Monitoring Trail: On Fast Link Failure Localization in All-Optical WDM Mesh Networks

Bin Wu, *Member, IEEE*, Pin-Han Ho, and Kwan L. Yeung, *Senior Member, IEEE*

Abstract—We consider an optical layer monitoring mechanism for fast link failure localization in all-optical wavelength-division-multiplexing (WDM) mesh networks. A novel framework of all-optical monitoring, called monitoring trail (m-trail), is introduced. It differs from the existing monitoring cycle (m-cycle) method by removing the cycle constraint. As a result, m-trail provides a general all-optical monitoring structure, which includes simple, nonsimple m-cycles, and open trails as special cases. Based on an in-depth theoretical analysis, we formulate an efficient integer linear program (ILP) for m-trail design to achieve unambiguous localization of each link failure. The objective is to minimize the monitoring cost (i.e., monitor cost plus bandwidth cost) of all m-trails in the solution. Numerical results show that the proposed m-trail scheme significantly outperforms its m-cycle-based counterpart.

Index Terms—Fast link failure localization, integer linear program (ILP), monitoring trail (m-trail), wavelength-division multiplexing (WDM).

I. INTRODUCTION

OPTICAL networks evolve toward increased transparency in the data plane and more intelligence in the control plane. Compared with conventional opaque networks, all-optical networks remove the electronic bottleneck. This not only reduces network cost, but also increases data transmission rate with much better quality of service (QoS) performance. However, it is extremely challenging to operate a dynamically reconfigurable all-optical network with high reliability and cost-efficiency [1], [2]. With wavelength-division-multiplexing (WDM) technology, a single fiber can carry hundreds of wavelengths, each working at 40 Gb/s [3] or higher data rate. On the other hand, optical networks are vulnerable to component failures such as fiber cuts. Due to the high speed nature of optical networks, a component failure can lead to a huge amount of data loss. To minimize data loss, it is important that the failed component can be immediately localized and bypassed. But it is generally not easy to accurately localize the failure in an instantaneous manner [2].

In all-optical networks, failure detection and localization is more complex than that in opaque networks [4]. Due to the lack of optoelectronic regenerators, the impact of a failure

propagates without electronic boundary, and a single failure can trigger a large number of redundant alarms. Meanwhile, protocols at different network layers may have their own failure management mechanisms. For example, routing protocols, such as open shortest path first (OSPF) and intermediate system–intermediate system (IS–IS), have built-in failure management functionality [5]. A failure at the optical layer (such as a fiber cut) may also trigger alarms in routing as well as other upper protocol layers [6]. It is reported that a single fiber cut with 16 disrupted wavelengths can lead to hundreds of alarms in the network [1]. This not only increases the management cost of the control plane, but also makes the failure localization difficult.

Without loss of generality, we call the device that monitors the health of a certain part of the network as a *monitor* [1], [2], [4]–[9]. Such a monitor is also responsible for generating alarm if a failure is detected. Alarm signals are then broadcasted in the control plane with the highest priority [10]. Based on the alarm signals, a component failure can be localized. If we can achieve accurate failure localization with a reduced number of monitors, less alarm signals will be generated, and the failure localization will become easier. This also makes the network more scalable by simplifying the fault management mechanism. Therefore, it is very important to reduce the number of monitors without sacrificing the accuracy of failure localization. In this paper, we focus on an optical layer mechanism to achieve fast link failure detection and localization, using only a small set of monitors. Link failure due to fiber cut is a common failure in optical networks. In this paper, we assume a single link failure in the network. Since a link failure disrupts all the lightpaths passing through the failed link, it is more critical than a channel-based failure or optical signal degradation (which can be detected by upper layer protocols). Generally, failure detection and localization at the optical layer is much faster than that carried out by upper layer protocols [4], [5], [11], [12]. At the optical layer, a monitor can detect a link failure by measuring optical power, analyzing optical spectrum, using pilot tones or optical time-domain reflectometry (OTDR) [4]. As pointed out earlier, a link failure tends to trigger a huge number of redundant alarms at different protocol layers. Fast link failure localization at the optical layer can help to suppress such redundant alarms.

Extensive studies have been reported on fault monitoring in all-optical mesh networks [4], [8], [9], [11], [13]–[17]. Conventional link-based monitoring is the most straightforward approach that requires one monitor at each link. To reduce the number of monitors, the concept of monitoring cycle (m-cycle) [13]–[17] is proposed, where a cyclic monitoring structure (i.e., m-cycle) is employed to monitor the health of multiple links on the cycle. A link failure is localized by decoding the alarm signals generated by the monitors on a set of m-cycles passing

B. Wu and P.-H. Ho are with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: b7wu@uwaterloo.ca; pinhan@bbcr.uwaterloo.ca).

K. L. Yeung is with the Department of Electrical and Electronic Engineering, The University of Hong Kong, Pokfulam, Hong Kong (e-mail: kyeung@eee.hku.hk).
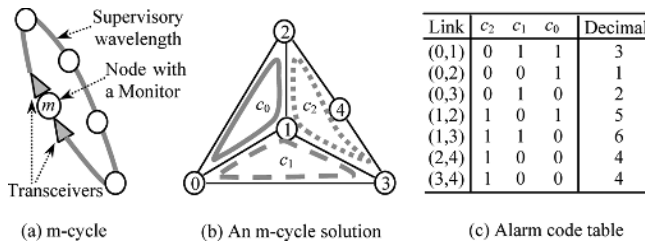
Fig. 1. Fast link failure localization using m-cycles.

Fig. 2. Two-edge cuts in a simple network. The two links incident on each dashed curve form a two-edge cut of the network topology.

through the failed link (detailed in Section II). Motivated by the fact that such a cyclic monitoring structure could limit the flexibility of monitoring resource allocation, this paper introduces a new framework of all-optical monitoring, called *monitoring trail* (m-trail). Compared with m-cycles, m-trails remove the cycle constraint, such that both cyclic and acyclic monitoring structures can be jointly considered to achieve unambiguous link failure localization. We also formulate an efficient integer linear program (ILP) for optimal design of m-trails. Numerical results show that the m-trail scheme significantly outperforms its m-cycle-based counterpart.

The rest of the paper is organized as follows. Section II provides a literature review on m-cycles. Section III introduces the m-trail concept. Section IV gives a theoretical analysis on our ILP-based approach, followed by the ILP formulation in Section V. Numerical results and discussions are presented in Section VI. We conclude the paper in Section VII.

II. LITERATURE REVIEW ON M-CYCLES

Fig. 1(a) shows the structure of an m-cycle, which is a pre-configured optical loop-back connection, using a supervisory wavelength on each link it traverses. Each m-cycle is associated with a pair of optical transceivers and a monitor. A supervisory optical signal is transmitted along the m-cycle using the optical transceivers and the supervisory wavelengths. If any link traversed by an m-cycle fails, optical signal in the supervisory wavelengths will be disrupted. The monitor detects the disruption and generates an alarm. Generally, an m-cycle solution consists of a set of M m-cycles $\{c_0, c_1, \ldots, c_{M-1}\}$ that covers every link of a given network. Upon a single link failure, monitors on all m-cycles traversing the failed link will alarm. This produces an *alarm code* $[a_{M-1}, \ldots, a_1, a_0]$, where $a_j = 1$ means that the monitor on m-cycle c_j alarms and $a_j = 0$ otherwise. Fig. 1(b) shows a solution with three m-cycles $\{c_0, c_1, c_2\}$. If link (0, 1) fails, the monitors on c_0 and c_1 will alarm to produce the alarm code $[0, 1, 1]$. Similarly, if link (0, 2) fails, the monitor on c_0 will alarm and the resulting alarm code is $[0, 0, 1]$. The alarm code table in Fig. 1(c) contains all possible alarm codes, based on which a particular link failure can be localized.

It is possible that a pure m-cycle solution is not sufficient to achieve *unambiguous link failure localization* (i.e., identify each link failure using a unique alarm code). For example, the failures at links (2, 4) and (3, 4) in Fig. 1(b) have the same alarm code, as shown in Fig. 1(c), and thus cannot be distinguished from each other. Define a *two-edge cut* of the network as a pair of links, where the network will be divided into two separate parts if the two links are removed. Fig. 2 gives an example, where each pair
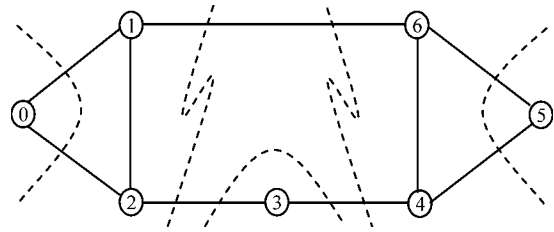
of links incident on a dashed curve forms a two-edge cut. For a two-edge cut, the two link failures cannot be distinguished by any cycle-based monitoring scheme, because the two links must be traversed by the same set of m-cycles. To achieve un-ambiguous link failure localization, extra link-based monitors are required [13]–[17]. Specifically, a link-based monitor can be used to monitor either link in a two-edge cut, such that the two link failures can be distinguished from each other. This is repeated until unambiguous link failure localization is achieved. In Fig. 1(b), an extra link-based monitor can be used at either (2, 4) or (3, 4). This increases the total number of monitors from 3 to 4, but it is still less than 7, as required by a pure link-based monitoring scheme.

Several algorithms [13]–[17] have been proposed for m-cycle design. In particular, HST [14] constructs m-cycles based on a spanning-tree of the network. The links in the spanning-tree are called *trunks*, and other links are *chords*. An m-cycle is generated from each chord, where all other links traversed by this m-cycle must be trunks. Let E be the set of all the links and V be the set of all the nodes in the network. Because a network has $\|V\| - 1$ trunks and $\|E\| - \|V\| + 1$ chords, an HST solution contains exactly $\|E\| - \|V\| + 1$ m-cycles/monitors, plus extra link-based monitors if required. Though the number of required monitors is generally less than $\|E\|$ (as required by link-based monitoring), it still increases linearly with the network size. Another algorithm M^2-CYCLE [16] always generates m-cycles with the minimum cycle length, where the *length* of a cycle is defined as the total number of links it traverses (assume that hopcount is the cost metric). It is proved [16] that M^2-CYCLE always outperforms HST by requiring less monitoring resources, but the required number of monitors still increases linearly with the network size. Both HST and M^2-CYCLE generate only simple m-cycles, where a simple m-cycle can traverse a node at most once. Besides, they do not allow any tradeoff between the monitor cost and the bandwidth cost (i.e., the cost of the supervisory wavelength-links), and thus the solutions are determined solely by the network topology. The work in [17] removes the aforementioned limitations. It introduces nonsimple m-cycles, where a nonsimple m-cycle can traverse a node multiple times, as shown in Fig. 3. An important contribution of the work in [17] is that, the m-cycle design problem is translated into binary coding of individual link failures, under the network topology and the cycle constraints. Since each m-cycle matches one bit in the binary alarm codes [see Fig. 1(c)], this dramatically reduces the required number of m-cycles from $O(\|E\| - \|V\| + 1)$ to $O(\log_2 \|E\|)$. Compared with simple cycles, nonsimple cycles are more flexible in exploiting mesh connectivity of a network
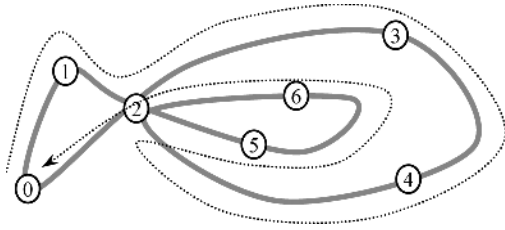
Fig. 3. Nonsimple m-cycle. The dotted arrow shows a possible connection pattern of the supervisory wavelengths.

Link	t_2	t_1	t_0	Decimal
(0,1)	1	0	1	5
(0,2)	1	1	1	7
(0,3)	1	0	0	4
(1,2)	0	1	1	3
(1,3)	1	1	0	6
(2,4)	0	0	1	1
(3,4)	0	1	0	2

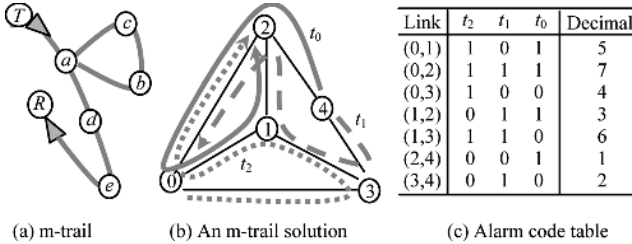(a) m-trail   (b) An m-trail solution   (c) Alarm code table

Fig. 4. Fast link failure localization using m-trails.

[18], but such flexibility is still limited by the cycle constraint. For example, without the aid of link-based monitors, link failures in a two-edge cut cannot be distinguished by any simple or nonsimple m-cycle solution. However, all existing m-cycle design algorithms [13]–[17] fail to achieve a joint optimization on both cycle-based and link-based monitoring. Besides, the monitoring structure is limited to either cycle based or link based, and other possible monitoring structures are not considered.

III. M-TRAIL CONCEPT

Though an m-cycle-based monitoring scheme [13]–[17] can greatly reduce the number of required monitors, the optical connection of the supervisory wavelengths is still constrained in a loop. By assuming that the optical transmitter and receiver of a single monitoring structure are not necessarily collocated at the same node, the cycle structure is broken, which leads to a new monitoring structure called *monitoring trail* (m-trail), as shown in Fig. 4(a). Though the cycle constraint is removed, an m-trail works in the same way as an m-cycle for fast link failure localization.

Similar to a nonsimple m-cycle, an m-trail can traverse a node multiple times but a link at most once. The node with the transmitter is defined as the *source* of the m-trail and is denoted by T. Similarly, the node with the receiver is defined as the *sink* of the m-trail and is denoted by R. A dedicated monitor is collocated with the receiver at sink R. In Fig. 4(a), the supervisory wavelengths can bepre-cross-connected in either $T \rightarrow a \rightarrow b \rightarrow c \rightarrow a \rightarrow d \rightarrow e \rightarrow R$ or $T \rightarrow a \rightarrow c \rightarrow b \rightarrow a \rightarrow d \rightarrow e \rightarrow R$. Differentpre-cross-connection patterns based on the same set of supervisory wavelengths will not affect the monitoring result, because we only care about whether the supervisory optical signal in an m-trail is disrupted or not. If an m-trail has a closed loop-back structure (i.e., a simple or nonsimple m-cycle), it is called a *closed trail*; otherwise it is an *open trail*. Therefore, the m-trail concept provides a general all-optical monitoring structure, which includes

simple, nonsimple m-cycles, and open trails (link-based monitoring and nonlink-based open trails) as special cases.

Our objective in m-trail design is to minimize the *monitoring cost*, which consists of the monitor cost and the bandwidth cost. For simplicity, the hardware cost of the transceivers is counted into the monitor cost. Since reducing the number of monitors greatly simplifies the failure management, we can also estimate the failure management cost and amortize it into the cost of each monitor. Let the *length* of an m-trail be the number of links it traverses. The bandwidth cost is denoted by the *cover length*, which is the length sum of all the m-trails in the solution, or the total number of supervisory wavelength-links required. Accordingly, we can formulate the monitoring cost given in the following:

$$
\begin{aligned}
Monitoring\ Cost =\ & \text{monitor cost} \\
& + \text{bandwidth cost} \\
=\ & \gamma \times \text{number of monitors} \\
& + \text{cover length.}
\end{aligned} \tag{1}
$$

The *cost ratio* γ determines the relative importance between the monitor cost and the bandwidth cost. Fig. 4(b) gives an m-trail solution for the network in Fig. 1(b) and Fig. 4(c) shows the alarm code table. We can see that only three m-trails $\{t_0, t_1, t_2\}$ (each with a dedicated monitor) are required to distinguish all the link failures. Though two links (2, 4) and (3, 4) form a two-edge cut of the network, they can be traversed by different m-trails, and no additional link-based monitoring is required to distinguish the two corresponding link failures. To achieve the same unambiguous link failure localization, the solution in Fig. 1(b) needs four monitors, three for m-cycles $\{c_0, c_1, c_2\}$, and one additional link-based monitor for the two-edge cut $\{(2, 4), (3, 4)\}$. By taking $\gamma = 5$, the solution in Fig. 4(b) only requires a monitoring cost of $3 \times 5 + 12 = 27$, whereas the solution in Fig. 1(b) requires $4 \times 5 + 11 = 31$. We can see that the m-trail solution cuts down the monitoring cost by 12.9%.

IV. THEORY BEHIND ILP

In what follows, we formulate an efficient ILP for m-trail design to achieve unambiguous link failure localization, with the objective of minimizing the monitoring cost in (1). Generally, ILP-based approaches need a long running time to generate solutions. To shorten the ILP running time, we simplify the optimization problem by minimizing the number of necessary ILP variables. This is achieved by formulating trail and unambiguous link failure localization constraints using some intelligent algorithms, as well as reducing the ILP solution space with a proper bound on the monitoring cost. We detail the aforementioned theoretical points in this section, whereas the ILP formulation is presented in Section V.

A. Trail Formulation

We use *on-trail vectors* (vectors for short) to denote the supervisory wavelengths of an m-trail t_j. Fig. 5(a) shows an m-trail consisting of four vectors. A vector $u \rightarrow v$ denotes a supervisory wavelength of t_j on link (u, v), where the supervisory optical signal is transmitted from node u to node v. Each m-trail t_j has a unique source–sink (i.e., $T - R$) node pair. Let Δ_u be
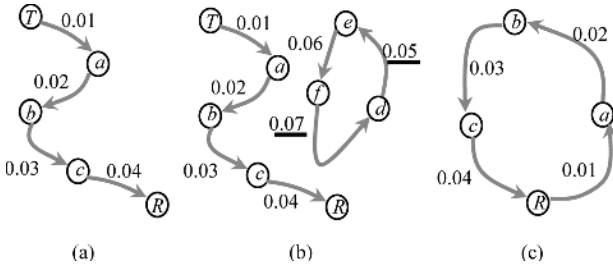
Fig. 5. Voltage constraint (assume that each node is traversed at most once).

Fig. 6. Voltage constraint (some nodes are traversed multiple times).

the difference of the number of outbound and inbound vectors at node u. For an open trail, we have $\Delta_T = 1$, $\Delta_R = -1$, and $\Delta_u = 0$ for $u \neq T$ and $u \neq R$. In other words, the vectors must obey flow conservation at each node, except at source T and sink R. For a closed trail, T and R denote the same node (we still use the term "$T - R$ node pair" for simplicity), and we have $\Delta_u = 0$ for each node u in the network.

Though we intend to formulate a single trail, the abovementioned formulation may result in multiple disjoint trails without any common node. Fig. 5(b) shows an example where the flow conservation property is obeyed at each node except T and R. Instead of having a single trail, two trails (an open trail and a cycle) are generated, and each trail needs a dedicated monitor. Generally, the exact number of disjoint trails is unknown. This makes it difficult to count the number of required monitors.

To generate a single trail t_j at a time, we define a positive *voltage* value q_{uv}^j for each directed vector $u \rightarrow v$ on t_j, as denoted by a fraction next to each vector in Fig. 5. If $q_{uv}^j > 0$, then $u \rightarrow v$ is a vector on m-trail t_j and $q_{vu}^j = 0$. If a link (u, v) is not traversed by t_j, then $q_{uv}^j = 0$ and $q_{vu}^j = 0$. For any node traversed by t_j except sink R, we require the sum of the voltage values of its outbound vectors to be larger than that of its inbound vectors. This is called the *voltage constraint*. The specific voltage values are not important as long as the voltage constraint is obeyed. Note that the voltage constraint does not apply to sink R. For simplicity, we first assume that t_j traverses any node at most once, as shown by the example in Fig. 5(a). Then, the voltage values of the vectors on t_j must keep increasing along the trail, except at sink R where a voltage decrease may occur. We can see that a feasible set of voltage values can be found in Fig. 5(a). On the other hand, Fig. 5(b) contains two disjoint trails (an open trail and a cycle), and the cycle $d \rightarrow e \rightarrow f \rightarrow d$ does not traverse sink R. Then, voltage values must monotonically increase along the cycle. This leads to a voltage conflict (i.e., a violation of the voltage constraint) at node d, as indicated by the two underlined voltage values. In contrast, Fig. 5(c) gives a valid m-cycle (where both the transmitter and the receiver are installed at node R), because the voltage value 0.01 of $R \rightarrow a$ can be smaller than 0.04 of $c \rightarrow R$ at sink R. The abovementioned voltage analysis can be extended to the case where more disjoint cycles are involved. With the voltage constraint, any cycle without traversing sink R will encounter a voltage conflict due to its cyclic structure.

We now consider a more general case, where a trail traverses some nodes multiple times. Consider the solid open trail in Fig. 6. Assume that the supervisory wavelength on link (T, a) ispre-cross-connected to that on (a, b), and the
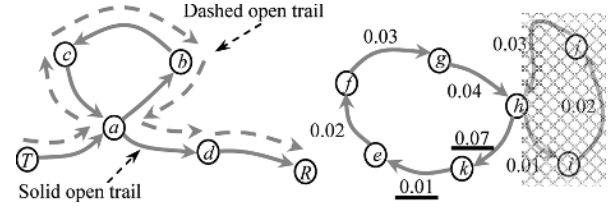
supervisory wavelength on (c, a) ispre-cross-connected to that on (a, d). At node a, the voltage constraint translates to $q_{ad}^j + q_{ab}^j > q_{Ta}^j + q_{ca}^j$ or $q_{ad}^j > q_{Ta}^j + (q_{ca}^j - q_{ab}^j)$. On the other hand, the voltage constraint ensures $q_{ca}^j > q_{bc}^j > q_{ab}^j$ along $a \rightarrow b \rightarrow c \rightarrow a$ and thus $q_{ca}^j - q_{ab}^j > 0$. As a result, we have $q_{ad}^j > q_{Ta}^j$. If we change thepre-cross-connection pattern of the supervisory wavelengths, as indicated by the dashed arrows in Fig. 6 [where the supervisory wavelength on (T, a) ispre-cross-connected to that on (a, c), and the supervisory wavelength on (b, a) ispre-cross-connected to that on (a, d)], we can still prove $q_{ad}^j > q_{Ta}^j$ with similar analysis. This interesting observation means that, if a local loop-back (such as the solid $a \rightarrow b \rightarrow c \rightarrow a$ or the dashed $a \rightarrow c \rightarrow b \rightarrow a$ in Fig. 6) does not traverse sink R, it can be bypassed in our voltage analysis. No matter what is thepre-cross-connection pattern of the supervisory wavelengths, the voltage constraint always ensures increasing voltage values along the trail by ignoring the local loop-back. For the solid open trail in Fig. 6, even if we allow $q_{ab}^j < q_{Ta}^j$ at node a (i.e., the voltage value of $a \rightarrow b$ on the local loop-back $a \rightarrow b \rightarrow c \rightarrow a$ is smaller than that of its upstream vector $T \rightarrow a$ outside the loop-back), the same result $q_{ad}^j > q_{Ta}^j$ can still be ensured if the voltage constraint holds at each individual node. If the nonsimple cycle in Fig. 6 (which does not traverse sink R) is considered, voltage values increase along $k \rightarrow e \rightarrow f \rightarrow g \rightarrow h$. The shadowed local loop-back $h \rightarrow i \rightarrow j \rightarrow h$ is bypassed and $q_{hk}^j > q_{gh}^j$ is ensured. Then, $q_{ke}^j < q_{hk}^j$ violates the voltage constraint at node k. Generally, we have the following theorem, which is strictly proved in the Appendix.

Theorem: With the voltage constraint at each individual node, any cycle without traversing sink R will encounter a voltage conflict.

In our ILP, we allow only a unique $T - R$ node pair in each t_j, and the vectors in t_j must obey flow conservation at each node (except at T and R). With the voltage constraint at each node, this sufficiently ensures a single (open or closed) trail in t_j. Otherwise either the flow conservation property or the voltage constraint will be violated. On the other hand, the voltage constraint can properly work without knowing thepre-cross-connection pattern of the supervisory wavelengths. This removes the need of formulating thepre-cross-connection pattern of each m-trail. As a result, the ILP formulation can be greatly simplified.

B. Unambiguous Link Failure Localization

To localize each link failure without any ambiguity, every link in the network must have a unique alarm code. With binary alarm codes [see the middle column in Figs. 1(c) and 4(c)], we

have to carry out bitwise comparisons in order to determine whether two alarm codes are different or not. To remove bitwise comparisons, we introduce *decimal alarm code*, which is a decimal translation of the corresponding binary alarm code, as shown in the last column in Figs. 1(c) and 4(c). Due to the one-to-one mapping between binary and decimal codes, checking if two binary codes are different is equivalent to checking if the corresponding decimal codes are unequal. Since bitwise comparisons are removed, the number of ILP variables and constraints can be greatly reduced.

Though checking the inequality of two decimal codes seems to be an easy task, we may not be able to easily formulate it in an ILP constraint due to its nonlinear nature. Assume that there are at most J m-trails in the solution. Since each m-trail matches one bit in a binary alarm code, the candidate set of all possible decimal alarm codes is $\{1, 2, \ldots, 2^J - 1\}$. To formulate a unique alarm code for each link failure, the algorithm in [17] adopts a set of $\|\boldsymbol{E}\| \times (2^J - 1)$ auxiliary ILP variables to choose a unique value from $\{1, 2, \ldots, 2^J - 1\}$. Since the number of auxiliary ILP variables increases exponentially with J, this approach generally requires a long running time to generate a solution.

In this paper, we use a new approach to reduce the required number of ILP variables. We denote two distinct links (u, v) and (x, y) by $(u, v), (x, y) \in \boldsymbol{E} : (u, v) \neq (x, y)$ and the corresponding decimal alarm codes by α_{uv} and α_{xy}. A binary variable f_{uv}^{xy} is defined to indicate the inequality between α_{uv} and α_{xy}. Specifically, $f_{uv}^{xy} = 1$ means $\alpha_{uv} > \alpha_{xy}$ and $f_{uv}^{xy} = 0$ means $\alpha_{uv} < \alpha_{xy}$. With a predefined small positive constant β, the following constraint can efficiently ensure $\alpha_{uv} \neq \alpha_{xy}$. See equation at the bottom of the page. The specific value of β is not important, as long as it is small enough to ensure $\beta + \beta \times |\alpha_{uv} - \alpha_{xy}| \leq 1$. For example, if we allow nine m-trails in the solution, the candidate set of the decimal alarm codes is $\{1, 2, \ldots, 2^9 - 1\}$. As a result, β can be predefined in $0 < \beta \leq 1/512$. In this approach, the required number of auxiliary variables (i.e., f_{uv}^{xy}) is only $\|\boldsymbol{E}\| \times (\|\boldsymbol{E}\| - 1)/2$.

C. Maximum Number of M-Trails (J)

In our ILP, we need to predefine an integer J, which denotes the maximum number of m-trails allowed in the solution. Obviously, taking a smaller value of J will reduce the number of variables and constraints in the ILP and thus shorten the computation time in solving the ILP. However, if the value of J is smaller than that required by an optimal solution, the ILP will never return an optimal solution (or even cannot find a feasible solution). On the other hand, if J is predefined large enough, the optimality of the solution can be ensured if less than J m-trails are obtained. Therefore, the actual number of m-trails is a variable upperbounded by J and is determined by solving the ILP.

It is important to predefine a proper value of J, such that ILP solutions can be obtained in a reasonable running time.

Since the network contains $\|\boldsymbol{E}\|$ links, at least $\lfloor \log_2 \|\boldsymbol{E}\| \rfloor + 1$ bits are required in the binary alarm codes to achieve unambiguous link failure localization. As a result, the lower bound $J_{\min}$ for the required number of m-trails is

$$J_{\min} = \lfloor \log_2 \|\boldsymbol{E}\| \rfloor + 1. \tag{2}$$

The actual number of m-trails required in a solution tends to be close to the lower bound $J_{\min}$ in (2). This is because adding an additional m-trail in the solution means one more bit in the binary alarm codes, which will double the size of the candidate set $\{1, 2, \ldots, 2^J - 1\}$ of decimal alarm codes. For the SmallNet topology in Fig. 11 with $\|\boldsymbol{E}\| = 22$ links, at least $J_{\min} = \lfloor \log_2 \|\boldsymbol{E}\| \rfloor + 1 = \lfloor \log_2 22 \rfloor + 1 = 5$ m-trails are required. If we set $J = 12$, the candidate set of decimal alarm codes will be $\{1, 2, \ldots, 4095\}$, with a size much larger than $\|\boldsymbol{E}\| = 22$. This gives very high flexibility in choosing only 22 distinct alarm codes from 4095 candidates. Let δ be a small positive integer. Generally, we can predefine the value of J according to (3).

$$J = \lfloor \log_2 \|\boldsymbol{E}\| \rfloor + \delta. \tag{3}$$

From (3), J (i.e., the maximum number of m-trails in the solution) is generally small even in a large-size network. This also allows us to keep the ILP problem size small.

D. Lower Bound on the Monitoring Cost

To reduce the solution space of the ILP, we can formulate a lower bound on the monitoring cost in (1). In an alarm code table [as the one in Fig. 4(c)], a "1" in each binary alarm code means that the corresponding link is traversed by an m-trail. If a solution contains exactly k m-trails, each binary alarm code should have k bits. Define the *cover times* of a link as the total number of "1" bits in its binary alarm code. Among all the $\|\boldsymbol{E}\|$ links, we can have at most $C_k^1 = k$ links with a cover times of 1, and other links must have a cover times no smaller than 2. Otherwise, there must be two links with identical alarm codes and thus the corresponding link failures cannot be uniquely identified. Assume $R_1 = \|\boldsymbol{E}\| - C_k^1 \geq C_k^2$. With similar analysis, it is easy to see that among the remaining R_1 links, at most C_k^2 links can have a cover times of 2, and so on. Fig. 4(c) gives an example with $k = 3$ m-trails in the solution. To achieve unambiguous link failure localization, there are at most $C_3^1 = 3$ links (0, 3), (2, 4) and (3, 4) with a cover times of 1, and another $C_3^2 = 3$ links (0, 1), (1, 2), and (1, 3) with a cover times of 2. Obviously, the remaining link (0, 2) can only have a cover times of 3 (or above).

$$\begin{cases} \beta + \beta \left(\alpha_{uv} - \alpha_{xy} \right) \leq f_{uv}^{xy} \\ \beta + \beta \left(\alpha_{xy} - \alpha_{uv} \right) \leq 1 - f_{uv}^{xy} \end{cases} \qquad \forall (u, v), (x, y) \in \boldsymbol{E} : (u, v) \neq (x, y).$$

Based on the above analysis, we can formulate a lower bound B_k for the monitoring cost by assuming that the solution contains exactly k m-trails. Since the cover length in (1) equals to the sum of cover times of all the links, we have the following equation(4):

$$\begin{cases} R_1 = \|\boldsymbol{E}\| - C_k^1 = \|\boldsymbol{E}\| - k \\ R_i = R_{i-1} - C_k^i, \quad i \geq 2 \\ B_k = \gamma k + k + \sum_{i \geq 1, R_i > 0} (i+1) \times \min\{R_i, \; C_k^{i+1}\}. \end{cases} \quad (4)$$

For example, the SmallNet in Fig. 11 has $\|\boldsymbol{E}\| = 22$ links. Assume $\gamma = 5$ and $k = 6$. Because $C_6^2 = 15$ and $C_6^3 = 20$, we have $R_1 = 16$, $R_2 = 1$, and $R_3 = -19$ (note that negative values are avoided in (4)). As a result, $B_6 = \gamma k + k + 2C_2^6 + 3R_2 = 5 \times 6 + 6 + 2 \times 15 + 3 \times 1 = 69$.

By taking each k in $J \geq k \geq J_{\min}$ into consideration, the lower bound B on the monitoring cost in (1) can be determined by

$$B = \min_{J \geq k \geq J_{\min}} \{B_k\}. \quad (5)$$

For the SmallNet in Fig. 11, the lower bound on the required number of m-trails is $J_{\min} = \lfloor \log_2 \|\boldsymbol{E}\| \rfloor + 1 = \lfloor \log_2 22 \rfloor + 1 = 5$. Assume $J = 9$ and $\gamma = 5$. From (4), we have $B_5 = 71$, $B_6 = 69$, $B_7 = 72$, $B_8 = 76$, and $B_9 = 80$. According to (5), the lower bound on the monitoring cost is $B = B_6 = 69$.

V. ILP FORMULATION

Notation List

J The maximum number of m-trails allowed in the solution.

j m-trail index, where $j \in \{0, 1, \ldots, J-1\}$.

$\boldsymbol{E}$ The set of all the links in the network.

$\boldsymbol{V}$ The set of all the nodes in the network.

c_{uv} Predefined cost of a supervisory wavelength on link (u, v). Either hopcount or distance-related cost can be used (hopcount is used in this paper).

L Predefined length limit of each m-trail.

γ Predefined cost ratio of a monitor to a supervisory wavelength link.

λ A predefined small positive value ($\|\boldsymbol{E}\|^{-1} \geq \lambda > 0$). It is the minimum step of voltage increase in the voltage constraint.

B Lower bound of the monitoring cost as formulated in (5).

β A predefined small constant and $2^{-J} \geq \beta > 0$.

e_{uv}^j Binary variable. It takes 1 if $u \rightarrow v$ is an on-trail vector of m-trail t_j, and 0 otherwise.

m^j Binary variable. It takes 1 if t_j is an m-trail, and 0 otherwise.

t_u^j Binary variable. It takes 1 if $u = T$ on m-trail t_j (i.e., node u is the source), and 0 otherwise.

r_u^j Binary variable. It takes 1 if $u = R$ on m-trail t_j (i.e., node u is the sink), and 0 otherwise.

z_u^j Binary variable. It takes 1 if node u is traversed by m-trail t_j, and 0 otherwise.

q_{uv}^j Nonnegative fractional variable. It is the voltage of vector $u \rightarrow v$ on m-trail t_j. It takes 0 if $u \rightarrow v$ is not an on-trail vector of t_j.

α_{uv} General integer variable. It is the decimal alarm code assigned to link (u, v).

f_{uv}^{xy} Binary variable. For two distinct links (u, v) and (x, y), it takes 1 if $\alpha_{uv} > \alpha_{xy}$, and 0 if $\alpha_{uv} < \alpha_{xy}$.

A. ILP Formulation

Given a network topology $\boldsymbol{G}(\boldsymbol{V}, \boldsymbol{E})$, the cost c_{uv} of a supervisory wavelength on each link $(u, v) \in \boldsymbol{E}$ and the cost ratio γ of a monitor to a supervisory wavelength-link, the ILP formulated in (6)–(21) can generate an optimal m-trail solution with the minimum monitoring cost to achieve unambiguous link failure localization. See (6)–(21) at the bottom of the next page.

Objective (6) aims at minimizing the monitoring cost in (1). Constraints (7) and (8) allow a single $T - R$ node pair in each m-trail t_j. Constraint (9) formulates the flow conservation property at each node. If a node u is neither T nor R (i.e., $t_u^j - r_u^j = 0$), it must have an equal number of inbound and outbound vectors. For an open trail, we have $t_u^j - r_u^j = 1$ at source T and $t_u^j = r_u^j = -1$ at sink R. For a closed trail (or cycle), flow conservation is ensured at every node. Constraint (10) allows at most a single directed vector on each link (u, v), either $u \rightarrow v$ or $v \rightarrow u$, or none. Constraint (11) indicates that a node is traversed by an m-trail t_j if it has at least one inbound or outbound vector on t_j. Constraint (12) says that only an on-trail vector can have a positive voltage value. The voltage constraint is formulated in (13). If a node is traversed by an m-trail t_j and is not the sink R, the voltage sum of its outbound vectors must be larger than that of its inbound vectors. Constraint (14) specifies the lower bound on the required number of m-trails. Constraint (15) means that, if the solution contains $\sum_j m^j$ m-trails, only $\sum_j m^j$ links can have a cover times of 1, and all other links must have a cover times no smaller than 2. Constraint (16) stipulates the lower bound on the monitoring cost. Since the required number of m-trails in the solution may be less than J, there could be some *empty trails* without traversing any link. Constraint (17) identifies whether a trail t_j is empty or not by checking its source T. Constraint (18) translates binary alarm codes into decimal ones, and constraint (19) prevents zero alarm codes. Finally, constraints (20) and (21) ensure a unique alarm code for each link failure. If we also have a length limit L for each m-trail, we can add an additional constraint (22) below.

$$\sum_{(u,v) \in \boldsymbol{E}} c_{uv} \left(e_{uv}^j + e_{vu}^j \right) \leq L \; \forall j \quad (22)$$

VI. NUMERICAL RESULTS AND DISCUSSIONS

We use ILOG CPLEX 11.0 [19] to implement the ILP on a server with 3 GHz Intel Xeon CPU 5160. The CPLEX environment parameters are set as follows.

$$
\begin{cases}
1 \rightarrow & \text{emphasis mip} \\
2 \rightarrow & \text{mip strategy probe} \\
3 \rightarrow & \text{mip strategy rins} \\
3 \rightarrow & \text{mip strategy heuristicfreq} \\
2 \rightarrow & \text{mip cuts all} \\
3 \rightarrow & \text{mip strategy dive} \\
3 \rightarrow & \text{preprocessing symmetry.}
\end{cases}
\tag{23}
$$

The same set of predefined parameters $J = 12$, $\gamma = 5$, $\lambda = 0.01$, and $\beta = 0.0001$ are used for all examples. Since the final solution may contain less than J m-trails, we add an additional constraint $m^j \geq m^{j+1}\,(J - 1 > j \geq 0)$ to the ILP. Then,

the set of nonempty m-trails will be sequentially indexed by $j \in \{0, 1, 2, \ldots\}$. Meanwhile, empty trails without traversing any link are removed from the alarm code tables.

Fig. 7 shows an optimal m-trail solution for the network in Fig. 2, which consists of three open trails t_0, t_1, t_3, and a closed trail (simple cycle) t_2. Recall that for each two-edge cut in Fig. 2, the corresponding link failures cannot be distinguished by a pure m-cycle solution without the aid of additional link-based monitors. The m-trail solution in Fig. 7 does not have this problem, as indicated by the alarm code table. Fig. 8 gives an m-trail solution for the ARPA2 network [14]. Among the eleven m-trails obtained, three m-trails t_0, t_5, and t_6 carry out link-based monitoring. The examples in Figs. 7 and 8 clearly show that m-trail is a generalization of both m-cycle and open trail (including link-based monitoring). An m-trail-based monitoring scheme can always achieve unambiguous link failure localization due to the general and flexible monitoring structure of m-trails. In addition, the monitoring cost can be

Objective :

$$
\text{minimize} \left\{ \gamma \sum_j m^j + \sum_j \sum_{(u,v) \in \boldsymbol{E}} c_{uv} \left(e^j_{uv} + e^j_{vu} \right) \right\}
\tag{6}
$$

Subject to :
$$
\sum_{u \in \boldsymbol{V}} t^j_u \leq 1 \qquad \forall j
\tag{7}
$$

$$
\sum_{u \in \boldsymbol{V}} r^j_u \leq 1 \qquad \forall j
\tag{8}
$$

$$
\sum_{(u,v) \in \boldsymbol{E}} \left(e^j_{uv} - e^j_{vu} \right) = t^j_u - r^j_u \qquad \forall u \in \boldsymbol{V}, \quad \forall j
\tag{9}
$$

$$
e^j_{uv} + e^j_{vu} \leq 1 \qquad \forall (u,v) \in \boldsymbol{E}, \quad \forall j
\tag{10}
$$

$$
z^j_u \geq e^j_{uv} + e^j_{vu} \qquad \forall u \in \boldsymbol{V} : (u,v) \in \boldsymbol{E}, \quad \forall j
\tag{11}
$$

$$
q^j_{uv} \leq e^j_{uv}, q^j_{vu} \leq e^j_{vu} \qquad \forall (u,v) \in \boldsymbol{E}, \forall j
\tag{12}
$$

$$
r^j_u + \sum_{(u,v) \in \boldsymbol{E}} \left(q^j_{uv} - q^j_{vu} \right) \geq \lambda \times z^j_u \qquad \forall u \in \boldsymbol{V}, \forall j
\tag{13}
$$

$$
\sum_j m^j \geq \lfloor \log_2 \|\boldsymbol{E}\| \rfloor + 1
\tag{14}
$$

$$
\sum_j \sum_{(u,v) \in \boldsymbol{E}} c_{uv} \left(e^j_{uv} + e^j_{vu} \right) \geq 2 \|\boldsymbol{E}\| - \sum_j m^j
\tag{15}
$$

$$
\gamma \sum_j m^j + \sum_j \sum_{(u,v) \in \boldsymbol{E}} c_{uv} \left(e^j_{uv} + e^j_{vu} \right) \geq B
\tag{16}
$$

$$
m^j \geq \sum_j t^j_u \qquad \forall j
\tag{17}
$$

$$
\alpha_{uv} = \sum_j 2^j \left(e^j_{uv} + e^j_{vu} \right) \qquad \forall (u,v) \in \boldsymbol{E}
\tag{18}
$$

$$
\alpha_{uv} \geq 1 \qquad \forall (u,v) \in \boldsymbol{E}
\tag{19}
$$

$$
\beta + \beta \left(\alpha_{uv} - \alpha_{xy} \right) \leq f^{xy}_{uv} \qquad \forall (u,v), (x,y) \in \boldsymbol{E} : (u,v) \neq (x,y)
\tag{20}
$$

$$
\beta + \beta \left(\alpha_{xy} - \alpha_{uv} \right) \leq 1 - f^{xy}_{uv} \qquad \forall (u,v), (x,y) \in \boldsymbol{E} : (u,v) \neq (x,y).
\tag{21}
$$
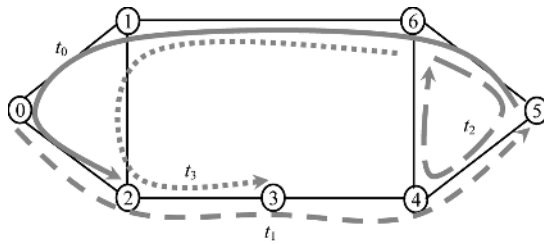
Link	t_3	t_2	t_1	t_0	Decimal	Link	t_3	t_2	t_1	t_0	Decimal
(0,1)	0	0	0	1	1	(3,4)	0	0	1	0	2
(0,2)	0	0	1	1	3	(4,5)	0	1	1	0	6
(1,2)	1	0	0	0	8	(4,6)	0	1	0	0	4
(1,6)	1	0	0	1	9	(5,6)	0	1	0	1	5
(2,3)	1	0	1	0	10						

Predefined parameters: J=12, γ=5, λ=0.01, β=0.0001, B=34

Results: ILP running time=12.70 sec, Gap to optimality=0% (Optimal),

Number of m-trails=4, Cover length =14, Monitoring cost=34

Fig. 7.   Optimal m-trail solution for the network in Fig. 2.
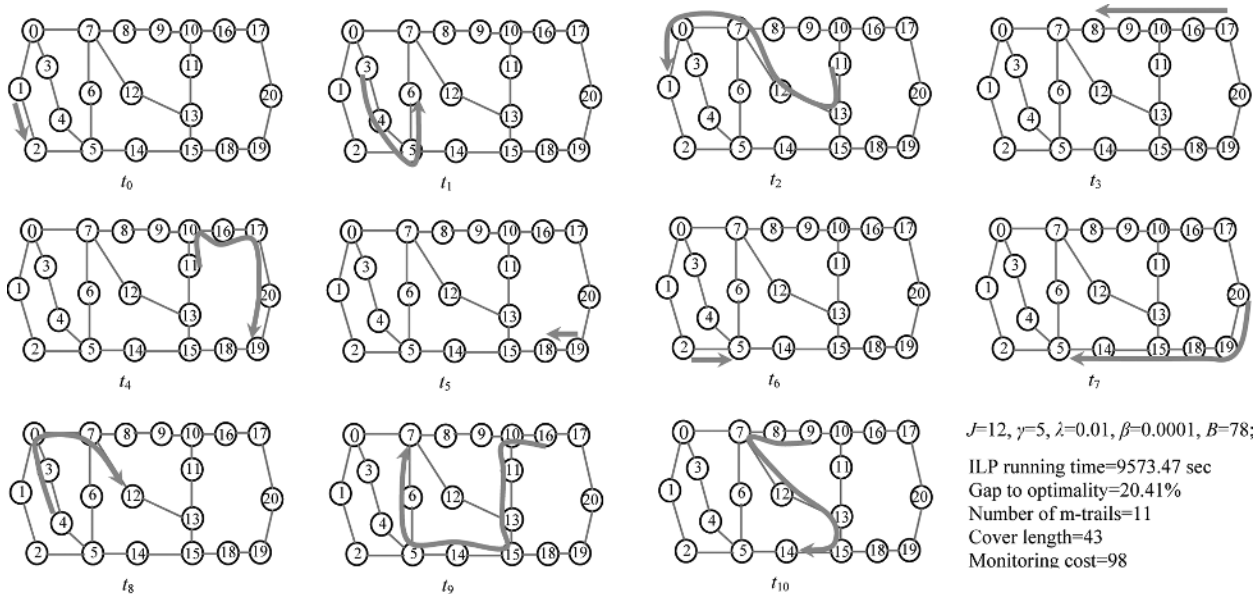
J=12, γ=5, λ=0.01, β=0.0001, B=78;

ILP running time=9573.47 sec
Gap to optimality=20.41%
Number of m-trails=11
Cover length=43
Monitoring cost=98

Link	t_{10}	t_9	t_8	t_7	t_6	t_5	t_4	t_3	t_2	t_1	t_0	Decimal	Link	t_{10}	t_9	t_8	t_7	t_6	t_5	t_4	t_3	t_2	t_1	t_0	Decimal	Link	t_{10}	t_9	t_8	t_7	t_6	t_5	t_4	t_3	t_2	t_1	t_0	Decimal
(0,1)	0	0	0	0	0	0	0	1	0	0		4	(6,7)	0	1	0	0	0	0	0	0	0	0	0	512	(13,15)	1	1	0	0	0	0	0	0	0	0	0	1536
(0,3)	0	0	1	0	0	0	0	0	0	0	0	256	(7,8)	1	0	0	0	0	0	0	0	0	0	0	1024	(14,15)	1	1	0	1	0	0	0	0	0	0	0	1664
(0,7)	0	0	1	0	0	0	0	0	1	0	0	260	(7,12)	1	0	1	0	0	0	0	0	1	0	0	1284	(15,18)	0	0	0	1	0	0	0	0	0	0	0	128
(1,2)	0	0	0	0	0	0	0	0	0	0	1	1	(8,9)	1	0	0	0	0	0	0	1	0	0	0	1032	(16,17)	0	0	0	0	0	1	1	0	0	0	0	24
(2,5)	0	0	0	0	1	0	0	0	0	0	0	64	(9,10)	0	0	0	0	0	0	0	1	0	0	0	8	(17,20)	0	0	0	0	0	1	0	0	0	0	0	16
(3,4)	0	0	1	0	0	0	0	0	0	1	0	258	(10,11)	0	1	0	0	0	0	1	0	0	0	0	528	(18,19)	0	0	0	1	0	1	0	0	0	0	0	160
(4,5)	0	0	0	0	0	0	0	0	0	1	0	2	(10,16)	0	1	0	0	0	0	1	1	0	0	0	536	(19,20)	0	0	0	1	0	0	1	0	0	0	0	144
(5,6)	0	1	0	0	0	0	0	0	0	1	0	514	(11,13)	0	1	0	0	0	0	0	0	1	0	0	516													
(5,14)	0	1	0	1	0	0	0	0	0	0	0	640	(12,13)	1	0	0	0	0	0	0	1	0	0	0	1028													

Fig. 8.   m-trail solution for the ARPA2 network with 21 nodes and 25 links.

optimally minimized by jointly considering both m-cycles and open trails.

To compare the monitoring cost in m-trail and m-cycle solutions, we can slightly modify the ILP in (6)–(21) as follows to generate solutions with only m-cycles: 1) because a pure m-cycle solution may not be able to achieve unambiguous localization for each link failure, the bounds in (14)–(16) cannot be applied and thus removed; 2) constraints (20)–(21) formulate unequal alarm codes and thus should be applied only to those distinguishable link failures; 3) the following constraint is added to the ILP to allow only m-cycles.

$$t_u^j = r_u^j \qquad \forall u \in V, \ \forall j. \qquad (24)$$

We first find an optimal m-cycle solution for the network in Figs. 2 and 7, as shown in Fig. 9. To achieve unambiguous link failure localization, four additional link-based monitors and supervisory wavelength-links are required to distinguish the link failures in the five two-edge cuts (see Fig. 2). As a result, the monitoring cost (m-cycle cost plus link-based monitoring cost) in Fig. 9 is 50. We can see that the m-trail solution in Fig. 7 cuts down the monitoring cost by 32%. To get an optimal m-cycle solution for ARPA2, we simplify the original ARPA2 topology in Fig. 10(a) using another equivalent topology in Fig. 10(b), where some links on the same segment in Fig. 10(a) are merged into a single "link" in Fig. 10(b). A number next to each "link" in Fig. 10(b) gives the cost of that "link," which is obtained by
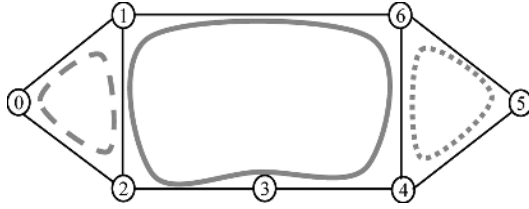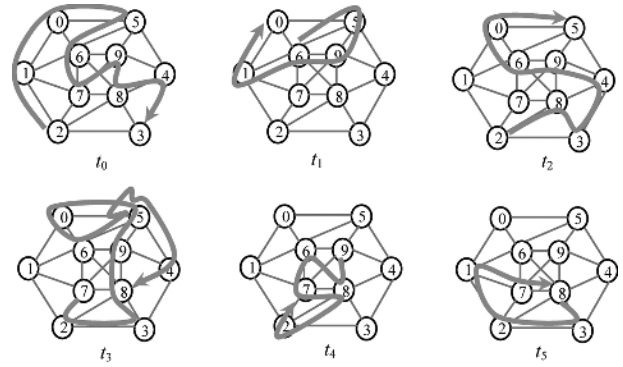
Fig. 9. Optimal pure m-cycle solution for the network in Fig. 2 with 3 m-cycles and 11 supervisory wavelength-links. To achieve unambiguous link failure localization, four additional link-based monitors and supervisory wavelength-links are required, and the total monitoring cost is 50.
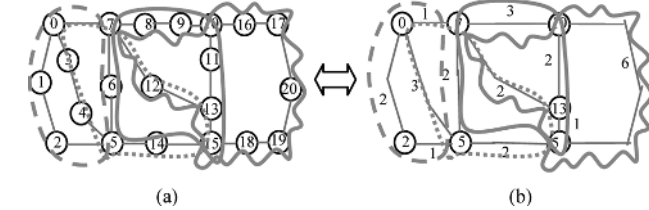

Fig. 10. Optimal m-cycle solution for ARPA2 with 4 m-cycles and 37 supervisory wavelength-links. To achieve unambiguous link failure localization, 15 additional link-based monitors and supervisory wavelength-links are required, and the total monitoring cost is 147.

adding up the costs of the corresponding links in Fig. 10(a). Since the topology in Fig. 10(b) is much simpler, the optimality of the m-cycle solution in Fig. 10(a) can be equivalently proved in Fig. 10(b). The optimal m-cycle solution includes four m-cycles with a cost of 57, as shown in Fig. 10(a). To achieve unambiguous link failure localization, we still need another 15 link-based monitors and supervisory wavelength-links. As a result, the total monitoring cost is 147. In contrast, the m-trail solution in Fig. 8 (with a monitoring cost of 98) cuts down the monitoring cost by 33.33%. Note that the m-trail solution in Fig. 8 has a gap-to-optimality of 20.41%, and it is obtained in 9573.47 s. Due to the NP-hardness of the optimization problem, allowing a longer ILP running time does not improve the solution quality too much in this example. Nevertheless, even the suboptimal m-trail solution can achieve a significant monitoring cost saving of 33.33% over the optimal m-cycle-based counterpart. The above examples show that m-trail solutions significantly outperform their m-cycle-based counterparts.

Fig. 11 shows another example based on the SmallNet topology [14]. Though we set $J = 12$, only six m-trails (five open trails $t_0 - t_4$ and a cycle t_5) are required for unambiguous link failure localization. Despite of the large value of J, CPLEX 11.0 only needs 1543.49 s to generate the solution with a gap-to-optimality of 4.17%. For the ARPA2 network in Fig. 8 and the SmallNet in Fig. 11, the ILP cannot find an optimal solution in an acceptable running time, but good feasible solutions can be obtained reasonably fast.

So far, we have focused on proposing the m-trail concept and formulating an ILP to demonstrate its superior performance over the existing monitoring schemes. But, the ILP-based design approach is not scalable to large network size, and fast heuristic algorithms are desired in practical engineering designs. Most recently, a heuristic [20] was proposed for m-trail design in large-size networks. It includes two steps: random code assignment (RCA) and random code swapping (RCS). In RCA,

Link	t_5 t_4 t_3 t_2 t_1 t_0	Decimal	Link	t_5 t_4 t_3 t_2 t_1 t_0	Decimal
(0,1)	0 0 0 0 1 1	3	(4,5)	0 0 1 0 0 0	8
(0,5)	0 0 1 1 0 1	13	(4,8)	0 0 1 0 0 1	9
(0,6)	0 0 1 1 0 0	12	(4,9)	0 0 0 1 0 0	4
(1,2)	1 0 0 0 0 1	33	(5,6)	0 0 1 0 1 1	11
(1,6)	0 0 0 0 1 0	2	(5,9)	0 0 1 0 1 0	10
(1,7)	1 0 0 0 0 0	32	(6,7)	0 1 0 0 0 1	17
(2,3)	1 0 1 0 0 0	40	(6,8)	0 1 0 0 0 0	16
(2,7)	0 1 1 0 0 0	24	(6,9)	0 0 0 1 1 0	6
(2,8)	0 1 0 1 0 0	20	(7,8)	1 1 0 0 0 0	48
(3,4)	0 0 0 1 0 1	5	(7,9)	0 0 0 0 0 1	1
(3,8)	1 0 1 1 0 0	44	(8,9)	0 1 1 0 0 1	25

Predefined parameters:

$J=12$
$\gamma=5$
$\lambda=0.01$
$\beta=0.0001$
$B=69$

Result:

ILP running time=1543.49 sec
Gap to optimality=4.17%
Number of m-trails=6
Cover length=42
Monitoring cost=72

Fig. 11. m-trail solution for SmallNet with 10 nodes and 22 links.

a unique (temporary) alarm code is randomly assigned to each link in the network. Because this initial random alarm code assignment does not ensure that the supervisory wavelengths are organized in a set of m-trails, the second step RCS is carried out to shape the monitoring structures into m-trails by swapping the alarm codes among the links, where each binary bit in the alarm codes (or each m-trail) is shaped one by one to sequentially generate the set of m-trails. Details of this heuristic can be found in [20].

Note that the performance gain of m-trails over (nonsimple) m-cycles is achieved by removing the cycle constraint. In engineering practice, it is possible that a cyclic monitoring structure is preferred. For example, if we hope that the same node can transmit and receive the supervisory optical signal to facilitate a signal comparison, then (nonsimple) m-cycle is desired. It is also possible that we need to consider additional cost for separating the transmitter and receiver of the supervisory optical signal in m-trails, but such cost considerations (if any) can be easily incorporated by slightly modifying the ILP formulated in this paper.

VII. CONCLUSION

We proposed a new framework of all-optical monitoring, namely *monitoring trail* (m-trail), for fast link failure localization in all-optical WDM mesh networks. Compared with the existing monitoring cycle (m-cycle) method, m-trail provides a general all-optical monitoring structure by removing the cycle
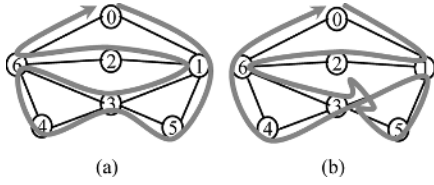
Fig. 12. Examples of complex m-trails.

constraint. As a result, m-trails can be taken as a generalization of simple and nonsimple m-cycles as well as open trails (including link-based monitoring), and an optimal m-trail solution can be obtained by jointly considering all these monitoring structures. Due to the flexible monitoring structure of m-trails, an m-trail solution can always achieve unambiguous link failure localization with the least amount of monitoring resources. We also formulated an efficient ILP for optimal m-trail design, with the objective of minimizing the overall monitoring cost in the network. Numerical results showed that the proposed m-trail scheme can significantly cut down the required monitoring cost compared with that by the m-cycle based counterpart.

APPENDIX
MORE ON VOLTAGE CONSTRAINT

In Figs. 5 and 6, we have illustrated how the voltage constraint works using some simple examples. Generally, a trail may have a much more complex pattern than those given in Figs. 5 and 6, as shown by the two examples in Fig. 12. We need to strictly prove that the voltage constraint can always properly work no matter how complex the pre-cross-connection pattern is. Due to the flow conservation property, it is not possible that a disjoint open trail without traversing the unique $T - R$ node pair can exist. So, we consider only closed trails (i.e., cycles) below.

The two closed trails (i.e., nonsimple cycles) in Fig. 12 have the same set of vectors but different pre-cross-connection patterns. Assume that both of them do not traverse sink R. We show that a voltage conflict must occur in either case. For simplicity, we consider only those nodes and vectors on the cycles by ignoring other parts of the network (if any).

Let q_{uv}^j be the voltage of a vector $u \to v$. In Fig. 12(a), we can translate the voltage constraint at each node as follows.

$$\begin{cases} q_{01}^j > q_{60}^j, & \text{at node } 0 \\ q_{15}^j + q_{12}^j > q_{01}^j + q_{31}^j, & \text{at node } 1 \\ q_{26}^j > q_{12}^j, & \text{at node } 2 \\ q_{34}^j + q_{31}^j > q_{53}^j + q_{63}^j, & \text{at node } 3 \\ q_{46}^j > q_{34}^j, & \text{at node } 4 \\ q_{53}^j > q_{15}^j, & \text{at node } 5 \\ q_{60}^j + q_{63}^j > q_{46}^j + q_{26}^j, & \text{at node } 6. \end{cases} \quad (25)$$

Adding up all the inequalities in (25), we have

$$q_{01}^j + q_{15}^j + q_{12}^j + q_{26}^j + q_{34}^j + q_{31}^j + q_{46}^j + q_{53}^j + q_{60}^j + q_{63}^j >$$
$$q_{60}^j + q_{01}^j + q_{31}^j + q_{12}^j + q_{53}^j + q_{63}^j + q_{34}^j + q_{15}^j + q_{46}^j + q_{26}^j. \quad (26)$$

In (26), each voltage value appears exactly once at both sides of ">." In fact, due to the flow conservation property, each node has the same number of inbound and outbound vectors, and any outbound vector of a particular node must be an inbound vector of one of its neighbors. For example, vector $0 \to 1$ in Fig. 12(a) is an outbound vector of node 0 but an inbound vector of node 1. As a result, q_{01}^j appears at the left-hand side of the first inequality in (25), but the right-hand side of the second inequality. Obviously, inequality (26) cannot hold because the two sums at both sides of ">" are exactly the same. Consequently, the set of all inequalities in (25) cannot hold at the same time, and thus there must be a voltage conflict.

Similar analysis can be applied to Fig. 12(b) to generate the same result. This is independent of the pre-cross-connection pattern of the trails. Generally, we can prove the theorem in Section IV-A (as also cited shortly).

Theorem: With the voltage constraint at each individual node, any cycle without traversing sink R will encounter a voltage conflict.

Proof: Consider an arbitrary cycle t_j with an arbitrary pre-cross-connection pattern. Let $\boldsymbol{V}_C$ be the set of all the nodes and $\boldsymbol{E}_C$ be the set of all the vectors traversed by t_j. Since t_j does not traverse sink R, the voltage constraint must be obeyed at each node in $\boldsymbol{V}_C$, or

$$\sum_{(u \to v) \in \boldsymbol{E}_C} q_{uv}^j > \sum_{(v \to u) \in \boldsymbol{E}_C} q_{vu}^j \qquad \forall u \in \boldsymbol{V}_C. \quad (27)$$

As a result, we have

$$\sum_{u \in \boldsymbol{V}_C} \sum_{(u \to v) \in \boldsymbol{E}_C} q_{uv}^j > \sum_{u \in \boldsymbol{V}_C} \sum_{(v \to u) \in \boldsymbol{E}_C} q_{vu}^j. \quad (28)$$

In (28), the term at the left-hand side of ">" is the voltage sum of all outbound vectors at all nodes $u \in \boldsymbol{V}_C$. Due to the flow conservation of the vectors in $\boldsymbol{E}_C$, each node $u \in \boldsymbol{V}_C$ must have the same number of inbound and outbound vectors, and each outbound vector of a particular node must be an inbound vector of one of its neighbors. Accordingly, summing up the voltage of all outbound vectors at all nodes is equivalent to summing up the voltage of all inbound vectors, or

$$\sum_{u \in \boldsymbol{V}_C} \sum_{(u \to v) \in \boldsymbol{E}_C} q_{uv}^j = \sum_{v \in \boldsymbol{V}_C} \sum_{(u \to v) \in \boldsymbol{E}_C} q_{uv}^j$$
$$= \sum_{u \in \boldsymbol{V}_C} \sum_{(v \to u) \in \boldsymbol{E}_C} q_{vu}^j. \quad (29)$$

However, (29) contradicts (28). Accordingly, the voltage constraint as formulated in (27) cannot be obeyed at every node $u \in \boldsymbol{V}_C$. In other words, there must be a voltage conflict.

REFERENCES

[1] M. W. Maeda, "Management and control of transparent optical networks," *IEEE J. Sel. Areas Commun.*, vol. 16, no. 7, pp. 1008–1023, Sep. 1998.
[2] I. Tomkos, "Dynamically reconfigurable transparent optical networking based on cross-layer optimization," in *Proc. 9th Int. Conf. Transparent Optical Networks (ICTON 2007)*, Jul. , vol. , p. 327.

[3] D. Z. Chen, G. Wellbrock, S. J. Penticost, D. Patel, C. Rasmussen, M. C. Childers, X. Yang, and M. Y. Frankel, "World's first 40 Gbps overlay on a field-deployed, 10 Gbps, mixed-fiber, 1200 km, ultra long-haul system," in *Proc. Opt. Fiber Commun. Conf. 2005. Technical Digest. OFC/NFOEC*, Mar. .

[4] C. Mas, I. Tomkos, and O. K. Tonguz, "Failure location algorithm for transparent optical networks," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 8, pp. 1508–1519, Aug. 2005.

[5] M. Goyal, K. K. Ramakrishnan, and W.-C. Feng, "Achieving faster failure detection in OSPF networks," in *Proc. IEEE Int. Conf. Communications (ICC 2003)*, May , vol. , pp. 296–300.

[6] P. Demeester, M. Gryseels, A. Autenrieth, C. Brianza, L. Castagna, G. Signorelli, R. Clemenfe, M. Ravera, A. Jajszczyk, D. Janukowicz, K. Van Doorselaere, and Y. Harada, "Resilience in multilayer networks," *IEEE Communications Mag.*, vol. 37, no. 8, pp. 70–76, Aug. 1999.

[7] Y. Hamazumi, M. Koga, K. Kawai, H. Ichino, and K. Sato, "Optical path fault management in layered networks," in *Proc. IEEE GLOBECOM 1998*, Nov. , vol. , pp. 2309–2314.

[8] C.-S. Li and R. Ramaswami, "Automatic fault detection, isolation, and recovery in transparent all-optical networks," *J. Lightw. Technol.*, vol. 15, no. 10, pp. 1784–1793, Oct. 1997.

[9] S. Stanic, S. Subramaniam, H. Choi, G. Sahin, and H.-A. Choi, "On monitoring transparent optical networks," in *Proc. Int. Conf. Parallel Processing Workshops*, Aug. 2002, pp. 217–223.

[10] S. Ramasubramanian, "Supporting multiple protection strategies in optical networks," *IEEE/ACM Trans. Netw.*, vol. 16, no. 6, pp. 1352–1365, Dec. 2008.

[11] Y. G. Wen, V. W. S. Chan, and L. Z. Zheng, "Efficient fault diagnosis algorithms for all-optical WDM networks with probabilistic link failures (Invited Paper)," *J. Lightw. Technol.*, vol. 23, no. 10, pp. 3358–3371, Oct. 2005.

[12] C. Assi, Y. Ye, A. Shami, S. Dixit, and M. Ali, "A hybrid distributed fault-management protocol for combating single-fiber failures in mesh-based DWDM optical networks," in *Proc. IEEE GLOBECOM 2002*, Nov. , vol. , pp. 2676–2680.

[13] H. Zeng, C. Huang, A. Vukovic, and M. Savoie, "Fault detection and path performance monitoring in meshed all-optical networks," in *Proc. IEEE GLOBECOM*, 2004, pp. 2014–2018.

[14] H. Zeng, C. Huang, and A. Vukovic, "A novel fault detection and localization scheme for mesh all-optical networks based on monitoring-cycles," *Photon. Netw. Commun.*, vol. 11, no. 3, pp. 277–286, May 2006.

[15] H. Zeng and A. Vukovic, "The variant cycle-cover problem in fault detection and localization for mesh all-optical networks," *Photon. Netw. Commun.*, vol. 14, no. 2, pp. 111–122, 2007.

[16] B. Wu and K. L. Yeung, "M^2-cycle: An optical layer algorithm for fast link failure detection in all-optical mesh networks," in *Proc. IEEE GLOBECOM 2006*, pp. 1–5.

[17] B. Wu, K. L. Yeung, and P.-H. Ho, "Monitoring cycle design for fast link failure localization in all-optical networks," *J. Lightw. Technol.*, vol. 27, no. 10, pp. 1392–1401, May 2009.

[18] B. Wu, K. L. Yeung, and P.-H. Ho, "A comparative study of fast protection schemes in WDM mesh networks," *Proc. IEEE Int. Conf. Communications (ICC 2008)*, pp. 5160–5164.

[19] (2009). [Online]. Available: [Online]. www.ilog.com

[20] J. Tapolcai, B. Wu, and P.-H. Ho, "On monitoring and failure localization in mesh all-optical networks," in *Proc. IEEE INFOCOM 2009*, Apr. 2009, pp. 1008–1016.

Bin Wu (S'04–M'07) received the B.Eng. degree from Zhe Jiang University, Hangzhou, China, in 1993, the M.Eng. degree from the University of Electronic Science and Technology of China, Chengdu, China, in 1996, and the Ph.D. degree from The University of Hong Kong, Pokfulam, Hong Kong, in 2007.

During 1997–2001, he was the Department Manager of Texas Instruments (TI)-Huawei Digital Signal Processor (DSP) Co-Lab in Huawei Technologies Company Ltd., Shenzhen, China. He is currently a Postdoctoral Research Fellow at the University of Waterloo, Waterloo, ON, Canada.

Pin-Han Ho received the B.Sc. and M.Sc. degrees in electrical and computer engineering from the Department of Electrical and Computer Engineering, National Taiwan University, Taipei, Taiwan, in 1993 and 1995, respectively, and the Ph.D. degree, in 2002, from Queen's University, Kingston, ON, Canada, where he was engaged in research on optical communications systems, survivable networking, and QoS routing problems.

In 2002, he joined the Electrical and Computer Engineering Department at the University of Waterloo, Waterloo, ON, Canada, as an Assistant Professor. He is the author or coauthor of more than 100 refereed technical papers and book chapters, and the coauthor of a book on optical networking and survivability.

Dr. Ho is the recipient of the Distinguished Research Excellence Award in the ECE Department at the University of Waterloo, the Early Researcher Award in 2005, the Best Paper Award at International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS 2002) and International Conference on Communications (ICC 2005), Optical Networking Symposium, and the Outstanding Paper Award in High Performance Switching and Routing (HPSR 2002).

Kwan L. Yeung (S'93–M'95–SM'99) was born in 1969. He received the B.Eng. and Ph.D. degrees in information engineering from The Chinese University of Hong Kong, Pokfulam, Hong Kong, in 1992 and 1995, respectively.

He was an Assistant Professor in the Department of Electronic Engineering, City University of Hong Kong, Kowloon, Hong Kong, where he has spent five years. During the summer of 1993, he was with the Performance Analysis Department, American Telephone and Telegraph Company (AT&T) Bell Laboratories (now Bell Labs, Lucent Technologies), Holmdel, NJ, as a Member of Technical Staff. In July 2000, he joined the Department of Electrical and Electronic Engineering, The University of Hong Kong, where he is currently an Associate Professor and the Information Engineering Program Codirector. He is the author or coauthor of more than 120 papers published in international journals and conferences. He is the holder of two patents. His current research interests include next-generation Internet, packet switch/router design, all-optical networks, and wireless data networks.