A simulation program for optical networks

Schuman, E.

Award date:
1994

7250

Technische Universiteit Eindhoven

University of Technology
Den Dolech 2
P.O. Box 513
5600 MB Eindhoven, The Netherlands
Telephone: (040) 479111

Faculty of Electrical Engineering
Telecommunications group

A Simulation Program for Optical Networks

E. Schuman

December 13, 1994

REPORT OF GRADUATION STUDY

COACH: ir. H.P.A. v.d. Boom
SUPERVISOR : prof. ir. G.D. Khoe

The faculty of Electrical Engineering of the Eindhoven university of technology does not accept any liability for the contents of student and graduation reports

Summary

In this report a simulation program for networks of interconnected optical nodes is described. Optical nodes are based on a Local Area Network, consisting of an optical star, to which an optical cross connect overlay is added. WDM and coherent transceivers are connected to the star, and the node is provided with a wavelength and allocating scheme. The optical switches, WDM devices and optical amplifiers in the overlay are used to transfer signals between users connected to the Local Area Network and users in distant nodes.

A model of an optical node is developed in which the components are implemented as objects. The use of objects ensures a clear and flexible structure, which can easily be extended.

Using this model, a computer program is developed, to construct and simulate a network of several nodes. Distance between nodes, the number of connections and the components' properties are among the parameters that can be set. At this stage the emphasis is on the power budget calculations, but other types can be added to the program in the future. The program is written for the MS-Windows environment and is object-oriented. This gives it a clear and consistent user-interface and makes extensions to the computer model possible. Simulation results are displayed in graphs and can be exported to other programs for further use. Results of computer simulations are compared with similar simulations in literature.

An implementation of the optical node's Control and Management section is described, which is responsible for the routing of signals, and establishing and monitoring connections between users.

Contents

CHAPTER 1

Introduction

In modern telecommunications, optical communication systems have become very important. This is mainly due to the high bandwidth and large transmission distance that can be achieved. Aside from use in normal point to point connections (e.g. telephone), optical communication systems are also being used in short range high capacity networks.

Optical LANs are used in those cases, where high bandwidths are required. In a TV studio, for example, High-Definition TV signals are transmitted between editing recording rooms and editing facilities. A factory plant and medical environment are other places where high capacity LANs are used. Most of these are based on the combination of passive and transparent optical star structures and wavelength domain multiplexing techniques. LANs are used to communicate over short distances, between stations located in the same building. Traffic conducted between stations is frequent in nature, but short.

It is clear that there will be a need towards the broadband interconnection of LANs. For this purpose an "Optical cross connect node overlay" is proposed in [1] and [2], describing a way of building a network of optical nodes. Such an overlay would provide a transparent optical connection between stations of different nodes, and a control system that would regulate traffic between nodes, for example by allocating wavelengths and channels. The network could span distances of hundreds of kilometers, accommodate many nodes, each with several hundreds of stations.

To get an idea of the feasibility of such a complex system simulations will have to be done. In this report a simulation program for a small optical network consisting of nodes and overlays is described. The program lets the user design a small network, indicate and set a number of variables and run simulations on it. The program deals mostly with the calculation of signals in the network, but the last part of the report describes a simple Control and Management system that could be used. The first part describes optical networks networks and their components.

CHAPTER 2

An optical cross-connect node overlay

2.1 INTRODUCTION

An optical node described by Hill [3] consists of a passive, transparent, optical star structure, which connects the local users, and an optical cross connect node (OXN). Interconnections to other nodes can be made via optically transparent or non-transparent link as needed. In this chapter the node and its components are described in general. In the next chapter a model of a simplified node is described. Figure 2.1 gives an example of an optical node, which will be described in the next section.

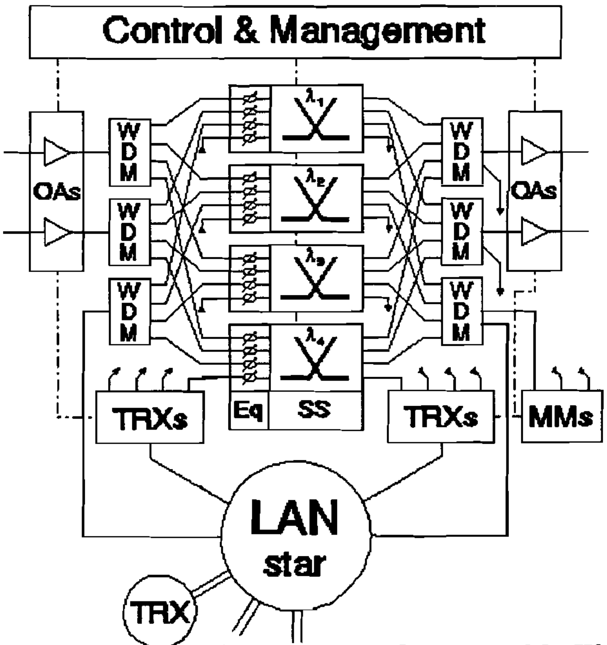


FIGURE 2.1: *An optical node. OAs: Optical Amplifiers; WDM: Wavelength Division Multiplexer; TRX: Transceiver; Eq: Optical Equalizer; MM: Monitor Module*

2.2 The node

The OXN consists basically of integrated Wavelength Division Multiplex (WDM) modules and wavelength dedicated space switches which are used to route optical signals. WDM devices are used because integrated optical switches are optimized for operation at just one wavelength. Optical equalizers may be added to compensate the non-uniform gain profiles of the optical amplifiers. Channel routing is achieved through a combination of wavelength choice (tunable filter or local oscillator wavelength) and space switch setting.

Each space switch is used for only one wavelength (or wavelengths within a small band) or optical carrier (or a group of closely spaced optical carriers). It is therefore possible to optimize the space switches and equalizers for a specific wavelength. This is an advantage when integrated optical components are used.

The number of channels is not fixed, it will depend on the number of stations connected to the node, the amount of traffic and the number of possible routes between transmitter and receiver.

The use of single mode optical fibre enables the wavelengths in the range of 1200-1600 nm to be used. Optical amplifiers limit the wavelengths much more, to about 1500-1550 nm.

Several connections inside the node are possible. For a simple link, it is common to use a transmitter laser and a Local Oscillator laser with fixed frequencies. In a distribution system, each heterodyne receiver can be tuned to any of the incoming signals. In a multi-access system, each of the transmitters and receivers can be tuned to each other to establish a communication channel. Several channels can exist at the same time without disturbing each other owing to the perfect filtering properties of the receiver. If receivers and transmitters can be tuned to an incoming signal in a very short time, packet switching is possible as well. At this stage, it is assumed that a connection between receiver and transmitter remains in place until it is terminated by caller or receiver.

Because of the attenuation in the fibre path and in the components, optical amplifiers may be needed to boost the signal level at input and output.

A Control and Management system is provided for the allocation and control of wavelength channels. This is connected directly to the incoming and outgoing fibres. Transceivers connected to the LAN itself can be used to monitor all signals available in the optical star. Monitor Modules are provided to control the level of the signals.

Non-transparent traffic is possible if extra receivers and transmitters are connected to the optical star. This may be needed if no channel can be found that is free in all the nodes along a signal's route. The Control and Management system is described in more detail in chapter 6.

2.2.1 A short description of the parts

Star coupler

The Local Area Network consists, for example, of a passive optical transmittive star, built with directional couplers. The star structure puts all transmitted signals together and then distributes them equally among the receivers. One input and output are connected to the OXN, the others are available for transmitters and receivers. Figure 2.2 shows an example of an optical star structure with 4 couplers.

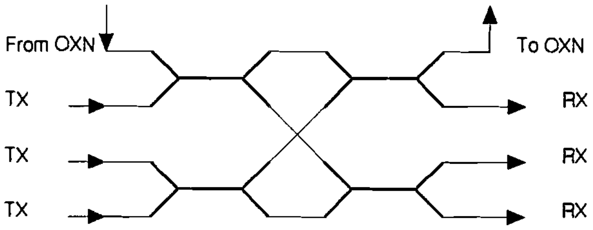


FIGURE 2.2: *An optical transmittive star*

Wavelength (De)-Multiplexer

A WDM is used to separate various signals in a fibre according to their wavelengths. The number of outputs of a WDM determines the number of "bands" these signals are put in to. Each output of the WDM specifies a band with a certain bandwidth and central wavelength. This is also called a "channel". The signals are transported from input to all outputs, taking into account the wavelength dependent attenuation of each channel. So a signal will appear at all output fibres of a WDM, once as a real signal, with a attenuation of about 2 - 3 dB, the other times as crosstalk, with losses < -16 dB. A WDM makes it possible to switch each channel (one or more wavelengths in a narrow band) separately. Such a WDM could be a phased array multiplexer.

An example is the WDM described in [4], which has four channels with a 2nm channel spacing and operates at 1540 nm.
It works by means of a dispersive waveguide array with a constant path length difference between adjacent arms, which will direct the signals in each channel to a specific output waveguide. Some characteristics of this WDM are:

on-chip attenuation	5 dB
crosstalk	< -25 dB
channel spacing	1.8 nm
full-width at half maximum	0.7 nm

Optical switches

Optical waveguide switches can be realized with directional couplers, digital optical switches, or Mach-Zehnder interferometric devices. An example of the last is a polarization-insensitive low-voltage optical waveguide switch using InGaAsP/InP four-port Mach-Zehnder interferometer, described in [5].

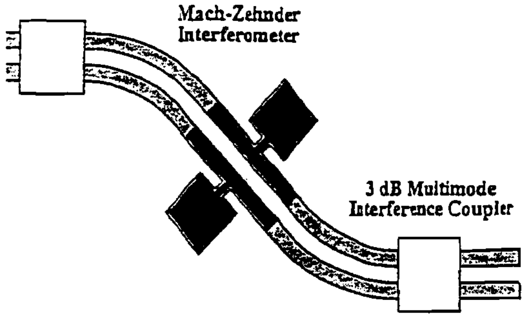


FIGURE 2.3: *A four port Mach-Zehnder interferometer optical switch*

Figure 2.3 shows the layout of a 2 X 2 optical waveguide switch.

Some characteristics of this switch are:

	total on-chip insertion loss [dB]	crosstalk [dB]
TE-polarized source	2.5	16
TM-polarized source	3.2	16

The insertion loss is the attenuation a signal encounters from an input to the selected output of the switch. Crosstalk is the attenuation from an input to any other output. The values above are measured with a light source of 1530 nm connected to one input. A voltage of -5 to -10 V determines the state of the switch.

Fibre

Nodes are connected by optical fibre. In most cases, monomode fibre will be used, because of the much lower dispersion. Only the fibre's attenuation and other properties like material and waveguide dispersion will then effect the maximum distance between nodes. In most cases, attenuation will be the limiting factor.

Receivers and transmitters

Various types of optical transmitters and receivers can be used,[1], but heterodyne systems appear to have much advantages over, for example direct detection receivers:

- They can be used with optical multicarriers (channels) that are allocated in a flexible way. They can be adjusted to changing wavelengths.
- They are not restricted to the passband of optical filters or amplifiers. Heterodyne systems can be used over the entire wavelength range offered by the single-mode optical fibre, from 1250 to 1600 nm.

- They give a superior receiver sensitivity and related transmitter span. An optical receiver operating without the heterodyne method with a similar sensitivity has to be equipped with a optical fibre pre amplifier and some other special measures.

System details depend on modulation formats such as amplitude shift keying (ASK), frequency shift keying (FSK), and differential phase shift keying (DPSK). Bit rates ranging from 155 Mb/s to 2.5 Gb/s are possible.

Of the above mentioned, FSK is the most common modulation format. Modulation of the transmitter laser frequency can be obtained by changing the current through the semiconductor laser. Principal parameters for a FM transmitter are linewidth (frequency spectrum), FM response and output power. Typically these have values of 0.50 MHz (linewidth), 100kHz-10 GHz (FM response) with an output power of 40 mW.

Continuous tuning of the Local Oscillator in receivers and transmitters is preferred, because only in that case it is possible to tune at incoming signal frequency within a given range. This can be done electrically and thermally. Electrical tuning is fast and gives a range of about 7 nm, thermal tuning is much slower but has a larger range.

2.2.2 Control and Management

Its function is in general to control the traffic both inside and outside the node. For example: to allocate channels to stations who wish to open a connection, to close a channel when a connection is no longer needed.

To do this, there has to be another way of communication between stations and C & M and between two C & Ms about the duration of the transmission and the wavelengths used.

A simple solution would be to use one channel of a different wavelength (e.g. 1300 nm). This has the advantage that it is easy to separate this signalling information from the normal traffic (with a WDM).

It is necessary to distinguish between local signalling and inter-node signalling. The first case is between a station and the C & M. The C & M maintains a strict control of the transmitters because the number of wavelengths is a limited resource.

Note that contrary to "normal" LANs, traffic between nodes is not of a packet nature but is a point-to-point connection. A full duplex connection between stations therefore occupies two wavelengths. They cannot be used for anything else before the connection is terminated. If the nodes are used in a "broadcasting environment" for the transport of TV signals, this can take quite some time. If no free wavelength can be found, a station will have to wait.

The second case is more about cooperation between nodes and negotiation about paths and wavelengths used. Nodes will have to exchange information about channels and routes. The possibility of wavelength switching must be taken into account. A more detailed description of the C & M and a possible implementation is described in chapter 6.

2.2.3 Non-transparent connections

In some cases it will not be possible to choose a channel (wavelength) that is not used by any node along the link. In this case, it may be necessary to change the wavelength of a signal at some point. For this purpose extra receivers and transmitters are located in the star. To change the wavelength of a signal passing through a node, it is diverted to a receiver/transmitter combination. After being transmitted to the star, it is switched to one of the node's output fibres. Note that for each non-transparent connection, a receiver/transmitter pair has to be available, as well as a free wavelength channel.

It is not clear how much of these facilities must be provided. In an article about Wavelength requirements and Survivability in WDM Cross-connected Networks [6], it is shown that the difference in wavelength requirements between "Virtual Wave Paths" (paths with a wavelength translation), and ordinary Wave Paths is very limited. Therefore wavelength translation is not necessary in every node. A hybrid solution, where wavelengths are translated only in specific nodes, e.g. the nodes where the paths are electrically regenerated, gives no significant drawbacks.

Chapter 3

Simulation of an optical node

3.1 Introduction

In this chapter a simple model of an optical node will be described. This model consists of a number of objects. The method of simulation will be described, together with a description of the parts of the network and the methods of calculation.

In a network, information flows from a transmitter to one (or several) receivers. It flows through many components like couplers, splitters, amplifiers, etc. before it reaches its destination.

In the simulation we want to study the performance of the network while changing some of its properties.

Some of the variables are:

- the number of users in a node,
- the distance between nodes,
- the use of amplifiers between nodes,
- power and wavelength of transmitters,
- the routing of signals.

The model makes it possible to calculate levels of signals anywhere in the network. It takes into account losses and crosstalk of the node's components and attenuation of the fibres. Amplifiers are also included.

As an example and as a start of the development of the simulation program the optical node of chapter 2 will be taken. One of the objectives was to maintain the structure of the original node as closely as possible. Therefore the modeled node also exists of a passive optical star, WDMs, switches, etc. This was done to make the set-up as flexible as possible. Different (larger) nodes can be constructed from the same parts. Modifications and extensions can be added very easily, because the structure is very clear and the way of exchanging data is the same throughout the network.

Limitations of the model:
As far as it is developed now, the model has some limitations:

- Signals in the model are available only in the frequency domain. It is not yet possible to include signals as a function of time, for example the shape of pulses at the input fibres to account for dispersion.
- Some components in the model are very elementary. The models of transmitters and receivers will have to be extended to make the calculation of the Bit-Error rate possible.
- Some components are not yet available. To provide non-transparent connections, transmitter/receiver pairs or frequency converters will have to be added.

The following describes the way information in the model is stored and handled. To make this clear, it is necessary to first say something about the programming concepts behind it.

3.2 Signals

The most basic form of information in the model is the *signal.* A signal is "transmitted by something" and "received by something else". Such a signal has the following properties:

1. a wavelength
2. a (power)level
3. a bitrate
4. a destination

Bandwidth can be added to the list of properties. It is at this stage not defined, but in some parts (amplifiers) assumed to be 1 nm. The last two items are added for administrative purposes only, to help guide the signals through the network and to calculate the Bit-Error rate at the receiver. Other information, for example the modulation format can be stored here as well, if this is required.

These signals are stored in a list. A list can be seen as the equivalent of a fibre. It has a number of signals with different power levels and wavelengths passing through it. The of use (linked) lists makes the handling of signals very flexible. In this way, any number of signals can be stored in a list. If needed, a spectrum of a certain shape can be modeled using several signals that are very close together. On the other hand, no memory is wasted when the number of signals is small. On a PC, the number of signals in the model is limited by memory, and is in the order of several thousands. A list is the most simple form of *object* in the model. Object are structures used in programming that contain both data and programming functions into one part. An object like *LIST* contains functions, e.g. *append()* and *delete()* to add and delete signals to the list , but also data (the signals, the number of signals). Only certain information in the object is available to the programmer. It is possible to shield some sensitive variables and structures. For example, the pointer to the list of signals, which holds the address where the list is stored in memory is hidden from the programmer. Changing the contents of the pointer would mean losing the list of signals and probably crashing the program.

Defining the model
One level higher are the of the model's components. Objects like star, WDM etc, are used to simulate their corresponding components. Most objects are built the same way: they read signals from input lists, these signals are combined, copied or discarded. The results are then written to output lists and sent to the next object.

The "next" object is the one that this object is connected to. Connections exist in the form of pointers to lists of signals. As shown in figure 3.1, objects are linked together trough the output lists they contain. Each component generates and contains lists of signals. Because these lists will always start at a fixed point in memory, the address of the output list is a good point to connect the next object to.

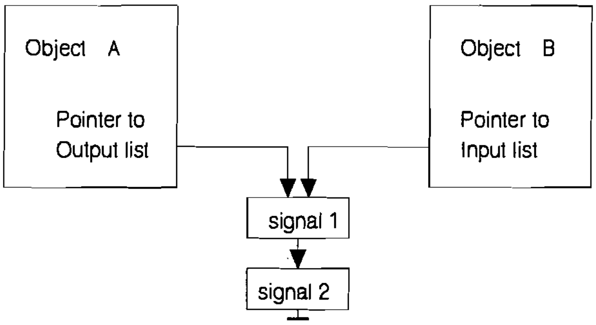


FIGURE 3.1: *Connections between objects*

The next object requires a list of signals for its input. Giving not a list, but a pointer to a list has some advantages:

- It saves memory. The output list of one object does not have to be copied to the input list of another object.
- It is a more direct connection. There are no duplicates of the same list in the network.

The contents of a list may change but the address (starting point in memory) will remain the same throughout the program run. In this way, small objects are used to make larger ones.

During calculation, the *calculate()* function of the node is executed. As a node is a hierarchical structure, this function first calls all *calculate* functions of the objects it contains en then performs it's own calculations. This ensures that all *calculate* functions are called in the right order.

Loops in the network.
Normally, every calculating function is executed only once. If the network contains loops, it may be necessary to re-calculate the entire network several times until some stable situation is reached.

Signal limits
To avoid unnecessary calculations, signals with a level below a certain value are discarded. This lower signal limit can be set in the program, and is a way to control the accuracy of calculations.

3.3 Description of the objects in the node

The following describes the various objects the node contains. A set of simple objects is chosen from which a node can be constructed:

- a passive optical star object,
- WDM objects with one input and a variable number of outputs,
- an optical switch object with 2 inputs and 2 outputs,
- reversed WDM objects with a variable number of inputs and one output,
- optical amplifier objects,
- receiver and transmitter objects.

These combined form a node object with 2 input and 2 output fibre connections. Fibre objects are then used to link nodes together. The following sections describe the objects in detail.

3.3.1 Star

The star object is the part in the node the transmitters and receivers are connected to. It receives its inputs from one output fibre of the OXN and the transmitters. To accommodate a variable number of transmitters, each transmitters adds its transmitted signal to the OXN output fibre list. Its output is connected to a WDM. The star's attenuation depends on the number of directional couplers it contains. N transmitters can be connected to a first stage of $\frac{N}{2}$ 3 dB couplers. In total, there are $\log_2 N$ stages to distribute the transmitter's signal to all N outputs. Therefore the number of couplers C is

$$C = \frac{N}{2} \log_2 N \tag{3.1}$$

and the attenuation is

$$L = \log_2 N L_c \tag{3.2}$$

with L_c the "loss" of a single 3 dB coupler. This loss is stored in the object but is assumed equal for all couplers in the star. The coupler has an extra loss L_x as a result of connectors, which makes the total loss L:

$$L = \log_2 N(L_c + L_x) \tag{3.3}$$

3.3.2 WDM

The WDM object transports signals from one input to all outputs, taking into account a different wavelength dependent loss for each output. The number of outputs corresponds to the number of channels in the node, which is variable. Each of the WDM's outputs has a attenuation curve based around a central wavelength λ_c. These central wavelengths λ_c, are evenly distributed in the working area of the WDM, between 1500-1550 nm. The curves themselves do not changed with the number of channels, i.e. the width between the 3-dB points is unchanged. This is done, because probably another WDM device with a different number of outputs has different attenuation curves and it is very easy to assign a different attenuation curve to a multiplexer. For each WDM, the attenuation curves for all channels are assumed to be equal, therefore only one attenuation curve is stored in the object. The curve is read from an *WDM information file.* These files are described in detail in section 4.4. The wavelengths used in the curve are relative to the central wavelength λ_k.

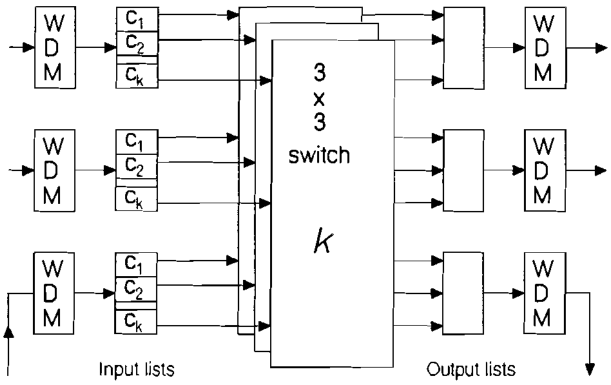


FIGURE 3.2: *Connections between WDMs and switches in a node with K channels*

Because the number of outputs of a WDM is variable, it is easier to store the output signals for all channels in one big list. The signals receive an extra tag to record to which channel they belong. In the next stage, the switch will process signals on a channel by channel basis, so it is vital that this information is not lost. Figure 3.2 depicts the connections between WDMs and switches.

Reversed WDMs
The WDMs used on the "output side" of the node are built the same way. The tagged signals from the switches are processed channel by channel. Wavelength dependent attenuation for each channel is calculated and the signals are moved to a single output. The tags are removed in the process.

```
a  -----X---X---X
        |   |   |
b  -----X---X---X
        |   |   |                   1
c  -----X---X---X                   |
        |   |   |   x : switch    2-x-3
        |   |   |                   |
        x   y   z                   4
```

FIGURE 3.3: *A 3 x 3 non-blocking switch*

3.3.3 Switch

Properties of switches, such as loss and crosstalk, depend on the number of inputs and outputs. Optical switches in the node are defined as a 3-by-3 type, which means that a full non-blocking switch is made of 9 switches. To calculate the loss it is necessary the know the paths a signal can take.

Directly from one input (a,b,c) to an output (x,y,z) , the signals pass through 1 to 5 closed switches. If the signal takes another path, it will travel through at least two more opened switches. See figure 3.3. Each switch has 4 terminals. Either connection 1-4 is open, or connection 2-3. The loss as a result of an open connection is L_o. The loss to one of the other terminals is then L_c.

So the loss L for the switched signal is

$$L_c \leq L \leq 5L_c \tag{3.4}$$

Signals produced by crosstalk are at least $2L_o$ weaker than the switched signal, that is, the loss through crosstalk is

$$L_x > 3L_c + 2L_o \tag{3.5}$$

The signals are transported to the selected output of the switch, taking into account the maximum L, and to the other two outputs using the worst case L_x.

Like the number of channels, the number of 3-by-3 switches in a node is variable. It is easier to implement this as one 3-by-3 by K switch (K number of channels). This switch just processes all channels sequentially. Because in reality each channel has its own switches, there is no crosstalk between channels. The switch object has K outputs, through which signals are sent to the next stage, the reversed WDMs.

As explained before, in the WDMs and switches, the signals are tagged. The channel number is coded into the signal's *destination* field. Figure 3.2 depicts the connections to the switches in the node.

3.3.4 Fibre

In the node a very simple optical fibre model is used. The fibre is assumed to be monomode and only the attenuation is taken into account. The attenuation is

$$L_f = lL_0 \text{ dB (l in km)} \tag{3.6}$$

with L_0 the attenuation in dB/km. The attenuation per kilometer is wavelength dependent, and is stored inside the object. It is given in a curve as a function of the wavelength for some points, and calculated for intermediate points by linear interpolation. The effect of dispersion is, at this stage, neglected. If necessary, it could be added to the model.

3.4 The Optical Amplifier

Probably, in the case of longer distances and more users in each node, the use of amplifiers becomes inevitable. Optical amplifiers are then preferred above electrical ones, because they provide a transparent connection between transmitter and receiver, giving larger bandwidths and thus providing more channels.

EDFAs (Erbium doped fibre amplifiers) can amplify signals in the range of 1500-1550 nm, with gains up to 20 dB. Such amplifiers will have to be placed and tuned carefully, as not to become saturated by large input signals. The difference between signals "passing through" a node and signals that have just been transmitted must also be within certain limits.

Because a signal's gain depends on all input signals, and to take into account saturation and the generated Amplified Spontaneous Emission, a fairly comprehensive model is used.

The "spectrally resolved model" described in [7] gives a good numerical solution to the rate and propagation equations that describe the amplifier. This method is also used and described in [8].

In the model, the light in the amplifier is thought to be propagating as a number of optical beams of wavelength $\lambda_k = c/\nu_k$ from one end of the amplifier's fibre to the other. k is the beam's index and corresponds to a wavelength. All beams have an equal width $\Delta\nu_k$. Both narrowband signals and wideband Amplified Spontaneous Emission (ASE) noise are described in this manner.

The equation 3.8 for the change of power in a beam dP_k/dz is used to calculate the power in the beams P_k. This depends on the state of inversion in the fibre. Equation 3.7 gives the ratio of ratio of the average density of ions in the metastable state $\overline{n}_2$ to the average total erbium density $\overline{n}_t$. The state of inversion depends on the fibre's properties (the gain and absorption spectra) and on the total power in all beams.

Note that beams travel in both directions. The equations must be used repeatedly in both positive and negative z direction, until the situation is stable (derivatives dP_k/dz are near zero).

Boundary conditions at z=0 are given by the signals that are put into the doped fibre. These include the signal of the pump laser (P_p) at a wavelength λ_p. The remaining P_k are 0 because there is no ASE at z=0.

After solving the model's equations 3.7 and 3.8 over z using a trapezium method, the amplified output signals, the residue of the pump signal and Amplified Spontaneous Emission are present at $z = L$.

If a number of simplifications are made (the signals are assumed to be constant in time), the equations of the spectrally resolved, spatially integrated model are

$$\frac{\overline{n}_2}{\overline{n}_t} = \frac{\sum\limits_k \frac{P_k(z)\alpha_k}{h\nu_k\zeta}}{1+\sum\limits_k \frac{P_k(z)(\alpha_k+g_k^*)}{h\nu_k\zeta}} \tag{3.7}$$

$$\begin{aligned}\frac{\mathrm{d}P_k}{\mathrm{d}z} &= u_k(\alpha_k+g_k^*)\frac{\overline{n}_2}{\overline{n}_t}P_k(z) \\ &+ u_k g_k^* \frac{\overline{n}_2}{\overline{n}_t} m h\nu_k\Delta\nu_k - u_k(\alpha_k - l_k)P_k(z)\end{aligned} \tag{3.8}$$

with:

$\frac{\overline{n}_2}{\overline{n}_t}$: inversion parameter
$P_k(z)$: power [W]
α_k : gain spectrum [Np/m]
ν_k : signal frequency [Hz]
ζ : saturation parameter $[m^{-1}s^{-1}]$
g_k^* : loss spectrum [Np/m]
l_k : excess loss [dB]
m : the number of modes
$\Delta\nu_k$: width of frequency band [Hz]
u_k : propagation direction =1 : positive z dir.

Equation 3.7 gives the ratio of the average density of ions in the metastable state $\overline{n}_2$ to the average total erbium density $\overline{n}_t$. Equation 3.8 is an expression for the propagation of beams in the amplifier's z direction.

The saturation parameter

$$\zeta = \pi b_{eff}^2 \overline{n}_t/\tau$$

is the ratio of the linear density of ions to the metastable lifetime. m is 2, being the number of polarization states of the lowest order optical mode that the fibre supports. $\Delta\nu_k$ is set at 125.10^9 Hz, this corresponds to a width of 1 nm at 1550 nm. b_{eff} is the equivalent radius of the erbium dopant in the fibre.

3.4.1 Implementation

Several parameters, among which are the gain and loss spectra are stored in the object. The length L, the saturation parameter ζ, and the pump power P_p can be changed in the program. The amplifier's input signals are stored in the array P_k together with a pump signal (set at 1480 nm). During calculation, the powers $P_k(z)$ and the derivatives $\frac{dP_k}{dz}$ are stored in arrays to speed up the calculation.

The equations are first used in forward direction, from $z = 0$ to $z = L$, and then repeatedly backwards and forwards until the powers P_k have reached a stable value. The available wavelength ranges from 1450-1550 nm, so each "slot" in the array represents a signal with a bandwidth of 1nm. Gain and absorption spectra are normally stored in curve lists in the amplifier object, but are loaded into a sort of cache arrays during the calculation to speed up things.

Using z as an independent variable, after reaching $z = L$ for the last time, all output signals are present in the array P_k and can be put into lists, for further processing.

3.5 ASE and noise in the model

An expression for the Noise Figure F

The ASE output power in an optical bandwidth $\Delta\nu_k$ is (see [8] and [9]) :

$$P_n = 2n_{sp}(G-1)h\nu\Delta\nu_k \tag{3.9}$$

and the Noise Figure:

$$F = \frac{1 + \frac{P_n}{h\nu\Delta\nu_k}}{G} \tag{3.10}$$

with
G the Amplifier gain for some wavelength
ν the signal frequency
$\Delta\nu_k$ the bandwidth, set to 125 GHz

Normally proper transmission is possible in case the coherent receiver input is larger than the specified receiver sensitivity. However, noise contributions due to ASE may degrade the receiver sensitivity. In case of a balanced coherent detection, ASE-ASE beat noise can be neglected and only LO-ASE beat noise will strongly dominate the amplifier induced receiver noise. The amplifier induces a penalty, which is dependent on the amplifier noise figure F, the post-amplifier loss L_{post}, and amplifier gain G. The impact of LO-ASE beat noise is dependent on properties of the receiver itself: the optical signal loss inside the coherent receiver D_{loss}, and receiver degradations caused by noise already present at the coherent receiver D_{noise}. This penalty can then be written as

$$\text{Penalty} = \frac{1}{1 + \mathrm{D_{loss}D_{noise}L_{post}(GF-1)}}\ [\mathrm{dB}] \tag{3.11}$$

where $(0 \leq D_{\text{loss}} \leq 1)$,$(0 \leq D_{\text{noise}} \leq 1)$ and $(0 \leq L_{\text{post}} \leq 1)$.

Because we are foremost interested in the power budget calculations, and the properties of receivers are not clearly defined, the impact of amplifier noise is accounted for by subtracting the penalty from the amplifier output power. Note that in this case, the receiver sensitivity is considered to remain constant while the receiver input power is adapted to compensate for the penalty.

CHAPTER 4

Description of the program

4.1 INTRODUCTION

This chapter describes a program that has been developed to design and simulate a network of optical nodes.
The objective was, to develop a Windows program that:

1. lets the user construct a network consisting of several nodes,
2. is able to calculate signals throughout the network,
3. accepts user input through dialog windows for network settings,
4. displays results of calculations in graphs or lists,
5. can send its output to other programs for further evaluation,
6. can easily be extended,
7. is easy to use.

The program is roughly divided into two parts, a presentation part (user interface), and a part that performs the calculations and contains the network. The first part has to do with windows.

4.2 WINDOWS

The main window, shown in figure 4.1 displays the network of nodes that is being simulated. At the top of the window is a menu structure, which can be used to select actions. The user can select components of the network and change their properties by clicking the mouse inside their image on the screen.

The main menu has the following items:

- *File.* This part lets the user load a designed network from disk or save one. The option *Quit* ends the program.
- *Network.* This is the part that constructs a network by adding nodes and linking fibres. Also the *Global settings* can be reached here.

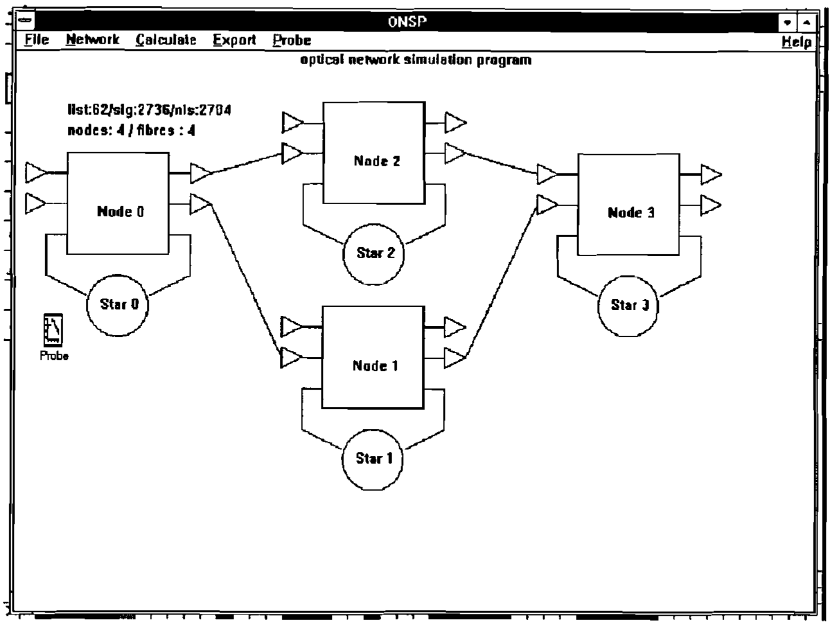


FIGURE 4.1: *main window*

- *Calculate.* The possibilities have to do one calculation with the network in its current state, or do a series of calculations with specific values for certain parameters.

- *Export.* Switch *Dynamic Data Exchange* on or off. This is a method of transporting results of calculations to other programs running under the Windows environment.

- *Probe.* Determines which signal in the network is displayed in the Probe window.

- *Help* This option gives some explanations about the program's use.

Selecting an item in the menu bar will cause a dialog window to appear. In this window more selections and commands can be made about a specific subject.

4.2.1 Creating a network

When the program is started, the network contains only a single node. To add more nodes, and connect these, select to menu item *Network / Add Node.* A dialog window appears to choose the connections of the new node. Currently, nodes can have at most two input connections. For each of the inputs, a list of possible connection points (in general outputs of other nodes) are displayed.

To leave a connection open, choose the option `-nothing-`. A new node is then displayed on the screen, with its connecting fibres.

The new node is initially positioned in the right side of the window, but it can be moved by clicking on the node and using the `position` buttons in the *node dialog* window described below.

It is not yet possible to delete nodes or fibres or change connections between nodes.

4.2.2 The node dialog

To change the settings of a node, or the parts belonging to it, click the mouse on the image on screen. Settings that can be edited are:

- the properties of the components : WDMs, switches,
- the position of the node on the screen,
- functions of the node's amplifiers.

4.2.3 The star dialog

Star-settings can also be changed by clicking on the star image. They include:

- the number of transmitters / receivers in the star,
- the power-level, bitrate, wavelength and destination of the transmitted signals,
- settings for the receivers.

Each node contains four amplifiers, connected to the two input and output fibres. They can be switched on or off through the *node* dialog window. Other settings can be edited by clicking the mouse directly on the amplifier:

- amplifier on or off,
- name of the amplifier information file,
- Pump power,
- amplifier fibre length,
- the saturization parameter,
- the properties of the fibre connected to the amplifier:
  the fibre length,
  the fibre information file (contains the attenuation curve).

These can be set for each amplifier in the network. Fibre settings can only be selected if a fibre is connected to the amplifier.

4.2.4 General settings

Some things that have to do with the whole network can be reviewed in the General settings dialog window. They are:

- the number of nodes, fibres and signals used,
- the number of channels used in WDMs and switches,
- the lower signal limit, below which signals are discarded,
- DDE settings, the location of Quattro Pro for Windows and the filename for the GnuPlot file

4.2.5 Probe window

The Probe window displays a list of signals either as a graph or in print. During the calculation a specific point in the network is "connected" to the probe window. Until another list is connected, any signals in this list are immediately displayed in the Probe window.

The user can choose to display this in text form or as a graph. The graph only displays power level and wavelength, and only the largest signal with a certain wavelength will be visible. In text form, all items are displayed. These options can be selected through the window's *system menu* (the button on the left side of the menu bar). The user can export the list of signals in the window with DDE to e.g. the spreadsheet program Quattro Pro.

4.3 Calculations

Calculations on the network can be performed in two ways:
- only once, with the network in its current state.
- in a series, with different values for selected variables.

In the first case, the nodes and fibres are calculated one by one, in the order of signal propagation. The results of these calculations can be seen in the Probe window.

4.3.1 Series calculations

In this case, a series of values is inserted into certain variables before the calculation takes place. Results of these calculations are displayed in the Graphics window.

This selection of variables is made through the *series calculation* dialog window. Here, the subject the of calculation, a first and a second parameter can be selected.

Currently, the only thing that can be calculated is the signal power at a receiver, so this is always the first parameter. Other subjects could be the BER at a receiver, or the Signal-to-Noise ratio.

All possible choices for the first and second parameters are shown in two listboxes. They include:

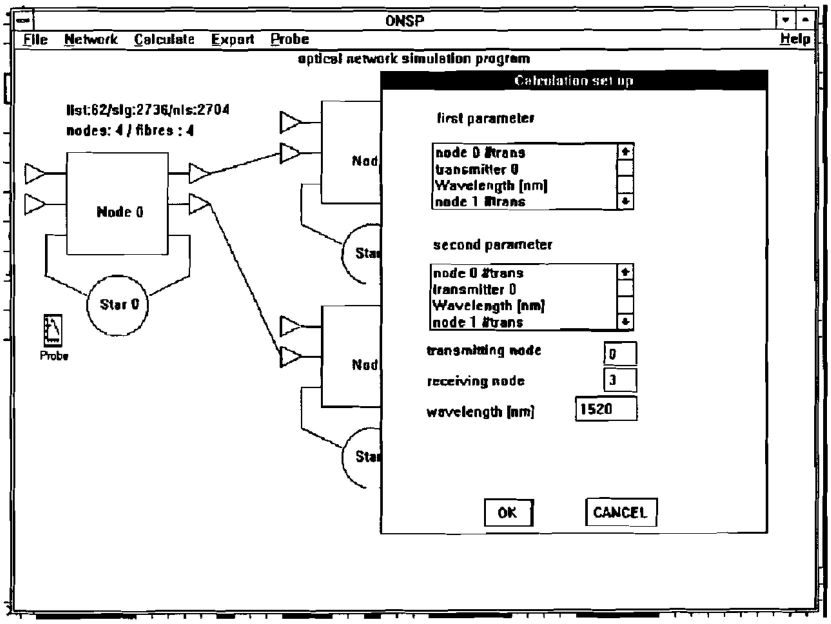
- the number of transceivers for any node,
- signal power for a transmitter,
- the length of any fibre,
- the wavelength of the transmitted signal.

Of course, the first and second parameters must be different. Figure 4.2 depicts the calculation set-up. The receiving and transmitting node can be set, as well as the wavelength of the transmitted signal.

4.3.2 Graphics Window

In this window, the chosen type of calculation is set-up and executed and a graph of the results is displayed. For each of the three values of the second parameter, a graph is drawn of the results of the calculation in which the first parameter is iterated from start till end value.

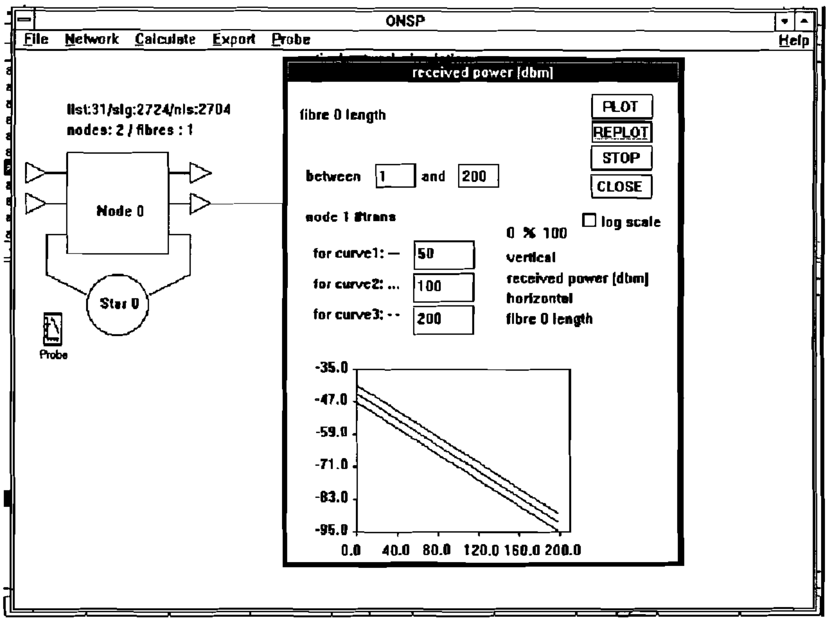
Before the calculation can start, the user has to give a start value and end value for the first parameter and the three values for the second parameter. Then he can choose
Plot to start calculations,
Replot to repaint the screen after calculations have finished,
Stop to stop calculations, or
Close to end the dialog.



list:62/sig:2736/nls:2704
nodes: 4 / fibres : 4

transmitting node 0
receiving node 3
wavelength [nm] 1520

FIGURE 4.2: *Selecting parameters for calculation*



list:31/sig:2724/nls:2704
nodes: 2 / fibres : 1

fibre 0 length

between 1 and 200

node 1 #trans

for curve1: — 50

for curve2: ... 100

for curve3: - - 200

0 % 100

vertical
received power [dbm]
horizontal
fibre 0 length

FIGURE 4.3: *Calculations in the Graphics window*

Choosing *Plot* starts 3 series of 50 calculations, where for each calculation the second parameter has one of the given values, and the first parameter iterates from start- to end value. Three graphs are drawn on a small display window at the bottom of the dialog window. If the DDE option is switched on, the results are sent to Quattro Pro. Figure 4.3 is a picture of this window.

4.3.3 Calculation Order

Calculations start with the `calculate()` function in node 0. This is the first node and it is always present. Each calculating function will first erase the previous signal lists. The calculating functions are then called in the following order:

◇ transmitter
◇ all EDFAs
◇ STAR
◇ all WDMs
◇ SWITCH
◇ all reversed WDMs
◇ all EDFAs
◇ STAR
◇ receivers

first node

◇ fibres
◇ next node

The nodes are calculated in the same order as they were created.

Fibres between nodes will be calculated in the right order, i.e. fibres between two consecutive nodes will be calculated regardless of their numbers.

4.4 Configuration files

The program uses a text file to store information regarding the whole network as well as other files that are particular to one object. The *network file* is loaded and saved through the *File* menu options. This file include global settings for the network, descriptions of the nodes and fibres and their connections, and references to other information files. Each WDM, star, amplifier and fibre in the model can have it's own *information file.* In such a file, the object's more permanent parameters are stored. Some object's files include curves (fibre attenuation curve, amplifier loss and gain curves). These information files must be present in the program's directory. Contents of information files cannot be changed inside the program, but this can be done easily with a text editor. For a description of the network file, see appendix B.

4.5 Dynamic Data Exchange

Quattro Pro for Windows (QPW)
It is possible to transfer results of calculations to other Windows programs. These programs must be able to work as a DDE-server. At this stage, only Quattro Pro for Windows (a spreadsheet) can be used. When this option is chosen, the program will attempt to
- locate and start Quattro Pro for Windows
- request the name of the sheet that is active
- establish a connection to the current sheet in QPW.

At this point, it is possible to send things (values, numbers, text) to cells in the spreadsheet. For each series of calculations, all parameters are sent to QPW, as well as the result of the calculation. Once the results are transferred, it is possible to save them as a spreadsheet or make graphs, etc.

Gnuplot
While this is not really DDE, it is very useful to have the result values in a simple ascii file. Such a file can be used for example, by Gnuplot to make graphs. The name of the file is specified in the `General settings` dialog window, and it contains values in de x,y format, with each pair of values on a single line.

4.6 Windows programs

Windows is not really a multitasking system. It is possible for more than one program to run at the same time, when each program runs only for a short time and then gives control back to the Windows Program Manager. The

Program Manager will then let another program run for a while. Only during long calculations (when optical amplifiers are switched on), does the program regularly give back control to Windows. This enables other programs to run at the same time. If no amplifiers are switched on, calculation time is much less, and the program is interrupted less often. It can be seen that during calculations, other programs run slowly, and also the repainting of windows on the screen takes more time.

During calculation, the program can be run in the background (i.e. as an icon). When one graph has been completed, it will "popup" (display the graph on the foreground window).

The simulation program uses a lot of memory and computing power. It runs comfortable on a PC with a 80486 microprocessor and 4 Mb ram. To run more programs at the same time, more memory is required. The number of nodes is at this stage limited to 10, and the number of fibres to 20. The amount of memory used and the calculation time is determined largely by the number of nodes, the number of amplifiers and the lower signal limit.

CHAPTER 5

Results

In this chapter, some results of simulations are presented, and compared with figures derived from a similar study by PTT research.

5.1 Transmission without amplifiers

In this simulation, we consider a network, consisting of 2 nodes, each with a number of users, and connected by a monomode fibre. A transmitter is situated in node 0, and a receiver in node 1. For transmission without amplification, the received power can be calculated as

$$\begin{aligned} P_{\mathrm{RX}} &= P_{\mathrm{TX}} - L_{\mathrm{STAR1}} - L_{\mathrm{WDM1}} - L_{\mathrm{SW1}} - L_{\mathrm{WDM1}} - L_{\mathrm{FIBRE}} \\ &- L_{\mathrm{WDM2}} - L_{\mathrm{SW2}} - L_{\mathrm{WDM2}} - L_{\mathrm{STAR2}} \end{aligned} \tag{5.1}$$

with

P_{RX}	received power [dBm]
P_{TX}	transmitter power [dBm]
L_{STAR1}	loss of star of transmitting node 0 [dB]
L_{WDM1}	loss of one WDM coupler in transmitting node 0 [dB]
L_{SW1}	loss of series of switches in transmitting node 0 [dB]
L_{WDM1}	loss of one reversed WDM coupler in transmitting node 0 [dB]
L_{FIBRE}	attenuation of monomode fibre [dB]
L_{WDM2}	loss of one WDM coupler in receiving node 1 [dB]
L_{SW2}	loss of series of switches in receiving node 1 [dB]
L_{WDM2}	loss of one reversed WDM coupler in receiving node 1 [dB]
L_{STAR2}	loss of star of receiving node 1 [dB]

Some typical values are:

Transmitter output power	0dBm
Excess loss star coupler	0.2 dB
Receiver loss	-3.0 dB
Receiver sensitivity	-48 dBm
Optical switch loss	3.0 dB
WDM loss	1.0 dB
Fibre loss (10 km)	3.5 dB

Figure 5.1 displays the calculated received power vs. the number of stations at the transmitting node (N_{node1}) and receiving node (N_{node2}).

The other graphs are from a COBRA-BCPN demonstrator concept by PTT-Research. The difference between the results of PPT results and ours is caused by the use of an extra 3-dB coupler for bi-directional traffic and somewhat larger losses in connectors and components.

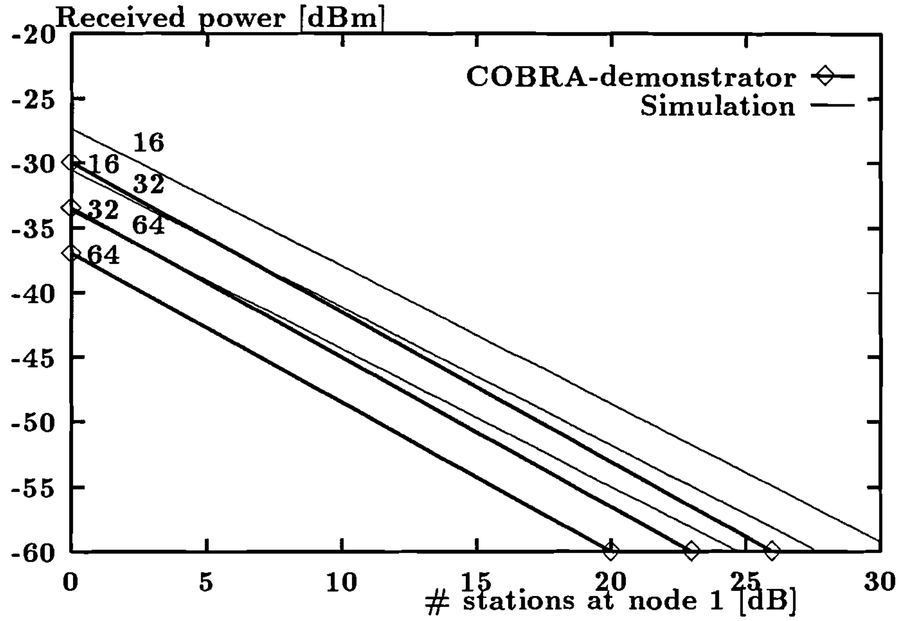


FIGURE 5.1: RX input power vs. N_{node1} for $N_{\text{node2}} = 16, 32, 64$

The graph shows that for the given distance and sensitivity, the number of users is limited to about 20.

5.2 TRANSMISSION WITH AMPLIFIERS

The number of users can be increased if an amplifier is used between node 0 and node 1. In figure 5.2 the received power is displayed when one amplifier is used at the output of node 0. This amplifier has a gain of about 10 dB, which brings the number of users per node to about 100.

Using two amplifiers, the distance between the nodes can be increased. In figure fig:2amp the received power is displayed vs. the distance between nodes, with a post- and pre-amplifier.

This shows that the distance is in the order of several tens of kilometers. The amplifier's pump power $p_p = 2$ mW, P_{TX}=0 dBm.

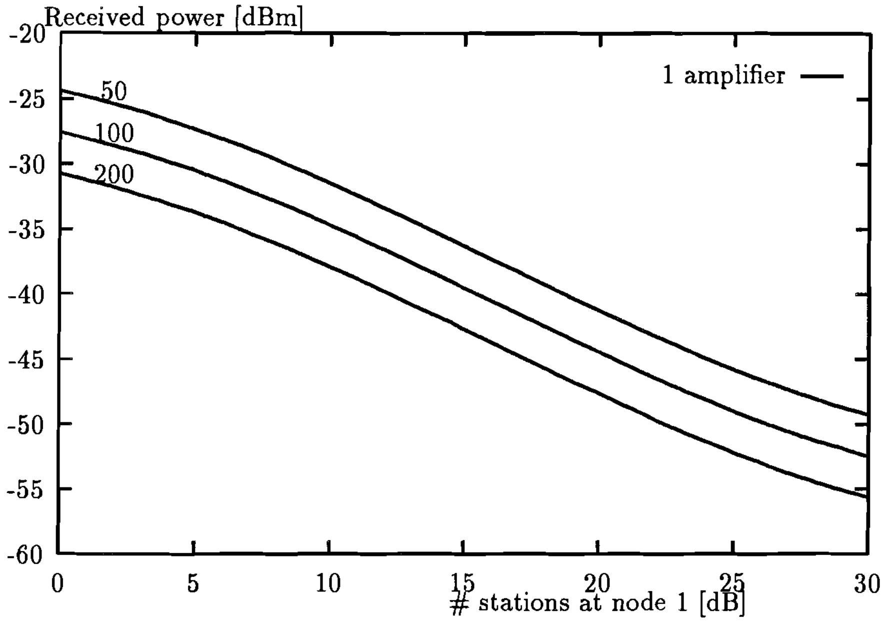


FIGURE 5.2: RX input power vs. N_{node0} for $N_{\mathrm{node1}} = 50, 100, 200$

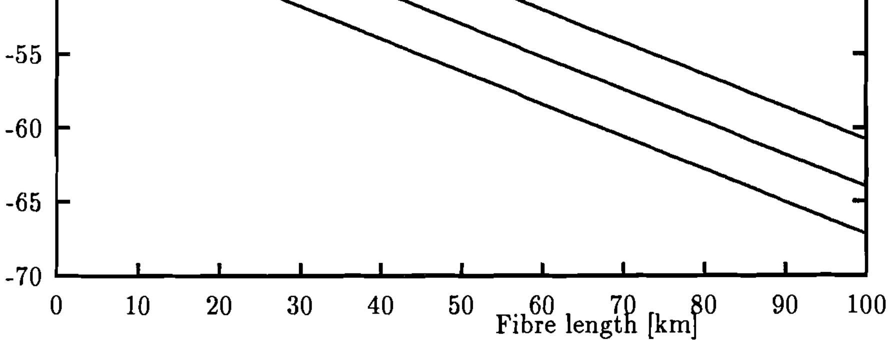


FIGURE 5.3: RX input power vs. fibre length for $N_{\mathrm{node1}} = 20, 40, 80$ and $N_{\mathrm{node2}} = 50$

CHAPTER 6

Signalling system

In the following part, several properties of signalling system are described, together with a possible implementation in an optical node in the program as well as in a demonstrator.

6.1 INTRODUCTION

In the recent past, high speed optical fibre bases systems made their way to point-to-point links, long distance transmission, as well as in computer networks. When local computer networks are considered, the bus and the star are the most popular physical topologies in optical networks. However, for a given transmitter power, the star can support more users than a bus topology. These networks come under the class of broadcast networks. That is, signals transmitted by each user are broadcast to all other users in the network and each receiver tunes in to the desired signal. Since fibre bandwidth is divided into a set of logical channels using WDM, each user in the network has access to a complete or partial set of these channels, depending on the capability of the user's transmitter and receiver.

Since users have access to many channels in these networks, co-ordination in the form of signalling and protocols become necessary, so that users can tune their receivers and transmitters to the approperiate channels to establish a connection, but also to prevent collisions between signals.

6.2 PROTOCOLS

When describing protocols, it is necessary to distinguish between two situations:

- When there are more channels than users, each pair of users can use a wavelength channel for a connection.

- If users must share a channel, the protocol will have to deal with timeslots, and the transmission of packets.

If the number of users and channels is small, a simple protocol to use is Time Division Multiplex (TDM). In this case, the time axis on each channel is divided into frames and each frame is subdivided into slots. These slots are pre-assigned to users in a non-conflicting way. This, and more protocols are described in [10].

It is assumed that, if packets are used, the communication between 2 stations through one channel is "regular traffc", i.e. packets are sent at regular intervals. This means that the controller does not have to deal with collisions between packets. In any case, whether packets are used or not, the connection between two users over a channel can be seen as a circuit. In both cases, controller and users will have to exchange signalling traffic.

6.3 Signalling traffic

Connections between customers are made on demand, and released when the communications is completed. The circuits within the transmission links can then be used by others.

Signalling information can be transferred in two ways. The first type is termed "Channel Associated". In such a system, a dedicated means of transferring signalling information is provided for each circuit. In channel-associated signalling (CAS) systems, information is either transferred over the circuit itself or dedicated signalling capacity is provided for each circuit within the transmission link.

The second type is termed "Common Channel Signalling" (CCS). Here, a common-signalling path is provided for a number of circuits. Hence, in CCS systems signalling capacity is provided in a common pool and allocated for use by each circuit as and when required, [11].

6.3.1 Channel-associated signalling

In CAS systems, the transfer of signalling information is conducted over signalling capacity that is specific to a particular circuit. fig 6.1 illustrates the conecpt of CAS systems. For each circuit between exchanges there is a directly-associated means of transferring signalling information. A physical tie exists between the signalling path and the traffic circuit. In the case of optical transmission through fibres there would be a signalling channel for each traffic channel.

In the figures, the broken lines represent the signalling channels.

6.3.2 Common-channel signalling

In Common-channel signalling systems, the physical tie between signalling path and the traffic circuit is removed. All signalling transfer relating to a transmission link takes place over a dedicated signalling channel (fig 6.2). Hence, a common signalling channel handles the transfer of signalling information for numerous traffic circuits. Signalling capacity is not reserved for each traffic circuit, but is allocated dynamically as and when required.

The transfer of signalling information is achieved by sending "messages" down the common signalling path.

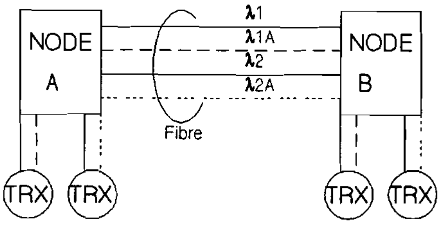


FIGURE 6.1: *Channel Associated Signalling.* λ_1 *and* λ_2 *are traffic wavelenghts,* λ_{1a} *and* λ_{2a} *are used for signalling*

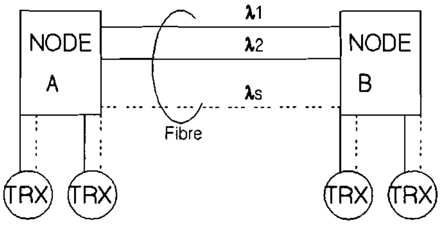


FIGURE 6.2: *Common Channel Signalling.* λ_1 *and* λ_2 *are traffic wavelenghts,* λ_s *is used for the common signalling channel*

The CCS system has a number of advantages:

1. The equipment for signalling is less complex. In the case of CAS, each wavelength channel would need its own transmitter/receiver pair in the signalling-wavelength band. This means that coherent receivers or direct detection receivers with optical filters have to be used. With CCS, each node needs only one receiver/transmitter for signalling for each connected fibre.

2. Signalling information is not restricted to the control of traffic circuits. The signalling channel can be used as a general data-transfer mechanism.

6.4 A POSSIBLE IMPLEMENTATION

In this part, an implementation for the Control and Management block is suggested. Next, the implementation of the C & M in the simulation program is described.

Normal traffic between users is conducted in the 1500nm band. The wavelength of signalling traffic must be chosen in such a way that:
there is no interference between user- and signalling traffic
- signalling traffic can be easily added and removed from the signals in the fibre
- signalling traffic inside and between nodes is completely separated.

These conditions can be satisfied by setting the signalling wavelength to about 1300nm. The wavelengths for traffic and signalling are so far apart that no interference exists. Crosstalk of the various components in the node is also non existent. Signalling traffic can be coupled into the inter-node fibre using directional couplers (of WDM couplers?). It can be recaptured by means of the same couplers and directed to the C & M. Finally, both amplifiers and WDMs will block all signalling traffic at the node's entrance.

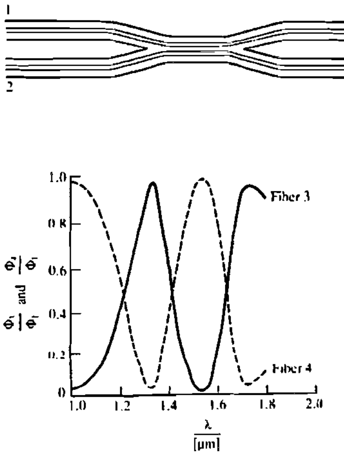


FIGURE 6.3: *A directional coupler can be used to separate signalling information*

Because of the difference in nature, C&M can be divided into two parts:
-inside the star (for local traffic)
-between nodes (trunk to trunk)
Figure 6.4 depicts an implementation of the C&M block.

6.4.1 Inside the star

The C & M is connected to the star by a direct detection transmitter/receiver. All stations are also equipped with such a device. Signalling traffic inside the node is used for making a local or distant connection. To distinguish between stations each station connected to the star has an unique address.

The signalling channel can only be used by one station at a time.

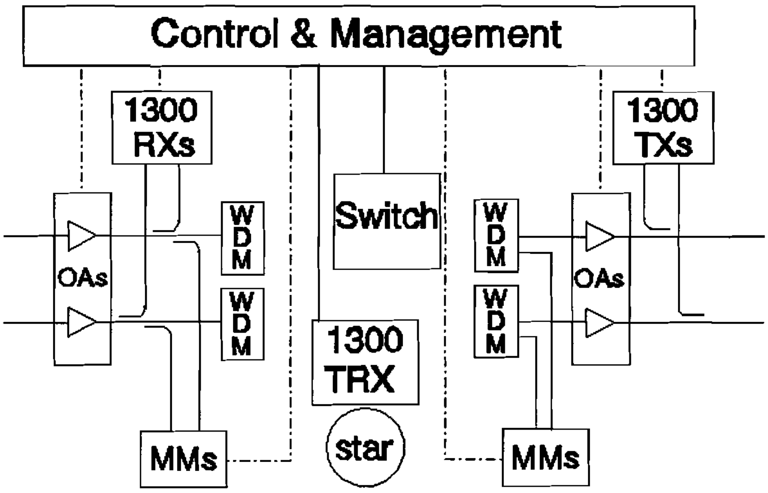


FIGURE 6.4: *An implementation of the Control and Management block*

Therefore C&M polls each station one by one. A station can then

- request a circuit to send to a another local station.,
- request a circuit and a fibre connection to a distant station,
- signal that it is online,
- signal that it will go offline,
- be called by another station,
- terminate a connection,
- report an error condition.

Because connections will exist for a longer time and the time required to establish a connection with a distant node is quite long, the time loss caused by the polling is not very important.

To make a connection, C & M needs information about all the wavelength channels and circuits. The use of circuits varies locally, so each C & M will store information about channels and circuits in the node:

- * the station that is receiving at a circuit,
- * the station that is transmitting at a circuit,
- * the address of the destination of the call: *local* or *distant*
- * the external calls from distant node to a distant node,
- * free circuits,
- * the circuits that cannot be used due to some error.

If non-transparant connections are possible, information about wavelength converters should be stored as well.

6.4.2 Signalling between nodes

Signalling between nodes is done in a simular way as inside the star. Instead of synchroneous polls of all the node's stations, signalling traffic is conducted in a asynchroneous method messages. These messages are exchanged between the two nodes connected to a fibre over one channel, so the traffic is half-duplex. Note that because signalling traffic signals are restricted to one fibre, the same wavelength can be used by all nodes in all fibres.

Nodes use messages to negotiate about circuits. They can

- request a connection to a station with address X at node Y at wavelength λ,
- terminate a connection to a station,
- acknowledge a connection,
- report a station busy,
- report a wavelength or a circuit being used,
- report a connection lost.

If a connection spans many nodes, messages could go back and forth for a long time before a circuit is found that is free everywhere. In the last case, (connection lost) the nodes connected to the broken fibre are responsible for conveying this information to all other nodes in the network.

6.4.3 Control system

In addition to the tasks of signalling, C & M deals with the control of optical switches and the monitoring of signal levels. Optical switches are used to route signals, as described elsewhere. Monitor modules connected to the output WDMs are used to register the levels of signals leaving the node. Either equalizers or amplifiers could be used to keep the level within certain limits.

6.4.4 Control and Management in the simulation program

Because its main purpose is the calculation of signal levels, the signalling system as described above is not needed in the program. It is possible to set the optical switches, and thus route the signals. For each node and each channel, an input fibre can be switched to an output fibre, through a *C & M button* in the node dialog window as described in chapter 4. To prevent having to set each switch in the network, a default situation exists: the upper fibres are connected (input 1 is switched to output 1), input fibre 2 is switched to the star, and signals from the star are routed to output 2. This is shown in figure 6.5.

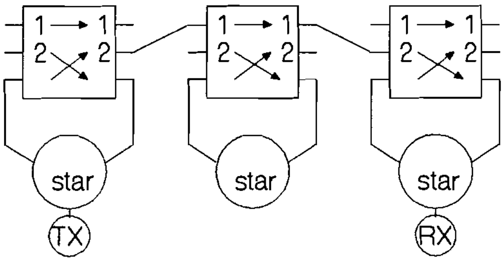


FIGURE 6.5: *Default settings of the optical switches, for all channels*

Chapter 7

Conclusions and recommendations

A study on the interconnections of optical networks has shown that their structure is very complicated, and to get an idea of their performance, simulations are necessary.

In this report a model of an optical node is presented. This model consists of a passive opticial transmittive star and an overlay with Wavelength Division Multiplexers, optical switches and optical amplifiers.

A computer program using this model enables the user to construct and simulate networks of optical nodes. The program is able to perform calculations on networks, with the emphasis on power budget calculation. The program was developed in such a way that other types of calculation and the simulation of more complex networks can be added in the future.

A comparison with other methods has shown that Common Channel Signalling is the best method for the Control and Management system to control and regulate traffic inside and between nodes. Signalling information using a different wavelength can be sent between nodes together with regular traffic and captured using WDMs.

Recommendations

- Several extentions can be added to the model: receivers, transmitters, optical filters to obtain a more complete optical node.
- Noise calculations could be included in the model to take into account the effects of crosstalk and Amplified Spontaneous Emission
- Other types of nodes with more connections can be defined to simulate more realistic networks.
- Other types of communications (non-transparent connection) and various modulation methods can be implemented in the model.
- Simulations of a Control and Management system could be performed, either by including such a system to the simulation program or in a separate program.

References

[1] Khoe, G. Coherent multicarrier technology for implementation in the customer access. *Journal of Lightwave Technology*, vol. 5/6 p.695–713, May/June 1993.

[2] Khoe, G., Boom v.d., H., and Put v.d., E. LAN interconnections based on a cross-connected multi-wavelength layer. In *LEOS 1994*, pages 40–41, 1994.

[3] Hill, G. et al. A transport layer based on optical network elements. *Journal of Lightwave Technology*, vol. 5/6 p.667–679, May/June 1993.

[4] Amersfoort, M. et al. High performance 4-channel PHASAR wavelength demultiplexer integrated with photodetectors. In *Proceedings 19th. ECOC*, volume 3, pages 49–52, September12–16 1993.

[5] Bachmann, M. et al. Polarization-insensitive low-voltage optical waveguide switch using InGasP/InP four-port Mach-Zehnder interferometer. In *Proceedings OFC/IOOC*, pages 32–33, February 1993.

[6] N. Wauters, P. D. Wavelength requirements and survivalibility in WDM cross-connected networks. In *Proceedings 20th. ECOC*, volume 2, pages 125–128, September 25–29, 1994.

[7] Giles, C. Modeling erbium doped fiber amplifiers. *Journal of Lightwave Technology*, vol. 9 no. 2 p.271–283, February 1992.

[8] Osch van, A. Measurements and models on an erbium-doped fibre amplifier. Graduation report, TU Eindhoven, Faculty of electrical engineering, Telecommunications division, 1994.

[9] Saleh, A., Jopson, R., et al. Modeling of gain in Erbium-Doped Fiber Amplifiers. *IEEE Photonics Technology Letters*, vol. 2 no. 10, October 1990.

[10] Sudhakar, G., Kavehrad, M., and Georganas, N. Access protocols for passive optical star networks. *Computer networks and ISDN systems*, vol. 26 p.913–930, 1994.

[11] Manterfield, R. *Common-channel signalling.* Peregrinus, London, 1991.

[12] Petzhold, C. *Programming Windows 3.1.* Microsoft Press, Redmond, Washington, 1992.

List of Abbreviations

ASE	Amplified Spontaneous Emission
BER	Bit-Error Rate
CAS	Channel Associated Signalling
CCS	Common-Channel Signalling
C & M	Control and Management (Block)
DDE	Dynamic Data Exchange
DPSK	Differential Phase Shift Keying
EDFA	Erbium-doped Fibre Amplifier
LAN	Local Area Network
OA	Optical Amplifier
OXN	Optical Cross-connect Node
TRX	transceiver
WDM	Wavelength Division Multiplexer

APPENDIX A

The software

A.1 INTRODUCTION

In this chapter, a bit will be explained about the way the program has been made and the choices that have been made.

Programming language
The calculating part of the program was written in C++, while the user interface using Windows functions uses more traditional C. It was written using the Borland C 3.1 compiler for windows. The program runs under Windows 3.1 and is a standard 'enhanced mode' Windows application. It requires a at least a 80386 microprocessor. A 80486 or mathematical coprocessor is recommended. Much of the code for the Windows user interface is based on the book "Programming Windows 3.1", by C. Petzhold [12]. This book gives many interesting examples on how to program with Windows. Parts on file handling and DDE were very useful. Object oriented class additions to Borland C were not used, because a lot of the examples were written in ordinary C, and the program's structure is a lot clearer this way.

Extending the program can be done by defining other nodes. To make another node than the 2-by-2 type, it is necessary to make a new object. This object must contain all the components (star, WDMs, switches, reversed WDMs, amplifiers). Only the switches object would have to be changed, to accommodate more inputs and outputs.

Speed
Most of the time is spent deleting and allocating signals. It takes time to request a piece of memory from the operating system, use it for a short time and then return it. Another part of the program that takes time is de *edfa* routines. The method of solving the differential equations (trapezium method) is simple, but not very efficient. A higher-order method using a variable stepsize could make a difference in speed.

Memory
Signals are stored in memory using the implied `new` and `delete` commands in the *constructor* and *destructor* functions. Note thet because some of the objects are references, the objects' desctructors are not executed automatically. The memory for storage is marked "Non-movable" and "Permanent". This means

that signals cannot be moved in memory while the program is running, but also that a lot of memory fragmentation takes place. The program will use a lot of memory because of this, and will be slower.

A.2 Program and header files

The program's source consists of the following files:

mwin.cpp	main program, message loop, windows-dependent object functions
dde.cpp	dde functions
popfile.cpp	file support functions
star.cpp	objects' functions for data management and calculations
dialogs.cpp	window's dialog screen functions
edfa.cpp	fibre amplifier functions
plot.cpp	print and plot functions

Header files:

entete.h	all objects
globals.h	global variables
functions.h	function headers
constant.h	constants and macros
node.rc	resource file
node.prj	Borland C project file

Additional files needed: `bwcc.dll`, for the dialog functions and `commdlg.dll`.

A.3 Data structures and objects

All objects have the same structure: they contain data that is specific to the object. If these variables are declared `public`, they can be accessed anywhere in the program, otherwise they are availible only in the object's functions.

Most objects have a *constructor* and a *destructor* function. The contructor function has the same name as the object. It is executed automatically immediately after the object's declaration and is used for creating an object and setting it in a default state. The destructor function has the object's name with a ~ before it and is called automatically when an object is no longer valid, to remove it from memory.

All larger objects have a `connect` function. It takes as argument one or more addresses of lists and is used to create a node or network. Objects have also a `calculate` function. This function first deletes old results and then calculates new signals.

A.4 Variables

Contrary to good programming practice, a lot of global variables were used. Most of those are used for storing and controlling the network. The nodes are indirectly stored in the array `nodes`, the fibres in array `fiber`. `nonodes` and `nofibres` are the amount of nodes and fibres in the network, as well as the index of the last node, or fibre, that was allocated. All relevant public variables in the network are of type `double`. During the series calculation, the addresses of these variables are stored with the variables' description in an array of structures `outputvars`. Parameters in the calculation are written to the addresses of those variables chosen during the clculation set-up. This way, every variable of type double in the network can be used as a parameter in a simulation.

A.5 SUGGESTIONS

A solution to the fragmentation problem would be to allocate memory for signals yourself. Use a large memory block. that contains thousands of signals, for the storage of all signals in the network. Requesting memory from this block and returning it after use will be much faster than bothering the operating system with this. Blocks can then even be dynamically allocated and moved in memory if Windows should require this.

APPENDIX B

Configuration files

In the network and information files, the properties and settings of the metwork are stored. These files contain one instruction on each line, with a comment. Because the program just reads the information without looking at the comments, the order and place of instructions cannot be changed. The network configuration file is not meant to be changed, except in special circumstances. For example, the only way to make a network with loops, is to add fibres to the configuration file. Below is a sample network description file for a network consisting of 2 nodes.

```
2  // #nodes
0  // node nr
1520                    // description of transmitted signals
1520
1520
10  1520  1.55e+09  3
~600  1547  1.55e+09  1
200  // # rec. @ star
4   // no channels
50 100  // position
-1 -1  // fibres connected left
-1 0  // fibres connected right
C:\qpw\qpw  // Quattro
c:\gnu.plt  // GnuPlot
-90.000000  // lower limit signals
CURVE1.i  // WDM informationfile
CURVE1.i  // WDM informationfile
CURVE1.i  // WDM informationfile
CURVE1.i  // WDM informationfile
CURVE1.i  // WDM informationfile
CURVE1.i  // WDM informationfile
edfa.i   // edfa infofile
10.000000  // edfa length
1.5e15 // edfa saturation parameter
0.000000  // edfa extra loss
0.010000  // edfa pump power
0   // edfa on/off switch
edfa.i   // edfa infofile
```

```
10.000000  // edfa length
1.5e15 // edfa saturation
0.000000  // edfa extra loss
0.010000  // edfa pump power
0   // edfa on/off switch
edfa.i   // edfa infofile
10.000000  // edfa length
1.5e15   // edfa saturation
0.000000  // edfa extra loss
0.010000  // edfa pump power
0   // edfa on/off switch
edfa.i   // edfa infofile
10.000000  // edfa length
1.5e15 // edfa saturation
0.000000  // edfa extra loss
0.010000  // edfa pump power
0   // edfa on/off switch
1  // node nr

 settings for the other node are identical

1  // #fibres
0  // fibre number
FIBERC.I 10.000000      // fiber attenuation curve and length
0 1  // connect nodenr, output
1    // other nodenr
140 50 -40 50   // fiber connections position
```

Objects' information files contain the more permanent settings, like loss- and gain curves. They cannot be changed inside the program, but can easily be edited by hand. Below is displayed the file `edfa.i`, describing the absorption and gain spectra of a erbium doped fibre.

```
1400   1.12    1.580  // gain and absorption spectrum [Np/m2]
1480   1.20    1.77   // of a erbium doped fibre amplifier
1490   1.32    2.23
1500   1.47    2.45
1510   1.58    2.81
1520   3.54    5.0
1530   7.9     7.8
1540   3.8     3.1
1550   3.2     3.1
1600   2.0     2.0
```

The last information file describes the losses of a WDM. The wavelengths are relative to the central wavelength λ_c for each channel.

```
-200 30                  // loss [dB] in a WDM
-100 30                  // relative to central wavelength
-10  30
-9   24
```