Efficient Fault Diagnosis Algorithms for All-Optical WDM Networks

Yonggang Wen, *Student Member, IEEE*, Vincent W.S. Chan, *Fellow, IEEE*, and Lizhong Zheng, *Member, IEEE*

(Invited Paper)

***Abstract*—This paper investigates the fault diagnosis problem for all-optical wavelength-division multiplexing (WDM) networks. We propose a family of failure localization algorithms that exploit the unique properties of all-optical networks. Optical probe signals are sequentially sent along a set of designed lightpaths and the network state is inferred from the result of this set of end-to-end measurements. The design objective is to minimize the diagnosis effort (e.g., the average number of probes) to locate failures.**

By establishing a mathematical equivalence between the fault diagnosis problem and the source coding problem in Information Theory, we obtain a tight lower bound for the minimum average number of probes per edge (of the network modeled as a graph) as $H_b(p)$, the entropy of the individual edges. Using the rich set of results from coding theory to solve the fault diagnosis problem, we show that the '2^m-splitting' probing scheme is optimum for the special case of single failure over a linear network. We then develop a class of near-optimum run-length probing schemes that have low computation complexity. Analytical and numerical results suggest that the average number of probes per edge for the run-length probing scheme is uniformly bounded above by $(1+\varepsilon)H_b(p)$, and converges to the entropy lower bound as the failure probability decreases. From an information theoretic perspective, we show that the run-length probing scheme outperforms the greedy probing scheme of the same computational complexity. The investigation reveals a *guideline* for efficient fault diagnosis schemes: *each probe should provide approximately 1-bit of information and the total number of probes required is approximately equal to the entropy of the state of the network.* This result provides an insightful guideline to reduce the overhead cost of fault management for all-optical networks and can further the understanding of the relationship between information entropy and network management. We also address several practical issues in the implementation of run-length probing schemes over all-optical WDM networks.

***Index Terms*— All-optical networks, run-length code, fault diagnosis, fault management, network management.**

Manuscript received February 15, 2005. The research in this paper was supported in part by the Defense Advanced Research Projects Agency (DARPA).

The authors are with the Laboratory for Information and Decision Systems (LIDS), Massachusetts Institute of Technology (MIT), Cambridge, MA 02139 USA (Phone: 617-452-3443; Fax: 617-253-3578; E-mail: {eewyg, chan, lizhong}@mit.edu).

I. Introduction

ALL-optical networks [1][2], where data traverses along lightpaths without any optical-to-electrical conversion, will be increasingly prevalent in future broadband networks due to its inherent large transmission bandwidth, lower cost and transparency to different signal formats and protocols. However, similar to other networks, all-optical networks are also vulnerable to failures [3], such as fiber cuts and transmitter/receiver breakdowns. Moreover, there are new types of failures that are unique to all-optical networks – failures related to subtle changes in signal power, optical signal-to-noise ratio, cross-talk, Kerr effects, or other non-linear effects. 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. At the same time, the cost to detect and locate failures must be small to keep the network cost low. In this paper, we propose a family of efficient fault diagnosis algorithms that exploit the unique property of all-optical WDM networks where optical signals are not usually detected at intermediate nodes along lightpaths (mostly for cost reasons).

According to the scale of their effect, failures in all-optical WDM networks can be classified into two categories. One category is a wavelength-level failure which impacts the quality of transmission of each individual lightpath, e.g. transmitter/receiver failures in the case of one dedicated transmitter/receiver per wavelength, optical filter failures and individual channel failures of a frequency selective switch. The other category is a fiber-level failure which affects all the lightpaths on an individual fiber, for example, fiber cuts, EDFA breakdowns and transmitter/receiver failures in the case of only one tunable transmitter/receiver per fiber. From a graph theory perspective, we can attribute both categories to edge failures in a network (graph) topology. In this paper, we focus on the ON/OFF edge failure which is modeled by a binary-value function of value 0 if the required quality of transmission is met and of value 1 otherwise. Besides, all the failures that do not belong to the same risk group are considered independent. In real life, there may be failure correlation among risk groups due to physical proximity or accessibility from the same malicious attack entry point. In those cases the results in this paper can still provide very useful upper and lower bounds to the diagnostic effort required to localize the failures.

Since the fault diagnosis problem [4] was first proposed in 1967, it has been investigated extensively in electrical networks under a system diagnosis context [5-7]. In this context, most current research is focused on a "single hop" test model, i.e., signals are transmitted between adjacent nodes to determine whether failure occurs on the edge connecting them. The result of each test can be represented as one bit of diagnosis information: 1 or 0, corresponding to "failure" or "no failure". Indeed for SONET networks each SONET link (single hop) checks the health of the link using parity checks within the SONET receiving chips. However, in all-optical networks, this 'single-hop test' assumption will usually not be applicable due to the unique property that optical signals are not typically detected at every (optically switched) node along the lightpath.

For SONET networks, the network management system employs mechanisms such as BER measurement, optical trace and alarm management to perform fault management at each regenerator. In particular, these functionalities may be carried over various types of optical layer overhead [8], including pilot tone, subcarrier-modulated overhead, optical supervisory channel, rate-preserving overhead and digital wrapper overhead. To some degree, all these overheads are detected at some intermediate nodes along the lightpath. This, in fact, breaks the spirit of the transparency paradigm and adds to the complexity and cost of future all-optical networks which do not need signal detection along a lightpath.

Currently, to diagnose failures in future all-optical WDM networks, researchers typically consider an optical (channel) performance monitoring solution, where optical performance monitors are employed at a set of network nodes to watch for possible failures and report them to the network management system [9]. However, little work has been done to quantify the overhead cost that this monitoring solution might incur. Instead, most research literatures [10, 11] follow essentially the same design approach as their electrical counterparts, implicitly assuming that each network node, or even each active optical component such as optical amplifiers and OADMs (Optical Add-Drop Multiplexer), is equipped with a performance monitoring module which is active and reporting all the time. While this is an acceptable solution in the near-term since signal detection comes for free at every regeneration point, it is desirable to develop more efficient and less costly methods when the all-optical network paradigm is fully implemented and the network size grows significantly. Reduced complexity is good for the following reasons. First, the total amount of monitored information and signaling grows quickly with the number of network elements (i.e., network nodes and edges). The huge amount of management information, together with faster switching speeds in the network, complicates the network management system and stresses the limited capability of current network processing units and a mechanism based on constant sensing and reporting of numerous individual active monitors does not scale well with the size and tuning agility of future all-optical networks. Second, since each monitor only tests one component without taking into consideration of its failure statistics, the diagnosis overhead cost (e.g., the required number of tests per unit time with the interval between monitoring drawn from QoS specifications) of such a mechanism can be prohibitively high, limiting the efficacy and ultimately ubiquitous deployment of all-optical networks.

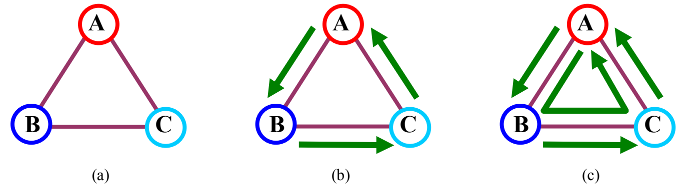

Fig.1. Comparison between diagnosis paradigms of electrical networks and all-optical networks: (a) a 3-node ring network; (b) diagnosis with 3 single-hop tests, and (c) diagnosis with one 3-hop test and three single-hop tests.

In this paper we seek more efficient and elegant methods that greatly lower the hardware and computational complexity and cost of such functions for future all-optical WDM networks. Specifically we develop a family of failure identification algorithms that exploit the unique properties of all-optical networks to reduce the average number of diagnostic probes per unit time. In particular, optical signals will be sequentially sent along a set of lightpaths over an all-optical network to probe its state of health. The network state (i.e., failure pattern) is then inferred from the 'syndromes' of this set of end-to-end measurements. To keep the required number of probes small, each successive probe is dynamically chosen among the set of permissible probes according to the results of the previous tests. Under this generalized model, the traditional diagnosis mechanism based on single-hop probes is then a special case and will be proven to be rather inefficient compared to our designs.

In all-optical networks, the fact that optical signals can be carried over a lightpath of a number of interconnected edges without necessarily being detected by the intermediate nodes allows "multi-hop" tests to probe several edges simultaneously. This technique can be used to greatly reduce the amount of diagnosis effort, as illustrated with the 3-node ring network in Fig. 1. In this example, we assume that each edge fails independently with probability of 0.1. If only 'single-hop' tests are allowed as in Fig. 1(b), the total number of tests to identify all edge states is 3 by employing three single-hop tests (A-B, B-A, C-A). Note that the number of tests required is independent of the edge failure probability. On the other hand, if multi-hop tests are allowed, we can first perform a three-hop test (A-B-C-A) as shown in Fig. 1(c). With a probability of $0.9^3 = 0.729$, we will find out that all edges are fault-free and conclude the diagnosis with only one test. We will resort to the single hop tests only if we know there is at least one failure from the result of the first test, which has a probability of 1-0.729 = 0.271. Thus, on the average, it requires only 0.729 × 1 + 0.271 × (1 + 3) = 1.813 tests to fully diagnose this network. Intuitively, in most cases, the probability that a particular edge has failed is low when network diagnosis is performed; hence it makes sense to test several edges together. Here, reducing the average number of tests required for network diagnosis, which is used in this paper as a measure of the cost, or efficiency, of the diagnosis process, is equivalent to reducing the network diagnosis information bits [14] obtained in the probing sequence.

This example suggests that the fault diagnosis problem can be better understood from an information theoretic prospective. The network state can be viewed as a collection of binary valued random variables; each associated with an edge in the network, indicating failure/no failure on that edge. The objective of a fault diagnosis algorithm is to use a number of tests, whose results, also called the 'syndromes', can be used to uniquely identify the network state. To put it simply, we use probes to dig out all the information hidden in the unknown network state. In the above example, with the single-hop tests, the result of each test is '0' (for no failure) with a probability of 0.9 and '1' (for failure) with a probability of 0.1. Thus the information about the network state contained in this test result is the entropy $H_b(0.1) = 0.469$ bits where $H_b(x)=-x\log_2 x-(1-x)\log_2(1-x)$. In comparison, the three-hop test (A-B-C-A) contains $H_b(0.271) = 0.843$ bits of information. The information contained in a three-hop test is obviously larger than that of the single-hop test, indicating that multi-hop tests are more informative than single-hop tests for this case. As a result, in the second approach, the network state can be identified by a smaller number of probes, or equivalently, the network state is represented by the test syndrome in a more efficient way (this can be understood as encoding the network state with syndromes.) Similar approaches can be used to determine the efficiency of a probe in a general network. More importantly, the design of efficient fault diagnosis algorithm is thus similar to the well-studied source-coding problem, whose goal is to use the minimum average number of bits to represent the source, which is also a collection of random variables.

Applying the above approach under a probabilistic failure model where each edge is assumed to fail independently with a prior failure probability, we obtain the following main results. First, we establish the mathematical equivalence between the fault diagnosis problem and the source-coding problem, which indicates that the minimum average number of probes required is lower bounded by the entropy of the network states. Second, since the sequential diagnosis problem for general network topology is NP-complete, we propose a family of novel near-optimal polynomial time algorithms based on run-length codes [12], whose performance asymptotically approaches the theoretical entropy limit for large networks. Analytical and numerical investigation reveals a guideline for efficient probing schemes: ***each probe should be designed to provide approximately 1-bit of information on the network state and the number of probes required is approximately equal to the information entropy of the network states***. Finally, we apply the family of near-optimum fault probing schemes to different WDM network scenarios. This general approach provides good insights and can be generalized to study other network failure models, including failures on both nodes and edges, correlated failures and transient failures.

This paper is organized as follows. In Section II, we formulate the fault diagnosis problem under a probabilistic edge failure model. In Section III, we establish the source-coding/fault-diagnosis equivalence which suggests a tight lower bound for the minimum average number of probes required, and show an optimal probing scheme for the special case of finding a single failure in the network. In Section IV, we develop the near-optimum run-length probing scheme and analyze its performance in closed-form. In Section V, we compare the performance of the run-length probing scheme to that of the greedy probing scheme within the same framework. In Section VI, we address some practical issues in the implementation of the run-length probing scheme.

II. Fault Diagnosis Problem Formulation

A. Probabilistic Edge 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 of size n, and E is the set of edges of size m. In this paper, we first focus on Eulerian network topologies that have at least one Euler trail [13], which is a sequence of interconnected edges containing all the edges in the topology without repetition. Our results obtained in this paper can also be generalized to non-Eulerian topologies.

In this paper, we characterize the vulnerability of future all-optical WDM networks by the following probabilistic edge failure model:

1. *Nodes are invulnerable(the vulnerable node case will be treated later);*
2. *Edges are vulnerable, and assumed to fail independently with a prior probability of* $p\left(0 \le p \le 1\right)$*;*
3. *We assume that the states of the edges do not change over the duration of the fault diagnosis process, (or only non-ergodic failures occur) [14].*

For a given network topology, we label each edge along an Euler trail with an index, $\beta = 1,2,\ldots,m$. The state of the β^{th} edge is represented by a Bernoulli random variable, F_β, called the *edge state*. We assume for the moment that the edge states $F_\beta, \beta = 1,...,m$ are statistically independent, and identically distributed with $\Pr\left(F_\beta = 1\right) = p$ for an edge failure and $\Pr\left(F_\beta = 0\right) = q\left(=1-p\right)$ for no failure.

We refer to the network state as a realization of the set of edge states $\left\{F_\beta\right\}_{\beta=1}^{m}$, written as $s = f_1 f_2 \cdots f_m \in S = \left\{0,1\right\}^m$. The set of all possible network states is denoted as S. Using

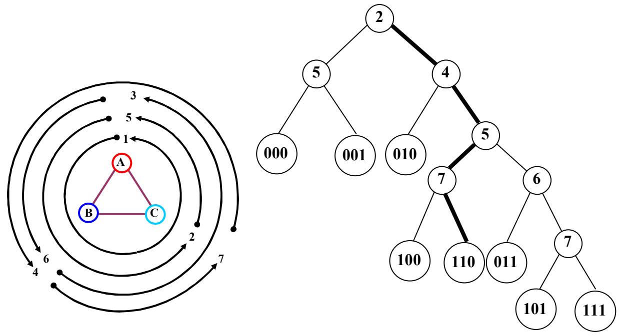

Fig. 2. (a) The set of permissible probes over the 3-node ring topology. Total number of probes is 7. Each probe is indexed with a number near the arrow. (b) A probing scheme (decision tree) for the 3-node ring topology.

the fact that all edges fail independently, we obtain the prior probability of a particular realization of the network state $s = f_1 f_2 \cdots f_m$ as the product of prior probabilities of all edges, i.e.,

$$\Pr(s) = p^{\sum_{\beta=1}^{m} f_\beta} q^{m - \sum_{\beta=1}^{m} f_\beta} . \quad (1)$$

B. Sequential Probing Model

In this paper, we focus on diagnosing network states 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 *permissible probe* t over an Eulerian network topology is a trail (a sequence of adjoined edges without repetition) over the graph. For a finite network, we label each probe with an index $t \in T = \{1, 2, \cdots, |T|\}$ where $|T|$, the cardinality of the set T, is the number of distinct probes over the network. As an example, the 3-node ring topology has 7 permissible probes, as shown in Fig. 2(a).

When an optical signal is sent along a permissible probing trail, the signal either arrives at the destination when all the edges along the probe are ON, or never reaches the destination (or the quality of the signal is unacceptable) when any of the edges along the probe is OFF. The result of each probe is called the probe *syndrome*, denoted as $r_t = 0$ if the probing signal arrives successfully; and $r_t = 1$ otherwise.

A *probing scheme* π is a sequential employment of probes such that any network state can be identified. The successive probe can be sequentially determined according to the syndromes of previous probes. Due to this sequential decision making property, any probing scheme is equivalent to a binary decision tree, whose leaves are network states and inner nodes are probes. For example, a probing scheme for the 3-node ring network is shown in Fig. 2(b), where each inner node is labeled with the probe employed. We adopt the convention that at any inner node, if the probe syndrome is 0 (no failure), the subsequent probe is given in the left child; otherwise if the probe syndrome is 1, the probing process continues on the right child.

The set of all probing decision trees for the topology G is denoted as $\Pi(G)$. Without loss of optimality, we assume that any efficient probing scheme has the following properties:

(a) A probe will not be employed if its syndrome can be inferred from previous syndromes. For example, if a probe returns no failure, it means that no edge in that probe has failed; hence no probe that involves only a subset of these edges is performed thereafter.

(b) When two probes are expected to reveal the same information, the probe with fewer hops is preferred. For example, if the state of an individual edge is known, then one should not start or end a probe with this edge, since dropping it loses no information.

C. Fault Diagnosis Problem

To analyze the additional effort that different probing schemes incorporate into the network management system, we need to associate a probing scheme with some cost metric.

In this paper, each probe $t \in T$, if employed, is assumed to cost one unit of diagnosis effort. Consequently, the probing cost of the state s, denoted by l_s^π, is equal to the number of probes from the root to the leaf node s in the probing decision tree π, called the probing depth of the state s. For example, as shown in Fig. 2(b), the probing depth of state 110 is 4.

Given a probing scheme $\pi \in \Pi(G)$, the average number of probes is

$$\mathcal{L}_\pi = \sum_{s \in S} \Pr(s) l_s^\pi , \quad (2)$$

where l_s^π is the probing depth of network state s and $\Pr(s)$ is the prior probability of this state. We observe that the average number of probes scales with the size of network topologies. In this research, to suppress the scaling effect, we focus on the average number of probes *per edge* which is defined as

$$\overline{\mathcal{L}}_\pi = \frac{1}{m} \sum_{s \in S} \Pr(s) l_s^\pi , \quad (3)$$

where m is the number of edges in the network topology. In this research, we use the average number of probes per edge as the cost metric to design optimum probing schemes.

For a given network topology $G(V, E)$, we want to find the optimal probing scheme that minimizes the average number of probes per edge, and thus to minimize the network monitoring overhead cost. Mathematically, it is formulated as the following optimization problem,

$$\min_{\pi} \ \overline{\mathcal{L}}_\pi = \frac{1}{m} \sum_{s \in S} \Pr(s) l_s^\pi . \quad (4)$$
$$s.t. \quad \pi \in \Pi(G)$$

The resulted minimum average number of probes per edge is written as

$$\overline{\mathcal{L}}^* = \min_{\pi \in \Pi(G)} \left\{ \frac{1}{m} \sum_{s \in S} \Pr(s) l_s^\pi \right\} = \frac{1}{m} \sum_{s \in S} \Pr(s) l_s^{\pi^*} , \quad (5)$$

where π^* is the optimum probing decision tree.

III. Optimum Fault Diagnosis Schemes

In this section, we characterize some properties of the optimal probing schemes for Eulerian networks and the achievable performance. The insights developed in this section will provide guidance for designing near-optimum diagnosis schemes.

A. Source-Coding/Fault-Diagnosis Equivalence

In this sub-section, we first build a connection between the fault diagnosis problem and the source coding problem. This connection inspires the use of the source coding theory to design efficient network diagnosis schemes.

Given a probing scheme $\pi \in \Pi(G)$, we denote the probe syndrome of network state s as $r(s) = r(t_1^s) r(t_2^s) \cdots r(t_{l_s^\pi}^s)$, where l_s^π is the probing depth of state s, and $\{t_1^s, t_2^s, \dots t_{l_s^\pi}^s\}$ is the sequence of probes employed to identify state s. For example, the sequence of probes for state $s = 011$ in Fig. 2(b) is {2, 4, 5, 6} and the probe syndrome is $r(s) = 1110$.

We observe that, given a probing scheme $\pi \in \Pi(G)$, there is a one-to-one correspondence between any network state s and its probe syndrome $r(s)$. Besides, we have the following conceptual mapping between the fault diagnosis problem and the source coding problem:

Network states	⇔	Source symbols
Prior probability of states	⇔	Prior probability of symbols
Probe syndromes	⇔	Coded symbols
Average # of probes	⇔	Average code length

It follows that the set of probe syndromes constitutes a uniquely-decodable code for the set of network states [15], summarized as follows.

Theorem 1 *For any valid probing decision tree $\pi \in \Pi(G)$, the set of probe syndromes $R(\pi) = \{r(s), s \in S\}$ forms a uniquely-decodable code.*

This source-coding/network-probing equivalence has some important implications. First, our design objective, to minimize the average number of probes per edge, is the same as that of the source coding problem: to use the minimum number of coded symbols to represent the source. It follows from the lossless source coding theory [16] that the minimum average number of probes per edge is larger than the information entropy of individual edge, i.e.,

$$\overline{\mathcal{L}}^* \geq H_b(p). \tag{7}$$

Second, inspired by the source coding problem, we observe that it is easier to focus on the diagnosis of large networks, and develop simpler algorithms to find optimal probing schemes that minimize the average number of probes per edge as the network size, or the number of edges m, grows large. Note that this objective is weaker than the original optimization problem (4) because the result applies to large size networks. However, from the engineering prospective, the difference between the two, say, a few more probes in the diagnosis of the moderately sized network, is usually insignificant. On the other hand, the design of such near-optimal schemes can be much simpler.

Finally, this equivalence suggests an information-theoretic approach to translating existing source coding algorithms into efficient network diagnosis schemes. However, this translation is not trivial. In fact, not all source coding algorithms can be transformed into fault diagnosis algorithms. For instance, the well-known optimal source coding algorithm, the Huffman code, is in general not applicable to fault diagnosis problems. Huffman codes can be best understood as a sequence of YES/NO questions, each of which corresponds to an inner node of the code tree. The optimality of the Huffman code comes from the optimal sequence of questions in the form of "Is the source realization in the set A?" Translated into the context of fault diagnosis, this corresponds to questions such as "Is edge 1 UP?" or "Is Edge 1 UP and edge 3 DOWN and edge 5 UP?" Clearly, not all of such questions are physically realizable probes, which can only test consecutive edges and ask questions such as "Are edges 2,3,4 all UP?" Thus, the nature of realizable probes posts an extra restriction that only a special class of questions can be asked. In the rest of this paper, we refer to this restriction as the "*consecutive probing constraint*", and study the fault diagnosis problems, or the equivalent source coding problems, under this constraint.

B. Edge-Wise Probing Schemes

As discussed in the introduction section, the edge-wise probing scheme, by probing each individual edge separately, is in general not optimal, especially when the failure probability of each edge, p, is small. On the other hand, if each edge fails with a high probability (which is unrealistic), the edge-wise probing scheme becomes more efficient. Generalizing from the break-point theorem of the group testing problem [23], we obtain the condition under which the edge-wise probing scheme is optimal.

Theorem 2 *For any non-trivial network topology with $m \geq 2$ edges and at least one path of more than one edge, the edge-wise probing scheme is optimal if and only if the edge failure probability is larger than $(3-\sqrt{5})/2$.*

The theorem suggests that the edge-wise probing schemes based on single-hop tests, as used in the electrical networks and some of the current optical monitoring schemes, are strictly sub-optimal in all-optical networks if $p < (3-\sqrt{5})/2$ (≈ 0.382), which is the situation in most network monitoring scenarios.

According to the theorem, the edge-wise probing scheme is optimal for the case that $p > 1/2$. As p increases to 1, the lower bound $H_b(p)$ decreases, while the optimal approach is always 1 probe per edge. Intuitively, for large values of p, if all the edges fail with a high probability, we could reduce the

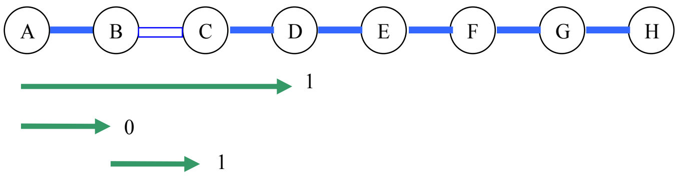

Fig. 3. Optimal probing scheme for the linear network with 7 edges and one faulty edge of BC. The syndrome is 101.

number of probes required if there were a probe to test the scenario where a collection of edges are all in OFF states. Since such a probe cannot be implemented, our information theoretical bound becomes less meaningful in the range of $p > 1/2$. Nevertheless, in almost all practical situations, the edge failure probabilities are small, thus in the remainder of this paper, we always assume $p \le 1/2$.

C. Optimal Probing Schemes for Lightpath with Single Failure

In this sub-section as a special case to illustrate the technique, we focus on a linear network topology (i.e., bus) with h edges and only one faulty edge. In an all-optical network context, this can be understood as the case where there is only one failure along a particular lightpath. Conditioning on the fact that there is one and only one faulty edge, each edge has a uniform distribution of being the faulty one. For this case, the optimum probing scheme to minimize the average number of probes (per edge) has been found in [17].

The optimum probing scheme works as follows. Given that the linear network topology has h edges and the number of faulty edges is one, we first split the path of length h into two sub-paths of length h_l and h_r according to the following criteria:

$$h_l = g(h) = \begin{cases} 2^{\lfloor \log_2 h \rfloor - 1} & \text{if } 2 \le h \le 3 \cdot 2^{\lfloor \log_2 h \rfloor - 1} \\ h - 2^{\lfloor \log_2 h \rfloor} & \text{if } 3 \cdot 2^{\lfloor \log_2 h \rfloor - 1} < h \le 2^{\lfloor \log_2 h \rfloor + 1} - 1 \end{cases}, \quad (8)$$

and $h_r = h - h_l$. (9)

Now we probe the first sub-path of length h_l. If the syndrome is ONE meaning the faulty edge is in the first sub-path, we continue to split the first sub-path according to rule (8) and probe the resulted first sub-path. If the syndrome is ZERO meaning that the first sub-path is fault-free and the faulty edge is in the second sub-path, we split the second sub-path using (8) and probe the resulted first sub-path. This process continues until the faulty edge is located.

Intuitively, when dividing the path into two sub-paths of length h_l and h_r respectively, it is desirable to cut the path into equal halves, thus the probability that the faulty edge is in the first sub-path is as close to ½ as possible. However, such approach is in fact only locally optimal: it makes the probe over the current h_l edges information efficient, but may cause the subsequent probes to be inefficient. Equation (8) says that it is globally optimal to balance the two halves while making sure one of the sub-paths has a length of an integer power of 2. The resulted probing scheme is called the "2^m-splitting" probing scheme. Note that, in both local and global optimums, we are trying to balance the probabilities of syndrome 0 and syndrome 1, indicating that each efficient probe should provide approximately one bit of network state information.

To illustrate the "2^m-splitting" probing scheme, let us consider a linear topology with 7 edges. As shown in Fig. 3, we assume that the 2nd edge (BC) fails. The probing algorithm outputs the syndrome 101 for the network state 0100000.

It is also important to observe that, if the problem is changed into the scenario where there is at least one faulty edge in the linear network, and our objective is to locate the first (leftmost) one, the optimal probing scheme is exactly the same as the one described above, since the algorithm never tests a sub-path without knowing that every edge to the left is fault free. It turns out that this is crucial in developing the run-length probing algorithm in the next section.

IV. Run-Length Probing Algorithms

The optimization problem (4) is, in fact, equivalent to designing an optimal binary decision tree for an equivalent decision problem. In general, the design of optimal binary decision tree has been approached with well-established dynamic programming algorithms [18, 19]. However, it has been shown that the sequential diagnosis problem is Co-NP complete [20], meaning that the computational complexity of probing algorithms grows exponentially with the network size. From a practical point-of-view, we are thus more interested in finding simpler algorithms to design near-optimum probing schemes.

In this section, as a trade-off between complexity and performance, we seek a class of near-optimum probing schemes whose computational complexity is on the polynomial order of network size m. This class of near-optimum probing schemes have probe syndromes consisting of a series of cascaded run-length codes. We show that this probing scheme is asymptotically optimal in that it achieves the minimum average number of probes per edge for large enough networks. Furthermore, the run-length probing algorithm is easy to implement and its performance can be obtained in closed-form.

A. Probing Schemes Based on Run-Length Codes

In Section III.C, we have shown the optimal probing scheme for the scenario where only one failure exists over a linear topology. We have also shown that the same probing scheme is optimal for the scenario where multiple failures exist and the objective is to locate the first (leftmost) failure on the linear topology.

At the same time, given any network state $s = f_1 f_2 \cdots f_m$, it must have the format of $0^{i_1}10^{i_2}1\cdots0^{i_L}1$ where $i_1, i_1, \cdots i_L$ are non-negative integers and 0^i means a run of i '0', and each of

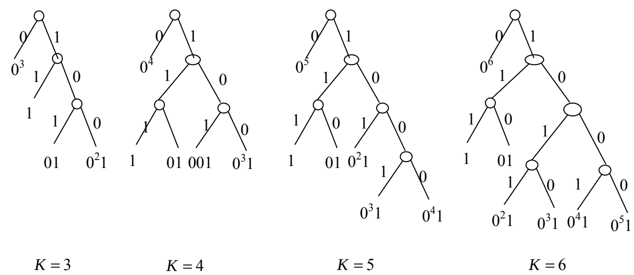

Fig. 4. The run-length probing decision trees for $K=3,4,5,6$, which are also the optimal Huffman code trees for the corresponding intermediate symbol sets Z .

the segments, $0^{i_j}1$, is called a sub-state. In order to encode the network state, it is natural to encode each of such sub-states separately, since any probe can locate at most one faulty edge at a time. This idea suggests that we can, instead of coding for binary input streams, code on the symbols in the set $Z_0=\{0^i1\}_{i=0}^{\infty}$, corresponding to always finding out the first (leftmost) faulty edge along the Euler trail.

The problem of such an approach is that the input alphabet has a size of infinity, which makes it difficult to encode. To this end, it is natural to have an upper limit on the number of edges being considered at a time, that is, to code over the finite alphabet set

$$Z=\left\{0^K,0^i1\right\}_{i=0}^{K-1}, \tag{10}$$

This corresponds to finding the first faulty edge, if any, in K consecutive edges. The prior probability distribution of the possible symbols in Z is given by

$$\Pr\left(0^i1\right)=q^ip,\quad 0\le i\le K-1, \tag{11}$$

and $\Pr\left(0^K\right)=q^K$. (12)

Given this set-up, a simple probing scheme follows. First, a probe is sent over a set of K consecutive edges along the Euler trail. If all the edges are fault-free, we move onto the next set of K consecutive edges along the Euler trail. If on the other hand the first probe says that there is at least one faulty edge in this group, we can employ the "2^m-splitting" approach, described in section III.C to locate the first faulty edge. The process resumes with another group of K edges along the trail right after this faulty edge.

The parameter K is called the *maximum probing length*. Intuitively, for small values of p, or in other words if the edges are more reliable, one should probe more edges at a time, i.e., choose a larger value of K. The optimal value of K can be chosen such that the first probe of K consecutive edges returns ON or OFF with approximately equal probabilities, so that this probe is more information efficient. This can be achieved by choosing K as the unique positive integer satisfying the inequality

$$q^K+q^{K+1}\le 1<q^K+q^{K-1}\ . \tag{13}$$

Solving this inequality, we obtain the maximum probing length as $K=\left\lceil -\log_q\left(1+q\right)\right\rceil$.

It turns out that the probing algorithm developed here is equivalent to the run-length coding procedure for the source-coding problem. For a detailed description of the run-length code, please refer to [12, 21]. For the rest of this paper, we refer to the algorithm above as the *run-length probing scheme*. A detailed description of this algorithm is given below.

Let P_i^j be a path (i.e., a permissible probe) that covers edge i to edge j over the Euler Trail. When *that* path is being probed, it is active. Let h_l denote the number of edges in the current active path, and h_r denote the number of edges in the subsequent active path which is to be probed if all the edges in the current active path are fault-free or the current active path has only one edge and it fails, and i be the start point of the active path. The run-length probing scheme is given by the following algorithm.

Run-Length Probing Algorithm (RLPA):

Step 0:
  Set $i=0$.
Step 1:
  Set $h_l=h_r=K$.
Step 2:
  Probe the path $P_{i+1}^{i+h_l}$;
    If the syndrome $r\left(P_{i+1}^{i+h_l}\right)=0$
      Set $i=i+h_l$, $h_l=h_r$ and $h_r=K$,
      Go to Step 2;
    Else if the syndrome $r\left(P_{i+1}^{i+h_l}\right)=1$
      Set $h=h_l$,
      Set $h_l=g\left(h\right)$ (function $g(\cdot)$ is given by (8),
      Set $h_r=g\left(h-h_l\right)$,
      If $h_l\ge 2$
        Go to Step 2,
      Else if $h_l=1$
        The edge $P_{i+1}^{i+h_l}$ fails,
        Set $h_l=h_r=K$, $i=i+1$,
        Go to Step 2.

To understand the run-length probing scheme pictorially, we illustrate the corresponding probing decision trees for different K's in Fig. 4. Note that, these trees are also the optimal Huffman code trees for the finite symbol set (10) for different K's . It turns out that for the particular set $Z=\left\{0^K,0^i1\right\}_{i=0}^{K-1}$, the Huffman code can in fact be implemented under the consecutive probing constraint. This should not be very surprising since we have already known (a) the Huffman code is always optimal for any given alphabet, (b) the algorithm above is optimal in locating the first faulty edge on a lightpath. The only missing logical step is that the symbol 0^K is always assigned to a length-one codeword, corresponding to a single probe. It can be shown that this is optimal from a coding perspective since K is chosen such that the symbol 0^K is much more likely than the other symbols in Z . In fact,

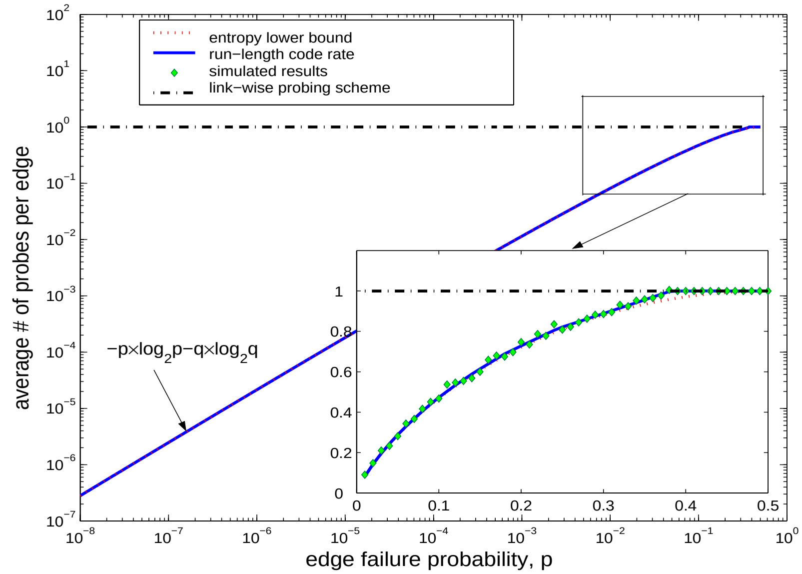

Fig. 5. The simulated average number of probes per edge for an Eulerian topology with 50 edges is compared with the run-length code rate and the entropy lower bound.

one can easily verify that the probability of 0^K is larger than 2/5, which assures the optimality of assigning to it a length 1 codeword [16].

In summary, the run-length probing algorithm is natural for the fault diagnosis problem due to the following two reasons:

1. Each probe can at most locate one faulty edge, thus it makes sense to split the network state into sub-states and locate the faulty edges one-by-one;
2. The probing algorithm can achieve the information theoretical optimum in locating the individual faulty edges.

Note that the run-length probing algorithm is restricted on an Euler Trail of the network, and ignores other connections. In general, this restriction may seriously reduce the set of admissible probes, thus one cannot claim a general optimality of this algorithm over all possible sequential probing schemes. However, this probing scheme is optimal within the class of 'nested' probing schemes where each successive probe includes only a subset of edges from the previous probe or a set of edges that are not tested in the previous probe. Wolf [22] has derived similar results under a totally different context of using group testing approach to resolve the conflict in multi-access communications, and showed that a similar scheme is optimal within the class of nested group testing algorithms [23].

One advantage of the run-length probing scheme is that, it outperforms the greedy algorithm although it has the same minimum computational complexity as the greedy probing scheme [19], which will be addressed in Section V. Another distinguished advantage of the run-length probing scheme is that, with information theoretic insights, one can compare the performance of the run-length algorithm against the entropy lower bound of the global optimum as given in (7). To do that, we will start in the next sub-section by deriving the average number of probes per edge of the run-length probing scheme, again by taking the advantage of the known results on the run-length source codes.

B. Average Number of Probes Required for Run-Length Probing Schemes

The following Lemma characterizes the average number of probes per edge required for run-length probing schemes.

Lemma 1 *The average number of probes per edge required by the run-length probing scheme to fully identify the network state,* $\overline{\mathcal{L}}_{\text{Run-Length}}$*, satisfies*

$$\overline{\mathcal{L}}_{\infty} \le \overline{\mathcal{L}}_{\text{Run-Length}} \le \overline{\mathcal{L}}_{\infty} + \frac{1}{m}, \tag{14}$$

where m *is the total number of edges in the network, and*

$$\overline{\mathcal{L}}_{\infty} \triangleq p \cdot \left(\lfloor \log_2 K \rfloor + 1 + \frac{q^k}{1-q^K} \right), \tag{15}$$

with $K = \lceil -\log_q (1+q) \rceil$ *and* $k = 2^{\lfloor \log_2 K \rfloor + 1} - K$.

This lemma can be proved using the results from the run-length code [21].

We observe that for a reasonably large network, $\overline{\mathcal{L}}_{\infty}$ is independent of the network size (n and m) and is a good approximation of the actual performance. To verify this, the values of $\overline{\mathcal{L}}_{\infty}$ as a function of the edge failure probability $p \in (0, 1/2)$ are plotted in Fig.5, together with the simulated results of the actual number of probes required, $\overline{\mathcal{L}}_{\text{Run-Length}}$, the performance of the edge-wise probing scheme, and the entropy lower bound given in (7), for Eulerian networks with 50 edges. We have two observations from this plot. First, the plot indicates that $\overline{\mathcal{L}}_{\infty}$ is a good approximation of the actual performance of the run-length probing scheme over a broad range of reliability regime. This suggests that for a large Eulerian network ($m \gg K$) we can approximate the average number of probes for the run-length probing scheme as

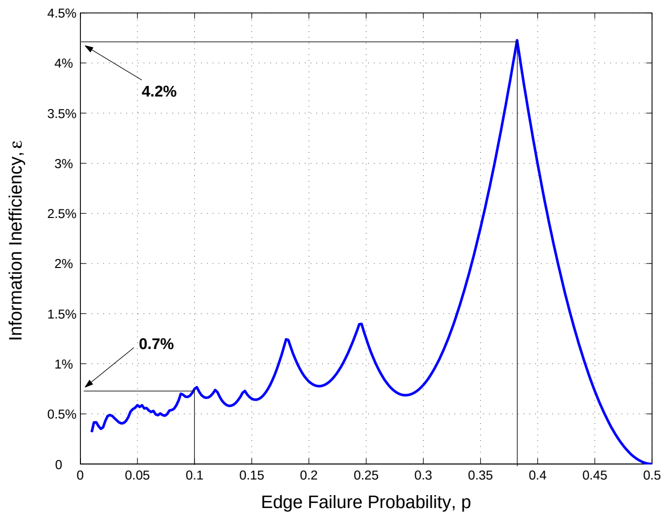

Fig. 6. The information inefficiency of run-length probing schemes for different edge failure probability.

$\mathcal{L}_{\text{run-length}} \approx m \cdot \overline{\mathcal{L}}_{\infty}$.

Second, when the edge failure probability is small (of greater engineering interests), the average number of probes required is close to the entropy lower bound. For example, for an Eulerian network with 1000 edges and edge failure probability $p = 0.01$, the run-length probing scheme requires only 81.05 probes on the average. Compared to the entropy lower bound of 80.79 probes, it requires only an additional number of 0.26 probes. This suggests that even though most of the information of the network topology is ignored, the run-length probing scheme still achieves a near-optimal performance.

To gain a clearer view of this performance, we define the *information inefficiency* of the probing scheme as the ratio between the extra number of probes per edge (compared to the entropy lower bound) and the entropy of each edge, i.e.,

$$\varepsilon(p) = \frac{\overline{\mathcal{L}}_{\pi} - H_b(p)}{H_b(p)}, \tag{16}$$

where $\overline{\mathcal{L}}^{\pi}$ is the average number of probes per edge of any probing decision tree $\pi \in \Pi(G)$. In Fig.6, we plot the information inefficiency of the run-length probing scheme, again with the actual value of $\overline{\mathcal{L}}_{\text{Run-Length}}$ approximated by $\overline{\mathcal{L}}_{\infty}$. As shown in Fig. 6, when the edge failure probability p decreases, the run-length algorithm becomes more efficient, with the fluctuation due to the change of the choice of the maximum probing length K, which takes on only integer values. In particular, if the edge failure probability is less than 0.1, the average number of probes per edge of the run-length probing scheme, for large networks, is upper bounded by

$$\overline{\mathcal{L}}_{\text{Run-Length}} \leq 1.007 H_b(p), \tag{17}$$

which is only 0.7% higher than the entropy lower bound. Moreover, the difference between the achieved performance and the entropy lower bound is uniformly bounded. In the range of $p \in (0, 0.5]$, the worst case occurs at $p = (3-\sqrt{5})/2$, where $\overline{\mathcal{L}}_{\text{Run-Length}} \approx 1.0423 H_b(p)$, (18)

meaning that the actual performance of the run-length probing scheme is less than 5% larger than the lower bound.

Combing (7), (17) and (18), we conclude that the performance of the run-length probing scheme is bounded by the following inequalities,

$$H_b(p) \leq \overline{\mathcal{L}}_{\text{Run-Length}} \leq (1+\varepsilon(p)) H_b(p), \tag{19}$$

where $\varepsilon(p)$ tends to decrease with smaller edge failure probability and we can approximate $\varepsilon(p) < 0.01$ for $p \leq 0.1$ and $\varepsilon(p,q) < 0.05$ for $0.1 < p \leq 0.5$.

Finally, the performance of the run-length probing scheme can be used as an upper bound for the minimum average number of probes per edge. In practical networks whose failure probability is usually small, both the upper bound and the lower bound are reduced to the entropy of individual edge as indicated in Fig. 5, and thus are both tight. The convergence of both upper bound and lower bound to the entropy and the scaling function (19) indicate that the minimum probing effort approximately equals to the entropy of the network states. This actually follows from the fact that, in an efficient probing scheme (e.g., the run-length probing scheme), each probe is designed to provide approximately one bit of state information. Since the amount of unknown information in the network state is equal to the entropy, the number of probes required to identify the network state is on the average equal to the entropy.

C. *Complexity of Run-Length Probing Schemes*

In this subsection, we address the computational advantages of run-length probing schemes including the constant storage space requirement and the polynomial running time.

First, the run-length probing scheme is an online algorithm which choose the successive probes dynamically and does not store the whole probing decision tree beforehand. In fact, the run-length probing scheme only store three variables (i.e., i, h_l and h_r), and reduces the requirement for the storage space to a constant of 3 (i.e., $O(1)$). On the contrary, for a network with m edges, any offline algorithm, which the run-length probing scheme does not belong to, store the whole probing decision tree and thus the requirement for the storage space is $2 \cdot 2^m - 1$ (i.e, $O(2^m)$), which grows exponentially with the network size.

Second, the complexity of the run-length probing scheme is proportional the number of executions of Step 2 in the algorithm, since each execution of Step 2 in the run-length probing algorithm costs roughly a constant time. In the worst case, all the edges fail simultaneously and the run-length probing scheme employs $\log K$ probes to identify each edge failure. This corresponds to executing Step 2 $\log K$ times for each edge failure. For a network with m edges, the total number of executions of Step 2 is at most $m \cdot \log K$, which suggests that the complexity of the run-length probing algorithm is $O(m \cdot \log K)$. It follows that the run-length probing scheme is a polynomial time algorithm.

V. Greedy Probing Algorithms

The information theoretic perspective in the context of fault diagnosis is in general also very useful in understanding, comparing, and improving the network probing schemes based on existing heuristics. In particular, we will in this section study the fault diagnosis approaches based on dynamic programming.

As aforementioned, dynamic programming solutions to the network diagnosis problem (equivalently, the optimal binary decision tree design) are in general NP-complete. As a compromise, various sub-optimal greedy algorithms [25] are proposed based on local optimization heuristics. The performance of such heuristic algorithms is usually studied only via simulations. With our information theoretic viewpoint of the problem, it is natural to connect these problems to their counterparts of source coding problems with dynamic programming approaches, which have been thoroughly studied for decades.

In this section, we will first review the dynamic programming formulation of the network diagnosis problem, with the focus of a particular greedy algorithm that maximizes the local information gain at each stage [25]. We will then compare the performance of this scheme with that of the run-length algorithm to gain more insights.

A. Dynamic Programming and Greedy Algorithms

We first introduce some useful notations. The design of optimal fault diagnosis algorithms is equivalent to the design of optimal binary decision trees. We denote $\mathcal{I}_\pi$ as the set of inner nodes of the decision tree π. Let $\varsigma \in \mathcal{I}_\pi$ denote one of the inner nodes, and P_ς the probability that ς is reached. It follows that P_ς equals to the sum of the prior probabilities of the network states that are descendents of the node ς [26]. Let t_ς be the probe employed at inner node ς, $\Pr(0 \mid \varsigma)$ and $\Pr(1 \mid \varsigma)$ be the probabilities that this test returns 0 and 1, corresponding to the probabilities that the network state lies in the left or right sub-trees of inner node ς, respectively. Furthermore, let $\mathcal{L}_\varsigma$ be the average number of successive probes required when the inner node ς is reached.

Now to design the diagnosis algorithm with the minimum average number of probes, we need to, at each inner node ς, choose a probe t_ς to minimize

$$\mathcal{L}_\varsigma = 1 + \Pr(0 \mid \varsigma) \times \mathcal{L}^*_{\varsigma,o} + \Pr(1 \mid \varsigma) \times \mathcal{L}^*_{\varsigma,1}, \tag{20}$$

where $\mathcal{L}^*_{\varsigma,0}$ and $\mathcal{L}^*_{\varsigma,1}$ are the minimum average number of probes required by the left and right sub-trees from the inner node ς, respectively. In particular, taking ς as the root of the entire tree, the solution of this optimization problem gives the optimal fault diagnosis scheme.

Note that the difficulty of such a problem comes from the fact that the optimization problems at different steps are coupled. In choosing t_ς, one needs to cater for the future optimizations of $\mathcal{L}^*_{\varsigma,0}$ and $\mathcal{L}^*_{\varsigma,1}$. As a result, the computational complexity of this problem grows exponentially with the number of edges m. Some results of using dynamic programming in designing binary decision trees can be found in [18, 19].

Now from an information theoretic perspective, the performance, in terms of the average number of probes, can be computed from the local information efficiencies as follows. For a given probing tree π, the average number of probes required to reach the leaves can be computed as

$$\begin{aligned}\mathcal{L}_\pi &= \sum_{s\in S} \Pr(s) \times \text{(number of probes to reach state } s)\\ &= \sum_{s\in S} \Pr(s) \times \text{(number of ancestors of } s)\\ &= \sum_{\varsigma\in\mathcal{I}_\pi} \Pr(\varsigma) \times 1\end{aligned}, \tag{21}$$

On the other hand, one can write $H(\varsigma)$ as the amount of information in bits, obtained by employing the probe t_ς as node ς is reached, that is,

$$H(\varsigma) \triangleq H_b\left(\Pr(0 \mid \varsigma), \Pr(1 \mid \varsigma)\right). \tag{22}$$

By running this fault diagnosis algorithm, one can always find out the network state, which contains on the average $m \cdot H_b(p)$ bits of information, and can be viewed as the sum of the information obtained in each step, i.e.,

$$m \cdot H_b(p) = \sum_{\varsigma\in\mathcal{I}_\pi} \Pr(\varsigma) \times H(\varsigma). \tag{23}$$

Hence the total inefficiency of the algorithm, in terms of the average number of probes required in excess of the information minimum $m \cdot H_b(p)$ is

$$\mathcal{L}_\pi - m \cdot H_b(p) = \sum_{\varsigma\in\mathcal{I}_\pi} \Pr(\varsigma) \times (1 - H(\varsigma)). \tag{24}$$

where $(1 - H(\varsigma))$ is referred as the *local inefficiency* of the algorithm π at the inner node ς. Intuitively, one probe is used to return only $H(\varsigma)$ bits of information. Hence the difference between the two measures is the information inefficiency of employing this probe, and the weighted sum of the inefficiency over the tree gives the total number of extra probes required by the given probing scheme.

Following such discussions, we observe that in order to design efficient network diagnosis algorithms, it is desirable to minimize the local inefficiency at each stage. Intuitively, by always asking the question to which the answer is completely without bias, one would expect to figure out the network state with fewer questions. This corresponds to making the left and right sub-trees as balanced as possible, i.e., to chose a probe t_ς to minimize

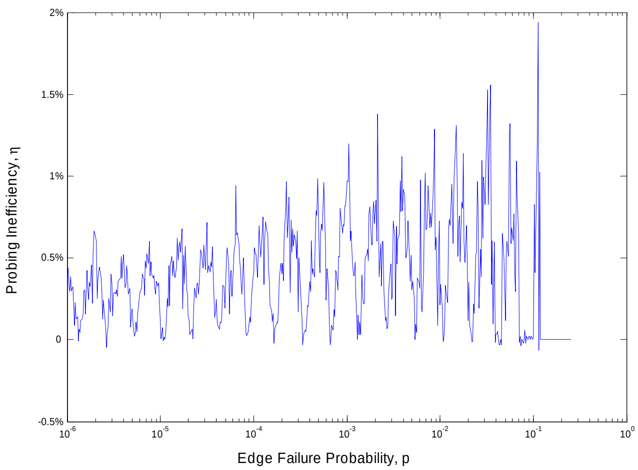

Fig. 7. The performance comparison between the run-length probing algorithm and the greedy probing algorithm.

$$t_\varsigma^* = \arg\min_{t_\varsigma} \left| \Pr(0 \mid \varsigma) - \Pr(1 \mid \varsigma) \right| . \qquad (25)$$

Such intuition of balancing the probabilities of the outcomes of a probe is in general very useful. We have just used that to design the maximum probing length K for the run-length algorithm. For the first probe over K links, the probabilities of UP and DOWN are, respectively, q^K and $1-q^K$. It can be shown that the choice of K in inequality (13) indeed minimizes the difference between these two probabilities.

It is important to note that such an approach, by maximizing the local information gain $H(\varsigma)$, may not necessarily be the globally optimum choice. As an example, in the example of locating a single failure in section III.C, it is globally optimal to split the path as in (8), to make sure that the length of one of the sub-paths is an integer power of 2. On the other hand, a greedy design based on local optimizations would simply split the path into equal halves.

Correspondingly, in diagnosing a network over the Euler Trail, the difference between a greedy algorithm and the run-length scheme is that when it is observed that there is at least one faulty link within a path of length h, a greedy algorithm would continue to probe the left half of this path with length $h_l = \lfloor h/2 \rfloor$, instead of using (8) as in the run-length probing scheme. In this way the local information gain is maximized.

The greedy algorithm presented above to maximize the local information gain is in fact one of many variations [25]. Such algorithms are sometimes preferred due to their conceptual simplicity. However, the run-length algorithm has the same order of computational complexity as these algorithms. In the next sub-section, we will compute the performance of the greedy algorithm based on maximizing local information gains.

B. Performance Comparison

Using Monte Carlo simulation, we simulate the performance of the greedy probing algorithm and compare it with that of the run-length probing algorithm in this sub-section.

To compare these two algorithms in a finer scale, we plot the probing inefficiency of the greedy probing scheme over the run-length probing scheme, i.e.,

$$\eta(p) = \frac{\overline{\mathcal{L}}_{\text{greedy}} - \overline{\mathcal{L}}_{\text{run-length}}}{\overline{\mathcal{L}}_{\text{run-length}}} , \qquad (26)$$

as a function of the link failure probability p in Fig.7. In (26), $\overline{\mathcal{L}}_{\text{run-length}}$ and $\overline{\mathcal{L}}_{\text{greedy}}$ are the average number of probes per link for the run-length probing algorithm and the greedy probing algorithm, respectively.

We observe that for some range of link failure probability, both the run-length probing scheme and the greedy probing scheme have the same average number of probes per link. It can be verified that this happens for all the link failure probabilities such that the maximum probing length K is an integer power of 2. Under this scenario, splitting a path of length K automatically gives sub-paths with lengths as powers of 2, hence the local and global optimums coincide. On the other hand, when the link failure probability is in the range such that K is not an integer power of 2, the greedy algorithms are strictly sub-optimal. As a result, in Fig.7, there is a periodic pattern in the log plot: when p is such that $K(p)$ equals to power of 2, the probing inefficiency is equal to 0; as p increases or decreases such that $K(p)$ does not equal a power of 2, the probing inefficiency is strictly non-zero.

Therefore, although both probing schemes have the same computational complexity, the run-length probing scheme provides some cost-saving over the greedy probing scheme. However, the difference between the two algorithms is quite limited. Intuitively, this is because that the global optimum solution always makes sure that all but one sub-path have lengths of powers of 2, in which case the greedy algorithm is also optimum.

VI. Applications of Run-Length Probing Schemes for All-Optical WDM Networks

In this section, we first address some practical issues in the implementation of the run-length probing scheme, and then apply the run-length probing scheme to different WDM network scenarios.

To employ the run-length probing scheme, we assume for simplicity that a centralized fault management agent communicates with all the network nodes through a reliable out-of-band control channel. In real life this limit is approached even with in-band signaling for fairly reliable networks and when flood routing for signaling is used, as long as the network remains connected. First, the fault management agent processes all the previous probe syndromes to decide the attributes of the successive probe, including the source node, the destination node and the routing path of the probe. Then, the agent signals the source node to send the probe along the specified lightpath to the destination node. After the destination node determines the probe syndrome, it reports the syndrome to the agent through the reliable out-of-band control channel. This process continues until all the edge states have been identified and the entire process is then repeated in a pre-determined or adaptive fashion.

In developing the run-length probing scheme, we have made another assumption that the network is abstracted as an undirected graph, meaning that each optical edge is bidirectional. However, in practical all-optical networks, each optical edge is seldom bidirectional due to the prohibitively high cost of incorporating two-way traffic in a single fiber. Instead, each bidirectional logical edge includes two parallel unidirectional optical fibers with signals in only one direction. To apply the run-length probing scheme to such networks, we replace each edge in the undirected graph with two parallel directed edges. It follows that the undirected network is converted into a directed graph. Under such a conversion, it can be shown that an Euler trail always exists for a directed graph converted from any undirected graph. To diagnose failures, we simply employ the run-length probing scheme over a directed Euler trail in the directed graph. When the maximal probing length is much less than the number of edges, we can approximate the average number of probes per edge requires for a practical network with unidirectional edges with the information entropy of individual directed edge.

Without loss of generality, we assume that there are W wavelengths per fiber in an all-optical WDM network, which is subjected to different kinds of failures. According to the scale of their effect, these failures can be classified into two categories. One category is wavelength-level failures which affect a particular wavelength channel, e.g., transmitter/receiver failures with one dedicated transmitter/receiver per wavelength and single-bandwidth optical filter or single channel frequency selective switch failures. The other category is fiber-level failures which affect all the wavelength channels within an individual fiber, such as fiber cuts, EDFA breakdowns and transmitter/receiver failures in the case of only one tunable transmitter/receiver per fiber (which rarely happens).

We observe that, although that the wavelength-level failure and the fiber-level failure are statistically independent, all the wavelength channels passing through an EDFA fail simultaneously when the EDFA fails. This suggests that failures in different wavelength channels on the same fiber are dependent in that knowing one particular wavelength channel fails reveals some information about the failures of other wavelength channels. Therefore, the fault diagnosis algorithm for practical all-optical WDM networks must consider inter-dependence among failures in different wavelength channels. In particular, the application of the run-length probing scheme over practical all-optical networks depends on the relative dominance between wavelength-level failures and fiber-level failures. In other words, the relationship between p_F (i.e., the prior probability of individual fiber-level failure) and p_W (i.e., the prior probability of individual wavelength-level failure) determines how the run-length probing scheme should be implemented over practical all-optical WDM networks.

In one extreme, for an all-optical network where the wavelength-level failures dominate the fiber-level failures (i.e., $p_w \gg p_F$), we can view the network as a graph where each physical link is represented with W parallel edges and employ the run-length probing scheme over an Euler trail of the resulted graph. We call this the wavelength-level implementation. For a reasonable large number with $m \cdot W \gg K$, the average number of probes required by the run-length probing scheme can be approximated by $W \cdot m \cdot H_b(p_W)$.

In the other extreme, for an all-optical WDM network where the fiber-level failures dominate the wavelength-level failures (i.e., $p_F \gg p_W$), we can view the network as a graph where each physical link is represented with one edge and employ the run-length probing scheme over an Euler trail of the resulted graph. We call this the fiber-level implementation. For a large network of $m \gg K$, the average number of probes required is approximately equal to $m \cdot H_b(p_F)$.

Finally, for an all-optical WDM network subjected to a comparable (in terms of probability of occurrence) mixture of both fiber-level failures and wavelength-level failures, we can still use the fault-diagnosis/source-coding equivalence to obtain a useful lower bound for the minimum average number of probes required as the information entropy of network states, i.e,

$$\mathcal{L}^* \geq m \cdot H(F_1, F_2, \cdots, F_W), \tag{27}$$

where F_i's are dependent random variables indicating states of wavelength channel i's, and $H(\cdot)$ is the information entropy function. The entropy function $H(F_1, F_2, \cdots, F_W)$ can be calculated through the summation of a sequence of conditional entropy functions, i.e.,

$$H(F_1, F_2, \cdots, F_W) = \sum_{i=1}^{W} H(F_i \mid F_1, \cdots, F_{i-1}). \tag{28}$$

However, we are not yet clear whether the entropy lower bound (27) can be achieved, or if achievable, how we can develop probing schemes to achieve this lower bound. Fortunately, in real life either wavelength or fiber integrity but not both at the same time, is the dominant reliability factor.

Actually, we can view the network as a graph where each physical link is represented with W parallel edges and employ the run-length probing scheme over an Euler trail of the resulted graph. The performance of this wavelength-level implementation, which is hard to obtain due to the complicated dependence among failures in different wavelength channels, can certainly serve as an upper for the minimum average number of probes required by an optimal probing scheme. However, since the wavelength-level implementation does not consider the dependencies among failures in different wavelength channels, the run-length probing scheme is not optimum in general. The same conclusion that the information entropy is a lower bound and the run-length probing scheme might not be near-optimum, can be extended to a more general failure model which accommodates dependent failures and/or heterogeneous failures.

VII. Conclusion

In this paper, we investigated the fault diagnosis problem for all-optical WDM networks under a probabilistic edge failure model. Our investigation reveals that the complexity of the fault management system of all-optical networks can be reduced and thus the operational network cost can be kept low. This research can further the understanding of the relationship between the amount of network information gathered and the performance of network management.

Using the average number of probes required as the cost metric, we characterized the optimal fault diagnosis algorithms from an information-theoretical perspective. In particular, the mathematical equivalence between the fault diagnosis problem and the source coding problems suggests that the minimum average number of probes is lower bounded by the entropy of the network states. We also showed that the '2^m-splitting' probing scheme is optimum for the special case of single failure over a linear topology. Based on these heuristics, we developed a class of efficient network diagnosis algorithms, i.e., run-length probing schemes. Its performance is uniformly bounded above by $(1+\varepsilon)H_b(p)$ and converges to the entropy lower bound as the edge failure probability decreases to zero. We also compared the run-length probing scheme to the 'greedy' probing schemes, which indicates that the run-length probing scheme is the algorithm of choice since it outperform the greedy algorithm of the same computational complexity. Several practical issues in implementing the run-length probing scheme in future all-optical WDM networks were also addressed. The guideline for efficient probing schemes is that each probe should provide approximately 1-bit of state information and thus the number of probes required is approximately equal to the information entropy of the network states.

Although this research is done for all-optical networks, full understanding of the network diagnosis problem for all-optical networks can lead to deeper insight into network management under more general models (e.g., transient failures). For future work, the network diagnosis problem with simultaneous edge and node failures needs to be addressed. Other possible future work includes extending the centralized run-length probing scheme into a distributed implementation. Finally, using appropriate cost metrics, these diagnostic approaches can probably be applied to the diagnosis of network failures of other networks including wireless and sensor networks.

Acknowledgment

The authors wish to thank Prof. Muriel Medard, Mr. Eric Swanson, Dr. Poompat Saengudomlert, Dr. Chunmei Liu, Liwei Chen, Jun Sun, and Guy E. Weichenberg for helpful discussions.

References

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

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

[3] Y. G. Wen and V. W. S. Chan, "Ultra-reliable communication over vulnerable all-optical networks via lightpath diversity," to appear on *Journal of Selected Areas of Communication*.

[4] 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.

[5] A. K. Somani, V. K. Agarwal and D. Avis, "A Generalized Theory for System-Level Diagnosis," *IEEE Transactions on Computers*, Vol. c-36, no. 5, May 1987, pp. 538-546.

[6] M. Barborak, M. Malek and A. Dahbura, "The consensus problem in fault-tolerant computing," *ACM Computing Surveys*, Vol. 25, No. 2, June 1993, pp171-220.

[7] S. Lee and K. G. Shun, "Probabilistic diagnosis of multiprocessor systems," *ACM Computing Surveys*, Vol. 26, No. 1, March 1994, pp121-139.

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

[9] D. C. Kilper, R. Bach, D. J. Blumenthal, D. Einstein, T. Landolsi, L. Ostar, M. Preiss and A. E. Willner, "Optical performance monitoring,", *IEEE/OSA Journal of Lightwave Technology*, Vol. 22, No. 1, January 2004, pp. 294-304.

[10] W. G. Yang, "Sensitivity iIssues of optical performance monitoring," *IEEE Photonics Technology Letters*, Vol. 14, No. 1, January 2002, pp. 107-109.

[11] G. Rossi, T.E. Dimmick and D. J. Blumenthal, "Optical performance monitoring in reconfigurable WDM optical networks using subcarrier multiplexing," *IEEE/OSA Journal of Lightwave Technology*, Vol. 18, No. 12, December 2000, pp. 1639-1648.

[12] R. Gallager and D. Van Voorhis, "Optimal source codes for geometrically distributed integer alphabets," *IEEE Transactions on Information Theory*, Vol. IT-21, March 1975, pp. 228-230.

[13] B. Bollobas, Modern *Graph Theory*, Springer-Verlag New York, Inc. 1998.

[14] T. Ho, M. Medard, R. Koetter, ``An information-theoretic view of network management," *INFOCOM*, May 2003.

[15] R. Ahlswede and I. Wegener, *Search Problems*, John Wiley & Sons Ltd., 1987.

[16] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, John Wiley & Sons Ltd., 1991.

[17] E. Wong, "A linear search problem," *SIAM Review*, Vol. 6, No. 2, April 1964, pp. 168-174.

[18] H. J. Payne and W. S. Meisel, "An algorithm for constructing optimal binary decision trees," *IEEE Transactions on Computers*, Vol. C-26, No.9, September 1977, pp. 905-916.

[19] 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.

[20] V. Raghavan and A. Tripathi, "Sequential diagnosability is Co-NP complete," *IEEE Transactions on Computers*, Vol. 40, No. 5, May 1991, pp. 584-595.

[21] H. Tanaka and A. Leon-Garcia, "Efficient run-length encodings," *IEEE Transactions on Information Theory*, Vol. IT-28, No. 6, November 1982, pp. 880-890.

[22] J. K. Wolf, "Born again group testing: multi-access communications", *IEEE Transactions on Information Theory*, Vol. IT-31, No.2, March 1985, pp. 185-191.

[23] P. Ungar, "The cut-off point for group testing," *Comm. Pure Appl*. Math., 13 1960, pp. 49-54.

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

[25] S. R. Safavian and D. Landgerebe, "A survey of decision tree classifier methodology," *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 21, No. 3, May/June 1991, pp. 660-674.

[26] R. W. Yeung, *A First Course in Information Theory*, Kulwer Academic/Plenum Publishers, 2002.