FAULT LOCATION ALGORITHMS FOR OPTICAL NETWORKS

THÈSE N^{O} 2164 (2000)

PRÉSENTÉE AU DÉPARTEMENT DE SYSTÈMES DE COMMUNICATION

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES TECHNIQUES

PAR

Carmen MAS MACHUCA

Ingénieur en télécommunication, Universitat Politècnica de Catalunya, Barcelona, Espagne
de nationalité espagnole

acceptée sur proposition du jury:

Prof. P. Thiran, directeur de thèse
Dr P. Bayvel, rapporteur
Dr W. Denzel, rapporteur
Prof. R. Guerraoui, rapporteur
Prof. J.-Y. Le Boudec, rapporteur

Lausanne, EPFL
2000

Fault Location Algorithms for Optical Networks

To Dimitris and all my family.

Acknowledgements

First of all, I would like to express my sincere thanks to Prof. Patrick Thiran, who supervised the last crucial years of my thesis. Although it was a new domain for him, his interest and motivation helped me to conclude successfully this work. I greatly value his friendship and all the things he taught me.

I would also like to thank Prof. Jean-Yves Le Boudec who offered me the possibility to come to EPFL and guided me through the first determining years of the thesis. He provided me with the opportunity to participate in the European project COBNET, which gave me a lot of experience and a chance to meet interesting people. Among these people I would particularly like to mention Dr. Wolfgang Denzel, Jonathan Armitage, Andrew McCabe, Jacques Saint-Blancat and Bart Meekers. The development of the management platform by our Institute would not have been possible without the help of Olivier Crochat, Heiko Boch, Bipul Parua and Simon Znaty.

I am very grateful to all my colleagues at the Institute for computer Communications and their Applications (ICA) who played an important role in creating a friendly atmosphere. I owe special thanks to my officemates: Catherine Boutremans, Paul Hurley and Sam Manthorpe; to the system managers: Bruno Dufresne, Jean-Pierre Dupertuis and Hans Einsiedler; as well as to the secretaries: Danielle Alvarez, Holly Cogliati, Angela Devenoge, Yvette Dubuis and Jocelyne Plantefol for their efficiency, continuous availability and English corrections.

I am very grateful to the Office Fédéral de l'Éducation et de la Science (OFES) and Swiss National Research Fund (FNRS) who financed my work for my Ph.D.

I would also like to thank Artis-Software for allowing me test their simulation software and Dr. Daniel Rodellar for the stimulating discussions on the physical layer, his help performing simulations and his friendship.

Since life is not only work, I would like to thank all my friends in Lausanne, far too many to list here, who gave me support and courage when needed. I will treasure these years for the rest of my life!

Last but not least, I would like to thank Dimitris and all my family for believing in me, which gave me enough confidence to keep going. Despite the distance, they always felt close to me to encourage and make me laugh. A vosotros Dimitris, padres y hermanos, os queria agradecer el haber estado a mi lado en los buenos y malos momentos de la tesis, haberme mostrado vuestra confianza y haberme dado continuos ánimos.

Abstract

Today, there is no doubt that optical networks are the solution to the explosion of Internet traffic that two decades ago we only dreamed about. They offer high capacity with the use of Wavelength Division Multiplexing (WDM) techniques among others. However, this increase of available capacity can be betrayed by the high quantity of information that can be lost when a failure occurs because not only one, but several channels will then be interrupted. Efficient fault detection and location mechanisms are therefore needed.

Our challenge is to identify and locate failures (single or multiple) at the physical layer of an optical network in the presence of some lost and/or false alarms.

After briefly introducing optical networks and the multiplexing techniques that can be used, we study the most common components and their most usual failures. We propose a classification of all the network components based on their behaviour when failures occur. This classification gives an abstract model of the optical network, which is appropriate for developing algorithms to locate faulty elements.

Two algorithms that solve the fault location problem are proposed. Both algorithms cope with existence of false and missing alarms when locating single and multiple failures. The problem of locating multiple failures already in the absence of false or missing alarms, has been shown to be NP-complete.

The first algorithm, which is called Alarm Filtering Algorithm (AFA) is based on the combination of two approaches: *forward* and *backward*. The *forward* approach returns for each network element, their domain, which is the set of network elements that will send an alarm when the considered element fails. The *backward* approach returns the set of elements that are directly related to the received alarms. In this approach, the alarms that are considered to provide redundant information, are discarded. The combination of the results given by both approaches allows the location of multiple failures, given an allowed number of false and missing alarms.

However, this algorithm does not minimize the complexity when new alarms are received. Hence, a second algorithm, which is called Fault Location Algorithm (FLA), is proposed. The FLA concentrates the complexity in a pre-computation phase, so that when new alarms are received, the result of the algorithm is rapidly displayed. The FLA algorithm is based on the construction of a binary tree that realizes a non linear error correcting code. The FLA has also been extended to locate *soft* failures in addition to hard failures. Hard failures are unexpected failures, whereas soft failures are progressive failures due to equipment aging, misalignments or external factors such as temperature or pressure.

Both algorithms are compared on some simulated networks using different network topologies and failure cases. The comparison has also be done on the basis of their worst case complexity. Some conclusions indication with which settings each algorithm perform the best, were obtained.

Keywords: optical networks, fault location, wavelength division multiplexing, multiple failures, lost and false alarms.

Version Abrégée

Aujourd'hui, il n'y a pas de doute que les réseaux optiques sont la solution à l'explosion du trafic internet encore inimaginable il y a deux décennies. Les réseaux optiques offrent une grande capacité grâce, entre autres, à l'utilisation des techniques de multiplexage en longueur d'onde (Wavelength Division Multiplexing, WDM). Néanmoins, l'avantage procuré par cette augmentation de la capacité disponible doit être contre-balancé par la grande quantité d'information qui peut être perdue quand il y a une panne, car non seulement un, mais plusieurs canaux sont alors interrompus. Des mécanismes de détection et de localisation des pannes sont par conséquent nécessaires.

Notre objectif est l'identification et la localisation de pannes (simples et multiples) d'éléments de la couche physique d'un réseau optique en pésence de fausses alarmes et d'alarmes perdues.

Après une brève description des réseaux optiques et des techniques de multiplexage, nous commençons par étudier les composants optiques et leurs pannes les plus courantes. Nous proposons une classification de tous ces éléments basée sur leur comportement en cas de panne. Cette classification procure un modèle du réseau optique à un niveau d'abstraction suffisant pour développer deux algorithmes de localisation des éléments défectueux.

Les deux algorithmes tolèrent l'existence de fausses alarmes et d'alarmes perdues, et localisent des pannes simples, ou multiples. Notons que la localisation de pannes multiples, même sans fausses alarmes ou alarmes perdues, est un problème NP-complet.

Le premier algorithme, que nous nommons Alarm Filtering Algorithm (AFA), est basé sur la combinaison de deux approches: directe (*forward*) et inverse (*backward*). L'approche directe donne pour chaque élément du réseau, son domaine, c'est-à-dire l'ensemble des éléments qui produiront des alarmes en cas de panne de cet élément. La seconde approche fournit l'ensemble des éléments qui sont directement liés aux alarmes reçues. Les alarmes qui sont considérées donner une information redondante sont éliminées. La combinaison des résultats donnés par les deux approches permet de localiser des pannes multiples, malgré l'existence d'un nombre fixé a priori de fausses alarmes et d'alarmes perdues.

Cependant, cet algorithme ne minimise pas la complexité quand de nouvelles alarmes sont reçues. Par conséquent, un second algorithme, que l'on nomme Fault Location Algorithm (FLA), a été proposé. L'objectif du FLA est de concentrer la complexité dans une phase de pré-calcul, de telle sorte que les éléments défectueux soient localisés très rapidement dès que de nouvelles alarmes sont émises. L'algorithme FLA est basé sur la construction d'un arbre binaire ainsi que sur des propriétés de codes correcteurs non linéaires. Outre les pannes *dures* (ou soudaines), l'algorithme FLA a été étendu pour pouvoir également localiser pannes *douces* (ou progressives) qui sont causées par le vieillessement des équipements ou d'autres causes externes comme température et pression.

Les deux algorithmes sont comparés sur des réseaux simulés, en prenant des topologies différentes et divers types de pannes. La comparison a été aussi faite sur base de leur complexité dans le cas le plus défavorable.

Contents

Chapter 1

Introduction

There is no doubt today that optical networks are the solution to the explosion of Internet traffic that two decades ago we only dreamed about.
This breakthrough is due to two main factors:

- First of all, the possibility to increase the link *capacity* by multiplexing several channels with different wavelengths into one optical signal allowed higher bit rates. The number of multiplexed channels increased considerably with the appearance of Erbium Doped Fiber Amplifiers .
- Secondly, the *connectivity* of optical networks, i.e. the ability to interconnect pairs of network nodes simultaneously, was made possible by the appearance of static Add-Drop Filters and Add-Drop Multiplexers.

Optical networks will continue to evolve until they become completely re-configurable and all-optical networks with wavelength conversions and routing.
However, this increase of available capacity can be betrayed by the high quantity of information that can be lost when a failure occurs because not only one, but several channels will then be interrupted. Rapid restoration and fault identification are therefore vital to ensure performance and a safe operation of optical networks.

1.1 Objectives

Our challenge is to identify and locate failures (single or multiple) at the physical layer in the presence of some lost and/or false alarms.
Multiple failures and false and lost alarms are rarely taken into account in current fault management systems, yet they may quite easily occur. Take the example of alarms issued when some threshold is crossed by some variable. Thresholds set too high may conceal a failure by not sending the expected alarms; and conversely, thresholds set too low will prompt many false alarms.
Before outlining the contents in the next section, let us briefly summarize the evolution of this doctoral dissertation.

- This work started with the design and the implementation of the network management platform of the optical network developed within the ACTS European project COBNET [1]. A study of the optical components enabled us to classify them according to their behavior when *hard* failures occur. These *hard* failures are unexpected

failures, such as a fiber cut that interrupts suddenly the transmission of the optical signal.

- This classification enabled us to work with an abstract model of the network to be managed, and to design a first algorithm called Alarm Filtering Algorithm (AFA), which is able to locate multiple failures at the physical layer with the information delivered by the hardware components while tolerating a given number of false and lost alarms. The AFA algorithm combines two approaches: the *backward* phase, which filters as many redundant alarms as possible, and the *forward* phase, which locates the fault from the non redundant alarms. This algorithm is efficient in filtering the alarms, but does not minimize the time to locate failures upon reception of new alarms.

- A second algorithm was therefore developed, which minimizes this time by concentrating most of the complexity in a pre-computational phase that can be carried on off-line, before reception of new alarms. This algorithm, called Fault Location Algorithm (FLA), can be viewed, to some extent, as a non-linear error correcting code. The diagnosis phase (i.e., the failure location upon reception of a new alarm) is rapid, at the cost however of an increase of the over-all complexity due to the absence of alarm pre-filtering.

- In addition to hard failures, the FLA has been extended to locate *soft* failures, which are the result of equipment aging, misalignments or external factors such as temperature or pressure. To achieve this goal, the algorithms take information not only from the optical components, but also from other equipment, such as WDM monitoring equipment, SDH layer elements or IP routers.

- A complexity study of both algorithms was performed to assess which algorithm must be used for different parameters of the networks (number of active/passive elements, etc.)

1.2 Dissertation Outline

This thesis is organized as follows:
Chapter 2 gives a general introduction on optical networks and their management. It begins with a broad overview of optical networks, their multiplexing techniques and their classification. The management of networks is then introduced with a particular emphasis on fault management. The chapter ends with a state-of-the-art of the techniques used for fault diagnosis.
Chapter 3 describes the main components of an optical network and the components present in other layers such as WDM or SDH layers, which are components that may provide more accurate information about *soft* failures.
The next chapters form the core of this dissertation.
Chapter 4 introduces the first algorithm to solve the fault location problem at the physical layer. Because this algorithm is based on the alarms issued at the physical layer, a classification of the hardware components is given on the basis of their alarming properties when a *hard* failure occurs, which enables us to abstract the fault location problem and develop an algorithm to solve it, which is called Alarm Filtering Algorithm (AFA). Given

the alarms received by the hardware components, the AFA returns a list of fault candidates. This algorithm performs first some discarding of the alarms that can be considered redundant.
In Chapter 5, we present the second algorithm where not only *hard*, but also *soft* failures have to be located, given the information from physical and some other layers. The classification of the components of Chapter 4 is updated to include these new elements. This second algorithm is called Fault Location Algorithm (FLA) and concentrates most of the complexity in a Pre-Computing Phase (PCP) so that the computational complexity when alarms are received by the manager is minimized.
Chapter 6 illustrates mainly some simulation results of both algorithms. The chapter starts by briefly describing how both algorithms were implemented in Java. The algorithms are applied to different failure scenarios onto different network topologies. An optical system simulator `OptSim` © [2] was used to emulate the failures at the physical layer.
Chapter 7 studies the complexity aspects of both algorithms, and compares the settings in which each algorithm performs the best.
This dissertation concludes with perspectives on future work.

Chapter 2

Introduction to Optical Networks

2.1 Introduction

This chapter is a rather general introduction to the management of optical networks with a particular emphasis on fault management and diagnosis. We begin by reviewing the multiplexing techniques (Space Division Multiplexing (SDM), Time Division Multiplexing (TDM) and mainly Wavelength Division Multiplexing (WDM)) in Section 2.2, and we describe the WDM network developed in the framework of the European COBNET project. We then move to Section 2.3 that presents the control and management of optical networks with a description of the platform developed for the COBNET network as an example. We emphasize on the functions of fault management and we describe different techniques in fault diagnosis in Section 2.4.

2.2 Optical Telecommunication Networks

Telecommunication networks have grown tremendously during these last two decades. The major causes of this growth are the explosion of Internet and the World Wide Web. Future applications such as the simultaneous video and sound transfer in real time, or the everywhere access of information will increase even more the traffic in the Internet and thus the need for more bandwidth. To fulfill these high demands of bandwidth, communication networks have moved from electrical to optical, and the optical ones have also evolved using different multiplexing techniques to increase their capacity.
Optical fiber is an ideal transmission medium because it offers higher bandwidth and is less sensitive to interferences. The optical fiber began replacing copper when its capacity was exhausted. The fiber was used as a transmission medium and the switching and routing was performed electronically. These networks are called *First Generation Optical Networks*. Today, they are used everywhere, except in the local access networks where the high installation costs are usually not justified by the reduced capacity needed. Examples of *First Generation Optical Networks* are SONET and SDH networks that form the core of the communication networks in USA and Europe, and the COBNET network, which will be presented in Section 2.2.3. This Ph.D. work was realized in the framework of the COBNET project. So the COBNET network is used as an example of a WDM network and as a model to test the proposed fault location algorithms.
The continuous increase of traffic and the expensive installation cost of new fibers led engineers to search for techniques to increase the capacity of already installed fibers, by

multiplexing channels within the same fiber. Two solutions were developed: Time Division Multiplexing (TDM) and Wavelength Division Multiplexing (WDM). Both of them will be described in Section 2.2.1.
In recent years, optics have been used not only as a transmission medium but also to perform more complex functions, such as switching and routing, because electronics are becoming too slow to process the high rates achievable in optics. The networks that perform transmission, switching and routing in the optical domain are called *Second Generation Optical Networks*. One of the main features of these networks is their complete transparency to the data sent over the light path. One example is the telephone network: whenever a call is established, the user can send not only voice but also other kinds of traffic such as video, fax or data. However, in some cases the transparency may not be complete if the optical signal is converted into an electric signal in order to be regenerated or if the Bit Error Rate (BER) has to be computed for some performance measurement because in these cases, the type of signal has to be known in advance.

2.2.1 Multiplexing Techniques

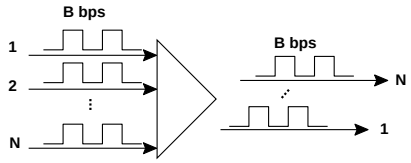
To increase the capacity of the optical fiber and to concurrently transmit different channels, several multiplexing techniques are used in optical networks. Interested readers may find more information in [3]. We describe here the three most popular ones:

- *Space Division Multiplexing (SDM):* This multiplexing technique does not exploit the capacity of the single fiber because it associates each channel to a different single-mode fiber, as shown in Figure 2.1 (a). The term *multiplexing* can therefore be questioned. However, this technique is deployed because of the non-critical synchronization, the low cross-talk, the possible use of laser arrays at transmitter, as well as receiver arrays, and their easy management.

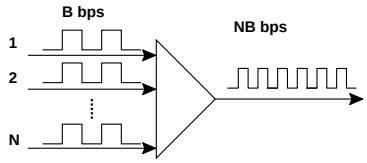
- *Time Division Multiplexing (TDM):* This multiplexing technique divides the time variable into time slots, and periodically allocates one time slot to one particular channel (see Figure 2.1 (b)). The optical signals at the output of the multiplexer must therefore be the sum of the incoming bit rates. This technique requires very good synchronization among the incoming signals in order to send the channel information in the correct time slot and to avoid overlaps between channels while making use of the full time slot.
  This technique is used today in commercial networks at bit rates up to 10 Gbps. Beyond this rate, the transmission through the fiber cannot be achieved at significant distances due to dispersion and other impairments that we will discuss later in Section 3.3.1.

- *Wavelength Division Multiplexing (WDM):* WDM is fundamentally identical to the Frequency Division Multiplexing used in radio systems. WDM sends different data simultaneously at different wavelengths over a single fiber (see Figure 2.1 (c)).
  At first, the wavelengths do not interfere with each other. Due to imperfections of the transmission systems, some effects however cause interference between channels. WDM commercial systems offer up to 32 wavelengths at 2.5 Gbps each or 16 wavelengths at 10 Gbps over a single fiber. The next section presents in detail optical communication networks based on this multiplexing technique.

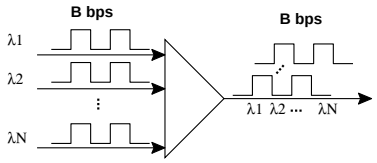
The SDM technique is used in every optical ribbon when several optical fibers are put together in one physical ribbon as an array. The TDM technique is used to increase the



(a) Space Division Multiplexing (SDM)



(b) Time Division Multiplexing (TDM)



(c) Wavelength Division Multiplexing (WDM)

Figure 2.1: Different Multiplexing Techniques in optical networks

capacity of one link but has a limitation of 10 Gbps for the synchronization [4]. The WDM technique is used to increase even more the capacity of one link and combined with TDM is able to achieve terabits per second [5].

Fiber failure effects for each multiplexing technique

The effect of a fiber failure changes with the multiplexing technique used:

- SDM: In this case, several optical fibers are grouped forming an optical ribbon. If only one fiber breaks, only one channel will be interrupted. If the failure affects more fibers or even the whole ribbon, more channels will be interrupted.
- TDM: When TDM is used, all the channels go through the same fiber. When the fiber is cut, all the channels are interrupted and the synchronization is lost.
- WDM: When WDM is used, as in the TDM case, all the channels that are traveling though the same fiber are interrupted when the fiber fails.

2.2.2 WDM Networks

Today, WDM networks are deployed in long-haul networks but in a few years they are expected to also be deployed in local exchange and access networks. Most local and metropolitan area networks use simple broadcast topologies: ring, star or buses. Let us show in Figure 2.2 the case of a star topology with 4 nodes. In this example, the network is all-optical and the nodes have fixed transmitters and tunable receivers. The connection is done via a passive star coupler [6]. Each node transmits on a given wavelength and receives all the wavelengths, so that each node can choose which wavelength to retrieve by tuning the receiver. These local topologies have a limited maximum number of nodes because the wavelengths can not be reused and the emitted power is split among all the receivers in the network by the star coupler. In contrast, wide area networks use mesh topology with nodes having switches to forward data to other nodes. These nodes can forward different wavelengths from the input signal to different output ports. These networks are able to reuse wavelengths in separated lightpaths as shown on the example in Figure 2.3. In this example, the path between A and F is established using λ_1. A path between A and D is established using λ_2 because λ_1 is already used between A and E. When a path between C and D has to be established, λ_1 can be used because it does not

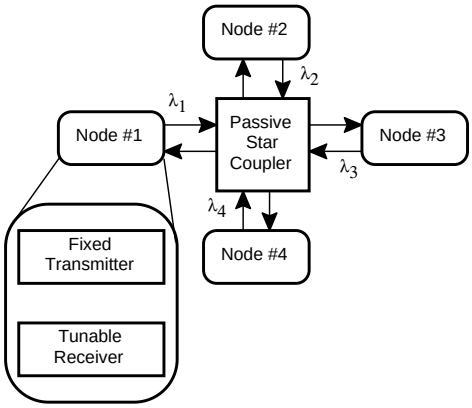


Figure 2.2: Example of a single-hop network with a Passive Star Coupler

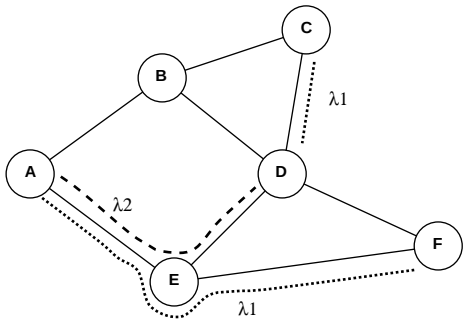


Figure 2.3: Meshed network able to establish lightpaths that reuse wavelengths

share any link with the lightpath between A and F.
The network becomes more dynamically configurable when the nodes can perform wavelength conversion so that at every link any wavelength can be chosen. Using the same example of Figure 2.3 and assuming that only two wavelengths are available in the network, if a path between E and C, through D, has to be established, it can not be done because both wavelengths are already used. If the network can perform wavelength conversion, the path can be established by using λ_1 between E and D and λ_2 between C and D (as shown in Figure 2.4). In general, the number of channels that can be established in

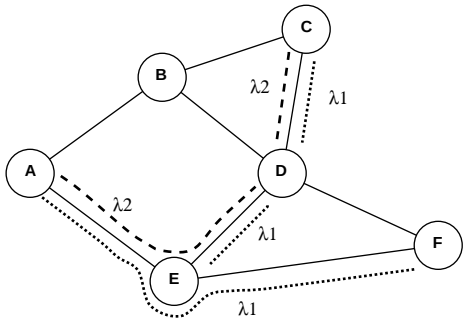


Figure 2.4: Meshed network able to establish lightpaths that convert wavelengths

a network using wavelength conversion is larger than those that do not, having the same number of nodes, links and available wavelengths.
Up to this point, we have given a possible classification of optical networks based on

whether they use wavelength conversion or not. A second possible classification is done based on whether they perform electrical conversion of the signal during data transmission or not. In this case, optical networks can be classified in two different groups: single-hop or multi-hop networks [7]. Single-hop networks are the networks that do not convert optical signals to electrical signals during data transmission, whereas multi-hop networks perform electrical conversion to be able to route or regenerate the signal during data transmission. Because the hardware technology has still to evolve and improve, multi-hop networks are more used.

2.2.3 Example: COBNET Network

The COBNET project [1] is a project member of the ACTS programme, which was established under the Fourth Framework Programme of European activities in the field of research and technological development and demonstration.
The overall goal of COBNET is to establish the architectural and technological concepts for the next generation of high-performance Corporate Networks taking advantage of new and evolving photonic technologies and existing switching/multiplexing/ routing techniques.
Corporate Networks are networks formed by the inter-connection of Customer Premises Networks (CPNs) via Wide Area Network (WAN) or Metropolitan Area Network (MAN) links from the Public Network Operator (PNO), or via leased lines from a private network provider. In this project each CPN consisted of two protected optical rings interconnected through a central switch. One protected ring used SDM multiplexing technique (for nodes distancing less than 2 km) and the other one used WDM multiplexing technique (for nodes distancing more than 2 km). The central switch also allowed the connection towards other CPNs through the Public Network, using a WDM point-to-point link (see Figure 2.5). One of the important characteristics of these CPNs is their low cost, which was achieved by avoiding the use of amplifiers. Another advantage is the allowance of a 'Clear Channel' connectivity, which is essentially a bit rate, protocol independent, and largely distance independent transmission technique.
Concerning the fault restoration and location, both rings of each CPN are protected, that is, they consist of two rings: one clockwise and another counter-clockwise. The former is the working ring and the latter is the protection ring. The information of each channel is sent through both rings so that when a fault occurs in the working ring, the information can still be retrieved from the protection one. The change from the working ring to the protection ring is done via a protection switch that reacts to the absence of incoming optical power. The problem is that when a failure occurs in the protection ring, it is not possible detect it until this ring has been used due to a second failure in the working ring.
A difference between the failures at SDM and WDM rings is that, as it was mentioned earlier, a failure at the WDM ring affects all the established channels whereas a failure at the SDM ring can affect a different number of channels depending on the number of optical fibers of the ribbon that were cut.
Because all switching and routing is done electrically, the COBNET network can be classified as a multi-hop meshed network.
Within the project, these concepts were developed and verified in a demonstrator, which was presented in November 1998 at British Telecom (BT) Labs, London. This demonstrator consisted of two CPNs, located in Ipswich (CPN1 on Figure 2.5) and in Borehamwood, London (CPN2 on Figure 2.5) interconnected via a public switched network.

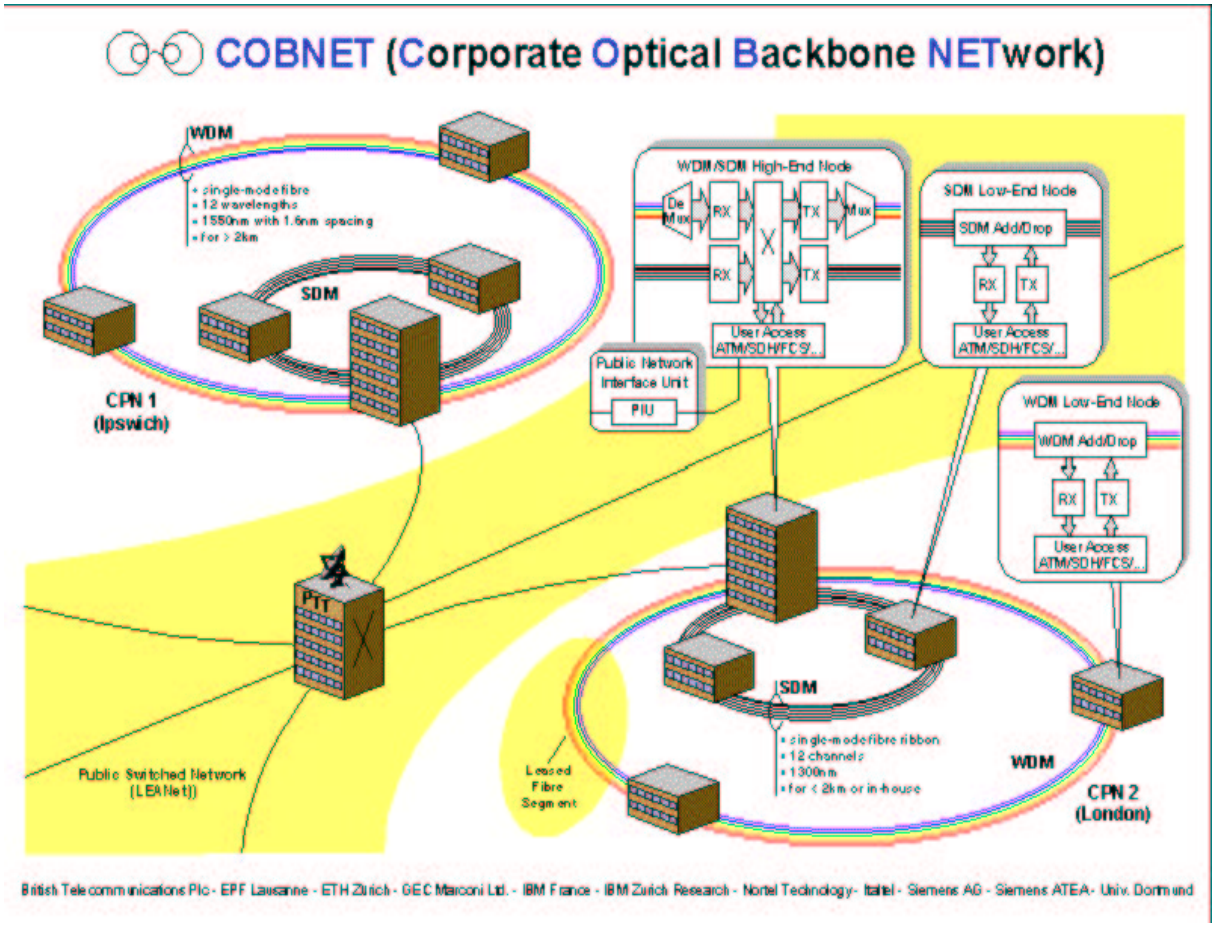


Figure 2.5: COBNET field trial

2.3 Control and Management

All communication networks need control and management functions that will allow them to be used more efficiently. Management functions, presented in Section 2.3.2, have been classified into different functional areas defined by the International Organization for Standardization (ISO).

2.3.1 Management platform structure

Most networks have a centralized management center that performs all the management functions in a centralized manner. This management platform implies that all the managed elements have to communicate with the same manager, which for large networks implies big delays. This management platform also implies that the manager has to perform all the management functions, which can cause large processing times. Some of the processes, such as the set up of new connections or the restoration of failed connections, need however fast response times that cannot be achieved by a centralized management. The delays of these management tasks can be reduced by distributing them. This is, for example, the case of an SDH/SONET network that can achieve restoration times of 60 ms due to management distribution. The increasing network size, making the quantity of information to managed considerably larger, calls also for distributing the management tasks.

The management structure of a network is composed of the *three* following components: the so-called *network elements* are the elements of the network that can be managed, the *agent* is the software that controls a set of network elements and communicates with the *manager* that performs the management functions for the network it is responsible for (see Figure 2.6). The management information is stored in the Management Information

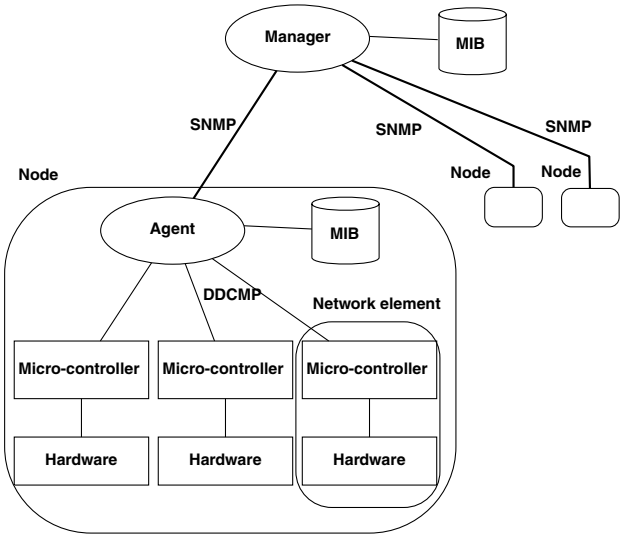


Figure 2.6: Elements and Protocols of a Management platform

Bases (MIBs) that are attached to each agent and to the manager. Section 2.3.3 will describe the MIB of a node used in the COBNET network. The term *node* encompasses the corresponding agent, the network elements it controls and its MIB.
There are two kinds of protocols used in the management structure. One protocol is used between the manager and its agents and the other is used between the agent and the network element.

- The protocols used in the communication between managers and agents are master-slave based, that is, there is a master that governs one or several slaves and the slave that responds to the master's commands. The manager has the master's role and the agent has the slave's role. This is the case, for example, with the Simple Network Management Protocol (SNMP) used to manage TCP/IP networks and the Common Management Information Protocol (CMIP) used to manage OSI based systems. The three basic operations common to both management protocols are: *set*, which allows the manager to give a certain value to some given variable of the MIB and make the hardware execute it; *get*, which allows the manager to retrieve the value of a certain parameter and *trap*, which allows the agent to send an information that was not explicitly asked by the manager. For example, when the manager wants to establish a connection, it will *set* parameters of the involved network elements such as lasers and switches; when the manager wants to check the position of a switch, it will *get* the associated variable; when a failure occurs, some elements will inform to the manager about the abnormal situation by sending *traps* (e.g. *NoInputPower* if the element stops receiving power).

- The protocol used between the agent and the network element is more hardware oriented. This is, for example, the case of Digital Data Communications Message

Protocol (DDCMP) [8], which was used in COBNET. Therefore, another role of the agent is to translate the management commands from one protocol to the other.

2.3.2 Management Functional Areas

To facilitate the description of the management requirements for the network design, the OSI has broken down the management functions into five different functional areas:

- Fault Management
- Configuration Management
- Performance Management
- Security Management
- Accounting Management

We give a brief description of each functional area.

Fault Management

Fault management deals with the detection, isolation and elimination of abnormal system behaviour [9]. Indeed, the performance deviation in the behaviour can be the definition of a fault and its manifestation as a failure. When a fault occurs in the network, several tasks have to be performed as rapidly as possible:

- locate the fault,
- isolate the fault so that the network can continue to operate,
- reconfigure the network to minimize the impact of the fault,
- and replace the failed components.

The detection of faults is based on monitoring the state of the network components. Simple fault detection mechanisms are often based on locally monitored variables. The faulty values reached by these variables are logged as errors. Critical errors are sent to the network manager as alarms. However, it is not always possible to detect complex faults only on the basis of locally monitored variables: it is then necessary to have a global knowledge of the network and to do some processing to diagnose the presence/absence, the nature and the location of the fault. Also, and because the fault can be propagated to components that depend on the failed component, the influences of faulty components on other components have to be taken in account to perform an efficient fault management.

Configuration Management

Configuration management deals with initializing the network components, establishing relationships among them, maintaining, adding and updating these relationships, and keeping the manager informed about the status of the components. These relationships between components are based on the connections established and cleared down in the network. Configuration management also has to be able to reconfigure the network when the user imposes changes. It also should include routines able to inform about any change at the configuration (for example, when a protection switch changes its position, the manager should be informed about the new path of the established channels).

Performance Management

Data communication networks are composed of a huge variety of components. Performance management has to monitor and control them. *Monitoring* is the function that tracks activities in the network, whereas the *control* function enables adjustments in the components to improve network performance. The main performance issues are: network capacity utilization, existence of excessive traffic and of bottlenecks, and increase of response time. Performance management collects information from the network and analyzes it so that the system manager can recognize situations of performance degradation.

Security Management

Security management deals with generating, distributing and storing encryption keys (like passwords or other authorization information). Also, security management monitors and controls access to computer networks and to management information.

Account Management

The services provided by many networks are charged to the users. The network management has some procedures that perform not only the internal accounting but also other tasks, such as checking the right use of the access privileges and avoiding the abuse of the network by some users at expenses of others.

2.3.3 Example: COBNET Network Management

The role of EPFL within the COBNET project presented in Section 2.2.3 was to design and implement a management platform able to control and manage the COBNET network. The management functionalities implemented were configuration and fault management. The management platform was realized in two levels:

- The *Management Level* where the fault and configuration management are implemented using software tools and protocols. It works on an abstract representation of the entire network, which includes all its network elements and their present status. It views the whole network as a unified structure, with addresses and labels assigned to each network element and its specific information. The Management level is implemented using existing SNMP standards.
- The *Control Level* is the interface between the Management level and the hardware. Its task is to offer to the Management level an up-to-date status of the hardware and to translate the configuration commands from the management into signals that modify the status of the hardware components accordingly. Active elements of the network provide regular feedback on their status. This level is a proprietary one, no architectural standard exists, and each manufacturer defines solutions for his/her own system, depending on the hardware used.

The overall COBNET management platform has a hierarchical structure consisting of three levels: a global manager level, a CPN level and a node level (see Figure 2.7). The global management level has a graphical user interface (GUI) (as shown in Figure 2.8) to the human manager, which enables him/her to act upon the network by performing actions such as establish and clear down channels, or act upon hardware equipment. The CPN management level behaves as an agent towards the global manager, and as a manager

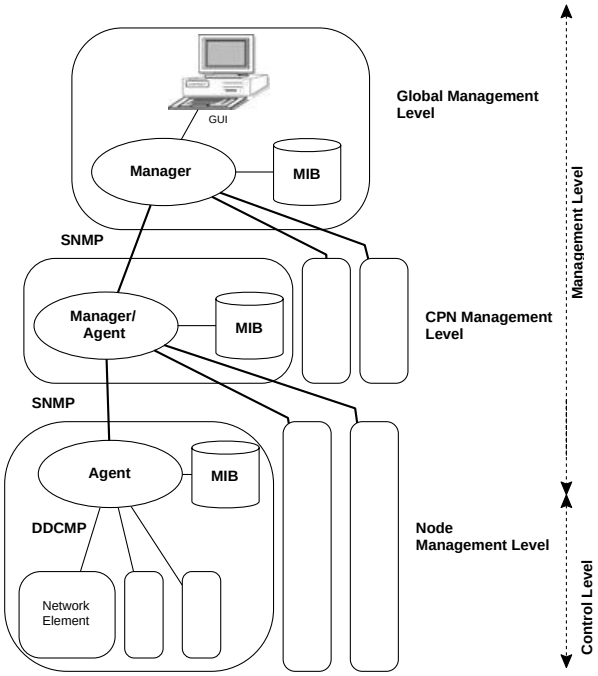


Figure 2.7: COBNET Management Platform in three management levels

towards the managed nodes. The lower level is the node management level that performs the communication with the hardware through the agent. Communications between manager and agents are SNMP-based, whereas communications between the node agents and the hardware elements use DDCMP [8], which is a freeware data link control procedure. The MIBs at each management level were defined and implemented in Abstract Syntax Notation.1 (ASN.1) using the framework given in the SNMP standard RFC.1155 [10] and the extensions defined in additional SNMP documents RFC.1213 [11] and RFC.1215 [12]. An example of the MIB storing the information of the node containing the central switch is shown in Figure 2.9. The MIB stores its data based on a tree distribution where the *Service* branch corresponds to the fixed data and the *Architecture* branch corresponds to the dynamic data, such as the position of the switch, and which input port is connected to which output port.

2.4 Fault Management Systems

Optical communication networks, and networks in general, need a fault management system able to perform fault diagnosis, that is to say, able to identify the faults that occur based on the information given by the network components. A *fault* can be defined as an unpermitted deviation of at least one characteristic property or variable of a network element from acceptable/usual/standard behaviour, whereas a *failure* can be defined as the manifestation of the fault. For example, if the ventilator in a laser stops and if the temperature increases and overpasses an accepted temperature limit, the fault is the stopped ventilator and the failure is the temperature of the laser, which is too high. Because both terms are closely related, we may use them indistincly. Indeed, considering the particular case of communication networks, when a failure occurs, several symptoms or events are issued to the network manager. As mentioned in Section 2.3.2, the network manager per-

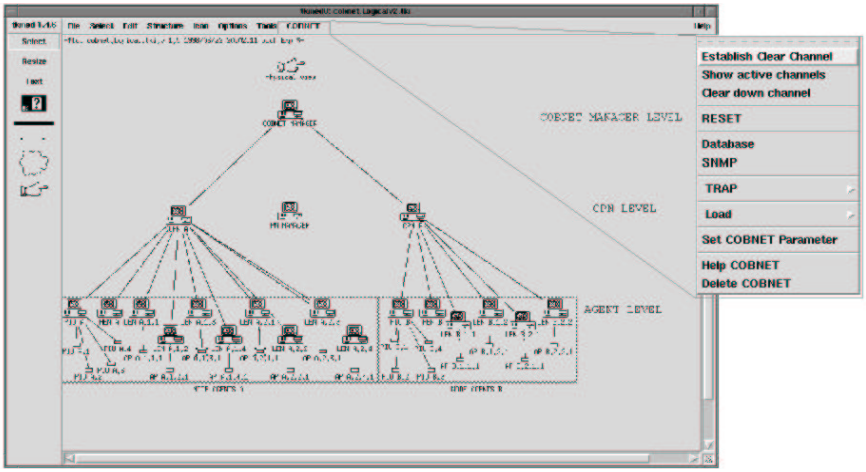


Figure 2.8: Graphical User Interface (GUI) used in COBNET

forms, among other management tasks, fault management that in principle should identify the network elements having failed.
This problem is solved by Fault Management Systems that take as inputs the events generated by the network elements (these events can be alarms, warnings or parameters of the network element itself), and produce an output, which is the set of network elements whose failure would explain the input events (see Figure 2.10).
These fault management systems differ in:

1. the way they solve the problem: using neural networks, storing previous cases, etc.
2. the information they need: failure propagation probabilities, timestamps, set of established channels, etc.
3. the assumptions on which they rely: existence of only single failures, existence of multiple failures but assuming than x failures are more likely than $x+1$ failures, etc.
4. the memory they use: memoryless, memory needs, etc.
5. the nature of their output: nature of the failure, network elements that have failed, absence of solution, etc.

2.4.1 State-of-the-Art

All techniques performing fault diagnosis rely on the analysis of symptoms and events that are generated during the occurrence of the fault. Amongst these techniques, we find expert systems technology, model based systems, case based systems and neural networks. They are briefly described in this section.

Model based systems

Model based systems are systems that construct an abstract model of the network that has to be diagnosed. The model can be of any kind, from logic to differential equations, and represents the failure dependence of the network elements. Depending of the kind of model, different approaches to the model can be used, for example, statistical approaches, artificial intelligence based approaches, or approaches using control theory [13]. Based on

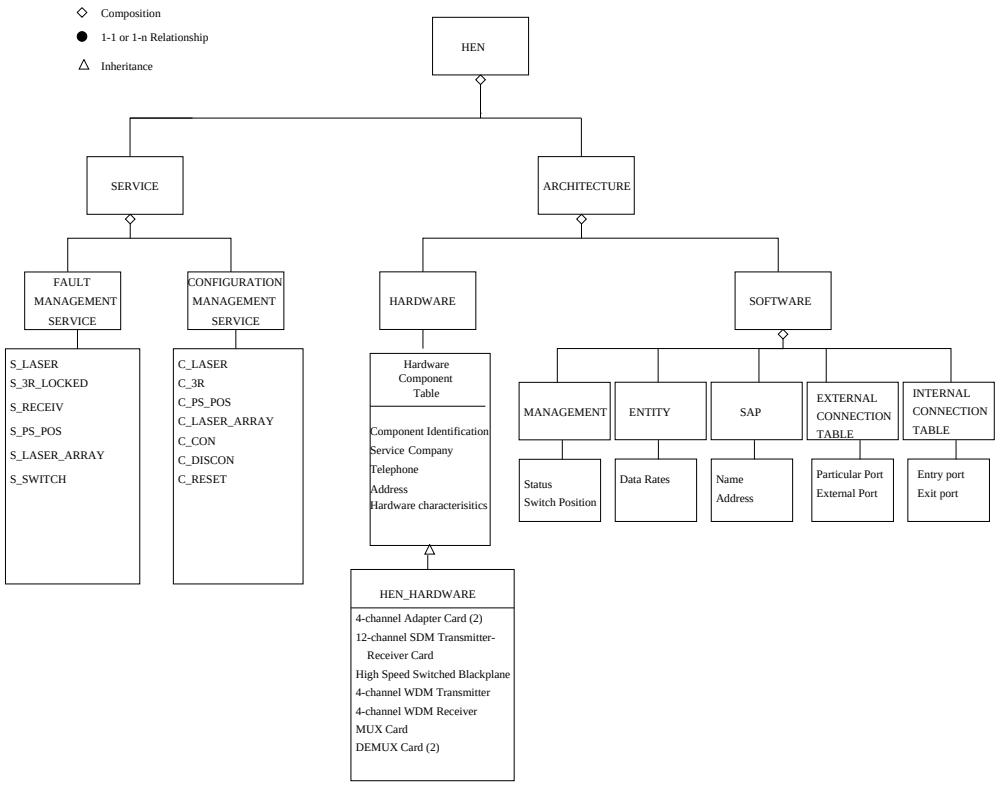


Figure 2.9: Management Information Base (MIB) of one of the nodes at the COBNET project.

the model, it is then possible to deliver the elements that best explain the discrepancies between the observations and the predictions of the model (see Figure 2.11).
The advantages of these systems are that:

- they are able to cope with incomplete information,
- they do not require learning,
- they can cope with unforeseen failures

The drawback is the difficulty of developing good model for complex networks.
The algorithms presented in Chapters 4 and 5 of this thesis belong to this category. Other examples include the Finite State Machine (FSM) model presented by Wang [14] and by Bouloutas [15], the probabilistic reasoning systems presented by Katzela [16] and by Wang [17], and the proprietary system `ECS` of Hewlett-Packard [18]. Let us briefly describe some of them:

- Wang [14] treats the system as a discrete event system whose behaviour can be described by concatenation of events. Some of the concatenations correspond to correct network behaviour, whereas others correspond to faulty behaviour. Wang considers the system as a FSM that is a 3-uple of elements issued from respectively the set of possible events, the set of states and the state transition function. Any fault violating the FSM specification will be detected.
- Katzela [16] models the network as a graph, which takes into account the dependencies among the different network components. Every link between two network

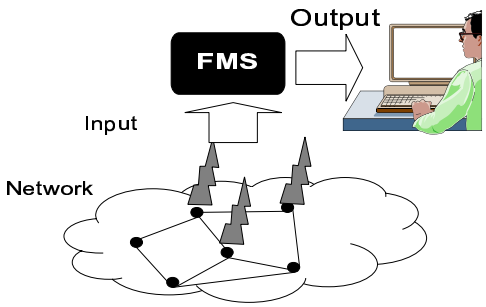


Figure 2.10: Fault Management System (FMS)

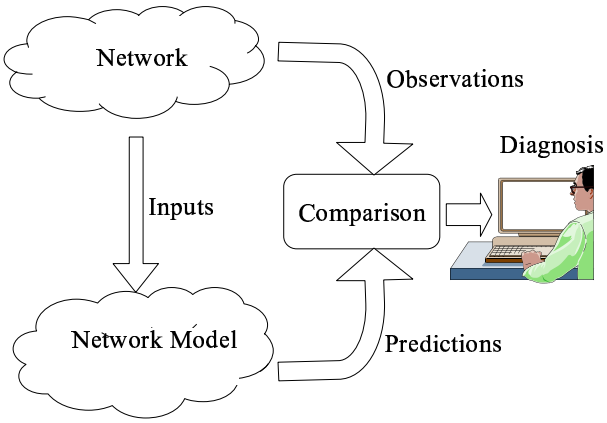


Figure 2.11: Model Based System

components has a weight, which is the probability that the failure of one particular network component propagates to the neighboring components. Once the graph modeling is done, a fault diagnosis algorithm based on the failure probability propagation is proposed for systems, where failures are assumed to be independent events.

- Wang [17] models the network as a set of nodes interconnected with links that can fail with a given probability. Depending on the channel connections that can and cannot be established between each pair of nodes, he is able to give the more likely failed link.

- `ECS` [18] stands for Event Correlation Services and has been developed by Hewlett-Packard as a part of the `OpenView` network management framework. This software allows to the system designer to model the system behaviour. The model uses a graph that interconnects *nodes* and where *events* circulate through. The *events* are the ones generated by the system and the *nodes* are modules that process the events by, for example, filtering, generating or combining them. This software can cope with events out of order and with alarms at a high rate. The drawback however is that the rules have to be defined by the user, which can become a difficult task for big networks.

Expert Systems

Expert systems use the knowledge of human experts acquired during their past experience with the network. They work in two phases (see Figure 2.12):

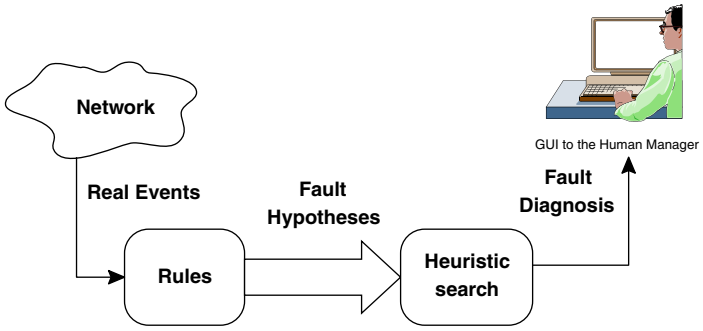


Figure 2.12: Expert System Approach

- The first phase is the *generation of rules*, based on the expert's knowledge. These rules associate fault hypothesis with (network/alarm) events, and have the form

  if `event` x **then** `fault hypothesis` y

- The second phase is the *operational or diagnosis phase*: whenever an event is prompted, the expert system will extract from the rules a fault hypothesis. When a large number of events is prompted, they will often result in different fault hypothesis. In this case, an heuristic search is performed to recover the most likely one.

Expert systems have the advantage of effectively taking human expertise into account, and coping with large scale systems. The operational phase is also rapid.
The drawback of expert systems is the required translation of the human expertise into a set of rules, which cover all cases in an exhaustive manner. This is a very difficult task, which may result in a large number of rules and therefore require a huge database.
Applications of expert systems to fault diagnosis have been proposed by Brugnoni [19] and Jakobson [20].

- Brugnoni [19] proposes `Sinergia`, which is an expert system for the isolation and diagnosis of faults in the Italian Telecommunications Network. The final fault diagnosis is carried out by maintenance experts. `Sinergia` reduces the amount of data to which the human expert is faced. This software defines, for every possible failure, a set of network elements affected by that particular fault. Brugnoni calls this set *Fault Influence Area*, which corresponds to our *Domain* definition of Chapter 4.

- In Jakobson's alarm correlation expert system [20], not all the faults are associated with alarm signals, but they can be recognized by correlating alarms, that is, by finding the reciprocal information that the alarms contain. The alarm correlation involves knowledge about the network elements and relations between events happening within these network elements when failures occur. The expert system is used in a commercial shell called Intelligent Management Platform for Alarm Correlation Tasks (`IMPACT`), which is a GTE proprietary system. `IMPACT` has been used

for both land-based and cellular telecommunications networks and can correlate up to 15 alarms per second [21]. Because it is a proprietary system, its implementation is kept confidential.

Case Based Systems

Case Based Systems are on-line learning systems that use earlier experiences as the basis for making decisions. To solve a particular problem, a case based system remembers the occurrence of a similar problem that was solved in the past, and adopts the old solution to solve the newly submitted problem. The system can be divided into *four* steps: *retrieve*, which finds the best match of a previous case, *reuse*, which finds what can be reused from old cases, *revise*, which checks if the proposed solution is correct and *retain*, which learns from the problem solving experience (see Figure 2.13).

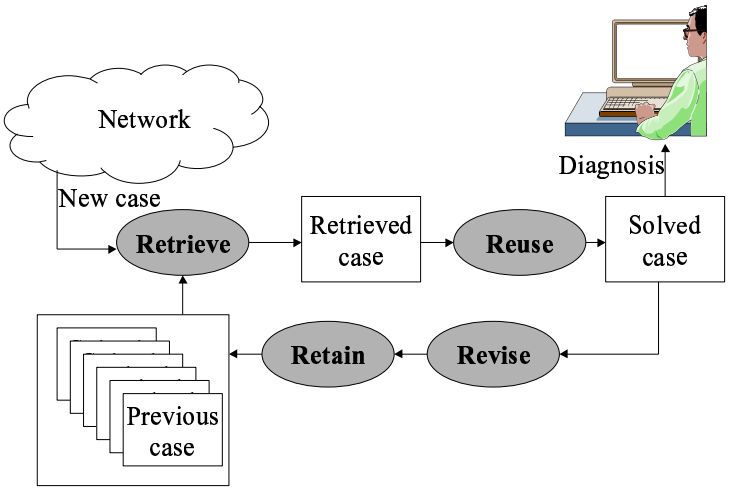


Figure 2.13: Case Based Systems Approach

The advantages of these systems are their efficiency and speed when the submitted problem is a previously submitted and solved case, the on-line learning that allows storing newly solved cases and their implementation simplicity.
The main disadvantages are that they need a huge memory to store all the past cases and that they cannot cope with new problems that were not previously encountered.
Lewis [22] proposes a case base system to solve fault diagnosis with its techniques to retrieve, adapt and embed knowledge in the case library. This system is called `CRITTER` and it applies case base reasoning to enhance a standard trouble ticketing system. A solved trouble ticket is a case that records a former problem and a trouble ticket database is the case library. The enhancement is done in three features:(i) retrieve the useful ticket when there is a new problem submitted to the system, (ii) if necessary adapt the ticket to recommend a solution and (iii) embed the experience in the trouble ticket database.

Artificial Neural Networks

An Artificial Neural Network (ANN) is a set of elements, the *neurons*, interconnected by parameters, and the *weights*. Systems using ANNs realize a black box modeling of the network using statistical learning. The mapping between the input and the output of the neural network is realized in two phases:

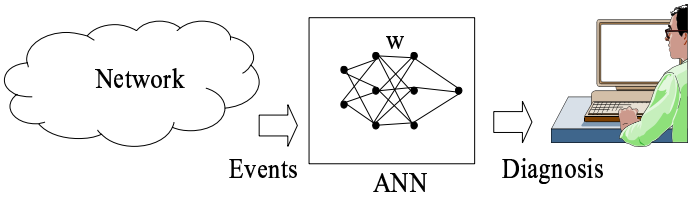


Figure 2.14: Artificial Neural Network Approach

- *Learning phase* The initial values of the weights are randomly chosen. This first phase updates the weights by learning from a database containing samples of pairs `(input events, desired output)`. In the case of surprised learning algorithm, the weights are updated by some form of a gradient descent of an error functional between the desired output and the actual output, as sketched in Figure 2.15. A unsurprised learning algorithm makes use only of the input network events. It is only at the end of the process, once the weights have reached their final values, that they are "labeled" and associated to a particular diagnosis output.

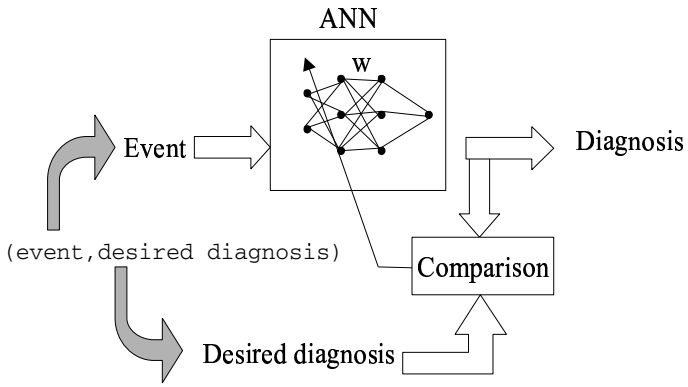


Figure 2.15: Learning phase of ANNs where w stands for *weight*

- *Operational phase* When the training is completed, the ANN should issue the correct diagnosis whenever an event is presented, provided it generalizes well from the samples in the training database to new samples.

The advantages of using ANNs are:

- the avoidance of explicit modeling,
- learning is not only achieved from past experiences but from any set of measurements,
- if the system is properly trained, it can cope with new cases,
- once the system is in the operational phase, the processing time for a new diagnosis is very low.

The drawbacks of these systems are mainly twofold: the training can be slow and the generalization can be poor. Moreover, neural networks are only useful as a tool for statistical classification from analog and noisy time series. In this thesis, the input events are mainly binary information, and the diagnosis is obtained by a deterministic mapping from

this input information. If noise corrupts the input information, some form of error-control coding theory will be more relevant for handling errors, as we will see in Chapter 7.
An example of neural networks applied to the fault diagnosis problem is proposed by Gardner [23]. In his work, Gardner uses Kohonen Self-Organizing Maps [24] to identify fault scenarios arising in an SDH-based network. He considered an SDH network with 11 interconnected Add-Drop Multiplexers nodes. Four different types of failures were simulated: line break, transmitter failure, framing error and pointer error.

2.5 Our contribution

Our objective is to develop a fault diagnosis algorithm avoiding the use of failure propagation probabilities and timestamps. The drawback of the use of these probabilities is that they change with time, that is, the failure propagation probability of a new network link is different from the failure probability after functioning for several years. The timestamp parameter can be useful to distinguish possible fault candidates but not as a main parameter because of the possibility of the existence of delays and of the absence of synchronization between all network components.
Within the framework of the COBNET [1] project, we studied which are the equipment commonly found in an optical network, their alarms when faults occurs and how they get propagated through the network: the results of this study will be given in the next chapter. A classification of these optical network components was then possible and allowed us to work on a problem abstraction. This abstraction, which can be considered as a modeling of the network, consideres these components as blocks with different alarming properties, and channels as ordered lists of blocks.
We first aimed to identify sudden failures at the physical level based on the information received from the network components and the defined propagation rules. This algorithm, called Alarm Filtering Algorithm (AFA), combines two different approaches (backward and forward approach) to give the fault candidates that best explained the received alarms. One of the main features was the discarding of redundant alarms.
Then, the need of an alarm discarding phase was questioned. Indeed, a second algorithm called FLA, which stands for Fault Localization Algorithm, was developed avoiding the backward approach but concentrating all the complexity to a pre-computing phase. The core of the FLA consists on building a binary tree whose leaves point to possible fault candidates. FLA was extended to also identify progressive failures on the basis of other information more subtle to these failures such as Bit Error Ratio (BER) or wavelength stability.
Any of the two algorithms needs training and has memory to remember old solved cases. Indeed, they are based on a common system modeling. This modeling is explained in detail in next chapter.

2.6 Conclusion

This chapter gave an introduction to optical communication networks. We started by shortly describing the evolution of communication networks towards optical networks, and by focusing on Wavelength Division Multiplexing (WDM) networks. We then gave an overview of the management and control of communication networks, with the particular example of the COBNET management platform. Special attention was given to

fault management, and different methods to perform fault diagnosis were described. The chapter concluded with an overview of the thesis work.

Chapter 3

Components of Optical Networks

3.1 Introduction

This chapter gives an overview of the components most frequently used in optical networks. Two different categories of components are distinguished: *hardware components* and *monitoring components*. The former ones described in Section 3.2, are the components at the optical layer and their failure must be identified by the fault management (some typical failures are described in Section 3.3). The latter ones, described in Section 3.4, are the components whose failures either will not interrupt the channels (as the WDM monitoring equipment) or are located by its own management platform (as IP routers), but which provide information about the failure of hardware component and so it will help to a better location of faults.

3.2 Hardware Components of Optical Networks

Let us describe some of the most common hardware components of an optical network.

3.2.1 Optical Fibers

Optical fibers are the medium for transmitting optical signals between two points. They offer low attenuation, low cost and the capability of transmitting simultaneously several information channels at different wavelengths.
An optical fiber consists of a cylindrical core surrounded by a cladding which is itself surrounded by a jacket, as shown by the cross section in Figure 3.1. Both the core and the cladding are made primarily of SiO_2, which has a refractive index of approximately 1.45. This index is the ratio between the speed of light in a vacuum and in that material. The index of the core n_1 is slightly higher than the index of the cladding n_2. Single-mode fibers (in which only one single path that a guided ray can take is possible) have cores with diameters of 8 to 12 μm and a cladding diameter of 125 μm, whereas multimode fibers have cores of approximately 50 μm in diameter.
Light is propagated optimally through the fiber when *total internal reflection* occurs, that is, when all the energy is reflected inside the cladding [4, 25]. Transmission losses in fibers are due to their attenuation, dispersion and nonlinear impairments and to the splice and bending effects as presented in Section 3.3.

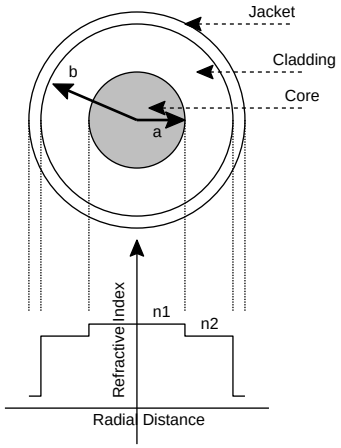


Figure 3.1: Cross section of an optical fiber with step refractive index

3.2.2 Transmitters

Light sources have evolved from Light-Emitting Diodes (LEDs) operating in the 850 nm range to semiconductor lasers, which are still evolving today. The choice of a light source is determined by the requirements that it should meet:

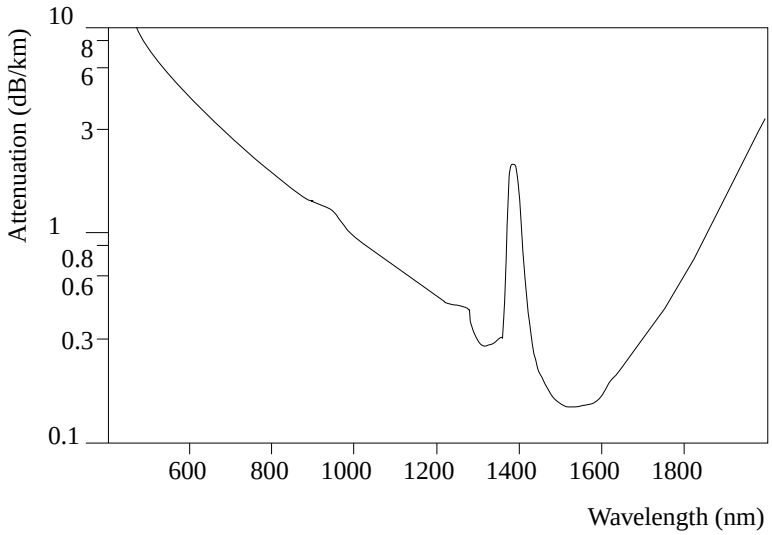
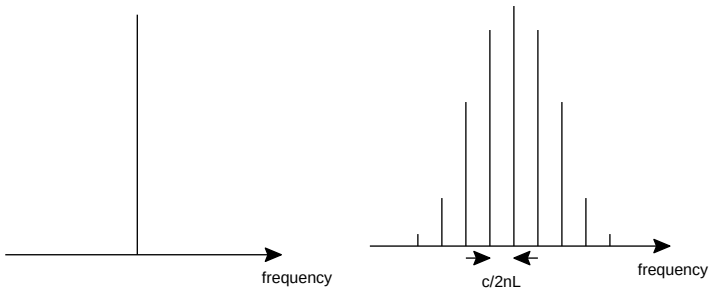


Figure 3.2: Attenuation Loss in silicon as a function of wavelength in μm

- The source wavelength should belong to one low-loss window of the two windows shown in Figure 3.2, which are centered around the 1.3 μm and 1.55 μm wavelengths. Wavelength stability is an important performance parameter that may limit the spacing between the different wavelengths of the different channels, and hence the number of channels in WDM networks (see Section 2.2.2).
- The spectral line-width of the source is the spectral width of the light generated by the source and should be as small as possible (around 1 nm). The line-width affects the minimum spacing between channels and the amount of dispersion when the light propagates along the fiber.

- Other desirable requirements are efficiency, reliability, price and compatibility.

It is possible to modulate the source at a given rate. The source is either directly modulated at the desired rate or, by an external modulator in tandem with a source that gives steady optical power. With some exceptions for short-distance and low bit-rate applications, all optical transmission systems use semiconductor lasers.
One parameter of the transmitter is the number of longitudinal modes, which is the number of wavelengths that the laser can amplify (see Figure 3.3). Single mode lasers amplify only one longitudinal mode, whereas multi-mode fiber amplify several modes.



(a) Single-mode Laser (b) Multi-mode Laser

Figure 3.3: Spectrum of two different types of laser

The word laser is an acronym of Light Amplification by Stimulated Emission of Radiation. Figure 3.4 shows a general representation of the structure of a laser. It consists of two

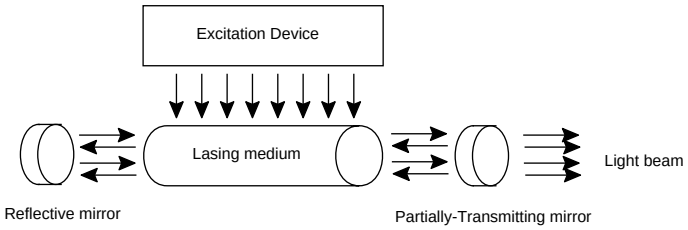


Figure 3.4: General representation of the laser structure

mirrors (one reflective and the other one partially-transmitting), which form a cavity. The cavity is actually filled by a lasing medium, which is made of a quasi-stable substance, in other words, a substance likely to stay in the excited state (the energy level reached by atoms after energy absorption) for long periods of time without steady excitation. The excitation device excites the electrons of the lasing medium, which causes the emission of light photons. The photons will reflect off the mirrors at each end of the capacity and will pass through the medium again. Stimulated emission occurs when a photon passes close to an excited electron and makes it release its energy onto a photon with the same direction and coherence as the stimulating photon. The partially transmitting mirror will allow some photons to escape in the form of a narrow-focused beam of light. In lasers with one cavity of length L, the wavelengths that are amplified verify that $n\lambda = 2L$ for some $n \in \mathbb{N}$.
So far, lasers operating on a fixed wavelength have been introduced. In some networks, there is a need for tuning the wavelength of the transmitter. This leads to the existence of tunable lasers which are able to emit at one wavelength within a given range. Some characteristics of tunable lasers are: tuning range, tuning time and whether the laser is

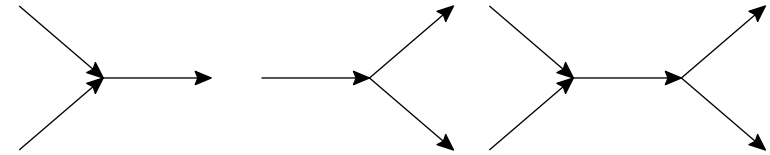
continuously or discretely tunable (over a range or to a set of wavelengths) [26]. Tunable lasers are mostly used as transmitter of the protection link in 1:N protection schemes.

3.2.3 Optical Detectors

An optical detector is a device that converts an optical signal into an electrical signal that can then be amplified and processed. There are different types of photo-detectors such as vacuum photo-diodes, semiconductor photo-diodes and photo-multiplier tubes. Semiconductor photo-diodes are the most popular, as they are small, low cost and yet provide good performance.
The basic principle of these detectors is optical absorption: when light incises on a semiconductor, it may be absorbed depending on its wavelength. The two commonly used photo-diodes are PIN (p-doped, intrinsic, and n-doped layers) and APD (avalanche photo-diodes).

3.2.4 Couplers, Combiners and Splitters

Coupler is a general term used for devices that combine optical signals into a fiber or split them out of a fiber. Combiners (see Figure 3.5(a)) are couplers that merge the optical signals coming from two different fibers into one signal through one optical fiber. Splitters



(a) 2x1 Combiner (b)1x2 Splitter (c)2x2 Coupler

Figure 3.5: Couplers, Combiners and Splitters

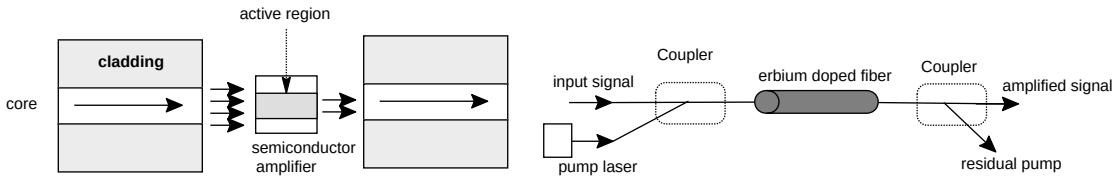
(see Figure 3.5(b)) are couplers that send the optical signal on one fiber in two or more fibers. A 2x2 coupler is, in general, a 2x1 combiner followed by a 1x2 splitter as shown in Figure 3.5(c). Due to careful design, couplers can be wavelength independent over a wide range, although for some applications they are designed to be wavelength selective so that the coupling coefficient depends on the wavelength. Such couplers are used to combine signals at 1310 nm and 1550 nm into a single fiber without losses. The same coupler can be used to separate signals at 1310 nm and 1550 nm coming from one fiber into separate fibers. Couplers can also be used in Erbium-Doped Fiber Amplifiers, described in Section 3.2.5.

3.2.5 Optical Amplifiers and Repeaters

Because fiber attenuation limits the reach of a non-amplified fiber span to 200 km for gigabit networks, wide area optical networks cannot work without line optical amplifiers. The amplification can be opto-electrical or all-optical.

- The opto-electrical amplifiers are able to re-shape, re-time and re-generate the power of the signal at the electrical domain. The device that performs these last three functions is called 3-R repeater. The drawbacks of repeaters are -because the optical signal is converted to an electrical data signal- the mandatory use of multiplexers and demultiplexers (Section 3.2.6), the need of one repeater for each channel and the delays added by the signal conversion.

- The all-optical amplifiers can only increase the optical power of the overall spectrum. They use the mechanism of stimulated emission, similar to those of a laser or a photodiode. The drawback is that optical shot noise is then also amplified with the signal; plus, optical amplifiers introduce spontaneous emission noise. The amplifiers are put every 160 km for ultra long hauls, 120km for very long hauls and 80 km for long hauls [27].
  The most characteristic parameters of an amplifier are its gain, gain bandwidth, gain saturation, polarization sensitivity and amplifier noise. *Gain* gives the ratio of the output optical power to its input power. *Gain bandwidth* gives the range of wavelengths over which the amplifier can effectively operate, which limits the number of wavelengths that can be used. *Gain saturation* is the amplification at which the output power is no longer sensitive to an increase of the input power. *Polarization sensitivity* gives the gain difference between different polarizations. *Amplifier noise* is mostly the Amplified Spontaneous Emission (ASE) of the amplifier.
  The two basic types of line amplifiers are Semiconductor Laser Amplifiers and Erbium-Doped Fiber Amplifiers. Although the latter ones are better amplifiers, the former ones, which were developed earlier, are still used in other applications such as in switches and wavelength converter devices.



(a) Semiconductor Laser Amplifier (b) Erbium-Doped Fiber Amplifier

Figure 3.6: Optical Amplifiers

  - *Semiconductor Laser Amplifiers (SLAs)* Figure 3.6 (a) shows the block diagram of a SLA, which is basically a pn-junction. The layer formed at the junction acts as an *active region* where the amplification is done. SLAs have a broadband gain characteristic which is the most positive feature of the device. The main drawbacks are the addition of crosstalk effects at gigabit rates, the polarization dependence and the coupling losses.
  - *Erbium-Doped Fiber Amplifiers (EDFAs)* Figure 3.6 (b) presents an EDFA that consists of an Erbium-Doped fiber, pumped by a laser typically at a wavelength of 980 nm or 1480 nm. Between the pump laser and the doped fiber, there is a coupler that combines the output of the laser with the input signal. After the doped fiber there is a second coupler which separates the amplified signal from the remaining pump optical power.

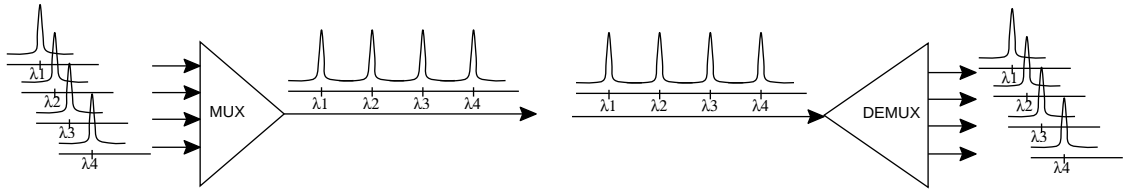
  EDFAs have been chosen in many communication systems because of several factors such as the availability of compact and reliable pump lasers, the independence to polarization, the avoidance of crosstalk addition and the facility to couple light in and out of the amplifier. But whereas the electronic amplification re-creates a perfect output signal, the fiber amplifier amplifies all what it receives, that is, pulse spreading and other effects can accumulate along a transmission path [5].

Line amplifiers have been described in this section. Two other kinds of amplifiers are used to adapt the optical signal within certain conditions. This is the case of *boosters* whose

function is to adapt the optical signal before sending it through the optical channel and of *pre-amplifiers* whose role is to adapt the optical signal before delivering it to the receiver.

3.2.6 Multiplexers, Demultiplexers and Add-Drop Filters

Multiplexers (MUXs) and Demultiplexers (DEMUXs) are devices used in WDM terminals as well as in wavelengths routers and filters. A Multiplexer is a device able to combine optical signals at different wavelengths into one optical fiber whereas a Demultiplexer performs the opposite, that is, separates different wavelengths from one optical fiber into independent optical signals (see Figure 3.7).



(a) Multiplexer (b) Demultiplexer

Figure 3.7: Functioning of Multiplexers and Demultiplexers

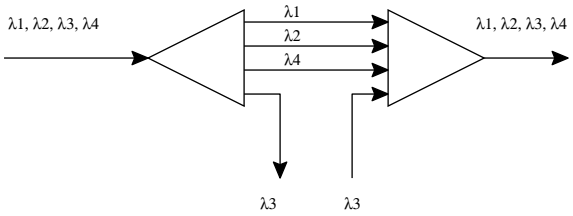


Figure 3.8: Filter built with MUXs and DEMUXs

An optical component that can be implemented with MUXs and DEMUXs is the Add-Drop Filter (ADF) shown in Figure 3.8. The main feature of these filters is their passband, which should be both as selective as possible to minimize the crosstalk with neighboring channels (see Figure 3.9), and as flat as possible to ensure that the overall passband at the output of a cascade of ADFs is still reasonably broad band. The two most important characteristics of the passband is their central frequency and their bandwidth (at 3 and 20 dB). In addition to the low crosstalk increase and the low flatness of the passband, the key characteristics of the filters are the need of having a low insertion loss, the polarization independence and the passband insensitivity to temperature variations. The bandpass filters can also be implemented by using multimoded overlays on polished fiber half blocks [28], or by using tunable receivers [26].
The Add-Drop Filters are used in networks where each node has to retrieve the data addressed to itself and modulated with its wavelength from a WDM signal. In future optical networks, not only one but several wavelengths will be added-dropped, thus making channel routing possible. This component is still under research and it is called Add-Drop Multiplexer.

3.2.7 Switches

These are the components that allow the cross-connection, i.e., the association of a particular input with a particular output. They can be classified as follows:

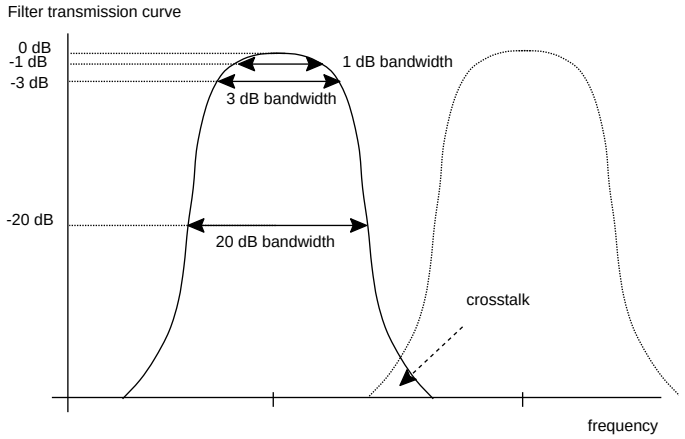


Figure 3.9: Filter Bandpass parameters: crosstalk measured with the overlapping of the neighboring channel and the flatness of the bandpass measured by the 1dB bandwidth.

- *Electrical switches* need to be preceded by receivers to convert the optical signal to electrical and to be followed by transmitters before sending the signal through the optical fiber. They have been on the market for a long time. Most of them are dynamic with a low switching time.
- *Optical switches* perform cross-connection without wavelength conversion. They are managed by the WDM layer and are therefore, transparent to upper layers. Two kinds of optical switches can be distinguished:
  - *Static optical switches*: The function of a wavelength static router is to exchange wavelengths between different input ports into different output ports. The mapping between input and output ports is not configurable. It can be implemented with MUXs and DEMUXs, as shown in Figure 3.10.

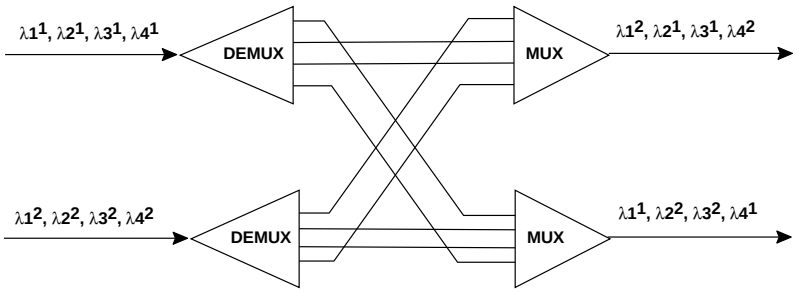


Figure 3.10: 2x2 Optical Router with 4 wavelengths

  - *Dynamic optical switches*, also called, optical cross-connects (OXC): They are new devices that can realize any mapping between input and output in the optical domain without needing an electrical conversion. The mapping is configurable.

3.3 Examples of failures in different hardware components

Hardware components usually have a micro-controller informing the manager about the status of the component and able to act over the hardware component when needed or requested by the manager. Except in some special cases, the information that the micro-controller delivers to the manager is binary (for example, a micro-controller of a laser will send an alarm if its temperature exceeds a given threshold, but will not send the value of the temperature). By disabling the micro-controllers to continuously send information to the manager, one avoids overloading the management communications network. If the manager needs the exact information, he/she may request the value from the micro-controller as we have seen in Section 2.3. We present some examples of failures of different hardware equipment.

3.3.1 Optical Fiber's Failures

Optical fibers are installed underground. It is quite frequent to have fiber cuts with an average of 1 fiber cut per day in the Unites States, due to agricultural engines, natural disasters or even animals. Also, with an average of one repair every five weeks [29], submarine cables are vulnerable to damage from anchors, fishing gear and submarines. The quantity of information lost in this situation is enormous, as are the number of alarms sent by other equipment able to detect the loss of signal.
Also, optical fibers can suffer from some impairments and defects as listed below.

- Attenuation: Although ideal fibers are a lossless medium, real fibers have attenuation losses due primarily to three effects:
  - material absorption, due to resonance of silicon molecules, as well as impurities of the fiber;
  - Rayleigh scattering, due to the non-uniformity of the medium, which causes variations of the refractive index. Because of this effect, the light is scattered so that the transmitted signal becomes attenuated;
  - waveguide imperfections, due to the imperfections of the fabrication of the fiber and to small bends and distortions in the fibers.

  The overall attenuation is given by the following coefficient, which is expressed in decibels per kilometer:

$$\alpha_{dB} = -\frac{10}{L} log_{10} \frac{P_R}{P_T} \tag{3.1}$$

  where L is the length of the fiber in kilometers, P_T is the power launched into the fiber and P_R is the power received at the end of the fiber of length L. This coefficient is a function of the fiber and of the wavelength, as shown in Figure 3.2. Hence, attenuation is a known parameter once the wavelength is chosen.
  There are two main windows where the attenuation reaches a local minimum and that have been used in communication systems: the first window is centered at 1.3 μm, has a width of 200 nm and an attenuation around 0.7 dB/km, whereas the second window is centered at 1.5 μm, has a width of 200 nm and an attenuation of 0.3 dB/km.

- Dispersion: A narrow pulse launched in a fiber tends to get wider as it propagates along the fiber. When a pulse broadens to the point that it overlaps neighboring pulses, the resulting inter-symbol interference (ISI) increases, and so does the Bit Error Rate (BER). Therefore, the phenomenon of dispersion limits the bit rate to a value, depending on the length of the dispersive fiber. Several modes of dispersion are present in optical communication systems. The most important ones are:
  - Modal dispersion, which appears only in multi-mode fibers where the different modes travel with different velocities [30]. To avoid this phenomenon, single-mode fibers are used in communication systems.
  - Polarization-mode dispersion, which is created when the fiber core is not perfectly circular, so that different polarizations of the signal travel with different group velocities. This phenomenon can be avoided using polarization independent equipment, such as optical filters, that have the same curve for all polarizations of the light.
  - Chromatic dispersion, which arises because signals at different wavelengths travel with different group velocities and thus arrive at the end of the fiber at different times. This dispersion is still an open issue.
- Nonlinear Effects: Although these effects are usually very small, they can become important over long, amplified, but non-regenerated links. The more important nonlinear effects are Stimulated Raman Scattering, Stimulated Brillouin Scattering, Four-wave mixing and Self- and cross-phase modulation. These effects increase with the number of simultaneous channels and with a narrower channel spacing.

Some of the losses of the fibers are due to two factors:

- Splice losses: This defect is due to the light losses when there is a junction between two fibers.
- Bending losses: These losses appear when the fiber gets bent or when it receives an external pressure.

3.3.2 Transmitter's Failures

Transmitters send alarms when either the temperature or the incoming power is beyond a prescribed range. In fact, for each variable (temperature or power) there are two ranges (see Figure 3.11): the first one, Ma, delimits the values for which the transmitter works correctly. Once the upper or lower margin is crossed by either the incoming power or the temperature, an alarm is sent informing that the corresponding variable is too high/too low, and hence that the emitted signal may be incorrect. When the larger range Mt is exceeded, not only a new alarm is sent but the transmitter is turned off, thus preventing any damage to the network.
In the example of Figure 3.11, at time t_1 an alarm is sent informing that Ma has been exceeded. Due to the local temperature or the power control of the board, the value is back to the expected range at t_2 and another alarm is sent to cancel the first alarm. Some time later, at t_3, Ma is again exceeded and a new alarm is sent. At t_4 the range Mt is also exceeded, a new alarm is sent and the transmitter is turned off (because the local control mechanism of the board has not been able to restore the normal behaviour).

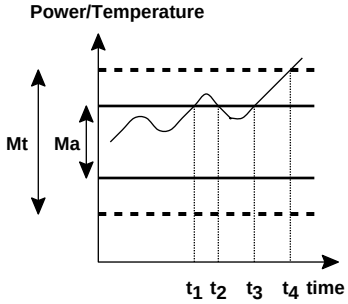


Figure 3.11: Transmitter Margins

3.3.3 Failures perceived by optical detectors

Optical detectors have two characteristic parameters: the sensitivity and the overload parameter. The sensitivity is the average optical power required to achieve a certain BER at a particular bit-rate, whereas the overload parameter gives the minimum input power that the receiver can accept (typically -7dBm).
Receivers are able to detect increases or decreases of the input optical power. Indeed, when the input optical power exceeds the accepted threshold, an alarm is sent to the manager (as the transmitter with the power or the temperature variation).

3.3.4 Optical Amplifier's Failures

The alarms from optical amplifiers depend on their implementation and the kind of amplifier but the two main ones failures are that

- (i) there is not enough input power to amplify,
- (ii) the pump laser is off or does not work properly.

In this case, this element is able to inform about external(i) and internal(ii) problems.

3.3.5 Add-Drop Filter's Failures

Add-Drop Filters have as a main characteristic its bandpass. This characteristic may degrade with the time or environmental conditions. Therefore, the optical power of the dropped signal can fall under the allowed margins so that can provoke alarms from other equipment, such as receivers or regenerators, that can not lock onto the incoming signal.

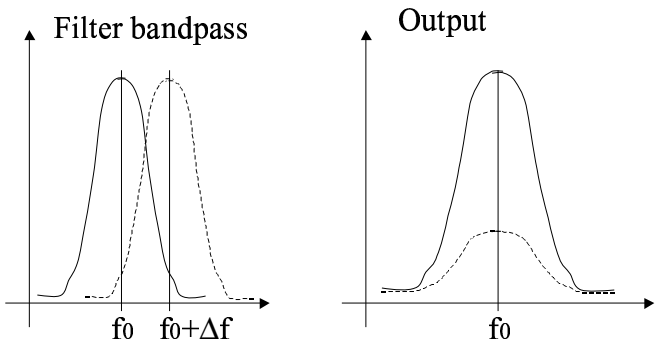


Figure 3.12: Example of a failure in an ADF

This is for example the case shown in Figure 3.12 when the central wavelength of a filter gets shifted: the optical power at the output decreases considerably and may fall under the allowed values. For this reason, these filters usually inform about the stability of their characteristic parameters: their central frequency and their bandwidth.

3.3.6 Switch's Failures

Both electrical and optical switches are able to send alarms when they have an internal problem. Examples of internal problems are the impossibility of the switch to connect or disconnect the requested pair (input, output) or the reset due, for example, to temperature problems.

3.4 Monitoring Equipment

Let us present the equipment used both at the WDM and higher layers to measure the quality of the transmitted signal. These measurements can give more accurate information than the hardware components since the latter can only send binary alarms based on optical power and other basic physical characteristics of an optical signal, whereas the monitoring equipment is able to provide not only binary but also analog information.

3.4.1 WDM Layer

Devices measuring directly the quality of the optical signal can be divided into the two following categories:

- Global Testing Equipment (GTE) measures the quality of the *overall* optical signal. This is for example the case of *Spectrum Analyzers* and *DWDM Monitors*. GTEs are equipment able to analyze the frequency characteristics of the incoming signal and display them on a graphical interface. We will focus on the spectrum analyzers since they are the most commonly used.
  The basic operation of a spectrum analyzer is the separation of a signal into multiple single signals using filters and the measurement of the amplitude of each of them using detectors. The main characteristics are:
  - Frequency Range, i.e. the range of frequencies where the spectrum analyzer can be used.
  - Frequency Span, i.e. the frequency scan width that can be monitored.
  - Frequency resolution, i.e. the minimal separation between two adjacent signals that allows them to be analyzed separately.
  - Average noise, i.e the mean value of the internal noise added by the spectrum analyzer
  - Sensitivity, i.e. the minimum detectable input signal that can be measured.

  The information spectrum analyzers can provide is:
  - *Frequency measurement* It returns: the power at each wavelength, the location of each channel, the bandwidth at 3dB, the channel spacing, etc.
  - *Time measurement* It shows: the signal in the time domain, the bursts, the transient response, etc.

The spectrum analyzers can be electrical or optical. Examples of electrical spectrum analyzers are the new models MS2665C and MS2667C developed by Anritsu [31] or the HP 70000 series of Hewlett-Packard [32]. Examples of optical spectrum analyzers are MS9720A and MS9715A developed by Anritsu [31], the Walics analyzer by Photonetics [33], the HP86142 by Hewlett-Packard [32] and the OSP102A by Wandel & Goltermann [27]. The parameters measured by this equipment, shown in Figure 3.13, are mainly the Signal to Noise Ratio (SNR), the used wavelengths, the maximum power at each wavelength, the gain slope and the wavelength and power stability based on a time measurement.

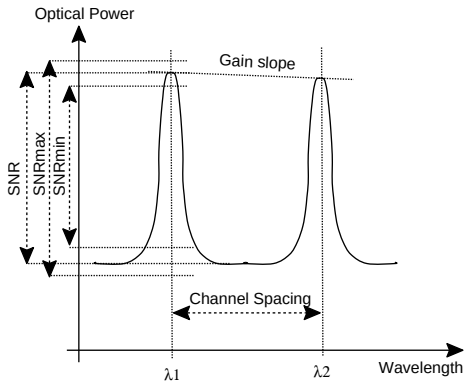


Figure 3.13: Parameters measured by an optical spectrum analyzer of a WDM signal

- Individual Testing Equipment (ITE) designates a device that can measure parameters relevant to a *single* channel, that is, of a single wavelength. Two examples are the network tester ANT-20 by Wandel & Goltermann [27], which can calculate the BER of a given channel and the SDH/PDH/ATM Analyzer MP1552A manufactured by Anritsu [31], which can evaluate the equipment of PDH, SDH and ATM networks as well as provide performance functions by detecting management signals and monitoring the traffic. The drawback of this monitoring equipment is that they depend on the transmission technology used (ATM, SONET, SDH, etc.). This detailed knowledge of the upper layer should be avoided since one of the advantages of WDM networks is transparency with respect to the transmission technologies deployed.

3.4.2 SDH/SONET Layer

Synchronous Digital Hierarchy (SDH) and Synchronous Optical NETwork (SONET) are a set of network interface standards and multiplexing schemes developed to support the adoption of optical fiber as a transmission medium. SDH is the European standard and SONET is the US counterpart. The main difference between SONET and SDH is that SDH adds additional network management information to each data frame. For simplicity we refer only to SDH, but the concepts apply to SONET as well.

Introduction

Prior to SDH and SONET, the existing infrastructure was based on the Pleosynchronous Digital Hierarchy (PDH). At that time, the analog voice signals were sampled and quantized at 8 bits per sample, leading to digital voice signals of 64 kbps. Several frames were

time-multiplexed to achieve higher rates. However, PDH had some drawbacks such as the complexity of multiplexers and demultiplexers, the difficulty to pick out a low bit-rate stream from a high bit-rate frame (to retrieve a particular low bit-rate stream, not only this stream but all the low bit-rate streams had to be demultiplexed) and the restoration time in case of a failure. All these difficulties led research into the direction of new multiplexing and transmission standards: SDH and SONET.
SDH offers the following advantages:

- High transmission rates (up to 10 Gbps in modern SDH systems) enabling SDH to be the technology used in backbones.
- Simplified add and drop of the channel, which avoids demultiplexing and multiplexing all the channels.
- High adaptability, which allows providers to react quickly to the customer demands. Providers can also use standardized equipment that can be monitored through a telecommunication network management system.
- Reliability due to the implemented automatic restoration in case of failures. Although the communications are restored, the failure has to be identified, which is the goal of the fault management.

SDH tends to work at high bit rates (10 Gbps). The high cost and the difficulties achieving synchronization at these rates makes WDM a necessary solution. WDM network examples with 16 wavelengths transmitting 2.5 Gbps frames at each wavelength, offer a capacity of 40 Gbps. Communication systems using 32 and even 64 wavelengths have been already announced.

SDH layers

The SDH layer can be divided in three sublayers:

- *Regenerator Section (RS) layer:* This is the layer common to all SDH equipment where the signal is regenerated, that is to say, where the frame is re-timed and reshaped. The segments between each pair of SDH equipment at the RS layer are called Regenerator Sections.
- *Multiplexer Section (MS) layer:* This is the layer where the channels are multiplexed and demultiplexed. The segments between each pair of SDH equipment at the MS layer are called Multiplex Sections.
- *Path layer:* This is the higher layer where the Virtual Containers (VCs), which will be defined later, are obtained. Some of the functions done at the Path layer are: VC path performance monitoring, signals for maintenance purposes and alarm status indications.

SDH Frame

The information unit used in SDH is called an STM-N frame, where STM stands for Synchronous Transport Module and where N is the number of basic STM in one frame, N can be equal to 1, 4, 16 or 64. The frame duration is 125 μs.
The STM-1 frame is made up from a byte matrix of 9 rows and 270 columns as shown in Figure 3.14. This matrix contains a section overhead, a pointer and a payload.

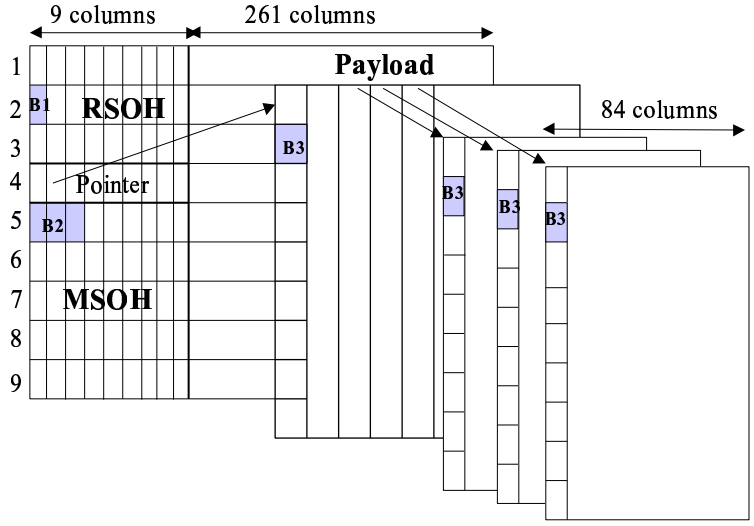


Figure 3.14: Sketch of an SDH STM Frame

- The section overhead is formed by the Regenerator Section Overhead (RSOH) and the Multiplexer Section Overhead (MSOH). These two overheads have some bytes called B-bytes that perform error control, as explained later. The first byte of the second row of RSOH is reserved for the B1 byte, which performs error control at the RS; whereas the three first bytes of the first row of MSOH are reserved to store the B2 bytes that perform error control at the RS.
- The pointer directs to the Path overhead (POH) start.
- The payload is a Virtual Container (VC) that is formed by a path overhead (POH) and a container. The container can be a 140 Mbps signal or several smaller VCs, which at the same time can be either a POH and container or a set of even smaller VCs. Figure 3.14 shows the case when the container is made from 3 smaller containers. The set of the first byte of all the rows of the VC form the POH. The byte of the second row is the B3 byte, which performs the error control of the VC.

SDH Components

Based on the layers that each component contains, three kinds of SDH elements can be obtained (see Figures 3.15 and 3.16):

- *Regenerator Section Terminating Element (RSTE)* is the element, which originates/terminates a Regenerator Section (RS). At one RSTE, the Bit Interleave Parity byte (BIP-8) is computed over all bits of the previous frame and stored at the B1 byte of the RSOH of the following frame. The next RSTE checks the byte parity of the preceding frame and detects errored blocks. Before re-transmitting the frame on the next RS, the value of the B1 byte is updated to recover the correct parity, thereby masking any detected error in the frame from the next RSTE. The characteristic information are STM-N frames as defined in G.707 [34].
- *Multiplex Section Terminating Element (MSTE)* originates/ terminates a Multiplex Section (MS). In the SDH layering, every MSTE covers both the RS and MS layers.

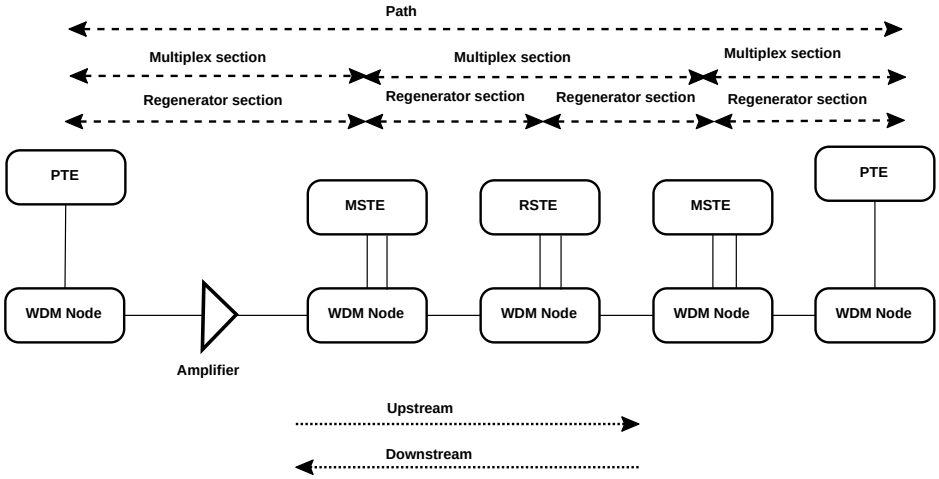


Figure 3.15: SDH layered structure and equipment

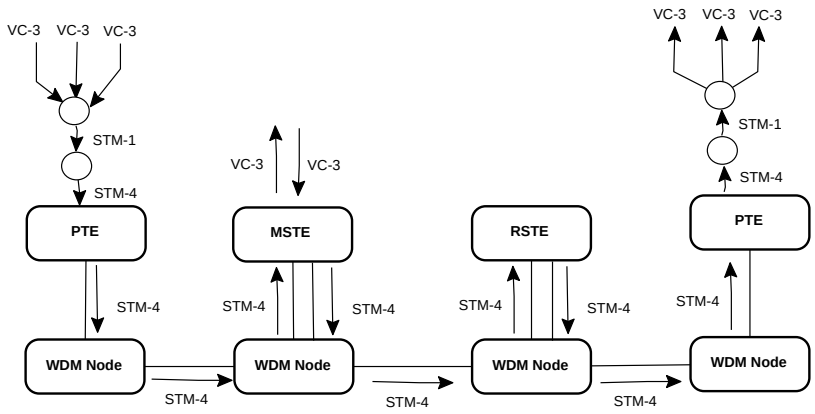


Figure 3.16: SDH equipment and their functions

The error monitoring function at the RS layer is the one of an RSTE, while the error monitoring function at the MS layer is performed by a BIP-24N code using even parity, as defined in G.707 [34]. At one MSTE, at the MS layer, the BIP-24N is computed over all bits of the previous frame (except those in the RSOH) and is placed in the 3xN respective B2 positions of the MSOH of the following frame. At the next MSTE, BIP-24N code is computed for the received frame and they are compared with the 3×N error monitoring B2 bytes recovered from the multiplex overhead [35] of the next frame. A difference between both values is an evidence of an errored block. Before re-transmitting the frame, bytes B2 are updated, thereby masking any detected error in the frame to the next MSTE. This layer consists of several Multiplex Sections whose characteristic information are STM-N frames, as defined in G.707 [34].

- *Path Terminating Element (PTE)* originates/terminates a Path. Its BIP is stored in the B3 bytes of the header. This is the last parity check before delivering the frame to the upper layers.

The error detection codes used at each layer are listed in Table 3.1. The number of errored blocks are counted continuously over two fixed 15-min and 24-h windows. The count is compared at every second to a threshold. Whenever the threshold is crossed, a notification

Level		No. bits/block	Error Detection Code
VC	VC-11	832	BIP-2
	VC-12	1120	BIP-2
	VC-2	3424	BIP-2
	VC-3	6120	BIP-8
	VC-4	18792	BIP-8
MS	STM-1	19224	BIP-24
	STM-4	76896	BIP-24x4
	STM-16	307584	BIP-24x16
RS	STM-1	19440	BIP-8
	STM-4	77760	BIP-8
	STM-16	311040	BIP-8

Table 3.1: This table gives the block size and the associated error detection code for each VC, MS and RS [36].

is sent to the manager and an Alarm Indication Signal (AIS) is sent to the following SDH equipment. This AIS informs the next equipment that there has been a failure so that the equipment does not have to sent any alarm to the manager. Due to this early filtering, the manager does not get overloaded by alarms that only provide redundant information about the failure. Moreover, at the end of the current window, the count is reset to zero, after having been registered in a file.

Forward Error Control Mechanisms

In long high capacity point-to-point optical links such as submarine cables, Forward Error Control (FEC) mechanism is used to correct the errors (see Figure 3.17) [37]. This mechanism works over defined data modules such as STM-16 frames for SDH submarine cables.

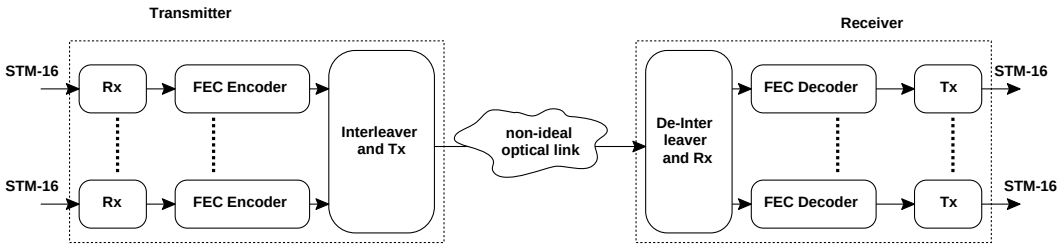


Figure 3.17: Block Diagram of a submarine system using FEC function

- The FEC encoder in the transmission equipment adds redundant bits to the payload. Therefore, the encoded data has to be sent at a higher bit-rate.
- A FEC decoder in the receiver equipment performs error correction while extracting the redundancy to regenerate the corrected payload.

The FEC code used to protect the STM-16 information is a Reed-Solomon code, which is a non-binary code. This code was chosen due to several advantages such as the important error correcting capacity (8 erroneous bytes in a single codeword of 255) and the low complexity of the needed encoder and decoder, among others.

The implementation of a FEC function allows in-line monitoring of the BER. If we consider the relation BER_{input}=$BER_{corrected}$+BER_{output} where BER_{input} is the BER before the FEC, $BER_{corrected}$ is the BER corrected by the FEC and BER_{output} denotes the BER of the non corrected errors. Normally, $BER_{input} \simeq BER_{corrected}$ except when the BER_{input} is less than 10^{-3} [37]. In this case, there will be an intermittent loss of FEC frame alignment.

3.4.3 Other Layers in Optical Communication Networks

In this section a summary of error control techniques performed in other layers than SDH/SONET is given. This study describes the error information that can be obtained at each of these layers. All this information is analog. We will focus on IP, following the trend of today's telecommunication networks.

Data Link Layer

Data Link layer is the second layer in the OSI model. Its main task is to receive information from the physical layer and to ensure an error-free transmission. Several protocols have been implemented to avoid lost, damaged or duplicated data. The most popular is Ethernet. It includes error detection capabilities by using a 32-bit Cyclic Redundancy Check (CRC) code. The generator polynomial is $x^{31}+x^{26}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x+1$. The CRC data packets consists on n bits where the first k bits are the original data and the $n-k$ bits are the redundancy. In Ethernet, n can range from 512 to 12144 and $n-k$ is 32. The transmitted packet can be expressed using the equation $T(x) = x^{n-k}C(x) + R(x)$, where $R(x) = x^{n-k}C(x)/G(x)$. $C(x)$ is the original data and $G(x)$ is the generator polynomial. The receiver reads in the message and computes the checksum of the k fist bits. If the received checksum is equal to the calculated one, no error is detected. If an error is detected, the frame is discarded and a re-transmission is requested.

ATM Layer

ATM is divided in two layers: the ATM Adaptation Layer (AAL) and the ATM Layer itself. AAL consists in three different AAL sublayers depending on the kind of data: AAL1, AAL3/4 and AAL5 [38].

- AAL1 offers connection-oriented services, and it is suitable for handling circuit-emulation applications, such as voice and video conferencing.
- AAL3/4 supports both connection-oriented and connectionless data, and provides switched data services. It creates protocol data units (PDU's) by prepending a beginning/end tag header to the frame and appending a length field as a trailer. If fragmentation is necessary, an error control is done by calculating the CRC-10 and appending it to each fragment. The generator polynomial is $x^{10}+x^9+x^5+x^4+x^1+1$. Errors can be detected when assembling the segments and then, a request to re-transmit the frame is sent.
- AAL5 is the primary AAL for data and it also supports both connection-oriented and connectionless data. It is used to transfer classical IP over ATM and LAN Emulation. A 32-bit CRC computed across the entire PDU is used for detecting bit errors. The generator polynomial is the same than the one used in Ethernet.

A variable that could be used for fault performance is the ATM Quality of Service parameter called *Cell Error Ratio*, which gives the ratio between the errored cells and the addition of successfully transferred cells and the errored cells [39]. This variable is expected to be primarily influenced by the error characteristics of the physical media. There is another variable called *Cell Loss Ratio* that gives the ratio between the lost cells and the total transmitted cells. This ratio depends on the errors in the cell header and the buffer overflows. This variable is influenced by errors at different layers and therefore it is not useful for our needs.

IP Layer

At IP layer (IPv4), routers can be considered monitoring equipment since they perform a header checksum before re-transmitting the packets. At such as IP router, a 16-bit one's complement sum of the header is calculated and compared with the checksum field of the received packet. If they are not equal, the packet is discarded and no error message is generated (it is up to the higher layer, such as TCP, to detect the missing datagram and ask for its retransmission). The checksum field is updated whenever the packet header is modified by a router. In this way, the packet is protected from undesirable modifications of the packet when the packet is not protected by the data link CRC check. The number of discarded packets is counted and stored in the MIB variable *ifInErrors* [11].

TCP and UDP Layer

At the TCP and UDP layers, the kernels of the end systems can be considered monitoring equipment since they perform a checksum of data and the header before accepting a packet. It is an end-to-end checksum whose purpose is to detect any modification of the data in transit. TCP checksum is mandatory whereas UDP checksum is optional. Both TCP and UDP checksums are calculated using a 12-bytes pseudo-header [40]. This pseudo-header contains fields from the IP header to allow a double check that the data has reached the right destination. The count of errored packets is stored in the MIB variables *tcpinErrs* and *udpInErrs* [11]

3.5 Conclusion

In this chapter we described the most common components in optical communication systems. We first described the hardware components at the physical layer, which are components whose failure has to be identified and that may or may not provide information about these failures. We then described monitoring equipment, i.e. equipment able to provide useful information about failures, at the WDM layer (spectral analyzers,...), at the SDH/SONET layer (BIP, EB,...), at the ATM layer and the TCP/IP layers. The failure of the monitoring equipment is beyond the scope of this work since each layer has its own methods for the failure location of their components.

Chapter 4

Alarm Filtering Algorithm (AFA)

4.1 Introduction

In this chapter we present a model based algorithm that solves the fault location problem on optical networks introduced in Section 2.4. The algorithm is called Alarm Filtering Algorithm (AFA).
Our modelization of an optical network begins with the classification of the optical components into categories depending on their behaviour when failures occur. This classification is given in Section 4.2. Based on this classification, an abstraction of the problem is given in Section 4.3 by introducing the information needed to solve the fault location problem, the expected result, and the most relevant concepts on which the algorithm is based.
Then, Section 4.4 describes the AFA algorithm [41]. This algorithm locates single and multiple failures at the physical layer. Indeed, when a single network component fails, all channels passing through this element are interrupted. Consequently, all the elements that were involved in the interrupted channels and that are able to send alarms to the manager will report a problem to him/her. The messages from these elements will be different and the manager will have to determine where the failure is. This becomes more complex when multiple failures occur almost simultaneously. In this case, alarms due to different failures will reach the management application during the same period of time and they will mix. As we will see later, to increase the filtering capacity of the algorithm, we will sometimes use the (reasonable) assumption that a number n of failures is more probable that a number $n+1$ of failures.
The AFA performance becomes more difficult to achieve under non-ideal conditions. Two abnormal cases can arise: the existence of (i) missing alarms (alarms that do not reach the manager) and (ii) false or unexpected alarms. The first case occurs when some alarms are lost or arrive with such a delay that they cannot be considered during the ongoing computation of the algorithm. The second case occurs when, due to an abnormal situation, an element sends an alarm although there is no real failure. These alarms are not caused by a malfunction but by an exceptional problem, for example, an incorrectly measured value. To handle these cases, a *mismatch threshold* is given (which was previously introduced in Subsection 4.3.1).
One of the main features of the AFA algorithm is that it filters the received alarms in order to work with a reduced set, that is, it discards what are called redundant alarms (alarms that do not provide additional information about the failure). Another feature that allows the location of multiple alarms with the existence of false and/or lost alarms,

is the combination of two approaches: *backward* and *forward*. This combination does not however minimize the worst-case computation time when alarms reach the manager, as does our second proposed algorithm (developed in Chapter 5).
Before concluding with the main results, Section 4.5 gives some applications of the AFA algorithm to two different network topologies.

4.2 Classification of optical network components

Walrand [42] defines communication networks as a set of nodes that are interconnected to allow the exchange of information. In this case, the term node can refer to bridges, switches and access ports. Such a definition is well suited for traffic control but is not very helpful for fault management because the view of the network is different. Fault management needs to know which are the network components, what are the alarms and their content and in which situation these alarms are sent. A model of optical communication networks used to demonstrate AFA is presented in this section. This model is based on channel relations between network components that may fail.

4.2.1 Channel Relation

Channels are established between nodes that want to exchange information and can be unidirectional or bi-directional. In our model we will consider channels to be unidirectional because a bi-directional channel can always be considered as two unidirectional channels.

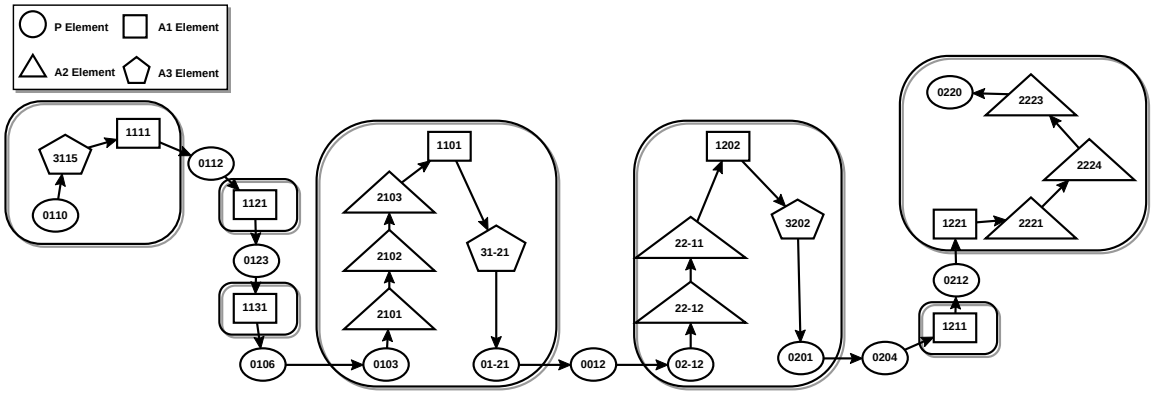


Figure 4.1: Channel between two WDM rings

Unidirectional channels are ordered sets of network components. They correspond to the Telecommunication Management Network (TMN) M.3100 [43] *trail* concept where the *Trail termination Source point* is the sender access port and the *Trail termination Sink point* is the receiver access port. For example, the channel shown in Figure 4.1 is an ordered set of 30 network components where the first one is (0 1 1 0), the 14th one is (3 1 -2 1) and the last one is (0 2 2 0). In this example, the reason the components have a four digit identifier, is due to the identifiers used in the COBNET [1] (Corporate Optical Backbone NETwork) and ARPA2 networks, as explained in Section 4.5. The explanation about the component classification as P, $A1$, $A2$ and $A3$ elements will be given in Section 4.2.3.

4.2.2 Alarms

Alarms are messages sent to the manager by the components of a network indicating an abnormal condition (e.g. some parameter of a component out of range, or a missing signal). As we will see in detail in the next subsection, only the so-called 'alarming' components can send alarms to the manager. The content of these messages depend on the component [44] but always contains the identification of the component that sent the alarm, the nature of the alarm, the abnormal value of the parameter and/or the time when the alarm was sent (timestamp). The timestamp can be useful in locating the failure but it can also lead to errors when there are multiple failures and has therefore not been considered in our analysis. Nevertheless, this parameter could be taken into account in other versions of the algorithm. Our algorithm needs only the minimal amount of information present in all the alarms, that is, the alarm origin and its nature.

4.2.3 Alarming Properties of hardware components

This section gives a classification of the hardware components of optical networks that were presented in Chapter 3. The classification has been based on the alarming properties of these components, that is, on the behaviour of these components when a failure has occurred. Three features can be distinguished:

Self-alarmed This property specifies whether a network component is able to send an alarm informing about its *own failure* or not. This alarm corresponds to the *equipment alarm* specified in X.721 [45]. An example of a self-alarmed component is a transmitter whose micro-controller controls the power and the temperature and sends an alarm whenever one of these parameters exceeds a given threshold.

Out-alarmed This property applies to the components that communicate with the manager and send alarms about a *failure external* to them. These alarms correspond to the *communication and environmental alarms* specified in X.721 [45]. For example, receivers are able to detect that there is no incoming power and send the corresponding alarm to the manager even if they themselves are working correctly. On the contrary, multiplexers are unable to detect whether some inputs are missing, hence they are not out-alarmed components.

Failure masking This property specifies whether the considered network component masks the failure from the hardware components that follow it on the channel (remember that a channel is an ordered set of components as explained in Section 4.2.1). For example, the laser of a transmitter sends power even if there is no incoming signal (due to a failure of some component located before the laser). Therefore, any out-alarmed component located after this transmitter on the channel will not send any alarm because it will keep receiving power, even if it does not receive data any more.

We can now check which of these three properties applies to each of the kinds of optical components presented in Chapter 3. This analysis is based on the capabilities of real optical components, in particular, on the components of the COBNET optical network. The results of this analysis are summarized in the three central columns of Table 4.1.
The properties of Table 4.1 enable us to classify optical components in the following *categories*:

Network Component	Self-alarmed	Out-alarmed	Failure masking	Category
Optical Fiber	No	No	No	P
Transmitters	Yes	No	Yes	$A3$
Receivers	No	Yes	No	$A2$
Add/Drop Filters	Yes	No	No	$A1$
3R	No	Yes	No	$A2$
Protection Switch	No	Yes	No	$A2$
MUX/DEMUX	No	No	No	P
Switch	Yes	No	No	$A1$
Optical Amplifier	Yes	Yes	No	$A1$ and $A2$

Table 4.1: Alarm properties of the Network Components and the resulting classification

1. **'Non-alarming' components**: These are the components that do not give any information to the manager because they do not have any micro-controller. They are denoted by P and they are represented in the figures by a circle.

2. **'Alarming' components**: These are the components that are able to communicate with the manager because they have some programmable software at the computer that controls the equipment. This group contains three subgroups:

   (a) The $A1$ **components** are the self-alarmed components that do not mask any kind of failure. They are represented in the figures by a square.

   (b) The $A2$ **components** are the out-alarmed components. They are represented by a triangle.

   (c) The $A3$ **components** are the components that are self-alarmed and mask previous failures. They are represented by a pentagon.

   If a component has both self and out-alarmed properties, it will be represented by a tandem of an $A2$ element followed by an $A1$ element. This is the case of an Optical Amplifier.

Comparing this classification with the TMN standard M.3100 [43], the $A3$ elements correspond to the *source* components. The $A2$ elements may correspond to the *sink* components but not exclusively.
In this way, all the optical components have been classified as presented in the last column of Table 4.1.

4.3 Problem abstraction

The classification of Section 4.2 enables us to derive and implement the Alarm Filtering Algorithm (AFA). Before describing the AFA itself, we first need the following definitions.

4.3.1 Alarm Mismatching thresholds

Some sets of received alarms may remain unexplained by any combination of failing components. This means that at least one alarm was lost or false. To cope with this, we introduce the *alarm mismatching threshold* parameter that reflects the reliability of the

management channel and of the management functions of the equipment. This parameter, which is denoted by m, is the addition of two parameters (denoted by m_1 and m_2), which give the maximum number of respectively lost and false alarms that are tolerated. We will refer to the scenario where all the alarms are correctly issued and retrieved (no alarms are lost or false) as the *ideal* scenario. In this case, one takes of course $m = m_1 = m_2 = 0$. The value of m_1 and m_2 (thus of m) can be set a priori by the network manager. The availability to cope with non-ideal scenario is of crucial importance. We provide here some examples:

- When an $A1$ element fails, it is supposed to send an alarm. For example, a switch having an internal failure may not be able to send the expected alarm and therefore, its alarm will be considered as a lost alarm.
- At the physical layer, the alarms are binary but some of them are obtained by thresholding analog values such as 'Power_Out_Of_Range'. This can lead to errors, for example a false alarm being sent when no problem has occurred.
- When establishing a new channel, transient conditions may prompt an element to send an alarm, although no failure has occurred. The alarm will be considered as a false alarm.

4.3.2 Inputs of the algorithm

The objects manipulated by the AFA are:

- **Component *comp*** is a network component that belongs to one of the aforementioned categories. It has an identifier with two fields: the first one specifies the category and the second identifies the component within the category. The set of network components is denoted by $\mathcal{V}$ and its cardinality by n.
- **Alarm *a*** is an object with two fields: the first one is the component who issued the alarm and the second is the informational content of the alarm.
- **Channel** $CH_i = \{comp_j\}$ is an ordered list of components. The channels are considered here as unidirectional.
  Function $Pos(comp, CH_i)$ returns the position of $comp$ within the channel CH_i if $comp$ belongs to this channel, and 0 otherwise. In other words,

$$Pos(comp, CH_i) = \begin{cases} 0 & \text{if } \forall\, comp_j \in CH_i, comp_j \neq comp \\ j & \text{if } \exists\, comp_j \in CH_i, comp_j = comp. \end{cases} \tag{4.1}$$

The inputs of the algorithm are:

- the set of established channels, which is denoted by $\mathcal{CH}= \{CH_i\}$. This set is updated every time there is a channel is established, modified or cleared down,
- the set of alarms received by the manager $\mathcal{R}= \{a_j\}$. Every time there is a new alarm, the algorithm returns new results based on the new $\mathcal{R}$,
- the mismatching threshold $m = m_1 + m_2$.

4.3.3 Domain Definition

Domain(comp) is defined as the set of network elements that will send an alarm when *comp* fails. The computation of the domain of each network component is based on the established channels and use the two following functions:

- If an element e_1 suffers a failure, an out-alarmed component $e_2 \in A2$ of any established channel will send an alarm if both of them belong to the same channel and if there is no $A3$ element between them. Mathematically it can be expressed by the Boolean relation:

$$\begin{array}{l} e_1\ FP1\ e_2 = 1 \text{ if and only if} \\ \quad \bullet e_2 \in A2 \\ \quad \bullet \exists\, CH_i \in \mathcal{CH} \text{ with } 0 < Pos(e_1, CH_i) < Pos(e_2, CH_i) \\ \quad \bullet \forall e_j \text{ with } Pos(e_1, CH_i) < Pos(e_j, CH_i) < Pos(e_2, CH_i),\ e_j \notin A3. \end{array} \tag{4.2}$$

- An element e_1 will send an alarm when it fails if it is self-alarmed, which can be recast mathematically as:

$$e_1\ FP2\ e_2 = 1 \text{ if and only if } e_1 = e_2 \in A1 \cup A3. \tag{4.3}$$

Based on these two functions, we can define *Domain* as follows:

- $Domain(e_1)$ is the set of elements whose alarms are expected when e_1 fails. These elements are (i) e_1 itself if $e_1 \in A1 \cup A3$, and (ii) the $A2$ components that follow e_1 in at least one channel and do not have any $A3$ component between them. Mathematically, $Domain(e_1)$ can be expressed as follows:

$$Domain(e_1) = \{e_2 \in \mathcal{V} |\ (e_1\ FP1\ e_2 = 1)\ or\ (e_1\ FP2\ e_2 = 1)\}. \tag{4.4}$$

4.4 Alarm Filtering Algorithm (AFA)

The AFA has as inputs the established channels in the network $\mathcal{CH} = \{CH_i\}$ and the received alarms by the manager $\mathcal{R} = \{a_j\}$. These inputs are updated whenever a new event occurs: either a channel event, which updates $\mathcal{CH}$, or an alarm event, which updates $\mathcal{R}$. By channel event, we mean a modification of the set of established channels by the addition or deletion of a channel or by a change of some components involved in a channel. By alarm event, we mean the of a new alarm arrival to the manager. The two kinds of events are independent.
The AFA is based on the discarding of some of the received alarms. This discarding phase is presented in two different modules: (1) *Alarm_Discarding_1* and (2) *Alarm_Discading_2*. The former module filters, among others, the $A1$ and $A3$ alarms and the latter filters the $A2$ alarms.
The output of the algorithm is a list of different sets of network components that may be faulty and that are an explanation of the observed symptoms with up to m erroneous alarms.
The AFA combines two different approaches: the *backward* approach, which gives the fault candidates based on the received alarms $\mathcal{R}$; and the *forward* approach, which gives the alarms that will be issued when an element fails. These two approaches have been implemented in 4 modules (modules 1, 2 and 3 perform the *backward* phase and module

4 performs the *forward* phase). A last module combines the results from both approaches allowing the location of multiple alarms coping with lost and false alarms. Figure 4.2 shows the AFA scheme.

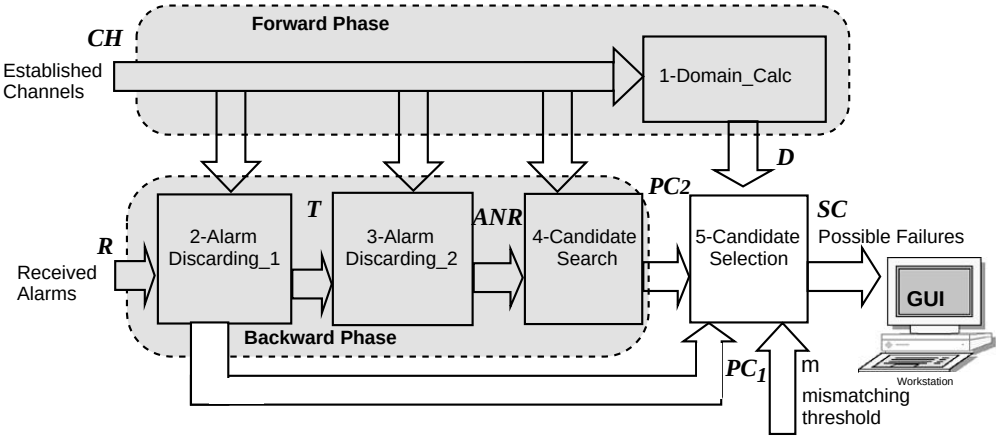


Figure 4.2: Scheme of the Alarm Filtering Algorithm (AFA)

4.4.1 Forward Phase

Domain_Calc

This module is called every time $\mathcal{CH}$ changes. It computes, for each element of each channel belonging to $\mathcal{CH}$, the set of *alarming* components that would send alarms if this element fails. This set is called *Domain*(e_0), for a given e_0 that belongs to $\mathcal{CH}$ and it is calculated based on the Function (4.4) presented in Section 4.3.3.

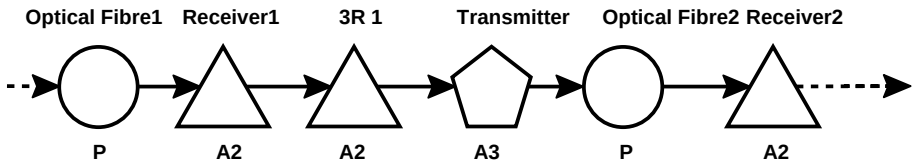


Figure 4.3: Second channel example

Let us give an example using the channel of Figure 4.3. If *Optical Fibre1* fails, *Receiver2* will not detect the failure because *Transmitter* is still sending optical power (even if it does not have any incoming data to modulate). In this case, *Domain(Optical Fibre1)*={*Receiver1* , *3R 1*} because *Receiver1* and *3R 1* are the two network components sending an alarm when *Optical Fibre1* fails.
The set $\mathcal{D}$ stores the list of the *Domain* of all the components that belong to each channel of the set $\mathcal{CH}$.

4.4.2 Backward Phase

This phase has been implemented in tree modules: *Alarm_Discarding_1, Alarm_Discarding_2* and *Candidate_Search*.

Alarm_Discarding_1

This module is the first alarm discarding phase of the algorithm. It produces a subset $\mathcal{T}$ of alarms from the received alarms $\mathcal{R}$, which contains the alarms having passed successfully the sequence of the three following tests, as shown in Figure 4.4:

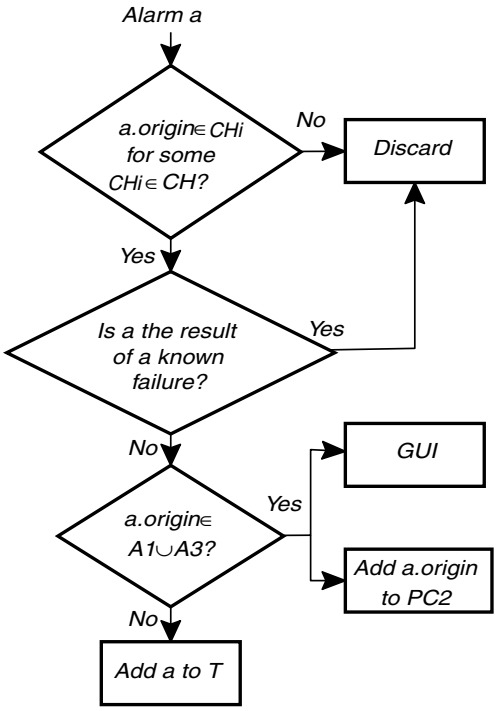


Figure 4.4: Alarm_Discarding_1 Module with the three tests that should be done before including an alarm in $\mathcal{T}$

1. The alarm must be sent by a component belonging to one channel $comp \in CH_i$. If this is not the case, the alarm is discarded because it has been generated by transient values reached by the network variables during the establishment of a channel.

2. If the new alarm is a consequence of an already known failure, it is discarded because it is a delayed alarm. This can be performed by storing the missing alarms that correspond to the known failure and by checking whether the new alarm is one of them. In this case the use of timestamps can be useful.

3. Is the alarm sent by a $A2$ component? Only in this case, the alarm will be added to $\mathcal{T}$. The alarms sent by $A1$ and $A3$ components are forwarded to the Graphical User Interface (GUI) to be presented to the Human Manager and included in the set $\mathcal{PC}_1$*(Possible_Candidates_1)*, which is input of the *Candidate Selection* module, so that the origin of the alarm is considered a likely faulty component:

$$\mathcal{PC}_1 = \{e \in \mathcal{V} \,|\, \exists CH_i \in \mathcal{CH} \text{ with } Pos(e, CH_i) \neq 0 \text{ and } \exists a \in \mathcal{R} \text{ with } e = a.origin\} \tag{4.5}$$

The *Alarm_Discarding_1* procedure is called each time a new event occurs. If the new event is a channel event, all the alarms $\mathcal{R}$ have to succeed the three sequential tests with

the new set $\mathcal{CH}$, in case one of the previously discarded alarms do indeed belong to the recently established channel. If the new event is an alarm event, the sequential conditions must be succeeded only by the new alarm in order for this one not to be filtered out as the other alarms are not changed.

Alarm_Discarding_2

This module is the second discarding phase. It has as input $\mathcal{T}$, the reduced set of received alarms obtained after *Alarm_Discarding_1*. Note that the set $\mathcal{T}$ contains only the alarms originated by $A2$ elements that belong to $\mathcal{CH}$ and are not the consequence of a known failure.
Some of these alarms can be discarded making the assumption that a number of n failures is always more likely than $n+1$ multiple failures. Indeed, consider two contiguous out-alarmed elements e_1 and e_2 belonging to a channel CH_i (i.e. such that $Pos(e_1, CH_i)+1 = Pos(e_2, CH_i)$). Suppose that an element e, with $Pos(e, CH_i) < Pos(e_1, CH_i)$, fails. If there is no $A3$ element between e and e_1, both out-alarmed elements e_1 and e_2 send an alarm. We can disregard the alarm from e_2 because it does not give more information about the failure than the alarm from e_1. Indeed e_1 is closer to the failure than e_2. The same principle makes us discard the alarm sent by e_2 when these two out-alarmed elements are not contiguous but have P and $A1$ between them. Consequently, when several alarms issued by out-alarmed elements having only P and $A1$ elements between them are received, only the one whose origin has the smallest position in the channel will be retained. Let us

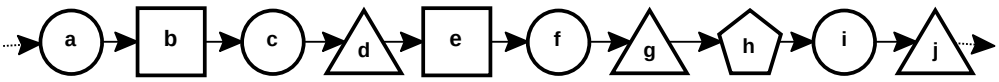


Figure 4.5: Channel example

illustrate on the channel example of Figure 4.5. If c fails, the components that will send alarms are d and g. In this case, the alarm from g will be discarded by the alarm from d. However, if for example f also fails, it will not be detected: only the failure from c will be located. It is only once c is repaired that, as explained above, a failure in f can be detected. This situation, of a double failure, is assumed much less likely than a single one, and hence one prefers to discard the alarm issued by g which most probably is redundant. This second discarding phase results in a set of non-redundant alarms issued by out-alarmed elements denoted by $\mathcal{ANR}$ *(Alarms_Non_Redundant)* and defined by:

$$\mathcal{ANR} = \{a \in \mathcal{T} \mid \text{for all } a_0 \in \mathcal{T},\ a_0.origin\ \beta\ a.origin = 0\} \tag{4.6}$$

where the function $e_1 \beta e_2$ is a Boolean function that returns 1 when there exists at least one channel that contains both components and such that all components between them are of class P or $A1$, and returns 0 otherwise. Mathematically this function can be expressed by:

$$e_1 \beta e_2 = \begin{cases} 1 & \text{if } \exists CH_i \in \mathcal{CH} \mid \begin{cases} \bullet 1 \le Pos(e_1, CH_i) \le Pos(e_2, CH_i) \text{ and} \\ \bullet \forall e \in \mathcal{V} \text{ with} \\ \quad Pos(e_1, CH_i) < Pos(e, CH_i) < Pos(e_2, CH_i), \\ \quad\quad e \in P \cup A_1 \end{cases} \\ 0 & \text{otherwise} \end{cases} \tag{4.7}$$

The general pseudo-code that performs this module and the next one is presented in Appendix C.

Candidate Search

After having obtained the set of non redundant alarms $\mathcal{ANR}$ from out-alarmed elements, one searches for the network components of $\mathcal{CH}$ whose failure may have prompted these alarms, called *fault candidates*. Their set, $\mathcal{PC}_2$ *(Possible_Candidates_2)*, may contain elements of all categories.
An element e is *fault candidate* of an alarm a, when e may have caused a, that is, when the origin of the alarm is an $A2$ component that belongs to the same channel than e and there are only P and $A1$ components between them. Mathematically it can be expressed by:

$$\mathcal{PC}_2 = \left\{ \begin{array}{r} e \in \mathcal{V} \mid \exists CH_i \in \mathcal{CH} \text{ with } Pos(e, CH_i) \neq 0 \text{ and} \\ \exists a \in \mathcal{ANR} \text{ with } e \,\beta\, a.origin = 1 \end{array} \right\} \tag{4.8}$$

where the function β is the same as Function (4.7).

At this point, we have obtained the set $\mathcal{D}$ from the *forward* phase and the set $\mathcal{PC} = \mathcal{PC}_1 \cup \mathcal{PC}_2$ from the *backward* phase. The set $\mathcal{D}$ gives, for each network element e, the set of 'alarming' elements that will issue an alarm when e would fail, whereas the set $\mathcal{PC}$ gives all the components whose likely failures account for the observed alarms (either $A1$ and $A3$ alarms, sent by elements in $\mathcal{PC}_1$, or $A2$ alarms, sent because a failure of an element in $\mathcal{PC}_2$). As it will be shown in Chapter 7 the complexity of all modules up to this point is polynomial. The final result, presented to the Human Manager, is not however the whole set $\mathcal{PC}$ but a refinement of this set, denoted by $\mathcal{SC}$ *(Subset Candidates)*, which is a set of subsets of $\mathcal{PC}$ given by the last module *Candidate Search*. We will see in Chapter 7 that the determination of this set of subsets of $\mathcal{PC}$ has a non polynomial complexity.

4.4.3 Candidate Selection

This module combines the result of both phases to perform the refinement of the $\mathcal{PC}$. It is called repeatedly, as long as there is no updating of the entries. The inputs of this module are: the mismatch threshold $m = m_1 + m_2$, the two sets $\mathcal{PC}_1, \mathcal{PC}_2$ given by the *backward* approach and the set $\mathcal{D}$ given by the *forward* approach. Let us distinguish the ideal from the non-ideal case:

(i) In the ideal scenario where there are neither false nor lost alarms (m=0), one looks for the *smallest* subset(s) of elements of $\mathcal{PC}$ that explain all the received alarms $\mathcal{R}$. To obtain this minimal set $\mathcal{SC}$ one builds iteratively sets $\mathcal{NS}(i)$, by starting from $\mathcal{NS}(0) = \mathcal{PC}_1$ and by adding subsets of $\mathcal{PC}_2$ having each a number i of elements at each iteration i. One begins therefore with singletons: all those that account for all the alarms in $\mathcal{R}$ are included in $\mathcal{SC}$. If none of them qualifies, pairs of elements of $\mathcal{PC}$ are then examined, and so on, until a subset of sufficient size that explain all the received alarms has been found.
The successive iterations are therefore:

Iteration 0: $\mathcal{NS}(0)$=$\mathcal{PC}_1$
Iteration 1: $\mathcal{NS}(1)$=$\mathcal{PC}_1\cup$ singletons of $\mathcal{PC}_2$
Iteration 2: $\mathcal{NS}(2)$=$\mathcal{PC}_1\cup$ pairs of $\mathcal{PC}_2$
Iteration 3: $\mathcal{NS}(3)$=$\mathcal{PC}_1\cup$ triples of $\mathcal{PC}_2$ ...

At each iteration i, the union of the domains of each element of one subset of $\mathcal{NS}(i)$

is computed and compared with the set of elements that have originated $\mathcal{R}$. The last iteration i is reached when at least one of the subsets of $\mathcal{NS}(i)$ contains exactly all the elements that have issued the alarms $\mathcal{R}$.
In other words, if we define $\mathcal{R}_{orig}$ as the set of network components that are the origins of the received alarms $\mathcal{R}$:

$$\mathcal{R}_{orig} = \{e \in \mathcal{V} \,|\, \exists a \in \mathcal{R} \text{ with } a.origin = e\}, \tag{4.9}$$

the last iteration i is reached when the elements of at least one subset of $\mathcal{NS}(i)$, $\{e_1, e_2, ..., e_q\}$ verify Equation (4.10)

$$\mathcal{R}_{orig} = \bigcup_{j=1}^{q} Domain(e_j) \tag{4.10}$$

Then, all the subsets $\mathcal{NS}(i)$ verifying this equation are included in $\mathcal{SC}$.
Let us consider the example of the channel shown in Figure 4.5. If there are only two received alarms, and if they are issued by d and g, the elements included in $\mathcal{PC}_2$ after the two discarding and the *Candidate Search* modules will be a, b and c. Therefore, since there is no alarm issued by a self-alarmed element, $\mathcal{PC} = \mathcal{PC}_1 \cup \mathcal{PC}_2 = \emptyset \cup \{a, b, c\} = \{a, b, c\}$.
At Iteration 0, $\mathcal{NS}(0) = \mathcal{PC}_1 = \emptyset$ and therefore, no check is done. At Iteration 1, $\mathcal{NS}(1) = \{\{a\}, \{b\}, \{c\}\}$. In this case, singletons $\{a\}$ and $\{c\}$ satisfy Equation (4.10) because their failure explain alarms issued by d and g, whereas singleton $\{b\}$ does not fulfill the equation because if it had failed, one would have had $\mathcal{PC}_1$=$\{b\}$. Therefore, Iteration 1 is the last iteration. In this case, $\mathcal{SC} = \{\{a\}, \{c\}\}$.

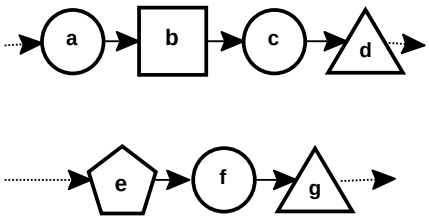


Figure 4.6: Double channel example

Let us now consider the channels of Figure 4.6. Because a double failure occurs, the only received alarms are issued by d and g. In this case, $\mathcal{PC}_1 = \emptyset$ and $\mathcal{PC}_2 = \{a, b, c, e, f\}$. At Iteration 0, $\mathcal{NS}(0)$ is again $\emptyset$. At iteration 1, $\mathcal{NS}(1) = \{\{a\}, \{b\}, \{c\}, \{e\}, \{f\}\}$ but in this case no singleton can explain both received alarms. Therefore, Iteration 2 is done considering pairs of the elements of $\mathcal{PC}_2$: $\mathcal{NS}(2) = \{\{a, b\}, \{a, c\}, \{a, e\}, \{a, f\}, \{b, c\}, ...\}$. The subsets $\{a, f\}, \{c, f\}$ verify Equation (4.10). Hence, in this scenario, $\mathcal{SC} = \{\{a, f\}, \{c, f\}\}$.

(ii) In the non-ideal scenario, the output $\mathcal{SC}$ is the set of subsets of elements that when failing will cause a set of alarms that will differ from the received alarms by a mismatching value lower than the given threshold m.
The sets $\mathcal{NS}(i)$ are constructed iteratively as in the ideal case. Also, the last iteration i is reached when at least one of the subsets of $\mathcal{NS}(i)$ contains exactly all the elements of $\mathcal{R}_{orig}$. The subsets included in $\mathcal{SC}$ are the sets where the union of the domains of

their components has a difference up to m elements with $\mathcal{R}_{orig}$. Mathematically, it can be expressed as:

$$| (UnionDom(\mathcal{NS}(i)) \setminus \mathcal{R}_{orig}) \bigcup (\mathcal{R}_{orig} \setminus UnionDom(\mathcal{NS}(i))) | \leq m \qquad (4.11)$$

where $UnionDom(\mathcal{NS}(i)) = \bigcup_{j=1}^{q} Domain(e_j)$.
Then $UnionDom(\mathcal{NS}(i)) \setminus \mathcal{R}_{orig}$ is the set of components that should have sent an alarm due to the failure of the components of the $\mathcal{NS}(i)$ subset but whose alarms did not reach the manager (i.e. *lost alarms*), whereas $\mathcal{R}_{orig} \setminus UnionDom(\mathcal{NS}(i))$ is the set of components having sent an alarm but which cannot be explained by the failure of the components of the $\mathcal{NS}(i)$ subset (i.e. *false alarms*).
In this case, the output $\mathcal{SC}$ is the set of all subsets of $\mathcal{NS}(i)$ that verify Equation (4.11).
Clearly, if m=0, Equation (4.11) becomes Equation (4.10).
In the example of Figure 4.5, if the only received alarms come from d and g, $\mathcal{PC}_1 = \emptyset$ and $\mathcal{PC}_2 = \{a, b, c\}$. Assume that m=1, that is, one mismatch between the received alarms and the expected alarms is tolerated. At Iteration 0, $\mathcal{NS}(0) = \mathcal{PC}_1 = \emptyset$ and therefore, no check is done. At Iteration 1, $\mathcal{NS}(1) = \{\{a\}, \{b\}, \{c\}\}$. In this case, the three singletons $\{a\}$, $\{c\}$ and $\{b\}$ satisfy Equation (4.11), the two first ones with 0 mismatches and the latter one with 1 mismatch because when it is faulty an alarm from b itself is also expected. The resulting output is therefore $\mathcal{SC} = \{\{a\}, \{b\}, \{c\}\}$.

4.5 Examples of the AFA application

The AFA is applied on two different network topologies. The first network has a meshed topology, the ARPA2 topology and its nodes are either local nodes (the ones with Add-Drop Filter (ADF)) or central nodes (the ones with switches). The second network has two protected WDM rings interconnected through the main node of each ring, which is the only node having a switch. The results of the AFA are shown in different scenarios: single failures, double failures and for some missing and false alarms. We will see that the location of the failure becomes more exact when the more channels are established. This is due to the fact that the information the manager receives, is larger. In some cases, the AFA returns several components as possible failures because none of them is a better candidate than the others. In this case, the possible existence of another parameter (as for example, the failure history of the component timestamps or how old is it) may help to refine the list of fault candidates.

4.5.1 Meshed optical network with ARPA2 topology

ARPA2 is a well-known meshed topology that is going to be considered as the topology of the studied optical network (see Figure 4.7). We will consider this network as a multi-hop network so that each node uses a different wavelength and that each link between two nodes consists on two separate optical fibers. Due to this bi-connectivity, the information addressed to a certain node can reach it from at least two different paths and the protection switch will choose the best input among all (for example, Node 12 can receive information from two different fibers and Node 13 from 3 different ones). There are two kinds of nodes: local nodes and central nodes. The former ones are the nodes that have an ADF and the latter ones are the nodes with a switch (contrary to local nodes, central nodes have more

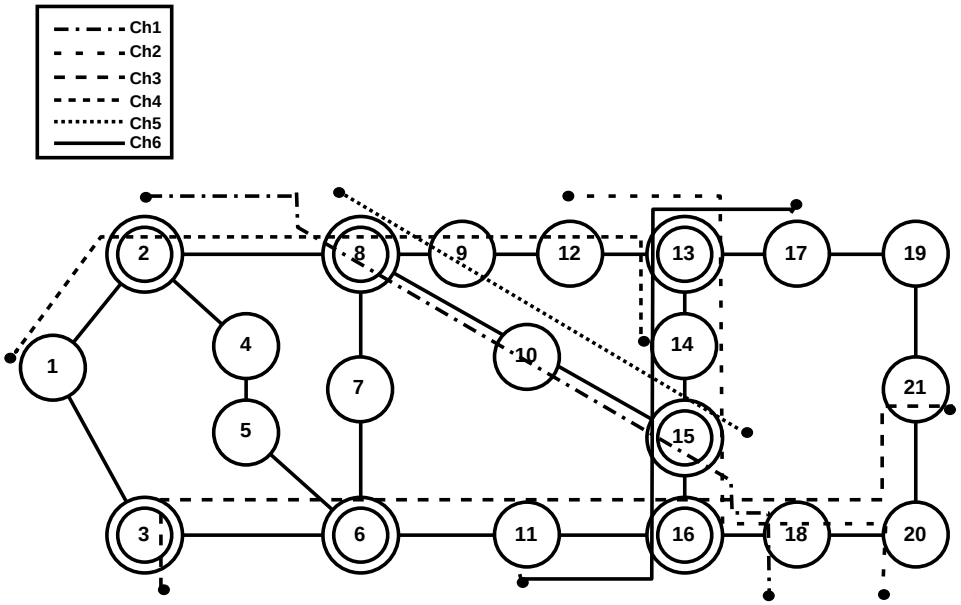


Figure 4.7: ARPA2 Network with optical links

than two connected nodes). In the example of Figure 4.7, Nodes 1, 4, 5, 7, 9, 10, 11, 12, 14, 17, 18, 19, 20 and 21 are local nodes (marked with a single circle) and Nodes 2, 3, 6, 8, 13, 15 and 16 are central nodes (represented by a double circle). Each kind of node has a different internal structure hence its modeling results to be different. During the modeling, each element is associated to a four digits identifier that is unique in the network and helps to a fast location within the network.

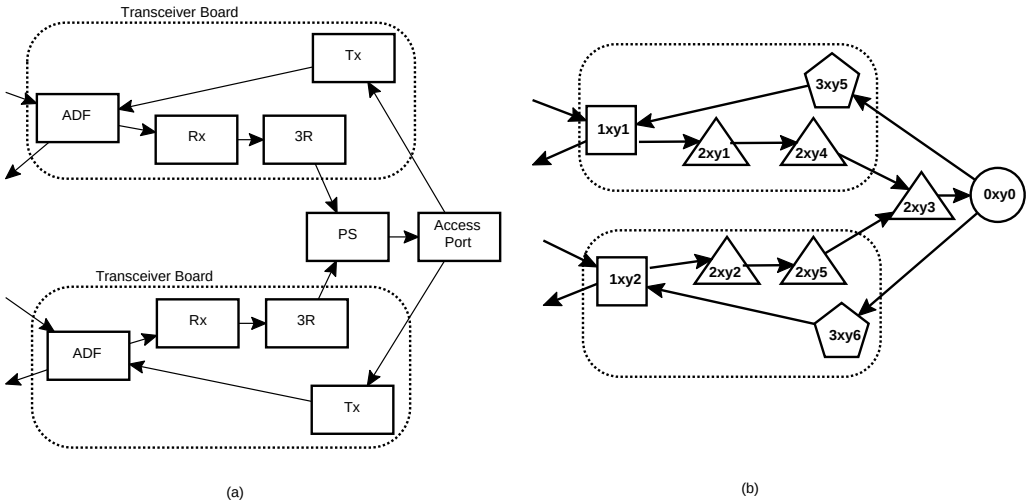


Figure 4.8: Local Node Internal structure (a) and its modeling (b)

Figure 4.8(a) presents the internal structure of a local node and Figure 4.8(b) presents its modeling. Each element has its identifiers in accordance with Table 4.2 where the first digit gives the class of element, according to Section 4.2.3, the second digit x is the node identifier, the third digit is 0 and the fourth one determines the element within the node. On the other hand, Figure 4.9(a) shows the components of a central node where k is the number of nodes connected to it. Figure 4.9(b) shows its modeling. The element identifiers

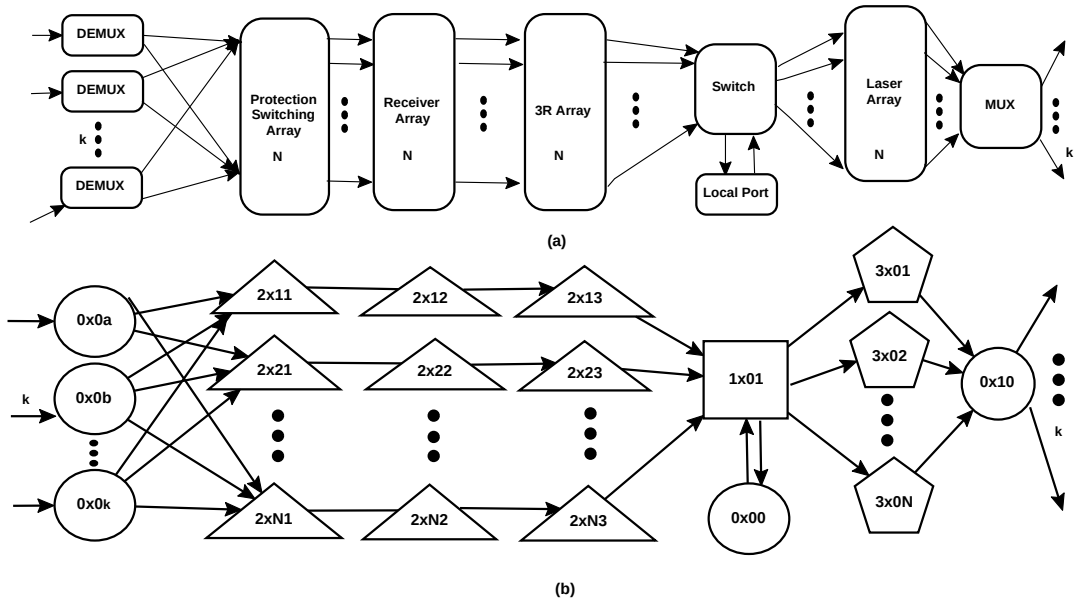


Figure 4.9: Central Node Internal structure (a) and its modeling (b)

Component	Identifier
ADF CW node x	1 x 0 1
ADF CCW node x	1 x 0 2
Rx CW node x	2 x 0 1
Rx CCW node x	2 x 0 2
PS node x	2 x 0 3
3R CW node x	2 x 0 4
3R CCW node x	2 x 0 5
Local Access Port node x	0 x 0 0
Laser CW node x	3 x 0 5
Laser CCW node x	3 x 0 6

Table 4.2: Identifiers of a local node hardware components

have been assigned in accordance with Table 4.3 where the first digit gives the class of element, the second digit x is the node identifier and the combination of digits third and fourth is the identification of the component within the node.
For this example, six channels have been established, which are listed in Table 4.4. The routing of the channels has been done arbitrarily and does not enter within the scope of this work. The input $\mathcal{CH}$ to the AFA is the ordered list of components of each channel (presented in Appendix A.1). For example, the ordered list related to Ch_1 is: **(0 2 0 0)**(1 2 0 1)(3 2 0 18)(0 2 1 0)(0 0 2 8)(0 8 0 2)(2 8 18 1)(2 8 18 2)(2 8 18 3)(1 8 0 1)(3 8 0 18) (0 8 1 0)(0 0 8 10)(1 10 0 1)(0 0 10 15)(0 15 0 10)(2 15 18 1)(2 15 18 2)(2 15 18 3)(1 15 0 1)(3 15 0 18)(0 15 1 0) (0 0 15 16)(0 16 0 15)(2 16 18 1)(2 16 18 2)(2 16 18 3)(1 16 0 1)(3 16 0 18)(0 16 1 0)(0 0 16 18)(1 18 0 1) (2 18 0 1)(2 18 0 4)(2 18 0 3)**(0 18 0 0)**
Several failure scenarios have been tested with this configuration. The results are listed in Table 4.5.

Failure of an optical fiber : In Scenario, 1 we have considered the failure of an optical fiber, as described in Chapter 3. In this scenario, the broken optical fiber connects

Component	Identifier
MUX node x	0 x 1 0
DEMUX node x from node y	0 x 0 $y+1$
PS node x at wavelength λ_z	2 x z 1
3R node x at wavelength λ_z	2 x z 3
Rx node x at wavelength λ_z	2 x z 2
Switch node x	1 x 0 1
Local Access Port node x	0 x 0 0
Laser node x at wavelength λ_z	3 x 0 z

Table 4.3: Identifiers of the central node hardware components

Channel	Input Node	Output Node	Intermediate Nodes
Ch_1	2	18	2-8-10-15-16-18
Ch_2	12	20	12-13-14-15-16-18-20
Ch_3	3	21	3-6-11-16-18-20-21
Ch_4	1	14	1-2-8-9-12-13-14
Ch_5	8	15	8-10-15
Ch_6	11	17	11-16-15-14-13-17

Table 4.4: Established channels in the ARPA2 network

Node 16 with Node 18, which is identified by (0 0 16 18). In this case, three channels are interrupted: Ch_1, Ch_2 and Ch_3. Therefore, the components that will send an alarm are the $A2$ components located behind the failure and without any $A3$ component between the failure and themselves. These components are: (2 18 0 1)(2 18 0 4)(2 18 0 3)(2 20 0 1)(2 20 0 4)(2 20 0 3)(2 21 0 1)(2 21 0 4)(2 21 0 3), that is, the three Protection Switches, 3Rs and Receivers at respectively Nodes 18, 20 and 21. The solution of the AFA having *m=0*, is either MUX (0 16 1 0) or Optical Fiber (0 0 16 18). The AFA is not able to distinguish which of these adjoint *non-alarming* elements has failed because any of them provide information to the manager. The agent that controls the hardware components, informs to the manager that the Protection Switch has changed its position so that the channel path should be updated. In this case, $\mathcal{CH}$ changes. For example, for Ch_1, the channel path will pass by Nodes 2, 8, 9, 12, 13, 17, 19, 21, 20 and 18. This is one possibility that will depend on the channel routing applied and is outside the scope of this paper. When the Protection Switch changes its position, both paths are stored so that the failure that provoked this switch can be located given the previous path. After locating the failure, the path is removed from $\mathcal{CH}$ and the alarms $\mathcal{R}$ are updated in case that now some previous non-considered alarms have now to be considered and viceversa.

Laser wavelength instability In Scenario 2, the failure presented in Section 3.2.2 has been considered. The laser at Node 6 that emits at wavelength λ_{16} has a problem. If the micro-controller detects any anomaly at the laser, as for example the temperature out of the *Ma* range, it will send an alarm. Because this alarm is sent by an $A3$ component, this alarm will be processed by the *Alarm_Discarding_1* module and the origin of the problem will be immediately located. But if the micro-controller of the laser does not detect any anomaly, no alarm is sent (Scenario 2'). In this case, the resulting alarms from this problem are processed in the *Alarm_Discarding_2* module.

Scenario	# of al.	Channels	AR	Results
1	8	Ch_1,Ch_2,Ch_3	62%	MUX(0 16 1 0)(m=0) O. F.(0 0 16 18)(m=0)
2	4	Ch_3	50%	Laser(3 6 0 21)(m=0)
2'	3	Ch_3	66%	Laser(3 6 0 21)(m=1)
3	9	Ch_2,Ch_4,Ch_6	66%	O. F.s (0 0 13 14) & (0 0 14 13)(m=0)

Table 4.5: Testing results using the ARPA2 network topology. # of al. gives the cardinality of $\mathcal{R}$ and AR stands for alarm reduction after the discarding phase.

These alarms are sent by the $A2$ components that after demultiplexing the WDM signal detect the levels of each of the demultiplexed signals. In this case the alarms are coming from (2 16 21 1)(2 16 21 2)(2 16 21 3). The result of the algorithm in this case will be also the correct one but, with the mismatching value to 1 because the alarm from the laser was expected and was lost.

Two simultaneous failures In Scenario 3, two simultaneous failures have been considered: the two optical fibers between Node 13 and Node 14 fail: (0 0 13 14) and (0 0 14 13). This is the case when an optical ribbon is broken. In this case 3 channels are interrupted and 9 alarms are issued. Ch_2 and Ch_4 use the fiber from Node 13 to Node 14 and Ch_6 uses the optical fibre from Node 14 to Node 13. The algorithm locates the failure perfectly.

4.5.2 A WDM ring network: COBNET

This Alarm Filtering Algorithm is applied in the COBNET Project [1] presented in Chapter 3. The COBNET network is shown in Figure 4.10 where two CPNs are connected through the PN).

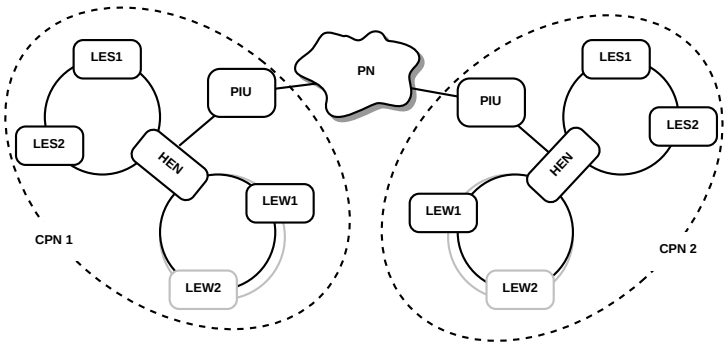


Figure 4.10: The COBNET network with two nodes at each protected ring

As introduced in Section 2.2.3, CPNs are composed of:

- an interface to the PN called Public Interface Unit (PIU),
- a central node called High End Node (HEN) that contains the main switch. The internal structure of this node is shown in Figure 4.11(a). This switch interconnects inputs from/outputs to any ring, its own local ports and the ports to/from the PN that are located at the PIU,

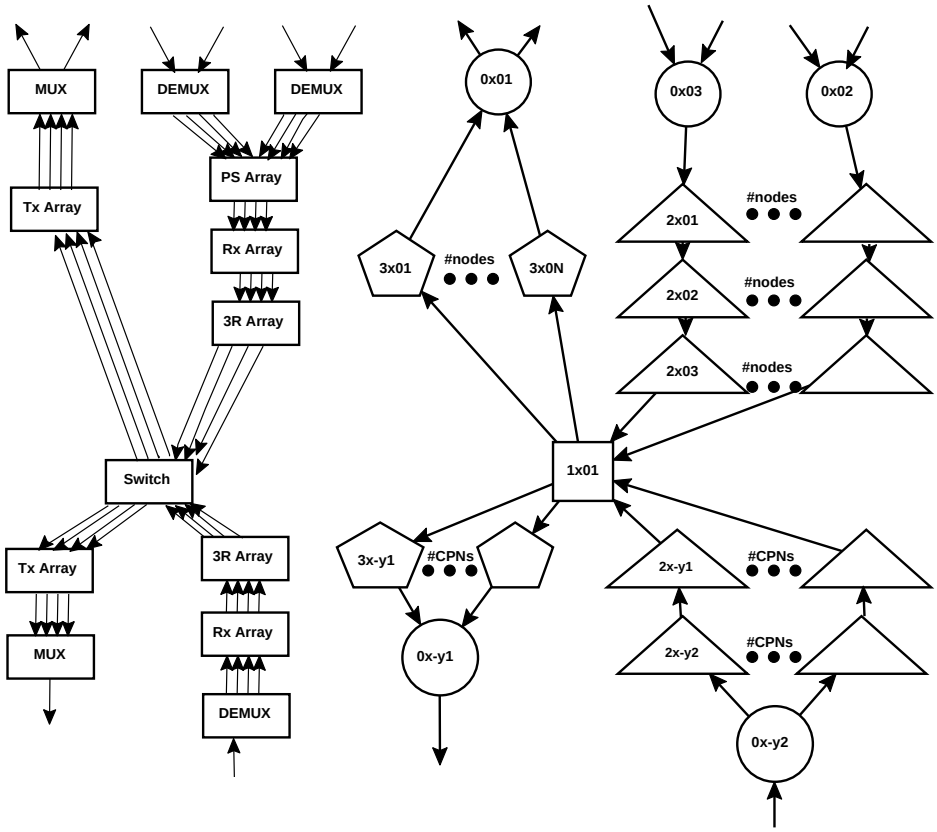


Figure 4.11: HEN internal structure (a) and its modeling (b)

- 2 rings: a WDM protected ring for the network nodes separated by more than about 2 km and a SDM protected ring for nodes that are closer to each other. Both rings are duplicated in hardware, so that the CPNs are less vulnerable to failures. A WDM ring is composed of only one optical fiber whereas an SDM ring has as many optical fibers as nodes in the ring. The Low End Nodes (LENs) of the rings are denoted by LEW for the WDM ring and by LES for the SDM ring. Each LEW node, whose structure is shown at Figure 4.8(a), has two transmitters and two receivers because of the protected ring.

Here we consider only the WDM part of the CPNs for two reasons:

(i) it is the technology that will be used in future optical networks and

(ii) these are the rings where the failures are the most difficult to locate because of the higher number of channels interrupted by a single failure and of the resulting increase of the number of alarms.

In order to apply our algorithm to the COBNET network, the CPN has been modeled as the interconnection of network components through a protected WDM ring. The identifiers used for the LENs are the same as the local nodes of the previous example (Figure 4.8(a) and (b)) where x is now the ring identifier and the third digit is the node identifier (instead of 0 as in the local nodes of the meshed network). The identifiers used in the HEN have four digits: the first digit gives the class of element, the second digit x is the ring identifier and the combination of the third and fourth digits is the identification of the component

Channel	Input Node	Output Node	Channel	Input Node	Output Node
Ch_1	(1 4)	(2 2)	$Ch_{1'}$	(2 2)	(1 4)
Ch_2	(2 1)	(1 3)	$Ch_{2'}$	(1 3)	(2 1)
Ch_3	(1 2)	(2 3)	$Ch_{3'}$	(2 3)	(1 2)
Ch_4	(1 1)	(1 5)	$Ch_{4'}$	(1 5)	(1 1)
Ch_5	(2 4)	(2 5)	$Ch_{5'}$	(2 5)	(2 4)

Table 4.6: Established channels in the COBNET network

within the node where y is the connected node identifier to this HEN.

The overall network with the ring topology that we have considered is shown in Figure 4.12.

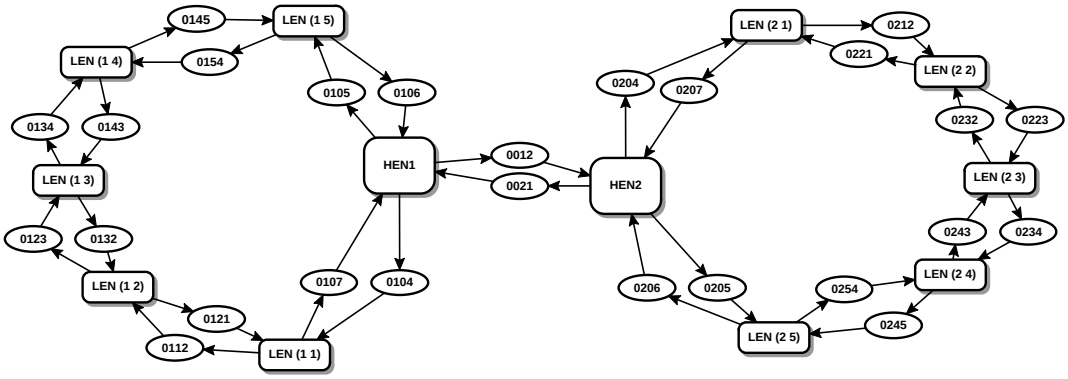


Figure 4.12: Optical network with 5-node protected rings

Due to the double ring structure, the established connections in the network are not interrupted if there is a failure. The protection mechanism works as follows. Let us consider a bidirectional communication between LENs Nodes (1 4) and (2 2). The protection switch (PS) at each node is set at the clock position by default. In this case, the two paths for the two unidirectional channels (Ch_1 and $Ch_{1'}$) are:

- Ch_1 from LEN (1 4) to LEN(2 2): *LEN(1 4)-LEN(1 5)-HEN1-HEN2-LEN(2 1)-LEN(2 2)*
- $Ch_{1'}$ from LEN(2 2) to LEN(1 4): *LEN(2 2)-LEN(2 3)-LEN(2 4)-LEN(2 5)-HEN2-HEN1-LEN(1 1)-LEN(1 2)-LEN(1 3)-LEN(1 4)*

If the LEN stops receiving signal from the CW ring, its PS switches and receives the signal from the CCW ring so that the channel is not interrupted. Following the example, if the fibre (0 1 1 2) between LEN(1 1) and LEN(1 2) breaks, the PS at LEN(1 4) will switch to the CCW ring so that the return path of Ch_1 will become:

- Ch_1 from LEN(2 2) to LEN(1 4): *LEN(2 2)-LEN(2 3)-LEN(2 4)-LEN(2 5)-HEN2-HEN1-LEN(1 5)-LEN(1 4)*

Even if the connection is restored and no data is lost, the failure has to be located and repaired. We have studied different failure scenarios considering the unidirectional channels listed at Table 4.6. Each bi-directional connection (*node a-node b*) is considered as two unidirectional channels Ch_x and $Ch_{x'}$ where the former is the direct path (from *node a* to *node b*) and the latter is the reverse path (from *node b* to *node a*). There are 10 established channels that are the $\mathcal{CH}$ input of the AFA. The channel paths are the following are presented in Appendix A.2. Several scenarios have been considered:

Failure of an optical fiber : Scenario 1 considers the hardware failure of the Optical Fiber (0 1 2 3), as presented in Section 3.3.1. In this case, four channels are interrupted $Ch_{1'}$, Ch_2, Ch_3, Ch_4 and restored due to the PS. Twelve alarms are sent to the manager: (2 1 4 1)(2 1 5 4)(2 1 4 4)(2 1 3 1)(2 1 4 3)(2 1 0 6)(2 1 3 3)(2 1 3 4)(2 1 0 4)(2 1 5 1)(2 1 0 5) and (2 1 5 3). The solution of the algorithm gives as perfect matching (*m=0*) the optical fiber that connects LEN(1 2) with LEN(1 3).

Two simultaneous failures : Scenario 2 considers two simultaneous failures in two different optical fibers: (0 1 2 3) and (0 1 0 6). In this case, the alarms related to both failures reach the manager simultaneously and intermingled. The manager receives a total of 21 alarms. The failures are well identified, with $\mathcal{SC}$ being the two sets: ((0 1 2 3) and (0 1 0 6)) or ((0 1 2 3) and (0 1 0 3)) for a perfect matching *m=0*. The algorithm can not distinguish between (0 1 0 6) -Optical Fibre- and (0 1 0 3) -Demultiplexer- because they are two consecutive *non-alarming* components with no *alarming* component between them to give more information to the manager (same case than Scenario 1 of ARPA2).

Two simultaneous failures with lost alarm : Scenario 3 studies the two previous simultaneous failures but considering the existence of a lost alarm. In this scenario the alarms related to both failures also reach the manager simultaneously and mixed but there is one that get lost. Giving *m=2*, the AFA result is also ((0 1 2 3) and (0 1 0 6)) or ((0 1 2 3) and (0 1 0 3)), which are the best candidates since they achieve the lowest mismatching value.

Two simultaneous failures with false alarm In this fourth scenario we studied the behaviour of the AFA when a false alarm is generated. Having the same double failure as in Scenario 2, the received alarms at the manager level are the same 21 alarms and the alarm from (2 1 5 4). In this scenario, the AFA result is also ((0 1 2 3) and (0 1 0 6)) or ((0 1 2 3) and (0 1 0 3)), which are the best candidates since they achieve the lowest mismatching value 1 which corresponds to the unexpected alarm.

Vague result : In this scenario 5 we considered that they were established only Ch_1, Ch_2 and Ch_3. After a simulated failure, the manager received 9 alarms coming from: (2 1 4 1)(2 1 3 4)(2 1 2 3)(2 1 4 4)(2 1 3 3)(2 1 2 1)(2 1 4 3)(2 1 3 1)(2 1 2 4). In this case the result of the AFA is weak because it gives three possible components: (0 1 0 1)(0 1 0 4)(0 1 1 2). A possible solution is to establish a ghost channel such that involves the node in the failure domain so that the manager will have extra alarms with their information and the AFA will be able to locate the failure better. In this case, considering as ghost channel the one going from LEN(1 1) to LEN(1 1) the new alarms will be (2 1 1 1)(2 1 1 4) and (2 1 1 3), and the new AFA result with perfect matching is: (0 1 1 2).

4.6 Conclusion

This chapter describes a model based algorithm called Alarm Filtering Algorithm (AFA). First, a description of the modelization of the system is given by classifying the optical network components into different categories and by considering the channels as ordered sets of categorized elements. The categories are based on the alarming properties of the

Scenario	# of alarms	Channels	Results
1	12	$Ch_{1'}$, Ch_2,Ch_3,Ch_4	O. F.(0 1 2 3)(m=0)
2	21	Ch_1,$Ch_{1'}$,Ch_2,$Ch_{2'}$, Ch_3,Ch_4,$Ch_{4'}$	O. F. (0 1 2 3) and (0 1 0 6)(m=0)
3	20	Ch_1,$Ch_{1'}$,Ch_2,$Ch_{2'}$, Ch_3,Ch_4,$Ch_{4'}$	O. F. (0 1 2 3) and (0 1 0 6)(m=1)
4	22	Ch_1,$Ch_{1'}$,Ch_2,$Ch_{2'}$, Ch_3,Ch_4,$Ch_{4'}$	O. F. (0 1 2 3) and (0 1 0 6)(m=1)

Table 4.7: Testing results using the COBNET topology with 10 unidirectional channels

network components.
From this model, the AFA algorithm is developed, to locate multiple failures, of both '*alarming*' and '*non-alarming*' elements. It requires a minimal amount of information as input, namely the established channels in the network, and the origin and type of alarms. No knowledge of the network topology nor of failure probabilities is needed.
The AFA algorithm consists in two approaches (*backward* and *forward*), which are implemented in four modules. Combining the result of these approached in a fifth module, the AFA is able to locate multiple failures coping with lost and/or false alarms. The output of the algorithm is the set of subsets containing the elements most likely to have failed and to have prompted the alarm messages received by the manager. The larger the number of established channels, the larger the number of alarms will be issued, that is, the more computation time is needed to locate the failure(s); and the more information about the failure, that is, the more accurate location of the failure is possible. Finally, the application of the AFA to two different network topologies has been described: an ARPA2 meshed network and the European ACTS project COBNET network.

Chapter 5

Fault Location Algorithm (FLA)

5.1 Introduction

The previous chapter described an algorithm able to identify sudden failures that cause the interruption of channels. We define these failures as *hard* failures. But this algorithm does not minimize the complexity of the modules that have to be computed when alarms reach the manager. A second algorithm called Fault Location Algorithm (FLA) is developed for this purpose. This second algorithm is based on the building of a binary tree and the filling of its leaves during a Pre-computation phase so that when alarms reach the manager, the faulty elements are obtained just by traversing the binary tree and finding a particular leaf.
This algorithm is extended to locate not only *hard* but also *soft* or progressive failures. *Hard* failures are not the only ones that may occur in a communication network. Indeed, due to the equipment aging, external factors such as temperature or pressure, or misalignements, progressive failures may occur in the hardware equipment causing, for example, an increase of distortion in an optical fiber or a shift of the emitted wavelength. These soft failures can sometimes be detected at the optical layer if proper testing equipment is deployed, but often require performance monitoring at a higher layer (SDH, ATM or IP). Moreover, we must continue tolerating that some alarms may be false and/or lost. For example, the existence of thresholds may conceal a failure by not sending the expected alarms because the threshold is set higher than the measured parameter, or conversely, may cause false alarms when the threshold is too low.
The proposed FLA algorithm that solves the targeted problem, has been designed and implemented. Three important features of this algorithm are: (i) the minimal diagnosis time, i.e. the time to locate failures when the management function receives alarm(s), (ii) the location of multiple, simultaneous failures and (iii) the tolerance of false and lost alarms. The algorithm is developed mainly for SDH/WDM networks and IP/WDM networks [46] but can be generalized to other networks.
This chapter is organized as follows: Section 5.2 gives an overview of the signals available at different layers that provide information about failures in an optical network. Section 5.3 describes the behaviour of both the network components and the monitoring equipment when a failure occurs, and also presents a resulting classification that will enable the abstraction of the failure location problem in Section 5.4 and the development of an efficient algorithm in Section 5.5.

5.2 Available Failure Indications in an Optical Network

Some signals contain only binary information about failures in an optical network, such as the indication of 'Loss_Of_Signal' in a receiver or 'Temperature_Out_of_Range' in a laser, and they are issued when a failure occurs. Others are analog or may take a large range of discrete values, such as the Bit Error Rate (BER), and they are usually sent periodically to the manager.
The physical layer provides binary alarms that indicate either an internal problem of the equipment or a problem with the incoming signal. The WDM layer can provide analog information, such as distribution of power of individual carriers over the full bandwidth (this parameter indicates whether a channel has dropped out or not), channel wavelength and channel spacing (it detects wavelength shifts for individual lasers in the system), Signal to Noise Ratio (SNR) (it ensures error-free transmission in each data channel) and crosstalk (it provides a quality indication of optical WDM couplers).
Signals from the WDM layer are not sufficient to locate all the progressive failures because they do not give enough information about the transmission quality. For example, if the SNR is low, it means that there has been a system error; but the contrary is not always true. The decisive parameter that determines the transmission quality of a system is the BER that is transmission technology dependent. Assuming that the transmission technologies are known, BER can either be measured by a network tester or be delivered by other layers, such as SDH through parity check, CRC used in Ethernet, or block checksum used in TCP/IP.

5.3 Optical Network Components

We distinguish two classes of network components. The first one includes the *Hardware components* at the optical layer, described in Section 3.2, whose failure needs to be identified because it will degrade or interrupt the channels. The second one includes the *Monitoring equipments*, described in Section 3.4, that are present at one or more layers, and that provide additional information about the transmission quality. Their failure does not interrupt the channel and the second part of this section focuses only on the information that they can provide about soft failures occurring in the hardware network components. We do not seek to locate failures at layers other than the physical one, and therefore the algorithm will locate failures in hardware components, but not in monitoring components.

5.3.1 Alarming Properties

Let us study the behaviour of network elements when a failure occurs:

- **Hard or Sudden failure** When a hard failure occurs, such as a fiber cut or the turn off of a transmitter, the alarms that are generated are the signals from the hardware components and the signals from the monitoring equipment. Depending on the location of the monitoring equipment, the measurements may help to refining the hard failure location.
  Let us consider, as an example, the system shown in Figure 5.1. Two channels are established in this network and modeled as shown in Section 4.2.3 (see Figure 5.2). If the optical fiber OF1 gets cut, the alarms that will be received when no monitoring equipment is used, are those coming from Rx1 and Rx3. With this information, the

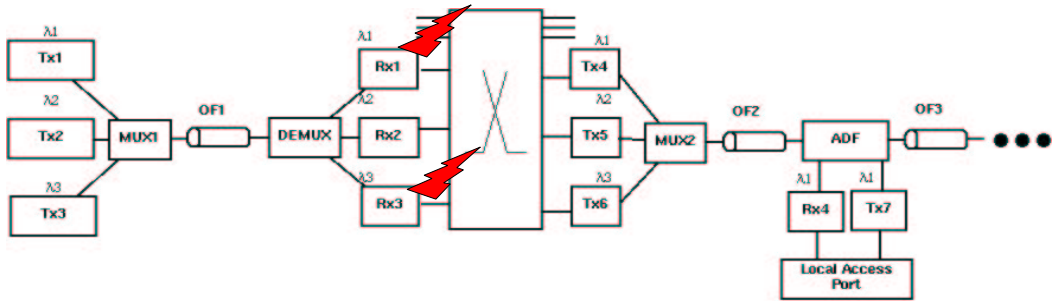


Figure 5.1: Example of a WDM System

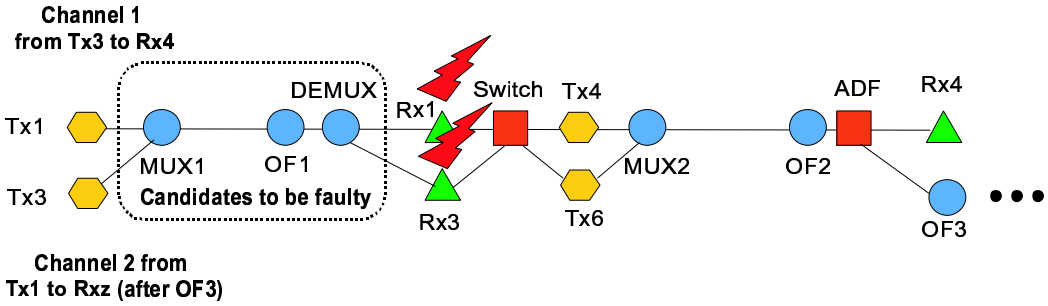


Figure 5.2: Modelization of two channels established in the network of Figure 5.1 where the fault candidates have been encircled

fault candidates are: DEMUX, OF1 and MUX1.

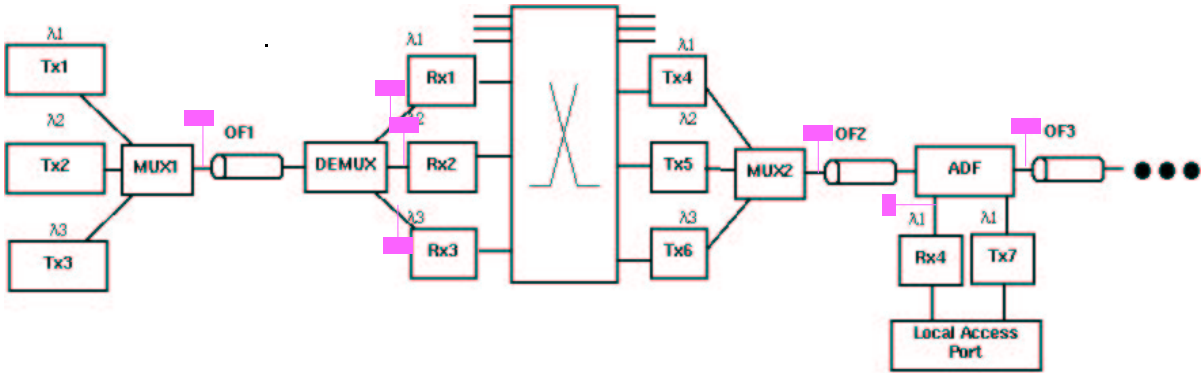


Figure 5.3: Example of a WDM System with monitoring equipment represented by a grey rectangle.

If some monitoring equipment is inserted in the system, the available information will depend on the location of the monitoring equipment. For example, if there is monitoring equipment at the output of MUX, DEMUX and ADFs (as represented by grey boxes in Figure 5.3 and by rhombus in Figure 5.4), the additional available information when hard failures occur, is more accurate (the alarms are marked by arrows and the information from the monitoring equipment by symbolic graphs in a square), enabling us to reduce the set of fault candidates to OF1 and DEMUX, because the monitoring equipment at the output of MUX1 has not detected any problem. If there was no monitoring equipment between MUX1 and the receivers, the set of fault candidates would have also included DEMUX, OF1 and MUX1 as

before. In this example, the knowledge of signals from higher layers would not have helped in locating the failure more accurately.

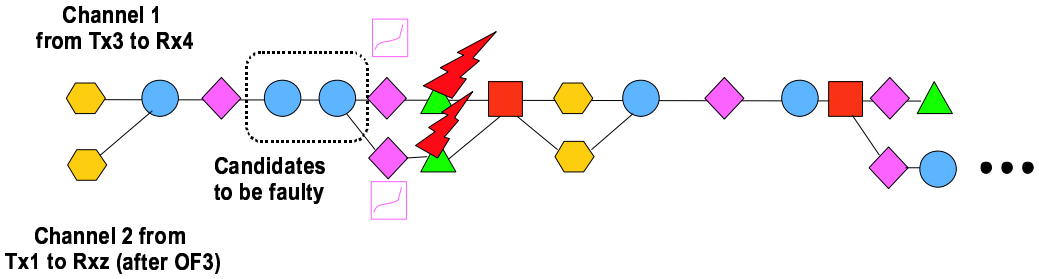


Figure 5.4: Modelization of two channels established in the network shown in Figure 5.3 where the fault candidates have been encircled

- **Soft or Progressive failure** Progressive failures, such as the shift of the emitted wavelength by a laser, do not prompt binary signals from the hardware components, but rather a variation of some analog signals. These analog signals can be associated to one or to all the transmitted channels. Let us consider as an example the system shown in Figure 5.5 with the same pair of established channels as in the previous example. If the emitted wavelength of Laser3 is shifted, the quality of the signal will

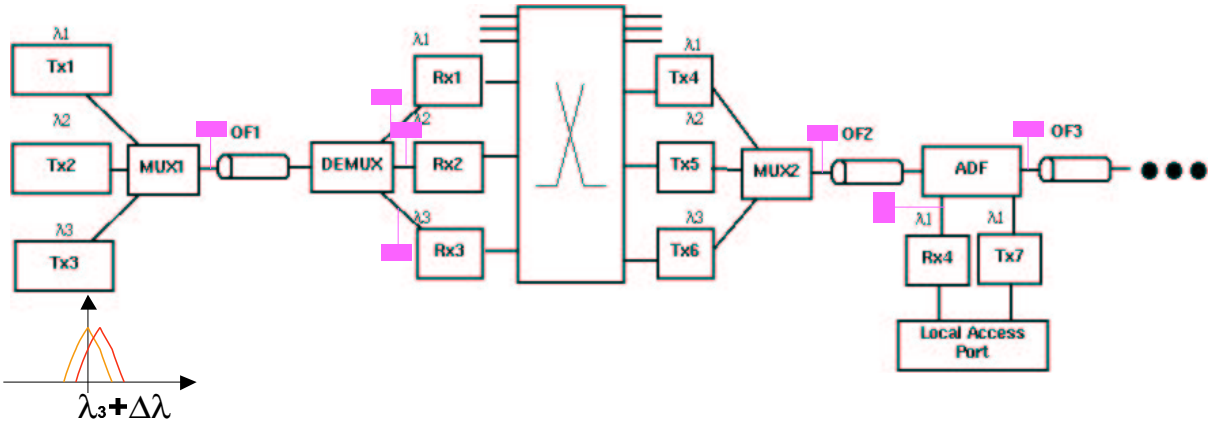


Figure 5.5: Example of a WDM network with monitoring equipment and a transmitter with a shifted wavelength.

degrade. In this case, the parameter measured by the monitoring equipment after MUX1 and before Rx3 will deteriorate. The Fault Localization Algorithm should be able to detect that there is a problem, and to locate the failure.

Alarming properties of the hardware components

The alarming properties of the network components, based on their behaviour when a failure occurs, will now depend on the kind of failure. The three *features* introduced in Chapter 4 and recalled here, still apply to this scenario. But we must now specify that they stand for *hard* failures.

Self-alarmed This property specifies whether a network component is able to send an alarm informing about its *own hard failure* or not.

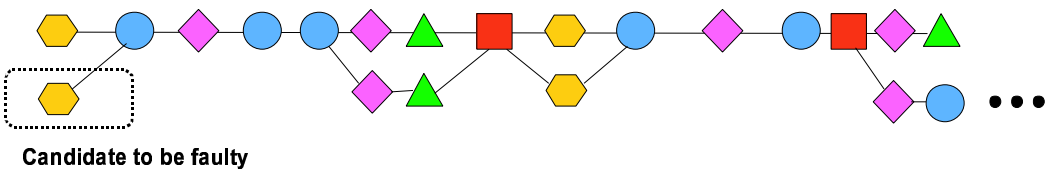


Figure 5.6: Modelization of the two established channels with the fault candidate.

Out-alarmed This property applies to the components that communicate with the manager and that send alarms about a *hard failure external* to them.

Hard failure masking (HF masking) This property specifies whether the network component masks the hard failure from the hardware components that follow it on the channel.

Alarming properties of the monitoring equipment

All monitoring equipment is out-alarmed and sensitive to the quality of the signal. Therefore, they all possess the property of *Soft failure sensitivity*. They differ however in their ability to mask failures from other monitoring equipment:

Failure masking This property specifies whether this monitoring equipment masks the failure from any other monitoring equipment of its own layer. For example, a WDM spectrum analyzer does not mask any failure from the next network component because it does not act on the content of the signal, whereas an SDH MSTE masks progressive failures from the next MSTE and RSTE because it updates the header of the retransmitted frame so that the next MSTE or RSTE will not be able to detect any failure, but it does not mask them from the next PTE because the associated B3-bytes remains intact.

5.3.2 Optical Network Components Classification

Hardware components classification

The classification of the hardware components of Chapter 4 remains unchanged and it is recalled in Table 5.1.

Monitoring equipment classification

The monitoring equipment is also classified according to their failure masking properties. Let M denote the class of all monitoring components. Let $M0$ denote the monitoring equipment that are sensitive to soft failures and do not mask any failures to other monitoring components following it on the same channel. An Mq component, where $q \in \mathbb{N}_0$, is a monitoring component masking soft failures to other monitoring components following it on the same channel and belonging to Category Mp with $0 \leq p \leq q$. For example, $M1$ denotes the monitoring equipment that are sensitive to soft failures and do mask failures to $M0$ and $M1$ monitoring components following it on the same channel. $M0$ components are represented by a rhombus, $M1$ by a thick square, $M2$ by a thick square with an oblique line and $M3$ by a thick square with a cross.

Network Component	Self-alarmed	Out-alarmed	HF masking	Category
Optical Fiber	No	No	No	P
Transmitters	Yes	No	Yes	$A3$
Receivers	No	Yes	No	$A2$
Add/Drop Filters	Yes	No	No	$A1$
3R	No	Yes	No	$A2$
Protection Switch	No	Yes	No	$A2$
MUX/DEMUX	No	No	No	P
Switch	Yes	No	No	$A1$
Optical Amplifier	Yes	Yes	No	$A1$ and $A2$

Table 5.1: Alarm properties of the Network Components and the resulting classification

	ITE	GTE	Router	Category
ITE	No	No	No	$M0$
GTE	No	No	No	$M0$
Router	No	No	Yes	$M1$

Table 5.2: Masking relationships between monitoring components and the resulting classification. The Yes/No entry indicates whether the element listed at the left of the considered row masks failures to the element at the top of the considered column.

Some examples of the application of this classification of the monitoring equipment are as follows:

- In an IP over WDM network, IP performs a check of the header at each router and WDM retrieves information at each location of the testing equipment. WDM monitoring equipment is able to detect failures from elements within the same optical path, that is, when there is no electrical conversion of the signal between the failure and the monitoring equipment. The IP routers perform the error monitoring once the signal is electrical.
  In this case, the available equipment, the masking properties and the resulting classification are listed in Table 5.2.

	ITE	GTE	RSTE	MSTE	PTE	Category
ITE	No	No	No	No	No	$M0$
GTE	No	No	No	No	No	$M0$
RSTE	No	No	Yes	No	No	$M1$
MSTE	No	No	Yes	Yes	No	$M2$
PTE	No	No	Yes	Yes	Yes	$M3$

Table 5.3: Masking relationships between monitoring components and the resulting classification. The Yes/No entry indicates whether the element listed at the left of the considered row masks failures to the element at the top of the considered column

- In an SDH network, there may be testing equipment at the WDM layer in addition to the three different SDH monitoring equipments presented in Section 3.4.2, i.e. RSTE, MSTE and PTE. Table 5.3 lists which monitoring equipment masks failures to other pieces and the resulting classification. An example is shown in Figure 5.7.

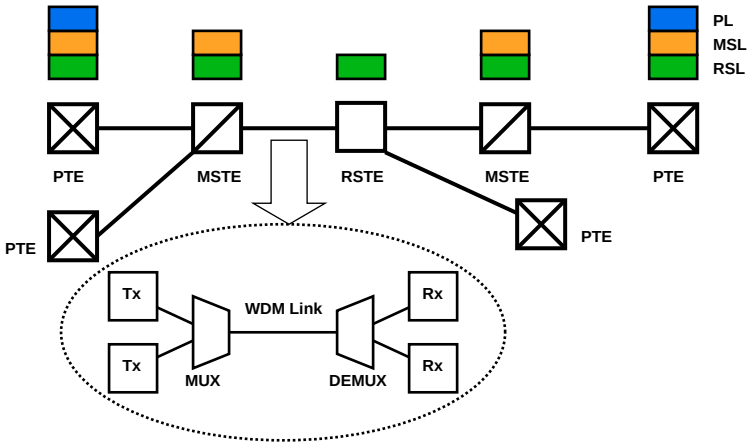


Figure 5.7: Example of two different SDH paths sharing some SDH section

5.4 Problem Abstraction

The classification of the previous section enables us to derive and implement the Fault Localization Algorithm (FLA). The FLA has to be able to locate the component or set of components that when hard or soft failures have caused the alarms received by the manager.

5.4.1 Inputs of the algorithm

The inputs of the algorithm are the same as the inputs of AFA, which are:

- the set of established channels $\mathcal{CH} = \{CH_i\}$
- the set of alarms received by the manager $\mathcal{R} = \{a_j\}$ and,
- the mismatching thresholds m_1 and m_2.

In this algorithm, the allowed number of lost alarms m_1 is distinguished from the allowed number of false alarms m_2. Obviously, $m = m_1 + m_2$.

5.4.2 New Domain Definitions

Domain(comp) has been defined as the set of network elements that will send an alarm when *comp* fails. Two different kinds of *Domains* can be distinguished for every network component based on the nature of the failure: *HDomain* when *comp* suffers a hard failure and *SDomain* when *comp* suffers a soft failure. The computation of the domains of each network component is based on the established channels and it uses the Boolean functions $FP1$ and $FP2$ presented in Chapter 4 and a new Boolean function denoted by $FP3$. The mathematical expressions of these functions are:

- If an element e_1 suffers a hard failure, an out-alarmed component $e_2 \in A2$ of any established channel will send an alarm, if both of them belong to the same channel and if there is no $A3$ element between them. Mathematically it can be expressed by the Boolean relation:

$$\begin{array}{l} e_1\ FP1\ e_2 = 1 \text{ if and only if} \\ \quad \bullet e_2 \in A2 \\ \quad \bullet \exists\, CH_i \in \mathcal{CH} \text{ with } 0 < Pos(e_1, CH_i) < Pos(e_2, CH_i) \\ \quad \bullet \forall e_j \text{ with } Pos(e_1, CH_i) < Pos(e_j, CH_i) < Pos(e_2, CH_i),\ e_j \notin A3. \end{array} \tag{5.1}$$

- An element e_1 will send an alarm when it fails if it is self-alarmed, which can be recast mathematically as:

$$e_1\ FP2\ e_2 = 1 \text{ if and only if } e_1 = e_2 \in A1 \cup A3. \tag{5.2}$$

- If an element e_1 suffers a hard or a soft failure, a monitoring element $e_2 \in Mq$ of any established channel will notice the problem, if both elements belong to the same channel, if the monitoring element follows the failing one and there is no other monitoring element that masks the failure to e_2. Mathematically it can be expressed as follows:

$$\begin{array}{l} e_1\ FP3\ e_2 = 1 \text{ if and only if} \\ \quad \bullet e_2 \in Mq \text{ for some } q \geq 0 \\ \quad \bullet \exists\, CH_i \in \mathcal{CH} \text{ with } 0 < Pos(e_1, CH_i) < Pos(e_2, CH_i) \\ \quad \bullet \text{ if } q \geq 1 \text{ then } \forall e_j \in Mp \text{ with} \\ \qquad\quad Pos(e_1, CH_i) < Pos(e_j, CH_i) < Pos(e_2, CH_i),\ p < q \\ \quad \text{else } \forall e_j \in \mathcal{V} \text{ with} \\ \qquad\quad Pos(e_1, CH_i) < Pos(e_j, CH_i) < Pos(e_2, CH_i),\ e_j \notin A3 \end{array} \tag{5.3}$$

Based on these three functions, we can define *HDomain* and *SDomain* as follows:

- $HDomain(e_1)$ is the set of elements whose alarms are expected when e_1 suffers a *hard failure*. These elements are (i) e_1 itself if $e_1 \in A1 \cup A3$ (ii) the $A2$ components that follow e_1 in at least one channel and do not have any $A3$ components between them, and finally (iii) the monitoring components that take into account their failure masking properties of Tables 5.2 and 5.3. Mathematically, $HDomain(e_1)$ can be expressed as follows:

$$HDomain(e_1) = \left\{ \begin{array}{l} e_2 \in \mathcal{V} \,|\, (e_1\ FP1\ e_2 = 1) \text{ or } (e_1\ FP2\ e_2 = 1) \\ \quad \text{or } (e_1\ FP3\ e_2 = 1) \end{array} \right\}. \tag{5.4}$$

- $SDomain(e_1)$ is the set of elements whose alarms are expected when e_1 suffers a *soft failure*. These elements are the monitoring equipment that follow e_1 and that are not masked by any other monitoring equipment. Mathematically, $SDomain(e_1)$ can be expressed as follows:

$$SDomain(e_1) = \{e_2 \in M |\ e_1\ FP3\ e_2 = 1\}. \tag{5.5}$$

5.5 Fault Localization Algorithm (FLA)

Time to locate the failure(s) is critical, and the FLA must locate failures as fast as possible. Unfortunately, the multiple fault location problem is NP-hard, even in the ideal scenario, as will be shown in Section 7.2. Nevertheless, the computation that has to be carried out when a new alarm reaches the manager can be kept small, despite the potentially large size of the network, if we follow Rao's approach [47] to pre-compute as much as possible the functions that can be executed independently of the received alarms. This phase is called *Pre-Computation Phase (PCP)*. Once the manager starts receiving alarms from the network, the algorithm does not have to perform complex computations but simply to traverse a binary tree. The PCP phase of the algorithm is executed only when $\mathcal{CH}$ is updated, not when the alarms are received. This minimizes the time the algorithm needs to deliver results to the manager when failures occur.
The PCP phase has been implemented on the basis of the algorithm devised by Rao [48], to locate single failures in a network with two kinds of network components (P and $A2$) in the ideal scenario. We have extended this algorithm to the three categories of network components presented in Section 5.3.2, to multiple failures, and finally to the non-ideal scenario, which accepts the existence of lost and false alarms.
Let us present each of the extensions of the algorithm step by step. The final scheme of FLA is shown in Figure 5.8.

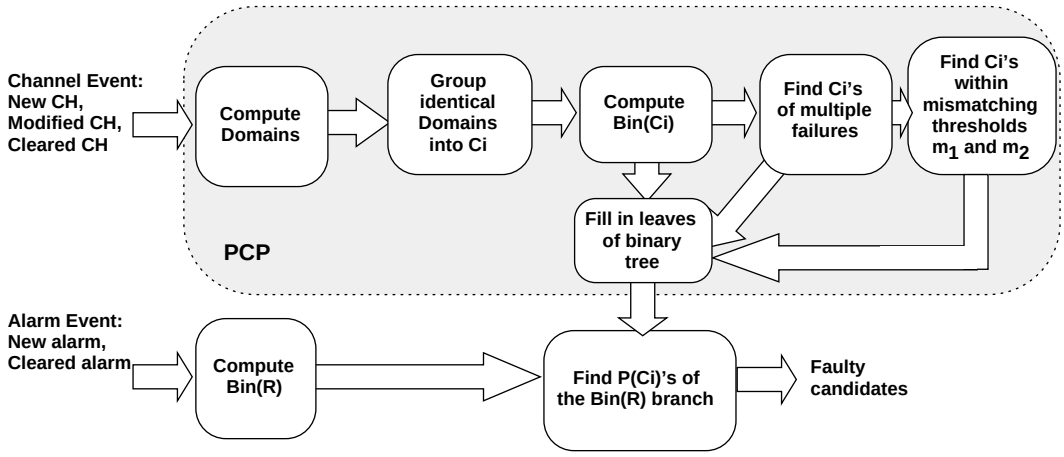


Figure 5.8: FLA Scheme: the Pre-Computation Phase (PCP) gathers most of the complexity and leaves only a few processing steps to be carried out in the FLA core

5.5.1 FLA to locate single failures in the ideal scenario

The network has only P and $A2$ components

Let us start solving the problem of locating a single hard failure within a network with two kinds of network components: P and $A2$ components. In this case, when a component fails, the 'alarming' components that follow it on a channel will send an alarm. In this section we assume that the only failures are single hard failures.
The steps of this algorithm are illustrated by the example of Figure 5.9, where two channels have been established ($\mathcal{CH} = \{CH_1, CH_2\}$). We denote by p the non-alarming components and by e the alarming components that can send an alarm when there is a hard failure in

one of the preceding elements. In the example, there are 9 non-alarming elements $p_1, ..., p_9$ and 4 alarming elements which are e_1, e_2, e_3 and e_4. The PCP consists of the following modules:

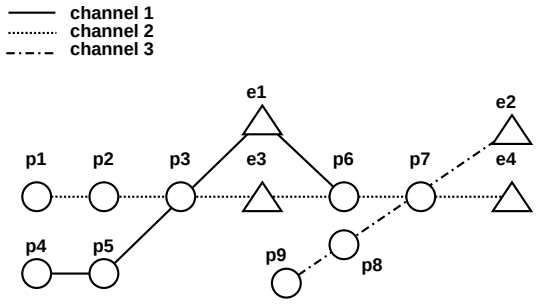


Figure 5.9: A simple toy network example to introduce the algorithm

1. Compute the domain of each element of the established channels.
   $HDomain\{p_1\} = \{e_3, e_4\}$ $HDomain\{p_2\} = \{e_3, e_4\}$
   $HDomain\{p_3\} = \{e_1, e_3, e_4\}$ $HDomain\{p_4\} = \{e_1\}$
   $HDomain\{p_5\} = \{e_1\}$ $HDomain\{p_6\} = \{e_4\}$
   $HDomain\{p_7\} = \{e_2, e_4\}$ $HDomain\{p_8\} = \{e_2\}$
   $HDomain\{p_9\} = \{e_2\}$ $HDomain\{e_1\} = \emptyset$
   $HDomain\{e_2\} = \emptyset$ $HDomain\{e_3\} = \{e_4\}$
   $HDomain\{e_4\} = \emptyset$

2. Group the identical domains into equivalence classes $C_1, C_2, ..., C_b$ $(b \leq n)$.
   $C_1 = HDomain\{p_1\} = HDomain\{p_2\} = \{e_3, e_4\}$
   $C_2 = HDomain\{p_3\} = \{e_1, e_3, e_4\}$
   $C_3 = HDomain\{p_4\} = HDomain\{p_5\} = \{e_1\}$
   $C_4 = HDomain\{p_6\} = HDomain\{e_3\} = \{e_4\}$
   $C_5 = HDomain\{p_7\} = \{e_2, e_4\}$
   $C_6 = HDomain\{p_8\} = HDomain\{p_9\} = \{e_2\}$
   The domains of e_1, e_2 and e_4 are empty so that the failure of these elements cannot be detected.

3. Associate to each C_i a binary vector $\vec{g_i} = Bin(C_i)$ with as many elements as alarming and monitoring components (this example, $A2$ components) in the established channels. Let us recall that the number of alarming and monitoring components is given by n_a, and the vectors $\vec{g_i}$ are therefore binary n_a-uples. The jth component of $Bin(C_i)$ is equal to 1 if the jth $A2$ element belongs to C_i, and to 0 otherwise. Each component of the binary vector is associated to one 'alarming' component according to the order of the channel establishment. In our example the binary vectors attached to each class $\vec{g_i} = Bin(C_i)$ are:
   $\vec{g_1} = Bin(C_1) = (0\,0\,1\,1)$ $\vec{g_2} = Bin(C_2) = (1\,0\,1\,1)$
   $\vec{g_3} = Bin(C_3) = (1\,0\,0\,0)$ $\vec{g_4} = Bin(C_4) = (0\,0\,0\,1)$
   $\vec{g_5} = Bin(C_5) = (0\,1\,0\,1)$ $\vec{g_6} = Bin(C_6) = (0\,1\,0\,0)$
   We will see in Section 5.6 that these vectors generate a set $\mathcal{C}$ of binary vectors which can be regarded as a non linear code.

4. Let $P(C_i)$ be the set of elements whose Domain is C_i:

$$P(C_i) = \{comp \in \mathcal{V} | HDomain(comp) = C_i\}. \quad (5.6)$$

In our example these sets are:
$P(C_1) = \{p_1, p_2\}$ $P(C_2) = \{p_3\}$
$P(C_3) = \{p_4, p_5\}$ $P(C_4) = \{p_6, e_3\}$
$P(C_5) = \{p_7\}$ $P(C_6) = \{p_8, p_9\}$

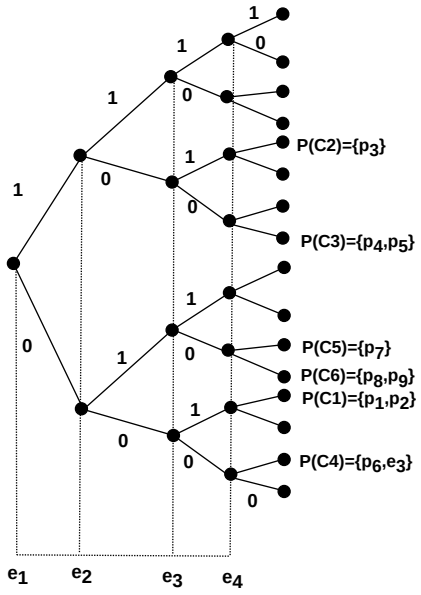


Figure 5.10: Binary Tree of the example shown in Figure 5.9

5. A binary tree is built with a depth equal to the number of active components and whose leaves correspond to different the binary combinations. Occupied leaves point to the set $P(C_i)$ whose corresponding $Bin(C_i)$ is the path from the root of the tree to the leaves. Figure 5.10 shows the binary tree of our example.

These steps can be pre-computed off-line before receiving any alarm so that the major computational complexity is placed in this *PCP* module. Once the manager receives alarms $\mathcal{R}$, a last and simple step has to be performed: $\vec{y} = Bin(\mathcal{R})$ is computed, and the binary tree is traversed from the root to the corresponding leaf. For example, if the manager receives alarms from e_2 and e_4, $Bin(\mathcal{R}) = (0\ 1\ 0\ 1)$ and the leaf that will be reached points to $P(C_5) = \{p_7\}$: p_7 is therefore the fault candidate.

The network has all categories of components $(\mathbf{P}, \mathbf{A1}, \mathbf{A2}, \mathbf{A3}, \mathbf{M})$

The modules of the algorithm are the same as those in the previous section. For each component, two different domains *HDomain* and *SDomain* have to be computed based on Equations (5.4) and (5.5) and merged. $P(C_i)$ will therefore contain two fields: the first one is the set of elements whose Domain is C_i, the second one is the indication of the nature (hard or soft) of the failure:

$$P(C_i) = \{(comp, F) \text{ with } comp \in \mathcal{V} \mid FDomain(comp) = C_i \text{ and } F = H \text{ or } S\}. \quad (5.7)$$

Let us illustrate the algorithm on the two following examples:

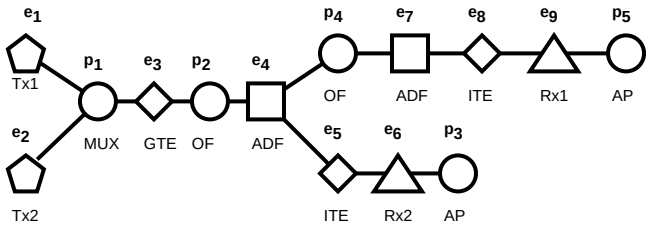


Figure 5.11: Modelization of two channels of a WDM network where the Individual Testing equipments (ITE) are BER testers

Classes of identical domains	C_i	$\vec{g_i} = Bin(C_i)$	$P(C_i)$
$HDomain\{e_1\}$	C_1	(1 0 1 0 0 0 0 1 1)	$P(C_1) = \{(e_1, H)\}$
$SDomain\{e_1\}$	C_2	(0 0 1 0 0 0 0 1 0)	$P(C_2) = \{(e_1, S)\}$
$HDomain\{e_2\}$	C_3	(0 1 1 0 1 1 0 0 0)	$P(C_3) = \{(e_2, H)\}$
$SDomain\{e_2\}$	C_4	(0 0 1 0 1 0 0 0 0)	$P(C_4) = \{(e_2, S)\}$
$HDomain\{e_3\} = HDomain\{p_1\}$	C_5	(0 0 1 0 1 1 0 1 1)	$P(C_5) = \{(e_3, H), (p_1, H)\}$
$SDomain\{e_3\} = SDomain\{p_1\}$	C_6	(0 0 1 0 1 0 0 1 0)	$P(C_6) = \{(e_3, S), (p_1, S)\}$
$HDomain\{e_4\}$	C_7	(0 0 0 1 1 1 0 1 1)	$P(C_7) = \{(e_4, H)\}$
$SDomain\{e_4\} = SDomain\{p_2\}$	C_8	(0 0 0 0 1 0 0 1 0)	$P(C_8) = \{(e_4, S), (p_2, S)\}$
$HDomain\{e_5\}$	C_9	(0 0 0 0 1 1 0 0 0)	$P(C_9) = \{(e_5, H)\}$
$SDomain\{e_5\}$	C_{10}	(0 0 0 0 1 0 0 0 0)	$P(C_{10}) = \{(e_5, S)\}$
$HDomain\{p_2\}$	C_{11}	(0 0 0 0 1 1 0 1 1)	$P(C_{11}) = \{(p_2, H)\}$
$HDomain\{e_7\}$	C_{12}	(0 0 0 0 0 0 1 1 1)	$P(C_{12}) = \{(e_7, H)\}$
$SDomain\{e_7\} = SDomain\{p_4\}$	C_{13}	(0 0 0 0 0 0 0 1 0)	$P(C_{13}) = \{(e_7, S), (p_4, S)\}$
$HDomain\{e_8\} = HDomain\{p_4\}$	C_{14}	(0 0 0 0 0 0 0 1 1)	$P(C_{14}) = \{(e_8, H), (p_4, H)\}$

Table 5.4: Domains of the network components and their binary vectors

1. Let us consider two channels that have been established in an optical network (similar to the one of Figure 5.1 but without the electro-optical switch), and that are modeled in Figure 5.11. In this case there are 9 'alarming' components and 5 'non-alarming' components. The monitoring equipment that have been installed are BER testers. The C_i classes and their corresponding vector $Bin(C_i)$ are listed in Table 5.4. Once the $Bin(C_i)$ are computed, the associated branches are occupied and the algorithm performs as described in the previous section.

2. Let us present now the example of an SDH Network in Figure 5.12 where two paths have been established. Two layers are presented in this figure: the SDH layer on the top, which contains the SDH equipment presented in Section 3.4.2 and the WDM layer on the bottom, which comprises the transmission hardware equipment and some WDM monitoring equipment. Between every pair of SDH equipment there is a WDM connection. These two paths can be modeled as shown in Figure 5.13, and contain a total of 19 'alarming' components and 6 'non-alarming' components. In this case the length of the binary vectors is 19. For example, the binary vectors associated to the domains of (P,2) are:
SDomain((P,2))=(0 0 0 0 0 0 1 0 0 1 0 0 1 0 0 0 0 0 1)
HDomain((P,2))=(0 0 0 0 0 1 1 0 0 1 0 0 1 0 0 0 0 0 1)

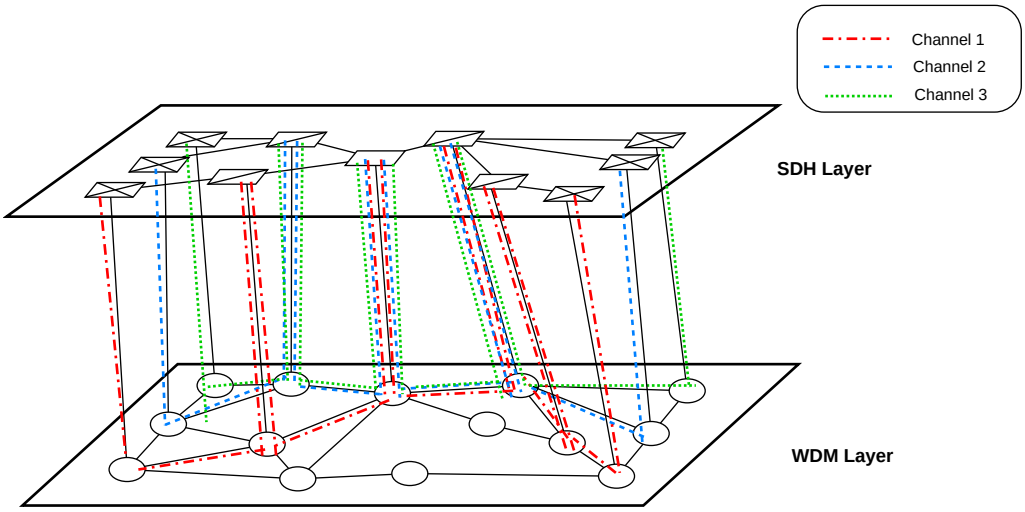


Figure 5.12: SDH over WDM network

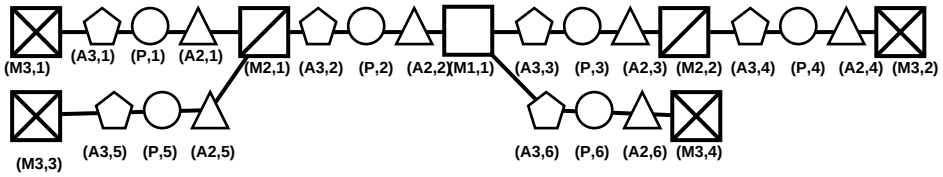


Figure 5.13: Modelization of the two SDH/WDM channels shown in Figure 5.7

5.5.2 FLA to locate multiple failures in the ideal scenario

Let us now consider the case when several failures happen in a short interval of time so that the alarms reaching the manager are intermingled with each other. The algorithm has to be able to distinguish the failures based on the received alarms. To solve this problem, the domains of simultaneous failures have to be computed.
We begin with double failures. Let $C_i = F\ Domain(e_i)$ and $C_j = F\ Domain(e_j)$ with $F = H$ or S, be the domains to two single failures of elements e_i and e_j respectively, and let $\vec{g_i} = Bin(C_i)$ and $\vec{g_j} = Bin(C_j)$ be their corresponding binary vectors. Then, the domains of a double failure of e_i and e_j will be $C_k = C_i \cup C_j$, and the associated binary vector will be

$$\vec{x_k} = Bin(C_k) = Bin(C_i \cup C_j) = Bin(C_i) \vee Bin(C_j) = \vec{g_i} \vee \vec{g_j}. \tag{5.8}$$

where $\vee$ stands for the point-wise OR operation. The set $P(C_k)$ will therefore contain different sets of pairs whose fields are $(comp, F)$: $comp$ is an element whose Domain is C_i, F is the indication of the nature of the failure (if it is a hard failure then $F = H$ and if it is a soft failure then $F = S$). This translates in the following equation:

$$P(C_i \cup C_j) = \{\{(e_i, F_i), (e_j, F_j)\}\}. \tag{5.9}$$

If equation (5.8) returns a binary vector $\vec{x_k}$ equal to one of the 'generator' vectors $\vec{g_l}$, $1 \leq l \leq t$, obtained for single failures, is not considered and not action is perform. As in the previous chapter, we can reasonably assume that a single failure is more likely than a multiple one, so that the occupied leaf points to the more likely single failure. Conversely, if equation (5.8) returns a binary vector $\vec{x_k}$ different from any of the existing generator

vectors $\vec{g_1}, ..., \vec{g_t}$, a new leaf is then occupied and points to the double failure. Once all the new leaves corresponding to double failures are filled, we proceed likewise for triple failures, etc.
Let $\mathcal{C}$ be the set of all vectors obtained by these successive OR operations, to which we can adjoin the null vector $\vec{0} = \{0, .., 0\}$, which corresponds to the absence of failure. We note that the set $\mathcal{C}$ has the property that for any $\vec{x_i}$, $\vec{x_j} \in \mathcal{C}$, the vectors $\vec{x_k}$,

$$\vec{x_k} = \vec{x_i} \vee \vec{x_j}. \tag{5.10}$$

where $\vec{x_i}, \vec{x_j} \in \mathcal{C}$.
Note also, that if at some point of this procedure, there is a $\vec{g_k}$ corresponding to a single failure which is such that for all the already computed $\vec{x_i}$'s,

$$\vec{x_i} \vee \vec{g_k} = \vec{x_i} \quad \text{or} \quad \vec{x_i} \vee \vec{g_k} = \vec{g_k}. \tag{5.11}$$

then $\vec{g_k}$ will not contribute to any new leaf anymore. Therefore, it needs not be considered for further steps. This property allows us to decrease the number of binary vectors corresponding to single failures needed for computing the domains of multiple ones. The process is finished when the set of single failures becomes empty.

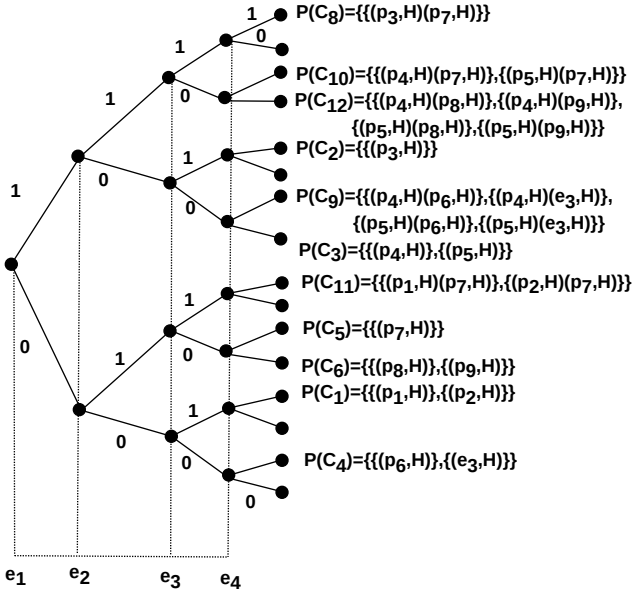


Figure 5.14: Binary Tree of network 5.9 with multiple failures

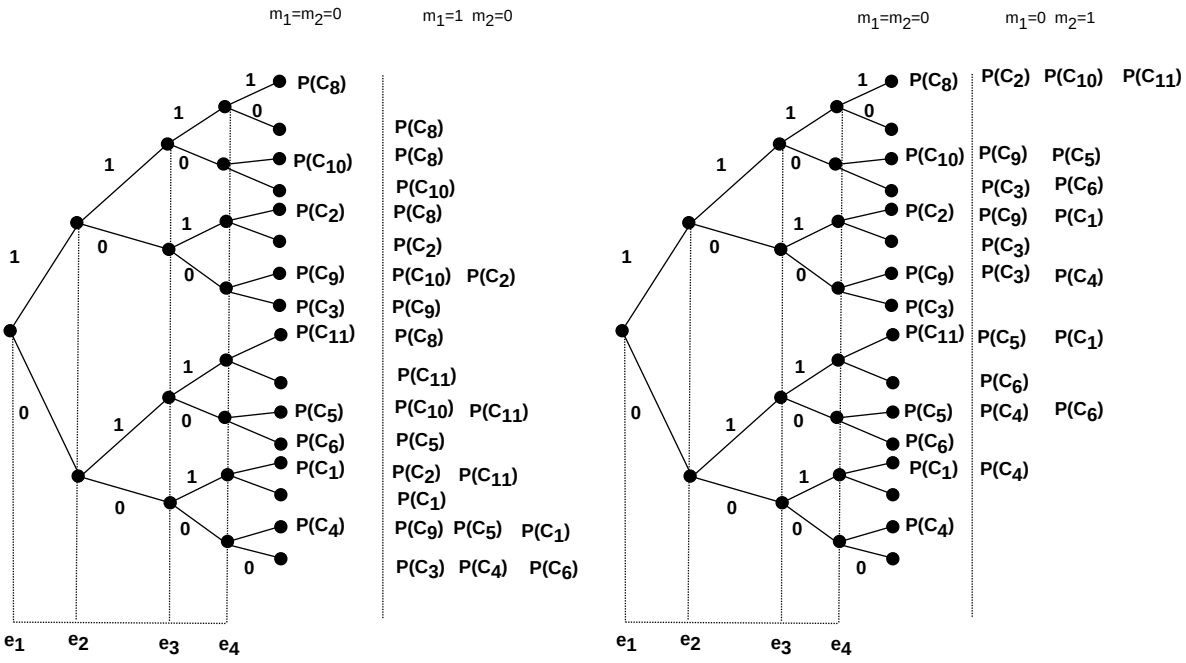
Appendix B gives the pseudo-code implementing this routine.
Let us consider the example shown in Figure 5.9. The output of this part of the algorithm is the following:
$Bin(C_1) \vee Bin(C_3) = Bin(C_2) = \vec{g_2}$
$Bin(C_1) \vee Bin(C_5) = Bin(C_{11}) = (0\,1\,1\,1) = \vec{x_1}$
$Bin(C_1) \vee Bin(C_6) = Bin(C_{11}) = \vec{x_1}$
$Bin(C_2) \vee Bin(C_5) = Bin(C_8) = (1\,1\,1\,1) = \vec{x_2}$
$Bin(C_2) \vee Bin(C_6) = Bin(C_8) = \vec{x_2}$
$Bin(C_3) \vee Bin(C_4) = Bin(C_9) = (1\,0\,0\,1) = \vec{x_3}$
$Bin(C_3) \vee Bin(C_5) = Bin(C_{10}) = (1\,1\,0\,1) = \vec{x_4}$
$Bin(C_3) \vee Bin(C_6) = Bin(C_{12}) = (1\,1\,0\,0) = \vec{x_5}$

$Bin(C_4) \vee Bin(C_6) = Bin(C_5) = \vec{g_5}$
Some of the found C_i's are new and they are given with their binary vector between parenthesis. The resulting binary tree is shown in Figure 5.14, which has more occupied leaves than the tree of Figure 5.10.

5.5.3 FLA to locate multiple failures in the non-ideal scenario

Note that in the previous example some leaves of the binary tree remain empty regardless of the number of failures. If the alarms received by the manager correspond to one of these empty leaves, there must have been lost and/or false alarms. In this case, what is the result that should be presented to the Human Manager?
The tree can be viewed as a particular block error-correcting code, whose codewords have the property that the logical OR of any two codewords is another codeword, as we have seen in Equation(5.10). One empty leaf of the tree corresponds to an erroneous word, and the error correction would be to replace it with the codeword whose Hamming distance with the received word is minimal. Note at this point that this code is not linear, and that it will have very poor performances in terms of minimal distance. Of course, in our case we do not have freedom in the choice of the generator codewords $\vec{g_1}, ..., \vec{g_t}$ as these are dictated only by the network topology and the established optical channels. We will discuss further the links with error correcting codes in Section 5.6.
Now, contrary to the use of error-correcting codes for data transmission, the manager of the network does not require a unique decoding. Indeed, he would prefer to get the set of all fault candidates whose domains are close to the received alarms. In fact, our proposed solution gives all the binary vectors that realize the given alarm mismatching thresholds. For example, if $m_1 > 0$ it means that we accept that a maximum of m_1 alarms will be lost, and therefore, the binary vectors that fall within this margin from the correct codewords are the binary vectors having a '1' when $Bin(\mathcal{R})$ has '0' at most m_1 positions. If $m_2 > 0$ it means that we accept a maximum of m_2 false alarms, and therefore, the binary vectors that lie within this margin from the correct codewords are the binary vectors having '0' when $Bin(\mathcal{R})$ has '1' at most m_2 positions.
This part can be computed off-line, in the *PCP* module, or on-line, when the alarms are received. In the first case, one computes for each occupied leaf (i.e., for each codeword), the leaves whose number of 0's and 1's differ by at most m_1 and m_2 positions, and the corresponding C_i's are added to the list pointed by this leaf (see for example in Figure 5.15(a) the new binary tree when $m_1 = 1$ and $m_2 = 0$ and in Figure 5.15(b) the new binary tree when $m_1 = 0$ and $m_2 = 1$). If the computation is done on-line, in the main part of the algorithm, the complexity is lower, as the previous computation is carried out only for $Bin(\mathcal{R})$ and not for all the leaves. The result of the algorithm is the set of components whose Domain(s) corresponds to one of these binary vectors.
Let us illustrate the algorithm with different scenarios in the example of Figure 5.9 when the set of received alarms is $\mathcal{R} = \{a_1, a_2\}$ (a_1 issued by e_1 and a_2 issued by e_2). Hence, $Bin(\mathcal{R})$ = (1 1 0 0) which corresponds to an empty leaf of the tree in Figure 5.14. Let us check different scenarios:

- $m_1 = 1, m_2 = 0$: One lost alarm and no false alarms are tolerated. In this case (see Figure 5.15(a)) the output of the algorithm is the leaf $Bin(C_{10})$ = (1 1 0 1) with one mismatch which corresponds to the two following solutions:
  `Failure of` p_7 `and` p_4 `with 1 mismatch`
  `Failure of` p_7 `and` p_5 `with 1 mismatch`



(a) Binary Tree when $m_1 = 1$ and $m_2 = 0$ (b) Binary Tree when $m_1 = 0$ and $m_2 = 1$

Figure 5.15: Binary Tree accepting mismatches

- $m_1 = 0, m_2 = 1$: One false alarm and no lost alarms are tolerated. In this case (see Figure 5.15(b)) the result is the leaf $Bin(C_3) = (1\ 0\ 0\ 0)$ and $Bin(C_6) = (0\ 1\ 0\ 0)$ with one mismatch which correspond to the following four solutions:
  `Failure of` p_4 `with 1 mismatch`
  `Failure of` p_5 `with 1 mismatch`
  `Failure of` p_8 `with 1 mismatch`
  `Failure of` p_9 `with 1 mismatch`

- $m_1 = m_2 = 1$: One alarm can be lost and another alarm can be false. In this case (see Figure 5.16) the additional results (to those of the two previous scenarios) are the new leaves $Bin(C_5) = (0\ 1\ 0\ 1)$ and $Bin(C_9) = (1\ 0\ 0\ 1)$, which correspond to:
  `Failure of` p_7 `with 2 mismatches`
  `Failure of` p_6 `and` p_4 `with 2 mismatches`
  `Failure of` e_3 `and` p_4 `with 2 mismatches`
  `Failure of` p_6 `and` p_5 `with 2 mismatches`
  `Failure of` e_3 `and` p_5 `with 2 mismatches`
  In the considered example that only has four $A2$ components, the tolerance $m_1 = m_2 = 1$ is too loose because it amounts the acceptance of up to 50% erroneous alarms, which is why there are so many $P(C_i)$ values at each leaf of the tree. In a more realistic case, the number of active elements is much larger and the number of $P(C_i)$ values per leaf is much lower. Hence the result is more selective.

5.6 Links between FLA and error-correcting codes

It is worth at this point to discuss the structure of the set $\mathcal{C}$ containing the n_a-uples $\vec{x}_i = Bin(C_i)$ corresponding to the different domains for single or multiple failures. As

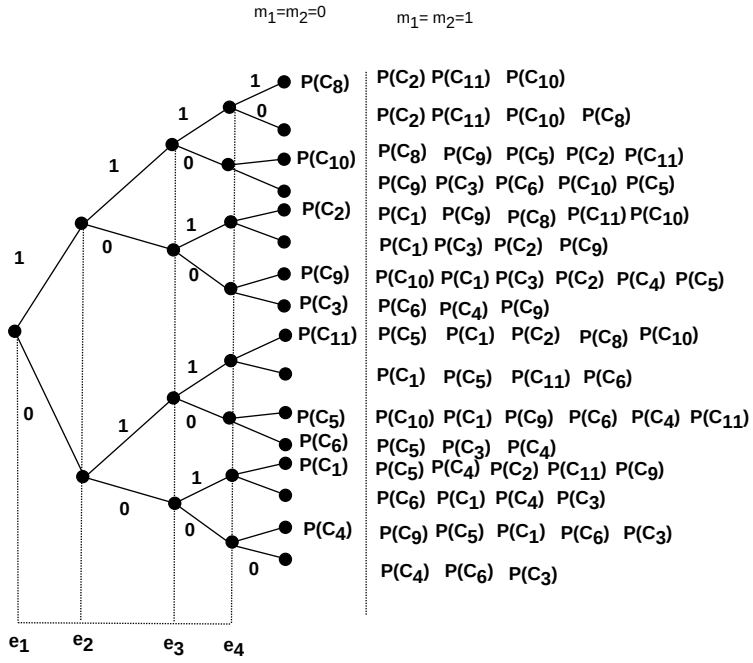


Figure 5.16: Binary Tree when $m_1 = m_2 = 1$

mentioned in the previous section, these binary vectors can be seen as codewords, and false or erroneous alarms will produce corrupted codewords. The FLA needs to fill all the leaves of a binary tree corresponding to codewords to perform decoding and error correction, and one can wonder whether a more efficient decoding, using the structure of the code, would not be possible and more efficient. It is therefore useful to compare this "code" $\mathcal{C}$ with traditional linear error-correcting codes.

A linear block code $\mathcal{C}_{lin}$ of size (n_a, t) generated by t codewords of n_a components each is defined by the property that any codeword $\vec{x} \in \mathcal{C}_{lin}$ can be obtained as a linear combination of the generator vectors $\vec{g}_1, \ldots, \vec{g}_t$, which are linearly independent and where the operation of addition is defined modulo 2 (i.e., is the logical XOR operation). In other words, with $\alpha_j \in \{0, 1\}$ for all $1 \leq j \leq t$,

$$\vec{x} = \alpha_1 \vec{g}_1 \oplus \ldots \oplus \alpha_t \vec{g}_t = \bigoplus_{j=1}^{t} \alpha_j \vec{g}_j. \tag{5.12}$$

The set of vectors generated by this rule forms a vector sub-space of the vector space of all binary vectors of n_a components, endowed with the operation of addition modulo 2 and multiplication by a binary scalar, which is denoted by $\{GF(2)^{n_a}, GF(2), \oplus\}$. $GF(2)$ stands for Galois Field, i.e. the set $\{0, 1\}$ endowed with operations $\oplus$ (addition modulo 2) and $\times$ (multiplication).

The sub-space $\{\mathcal{C}_{lin}, GF(2), \oplus\}$, or with a simpler notation, $\mathcal{C}_{lin}$, is the code. The fact that the code has an algebraic structure of vector space is very useful in determining all properties and performing error detection/correction. For example, the fact that $\mathcal{C}_{lin}$ is a vector subspace means that any linear combination of vectors of $\mathcal{C}_{lin}$ is again a vector of $\mathcal{C}_{lin}$, and that there will be exactly 2^t different codewords. Error detection, i.e., the verification that a given n_a-uple $\vec{y}$ belongs to $\mathcal{C}_{lin}$, is very simple. A set of $n_a - t$ linearly independent vectors $h_1, \ldots, h_{n_a-t}$ orthogonal to the subspace $\mathcal{C}_{lin}$ is first computed. It

suffices to check whether $\vec{y}$ is orthogonal or not to these $n_a - t$ vectors, which amounts to multiply $\vec{y}$ by a matrix whose columns are these $n_a - t$ vectors, to conclude that the given vector $\vec{y}$ is a codeword or not (this is known as syndrome decoding). Error correction also uses properties of the vector sub-space, and does not require a tree decoding to be efficient.
Here we have a code $\mathcal{C}$ generated by t codewords of n_a components each, and defined by the property that any codeword $\vec{x} \in \mathcal{C}$ can be obtained as a logical OR of any number of the generator vectors $\vec{g}_1, \ldots, \vec{g}_t$. In other words, with $\alpha_j \in \{0, 1\}$ for all $1 \leq j \leq t$,

$$\vec{x} = \alpha_1 \vec{g}_1 \vee \ldots \vee \alpha_t \vec{g}_t = \bigvee_{j=1}^{t} \alpha_j \vec{g}_j. \tag{5.13}$$

Although the only difference with the linear code generated by the rule (5.12) is that the operation $\oplus$ is now replaced by $\vee$, the set of vectors generated by (5.13) does however no longer form a vector space, and the code is non-linear. The reason why $\{\mathcal{C}, GF(2), \vee\}$ is no longer a vector space is due to the fact that $\{\mathcal{C}, \vee\}$ is no longer a commutative group, because the axiom of cancellation of the 'addition' (which is here the operation $\vee$) does no longer hold. This axiom states that for any vector $\vec{x} \in \mathcal{C}$, there exists a vector $-\vec{x} \in \mathcal{C}$ such that the addition of these two vectors is equal to the zero vector $\vec{0}$. If the 'addition' is $\vee$, there is no vector $(-\vec{x}) \in \mathcal{C}$ such that $\vec{x} \vee -\vec{x} = \vec{0}$. On the contrary, this axiom held for $\{\mathcal{C}_{lin}, \oplus\}$, because there always existed $(\vec{-x}) = \vec{x} \in \mathcal{C}$ such that $\vec{x} \oplus (\vec{-x}) = \vec{0}$. Therefore properties of linear codes do no longer apply. For example, the number of codewords is not equal to the same value 2^t, which is the dimension of the vector space spanned by the t generator vectors as before, but can be any number between 1 et 2^{n_a}, depending on the particular set of generator vectors.
The code $\mathcal{C}$ is non-linear, but it still possesses some structure, because all other axioms (associativity, commutativity, distributivity, etc) but cancellation still hold. Such a structure is known as a *moduloid* [49]. Although it is a structure weaker than a vector space, it is used in max-plus algebra to derive some system theoretic results. One is therefore tempted to look for some decoding properties similar to the syndrome decoding of linear code. In fact, error detection, i.e., the verification that a given n_a-uple $\vec{y}$ belongs to $\mathcal{C}$ is still possible without needing to have all codewords pre-computed and stored in a tree, thanks to the following theorem, which we state after a few simple definitions and a lemma.

Definition 1 *The Difference distance between two vectors $\vec{a}$,$\vec{b}$ is defined as*

$$d_D(\vec{a}, \vec{b}) = \begin{cases} 0 \ \textit{if} \ \ \vec{a} = \vec{b} \\ 1 \ \textit{if} \ \ \vec{a} \neq \vec{b} \end{cases}$$

One can easily check that it verifies all axioms defining a distance.

Definition 2 *A vector $\vec{a}$ is lower or equal to another vector $\vec{b}$ when they verify:*

$$\vec{a} \leq \vec{b} \quad \Leftrightarrow a_i \leq b_i \quad 1 \leq i \leq n_a.$$

We now state a few useful relations in the following lemma, where $\wedge$ stands for the pointwise logical AND (or minimum) operation. The proofs are trivial and left to the reader.

Lemma 1 *For any vectors* $\vec{a}$, $\vec{b}$,

$$\begin{aligned} \vec{a} \leq \vec{b} &\quad \Leftrightarrow \quad \vec{a} \wedge \vec{b} = \vec{a} \\ \vec{a} \leq \vec{b} &\quad \Leftrightarrow \quad \vec{a} \vee \vec{b} = \vec{b} \\ \vec{a} \vee \vec{b} &\quad = \quad \vec{a} + \vec{b} - (\vec{a} \wedge \vec{b}) \\ (\vec{a} \vee \vec{b}) \wedge \vec{c} &\quad = \quad (\vec{a} \wedge \vec{c}) \vee (\vec{b} \wedge \vec{c}). \end{aligned}$$

We now can state and prove the following theorem, which gives a direct answer to the question of determining whether a given vector $\vec{y}$ does or does not belong to $\mathcal{C}$, without having to compute all the codewords.

Theorem 1 *Given the (non-zero) generator vectors* $\vec{g_1}, ..., \vec{g_t}$ *of the code* $\mathcal{C}$*, and a binary* n_a*-uple* $\vec{y}$*, we have that*

$$\vec{y} \in \mathcal{C}$$

if and only if there exist $\alpha_j \in \{0, 1\}$, $1 \leq j \leq t$ *such that*

$$\vec{y} = \bigvee_{j=1}^{t} \alpha_j \vec{g}_j \tag{5.14}$$

with

$$\alpha_j = 1 - d_D(\vec{y} \wedge \vec{g}_j, \vec{g}_j). \tag{5.15}$$

Proof:

($\Rightarrow$) By definition of the code $\mathcal{C}$, if $\vec{y} \in \mathcal{C}$, we can express $\vec{y}$ by (5.14) for some binary constants $\alpha_j \in \{0, 1\}$, $1 \leq j \leq t$. We must show that there is such a set of binary constants verifying equation (5.15). Consider a set $\{\alpha_j, 1 \leq j \leq t\}$ that satisfies (5.14), and pick any of these α_j.

(i) Assume first that $\alpha_j = 1$. Then, because of the previous lemma, we have that

$$\begin{aligned} \vec{y} \wedge \vec{g}_j &= \left(\bigvee_{i=1}^{t} \alpha_i \vec{g}_i \right) \wedge \vec{g}_j \\ &= \bigvee_{i=1}^{t} \alpha_i (\vec{g}_i \wedge \vec{g}_j) \\ &= \alpha_j (\vec{g}_j \wedge \vec{g}_j) \vee \left(\bigvee_{i=1, i \neq j}^{t} \alpha_i (\vec{g}_i \wedge \vec{g}_j) \right) \\ &= \vec{g}_j \vee \left(\bigvee_{i=1, i \neq j}^{t} \alpha_i (\vec{g}_i \wedge \vec{g}_j) \right) \\ &\geq \vec{g}_j. \end{aligned}$$

On the other hand, one always have $\vec{y} \wedge \vec{g}_j \leq \vec{g}_j$ so that combining these two inequalities yields that $\vec{y} \wedge \vec{g}_j = \vec{g}_j$, and hence (5.15) is verified whenever $\alpha_j = 1$.

(ii) Assume next that $\alpha_j = 0$. We distinguish two subcases. In the first subcase, $\vec{y} \wedge \vec{g}_j \neq \vec{g}_j$,

so that Equation (5.15) is clearly satisfied. In the second subcase, $\vec{y} \wedge \vec{g}_j = \vec{g}_j$. Then $\vec{y}$ can also be expressed by

$$\vec{y} = \bigvee_{i=1}^{t} \alpha'_i \vec{g}_i$$

where

$$\alpha'_i = \begin{cases} 1 & \text{if } i = j \\ \alpha_i & \text{if } i \neq j. \end{cases}$$

Indeed, since $\alpha_j = 0$ and using the lemma, we can write that

$$\begin{aligned} \bigvee_{i=1}^{t} \alpha'_i \vec{g}_i &= \vec{g}_j \vee \left(\bigvee_{i=1, i\neq j}^{t} \alpha_i \vec{g}_i \right) \\ &= \vec{g}_j \vee \left(\bigvee_{i=1}^{t} \alpha_i \vec{g}_i \right) = \vec{g}_j \vee \vec{y} \\ &= \vec{g}_j + \vec{y} - (\vec{g}_j \wedge \vec{y}) = \vec{g}_j + \vec{y} - \vec{g}_j = \vec{y} \end{aligned}$$

Clearly, $\alpha'_j = 1$ satisfies Equation (5.15).
We have therefore shown that whenever $\vec{y} \in \mathcal{C}$, there always exist a set of t binary constants $\alpha_1, ..., \alpha_t$ that verify Equations (5.14) and (5.15).

($\Leftarrow$) Obvious. ■

From this theorem, one can therefore check whether the binary vector associated to the received alarms, which is $Bin(\mathcal{R})$, corresponds to a single or multiple failure when no alarms are lost or false. In this case, $\vec{y} = Bin(\mathcal{R})$ and one checks whether $d_D(\vec{y}, \vee_{i=1}^{t} \alpha_i \vec{g}_i) = 0$ with the α_i given by (5.15), is verified. If so, the fault candidates are the sets $P(C_i)$ pointed by the vectors $\vec{g}_i$ associated to the coefficients $\alpha_i = 1$. This set of fault candidates is *not* however the minimal set of elements whose failure explain the received alarms, but on the contrary, the maximal set. Hence the result of this simple check is not the solution to our problem.

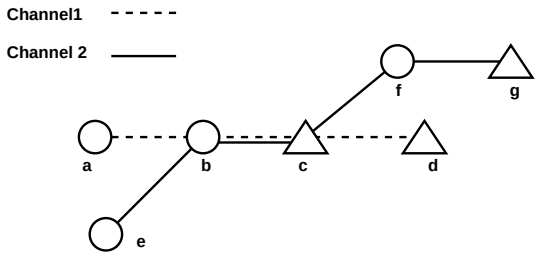


Figure 5.17: Two established channels with two kinds of network components.

As an example, take the network of Figure 7.1. Here $\mathcal{V} = \{a, b, c, d, e, f, g\}$ and their domains are:
$Domain(a) = \{c, d\} = C_1$
$Domain(b) = \{c, d, g\} = C_2$
$Domain(c) = \{d, g\} = C_3$
$Domain(e) = \{c, g\} = C_4$
$Domain(f) = \{g\} = C_5$
$Domain(d) = Domain(g) = \emptyset$

Therefore, $n_a = 3$ and $t = 5$, as there are here five generator codewords, namely

$$\vec{g_1} = Bin(C_1) = (1\,1\,0)$$
$$\vec{g_2} = Bin(C_2) = (1\,1\,1)$$
$$\vec{g_3} = Bin(C_3) = (0\,1\,1)$$
$$\vec{g_4} = Bin(C_4) = (1\,0\,1)$$
$$\vec{g_5} = Bin(C_5) = (0\,0\,1).$$

Suppose that the received alarms are issued by c, d, g. The resulting coefficients from Equation (5.15) are $\alpha_1 = \alpha_2 = \alpha_3 = \alpha_4 = \alpha_5 = 1$, which gives the maximum number of faulty elements that explain the received alarms: $\{a, b, c, e, f\}$. The minimal set is obtained with $\alpha_1 = 0, \alpha_2 = 1, \alpha_3 = 0, \alpha_4 = 0, \alpha_5 = 0$, which corresponds to the single failure of $\{b\}$.
In fact, the use of this theorem brings us to a somewhat similar stage as the backward approach in the AFA, namely a set of fault candidates from which a minimal subset needs to be extracted. This is the NP-hard part, as we will see in Chapter 7. As the backward phase can be computed only upon reception of the alarms, and not off-line in the PCP phase, we decided not to apply this check in the FLA.
The second difference with error correction is that, as mentioned in Section 5.5.3, the decoding is not unique. These differences oblige us to pre-compute all codewords in a binary tree during the PCP phase, as explained in the previous section.

5.7 Examples

In this section we will present the FLA algorithm behaviour in two different kinds of networks. First, an SDH network will be used as failure scenario and then an IP/WDM network with different kinds of failures.

5.7.1 SDH Network

The SDH network shown in Figure 5.12 has been used to test the proposed Fault Localization Algorithm (FLA). In this network, three channels have been considered and

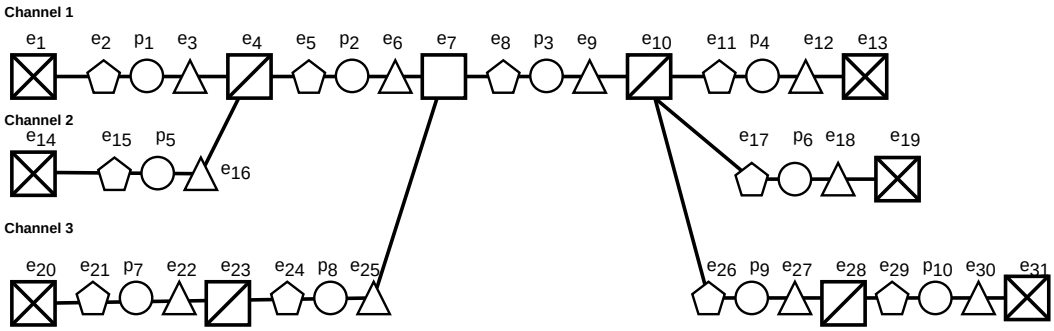


Figure 5.18: Modelization of the three SDH channels over WDM

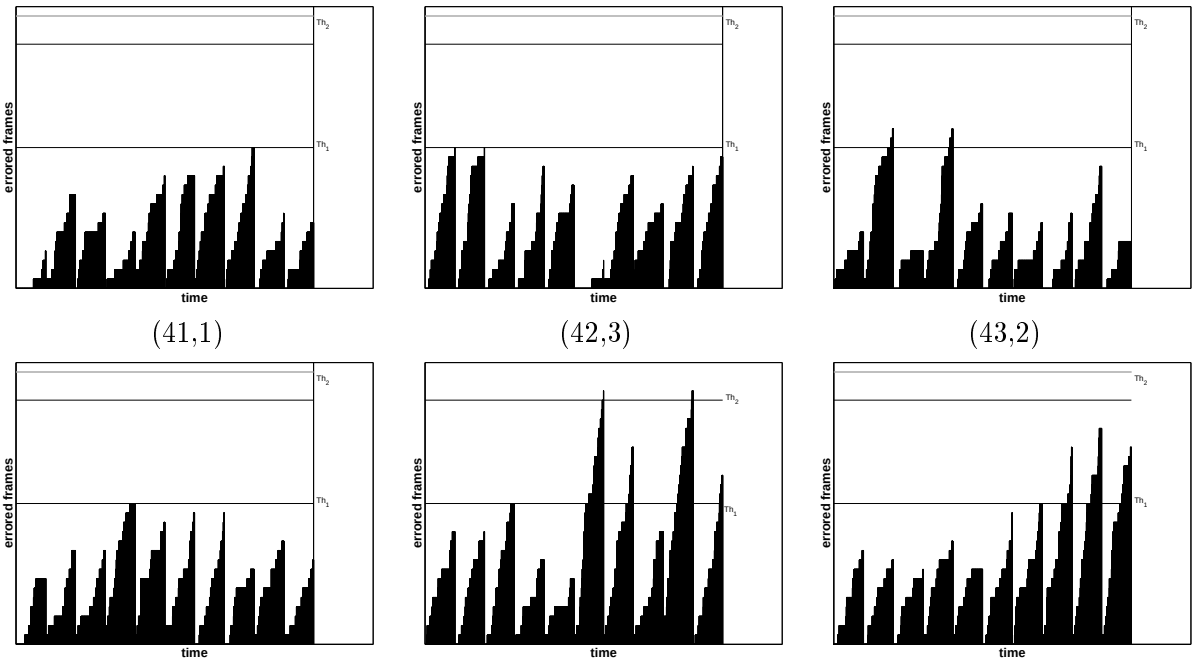
modeled in Figure 5.18 using the abstraction of the hardware components and monitoring equipment of Section 5.3. As it was presented in Section 3.4.2, the SDH monitoring ele-

e_1	(40 1)	e_2	(3 1)	e_3	(2 1)	e_4	(42 1)	e_5	(3 2)	e_6	(2 2)
e_7	(41 1)	e_8	(3 3)	e_9	(2 3)	e_{10}	(42 3)	e_{11}	(3 4)	e_{12}	(2 4)
e_{13}	(43 2)	e_{14}	(43 3)	e_{15}	(3 5)	e_{16}	(2 5)	e_{17}	(43 6)	e_{18}	(2 6)
e_{19}	(43 4)	e_{20}	(43 5)	e_{21}	(3 7)	e_{22}	(2 7)	e_{23}	(42 2)	e_{24}	(3 8)
e_{25}	(2 8)	e_{26}	(3 9)	e_{27}	(2 9)	e_{28}	(42 4)	e_{29}	(3 10)	e_{30}	(2 10)
e_{31}	(43 6)	p_1	(0 1)	p_2	(0 2)	p_3	(0 3)	p_4	(0 4)	p_5	(0 5)
p_6	(0 6)	p_7	(0 7)	p_8	(0 8)	p_9	(0 9)	p_{10}			

Table 5.5: Mapping between the identifiers used in Figure 5.18 and the presented results of FLA

ments count the number of errored blocks over time-windows of 15 minutes or 24 hours. The error counter is reset at each new time window. Whenever the number of errored blocks overpasses a threshold, an alarm ("Degraded Signal" or "Excessive Error") is sent, as specified by the ITU standard G.783, which assumes either a Poisson distribution or a bursty distribution of the errored blocks [35].

Suppose first that the distribution is Poisson, with two parameters $\lambda_1 < \lambda_2$: λ_1 corre-



(41,1) (42,3) (43,2)

(43,4) (42,4) (43,6)

Figure 5.19: Block Error Counting when errors follow a Poisson distribution with λ_1 for (41,1),(42,3),(43,2) and (43,4) and with $\lambda_2 > \lambda_1$ for (42,4)and (43,6)

sponds to the correct functioning (no failure) and λ_2 to the situation where a progressive failure has occurred. We have simulated the time-series of the error counts registered at the six monitoring components (41,1), (42,3), (43,2), (43,4), (42,4) and (43,6), shown in Figure 5.19. The four first ones correspond to a good-functioning (that is, the error distribution has λ_1) whereas two last ones should indicate that a soft failure occurred during the fifth time window (that is, the error distribution has λ_2). We present the results of the FLA for two different thresholds $Th_1 < Th_2$.

If the threshold is set too high (as is the case with Th_2), no false alarm is sent, but there

are missing alarms as the alarm from *(43,6)* in Figure 5.19. Running the algorithm with $m_1 = 1$ and $m_2 = 0$ yields the following output:

```
(3,9) 1 mismatch
(0,9) 1 mismatch
(2,9) 1 mismatch
```

Conversely, if the threshold is set too low (as is the case with Th_1), no missing alarm is sent, but false alarms are received from *(43,2)* in Figure 5.19. Running the algorithm with $m_1 = 0$ and $m_2 = 1$ gives the following output:

```
(3,9) 1 mismatch
(0,9) 1 mismatch
(2,9) 1 mismatch
(3,9) and (3 4) or (0 4) or (2 4) 0 mismatch
(0,9) and (3 4) or (0 4) or (2 4) 0 mismatch
(2,9) and (3 4) or (0 4) or (2 4) 0 mismatch
```

We note that in both cases the set of fault candidates presented to the manager include the actual fault elements, but few correctly working elements. The set of fault candidates is therefore both complete and selective, which are two desired quality of a fault location method. Note also that running the algorithm in the ideal scenario would have missed all fault candidates in both cases, but that other thresholds may require higher values of m_1 and/or m_2.

If the distribution of the errored block is heavy tailed, an appropriate value of the thresholds is even more difficult to pick, and the number of false or lost alarms will be larger. A typical error count assuming a Pareto distribution is shown on Figure 5.20. In this case, a higher value of the mismatching thresholds is needed, unless more sophisticated processing (filtering) of the time series of the errored blocks than a elementary thresholding over fixed time windows of 15 minutes can be implemented. It is indeed desirable to keep small mismatching thresholds, for a better selectivity in the fault candidates, and for a smaller complexity of the pre-computation phase of the FLA. This is left for future work.

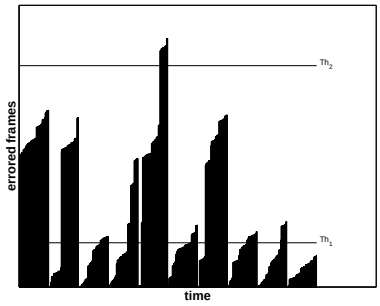


Figure 5.20: Block Error Counting when errors follow a Pareto distribution

5.7.2 IP/WDM Network

Let us illustrate the algorithm in the following example of an IP/WDM network, where WDM monitoring equipment has been introduced after the add/drop filters and after the full-optical switch. We have considered an IP/WDM network with the same topology as in Figure 5.21. Four channels have been considered, as modeled in Figure 5.22. The number of elements in the network is higher: 44 network components, 13 of them 'non-alarming'. Several failure situations have been simulated.

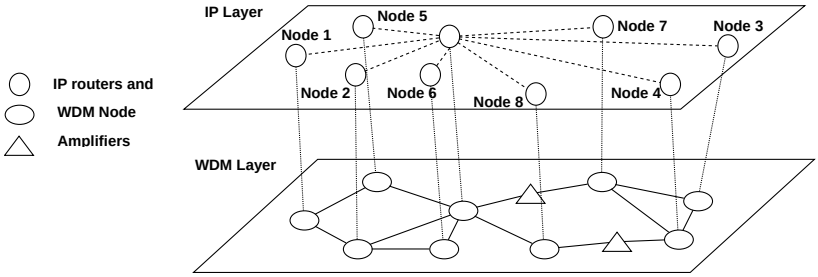


Figure 5.21: Example of an IP/WDM Network where two channels have been established: from Node 1 to Node 3 and from Node 2 to Node 4. Each of these nodes has an IP address

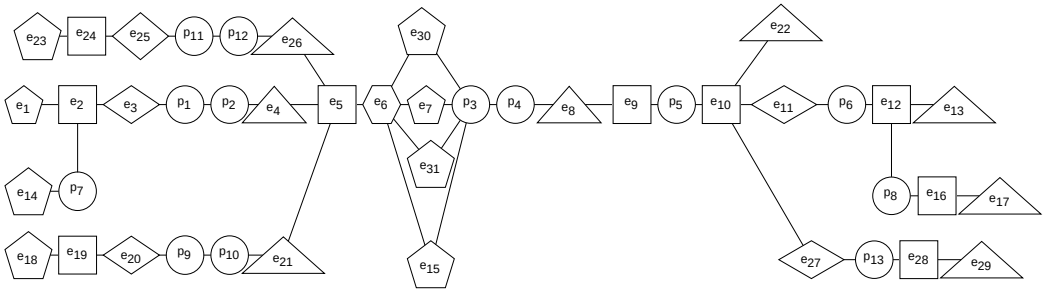


Figure 5.22: Modelization of 4 channels established in the network of Figure 5.21

Hard failure example

The alarms received by the manager are coming from elements e_{13} and e_{17} which are receivers, thus only sensitive to hard failures. The result with the mismatching thresholds $m_1 = m_2 = 0$ is shown in Figure 5.23. When the first alarm reached the manager, the associated leaf was empty. Once the second alarm reached the manager, the associated leaf was not longer empty but pointed to the p_6 component.

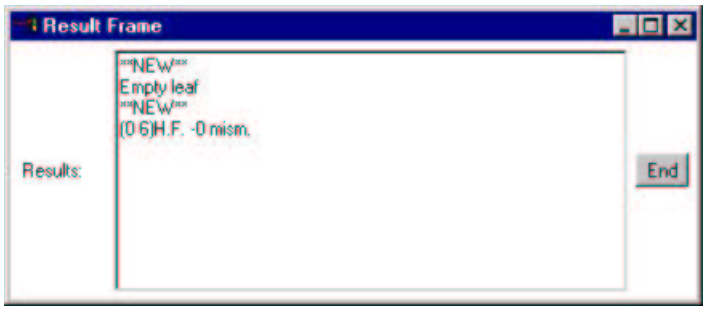


Figure 5.23: Result screen when the received alarms are issued by e_{13} and e_{17} and $m_1 = m_2 = 0$

Soft failure example

In this case the alarms received by the manager have been issued by e_{25} and e_6 but not from the receiver which is between them. Considering the mismatching thresholds $m_1 = m_2 = 0$, the result of our FLA are:

e_{23} `S.F. 0 mism.` e_{24} `S.F. 0 mism.`

e_1	(3 2)	e_2	(1 2)	e_3	(40 2)	e_4	(2 1)	e_5	(1 4)	e_6	(41 1)
e_7	(3 5)	e_8	(2 4)	e_9	(1 5)	e_{10}	(1 6)	e_{11}	(40 4)	e_{12}	(1 7)
e_{13}	(2 7)	e_{14}	(3 3)	e_{15}	(3 6)	e_{16}	(1 8)	e_{17}	(2 8)	e_{18}	(3 4)
e_{19}	(1 3)	e_{20}	(40 3)	e_{21}	(2 3)	e_{22}	(2 5)	e_{23}	(3 1)	e_{24}	(1 1)
e_{25}	(40 1)	e_{26}	(2 2)	e_{27}	(40 5)	e_{28}	(1 9)	e_{29}	(2 6)	e_{30}	(3 7)
e_{31}	(3 8)	p_1	(0 1)	p_2	(0 2)	p_3	(0 3)	p_4	(0 4)	p_5	(0 5)
p_6	(0 6)	p_7	(0 7)	p_8	(0 8)	p_9	(0 9)	p_{10}	(0 10)	p_{11}	(0 11)
p_{12}	(0 12)	p_{13}	(0 13)								

Table 5.6: Mapping between the identifiers used in Figure 5.22 and the result of FLA shown in Figure 5.23

Multiple failure example

The alarms received by the manager have been issued by e_{25}, e_{21} and e_6. Accepting one lost alarm, that is, $m_1 = 1$ and $m_2 = 0$, the result of the algorithm is:

e_{23} `S.F. and` p_9 `H.F. 0 mism.`
e_{24} `S.F. and` p_9 `H.F. 0 mism.`
e_{23} `S.F. and` e_{18} `H.F. 1 mism.`
e_{24} `S.F. and` e_{18} `H.F. 1 mism.`
e_{23} `H.F. and` p_9 `H.F. 1 mism.`
e_{24} `H.F. and` p_9 `H.F. 1 mism.`

e_{23} `S.F. and` p_{10} `H.F. 0 mism.`
e_{24} `S.F. and` p_{10} `H.F. 0 mism.`
e_{23} `S.F. and` e_{19} `H.F. 1 mism.`
e_{24} `S.F. and` e_{19} `H.F. 1 mism.`
e_{23} `H.F. and` p_{10} `H.F. 1 mism.`
e_{24} `H.F. and` p_{10} `H.F. 1 mism.`

5.8 Conclusion

We have presented a Fault Localization Algorithm (FLA) able to detect multiple hard and soft failures in a WDM network. Most of the processing time of the FLA is spent in a Pre-Computation Phase, so that the fault location when alarms are received by the manager is rapid. This makes it particularly appropriate for large meshed networks, the advantage for small ring networks, such as the COBNET network, or the networks used with the simulator `OptSim` is less critical.
The second attractive feature of our algorithm is the combination of the information at the WDM layer on hard failures with information at WDM, SDH or IP layers on soft failures. The latter failures are revealed by signals such as BER, for which thresholds are not easy to set, potentially yielding false or lost alarms. The robustness of our FLA to these is therefore a third feature important in practice.
Finally, we discussed the similarities and differences between our FLA and error-correcting codes.

Chapter 6

Simulation Results

6.1 Introduction

In this chapter we present the simulation results of the fault diagnosis algorithms that have been introduced in the previous chapters. These results cover different failure scenarios and different networks.
We begin by briefly describing the implementation of the AFA and FLA algorithms in Java, together with the Graphical User Interface where the inputs and outputs are displayed. We then apply the algorithms to different failure scenarios. Unfortunately, we did not have the opportunity to perform tests on real optical testbeds. In order to obtain analog information from the WDM layer as close to reality as possible, we used an optical system simulator `OptSim` © [2] to emulate the hard and soft failures that may occur, and generate the corresponding alarm signals, as explained in Section 6.3. With the inputs obtained from this simulator tool, Section 6.4 reports the performance of both algorithms for both hard and soft failures at the WDM layer. The chapter concludes with a summary of the performance results of both algorithms.

6.2 Implementation of the algorithms

We begin the description of the algorithm implementations by presenting some key programming features of the code.

6.2.1 Code

The algorithms have been implemented in Java so that they can be easily integrated in Web-based management systems. We will present two main features of the algorithm implementation.

Dynamic Lists

Dynamic lists have been used for their adaptability to any modification of the input. Both the set of established channels $\mathcal{CH}$ and the set of received alarms $\mathcal{R}$ are dynamic lists, like the one shown in Figure 6.1. The dynamic list is called `ListGeneral` and every element of this dynamic list is called `List`. Each *List* has two pointers, one to the previous `List` and the other to the next `List`, and an element that can be an alarm, a channel or any other type of variable. Every event from the network will update the associated `ListGeneral`.

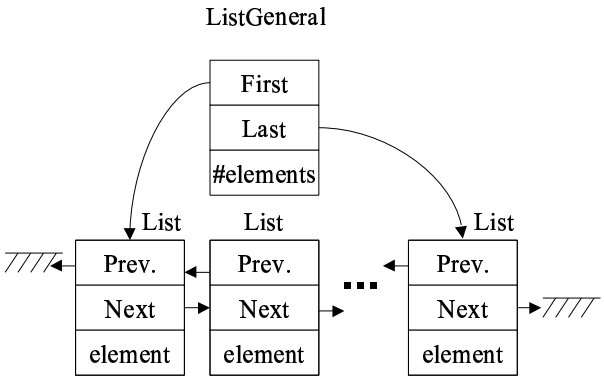
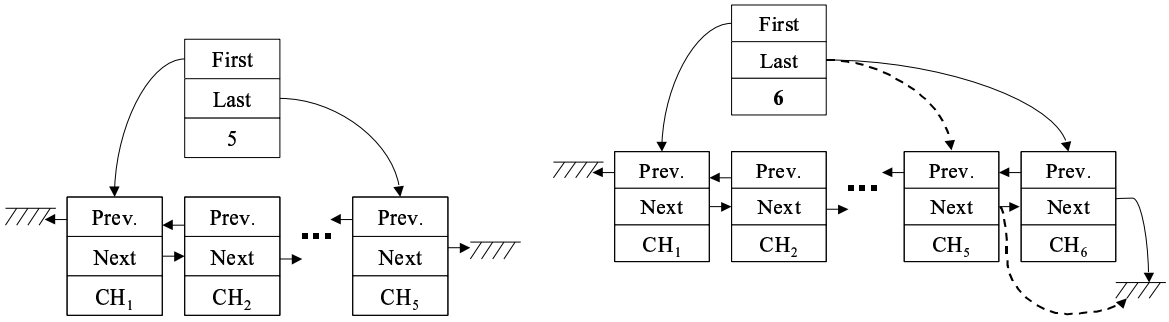


Figure 6.1: General structure of a dynamic list, where *#elements* gives the number of elements in `ListGeneral`

- If a new channel is established, which corresponds to a channel event, the `ListGeneral` $\mathcal{CH}$ will be updated as shown in the example of Figure 6.2. In (a) there is the `ListGeneral` $\mathcal{CH}$ with 5 channels and in (b) the updated the `ListGeneral` $\mathcal{CH}$ is shown.
  The AFA algorithm will have to update the domain of the new channel's components and re-compute all the modules to give a new result. The FLA algorithm will compute the domains of the elements belonging to the new channel and find the associated leaves of the binary tree.



(a) $\mathcal{CH}$ with 5 established channels (b) $\mathcal{CH}$ with 6 established channels

Figure 6.2: `ListGeneral` $\mathcal{CH}$ updated when a new channel is established. The arrows represent pointers: dashed arrows are the pointers before the event occurs and plain arrows are the ones after the event.

- If a channel is cleared down, which is one kind of channel event, the `ListGeneral` $\mathcal{CH}$ will be updated. For example, if CH_2 of Figure 6.2(a) is removed, the `ListGeneral`$\mathcal{CH}$ will be updated as shown in Figure 6.3.

The same flexibility occurs with the `ListGeneral` of received alarms $\mathcal{R}$. The advantage of these lists is their ease of updating which avoids a complete re-computation of the algorithm.

Parallelism

In order to reduce the computation time when an alarm event occurs and present the possible fault candidates as fast as possible to the Human Manager, the modules that

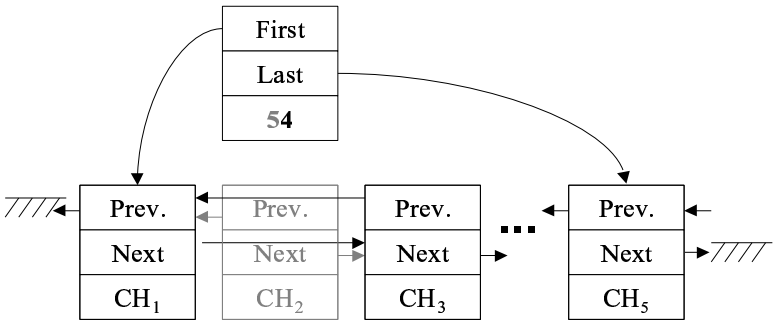


Figure 6.3: `ListGeneral` $\mathcal{CH}$ after removing CH_2

depend only on the channel events have been implemented independently from those that depend on the alarm events. In this way, these modules can be computed before alarms reach the manager.

- The AFA algorithm computes the *Domain_Calc* module for every channel event. When there is an alarm event, this module does not have to be computed contrary to the others.
- The FLA algorithm keeps most of the complexity in the modules which are computed when a channel event occurs. These modules form the Pre-Computing Phase (PCP) of the FLA algorithm. When there is an alarm event, the only modules that have to be computed are: to compute $Bin(\mathcal{R})$ and to find and present the associated $P(C_i)$. These modules have a minimal complexity as it is presented in Section 7.3.

6.2.2 Graphical User Interface

In this section we will present the Graphical User Interface (GUI) of the proposed algorithms. This GUI has two different sets of windows: the windows for the entry of the input to the algorithm and the windows illustrating the result of the algorithm.

Input to the algorithm

The algorithms have three different inputs: channels, mismatching thresholds and alarms. The two first inputs are set by the manager, whereas the last input is directly fed by the management platform that interacts with the network components. In our case, all the inputs will be set by us except for the alarms, which will sometimes also be set by `Optsim` tool.
Channels and alarms share the same window, as the one presented in Figure 6.4. The

Figure 6.4: Example of the window after when inserting a channel

mismatching thresholds in the actual implementations are inserted through a separate window, as the one shown in Figure 6.5. First the manager is asked to enter the allowed number of lost alarms and next, the allowed number of false alarms. Some further improvements can be done on this point, as it has been presented in Section 8.

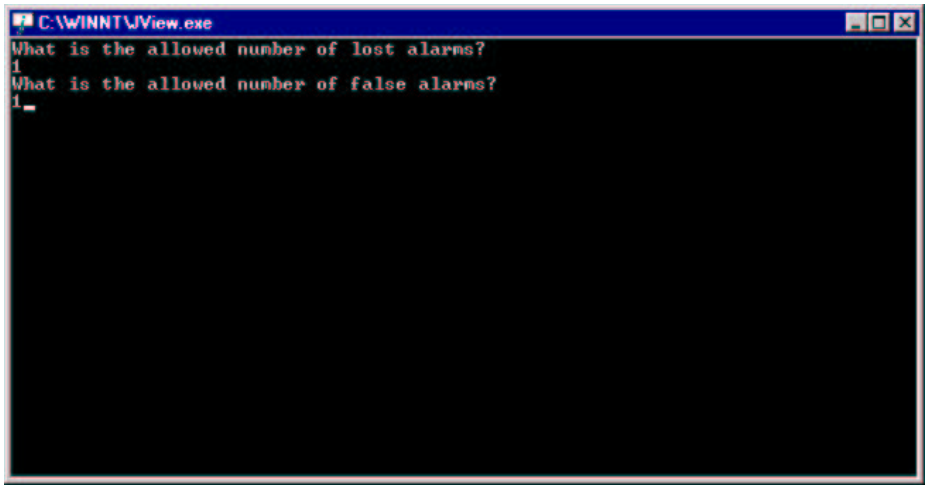


Figure 6.5: Window that allows inserting the mismatching thresholds

Output of the algorithm

Two windows display the result of the algorithms.
The first one presents the list of fault candidates together with their corresponding mismatch value (see Figure 6.6). Every time the algorithm provides a new result, that is, every time there is a new alarm or channel event, a *"NEW"* or *"New result"* message is written on the screen. At the FLA output, the type of failure is specified on the screen: *H.F.* stands for hard failure and *P.F.* stands for progressive failure.

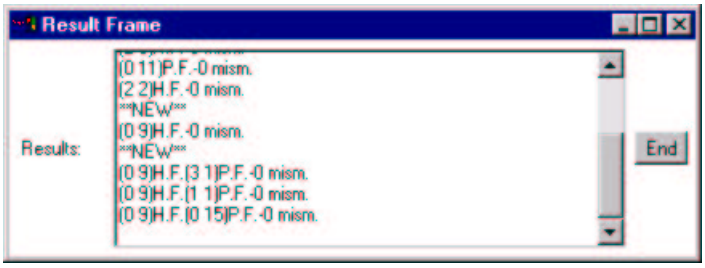


Figure 6.6: Window displaying the list of fault candidate elements

The second window displays the established channels, coloring in red the elements that have sent the received alarms and in green the elements whose failure corresponds to the ideal case (when the received alarms are exactly the expected alarms when these elements fail). An example is shown in Figure 6.7.

6.3 `OptSim` Optical System Simulation Tool

`OptSim` is a high-end optical systems simulator developed, marketed and supported by *ARTIS Software Corporation* [2]. `OptSim` includes a vast library of the most frequently used components in electro-optical systems focusing, among others, on WDM systems. Two advantages of this software are that it is a stand-alone product and has a user friendly interface.
This tool offers the user the possibility to choose the network components from a wide list of possible components, to modify some of their characteristic parameters and to put them

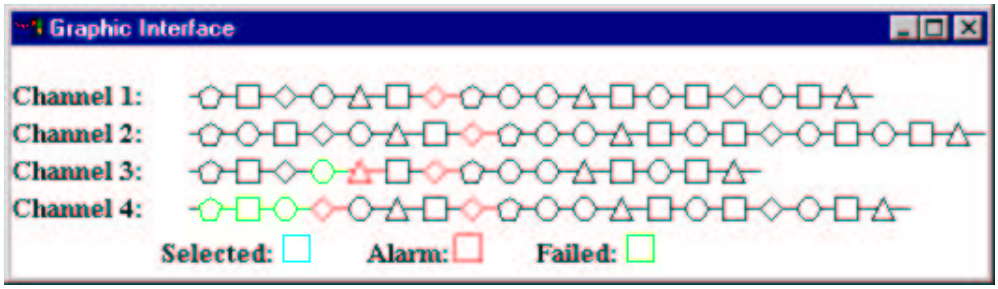


Figure 6.7: Window displaying an schematic view of the established channels that points out the alarming elements and the perfect matching candidates

together so that they form the desired topology. The user can then chose the monitoring equipment(s) he/she needs and add it(them) at the wished points. The next section lists the most important monitoring equipment.
`OptSim` offers a user-friendly graphical interface, as the one presented in Figure 6.8. On the left side different components can be chosen to be included in the right side of the screen. Several parameters of these components can be modified, depending on the user's needs.

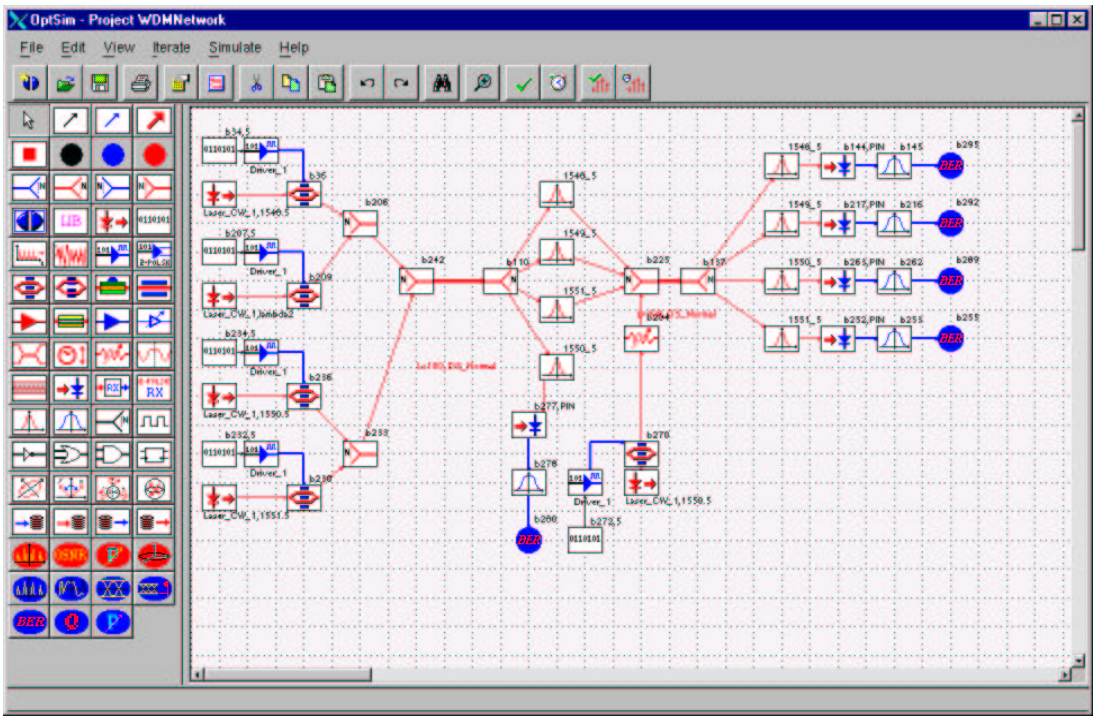


Figure 6.8: Graphical Interface of `OptSim` presenting the simulated 4-channel system

A useful feature of `OptSim` is the possibility of running several simulations differing some parameters by their numerical value. This feature is called *Multiple run* and it will allow us to observe the variation of performance variables such as BER, when a parameter of a network component is altered (for example, the central wavelength of a filter, as we will see in Section 6.4).
The monitoring equipment provided by the software can be classified in two categories:

electrical equipment measure parameters of electrical signals and *optical equipment* measure parameters of optical signals. The most interesting components for our simulations, either for designing a good-functioning network or as monitoring equipment, are:

6.3.1 Electrical measurement components

- Eye Diagram: This module generates the eye diagram of the input electrical signal. In laboratory measurements, such a diagram would be displayed by oscilloscopes [50]. The concept of an eye diagram is illustrated by Figure 6.9. This diagram overlaps

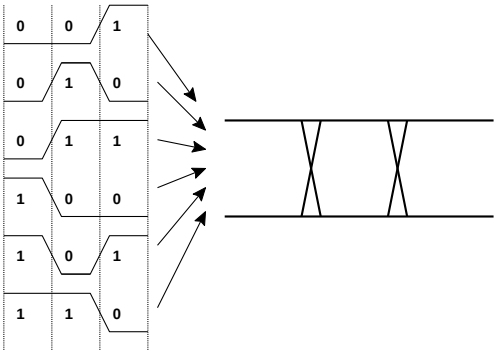


Figure 6.9: Sketch of an eye diagram

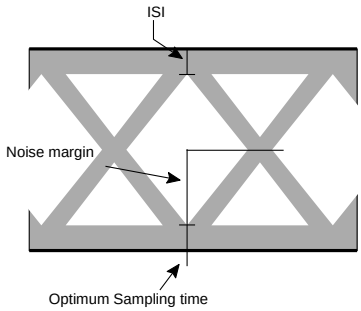


Figure 6.10: Binary eye pattern

the waveforms measured at some point in the network for possible input binary patterns. It is a multi-valued display because for every point on the time axis, several voltage values are associated with it.
The quality of the signal is measured by the eye shape, that is, by the amplitude and the width of the eye. The more open the eye, the better the quality of the signal. Figure 6.10 represents a generalized binary eye pattern with labels identifying the most significant features [51]: the *optimum time sampling* corresponds to the maximum eye opening. *ISI* at this time partially reduces the eye and reduces the *noise margin*.
The eye diagram can not be used as an input of the algorithm due to the way data is presented, whereas some parameters calculated from the eye, such as the BER or the Q-parameter, can.

- BER Estimator: This module estimates the BER of a received electrical signal for a binary modulation. The evaluation of the BER in an optical simulation is not a trivial task. The error counting is impractical because the BER is typically to the

order of 10^{-9} or less. Therefore, the calculation of the BER is based on the eye diagram and its histogram. The estimator assumes that '1' and '0' are equiprobable

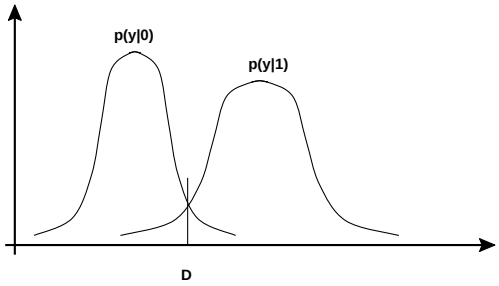
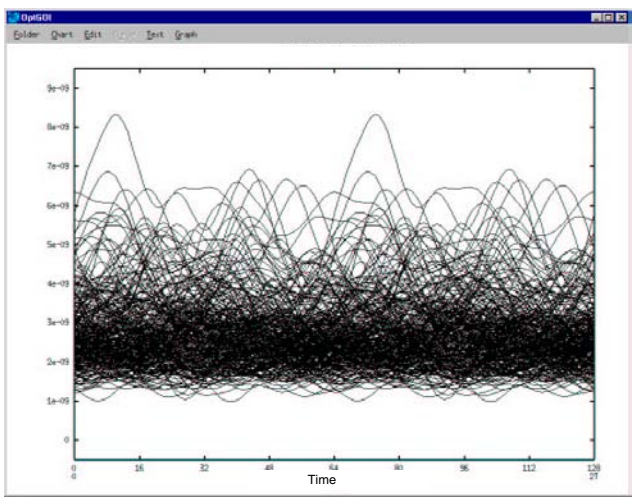


Figure 6.11: Ideal Detection

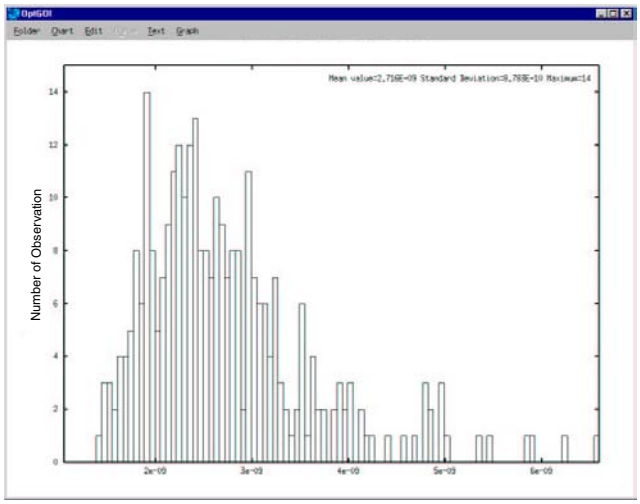
and it finds the first two moments of the received signals at the optimum sampling instant (m_0, σ_0 and m_1, σ_1) using a Gaussian or a χ^2 distribution (see Figure 6.11). These values give the BER based on the following expression:

$$P(e) = \frac{1}{4}[erfc(\frac{m_1 - V_{th}}{\sqrt{2}\sigma_1}) + erfc(\frac{m_0 - V_{th}}{\sqrt{2}\sigma_0})] \tag{6.1}$$

where V_{th} is chosen to minimize $P(e)$ and `erfc` is the complementary error function. The Q-parameter can be obtained from the eye diagram by using the formula $Q = \frac{m_1 - m_0}{\sigma_0 + \sigma_1}$.



(a) Extremely noisy eye diagram



(b) Histogram associated to the eye diagram

Figure 6.12: Example of an eye diagram yielding on erroneously optimistic value of BER

This estimation of the BER is far from being rigorous, and the estimated BER can be very wrong. An example found during our tests is shown in Figure 6.12. In this case, the eye diagram of (a) is completely deformed, and the expected BER must be very high. But surprisingly, the estimated BER from the method described above, is 1.50 10^{-7}, which is significantly better than expected. Therefore, the conclusion, as mentioned in the previous chapter, is that the estimated BER is not always a reliable measurement parameter by itself, whereas its variation can be a sign of improvement or degradation of the signal. Some work is being done to improve the

BER evaluation as proposed by Weinert [52]. Nevertheless, the difficulty to estimate correctly the BER justifies the need of a fault management system that accepts the existence of false and lost alarms, as we propose.

6.3.2 Optical measurement components

- Optical Spectrum Analyzer: This module emulates an optical spectrum analyzer for optical signals. The power spectrum is estimated using the method of *modified periodograms* that consists of sectioning the entire data sequence into a number of parts each containing N samples. The Fourier transform is computed for each segment and the amplitudes are squared and averaged. An example is shown in Figure 6.13. This diagram can not be an input of the algorithm itself but some of its parameters, such as wavelength stability or OSNR, can.

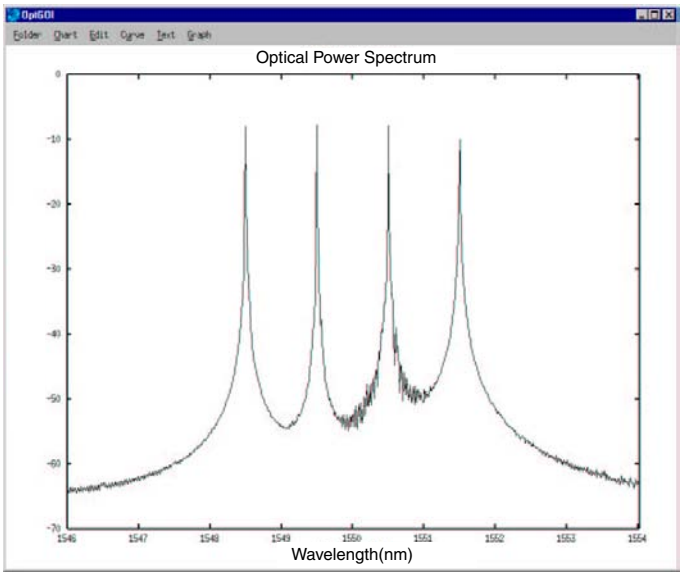


Figure 6.13: Example of the a spectrum emulated by an optical spectrum analyzer

- Optical Signal to Noise Ratio (OSNR): This modules searches the peaks in the Optical Signal Spectrum (given by the Optical Spectrum Analyzer), interprets them as channels, and for each of them, evaluates three parameters:
  - the average power of the channel over the *integration bandwidth* given by the user. The *integration bandwidth* is the expected bandwidth of the channel.
  - the average noise of the channel over the *integration bandwidth* given by the user.
  - the ratio between the two previous parameters.

  Channels are recognized when the difference between the value at one point, which corresponds to a relative maximum in the spectrum, and the value at the *integration bandwidth* around that point is greater than the *level Difference* parameter (see Figure 6.14).

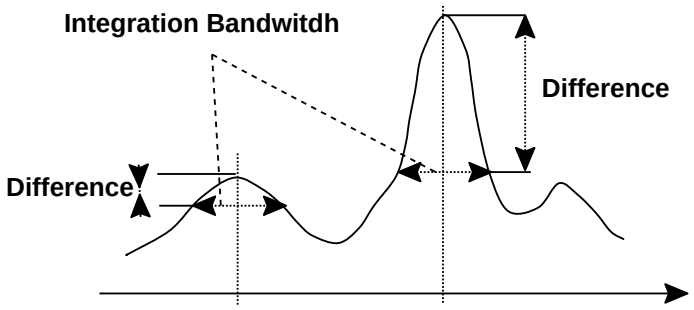


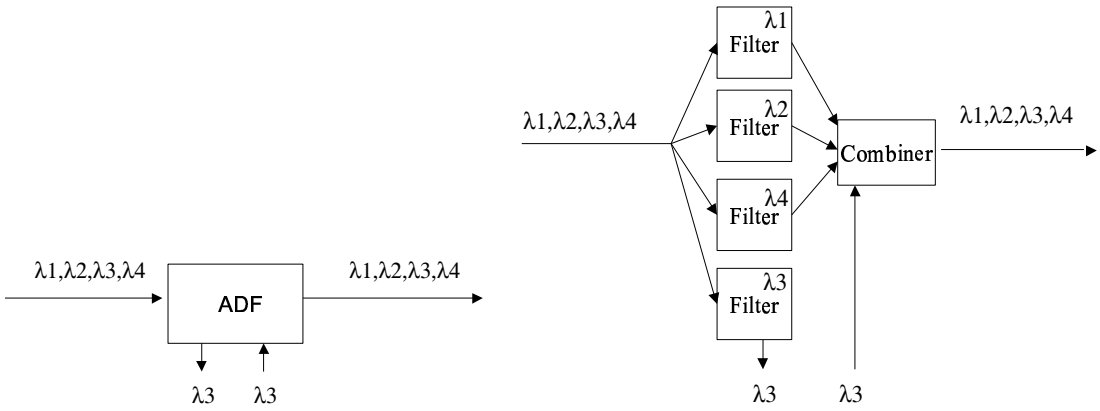
Figure 6.14: Functioning of the OSNR: When *Difference* is greater than the *level Difference* parameter, the OSNR considers it a channel and the evaluation of the parameters is performed

6.4 WDM Networks Simulations

We have used two different WDM network topologies, and we have simulated different failures for each topology. The performance values obtained by the `OptSim` simulator, as described in Section 6.3, are fed as input of the measuring equipment of the FLA algorithm and the expected alarms from the hardware components, given by us, as input of both AFA and FLA algorithms.
The goal of these simulations was to study the performance of both algorithms for each failure scenario, while paying special attention to the problem of setting the alarm threshold. Indeed, the problem appears when the manager has to decide where to put this threshold, that is, at which value of the measured parameter the manager considers that there is a problem. Because this problem does not have a solution, missing and false alarms have to be considered as a reality and our proposed algorithms have to cope with this possibility.

6.4.1 4-channel WDM link

A 4-channel 1-nm spaced WDM system has been simulated with `OptSim`. The Graphical User Interface of the simulated system is shown in Figure 6.8. Each channel runs at 5



(a)ADF (b) ADF Model

Figure 6.15: Modeling of an Add-Drop Filter

Gbps rate and is transmitted over a medium haul, non-repeater system. The emitted

Monitoring Equipment	BER
At ADF output	0
Channel1	$1.31\ 10^{-11}$
Channel2	$5.69\ 10^{-10}$
Channel3	$5.86\ 10^{-12}$
Channel4	$5.60\ 10^{-10}$

Table 6.1: BER at the monitoring equipment during normal functioning of the network.

wavelengths are λ_1=1548.5 nm, λ_2=1549.5 nm, λ_3=1550.5 nm and λ_4=1551.5 nm and they are carrying Channel1, Channel2, Channel3 and Channel4 respectively. The four channels are generated by modulating a Continuous Wave (CW) laser with an external modulator, pre-amplified using an Erbium Dopped Fiber Amplifier (EDFA), and sent on a first optical link, which is a 100 km long fiber. After this first optical link, the third channel is dropped and added by an ADF (Add-Drop Filter). This ADF has been modeled by demultiplexing all the channels with different optical filters, as shown in Figure 6.15. These optical filters have a bandwidth of 10 GHz. After the ADF, there is a second optical link, which is another 100 km long fiber, at the end of which there is a filter array where each filter is centered at each channel frequency (i.e. 1548.5 nm, 1549.5 nm, 1550.5 nm and 1551.5 nm respectively) and has a bandwidth of 37.5 GHz. The larger bandwidth makes these filters less ideal than the optical filters in the ADF. After the optical filtering, a photodiode converts the signal into the electrical domain. The electrical signal is again filtered, and then the BER is evaluated. The BER values recovered by each monitoring equipment during normal functioning are listed in Table 6.1. The smaller value measured

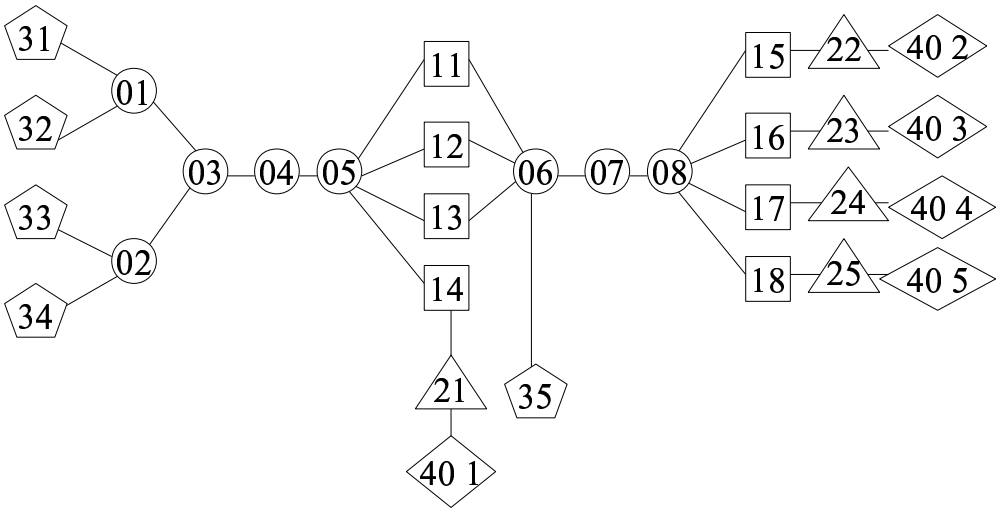


Figure 6.16: Model of the WDM network considered in the simulator

for Channel3 is due to the fact that it is dropped and added in the ADF, so that it is modeled by two different channels: one from the first transmitter to the receiver in the ADF and the second channel from the transmitter in the ADF to the end receiver.
Figure 6.16 shows the modeling of the system when 4 channels are established. The correspondence between the network components and the identifiers is listed in Table 6.2.

Network Equipment	AFA Identifier	FLA Identifier
Channel1 Transmitter	(3 1 0 0)	(3 1)
Channel2 Transmitter	(3 2 0 0)	(3 2)
Channel3 Transmitter	(3 3 0 0)	(3 3)
Channel4 Transmitter	(3 4 0 0)	(3 4)
Coupler Channel1+Channel2	(0 1 0 0)	(0 1)
Coupler Channel3+Channel4	(0 2 0 0)	(0 2)
Coupler (0 1)+(0 2)	(0 3 0 0)	(0 3)
Optical Fiber first optical link	(0 4 0 0)	(0 4)
Demultiplexer after (0 4)	(0 5 0 0)	(0 5)
Channel1 Filter	(1 1 0 0)	(1 1)
Channel2 Filter	(1 2 0 0)	(1 2)
Channel3 Filter	(1 3 0 0)	(1 3)
Channel4 Filter	(1 4 0 0)	(1 4)
Channel3 Receiver at ADF	(2 1 0 0)	(2 1)
BER estimator at ADF	-	(40 1)
Channel3 Transmitter at ADF	(3 5 0 0)	(3 5)
All channels Coupler	(0 6 0 0)	(0 6)
Optical Fiber second optical link	(0 7 0 0)	(0 7)
Demultiplexer after (0 7)	(0 8 0 0)	(0 8)
Channel1 Filter	(1 5 0 0)	(1 5)
Channel2 Filter	(1 6 0 0)	(1 6)
Channel3 Filter	(1 7 0 0)	(1 7)
Channel4 Filter	(1 8 0 0)	(1 8)
Channel1 Receiver	(2 2 0 0)	(2 2)
Channel2 Receiver	(2 3 0 0)	(2 3)
Channel3 Receiver	(2 4 0 0)	(2 4)
Channel4 Receiver	(2 5 0 0)	(2 5)
BER estimator Channel1	-	(40 2)
BER estimator Channel2	-	(40 3)
BER estimator Channel3	-	(40 4)
BER estimator Channel4	-	(40 5)

Table 6.2: Correspondence between the network components and the used identifiers in Figure 6.16.

Run	Wavelength (nm)
1	1549.5
2	1549.52
3	1549.54
4	1549.56
5	1549.58

Table 6.3: Wavelengths transmitted by the laser of Channel2

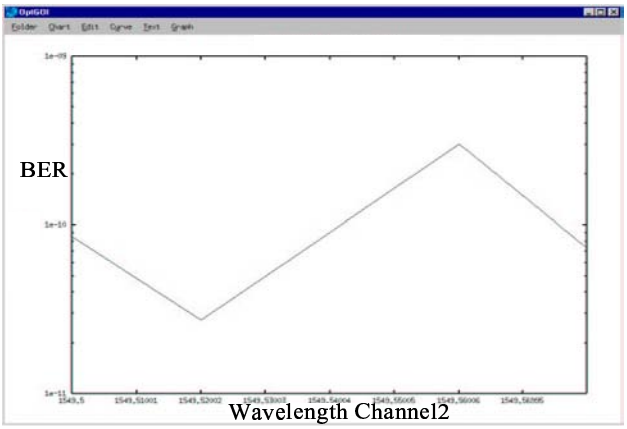
Several failure scenarios have been simulated:

1. Progressive failure: The emitted wavelength λ_2 of the second transmitter (represented by `(3 2)`) has been shifted from 1549.5 nm to 1549.58 nm in steps of 0.02 nm as listed in Table 6.3.
   In this scenario, the BER curves at each of the monitoring equipment changes as shown in Figure 6.17. As mentioned before, the BER is an estimated value and its exact value may be far from the real value, but the evolution can indicate an improvement or a degradation of the monitored signal.
   Where should the BER threshold be set? It has to be as low as possible to detect a BER degradation as soon as possible but it has to be high enough to avoid the variations of the BER calculations that are not a sign of a signal degradation. Figure 6.17 shows that the usual variations of the BER are under 10^{-9} hence the threshold can be set to this value.

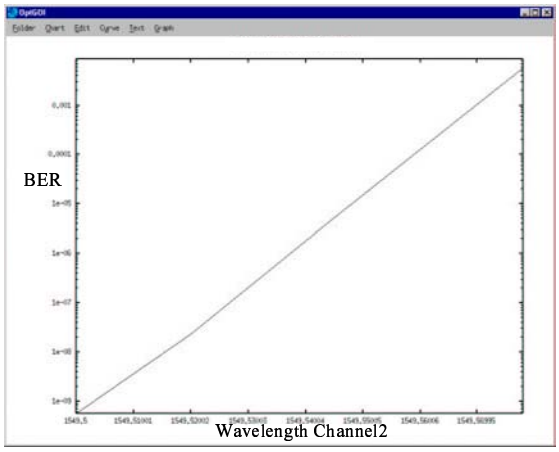
   - Under the assumption that the BER threshold is set to 10^{-9}, the alarm that the manager would receive is issued by the monitoring equipment associated to Channel2. In this case, the result of FLA, shown in Figure 6.18, gives as candidate elements the network components only related with Channel2, which are: the transmitter `(3 2)`, the first optical filter `(1 2)`, the second optical filter `(1 6)` and the receiver `(2 3)` (all of them having a progressive failure).
   - If the BER threshold is set to 4 10^{-9} then the element `(40 5)` will also send an alarm which will be considered as false. In this case, the result of FLA comprises
     - a simultaneous failure of at least one network element of Channel2 with at least one network element of Channel4
     - the failure of one network element of Channel2 (`(3 2),(1 2),(1,6),(2 3)`) with one false alarm,
     - the failure of one network element of Channel4 (`(3 4),(1 3),(1,8),(2 5)`) with one false alarm,

     as shown in Figure 6.19.

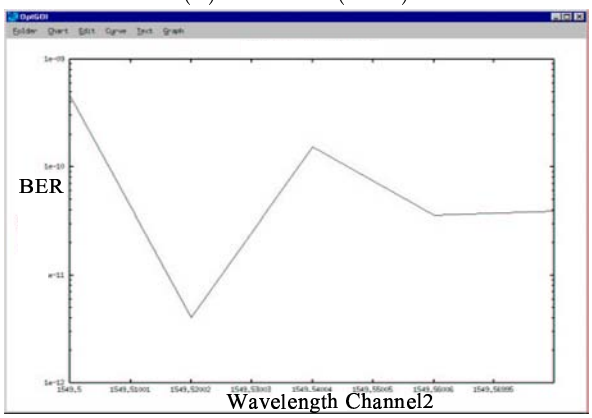
2. Hard failure: In this scenario, the first optical link is faced with a drastic increase of the attenuation (from 0,2 dB/km to 20 dB/km) so that the quality of the signals drops substantially. For example, the output of the second link is shown in Figure 6.20. The BERs of each channel have changed, as listed in Table 6.4.
   The value of `(40 4)` is within normal margins, but the other monitoring equipment will send an alarm due to the high BER. Therefore, the alarms that will be sent to the manager are issued by `(40 1)(2 1)(40 2)(40 3)(2 2)(2 3)(40 5)(2 5)`.



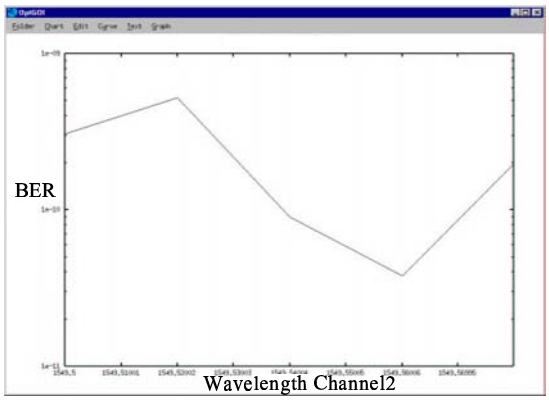
(a) BER at (40 2)



(b) BER at (40 3)



(c) BER at (40 4)



(d) BER at (40 5)

Figure 6.17: Estimated BER at different monitoring equipment when λ_2 changes

- The results given by FLA are shown in Figures 6.21 and 6.22. The results confirm our simulation thus presenting the optical fiber as one of the two candidates.
- The alarms that are input to the AFA algorithm are issued by `(2 1)(2 2)(2 3)(2 5)`. Taking into account that the AFA implementation considers four integers as network component identifiers (therefore the element `(2 3)` becomes `(2 3 0 0)`), the result of this failure scenario is shown in Figure 6.23 and also considers the optical link `(0 4 0 0)` as fault candidate.

In this failure scenario, the existence of monitoring equipment has not helped to better locate the failure. If monitoring equipment could have been installed between the optical fiber and the receiver, the failure would have been better located.

6.4.2 Star Local Area Network

The second network simulated using `OptSim` is an optical Local Area Network (LAN) with a star topology that was used in the project "Ultra High Capacity FDM Optical Local Area Network" [53], [54] sponsored by the National Swiss Fund.

Figure 6.18: FLA result when there is a single alarm from (40 3)

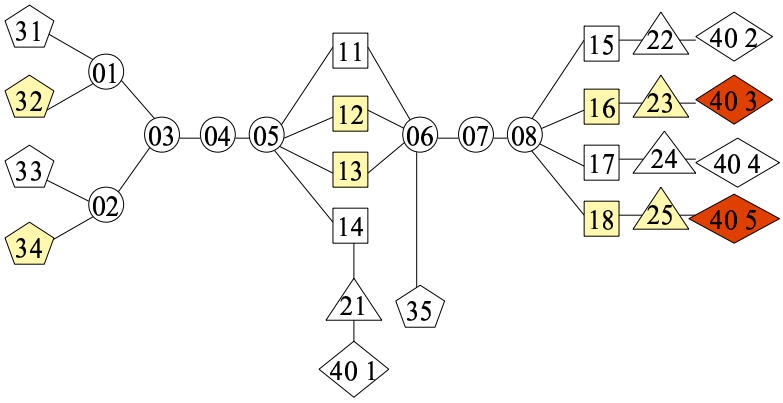


Figure 6.19: Alarms and result for a progressive failure scenario: the darkest elements have issued the alarms and the grey elements are the fault candidates

This network consists in a central coupler that interconnects two nodes in a star topology. Node1 transmits at 140 Mbps in λ_1 and Node2 transmits at 140 Mbps in λ_2. A local oscillator at 227.45 THz has been used to be able to recover the channels with a channel spacing very narrow (Channel1 at 227.451 THz and Channel2 at 227.45156 THz what corresponds to λ_1=1318.05 nm and λ_2=1318.05 nm respectively). Both nodes receive both wavelengths so that they can choose which channel they want to retrieve. In our case, both nodes retrieve both channels. A band pass filter (BPF) with a narrow bandwidth is used to retrieve the desired frequency. Then, an envelope detector is used to recover the signal before delivering it to the monitoring equipment. The optical fibers that connect the transmitter and the coupler have been considered as ideal, whereas the optical fibers between coupler and receivers have been considered as non-ideal (with a length of 50 km and a loss of 0.2 dB/km). The view of the simulated network in the simulator is shown in Figure 6.24 and the scheme of this LAN is shown in Figure 6.25.

Monitoring Equipment	BER
(40 1)	0.061
(40 2)	0.0666
(40 3)	0.0034
(40 4)	$5.72\ 10^{-12}$
(40 5)	0.0054

Table 6.4: BER at different monitoring equipment after hard failure in (0 4)

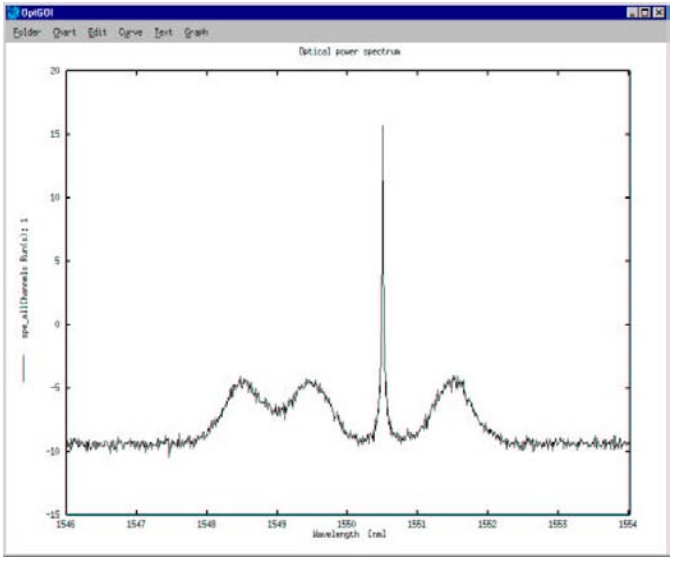


Figure 6.20: Output of the second optical link when the attenuation is drastically increased for all the channels except the third one that has been included after the failure. The x-axis represents the frequencies and the y-axis, the power.

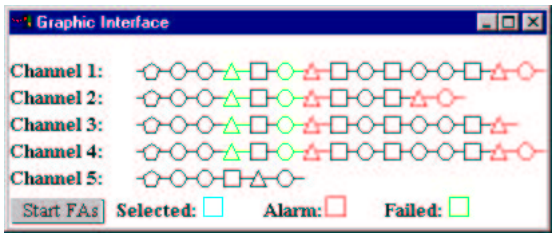


Figure 6.21: Graphical interface of the result in failure scenario 3

The BER values retrieved during the normal behaviour of the network are listed in Table 6.5 and an example of the eye diagram in one of them is displayed in Figure 6.26. Several failure scenarios have been simulated:

1. Hard failure: If the optical fiber OF1 that transmits λ_1 breaks, the estimated BER at each of the monitoring equipment will change as listed in Table 6.6. The BER of Channel2 has improved due to the absence of interference from Channel1. Therefore, the alarms that the manager will receive are issued by the monitoring equipment related with Channel1 which are `(40 1)` and `(40 3)` and the associated receivers `(2 1)` and `(2 3)`.

Monitoring Equipment	BER
Node1-Channel1	$7.14\ 10^{-15}$
Node1-Channel2	$1.05\ 10^{-13}$
Node2-Channel1	$6.44\ 10^{-15}$
Node2-Channel2	$2.96\ 10^{-13}$

Table 6.5: BER at different monitoring equipment during normal network functioning

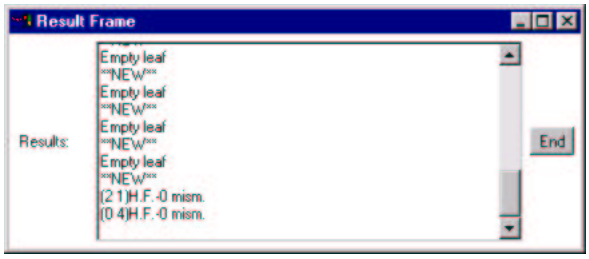


Figure 6.22: FLA Result in failure scenario 2

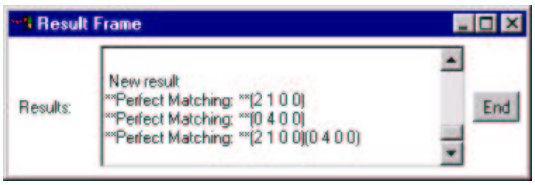


Figure 6.23: AFA result in failure scenario 2

- The AFA algorithm has as inputs the alarms coming from the receivers. The identifiers of the elements consist on 4 integers as explained in Chapter 4. The AFA result in this fault scenario, shown in Figure 6.27, proposes the element `(0 1 0 0)` as fault candidate, which corresponds to the fiber OF1.
- The FLA result is shown in Figure 6.28. FLA gives as fault candidate the element `(0 1)` which corresponds to the fiber OF1.

Both algorithms point to the same fault candidate when having a hard failure. The result is the expected one and both algorithms locate correctly the failure.

2. Two simultaneous hard failures: The optical fiber OF1 that transmits the signal from Node1 to the central coupler is likely to be installed in the same ribbon than the optical fiber OF3 that transmits the signals from the coupler to Node1. Let us simulate the cut of this ribbon, which corresponds to a double failure because each fiber is considered as a single network element. The estimated BERs at the monitoring equipment after this double failure are listed in Table 6.7. The alarms are therefore issued by `(40 1)`, `(40 2)`, `(2 1)`, `(2 2)`, `(2 3)` and `(40 3)`.

   - The AFA result after receiving these alarms is presented in Figure 6.29.
   - The FLA result also presents the elements `(0 1)(0 3)` as double failure.

   These results are shown in Figure 6.30

Monitoring Equipment	BER
Node1-Channel1	0.0387
Node1-Channel2	$3.584\ 10^{-28}$
Node2-Channel1	0.0394
Node2-Channel2	$2.286\ 10^{-28}$

Table 6.6: BER at different monitoring equipment when the optical fiber OF1 breaks

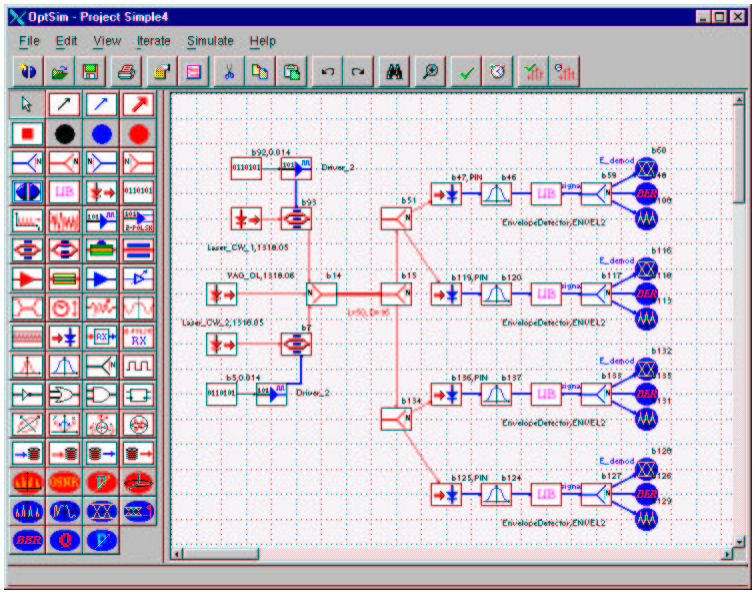


Figure 6.24: View of the star LAN network simulated with `OptSim`

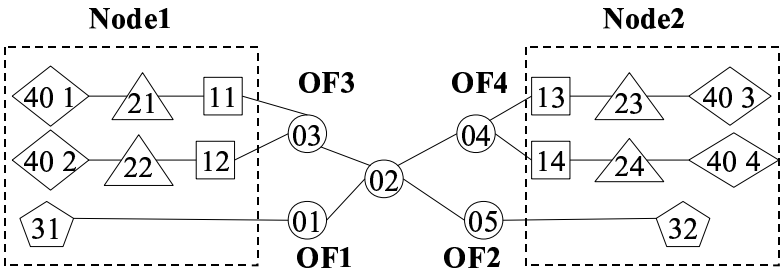


Figure 6.25: Scheme of the LAN with star topology

3. Progressive failure: Suppose now that the transmitter at Node1 has its frequency shifted from 227.451 THz to 227.476 THz as presented in Table 6.8. The variation of the estimated BER at the monitoring equipment of Node1 is presented in Figure 6.31. The estimated BER of Channel1 increases, whereas the estimated BER of Channel2 decreases due to the less interference caused by Channel1. The variation of the BER estimation at Node2 is very similar to the estimation done at Node2. The alarms received in this failure scenario are issued by `(40 1)` and `(40 3)`. In this failure scenario, the FLA proposes the following fault candidates: `(3 1)` and `(0 1)`. If monitoring equipment was installed between the transmitter `(3 1)` and the optical fiber `(0 1)`, the failure would have been better located (only one of these

Monitoring Equipment	BER
Node1-Channel1	0.0340
Node1-Channel2	0.0458
Node2-Channel1	0.0358
Node2-Channel2	$1.749\ 10^{-28}$

Table 6.7: BER at different monitoring equipment when the ribbon to Node1 breaks

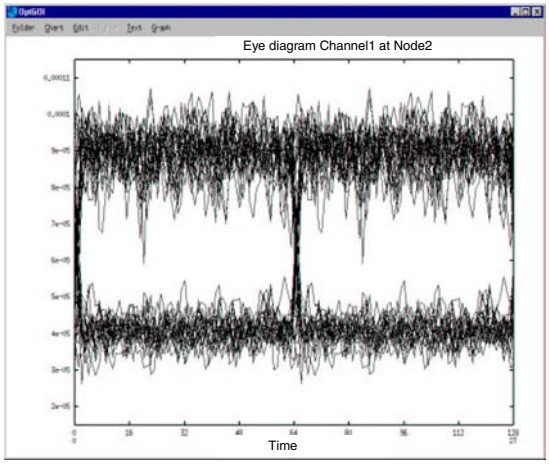


Figure 6.26: Eye diagram of Channel1 at Node2

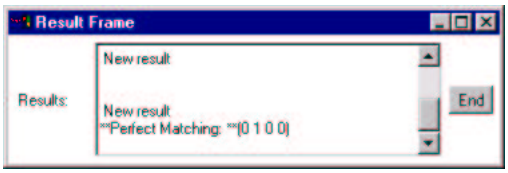


Figure 6.27: AFA result in scenario of a single hard failure in the star topology

two candidates would have been kept).

4. Progressive failure: Finally, suppose that the filter associated to Channel1 at Node1 has its central frequency shifted from 1 GHz to 1.3 GHz as presented in Table 6.9. The variation of the estimated BER at the monitoring equipment of Node1 is shown in Figure 6.32. The other estimated BER, which do not change with the simulated failure, are listed in Table 6.10. In this case, the only alarm that will be received is the one issued by the monitoring equipment of this channel, which is `(40 1)`. The result of FLA, shown in Figure 6.33, presents the filter and the receiver as fault candidates because any of them would send an alarm in case they fail.

6.4.3 Summary of the results

Two network topologies have been considered. The first one is a point-to-point WDM link, whereas the second one is a star topology. In each network several failure scenarios have been simulated. Some conclusions can be drawn from the simulations results:

Run	Frequency(THz)
1	227.451
2	227.456
3	227.461
4	227.466
5	227.471
6	227.476

Table 6.8: Variation of the transmitter frequency of Node1

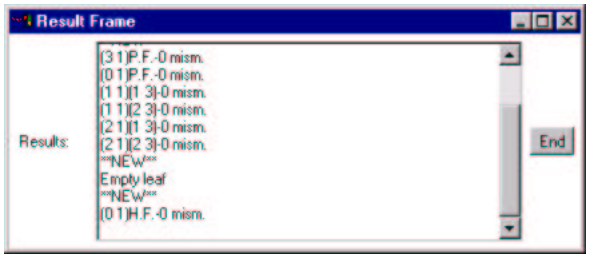


Figure 6.28: FLA result in scenario of a single hard failure in the star topology

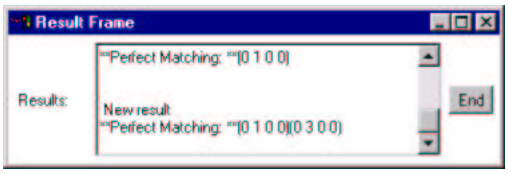


Figure 6.29: FLA result in the scenario of a double failure in star topology. In this window, there are two results: the first one, which is on the top of the window is an old result when only part of the alarms where received, whereas the second one is the last, definitive result delivered by the algorithm giving, for the full set of alarms received at that time, the fault candidates.

- When a progressive failure was simulated, the results were given by the FLA algorithm because it is the only algorithm that considers the information delivered by the monitoring equipment. We have shown that the accuracy of the failures location depends on the location of the monitoring equipment.
- When hard failures were simulated, the results delivered by each of both algorithms were compared, and did coincide.
- The algorithm results for each of the failure scenarios were the expected results because they always included the network component that was simulated to be faulty, and included only the minimal set of faulty candidates that could be retained given the alarms.
- The considered networks have still a small size but they are networks that have actually been implemented in experimental projects. Future trends point towards larger and all-optical networks so that the location of failures gets more complex due to the larger number of network components and the longer range failure propagation effects. Concerning the failure propagation, the AFA algorithm is well suited because by performing alarm discarding reduces the number of alarms to be treated.

Run	Central Frequency
1	1 GHz
2	1.1 GHz
3	1.2 GHz
4	1.3 GHz

Table 6.9: Variation of the central frequency of the filter to retrieve Channel1 at Node1

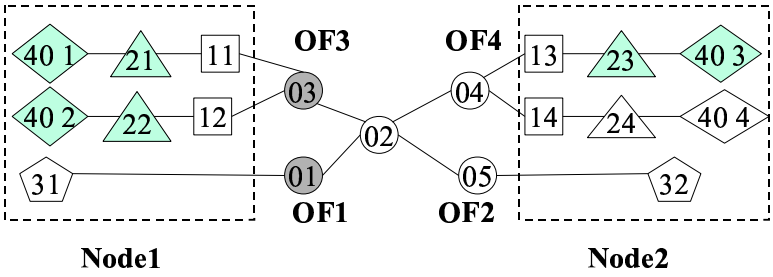
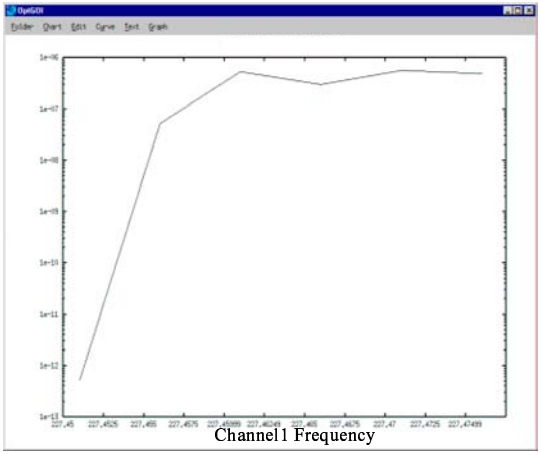
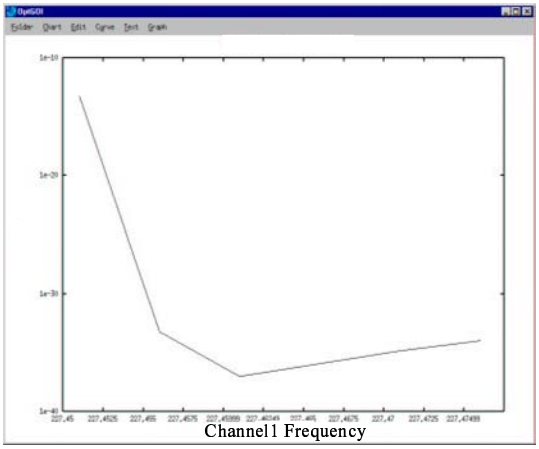


Figure 6.30: Alarms and result for a two simultaneous failure scenario: the grey elements have issued the alarms and the darkest elements are the fault candidates



(a) BER at Node 1 Channel1



(b) BER at Node 1 Channel2

Figure 6.31: Estimated BER variation when a progressive failure occurs

The FLA algorithm does not perform alarm discarding but it will present the fault candidates very fast. Concerning the larger number of network components, in order to check and verify whether the algorithms are suited for big, complex networks, a study of their complexity has been done in the next chapter.

6.5 Conclusion

In this chapter we have began introducing the most important features of the implementation of the proposed algorithms that perform fault location. Because we did not have the opportunity to test the algorithms on a real optical network, an optical network simulator called `OptSim` was used to emulate the physical layer. The algorithms were tested on two different network topologies and with different network scenarios. When hard failures

Monitoring Equipment	BER
Channel2 Node1	$4\ 10^{-12}$
Channel1 Node2	$2.21\ 10^{-12}$
Channel2 Node2	$4.08\ 10^{-12}$

Table 6.10: Estimated BER values during the shift of the filter of Channel1 at Node1

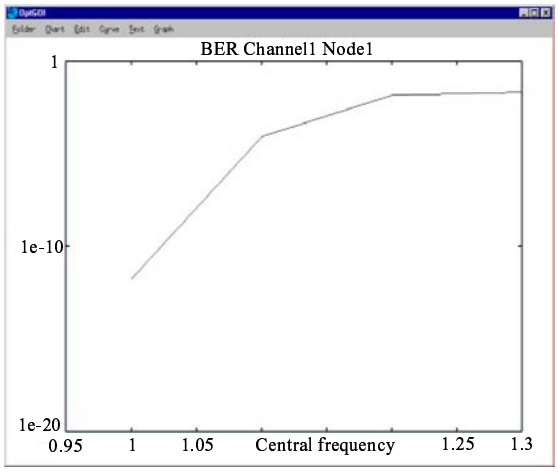
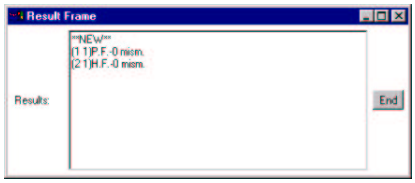
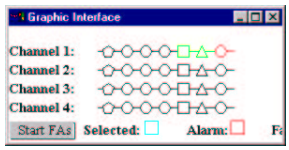


Figure 6.32: Estimated BER variation when the filter that retrieves Channel1 at Node1 ages



(a) Faulty Candidates



(b) Graphical User Interface

Figure 6.33: Result given by FLA algorithm

occurred, both algorithms delivered the same fault candidates. The FLA algorithm was also able to give fault candidates when progressive failures occurred.

Chapter 7

Comparison of the algorithms

7.1 Introduction

This chapter is devoted to the study of the worst case complexity of both algorithms. The algorithms are then compared to know the settings in which each of them performs the best. For this comparison, we do not take into account that FLA is able to locate progressive failures because of two reasons:(i) it does not change the complexity of the other modules and, (ii) the AFA algorithm could also include this point.
The problem of locating multiple failures has been shown by Rao [47] to be NP-complete. After establishing the NP-completeness of the problem in Section 7.2, the study of the complexity of the AFA and FLA algorithms is given in Section 7.3.
The main difference between both algorithms is the core of their procedures: the AFA algorithm combines two phases *backward* and *forward* to locate multiple failures, allowing a certain number of lost and false alarms; whereas the FLA algorithm is based on coding each of the domains into binary vectors and building a binary tree to locate multiple failures, allowing also a certain number of lost and false alarms.
It is important to underline here that a simple version of the fault location problem can be solved in the ideal case in a polynomial time by both algorithms. This problem is the determination of a subset of elements belonging to the established channels $\mathcal{CH}$, which include all the elements whose failure explain the received alarms $\mathcal{R}$, but do not include any element that can be certified not to have failed, because the associated alarms are not present in the received alarms $\mathcal{R}$.
Let us consider the example of two established channels as shown in Figure 7.1. Suppose that the received alarms are issued by c and g, so that $\vec{y} = Bin(\mathcal{R}) = (1\,0\,1)$. Then, the

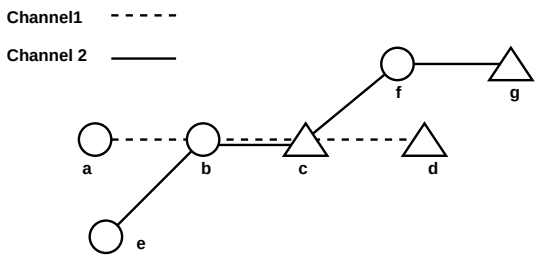


Figure 7.1: Two established channels with two kinds of network components.

solution of this subproblem is the set of elements $\{e, f\}$, because we know for sure that all

failed elements are included in this list and that neither a nor b have failed (if we assume that all alarms are correctly received) because no alarm was received from d.
Running the AFA on this example, the result of the backward phase is $\mathcal{PC} = \mathcal{PC}_2 = \{a, b, e\}$, because they are the elements directly related with the non discarded alarm issued by c. The AFA needs the forward phase to deliver a smaller set of fault candidates, which in this case is $\{e\}$ for the ideal case.
The FLA can provide a similar result without requiring the computation of all codewords associated with multiple failures, but using instead Theorem 1 of Section 5.6. Applying this theorem to the example of Figure 7.1 with $\vec{y} = (1\ 0\ 1)$ will indeed yield that $\alpha_1 = 0$, $\alpha_2 = 0$, $\alpha_3 = 0$, $\alpha_4 = 1$, $\alpha_5 = 1$ and therefore giving the set $\{e, f\}$ as solution.
The obtaining of these non-minimal sets can be shown to have a polynomial complexity, but unfortunately this is not the solution to our problem. What we are looking for is the *minimal* subset of elements that explain all the received alarms. In the example above, the desired solution is $\{e\}$. To recover this minimal subset is the point where the non polynomial complexity comes up.
Section 7.4 gives a comparison between the algorithms. The chapter will end summarizing the most important conclusions.

7.2 NP-Completeness of the multiple failure problem

Let us recall that $\mathcal{V}$ is the set of network components that may fail and that $\mathcal{R}_{orig}$ is the set of network elements that have indeed issued alarms as defined in Sections 4.3.1 and 4.4.3 respectively. Let us denote by $\mathcal{AL}$ the set of network elements able to issue alarms. The cardinality of $\mathcal{V}$ is $|\mathcal{V}| = n$, the cardinality of $\mathcal{AL}$ is $|\mathcal{AL}| = n_a$ and the cardinality of $\mathcal{R}_{orig}$ is $|\mathcal{R}_{orig}| = k$. The problem of locating multiple failures in the ideal scenario where $m = 0$ can be defined as follows: Does there exist a subset of $\mathcal{V}$ with a number s of elements as small as possible, that precisely causes all the received alarms $\mathcal{R}$? Mathematically, the problem can be reformulated as follows. What is the smallest integer s, $1 \leq s \leq n$, such that there is a set $\mathcal{V}_s \subseteq \mathcal{V}$ with $\mid \mathcal{V}_s \mid = s$ such that

$$(\cup_{e \in \mathcal{V}_s} Domain(e)) \cap \mathcal{R}_{orig} = \mathcal{R}_{orig} \tag{7.1}$$

and

$$Domain(e) \cap (\mathcal{A} - \mathcal{R}_{orig}) = \emptyset \quad \forall e \in \mathcal{V}_s \quad ? \tag{7.2}$$

If such a set $\mathcal{V}_s$ exists, then s simultaneous failures have occurred.
This problem is NP-complete because, for the special case where $\mathcal{AL} = \mathcal{R}_{orig}$ (that is, when all the alarms are issued), it can be polynomially reduced to the Minimum Cover (MC) problem, which is known to be NP-complete [55]. MC can be defined as follows: Given a collection $\mathcal{B}$ of subsets of a finite set $\mathcal{K}$, what is the minimal cardinality s of a subset $\mathcal{B}_s \subset \mathcal{B}$ such that every element of $\mathcal{K}$ belongs to at least one member of $\mathcal{B}_s$, if the solution exists?
Let us present the mapping between both problems with an example. Let $\mathcal{K}$ be a set of 6 elements: $\mathcal{K} = \{b, c, g, j, l, m\}$ and $\mathcal{B}$ be set of 9 subsets made of elements of $\mathcal{K}$: $\mathcal{B} = \{\{b, c, g\}, \{c, g\}, \{g, j\}, \{g, j, m\}, \{g\}, \{c, j\}, \{j\}, \{l, m\}, \{m\}\}$. Let us construct a two level graph connecting $\mathcal{K}$ in the first level and $\mathcal{B}$ in the second one, as shown in Figure 7.2.

The two levels are connected with lines that link an element belonging to $\mathcal{K}$ with the

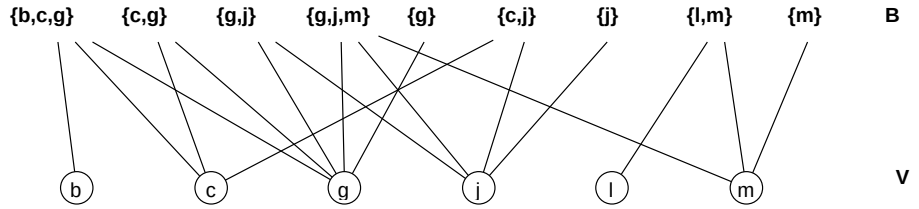


Figure 7.2: Two level graph to map the MC problem to the multiple failures problem when $\mathcal{AL} = \mathcal{R}_{orig}$

subsets of $\mathcal{B}$ containing it. Following the example given in Figure 7.2, the element c has one link to each of the four subsets of $\mathcal{B}$ that contain c which are: $\{b, c, g\}, \{c, g\}$ and $\{c, j\}$. In our example, with $s = 1$ and with $s = 2$, it is not possible to find $\mathcal{B}_s$. The solution of the problem is $s = 3$ because with three of the subsets we can cover the set $\mathcal{K}$. For example $\mathcal{B}_s$ can be $\{\{b, c, g\}, \{g, j, m\}, \{l, m\}\}$.
In order to map the multiple failures problem with the MC problem, we have to consider the following mapping: the set $\mathcal{K}$ of the MC problem corresponds to the set of network components able to issue alarms $\mathcal{AL}$, and the set $\mathcal{B}$ corresponds to the set of Domains of all the network elements $\mathcal{V}$. For example, let us consider the three channels shown in Figure 7.3. In this case, $\mathcal{V} = \{a, b, c, d, e, f, g, h, i, j, k, l, m\}$ and their domains are:

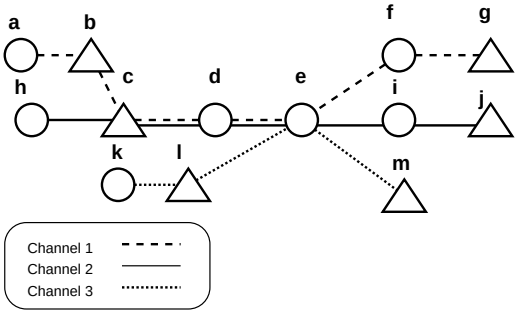


Figure 7.3: Two established channels with two kinds of network components.

$Domain(a) = \{b, c, g\} = C_1$
$Domain(b) = \{c, g\} = C_2$
$Domain(c) = \{g, j\} = C_3$
$Domain(d) = \{g, j\} = C_3$
$Domain(e) = \{g, j, m\} = C_4$
$Domain(f) = \{g\} = C_5$
$Domain(h) = \{c, j\} = C_6$
$Domain(i) = \{j\} = C_7$
$Domain(k) = \{l, m\} = C_8$
$Domain(l) = \{m\} = C_9$
The two levels mapping in this example is the same as the one in Figure 7.2. In this example, the solution is found for $s = 3$ and the set $\mathcal{B}_s$ can be $\{\{b, c, g\}, \{g, j, m\}, \{l, m\}\}$ which is associated to the fault candidates a, e and k. A triple failure of these three elements will indeed prompt all possible alarms in the network.

7.3 Complexity

This section presents the study of the complexity of both algorithms, as a function of the following parameters:

- n, which is the number of network components in the network: $|\mathcal{V}| = n$.
- l, which is the number of established channels: $|\mathcal{CH}| = l$.
- k, which is the number of received alarms: $|\mathcal{R}|$. k is also the cardinality of the set $|\mathcal{R}_{orig}| = k$.
- n_a, which is the number of alarming elements:$|\mathcal{AL}| = n_a$
- t, which is the number of different domains, or equivalence classes C_i, associated to single failures.

First we will present the complexity of the AFA algorithm, followed by the complexity of the FLA algorithm.

7.3.1 AFA Complexity

The AFA algorithm proposed in Chapter 4 combines two different approaches: *backward* and *forward* approach.
Given the received alarms $\mathcal{R}$, the *backward* phase finds the set of fault candidates, which is the union of the sets $\mathcal{PC}_1$ and $\mathcal{PC}_2$. This approach corresponds to the first part mentioned in Section 7.1 and has a polynomial complexity.
The *forward* phase finds, for each component of the established channels, the set of components that will send an alarm in case this component fails. This set was defined as *Domain* of a component.
Combining both phases, the algorithm is able to cope with false and/or lost alarms and to identify multiple failures as we have seen in Chapter 4. The AFA returns the list of components which may have failed and caused the received alarms, if one tolerates a given number of lost and false alarms, given by the mismatching threshold.
In this section we will present the complexity of each module of the algorithm.

Backward Approach

The *backward* approach consists in three different modules: *Alarm_Discarding_1, Alarm_Discarding_2* and *Candidate Search.*

Alarm_Discarding_1 This module produces two subsets from the set of the received alarms $\mathcal{R}$: $\mathcal{PC}_1$ and $\mathcal{T}$.

- $\mathcal{PC}_1$ (Possible_Candidates_1) is a subset of $\mathcal{R}_{orig}$ that contains the *A1* and/or *A3* components that belong to at least one established channel.
- $\mathcal{T}$ is the subset of $\mathcal{R}$ sent by *A2* components belonging to at least one established channel

The complexity of this module is $O(nk)$ because for each of the k received alarms, there is a check to decide whether its origin is a component participating in at least one of the established channels.

Alarm_Discarding_2 and Candidate Search These two modules can be implemented within one single procedure as shown in Appendix C. Given the set of alarms $\mathcal{T}$, the procedure discards redundant alarms as explained in Section 4.4.2 and returns the set $\mathcal{PC}_2$ which contains the network components directly related with the non discarded alarms. The complexity of this module depends on the alarms that belong to $\mathcal{T}$, whose number is at most k. Given the origin of each alarm of $\mathcal{T}$ and the channels to which belongs (which is at most l), one checks the elements located before it on these channels (which are at most n) to know first, whether their alarms have to be discarded or not (this is the Alarm_Discarding_2 module which is illustrated in Figure 7.4), and if it is not discarded, to determine the fault candidates to add to $\mathcal{PC}_2$ (this is the Candidate Search module). Note that in this particular figure, the circular elements represent network elements of all categories. The complexity of this module is therefore $O(nkl)$.

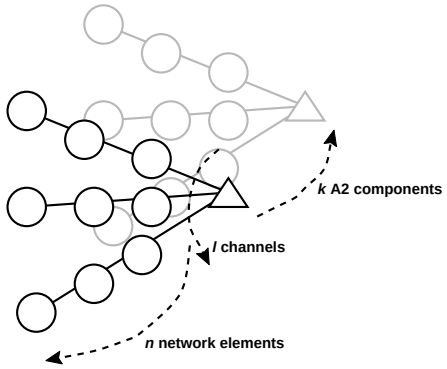


Figure 7.4: Alarm Discarding module: for each network element that issued an alarm, and for each of the channels where the element belongs to, there is a *backward* search to find if there are immediate alarms or to add the immediate elements into $\mathcal{PC}_2$

Forward Approach

This approach consists only of one module called *Domain_Calc*, which computes the *Domain* of each component belonging to at least one established channel and stores it in the list of domains $\mathcal{D}$.
The main core of this module is presented in Appendix C. For each element of $\mathcal{V}$ (which are at most n), and for each channel to which belongs (which is at most l), there is a check of the $A2$ components that follow it on the channel, to know whether they should be added in the domain. Since there are at most n components that belong to the l established channels (as shown in Figure 7.5, where the circular elements represent network elements of all categories), the complexity of this module is $O(nl\ (n-1)) \approx O(n^2 l)$.

Candidate Selection

This last module combines the results from both *backward* and *forward* phases in order to find the sets of elements that explain the received alarms, allowing up to m lost and false alarms.
The inputs are the two sets $\mathcal{PC}_1$ and $\mathcal{PC}_2$ given by the *backward* phase and the set $\mathcal{D}$ given by the *forward* phase. The cardinality of $\mathcal{PC} = \mathcal{PC}_1 \cup \mathcal{PC}_1$ is at most k because we assume

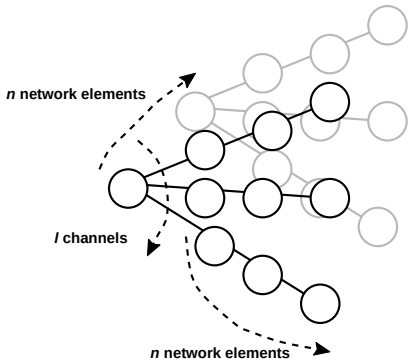


Figure 7.5: Domain methodology: for each network element e_i and for each channel to which e_i belongs, this module finds the elements that belong to the same channel as e_i, and that would send an alarm when e_i fails.

that when k alarms have been issued, there are at most k faulty elements. The new sets that this module constructs are made iteratively from the union of $\mathcal{PC}_1$ with subsets of $\mathcal{PC}_2$ as described in Section 4.4.3.
Let us study the complexity time of this module.

- The number of iterations are at most k because this is the maximal cardinality of $\mathcal{PC}$.
- The cardinality of $\mathcal{PC}_2$ is at worst n, which implies that for the ith iteration, there are C_n^i subsets of $\mathcal{NS}(i)$.
- The number of operations to perform the union of the domains $UnionDom(\mathcal{Q})$ of the elements of one subset $\mathcal{NS}$ at the i-iteration is at most $(k+i)n_a$ where $k+i$ is the number of elements of each subset of $\mathcal{NS}(i)$ and where n_a is the maximum number of elements in a Domain.
- The comparison of the union of domains $UnionDom(\mathcal{Q})$ with the set $\mathcal{R}_{orig}$ can be done in at most n_a since $|UnionDom(\mathcal{Q})| = n_a$ in the worst case.

The complexity of this module is therefore $O(\sum_{i=0}^{k} C_n^i i\, n_a^2)$. Note that $k \leq n_a$, so that a loose upper bound on the complexity of this module is $O(n_a^2\, n\, 2^{n-1})$.

Overall AFA complexity

Each of the two phases of the AFA algorithm can be done in parallel. For this reason, the overall complexity of the AFA algorithm is the maximum complexity between the *backward* and *forward* phase plus the complexity of the *Candidate Selection* module. After comparing the *backward* complexity which is $O(nk + nkl) = O(nkl)$ with the *forward* complexity which is $O(n^2 l)$ and taking into account that $k \leq n$, the dominant complexity comes from the *forward* phase. Nevertheless, the high complexity is centered in the last module *Candidate Selection* which performs the NP-completeness of the problem. The total complexity of AFA is given by $O(n^2 l + \sum_{i=0}^{k} C_n^i i\, n_a^2)$, which is, as expected, non-polynomial.

7.3.2 FLA Complexity

Let us recall that $\mid \mathcal{AL} \mid = n_a$ is the depth of the binary tree and hence the size of the binary vectors.
The FLA algorithm can be divided into two parts: one part, which is called *pre-computing phase*, is independent from the received alarms, whereas the second part, which is called *main core* is computed when alarms reach the manager.
Let us present the complexity of each of the modules of the FLA algorithm.

Pre-Computing Phase (PCP)

Computation of *Domains*

For each element of the established channels, a domain is computed applying the same methodology as the one used in AFA, which is presented in Figure 7.5. The complexity of this procedure is at most of $O(nl(n-1)) \approx O(n^2 l)$.

Grouping of *Domains* into equivalent classes C_i and computation of the codewords $\vec{g_i} = Bin(C_i)$

Let us present step by step, the complexity of this module:

- The domains having the same elements are associated to one set C_i. There are at most n different domains and therefore n different sets.
- Having n different sets, there is a check of $C_n^2 = n(n-1)/2$ pairs of sets to group identical sets in the same equivalence class.
- Each of these comparisons is done on n_a elements.

The complexity of this part is therefore $O(n(n-1)n_a/2)$. Then, for each class C_i ones computes the n_a-dimensional binary vector $\vec{g_i} = Bin(C_i)$ which is an operation of $O(n_a)$.
The complexity of this module that groups identical domains and computes the codeword $\vec{g_i} = Bin(C_i)$ is therefore at most of $O(n^2 n_a^2/2)$.
So far, we have obtained the codewords $\vec{g_i}$ that correspond to single failures. The cardinality of the set containing these codewords is denoted by t. Vectors corresponding to multiple failures can be found by computing the point-wise OR operation between each pair of codewords of $\mathcal{C}$. The codewords of the single failures are the generator vectors of the moduloid $\mathcal{C}$ which includes single and multiple failures.

Computation of the domains an codewords of multiple failures

There is no easy way to evaluate the total number of codewords of the non linear code $\mathcal{C}$ described in Section 5.6.

- A first method is to count the number of codewords generated by successively 'OR'ing the different generator vectors $\vec{g_i}$. At iteration 1, the codewords are the t initial generator vectors $\vec{g_1}, ..., \vec{g_t}$. At iteration 2, one computes the codewords $\vec{g_i} \vee \vec{g_j}$, $1 \leq i \neq j \leq t$, associated with double failures, and we proceed likewise up to a maximum of t iterations. Each iteration b, $1 \leq b \leq t$, brings at most C_t^b new different codewords. The total number of codewords is therefore at most $\sum_{b=1}^{t} C_t^b = 2^t - 1 \simeq 2^t$.
- Another reasoning is that there are at most 2^{n_a} different binary n_a-dimensional vectors $\vec{x} \in \mathcal{C}$.

- Combining these two points, we get a bound on the number of codewords, which is $2^{min(n_a,t)}$.
- We compare the n_a components of each pair of codewords to eliminate all the identical ones. This operation involves at most $2^{min(n_a,t)}(2^{min(n_a,t)}-1)/2 \simeq 4^{min(n_a,t)}$ comparisons.

Consequently, the complexity of this module is at most $O(4^{min(n_a,t)}n_a) \leq O(4^{n_a}n_a)$ but the bound can be loose if t is much smaller than n_a.

Computation of codewords with mismatching thresholds m_1 and m_2

For each codeword $\vec{x_i} \in \mathcal{C}$, we find all the vectors $\vec{z}$ such that $d_1(\vec{x_i},\vec{z}) \leq m_1$ and $d_2(\vec{x_i},\vec{z}) \leq m_2$, where d_1 is the function which returns the number of positions where $\vec{x_i}$ has '1' and $\vec{z}$ has '0'; and d_2 is the function which returns the number of positions where $\vec{x_i}$ has '0' and $\vec{z}$ has '1'. Then, the sets already pointed by leaf $\vec{z}$, if any, are added to $P(C_i)$. Let us denote by r_i the number of zeros in $\vec{x_i}$. The complexity depends on the number of vectors $\vec{z}$ for each codeword $\vec{x_i} \in \mathcal{C}$. These vectors have up to m_2 '1' at the positions where $\vec{x_i}$ has '0' (there are r_i such positions available) and have up to m_1 '0' at the positions where $\vec{x_i}$ has '1' (there are $n_a - r_i$ such positions available). The number of such vectors is therefore $(\sum_{i=0}^{m_2} C_{r_i}^i)(\sum_{j=0}^{m_1} C_{n_a-r_i}^j)$.
As an example, let us consider the vector $\vec{x_i} = (0\,1\,0\,0\,1\,0)$, and suppose that $m_1 = 1$ and $m_2 = 2$. In this case, the vectors $\vec{z}$ that verify $d_2(\vec{x_i},\vec{z}) \leq m_2 = 2$ are:

- the vector $\vec{x_i}$ itself,
- the $C_4^1 = 4$ vectors obtained by changing a '0' of $\vec{x_i}$ into a '1',
- the $C_4^2 = 6$ vectors obtained by changing a pair of '0' of $\vec{x_i}$ into a pair of '1',

i.e. a total of $C_4^0 + C_4^1 + C_4^2 = 11$ vectors.
Similarly, the vectors $\vec{z}$ that verify $d_1(\vec{x_i},\vec{z}) \leq m_1 = 1$ are:

- the vector $\vec{x_i}$ itself,
- the $C_2^1 = 2$ vectors obtained by changing a '1' of $\vec{x_i}$ into a '0',

i.e. a total of $C_2^0 + C_2^1 = 3$ vectors.
Now, the vectors that verify $d_1(\vec{x_i},\vec{z}) \leq m_1 = 1$ and $d_2(\vec{x_i},\vec{z}) \leq m_2 = 2$ are at most the product of the two previously numbers, i.e. 33.
The previous operations are repeated for all the vectors $\vec{x_i} \in \mathcal{C}$, whose number is at most $2^{min(n_a,t)}$, as we have seen above.
The worst case complexity is therefore upper bounded by

$$2^{min(n_a,t)} \sum_{i=0}^{m_2} C_{r_i}^i \sum_{j=0}^{m_1} C_{n_a-r_i}^j \leq 2^{min(n_a,t)} \sum_{i=0}^{r_i} C_{r_i}^i \sum_{j=0}^{n_a-r_i} C_{n_a-r_i}^j$$

because the maximal number of bits '0' and '1' that can be changed in a codeword $\vec{x_i}$ with r_i '0' and $n_a - r_i$ '1' are respectively $min(m_1, n_a - r_i)$ and $min(m_2, r_i)$. Therefore, the complexity of this module is $O(2^{min(n_a,t)}2^{r_i}2^{n_a-r_i}) = O(2^{n_a}2^{min(n_a,t)}) \leq O(4^{n_a})$.
Note that the inequalities in the previous paragraph, may give rather loose bounds for small values of m_1, m_2 and/or t.

n_a	t	k	Algorithm with lower complexity
20	50	80	FLA $n < 35$ AFA $n > 35$
50	50	80	FLA $n < 10$ AFA $n > 10$
80	50	80	AFA
50	50	20	FLA $n < 10$ AFA $n > 10$
50	20	80	FLA $n < 50$ AFA $n > 50$
50	35	80	FLA $n < 20$ AFA $n > 20$
50	80	80	AFA

Table 7.1: Algorithm with lower complexity

FLA Main Part

1. Compute $Bin(\mathcal{R})$: this module has a complexity of $O(n_a)$.
2. Find the $P(C_i)'s$ pointed by $Bin(\mathcal{R})$: the complexity of this module is $O(n_a)$, which is the time that is needed to go through the tree.

Overall FLA complexity

The overall complexity of the FLA algorithm when a new alarm reaches the manager is $O(n_a)$. This low complexity is due to the transfer of complexity towards the Pre-computing Phase (PCP) which has a complexity of $O(4^{min(n_a,t)}n_a)$. This algorithm is therefore well suited for networks with long term channels (networks whose channel events are spaced by long time intervals).

7.3.3 Complexity Comparison

This section gives the complexities of each algorithm in terms of the same parameters.

- AFA: The complexity of the polynomial part of AFA is n^2l, whereas the overall complexity is $O(n^2l + \sum_{i=0}^{k} C_n^i\, i\, n_a^2))$.
- FLA: The complexity of the polynomial part of FLA is $n_a^2 n$, whereas the overall complexity is $O(4^{min(n_a,t)}n_a) \leq 4^{n_a}n_a$

The comparison of the polynomial parts of both algorithms does not provide any useful information because the resulting sets are not equivalent and cannot be compared. Hence, we focus on the comparison of the overall complexity of both algorithms. We denote the complexity of the AFA algorithm with C_{AFA}, and the complexity of the FLA algorithm with C_{FLA}. In order to compare them, we have displayed in graphs, the logarithmic difference of the complexity, that is, $log(C_{AFA}) - log(C_{FLA})$. When the difference is positive, it means that the FLA has a lower complexity than the AFA, whereas if the difference is negative, the AFA has a lower complexity than the FLA.

We consider different scenarios by modifying the values of the number of 'alarming' components n_a, the number of generating codewords t and the number of received alarms k. The computational time at each scenario can be represented in a graph and the data obtained for these graphs is summarized in Table 7.1.

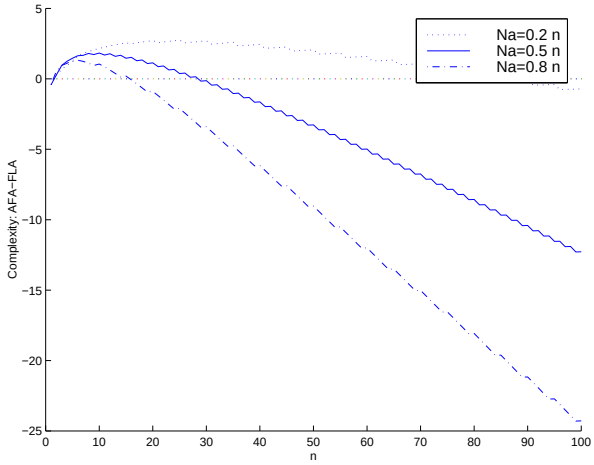


Figure 7.6: Logarithmic representation of the complexity of the AFA and FLA implementations when $k = 50\%$ of n_a and $t = 30\%$ of n_a.

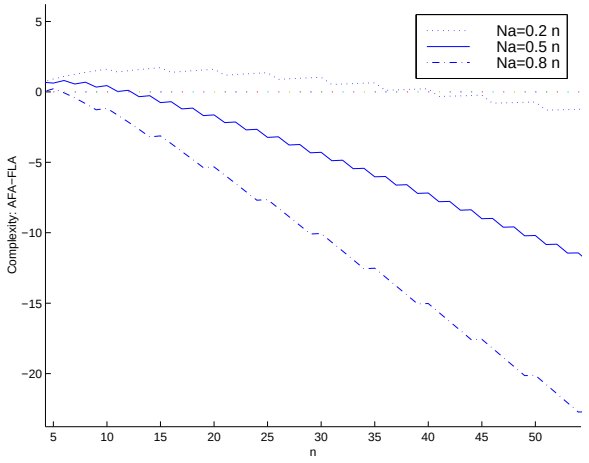


Figure 7.7: Logarithmic representation of the complexity of the AFA and FLA implementations when $k = 50\%$ of n_a and $t = 50\%$ of n_a.

Figure 7.6 illustrated the graph obtained for $k = 50\%$ of n_a and $t = 30\%$ of n_a, Figure 7.7 illustrated the graph obtained for $k = 50\%$ of n_a and $t = 50\%$ of n_a and Figure 7.8 illustrates the graph obtained for $k = 50\%$ of n_a and $t = 80\%$ of n_a.
As a result of the different tested scenarios, we give some conclusions about when each of the algorithms behaves better than the other:

- First, we changed the number of alarming components n_a. For a high percentage of n_a out of the total number of elements n, the AFA has a lower computational time than the FLA. This is expected since FLA depends directly on n_a because is the size of the binary tree. On the other hand, for a high number of non-alarming

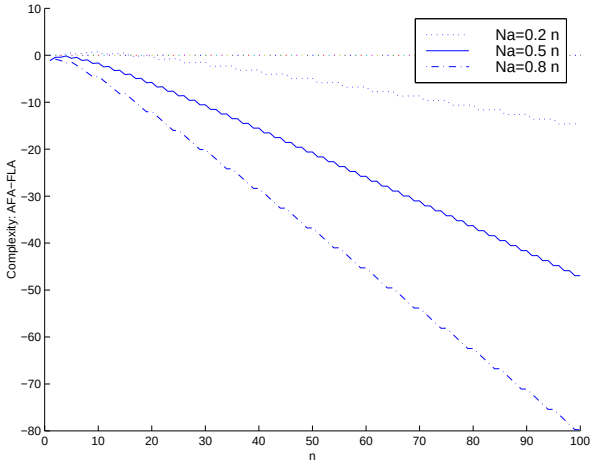


Figure 7.8: Logarithmic representation of the complexity of the AFA and FLA implementations when $k = 50\%$ of n_a and $t = 80\%$ of n_a.

components, the FLA has a lower computational time than the AFA for some values of n. In this scenario, the FLA has a small binary tree, which is easy to compute.

- Some tests were performed by changing the value of t in terms of n_a. For high values of t, (that is, when the number of different domains is high), the computational time of FLA is higher than the AFA. On the contrary, for low values of t, the FLA has a lower computational time, because the number of different domains is drastically reduces so that it 'plays' with few codewords.

- The last tests were based on modifying the k value in terms of n_a. The difference between the computational time of both algorithms does not change significantly.

7.4 Comparison of both algorithms

As a result of the complexity study of the algorithms, we give the scenarios about when each of the algorithms behaves better than the other:

- If the established channels change often, both algorithms behave similarly because both algorithms need to compute the domains of the new/old channels. Indeed, both algorithms are well suited for networks with long term connections. In particular, the FLA is the best suited algorithm due to the concentration of complexity in the so-called Pre-Computing Phase, which minimizes the computational time once alarms reach the manager. If the network has short term connections, the AFA should be used preferably because the FLA would remain computing the PCP part and never deliver a result.

- If the percentage of n_a ('alarming' components and of monitoring equipment) out of n (the total number of components) is high, and hence, if the percentage of 'non-alarming' components is low, the AFA has a lower complexity than the FLA algorithm. This fact is due to the absence of large vectors and the possibility to discard redundant alarms. On the contrary, if the number of 'non-alarming' components is high, the FLA algorithm has a lower complexity than AFA.

- The number of received alarms k does not influence the selection of the algorithm, which indeed will depend on the number of elements and its relation with the number of 'alarming' elements.

7.5 Conclusion

This chapter has illustrated the complexities of the AFA and FLA algorithms.
It shows the NP-completeness of the multiple location problem which justifies the non-polynomial complexity of the algorithms. The chapter also provides the complexity of each of the algorithms. It distinguishes two parts of each algorithm: the polynomial part that returns a non-minimal set of fault candidates in the ideal scenario and the non-polynomial part that gives the minimal set of faulty elements allowing a given number of lost and/or false alarms. A comparison of both algorithms is then possible and allows us to give the settings in which each algorithm behaves the best.

Chapter 8

Conclusion

In this dissertation we developed two algorithms to solve the multiple failure location problem in optical networks in presence of false and/or lost alarms.
Chapter 2 gave a general introduction on optical networks and their management. With the existing components to date, optical networks are not yet completely reconfigurable but that in a few years, fully reconfigurable optical networks are expected to appear. The management of networks, with a particular emphasis on fault management, was reviewed. The state-of-the-art of the techniques used for fault diagnosis concluded the chapter.
Chapter 3 described the most common components of an optical network, beginning with the hardware components of the physical layer and moving then to the components in other layers, such as WDM or SDH layers, which may provide more accurate information about failures.
Chapter 4 introduced the first algorithm to solve the fault location problem at the physical layer. This algorithm is called Alarm Filtering Algorithm (AFA). An abstraction of the fault location problem was done from the classification of the *hardware* components on the basis of their alarming properties when a *hard* failure occurs. This problem abstraction allowed us to develop the AFA algorithm. The AFA, based on the combination of two approaches, *forward* and *backward*, provides a list of fault candidates that explain the received alarms, while tolerating a given number of false/lost alarms. The *forward* approach finds the Domains of all the elements (i.e. the sets of elements that should issue alarms if the considered element fails), whereas the *backward* approach performs some discarding of the received alarms that can be considered as redundant. This algorithm, due to its structure, does not minimize the complexity when new alarms reach the manager.
This property was the objective for our second proposed algorithm presented in Chapter 5, called Fault Location Algorithm (FLA). It was introduced to shift the non-polynomial complexity from the diagnosis phase to a pre-computing phase which is carried out independently of the received alarms. This algorithm was initially designed to locate single and *hard* failures only. We extended this algorithm to locate not only *hard*, but also *soft* failures, given the information from the physical and some of the other layers. The classification of the components of the previous chapter was updated to include these *monitoring* equipment. This algorithm was also extended to multiple failures and to the non-ideal scenario with lost and false alarms.
Chapter 6 illustrated some simulation results of both algorithms. The algorithms were applied to different failure scenarios onto different network topologies. An optical system simulator `OptSim` © [2] was used to emulate the failures at the physical layer.

Chapter 7 studied the complexity aspects of both algorithms, to compare the settings in which each algorithm performed the best. We showed that if there are short term connections, both algorithms behave similarly, whereas if there are long term connections, the FLA has better behaviour than AFA because it delivers the result in shorter time. We also showed that if the percentage of alarming and monitoring equipment out of the total number of components is high, the AFA has lower complexity than FLA, whereas if the percentage is low, the FLA has lower complexity than AFA.
The proposed algorithms are based on a problem abstraction, which allows to apply them into other systems providing binary signals.

Future Work

Optical networks evolve rapidly and new optical components, both for transmission and monitoring, appear continuously on the market.

- One of the difficulties to locate failures and even to know that a failure has occurred, is to interpret the analog signals delivered both by monitoring equipment, such as BER or channels wavelengths; and by hardware components, such as temperature or optical power. To date, the method used is to set the threshold to a fixed value that separates the 'correct' from the 'faulty' situation, what causes false and missing alarms, as we have seen in this dissertation. This proposed study should be targeted to analyze the behaviour of the signal in order to better set the threshold.
  More sophisticated signal processing methods ranging from linear time series analysis to neural networks, may help to replace the existence of thresholds by correlating analog signals themselves (and not only the binary signals issued when the thresholds are crossed, which have a much poorer informational content).

- Another important task is to test of the algorithms on real optical networks. Unfortunately, we did not have the opportunity to have a real testbed to perform tests. A performance test of the algorithms on real optical networks and if possible, on large networks involving a large number of optical components, is needed to confirm and extend the conclusions obtained by simulations in Chapter 7.

- The choice of the mismatching threshold could be automatized. In this dissertation, the Human Manager was asked to give a mismatching threshold m (or m_1/m_2), which corresponded to the accepted number of false and lost alarms. Future work could include an automatic selection of this threshold based on the results found by the algorithm and the network topology.

- The algorithms could be extended to other kinds of systems. Indeed, the separation realized by the problem abstraction between the actual system and the algorithms help to the apply the algorithms to other systems. For example, the algorithms could be used to locate failures in power supply networks by modeling them as sets of interconnected elements belonging to different categories.

- A more detailed study on the properties of the moduloid $\mathcal{C}$ generated by the codewords $\vec{g_1}, ..., \vec{g_t}$ associated to single failures could help finding properties that would allow to reduce the complexity of the pre-computing phase of the FLA algorithm, especially for correcting the erroneous codewords corresponding to false or missing alarms.

Appendix A

Established channels for different topologies

This appendix provides the set of channels $\mathcal{CH}$ for the examples of Section 4.5.

A.1 ARPA meshed network

Ch_1 : **(0 2 0 0)**(1 2 0 1)(3 2 0 18)(0 2 1 0)(0 0 2 8)(0 8 0 2)(2 8 18 1)(2 8 18 2)(2 8 18 3)(1 8 0 1)(3 8 0 18) (0 8 1 0)(0 0 8 10)(1 10 0 1)(0 0 10 15)(0 15 0 10)(2 15 18 1)(2 15 18 2)(2 15 18 3)(1 15 0 1)(3 15 0 18)(0 15 1 0) (0 0 15 16)(0 16 0 15)(2 16 18 1)(2 16 18 2)(2 16 18 3)(1 16 0 1)(3 16 0 18)(0 16 1 0)(0 0 16 18)(1 18 0 1) (2 18 0 1)(2 18 0 4)(2 18 0 3)**(0 18 0 0)**
Ch_2 : **(0 12 0 0)**(3 12 0 5)(1 12 0 1)(0 0 12 13)(0 13 0 12)(2 13 12 1)(2 13 12 2)(2 13 12 3)(1 13 0 1) (3 13 0 20)(0 13 1 0)(0 0 13 14)(1 14 0 1)(0 0 14 15)(0 15 0 14)(2 15 20 1)(2 15 20 2)(2 15 20 3)(1 15 0 1) (3 15 0 20)(0 15 1 0)(0 0 15 16)(0 16 0 15)(2 16 20 1)(2 16 20 2)(2 16 20 3)(1 16 0 1)(3 16 0 20)(0 16 1 0) (0 0 16 18)(1 18 0 1)(0 0 18 20)(1 20 0 1)(2 20 0 1)(2 20 0 4)(2 20 0 3)**(0 20 0 0)**
Ch_3 : **(0 3 0 0)**(1 3 0 1)(3 3 0 21)(0 3 1 0)(0 0 3 6)(0 6 0 3)(2 6 21 1)(2 6 21 2)(2 6 21 3)(1 6 0 1)(3 6 0 21) (0 6 1 0)(0 0 6 11)(1 11 0 1)(0 0 11 16)(0 16 0 11)(2 16 21 1)(2 16 21 2)(2 16 21 3)(1 16 0 1)(3 16 0 21)(0 16 1 0) (0 0 16 18)(1 18 0 1)(0 0 18 20)(1 21 0 1)(0 0 20 21)(1 21 0 1)(2 21 0 1)(2 21 0 4)(2 21 0 3)**(0 21 0 0)**
Ch_4 : **(0 1 0 0)**(3 1 0 5)(1 1 0 1)(0 0 1 2)(0 2 0 1)(2 2 1 1)(2 2 1 2)(2 2 1 3)(1 2 0 1)(3 2 0 14)(0 2 1 0) (0 0 2 8)(0 8 0 2)(2 8 14 1)(2 8 14 2)(2 8 14 3)(1 8 0 1)(3 8 0 14)(0 8 1 0)(0 0 8 9)(1 9 0 1)(0 0 9 12)(1 12 0 1) (0 0 12 13)(0 13 0 12)(2 13 14 1)(2 13 14 2)(2 13 14 3)(1 13 0 1)(3 13 0 14)(0 13 1 0)(0 0 13 14)(1 14 0 1) (2 14 0 1)(2 14 0 4)(2 14 0 3)**(0 14 0 0)**
Ch_5 : **(0 8 0 0)**(1 8 0 1)(3 8 0 15)(0 8 1 0)(0 0 8 10)(1 10 0 1)(0 0 10 15)(0 15 0 10)(2 15 15 1)(2 15 15 2) (2 15 15 3)(1 15 0 1)**(0 15 0 0)**
Ch_6 : **(0 11 0 0)**(3 11 0 5)(1 11 0 1)(0 0 11 16)(0 16 0 11)(2 16 11 1)(2 16 11 2)(2 16 11 3)(1 16 0 1)(3 16 0 17) (0 16 1 0)(0 0 16 15)(0 15 0 16)(2 15 17 1)(2 15 17 2)(2 15 17 3)(1 15 0 1)(3 15 0 17)(0 15 1 0)(0 0 15 14) (1 14 0 1)(0 0 14 13)(0 13 0 14)(2 13 17 1)(2 13 17 2)(2 13 17 3)(1 13 0 1)(3 13 0 17)(0 13 1 0)(0 0 13 17) (1 17 0 1)(2 17 0 1)(2 17 0 4)(2 17 0 3)**(0 17 0 0)**

A.2 COBNET network

Ch_1 : **(0 1 4 0)**(3 1 4 5)(1 1 4 1)(0 1 4 5)(1 1 5 1)(0 1 0 6)(0 1 0 3)(2 1 0 1)(2 1 0 2)(2 1 0 3)(1 1 0 1)(3 1 -2 4)(0 1 -2 1)(0 0 1 2)(0 2 -1 2)(2 2 -1 2) (2 2 -1 1)(1 2 0 1)(3 2 0 2)(0 2 0 1)(0 2 0 4)(1 2 1 1)(0 2 1 2)(1 2 2 1)(2 2 2 1)(2 2 2 4)(2 2 2 3)**(0 2 2 0)**
$Ch_{1'}$: **(0 2 2 0)**(3 2 2 5)(1 2 2 1)(0 2 2 3)(1 2 3 1)(0 2 3 4)(1 2 4 1)(0 2 4 5)(1 2 5 1)(0 2 0 6)(0 2 0 3)(2 2 0 4)(2 2 0 5)(2 2 0 6)(1 2 0 1)(3 2 -1 2) (0 2 -1 1)(0 0 2 1)(0 1 -2 2)(2 1 -2 4)(2 1 -2 3)(1 1 0 1)(3 1 0 4)(0 1 0 1)(0 1 0 4)(1 1 1 1)(0 1 1 2)(1 1 2 1)(0 1 2 3)(1 1 3 1)(0 1 3 4)(1 1 4 1) (2 1 4 1)(2 1 4 4)(2 1 4 3)**(0 1 4 0)**
Ch_2 :**(0 2 1 0)**(3 2 1 5)(1 2 11)(0 2 1 2)(1 2 2 1)(0 2 2 3)(1 2 3 1)(0 2 3 4)(1 2 4 1)(0 2 4 5)(1 2 5 1)(0 2 0 6)(0 2 0 3)(2 2 0 1)(2 2 0 2)(2 2 0 3)(1 2 0 1) (3 2 -1 1)(0 2 -1 1)(0 0 2 1)(0 1 -2 2)(2 1 -2 2)(2 1 -2 1)(1 1 0 1)(3 1 0 3)(0 1 0 1)(0 1 0 4)(1 1 1 1)(0 1 1 2)(1 1 2 1)(0 1 2 3)(1 1 3 1)(2 1 3 1) (2 1 3 4)(2 1 3 3)**(0 1 3 0)**
$Ch_{2'}$: **(0 1 3 0)**(3 1 3 5)(1 1 3 1)(0 1 3 4)(1 1 4 1)(0 1 3 5)(11 5 1)(0 1 0 6)(0 1 0 3)(2 1 0 7)(2 1 0 8)(2 1 0 9)(1 1 0 1)(3 1 -2 3)(0 1 -2 1)(0 0 1 2) (0 2 -1 2)(2 2 -1 6)(2 2 -1 5)(1 2 0 1)(3 2 0 1)(0 2 0 1)(0 2 0 4)(1 2 1 1)(2 2 1 1)(2 2 1 4)(2 2 1 3)**(0 2 1 0)**
Ch_3 : **(0 1 2 0)**(3 1 2 5)(1 1 2 1)(0 1 2 3)(1 1 3 1)(0 1 3 4)(1 1 4 1)(0 1 4 5)(1 1 5 1)(0 1 0 6)(0 1 0 3)(2 10 4)(2 1 0 5)(2 1 0 6)(1 1 0 1)(3 1 -2 2) (0 1 -2 1)(0 0 1 2)(0 2 -1 2)(2 2 -1 4)(2 2 -1 3)(1 2 0 1)(3 2 0 3)(0 2 0 1)(0 2 0 4)(1 2 1 1)(0 2 1 2)(1 2 2 1)(0 2 2 3)(1 2 3 1)(2 2 3 1) (2 2 3 4)(2 2 3 3)**(0 2 3 0)**
$Ch_{3'}$: **(0 2 3 0)**(3 2 3 5)(1 2 3 1)(0 2 3 4)(1 2 4 1)(0 2 4 5)(1 2 5 1)(0 2 0 6)(0 2 0 3)(2 2 0 7)(2 2 0 8)(2 2 0 9)(1 2 0 1)(3 2 -1 3)(0 2 -1 1)(0 0 2 1)(0 1 -2 2)(2 1 -2 6)(2 1 -2 5)(1 1 0 1)(3 1 0 2)(0 1 0 1)(0 1 0 4)(1 1 1 1)(0 1 1 2)(1 1 2 1)(2 1 2 1)(2 1 2 4)(2 1 2 3)**(0 1 2 0)**
Ch_4 : **(0 1 1 0)**(3 1 1 5)(1 1 1 1)(0 1 1 2)(1 1 2 1)(0 1 2 3)(1 1 3 1)(0 1 3 4)(1 1 4 1)(0 1 4 5)(1 1 5 1)(2 1 5 1)(2 1 5 4)(2 1 5 3)**(0 1 5 0)**
$Ch_{4'}$: **(0 1 5 0)**(3 1 5 5)(1 1 5 1)(0 1 0 6)(0 1 0 3)(2 1 0 10)(2 1 0 11)(2 1 0 12)(1 1 0 1)(3 1 0 1)(0 1 0 1)(0 1 0 4)(1 1 1 1)(2 1 1 1)(2 1 1 4)(2 1 1 3)**(0 1 1 0)**
Ch_5 : **(0 2 4 0)**(3 2 4 5)(1 2 4 1)(0 2 4 5)(1 2 5 1)(2 2 5 1)(2 2 5 4)(2 2 5 3)**(0 2 5 0)**
$Ch_{5'}$: **(0 2 5 0)**(3 2 5 5)(1 2 5 1)(0 2 0 6)(0 2 0 3)(2 2 0 10)(2 2 0 11)(2 2 0 12)(1 2 0 1)(3 2 0 4)(0 2 0 1)(0 2 0 4)(1 2 1 1)(0 2 1 2)(1 2 2 1)(0 2 2 3)(1 2 3 1)(0 2 3 4)(1 2 4 1)(2 2 4 1)(2 2 4 4)(2 2 4 3)**(0 2 4 0)**

Appendix B

Routine for multiple failures

This appendix gives a routine in pseudo-code to compute the leaves corresponding to multiple failures following the procedure of Section 5.5.2.

$setC = setC0$=$Bin(C_i)$ the set of all the binary vectors that correspond to the domain of single failures.
$newSetC$=`FindNewSet`($setC0, setC$);
`while` ($setC0 \neq$null)
    $newSetC$=`FindNewSet`($setC0$,$newSetC$);

where the function `setK=FindNewSet(setA, setB)` can be computed as follows:

```
setC=FindNewSet(setA, setB)
boolean newLeaf;
for ∀ Bin(C_i) ∈ setA
{
    newLeaf=false;
    for ∀Bin(C_j) ∈ setB, Bin(C_j) ≠ Bin(C_i)
        if(Bin(C_i) ∨ Bin(C_j) = Bin(C_k) is a new binary vector)
        {
          if ((Bin(C_k)∉ setA)&&(Bin(C_k)∉ setB))
              add Bin(C_k) to setK
          else
              update P(C_k)
        }
    if  (newleaf==false) setC0.remove(Bin(C_i));
}
return  setK;
```

Appendix C

Pseudo-code of some AFA modules

C.1 Alarm_Discarding_2 and Candidate Search procedure

This section gives the pseudo-code of the procedure that performs the second alarm discarding and the search of the candidates. This procedure covers the modules Alarm_Discarding_2 and Candidate Search.

```
boolean STOP = false
∀a_i ∈ R
    ∀CH_j ∈ CH with Pos(a_i.origin, CH_j) ≠ 0
        Find e_l ∈ V with Pos(e_l, CH_j) = Pos(a_i.origin, CH_j) − 1
        While STOP==false
            e_k = e_l
            If  e_k ∈ P then add e_k in temporalSet
            If e_k ∈ A1
               If  e_k ∈ R_orig then
                   a_i is discarded
                   STOP=true
               else
                   add e_k in temporalSet
            If e_k ∈ A2
               If  e_k ∈ R_orig then
                   a_i is discarded
                   STOP=true
               else
                   add e_k in temporalSet
            If e_k ∈ A3
               If  e_k ∈ R_orig then
                   a_i is discarded
                   STOP=true
                   add temporalSet in PC_2
               else
                   add e_k in temporalSet
                   STOP=true
            Find e_l ∈ V with Pos(e_l, CH_j) = Pos(e_k, CH_j) − 1
```

C.2 Domain_Calc procedure

This section presents the pseudo-code of the module that performs the *forward* approach of the AFA algorithm.

```
boolean STOP = false
∀e_i ∈ V_C
        ∀CH_j ∈ CH with Pos(e_i, CH_j) ≠ 0
                If e_i ∈ (A1 or A3) then
                    Add e_i in Domain(e_i)
                Find e_l ∈ V_C with Pos(e_l, CH_j) = Pos(e_i, CH_j) + 1
                While STOP=false
                        e_k = e_l
                        If   e_k ∈ A2 then add e_k in Domain(e_i)
                        If   e_k ∈ A3 then STOP=true
                        Find e_l ∈ V with Pos(e_l, CH_j) = Pos(e_k, CH_j) + 1
                Add Domain(e_i) in set of lists D
```

Appendix D

Glossary

AAL	ATM Adaptation Layer
ADF	Add-Drop Filters
AFA	Alarm Filtering Algorithm
AIS	Alarm Indication Signal
ANN	Artificial Neural Network
APD	Avalanche Photodiodes
ASE	Amplified Spontaneous Emission
ASN.1	Abstract Syntax Notation 1
ATM	Asynchrouns Transfer Mode
BER	Bit Error Rate
BIP	Bit Interleave Parity
BPF	Band Pass Filter
CBR	Case Based Reasoning
CMIP	Common Management Information Protocol
COBNET	Corporate Optical Backbone NETwork
CPN	Customer Premise Network
CRC	Cyclic Redundancy Check
CW	Continuous Wave
DDCMP	Digital Data Communications Message Protocol
DEMUX	Demultiplexer
DWDM	Dense WDM
EDFA	Erbium-Doped Fiber Amplifiers

FEC	Forward Error Control
FLA	Fault Location Algorithm
FMS	Fault Management System
FSM	Finite State Machine
GTE	Global Testing Equipment
GUI	Graphical User Interface
HEN	High End Node
HF	Hard Failure
IMPACT	Intelligent Management Platform for Alarm Correlation Tasks
IP	Internet Protocol
ISI	Inter-Symbol Interference
ISO	International Organization for Standardization
ITE	Individual Testing Equipment
LAN	Local Area Network
LASER	Light Amplification by Stimulated Emission of Radiation
LED	Light-Emitting Diodes
LEN	Low End Node
MAN	Metropolitan Area Network
MC	Minimum Cover problem
MIB	Management Information Base
MS	Multiplex Section
MSOH	Multiplex Section Overhead
MSTE	Multiplex Section Terminating Element
MUX	Multiplexer
NP	Non-Polynomial
OF	Optical Fiber
OSNR	Optical Signal to Noise Ratio
PCP	Pre-Computing Phase
PDH	Pleosynchrous Digital Hierarchy
PDU	Protocol Data Unit
PF	Progressive Failure
PIN	P-doped, Intrinsic, and N-doped photodiode
PIU	Public Interface Unit
PN	Public Network
PNO	Public Network Operator
POH	Path Overhead
PTE	Path Terminating Element

RS	Regenerator Section
RSOH	Regenerator Overhead
RSTE	Regenerator Section Terminating Element
Rx	Receiver
SDH	Synchronous Digital Hierarchy
SDM	Space Division Multiplexing
SLA	Semiconductor Laser Amplifiers
SONET	Synchronized Optical NETwork
SNMP	Simple Network Management Protocol
SNR	Signal to Noise Ratio
STM	Synchronous Transport Module
TCP	Transmission Control Protocol
TDM	Time Division Multiplexing
TMN	Telecommunication Management Network
Tx	Transmitter
UDP	User Datagram Protocol
VC	Virtual Container
WAN	Wide Area Network
WDM	Wavelength Division Multiplexing

Appendix E

Notations

$\mathcal{A}$	set of 'alarming' network components
$\mathcal{AL}$	set of network elements able to issue alarms, $A \cup M$.
$\mathcal{ANR}$	Alarms_Non_Redundant
$\mathcal{CH}$	Established channels
CH_i	Channel
$\mathcal{GF}$	Galois Field $\{0, 1\}$
k	Number of received alarms, cardinality of $\mathcal{R}$
l	Number of established channels, cardinality of $\mathcal{CH}$
M	Monitoring Equipment
m	Mismatching threshold $m = m_1 + m_2$
m_1	Allowed number of lost alarms
m_2	Allowed number of false alarms
n	Number of network elements, cardinality of $\mathcal{V}$
n_a	Number of network elements able to issue alarms, cardinality of $\mathcal{AL}$
$\mathcal{NS}_i$	union of $\mathcal{PC}_1$ with set of subsets of $\mathcal{PC}_2$ having cardinality i
P	Non-alarming components
$\mathcal{PC}$	Possible_Candidates=$\mathcal{PC}_1 \cup \mathcal{PC}_2$
$\mathcal{PC}_1$	set of $A1$ and $A3$ elements that have issued an alarm
$\mathcal{PC}_2$	set of elements related with the $A2$ non-discarded alarms
$\mathcal{R}$	Received alarms
$\mathcal{R}_{orig}$	Components that have issued the alarms $\mathcal{R}$
$\mathcal{SC}$	Subset_Candidates
$\mathcal{V}$	Network components

List of Figures

List of Tables

References

[1] COBNET Consortium. *COBNET Corporate Optical Backbone NETwork.* URL: http://lrcwww.epfl.ch/COBNET/index.html, 1998.

[2] ARTIS Software Corporation. Optsim, 1999.

[3] D. J. G. Mestdagh. *Fundamentals of Multi-access Optical Fiber Networks.* Artch House, 1995.

[4] R. Ramaswami and K. Sivarajan. *Optical Networks, a practical prespective.* Morgan Kaufmann Publishers, 1998.

[5] Alan Eli Willner. Mining the optical bandwidth for a terabit per second. *IEEE Spectrum*, pages 32–42, April 1997.

[6] R. Chipalkatti, Z. Zhang, and A. S. Acampora. Protocols for optical star-coupler network using WDM: performance and complexity study. *IEEE Journal on Selected Areas in Communications*, 11(4):579–589, May 1993.

[7] J. M. Senior, M. R. Handley, and M. S. Leeson. Developments in Wavelength Division Multiple Access Networking. *IEEE Communications Magazine*, pages 28–36, December 1998.

[8] EPFL Editor. Design of the control and management functions. Deliverable COBNET DW142, COBNET Consortium, May 1997.

[9] S. Abek H. Hegerin and B. Neumair. *Integrated Management of Networked Systems.* Morgan Kaufmann Publishers, 1998.

[10] M. Rose and K. McCloghrie. RFC.1155: Structure and Identification of Management Information for TCP/IP-based internets, 1990.

[11] M. Rose and K. McCloghrie. RFC1213: Management Information Base for Network Management of TCP/IP-based internets: MIB-II, 1991.

[12] M. Rose. RFC1215: A Convention for Defining Traps for use with SNMP, 1991.

[13] M. Nyberg. *Model based Fault Diagnosis: Methods, Theory and Automotive Engine Applications.* Linköping University, 1999.

[14] C. Wang and M. Schwartz. Fault Detection with Multiple Observers. *IEEE INFOCOM Proc.*, pages 2187–2196, 1992.

[15] A.T. Bouloutas et al. Fault Identification Using a FSM model with Unreliable Partially Observed Data Sequences. *IEEE Trans. on Communications*, 41(7):1074–1083, July 1993.

[16] I. Katzela and M. Schwartz. Schemes for Fault Identification in Communication Networks. *IEEE/ACM Trans. on Networking*, 3(6), December 1995.

[17] C. Wang and M. Schwartz. Identification of faulty links in dynamic-routed networks. *IEEE Journal on Selected Areas in Communications*, 11(9):1449–1460, December 1993.

[18] K. R. Sheers. Hp OpenView Event Correlation Sevices. *Hewlett-Packard Journal*, October 1996.

[19] S. Brugnoni et al. An expert system for real time fault diagnosis of the italian communications network. *Integrated Network Management*, III:617–628, 1993.

[20] R. Weihmayer G. Jakobson and M. Weissman. A domain-oriented Expert System for Telecommunication Network Aalarm Correlation. *Network Management and Control*, 2, 1994.

[21] R. Gardner and D. Harle. Methods and Systems for Alarm Correlation. *Globecom 96 proceedings*, pages 136–140, 1996.

[22] H.-G. Hegering and Y. Yemini (Editors. *Integrated Network Management.* Elsevier Science Publishers B.V., 1993.

[23] R. Gardner and D. Harle. Alarm Correlation and Network Fault Resolution using Kohonen Self-Organising Map. *Globecom 97 proceedings*, pages 1398–1402, 1997.

[24] T. Kohonen. *Self-Organizing Maps.* Springler-Verlag, 1995.

[25] T. E. Stern and K. Bala. *Multiwavelength Optical Networks, a layered approach.* Addison-Wesley, 1999.

[26] B. Mukherjee. *Optical Communication Networks.* Mc Graw-Hill Series, 1997.

[27] Wandel & Goltermann. Understanding Dense WDM.

[28] A. Ghatak and K. Thyagarajan. *Introduction to Fiber Optics.* Cambridge University Press, 1998.

[29] Submarine Networks. *FibreSytems*, 3(5):26, June 1999.

[30] E. Hecht. *Optics.* Addison-Wesley publishing company, 1987.

[31] Anritsu. Catalog of Electronic Measuring Instruments, 1999.

[32] Hewlett Packard. Test & Measurement 1999 Catalog. http://www.hp.com/go/tmc99.

[33] Photonetics. Multi-wavelength analyzer. http://www.phononetics.com/.

[34] ITU-T Rec. G.707. Network Node Interface for the SDH, 1996.

[35] ITU-T Rec. G.783. Characteristics of SDH equipment functional blocks, 1997.

[36] Edited by C. A. Siller. *From SONET/SDH: a sourcebook of synchrouns networking.* IEEE Press, 1996.

[37] ITU-T Rec. G.975. Forward Error Correction for Submarine Systems, 1996.

[38] M. Prycker. *Asynchronous Transfer Mode.* Prentice Hall, 1995.

[39] ATM Forum. *ATM: User-Network Interface Specification.* Prentice Hall, 1993.

[40] R. Stevens. *TCP/IP Illustrates: Volume I.* Addison-Wesley, 1994.

[41] C. Mas et al. Fault location at the WDM Layer. *Photonic Network Comm.*, 1(3):235–255, November 1999.

[42] J. Walrand. *Communication Networks.* Aksen Associates.

[43] ITU-T Rec. M.3100. Generic network information model, 1995.

[44] A.T. Bouloutas et al. Alarm Correlation and Fault Identification in Communication Networks. *IEEE Trans. on Communications*, 42(2/3/4):523–533, Feb/March/April 1994.

[45] ITU-T Rec. X.721. Information Technology-Open Systems Interconnection-Structure of management Information: Definition of management information, 1992.

[46] Carmen Mas and Patrick Thiran. An Efficient Location Algorithm for IP/WDM Networks. *in proceedings of IEEE/ACM/SPIE Workshop on Optical Networks*, January 2000.

[47] N. S. V. Rao. Computational Complexity Issues in Operative Diagnosis of graph-based Systems. *IEEE Trans. on Computers*, 42(4):447–457, April 1993.

[48] N. S. V. Rao. On Parallel Algorithms for Single-Fault Diagnosis in Fault Propagation Graph Systems. *IEEE Trans. on Parallel and Distributed Systems*, 7(12):1217–1223, December 1996.

[49] F. Baccelli et al. *Synchronization and Lineraity.* John Wiley & Sons, 1992.

[50] D. Derickson. *Fiber Optic Test and Measurement.* Prentice Hall, 1998.

[51] A. Bruce Carlson. *Communications Systems.* Mc. Graw-Hill Book Company, 1986.

[52] C. M. Weinert. BER Evaluation for optical signals with combined EDFA and Interference noise. *2000 International Zurich Seminar on Broadband Communications*, pages 85–87, February 2000.

[53] Caroline Brisson. *Couche Physique d'un réseau optique local en étoile: étude et réalisation experimentale.* Ph. D. Thesis: Ecole Polytechnique Fédérale de Lausanne, 1999.

[54] Daniel Rodellar Gomez. *Performance analysis of Multi-Channel protocols for optical local area networks exploiting wavelength division multiplexing.* Ph. D. Thesis: Ecole Polytechnique Federal de Lausanne, 1999.

[55] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the theory of NP-Completeness.* W. H. Freeeman and Company, 1979.

Curriculum Vitae

Carmen MAS MACHUCA
Born the 30th October 1970 in Huesca (Spain).
Languages: Spanish (mother tongue), French and English (fluent), Greek (basic).

EDUCATION

Oct 94 - Mar 95 **Diploma Thesis**
Laboratory for Photonic Computing and Perception
Vrije Universiteit Brussel, Belgium
Diffractive Optical Elements for Optical Interconnections

Oct 90 - Mar 95 **M.S. and B.S. Telecommunication Engineering**
Universitat Politècnica de Catalunya, Barcelona (Spain)

EXPERIENCE

May 96 - to date **Research Assistant**
Ecole Polytechnique Fédérale de Lausanne, Switzerland
Partner Manager of the COBNET ACTS european project
5 graduate and undergraduate projects definition/supervision
3 presentations at international conferences

Jan 95 - Jun 95 **Research Assistant**
Laboratory for Photonic Computing and Perception
Vrije Universiteit Brussel, Belgium
Design of diffractive lenses for use in computing systems

Oct 93 - Feb 94 **Laboratory Assistant**
Image laboratory, Universidad Politecnica de Catalunya, Barcelona (Spain)

HONOURS

- Achievement Award, EPFL, October 1999.
- La Caixa Fellowship, for graduate studies in the King's College London, Great Britain, 1996.
- Erasmus Scholarship, for student exchange within the European Union, 1994-1995

PUBLICATIONS

- **An efficient Fault Localization Algorithm for IP/WDM Networks**
  Carmen Mas, Patrick Thiran
  IEEE/ACM/SPIE Workshop on Optical Networks
  Dallas, 2000.
- **Fault Localization at the WDM Layer**
  Carmen Mas, Patrick Thiran, Jean-Yves Le Boudec
  Photonic Network Communications Journal
  Volume 1, Number 3, November 1999, p. 235-255.
- **Fault Localisation in an Optical Network**
  Carmen Mas, Olivier Crochat, Jean-Yves Le Boudec
  SPIE'98 Voice, Video, and Data Communications
  All-Optical Networking: Architecture, Control, and Management Issues Proceedings p. 408-419, Boston, 1998.
- **An Alarm Filtering Algorithm for Optical Networks**
  Carmen Mas, Jean-Yves Le Boudec
  Management of Multimedia Networks and Services
  MMNS97 Proceedings p. 205-218, Montreal, Canada, 1997.

TECHNICAL REPORTS

- **COBNET control and management evaluation report**
  DW144 COBNET Deliverable, *EPFL Eds*, December 1998.
- **Specification of Control and Management Functions for COBNET**
  DW143 COBNET Deliverable, *EPFL Eds*, September 1997.
- **Design of the Control and the Management Functions**
  DW142 COBNET Deliverable, *EPFL Eds*, March 1997.