# On Failure Localization in Optical Networks

**Elias A. Doumith**, **Sawsan Al Zahr**, **Ahmed Haddad**
Contact: elias.doumith@telecom-paristech.fr,

Institut TELECOM - TELECOM ParisTech - LTCI CNRS
Networks and Computer Sciences Department
46, rue Barrault
F 75634 Paris Cedex 13 - France

*Abstract*— **Recent advances in WDM technology enable an optical fiber to carry up to 200 wavelengths operating at 40 Gbps each. In such networks, service disruptions caused by network faults (e.g., fiber cut, amplifier dysfunction) may lead to high data losses. Therefore, it is mandatory for a network operator to be able to find such faults promptly. Fault detection and localization in meshed WDM networks have been deeply investigated in the literature. Numerous schemes have been proposed to achieve fast but accurate failure localization. In this field, four main paradigms have proved their mettle, namely link-based, cycle-based, trail-based, and tree-based monitoring schemes. In this paper, we provide a comparative study of the aforementioned monitoring schemes through numerical analysis.**

## I. INTRODUCTION

Optical transport networks evolve towards higher data rates and increased wavelength density in wavelength division multiplexing (WDM) systems. Optical component failures such as fiber cuts, amplifier dysfunctions, or laser frequency shifts can lead to a huge amount of data loss. In transparent and translucent optical networks, failure detection and localization is a challenging issue to operate dynamically reconfigurable networks with high reliability. Since failure recovery protocols are implemented at different layers, a failure event at the optical layer, such as a fiber cut, may also trigger alarms at upper layers [1]. An upper layer protocol generally requires a much longer detection time than an optical/physical layer protocol. Therefore, an intelligent and cost-effective monitoring mechanism dedicated to the network optical layer is mandatory.

Fault detection and localization in WDM meshed networks have been extensively addressed and many related studies have been reported in the literature. Most existing approaches [2]–[10] consist in deploying optical monitors responsible for generating alarms upon a single link failure. Monitoring information (*i.e.* alarms generated by the monitors) are then submitted to the control plane of the optical network so that any routing entity is able to localize the failure and to perform a real time traffic restoration. In the proposed approaches, dedicated supervisory channels are used for monitoring purposes at the detriment of operational lightpaths. In other terms, supervisory channels cannot carry real traffic. Such monitoring schemes are referred to as "*out-of-band monitoring*" as opposed to "*in-band monitoring*" where monitors are supervising operational lightpaths.

In-band monitoring [11], [12] has been proposed in order to reduce the overhead induced by its counterpart out-of-band monitoring in terms of the number of dedicated lightpaths, required transponders and monitors, dedicated bandwidth, lightpath provisioning and maintenance, etc. In-band monitoring techniques are capable of monitoring individual wavelengths on each fiber and may also allow for estimation of the channel's performance. In this context, the objective is to optimally deploy optical monitors that will supervise already provisioned lightpaths in order to uniquely localize each network element failure. However, such an approach highly depends on the traffic itself and may need to evolve with time.

Hybrid approaches can be proposed as a mid-way solution between out-of-band and in-band monitoring schemes. In such approaches, the objective is to jointly use existing operational lightpaths in addition to a minimum set of complementary out-of-band lightpaths in order to achieve unambiguous failure localization.

The major concern of all previous approaches is to minimize the monitoring cost while achieving an unambiguous failure localization. For in-band monitoring schemes, the monitoring cost only accounts for the number of required optical monitors. For hybrid and out-of-band monitoring schemes, the previous cost is augmented by the number of required laser diodes as well as the number of required supervisory channels.

In this paper, we conduct a complete survey of recent researches dealing with out-of-band monitoring schemes for fault detection and localization in WDM networks. From this survey, we can clearly identify four main paradigms, namely conventional link-based monitoring (*m*-links), cycle-based monitoring (*m*-cycles), trail-based monitoring (*m*-trails), and tree-based monitoring (*m*-trees). Sections II-V introduce the basic principle of each paradigm as well as the main approaches proposed to design such monitoring schemes. Section VI is intended to present a comparative study of these schemes, while Section VII recaps the main ideas addressed in this paper and presents our conclusions.

## II. LINK-BASED MONITORING

In conventional link-based monitoring scheme, the position of the laser diodes/optical monitors is straightforward. Each fiber-link is equipped with a laser diode and an optical monitor at each of its ends, respectively. Moreover, an optical supervisory channel is reserved on each link in order to detect any failure occurring on that link. Consequently, this approach is able to detect and locate without any ambiguity any single link failure as well as multiple link failures in the network. Such a monitoring scheme costs $|E|$ laser diodes, $|E|$ optical monitors, and $|E|$ supervisory channels, where $|E|$ denotes the number of network links. For the network topology shown in Figure 1, a link-based monitoring solution requires 7 laser diodes, 7 monitors, and 7 supervisory channels. On the left-hand-side of Figure 1, we can see the monitoring information associated to each single link's failure.

To sum up, a link-based monitoring scheme requires one laser diode and one monitor per link and consumes one optical supervisory channel on each link of the network. Although this approach consumes the theoretical minimum number of supervisory channels, it consumes an excessive number of laser diodes and optical monitors which makes it less attractive for large networks.

|   | $\ell_a$ | $\ell_b$ | $\ell_c$ | $\ell_d$ | $\ell_e$ | $\ell_f$ | $\ell_f$ |
|---|---|---|---|---|---|---|---|
| **a** | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| **b** | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| **c** | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| **d** | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| **e** | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| **f** | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| **g** | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

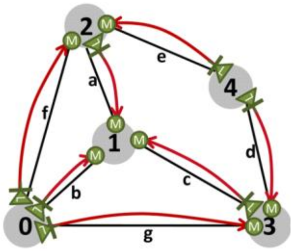

Fig. 1. Link-based monitoring scheme.

More sophisticated approaches aim at reducing the number of monitors in the network while achieving unambiguous failure localization. In the late 2000s, multiple paradigms for failure detection and localization have been proposed, namely monitoring cycles (*m*-cycles), monitoring trails (*m*-trails), and monitoring trees (*m*-trees).

## III. MONITORING CYCLES

In [2], the authors propose a monitoring scheme based on decomposing the transparent network into a set of cycles so that all nodes and links in the network appear in at least one of these cycles as shown in Figure 2.

|   | $c_1$ | $c_2$ | $c_3$ |
|---|---|---|---|
| **a** | 0 | 1 | 1 |
| **b** | 1 | 1 | 0 |
| **c** | 1 | 0 | 1 |
| **d** | 0 | 0 | 1 |
| **e** | 0 | 0 | 1 |
| **f** | 0 | 1 | 0 |
| **g** | 1 | 0 | 0 |

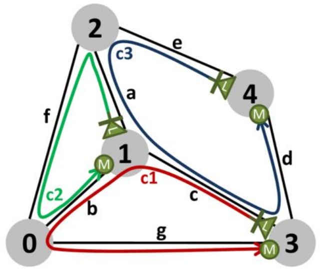

Fig. 2. *m*-cycle based monitoring scheme for fast link failure localization.

An *m*-cycle is defined as a loopback connection associated with a laser diode-optical monitor pair. A supervisory optical signal is transmitted along the cycle consuming one optical channel on each link it traverses. If a failure occurs on a link, the supervisory optical signal will be disrupted and an alarm is generated by its associated monitor.

A monitoring scheme based on the concept of *m*-cycles generally consists of $M$ *m*-cycles $\{c_1, c_2, c_3, \dots, c_M\}$ that cover every link of the considered network. Upon a single link failure, monitors associated to the cycles traversing that link will generate alarms. Monitoring information produce an alarm code $[a_1, a_2, \dots, a_M]$, where:

$$a_j = \begin{cases} 1, & \text{if cycle } c_j \text{ traverses the failed link;} \\ 0, & \text{otherwise.} \end{cases}$$

Figure 2 presents a monitoring scheme composed of three *m*-cycles $\{c_1, c_2, c_3\}$. If '*link a*' fails, the monitors associated to cycles $c_2$ and $c_3$ will generate alarms and the monitoring information produce the alarm code $[0, 1, 1]$. Similarly, if '*link b*' fails, the alarm code $[1, 1, 0]$ will be generated. All possible alarm codes are summarized on the left-hand-side of Figure 2. From this table, we notice that '*link d*' and '*link e*' failures generate the same alarm code. Consequently, when receiving the alarm code $[0, 0, 1]$, network operator is not able to precisely localize the failed link. One solution to eliminate such an ambiguity consists in using an additional link-based monitor either for '*link d*' or '*link e*' which increases the number of optical monitors as well as the number of required supervisory channels. However, the number of required monitors remains less than the 7 monitors required in a pure link-based monitoring scheme. In summary, the *m*-cycle monitoring scheme presented in Figure 2 costs 4 monitors and 11 optical channels.

In [2]–[4], three algorithms have been proposed to construct *m*-cycles, namely heuristic depth first searching (HDFS), shortest path Eulerian matching (SPEM), and heuristic spanning tree (HST). Given a network topology, these algorithms find a set of *m*-cycles that enable to localize any failure[1] in the network while minimizing the network resources' consumption (*i.e.* optical monitors, optical channels). Based on a carefully designed spanning tree, the HST algorithm yields the best performance in terms of localization degree and number of

---

[1] It worth noting that if any ambiguity exists, a link-based monitoring is considered for the ambiguous failure.

required wavelengths even though it introduces a larger number of monitors.

Proposed in [5], $m^2$-cycles present a more efficient mechanism for link failure localization. Minimum length $m$-cycles ($m^2$-cycles) are constructed in order to consume the minimum amount of network resources and to achieve the most accurate link failure detection and localization. It has been proved in [5] that $m^2$-cycle outperforms any spanning tree-based approach, no matter how the spanning tree is constructed. Moreover, numerical results show that $m^2$-cycles require much less network resources than HST.

In summary, the *m*-cycles have been proposed with the objective to reduce the number of required laser diodes and optical monitors, and subsequently reducing the network monitoring cost. An *m*-cycle is a loop-back optical connection using a supervisory optical channel on each link it traverses, with a laser diode and an optical monitor placed back to back at any node along the loop. However, the major drawback of the *m*-cycles is their inability to distinguish in some cases between single link failures occurring on different links. These links usually belong to the same network segments; a segment is defined as a path made of at least two consecutive links where intermediate nodes have a nodal degree of two. In order to localize each link failure without any ambiguity, extra link-based monitors are required.

## IV. MONITORING TRAILS

As stated previously, an *m*-cycle monitoring scheme may not be able to distinguish between the failure of the different links of the same network segment. Consequently, a loopback monitoring scheme is not the best suited approach for network topologies containing segments.

|  | $t_1$ | $t_2$ | $t_3$ |
|---|---|---|---|
| **a** | 0 | 1 | 0 |
| **b** | 1 | 0 | 0 |
| **c** | 1 | 1 | 0 |
| **d** | 1 | 0 | 1 |
| **e** | 0 | 0 | 1 |
| **f** | 0 | 1 | 1 |
| **g** | 1 | 1 | 1 |

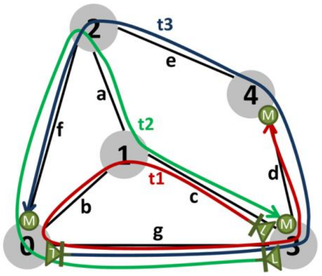

Fig. 3. *m*-trails monitoring scheme for fast link failure localization.

In order to cope with ambiguity, the concept of monitoring trails (*m*-trails) [6]–[9] has been introduced. *M*-trails break the structure of the cycle by assuming that the laser diode and the optical monitor are not necessarily collocated at the same node as shown in Figure 3. Although the cycle structure constraint is removed, an *m*-trail works exactly in the same way as an *m*-cycle for fast link failure localization. Moreover, an *m*-trail may be an *m*-cycle or a non-simple *m*-cycle (*i.e.* a non-simple *m*-cycle refers to an *m*-cycle that traverses the same node multiple times).

Similar to a non-simple *m*-cycle, an *m*-trail can traverse a node multiple times but it traverses a link at most once. From Figure 3, we can see that an *m*-trail monitoring scheme is able to associate to each link in the network a unique alarm code and thus, to unambiguously locate any single link failure occurring in the network. In the considered example, the *m*-trails consume 3 laser diodes, 3 optical monitors, and 12 optical channels.

The original approach proposed in [6] for *m*-trails design is based on an integer linear program (ILP) formulation. In this formulation, the number of trails is allowed to vary within a given range where the theoretical minimum number of trails is given by:

$$m = \left\lceil \log_2(|E| + 1) \right\rceil$$

However, the number of supervisory optical channels is larger than $|E|$, and the theoretical minimum number of optical channels $B$ can be computed for a given number $k (k \leq m)$ of *m*-trails as follows:

$$B = \sum_{\substack{i \geq 1 \\ R_i > 0}} i \times \min(R_{i-1}, C_k^i)$$

where $R_0 = |E|$, and $R_i = R_{i-1} - C_k^i,\ i \geq 1$.

One optimal *m*-trail solution minimizes the monitoring cost (*i.e.* number of laser diode-optical monitor pairs and number of supervisory channels) while guaranteeing an unambiguous failure localization. In other terms, the aim of the ILP is to minimize the number of required *m*-trails as well as their lengths subject to a unique alarm code per link. Numerical results reported in [6],[8] show that *m*-trail-based monitoring schemes can significantly cut down the monitoring cost compared to the *m*-cycle-based monitoring schemes and to pure link-based monitoring schemes.

In order to cope with the execution time of the ILP formulation, the authors in [7] proposed a heuristic approach for the design of *m*-trail solutions based on the random code assignment-random code swapping (RCA-RCS) concept. The proposed algorithm starts by assigning to each link in the network a unique alarm code (RCA). Such an initial *m*-trail solution is characterized by a null ambiguity but the trails are not necessarily valid since they may contain many isolated segments and a large number of odd-degree nodes. Then, the algorithm implements a greedy approach for swapping alarm codes (RCS) in order to help interconnecting isolated trail segments, reducing the numbers of odd-degree nodes, and minimizing the number of required supervisory channels.

Simulation results carried out in [9] show that designing an *m*-trail solution through RCA-RCS strongly depends on the initial conditions. Indeed, RCS is only able to modify an *m*-trail by adding and/or removing a single link at a time. Thus, the final solutions can be relatively far from optimal in the case of large networks. In addition, the authors propose an original meta-heuristic approach referred to as meta-heuristic for monitoring trail assignment (MeMoTA) algorithm. As opposed to RCA-RCS, MeMoTA deals always with valid *m*-trails and tries to reshape these trails in order to eliminate in a first step the ambiguity, then to reduce the monitoring cost. The proposed approach is based on the Tabu-Search algorithm and tries to iteratively improve a given *m*-trail solution. Numerical results show that MeMoTA provides near-optimal solutions with much shorter computing delays than the ILP formulation. In addition, unlike RCA-RCS, MeMoTA is less sensitive to initial conditions.

Concisely, the *m*-trails have been proposed as an alternative for the *m*-cycles with the objective to localize without any ambiguity any single link failure while still reducing the number of required laser diodes and optical monitors. An *m*-trail works in the same way as an *m*-cycle, but the optical connection of the supervisory channels does not necessarily need to be a loop. Thus, the laser diode and the optical monitor are not necessarily collocated together to maintain the cycle structure. As a result, both link-based and *m*-cycles monitoring are special cases of *m*-trails. The former approach is similar to the *m*-trails approach where all the trails are composed of a single supervisory channel. The latter approach is similar to the *m*-trails approach where all the trails have a loop shape. It should be noted that the lower the number of laser diodes and optical monitors deployed in the network, the higher the number of optical supervisory channels required for unambiguous failure detection. Thus, the *m*-trail approach tries to find a tradeoff between the cost penalty due to the additional number of supervisory channels and the cost benefit due to the reduction of the number of laser diodes and optical monitors.

## V. MONITORING TREES

One of the limitations of the *m*-trails is inherent to the fact they consume optical channels in the C-band at the detriment of operational lightpaths. Since wavelength resources are scare in optical networks, the authors in [10] propose a novel approach for fast link failure localization referred to as "*monitoring-trees*" (*m*-trees). The concept of *m*-trees makes use of the broadcasting capability within a network node. Broadcast implies that an optical signal passing through a node can be duplicated and forwarded over two or more outgoing fibers. This functionality is highly available in current WDM networks. Indeed, commonly used fabrics are based on wavelength selective switch (WSS) technology enabling broadcast and select architecture. Such switch fabrics can provide multicasting and broadcasting facilities for every input channel in a truly non-blocking manner.

As opposed to the *m*-trails which may use multiple optical supervisory channels per link on different wavelengths, the *m*-trees use of a single optical channel per link. Moreover, as the signal duplication is performed in the optical domain, the optical supervisory signal is carried by the same wavelength on all the network links. This does not only reduce the blocking ratio in the network due to the lack of network resources, but also reduces the blocking ratio due to the wavelength continuity constraint.

In the *m*-tree approach, a single laser diode is usually sufficient to monitor all the network. This laser diode is placed at a node and is transmitting its supervisory signal over a single link referred to as the "*head of the tree*". Arriving at a node along an input link, the supervisory signal can be terminated at the node, forwarded over a single outgoing link, or duplicated and sent over two or more outgoing links. By definition, a supervisory signal terminated at a node should be monitored at that node. Moreover, one may choose to monitor the supervisory signal at different locations in the network in order to be able to distinguish between different single link failures. A link with a monitor deployed at its end is referred to as a "*leaf of the tree*". In short, for a network composed of $|V|$ nodes and $|E|$ links, the *m*-tree approach requires a single laser diode, $|E|$ optical supervisory channels, and less than $|E|$ optical monitors in order to localize without any ambiguity any single link failure in the network. It should be noted that the number of supervisory channels required by the *m*-trees is equal to the number of supervisory channels required by the link-based monitoring approach which corresponds to the theoretical minimum number of supervisory channels required for an unambiguous failure localization.

| | $m_b$ | $m_d$ | $m_f$ | $m_g$ |
|---|---|---|---|---|
| **a** | 1 | 1 | 0 | 1 |
| **b** | 1 | 0 | 0 | 0 |
| **c** | 0 | 1 | 0 | 1 |
| **d** | 0 | 1 | 0 | 0 |
| **e** | 1 | 1 | 1 | 1 |
| **f** | 0 | 0 | 1 | 0 |
| **g** | 0 | 0 | 0 | 1 |

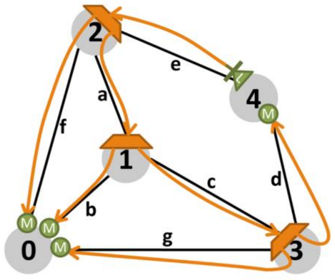

Fig. 4. *m*-tree monitoring scheme: a small example.

Let us consider a small example to provide a deeper insight into the *m*-trees concept (*cf*. Figure 4). A possible *m*-tree solution would consist in placing the laser diode at *'node 4'*. The supervisory signal generated at *'node 4'* is transmitted along *'link e'* towards *'node 2'*. At *'node 2'*, the supervisory signal is duplicated and sent towards *'node 0'* and *'node 1'* along *'link f'* and *'link a'*, respectively. At *'node 0'*, the supervisory signal along *'link f'* is terminated by a monitor which can detect any failure that occurs on any of *'link f'* and *'link e'*. The supervisory signal arriving at *'node 1'* is duplicated and sent towards *'node 0'* and *'node 3'* along *'link b'* and *'link c'*, respectively. At *'node 0'*, the supervisory signal along *'link b'* is terminated by a monitor which can detect any failure that occurs on any of *'link b'*, *'link a'*, and *'link e'*. Finally, the supervisory signal arriving at *'node 3'* is duplicated and sent towards *'node 0'* and *'node 4'* where these signals are terminated by two monitors. The monitor supervising *'link g'* at *'node 0'* can detect any failure that occurs on any of *'link g'*, *'link c'*, *'link a'*, and *'link e'*, while the monitor supervising *'link d'* at *'node 4'* can detect any failure that occurs on any of *'link d'*, *'link c'*, *'link a'*, and *'link e'*. In this example, *'link e'* corresponds to the head of the *m*-tree while *'link b'*, *'link d'*, *'link f'*, and *'link g'* are the leaves of the *m*-tree. In Figure 4, we have represented the *m*-tree solution as well as a table summarizing the links that are supervised by each monitor and the monitors that are alerted for each link failure.

It has been shown in [10] that the optimal *m*-tree solution consists in forwarding at each "*non-leaf*" node the incoming supervisory signal along only two outgoing links. Therefore, the optimal monitoring tree with the lowest number of optical monitors is a binary tree that duplicates, as much as possible, the supervisory signal into two copies whenever the supervisory signal passes through a node. It is shown in [13] that a binary tree composed of $n$ ($n$ odd) branches has $(n+1)/2$ leaves. Thus, the minimum number of optical monitors required to monitor a network composed of $|V|$ nodes and $|E|$ links is equal to $\lceil(|E|+1)/2\rceil$. The *m*-tree design problem has been formulated as an ILP and solved using linear solvers.

## VI. COMPARATIVE STUDY

In the following, we compare all the monitoring approaches that achieve unambiguous failure localization, namely *m*-link, *m*-trail, and *m*-tree monitoring schemes. To this end, we considered the Deutsche Telekom (DT) network composed of $|V| = 14$ nodes and $|E| = 23$ bi-directional links. For such a network, the theoretical minimum number of trails required to detect any single link failure without ambiguity is equal to 5. Consequently, the theoretical minimum number of supervisory channels is equal to 49. According to the current optical equipment market, the cost of a laser diode is equal to the cost of an optical monitor which is 2.5 times more expensive than the cost of a supervisory channel. Under this assumption, the optimal solution using 5 *m*-trails corresponds to a monitoring cost equal to 74. Figure 5 depicts these trails.

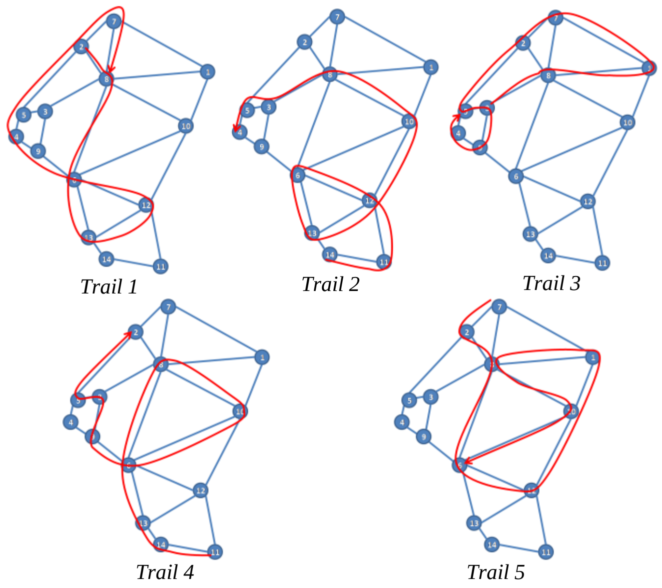
*Trail 1* *Trail 2* *Trail 3* *Trail 4* *Trail 5*

Fig. 5. Optimal *m*-trail solution for the Deutsch Telekom network.

For the same topology, the *m*-tree approach consumes a single laser diode and 23 supervisory channels. The theoretical minimum number of optical monitors required for an unambiguous detection and localization is equal to 12. However, the optimal *m*-tree solution is composed of 13 optical monitors (*cf.* Figure 6). This solution corresponds to a monitoring cost of 58. Table I summarizes the network resources occupied by the three compared approaches as well as their monitoring costs.

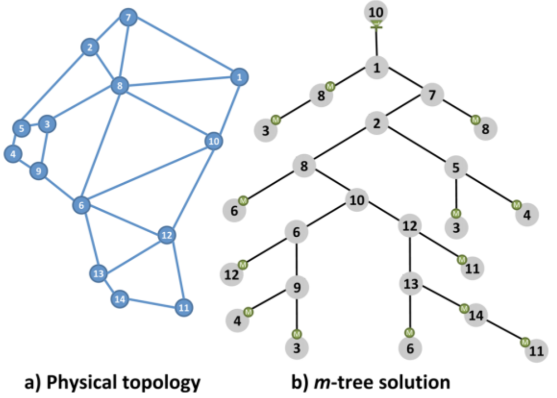
**a) Physical topology** **b) *m*-tree solution**

Fig. 6. Optimal *m*-tree solution for the Deutsche Telekom network.

TABLE I
3 CONCURRENT MONITORING SCHEMES FOR THE DT NETWORK

| | ***m*-link** | ***m*-trail** | ***m*-tree** |
|---|---|---|---|
| Number of supervisory channels | 23 | 49 | 23 |
| Number of laser diodes | 23 | 5 | 1 |
| Number of optical monitors | 23 | 5 | 13 |
| Monitoring Cost | 138 | 74 | 58 |

Consequently, the *m*-tree approach is less expensive than its counterparts while providing the same unambiguous failure detection and localization. It is worth noting that the *m*-tree approach remains economically more beneficial than the *m*-trail approach up to a ratio of 6.5 between the cost of an optical monitor and the cost of a supervisory channel. Moreover, we can note that the *m*-tree approach consumes less optical supervisory channels than the *m*-trail approach and thus, saves network resources for real traffic demands.

## VII. CONCLUSION

The design of efficient and cost-effective fault identification schemes is of paramount importance to achieve reliability in optical networks. In this paper, we conducted a complete survey on monitoring techniques that can be applied in such networks. We elaborated on 4 out-of-band monitoring schemes, namely link-based (*m*-links), cycle-based (*m*-cycles), trail-based (*m*-trails), and tree-based (*m*-trees) monitoring. These approaches have been compared in terms of monitoring cost expressed as a function of the number of laser diodes, optical monitors, and supervisory channels. We can conclude that the *m*-tree approach enables a gain in terms of monitoring cost of around 58% and 22% compared to the *m*-link and the *m*-trail approaches, respectively.

## REFERENCES

[1] P. Demeester, *et al.*, "Resilience in multilayer networks," *IEEE Communications Magazine*, vol. 37, no. 8, pp. 70-76, Aug. 1999.

[2] H. Zeng, C. Huang, A. Vukovic, and M. Savoie, “Fault detection and path performance monitoring in meshed all-optical networks, ” *Proc. of IEEE Globecom*, Nov. 2004, vol. 3, pp. 2014-2018.

[3] H. Zeng, C. Huang, and A. Vukovic, “Spanning-tree based monitoring-cycle construction for fault detection and localization in meshed all-optical networks,” *Proc. of IEEE ICC*, May 2005, vol. 3, pp. 1726-1730.

[4] H. Zeng, C. Huang, and A. Vukovic, “A novel fault detection and localization scheme for mesh all-optical networks based on monitoring cycles,” *Photon Network Communications*, vol. 11, no. 3, pp. 277-286, May 2006.

[5] B. Wu and K. L. Yeung, “$M^2$-Cycle: an optical layer algorithm for fast link failure detection in all-optical mesh networks,” *Proc. of IEEE Globecom*, Nov. 2006, pp. 1-5.

[6] B. Wu, P. H. Ho, and K. L. Yeung, “Monitoring Trail: a new paradigm for fast link failure localization in WDM mesh networks,” *Proc. of IEEE Globecom*, Nov. 2008, pp. 2709-2713.

[7] J. Tapolcai, B. Wu, P. H. Ho, “On monitoring and failure localization in mesh all-optical networks, ” *Proc. IEEE Infocom*, Apr. 2009, pp. 1008-1016.

[8] B. Wu, P. H. Ho, and K. L. Yeung, “Monitoring Trail: on fast link failure localization in all-optical mesh networks,” *IEEE/OSA Journal of Lightwave Technology*, vol. 27, no. 18, pp. 4175-4185, Sep. 2009.

[9] A. Haddad, E. A. Doumith, and M. Gagnaire. “A meta-heuristic approach for monitoring trail assignment in WDM optical networks,” *to appear in Proc. of the IFIP/IEEE RNDM*, Moscow, Russia, Oct. 2010.

[10] E. A. Doumith, S. Al Zahr, and M. Gagnaire. “Monitoring-tree: An innovative technique for failure localization in WDM translucent networks,” *to appear in Proc. of the IEEE Globecom*, Miami, USA, Dec. 2010.

[11] S. Stanic, G. Sahin, H. Choi, S. Subramaniam, and H.-A. Choi, “Monitoring and alarm management in transparent optical networks”, *Proc. IEEE Broadnets*, Sept. 2007, pp.828-836.

[12] S. Stanic, S. Subramaniam, H. Choi, G. Sahin, and H.-A. Choi, “On monitoring transparent optical networks,” *Proc. International Conference on Parallel Processing*, Aug. 2002, pp. 217-223.

[13] D. P. Mehta and S. Sahni, “Handbook of data structures and applications,” Chapman & Hall/Crc Computer and Information Science Series, Jan. 2005.