. Vista en el navegador del report basado en GNpy.

En él podemos observar el detalle de todos los cálculos para la OSNR lineal y no-lineal, así como la PMD del enlace y la potencia por canal, tanto las calculadas por OLE y GNpy como las obtenidas a través de los cálculos de Net2Plan.

Si analizamos los resultados, observamos que las diferencias en la OSNR entre los cálculos de la herramienta GNpy y los que obtenía Net2Plan anteriormente son únicamente de 0.04 dB, dando una estimación más optimista el modelo de GNpy.

LINK INFORMATION SUMMARY - Signal metrics at the end of the link for the central channel.

Link #	Length (km)	# EDFAs	# PCs	[OLE] OSNR-total central channel (dB)	[OLE] OSNR linear central channel (dB)	[OLE] OSNR non-linear central channel (dB)	[Net2Plan] OSNR linear (dB)	Polarization Mode Dispersion (ps)	[OLE] Power per channel (mW)	[Net2Plan] Power per channel (mW)
e0 (id 3723) (A --> B)	2000	20	2	11.22	15.13	13.49	15.11	18.04	0.794	0.794

LINK INFORMATION SUMMARY - Signal metrics at the end of the link for each channel.

Link	Channel 0	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5	Channel 6	Channel 7	Channel 8	Channel 9	Channel 10	Channel 11	Channel 12
e0 (id 3723)	OLE_Linear=15.15 NLP_Linear=15.11 Non-Linear=13.79 Total=11.5 Power=7.94E-4	OLE_Linear=15.15 NLP_Linear=15.11 Non-Linear=13.79 Total=11.41 Power=7.94E-4	OLE_Linear=15.15 NLP_Linear=15.11 Non-Linear=13.79 Total=11.37 Power=7.94E-4	OLE_Linear=15.15 NLP_Linear=15.11 Non-Linear=13.79 Total=11.34 Power=7.94E-4	OLE_Linear=15.15 NLP_Linear=15.11 Non-Linear=13.79 Total=11.32 Power=7.94E-4	OLE_Linear=15.15 NLP_Linear=15.11 Non-Linear=13.79 Total=11.3 Power=7.94E-4	OLE_Linear=15.15 NLP_Linear=15.11 Non-Linear=13.79 Total=11.29 Power=7.94E-4	OLE_Linear=15.15 NLP_Linear=15.11 Non-Linear=13.79 Total=11.28 Power=7.94E-4	OLE_Linear=15.15 NLP_Linear=15.11 Non-Linear=13.79 Total=11.27 Power=7.94E-4	OLE_Linear=15.15 NLP_Linear=15.11 Non-Linear=13.79 Total=11.26 Power=7.94E-4	OLE_Linear=15.15 NLP_Linear=15.11 Non-Linear=13.79 Total=11.26 Power=7.94E-4	OLE_Linear=15.15 NLP_Linear=15.11 Non-Linear=13.79 Total=11.25 Power=7.94E-4	OLE_Linear=15.15 NLP_Linear=15.11 Non-Linear=13.79 Total=11.25 Power=7.94E-4

LINK DETAILED INFORMATION

Number of links: 1

LINK # e0 (id 3723) (A --> B)

Position (km)	Position (description)	Channel 0	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5	Channel 6	Channel 7	Channel 8	Channel 9	Channel 10	Channel 11
0	Output of PC (loss 0.5 dB)	Linear=∞ Non-linear=∞ Total=∞ NLP_Linear=∞ Power=8.91E-4	Linear=∞ Non-linear=∞ Total=∞ NLP_Linear=∞ Power=8.91E-4	Linear=∞ Non-linear=∞ Total=∞ NLP_Linear=∞ Power=8.91E-4	Linear=∞ Non-linear=∞ Total=∞ NLP_Linear=∞ Power=8.91E-4	Linear=∞ Non-linear=∞ Total=∞ NLP_Linear=∞ Power=8.91E-4	Linear=∞ Non-linear=∞ Total=∞ NLP_Linear=∞ Power=8.91E-4	Linear=∞ Non-linear=∞ Total=∞ NLP_Linear=∞ Power=8.91E-4	Linear=∞ Non-linear=∞ Total=∞ NLP_Linear=∞ Power=8.91E-4	Linear=∞ Non-linear=∞ Total=∞ NLP_Linear=∞ Power=8.91E-4	Linear=∞ Non-linear=∞ Total=∞ NLP_Linear=∞ Power=8.91E-4	Linear=∞ Non-linear=∞ Total=∞ NLP_Linear=∞ Power=8.91E-4	Linear=∞ Non-linear=∞ Total=∞ NLP_Linear=∞ Power=8.91E-4
0	Output of SPAN (100.0 km, type SPM)	Linear=∞ Non-linear=32.04 Total=32.63 NLP_Linear=∞	Linear=∞ Non-linear=32.04 Total=32.04 NLP_Linear=∞	Linear=∞ Non-linear=32.04 Total=31.78 NLP_Linear=∞	Linear=∞ Non-linear=32.04 Total=31.62 NLP_Linear=∞	Linear=∞ Non-linear=32.04 Total=31.51 NLP_Linear=∞	Linear=∞ Non-linear=32.04 Total=31.42 NLP_Linear=∞	Linear=∞ Non-linear=32.04 Total=31.26 NLP_Linear=∞	Linear=∞ Non-linear=32.04 Total=31.20 NLP_Linear=∞	Linear=∞ Non-linear=32.04 Total=31.22 NLP_Linear=∞	Linear=∞ Non-linear=32.04 Total=31.19 NLP_Linear=∞	Linear=∞ Non-linear=32.04 Total=31.15 NLP_Linear=∞	Linear=∞ Non-linear=32.04 Total=31.15 NLP_Linear=∞

Fig. 24. Resultados de la ejecución del report basado en GNpy.

4.2 RESULTADOS PARA EL REPORT LINE ENGINEERING DE NET2PLAN

Siguiendo las instrucciones del manual de la sección 5.2 se puede ejecutar el report desarrollado en este proyecto y obtener el informe de resultados de desempeño de la red.

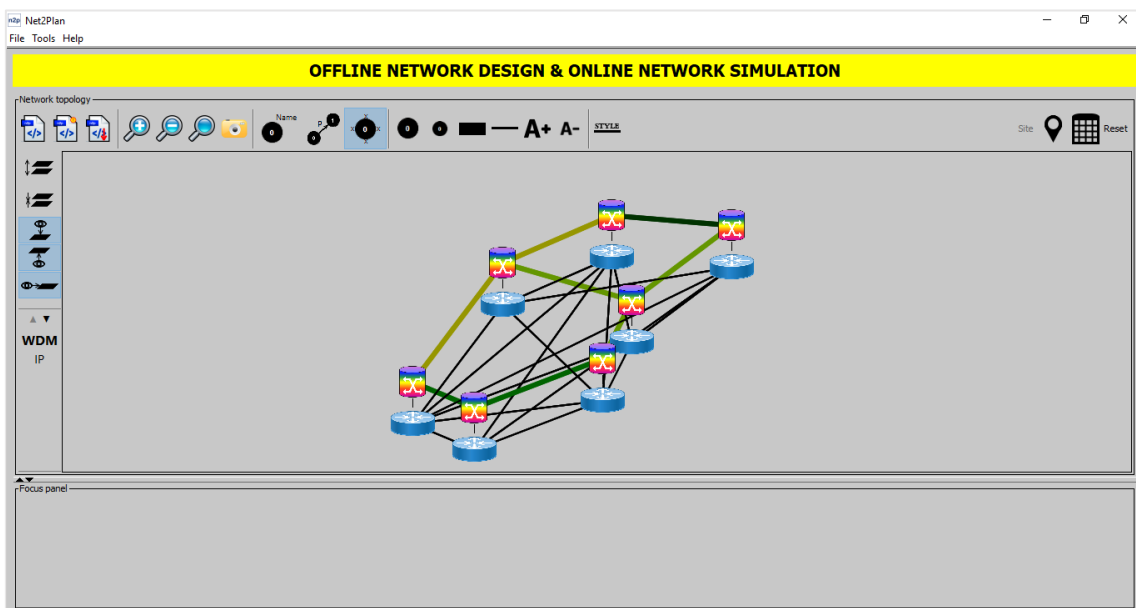


Fig. 25. Topología para ejecutar el report Line Engineering de Net2Plan.

Esta topología consta de siete nodos con capa tanto óptica como IP, con EDFAs situados a lo largo de los enlaces de fibra, sus ganancias asignadas y componentes pasivos incluidos.

Con una topología como la mostrada en la imagen anterior, ahora podemos ejecutar el report sin temor a que tarde demasiado por llamadas innecesarias a Python y con la seguridad de que los cálculos obtenidos serán correctos, pues son los mismos que en la sección anterior ya hemos contrastado y han sido testeados.

Net2Plan

The open-source network planner

Net2Plan Report

WDM line engineering report for lighthpath-based networks with GN calculations

This report shows line engineering information for WDM links in a multilayer optical network. The impairment calculations are based on the Gaussian Noise Model developed by Politecnico di Torino and their analytic formula. Other calculations are inspired in the procedures described in the 2009 ITU-T WDM manual "optical fibres, cables and systems".

The report assumes that the WDM network follows the scheme:

- In the net2plan object, nodes are OADMs, links are fiber links, and routes are lighthpaths: WDM channels optically switched at intermediate nodes.
- Nodes are connected by unidirectional fiber links. Fiber link distance is given by the link length. Other specifications are given by fiber_XXX input parameters, each one describing the parameter for the fiber types specified in fibers_types, in the same order and separated by spaces. The fiber can be split into spans of optical amplifiers (EDFAs) and/or passive components (PCs) are placed along the fiber. These spans can be of different fiber types as long as they are described in a link attribute called "FiberTypes". Must be separated by spaces and, in case that there were more spans than elements of the attribute, the default type given in "fiber_default_type" would be used.
- Optical line amplifiers (EDFAs) can be located in nodes, one or more positions in the fiber link, separating them in different spans. EDFAs are supposed to operate in the automatic gain control mode. Thus, the gain is the same, whatever the number of input WDM channels. EDFA positions (as distance in km from the link start to the EDFA location), EDFA gains (assumed in dB) and EDFA noise figures (in dB) are read from the "edfaPositions_km", "edfaGains_dB" and "edfaNoiseFigures_dB" attributes of the links. The format of all attributes are the same: a string of numbers separated by spaces. The nth number corresponding to the position/gain of the nth EDFA. If the attributes do not exist, it is assumed that no EDFAs are placed in the link. EDFA specifications are given by "edfa_XXX" parameters.
- There are not Dispersion compensating modules (DCMs) in the topology, since the Gaussian Noise Model is used.
- Passive components are described by the link attributes "pcPositions_km" and "pcLosses_dB". The nth number corresponding to the position/loss of the nth PC. If the attributes do not exist, it is assumed that no PCs are placed in this link. Other specifications for Passive Components will be described in link_pc_XXX input parameters.
- Fiber links start and end in OADM modules, that permit adding, dropping and optically switch individual WDM channels. OADMs have a pre-amplifier (traversed by drop and express channels) and a boost amplifier (traversed by add and express channels). They are supposed to equalize the channel power at their outputs, to a fixed value (added and express channels will thus have the same power in the fibers). Also, OADMs attenuate appropriately the optical signal coming from the pre-amplifier, in the drop channels, so that they fall within the receiver sensitivity range. OADM noise figures for add, drop and express channels are given as input parameters. PMD values for add, drop and express channels are computed assuming that: (i) add channel traverse a multiplexer and the booster, (ii) drop channels traverse the pre-amplifier and a demultiplexer, (iii) express channels traverse the two amplifiers. The required parameters are provided in oadm_XXX parameters.
- Each channel ends in a receiver, with specifications given by "rxp_XXX" parameters.

The basic checks performed are:

- For each link, signal power levels are within operating ranges at the oadm/edfas, both when the link has one single active channel, or when all the "gn_spec_rcch" are active
- For each route (lighthpath), OSNR (Optical Signal to Noise Ratio) is within the operating range at the receiver. A set of margins are considered to account to several not directly considered impairments.
- For each route (lighthpath), PMD (Polarization mode dispersion) is within the operating range at the receiver

Fig. 26. Informe Line Engineering de Net2Plan.

Una vez obtenemos los resultados, vemos en el navegador una imagen como la mostrada en las Fig. 26 y Fig. 27, donde podemos apreciar la relación señal a ruido de cada enlace y ruta, así como la potencia por canal, la PMD y los warnings que nos alertan si el modelo de transponder no soporta esa potencia de entrada o la OSNR recae por debajo del mínimo necesario.

Link #	Length (km)	# EDFAs	# PCs	OSNR total (dB)	Power per WDM channel (dBm)	Polarization Mode Dispersion (ps)	Warnings
#0:0(Madrid->Valencia) (Madrid -> Valencia)	132	1	0	14.91	-8.94	4.68	At 52.0km: Power at the EDFA output is (if all WDM channels were active) 19.1 dBm. It should be between [-6.0, 19.0] dBmAt 132.0km: OSNR at the RECEIVER is 11.72392245954801 dB. It is below the tolerance plus margin 7.0 dB + penalties 7.1 dB = 14.1 dB
#1:0(Madrid->Sevilla) (Madrid -> Sevilla)	171	2	0	2.7	0.26	5.33	At 106.0km: Power at the EDFA output is (if all WDM channels were active) 8.3 dBm. It should be between [-20.0, 2.0] dBmAt 106.0km: Power at the EDFA output is (if all WDM channels were active) 25.3 dBm. It should be between [-6.0, 19.0] dBmAt 171.0km: OSNR at the RECEIVER is 4.3384552479489 dB. It is below the tolerance plus margin 7.0 dB + penalties 7.1 dB = 14.1 dB
#2:0(Madrid->Zaragoza) (Madrid -> Zaragoza)	119	1	0	15.09	-6.34	4.45	At 52.0km: Power at the EDFA output is (if all WDM channels were active) 19.1 dBm. It should be between [-6.0, 19.0] dBmAt 119.0km: OSNR at the RECEIVER is 11.77985643675838 dB. It is below the tolerance plus margin 7.0 dB + penalties 7.1 dB = 14.1 dB
#3:0(Barcelona->Valencia) (Barcelona -> Valencia)	132	1	0	14.38	-7.44	4.68	At 58.0km: Power at the EDFA output is (if all WDM channels were active) 19.4 dBm. It should be between [-6.0, 19.0] dBmAt 132.0km: OSNR at the RECEIVER is 11.57241774033605 dB. It is below the tolerance plus margin 7.0 dB + penalties 7.1 dB = 14.1 dB
#4:0(Barcelona->Zaragoza) (Barcelona -> Zaragoza)	112	1	0	15.1	-4.44	4.32	At 54.0km: Power at the EDFA output is (if all WDM channels were active) 19.200000000000003 dBm. It should be between [-6.0, 19.0] dBmAt 112.0km: OSNR at the RECEIVER is 11.786909021817955 dB. It is below the tolerance plus margin 7.0 dB + penalties 7.1 dB = 14.1 dB
#5:0(Valencia->Madrid) (Valencia -> Madrid)	132	1	0	14.91	-8.94	4.68	At 52.0km: Power at the EDFA output is (if all WDM channels were active) 19.1 dBm. It should be between [-6.0, 19.0] dBmAt 132.0km: OSNR at the RECEIVER is 11.72392245954801 dB. It is below the tolerance plus margin 7.0 dB + penalties 7.1 dB = 14.1 dB
#6:0(Valencia->Barcelona) (Valencia -> Barcelona)	132	1	0	14.91	-8.94	4.68	At 52.0km: Power at the EDFA output is (if all WDM channels were active) 19.1 dBm. It should be between [-6.0, 19.0] dBmAt 132.0km: OSNR at the RECEIVER is 11.72392245954801 dB. It is below the tolerance plus margin 7.0 dB + penalties 7.1 dB = 14.1 dB
#7:0(Valencia->Murcia) (Valencia -> Murcia)	78	0	0	29.21	-15.6	3.6	
#8:0(Sevilla->Madrid) (Sevilla -> Madrid)	171	2	0	2.74	0.01	5.33	At 51.0km: Power at the EDFA output is (if all WDM channels were active) 19.05 dBm. It should be between [-6.0, 19.0] dBmAt 105.0km: Power at the EDFA output is (if all WDM channels were active) 8.25 dBm. It should be between [-20.0, 2.0] dBmAt 105.0km: Power at the EDFA output is (if all WDM channels were active) 25.25 dBm. It should be between [-6.0, 19.0] dBmAt 171.0km: OSNR at the RECEIVER is 4.382561438950743 dB. It is below the tolerance plus margin 7.0 dB + penalties 7.1 dB = 14.1 dB
#9:0(Sevilla->Málaga) (Sevilla -> Málaga)	69	0	0	29.34	-13.8	3.4	
#10:0(Zaragoza->Madrid) (Zaragoza -> Madrid)	119	1	0	15.01	-6.09	4.45	At 53.0km: Power at the EDFA output is (if all WDM channels were active) 19.15 dBm. It should be between [-6.0, 19.0] dBmAt 119.0km: OSNR at the RECEIVER is 11.757224506707718 dB. It is below the tolerance plus margin 7.0 dB + penalties 7.1 dB = 14.1 dB
#11:0(Zaragoza->Barcelona) (Zaragoza -> Barcelona)	112	1	0	15.1	-4.44	4.32	At 54.0km: Power at the EDFA output is (if all WDM channels were active) 19.200000000000003 dBm. It should be between [-6.0, 19.0] dBmAt 112.0km: OSNR at the RECEIVER is 11.786909021817955 dB. It is below the tolerance plus margin 7.0 dB + penalties 7.1 dB = 14.1 dB
#12:0(Málaga->Sevilla) (Málaga -> Sevilla)	69	0	0	29.34	-13.8	3.4	
#13:0(Málaga->Murcia) (Málaga -> Murcia)	141	1	0	14.83	-10.74	4.83	At 53.0km: Power at the EDFA output is (if all WDM channels were active) 19.1 dBm. It should be between [-6.0, 19.0] dBmAt 141.0km: OSNR at the RECEIVER is 11.697761742145712 dB. It is below the tolerance plus margin 7.0 dB + penalties 7.1 dB = 14.1 dB
#14:0(Murcia->Valencia) (Murcia -> Valencia)	78	0	0	29.21	-15.6	3.6	
#15:0(Murcia->Málaga) (Murcia -> Málaga)	141	1	0	14.93	-10.99	4.83	At 51.0km: Power at the EDFA output is (if all WDM channels were active) 19.05 dBm. It should be between [-6.0, 19.0] dBmAt 141.0km: OSNR at the RECEIVER is 11.72392245954801 dB. It is below the tolerance plus margin 7.0 dB + penalties 7.1 dB = 14.1 dB

Route #	Length (km)	# EDFAs	# PCs	OSNR total (dBm)	Power per WDM channel (dBm)	Polarization Mode Dispersion (ps)	Warnings
r0 (id 4790) (Zaragoza -> Sevilla)	290	3	0	2.45	0.26	6.94	At 53.0km: Power at the EDFA output is (if all WDM channels were active) 19.15 dBm. It should be between [-6.0, 19.0] dBmAt 225.0km: Power at the EDFA input is (if all WDM channels were active) 8.3 dBm. It should be between [-20.0, 2.0] dBmAt 225.0km: Power at the EDFA output is (if all WDM channels were active) 25.3 dBm. It should be between [-6.0, 19.0] dBmAt 290.0km: OSNR at the RECEIVER is 3.895402880007181 dB. It is below the tolerance plus margin 7.0 dB + penalties 7.1 dB = 14.1 dB

Fig. 27. Resultados de la ejecución del informe de Net2Plan.

5 MANUAL DE USUARIO

El manual de usuario se dividirá en dos secciones, cada una dedicada a una de las posibles ejecuciones desarrolladas durante el proyecto y que han sido detalladas en apartados anteriores.

5.1 USO DEL INFORME NET2PLAN-TIP-OPTICALIMPAIRMENTS-OLE-REPORT

Estas instrucciones se corresponden con la primera implementación detallada en la sección 3.2, y permiten obtener una copia del proyecto y ejecutarla en la máquina local, para propósitos de desarrollo y testeo.

5.1.1 Prerrequisitos

JAVA RUNTIME ENVIRONMENT

Dado que el trabajo se ha desarrollado sobre Net2Plan, herramienta basada en Java, es necesario tener descargado e instalado el JRE correspondiente en la máquina antes de poder ejecutar el programa. Esto puede realizarse desde la siguiente dirección web.

<http://www.oracle.com/technetwork/pt/java/javase/downloads/jdk8-downloads-2133151.html>

PYTHON 3

Tal como se ha explicado anteriormente, el software GNpy está escrito en Python y, por tanto, necesitaremos tener instalados tanto el mismo como las librerías matemáticas empleadas en GNpy.

La instalación difiere según el sistema operativo del host.

PARA USUARIOS DE WINDOWS

1. Descargar el instalador de Python 3 desde la página web oficial.

<https://www.python.org/downloads/>

Una vez descargado, ejecutarlo marcando las opciones para la instalación de PIP y la correspondiente a la adición de Python3 al path de sistema.

2. Descargar las librerías *numpy* y *scipy*. Las versiones para Windows pueden encontrarse en la siguiente dirección web:

<http://www.lfd.uci.edu/~gohlke/pythonlibs/>

Se debe seleccionar la apropiada para el Sistema operativo de la máquina y la versión de Python instalada. Por ejemplo, en el caso de Windows x64 y Python 3.6, debería ser '*numpy-1.13.3+mkl-cp36-cp36m-win_amd64.whl` and `scipy-1.0.0rc1-cp36-cp36m-win_amd64.whl`*'.

3. Instalar la librería *matplotlib* desde PIP. Para ello se debe ejecutar en la consola de comandos una línea similar a la que se describe a continuación¹.

```
python3 -m pip install matplotlib
```

PARA USUARIOS DE SISTEMAS UNIX

En sistemas basados en Unix el proceso para obtener estos paquetes es más sencillo, simplemente deben ejecutarse los comandos² descritos a continuación y se obtendrá una instalación funcional de Python y las librerías necesarias.

```
sudo apt-get install python3
sudo apt-get install python3-pip
sudo python3 -m pip install matplotlib
sudo python3 -m pip install numpy
sudo python3 -m pip install scipy
```

5.1.2 Instrucciones de instalación

Una vez contamos con todos los prerrequisitos en nuestra máquina, podemos pasar a la instalación de los programas propiamente dichos que ejecutarán nuestro informe.

GNPY

Descargar la herramienta GNpy desde el GitHub del Telecom Infra Project en la siguiente dirección web:

https://github.com/Telecominfraproject/gnpy/tree/feature_polito1

Descomprimirla e instalarla con el siguiente comando:

```
python3 -m pip install {path-to-GNPY}
```

¹ Nótese que el nombre del comando puede ser 'python' o 'python3' dependiendo del sistema.

² Al igual que en Windows, en sistemas basados en Unix el comando también puede variar de 'python' a 'pyhton3'.

NET2PLAN

Descargar la última versión de Net2Plan desde GitHub en la siguiente dirección:

<https://github.com/girtel/Net2Plan/releases>

Al ser una aplicación java, para ejecutarlo no tenemos más que descomprimirlo en una ruta y hacer clic en el fichero *Net2Plan.jar* o, desde la consola:

```
java -jar {Net2Plan-folder}/Net2Plan.jar
```

Por último, para acceder al plugin que nos permitirá incorporar el informe de desempeño en la herramienta, deberemos ir al menú superior y en la pestaña ‘Tools’ seleccionar ‘Offline network design & Online network simulation’, o simplemente por atajo de teclado presionar *ALT+I*.

5.1.3 Breve guía para la interfaz gráfica de usuario de Net2Plan

Una vez hemos realizado los pasos anteriores accedemos a la herramienta de simulación y diseño de redes offline de Net2Plan. En los pasos descritos a continuación se detalla el flujo de acciones necesarias para ejecutar de manera sencilla el informe desarrollado en este proyecto.

1. Descargar una topología de ejemplo y el informe en *.jar* desde:

<https://github.com/girtel/Net2Plan-TIP-OpticalImpairments-OLE-Report>

2. Cargarla mediante el botón ‘Load a network design’

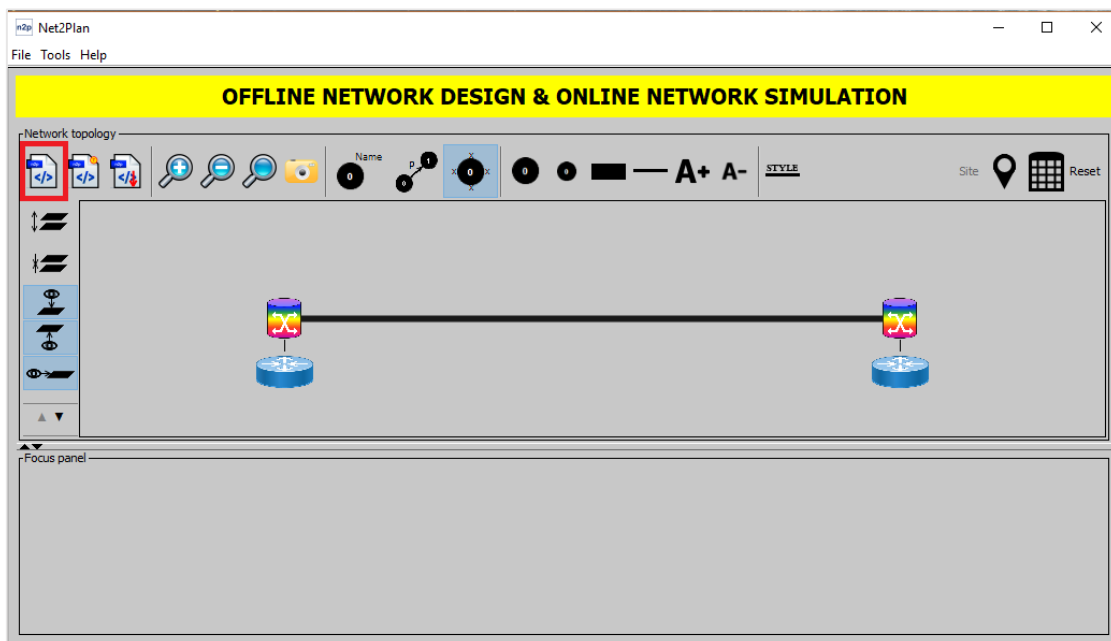


Fig. 28. Ventana principal de Net2Plan.
El primer botón a la izquierda es el correspondiente a la carga de topologías.

Para otras topologías distintas a la de ejemplo, seguir la creación de topologías habitual en Net2Plan e incorporar los atributos de enlace descritos en la sección 5.2.

3. En la ventana ‘Design tables and Network Control’, ir a la pestaña ‘View reports’ y cargar el informe descargado en *.jar*.
4. Cambiar el parámetro de entrada ‘*ole_GNPY_basePath*’ por la ruta en la que se ha instalado GNPy haciendo doble clic en la casilla y editando el texto.
5. Si fuera necesario, cambiar el parámetro de entrada ‘*pythonCommand*’ a ‘*python*’ o ‘*python3*’ dependiendo del comando de Python3 en la máquina local.

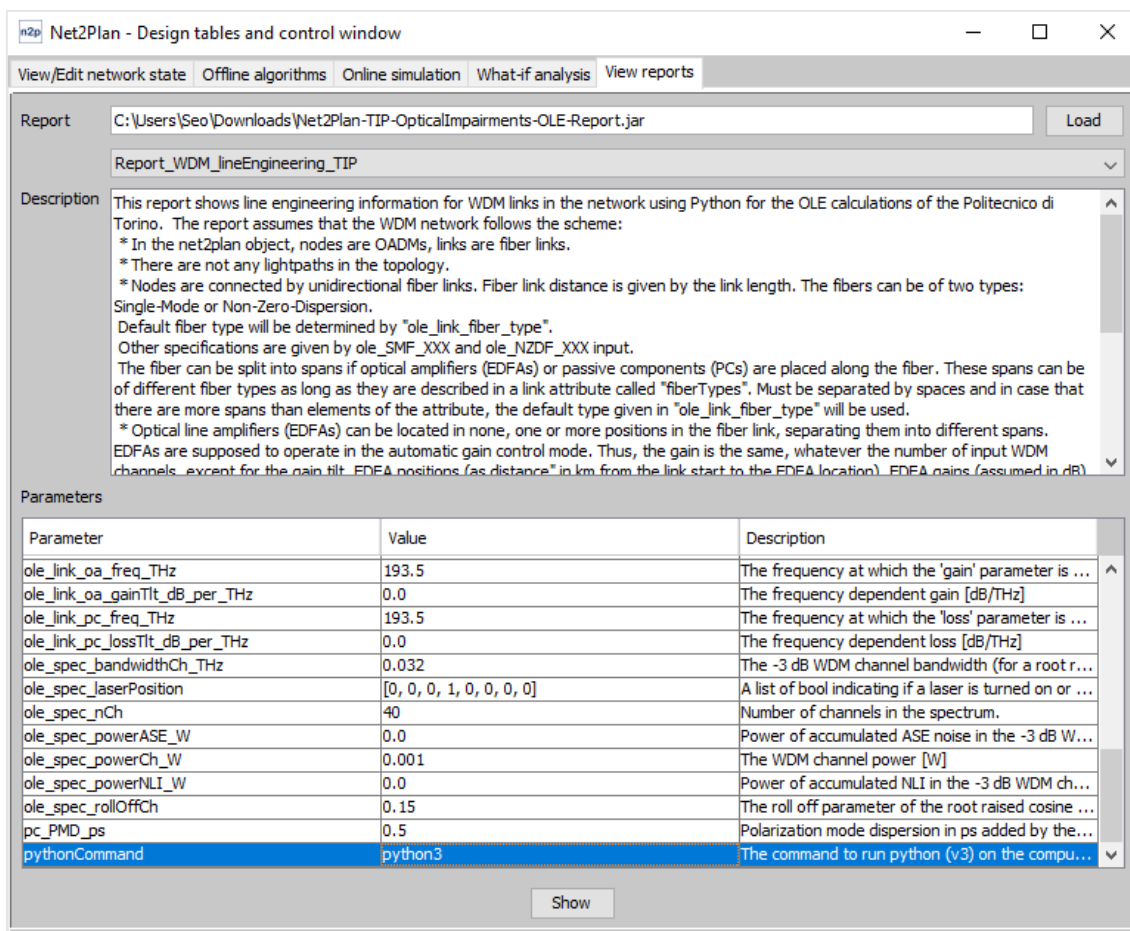


Fig. 29. Ventana 'Design tables and network control' de Net2Plan.

6. Clic en ‘*Show*’ y esperar unos segundos mientras se realizan los cálculos.
7. Una vez aparezcan las gráficas podremos cerrarlas y hacer clic en ‘*View in navigator*’ para ver el detalle completo del informe generado.

5.2 USO DEL INFORME LINE ENGINEERING DE NET2PLAN

En esta ocasión, al estar incluido en el propio Net2Plan no son necesarios prerequisites ni instalaciones externas como en el caso anterior.

Una vez tengamos una copia local de Net2Plan en nuestra máquina, simplemente cargamos una topología con el mismo botón ‘Load a network design’ utilizado anteriormente.

La red cargada deberá tener nodos y enlaces WDM, contando con los siguientes atributos en los enlaces:

- **fiberTypes**. Una cadena de texto con los nombres de los tipos de fibra de cada span separados por espacios. En el caso de no existir este atributo o tener menos posiciones que spans hay en el enlace, se empezará utilizando los que estén disponibles y el resto se tomará el valor por defecto especificado en los parámetros de entrada del informe.
- **‘edfaPositions_km’, ‘edfaGains_dB’ y ‘edfaNoiseFigures_dB’**. Cadenas de texto con los números de las posiciones en kilómetros del enlace en la que están situados los amplificadores y sus ganancias y figuras de ruido en orden y separados por espacios. El report comprobará que exista el mismo número de elementos en las tres listas.
- **‘pcPositions_km’ y ‘pcLosses_dB’**. Cadenas de texto con los números correspondientes a la posición en kilómetros y pérdidas en dB de los componentes pasivos del enlace. Se comprobará que exista el mismo número de elementos.

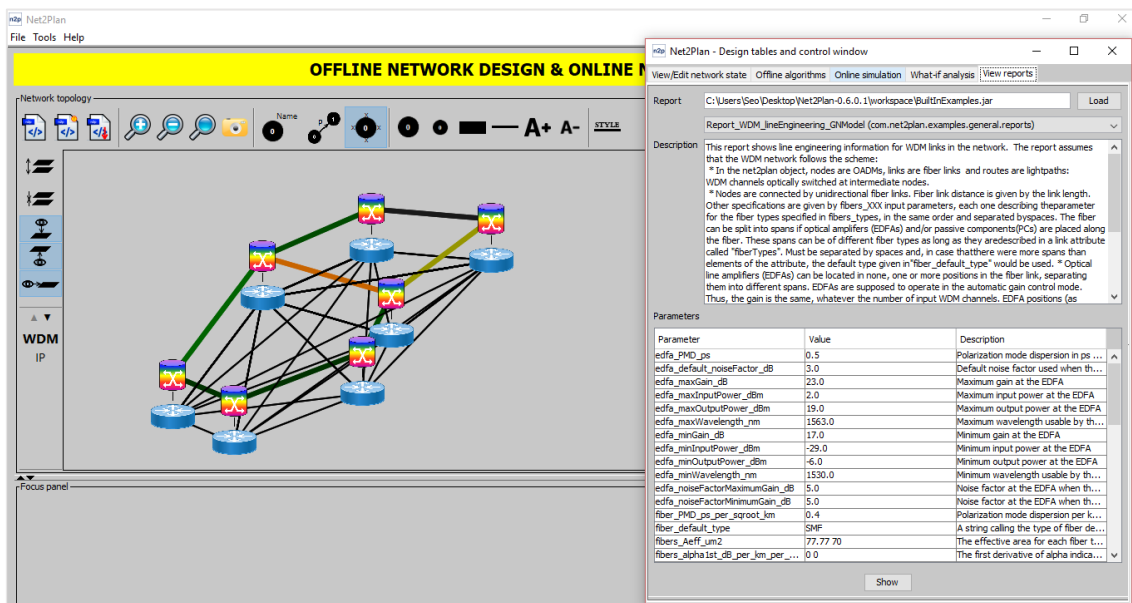


Fig. 30. Topología cargada y lista para ejecutar el report de Net2Plan.

Una vez la tenemos cargada, en la pestaña 'View reports' dejamos cargado el *BuiltInExamples* y seleccionamos del desplegable el Report_WDM_lineEngineering_GNModel. .

Para ejecutarlo, simplemente debemos hacer clic en 'Show' y obtendremos los resultados del informe.

6 CONCLUSIONES

El desarrollo de este Trabajo ha supuesto un avance en la manera en que se tratan los datos de las redes y la comodidad a la hora de calcular informes de desempeño. En la comunidad científica no existen muchas herramientas de diseño y simulación de redes, y menos aún que ofrezcan tantas posibilidades al usuario de manera comprensible e inmediata. Realizar informes de desempeño mejorados para esta herramienta permitirá estudios en profundidad de redes de telecomunicaciones de manera más exacta y eficiente.

Este trabajo ha supuesto un gran estudio en materias de fibra óptica, así como en la implementación de cálculos complejos en diversos lenguajes de programación, obteniéndose finalmente dos implementaciones individuales del objetivo inicial: una enmarcada en Python y empleando la herramienta GNpy, que obtiene datos de gran precisión para redes de WDM y otra sin necesidad de instalaciones adicionales que obtiene los mismos datos más rápidamente.

Como líneas futuras de este trabajo existe el estudio en mayor profundidad de los efectos no-lineales, así como implementar mejoras en Net2Plan que permitan el diseño de las redes de testeo de manera más sencilla y, como consecuencia de ambos puntos anteriores, mejorar los cálculos para hacerlos más eficientes una vez se disponga de las herramientas adecuadas.

7 BIBLIOGRAFÍA

- [1] G. P. Agrawal, *Fiber-Optic Communications Systems*, Third ed., Wiley-Interscience, 2002.
- [2] TIP, «Open Optical & Packet Transport website,» 2018. [En línea]. Available: <https://oopt.telecominfraproject.com/>.
- [3] Newport, «Fiber Optics: fiber basics,» *Photonics Technical Note* , nº 21.
- [4] International Telecommunication Union, *Optical Fibres Cables and Systems: ITU-T Manual*, 2009.
- [5] A. Carena, V. Curri, G. Bosco, P. Poggiolini y F. Forghieri, «Modeling of the Impact of Nonlinear Propagation Effects in Uncompensated Optical Coherent Transmission Links,» *Journal of Lightwave Technology*, vol. 30, nº 10, 15 Mayo 2012.
- [6] P. Pavon-Marino, «An open source network planning tool for bridging the gap between academia and industry,» *IEEE Network*, 29 Mayo 2015.
- [7] P. Poggiolini, «The GN Model of Non-Linear Propagation in Uncompensated Coherent Optical Systems,» *Journal of Lightwave Technology*, vol. 30, nº 24, pp. 3857-3879, 15 Diciembre 2012.
- [8] P. Pavón Mariño, «Net2Plan,» [En línea]. Available: <http://www.net2plan.com/>. [Último acceso: Septiembre 2018].
- [9] P. Pavón Mariño, «Net2Plan User's Manual,» 2018.
- [10] Grupo de Investigación de Redes de Telecomunicaciones, «Net2Plan,» [En línea]. Available: <https://github.com/girtel/Net2Plan/>.
- [11] R. Billington, «Effective Area of Optical Fibres - Definition and Measurement Techniques,» National Physical Laboratory, Centre for Optical and Environmental Metrology.
- [12] OOPT-PSE Telecom Infra Project Group, «GitHub OOPT-PSE TIP,» [En línea]. Available: <https://github.com/Telecominfraproject/oopt-gnpy/>.
- [13] Telecom InfraProject - OOPT PSE Group, «GNpy's documentation,» [En línea]. Available: <https://gnpy.readthedocs.io/en/develop/>.
- [14] TIP, «Telecom InfraProject website,» 2018. [En línea]. Available: <https://telecominfraproject.com/>.

- [15] G. Chauvel, «Dispersion in Optical Fibers,» Anritsu Corporation, 2008.