Neighborhood Failure Localization in All-Optical Networks via Monitoring Trails

János Tapolcai* *Member, IEEE,* Pin-Han Ho† *Member, IEEE,* Péter Babarczi* *Member, IEEE,* Lajos Rónyai‡
*MTA-BME Future Internet Research Group, Budapest University of Technology and Economics (BME), Hungary, {tapolcai, babarczi}@tmit.bme.hu
†Dept. of Electrical and Computer Engineering, University of Waterloo, Canada, p4ho@uwaterloo.ca
‡Computer and Automation Research Institute Hungarian Academy of Sciences and BME, ronyai@sztaki.hu

Abstract*—Shared protection, such as failure dependent protection (FDP), is well recognized for its outstanding capacity efficiency in all-optical mesh networks, at the expense of lengthy restoration time due to multi-hop signaling mechanisms for failure localization, notification, and device configuration. This paper investigates a novel monitoring trail (m-trail) scenario, called Global Neighborhood Failure Localization (G-NFL), that aims to enable any shared protection scheme, including FDP, for achieving all-optical and ultra-fast failure restoration. We firstly define *neighborhood* of a node, which is a set of links whose failure states should be known to the node in restoration of the corresponding working lightpaths (W-LPs). By assuming every node can obtain the on-off status of traversing m-trails and W-LPs via lambda monitoring, the proposed G-NFL problem routes a set of m-trails such that each node can localize any failure in its neighborhood. Bound analysis is performed on the minimum bandwidth required for m-trails under the proposed G-NFL problem. Then a simple yet efficient heuristic approach is presented. Extensive simulation is conducted to verify the proposed G-NFL scenario under a number of different definitions of nodal neighborhood which concern the extent of dependency between the monitoring plane and data plane. The effect of reusing the spare capacity by FDP for supporting m-trails is examined. We conclude that the proposed G-NFL scenario enables a general shared protection scheme, toward signaling-free and ultra-fast failure restoration like p-Cycle, while achieving optimal capacity efficiency as FDP.

***Index Terms*—neighborhood, monitoring trails, failure localization, all-optical networks**

I. Introduction

It is considered the best strategy to locally restore an optical layer failure (e.g., fiber cut) in the optical domain within as short time as possible before the failure maliciously affects the operation of upper layer protocols such as IP or TCP. Thus, an optical layer failure should be handled without relying on any electronic signaling protocol no matter the network optical domain has central or distributed control. Currently, only dedicated protection (i.e., $1+1$) and pre-configured Cycle (p-Cycle) based approaches can achieve 50 ms or shorter restoration time in mesh networks due to their simplicity and pre-configured spare capacity, but at the expense of 70% or higher redundancy [1]. Note that failures are rare events, and allocating a significant amount of redundancy for failure recovery is not considered economically reasonable.

On the other hand, failure dependent protection (FDP), as a general version of shared protection, trades control simplicity for better capacity efficiency, in which one or multiple *protection lightpaths* (P-LPs) are pre-computed for each *working lightpath* (W-LP). The switching capacity of the optical cross-connects (OXCs) and the wavelength links (WLs) along the P-LPs – although reserved – are shared among multiple P-LPs and will be configured only after the failure occurs. Thus, improved capacity efficiency up to 30% of redundancy can be achieved [1] at the expense of extensive signaling mechanisms for real-time fault management and device configuration for P-LP setup, which possibly leads to hundreds of milliseconds of recovery time.

Monitoring trail (m-trail) has been proposed as an effective approach to enable all-optical and ultra-fast failure restoration in the network optical domain. An m-trail is implemented as a pair of lightpaths along a common physical route in opposite directions for sensing/monitoring the health of the links along the route. Thus, each node traversed by an m-trail will sense loss of light (LOL) via lambda monitoring when a failure hits upon any link along the m-trail. By properly allocating a set of m-trails in the network, an all-optical monitoring system is formed, so that every node can unambiguously identify the failed link by only inspecting the m-trails traversing through the node. This is also referred to as the network-wide local unambiguous failure localization (NWL-UFL) scenario [2]. In [3] NWL-UFL was taken as a building block for constructing the first all-optical failure restoration framework, which enables a general shared protection scheme to be performed in an all-optical and signaling-free fashion.

Although theoretically sound, [2], [3] assumed each node able to unambiguously identify all possible failures. Thus, a node will monitor a remote link even if the node does not need to respond to the link failure. This approach results unnecessary monitoring resource consumption, high computation complexity, and very lengthy m-trails. Note that using lengthy m-trails not only causes various implementation issues

This work was supported by the High Speed Networks Laboratory (HSNLab) at the Budapest University of Technology and Economics. P.-H. Ho was supported by National Science and Engineering Research Council (NSERC), Canada. P. Babarczi was supported by the János Bolyai Research Scholarship of the Hungarian Academy of Sciences (MTA). Research of P. Babarczi and J. Tapolcai was partially supported by the Hungarian Scientific Research Fund (OTKA grant K108947). L. Rónyai was supported by the Hungarian Research Fund (OTKA grant NK105645). This document has been produced with the financial assistance of the European Union under the FP7 GÉANT project grant agreement number 605243 as part of the MINERVA Open Call project.

(because of physical layer impairments [4]) but also increases monitoring latencies.

In this paper, we investigate an on-demand m-trail allocation paradigm that enables a general shared protection scheme to perform signaling-free failure restoration as in $1+1$ and p-Cycle. A novel scenario of m-trails, called Global Neighborhood Failure Localization (G-NFL), is proposed [5]. The features of our approach are summarized in four points:

(1) The *neighborhood* of a node is defined as a set of links whose failures must be unambiguously localized by the node. Furthermore, each node only localizes the link failures in its neighborhood.
(2) The spare capacity of the P-LPs can be reused to support the m-trails in order to achieve better capacity efficiency.
(3) A node can monitor both traversing m-trails and W-LPs for failure status acquisition, which is referred to as *out-of-band* and *in-band* monitoring, respectively.
(4) Since each m-trail is at most one hop longer than the corresponding shortest path, a provisioned lightpath can be taken as an m-trail, a W-LP or both according to the operation requirement.

The G-NFL problem aims to find a set of m-trails with minimum consumed WLs (called *coverlength*) such that all the nodes can localize the link failures in their respective neighborhoods. Lower bounds of the G-NFL problem are derived for general graphs by using combinatorial group testing (CGT) arguments. This is a newly defined CGT paradigm where the cost of testing a group is dependent on the size of the group, and the developed theorem and proof are considered the first theoretical work under the associated application scenario. This contributes to the state-of-the-art CGT theories.

The rest of the paper is organized as follows. Section II provides background for the study. Section III precisely describes the proposed G-NFL m-trail allocation problem and the definition of neighborhood. In Section IV, lower bounds on the coverlength of the proposed G-NFL problem are analyzed via CGT theory. Section V presents our heuristic algorithm for solving the G-NFL problem. Section VI presents the simulation results, and Section VII concludes the paper.

II. Background

A. Standard Failure Restoration

The failure restoration process under a general shared protection scheme mainly includes two post-failure tasks: one is *fault management* and the other is *device configuration* (for the P-LP setup).

Fault management defined in Generalized Multi-protocol Label Switching (GMPLS) [6] includes *failure localization* and *failure notification*, which are two sequentially performed real-time actions. In the former, the nodes adjacent to the failure detect, localize, and isolate the failure. In the latter, these nodes send notifications to the switching nodes of the affected W-LPs. Then the switching node correlates the notifications to exactly identify the failed link, before it can initiate the pre-planned restoration process. At last, device configuration is performed at each OXC along the P-LP(s) via a hop-by-hop wake-up process using an electronic control packet such that the affected W-LP(s) can be switched over.

It is clear that both fault management and device configuration under GMPLS should be supported by multi-hop signaling mechanisms, which possibly incurs high control complexity and long restoration time. This issue can be addressed with the application of m-trails.

B. Signaling-Free Failure Localization via M-Trails

Failure localization via a central fault management system using m-trails has been extensively studied in the past decade [7]–[14]. Later, Local Unambiguous Failure Localization (L-UFL) [2], [15]–[17] was introduced, which provides a signaling-free failure restoration framework that can be purely operated in the optical domain. With a set of m-trails properly allocated, a node is *L-UFL capable* if the node can unambiguously identify any link failure according to locally available m-trail on-off status.

In [15] monitoring location (ML) was defined as a node which terminates the launched monitoring lightpaths and coordinates the sensed alarms. The study focused on designating as few MLs as possible to collect the alarms in order to collaboratively identify the failed link(s). Note that, when only a single ML is required, the ML is L-UFL capable. [16] improved [15] by exploring the scenario where not only the terminating node but also the intermediate nodes of an m-trail can obtain its on-off status via lambda monitoring. The study attempted to enable L-UFL for a given set of nodes via an integer linear program, and discovered the fact that the total coverlength scales very well with the number of L-UFL capable nodes. This is due to the sharing of on-off status information among the nodes traversed by a common m-trail. Motivated by the preceding result, similar ideas were developed in [17] via a novel heuristic approach, and were further extended in [2] in which all the nodes are made L-UFL capable under any single link failure. An efficient heuristic was developed for allocating m-trails in a shape of spanning tree via link code swapping. With all the nodes being L-UFL capable, NWL-UFL is achieved.

In [3], NWL-UFL was taken to facilitate all-optical and signaling-free failure restoration. The basic idea in NWL-UFL is to leave the two post-failure tasks, namely fault management and device configuration, to be autonomously performed in the optical domain without any aid by a multi-hop signaling protocol. To achieve this, the switching, intermediate, and merging nodes of a P-LP can start configuring their OXCs to form the required cross-connect right after the identification of the failure. This is possible owing to NWL-UFL, which allows every node to unambiguously localize any link failure. Thus, the affected W-LP can be switched over to the P-LP without waiting for the cross-layer signaling mechanisms as in GMPLS. Although technically sound, [2], [3], [17] used very lengthy m-trails to localize all link failures at each node. This is not considered feasible in some cases, which is the last hurdle of the practical implementation of NWL-ULF. We tackle this problem in this paper with the application of state-of-the-art CGT theory.

C. Non-Adaptive Combinatorial Group Testing and Separating Systems

The primary goal of a CGT construction is to identify up to d defective items among a given set through as few tests as possible. In our case, the set of items are the network links, the defective items are the failed links, and the tests are by way of allocating a set of m-trails in the network [13]. For more references on CGT the interested reader can refer to [18]. For $d = 1$ (i.e., the case of single link failures), the problem is also called *separating systems* [19], which was introduced by Alfréd Rényi [20] in 1961 in the context of information theory.

Our novel neighborhood failure localization problem is a newly defined CGT problem where the cost of testing a group is dependent on the size of the group. To the best of our knowledge, in all past studies on CGT the aim was to minimize the number of tests, i.e., the cost of each test was constant. Note that none of the results on traditional CGT can be applied to this generalization of the problem.

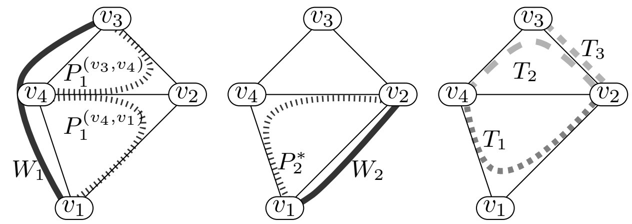
III. The G-NFL Scenario

Global Neighborhood Failure Localization (G-NFL) is proposed as a novel scenario of m-trails aiming to resolve all the potential issues in the previously reported studies. Given the W-LPs and P-LPs, the neighborhood of each node is defined, and a node is said to meet the *NFL requirement* if it can localize all the link failures in its neighborhood. Thus, a feasible G-NFL solution consists of a set of m-trails such that each node can localize the failed links in its neighborhood based on the on-off status of a subset of the m-trails and/or W-LPs that pass through the node.

Definition 1: The *neighborhood* of a node is defined as a set of links whose failures must be unambiguously localized by the node.

In particular, the neighborhood of a node should contain all the links along the W-LPs whose corresponding P-LPs traverse through the node. On the other hand, all the nodes traversed by a P-LP should be able to localize the link failure for which the P-LP is used to restore the disrupted W-LP. Therefore, the size of neighborhood of each node (and the resultant monitoring resource consumption) is expected to scale well with the network size.

Fig. 1 shows an example of G-NFL in a topology with 4 nodes and two W-LPs denoted as W_1 and W_2, each being provisioned with two physical lightpaths on the same route in opposite directions. W_1 is protected by two P-LPs, namely $P_1^{(v_3,v_4)}$ for link failure (v_3, v_4), and $P_1^{(v_4,v_1)}$ for (v_4, v_1) as shown in Fig. 1(a), while W_2 is protected by a single P-LP denoted as P_2^* as shown in Fig. 1(b). To ensure signaling-free restoration for W_1 and W_2, v_1 should be able to unambiguously identify the failure of (v_4, v_1) and (v_2, v_1), such that W_1 (or W_2) can be switched over to $P_1^{(v_4,v_1)}$ (or P_2^*) when (v_4, v_1) (or (v_2, v_1)) fails. Thus, the neighborhood of v_1 must contain the two links (v_4, v_1) and (v_2, v_1). On the other hand, since v_2 is traversed by all the three P-LPs (i.e., $P_1^{(v_3,v_4)}$, $P_1^{(v_4,v_1)}$, and P_2^*), it needs to react to any failure along W_1 or W_2. Therefore, the neighborhood of v_2 should contain (v_3, v_4), (v_4, v_1) and (v_2, v_1). Similarly, we can define the neighborhood of v_3 as (v_3, v_4), and that of v_4 as (v_2, v_1), (v_3, v_4), and (v_4, v_1).



(a) Connection 1 (b) Connection 2 (c) M-trails T_1,T_2,T_3

Link	T_1	T_2	W_1	switch to
(v_3, v_4)	0	1	1	$P_1^{(v_3,v_4)}$
(v_4, v_1)	1	0	1	$P_1^{(v_4,v_1)}$
(v_2, v_1)	1	0	0	P_2^*
(v_3, v_2)	0	1	0	
(v_2, v_4)	0	0	0	

(d) ACT at v_4

Link	T_2	T_3	W_1	switch to
(v_3, v_4)	1	0	1	$P_1^{(v_3,v_4)}$
(v_2, v_1)	0	0	0	
(v_3, v_2)	1	1	0	
(v_4, v_1)	0	0	1	
(v_2, v_4)	0	0	0	

(e) ACT at v_3

Link	T_1	W_1	W_2	switch to
(v_4, v_1)	1	1	0	$P_1^{(v_4,v_1)}$
(v_2, v_1)	1	0	1	P_2^*
(v_3, v_2)	0	0	0	
(v_3, v_4)	0	1	0	
(v_2, v_4)	0	0	0	

(f) ACT at v_1

Link	T_1	T_2	T_3	W_2	switch to
(v_3, v_4)	0	1	0	0	$P_1^{(v_3,v_4)}$
(v_4, v_1)	1	0	0	0	$P_1^{(v_4,v_1)}$
(v_2, v_1)	1	0	0	1	P_2^*
(v_3, v_2)	0	1	1	0	
(v_2, v_4)	0	0	0	0	

(g) ACT at v_2

Fig. 1. An illustrative example for the proposed G-NFL scenario.

To achieve the NFL requirement according to the above nodal neighborhoods, three m-trails T_1, T_2 and T_3 are needed as shown in Fig. 1(c), by which the *alarm code table (ACT)* for each node is formed as shown in Fig. 1(d)-(g). Each row of an ACT on top of the separator corresponds to a failure state within its neighborhood, and the rows below the separator are the alarm codes seen at the node but correspond to link failures outside of its neighborhood. Note that, all codes on top of the separator in an ACT should be unique. For example, v_1 keeps the ACT as in Fig. 1(f) by observing the on-off status of T_1, W_1 and W_2, so as to uniquely identify the failure of (v_4, v_1) or (v_2, v_1) in its neighborhood. If v_1 finds that T_1 and W_1 become unexpectedly off while W_2 is still on, an alarm code $[1, 1, 0]$ is obtained; so the node will consider link (v_4, v_1) as failed by matching the first row of its ACT and be ready to switch W_1 over $P_1^{(v_4,v_1)}$. In parallel, v_2 and v_4 will be able to identify the failure of (v_4, v_1) by matching the second row in their corresponding ACTs as in Fig. 1(g) and Fig. 1(d), respectively, and instantly configure their OXCs to support $P_1^{(v_4,v_1)}$. Thus W_1 can be restored in an all-optical and deterministic fashion upon the failure of (v_4, v_1) without relying on any real-time signaling mechanism.

A. Problem Definition

The input of the G-NFL problem is an undirected graph $G = (V, E)$ with node set V and link set E, where the number of nodes is denoted by $n = |V|$ and the number of links by $m = |E|$. Given a set of W-LPs, denoted by $\mathcal{W}$, each of which can be in-band monitored by the nodes traversed by it for failure status acquisition. If a working path W_i that traverses e is interrupted due to the failure of e, the corresponding protection path P_i^e should be activated at the switching node for restoration. Let the *neighborhood* of node v be denoted by E_v, which is a set of links whose failure states can be unambiguously identified by v. Conversely, let *visibility region* of e be denoted by V_e, as a set of nodes each being able to unambiguously identify the failure of e.

The single-link G-NFL problem is to establish a set of m-trails to meet the two requirements (R1) and (R2) as follows.

(R1): each m-trail is a loopless path of G, at most one hop longer than the shortest path between its end nodes.

In this research, each m-trail is not longer than the minimum hop distance between the endpoints plus one, so that a provisioned lightpath can switch its role between a W-LP and m-trail according to the traffic demand and monitoring requirement. Such flexibility is desired in an intelligent failure localization framework and cannot be achieved with lengthy m-trails as in the previous studies [2], [3], [17]. Furthermore, short m-trails bear much better physical-layer impairment properties [4] than the previously proposed long m-trails.

The set of m-trails is denoted by $\mathcal{T} = \{T_1, \ldots, T_b\}$ where b is the number of m-trails. The objective is to minimize:

$$\|\mathcal{T}\| = \sum_{i=1}^{b} |T_i|, \tag{1}$$

where $|T_i|$ is the number of links in m-trail T_i. We expect that each node $v \in V$ can achieve NFL according to the on-off status of m-trails and W-LPs in $\mathcal{T}^v$, which is the subset of $\mathcal{T} \cup \mathcal{W}$ containing the m-trails and W-LPs passing through v. Let $\underline{\underline{A}}^v$ denote the alarm code table (ACT) at node v, where the i^{th} bit of alarm code for link e at v will be denoted by $a_{e,i}^v$ for $1 \leq i \leq |\mathcal{T}^v|$, where $|\mathcal{T}^v|$ is the number of m-trails in $\mathcal{T}^v$. We have $a_{e,i}^v = 1$ if the i^{th} m-trail passing through node v has link e and 0 otherwise.

To achieve NFL at node v, (R2) should follow:

(R2): every link e in neighborhood E_v has a unique non-zero alarm code seen at v denoted by A_e^v, and meanwhile different from all the possible link codes that v can see outside the neighborhood.

We require (R2) for every $v \in V$.

The following theorem proves the feasibility of the proposed G-NFL problem in any connected graph.

Theorem 1: Given a connected graph $G = (E, V)$ with neighborhoods E_v for every $v \in V$, an m-trail solution for G-NFL can always be found.

Proof: One can use the argument of Theorem 1 in [2]. ∎

B. Performance Metrics of G-NFL

Resources for G-NFL are identified as *transponders* (or referred to as transmitters in the following context), *lambda monitors*, and *coverlength*.

1) Transmitters: are expensive optical devices. However, we claim that the number of transmitters should not be an issue due to the following two reasons. Firstly, network providers usually prepare some amount of spare transmitters available at each OXC for heavy traffic loads. Secondly, in case more than the spare ones are required at a node, an optical splitter can be used to support multiple m-trails originated from the node.

2) Lambda Monitors: have been built-in devices in commercial DWDM equipment such as OXCs and reconfigurable optical add-drop multiplexers (ROADMs) [21], [22] for the purpose of automatic power leveling. They are essential in adjusting the signal power of individual optical channels for all-optical amplification, mostly done by attaching a monitoring photo-diode at each channel port. Thus, very little cost is incurred due to the required lambda monitoring capability at each OXC.

3) Coverlength: is the total number of WLs taken by the m-trails, which has been taken as the metric to evaluate the m-trail solutions [2], [15]–[17], and will still be the performance measure of this study.

IV. Bound Analysis

This section presents our bound analysis for the coverlength in the proposed G-NFL problem. For the sake of simplicity and without loss of generality, let $\mathcal{W} = \emptyset$. We will first consider the lower bound on a generalized version of Combinatorial Group Testing (CGT) and then apply them to the NFL requirement at each node, which will give us a lower bound on the coverlength for general graphs. The key idea is to define a special cost function for the m-trails at each node such that the lower bound to meet the NFL requirement at each node can be summed up to get a lower bound on the total coverlength.

A. General Lower Bound for CGT

Let us consider a non-adaptive CGT problem where the goal is to find one faulty item among a set of items with group tests, where each group test is on a set of items and has two outcomes: the test contains a faulty item or not. Note that the NFL problem at each node v is a special version of CGT, where the tests are the m-trails passing through v, and the items are the links. We have two additional constraints:

- the links must form a path in the topology, and
- each of the links must have a non-zero code.

It is clear that a valid NFL solution at node v is a valid CGT solution over the links in the neighborhood of v.

Next, let us formalize the CGT problem with a cost function on each test. The cost of test T_i depends on its size according to a given cost function $\omega()$. The input of the CGT problem is a set of items denoted by $E = \{e_1, \ldots, e_m\}$ and a cost function ω, where $m = |E|$ is the number of items. The goal is to establish a set of b group tests, denoted by $T_1, \ldots, T_b$, where each group test consists of a set of items, such that a single faulty item can be unambiguously identified according to the outcomes of the group tests. It is also called *separating* test collection. Each test has a cost defined as follows

Definition 2: The cost of test T_i with $t_i = |T_i|$ is $\omega(t_i)$, where function ω has the following properties:

(i) $\omega(1) = 1$, means testing one element has a unit cost.
(ii) $\omega(x+1) \geq \omega(x)$ for every positive integer $1 \leq x \leq m-1$. Testing a larger group cannot decrease the cost.
(iii) $\frac{\omega(x)}{x} \geq \frac{\omega(x+1)}{x+1}$ whenever $1 \leq x < m$.

The goal is to identify the faulty item with minimum cost:

$$\text{Minimize } \Omega = \sum_{i=1}^{b} \omega(t_i) \tag{2}$$

Note that much of the prior work focused on the cases with $\omega(t) = 1$, i.e., the cost of a test does not depend on the number of items, and thus the goal is to reduce the number of tests. Our paper is among the first studies of the CGT problem where the cost of each test is dependent on the size of the group.

Theorem 2: Suppose there are $m > 1$ items and assume (i)-(iii) holds for the cost function ω. Then for the cost of finding precisely one faulty item with group tests is at least

$$\Omega \geq \min_{1 \leq x \leq \frac{m}{2}} \omega(x) \left(\log_2 x + \frac{m}{x} - 1 \right). \tag{3}$$

Note that the minimum is taken over the integers x of the interval $[1, \frac{m}{2}]$. The proof is relegated to the Appendix.

B. Lower Bound for G-NFL

Here we develop a lower bound on the cost with an assumption weaker than (R1), by merely assuming that the m-trails are connected subgraphs. Let $r(T_i)$ denote the number of nodes the m-trail T_i passes through. These nodes are aware of the on-off status of T_i. A trivial upper bound on $r(T_i)$ is $|T_i| + 1$; or formally

$$r(T_i) \leq |T_i| + 1. \tag{4}$$

Note that the equality holds if the m-trail is a tree.

We divide the cost of each m-trail equally among the nodes it traverses, and represent the cost in a matrix Ω which has n columns and b rows, where

$$\omega_{v,i} = \begin{cases} \frac{|T_i|}{r(T_i)} & \text{the } i\text{th m-trail traverses node } v, \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

The size of T_i can be expressed as

$$\sum_{v=1}^{n} \omega_{v,i} = \sum_{v \in T_i} \frac{|T_i|}{r(T_i)} = |T_i|. \tag{6}$$

Thus we have

$$\sum_{i=1}^{b} \sum_{v=1}^{n} \omega_{v,i} = \sum_{i=1}^{b} |T_i| = \|\mathcal{T}\| \tag{7}$$

which can be reordered as

$$\|\mathcal{T}\| = \sum_{i=1}^{b} \sum_{v=1}^{n} \omega_{v,i} = \sum_{v=1}^{n} \left(\sum_{i=1}^{b} \omega_{v,i} \right) = \sum_{v=1}^{n} \Omega_v \tag{8}$$

where

$$\Omega_v = \sum_{i=1}^{b} \omega_{v,i} = \sum_{i | v \in T_i} \omega_{v,i} \ , \tag{9}$$

because $\omega_{v,i} = 0$ if the i^{th} m-trail does not traverse node v. Note that if an m-trail traverses v, then we can define a lower bound on $\omega_{v,i}$ as a function of size T_i

$$\omega_{v,i} \geq \begin{cases} \dfrac{|T_i|}{1 + |T_i|} & \text{if } |T_i| \leq n - 1 \quad , \quad \text{(10a)} \\ \dfrac{|T_i|}{n} & \text{otherwise.} \quad \text{(10b)} \end{cases}$$

To give a lower bound on Ω_v, we may consider this subproblem as a CGT problem where the cost of a group test T_i is a function of its size (cardinality), denoted by $\omega(|T_i|)$. In this case, the cost function (10) is defined separately on two intervals, (10a) is a reciprocal function and (10b) is linear. For better interpretation, let us multiply (10) by 2 so the cost is 1 when the size of a test is $|T_i| = 1$ and meets the requirements in Definition 2. We define the cost of m-trail T_i which is a function of the size of T_i as follows

$$\omega(|T_i|) = \begin{cases} \dfrac{2|T_i|}{1 + |T_i|} & \text{if } |T_i| \leq n - 1 \quad , \quad \text{(11a)} \\ \dfrac{2|T_i|}{n} & \text{otherwise.} \quad \text{(11b)} \end{cases}$$

We have

$$\omega_{v,i} \geq \frac{\omega(|T_i|)}{2} \text{ , if } v \in T_i. \tag{12}$$

Theorem 3: The total cover length for an G-NFL solution is at least

$$\|\mathcal{T}\| \geq \sum_{v \in V} \left(1 - \frac{2}{m_v + 2} \right) \log_2(m_v) \tag{13}$$

if $n - 1 \geq \frac{m_v}{2}$ for all $v \in V$, where m_v is the cardinality of the neighborhood of v, i.e., $m_v = |E_v|$.

Proof: As a lower bound on the cost of each test we use $\omega(|T_i \cap E_v|) \leq \omega(|T_i|)$. By assumption, $n - 1 \geq \frac{m_v}{2}$, thus we need to consider $x = |T_i \cap E_v| \leq \frac{m_v}{2} \leq n - 1$ only. Putting together the lower bounds on the cost in (8), (12) and applying Theorem 2 on each node we get a lower bound on Ω_v

$$\Omega_v \geq \min_{1 \leq x \leq \frac{m_v}{2}} \frac{2x}{1 + x} \left(\log_2 x + \frac{m_v}{x} - 1 \right) \tag{14}$$

where inside the min there is a decreasing function for integer values of x as proved in Lemma 2 in [3]. Thus, it leads to

$$\begin{aligned} 2\Omega_v \geq \frac{2\frac{m_v}{2}}{\frac{m_v}{2} + 1} \left(\log_2 \left(\frac{m_v}{2} \right) + \frac{m_v}{\frac{m_v}{2}} - 1 \right) = \\ = \frac{2m_v}{m_v + 2} \left(\log_2(m_v) - 1 + 2 - 1 \right) = \\ = \left(2 - \frac{4}{m_v + 2} \right) \log_2(m_v). \end{aligned} \tag{15}$$

Putting it together with (8), we get (13). ■

V. G-NFL M-Trail Allocation

The computational complexity of the optimal m-trail allocation problem with general multi-link failures is NP-hard [23], while being an open question for single link failures (which is a special case of the multi-link failure scenario). Previous studies tackled the problem via heuristic methods that can approach the derived theoretical lower bounds [2], [3], [12], [15] in light of inefficiency of using integer linear programs to solve the problem. The paper focuses on developing a simple yet effective heuristic to solve the proposed G-NFL problem. The basic idea is to successively and incrementally construct the ACT at each node such that every link code in its neighborhood is unique, i.e., different from any other link code seen by the node.

A. The G-NFL Heuristic

The detailed description of the proposed heuristic is given in Algorithm 1 and is explained step by step as follows.

In Step (2) an initial solution is taken by using single-hop m-trails for every link. The W-LPs $\mathcal{W}$ are given as the input of the algorithm ($\mathcal{W} = \emptyset$ in the out-of-band schemes), and their visibility information is set in Step (3). In Step (4) each node $v \in V$ is considered one after the other to meet the NFL requirement such that each link code in the neighborhood E_v is unique. Specifically, E_v is loaded with W-LPs in Step (5), and the current ACT $\underline{\underline{A}}^v$ is constructed based on the m-trails traversing through node v in Step (6).

Then, the heuristic enters the loop in Steps (7)-(8) for each node v, by checking whether links e_1 and e_2, where $e_1 \in E_v$ and $e_2 \in E$, have the same alarm code seen at v or not. If yes in Step (9), we place an m-trail starting from v and traversing either e_1 or e_2, but not both. To make this information local at node v, we use Dijkstra's shortest path finding algorithm in Step (10) between v and the two adjacent nodes of the corresponding link, and select the one with the shorter distance in Step (11). Finally, we add the shortest possible path to $\mathcal{M}$ in Step (13) or Step (15), and refresh the ACT of v in Step (16).

Algorithm 1: Global Neighborhood Failure Localization

Input: $G = (V, E)$, W-LPs $\mathcal{W}$
Result: $\mathcal{T}$ set of m-trails
1 **begin**
2 Use a single-hop m-trail for each link as an initial guess;
3 Set W-LP visibility from $\mathcal{W}$;
4 **for** $v \in V$ **do**
5 Load the set of neighborhood links $E_v \subseteq E$;
6 Construct current ACT $\underline{\underline{A}}^v$ at node v;
7 **for** $e_1 = (u_1, w_1) \in E_v$ **do**
8 **for** $e_2 = (u_2, w_2) \in E$ **do**
9 **if** $e_1 \neq e_2 \wedge A^v_{e_1} = A^v_{e_2}$ **then**
10 Use Dijkstra's algorithm to get the shortest paths from v to $\{u_1, w_1, u_2, w_2\}$;
11 Set $\mathcal{P}_1$ and $\mathcal{P}_2$ to the shortest path to $\{u_1, w_1\}$ and $\{u_2, w_2\}$, respectively;
12 **if** $|\mathcal{P}_1| \leq |\mathcal{P}_2|$ **then**
13 Add m-trail $\forall e \in \mathcal{P}_1 \cup e_1$ to $\mathcal{T}$;
14 **else**
15 Add m-trail $\forall e \in \mathcal{P}_2 \cup e_2$ to $\mathcal{T}$;
16 Refresh $\underline{\underline{A}}^v$;

The computational complexity of Algorithm 1 is described as follows. We have three iterations in Steps (4), (7) and (8) with $O(|V| \cdot |N| \cdot |E|)$ runs in total, where $|N|$ is the maximum size of the neighborhoods. In each iteration we compare the alarm codes, and compute a shortest path for any collided code pair with Dijkstra's shortest path finding algorithm, which can be done in $O(|E| + |V| \log |V|)$ steps. Altogether Algorithm 1 has $O(|V| \cdot |E| \cdot |N| \cdot (|E| + |V| \log |V|))$ steps in the worst case. We note here that the NL-LCC algorithm presented in [3] is subject to $O(|V| \cdot |E|^2 \cdot (|E| + |V| \log |V|))$ computational complexity, as in Step (7) each edge should be considered in the graph. As $|N| << |E|$, the G-NFL problem is considered to be more efficient and scalable than NL-LCC.

TABLE I
M-TRAIL RECONFIGURATION UPON DYNAMIC DATA PLANE CHANGES

	SOD-O	SOD-IO	LOD-O	LOD-IO
W-LP deployment	X	X		
W-LP release		X		X

B. Practical Neighborhood Scenarios

An important feature of the proposed G-NFL scenario is that each node only monitors unambiguously the links in its neighborhood. The following two classes of neighborhood definitions are studied, while the dependencies between the monitoring and data plane are summarized in Table I.

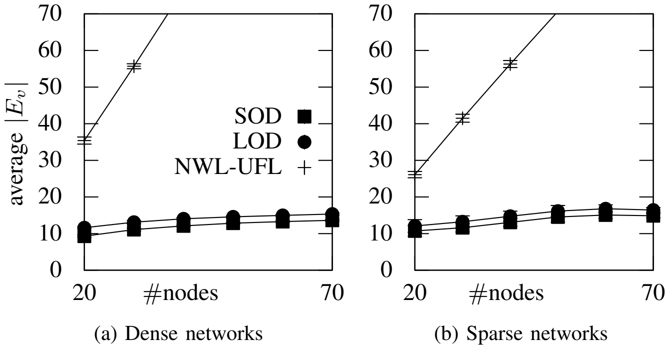
1) Strictly On-Demand (SOD): enables a node v to failure-localize link e only if node v is involved in the restoration process of the link failure e *according to the current traffic distribution*; i.e., v is either the switching, intermediate, or merging node of a P-LP which protects link e along an active W-LP. It is expected to achieve the most efficient allocation of monitoring resources due to the strictly on-demand nature, but at the expense that the dynamically changing W-LP and P-LP route information has to be considered. Such strong dependency between monitoring and data planes imposes the need for frequent reconfiguration of m-trails upon traffic distribution variations.

We consider two versions of SOD according to whether the W-LPs are taken for in-band monitoring or not, namely *SOD with out-of-band monitoring (SOD-O)*, and *SOD with in-band and out-of-band monitoring (SOD-IO)*. With the former, each node relies only on out-of-band monitoring for network failure status acquisition; while with the latter, a node can perform both out-of-band and in-band monitoring.

Obviously, both SOD-O and SOD-IO are subject to reconfiguration of m-trails upon any newly allocated W-LP. In the case of connection release, SOD-IO needs to check whether the W-LP is currently being used for in-band monitoring. If yes, the W-LP should be kept and automatically turned into an m-trail instead of being torn down immediately.

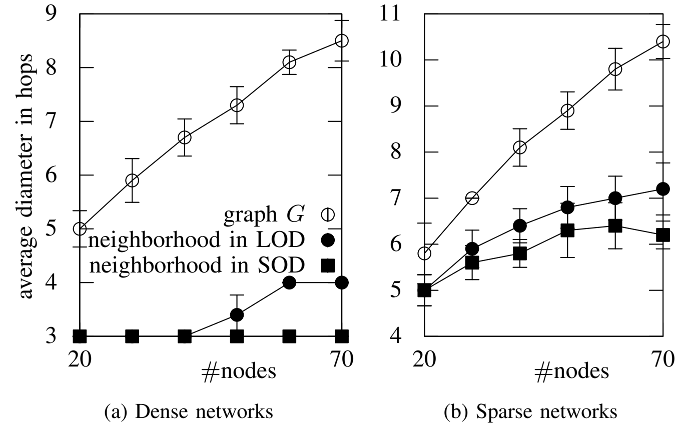
2) Loosely On-Demand (LOD): aims to significantly reduce or completely avoid reconfiguration of m-trails by maintaining a clean separation between the monitoring and data planes. It defines the neighborhood of each node by *considering all the future possible traffic* (or when the network is *fully loaded* in the following context). In this paper, we define that the network is *fully loaded* when every node pair has at least one shortest-path-routed W-LP that is protected by one or a set of P-LPs under failure dependent protection. If two W-LPs are allocated across a common node pair, they will be routed via a common route.

Two versions of LOD are implemented, namely *LOD-O* and *LOD-IO*. The former considers only out-of-band monitoring, such that the m-trails can completely ignore the arrival and departure of the W-LPs. LOD-IO is different from LOD-O



(a) Dense networks (b) Sparse networks

Fig. 2. The average size of neighborhood of each node.



(a) Dense networks (b) Sparse networks

Fig. 3. The diameter in hops of the graph G and the neighborhood regions.

by taking in-band monitoring, which results in some extent of dependency between the monitoring and data planes; i.e., when a W-LP currently used for in-band monitoring is being released, the W-LP should be automatically turned into an m-trail for supporting the monitoring plane instead of being torn down immediately.

VI. Experimental results

A set of experiments was conducted to verify the proposed G-NFL scenario[1]. Two classes of random planar graphs were generated: one for dense and the other for sparse networks, typically with the number of nodes for inner faces is between 4 and 7, and an average nodal degree 4.0 and 2.8, respectively. 30% of all node pairs are randomly selected for being loaded, where a pair of W-LPs are shortest-path routed for each loaded node pair on the same route in both directions, which is protected by a set of P-LPs shortest and diversely routed from each link of the W-LP.

A. Size of Neighborhood

We first investigated the size of neighborhood under SOD and LOD scenarios as shown in Fig. 2 (based on the randomly selected 30% and 100% of node pairs loaded with W-LPs and P-LPs, respectively). It clearly shows that the sizes of nodal neighborhoods grow very mildly as the network size increases, compared with NWL-UFL where all the links are contained in the neighborhood of each node.

We note here that the effectiveness of the FDP on NWL-UFL based on Network-Wide Local Link Code Construction (NL-LCC) was demonstrated and compared to traditional protection approaches, such as p-Cycles [3]. Under our G-NFL framework – as shown in Fig. 2 – only a fraction of the links needs to be localized, i.e., based on the W-LPs and P-LPs of the same FDP solution, G-NFL will always outperform the NL-LCC framework in S-LP allocation. Thus, G-NFL will outperform p-Cycle and other shared protection approaches as well in bandwidth consumption, as already NL-LCC did it.

[1]The simulators are available online at http://lendulet.tmit.bme.hu/demo/mtrail/

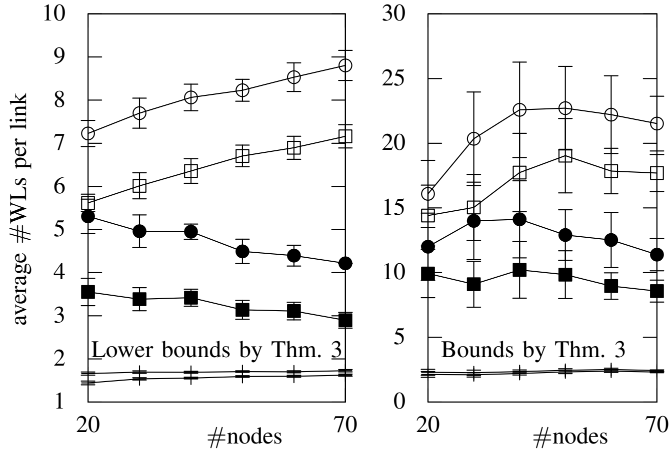
B. Restoration Time Analysis

Fig. 3 shows the average diameter d of the neighborhood of each node under SOD and LOD, which corresponds to 30% and 100% of loaded node pairs, respectively. The diameter of the neighborhood serves as an important parameter to determine the maximum length of m-trails, that is $d + 1$, which is further related to the maximum restoration time. Note that the restoration time of a W-LP under the proposed G-NFL can be simply modeled as the light propagation delay of the m-trails plus the latencies for LOL detection by the lambda monitors ($\sim$ 5 ms), nodal processing for look-up-take ($\sim$ 5 ms), and OXC configuration ($<$ 20 ms). Thus, for an m-trail of 200-400 km in length which is subject to a propagation delay of about 15 ms, we claim that the restoration time of any W-LP can be well below 50 ms.

C. Coverlength of the G-NFL Solution with FDP

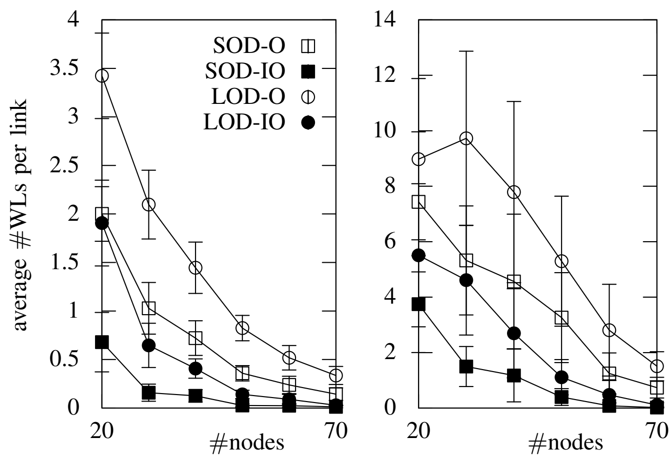
Fig. 4 shows the average WLs per link under the four definitions of nodal neighborhood. Firstly we have seen that the number of WLs per link under all the nodal neighborhood definitions scales very well when the network sizes increase; and LOD-O is outperformed by all the other schemes. This is due to the fact that it purely relies on out-of-band monitoring while having the largest possible neighborhood (as we are assuming full load). The worst performance is the price paid for the complete independence between the monitoring and data planes (see Table I for details). On the other hand, SOD-IO consumes the fewest WLs per link since it jointly considers in-band and out-of-band monitoring on the smallest possible neighborhoods according to the actual W-LPs and P-LPs (i.e., 30% loaded node pairs). The good performance is at the expense of higher m-trail reconfiguration complexity.

LOD-IO yields the second best performance among the four, and is considered a good compromise between the consumed WLs per link and the m-trail reconfiguration complexity. Note that the adoption of in-band monitoring improves performance but with little price paid as explained in Section V-B. The results derived in Theorem 3 are also sketched in Fig. 4. It is seen that some gap exists between the derived lower bounds and the SOD and LOD schemes, mostly due to the fact that



(a) Dense networks (b) Sparse networks

Fig. 4. Average number of WLs per link under the four neighborhood definitions (see Fig 5(a) for the legend) and for the lower bounds by Theorem 3 (denoted by $+$).



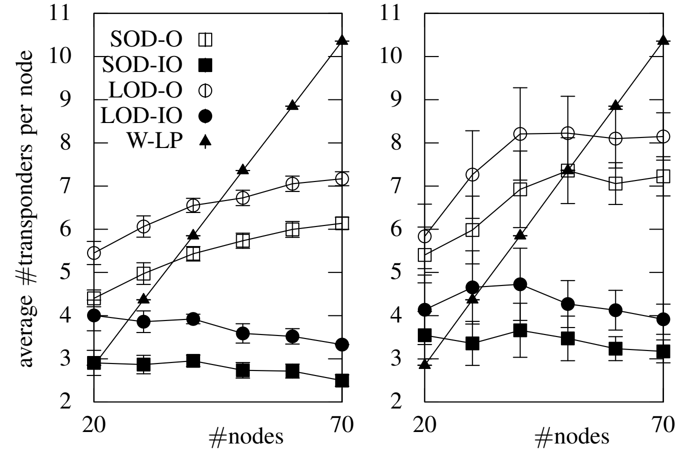
(a) Dense networks (b) Sparse networks

Fig. 5. Monitoring overhead that cannot be hidden by the spare capacity.

the analysis was purely conducted based on CGT theory and can only modestly capture the additional complexity of the G-NFL problem. However, we claim that the analytical results not only contribute to the general CGT topics, but serve as a design guideline for the proposed solutions, too.

Fig. 5 shows the average WLs per link consumed when the m-trails are allowed to reuse the spare capacity of the deployed P-LPs. We have seen that the consumed WLs per link can be significantly reduced due to the reuse; and when the network sizes are increased, such monitoring overhead can be almost hidden. In this case, the network capacity efficiency approaches to that of FDP, which has been rightly claimed as the optimal among all possible protection strategies.

Fig. 6 shows the number of required transmitters for supporting the m-trails under each neighborhood definition. We have seen superb scalability of the proposed G-NFL scenario where the number of transmitters at each node, particularly for SOD-IO and LOD-IO, becomes a very small portion among



(a) Dense networks (b) Sparse networks

Fig. 6. The required transmitters in the G-NFL solution.

the totally consumed as the network size increases.

VII. Conclusions

The paper studied a novel scenario of m-trails, called Global Neighborhood Failure Localization (G-NFL), which is uniquely characterized by signaling-free fault management, on-demand monitoring resource allocation, near shortest m-trails, and both out-of-band and in-band monitoring at each node. By assuming the capability of lambda monitoring at each node, we justified the use of m-trails as an effective alternative to the current industry practices. We exemplified how signaling-free and all-optical restoration under general shared protection can be achieved, and reasoned the use of total coverlength as the metric instead of any other such as the number of lambda monitors and transmitters. In particular, the neighborhood of a node is defined as a set of links that the node has to respond when any of them fails, by which the proposed G-NFL problem was formulated.

Bound analysis was conducted via a novel CGT theory which was applied to the proposed G-NFL problem. A simple yet effective heuristic was developed which incrementally constructs the alarm code table at each node. Extensive simulation was conducted to verify the proposed G-NFL scenario. We first examined the average sizes of some practical neighborhood definitions under FDP and identified the desired scalability, which ensures nearly constant monitoring resource consumption when the network size grows under all the nodal neighborhood definitions. We further demonstrated that the monitoring overhead can be almost hidden by the spare capacity of FDP when the network size is getting larger, while taking a reasonable amount of transmitters at each node. Finally, we conclude that the proposed G-NFL scenario can effectively enable a general shared protection scheme, such as FDP, to yield ultra-fast and all-optical failure restoration as in $1+1$ and p-Cycle, while possibly achieving the optimal capacity efficiency as in FDP.

References

[1] W. Grover, J. Doucette, M. Clouqueur, D. Leung, and D. Stamatelakis, "New options and insights for survivable transport networks," *IEEE Commun. Mag.*, vol. 40, no. 1, pp. 34–41, Jan. 2002.

[2] J. Tapolcai, P.-H. Ho, L. Rónyai, and B. Wu, "Network-wide local unambiguous failure localization (nwl-ufl) via monitoring trails," *IEEE/ACM Transactions on Networking*, vol. 20, no. 6, pp. 1762–1773, 2012.

[3] J. Tapolcai, P.-H. Ho, P. Babarczi, and L. Rónyai, "On signaling-free failure dependent restoration in all-optical mesh networks," *IEEE/ACM Transactions on Networking*, pp. 1–12, 2014, in press.

[4] É. Hosszu, E. S. Moghaddam, J. Tapolcai, and D. Mazroa, "Physical impairments of monitoring trails in all optical transparent networks," *IET Networks*, vol. 2, no. 4, pp. 196–203, 2013.

[5] J. Tapolcai, P.-H. Ho, P. Babarczi, and L. Rónyai, "On achieving all-optical failure restoration via monitoring trails," in *Proc. IEEE INFOCOM*, 2013, pp. 380–384.

[6] D. Papadimitriou and E. Mannie, "Analysis of generalized mpls-based recovery mechanisms," Internet Draft, Sep. 2003.

[7] H. Zeng, C. Huang, and A. Vukovic, "A Novel Fault Detection and Localization Scheme for Mesh All-optical Networks Based on Monitoring-cycles," *Photonic Network Comm.*, vol. 11, no. 3, pp. 277–286, 2006.

[8] C. Li, R. Ramaswami, I. Center, and Y. Heights, "Automatic fault detection, isolation, and recovery in transparentall-optical networks," *IEEE/OSA J. Lightwave Technol.*, vol. 15, no. 10, pp. 1784–1793, 1997.

[9] Y. Wen, V. Chan, and L. Zheng, "Efficient fault-diagnosis algorithms for all-optical WDM networks with probabilistic link failures," *IEEE/OSA J. Lightwave Technol.*, vol. 23, pp. 3358–3371, 2005.

[10] 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, pp. 2676–2680.

[11] B. Wu, P.-H. Ho, and K. Yeung, "Monitoring trail: On fast link failure localization in all-optical WDM mesh networks," *IEEE/OSA J. Lightwave Technol.*, vol. 27, no. 18, pp. 4175–4185, 2009.

[12] J. Tapolcai, B. Wu, P.-H. Ho, and L. Rónyai, "A novel approach for failure localization in all-optical mesh networks," *IEEE/ACM Transactions on Networking*, vol. 19, no. 1, pp. 275–285, 2011.

[13] N. Harvey, M. Patrascu, Y. Wen, S. Yekhanin, and V. Chan, "Non-Adaptive Fault Diagnosis for All-Optical Networks via Combinatorial Group Testing on Graphs," in *Proc. IEEE INFOCOM*, 2007, pp. 697–705.

[14] P. Babarczi, J. Tapolcai, and P.-H. Ho, "Adjacent link failure localization with monitoring trails in all-optical mesh networks," *IEEE/ACM Transactions on Networking*, vol. 19, no. 3, pp. 907 – 920, June 2011.

[15] S. Ahuja, S. Ramasubramanian, and M. Krunz, "Single link failure detection in all-optical networks using monitoring cycles and paths," *IEEE/ACM Trans. Networking*, vol. 17, no. 4, pp. 1080–1093, 2009.

[16] B. Wu, P.-H. Ho, J. Tapolcai, and X. Jiang, "A novel framework of fast and unambiguous link failure localization via monitoring trails," in *IEEE INFOCOM WIP*, San Diego, 2010, pp. 1–5.

[17] W. He, P.-H. Ho, B. Wu, and J. Tapolcai, "On identifying srlg failures in all-optical networks," *Optical Switching and Networking*, vol. 10, no. 1, pp. 77 – 88, 2013.

[18] D. Du and F. K. Hwang, *Combinatorial Group Testing and Its Applications*. World Scientific, 2000.

[19] B. Bollobás and A. Scott, "On separating systems," *European Journal of Combinatorics*, vol. 28, no. 4, pp. 1068–1071, 2007.

[20] A. Rényi, "On random generating elements of a finite boolean algebra," *Acta Sci. Math. (Szeged)*, vol. 22, no. 1–2, pp. 75–81, 1961.

[21] "Cisco ONS 15454 Multiservice Transport Platform," 2012, data sheet. [Online]. Available: http://www.cisco.com

[22] "Huawei OptiX BWS 1600A DWDM System," 2012, product brochure. [Online]. Available: http://www.huawei.com

[23] P. Babarczi, J. Tapolcai, and P.-H. Ho, "SRLG failure localization with monitoring trails in all-optical mesh networks," in *Proc. IEEE DRCN*, Krakow, Poland, 2011, pp. 188–195.

Appendix

Suppose there are $m > 1$ items and assume (i)-(iii) holds for the cost function ω. Then for the cost of finding precisely one faulty item with group tests is at least

$$\Omega \geq \min_{1\leq x\leq \frac{m}{2}} \omega(x)\left(\log_2 x + \frac{m}{x} - 1\right) \tag{16}$$

Proof: Let us sort the tests in descending size, so that T_1 has the largest number of items while T_b has the least: we assume that

$$t_1 \geq t_2 \geq \cdots \geq t_b$$

where $t_i = |T_i|$ denotes the number of items in test T_i.

Also, we may assume that $t_i \leq \frac{m}{2}$ for every i. Indeed, a test set T_i with $|T_i| \geq \frac{m}{2}$ can be replaced by its complementary set $E\setminus T_i$. The resulting test collection still remains separating if the original one was separating.

We build up the $b\times m$ matrix whose rows are the characteristic vectors of the tests $T_i \subseteq E$, by adding the rows one-by-one, and in each step we count the number of different columns in the matrix. Let f_i denote the number of different columns when the matrix has i rows, i.e. tests $T_1,\ldots,T_i$ are present, the others are not. For convenience we set $f_0 = 1$. Adding a row the number of different columns cannot decrease, thus $f_{i-1} \leq f_i$ for $i = 1,\ldots,b$. As we have a separating system, all the m columns will be different when the last row is added, giving that $f_b = m$.

When we add T_i, the number of different columns is at most doubled, hence $f_i \leq 2f_{i-1}$, or

$$\log_2(f_i) - \log_2(f_{i-1}) \leq 1 \tag{17}$$

for $i = 1,\ldots,b$.

Similarly, by adding test T_i to the collection $T_1,\ldots,T_{i-1}$ can increase the number of different columns in the matrix by at most t_i, giving $f_i \leq f_{i-1} + t_i$, or

$$\frac{f_i - f_{i-1}}{t_i} \leq 1 \tag{18}$$

for $i = 1,\ldots,b$.

Now fix an integer k with $1 \leq k < b$. We have

$$\Omega = \sum_{i=1}^{b}\omega(t_i) \geq \sum_{i=1}^{k}\omega(t_i)\left(\log_2(f_i) - \log_2(f_{i-1})\right) + \\ + \sum_{i=k+1}^{b}\omega(t_i)\left(\frac{f_i - f_{i-1}}{t_i}\right). \tag{19}$$

In the first sum we used (17), and (18) in the second.

The sequence $\omega(t_i)$ is nonincreasing for $i = 1,\ldots,b$ by (ii) and our numbering of the tests, hence

$$\sum_{i=1}^{k}\omega(t_i)(\log_2(f_i) - \log_2(f_{i-1})) \geq \\ \geq \sum_{i=1}^{k}\omega(t_k)(\log_2(f_i) - \log_2(f_{i-1})) = \omega(t_k)\log_2(f_k). \tag{20}$$

Similarly, the sequence $\frac{\omega(t_i)}{t_i}$ is nondecreasing because of (iii) and our numbering of the tests, giving that

$$\sum_{i=k+1}^{b}\omega(t_i)\left(\frac{f_i - f_{i-1}}{t_i}\right) \geq \\ \geq \sum_{i=k+1}^{b}\omega(t_{k+1})\left(\frac{f_i - f_{i-1}}{t_{k+1}}\right) = \frac{\omega(t_{k+1})}{t_{k+1}}(m - f_k). \tag{21}$$

By substituting (20) and (21) into (19), we have

$$\Omega \geq \omega(t_k)\log_2(f_k) + \frac{\omega(t_{k+1})}{t_{k+1}}(m - f_k) \geq$$
$$\geq \omega(t_k)\left(\log_2(f_k) + \frac{m-f_k}{t_k}\right). \quad (22)$$

This inequality is valid for any k with $1 \leq k < b$. Let we set now k to be the first index j for which $t_j \leq f_j$. Such index clearly exists and $k < b$ because $f_{b-1} \geq \frac{m}{2}$, while $t_i \leq \frac{m}{2}$ for every i. We need to consider two cases:
(1) If $f_{k-1} \leq t_k$. We start from

$$\Omega \geq \omega(t_k)\left(\log_2(f_k) + \frac{m-f_k}{t_k}\right).$$

Note that $t_k \leq f_k \leq 2f_{k-1} \leq 2t_k$, hence for δ defined by $f_k = t_k + \delta$ we have $0 \leq \delta \leq t_k$. Moreover,

$$\Omega \geq \omega(t_k)\left(\log_2(t_k+\delta) + \frac{m - t_k - \delta}{t_k}\right) =$$
$$= \omega(t_k)\left(\log_2(t_k+\delta) - \frac{\delta}{t_k} + \frac{m}{t_k} - 1\right). \quad (23)$$

On the interval $0 \leq x \leq 1$ we have the inequality $x \leq \log_2(1+x)$. We apply this for $x = \frac{\delta}{t_k}$. Note that $0 \leq \delta \leq t_k$ implies that $0 \leq x \leq 1$. We obtain the following inequality

$$\log_2(t_k+\delta) - \frac{\delta}{t_k} \geq \log_2(t_k+\delta) - \log_2\left(1 + \frac{\delta}{t_k}\right) =$$
$$= \log_2\left(\frac{t_k+\delta}{1+\frac{\delta}{t_k}}\right) = \log_2\left(\frac{t_k+\delta}{\frac{t_k+\delta}{t_k}}\right) = \log_2(t_k). \quad (24)$$

Substituting (24) into (23) we get

$$\Omega \geq \omega(t_k)\left(\log_2(t_k) + \frac{m}{t_k} - 1\right).$$

(2) If $t_k < f_{k-1}$, then $k > 1$ because $f_0 = 1$ by definition. Thus $f_{k-1} < t_{k-1}$ and based on (22) we have

$$\Omega \geq \omega(t_{k-1})\log_2(f_{k-1}) + \frac{\omega(t_k)}{t_k}(m - f_{k-1}).$$

Since $\frac{\omega(t)}{t}$ is a nonincreasing function of t, we have

$$\Omega \geq \omega(f_{k-1})\log_2(f_{k-1}) + \frac{\omega(f_{k-1})}{f_{k-1}}(m - f_{k-1}) \geq$$
$$\geq \omega(f_{k-1})\left(\log_2(f_{k-1}) + \frac{m}{f_{k-1}} - 1\right). \quad (25)$$

In both cases there is an integer x in the interval $[1, \frac{m}{2}]$ such that

$$\Omega \geq \omega(x)\left(\log_2(x) + \frac{m}{x} - 1\right).$$

This is because $f_{k-1} < t_{k-1} \leq \frac{m}{2}$ and $t_k \leq \frac{m}{2}$ hold. This proves the theorem. ■

János Tapolcai received his M.Sc. ('00 in Technical Informatics), and Ph.D. ('05 in Computer Science) degrees from Budapest University of Technology and Economics (BME), Budapest, and D.Sc. ('13 in Engineering Science) from Hungarian Academy of Sciences (MTA). Currently he is an Associate Professor at the High-Speed Networks Laboratory at the Department of Telecommunications and Media Informatics at BME. His research interests include applied mathematics, combinatorial optimization, optical networks and IP routing, addressing and survivability. He is an author of over 100 scientific publications, he is the recipient of the Google Faculty Award and Best Paper Award in ICC'06, in DRCN'11. He is a TPC member of leading conferences such as IEEE INFOCOM (2012 - 2015), and is a winner of MTA Lendület Program and Google Faculty Award.

Pin-Han Ho received his B.Sc. and M.Sc. degrees from the Electrical and Computer Engineering, Department of National Taiwan University in 1993 and 1995, respectively. He started his Ph.D. studies in 2000 at Queen's University, Kingston, Ontario, Canada, focusing on optical communications systems, survivable networking, and QoS routing problems. He finished his Ph.D. in 2002, and joined the Electrical and Computer Engineering Department at the University of Waterloo as an assistant professor in the same year. He is the author/co-author of more than 100 refereed technical papers and book chapters, and the co-author of a book on optical networking and survivability. He 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 SPECTS '02 and the ICC '05 Optical Networking Symposium, and the Outstanding Paper Award in HPSR '02.

Péter Babarczi is currently working as an Assistant Professor in the High-Speed Networks Laboratory at the Department of Telecommunications and Media Informatics at the Budapest University of Technology and Economics, Hungary, where he received the M.Sc. and Ph.D. degree with summa cum laude in computer science in 2008 and 2012, respectively. In 2012 he held an appointment as a Post-Doctoral Research Associate at the University of Waterloo, Canada and at the University of Oklahoma, USA. In 2013 he won the János Bolyai Research Scholarship of the Hungarian Academy of Sciences. His research interests include all-optical failure localization, combinatorial optimization, dedicated protection and network coding.

Lajos Rónyai is a research professor with the Informatics Laboratory of the Computer and Automation Institute of the Hungarian Academy of Sciences. He leads a research group there which focuses on theoretical computer science and discrete mathematics. He is also a full professor at the Mathematics Institute of the Budapest University of Technology and Economics. He received his PhD in 1987 from the Eötvös Loránd University, Budapest. His research interests include efficient algorithms, complexity of computation, algebra, and discrete mathematics.