This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2007 proceedings.

Non-Adaptive Fault Diagnosis for All-Optical Networks via Combinatorial Group Testing on Graphs

Nicholas J.A. Harvey[2], Mihai Pătraşcu[2], Yonggang Wen[1], Sergey Yekhanin[2] and Vincent W.S. Chan[1]
[1]Laboratory for Information and Decision Systems (LIDS)
[2]Computer Science and Artificial Intelligence Laboratory (CSAIL)
Massachusetts Institute of Technology
Cambridge, MA 02139
{nickh, mip, eewyg, yekhanin, chan}@mit.edu

Abstract—**We consider the problem of detecting failures for all-optical networks, with the objective of keeping the diagnosis cost low. Compared to the passive paradigm based on parity check in SONET, optical probing signals are sent proactively along lightpaths to probe their state of health and failure pattern is identified through the set of test results (i.e., probe syndromes). As an alternative to our previous adaptive approach where all the probes are sent sequentially, we consider in this work a non-adaptive approach where all the probes are sent in parallel. The design objective is to minimize the number of parallel probes, so as to keep network cost low. The non-adaptive fault diagnosis approach motivates a new technical framework that we introduce: combinatorial group testing with graph-based constraints. Using this framework, we develop several new probing schemes to detect network faults for all-optical networks with different topologies. The efficiency of our schemes often depends on the network topology; in many cases we can show that our schemes are optimal in minimizing the number of probes.**

I. INTRODUCTION

Network management [1] is a crucial but expensive component of any network operation whose cost constitutes a significant portion of total network cost. Typical network management activities include configuration, performance, security, account management and fault management. In current networks, fault management [2], specifically failure detection and isolation, contributes a significant amount of network operating effort. Owing to the cost and importance of fault diagnosis, it has been an active research topic in various contexts, such as the Internet [3-5], wireless networks [6, 7], and optical networks [8-10]. In this work, we focus on fault diagnosis for all-optical networks. The unique characteristics of all-optical networks yield not only technical challenges but also cost-reduction opportunity for fault diagnosis, as we explain below.

All-optical networks promise significant cost benefits such that broadband network services can potentially be delivered to large populations at much lower cost than today's technologies [11,12]. The significant cost savings are due to optical switching of high data-rate lightpaths at intermediate network nodes, thereby reducing electronic processing costs. However, all-optical networks are susceptible to various physical failures, e.g., fiber cuts, switch node failures, transmitter/receiver breakdowns, and optical amplifier breakdowns. These failures can result in the disruption of communication, and can be costly to detect and localize within the current management framework. Since all-optical networks lack parity checks at intermediate nodes as SONET does, either optical signal is tapped out at each intermediate node for parity check or new mechanisms are needed to diagnose link failures. If tapping is indeed done, a lot of cost gains of all-optical networks will be mitigated.

Instead of the passive paradigm based on parity check in SONET, we have proposed a proactive fault diagnosis paradigm in [8]: optical probing signals are sent along some lightpaths to test the health of the network, and *probe syndromes* (i.e., results of the probes) are used to differentiate failure patterns. The design of proactive fault diagnosis schemes for all-optical networks bears two key objectives: (i) detecting faults quickly, and (ii) keeping the diagnosis cost low. The importance of objective (i) stems from the current SONET standard [13], in which the 50-ms restoration time leaves little room for fault detection and localization. This will probably be reduced further in future all-optical networks to avoid large amount of data loss during a short period of communication disruption. Hence, when parts of a network are malfunctioning, it is critical to locate and identify these failures as soon as possible. At the same time, the cost of fault diagnosis has to be kept low such that the cost advantage of all-optical networks, compared to traditional optical networks, can materialize.

We believe that the two design objectives could be tightly related to two parameters of proactive fault diagnosis schemes (i.e., the number of probes and the number of probing steps[1]). First, the number of probes could serve as the manifestation of fault management effort. In particular, each probe requires certain amount of effort in both network management/control plane (e.g., signaling) and data plane (e.g., transmission and detection) that otherwise could be used to generate revenue. In addition, each probe results in one bit of management information, whose transportation, storage and processing consumes additional network resources. Second, under the assumption that each step takes approximately equal amount of time, the number of probing steps indicates how fast the fault pattern could be identified. In this research, we exploit two alternative designs for choosing probes (i.e, *adaptive* probing, and *non-adaptive* probing) to balance these two objectives.

In adaptive fault diagnosis schemes [8-10, 14], probing

The research in this paper was supported in part by the Defense Advanced Research Projects Agency (DARPA), by a Natural Sciences and Engineering Research Council of Canada PGS Scholarship, by NSF contract CCF-0515221 and by ONR grant N00014-05-1-0148. Student authors in this paper are listed in an alphabetical order.

---

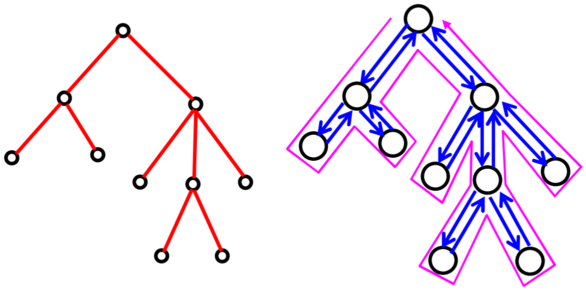
[1] One probing step corresponds to a set of parallel probes.

signals are sequentially sent to probe the health of the network until the failure pattern is identified. Owing to its sequential nature, successive probes can be chosen according to previous probe syndromes, and thus the number of probes required is usually quite small. Indeed, our previous research on adaptive fault diagnosis schemes for all-optical networks with probabilistic node/link failures [8-10] has shown that the average number of probes is lower bounded by the information entropy of network states. Based on information theoretic insights, we have also developed the run-length probing scheme and proved its performance to be within 5% of the entropy lower bound. However, the number of probing steps might be quite large for some network failure patterns and/or in some large networks.

To keep the number of probing steps small, in this paper, we consider an alternative non-adaptive approach [15] to diagnosing failures in all-optical networks. Instead of sending optical probing signals sequentially, a pre-determined set of probing signals are sent in parallel to probe the network state of health. In addition, compared to the probabilistic failure model (i.e., each link fails independently and no upper bound on the number of failures) used in our previous work, we also assume a worst-case failure model in that the number of simultaneous failures is upper bounded by a constant. Under such a framework, the design objective is to minimize the number of parallel probes for non-adaptive fault diagnosis schemes.

Our fault detection methods are based on techniques from the field of *combinatorial group testing (CGT)* [16], where defected samples are identified through a set of parallel testing on different combinations of unknown samples. This field has a wide variety of practical applications, such as HIV screening, DNA testing, MAC design, and much more [17]. It has also been used in network management applications (see, e.g., [18]), but only to a limited degree. We believe that CGT is a powerful tool that can be used in a wide variety of network failure detection contexts, and we hope that our work will inspire its use more widely. The present paper considers only the context of all-optical networks since their unique characteristics lead to a natural application of CGT. In particular, the absence of optical-to-electrical conversions at intermediate nodes of all-optical networks allows us to probe a lightpath of several interconnected links at approximately the same amount of effort of probing one single link.

In this work, we propose a variant of classical CGT in which the valid tests are determined by the structure of a graph. In the all-optical network context, this graph corresponds to the network topology, and the constraint on valid tests is due to the fact that lightpaths can only traverse a set of interconnected edges. To the best of our knowledge, this is a novel framework for CGT[2], and we believe it to deserve further study. We formally analyze the number of tests needed for certain interesting classes of graphs, and even arbitrary graphs (with performance depending on the topology). In some cases, we can give matching upper- and lower-bounds on the number of tests needed. Our fault diagnosis schemes have a common theme, which suggests a practical rule-of-thumb for efficient fault diagnosis schemes: a fault-free sub-graph in the network topology should be identified, and used as a "hub" to diagnose other failures in the network.



(a) (b)

Fig. 1. A walk over undirected graph can be implemented with a lightpath in a practical all-optical network.

The remainder of this paper is organized as follows. In Section II, we formulate the non-adaptive fault diagnosis problem. In Section III, we reinterpret this problem as the combinatorial group testing problem on graphs. In Section IV, we describe algorithms and lower bounds for various classes of important network topologies: linear networks, complete networks, grid networks. In Section V, we consider trees and arbitrary graphs, and obtain efficient algorithms when the diameter is small and/or the graph does not have small cuts. Section VI concludes this paper.

II. NON-ADAPTIVE FAULT DIAGNOSIS SCHEMES FOR ALL-OPTICAL NETWORKS

A. Permanent Link Failure Model

In this paper, all-optical networks are abstracted as undirected graphs. An *undirected graph* G is an ordered pair of sets (V,E), where V is the set of nodes, and E is the set of edges, which are unordered pairs of nodes. The number of nodes is n and the number of edges is m. The terms links and edges are used interchangeably in this paper.

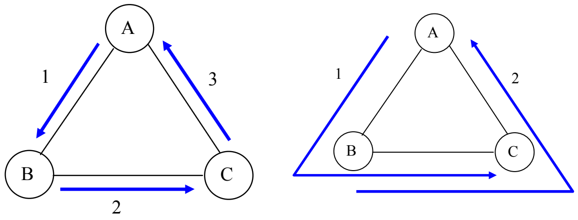
In our model, we assume links fail and nodes do not. Insights from this limited case could facilitate to address fault diagnosis for both node and link failures. In addition, we consider a permanent failure model, i.e., an edge is either *failed* or *intact*, and the failure status does not change over the period of diagnosis. Since it is unlikely that numerous edge failures happen simultaneously, we assume that the number of edges failures is upper bounded by a constant $s(\leq m)$ at any instant. In this paper, we generally allow s to be arbitrary, although the case of $s=1$ is often central.

B. Non-Adaptive Fault Diagnosis Scheme

In this paper, we diagnose network failures by sending

---

[2] There is another notion of group testing on graphs [16, Chapter 12], although it is completely unrelated to the framework that we propose herein.



	AB	BC	CA	Ø
1	**1**	**0**	**0**	**0**
2	**0**	**1**	**0**	**0**
3	**0**	**0**	**1**	**0**

(a)

	AB	BC	CA	Ø
1	**1**	**1**	**0**	**0**
2	**0**	**1**	**1**	**0**

(b)

Fig. 2. Two non-adaptive fault diagnosis algorithms for the 3-node ring network.

optical probing signals along certain lightpaths to determine the network's state. We illustrate this probing model in this sub-section.

A *probe* in the network corresponds to a walk (a sequence of adjacent edges, allowing repetitions) in the corresponding graph. Physically, each probe corresponds to a lightpath in the network. For example, a walk in the graph can constitute a sub-tree in the graph as in Fig. 1(a), which can be translated to a lightpath in practical all-optical networks as in Fig. 1(b). In Fig. 1(a), the network is abstracted as undirected graph, whose nodes correspond to the optical switches and links correspond to the optical fibers. In practical all-optical networks, each link represents two parallel optical fibers transmitting signals in opposite directions. As shown in Fig. 1(b), we can replace each link in Fig. 1(a) by two directed arcs in opposite directions. In this way, each walk can be implemented as a probe by sending a diagnosis signal along the directed lightpath, as illustrated in Fig. 1(b). Moreover, to avoid the potential fiber loop lasing effect [25], a physically feasible probe must satisfy one additional property: each network link is traversed at most once in each direction. We call such a probe a *permissible probe*. The probes generated by our algorithms in Section IV and V are all permissible probes.

When an optical signal is sent along a given lightpath, the signal will arrive at the destination if all edges along the lightpath are intact. Otherwise, if there is at least one failed edge on the lightpath, the signal never reaches the destination (or the quality of the signal is unacceptable). The result of each probe is called the probe *syndrome*, denoted as $r=0$ if the probing signal arrives successfully; and $r=1$ otherwise.

A *non-adaptive fault diagnosis scheme* is a method for sending optical signals (i.e., probes) along a set of pre-determined lightpaths in the network such that up to s edge failures can be identified by examining the set of probe syndromes. For example, as shown in Fig. 2, both sets of probes can identify any single edge failure. Although both schemes can identify any single edge failure, one would prefer Scheme (b) to Scheme (a) since Scheme (b) uses less probes than Scheme (a). Indeed, to keep the fault diagnosis low, we would like to develop efficient non-adaptive fault diagnosis schemes using the minimum number of probes.

	0	*1*	*2*	*3*	*4*	*5*	*6*	*7*
1	**0**	**0**	**0**	**0**	**1**	**1**	**1**	**1**
2	**0**	**0**	**1**	**1**	**0**	**0**	**1**	**1**
3	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**

Fig. 3. The diagnosis matrix for the logarithmic testing procedure (LTP) with *m* = 7. Columns correspond to elements to be tested, and rows correspond to tests.

III. COMBINATORIAL GROUP TESTING (CGT) ON GRAPHS

In this section, we present theoretical background on combinatorial group testing (CGT) and its connection to the non-adaptive fault diagnosis problem.

The general CGT problem is defined as follows. Consider a set S of m elements, each of which is either intact or failed. The maximum number of failed elements is bounded by s, which is smaller than m. We are allowed to perform group tests of the following form: specify a subset $t \subset S$, run the test on t, and learn if there is at least one failed element in t. Our objective is to discover all faulty elements, while using the smallest possible number of group tests. It has been shown that the non-adaptive combinatorial group testing problem is equivalent to the superimposed code problem [20] in Information Theory.

Let $T^*(m,s)$ denote the minimum number of non-adaptive group tests needed to locate up to s failed elements in a set of size m. It is obvious that $T^*(m,s) \le m$, since we can test each element individually. The total number of failure patterns is $N(m,s) = \sum_{k=0}^{s} \binom{m}{k}$, so the minimum number of probes needed to distinguish between these patterns is at least $\log_2 N(m,s)$. Hence, $\log_2 N(m,s) \le T^*(m,s) \le m$. In particular, if $s=1$, the minimum number of non-adaptive probes needed is bounded as follows:

$$\log_2(m+1) \le T^*(m,1) \le m. \tag{1}$$

For arbitrary s and sufficiently large m, it has been shown that $T^*(m,s)$ can be bounded[3] as,

$$\Omega\left(\frac{s^2}{\log s}\log m\right) \le T^*(m,s) \le O\left(s^2 \log m\right), \tag{2}$$

where the upper bound comes from [20] and is essentially based on a simple random superimposed coding argument, and the lower bound is due to D'yachkov and Rykov [21].

Any non-adaptive combinatorial group testing algorithm with $T(m,s)$ tests can be expressed as a testing matrix C with $T(m,s)$ rows and $N(m,s)$ columns, where each row

---

[3] $f(n)=O(g(n))$ means that there exists a constant c and integer N such that $f(n) \le cg(n)$ for all $n > N$. $f(n)=\Omega(g(n))$ means that $g(n)=O(f(n))$. $f(n)=\Theta(g(n))$ means both $f(n)=O(g(n))$ and $f(n)=\Omega(g(n))$.

corresponds to a group test and each column corresponds to a failure pattern. We set $c_{ij} = 1$ if group test i would fail under failure pattern j; otherwise, $c_{ij} = 0$. As a simple illustration, consider the case of $s = 1$ and $m = 7$; the testing matrix is shown in Fig. 3. In this case, the algorithm performs three group tests. The elements involved in these tests are respectively {4, 5, 6, 7}, {2, 3, 6, 7} and {1, 3, 5, 7}. If element i has failed, the results of the tests are identical to column i, which is the binary representation of i. If no element has failed, all tests return zero. Thus $T(7,1) = 3$, which corresponds to the lower bound of (1).

A similar construction yields an efficient procedure to find a single failed element in any group of m elements. This procedure plays an important role in the fault diagnosis algorithms of Section IV and V. The construction involves a matrix with $\lceil \log(m+1) \rceil$ rows (corresponding to the tests) and m+1 columns (corresponding to the m+1 possible failure patterns). Column 0 corresponds to the scenario in which all elements are intact, and column i ($i = 1, \cdots, m$) corresponds to the scenario in which element i has failed. We set column i of the matrix to be the binary representation of i. Each row corresponds to a group test which tests the subset of objects which have a 1 entry in the row of the diagnosis matrix. It is easy to see that if item i has failed then the outcome of the tests will be precisely the binary representation of i. For convenience, we refer to this procedure as the *logarithmic testing procedure* (LTP).

The non-adaptive network fault diagnosis problem can be formulated as a non-adaptive combinatorial group testing problem, under some additional constraints. In particular, in our formulation of the non-adaptive fault diagnosis problem, there are up to s edge failures among the set of m network edges. A set of permissible probes are sent concurrently to test whether any edge of the corresponding walk has failed. It follows that the non-adaptive fault diagnosis problem is equivalent to a non-adaptive combinatorial group testing problem, under the constraint that the group test can be performed only if it corresponds to a permissible probe. We call this variant of CGT the problem of *combinatorial group testing on graphs*. We address the non-adaptive fault diagnosis problem by proving several results concerning combinatorial group testing on graphs.

IV. EFFICIENT FAULT DIAGNOSIS SCHEMES IN CERTAIN NETWORK CLASSES

In this section, we present efficient non-adaptive fault diagnosis algorithms for certain classes of network topologies, and we characterize the minimum number of non-adaptive probes to identify up to s failed edges in the network topology G. This quantity is denoted $L^*(G,s)$. The algorithms that we present can also be considered algorithms for combinatorial group testing on graphs.

A. Networks with Linear or Ring Topologies

Linear topologies are used mostly for distribution networks in optical networks. Ring topologies are also widely used and are largely similar to linear networks, from a fault diagnosis perspective.

Consider a linear network consisting of n nodes, indexed by integers $0, 1, \cdots, n-1$. The edges are $\{i, i+1\}$ for $0 \le i \le n-2$. For linear networks, we can establish the following result.

Theorem 1: The minimum number of non-adaptive probes to locate up to a single edge failure in a linear network of n nodes, i.e., $L^*(G, s=1)$, is precisely $\lceil n/2 \rceil$.

Proof:

Let t be an arbitrary probe in a linear network. Let a the node with smallest index that is contained in t, and b the node with largest index contained in t. Note that probe t is equivalent to a path from node a to node b. We use the notation $t = [a,b]$ and call $a(b)$ the head (tail) of t.

First we establish the lower bound. Let $T = \{t_1, \cdots, t_l\}$ be a set of probes that can detect a single edge failure. Suppose $2l < n$; then there exists a node i that is neither a head or a tail of any test t_j. Considering the following two cases:

- $i = 0$ or $n-1$: In this case, no probe t_j includes an edge that is adjacent to node i. Therefore, the probe algorithm cannot identify whether the edge adjacent to node i has failed or not.
- $1 \le i \le n-2$: In this case, every test t_j either contains both edge $\{i-1, i\}$ and edge $\{i, i+1\}$, or contains neither. Therefore, the probe algorithm cannot distinguish between the case when edge $\{i-1, i\}$ has failed and the case when edge $\{i, i+1\}$ has failed.

In both cases, we arrive at a contradiction and conclude that $l \ge \lceil n/2 \rceil$ is a necessary condition.

Now, we proceed to the upper bound. Consider the probe test $\{t_j\}$, where $t_j = [j, j + \lfloor n/2 \rfloor]$ for $0 \le j \le \lceil n/2 \rceil - 1$. Clearly, every edge e belongs to some test t_j. Therefore all we need to show is that, for every pair of edges $e_1 \ne e_2$ there is a test t_j that contains exactly one of the edges. This will imply that, given all the probe syndromes, one can locate the faulty edge or decide that no failure has occurred. Let $e_1 = [t_1, h_1]$ and $e_2 = [t_2, h_2]$. Without loss of generality, we assume $h_1 \le t_2$. Consider the following two cases:

- $h_1 \ge \lceil n/2 \rceil$: In this case, the test $[h_1 - \lfloor n/2 \rfloor, h_1]$ contains e_1 but not e_2.
- $h_1 < \lceil n/2 \rceil$: In this case, either the test $[h_1, h_1 + \lfloor n/2 \rfloor]$ or the test $[\lceil n/2 \rceil - 1, n-1]$ contains e_2 but not e_1.

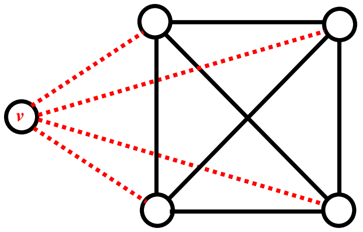


Fig. 4. The complete graph with $n=5$, where node v and its neighborhood are used to route probes.

This completes the proof. Q.E.D.

This $\Theta(n)$ bound for linear networks is much larger than the lower bound of $\log(n)$ in (1). Intuitively, the low connectivity of the linear/ring network topology restricts the possible tests to such an extent that testing becomes inefficient. Note that with a linear lower bound, s becomes irrelevant (we can handle any s with $m=n-1$ probes). It can be shown that the same result can be proved for ring networks by simply cutting the ring network into a line network at any node.

B. Networks with Fully-Connected Topologies

This sub-section deals with the non-adaptive fault diagnosis problem for all-optical networks whose topologies are fully-connected (i.e., complete graphs). For a topology of n nodes, denoted K_n, each node is connected to all other nodes in the network, resulting in $m=n(n-1)/2$ edges in the network. The case $n=5$ is illustrated in Fig. 4. For such a network, we propose the following non-adaptive fault diagnosis algorithm.

Algorithm 1: *Testing for a single failure in complete networks*

Step 1a:

Arbitrarily pick a node v and define its neighborhood sub-graph $B(v)$ as the $n-1$ edges that connect it to all other nodes. As shown in Fig. 5, the neighborhood is a star centered at node v.

Step 1b:

Perform the LTP on the sub-graph $B(v)$. Each LTP test becomes a valid probe due to the star topology.

Step 2:

Perform the LTP on the subgraph obtained by deleting node v. The sub-graph $B(v)$ is used to route the probes as needed.

We now discuss the correctness of Algorithm 1. If the network topology did not impose any constraints on the choice of probes then (that is, if an arbitrary subset of edges formed a permissible probe) then one could directly apply the LTP procedure, using the individual edges as elements to be tested. Unfortunately, the topology restricts our choice of probes to sequences of adjacent edges, so the probes are chosen more carefully. At a high level, the approach is first to identify a fault-free sub-graph, then to use this sub-graph to route the probes for an LTP procedure. Algorithm 1 uses two LTPs, of size $(n-1)$ and size $(n-1)(n-2)/2$ respectively, and therefore the total number of probes required is $\mathrm{O}(\log n)$. Combining this result with the lower bound of (1), we have established our main result for complete networks as follows.

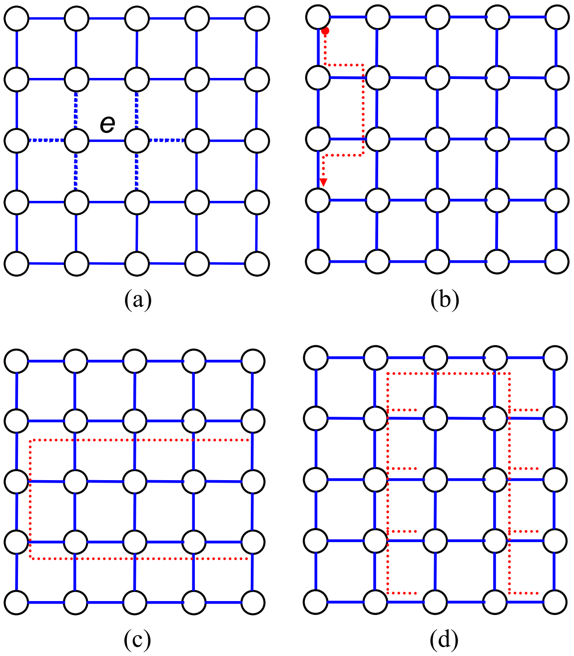


Fig. 5. (a) A 2-D grid with 25 nodes. If at most 7 failures are allowed, then Theorem 4 shows that the failure of edge e cannot be detected efficiently by non-adaptive tests. (b) A single probe to test edge 1 and edge 3 on column 1. (c) A single probe to test column 2 and column 4. (d) Single probes to test the 2nd edge on all rows and the 4th edge on all rows.

Theorem 2: $\in(\log n)$ probes are necessary and sufficient to identify a single edge failure in a fully connected network with n nodes.

C. Networks with 2-D Grid Topologies

The sub-section considers two-dimensional grid networks of size $\sqrt{n}\times\sqrt{n}$. Such structures are also commonly used as interconnection networks [22]; in the context of all-optical networks, they are sometimes called Manhattan networks. Fig. 5 illustrates the case of n = 25. The following algorithm gives an optimal non-adaptive fault diagnosis scheme for 2-D grids.

Algorithm 2: *Testing for a single failure in 2-D grid networks*

Step 1a:

Test all edges in column 1 using a single probe.

Step 1b:

Perform the LTP on the edges in column 1 using edges between column 1 and column 2 and edges in column 2 to route the probes as necessary. Fig. 5(b) illustrates a single probe to test edge 1 and edge 3 in column 1, numbering the edges in increasing order from top to bottom.

Step 2a:

Test all edges in row 1 using a single probe.

Step 2b:

Perform the LTP on the edges in row 1 using edges between row 1 and row 2 and edges in row 2 to route the probes as necessary. (This is similar to Step 1b.)

Step 3a:

Perform the LTP on row 2 through row $\sqrt{n}$. This step differs from Steps 1b and 2b in that an entire row is treated as a single element for testing purposes. The edges in column 1 are used to route between rows. Fig. 5(c) illustrates a single probe to test row 2 and row 4.

Step 3b:

Perform the LTP on the individual row edges (the elements are $s_1, \cdots, s_{\sqrt{n}}$, where $s_i = \{i^{\text{th}}$ edge of row $j : 2 \le j \le \sqrt{n}\}$). The column edges and the edges of row 1 are used to route between rows. Fig. 5(d) illustrates a single probe to test the 2nd edges in all rows and the 4th edge in all rows.

Step 4a:

Perform the LTP on column 2 through column $\sqrt{n}$, in a manner analogous to Step 3a.

Step 4b:

Perform the LTP on the column edges, in a manner analogous to Step 3a.

The correctness of Algorithm 2 is shown in *Appendix A*. As with Algorithm 1, the strategy is first to identify a fault-free sub-graph (either column 1 or row 1), and then to use the fault-free sub-graph to route the necessary probes required by the LTPs. Algorithm 2 uses only 6 LTPs, each over a set of $\sqrt{n}$ elements, plus two additional probes. It follows that the total number of probes used is only $O(\log n)$. Combining this result with the lower bound of (1), we have established our main result for 2-D grid networks as follows.

Theorem 3: $\Theta(\log n)$ probes are needed to identify a single edge failure in a 2-D grid network of size $\sqrt{n} \times \sqrt{n}$.

In general, if multiple failures can occur simultaneously, more probes are needed. This phenomenon can be intuitively explained as follows. An edge e can be isolated by a small cut which separates it from the rest of the network. If all the edges of this cut have failed, the only way to test whether edge e has also failed is to probe edge e by itself. Theorem 4 explains this phenomenon formally.

Theorem 4: If at least 7 failures can occur, $\Theta(n)$ probes are needed to identify all the edge failures in a 2-D grid network.

Proof:

Consider Fig. 5(a), in which the 6 edges adjacent to edge e have failed. The only way to test whether edge e has also failed is to probe edge e itself. However, the identity of edge e is not known when the algorithm chooses its probes, due to the non-adaptive nature of the algorithm. Therefore, the algorithm can only know whether edge e has also failed if it performs $\Omega(m) = \Omega(n)$ probes. Combining this with the upper bound of (1) completes the proof. Q.E.D.

V. EFFICIENT DIAGNOSIS WITH ARBITRARY TOPOLOGIES

We now provide efficient testing algorithms for arbitrary graphs and trees. The algorithms depend on the diameter[4] and/or the edge-connectivity[5] of the graph. On practical networks, we expect the diameter to be relatively small, and the connectivity to be large (for failure resilience).

A. Networks with Well-Connected Topologies

As shown in Section IV, identifying *multiple* failed edges in some networks (e.g., 2-D grid networks) requires exponentially more probes than required for a *single* failed edge. This high complexity is caused by edge failures that can hide behind small cuts. One might conjecture that this phenomenon does not occur in graphs with sufficiently high connectivity. The following theorem proves such a result.

Theorem 5: If a graph G contains $s+1$ edge-disjoint spanning trees[6], the minimum number of non-adaptive probes required to identify up to s failed edges, i.e., $L^*(G,s)$, is bounded by $T^*(m,s) \le L^*(G,s) \le O(s \cdot T^*(m,s))$, where $T^*(m,s)$ is as defined in Section III. In particular, this holds in a network topology with edge-connectivity at least $2(s+1)$.

Proof:

The lower bound is immediate since the non-adaptive fault diagnosis problem is simply the combinatorial group testing problem with an additional restriction on the feasible probes.

The Tutte–Nash-Williams Theorem [23,24] implies that a network with edge connectivity of at least $2(s+1)$ has at least $s+1$ edge-disjoint spanning trees. It follows that at least one of the spanning trees, call it G_T, contains no edge failures. A single probe suffices to test if all edges of a tree are intact, therefore we can identify G_T using only $s+1$ probes. For every non-tree edge $\{u,v\}$, we create a virtual node v' and replace $\{u,v\}$ with $\{u,v'\}$. After this transformation, all non-tree edges are at the bottom of G_T, i.e., they have height zero.

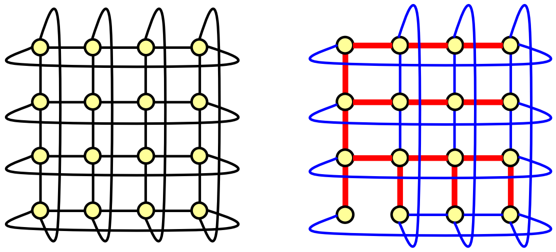
We now think of these non-trees edges as the elements to be tested, and we can use any CGT algorithm to do so. Pick a root for G_T arbitrarily; we think of the CGT algorithm as running at this root node. By our choice of G_T, the path from the root to each of the non-tree edges contains no failures. The CGT algorithm produces a sequence of tests, each of which specifies a set of elements to test. For each such set, we send a probe from the root node which traverses the tree and visits only the non-tree edges in the specified set. Therefore a probe fails if and only if one of the elements in the corresponding CGT test has failed. The results of these probes are returned to the CGT

---

[4] The diameter of a graph is the maximum shortest distance between any two nodes in the graph.

[5] Edge-connectivity means the minimum cardinality of any subset of edges whose removal disconnects the network.

[6] A spanning tree of a graph is an acyclic sub-graph containing all nodes.



(a) (b)

Fig. 6. (a) A 2-D torus of 4x4. (b) Two edge-disjoint spanning trees contained in the 2-D torus.

algorithm, and it identifies the failed edges.

To summarize, the optimal non-adaptive CGT algorithm can be applied to the set of non-tree edges, using the edges of G_T to route from the root to the non-tree edges. This approach uses $O(T^*(m,s))$ probes. Since we have to perform these tests for all $s+1$ trees, $O(s \cdot T^*(m,s))$ probes are sufficient.

Q.E.D.

We now illustrate this theorem by comparing it to our earlier results. A 2-D grid network has edge-connectivity 2, since the corner nodes have degree only 2. Therefore Theorem 5 yields no result for 2-D grids. On the other hand, consider a 2-D torus, i.e., a grid in which the edges wrap around. Such a graph is shown in Fig. 6(a). Any 2-D torus has edge connectivity 4, so it has two disjoint spanning trees. An example of two spanning trees in a 2-D torus is shown in Fig. 6(b). As consequences of Theorem 5, we have the following two corollaries.

Corollary 1: In a 2-D torus with n edges, $\Theta(\log n)$ probes are sufficient to identify a single edge failure.

Corollary 2: In a complete (i.e., fully connected) network with n nodes, $O(s \cdot T^*(m,s))$ probes are sufficient to identify up to $s \le (n-3)/2$ failed edges.

Theorem 5 also suggests the following general paradigm for applying classical CGT procedures (such as LTP) to problems on graphs.

Preprocessing:

1. Identify $s+1$ edge-disjoint connected sub-graphs. Each sub-graph will be used in turn as a "hub" to reach the edges of the graph outside itself.
2. For each hub, use a CGT algorithm to generate tests for the set of edges outside it.

Probing the network non-adaptively:

3. For each hub, verify that its edges are intact.
4. For each hub, each test from Step 2 is implemented by a permissible probe as follows: the probe traverses the interior of the hub, and steps out only onto the neighboring edges that are to be tested. Note that, *assuming* the hub is intact, the probe fails if and only if one of the edges to be tested has failed.

Diagnosis:

5. Since there are at most s failures and $s+1$ edge-disjoint hubs, at least one contains no failed edge. Such a hub can be identified based on the results of Step 3. All other hubs are ignored by the diagnosis algorithm.
6. Run the CGT algorithm on the results of Step 4 for the good hub, thus identifying all failed edges.

It can be seen that Algorithm 1 is a special case of this general procedure with $s=1$. Similar fault diagnosis algorithms can be designed for other regular networks of degree d.

B. Networks with Tree Topologies

We now consider networks with tree topologies, and obtain bounds in terms of the diameter[7]. Note that the depth, the most commonly used measure for trees, is within a factor of 2 of the diameter, for any choice of a root.

Theorem 6: For any tree G^T, when $s=1$, we have:

$$\Omega(D+\log n) \le L^*(G^T,1) \le \min\left\{ \begin{matrix} O(D \cdot \log n) \\ O(D+\log^2 n) \end{matrix} \right\}, \qquad (3)$$

where D is the diameter of the graph G^T.

The proof of Theorem 6 is given in *Appendix B*.

C. Networks with Arbitrary Topologies

In this sub-section, we address the fault diagnosis problem for networks with arbitrary topologies. The main result is summarized as follows.

Theorem 7: If a graph G contains s edge-disjoint spanning trees $T_1,\ldots,T_s$, then the minimum number of non-adaptive probes to identify up to s failed edges is upper bound by

$$L^*(G,s) \le O\left(s \cdot T^*(m,s) \;+\; \sum_{i=1}^{s} L^*(T_i, s=1) \right). \qquad (4)$$

Proof:

For *each* chosen spanning tree, we perform the following probes independently:

1. Probe the entire spanning tree.
2. *Assuming* there is exactly one failure in the edges of the spanning tree, use $L^*(T_i, s=1)$ probes to find the failure.
3. *Assuming* there is no failure inside the spanning tree, use it as a hub to diagnose at most s failures among the remaining edges. This needs $T^*(m,s)$ probes.

The diagnosis algorithm proceeds as follows. If one of the spanning trees contains no failure (this can be inferred from Step 1), the information gathered in Step 3 for this spanning tree will solve the problem. Otherwise, each tree contains exactly one failed edge. Step 2 identifies a unique failed edge inside each spanning tree. Q.E.D.

Theorem 7 implies an upper bound for arbitrary graphs as follows.

---

[7] The diameter of a graph is the maximum shortest distance between any two nodes in the graph.

Corollary 3: For an arbitrary graph G and $s=1$, we have:

$$L^*(G,1) \le \mathrm{O}(D+\log^2 n)\,, \tag{5}$$

where D is the diameter of the graph.

Proof:

Choose the spanning tree to be a shortest path tree from an arbitrary starting node. This guarantees that the depth of the tree is at most the diameter of G. It follows from Theorem 6 that $L^*(T_i,s=1)=\mathrm{O}\left(D+\log^2 n\right)$ and from the LTP that $T^*\left(m,s=1\right)=\log n$. Q.E.D.

VI. CONCLUSION

In this paper, we have considered the fault diagnosis problem for all-optical networks. We focused on a proactive fault diagnosis framework, in which a set of probes are sent along lightpaths to test whether they have failed; the network failure pattern is identified using the results of the probes. As an alternative to our previous adaptive design, we proposed a non-adaptive probing design, where the set of pre-determined probes are sent in parallel so that the number of probing steps is always one. The key objective of our design is to minimize the number of probes sent in parallel, so as to keep the fault diagnosis cost low.

The non-adaptive fault diagnosis problem for all-optical networks is equivalent to the combinatorial group testing problem on graphs. In the latter problem, probes can only be sent over walks over the graph, and therefore such probes correspond to lightpaths in all-optical networks. In this framework, we developed efficient fault diagnosis algorithms (some of them are optimal in achieving the lower bound) for different classes of network topologies, and obtained upper and/or lower bounds on the number of non-adaptive probes needed. The efficient non-adaptive fault diagnosis algorithms that we developed share a common theme: a fault-free sub-graph should be identified in the network and serve as a hub to route other necessary probes to diagnose failures in the network.

Although this research was presented in the context of all-optical networks, we believe that our methods based on combinatorial group testing on graphs can be employed in other network contexts to solve fault diagnosis problems.

APPENDIX

A. Correctness of Algorithm 2

The correctness of Algorithm 2 can be established as follows.

- Suppose that the edge failure happens in column 1. This fact will be uncovered in Step 1a. The edges in all other columns and in all rows are intact, and therefore it is valid to use them for routing in Step 1b. It follows that Step 1b correctly performs the LTP on the edges of column 1 and identifies the edge failure.
- Suppose that the edge failure happens in row 1. A similar argument shows that Step 2 identifies the edge failure.
- Suppose that the edge failure happens on the i^{th} edge in row $j\ge 2$. All column edges are intact, and can be used to route probes in Step 3a. It follows that Step 3a correctly performs the LTP on all rows and identifies the row containing the edge failure. The edges of row 1 are intact, and can be used for routing probes in Step 3b to identify the edge failure.
- Suppose that the edge failure happens on the i^{th} edge in column $j\ge 2$. A similar argument shows that Step 4 identifies the edge.

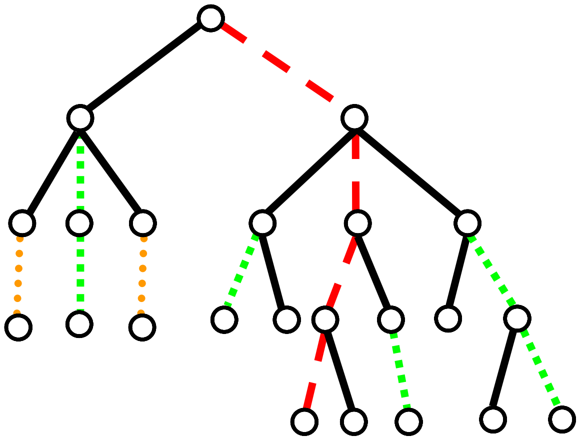


Fig. 7. An illustration of the heavy-light decomposition. Preferred paths at different depths are labeled with different colors and indexes.

B. Proof of Theorem 6 for Tree Topologies

For the proof, we fix an arbitrary root. First consider the lower bound. The $\Omega(\lg n)$ bound is inherited from the CGT lower bound of (1). The $\Omega(D)$ bound follows from the lower bound for linear networks of Theorem 2, as follows. Consider only the path from the root to the deepest leaf, which has length at least $D/2$. By truncating every probe to its intersection with this path, we obtain a solution to the problem on the path (a linear network).

We now show the upper bound of $\mathrm{O}(D\cdot\lg n)$. This dominates in the case of trees of sub-logarithmic depth (which necessarily have high degree). The strategy is quite simple. For each depth $d\in[0,D-1]$, we do the following:

1. Probe the sub-tree containing the root and all nodes up to depth d.
2. Assuming that the failed edge is at level $d+1$, use the sub-tree of depth d as a hub to test nodes at depth $d+1$.

The diagnosis algorithm first looks at probes of type 1, and determines the level at which the failure occurred. Then, it uses the probes of type 2 made at the relevant level.

The problem with this direct approach is that it involves a CGT step at every level of the tree, which is potentially wasteful when the tree has depth much larger than $\lg n$. To handle unbalanced trees more efficiently, we use a technique known as the *heavy-light* decomposition.

Define the weight of a node to be the number of nodes under it. For each non-leaf node, define a heavy edge to go from the node to its heaviest child. Preferred paths are defined as the maximal paths in the graph containing only heavy edges. For convenience, we will also consider the light edge immediately above a preferred path to be a part of the path. Thus, all edges of the tree are in a preferred path.

For each edge, its *light depth* is the number of light edges on the path from the edge to the root. All heavy edges in a preferred path have the same light depth, so we can also talk about the light depth of a preferred path. A standard argument shows the light depth is always $O(\lg n)$, because each time we follow a light edge, the number of nodes under the current node decreases by at least a factor of 2. An example of heavy-light decomposition is illustrated in Fig. 7.

Our solution performs the following probes:

1. For each depth $d \in [0, D-1]$, probe the sub-tree containing the root and all nodes up to depth d.
2. For each light depth ℓ:
   A. Probe the sub-tree containing all preferred paths up to light depth ℓ.
   B. Under the hypothesis that the sub-tree does not contain a failure, use it as a hub to test the preferred paths at light depth $\ell + 1$. Such a preferred path is viewed as a single element for the CGT algorithm; each probe either includes all edges in the path or none.

The diagnosis algorithm works as follows. By examining data from Step 1, it determines the depth of the failed edge. Then, it only needs to find out the preferred path containing the failure. From the data of Step 2A, one can gather the light depth of the failure. Finally, the analysis only considers the relevant hub among the data from Step 2B.

REFERENCES

[1] M. Subramanian, *Network Management: Principles and Practice*, 1^{st} edition, Addison-Wesley Lognman, 2000.

[2] R. Ramaswami and K. N. Sivarajan, *Optical Networks: A Practical Perspective*, 2^{nd} edition, Morgan Kaufmann Publishers, 2002.

[3] N. Feamster and H. Balakrishnan. "Detecting BGP Configuration Faults with Static Analysis". In Proceedings of the 2nd Symposium on Networked Systems Design and Implementation (NSDI), May 2005.

[4] S. Kandula, D. Katabi, J.-P. Vasseur, "Shrink: A Tool for Failure Diagnosis in IP Networks," MineNet Workshop, SIGCOMM, August 2005.

[5] J. Kleinberg. "Detecting a Network Failure". In Proceedings of the 41st Annual Symposium on Foundations of Computer Science (FOCS), November 2000.

[6] A. Adya, P. Bahl, R. Chandra and L. Qiu, "Architecture and techniques for diagnosing faults in IEEE 802.11 infrastructure networks," In Proceedings of the 10th Annual international Conference on Mobile Computing and Networking (MOBICOM), Sept. 2004.

[7] L. Qiu, P. Bahl, A. Rao and L. Zhou, "Troubleshooting multihop wireless networks," In Proceedings of ACM SIGMETRICS (extended abstract), June 2005.

[8] Y.G. Wen, V. W. S. Chan and L. Z. Zheng, "Efficient Fault Diagnosis Algorithms for All-Optical WDM Networks with Probabilistic Link Failures (*invited paper*)," *IEEE/OSA Journal of Lightwave Technology*, Vol. 23, No. 10, October 2005, pp.3358-3371.

[9] Y. G. Wen, V. W. S. Chan, and L. Z. Zheng, "Efficient Fault Diagnosis Algorithms for All-Optical Networks: An Information Theoretical Approach," *2006 IEEE International Symposium on Information Theory (ISIT), Seattle, WA, July 2006.*

[10] Y. G. Wen, V. W. S. Chan, and L. Z. Zheng, "Efficient Fault Detection and Localization for All-Optical Networks," *2006 IEEE Global Communications Conference (GlobeCom), San Francisco, December 2006*.

[11] V. W. S. Chan, et al, "A pre-competitive consortium on wide-band all-optical networks," *IEEE/LEOS Journal of Lightwave Technology*, Vol. 11, No. 5/6, May/June, 1993, pp. 714-735.

[12] V. W. S. Chan, "All-optical networks," *Scientific American*, No. 3, September 1995, pp. 56-59.

[13] "Architecture of Optical Transport Networks," ITU-T Recommendation G. 872, Feb. 1999.

[14] K. R. Pattipati and M. G. Alexandridis, "Application of heuristic search and information theory to sequential fault diagnosis," *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 20, No. 4, July/August 1990, pp. 872-886.

[15] F. P. Preparata, G. Metze and R. T. Chien, "On the connection assignment problem of diagnosable systems," *IEEE Transaction on Electronic Computers*, Vol. 16, No. 6, December 1967, pp. 848-854.

[16] D. Du and F. Hwang, *Combinatorial group testing and its applications*, World Scientific, 2000.

[17] C. J. Colbourn, J. H. Dinitz, and D. R. Stinson. "Applications of combinatorial designs to communications, cryptography, and networking". In Surveys in Combinatorics, 1993, Walker (Ed.), London Mathematical Society Lecture Note Series 187. Cambridge University Press, 1999.

[18] A. Bar-Noy, F. K. Hwang, I. Kessler, S. Kutten, "A New Competitive Algorithm for Group Testing," Discrete Applied Mathematics 52(1): 29-38 (1994).

[19] G. Weichenberg, *High-Reliability Architectures for Networks under Stress*, M.Sc. Thesis, EECS, MIT, 2003.

[20] W. H. Kautz and R. C. Singleton, "Nonrandom binary superimposed codes," IEEE Transaction on Information Theory, Vol. 4, No. 10, October 1964, pp. 363-377.

[21] A. G. D'yachkov and V. V. Rykov, "Bounds for the length of disjunctive codes," Problems of Information Transmission, vol. 18, no. 3, pp. 7-13, 1982.

[22] F. T. Leighton. *Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes*. Morgan-Kaufman, 1991.

[23] C. St. J. A. Nash-Williams, "Edge-disjoint spanning trees of finite graphs," Journal of the London Mathematical Society, 36 (1961), pp. 445-450.

[24] W. T. Tutte, "On the problem of decomposing a graph into n connected factors," Journal of the London Mathematical Society, 36(1961), pp. 221-330.

[25] W.J. Lai, P. Shum and L.N. Binh, "NOLM-NALM fiber ring laser," IEEE Journal of Quantum Electronics, Vol. 41, No. 7, July 2005, pp. 986-993.