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 fault diagnosis problem in all-optical networks, focusing on probing schemes to detect faults. Our work concentrates on non-adaptive probing schemes, in order to meet the stringent time requirements for fault recovery. This fault diagnosis problem motivates a new technical framework that we introduce: group testing with graph-based constraints. Using this framework, we develop several new probing schemes to detect network faults. The efficiency of our schemes often depends on the network topology; in many cases we can show that our schemes are near-optimal by providing tight lower bounds.**

I. Introduction

Network management is a crucial but expensive aspect of any modern network [1]. Network management provides five functionalities including management of faults, configuration, performance, security and accounts. Currently, much of the effort and cost lies in fault management [2], specifically in detection and isolation of the failures. Due to the cost and importance of fault management, it has been an active research topic in a variety of contexts, such as the Internet [22-24], wireless networks [3,4], and optical networks [5-7]. In this work, we focus on fault management in all-optical networks. The qualities and performance characteristics of all-optical networks yield interesting challenges for fault management, as we explain below.

The promise of all-optical networks is compelling: broadband network services can potentially be delivered to large populations at much lower cost than today's technologies [8,9]. The significant cost savings are due to optical switching of high data-rate lightpaths at network nodes, thereby reducing electronic processing costs. This reduction of electronic equipment necessitates new mechanisms to diagnose link failures, since the side benefit of electronic switching, namely parity checks at the end of each SONET line, is now absent. As with other networks, all-optical networks are susceptible to various 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 difficult to detect, localize and repair. Hence, when parts of a network are malfunctioning it is critical to locate and identify these failures as soon as possible. In the current SONET standard [10], the fault detection and restoration time is around 50ms, and this will probably be reduced further in future all-optical networks. The fault diagnosis solution that we propose in this paper is designed to address both the cost and timeliness concerns.

Our fault diagnosis methodology is based on the proactive probing paradigm. In particular, probing signals are sent to test the health of the network, and the *probe syndromes* (i.e., results of the probes) are used to identify any potential failures. Within this paradigm, two alternative designs predominate: *adaptive* probing, and *non-adaptive* probing. Each design has its advantages and disadvantages, as outlined below.

In adaptive fault diagnosis schemes [11], a set of probing signals are sequentially sent to probe the health of the network, and subsequent probes are chosen based on the results of previous probes. Due to the sequential nature of this process, the number of probes used by adaptive schemes is usually quite low. On the other hand, the delay to identify failures, i.e., the Capture Time, might be quite large for some failure patterns. Furthermore, the coordination required among all the monitoring modules to send these probes sequentially might be quite complicated.

In non-adaptive fault diagnosis schemes [12], a set of probing signals are sent independently, within a time window specified by the standard, and network failures are identified through the set of probe syndromes. We now outline a natural design for implementing non-adaptive schemes in practice. First, the network management system (NMS) examines the topology of the network, and decides on an appropriate choice of probing signals. Next, the NMS configures network nodes to send these probing signals periodically. The probe syndromes are gathered at one or more central nodes (via out-of-band mechanisms, or a dedicated control channel). Should a network fault occur, this can be detected at the central nodes, simply by examining the probe syndromes.

This non-adaptive probing scheme offers several practical benefits. First, it requires very little synchronization, in contrast with the adaptive fault diagnosis scheme; this quality makes it more appealing for use in a distributed system. Second, a snapshot of the entire network health is available at any time at the central nodes. However, non-adaptive schemes might require more probes than adaptive schemes since the information provided by a probe cannot be leveraged to choose subsequent probes.

Previous research on all-optical networks [5-7] has

*The research in this paper was supported in part by the Defense Advanced Research Projects Agency (DARPA). Student authors in this paper are list in an alphabetical order.

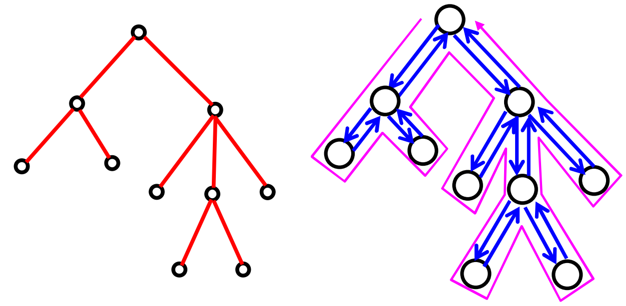
investigated adaptive fault diagnosis schemes for networks with probabilistic node/link failures. In particular, based on information theoretic insights, the run-length probing scheme has been proposed, and its performance has been shown to be within 5% of the entropy lower bound. However, this scheme may require significant delay to identify failures in some large networks, or under certain network failure patterns. This undesirable delay is due to the adaptive nature of the probing scheme.

In this paper, we consider non-adaptive approaches to the fault diagnosis problem for all-optical networks. Instead of the probabilistic failure model used in previous work, this paper allows arbitrary failure patterns, assuming only that the number of failures is bounded, i.e., a worst-case failure model. Instead of sending optical probing signals sequentially, a pre-determined set of optical probing signals are sent independently to probe the network state of health. The objective of this research is to develop algorithms to choose efficient non-adaptive probing schemes, thereby obtaining inexpensive fault diagnosis mechanisms.

Our fault detection methods are based on techniques from the field of *combinatorial group testing (CGT)* [13]. This field has a wide variety of practical applications, such as HIV screening, DNA testing, MAC design, and much more [15]. It has also been used in network management applications (see, e.g., [16]), 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 instigate its use more widely. The present paper considers only the context of all-optical networks since their performance characteristics lead to a natural application of CGT. More specifically, the switching characteristics of all-optical networks imply that probing several interconnected edges costs no more (in terms of transmitter/receiver cost) than probing a single network 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 obvious condition that lightpaths can only traverse interconnected edges. To the best of our knowledge, this is a novel framework for CGT, 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 algorithms have a common theme, which suggests a practical rule-of-thumb for efficient fault diagnosis: a fault-free sub-graph in the network topology should be identified, and used as a "hub" to diagnose other failures in the 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,



(a) (b)

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

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. THE NON-ADAPTIVE FAULT DIAGNOSIS PROBLEM

In this section, we present the permanent link failure model and a formal definition of a non-adaptive fault diagnosis algorithm from an engineering perspective.

A. *Permanent Link Failure Model*

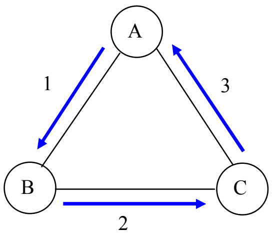
In this paper, all-optical networks are abstracted as undirected graphs. An *undirected graph* G is an ordered pair of sets (V, E), where V is the set of nodes, and E is the set of edges. The number of nodes is n and the number of edges is m. The terms links and edges are used interchangeably in this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

In our model, we assume links fail and nodes do not. This assumption is realistic: in all-optical networks, the passive optics used in network nodes is highly reliable. On the other hand, graph edges correspond to fiber links, optical amplifiers and transmitter/receivers, which are significantly more prone to failures [17]. 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. We do not place any restriction on the edge failure patterns. Since it is unlikely that numerous edge failures happen simultaneously, we assume that the number of edges failures is upper bounded by a constant $s(\le m)$ at any instant. In this paper, we generally allow s to be arbitrary, although the case $s = 1$ is often central.

B. *Non-Adaptive Fault Diagnosis Algorithm*

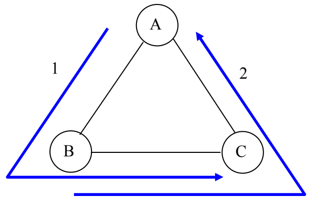
In this paper, we focus on diagnosing network failures via the measurements of end-to-end probing signals. Specifically, each probe corresponds to sending an optical signal along some lightpath. We will illustrate the probing model in this sub-section.

A *probe* in the network corresponds to a walk (a sequence of adjacent edges, allowing repetitions) in the corresponding



	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.

	Scheme (a)	Scheme (b)
Number of Wavelengths	**1**	**2**
Number of Tx/Rx	**3**	**2**

Fig.3 The cost comparison between the two non-adaptive fault diagnosis algorithms in Fig. 2.

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 and transmitters/receivers. 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, whose failures are perfectly correlated since they are usually contained in the same conduit. 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, 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 which has failed.

	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. 4 The diagnosis matrix for the logarithmic testing procedure with m=7. Columns correspond to elements to be tested, and rows correspond to tests.

Besides the capture time, another key design consideration of our research is the cost of the fault diagnosis scheme. For all-optical networks, the cost of a lightpath probe includes the cost of the transmitter/receiver hardware, and the wavelength usage. For the scheme in Fig. 2(a), the number of wavelengths used is 1 because no probed lightpath overlaps with another. The number of transmitters/receiver pairs is 3 because three probe lightpaths are employed. For the scheme in Fig. 2(b), the number of wavelength is 2 because of the overlapped lightpaths, and the number of transmitters/receivers pairs is 2. This comparison is summarized in Fig. 3. If the usage of wavelength is the dominant cost, Scheme 2(a) is preferable to Scheme 2(b). On the other hand, if the cost of transmitter/receiver pairs dominates, Scheme 2(b) is preferable. Our design considers only the transmitter/receiver hardware cost because in practical optical networks, the cost of wavelength is negligible compared to the cost of hardware. Therefore minimizing the cost is equivalent to minimizing the number of probes sent. This design objective has an additional benefit: minimizing the number of probes also reduces the amount of management information, with commensurate improvements in transferring, storing and processing this information.

In summary, the objective of this research is to develop efficient non-adaptive fault diagnosis schemes with minimum number of probes, thereby minimizing the fault diagnosis cost.

III. Combinatorial Group Testing (CGT)

In this section, we present relevant theoretical background on the combinatorial group testing (CGT) problem.

The general 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 we think of as being small relative to m. We are allowed to perform group tests of the following form: specify a subset $t \subset S$, run the group test on t, and learn if there is at least one faulty 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 [14].

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:

$$\log_2(m+1) \le T^*(m,1) \le m. \qquad (1)$$

For arbitrary s and sufficiently large m, it has been shown that $T^*(m,s)$ can be bounded[1] as,

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

where the upper bound comes from [14] and is essentially a simple random superimposed coding argument, and the lower bound is due to D'yachkov and Rykov [18].

From any non-adaptive combinatorial group testing algorithm with $T(m,s)$ tests, we construct a testing matrix with $T(m,s)$ rows and $N(m,s)$ columns, where each row corresponds to a group test and each column corresponds to a failure pattern. We set $c_{ij}=1$ if failure pattern j fails in group test i; otherwise, $c_{ij}=0$. As a simple illustration, consider the case of $s=1$ and $m=7$; the testing matrix is shown in Fig 4. 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 ceil(log(m+1)) rows (corresponding to the tests) and m+1 rows (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 1 entry in the row of the diagnosis matrix. It is easy to see that if item *i* is 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).

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 In Regular Networks

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

A. Networks with Linear or Ring Topologies

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

Consider a network consisting of n nodes, indexed by integers $0,1,\cdots,n-1$. For all $1 \le i \le n-2$, node i is connected to node $i-1$ and node $i+1$, so the number of edges is $m=n-1$. For a linear network with m edges, we can establish the following result.

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

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 allows locating 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 such case, no probe t_j includes an edge that is adjacent to node i. Therefore, probe algorithm cannot separate between the case when all edges are intact and the case when edge adjacent to node i has failed.
- $1 \le i \le n-2$: In such case, every test t_j either contains or does not contain both edge $(i-1,i)$ and edge

[1] $f(n)=O(g(n))$ 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 $f(n)=\Omega(g(n))$.

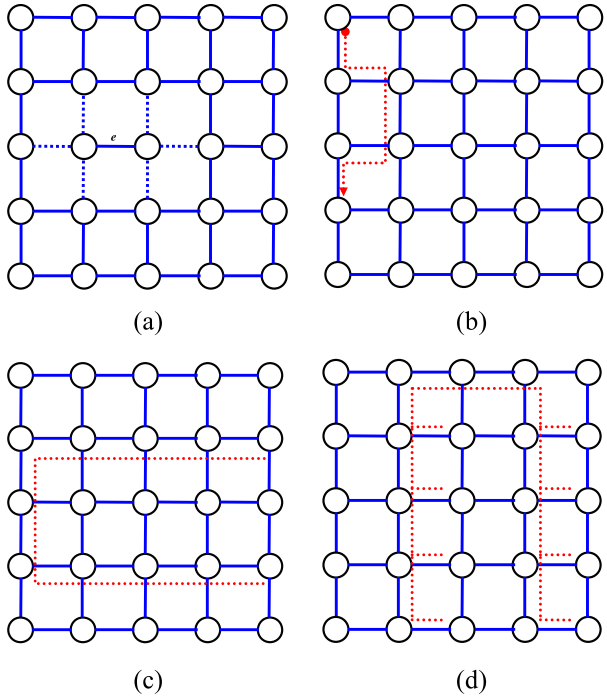


Fig. 5. (a) A 2-D grid with 25 nodes with at most 7 failures. The failure of edge e cannot be detected efficiently by non-adaptive tests. (b) a single probe to test edge 1 and edge 3 on column 1; (c) a single probe to test column 2 and column 4; (d) single probes to test the 2nd edge on all rows and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

$(i, i+1)$ simultaneously. Therefore, the probe algorithm cannot separate between the case when edge $(i-1, i)$ has failed and the case when failure edge $(i, i+1)$ has failed.

In both cases, we arrive at a contradiction and conclude that $l \geq \lceil n/2 \rceil$ 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 $0 \leq j \leq \lceil n/2 \rceil - 1$. Clearly, every edge e belongs to some test t_j. Therefore all we need to show is that, for every pair of edges $e_1 \neq e_2$ 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 $h_1 \leq t_2$. Consider the following two cases:

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

This completes the proof. Q.E.D.

This $\Theta(m)$ bound for linear networks is much larger than the upper bound of $O(\log m)$ for the LTP procedure. Intuitively, the low connectivity of the 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 probes).

B. Networks with 2-D Grid Topologies

The sub-section considers two-dimensional grid networks of size $\sqrt{m} \times \sqrt{m}$. Such structures are also commonly used as interconnection networks [19]; in the context of all-optical networks, they are sometimes called Manhattan networks. Fig 5 illustrates the case of $m = 25$. The following algorithm gives an optimal non-adaptive fault diagnosis scheme for 2-D grids.

Algorithm 1: *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 edges on column 1 using edges between column 1 and column 2 and edges on column 2 to navigate necessary probes. Fig. 5(b) illustrates a single probe to test edge 1 and edge 3 on column 1.

Step 2a:

Test all edges in row 1 using a single probe;

Step 2b:

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

Step 3a:

Perform the LTP on row 2 through row $\sqrt{m}$ (an entire row is treated as a single element for testing purposes). The edges on column 1 are used to navigate between rows. Fig. 6(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 $s_1, \cdots, s_{\sqrt{m}}$, where $s_i = \{i^{\text{th}} \text{ edge of row } j\text{: } 2 \leq j \leq \sqrt{m}\}$). The column edges and the edges of row 1 are used to navigate between rows. Fig. 5(d) illustrates a single probe to test the 2nd edges on all rows and the 4th edge on all rows.

Step 4a:

Perform the LTP on column 2 through column $\sqrt{m}$, 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 1 is shown in Appendix A. Since only permissible probes can be used, we cannot probe an arbitrary subset of edges in the network topology, and thus the combinatorial group testing algorithms cannot be applied directly. Instead, our algorithm first identifies a fault-free sub-graph in the topology, and then uses the fault-free sub-graph to navigate the necessary probes required by the optimal logarithm testing procedure. Algorithm 1 uses only 6 LTPs, each over a set of $\sqrt{m}$ elements, plus two additional probes. It

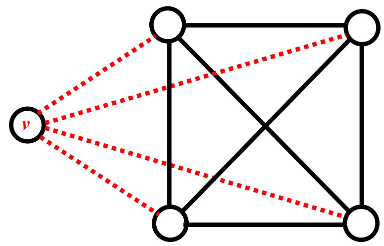


Fig.6. Complete graph of $n=5$, where node v and its neighborhood is chosen to navigate probes.

follows that the total number of probes used is only $O(\log m)$. Combining this result with the lower bound of (1), we have established our main result for 2-D grid networks as follows.

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

In general, if multiple failures can occur simultaneously, more probes are needed. This phenomenon can be intuitively explained as follows. An edge e can hide behind 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 3 illustrates this phenomenon formally.

Theorem 3: If at least 7 failures can occur, $\Theta(m)$ 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 $\Omega(n)$ probes. Combining the upper bound of (1), this completes the proof. Q.E.D.

C. *Networks with Fully-Connected Topologies*

This sub-section deals with the non-adaptive fault diagnosis problem for all-optical networks whose topologies are complete graphs. In particular, for a topology of n nodes, each node is connected to all other nodes in the network, resulting in $m=n(n-1)/2$ edges in the network. Complete graphs of n nodes are denoted as K_n, where $n=5$ is illustrated in Fig. 6. For such a network, we propose the following non-adaptive fault diagnosis algorithm.

Algorithm 2: *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 which connects it to al other nodes. As shown in Fig. 6, it is a star topology centered at node v.

Step 1b:

Perform the LTP on the sub-graph $B(v)$.

Step 2:

Perform the LTP on the rest of the graph, i.e., an K_{n-1}, using the sub-graph $B(v)$ to navigate all the probes if needed.

The correctness of Algorithm 1 can be shown similarly as for Algorithm 1. As in Algorithm 1, Algorithm 3 first identifies a fault-free sub-graph in the network and uses it to navigate the LTP procedure. Algorithm 3 uses two LTPs of size $(n-1)$ and size $(n-1)(n-2)/2$, and therefore the number of probes requires is $O(\log m)$. Combining this result with the lower bound of (1), we have established our main result for complete networks as follows.

Theorem 4: $\Theta(\log m)$ probes are necessary and sufficient to identify a single edge failure in a fully connected network with m edges.

V. Efficient Diagnosis With Arbitrary Topologies

We now provide efficient testing algorithms for arbitrary graphs and arbitrary trees. The algorithms depend on the diameter and/or the connectivity 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 that for a *single* failed edge. As was illustrated in Fig. 6(a), the high complexity is caused by edges 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[2], 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 the minimum number of tests to identify s positive elements in a set of size m. In particular, this holds in a network topology with edge-connectivity[3] at least $2(s+1)$.

Proof:

First, the lower bound follows from the fact that the non-adaptive fault diagnosis problem is equivalent to the combinatorial group testing problem with an additional constraint of feasible probes.

Second, the Tutte-Nash-Williams theorem [20,21] implies that such a network with edge connectivity of at least

[2] A spanning tree of a graph is a subgraph that contains all the nodes and is tree.

[3] Edge connectivity means the minimum cardinality of all edge cuts in the network.

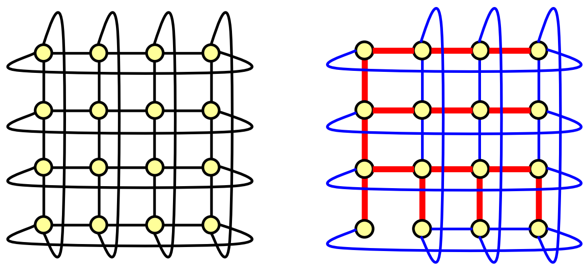


Fig. 7. (a) A 2-D torus of 4x4; (b) two edge-disjoint spanning trees contained in the 2-D torus.

$2(s+1)$ 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 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 navigate from the root to the non-tree edges. This approach use $O(T^*(m,s))$ 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 provides no results 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. 7(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. 7(b). As a consequence of Theorem 5, we have the following two corollaries.

Corollary 1: In a 2-D torus with m edges, $\Theta(\log m)$ probes are sufficient to identify up 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 subgraphs. Each subgraph 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 group-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 2 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 look at all-optical networks with tree topologies, and obtain bounds in terms of the diameter[4]. Note that depth, the most commonly used measure for trees, is within a factor of 2 of the diameter, for any choice of a root.

Theorem 7: 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.

Proof of Theorem 7 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 8: If a graph G contains s edge-disjoint spanning trees $T_1,...,T_s$, the minimum number of non-adaptive probes to identify up to s failed edges is upper bound by

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

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

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 (as seen 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 8 implies an upper bound for arbitrary graphs as follows.

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

$$L^*(G,1) \le O(D + \log^2 n), \qquad (5)$$

where D is the diameter of the graph.

Proof:

Choose as a spanning tree 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 7 that $L^*(T_i, s=1) = O\left(D + \log^2 n\right)$ 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 the 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. We proposed a non-adaptive probing design, due to its asynchronous nature and ease of implementation. The key objective of our design is to minimize the number of probes sent, in order to minimize the total hardware cost.

The non-adaptive fault diagnosis problem for all-optical networks is equivalent to the combinatorial group testing problem over 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 for different classes of network topologies, and obtained upper and/or lower bounds on the number of non-adaptive probes needed. The non-adaptive fault diagnosis algorithms that we proposed share a common theme: a fault-free sub-graph should be identified in the network and serve as a hub to navigate 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. Moreover, we would like to investigate the non-adaptive fault diagnosis problem under probabilistic failure model from an information theoretic perspective.

Appendix

A. Correctness of Algorithm 1

The correctness of Algorithm 1 can be established as follows.

- Suppose that the edge failure happens on 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 navigation in Step 1b. It follows that Step 1b correctly performs the GTP on the edges of column 1 and identifies the edge failure.
- Suppose that the edge failure happens on 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 navigate in Step 3a. It follows that Step 3a correctly performs the GTP on all rows and identify the row containing the edge failure. The edges of row 1 are intact, and can be used for navigation 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.

B. Proof of Theorem 7 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. We can concentrate just 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 $O(D \cdot \lg n)$. 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 subtree containing the root and all nodes up to depth d.
2. Assuming that the failed edge is at that level $d+1$, use the subtree 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

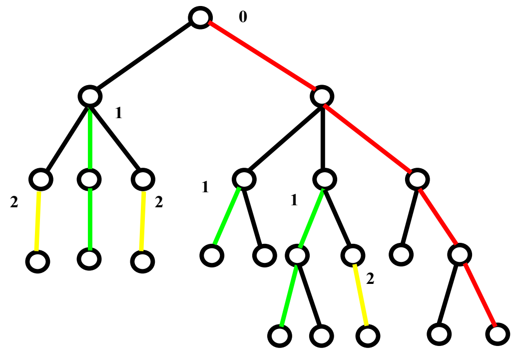


Fig. 8. An illustration of the heavy-light decomposition. Preferred paths on different depths are labeled with different colors and indexes.

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 economically, 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. 8.

Our solution performs the following probes:

1. For each depth $d \in [0, D-1]$, probe the subtree containing the root and all nodes up to depth d.
2. For each light depth ℓ:
   A. Probe the subtree containing all preferred paths up to light depth ℓ.
   B. Under the hypothesis that the subtree 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*, 1st edition, Addison-Wesley Lognman, 2000.

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

[3] 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.

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

[5] 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.

[6] Yonggang Wen, Vincent W. S. Chan, and Lizhong 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.*

[7] Yonggang Wen, Vincent W. S. Chan, and Lizhong Zheng, “Efficient Fault Detection and Localization for All-Optical Networks,” *2006 IEEE Global Communications Conference (GlobeCom), San Francisco, December 2006.*

[8] 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.

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

[10] “Architecture of Optical Transport Networks,” ITU-T Recommendation G. 872, Feb. 1999.

[11] 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.

[12] 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.

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

[14] 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.

[15] 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.

[16] A. Bar-Noy, Frank K. Hwang, Ilan Kessler, Shay Kutten: A New Competitive Algorithm for Group Testing. Discrete Applied Mathematics 52(1): 29-38 (1994).

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

[18] 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.

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

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

[21] 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.

[22] 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.

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

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