Middle-East Journal of Scientific Research 6 (4): 324-328, 2010
ISSN 1990-9233

A Probabilistic Failure Localization Method in Optical WDM Networks

Mehdi Khani, Mohammad Ghasemzadeh, Fazlollah Adibnia and Mehdi Sarram

Department of Electrical and Computer Engineering, Yazd University, Yazd, Iran

Abstract: Optical fibre as the basic components of broad band optical networks are less vulnerable to electrical noise than normal network cables. The main drawback in using optical fibres is that they are very vulnerable to physical damages. Specially in DWDM networks, considering the high rate of data transfer, an interruption can cause a huge loss of information. This has made fault localization an important research topic in computer networking. In this paper we pre-sent a new technique which let us localize faults in optical DWDM networks effectively. It is very fast and very robust to alarm errors. The method is based on establishing normalized fault vectors and construction of components matrix. It also benefits from Probabilistic measures to overcome multiple simultaneous faults as well as false lost alarms. The processing and memory usage of the technique are also lower than previous existing methods.

Key words: Fault localization · Network · Optical fibre · WDM networks · Components

INTRODUCTION

Regarding the probable physical damages, optical fibre is more vulnerable than usual metallic cables but much less to electric fields and power loss. So they are widely used in long haul backbone net-works and faulty states, like fiber cuts seem to be inevitable. Although the high bandwidth is desirable in many fields, on a faulty state, the higher the bandwidth, the more loss of information bits in a single unit of time could happen. In an ordinary WDM network, a fibre break leads to interruption of hundreds of thousands of flow lines and loss of thousands of megabits of information [1]. So it is important to restore the faulty state very soon. Investigations has shown [2], from the three steps of restoration (Detection, Localization and Repair) the time needed for the second step completely dominates the other two steps.

It is pointed by [3] that almost 80% of faults in networks are caused by human or software geared by human. Therefore in the future, fast failure localization by an automated method is a critical need of any optical network. The main purpose of this paper is to introduce a very fast and applicable method to localize any fault in the optical WDM networks.

Hardware Optical Components: In an optical network, there are two main groups of hardware components, *ordinary components* and *monitoring components*. Some network components such as transmitters can be considered as a two-part set, each of which belonging to the corresponding group. More details about the characteristics of network components could be found in [2]. The main responsibilities of the above two groups of components are as follows:

Ordinary Components: These components are used to make a network perform properly. They are directly related to usual network tasks such as data transformation. Some of these components as listed in [4] are:

- Transmitters (Txs)
- Receivers (Rxs)
- Optical switches
- Amplifiers
- Optical regenerators/wavelength converters
- Couplers (Splitters/combiners)
- Optical filters
- Protection switches

Monitoring Components: These components are deployed in a WDM network to continuously check the network parameters. Each one is responsible to check a special parameter within its domain. The domain is the area on which the monitor-ing component has been placed. Because of hardware constraints, a monitoring component can only check the network within its domain. Some monitoring equipments in optical WDM networks are [4]:

Corresponding Author: Mohammad Ghasemzadeh, Yazd University, Iran, E-mail: m.ghasemzadeh@yazduni.ac.ir.
*This research work was partially supported by Iranian Telecommunication Research Center, Contract Nr. T/500/6257

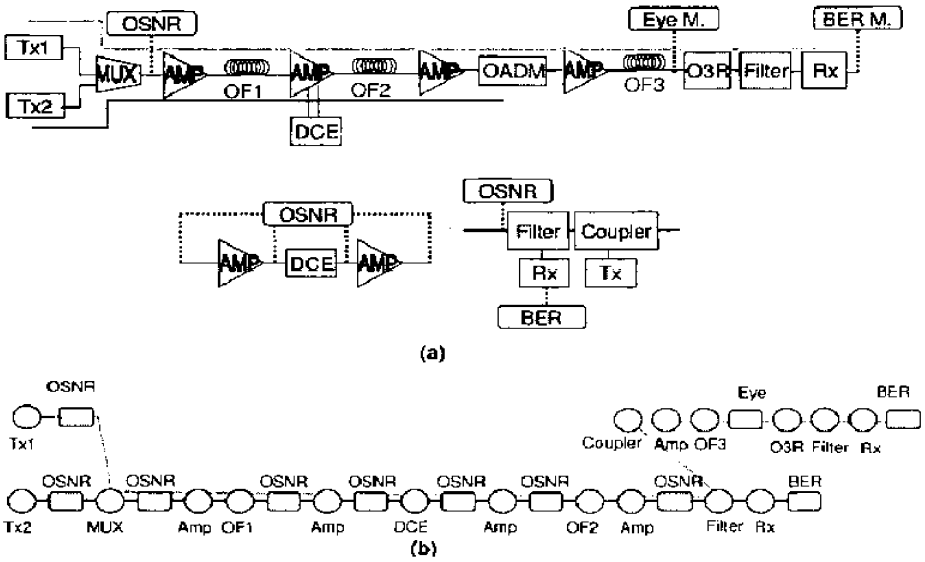

Fig. 1: A long haul optical WDM network map [4]

- Optical Power Meter
- Optical Spectrum Analyzer [5]
- Eye Monitoring [6, 7].
- BER Monitoring
- Wave-meter
- Pilot tones
- Optical Time Domain Reflectometry [8, 9].

Creating a Network Model: We suppose an optical network map is given, comprising of the network path graph, locations and types of optical network components and monitoring equipments. A sample of this kind of maps that is used by [4] is shown in Figure 1. In Fig. 1-a, components like transmitters, amplifiers, optical fibers, receiver and so on, can be seen as usual optical components of the networks and the other components such as OSNRs, eye monitoring and filters are monitoring equipments. Normally, if one of the optical components causes a problem (e.g. fiber cuts), a subset of monitoring equipments near the faulty component, will generate and propagate alarms in the network.

The alarms will be reported to the managerial center of the network. In Figure 1-b, the monitoring components are shown by rectangles and ordinary components are shown by circles. This is the network model needed by the managerial center to find the actual faulty component of the network. In order to clear the situation:

- The alarm domain of any component should be known by the managerial.
- Each of the monitoring equipments can be a member of one or more domains.
- There may be more than one faulty component at a time.

The point is to find the faulty component or components with the information of domains and alarms that are raised together. The general problem of finding these faulty components has shown to be NP-hard by [13]. Obviously there is no such time to use when the fault occurs. We show in the next section that by using some pre-processing in reasonable time, it is possible to find the faulty component, much faster.

The Algorithm: First, we introduce a definition for "Fault vector". A fault vector is a vector associated to every single component of the network. Each element of a vector can only get "0" or "1" values, but its value can change in other steps. All fault vectors have the same length which is equal to the number of monitoring elements in the network topology. We implement our algorithm according to the following steps:

Step1: We associate one vector to each vulnerable network component, but before that, we need to set a unique name to each network components in our network model. Figure 2 [4] shows the previous network we mentioned in figure1 with named components.

Step2: An alarm domain of the network part 'p', is the set of monitoring components which raise alarm when 'p' does not work properly [4]. The alarm domain for all network components of Figure 2 is shown in Figure 3 by [4].

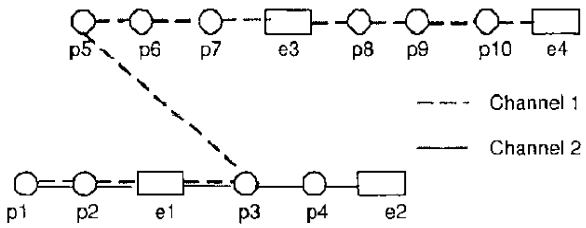

Fig. 2: A network model with named parts [4]

$$Domain(p_1) = \{e_1, e_2, e_3, e_4\}, Domain(p_2) = \{e_1, e_2, e_3, e_4\},$$
$$Domain(p_3) = \{e_2, e_3, e_4\}, Domain(p_4) = \{e_2\},$$
$$Domain(p_5) = \{e_3, e_4\}, Domain(p_6) = \{e_3, e_4\},$$
$$Domain(p_7) = \{e_3, e_4\}, Domain(p_8) = \{e_4\},$$
$$Domain(p_9) = \{e_4\}, Domain(p_{10}) = \{e4\}.$$

Fig. 3: The alarm domains for networks elements of Figure 2 [4].

$$C_1 = Domain(p_1) = Domain(p_2) = \{e_1, e_2, e_3, e_4\},$$
$$C_2 = Domain(p_3) = \{e_2, e_3, e_4\},$$
$$C_3 = Domain(p_5) = Domain(p_6) = Domain(p_7) = \{e_3, e_4\},$$
$$C_4 = Domain(p_4) = \{e_2\},$$
$$C_5 = Domain(p_8) = Domain(p_9) = Domain(p_{10}) = \{e_4\}.$$

Fig. 4: Component classes generated from network domains

Step3: A component class is a set of components which have the same alarm domains. It can be used to breakdown similar domains to reduce the alarm vector database size. Figure 4 indicates the alarm classes for the previous example.

Step4: With these classes, it is easy to create the primitive fault vector for all network vulnerable parts. The '*i*'th element in a primitive fault vector, assigned to a component class '*c*' is representing the alarm capability of monitoring element e_i, for the corruption of any member of class '*c*'. In fact it is assigned to all network parts which are members of '*c*'. A '1' basically means that it makes alarm in a fault situation for any member of '*c*' and a '*0*' shows that there will be no alarm generated in that conditions. The basic alarm vector for all component classes of previous example is:

$$v_1 = (1\ \ 1\ \ 1\ \ 1)$$
$$v_2 = (0\ \ 1\ \ 1\ \ 1)$$
$$v_3 = (0\ \ 0\ \ 1\ \ 1)$$
$$v_4 = (0\ \ 1\ \ 0\ \ 0)$$
$$v_5 = (0\ \ 0\ \ 0\ \ 1)$$

Step5: Considering the role of monitoring elements in the network topology, we find out that these elements are not of the same importance for any network part *p*. In other words, it is unfair to give the general amount '1', for all elements of all alarm vectors. To clear this, we should change the fault vector element values to make the sum of all '*i*'th fault vector elements, independent to '*i*'. It means that the alarm from monitoring element which rarely raises alarm is more important than the alarm raised from a monitoring element which usually raises alarm because of frequent failure situations. In this way, our algorithm will be more agile in situations with false alarms and lost alarms. First, we calculate the number of alarming situations for all monitoring elements which is sum of all 'i' values of fault vectors. We show these values in a vector called '*v*':

$$v = (1\ \ 2\ \ 3\ \ 4)$$

Next, we calculate the smallest common denominator for the four numbers and then multiply all the alarm vector elements of all component classes to the resulted which is 12 in this example, then we divide 'i'th elements of fault vectors by the '*i*'th element of vector '*v*'. The resulting fault vectors are called weighted fault vectors. The weighted fault vectors for the previous example are shown below:

$$v_1 = (12\ \ 4\ \ 4\ \ 3)$$
$$v_2 = (0\ \ 4\ \ 4\ \ 3)$$
$$v_3 = (0\ \ 0\ \ 4\ \ 3)$$
$$v_4 = (0\ \ 4\ \ 0\ \ 0)$$
$$v_5 = (0\ \ 0\ \ 0\ \ 3)$$

Using a maximum number as a threshold (which is shown by 'T') for non-breaking quiet periods, we can divide the running time of the network to some intervals. It means that the boundary between two time intervals will be the last moment of the first quiet period which is longer than the threshold happening after the beginning of the first interval. After finishing this quiet period, another time period begins and continues until another long quiet period appears. More details about the slicing method can be found in [2]. Using this method, any number of alarms which are temporally near each other, will be counted as a unique observation which can be resulted from a unique failure or a unique set of simultaneous failures. In other words, the fact that the probability of existence of causal relation (s) between alarms which are temporally related together is higher, has been taken to account. The raised alarms within each interval make a primitive alarm vector which is called the *situation vector*. The situation vector,

in other words, typically shows that which alarms are raised in that time interval. An example of situation vector could be like:

$$sv_t(1 \quad 0 \quad 0 \quad 1)$$

It is obvious that the situation vector can have errors, both in reception and time slicing phasees. Now, using cross products between 'sv_t' for any time t and any of the weighted fault vectors, the faulty component(s) can be identified. More precisely the result of cross product between 'sv_t' and fault vector 'vi', represents the probability of the corruption of the elements belonging to the component class 'c_i'.

The list of components can be prioritized by the calculated product and a certain number of the most probable faulty components reported as the result of the algorithm. The number of results is depended to the size of the network. We can see the result of cross product for the previous example below:

$$v1 = 15$$
$$v2 = 3$$
$$v3 = 3$$
$$v4 = 0$$
$$v5 = 3$$

The results show that most probably, a member of C_1 is failed. With less probability it is true for C_2, C_3, or C_5 and the probability of being faulty for the member(s) of C_4 is almost equal to zero.

Dealing with False/lost Alarms: There can be many reasons to yield a monitoring equipment to raise an alarm without a true failure. A good method should consider these mistakes and predict such situations. The represented method in this paper has some kinds of intelligence about false alarms. Because of weighted values, depending on the situation and the number of false alarms and lost alarms, it can handle a reasonable amount of mistakes in alarm generation. In the worse case, there could be some differences in the priority of faultinesses identified by the algorithm but it is not a total failure since the failed component may still be on the list. With enough monitoring components used against reasonably limited false or lost alarms, the effect of false or lost alarms could be reduced. More precisely we can say that the probability of finding the true failure(s) as first ones in the list, is depended on the mean value of the percentage of false or lost alarms to the total number of monitoring elements used in network topology. To find a measure for that probability first we must define a concept called 'Distance' between two vectors which means the number of zero elements in both vectors where there is no zero in the same position in the other vector. For example consider the two vectors v_1 and v_2:

$$v_1 = (1 \quad 1 \quad 1 \quad 1)$$
$$v_2 = (0 \quad 1 \quad 1 \quad 1)$$

The distance between v_1 and v_2 will be:

$$D(v_1,v_2) = 0 + 1 = 1$$

The mean distance between each two weighted fault vectors, supposed to be four times greater than the mean number of false or lost alarms. If this is the case, the probability of finding the true faulty component as the first one on the list, will be greater than 0.9, but if it did not happen, the probability of finding the true failure as the second one will be 0.9 and so on. So by expanding the result list we can increase the probability of finding the faulty component(s). In above case, with a list of 5 candidates, the probability of finding the faulty component in the list, will be greater than 0.99999 which is almost equal to 1. In short, the greater the minimum distance the more alarm errors, could be tolerated.

Dealing with Multiple Alarms: Using the productions introduced in step 5, the presented method can work in situations which multiple simultaneous faults, could happen. But the more simultaneous faults, should be taken to account, the more monitoring components should be used to keep the probability mistakes low. It is possible that the effect of simultaneous alarms be just like that the effect of false alarms in the system. It means that another failure, would raise some alarms which would not be raised if that failure did not exist. So the added alarms could be counted as false alarms for other failures which are simultaneous with this failure. But as we mentioned before by enlarging the list of candidates and/or increasing the mean distance between weighted fault vectors, this effects could be handled.

Algorithm Time Complexity: The algorithm runtime complexity, is almost equal to the time needed for cross products which is finally equal to $u*v$ where u is the number of network vulnerable parts and v is the number of monitoring components.

CONCLUSION

We introduced a new method with the name of Probabilistic Failure Location (PFL) to localize the faulty component (s) in Optical networks and especially useful in WDM networks due to their high bandwidth. Our algorithm, gives a prioritized list of component classes which are identified to be faulty. It works well with multiple fault situations and even with a reasonable amount of false or lost alarms.

REFERENCES

1. Metz, C., 2000. IP Protection and Restoration. IEEE Internet Computing, pages 97–102, March-April 2000 Metz.
2. Kompella. R.R., 2007. Fault localization in backbone networks. University of California. San Diego.
3. Bush, R. and D. Meyer, 2002. Some Internet architectural guidelines and philosophy. RFC, 3439, IETF.
4. Carmen Mas Machuca, H. Nguyen and Patrick Thiran, 2004. *Failure location in WDM network,* Kluwer Publishers.
5. Park, K.J., S.K. Shin and Y.C. Chung, 2000. A novel optical signal-to-noise ratio monitoring technique for WDM networks, *Proc. IEEE Optical Fiber Communication conference (OFC),* (Baltimore, MD), pp: 182-184.
6. Mueller, K., *et al*. 1998. Application of amplitude histograms for quality of service measurements of optical channels and fault identification, Proc. ECOC 1999, (Nice, France), Sept. 1999. Madrid, 1998.
7. Shake, I., H. Takara, S. Kawanishi and Y. Yamabayashi, 1998. Optical signal quality monitoring method based on optical sampling. *Electronic Letters,* 34(22):2152, October 1998.
8. Walker, M., 1999. *Cryptography and Coding*, In 8th IMA International Conference on Cryptography and Coding, Cirencester, United Kingdom. 1999.
9. Gardner, R. and D. Harle, 1997. Alarm Correlation and Network Fault Resolution using Koho-nen Self-Organising Map. Proc. of IEEE GLOBECOM, pp: 1398-1402.