SRLG Failure Localization in Optical Networks

Satyajeet S. Ahuja, Srinivasan Ramasubramanian, and Marwan Krunz
Department of ECE, University of Arizona, Tucson, AZ 85721
{ahuja, srini, krunz}@ece.arizona.edu

***Abstract*—We introduce the concepts of monitoring paths (MPs) and monitoring cycles (MCs) for unique localization of shared risk linked group (SRLG) failures in all-optical networks. An SRLG failure causes multiple links to break simultaneously due to the failure of a common resource. MCs (MPs) start and end at the same (distinct) monitoring location(s). They are constructed such that any SRLG failure results in the failure of a unique combination of paths and cycles. We derive necessary and sufficient conditions on the set of MCs and MPs needed for localizing any single SRLG failure in an arbitrary graph. When a single monitoring location is employed, we show that a network must be $(k + 2)$-edge connected for localizing all SRLG failures, each involving up to k links. For networks that are less than $(k+2)$-edge connected, we derive necessary and sufficient conditions on the placement of monitoring locations for unique localization of any single SRLG failure of up to k links. We use these conditions to develop an algorithm for determining monitoring locations. We show a graph transformation technique that converts the problem of identifying MCs and MPs with multiple monitoring locations to a problem of identifying MCs with a single monitoring location. We provide an integer linear program and a heuristic to identify MCs for networks with one monitoring location. We then consider the monitoring problem for networks with no dedicated bandwidth for monitoring purposes. For such networks, we use passive probing of lightpaths by employing optical splitters at various intermediate nodes. Through an integer linear programming formulation, we identify the minimum number of optical splitters that are required to monitor all SRLG failures in the network. Extensive simulations are used to demonstrate the effectiveness of the proposed monitoring technique.**
Index Terms: Network Monitoring, Failure localization, SRLG failure, Optical networks.

I. Introduction

Optical networks have gained tremendous importance due to their ability to support very high data rates using the dense wavelength division multiplexing (DWDM) technology. At such high rates, a brief service disruption in the operation of the network can result in the loss of a large amount of data. Commonly observed service disruptions are caused by fiber cuts, equipment failure, excessive bit errors, intrusion, and human error. To ensure robust network operation, it is highly desired that these faults be *uniquely* identified and corrected at the physical layer before they are noticed at higher layers. Therefore, it is critical for optical networks to employ fast and effective methods for detecting and localizing network failures. Some failures, such as optical cross-connect port blocking and intrusion, can affect a single or a specific subset of wavelengths within a link. Other failures, including fiber cuts and high bit error rates (BERs), may affect all the wavelengths that pass through a link. In this paper, we focus on the latter type of failures. In addition, links in an optical network may share a common resource, such as a duct or conduit through which multiple links are laid out. The failure of this resource results in the simultaneous failure of multiple links. Such failures are referred to as *Shared Risk Link Group* (SRLG) failures [14].

Failure detection and localization may be performed either at the physical or the IP layer. Routing protocols at the IP layer, such as OSPF, often have an inherent failure detection capability. However, such a capability suffers from long detection time (a few seconds), and hence are not suitable for networks that require fast recovery. Some parameters in OSPF can be optimized to achieve fast failure detection [8]. A cross-layer (optical/IP) recovery method was proposed in [3], but its relatively slow recovery time prohibits its use for failure detection in all-optical networks.

Optical-level mechanisms for single-link failure detection were proposed in the literature [9], [3]. These include optical spectral analysis, optical power detection, pilot tones, and optical time domain reflectometry (OTDR). In [9], a failure detection scheme was proposed, in which monitors are assigned to each optical multiplexing and transmission section. Such a scheme requires an excessive number of inherently expensive monitors, and is not scalable and cost-effective for large-scale networks. In [16], an adaptive technique for fault diagnosis using "probes" was presented. According to this scheme, probes are established sequentially, each time using information about already established probes. While sequential probing helps achieve adaptiveness, it also increases the fault localization time. In [10], a non-adaptive fault diagnosis through a set of probes (lightpaths) was developed, where all the probes are employed in advance. The techniques presented in [16] and [10] assume that a probe can originate and terminate at any location, i.e., there is no restriction on the number of monitoring locations. In the worst case, all the nodes in the network need to have monitors, resulting in high setup cost and protocol overhead. In [2], a failure localization techinque was presented for single link failures by using monitoring cycles and paths.

Monitoring trails were introduced in [13], [17] for localizing link failures in all-optical networks. A monitoring trail is a non-simple light-path that is associated with a monitor at the receiver. A monitor triggers an alarm after identifying a link failure and flood the network with the alarm in the control domain. A remote network entity reads the alarm pattern to localize link failure. Note that such a technique is based on alarm flooding and is likely to be slow.

A practical approach to fault detection and localization that

This research was supported in part by NSF (under grants CNS-1016943, CNS-0435490, CNS-0904681, and IIP-0832238), Raytheon, and the "Connection One" center. Any opinions, findings, conclusions, or recommendations expressed in this paper are those of the author(s) and do not necessarily reflect the views of the NSF. An abridged version of this paper was presented at the *IEEE INFOCOM Conference*, Phoenix, Apr. 15-17, 2008.

we advocate in this paper is to equip only a few nodes in the network with monitors, referred to as *monitoring locations*. SRLG failures can then be uniquely localized using monitoring paths (MPs) and monitoring cycles (MCs). An MP starts and ends at distinct monitoring locations, whereas an MC starts and ends at a given monitoring location. The MPs and MCs should be selected such that the failure of any SRLG results in the failure of a unique combination of MPs and MCs. One wavelength may be dedicated for each MC/MP on every link the MC/MP traverses, or a single wavelength may be time-shared by all MCs/MPs. Similarly, the monitoring location may employ multiple monitors (one for each MC/MP) or it may use one monitor that is time-shared by various MCs/MPs. For networks that can provide only a small amount of bandwidth for monitoring, the network operator cannot reserve MCs/MPs between different monitoring locations. Instead, the operator can exploit the failure information of lightpaths carrying traffic and can monitor the status of lightpaths at various intermediate nodes using optical splitters.

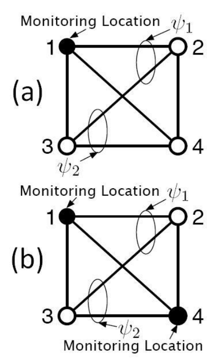
It should be noted that we do not allow a MC/MP to traverse a link twice. This is done for two reasons: First, we want to exploit already established (operational) lightpaths that carry user payload for monitoring purposes, so as to reduce the monitoring overhead. Such light-paths never traverse a link in both directions. Second, in many practical scenarios, the two directions of a link are highly correlated in terms of their failure patterns (i.e., if one direction fails, the other is also likely to fail). This is the case when the two directions operate on separate fibers, but which belong to the same duct. Correlation between the failures of the two links means that incorporating a reverse direction, will not change the syndrome of a given SRLG (i.e., if a MC/MP traverses a link multiple times, it does not provide any extra information that can be use for failure localization of that link), but at the same time will likely result in longer MCs/MPs (more monitoring wavelengths per link).

Although we focus on failure detection, our techniques may also be used to assess other performance metrics, such as optical power, optical signal-to-noise ratio (OSNR), and BER. Because a combination of MPs and MCs is unique to a given SRLG, the observed phenomenon (link failure, low optical power, etc.) can be uniquely attributed to that SRLG.

The contributions of this paper are as follows: (1) we develop necessary and sufficient conditions on the set of MCs and MPs needed for localizing SRLG failures in arbitrary topologies (Section II); (2) we prove that a network must be $(k+2)$-edge-connected in order to localize any single SRLG failure of up to k links using one monitoring location (Section II); (3) we develop necessary and sufficient conditions on the placement of monitoring locations for unique localization of all SRLG failures of up to k links (Section III); (4) we derive the minimum number of monitors required to localize any SRLG failure of up to k links in an arbitrary topology (Section IV); (5) we develop an algorithm for the placement of the minimum number of monitors (Section IV); (6) we develop an integer linear programming (ILP) formulation and a heuristic solution for selecting monitoring cycles in networks with one monitoring location (Section V); (7) we use a graph transformation technique to solve the problem of selecting MCs and MPs in a network that employs multiple monitoring locations, and we use this technique to show that the problem of selecting MCs/MPs with multiple monitoring locations can be reduced into an MC selection problem in the transformed graph with one monitoring location; and (8) we develop a monitoring technique for optical networks with no dedicated bandwidth for monitoring purposes. We employ optical splitters to passively monitor different lightpaths at various locations. To this end, we develop an ILP formulation to minimize the number of optical splitters and establish the conditions for the placement of optical splitters at intermediate nodes.

II. Necessary and Sufficiency Conditions for Localizing SRLG Failures

Consider an all-optical network whose topology is modeled as a graph $\mathcal{G}(\mathcal{N}, \mathcal{L})$, where $\mathcal{N}$ is the set of nodes and $\mathcal{L}$ is the set of links. Let Ψ denote the set of all distinct SRLGs in the network. An element $\psi \in \Psi$ is a subset of $\mathcal{L}$. It denotes a set of links that would fail simultaneously due to the failure of a shared resource. The primary objective of the monitoring approach is to minimize the number of monitor required for failure localization. For a given placement of monitors, the objective is to localize[1] an SRLG failure by observing failed MPs and MCs. An MC/MP is affected by an SRLG failure ψ if it passes through at least one link $\ell \in \psi$. We refer to the set of monitoring paths and cycles that can uniquely localize all SRLG failures as the *fault localization* (FL) set. Note that, while constructing an FL set, we do not prefer MCs (MPs) over MPs (MCs). The default case of no SRLG failure is indicated by no failures in any of the paths or cycles in the FL set. Hence, our approach also detects the case of no failure in the network. By design, every SRLG failure must impact at least one MP or MC. In addition, an SRLG failure should result in a unique *syndrome* – a combination of failed MPs and MCs.



SRLG	c_1	c_2	c_3	c_4	p_1	p_2	p_3	p_4
ψ_1	×	×	×		×		×	×
ψ_2	×	×		×		×	×	×
$(1,2)$	×		×		×		×	
$(2,3)$	×	×					×	×
$(3,4)$	×			×		×	×	
$(1,4)$	×	×	×	×				×
$(2,4)$		×	×		×			×
$(1,3)$		×		×		×		×

(c)

Fig. 1. Example network topology, SRLG failures, and associated syndromes.

To illustrate, consider the example in Figure 1, where $\psi_1 = \{(1,2),(2,3)\}$ and $\psi_2 = \{(2,3),(3,4)\}$ are the two SRLGs to be monitored. When node 1 is the monitoring location (part (a) of Figure 1), an FL set can be obtained using four MCs: $c_1 =$

[1]Localizing an SRLG failure means identifying the shared resource that failed and in return caused the failure of links that are associated with the SRLG.

(1–2–3–4–1), c_2 = (1–3–2–4–1), c_3 = (1–2–4–1), and c_4 = (1–3–4–1). If nodes 1 and 4 are both monitoring locations (part (b) of Figure 1), then an FL set can be obtained using three MPs and one MC: p_1 =(1–2–4), p_2 = (1–3–4), p_3 = (1–2–3–4), and p_4 = (1–3–2–4–1). The associated syndromes are shown in part (c) of the Figure 1.

Theorem 1: The necessary and sufficient conditions for the existence of an FL set that can uniquely localize any single SRLG failure from a given set Ψ of SRLG failures in an arbitrary network are:

1) Each SRLG failure affects at least one monitoring path or cycle; and
2) For any two SRLGs ψ_1 and ψ_2 in Ψ, there exists a path or cycle in the FL set that is affected by ψ_1 and not ψ_2 or vice versa.

Proof: The necessity aspect is proved by contradiction. Assume to the contrary that there is a feasible solution but at least one of the two conditions presented is not satisfied. First assume that condition 1 is not satisfied for an SRLG ψ_i. Then, no MC or MP passes through any link in ψ_i, and hence the failure of ψ_i cannot be monitored. Thus, any feasible solution must satisfy condition 1. If condition 2 is not satisfied for a pair of SRLGs ψ_1 and ψ_2, then every cycle or path that is affected by ψ_1 is also affected by ψ_2, contributing to the syndromes of both ψ_1 and ψ_2. Thus, the failure of these two SRLGs cannot be uniquely identified, leading to a contradiction. Therefore, a feasible solution may be obtained only if the two conditions presented above are satisfied.

We prove the sufficiency part by construction. For any two SRLGs ψ_1 and ψ_2 in the network, we define three types of paths and cycles: $\mathcal{T}_1$: Set of paths and cycles affected by ψ_1 but not ψ_2. $\mathcal{T}_2$: Set of paths and cycles affected by ψ_2 but not ψ_1. $\mathcal{T}_3$: Set of paths and cycles affected by both ψ_1 and ψ_2.

To distinguish between the failure of ψ_1 and the failure of ψ_2, an FL set must have paths and/or cycles that fall in at least two of these three types. For example, if the FL set contains a cycle $c_1 \in \mathcal{T}_1$ and a cycle $c_2 \in \mathcal{T}_3$, then the failure of ψ_1 will result in the failure of c_1 and c_2, while the failure of ψ_2 will result in the failure of c_2 only. Based on condition 2, we know that cycles or paths of type $\mathcal{T}_1$ or $\mathcal{T}_2$ must exist. Assume that a monitoring path/cycle of type $\mathcal{T}_1$ exists for SRLGs ψ_1 and ψ_2. Given the monitoring locations, a path or cycle of type $\mathcal{T}_1$ may be obtained as follows: First, merge all the monitoring nodes to form one super-node m[2]. This transformation may result in multiple links from a given node to the super-node m. In addition, any link between two monitoring locations will transform into a loop at node m. Second, remove links belonging to SRLG ψ_2 from the network. Third, if any of the links in $\psi_1 \setminus \psi_2$ forms a loop at node m, then the loop is a (one link) path of type $\mathcal{T}_1$. Otherwise, construct a cycle traversing node m and link $\ell \equiv (x_\ell, y_\ell) \in \psi_1 \setminus \psi_2$, as shown in Figure 2. Such a cycle must exist for at least one link, due to the necessity of condition 2. When the merged super node is expanded, the cycle either remains a cycle or transforms into a monitoring path. Similarly, if the MC/MP of type $\mathcal{T}_2$ exists, then it can be constructed using the above method by interchanging ψ_1 and ψ_2. We now have at least one cycle/path of type $\mathcal{T}_1$ or $\mathcal{T}_2$.

1) Add a virtual node v and two virtual links (v, x_ℓ) and (v, y_ℓ). An example for virtual link construction for link $(1, 3)$ is shown in the figure below.
2) Remove link ℓ.
3) Obtain two link-disjoint paths $\mathcal{P}_1$ and $\mathcal{P}_2$ from v to m. For the example topology shown below, $\mathcal{P}_1$ and $\mathcal{P}_2$ are green paths.
4) Form a cycle c by joining $\mathcal{P}_1$ and $\mathcal{P}_2$ after removing the virtual links and adding the link ℓ.

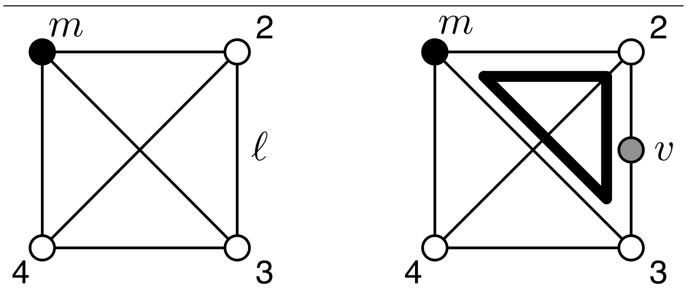


Fig. 2. Procedure and example to obtain a cycle that traverses a given node m and link ℓ.

Finally, assume that for SRLGs ψ_1 and ψ_2 we have a cycle/path of type $\mathcal{T}_1$ but not $\mathcal{T}_2$[3]. Then, in order to satisfy condition 1, there must be a cycle/path of type $\mathcal{T}_3$ (otherwise ψ_2 will not be covered by any MC/MP). An MC/MP of type $\mathcal{T}_3$ can be obtained using the procedure presented in Figure 2 by trying all links $\ell \in \psi_2$. In this case, we do not remove links in ψ_1. By adding MCs/MPs of at least two of the types $\mathcal{T}_1$, $\mathcal{T}_2$, and $\mathcal{T}_3$ for each SRLG pair (ψ_i, ψ_j), we can construct a solution. ■

The procedure to place monitoring locations in the network must enable the generation of MCs and MPs that satisfy the conditions in Theorem 1. Before presenting such a procedure, we need to identify the number of monitoring locations required to localize the SRLG failures in Ψ. We first establish the necessary and sufficient conditions for localizing SRLG failures using only one monitoring location. We assume that the SRLG failures contain all possible link failures involving up to k links.

Theorem 2: (k+2)-edge connectivity is a necessary and sufficient condition to localize all possible failures involving up to k links with one monitoring location.

Proof: We prove the necessary part by contradiction. Assume to the contrary that there is an FL set with one monitoring location for a network that is (k+1)-edge but not (k+2)-edge connected. Note that because there is only one monitoring location, the FL set must consist of MCs only. Since the network is not (k+2)-edge-connected, there exists a set of k+1 links whose removal disconnects the network, as described in Figure 3. We number these links as $\ell_1, \ell_2, ..., \ell_{k+1}$. Consider now two SRLGs $\psi_1 \equiv \{\ell_1, \ell_2, ..., \ell_k\}$ and $\psi_2 \equiv \{\ell_2, \ell_3, ..., \ell_{k+1}\}$. Notice that any cycle that traverses through one or more links in ψ_1 must also pass through one or more links in ψ_2. Thus, the failure of

[2] An example of merge operation for the network in Figure 4 is presented in Figure 6.

[3] The case of $\mathcal{T}_2$ but not $\mathcal{T}_1$ is similar to the case of $\mathcal{T}_1$ but not $\mathcal{T}_2$ and hence can be constructed by following the same argument.

any of these two SRLGs cannot be uniquely identified through MCs, leading to a contradiction. Therefore, an FL set cannot be obtained in a (k+1)-edge-connected network, i.e., (k+2)-edge connectivity is a necessary condition for the existence of a solution.

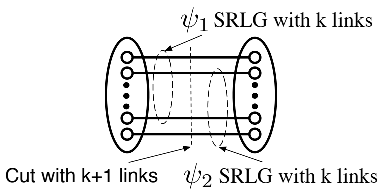


Fig. 3. A cut in the graph with $k+1$ links.

We prove the sufficiency part by construction. For any two SRLGs ψ_1 and ψ_2 in the network, consider the cycles of types $\mathcal{T}_1$, $\mathcal{T}_2$, and $\mathcal{T}_3$, as defined in the proof of Theorem 1. Since the network is (k+2)-edge connected, the removal of any SRLG leaves the network at least two-edge connected. In the remaining network we can find MCs passing through any link ℓ using the procedure presented in Figure 2. Hence, cycles of types $\mathcal{T}_1$ and $\mathcal{T}_2$ can be constructed easily for any two SRLGs ψ_i and ψ_j. ■

Corollary 1: Three-edge connectivity is a necessary and sufficient condition to uniquely localize any *single-link* failure with one monitoring location.

III. Necessary and Sufficient Conditions for Monitor Placement

We have shown that with the use of one monitoring location, (k+2)-edge connectivity is necessary and sufficient to uniquely localize all possible failures involving up to k links. Even to identify single-link failures, the network needs to be at least three-edge connected. However, many optical networks in real life may not satisfy such high connectivity requirement and hence cannot be monitored using a single monitoring location. We now give an example to show that with multiple monitoring locations and by employing monitoring paths (MPs) along with MCs, all SRLGs in an arbitrarily connected network can be monitored. Consider the network in Figure 4. Because the network is not three-edge connected, single-link failures cannot be monitored using a single monitoring location. With two monitoring locations at node 1 and 6, an FL set consisting of six cycles c_1 = 1–2–3–4–1, c_2 = 1–3–2–4–1, c_3 = 1–2–4–1, c_4 = 6–5–8–7–6, c_5 = 6–8–5–7–6, c_6 = 6–5–7–6 and a path p_1 = 1–2–5–6 can localize any single-link failure. To identify an SRLG failure, the monitoring locations must share information about failed MCs and MPs.

Theorem 3: To uniquely localize all possible failures involving up to k links in an arbitrary network, it is necessary and sufficient that every component of the disconnected graph that results from the removal of any set of $k+1$ links contains a monitoring location.

Proof: For the necessary part of the proof, consider the removal of any k+1 links $\ell_1, \ell_2, ..., \ell_{k+1}$ from $\mathcal{G}$. The removal of these links can reduce the graph into one, two, ..., or $k+2$ connected components. Let the removal of these links result in $\mathcal{R}$ components $C_1, C_2, ..., C_{\mathcal{R}}$ of respective degrees $d_1, d_2, ..., d_{\mathcal{R}}$ (The degree of a component C_i is the number edges that are incident on nodes in C_i from nodes outside of C_i). Notice that $0 \leq d_i \leq k+1$ for all i. If the removal of these links results in one connected component of degree zero, then this component must contain one monitoring location; otherwise, the network will be without any monitoring location. If it results in more than one connected component, then consider any such component C_i of degree $d_i \leq k+1$. Suppose that C_i is connected to other components using links $\ell_1, \ell_2, ..., \ell_{d_i}$, as shown in Figure 5.

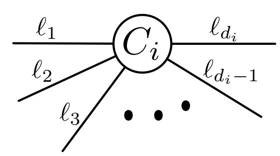


Fig. 5. Connected component C_i of degree d_i.

We now show that any feasible placement of monitors must include a monitoring location inside C_i. An SRLG may contain up to k links that can fail simultaneously. Consider two SRLGs with $d_i - 1$ links: $\psi_1 \equiv \{\ell_1, ..., \ell_{d_i-1}\}$ and $\psi_2 \equiv \{\ell_2, ..., \ell_{d_i}\}$, $d_i \leq k+1$. If C_i does not have a monitoring location, then any MP/MC that passes through ψ_1 must also pass through ψ_2 because an MC/MP cannot terminate inside C_i. Hence, the failure of ψ_1 and ψ_2 cannot be distinguished without placing a monitor inside C_i. The above argument is true for any connected component of degree $\leq k$+1. Because the maximum degree of a component is at most $k+1$, every component that results after the removal of k+1 links must have a monitoring location.

For the sufficiency part, consider a set of monitoring locations $\mathcal{M}$ that satisfy the necessary condition. Construct a transformation graph by merging all the nodes $m \in \mathcal{M}$ into a *central monitoring node* $\mathcal{J}$. For each link (m, v) with $m \in \mathcal{M}$, add a link $(\mathcal{J}, v)$. The graph transformation of the example graph in Figure 4 is shown in Figure 6. Notice that since the set of monitoring locations satisfies the necessary condition, the removal of any k+1 links does not disconnect

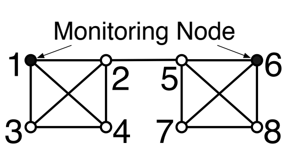


Fig. 4. Example graph to illustrate monitoring with two monitoring locations.

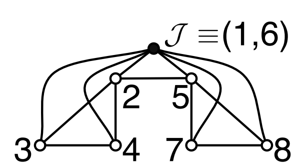


Fig. 6. Transformation graph for the example network in Figure 4.

the transformation graph[4]. Hence, the transformation graph is (k+2)-edge connected. Finally, by using Theorem 2, we conclude that such a placement is sufficient for monitoring an arbitrary set of k-link failures. A monitoring cycle in the transformed graph will be either a monitoring cycle or a monitoring path in the original graph. ■

Corollary 2: To distinguish among arbitrary k-link failures in a tree network, every node with degree $\leq k+1$ must be a monitoring location.

Corollary 3: To uniquely identify all single-link failures in a tree network, all the nodes of degree one or two must be monitoring locations.

Corollary 4: To uniquely identify all single-link failures in line and ring networks, all the nodes must be monitoring locations.

Corollary 5: To uniquely identify all possible failures involving up to k links in a network, every j-edge-connected component ($j \leq k+2$) of degree $k+1$ or less must have a monitoring location.
The above corollaries can be easily derived using Theorem 3.

IV. Monitor Placement Algorithm For up to k Link Failures in Arbitrary Networks

In this section, we provide a monitor placement algorithm for localizing up to k-link failures in an arbitrarily connected network. Before proceeding further, we define the following terms:

1) The *k-super-graph* (or simply *k-graph*) of a network graph $\mathcal{G}(\mathcal{N}, \mathcal{L})$ is a graph that represents the interconnectivity of the k-edge-connected components of $\mathcal{G}$ [11]. In other words, each node i of the k-graph is a k-edge-connected component of $\mathcal{G}$.
2) A *connector link* $\ell \in \mathcal{L}$ is a link that connects nodes belonging to two different k-edge-connected components of $\mathcal{G}$. Such a link is also a link of the k-graph.

For a given network $\mathcal{G}$ and a given integer k, the k-graph of the network is unique [11]. The k-graph can be at most $(k-1)$-edge-connected [11], [7], [12]. In this paper, we employ 2-, 3-, ..., (k+2)-graphs to determine the minimum number of monitors required to uniquely localize up to k-link failures.

Using the corollaries described above, the minimum number of required monitoring locations in an arbitrary network that can have up to k-link failures can be obtained using the `Placement` procedure described in Figure 7. In Step 2 of this procedure, the degree of a k-edge-connected component is that of the corresponding node on the k-graph, $k \geq 2$.

An example of the 2-graph and the 3-graph is shown in Figure 8.

Notice that the 2-graph in this example is a tree. To localize single-link failures, each three-edge-connected component of degree two or less must have at least one monitoring location inside it. These monitoring locations are nodes in Figure 8 with number 3. Similarly, each two-edge-connected component of degree two or less must have a monitoring location inside it. For example, component A is a two-edge-connected component of degree two[5]. Hence, it needs at least one monitoring location inside it. However, all three-edge-connected components within A itself have degree three and according to step 2, they do not need a monitoring location. Hence, any node is picked in Step 4(b) for $j = 2$ as a monitoring location inside A. Similarly, component B has degree two but a monitoring location is not assigned to it in Step 4(b) for $j = 2$ because it already gets a monitoring location in one of its three-edge-connected components in Step 2.

Procedure `Placement`

1) Obtain the $(k+2)$-graph of the original network graph $\mathcal{G}(\mathcal{N}, \mathcal{L})$.
2) Assign a monitoring location to every $(k+2)$-edge-connected component of degree $k+1$ or less.
3) $j = k+1$
4) While $j \geq 2$
   a) Obtain the j-graph by merging the $(j+1)$-edge-connected components.
   b) Assign a monitoring location to every j-edge-connected component of degree $k+1$ or less if such a component does not contain a monitoring location.
   c) $j = j-1$

Fig. 7. Procedure for finding the minimum number of monitoring locations and their placement for an arbitrary network with up to k-link failures.

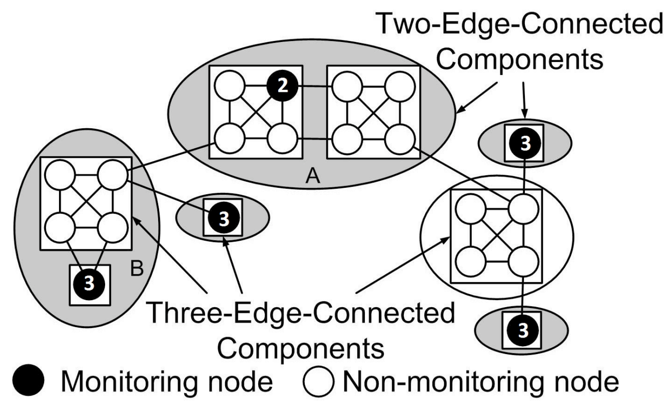


Fig. 8. An example showing the 2-graph, 3-graph, and the monitor placement in an arbitrarily connected network with single-link failures. Two-edge (3-edge) connected components are surrounded by ovals (squares).

Complexity: Unique decomposition of a graph into its $2-$graph , $3-$graph can be done in $\mathcal{O}(|\mathcal{L}|)$ time [11]. For all k-graphs, $k > 3$, we first obtain the maximum number of link-disjoint paths[6] between every pair of nodes in the graph. We maintain a two-dimensional array that stores the number of

[4] The removal of k+1 links would result in a maximum of k+2 components, each having at least one monitoring location. Because monitoring locations are merged to form $\mathcal{J}$, the transformation graph remains connected.

[5] This monitoring location is assigned with number 2 in Figure 8.

[6] If maximum number of link-disjoint paths is more than k, then find only first k-disjoint paths.

disjoint paths found between a node pair. A node is associated with a k-graph if it has more than k disjoint paths to all other nodes in the k-graph. Such a collection of nodes can now be identified using table lookup. Identifying k link-disjoint paths between all node pairs can be done in $\mathcal{O}(k|\mathcal{N}|^4)$. Identifying the degree of a component by identifying its outgoing edges can be done using $\mathcal{O}(|\mathcal{L}|)$ computations. Hence, the total complexity of the algorithm is $\mathcal{O}(k|\mathcal{N}|^4)$ for localizing up to k-link failures.

Theorem 4: The monitoring locations placed according to the `Placement` procedure are sufficient to uniquely identify all k-link failures.

Proof: The proof is by contradiction. Assume to the contrary that the placement of monitoring locations provided by the `Placement` procedure does not satisfy the necessary and sufficient condition of Theorem 3 for a set of $k+1$ links $\{\ell_1,\ \ell_2, ...,\ \ell_{k+1}\}$. Assume that the removal of these links creates $\mathcal{R}$ connected components $C_1,\ C_2,\ ...,\ C_{\mathcal{R}}$ of respective degrees $d_1,\ d_2, ..., d_{\mathcal{R}}$. Notice that $d_i \leq k+1\ \forall j \in \{1, 2, ..., \mathcal{R}\}$. Assume that the `Placement` procedure does not provide a monitor to component C_i. We show that this assumption leads to a contradiction.

We show during Step 4(b) that the `Placement` procedure assigns at least one monitoring location inside component C_i. Then,

1) If C_i is a single node of degree $d_i \leq k+1$, then it gets assigned monitor in Step 2.
2) If C_i is a one-edge-connected component consisting of at least two nodes, then in the 2-graph, it will form a tree of two-edge-connected components. If C_i forms a tree of n two-edge-connected components, then the sum of the degree of all the components is $2(n-1)+d_i$. Now if the `Placement` procedure does not assign a monitoring location to any of the two-edge-connected components, then all these components must have degrees $\geq k+2$. Hence, the sum of their degree is $2(n-1)+d_i \geq n(k+2)$, which is not true for any $n \geq 2$ because $d_i \leq k+1$. Therefore, at least one monitor must be assigned to C_i.
3) If C_i is r-edge-connected, where $r \geq 2$, then it is a node in a j-graph, where $j \geq d_i$. Hence, C_i gets a monitor because its degree is $\leq k+1$.

In summary, for all possible cases, component C_i gets a monitoring location. Note that the above procedure places exactly one monitor if the network is $(k+2)$-edge connected. ■

We now determine the number of monitoring locations provided by the `Placement` procedure. Let N_i denote the number of i-edge-connected components of degree $k+1$ or less that do not have inside them any $(i+j)$-edge-connected components $(j>0, i+j \leq k+2)$ of degree $k+1$ or less . The number of monitoring locations provided by the `Placement` algorithm is given by $\sum_{i=2}^{k+2} N_i$.

Correctness and Minimality: Given the j-graph of a graph $\mathcal{G}$ and using Corollary 5, any feasible set of monitoring locations must have at least one monitoring location in each of the j-edge-connected components of degree k+1 or less. This is ensured by Step 4 of the `Placement` procedure. Step 4 does not assign a monitoring location inside an i-edge-connected component if at least one monitoring location is already present inside this component, i.e. this component has a monitoring location because one of the $(i+j)$-edge-connected component $(j>0)$ inside it required a monitoring location in the earlier iterations of the `Placement` procedure. Finally, we also know that the k-graph of a graph $\mathcal{G}$ is unique [7], [11], [12]. Hence, the `Placement` procedure provides the minimum number of monitoring locations.

A. Graph Transformation

We now provide a graph transformation for constructing MCs/MPs to localize SRLG failures in an arbitrary connected graph $\mathcal{G}$. Given the set of monitoring locations $\mathcal{M}$, we introduce a *central monitoring node* $\mathcal{J}$ by merging all the monitoring locations in $\mathcal{M}$. The graph transformation of the network in Figure 4 is presented in Figure 6. We can now solve a simple version of the monitoring problem involving a single monitoring location $\mathcal{J}$. Note that such a transformation will provide a feasible solution only if the monitoring locations satisfy the necessary and sufficient conditions. Moreover, as proved in Theorem 3, if a set of monitoring locations satisfies these conditions, the transformation graph is $(k+2)$-edge connected.

Note that the merging of monitoring locations may result in self-loops at $\mathcal{J}$ (due to links between the monitoring locations) and multiple links between the monitoring node and a non-monitoring node. A monitoring cycle in the transformed graph will result in a monitoring cycle or a monitoring path in the original graph. Thus, it is sufficient to develop algorithms for identifying monitoring cycles for networks with one monitoring location only. It is worth noting that when the network employs multiple monitoring stations, knowledge of MP/MC failures must be shared among all monitoring locations to localize SRLG failures.

V. Identifying Monitoring Cycles in Networks with One Monitoring Location

We now consider the situation when only one monitoring location is used to localize SRLG failures. The goal is to identify the MCs for the FL set.

Problem MC-1 (Monitoring Cycle Problem With One Monitoring Location): Given a (k+2)-edge-connected network graph $\mathcal{G}(\mathcal{N}, \mathcal{L})$, a monitoring location $m \in \mathcal{N}$, and a set of SRLG failures Ψ, find a set of MCs $\mathcal{C}$ such that every SRLG failure affects at least one MC and every SRLG failure results in the failure of a unique subset of cycles in $\mathcal{C}$.

A. Bounds on the Required Number of Cycles

Let $T(\Psi)$ be the total number of MCs and MPs required to monitor Ψ SRLG failures. A failure localization technique needs to detect all SRLG failures and the no-failure scenario. A network with $|\Psi|$ SRLGs requires at least $\lceil \log_2(|\Psi|+1) \rceil$ cycles for fault localization for any arbitrary choice of the monitoring node. We assume that, at most one SRLG failure can occur at a time. When there is at most one SRLG failure, an MC/MP can be in one of the two states: working (State = 1) or failed (State = 0). Thus, an MC/MP is acting as a binary variable that can provide a maximum of $\log_2(2)$ =1 bit of information towards failure localization. In order to identify $|\Psi+1|$ SRLG failures, an FD set needs to represent $\log_2(|\Psi+1|)$ bits of information. In an optimal FL set, each MC/MP would contribute 1 bit of information, i.e., an optimal FL set must have $\lceil\log_2(|\Psi|+1)\rceil$ cycles. Accordingly, $T(\Psi) \geq \lceil\log_2(|\Psi|+1)\rceil$. For the example network in Figure 1, where $|\Psi| = 8$, the set of MCs with node 1 as the monitoring location and the set of MCs and MPs with nodes 1 and 4 as the monitoring locations were given in part (c) of the Figure 1. The presented solutions achieve the lower bound ($\lceil\log_2(8+1)\rceil$ = 4).

The sufficiency proof of Theorem 2 provides a method for obtaining a feasible solution by constructing cycles of type $\mathcal{T}_1$ ($\mathcal{T}_2$) and $\mathcal{T}_3$ for all possible SRLG pairs. This results in a loose upper bound of $|\Psi|(|\Psi|-1)$ on the required number of cycles. We now show that $\mathcal{O}(|\Psi|)$ is an upper bound on the number of cycles.

Theorem 5: The maximum number of monitoring cycles and paths required to monitor $|\Psi|$ SRLG failures is upper-bounded by $|\Psi|$, i.e., $T(\Psi) \leq |\Psi|$.

Proof: We prove the above theorem by induction. We first assume that we are given that an SRLG failure has occurred in the network. Given the SRLG failure has occurred, we show that we require at most $|\Psi|-1$ cycles. We then consider the case when the SRLG failure occurrence is not given, which results in the addition of at most one extra cycle, to complete the proof.

For the base case of this induction, given that a SRLG failure has occurred, and if $|\Psi| = 1$, then we do not need any cycles to localize the failure. Thus, T(1) = 0. Similarly, we consider the case of two SRLGs, i.e., $|\Psi| = 2$. Select an MC/MP that is affected by only one of the two SRLGs. Thus $T(|\Psi|) = 1$.

Suppose the theorem is valid for an SRLG set Ψ', where $|\Psi'| \leq |\Psi|-1$, i.e., $T(\Psi') \leq |\Psi'|-1$. Note that we are not counting the case of identifying no failure, which we will address separately. Now consider another SRLG set Ψ'', such that, $|\Psi''| = |\Psi|+1$. Select a cycle (or a path) c_1 that covers at least one SRLG in Ψ'' and add it to the set of MCs and MPs. Let Ψ_1 denote the set of SRLGs in Ψ'' that share at least a common link with c_1. Similarly, Ψ_2 be the set of SRLGs that do not share any common link with c_1, specifically, $\Psi_2 = \Psi \setminus \Psi_1$). Note that $|\Psi_1| \leq |\Psi|+1$, $|\Psi_2| \leq |\Psi|+1$, and $|\Psi_1|+|\Psi_2| = |\Psi|+1$. Furthermore, c_1 can distinguish failures between Ψ_1 and Ψ_2. c_1 also breaks the monitoring problem into two smaller sub-problems with SRLG sets Ψ_1 and Ψ_2. Using the above arguments, we have the following recurrence relation:

$$\begin{aligned} T(\Psi'') &\leq T(\Psi_1) + T(\Psi_2) + 1 \leq |\Psi_1| - 1 + |\Psi_2| - 1 + 1 \\ &\leq |\Psi| \leq |\Psi''| - 1 \end{aligned}$$

We have shown that we need at most $|\Psi|-1$ MPs/MCs to distinguish $|\Psi|$ SRLG failures, given that an SRLG failure has occurred. Observe that, among these SRLGs, at most one (say ψ_1) will not be present in any of the cycles. Thus, failure of ψ_1 will not result in the failure of any of the cycles. Any other SRLG failure will result in the failure of at least one cycle. In order to localize the failure without the knowledge that a failure has occurred, we need another cycle that is affected by ψ_1. With the addition of this cycle, any SRLG failure will result in the failure of at least one cycle. Thus, we need a total of at most $|\Psi|$ MPs and MCs to distinguish $|\Psi|$ SRLGs. ■

B. Construction of the Optimal FL Set

The monitoring location may contain dedicated hardware for each MC or it may have various MCs time-share the same hardware. In the first scenario, each cycle can be assigned its own wavelength, and cycles are monitored all the time. It is then reasonable to aim at constructing MCs that minimize the total bandwidth resources, i.e., minimize the sum of the hop lengths of the MCs. If the MCs time-share the monitoring equipment, then one wavelength, say λ_0, may be reserved on each link for monitoring purposes, and only one cycle may be monitored at a time. In this case, to maximize the frequency of monitoring, one should minimize the number MCs. In this paper, we consider the former scenario. We first formulate the problem of constructing a feasible solution that minimizes the sum of the cycle lengths as an ILP. We then develop heuristic solutions to it.

1) ILP Formulation: Consider a network graph $\mathcal{G}(\mathcal{N}, \mathcal{L})$, a given monitoring location $m \in \mathcal{N}$, and a set of SRLG failures Ψ. Let $\mathcal{C}_m$ be the set of all cycles that pass through m. The objective of the ILP is to find an FL set such that the average number of cycles per link is minimized. The ILP formulation takes $\mathcal{C}_m$ as input. Let α_c be a binary variable that specifies if cycle $c \in \mathcal{C}_m$ is part of the solution or not. Let L_c be the hop-length of cycle c. For an SRLG $\psi_i \in \Psi$, let $A_{\psi_i c}$ be a binary indicator that denotes whether at least one link in ψ_i is present in cycle c or not; $A_{\psi_i c} = 1$ if $\psi_i \cap c \neq \phi$, and 0 otherwise. Since $\mathcal{C}_m$ is an input to the ILP, $A_{\psi_i c}$, is known in advance $\forall \psi_i \in \Psi$ and $\forall c \in \mathcal{C}_m$. The ILP formulation is shown in Figure 9. The objective of this ILP is to minimize the sum of the hop-lengths of the selected MCs. Constraint C1 ensures that for any two SRLGs ψ_i and ψ_j, there exists at least one cycle in the FL set that overlaps with exactly one of the two SRLGs. Hence, the set of cycles that pass through an SRLG is different from those that pass through all other SRLGs by at least one cycle. Since this is true for all SRLG pairs, any SRLG failure can be uniquely identified provided there is at least one cycle passing through the failed SRLG. Constraint C2 ensures that every SRLG affects at least one cycle.

Minimize $\sum_{c\in\mathcal{C}_m} L_c \alpha_c$

Subject to:

C1: $\sum_{c\in\mathcal{C}_m} \{\alpha_c[A_{\psi_i c}(1 - A_{\psi_j c}) + (1 - A_{\psi_i c})A_{\psi_j c}]\} > 0, \forall \psi_i, \psi_j \in \Psi,\ \psi_i \neq \psi_j$

C2: $\sum_{c\in C_m} \alpha_c A_{\psi_i c} > 0, \quad \forall \psi_i \in \Psi$

Fig. 9. ILP formulation for finding the optimal FL set.

2) Heuristic: The ILP solution guarantees finding the optimal FL set but its worst-case complexity is exponential in the number of variables. It also requires enumerating an exponentially large number of cycles that pass through the

SRLG	ψ_1	ψ_2	(1,2)	(2,3)	(3,4)	(4,1)	(2,4)	(1,3)
t_ψ	7	11	5	3	9	15	6	10

TABLE I
TAG VALUES FOR VARIOUS SRLGS OF THE NETWORK IN FIGURE 1.

monitoring node. This complexity makes the ILP solution impractical for large networks. We now develop a heuristic for finding a feasible FL set. According to the heuristic, cycles are added incrementally to the solution set. When a cycle c is added, it is assigned a tag value. The tag for the ith added cycle is 2^{i-1}. As an example, consider the feasible FL set $\{c_1, c_2, c_3, c_4\}$ in Figure 1. The tag values assigned to the four cycles are 1, 2, 4, and 8, respectively. For each SRLG $\psi_i \in \Psi$, we maintain a tag t_{ψ_i} which denotes the sum of the tags of cycles selected thus far and that overlap with SRLG ψ_i. The tag values of various SRLGs of the example network in Figure 1 are shown in Table I. At any step in the algorithm, if two SRLGs have the same tag value, then they are covered by the same set of cycles and hence their failures cannot be uniquely identified. The algorithm ensures that no two SRLGs have the same tag value.

The input to the algorithm is a network graph $\mathcal{G}(\mathcal{N}, \mathcal{L})$, a monitoring location m, and a set of SRLG failures Ψ. The tag value for each SRLG t_{ψ_i} is initialized to zero. The algorithm starts by randomly selecting two SRLGs ψ_i and ψ_j with the same tag value. It then adds to the solution set $\mathcal{C}$ a least-weight cycle that traverses the monitoring node m and one of the links $\ell \in \psi_i \setminus \psi_j$ but does not overlap with ψ_j. The selected cycle is assigned a tag value. The SRLG tags are then updated. The above procedure is repeated until all SRLGs in the network have unique tag values. A pseudocode for the algorithm is shown in Figure 10.

Procedure `MC-1`
Input: $\mathcal{G}(\mathcal{N}, \mathcal{L}),\ m,\ \Psi$
Output: $\mathcal{C}$

1) $\forall \psi \in \Psi,\ t_\psi = 0, U = 0,\ \mathcal{C} = \phi,\ w_\ell = 1, \forall \ell \in \mathcal{L}$
2) While $\exists \{\psi_i, \psi_j\} \in \Psi$ s.t. $t_{\psi_i} = t_{\psi_j}$,
   a) $c =$ `FindCycle` $(m, \psi_i, \psi_j, w(.), \mathcal{G})$.
   b) $\forall \psi_k : \psi_k \cap c \neq \phi$,
      $w_{\psi_k} = w_{\psi_k} + |\mathcal{N}|,\ t_{\psi_k} = t_{\psi_k} + 2^U$
   c) $\mathcal{C} = \mathcal{C} \cup \{c\}$, $U = U + 1$
3) If $\exists \psi_x : t_{\psi_x} = 0$,
   $c =$ `FindCycle` $(m, \psi_x, \phi, w(.), \mathcal{G})$
   $\mathcal{C} = \mathcal{C} \cup \{c\}$, $U = U + 1$

Procedure `FindCycle`
Input: $m, \psi_x, \psi_y, w(.), \mathcal{L}$
Output: c
Choose a link ℓ randomly from the set $\psi_x \setminus \psi_y$. Find a cycle passing through m and ℓ by removing ψ_y and using the procedure mentioned in Figure 2. Two link-disjoint paths are constructed using minimum sum of two paths and with link weights as $w(.)$.

Fig. 10. Pseudocode for the `MC`-1 algorithm with one monitoring location.

At any stage in the algorithm, a cycle traversing the monitoring location m and link $\ell \in \psi_i \setminus \psi_j$ but not any of the links in ψ_j can be computed by removing all the links in ψ_j and then using the procedure presented in Figure 2. For a given pair of nodes, link-disjoint paths with the least sum of costs can be obtained using a modified version of Dijkstra's algorithm [4], [1], [5]. Notice that after finding an MC, the link weights associated with this cycle are incremented by $|\mathcal{N}|$. Such an adjustment along with the fact that we construct cycles by using two link-disjoint paths of minimum sum helps in evenly distributing the cycles among all the links, and hence it limits the number of cycles that pass through a particular link. Finally, if $\exists \psi_i : t_{\psi_i} = 0$, then an MC containing SRLG ψ_i is added to $\mathcal{C}$.

Complexity: A cycle traversing a node m and a link ℓ is obtained in $\mathcal{O}(|\mathcal{N}| \log |\mathcal{N}|)$ time by running two instances of Dijkstra's algorithm [1]. The worst-case number of cycles required to distinguish the failure of two links is $O(|\Psi|^2)$ (one cycle for every link pair). Hence, the worst-case complexity of our algorithm is $O(|\Psi|^2 |\mathcal{N}| \log |\mathcal{N}|)$.

VI. MONITORING WITH PRE-EXISTING PATHS

In this section, consider an optical network that has no dedicated bandwidth for monitoring. This situation may arise, for example, when the network is heavily loaded or when the network operator is not willing to dedicate bandwidth for monitoring purposes. In such networks, monitoring can only be done by probing various lightpaths that carry data traffic. Every lightpath carries traffic according to a certain framing format (e.g., SONET or GigE frames) and has the capability of detecting a loss of signal (for example, SONET end systems will generate LOS alarm if the lightpath experiences a fiber cut on any link it passes through). This loss of signal can be used to detect if the monitoring path has failed or not. Note that various framing techniques will also provide link quality information (such as BIP alarms in SONET), which can be used to localize SRLGs with high bit error rates.

We now consider the problem of monitoring an optical network that has a set of τ established lightpaths $\mathcal{H} = \{p_1, p_2, ..., p_\tau\}$. These lightpaths carry traffic between given sources and destinations. To monitor SRLG failures in such a network, we can use passive optical splitters, placed at various intermediate nodes. For example, consider a lightpath p in Figure 11, which originates at node x and terminates at node z, passing through node y. The failure of link ℓ_{xy} or ℓ_{yz} will result in the failure of p. However to distinguish the failures of links ℓ_{xy} and ℓ_{yz}, we need to add an optical splitter at node y. From a monitoring perspective, the inclusion of this optical splitter is equivalent to adding a new lightpath p' to the monitoring system, which provides an additional syndrome to link ℓ_{xy}. After each splitter, a filter is used to tap a specific wavelength carrying lightpath. We examine a specific wavelength and assume that a failure will affect all wavelengths.

Note that each path $p_i \in \mathcal{H}$ is monitored by its end nodes, i.e., p_i is part of the syndrome of all the SRLGs that have common links with p_i. The syndrome of an SRLG ψ_j is then

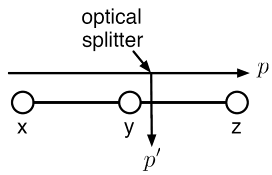


Fig. 11. Example ligthpath to demonstrate path splitting.

given by $\{p_i : p_i \cap \psi_j \neq \phi,\ p_i \in \mathcal{H}\}$. Note that the set of paths $\mathcal{H}$ may provide unique syndrome for some but not all SRLGs. For example, in Figure 12, we have three SRLGs: $\psi_1 = \{\ell_{12}, \ell_{23}\}$, $\psi_2 = \{\ell_{23}, \ell_{24}\}$, and $\psi_3 = \{\ell_{34}\}$, and two lightpaths: $p_1 = \{\ell_{12}, \ell_{24}\}$ and $p_2 = \{\ell_{12}, \ell_{23}, \ell_{34}\}$. While

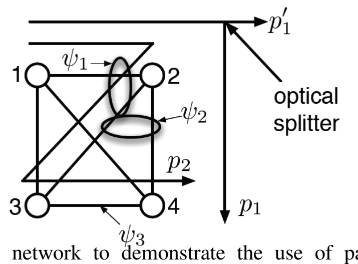


Fig. 12. Example network to demonstrate the use of path splitting for distinguishing SRLG failures.

ψ_3 can be uniquely identified using p_2, p_1 and p_2 cannot distinguish the failures of ψ_1 and ψ_2. An optical splitter at node 2 provides an additional monitoring path ($p_1' = \{\ell_{12}\}$) that can be used to distinguish the failures of ψ_1 and ψ_2.

Given the cost of adding and maintaining optical splitters, we would like to minimize the number of optical splitters placed at various nodes in the network. We now define the monitoring problem for a network with a set of established paths and with no dedicated monitoring bandwidth.

Problem MNDB (Monitoring With No Dedicated Bandwidth): Given a network graph $\mathcal{G}(\mathcal{N}, \mathcal{L})$, a set $\mathcal{H}$ of τ paths, and a set of SRLGs Ψ, find the minimum number of splitters and their placement such that the failure of an SRLG can be uniquely localized.

Note that SRLGs that have unique syndromes based solely on paths in $\mathcal{H}$ will not require any additional optical splitters. Hence, the problem is reduced to finding the locations of optical splitters for SRLGs that do not have a unique syndromes. Let $\Psi' \subset \Psi$ be the set of such SRLGs. Assume $\Psi' \neq \phi$. We now present an ILP formulation for finding the minimum number of optical splitters and their positions.

A. ILP formulation for MNDB problem

Consider the MNDB problem with SRLGs Ψ'. For a path $p_i \in \mathcal{H}$, let $\mathcal{H}_{p_i} \stackrel{\text{def}}{=} \{p_i^{(1)}, p_i^{(2)}, ..., p_i^{(r)}\}$ be the set of all possible paths resulting from the placement of optical splitters along path p_i. Let $\mathcal{H}' = \cup_{i=1,2,...,\tau} \mathcal{H}_{p_i}$. The ILP takes $\mathcal{H}'$ as input. Let I_p be a binary variable that specifies if path $p \in \mathcal{H}'$ is part of the solution set or not. For an SRLG $\psi_i \in \Psi'$, let $A_{\psi_i,p}$ be the binary indicator that denotes whether at least one link in ψ_i is present in path p or not; $A_{\psi_i,p} = 1$ if $\psi_i \cap p \neq \phi$, and 0 otherwise. The ILP formulation is shown in Figure 13. The objective is to minimize the number of paths selected from $\mathcal{H}'$. Each path in $\mathcal{H}'$ is associated with an optical splitter, hence

Minimize $\sum_{p \in \mathcal{H}'} I_p$

Subject to:

C1: $\sum_{p \in \mathcal{H}'} \{I_p[A_{\psi_i,p}(1 - A_{\psi_j,p}) + (1 - A_{\psi_i,p})A_{\psi_j,p}]\} > 0, \forall \psi_i, \psi_j \in \Psi',\ \psi_i \neq \psi_j$

Fig. 13. ILP formulation for finding the optimal splitter placement.

this is equivalent to minimizing the number of optical splitters. Constraint C1 ensures that for any two SRLGs ψ_i and ψ_j, there exists at least one path $p \in \mathcal{H}'$ among the chosen set of paths on which exactly one SRLG is present. We do not need to impose a constraint to ensure that each SRLG is at least part of a path $p \in \mathcal{H}'$ because if an SRLG is not part of any path then it is not carrying any traffic. Such an SRLG cannot be monitored without introducing an additional lightpath.

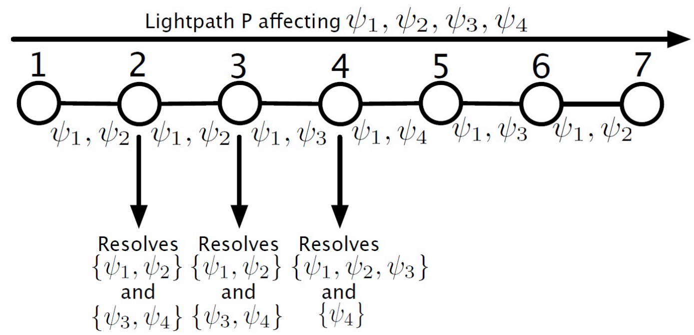


Fig. 14. Example of a lightpath with multiple optical splitters for distinguishing SRLG failures.

We now describe how optical splitter placement allows for distinguishing failures of a group of SRLGs with common syndrome. Consider the example in Figure 14, where a lightpath p affects four SRLGs $\{\psi_1, \psi_2, \psi_3, \psi_4\}$. Assume that $\{\psi_1, \psi_2, \psi_3, \psi_4\}$ have a common syndrome and path p is part of their syndrome. Note that an optical splitter at nodes 2 or 3 can distinguish failures of the $\{\psi_1, \psi_2\}$ and $\{\psi_3, \psi_4\}$. However, such an optical splitter cannot distinguish failures of ψ_1 and ψ_2, or between ψ_3 and ψ_4. Similarly, an optical splitter at node 4 can distinguish between $\{\psi_1, \psi_2, \psi_3\}$ and $\{\psi_4\}$ but not $\{\psi_1, \psi_2\}$, and ψ_3. Note that optical splitters at nodes 2 and 4 or at nodes 3 and 4 can resolve failures between $\{\psi_1, \psi_2\}$, $\{\psi_3\}$, and $\{\psi_4\}$. However, no optical splitter placement on path p can distinguish failures of ψ_1 and ψ_2.

We now present a general condition for the placement of optical splitters, which can help in resolving a set of SRLG failures. Consider a path p from node s to node t and consider a set of SRLGs $\mathcal{S}_p \equiv \{\psi_1, \psi_2, ..., \psi_r\}$ affected by p and that have a common syndrome. Let p_x be a subpath of p that originates at s and terminates at an intermediate node x on p, and let $\mathcal{S}_{p_x}$ be the set of SRLGs affected by p_x. Note that an optical splitter at node x will resolve failures between SRLGs $\mathcal{S}_{p_x}$ and $\mathcal{S}_p \setminus \mathcal{S}_{p_x}$. Hence, if $\mathcal{S}_p \setminus \mathcal{S}_{p_x} = \phi$, then an optical splitter at node x will not help in resolving any SRLG failures. Based on the above argument, we have the following condition for optical splitter placement:

For a ligthpath $p \in \mathcal{H}$ and an intermediate node x on p, an optical splitter must be placed at x if $\mathcal{S}_p \setminus \mathcal{S}_{p_x} \neq \phi$.

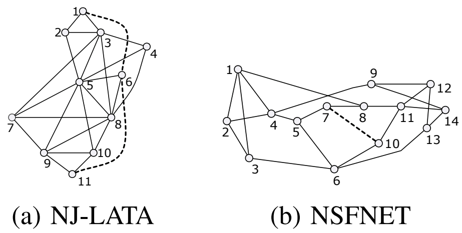


Fig. 15. Network topologies used in the simulations.

Note that we can significantly reduce the size of the set $\mathcal{H}'$ by choosing only those subpaths that satisfy the above condition.

VII. Simulation Results

We compare the performance of the MC-1 algorithm with the optimal results obtained using the ILP solution. We consider the two topologies (NJ-LATA and NSFNET) shown in Figure 15. These small topologies are selected because they enable us to obtain the ILP solution using the CPLEX solver [6] in tractable time. For both topologies, we add a few additional links to make them three-edge-connected, and we use MC-1 to find cycles that can localize any single-link failure (i.e., $\Psi = \mathcal{L}$). For a given topology, we vary the location of the monitoring node and report the average number of monitoring-related wavelengths per link, given by $\frac{\sum_{c\in\mathcal{C}} L_c}{|\mathcal{L}|}$, where L_c is the number of links in a cycle c and $\mathcal{C}$ is the set of MCs. This measure reflects the amount of network resources per link that are used for link-failure localization. We also report the number of cycles associated with the monitoring location, which reflects the setup cost associated with monitoring. Tables II depict the observed performance. For both topologies, approximately two wavelengths per link are found sufficient to localize single-link failures. This confirms the effectiveness of our fault localization approach. The results obtained from the $\mathtt{MC}-1$ algorithm are very close to the optimal ILP solution. For these experiments, we found that the running time of MC-1 is in the order of a few seconds ($\leq$ 60 secs). For the ILP, the running time is in the order of days. In general, we observed multiple orders of difference in the running times of ILP and MC-1 for small topologies (NJ-LATA, NSFNET) considered. These simulations were done on a dual-core, 2GHz-processor PC, with 2GB of RAM. For large topologies, it was infeasible to obtain the ILP results for large topologies.

Next, we consider mesh-torus topologies with 16 (4×4), 25 (5×5), 36 (6×6), 49 (7×7), and 64 (8×8) nodes. A mesh-torus is a four-edge-connected graph. We use these topologies with a single monitoring location to analyze the extra resources consumed when two links fail simultaneously as opposed to single-link and adjacent-link failures[7]. Figure 16 shows the average number of wavelengths per link that are used for monitoring purposes when single-link, two adjacent links, and arbitrary two-link failures take place. It is noted that compared with single-link failure monitoring, almost no additional resources are consumed in monitoring two-adjacent link failures. An additional 30% network resource is needed when the two failed links are arbitrarily located.

[7]Two links are adjacent if they have a common node.

Network	NJ-LATA: 11 nodes, 25 links				
Monitoring node	1	3	5	7	9
No. of MCs in FL set (ILP)	8	7	8	8	7
No. of MCs in FL set (MC-1)	13	13	16	13	12
No. of MCs / link (ILP)	0.32	0.28	0.32	0.32	0.28
No. of MCs / link (MC-1)	0.52	0.52	0.64	0.52	0.48
Avg. no. of **WLs**/link (ILP)	2.12	1.82	1.72	1.98	1.88
Avg. no. of **WLs**/link (MC-1)	3.16	2.42	2.38	2.56	2.7
Network	**NSFNET: 14 nodes, 23 links**				
Monitoring node	1	3	5	7	9
No. of MCs in FL set (ILP)	6	5	6	6	6
No. of MCs in FL set (MC-1)	11	11	12	10	10
No. of MCs / link (ILP)	0.26	0.22	0.26	0.26	0.26
No. of MCs / link (MC-1)	0.48	0.48	0.52	0.43	0.43
Avg. number of WLs/link (ILP)	2.06	2.13	2.17	2.11	2.04
Avg. number of WLs/link (MC-1)	3.04	2.65	2.73	2.91	3.00

TABLE II
SIMULATION RESULTS FOR THE MC-1 PROBLEM.

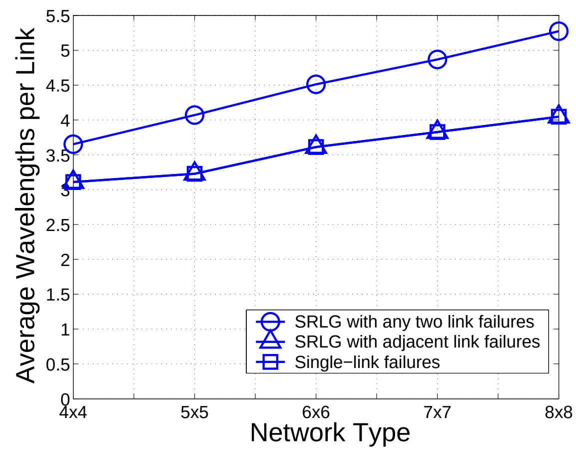


Fig. 16. Average number of wavelengths consumed for different mesh torus topologies.

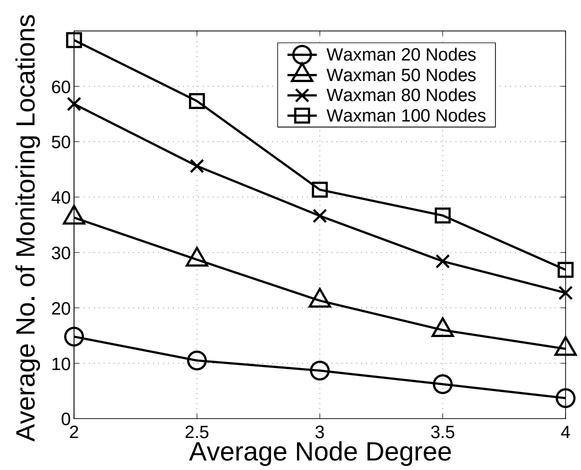


Fig. 17. Average number of monitors vs. average node degree.

A. Monitoring an Arbitrary Network Topology

In this part, we use randomly generated Waxman's topologies [15] of different average node degree. For a given topology, we determine the required number of monitoring locations that can detect all single-link failures, i.e. $k = 1$, by using the `Placement` procedure presented in Section IV. For any j-

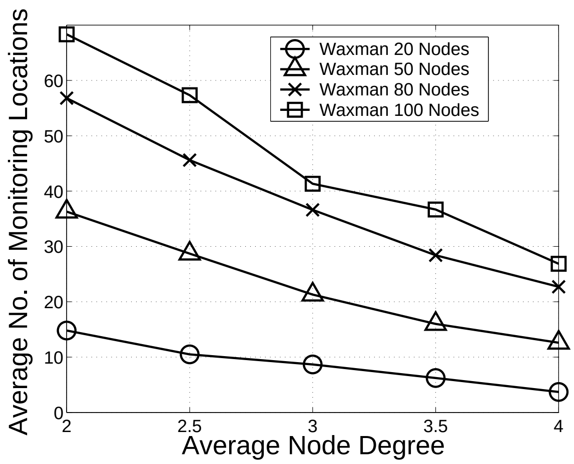


Fig. 18. Average number of wavelengths vs. average node degree.

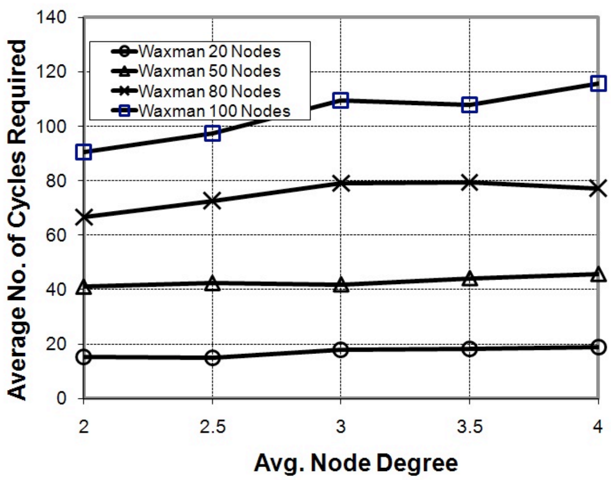


Fig. 19. Total number of wavelengths vs. average node degree.

edge-connected component, $j = 2, 3$, that requires a monitor, an internal node is chosen randomly to be the monitoring location. For the chosen set of monitoring locations, we use the graph transformation presented in Section IV and then apply the $\texttt{MC} - \texttt{1}$ algorithm to find the set of paths and cycles needed to monitor all single-link failures in the network. For each topology, we report the number of required monitoring locations and the average number of cycles per link. Figure 17 shows that the average number of required monitoring locations decreases with the average node degree. Intuitively, this is true because the average size of a three-edge-connected component increases with the increase in average node degree. Figure 18 shows the required average wavelengths per link for detecting all single-link failures. Figure 19 shows the total number of cycles required for detecting all single-link failures. We make the following observations from 17, 18, and 19:
(1) For sparse networks (average node degree ≤ 2), average number of MCs per link is small because for such network, number of required monitoring locations is large. In this case, many links get monitored by their adjacent nodes. (2) Average number of required cycles per link increases with the average node degree. (3) For dense topologies, less than 30% of nodes require a monitor, which is a clear improvement over the technique presented in [9], in which all nodes need to have monitor to detect single-link failures. (4) For small networks (≤ 80 nodes), less than three wavelengths per link are used, on average, for monitoring purposes. This shows the effectiveness and low overhead associated with the presented monitoring technique. (5) Increase in node degree has a little impact on the number of cycles required.

VIII. Conclusions

In this paper, we considered the problem of fault localization in all-optical networks. We described a fault localization mechanism that uniquely determines SRLG failures by using monitoring cycles and paths. We provided necessary and sufficient conditions on (1) the requirements of the fault localization set; (2) network connectivity for localizing failures with one monitoring location; and (3) the placement of monitoring locations to obtain a feasible solution. We developed an $\mathcal{O}(k|\mathcal{N}|^4)$ algorithm to calculate the minimum number of required monitoring locations to localize all possible failures involving up to k links. We described an ILP formulation and a heuristic approach (`MC-1`) to find the set of cycles that can localize SRLG failures using a single monitoring location. We also considered the problem of monitoring an optical network with no dedicated bandwidth for monitoring purposes. We employed optical splitters at various nodes to probe various lightpaths carrying traffic. We described an ILP formulation to identify the minimum number and locations of optical splitters needed to monitor all SRLGs in a network. Simulation results confirm the effectiveness of the proposed monitoring technique and the presented solutions.

References

[1] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network flows: Theory, Algorithm, and Applications*. Prentice Hall Inc., 1993.
[2] S. Ahuja, S. Ramasubramanian, and M. Krunz. Single link failure detection in all-optical networks using monitoring cycles and paths. *IEEE/ACM Transactions on Networking*, 17(4):1080–1093, Aug. 2009.
[3] 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. *Proceedings of the IEEE Globecom Conference*, 3:2676–2680, Nov. 2002.
[4] R. Bhandari. *Survivable Networks: Algorithms for Diverse Routing*. Kluwer Academic Publishers, 1999.
[5] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*. Prentice Hall Inc., 1998.
[6] CPLEX. http://www.cplex.com.
[7] Z. Galil and G. F. Italiano. Maintaining the 3-edge-connected components of a graph on-line. *SIAM Journal on Computing*, 22(1):11–28, 1993.
[8] M. Goyal, K. K. Ramakrishnan, and W. Feng. Achieving faster failure detection in OSPF networks. *Proceedings of the IEEE ICC Conference*, 1:296–300, May 2003.
[9] Y. Hamazumi, M. Koga, K. Kawai, H. Ichino, and K. Sato. Optical path fault management in layered networks. *Proceedings of the IEEE Globecom Conference*, 4:2309–2314, Nov. 1998.
[10] N. Harvey, M. Patrascu, Y. Wen, S. Yekhanin, and V. W. S. Chan. Non-adaptive fault diagnosis for All-optical networks via combinatorial group testing on graphs. *Proceedings of the IEEE INFOCOM Conference*, May 2007.
[11] J. Hopcroft and R. E. Tarjan. Dividing a graph into triconnected components. *SIAM Journal of Computing*, 2(3):135–158, 1973.
[12] S. Maclane. A structural characterization of planar combinatorial graphs. *DUKE Math Journal*, 3(3):460–472, 1937.

[13] J. Tapolcai, B. Wu, and P. H. Ho. On Monitoring and Failure Localization in Mesh All-Optical Networks. *Proceedings of the IEEE INFOCOM Conference*, Apr. 2009.

[14] A. Todimala and B. Ramamurthy. Survivable virtual topology routing under shared risk link groups in WDM networks. *First International Conference on Broadband Networks (BROADNETS)*, pages 130–139, Oct. 2004.

[15] B. M. Waxman. Routing of multipoint connections. *IEEE Journal on Selected Areas in Communications*, 69:1617–1622, Dec. 1988.

[16] Y. Wen, V. W. S. Chan, and L. Zheng. Efficient fault diagnosis algorithms for All-optical WDM networks with probabilistic link failures. *IEEE/OSA Journal of Lightwave Technology*, 23(10):3358–3371, Oct. 2005.

[17] B. Wu, P. H. Ho, and K. L. Yeung. Monitoring Trail: On Fast Link Failure Localization in All-Optical WDM Mesh Networks. *IEEE/OSA Journal of Lightwave Technology*, 27(18):4175–4185, 2009.

Satyajeet S. Ahuja (S'04 / M'08) received B.E. (Hons.) degree in Electronics and Communication Engineering from Maulana Azad College of Technology (MACT), Bhopal, India, in 1999. He received his M.E. (Hons.) degree in Electrical and Communication Engineering (Telecommunications) from Indian Institute of Science (IISc), Bangalore, India, in 2002 and Ph.D. degree in ECE from the University of Arizona, Tucson, USA in 2008. He held research positions at Google; Tejas Networks, Bangalore, India; and VSNL, Bangalore, India. He is currently working at Infinera Corporation, Sunnyvale, CA, USA. His research interests include design and analysis of algorithms for optical and wireless networks, network monitoring, fault localization, network design, routing, media streaming, and protocol design.

Srinivasan Ramasubramanian (S'99 / M'02 / SM'08) received the B.E. (Hons.) degree in Electrical and Electronics Engineering from Birla Institute of Technology and Science (BITS), Pilani, India, in 1997, and the Ph.D. degree in Computer Engineering from Iowa State University, Ames, in 2002. Since August 2002, he is an Assistant Professor in the Department of Electrical and Computer Engineering at The University of Arizona. He is a co-developer of the Hierarchical Modeling and Analysis Package (HIMAP), a reliability modeling and analysis tool, which is currently being used at Boeing, Honeywell, and several other companies and universities. His research interests include architectures and algorithms for optical and wireless networks, multipath routing, fault tolerance, system modeling, and performance analysis. He has served as the TPC Co-Chair of BROADNETS 2005, ICCCN 2008, and ICC 2010 conferences and LANMAN 2010 Workshop. He has served on the editorial board of the Springer Wireless Networks Journal from 2005 to 2009.

Marwan Krunz is a professor of ECE at the University of Arizona. He also holds a joint appointment at the same rank in the Department of Computer Science. He directs the wireless and networking group and is also the UA site director for Connection One, a joint NSF/state/industry IUCRC cooperative center that focuses on RF and wireless communication systems and networks. Dr. Krunz received his Ph.D. degree in electrical engineering from Michigan State University in 1995. He joined the University of Arizona in January 1997, after a brief postdoctoral stint at the University of Maryland, College Park. In 2010, he was a visiting researcher at the University of Carlos III, Madrid, Spain, where he held a Chair of Excellence ("Catedra de Excelencia") position. He previously held other visiting research positions at INRIA, HP Labs, University of Paris VI, and US West (now Qwest) Advanced Technologies. Dr. Krunz's research interests lie in the fields of computer networking and wireless communications, with recent focus on cognitive radios and SDRs; distributed radio resource management in wireless networks; channel access and protocol design; MIMO and smart-antenna systems; UWB-based personal area networks; energy management and clustering in sensor networks; media streaming; QoS routing; and fault monitoring/detection in optical networks. He has published more than 160 journal articles and refereed conference papers, and is a co-inventor on three US patents. M. Krunz is a recipient of the National Science Foundation CAREER Award (1998). He currently serves on the editorial boards for the IEEE Transactions on Mobile Computing, the IEEE Transactions on Network and Service Management, and the Computer Communications Journal. He previously served on the editorial board for the IEEE/ACM Transactions on Networking (2001-2008). He was a guest co-editor for special issues in IEEE Micro and IEEE Communications magazines. He served as a technical program chair for various international conferences, including WoWMoM 2006, SECON 2005, INFOCOM 2004, and Hot Interconnects 2001 Symposium. He has served and continues to serve on the executive and technical program committees of many international conferences and on the panels of several NSF directorates. He gave keynotes and tutorials, and participated in various panels at premier wireless networking conferences. He is a consultant for a number of companies in the telecommunications sector. He is an IEEE Fellow.