An Efficient Algorithm for Locating Soft and Hard Failures in WDM Networks

Carmen Mas and Patrick Thiran, *Member, IEEE*

***Abstract*—Fault identification and location in optical networks is hampered by a multitude of factors: the redundancy and the lack of coordination (internetworking) of the managements at the different layers (WDM, SDH/SONET, ATM, IP); the large number of alarms a single failure can trigger; the difficulty in detecting some failures; and the resulting need to cope with missing or false alarms. Moreover, the problem of multiple fault location is NP-complete, so that the processing time may become an issue for large meshed optical networks.**

We propose an algorithm for locating multiple failures at the physical layer of a WDM network. They can be either hard failures, that is, unexpected events that suddenly interrupt the established channels; or soft failures, that is, events that progressively degrade the quality of transmission; or both. Hard failures are detected at the WDM layer. 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). Both types of failures, and both types of error monitoring, are incorporated in our algorithm, which is based on a classification and abstraction of the components of the optical layer and of the upper layer. Our algorithm does not rely on timestamps nor on failure probabilities, which are difficult to estimate and to use in practice. Moreover, our algorithm also handles missing and false alarms. The nonpolynomial computational complexity of the problem is pushed ahead into a precomputational phase, which is done off-line, when the optical channels are set up or cleared down. This results in fast on-line location of the failing components upon reception of the ringing alarms.

***Index Terms*—Fault location, lost and false alarms, multiple failures, optical networks, wavelength division multiplexing (WDM).**

I. Introduction

THE RECENT explosion of Internet traffic has resulted in an evolution of backbone networks: from electrical to optical; and, among the latter ones, from space division multiplexing (SDM) and time division multiplexing (TDM) to wavelength division multiplexing (WDM). The evolution does not imply that there has been a reassignment of management tasks to other layers. On the contrary, the main drawback of today's networks is the existence of several management layers, each having its own management routines. This is the case, for example, with IP over (ATM over) SDH over WDM. Interoperability is needed to speed up the process and avoid task duplication. In particular, fault management is one of the most redundant tasks because it is performed at all layers and could be improved by cooperation between the different management layers. Indeed, the manager of each layer receives different information about the network behavior. At the physical layer, the manager receives information about the physical properties of the network such as optical power and temperature of the equipment. At the WDM layer, the manager obtains information about the quality of the optical signals such as signal-to-noise ratio (SNR) and crosstalk from some testing equipment such as spectrum analyzer [2], [3]. The manager at the upper layers receives more detailed information about the quality of the signal such as the bit error rate (BER), which is specific to each transmission technology. When a failure occurs, each manager tries to locate the failure based on the information obtained at his/her layer. For example, with SDH (synchronous digital hierarchy) over WDM links, an optical fiber break will trigger several alarms at the optical and SDH layers [4], [5]. In this paper we propose an algorithm that is able to cope with information from several layers to locate the failure(s). It is tailored to optical networks but is generic enough to be applied to existing networks using optical fibers that have several management layers (IP/ATM/SDH/WDM), or to future networks. It is believed that the SDH/SONET layer will always exist under some form until a new generation of high speed switches, routers, and alternative transport systems emerge that will provide the same functionality as SDH/SONET, but at the optical layer, and will allow direct IP over WDM.

Different methods have been proposed for fault location in the literature. They differ on the network model that is used (e.g., model based on the physical topology [6] or on the description of the established channels [7]); on the information needed (e.g., probabilities [8]) as input for the algorithm; and on the information processing methodology to locate the failures (e.g., finite state machine [9] or artificial neural networks [10]). The goal of all these methods is to identify the components whose failure has caused the received alarms. Other algorithms achieve the same goal by discarding redundant alarms. Our previous work [11] describes an alarm filtering algorithm where the considered failures are *hard* failures that are unexpected, sudden events that cause the interruption of the transmission channel, such as an optical fiber cut or a broken laser . The information obtained from the network in all these methods is minimal (binary). For example, a receiver sends an alarm when the power is below a certain threshold and the content of the alarm will be "loss of signal."

Our goal is to identify and locate rapidly not only *hard* but also *soft* failures that are the result of aging equipment or mis-

Manuscript received October 18, 1999; revised May 15, 2000. This work was supported in part by FN Grant 2100-050616.97.

C. Mas was with the ICA/DSC Department of Communication Systems, EPFL, CH-1015 Lausanne, Switzerland. She is now with the Development Programmes Department, Intracom, Athens, Greece.

P. Thiran is with Sprint ATL, Burlingame, CA 94010 USA, on leave from the ICA/DSC Department of Communication Systems, EPFL, CH-1015 Lausanne, Switzerland.

Publisher Item Identifier S 0733-8716(00)09008-9.

alignments and provide much richer (nonbinary) information (such as BER or SNR). Moreover, we must tolerate 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 too high or, conversely, may cause false alarms when the threshold is too low. At the same time, the time to locate the failure(s) when new alarms reach the manager must be kept as short as possible.

This paper is organized as follows. Section II describes the signals available at different layers that provide information about failures in an optical network. Section III first describes the hardware components of an optical network whose potential failures need to be identified, and then describes the monitoring equipment that provides additional information about the failures of the hardware components. The latter includes both measurement devices at the optical layer and performance monitoring functions implemented at higher layers. This study of the behavior of the network components and of the monitoring equipment results in a classification that abstracts the failure location problem in Section IV, and from which one can devise an efficient algorithm in Section V. Three important features of our fault location algorithm (FLA) are 1) the minimal diagnosis time, i.e., the time to locate failures when the management function receives alarm(s); 2) the location of multiple, simultaneous failures; and 3) the tolerance of false and lost alarms. Section VI shows results of the application of the FLA to examples on an SDH/WDM network; and Section VII concludes the paper.

II. Available Failure Indications in an Optical Network

This section describes the signals that can notify about failures in an optical network. Some signals contain only binary information, 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 informing about either an internal problem of the equipment or a problem with the incoming signal. The WDM layer can provide the following analog information: distribution of power of individual carriers over the full bandwidth, channel wavelength, signal-to-noise ratio (SNR), and crosstalk. 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 dependent on transmission technology. Assuming that the transmission technologies are known, BER can either be measured by a network tester or be delivered by the higher layers such as SDH through parity check, Ethernet by CRC computation or TCP/IP through the checksum, etc.

The management information can reach the manager either through a management network independent of the data network and assumed to be reliable, or through the data network itself, which is assumed to be a protected network allowing the manager to receive most of the alarms through alternative paths when failures occur.

III. Optical Network Components

We distinguish two classes of network components. The first one, described in Subsection A, contains the *hardware component* whose failure needs to be identified because it will degrade or interrupt the channels. The second one, described in Subsection B, contains the *monitoring equipment* that is present at one or more layers and provides 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.

A. Hardware Components of the Optical Layer

This subsection briefly reviews the hardware components found in optical networks, along with the most common information they can deliver to the manager. This information depends on the software that controls them. A more detailed description can be found in [11].

- *Optical Fibers (OFs)* 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. They cannot communicate with the manager.
- *Transmitters (Txs)* are lasers or laser arrays that convert the electrical signal into an optical signal at a certain wavelength. The resolution of the laser limits the spacing between the different wavelengths of the different channels, and hence the number of channels in WDM networks. New lasers used in advanced WDM networks are tunable and can change the emission wavelength within a prescribed range [12].

  Transmitters send alarms when either the temperature or the incoming power is beyond a prescribed range. If the analog value of these variables could be retrieved as such, a more proactive fault management of the laser would be possible. In this work, we have considered transmitters sending binary alarms and analog information when requested by the manager. In case the transmitters are able to continuously send analog information instead of binary alarms, they can be considered as pieces of both hardware and monitoring equipment.
- *Receivers (Rxs)* convert an optical signal, which corresponds to a certain wavelength, into an electrical signal. They send alarms when the input optical power is under an accepted level.
- *Add-Drop Filters (ADFs)* are able to drop and add a certain wavelength to an optical signal with several wavelengths without distorting the other wavelengths.
- *3Rs (Regenerator/Reshaper/Retimer)* are framers able to amplify the electrical signal, give the original shape of the signal, and readjust the time interval between pulses. They send alarms when they cannot lock to the incoming signal.

- *Protection Switches* receive more than one optical signal, and select one of them that has an acceptable power level. They send an alarm when they change the switch position due to an unacceptable incoming optical power.
- *Multiplexers (MUXs)* and *demultiplexers (DEMUXs)* are able to pass from several optical signals at different wavelengths into one optical signal that contains all the wavelengths, and vice versa.
- *Switches* allow cross-connection, that is, the connection of a particular input with a particular output. An alarm is sent when the connection cannot be established.
- *Amplifiers* output a signal at a higher power level than the input signal. Optical amplifiers do not perform conversion to electrical signal before amplifying. Most of the amplifiers add distortion to the signal.

B. Additional Equipment Sensitive to Progressive Failures

The alarms provided by the physical layer are sufficient to locate hard failures, but not soft ones. This information is provided either by performance monitoring at higher layers or, whenever available, by WDM monitoring equipment. The latter are described in Subsection 1, the former in Subsection 2 for SDH, and in Subsection 3 for performance equipment at other layers. We briefly indicate how ATM and TCP/IP provide similar information in Subsection 3. We refer to all the equipment as *monitoring components*.

1) WDM Measurement Equipment: Devices directly measuring the quality of the optical signal can be divided into the two following categories:

- *Individual Testing Equipment (ITE)* designates a device that can measure parameters of a *single* channel, that is, of a single wavelength. An example is the network tester ANT-20 [2], which can calculate the BER of a given channel and display the results. The drawback of measuring and relying on the BER is that in order to compute it, the testing equipment must be aware of the transmission technology (ATM, SONET, SDH, etc.). This detailed knowledge of the upper layer should be avoided since one of the advantages of WDM networks is the transparency with respect to the transmission technologies used.
- *Group Testing Equipment (GTE)* measures the quality of the *overall* optical signal, which includes the used wavelengths, the maximum power at each wavelength, and the SNR of each channel. Examples of these equipments are the spectrum analyzers (for example, OSP-102A [2] or MON-001 [3]).

2) SDH/SONET Equipment: Synchronous digital hierarchy (SDH) and synchronous optical network (SONET) comprise 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 U.S. 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. SDH works on three levels: i) regenerator section layer (RS layer), which deals with retiming and reshaping of frames; ii) multiplex section layer (MS layer), which performs the multiplexing and demultiplexing operation; and iii) path layer, which takes care of the transport assembly.

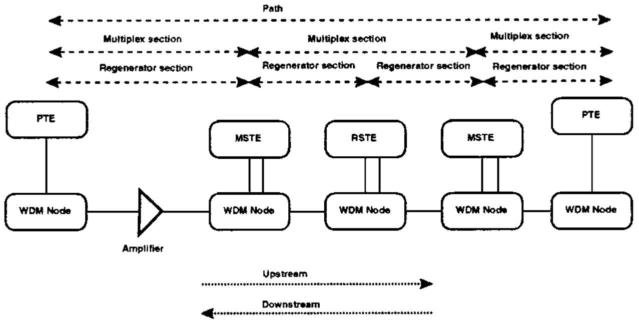


Fig. 1. SDH channel example.

Transmission performance is provided by parameters based on the detection defects and on the ratio of errored blocks (EB). A block is a set of consecutive bits monitored by means of an error detection code: the bit interleaved parity (BIP) which is a special case of the cyclic redundancy check (CRC) with polynomial of $x^n + x^0$ (for BIP-n) [13]. The BIP-n used at each layer are: BIP-8 at the RS layer, BIP-24N at the MS layer (where N is the number of basic STMs in one frame and which can be equal to 1,4, 16, or 64) and either BIP-2 or BIP-8 at the path layer, depending on the frame. The BIP result is stored in the B-bytes of the frame header. Three kinds of different SDH elements can be distinguished (see Fig. 1).

- *Regenerator Section Terminating Element (RSTE)* is the element which originates/terminates a regenerator section (RS). At an RSTE, the BIP-8 is computed over all bits of the previous frame and stored in the B1 byte of the following frame. The next RSTE checks the byte parity of the preceding frame and detects errored blocks. Before retransmitting 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.
- *Multiplex Section Terminating Element (MSTE)* originates/terminates a multiplex section (MS). In the SDH layering, every MSTE covers both the RS and MS layers. The error monitoring function at the RS layer is that 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 [14]. At one MSTE and at the MS layer, the BIP-24N is computed over all bits (except those in the regenerator overhead) of the previous frame and placed in the $3\times$ N respective B2 positions of the multiplex overhead of the following frame. At the next MSTE, BIP-24N code is computed for the received frame and they are compared to the $3\times$ N error monitoring B2 bytes recovered from the multiplex overhead [15] of the next frame. A difference between both values is an evidence of an errored block. Before retransmitting the frame, bytes B2 are updated, thereby masking any detected error in the frame from the next MSTE.
- *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.

3) Other Monitoring Equipments: Let us briefly mention some possible monitoring functionalities at higher layers.

- At the IP layer, routers can be considered as monitoring equipment since they perform a header checksum before retransmitting the packets.
- At the TCP and UDP layers, the kernels of the end systems can be considered as monitoring equipment since they perform a checksum of data and header before accepting a packet.
- In ATM, there is a CRC-10 error-detecting code at the trail termination that uses a common cell payload format.
- In submarine cables, a forward error control (FEC) mechanism is used to control the errors in a heavily used point-to-point WDM system [16].

C. Properties of Alarms

We now study the behavior of the network elements when a failure occurs:

- When a *hard* (or sudden) failure occurs, such as a fiber cut or the turnoff of a transmitter, the generated alarms are signals from both the hardware components and the monitoring equipment. Depending on the location of the latter, measurements may help to locate the hard failure better than just the binary alarms from the hardware equipment [17].
- *Soft* or progressive failures, such as the shift of a filter curve, do not cause binary signals from the hardware components but an increase/decrease of the involved measured analog signals. These analog signals can be associated to one or to all the transmitted channels. For example, if the curve of an ADF gets shifted, the quality of the signal at the output of the filter and in other further equipment will degrade. The FLA should be able to detect that there is a problem, and to locate the failure.

1) Alarming Properties of the Hardware Components: Let us define the different alarming properties of the network components based on their behavior when a failure occurs. The three following *features* can be distinguished.

Self-Alarmed: This property specifies whether a network component is able to send an alarm informing about its *own hard failure* or not. An example of a self-alarmed component is a transmitter whose microcontroller controls power and 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 *hard failure external* to them. For example, receivers are able to detect that there is no incoming power and to 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, therefore they are not out-alarmed components.

Hard Failure Masking (HF Masking): This property specifies whether the network component masks the hard failure to the hardware components that follow it on the channel. For example, the laser of a transmitter sends power even if there is no incoming signal (due to a hard 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.

2) 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 to other monitoring equipment.

Failure Masking: This property specifies whether the monitoring equipment masks the failure to any other monitoring equipment of its own layer or not. For example, a WDM spectrum analyzer does not mask any failure to the next measuring equipment because it does not act on the content of the signal. On the contrary, an SDH MSTE masks progressive failures from 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. It does not mask from PTE because the associated B3-byte remains intact.

D. Optical Network Components Classification

1) Hardware Components Classification: We can now check which of the three properties of Section III-C-1 applies to each hardware component. The properties of each component may change depending on the type of component. The classification proposed in this section is based on the components described in Section III-A. The properties of Table I enable us to classify the components in the following *categories*.

1) The *"nonalarming" components* do not give any information to the manager because they do not have any microcontroller. They are denoted by P and they are represented by a circle on the figures.
2) The *"alarming" components* are able to communicate with the manager because they have some programmable software on 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 on 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 hard 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 for an optical amplifier.

2) Monitoring Equipment Classification: Monitoring equipment is also classified according to its failure masking property, resulting in the classification of Table II. Let M denote the class of all monitoring components. An Mq component, where $q \in \mathbb{N}_0$, is a monitoring component that masks soft failures to other monitoring components Mp such that $1 \le p \le q$, and such that follow it on the same channel. $M0$ components do not mask any failure and are represented by a

TABLE I
ALARM PROPERTIES OF THE NETWORK COMPONENTS AND THE RESULTING CLASSIFICATION

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 II
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 OR NOT FROM THE ELEMENT AT THE TOP OF THE CONSIDERED COLUMN

	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

rhombus. SDH will introduce $M1, M2$, and $M3$ components, that will be represented, respectively, by a thick square, a thick square with an oblique line, and a thick square with a cross. Table II shows which monitoring piece of equipment masks failures from another piece, and the resulting classification.

IV. PROBLEM ABSTRACTION

The classification of the previous section enables us to derive and implement the FLA to locate the component(s) whose failure has caused the alarms received by the manager.

A. Alarm Mismatching Thresholds

Some sets of received alarms may remain unexplained by any combination of faulty components. This means that at least one alarm was lost or false. To cope with this, we introduce two *alarm mismatching thresholds,* which are two parameters that reflect the reliability of the management channel and of the management functions of the equipment. They are denoted by m_1 and m_2, and give the maximum number of lost and false alarms, respectively, 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 $m_1 = m_2 = 0$. The value of m_1 and m_2 can be set *a priori* by the network manager. The availability to cope with the nonideal scenario is of crucial importance, as supported by the following examples.

- When an $A1$ element fails, it may not send the alarm it is supposed to. For example, a switch having an internal failure may not be able to send the expected alarm (because the failed board is turned off) 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.
- At the SDH, IP, or ATM layers, there is a mechanism that checks whether there has been an error at the bit level. This mechanism (for example, BIP for SDH or CRC for ATM) may leave errors undetected. The alarms that were expected under the failure conditions will be considered as lost.
- 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.

B. Inputs of the Algorithm

The objects manipulated by the FLA are the following.

- *Component* $comp$ is a network component that belongs to one of the aforementioned categories. It has an identifier with two fields: the first 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 is the identifier of the component, which issued the alarm, and the second is the informational content of the alarm. The information is relevant only for alarms issued by a mon-

itoring component, because they give either an analog value (for example BER) or a binary value (such as BIP-8 mismatch at the RSTE) informing about *soft* failures.

- *Channel* $CH_i = \{comp_j\}$ is an ordered list of components. The channels are considered here as unidirectional. Bidirectional channels are equivalent to a pair of unidirectional channels. Function $Pos(comp, CH_i)$ returns the position of *comp* within the channel CH_i if *comp* belongs to this channel, and 0 otherwise.

The inputs of the algorithm are: the set of established channels $\mathcal{CH}$, which is updated every time a channel is established, modified, or cleared down; the set of alarms received by the manager $\mathcal{R}$ and the mismatching thresholds m_1 and m_2.

C. Domain Definition

Domain(comp) is defined as the set of network elements that will send an alarm when *comp* fails. Two different kinds of *Domain* 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 makes use of the three following functions.

1) 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

$e_1\ FP1\ e_2 = 1$ if and only if

- $e_2 \in A2$
- $\exists\, CH_i \in \mathcal{CH}$ with $0 < Pos(e_1, CH_i) < Pos(e_2, CH_i)$
- $\forall\, e_j$ with $Pos(e_1, CH_i) < Pos(e_j, CH_i) < Pos(e_2, CH_i), \quad e_j \notin A3.$ (1)

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 if there is no other monitoring element that masks the failure from e_2. Mathematically this can be expressed as follows:

$e_1\ FP2\ e_2 = 1$ if and only if

- $e_2 \in Mq$ for some $q \geq 0$
- $\exists\, CH_i \in \mathcal{CH}$ with $0 < Pos(e_1, CH_i) < Pos(e_2, CH_i)$
- if $q \geq 1$ then $\forall\, e_j \in Mp$ with $Pos(e_1, CH_i) < Pos(e_j, CH_i) < Pos(e_2, CH_i), p < q$

else $\forall\, e_j \in \mathcal{V}$ with $Pos(e_1, CH_i) < Pos(e_j, CH_i) < Pos(e_2, CH_i), \quad e_j \notin A3.$ (2)

3) An element e_1 will send an alarm when it fails if it is self-alarmed, that is,

$$e_1\ FP3\ e_2 = 1 \quad \text{if and only if } e_1 = e_2 \in A1 \cup A3. \quad (3)$$

Based on these relations, 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 $A3$ component between them and e_1, and finally iii) the monitoring components taking into account their failure masking properties of Table II. Mathematically, *HDomain*(e_1) can be expressed as follows:

$$HDomain(e_1) = \{e_2 \in V \,|\, (e_1\ FP1\ e_2 = 1) \text{ or } (e_1\ FP2\ e_2 = 1) \text{ or } (e_1\ FP3\ e_2 = 1)\}. \quad (4)$$

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

$$SDomain(e_1) = \{e_2 \in M \,|\, e_1\ FP2\ e_2 = 1\}. \quad (5)$$

V. Fault Location Algorithm (FLA)

Time to locate the failure(s) is critical, and the FLA must locate failures as quickly as possible. Unfortunately, the multiple fault location problem has been shown to be NP-complete already in the ideal scenario [1]. Nevertheless, the computation that has to be carried out when a new alarm reaches the manager can be kept short despite the potentially large size of the network, if we follow Rao's approach to precompute, as much as possible, the functions that can be executed independently of the received alarms. This phase is called *precomputation phase (PCP)*. Once the manager starts receiving alarms from the network, the algorithm does not have to perform complex computations but must simply traverse a binary tree. The precomputation phase of the algorithm is executed only when the set of channels $\mathcal{CH}$ 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 precomputation phase has been implemented on the basis of the algorithm devised by Rao [18], 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 III-D, to multiple failures, and finally to the nonideal scenario.

Let us present each of the extensions of the algorithm, step by step. The final scheme of FLA is presented in Fig. 2.

A. FLA to Locate Single Failures in the Ideal Scenario

1) The Network has Only P and $A2$ Components: We first solve the problem of locating a single hard failure within a net-

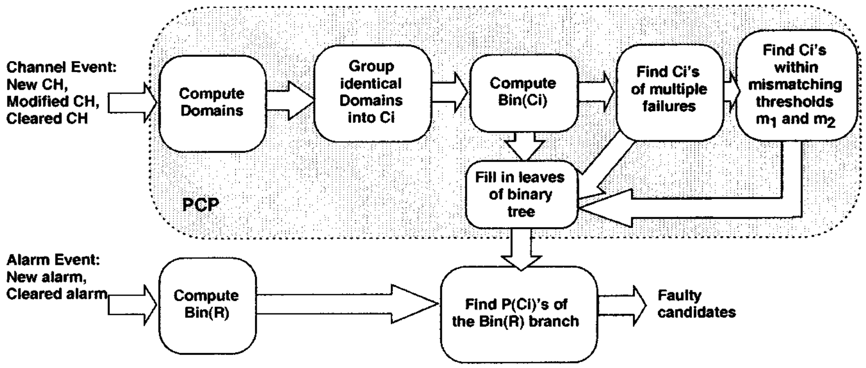

Channel Event:
New CH,
Modified CH,
Cleared CH
Compute Domains
Group identical Domains into Ci
Compute Bin(Ci)
Find Ci's of multiple failures
Find Ci's within mismatching thresholds m1 and m2
Fill in leaves of binary tree
PCP
Alarm Event:
New alarm,
Cleared alarm
Compute Bin(R)
Find P(Ci)'s of the Bin(R) branch
Faulty candidates

Fig. 2. FLA scheme: the precomputation phase (PCP) gathers most of the complexity and leaves few processing steps to be carried out in the FLA core.

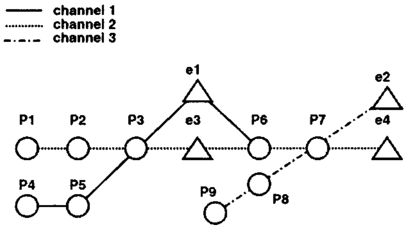


Fig. 3. A simple network example to introduce the algorithm.

work 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 only single, hard failures.

The steps of this algorithm are illustrated by the example of Fig. 3 where three channels have been established ($\mathcal{CH} = \{CH_1, CH_2, CH_3\}$). We denote by p the P components, and by e the $A2$ components. There are 9 passive elements $p_1, \cdots, p_9$, and 4 active elements which are e_1, e_2, e_3, and e_4. The PCP consists of the following modules.

1) Compute the domain of each element of the established channels. On the example you obtain:

$$\begin{array}{ll}
HDomain\{p_1\} = \{e_3, e_4\} & HDomain\{p_2\} = \{e_3, e_4\} \\
HDomain\{p_3\} = \{e_1, e_3, e_4\} & HDomain\{p_4\} = \{e_1\} \\
HDomain\{p_5\} = \{e_1\} & HDomain\{p_6\} = \{e_4\} \\
HDomain\{p_7\} = \{e_2, e_4\} & HDomain\{p_8\} = \{e_2\} \\
HDomain\{p_9\} = \{e_2\} & HDomain\{e_1\} = \emptyset \\
HDomain\{e_2\} = \emptyset & HDomain\{e_3\} = \{e_4\} \\
HDomain\{e_4\} = \emptyset. &
\end{array}$$

2) Group the identical domains into equivalence classes $C_1, C_2, \cdots, C_t$ ($t \leq n$). Here $t = 6$, and the classes are:

$$\begin{aligned}
C_1 &= HDomain\{p_1\} = HDomain\{p_2\} = \{e_3, e_4\} \\
C_2 &= HDomain\{p_3\} = \{e_1, e_3, e_4\} \\
C_3 &= HDomain\{p_4\} = HDomain\{p_5\} = \{e_1\} \\
C_4 &= HDomain\{p_6\} = HDomain\{e_3\} = \{e_4\} \\
C_5 &= HDomain\{p_7\} = \{e_2, e_4\} \\
C_6 &= HDomain\{p_8\} = HDomain\{p_9\} = \{e_2\}.
\end{aligned}$$

The domains of e_1, e_2, and e_4 are empty, so the failure of these elements cannot be detected.

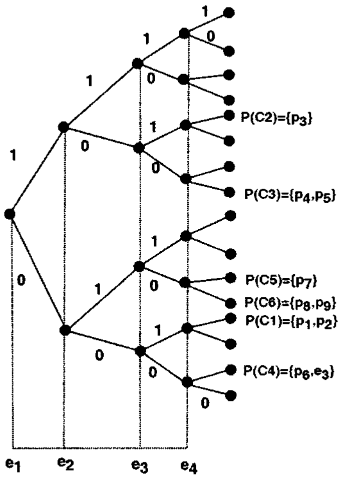


Fig. 4. Binary tree of the example presented in Fig. 3.

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. If we denote the number of alarming and monitoring components by n_a, the vectors $\vec{g}_i$ are therefore binary n_a-uples. The jth component of $\vec{g}_i = Bin(C_i)$ 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

$$\begin{array}{ll}
\vec{g}_1 = Bin(C_1) = (0\,0\,1\,1) & \vec{g}_2 = Bin(C_2) = (1\,0\,1\,1) \\
\vec{g}_3 = Bin(C_3) = (1\,0\,0\,0) & \vec{g}_4 = Bin(C_4) = (0\,0\,0\,1) \\
\vec{g}_5 = Bin(C_5) = (0\,1\,0\,1) & \vec{g}_6 = Bin(C_6) = (0\,1\,0\,0).
\end{array}$$

We will see later that these vectors generate a set $\mathcal{C}$ of binary vectors which can be regarded as a nonlinear code.

4) A binary tree is built with a depth equal to the number of active components and whose leaves correspond to different 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. Fig. 4 presents the binary tree of our example.

These steps can be precomputed 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 simple step has to be performed: $Bin(\mathcal{R})$ 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 only faulty candidate.

2) The Network has All Categories of Components (P, A1, A2, A3, M): The modules of the algorithm are the same as the ones in the previous section with the distinction that, for each component, two different domains *HDomain* and *SDomain* have to be computed based on (4) and (5). $P(C_i)$ will therefore contain two fields: the first is the set of elements whose Domain is C_i, the second is the indication of the nature (hard or soft) of the failure:

$$\begin{aligned} P(C_i) = \{&(comp, F) \text{ with } comp \in \mathcal{V} | FDomain(comp) = C_i \\ &\text{and } F = H \text{ or } S\}. \end{aligned} \tag{6}$$

B. FLA to Locate Multiple Failures in the Ideal Scenario

Let us now consider the case where several failures may happen in a short interval of time so that the alarms reaching the manager are intermingled. 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 $C_i = FDomain(e_i)$ and $C_j = FDomain(e_j)$, 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

$$\begin{aligned} \vec{x}_k &= Bin(C_k) = Bin(C_i \cup C_j) \\ &= Bin(C_i) \vee Bin(C_j) = \vec{g}_i \vee \vec{g}_j \end{aligned} \tag{7}$$

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 into the following equation:

$$P(C_i \cup C_j) = \{\{(e_i, F_i), (e_j, F_j)\}\}. \tag{8}$$

If (7) 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, no action is performed: 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 (7) returns a binary vector $\vec{x}_k$ different from any of the existing generator vectors $\vec{g}_1, \cdots, \vec{g}_t$, 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 $\vec{0} = \{0, \cdots, 0\}$, 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 $\vec{x}_k$,

$$\vec{x}_k = \vec{x}_i \vee \vec{x}_j \tag{9}$$

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,

$$\vec{x}_i \vee \vec{g}_k = \vec{x}_i \quad \text{or} \quad \vec{x}_i \vee \vec{g}_k = \vec{g}_k \tag{10}$$

then $\vec{g}_k$ will not contribute to any new leaf anymore. Therefore, it need 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. The output of this part of the algorithm in the example presented in Fig. 3 is the following:

$$\begin{aligned} Bin(C_1) \vee Bin(C_3) &= Bin(C_2) \\ Bin(C_1) \vee Bin(C_5) &= Bin(C_{11}) = (0\ 1\ 1\ 1) \\ Bin(C_1) \vee Bin(C_6) &= Bin(C_{11}) \\ Bin(C_2) \vee Bin(C_5) &= Bin(C_8) = (1\ 1\ 1\ 1) \\ Bin(C_2) \vee Bin(C_6) &= Bin(C_8) \\ Bin(C_3) \vee Bin(C_4) &= Bin(C_9) = (1\ 0\ 0\ 1) \\ Bin(C_3) \vee Bin(C_5) &= Bin(C_{10}) = (1\ 1\ 0\ 1) \\ Bin(C_3) \vee Bin(C_6) &= (C_{12}) = (1\ 1\ 0\ 0) \\ Bin(C_4) \vee Bin(C_6) &= Bin(C_5). \end{aligned}$$

These vectors correspond to double failures. Some are identical to those of single failures, but some others, with their binary vector between parenthesis, are new and fill new leaves of the binary tree, as shown in Fig. 5. Triple failures do not produce new binary vectors and, hence, no further steps are done.

C. FLA to Locate Multiple Failures in the Nonideal Scenario

Note that in the previous example, some leaves of the binary tree remain empty whatever 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, which 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. 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 $\vec{g}_1, \cdots, \vec{g}_t$ as these are dictated only by the network topology and the established optical channels.

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/she will prefer to get the set of

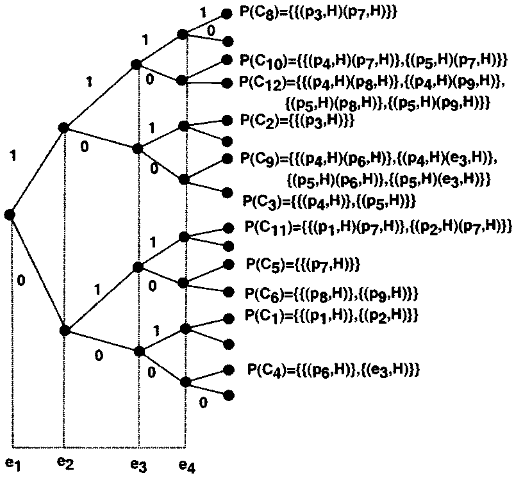

$P(C_8)=\{\{(p_3,H)(p_7,H)\}\}$
$P(C_{10})=\{\{(p_4,H)(p_7,H)\},\{(p_5,H)(p_7,H)\}\}$
$P(C_{12})=\{\{(p_4,H)(p_8,H)\},\{(p_4,H)(p_9,H)\},$
$\{(p_5,H)(p_8,H)\},\{(p_5,H)(p_9,H)\}\}$
$P(C_2)=\{\{(p_3,H)\}\}$
$P(C_9)=\{\{(p_4,H)(p_6,H)\},\{(p_4,H)(e_3,H)\},$
$\{(p_5,H)(p_6,H)\},\{(p_5,H)(e_3,H)\}\}$
$P(C_3)=\{\{(p_4,H)\},\{(p_5,H)\}\}$
$P(C_{11})=\{\{(p_1,H)(p_7,H)\},\{(p_2,H)(p_7,H)\}\}$
$P(C_5)=\{\{(p_7,H)\}\}$
$P(C_6)=\{\{(p_8,H)\},\{(p_9,H)\}\}$
$P(C_1)=\{\{(p_1,H)\},\{(p_2,H)\}\}$
$P(C_4)=\{\{(p_6,H)\},\{(e_3,H)\}\}$
1
0
e_1
e_2
e_3
e_4

Fig. 5. Binary tree of network shown in Fig. 3 with multiple failures.

all faulty 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 a maximum of m_1 alarms to be lost, and therefore the binary vectors that fall within this margin from the correct codewords are the binary vectors having "1" when $Bin(\mathcal{R})$ has "0" in 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" in at most m_2 positions.

Let us illustrate the algorithm with different scenarios on the example of Fig. 3 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 at Fig. 5. Let us check different scenarios.

- $m_1 = 1, m_2 = 0$: One lost alarm and no false alarms are tolerated. In this case [see Fig. 6(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.

- $m_1 = 0, m_2 = 1$: One false alarm and no lost alarms are tolerated. In this case [see Fig. 6(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 Fig. 7), the additional results to the previous ones are the new vectors $Bin(C_5) = (0\ 1\ 0\ 1)$ 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 ofp_6 and p_5 with 2 mismatches
  Failure of e_3 and p_5 with 2 mismatches.

  In this example, which has only four $A2$ components, the tolerance $m_1 = m_2 = 1$ is too loose because it amounts to accept up to 50% erroneous alarms, and leads to many $P(C_i)$ sets at each leaf of the tree. In a more realistic case, the number of active elements is much larger, and the number of sets $P(C_i)$ pointed by each leaf is much lower. Hence, the result is more selective.

D. Implementation and Algorithmic Complexity

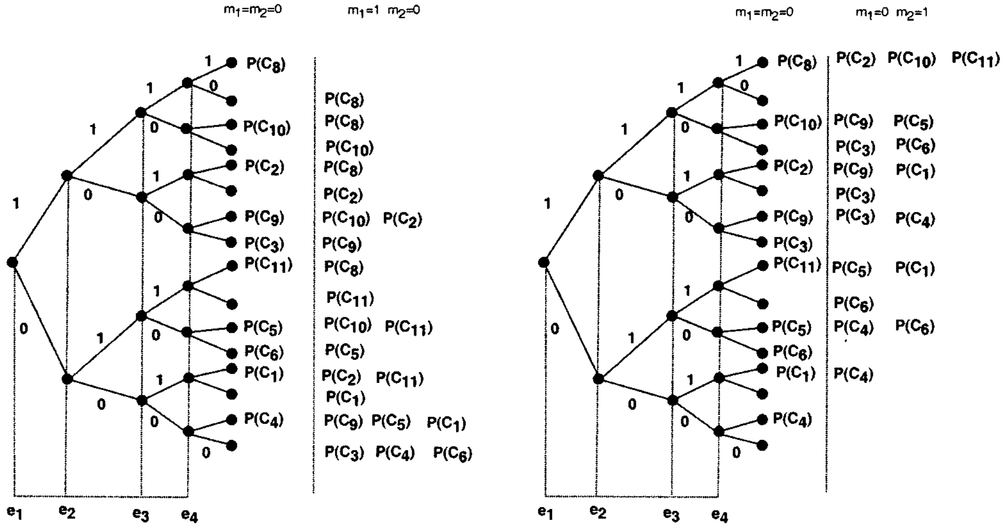
This algorithm has been implemented in Java and displays three windows: the input window, which allows entering the channels, alarms, and mismatching threshold; the result window, which displays the result of the algorithm, and the graphical window, which gives a graphical view of the established channels highlighting the elements that have issued alarms and the resulting faulty candidates [17].

Let us now briefly discuss the complexity of the FLA. A more detailed study can be found in [17]. Time complexity is critical for a fast diagnosis phase, whereas space complexity is critical if we have memory space limitations.

Let us begin with time complexity. As mentioned earlier, the problem of identifying multiple failures is NP complete [1]. However, the computationally intensive part is carried out off-line, in the precomputation phase (PCP), where all the codewords, including those accounting for nonideal scenarios, are determined. As a result, the computation time of the *on-line diagnosis part* is kept minimal, proportional to the number of alarming components n_a, whereas a bound on the worst case complexity of the PCP part is $O(4^{n_a})$ [17].

However, this minimal time complexity of the diagnosis phase comes at the expense of space complexity, as all codewords are explicitly computed and need to be stored. For the FLA, we have deliberately chosen to minimize time-complexity when alarms are received by the manager. One could have chosen the opposite approach, and to minimize space complexity. Indeed it might have been enough to store only the t generator codewords $\vec{g}_1, \cdots, \vec{g}_t$ corresponding to single failures, as all the binary vectors corresponding to multiple failures, in ideal and nonideal scenarios, are computed from these. However, remember that this computation is precisely the NP part of the algorithm which would now need to be carried out during the diagnosis phase, and no longer off-line in the PCP, resulting in worse time-complexity. An example of an algorithm that trades off time complexity for lower space complexity than the FLA is the alarm filtering algorithm described in [11].

A question remains open: is it possible to minimize both time and space complexities? We have already mentioned the



(a)

(b)

Fig. 6. Binary tree accepting mismatches. (a) Binary tree when $m_1 = 1$ and $m_2 = 0$. (b) Binary tree when $m_1 = 0$ and $m_2 = 1$.

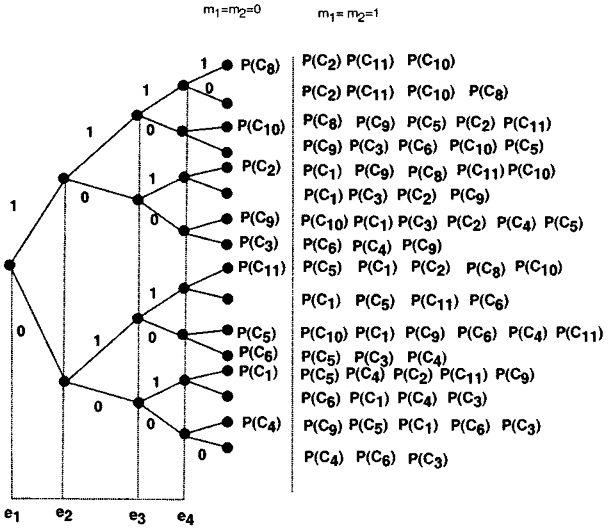

m1=m2=0
m1= m2=1
P(C8)
P(C10)
P(C2)
P(C9)
P(C3)
P(C11)
P(C5)
P(C6)
P(C1)
P(C4)
P(C2) P(C11) P(C10)
P(C2) P(C11) P(C10) P(C8)
P(C8) P(C9) P(C5) P(C2) P(C11)
P(C9) P(C3) P(C6) P(C10) P(C5)
P(C1) P(C9) P(C8) P(C11) P(C10)
P(C1) P(C3) P(C2) P(C9)
P(C10) P(C1) P(C3) P(C2) P(C4) P(C5)
P(C6) P(C4) P(C9)
P(C5) P(C1) P(C2) P(C8) P(C10)
P(C1) P(C5) P(C11) P(C6)
P(C10) P(C1) P(C9) P(C6) P(C4) P(C11)
P(C5) P(C3) P(C4)
P(C5) P(C4) P(C2) P(C11) P(C9)
P(C6) P(C1) P(C4) P(C3)
P(C9) P(C5) P(C1) P(C6) P(C3)
P(C4) P(C6) P(C3)
1
0
e1
e2
e3
e4

Fig. 7. Binary tree when $m_1 = m_2 = 1$.

analogy between the FLA and error-correcting codes. Linear error-correcting codes achieve fast error correction, i.e., have a good time-complexity, without requiring the computation of a binary tree, and so have also a good space complexity. This is achieved thanks to the fact that the set of codewords forming a linear code is a *vector space*. Here the set of codewords $\mathcal{C}$ does not enjoy this crucial property. It does, however, still possess a weaker algebraic structure [17], known as a *moduloid* in max-plus algebra [19]. Further research will address the possibility of making use of this structure for improving the space complexity while keeping the time complexity minimal.

VI. SDH/WDM Soft Failure Scenarios

We briefly show an example of an SDH/WDM network to illustrate how internetworking may help in locating a progressive failure, and how the difficulty of setting appropriate thresholds is (at least partially) alleviated by our algorithm ability to tolerate false and missing alarms.

We have modeled three SDH channels (shown in Fig. 8) using the abstraction of the hardware components and monitoring equipments of Section III-D. The SDH monitoring elements count erroroneous blocks over time windows of 15 min or 24 h. The error count is reset at each new time window. Whenever it crosses 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 [15].

We adopted here the assumption of Poisson distribution (similar results hold for bursty distribution), with two parameters $\lambda_1 < \lambda_2$, where λ_1 corresponds to the correct functioning (no failure), and is λ_2 the situation where a progressive failure has occurred. We have simulated the time series of the error counts registered at all the monitoring components (the ones of e_7, e_{10}, e_{28}, and e_{31} are shown in Fig. 9). The last two should indicate that a soft failure occurred during the fifth time window. 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. Running the algorithm with $m_1 = 1$ and $m_2 = 0$ yields the following output: $e_{26}, p_9,$ or e_{27} with 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. Running the algorithm with $m_1 = 0$ and $m_2 = 1$ gives the following output: e_{26}, p_9, or e_{27} with 1 mismatch.

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

Our algorithm can also be adapted to IP/WDM networks and can cope with soft failures by modeling the IP routers as $M1$ elements because they are not masked by any WDM monitoring equipment at the WDM layer, and they mask failures to the next

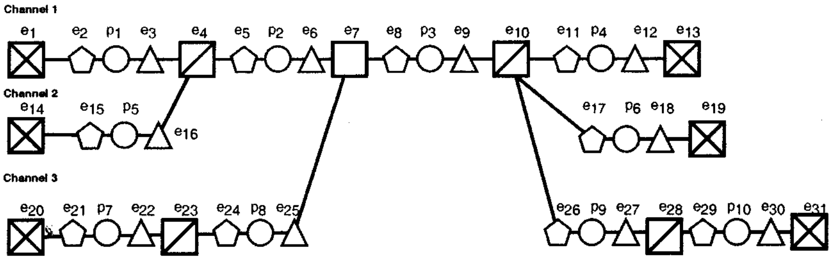
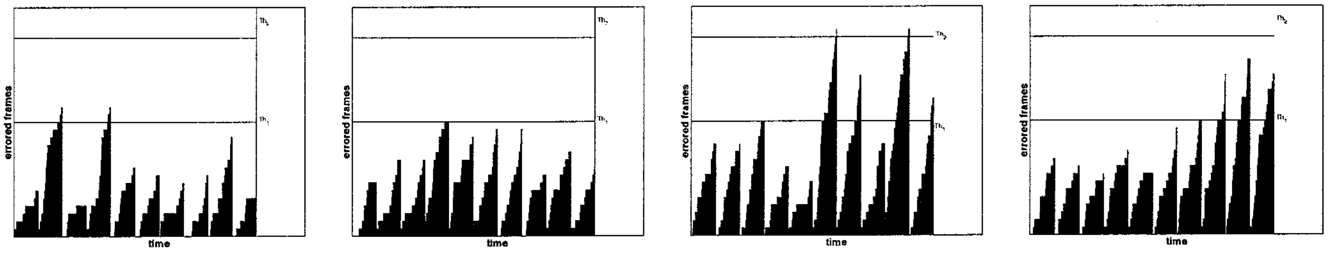

Channel 1
e1 e2 p1 e3 e4 e5 p2 e6 e7 e8 p3 e9 e10 e11 p4 e12 e13
Channel 2
e14 e15 p5 e16
e17 p6 e18 e19
Channel 3
e20 e21 p7 e22 e23 e24 p8 e25
e26 p9 e27 e28 e29 p10 e30 e31

Fig. 8. Modeling of the 3 SDH channels over WDM.



e_7 e_{10} e_{28} e_{31}

Fig. 9. Block error counting when errors follow a Poisson distribution.

IP router on the channel. An IP router discards packets that do not verify the checksum. The number of discarded packets is stored in the MIB variable *ifInErrors*.

VII. Conclusion

We have presented a fault location algorithm (FLA) able to detect multiple soft and hard failures in a WDM network. This algorithm improves the previous version implemented in the COBNET ring network [11].

First, most of the processing time of the second version of FLA developed in this paper is spent in a precomputation phase, so that the fault location when alarms are received by the manager is fast. This makes it particularly appropriate for large meshed networks, although the advantage for small ring networks like the ones of COBNET is less critical. The second attractive feature of our algorithm is the combination of 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.

Acknowledgment

The authors would like to thank Prof. J.-Y. Le Boudec (EPFL), Dr. A. McCabe (GEC Marconi), and H. Garcia (Lucent) for helpful discussions on SDH, as well as Dr. M. Zirngibl (Lucent) and Dr. L. Ping (Sprint ATL) for helpful discussions about error monitoring at the physical layer.

References

[1] N. S. V. Rao, “Computational complexity issues in operative diagnosis of graph-based systems,” *IEEE Trans. Computers*, vol. 42, pp. 447–457, Apr. 1993.

[2] Wandel and Goltermann, *Understanding Dense WDM*: Tech. Library, Apr. 1998.

[3] Ditech. The role of channel monitoring in DWDM networks. [Online]. Available: http://www.ditechcorp.com/Ditech/products/optical.products/application.notes/moon-001.htm

[4] N. Wauters *et al.*, “Survivability in a new pan-European carriers’ carrier network based on WDM and SDH technology: Current implementation and future requirements,” *IEEE Commun. Mag.*, vol. 37, pp. 63–69, Aug. 1999.

[5] K. Sato and S. Okamoto, “Photonic transport technologies to create robust backbone networks,” *IEEE Commun. Mag.*, vol. 37, pp. 63–69, Aug. 1999.

[6] A. T. Bouloutas and A. Finkel, “Alarm correlation and fault identification in communication networks,” *IEEE Trans. Commun.*, vol. 42, Feb./Mar./Apr. 1994.

[7] C. Mas *et al.*, “Fault location for optical networks,” in *Proc. All-Optical Networking: Architecture, Control Management Issues (SPIE’99)*, 1998, pp. 408–419.

[8] R. H. Deng, A. A. Lazar, and W. Wang, “A probabilistic approach to fault diagnosis in linear lightwave networks,” *IEEE J. Select. Areas Commun.*, vol. 11, pp. 1438–1448, Dec. 1993.

[9] A. T. Bouloutas *et al.*, “Fault identification using a FSM model with unreliable partially observed data sequences,” *IEEE Trans. Commun.*, vol. 41, pp. 1074–1083, July 1993.

[10] R. Gardner and D. Harle, “Alarm correlation and network fault resolution using Kohonen self-organizing map,” in *Proc. Globecom’97*, pp. 1398–1402.

[11] C. Mas, P. Thiran, and J.-Y. Le Boudec, “Fault location at the WDM layer,” *Photon. Network Commun.*, vol. 1, no. 3, pp. 235–255, Nov. 1999.

[12] B. Mukherjee, *Optical Communication Networks*. New York: McGraw-Hill, 1997.

[13] M. Sexton *et al.*, *Broadband Networking: ATM, SDH and SONET*. Norwood, MA: Artech House, 1997.

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

[15] *Characteristics of SDH Equipment Functional Blocks*, 1997. ITU-T Rec. G.783.

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

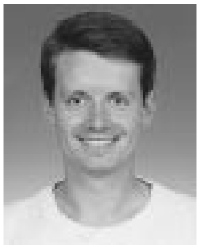
[17] C. Mas *et al.*, “Fault location algorithms for optical networks,” Ph.D. dissertation 2164, EPFL, 2000.

[18] N. S. V. Rao, “On parallel algorithms for single-fault diagnosis in fault propagation graph systems,” *IEEE Trans. Parallel Distributed Syst.*, vol. 7, pp. 1217–1223, Dec. 1996.

[19] F. Bacelli *et al.*, *Synchronization and Lineraity*. New York: Wiley, 1992.

Carmen Mas received the telecommunication engineering degree from the Universitat Politecnica de Catalunya (UPC), Barcelona, Spain, in 1995, and the Ph.D. degree from the Swiss Federal Institute of Technology at Lausanne (EPFL) in 2000.

In 1996, she became a Research Assistant at the Institute of Computer Communications and Applications, EPFL. She was a Teaching Assistant in several courses, and participated in the COBNET European project, where her contribution was the implementation of the management platform. After working with the Optical Systems and Network Group, National Technical University of Athens (NTUA), Athens, Greece, she is currently a Project Coordinator with Intracom, Athens.



Patrick Thiran (S'88–M'96) received the electrical engineering degree from the Universite Catholique de Louvain, Louvain-la-Neuve, Belgium, in 1989, the M.S. degree from the University of California at Berkeley in 1990, and the Ph.D. degree from the Swiss Federal Institute of Technology at Lausanne (EPFL) in 1996.

He joined the Institute of Computer Communications and their Applications of EPFL in 1996, where he was given the title of Professor in 1998. His research interests are in the fields of traffic control in communication networks, system theory, and neural networks. He is on leave from EPFL at Sprint Advanced Technology Labs, Burlingame, CA, until March 2001.

Dr. Thiran received the 1996 Doctoral Prize of EPFL, and he served as an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS, PART II, from 1997 until 1999.