

Link Fault Localization using Bi-directional M-Trails in All-Optical Mesh Networks

János Tapolcai*, Lajos Rónyai†, and Pin-Han Ho‡

* MTA-BME Future Internet Research Group, tapolcai@tmit.bme.hu

† Informatics Lab., Computer and Automation Research Inst., Hungarian Academy of Sciences (MTA SZTAKI),
2nd Inst. of Mathematics, Budapest University of Technology and Economics (BME), ronyai@sztaki.hu

‡ Dept. of Electrical and Computer Engineering, University of Waterloo, Canada, p4ho@uwaterloo.ca

Abstract—The paper considers the problem of single-link failure localization in all-optical mesh networks. Our study follows a generic monitoring approach using supervisory lightpaths (S-LPs), in which a set of bi-directional monitoring trails (bm-trails) are defined and closely monitored, such that the network controller can achieve unambiguous failure localization (UFL) for any single link by collecting the flooded alarms from the affected bm-trails. With a target of minimizing the number of bm-trails (or the length of alarm codes) required for single-link UFL, the paper provides optimal (or essentially optimal) solutions to the bm-trail allocation problem on a number of well known topologies. First we demonstrate that the theoretical lower bound of $\lceil \log_2(|E| + 1) \rceil$ bm-trails can be achieved in any $2 \cdot \lceil \log_2(|E| + 1) \rceil$ connected graph, where $|E|$ is the number of links. Next, we prove an essentially optimal solution for 1-by-N grid topologies (also known as *chocolate bar graphs*), where $\lceil 0.42 + \log_2(|E| + 2) \rceil$ bm-trails can be achieved. Based on the solution for chocolate bars, we further investigate bm-trail solutions to general 2-dimensional (2D) grid topologies, and the developed solution requires no more than $3 + \lceil \log_2(|E| + 1) \rceil$ bm-trails for UFL. Such an optimal (or essentially optimal) logarithmic behavior, although has been well observed in general topologies in our previous studies [1], [2], is formalized for the first time in this paper via a suite of polynomial-time deterministic constructions that consume less than a few seconds of running time in topologies of thousands of nodes.

I. INTRODUCTION

Fast and precise failure localization is a critical task in all-optical mesh backbone networks, and it has been extensively studied in the past decade [2]–[12]. Due to the transparency in the data plane, a single failure may trigger a large number of redundant alarms [13], [14], which not only increases the management complexity but also makes the failure localization more difficult. With a fast fault localization/diagnosis plane, the network operators can real-time monitor the network performance behaviour via very efficient redundancy placement strategies (e.g., failure dependent protection), and dynamically recover any unexpected failure event in a time scale of tens of milliseconds.

One of the most commonly adopted approaches is deploying a set of supervisor lightpaths (S-LPs) that are closely and actively monitored by the respective monitors, which generate

alarms when any failure/irregularity is detected. The alarms are then collected by the network controller such that any failure event can be unambiguously localized. Link monitoring serves as the most straightforward approach by deploying single-hop S-LPs to monitor each link; however, the simplicity is at the expense of $O(|E|)$ number of monitors and the alarm code length, where $|E|$ is the number of links. It is highly desired to keep the number of active monitors as small as possible while achieving unambiguous failure localization (UFL) for any failure event under consideration.

In general, a network topology is modeled as a graph with two directed links between some pairs of nodes in opposite directions. Each S-LP can be represented by a connected link set in the topology. In the literature, a number of different structures for the S-LPs are reported, and each structure brings about a specific restriction in the formation of the corresponding S-LPs. For example, a simple monitoring cycle (m-cycle) [4]–[7] is a connected graph where each node has a nodal degree 2; a non-simple m-cycle [15] is a connected subgraph where all nodes have even nodal degrees; and undirected monitoring trails (um-trails) [1], [13], [16] are connected graphs with no more than two odd-degree nodes. The most general structure is bi-directional monitoring trail (bm-trail), where each S-LP is simply a connected subgraph of the topology subject to no further restrictions. Such a structure for S-LPs has been explored in [16], [17] by assuming that each node can perform loop-back switching by which optical signals coming from an incoming fiber to a node can be switched to the same fiber segment leaving the node. It is clear that with less restriction on the structure of S-LPs, better performance can be achieved in terms of the number of S-LPs required for UFL.

Ideally, according to the binary coding mechanism [1], b S-LPs can localize any single link failure in a topology with up to $2^b - 1$ links. In this case, each S-LP corresponds to a single bit in the alarm code, which results in a monitoring system with alarm codes of b bits in length. In other words, ideally $\lceil \log_2(|E| + 1) \rceil$ S-LPs would be sufficient for localizing any single link failure in a topology with a set of links E . This lower bound on the number of S-LPs, nonetheless, may not be easily achieved (if not impossible) due to the connectivity constraint on routing each S-LP in different topologies. We have discovered in our previous studies [1], [2] via solving an optimal integer linear program (ILP) and a heuristic method,

R.L. was supported by the Hungarian Research Fund (OTKA grants NK 105645, K77476, and K77778). J.T. was supported by the project TÁMOP 421B-09/1/KMR-2010-0002. P. -H. Ho was supported by National Science and Engineering Research Council (NSERC), Canada. The paper was presented in part in Infocom 2010 (miniconference, 5 pages).

respectively, that a logarithmic solution can be achieved on most of the topologies without any degree-two nodes. However, it is still an open question on whether the logarithmic bound holds for large, particularly relevant topologies. Note that the ILP in [2] can only deal with topologies with less than 20 nodes due to the extremely large computation complexity. The heuristic in [1], managed to solve the uni-directional m-trail allocation problem in topologies with hundreds of nodes and links. Although it provides a convincing performance on typical backbone network topologies, we have also witnessed a degradation of solution quality in terms of the number of m-trails for UFL in some sparsely connected topologies with large diameters (such as N-by-1 grids).

To the best of our knowledge, the previously reported analytical studies that investigated the upper bound on the number of S-LPs for UFL were reported for rings and complete graphs [1], [17], where $c \cdot \log_2(|E| + 1)$ bound on the number of probes was found for localizing any link failure in a topology with $|E|$ links, with some constant $c \geq 2$. The result, on one hand, is very pleasing as it provides an excellent asymptotic upper bound on the problem. On the other hand, it opens up perhaps a no less challenging problem in constructing truly practical solutions that could be implemented and manipulated for engineering purposes. To accomplish this, one has to replace the constant $c = 2$ for densely connected and $c = 3$ for lattice topologies of [1], [17], and the obtained upper bounds should be of the form $\log_2(|E| + 1) + K$ where K is a small constant.

Motivated by the above observation, the paper investigates optimal (or essentially optimal) logarithmic solutions for single link failure localization using bi-directed m-trails (bm-trails) in $2 \cdot \lceil \log_2(|E| + 1) \rceil$ connected graphs, as well as in general 2-dimensional (2D) lattices (i.e. Manhattan grids), in view of the information theoretical lower bound $\lceil \log_2(|E| + 1) \rceil$. Note that a 2D grid graph is similar to circulant graphs with every node of nodal degree 4, which have been recognized as excellent candidate topologies for high-reliability network design in modern communication networks [18]–[21]. For example, circulant graphs of degree four have been considered for the design of WDM local networks and interconnection subsystems [18], [20], [21].

The contributions of the paper lie in the two theorems that demonstrate the proposed constructions. Firstly, we provide an optimal construction for bm-trail allocation on $2 \cdot \lceil \log_2(|E| + 1) \rceil$ -connected very dense networks as in Theorem 1; and this is the only optimal construction that has been published where the information theoretical lower bound $\lceil \log_2(|E| + 1) \rceil$ is tight. In Theorem 2 we demonstrate that, somewhat contrary to preliminary intuition, topologies with small constant degrees may also allow a very compact set of error locating bm-trails. This is established by using planar rectangular grids. Former results on the field are widely based on combinatorial and graph theoretical techniques and arguments. Theorem 2 does not seem to be amenable to such methods, while employing algebraic tools that appears to be entirely new in this field of communication engineering. Our constructions give essentially optimal results in settings where efficient and natural randomized approaches fail badly,

apparently by orders of magnitude.

The paper provides rigorous proofs for the correctness on the developed constructions and the obtained upper bounds, and in principle no simulation is needed for verification. Nonetheless, simulation is conducted to examine a number of most recently reported schemes for single-link UFL on N-by-1 grids - a special case of the 2D grid topologies, in order to demonstrate the difficulty for the schemes in handling the topologies with very large diameters. It shows that all the schemes take hours to converge while with far-from-optimal solutions, in contrast to the proposed construction which takes only a few seconds for essential optimality.

The rest of the paper is organized as follows. Section II presents the background and formulation for the S-LP allocation problem in communication networks. Section III provides a comprehensive analysis of the S-LP allocation problem for fully connected topologies. In Section IV, an essentially optimal solution is given for 2D lattice graphs. Section V conducts a comparison between the proposed construction on 2D grids and the a number of previously reported heuristics for the same purpose. Section VI concludes the paper.

II. BACKGROUND

A. Bi-directional Monitoring Trails (bm-trails)

In short, a bm-trail can traverse a link in different directions via loop-back switching and also a node multiple times. By properly allocating a set of bm-trail S-LPs, the network controller is expected to localize any link failure event by collecting the alarm signals issued by the corresponding monitors of the affected S-LPs in a timely manner. Let the transmitter and receiver of a bm-trail be denoted as T and R , respectively. As shown in Fig. 1(a), the bm-trail can be pre-cross-connected along the route $T \rightarrow a \rightarrow c \rightarrow a \rightarrow R$.

In general, a bm-trail S-LP solution consists of a set of b S-LPs t_1, t_2, \dots, t_b . Upon a single link failure, the monitor of any S-LP traversing the failed link will generate an alarm, which is sent to the network controller with the highest priority via any possible signaling protocol that supports point-to-point event-driven notifications. An alarm code $[a_1, \dots, a_{b-1}, a_b]$ can be formed at the network controller by reading the status of each S-LP, where $a_k = 1$ means that the monitor on S-LP t_k alarms and $a_k = 0$ otherwise. Fig. 1(b) shows a solution with three bm-trail S-LPs t_1, t_2, t_3 . If link $(0, 1)$ fails, the monitors on t_1 and t_3 will alarm to produce the alarm code $[1, 0, 1]$ at the network controller. Similarly, if link $(0, 2)$ fails, the monitors on all the three bm-trails S-LP will alarm and the resulting alarm code is $[1, 1, 1]$. The alarm code table (ACT) in Fig. 1(c) is available in the network controller, which maintains all the possible alarm codes corresponding to each failure state. Thus, the network controller can unambiguously localize a particular single link failure by matching the alarm code in the ACT.

B. Problem Definition - Deployment of S-LPs for UFL

Let a graph denoted as $G(E, V)$ be given with $|E|$ links and $|V|$ nodes. In order to achieve UFL, each link e must be assigned with a unique binary alarm code $c(e) = A_e =$

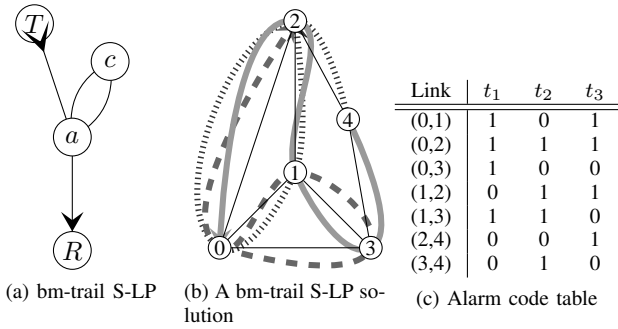


Fig. 1. Fast link failure localization based on bm-trail S-LPs.

$[a_1^e, a_2^e, \dots, a_b^e]$, where b is the length of alarm code, and a_l^e is a binary digit, which is 1 if the l^{th} bm-trail S-LP, denoted by t_l , traverses through this link and 0 otherwise. As discussed in Section I, various conditions could be imposed on t_l which lead to different failure location models, such as simple/non-simple cycles, um-trails and bm-trails. A corresponding solution is referred to as *S-LP formation*. In all three models, the theoretical lower bound on b is $b \geq \lceil \log_2(|E| + 1) \rceil$, (see [1]). In this paper, we are interested in when the above bound is essentially tight in the bm-trail model. An bm-trail S-LP formation is *essentially optimal* if $b = \lceil \log_2(|E| + 1) + c \rceil$ bm-trail S-LPs are sufficient for achieving UFL, where c is a small constant.

The main objective of the paper is to develop a suite of deterministic polynomial time constructions that achieve essentially optimal bm-trail S-LP formation on complete graphs and 2-dimensional lattice topologies.

C. Previous Work

The bm-trail S-LP formation problem is a structured variant of the combinatorial group testing (CGT) process [22], [23]. In [17] the problem was taken as *combinatorial group testing on graphs*. The idea of group testing dates back to World War II when millions of blood samples were analyzed to detect syphilis in US military. In order to reduce the number of tests it was suggested to pool the blood samples. From algorithmic aspects there are two significant differences between the tasks of pooling blood samples and monitoring of a group of links in a graph: (1) the blood samples can be pooled arbitrarily, while the monitored links must be connected and even in a valid shape, (2) the monitors are always pre-configured, and the probing is performed simultaneously without knowing the result of other tests (non adaptive CGT).

The primary goal of any CGT algorithm is to identify defective items among a given set of items through as few tests as possible. In this case the set of items are the links of a graph, the defective items are the failed links, and the tests are monitoring structures (e.g., bm-trails). The first general CGT method was given by Hwang and T. Sós [22], while the shortest real-world problem size non-adaptive CGT codes were developed by Epstein, Goodrich and Hirschberg [24]. Note that CGT algorithm codes are rather suitable to deal with the scenarios with multi-link SRLGs; on the other hand it is sufficient to have the code assigned to each link to

be distinguished from each other in the single-link SRLG scenarios. This immediately leads to a trivial lower bound $\lceil \log_2(|E| + 1) \rceil$ on the number of probes, where each probe, provided with a proper design, is realized by launching a single S-LP.

Table I summarizes the best known lower and upper bounds on the number of S-LPs reported in the literature for several special graphs. For ring topologies, the number of optimal bm-trails is exactly $\lceil |E|/2 \rceil$, which was proved first in [17], and later for um-trails in [1]. Note that in ring topologies each bm-trail should only be a simple path.

The study [17] developed a construction for any graph which contains two edge disjoint spanning trees, where an upper bound of $2 \cdot \lceil \log_2 |E| \rceil - 1$ bm-trails can be achieved. The key idea of the construction is to categorize the links in the topology into two disjoint sets E_1, E_2 of similar sizes, where $E_1 \cup E_2 = E$, $E_1 \cap E_2 = \emptyset$, and each set contains a spanning tree. We shall generate alarm codes of length $\lceil \log_2 |E_1| \rceil + \lceil \log_2 |E_2| \rceil$ for the links in E . The links in E_1 will have unique codes in the first $\lceil \log_2 |E_1| \rceil$ bits, and similarly the links in E_2 are coded uniquely in the last $\lceil \log_2 |E_2| \rceil$ bits. At this point every link has a unique individual alarm code, irrespective of the values in the bits in the alarm code that has not yet been specified. These unspecified bits can be used to make the resulting test sets connected and form bm-trails. Finally, we add one additional bm-trail covering every link in E_1 and none in E_2 , which can identify if the failed link belongs to E_1 or E_2 . In such a way, each link has a unique alarm code with a length:

$$\begin{aligned} & \lceil \log_2 |E_1| \rceil + \lceil \log_2 |E_2| \rceil + 1 \approx \\ & \lceil \log_2 \frac{|E|}{2} \rceil + \lceil \log_2 \frac{|E|}{2} \rceil + 1 = 2 \cdot \lceil \log_2 |E| \rceil - 1 \quad (1) \end{aligned}$$

We refine this idea further in Section III.

Nash-Williams and Tutte [25] showed that every $2k$ -connected graph has k link-disjoint spanning trees. Note that the disjoint spanning trees can be found in $O(|V||E| \log \frac{|E|}{|V|})$ time [26]. As a result, every 4-connected graph has always 2 link-disjoint spanning trees, thus the proof is valid for complete graphs with more than 5 nodes¹. For 2-dimensional square grid lattices, on the other hand, a similar technique was developed in [17] which results in $2 + 6 \cdot \lceil \log_2(n + 1) \rceil$ as an upper bound on the number of bm-trails, where the graph has $(n - 1) \times (n - 1)$ nodes. In fact due to $|E| = 2n^2 + 2n < 2(n + 1)^2$ in a square grid lattice, it leads to

$$2 + 6 \cdot \lceil \log_2(n + 1) \rceil \lesssim 2 + 6 \lceil \log_2 \sqrt{\frac{|E|}{2}} \rceil \approx 3 \lceil \log_2 |E| \rceil,$$

which is about 3 times of the theoretical lower bound: $\lceil \log_2(|E| + 1) \rceil$.

In [27], an observation made from extensive simulations on thousands of general topologies is that, the um-trail solution on a topology without degree-2 nodes can achieve the theoretical lower bound of $1 + \lceil \log_2(|E| + 1) \rceil$ provided sufficient running time for the construction. This was disproved by

¹Based on a similar approach, an upper bound $(6 + \lceil \log_2(|E| + 1) \rceil)$ for the um-trail formation problem was proved in [1].

Topology	trails	lower bound	upper bound	tight
ring	bm-	$\lceil E /2 \rceil$ [17]	$\lceil E /2 \rceil$ [17]	[17]
2D grid	bm-	$\lceil \log_2(E + 1) \rceil$	$\approx 3 \lceil \log_2 E \rceil$ [17]	Sec. IV. with +3
well connected	bm-	$\lceil \log_2(E + 1) \rceil$	$\approx 2 \lceil \log_2 E \rceil$ -1 [17]	Sec. III.
fully connected	m-	$\lceil \log_2(E + 1) \rceil$	$\lceil \log_2(E + 1) \rceil$ +6 [1]	Sec. III.

TABLE I

THE BEST KNOWN LOWER AND UPPER BOUNDS ON THE NUMBER OF S-LPS OF DIFFERENT STRUCTURES FOR RINGS, COMPLETE GRAPHS, AND 2D GRIDS

an example in [1]. Thus, we are motivated to develop a suite of polynomial-time deterministic constructions toward optimal (or essentially optimal) solutions for the bm-trail S-LP formation problem in densely meshed topologies and 2-dimensional grid lattices. The analytical constructions should be able to solve the bm-trail allocation problem in topologies with thousands of nodes by using only seconds.

III. OPTIMAL BM-TRAIL SOLUTION IN DENSELY MESHED GRAPHS

We shall need a simple inequity, which is proved in the appendix.

Lemma 1: The following inequality holds for every positive integer $b \geq 3$:

$$2 \cdot \lfloor \frac{2^b - 1}{b} \rfloor \geq \frac{2^{b+1} - 1}{b+1} \geq \lceil \frac{2^b - 1}{b} \rceil. \quad (2)$$

Next let us prove a lemma which is an important building block for the subsequent description on the proposed construction and its proof.

Lemma 2: The nonzero binary codewords of length b can be distributed into b buckets, where the i^{th} bucket contains codewords only with 1 for the i^{th} bit, and the size of each bucket is at least $\lfloor \frac{2^b - 1}{b} \rfloor$ and at most $\lceil \frac{2^b - 1}{b} \rceil$.

Proof: The proof is inductive, and we will give a recursive construction for such a distribution of codewords. See Fig. 2 as an illustration of each recursive step.

Clearly, for $b = 1, 2$ the statement trivially holds. Let us assume that the codewords of length b are already distributed into b buckets, where the i^{th} bucket has words only with 1 for the i^{th} bit, and the size of each bucket is at least $\lfloor \frac{2^b - 1}{b} \rfloor$ and at most $\lceil \frac{2^b - 1}{b} \rceil$. We define such a distribution as *almost uniform distribution of b bits*.

Next, we consider the nonzero codewords of length $b + 1$, and prove that the codewords can follow the almost uniform distribution of $b + 1$ bits. Clearly we can distribute the $2^b - 1$ codewords with 0 in the $(b + 1)^{\text{th}}$ bit such that the first b bits are distributed almost uniformly (according to the given assumption under the inductive proof); namely the first b buckets are filled up with at least $\lfloor \frac{2^b - 1}{b} \rfloor$ and at most $\lceil \frac{2^b - 1}{b} \rceil$ codewords. At the end these buckets must have at least $\lfloor \frac{2^{b+1} - 1}{b+1} \rfloor$ and at most $\lceil \frac{2^{b+1} - 1}{b+1} \rceil$ codewords.

Next, let us consider the rest of the codewords. Obviously, any of them can be placed into the $(b + 1)^{\text{th}}$ bucket, because they all have 1 bit at position $b + 1$. The codeword which

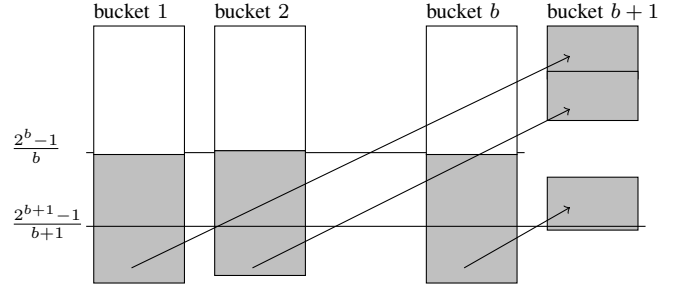


Fig. 2. Example of the construction in the proof of Lemma 2.

has 1 at the $(b + 1)^{\text{th}}$ position and 0 in the rest positions (i.e. $100 \dots 0$) must be placed into the $(b + 1)^{\text{th}}$ bucket. The remaining $2^b - 1$ codewords can be distributed by the first b bits almost uniformly into the b buckets. In such a way, each bucket has at least $2 \cdot \lfloor \frac{2^b - 1}{b} \rfloor$ codewords, which is at least $\lceil \frac{2^{b+1} - 1}{b+1} \rceil$ according to Lemma 1. Some of the newly added codewords must be moved to the $(b + 1)^{\text{th}}$ bucket, until every bucket has at least $\lfloor \frac{2^{b+1} - 1}{b+1} \rfloor$ and at most $\lceil \frac{2^{b+1} - 1}{b+1} \rceil$. Such an action is always possible. This is argued as follows: first, codewords are moved from each of the first b buckets to the $(b + 1)^{\text{th}}$ bucket so that every bucket among the first b has $\lceil \frac{2^{b+1} - 1}{b+1} \rceil$ elements. In case the $(b + 1)^{\text{th}}$ bucket has less than $\lfloor \frac{2^{b+1} - 1}{b+1} \rfloor$ codewords, one more codeword from each bucket is further moved to the $(b + 1)^{\text{th}}$ bucket until it has $\lfloor \frac{2^{b+1} - 1}{b+1} \rfloor$ codewords. Such a process will not get stuck at a position in which one bucket has less than $\lfloor \frac{2^{b+1} - 1}{b+1} \rfloor$ codewords while all the others have this number, because the total number of nonzero codewords is $2^{b+1} - 1$. In such a way, every bucket has at least $\lfloor \frac{2^{b+1} - 1}{b+1} \rfloor$ and at most $\lceil \frac{2^{b+1} - 1}{b+1} \rceil$ codewords. Thus, we proved Lemma 2. ■

Theorem 1: Let $G(V, E)$ be a $2 \cdot \lceil \log_2(|E| + 1) \rceil$ connected graph. $G(E, V)$ can be optimally covered with $\lceil \log_2(|E| + 1) \rceil$ bm-trails to achieve single-link UFL.

Proof: Let $b = \lceil \log_2(|E| + 1) \rceil$. Clearly at least b bm-trails are required for UFL in a graph with E links. In the following we will show that b is also the upper bound. Our goal for the proof of the theorem is to come up with a construction that achieves the theoretical lower bound, and then we will prove the correctness of the construction.

1) *The proposed construction:* Recall that the goal of the S-LP formation process is to assign a binary alarm code to each link so that t_l is a connected subgraph. This can be ensured if each t_l has a spanning tree as a subgraph. Since every $2k$ -edge-connected graph has k edge disjoint spanning trees [28], [29], the construction can achieve the desired lower bound if the graph is $2 \cdot b$ connected, which is sufficient to yield $b = \lceil \log_2(|E| + 1) \rceil$ edge disjoint spanning trees. Let S_i denote the i^{th} spanning tree, where $i = 1, \dots, b$, and the spanning trees are all disjoint (i.e. $S_i \cap S_j \equiv \emptyset$ if $i \neq j$).

According to Lemma 2, the $2^b - 1$ nonzero codewords of b bits in length can be grouped into b buckets of size at least $\lfloor \frac{2^b - 1}{b} \rfloor$, where the i^{th} bucket has alarm codes where the i^{th} bit is 1. Our construction simply assigns the codes of the i -th bucket to the i -th spanning tree S_i , while the remaining edges which are not in the $1^{\text{st}}, \dots, b^{\text{th}}$ spanning trees, namely

$E \setminus \{S_1 \cup S_2 \cup \dots \cup S_b\}$, will be assigned the left and unused codes arbitrarily. This finishes the construction.

2) *Correctness of the constructed bm-trail solution*: Since t_i contains S_i , each bm-trail must be connected and spans the whole graph G . Besides, each link has a unique alarm code because nonzero unique codewords were assigned to the links of the graph. To conclude the proof we need to show that each bucket has at least $|V| - 1$ codewords. By observing the equation $b \cdot (|V| - 1) \leq |E| \leq 2^b - 1$, we see that each bucket has at least $|V| - 1 \leq \lfloor \frac{2^b - 1}{b} \rfloor$ elements. Thus, we proved Theorem 1. ■

The theorem is applicable to complete graphs with at least 18 nodes because they have $\frac{18 \cdot 17}{2} = 153$ links that can be uniquely coded in 8 bits. In this case the graph is at least 16 connected.

IV. ESSENTIALLY OPTIMAL BM-TRAIL SOLUTION IN GENERAL GRID TOPOLOGIES

This section considers general 2D grids denoted by $S_{m,n}$, where m and n corresponds to the number of links in the vertical and horizontal direction, respectively. Harvey, *et al.* [17] provided an $3 \lceil \log_2 |E| \rceil$ upper bound on the number of bm-trails according to Eq. (1) in the case of $m = n$.

In this section, we generalize the study of [17] and investigate the scenario of 2D grid graphs with arbitrary m and n . We give a novel polynomial-time deterministic construction that requires no more than $3 + \lceil \log_2 (|E| + 1) \rceil$ bm-trails. We first solve the bm-trail allocation problem for a special case of $S_{m,n}$ with either $n = 1$ or $m = 1$ (called as *chocolate bar graphs*); and then a solution for general 2D grid topologies is developed based on the chocolate bar solution.

A. Solution for Chocolate Bar Graphs

A general chocolate bar graph is denoted as $C_n(E, V)$, which has $|V| = 2n + 2$ vertices denoted as $x_{1,0}, \dots, x_{1,n}$ (the lower points), and $x_{0,1}, \dots, x_{0,n}$ (the upper points). Fig. 3(a) shows an example of a chocolate bar with $n = 6$. For link set E , we have *lower horizontal links* $(x_{1,i}, x_{1,i+1}) \in E$, *upper horizontal links* $(x_{0,i}, x_{0,i+1}) \in E$ for $i = 0, \dots, n-1$, and the *middle vertical links* $(x_{0,i}, x_{1,i}) \in E$ whenever $i = 0, \dots, n$.

Theorem 2: For a chocolate bar graph $C_n(E, V)$ $b = \lceil \log_2(n+1) \rceil + 2$ bm-trails achieve single-link UFL for $b > 2$, which is at most $\lceil 0.42 + \log_2 (|E| + 2) \rceil$ bm-trails.

Proof: The proof is developed by way of a polynomial-time deterministic construction composed of two steps. We will first introduce the construction, and then explain in detail how the construction can achieve the desired bound on the number of bm-trails.

1) Alarm code assignment for the chocolate bar graph:

Let us assign binary alarm codes to the links of C_n in the following way (see also Fig. 3). We first generate n bitvectors $\mathbf{r}^1, \mathbf{r}^2, \dots, \mathbf{r}^n$ of length \mathfrak{B} , where \mathbf{r}^{i+1} is assigned to a lower horizontal link $(x_{1,i}, x_{1,i+1}) \in E$, where $i = 0, \dots, n-1$. The generation of these codes is provided in Lemma 3. On the other hand, to the higher horizontal link $(x_{0,i}, x_{0,i+1}) \in E$ we assign the bitwise complement of \mathbf{r}^{i+1} , denoted by $\bar{\mathbf{r}}^{i+1} = \mathbf{r}^{i+1} \oplus \mathbf{1}$ where \oplus stands for the bitwise modulo 2

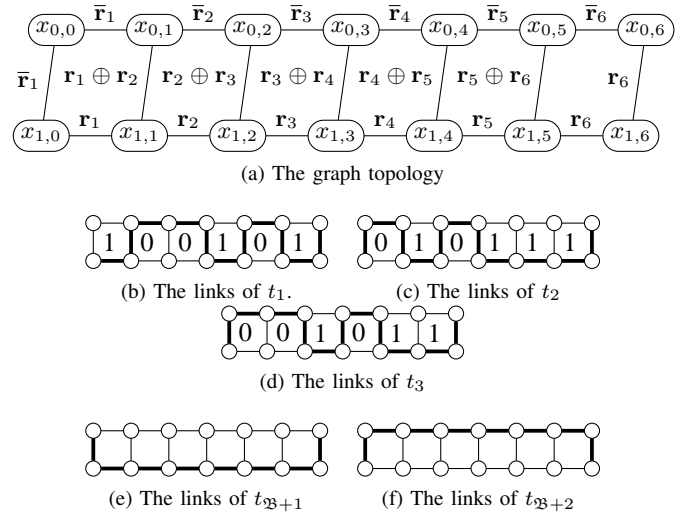


Fig. 3. An example of a chocolate bar graph and the corresponding optimal solution for bm-trails. The bit of each bit position is drawn in each 1×1 rectangular. The $\mathbf{r}^1, \mathbf{r}^2, \dots, \mathbf{r}^n$ codes assigned to the links are listed in the Table II

TABLE II
THE NONZERO ELEMENTS OF \mathbb{F}_8 AS BINARY POLYNOMIALS MODULO $1 + x + x^3$.

Exponential	Polynomial	Code
α^0	1	$\mathbf{r}^1 = 100$
α^1	x	$\mathbf{r}^2 = 010$
α^2	x^2	$\mathbf{r}^3 = 001$
α^3	$x^3 = 1 + x \pmod{1 + x + x^3}$	$\mathbf{r}^4 = 110$
α^4	$x + x^2$	$\mathbf{r}^5 = 011$
α^5	$x \cdot (x + x^2) = 1 + x + x^2 \pmod{1 + x + x^3}$	$\mathbf{r}^6 = 111$
α^6	$x \cdot (1 + x + x^2) = 1 + x^2 \pmod{1 + x + x^3}$	$\mathbf{r}^7 = 101$

addition (*XOR*) and $\mathbf{1}$ is the all 1 vector of length \mathfrak{B} . Also to the middle vertical link $(x_{1,i}, x_{0,i})$ we assign the bitvector $\mathbf{r}^i \oplus \mathbf{r}^{i+1}$ for $i = 1, \dots, n-1$. Finally to the link $(x_{1,0}, x_{0,0})$ bitvector $\mathbf{r}^1 \oplus \mathbf{1}$ is assigned, and to the link $(x_{1,n}, x_{0,n})$ we attach \mathbf{r}^n .

In choosing the list of bitvectors \mathbf{r}^i , for $i = 1, \dots, n-1$, we make the following three assumptions:

- (A1) The vectors \mathbf{r}^i are pairwise different for $i = 1, \dots, n$.
- (A2) The vectors $\mathbf{r}^i \oplus \mathbf{r}^{i+1}$ are all nonzero and pairwise different for $i = 1, \dots, n-1$.
- (A3) The first bits of the vectors \mathbf{r}^1 and \mathbf{r}^n are the same.

The following statement provides an approach to construct $n \leq 2^{\mathfrak{B}} - 1$ bitvectors \mathbf{r}^i which satisfy the requirements (A1), (A2), (A3).

Lemma 3: Let $\mathfrak{B} := \lceil \log_2(n+1) \rceil$ and $\mathfrak{B} > 2$. Then a series of $n \leq 2^{\mathfrak{B}} - 1$ nonzero codes $\mathbf{r}^1, \mathbf{r}^2, \dots, \mathbf{r}^n$ can be generated in polynomial time to satisfy properties (A1), (A2) and (A3).

Proof: With $\mathfrak{B} := \lceil \log_2(n+1) \rceil$, $q = 2^{\mathfrak{B}}$ is the smallest power of 2 which is greater than n . Following the widely used technique in classical error correcting codes, our code vectors will be vectors from a linear space over the two element field \mathbb{F}_2 . We shall consider the finite (Galois) field \mathbb{F}_q with q elements. See Appendix A for a short overview of such finite fields.

According to Theorem 2.5 in [30], \mathbb{F}_q always exists and

it forms a vector space of dimension \mathfrak{B} over its subfield \mathbb{F}_2 . This way we can identify \mathbb{F}_q with bit vectors of length \mathfrak{B} , where the all zero vector corresponds to the 0 element of \mathbb{F}_q . In particular, nonzero vectors correspond to the nonzero elements of the field. Also, according to Theorem 2.8 in [30], \mathbb{F}_q contains a primitive element α , which is a nonzero element such that all the powers $\alpha = \alpha^1, \alpha^2, \dots, \alpha^{q-1}$ are pairwise different. See also Table II where the elements and the related codes are listed for $q = 8$ ($\mathfrak{B} = 3$).

Finding a primitive element in \mathbb{F}_q can be done in polynomial time with exhaustive search, because any nonzero element α can be verified for being a primitive element by raising to a power and checking if the power equals to 1 with an exponent less than $q - 1$.

We now set \mathbf{r}^i to be the (bit vector of the) element α^{i-1} . Condition (A1) is satisfied as $n \leq 2^{\mathfrak{B}} - 1$.

Suppose now that (A2) fails. Then there must exist $0 \leq i < j < n - 1$ such that $\alpha^i \oplus \alpha^{i+1} = \alpha^j \oplus \alpha^{j+1}$ holds in \mathbb{F}_q . But then we have $\alpha^i(1 \oplus \alpha) = \alpha^j(1 \oplus \alpha)$ which (using that $\mathfrak{B} > 1$ and hence that $1 \oplus \alpha$ is not 0) would imply that $\alpha^i = \alpha^j$, contradicting to the fact that α is a primitive element.

To establish (A3), we note that (assuming $\mathfrak{B} > 2$) α and α^n span a subspace of dimension at most 2 of \mathbb{F}_q over \mathbb{F}_2 , hence we can select the basis of \mathbb{F}_q so that both element have 0 coordinates with respect to the first basis vector. ■

2) *The proposed construction for chocolate bar graphs:* In the chocolate bar construction, t_j is actually a simple path in C_n from $x_{1,0}$ to $x_{0,n}$. In the rest of the paper C_n can also be denoted as $C_{x_{1,0}, x_{0,n}}$. As a result, \mathfrak{B} bm-trails from $x_{1,0}$ to $x_{0,n}$ are formed in C_n , each corresponding to one bit position of the vectors. An example is given with $n = 5$, where the resultant 5 um-trails by the construction are shown in Fig. 3(b), 3(c), 3(d).

In addition to the above mentioned bm-trails, we need to add two more bm-trails. This is exemplified in Fig. 3(e) and 3(f). Let the two bm-trails correspond to $t_{\mathfrak{B}+1}$ and $t_{\mathfrak{B}+2}$, respectively, where $t_{\mathfrak{B}+1}$ is composed of the links $(x_{1,0}, x_{0,0}), (x_{1,n}, x_{0,n})$ and the path consisting of all the links $(x_{1,i}, x_{1,i+1})$ $i = 0, \dots, n-1$, while $t_{\mathfrak{B}+2}$ is composed of the links $(x_{1,0}, x_{0,0}), (x_{1,n}, x_{0,n})$ along with the path consisting of all the links $(x_{0,i}, x_{0,i+1})$, $i = 0, \dots, n-1$. As a result, $t_{\mathfrak{B}+1}$ and $t_{\mathfrak{B}+2}$ can identify whether a failed link was a horizontal or vertical link, and whether the link was $(x_{1,0}, x_{0,0})$ or $(x_{1,n}, x_{0,n})$.

Corollary 1: Each t_j $j = 1, \dots, \mathfrak{B} + 2$ forms a single bm-trail, and every bm-trail is a simple path.

The corollary clearly holds according to the construction.

3) *Correctness of the constructed solution:* We will show in the following paragraphs that the set of bm-trails $t_1, \dots, t_{\mathfrak{B}+2}$ are able to localize any single link failure in chocolate bar C_n . Obviously, t_1 , $t_{\mathfrak{B}+1}$ and $t_{\mathfrak{B}+2}$ can unambiguously localize any failed link among $(x_{1,0}, x_{0,0})$ and $(x_{1,n}, x_{0,n})$ because the faulty link can be one of $(x_{1,0}, x_{0,0})$ or $(x_{1,n}, x_{0,n})$ if and only if both $t_{\mathfrak{B}+1}$ and $t_{\mathfrak{B}+2}$ are faulty. If both $t_{\mathfrak{B}+1}$ and $t_{\mathfrak{B}+2}$ alarm (i.e., report failure), the status of t_1 can be used to determine which of the two links $(x_{1,0}, x_{0,0})$ or $(x_{1,n}, x_{0,n})$ is at fault according to (A3).

For the other links, the statuses of $t_{\mathfrak{B}+1}$ and $t_{\mathfrak{B}+2}$ can be

used to determine whether the faulty link is in the group of lower links, the group of upper links, or the group of middle links. With (A1), the links in the first two groups are pairwise different, while with (A2) it implies that the codes in the group of middle links are pairwise different. Therefore, all the links in each of the 3 groups are distinguishable such that unambiguous failure localization is possible within each group, and hence in C_n .

4) *The number of bm-trails in the construction:* Since the chocolate bar graph has $3n + 1$ links, we have

$$\mathfrak{B} = \lceil \log_2 \left(\frac{|E| - 1}{3} + 1 \right) \rceil < \lceil -1.58 + \log_2(|E| + 2) \rceil.$$

As a result the construction requires at most $b = \mathfrak{B} + 2 = \lceil 0.42 + \log_2(|E| + 2) \rceil$ bm-trails. ■

B. 2D Rectangular Grids

In this section, the construction for the chocolate bar graphs is generalized for 2D rectangular grids. A $(m+1)$ -by- $(n+1)$ grid graph is denoted as $S_{m,n}$, whose vertices are denoted as $x_{i,j}$ for $0 \leq i \leq m$ and $0 \leq j \leq n$. The vertical links of $S_{m,n}$ are $(x_{i,j}, x_{i+1,j})$ for $0 \leq i < m$ and $0 \leq j \leq n$. Analogously, the horizontal links of $S_{m,n}$ are $(x_{i,j}, x_{i,j+1})$ for $0 \leq i \leq m$ and $0 \leq j < n$.

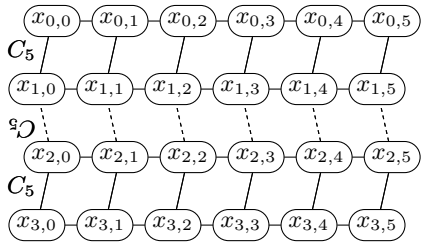
Theorem 3: A 2D rectangular grid graph $S_{m,n}(E, V)$ can be covered with $3 + \lceil \log_2(|E| + 1) \rceil$ bm-trails to achieve UFL, for $m, n \geq 1$.

Proof: We shall have two monitoring sets of bm-trails. The first set has size $b_1 = \lceil \log_2(m+1) \rceil + 2$, while the second has size $b_2 = \lceil \log_2(n+1) \rceil + 2$. Informally speaking, the first set gives the horizontal position of a failed link, while the other gives the vertical coordinate. This will be sufficient to locate the failed link unambiguously. In total, we shall have no more than $\mathcal{B} = b_1 + b_2$ monitoring bm-trails.

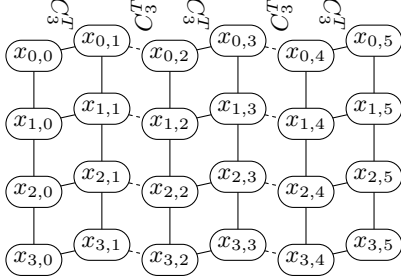
1) *The proposed construction:* We are going to extend the bm-trails t_i ($i = 1, \dots, b_1$) from the chocolate bar graph C_n to the whole square grid $S_{m,n}$. We do it step by step as follows: first we reflect the bm-trail t_i with respect to the line connecting $x_{1,0}$ to $x_{1,n}$, such that t_i is extended to the second chocolate bar defined by the vertices $x_{1,j}$ and $x_{2,j}$, for $j = 0, \dots, n$. The second chocolate bar is extended analogously by reflection to the third chocolate bar, defined by $x_{2,j}$ and $x_{3,j}$, and so on. This reflection process is repeated until the whole $S_{m,n}$ is covered, where the 2D rectangular grid is treated as a series of chocolate bar graphs of C_n . As shown in Fig. 4(a), at every second line the chocolate bar graph is upside down, and the i -th chocolate bar graph C_n consists of vertices $x_{i,0}, \dots, x_{i,n}$ and $x_{i+1,0}, \dots, x_{i+1,n}$, where $i = 0, \dots, m-1$. By applying the reflection process for all the bm-trails t_i ($i = 1, \dots, b_1$), we will obtain b_1 bm-trails.

It is clear that the result of the reflection process must be a connected subgraph without fragmentation, thereby its eligibility as a bm-trail is ensured.

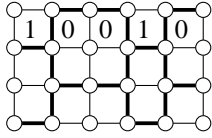
With the whole situation transposed, exactly the same method is applied to specify the vertical position i of the faulty link e . For the remaining b_2 bm-trails of the rectangular grid $S_{m,n}$ we start out with the vertically placed chocolate bar C_m^T



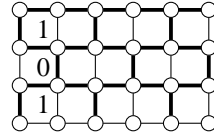
(a) $S_{3,5}$ decomposed into chocolate bars graphs in horizontal way



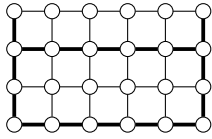
(b) $S_{3,5}$ decomposed into chocolate bars graphs in vertical way



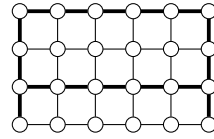
(c) An example of t_i for $i = 1, \dots, b_1 - 2$



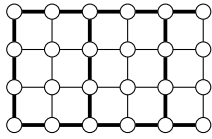
(d) An example of t_i for $i = b_1 + 1, \dots, \mathcal{B} - 2$



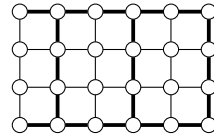
(e) The links of t_{b_1-1}



(f) The links of t_{b_1}



(g) The links of $t_{\mathcal{B}-1}$



(h) The links of $t_{\mathcal{B}}$

Fig. 4. An example of a 2D lattice graph of size 3×5 .

at the left end of the grid (see Fig. 4(b)) and extend the b_2 bm-trails of this C_m^T to the whole grid with the mirror-reflection procedure employed before, nonetheless from left to right in order to extend the bm-trails to all the vertical chocolate bars in the grid. By doing this b_2 bm-trails can be obtained.

With the b_1 and b_2 bm-trails, we complete the construction.

2) *Correctness of the constructed bm-trail solution:* In case of a single failure t_{b_1-1} , t_{b_1} , $t_{\mathcal{B}-1}$, and $t_{\mathcal{B}}$ (see also Fig. 4(e), 4(f), 4(g), and 4(h)) can identify whether a horizontal or a vertical link has failed and if the link is on the left or right border of the rectangular grid (it is on the first or last row/column). Since the failed link belongs to at least one of the horizontal chocolate bar graph C_n , the corresponding $b_1 - 2$ bm-trails can identify the column of the failed link. Similarly, the failed link belongs to at least one of the vertical chocolate bar graph C_m^T , the corresponding $b_2 - 2$ bm-trails can identify

the row of the failed link. As a result, it is known if the link is horizontal or vertical, and its column and row thus can be localized.

3) *The number of bm-trails in the construction:* Bm-trails t_{b_1-1} , t_{b_1} , $t_{\mathcal{B}-1}$, and $t_{\mathcal{B}}$ are only used to decide if the link is horizontal or vertical and if the link is on the boundary of the grid, which indeed can be done with only two bm-trails instead of four, namely $t_{b_1-1} \cup t_{b_1}$ and $t_{\mathcal{B}-1} \cup t_{\mathcal{B}}$. Since $S_{m,n}$ has totally $|E| = 2 \cdot m \cdot n + n + m$ links, the number of bm-trails is:

$$\begin{aligned} \mathcal{B} &= \lceil \log_2(m+1) \rceil + \lceil \log_2(n+1) \rceil + 2 \leq \\ &= \lceil 1 + \log_2(m+1) + \log_2(n+1) \rceil + 2 = \\ &= \lceil \log_2 2 + \log_2(m+1) + \log_2(n+1) \rceil + 2 = \\ &= \lceil \log_2 2 \cdot (m+1) \cdot (n+1) \rceil + 2 = \\ &= \lceil \log_2 2mn + 2n + 2m + 2 \rceil + 2 = \\ &= 2 + \lceil \log_2(2|E| - 2mn + 2) \rceil < \\ &= 2 + \lceil \log_2(2|E| + 2) \rceil = 3 + \lceil \log_2(|E| + 1) \rceil \quad (3) \end{aligned}$$

for $m, n \geq 1$. Note that the first inequality holds because of the general inequality $\lceil A \rceil + \lceil B \rceil \leq \lceil A + B \rceil + 1$, while the second follows from $m \cdot n > 0$. ■

More generally, a similar construction can be used to cover a cubic graph of any dimension for single link UFL with $O(1) + \log_2 |E|$ bm-trails. In this case the alarm code is divided into three parts, and each of them corresponds to a chocolate bar graph.

V. SIMULATION

We have seen the convincing performance of a number of previously reported heuristic approaches on general topologies, and these heuristics can efficiently solve the m-trail allocation problem in very densely meshed and square 2D grid networks. However, in our experiments those heuristics had difficulties to solve the problem in sparse topologies with relatively large diameters. To demonstrate the usefulness and uniqueness of the proposed constructions, we conduct simulation on chocolate bar graphs (or N-by-1 grids) to examine a number of previously reported heuristic approaches listed below, and we will show that the heuristics yield very awkward performance in the considered network settings even when granted with several hours of computation time, while the proposed construction can optimally solve the problem in milliseconds.

- 1) *Random Code Assignment - Random Code Swapping (RCA-RCS):* a heuristic by our previous study in [1].
- 2) *Monitoring Trail Allocation (MTA):* a heuristic by [31], which is a deterministic approach that builds the um-trails in parallel for achieving UFL.
- 3) *Random Next Hop policy (RNH):* a heuristic by [32] which is a randomized version of the MTA heuristic².
- 4) *Cycle Accumulation (CA):* a generic approach by employing Dijkstra's algorithm to distinguish each pair of links [16].

²We thank the authors for sharing the source codes.

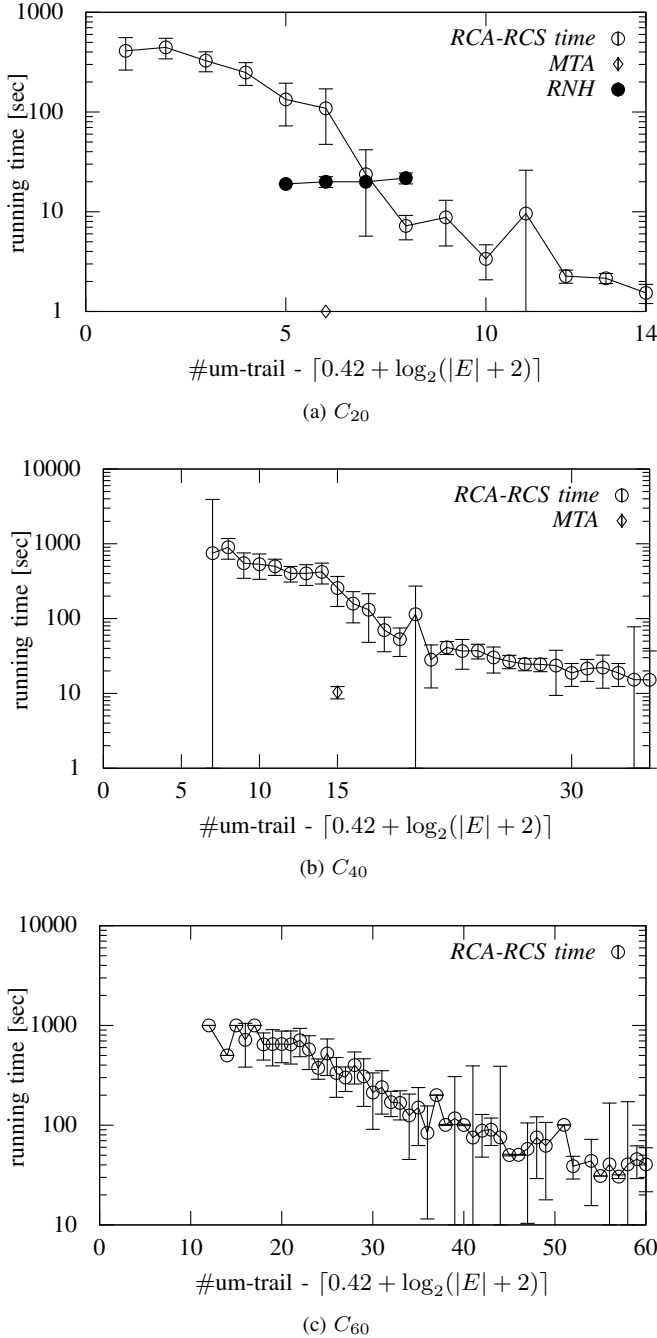


Fig. 5. Simulation results on C_{20} , C_{40} , and C_{60} for the number of bm-trails.

We consider the three schemes: RCA-RCS, MTA, and RNH in chocolate bar graphs C_{20} , C_{40} and C_{60} . A high-performance server with 3GHz Intel Xeon CPU 5160 was used in the simulation. The result of the proposed 2D grid construction is calculated first using the theoretical optimum given in Section IV-A2: $\lceil 0.42 + \log_2(|E| + 2) \rceil$, which yields the minimum number of bm-trails for UFL as 7, 7, and 8 for C_{20} , C_{40} and C_{60} with $|E| = 61$, 121, and 191, respectively. We are interested in the difference between the result by each considered heuristic and the one obtained via the proposed construction. It is important to note that RCA-RCS and RNH are both randomized approaches where longer computation time guarantees better performance (or smaller

numbers of bm-trails). Therefore, we are further interested to see how much long the two schemes can converge close to the optimal solution, where the computation time describes the efficiency (and inefficiency) of the two schemes in the considered scenarios.

Fig. 5 demonstrates the comparison results. Clearly, both RCA-RCS and RNH show better solution quality by granting longer running time, while MTA is a deterministic algorithm that iteratively finds the longest segment as the next um-trail. Since all the three topologies are very sparse whose diameters are much longer than the average nodal degree, MTA needs 6 and 15 more bm-trails in C_{20} and C_{40} than the optimum (i.e., 7), respectively, as shown in 5(a) and 5(b). On the other hand, both RCA-RCS and RNH are seen to converge very slowly as the network has a larger diameter. As shown in Fig. 5(a), RCA-RCS needs to take over 400 seconds to achieve one more bm-trail away from the optimal in C_{20} , but over 800 and 1,000 seconds to achieve about 10 bm-trails from the optimal in C_{40} and C_{60} , respectively, as shown in 5(b) and 5(c).

Note that CA requires 93 bm-trails which are not shown in the figures since it is out of the range. We do not show RNH in 5(b) and 5(c), and MTA in 5(c), because they were not solvable in the topologies C_{20} and C_{40} by running out of 2GB memory (which is the computation specification for the simulation). This is mainly because both methods are designed for networks when the average nodal degree is not much smaller than the diameter of the network. Therefore, the two schemes had a large amount of candidate segments which drained the memory usage.

In summary, the simulation results clearly show the merit of the proposed 2D grid construction compared with the previously reported heuristics.

VI. CONCLUSIONS

The paper considered the optimal S-LP allocation problem in all-optical mesh networks of very densely connected and grid topologies for achieving unambiguous failure localization (UFL) under any single link failure. We have derived essentially tight upper bounds on the number of bm-trails for the topologies of interest via deterministic constructions and rigorous proofs. The bounds obtained are $\lceil \log(|E| + 1) \rceil$ and $3 + \lceil \log_2(|E| + 1) \rceil$ for $2 \cdot \lceil \log_2(|E| + 1) \rceil$ connected and grid topologies, respectively, which are close to the theoretical lower bounds $\lceil \log(|E| + 1) \rceil$. We demonstrated via case studies in chocolate bar graphs that the derived bound can hardly be reached by some recently reported random algorithm based heuristics.

REFERENCES

- [1] J. Tapolcai, P.-H. Ho, B. Wu, and L. Rónyai, "A novel approach for failure localization in all-optical mesh networks," *IEEE/ACM Transactions on Networking (TON)*, vol. 19, pp. 275–285, 2011.
- [2] B. Wu, P.-H. Ho, and K. Yeung, "Monitoring trail: On fast link failure localization in all-optical wdm mesh networks," *IEEE/OSA Journal of Lightwave Technology*, vol. 27, no. 18, pp. 4175–4185, 2009.
- [3] C. Mas, I. Tomkos, and O. Tonguz, "Failure location algorithm for transparent optical networks," *IEEE Journal on Selected Areas in Communications (JSAC)*, vol. 23, no. 8, pp. 1508–1519, 2005.

- [4] H. Zeng and C. Huang, "Fault detection and path performance monitoring in meshed all-optical networks," in *IEEE GLOBECOM*, vol. 3, 2004, pp. 2014–2018.
- [5] 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 Communications*, vol. 11, no. 3, pp. 277–286, 2006.
- [6] H. Zeng and A. Vukovic, "The variant cycle-cover problem in fault detection and localization for mesh all-optical networks," *Photonic Network Communications*, vol. 14, no. 2, pp. 111–122, 2007.
- [7] B. Wu, K. Yeung, B. Hu, and P. H. Ho, "M²-cycle: an optical layer algorithm for fast link failure detection in all-optical mesh networks," *Computer Networks*, vol. 55, no. 3, pp. 748–758, February 2011.
- [8] B. Wu and K. Yeung, "Monitoring cycle design for fast link failure detection in all-optical networks," in *IEEE GLOBECOM*, 2007, pp. 2315–2319.
- [9] C. Li, R. Ramaswami, I. Center, and Y. Heights, "Automatic fault detection, isolation, and recovery in transparent all-optical networks," *IEEE/OSA Journal of Lightwave Technology*, vol. 15, no. 10, pp. 1784–1793, 1997.
- [10] S. Stanic, S. Subramaniam, H. Choi, G. Sahin, and H. Choi, "On monitoring transparent optical networks," in *International Conference on Parallel Processing Workshops (ICPPW)*, 2002, pp. 217–223.
- [11] Y. Wen, V. Chan, and L. Zheng, "Efficient fault-diagnosis algorithms for all-optical wdm networks with probabilistic link failures," *IEEE/OSA Journal of Lightwave Technology*, vol. 23, pp. 3358–3371, 2005.
- [12] C. Assi, Y. Ye, A. Shami, S. Dixit, and M. Ali, "A hybrid distributed fault-management protocol for combating single-fiber failures in mesh based dwdm optical networks," in *IEEE GLOBECOM*, 2002, pp. 2676–2680.
- [13] B. Wu, P.-H. Ho, and K. Yeung, "Monitoring trail: On fast link failure localization in wdm mesh networks," *IEEE/OSA Journal of Lightwave Technology*, vol. 27, no. 23, 2009.
- [14] M. Maeda, "Management and control of transparent optical networks," *IEEE Journal on Selected Areas in Communications (JSAC)*, vol. 16, no. 7, pp. 1008–1023, 2002.
- [15] B. Wu, K. Yeung, and P.-H. Ho, "Monitoring cycle design for fast link failure localization in all-optical networks," *IEEE/OSA Journal of Lightwave Technology*, vol. 27, no. 10, pp. 1392–1401, 2009.
- [16] 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 (TON)*, vol. 17, no. 4, pp. 1080–1093, 2009.
- [17] 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 *IEEE INFOCOM*, 2007, pp. 697–705.
- [18] E. Mohan Reddy and K. Reddy, "A dynamically reconfigurable wdm lan based on reconfigurable circulant graph," in *IEEE Military Communications Conference (MILCOM)*, vol. 3, oct. 1996, pp. 786–790 vol.3.
- [19] G. Weichenberg, V. Chan, and M. Medard, "High-reliability topological architectures for networks under stress," *IEEE Journal on Selected Areas in Communications (JSAC)*, vol. 22, no. 9, pp. 1830–1845, nov. 2004.
- [20] D. Wang, A. Kumar, M. Sivakumar, and J. McNair, "A fault-tolerant backbone network architecture targeting time-critical communication for avionic wdm lans," in *IEEE International Conference on Communications (ICC)*, jun. 2009, pp. 1–5.
- [21] P. Saengudomlert, E. Modiano, and R. Gallager, "On-line routing and wavelength assignment for dynamic traffic in wdm ring and torus networks," *IEEE/ACM Transactions on Networking (TON)*, vol. 14, no. 2, pp. 330–340, 2006.
- [22] F. K. Hwang and V. T. Sós, "Non-adaptive hypergeometric group testing," *Studia Sci. Math. Hungar.*, vol. 22, pp. 257–263, 1987.
- [23] D. Du and F. K. Hwang, *Combinatorial Group Testing and Its Applications*. World Scientific, 2000.
- [24] D. Eppstein, M. Goodrich, and D. Hirschberg, "Improved combinatorial group testing for real-world problem sizes," in *Workshop on Algorithms And Data Structures (WADS)*. Waterloo, ON, Canada: Springer, Aug. 2005, pp. 86–98.
- [25] R. Diestel, *Graph theory*. Springer New York, 2000.
- [26] H. N. Gabow and H. H. Westermann, "Forests, frames, and games: Algorithms for matroid sums and applications," *Algorithmica*, vol. 7, no. 1, pp. 465–497, 1992.
- [27] J. Tapolcai, B. Wu, and P.-H. Ho, "On monitoring and failure localization in mesh all-optical networks," in *IEEE INFOCOM*, Rio de Janeiro, Brasil, 2009, pp. 1008–1016.
- [28] W. Tutte, "On the problem of decomposing a graph into n connected factors," *Journal of the London Mathematical Society*, vol. 1, no. 1, pp. 221–230, 1961.
- [29] C. Nash-Williams, "Edge-disjoint spanning trees of finite graphs," *Journal of the London Mathematical Society*, vol. 1, no. 1, pp. 445–450, 1961.
- [30] R. Lidl and H. Niederreiter, *Introduction to finite fields and their applications*. Cambridge University Press, 1994.
- [31] Y. Zhao and S. Xu, "A new heuristic for monitoring trail allocation in all-optical WDM networks," in *IEEE GLOBECOM*, 2010.
- [32] Y. Zhao, S. Xu, B. Wu, X. Wang, and S. Wang, "Monitoring trail allocation in all-optical networks with the random next hop policy," in *submitted to IEEE ICC*, 2010.

APPENDIX

A. Proof of Lemma 1

We prove inequality

$$2 \cdot \lfloor \frac{2^b - 1}{b} \rfloor \geq \frac{2^{b+1} - 1}{b + 1} \geq \lceil \frac{2^b - 1}{b} \rceil.$$

For the first inequality one can readily check that it holds for $b = 3, 4, 5$. Note also that the inequality fails for $b = 2$. We have

$$\begin{aligned} 2 \cdot \lfloor \frac{2^b - 1}{b} \rfloor - \frac{2^{b+1} - 1}{b + 1} &\geq \\ &\geq 2 \cdot \left(\frac{2^b - 1}{b} - 1 \right) - \frac{2^{b+1} - 1}{b + 1}. \end{aligned} \quad (4)$$

After clearing denominators, the nonnegativity of the above quantity for $b \geq 6$ is equivalent to $2^{b+1} - (2b^2 + 3b + 2) \geq 0$. But for $b \geq 4$ we have $3b + 2 < 3b + b = 4b \leq b^2$, hence it suffices to see that $2^{b+1} - 3b^2 \geq 0$, or $f(b) := \frac{2^{b+1}}{3b^2} \geq 1$.

We have $f(6) = \frac{128}{108} > 1$. Moreover, for every real $b \geq 3$,

$$\frac{f(b+1)}{f(b)} = 2 \left(1 - \frac{1}{b+1} \right)^2 \geq 2 \left(1 - \frac{1}{4} \right)^2 = \frac{18}{16} > 1,$$

It implies that $f(b) > 1$ whenever $b \geq 6$ is an integer. The second inequality holds because $\frac{2^b - 1}{b}$ is a convex increasing function, and at $b = 4$ the difference is $\frac{2^5 - 1}{5} - \frac{2^4 - 1}{4} = 2.45 > 1$. This proves Lemma 1.

B. A Brief Introduction to Galois Fields

In the arithmetic of ordinary numbers there are infinitely many numbers, while the fields \mathbb{F}_{2^b} have only 2^b elements. However, the operations of addition, subtraction, multiplication and division (except division by zero) may be performed in a way that satisfies the familiar rules from the arithmetic of ordinary numbers. Concerning \mathbb{F}_{2^b} a widely accepted approach is to represent the elements as polynomials of degree strictly less than b over \mathbb{F}_2 . Operations are then performed modulo R where R is an irreducible polynomial of degree b over \mathbb{F}_2 . For example the field \mathbb{F}_8 can be interpreted as the polynomials modulo $1 + x + x^3$. This way we can consider \mathbb{F}_8 as the set of binary polynomials of degree at most 2 (indeed there are 8 such polynomials). Addition is the usual binary polynomial addition. For example

$$(1 + x + x^2) + (1 + x^2) = x.$$

Multiplication is the usual polynomial multiplication, followed by reduction if necessary (modulo $1 + x + x^3$). By reduction

we mean replacing x^3 by $x + 1$ as long as it is possible. For example

$$(1 + x + x^2)x^2 = x^2 + x^3 + x^4 = \\ x^2 + (x + 1) + x(x + 1) = x^2 + x + 1 + x^2 + x = 1.$$

In this representation x is a *primitive element*, indeed 7 is the smallest positive integer exponent m for which $x^m = 1$.



János Tapolcai received his M.Sc. ('00 in Technical Informatics), and Ph.D. ('05 in Computer Science) degrees in Technical Informatics from Budapest University of Technology and Economics (BME), Budapest, Hungary. 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, mathematical programming, optical and IP level routing and survivability, availability analysis and distributed computing. He is an author of over 80 scientific publications, and is the recipient of the Best Paper Award in ICC'06, in DRCN'11. He is a winner of Hungarian Academy of Sciences Momentum Program.



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.



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.