

# Towards Unbiased End-to-End Network Diagnosis

Yao Zhao, Yan Chen, and David Bindel

**Abstract**—Internet fault diagnosis is extremely important for end-users, overlay network service providers (like Akamai [1]), and even Internet service providers (ISPs). However, because link-level properties cannot be uniquely determined from end-to-end measurements, the accuracy of existing statistical diagnosis approaches is subject to uncertainty from statistical assumptions about the network. In this paper, we propose a novel *least-biased end-to-end network diagnosis* (in short, LEND) system for inferring link-level properties like loss rate. We define a *minimal identifiable link sequence* (MILS) as a link sequence of minimal length whose properties can be uniquely identified from end-to-end measurements. We also design efficient algorithms to find all the MILSs and infer their loss rates for diagnosis. Our LEND system works for any network topology and for both directed and undirected properties and incrementally adapts to network topology and property changes. It gives highly accurate estimates of the loss rates of MILSs, as indicated by both extensive simulations and Internet experiments. Furthermore, we demonstrate that such diagnosis can be achieved with fine granularity and in near real-time even for reasonably large overlay networks. Finally, LEND can supplement existing statistical inference approaches and provide smooth tradeoff between diagnosis accuracy and granularity.

**Index Terms**—Internet diagnosis, linear algebra, network measurement.

## I. INTRODUCTION

“WHEN something breaks in the Internet, the Internet’s very decentralized structure makes it hard to figure out what went wrong and even harder to assign responsibility.”—“Looking Over the Fence at Networks: A Neighbor’s View of Networking Research,” by the Committee on Research Horizons in Networking, National Research Council, 2001.

Internet fault diagnosis is important to end-users, overlay network service providers (like Akamai [1]), and Internet service providers (ISPs). For example, with Internet fault diagnosis tools, users can choose more reliable ISPs. Overlay service providers can use such tools to locate faults in order to fix them or bypass them; information about faults can also guide decisions about service provisioning, deployment, and

redirection. For ISPs, diagnosis tools can be used to verify services from provider/peering ISPs and to troubleshoot problems with the physical network.

The modern Internet is heterogeneous and largely unregulated, which renders link-level fault diagnosis an increasingly challenging problem. The servers and routers in the network core are usually operated by businesses, and those businesses may be unwilling or unable to cooperate in collecting the network traffic measurements vital for Internet fault diagnosis. Therefore, most research in this area is focused on end-to-end approaches.

*Internet tomography* is an important approach to fault diagnosis in which link properties [2]–[5] or shared congestion [6] are inferred from end-to-end measurements. In many of these systems, path properties are assumed to be linearly related to link quantities. For example, the latency along a path is the sum of the latencies along the links that make up the paths, and we may write this fact as a matrix equation relating unknown link latencies to measured path latencies. However, as we observed in [7], [8], the linear system is *fundamentally underconstrained*: There exist *unidentifiable links* with properties that cannot be uniquely determined from path measurements.

Because of the challenge of unidentifiable links, existing tomography systems must make certain assumptions in order to infer the property of each link. These modeling assumptions may not always hold in the Internet, and so these systems will have systematic inference errors with nonzero expected value. In other words, the error in the model assumptions prevents the inference error from converging to zero, even when there are many measurements available. We call this modeling error *bias*, and we call the problematic statistical assumptions *biased assumptions*.

In this paper, we advocate a different paradigm for network diagnosis: unbiased diagnosis. Note that there are two *fundamental* statistical assumptions for *any* end-to-end network diagnosis approach:

- End-to-end path properties can be measured accurately.
- The properties of each path are a known linear function of the properties of the link on the path.

Though these assumptions work well in practice (see Section III-A), they may still introduce some bias. However, this is the minimal amount of bias for *any* end-to-end diagnosis scheme. We call these assumptions *basic assumptions*. In this paper, we aim to only use the basic assumptions to achieve *the least biased diagnosis*. We define a *minimal identifiable link sequence* (MILS) as a link sequence of minimal length whose properties can be uniquely identified from end-to-end measurements without bias. We then propose the *least-biased end-to-end network diagnosis* (LEND) system, which uses linear algebra to find MILSs and infer their properties. Our LEND system achieves the finest diagnosis granularity possible without additional assumptions.

Manuscript received July 05, 2006; revised September 12, 2007 and September 03, 2008; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor D. Raz. First published September 29, 2009; current version published December 16, 2009. A subset of this work appears in the *Proceedings of ACM SIGCOMM’06*.

Y. Zhao and Y. Chen are with the Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, IL 60208 USA (e-mail: yzhao@cs.northwestern.edu; ychen@cs.northwestern.edu).

D. Bindel is with the Department of Computer Science, Cornell University, Ithaca, NY 14853 USA (e-mail: dbindel@math.nyu.edu).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TNET.2009.2022158

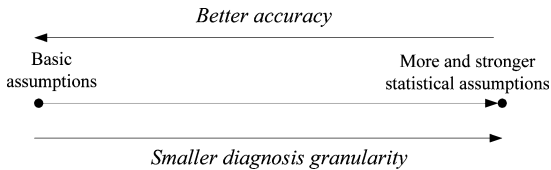


Fig. 1. The spectrum of network diagnosis methods.

When combined with statistical inference, our approach gives a full spectrum of network diagnosis methods with smooth tradeoff between diagnosis accuracy and diagnosis granularity,<sup>1</sup> as shown in Fig. 1. Because of the unidentifiable link problem, LEND cannot infer properties for each link. However, with more and stronger statistical assumptions, we can reduce the diagnosis granularity while introducing more bias and sacrificing diagnosis accuracy.

An interesting problem that we faced when designing the LEND system is that our inference methods behave much differently on directed graphs than they do on undirected graphs. For a simple network with symmetric link properties,<sup>2</sup> the network can be modeled as an undirected graph in which the nodes are connected by undirected links. For networks modeled as undirected graphs, we can use routing information to get the MILSs that are uniquely defined by the inherent path sharing of the networks, and we propose efficient algorithms to find all such MILSs. However, the real Internet has asymmetric link properties (e.g., loss rate), and so must be modeled as a directed graph. However, to find the MILSs in a directed graph is significantly more challenging.

In this paper, we make the following contributions in the algorithm design.

- We advocate the unbiased end-to-end diagnosis paradigm and introduce the concept of MILS.
- Taking a network as a directed graph, when only topology information is used, we prove that each path is a MILS: No path segment smaller than an end-to-end path has properties that can be uniquely determined by end-to-end measurements.
- To address the problem above, we observe that, in practice, there are many good paths with zero loss rates. Then, as a *fact* rather than a statistical assumption, we know all the links on such paths must also have no losses. Based on this observation, we propose a “good path” algorithm, which uses both topology and measurement snapshots to find MILSs with the finest granularity.
- We design efficient algorithms to incrementally update the MILSs and their loss rate estimates when the network topology or overlay measurement nodes change.
- We show that our approach complements other tomography techniques—it helps significantly reduce their complexity and improves their inference accuracy.

We evaluate the LEND system through extensive simulations and Internet experiments. Both give promising results. We define the diagnosis granularity of a path as the average of the

<sup>1</sup>We define *diagnosis accuracy* as the inference accuracy of the properties of the link sequences and *diagnosis granularity* as the length of the smallest consecutive link sequences whose properties are inferred. Formal definitions are in Section VI-A.

<sup>2</sup>A link property between routers  $A$  and  $B$  is symmetric if  $A \rightarrow B$  has the same property as link  $B \rightarrow A$ .

lengths of all the lossy MILSs contained in the path. For the experiments with 135 PlanetLab hosts (each from a different organization), the average diagnosis granularity is only four hops for all the lossy paths. This can be further improved with larger overlay networks, as shown through our simulation with a real router-level topology from [9]. This suggests we can do very fine-level accurate diagnosis with reasonably large overlay networks.

In addition, the loss rate inference on the MILSs is highly accurate, as verified through the cross-validation and IP spoof-based validation schemes. The LEND system is also highly efficient. For the PlanetLab experiments with 135 hosts, the average setup (monitoring path selection) time is 109.3 s, and the online diagnosis of 18 090 paths, 3714 of which are lossy, takes only 4.2 s.

The rest of the paper is organized as follows. In Section II, we survey related work. Then, we define MILSs in Section III, present algorithms to discover MILSs in Section IV, and describe how we validate our approach in Section V. We describe our evaluation of LEND in simulation in Section VI and the evaluation on an Internet testbed in Section VII. We discuss our results in Section VIII and conclude in Section IX.

## II. RELATED WORK

Ping and traceroute are the earliest Internet diagnosis tools, and they are still widely used. However, the asymmetry of Internet routing and of link properties makes it difficult to use these tools to infer properties of individual links. The latest work on network diagnosis can be put into two categories: *pure end-to-end approaches* [2]–[5], [7], [10], [11] and *router-response-based approaches* [12], [13].

### A. Pure End-to-End Approach

Most end-to-end tomography tools fall in one of two classes: tools that are based on temporal correlations among multiple receivers in a multicast-like environment [2]–[4], [10], [11] and tools that impose additional statistical assumptions beyond the linear loss model described in Section III-A [5], [7]. As we discuss below, none of these tools provides *unbiased diagnosis* as defined in Section I. As evidence of the utility of least-unbiased diagnosis, we show in Section VI that our inference is much more accurate than the inference of one statistical tool based on Gibbs sampling introduced in [5].

Under certain assumptions, tools in the first class infer a loss rate for each virtual link (i.e., sequence of consecutive links without a branching point) with high probability. Thus, these tools diagnose failures at the granularity of individual virtual links; obviously, this is a bound on the granularity obtainable by the end-to-end tomography system. Typically, these systems assume an ideal multicast environment, but since true multicast is very rare in the Internet, they use unicast for approximation. Thus, the accuracy of the probe measurements heavily depends on the cross traffic in the network, and there is no guarantee of their accuracy.

As for the second class of tools, the statistically based tools introduced in [5] and [7] use only uncorrelated end-to-end measurements to identify lossy network links. To see why these tools are insufficient, we consider a simple tree topology, shown in Fig. 2. The numbers in the figure are the loss rates of the corresponding paths or links. In this tree, we can only measure

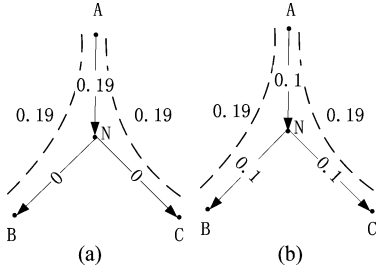


Fig. 2. Example of an underconstrained system.

the loss rates of two paths:  $A \rightarrow B$  and  $A \rightarrow C$ . In the figure, (a) and (b) show two possible link loss rates that lead to the same end-to-end path measurements. The linear programming approach in [5] and SCFS [7] will always obtain the result of (a) because they are biased toward minimizing the number of lossy link predictions, but such results may not be correct. As for the random sampling and Gibbs sampling approaches in [5], either (a) or (b) may be predicted. In fact, none of the loss rates for these three links are identifiable from end-to-end measurements. The LEND system will determine that none of the individual links are identifiable and will get MILSs  $A \rightarrow N \rightarrow B$  and  $A \rightarrow N \rightarrow C$ .

Other than the aforementioned two classes, Shavitt *et al.* use a linear algebraic algorithm to compute some additional “distances” (i.e., latencies of path segments) that are not explicitly measured [14]. The algorithm proposed in [14] has the same function as our link-level diagnosis algorithm in undirected graph model. However, our LEND system incorporates the scalable measurement approach designed in [8] and reuses its outputs to save the computational complexity for link-level diagnosis, and our LEND system is hence both measurement cost-efficient and computation-efficient. More importantly, the Internet should be modeled as a directed graph, in which the algebraic algorithm in [14] fails to do any link-level diagnosis, shown in Theorem 1 (see Section IV-C-1).

### B. Router-Response-Based Approach

All the router-based approaches to network diagnosis are based on response packets sent by interior routers. Unfortunately, interior routers may be unwilling to respond or may respond in an insufficiently informative manner. For example, because many routers implement ICMP filtering or ICMP rate limiting, some ICMP-based tools [12], [13] cannot measure the loss rate on each link. These systems also do not scale well to the task of simultaneously measuring many paths in a large overlay network; furthermore, the accuracy of measurements may be affected by ICMP cross traffic [12].

Tulip, the latest representative of this router-based approach [12], cannot accurately infer the loss rates of links or link sequences because of the following two problems. First, a Tulip probe involves two small ICMP packets and one large UDP data packet. To identify whether the loss happens on the forwarding path or not, Tulip only takes into account the case when only UDP packets are lost. About 40% of the time, a loss involves one of the ICMP packets as well. Tulip simply ignores these cases and consequently underestimates overall loss rates by about 40% [12]. Second, Tulip is sensitive to other simultaneous measurement probes. Tulip requires continuous IP-IDs of replies from the probed router, and it may fail to get accurate

TABLE I  
NOTATION

Symbols	Meanings
$n$	number of end hosts on the overlay
$r = O(n^2)$	number of end-to-end paths
$s$	# of IP links that the overlay spans on
$G \in \{0, 1\}^{r \times s}$	original path matrix
$\bar{G} \in \{0, 1\}^{k \times s}$	a basis of $G$
$k \leq s$	rank of $G$
$l_i$	loss rate on $i$ th link
$p_i$	loss rate on $i$ th measurement path
$x_i$	$\log(1 - l_i)$
$b_i$	$\log(1 - p_i)$
$v$	vector in $\{0, 1\}^s$ (represents path)
$p$	loss rate along a path
$\mathcal{R}(G^T)$	row(path) space of $G$ ( $= \text{range}(G^T)$ )
$G' \in \{0, 1\}^{r' \times s'}$	reduced $G$ after removing good paths & links
$s'$	# of links remaining in $G'$
$r'$	# of bad paths remaining in $G'$
$k' \leq s'$	rank of $G'$
$G''$	reduced $\bar{G}$ after removing good paths & links
$\bar{G}'$	a basis of $G''$ , also a basis of $G'$
$Q', R'$	QR decomposition of $\bar{G}'^T$ . $\bar{G}'^T = Q'R'$

loss rate if other measurements (e.g., another instance of Tulip) probe the router at the same time.

## III. MODELS AND ARCHITECTURE

In this section, we briefly describe the algebraic model and the system architecture of the LEND system. The algebraic model is widely used in Internet tomography and other measurement works [3], [8], [14]. However, the techniques for diagnosis require a significant amount of extra design over this framework, as we will describe in the paper—e.g., the MILSs introduced in Section IV-A.

### A. Algebraic Model

Here, we briefly introduce the algebraic model that is widely used in network diagnosis. For easy indexing, all the important notations in the paper can be found in Table I.<sup>3</sup> Suppose an overlay network spans  $s$  IP links. We represent a path by a column vector  $v \in \{0, 1\}^s$ , where the  $j$ th entry  $v_j$  is one if link  $j$  is part of the path, and zero otherwise. Suppose link  $j$  drops packets with probability  $l_j$ . Then, the loss rate  $p$  of a path represented by  $v$  is given by

$$1 - p = \prod_{j=1}^s (1 - l_j)^{v_j}. \quad (1)$$

In the equation above, we assume that packet loss is independent among links. We believe that such an assumption is supported by the findings of Caceres *et al.* [16]. They find that the diversity of traffic and links makes large and long-lasting spatial link loss dependence unlikely in a real network such as the Internet [16]. Our Internet experiments also show that the link loss dependence has little effect on the accuracy of (1). Formula (1) has also proven useful in other work on link/path loss inference [4], [5], [17], [18].

Suppose we take logarithms on both sides of (1). Then, by defining a column vector  $x \in \mathbb{R}^s$  with elements

<sup>3</sup>The standard algebraic terms in Table I can be found in [15].

$x_j = \log(1 - l_j)$ , and writing  $v^T$  as the transpose of the vector  $v$ , we can rewrite (1) as follows:

$$\log(1 - p) = \sum_{j=1}^s v_j \log(1 - l_j) = \sum_{j=1}^s v_j x_j = v^T x. \quad (2)$$

There are  $r = O(n^2)$  paths in the overlay network, thus  $r$  linear equations of the form (2). Putting them together, we form a rectangular matrix  $G \in \{0, 1\}^{r \times s}$  that represents these paths. Each row of  $G$  represents a path in the network:  $G_{ij} = 1$  when path  $i$  contains link  $j$ , and  $G_{ij} = 0$  otherwise. Let  $p_i$  be the end-to-end loss rate of the  $i$ th path, and let  $b \in \mathbb{R}^r$  be a column vector with elements  $b_i = \log(1 - p_i)$ . Then, we write the  $r$  in form (2) as

$$Gx = b. \quad (3)$$

Normally, the number of paths  $r$  is much larger than the number of links  $s$ . However, in general,  $G$  is rank deficient: i.e.,  $k = \text{rank}(G)$  and  $k < s$  [8]. In this case, we will be unable to infer the loss rate of some links from (3). These links are also called *unidentifiable* in the network tomography literature [7]. Fig. 2 shows an example in which no link is identifiable.

### B. System Architecture

Suppose  $n$  end-hosts belong to a single overlay network or to a confederation of overlay networks. They cooperate to share an overlay monitoring and diagnosis service and are instrumented by a central authority [e.g., an overlay network center (ONC)] to measure the routing topology and path loss rates as needed. First, the end-hosts measure the topology and report to the ONC, which selects a small number of paths to measure and instruments the end-hosts to execute such measurements. The end-hosts periodically report the measured loss rates to the ONC. Then, the ONC infers the loss rates of nonmeasured paths and locates the congestion/failure points. Applications can query the ONC for the loss rate or diagnosis of any path, or they can set up triggers to receive alerts when the loss rates of paths/links of interest exceed a certain threshold.

The core part of the LEND system consists of two stages. In the first stage, we select a small amount of paths to measure and then infer the loss rates of all end-to-end paths. The first stage is done in our previous work [8]. The basic idea of [8] is to find a minimal set of paths so that all the path properties can be inferred based on the measurement of the set of paths. Specifically, modeling with the algebraic model introduced in Section III-A, a path set that corresponds to a basis of the row(path) space of path matrix  $G$  ( $\mathcal{R}(G^T)$ ) satisfies our purpose. It is much more challenging to infer the properties on the *link* level in the second stage, which is our focus in this paper. Nevertheless, for overlay diagnosis, we naturally inherit the scalability and load balancing from [8] with this architecture. That is, to diagnose an overlay network of  $n$  nodes, we only need to measure  $O(n \log n)$  paths instead of all the  $O(n^2)$  paths. This load is evenly distributed across the end-hosts.

## IV. IDENTIFYING MILSS

In this section, we define a *minimal identifiable link sequence* (MILS) and introduce algorithms to identify MILSs and to infer

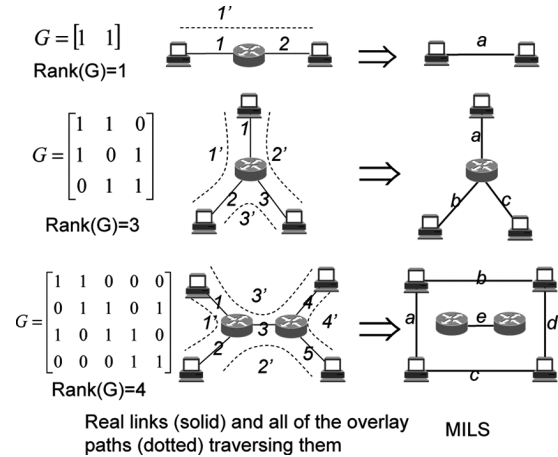


Fig. 3. Sample topologies and MILSs.

their properties. For simplicity, we first study link property inference for undirected graphs. We then turn to the more realistic problem of inferring link properties in directed graphs.

### A. Minimal Identifiable Link Sequence

As mentioned before, we know that not all the links (or the corresponding variables in the algebraic model) are uniquely identifiable. Thus, our purpose is to find the smallest path segments with loss rates that can be uniquely identified through end-to-end path measurements. We introduce *MILS* to define such path sequences. These path sequences can be as short as a single physical link or as long as an end-to-end path. Our methods are unbiased and work with any network topology. This provides the *first* lower bound on the granularity at which properties of path segments can be uniquely determined. With this information, we can accurately locate what link (or set of links) causes any congestion or failures.

Fig. 3 illustrates some examples for undirected graphs. In the top figure, we cannot determine the loss rates of the two physical links separately from one path measurement. Therefore, we combine the two links together to form one MILS. In the middle figure, three independent paths traverse three links. Thus, each link is identifiable, and thus each link is a MILS. In the bottom figure, there are five links and four paths. Each path is a MILS since no path can be written as a sum of shorter MILSs. However, link 3 can be written as  $(2' + 3' - 1' - 4')/2$ , which means link 3 is identifiable, and there are *five* MILSs. These examples show three features of the MILS set:

- The MILSs may be linearly dependent, as in the bottom example. We can shrink our MILS set to a basis for the path space by removing such linear dependence, e.g., by removing the MILS  $c$  in the bottom example in Fig. 3. However, it is helpful to keep such links for diagnosis.
- Some MILSs may contain other MILSs. For instance, MILS  $e$  is contained in MILSs  $b$  and  $c$  in the bottom example.
- The MILS is a *consecutive* sequence of links because, for diagnosis purposes, we often want to limit the range within the network where congestion/failure happens.

The problem of decomposing a network topology into MILSs is similar to the sparse basis problem in numerical linear algebra.

The sparse basis problem is to find a basis for the range of a matrix with as few nonzeros as possible. However, finding MILSs differs from the usual problem of finding a sparse basis for the following reasons:

- The sparse basis problem is an NP-hard problem, and nearly all the heuristic algorithms for this problem are based on a *nondegeneracy* assumption. In particular, these heuristics require that every submatrix of  $G$  with the order of  $\text{rank}(G)$  is nonsingular [19], an assumption does not hold for typical network path matrices.
- For Internet diagnosis, we want to locate the possible lossy links in a networking region that is as small as possible. Thus, we want to have vectors that correspond to consecutive link sequences. If we did not make this assumption, there could exist an exponentially large number of MILSs.

A MILS is a path segment and, like a path, it can be represented by a vector in  $\{0, 1\}^s$  whose nonzero entries denote the physical links used. Our requirement that the properties of MILSs must be determined by the end-to-end measurements is equivalent to the requirement that the vector  $v$  of the MILS is in the path space  $\mathcal{R}(G^T)$ . Compared to finding a basis of  $\mathcal{R}(G^T)$  made of end-to-end paths, which was addressed in [8], identifying MILSs is a more challenging task.

### B. MILSs in Undirected Graphs

As we have defined them, MILSs satisfy two properties: 1) they are minimal—i.e., they cannot be decomposed into shorter MILSs; and 2) they are identifiable—i.e., they can be expressed as linear combinations of end-to-end paths. Algorithm 1 finds all possible MILSs by exhaustively enumerating the link sequences and checking each for minimality and identifiability. An identifiable link sequence on a path will be minimal if and only if it does not share an endpoint with a MILS on the same path. Thus, as we enumerate the link sequences on a given path in increasing order of size, we can track whether each link is the starting link in some already-discovered MILS, which allows us to check for minimality in constant time. To test whether a link sequence is identifiable, we need only to make sure that the corresponding path vector  $v$  lies in the path space. Since  $Q$  is an orthonormal basis for the path space,  $v$  will lie in the path space if and only if  $\|v\| = \|Q^T v\|$ .

Now, we analyze the computational complexity of identifying MILSs. If a link sequence contains  $i$  links, then  $v$  will contain only  $i$  nonzeros, and it will cost  $O(i \times k)$  time to compute  $\|Q^T v\|$ . This cost dominates the cost of checking for minimality, and so the overall cost to check whether one link subsequence is a MILS will be at worst  $O(i \times k)$ . On a path of length  $l$ , there are  $O(l^2)$  link subsequences, each of which costs at most  $O(l \times k)$  time to check, so the total time to find all the MILSs on one end-to-end path is at most  $O(k \times l^3)$ . However, we can further reduce the complexity from  $O(k \times l^3)$  to  $O(k \times l^2)$  using dynamic programming (detail omitted). If we check every end-to-end path in the network, the overall complexity of Algorithm 1 will then be  $O(r \times k \times l^2)$ . However, our simulations and Internet experiments show that only a few more MILSs are obtained from scanning all  $r$  end-to-end paths than from scanning only the  $k$  end-to-end paths that are directly monitored. Furthermore, each physical link used by the network will be used by one of the  $k$  monitored paths, so the MILSs obtained from this smaller set of paths do cover every physical link. Therefore,

#### procedure Seek\_MILS

```

1 Let  $Q$  be an orthonormal basis of  $\mathcal{R}(G^T)$  which is pre-
  computed as in [8];
2 foreach path  $p$  in  $G$  do
3   start_mils := logical array of length( $p$ );
4   Clear start_mils to all false;
5   for  $i := 1$  to length( $p$ ) do
6     foreach segment  $S = p_k \dots p_l$  of length  $i$  do
7       if start_mils( $k$ ) then
8         continue;
9       else
10        Let  $v$  be the corresponding vector of  $S$ ;
11        if  $\|Q^T v\| = \|v\|$  then
12          start_mils( $k$ ) := true;
13           $S$  is a MILS;
14        else
15           $S$  is not a MILS;
16        end
17      end
18    end
19  end

```

Algorithm 1. Seeking All MILSs in an Undirected Graph.

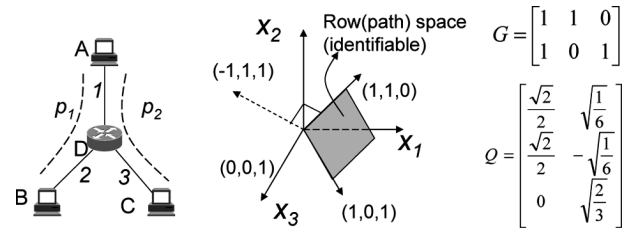


Fig. 4. MILSs in undirected graph.

in practice, we scan only the  $k$  monitored paths, which costs  $O(k^2 \times l^2)$  time, and we accept a slight loss of diagnosis granularity.

Once we have identified all the MILSs, we need to compute their loss rates. We do this by finding a solution to the underdetermined linear system  $\bar{G}x_G = \bar{b}$  system (see [8]). For example, in Fig. 4,  $x_G = \left(\frac{2x_1+x_2+x_3}{3}, \frac{x_1+2x_2-x_3}{3}, \frac{x_1-x_2+2x_3}{3}\right)^T$ . Obviously,  $x_G$  shows some identifiable vectors in  $\mathcal{R}(G)$ , but they may not be MILSs. Then, for each MILS with vector  $v$ , the loss rate is  $v^T x_G$ . The elements of  $x_G$  need not be the real link loss rates: Only the inner products  $v^T x_G$  are guaranteed to be unique and to correspond to real losses. We also note that because loss rates in the Internet remain stable over time scales on the order of an hour [20], the path measurements in  $\bar{b}$  need not be taken simultaneously.

It is worth mentioning that the same problem for undirected graph was solved in [14] with the same order of computational complexity. However, our focus of this paper is on the case of directed graphs, which is ignored in [14]. Furthermore, compared to [14], our approach inherits the key feature of measurement efficiency of [8] (i.e., requiring only  $O(n \log n)$  measurements of end-to-end paths instead of  $n^2$  measurements) and reuses the computational output of [8] such as  $x_G$  and  $Q$ .

### C. MILSs in Directed Graphs

1) *Special Properties for Directed Graphs:* Surprisingly, our MILS algorithm cannot be extended to directed graphs directly.

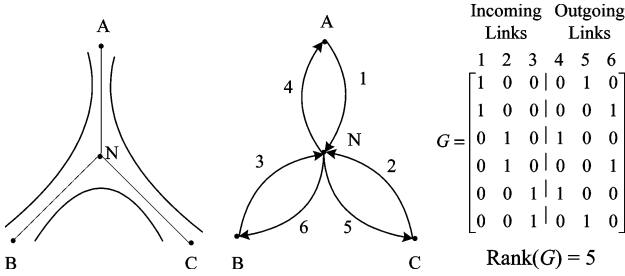


Fig. 5. Undirected graph versus directed graph.

We found that no path can be decomposed into more than one MILS, i.e., each path itself is a MILS. Fig. 5 shows a simple star topology as both an undirected graph and a directed graph. In the undirected graph on the left, the loss rate of each link is identifiable from the loss rate of the three paths. In contrast, in the directed graph on the right,  $\text{rank}(G) = 5$ , and none of the six links are identifiable from measurements of the six end-to-end paths. Only the end-to-end paths are identifiable in this case. This is typical of directed networks. In the case illustrated in Fig. 5, we can explain the lack of identifiable links as follows. We can split  $G$  into two submatrices, one containing only incoming links and the other only containing outgoing links of the router  $N$ . Thus, any vector  $v = [v_1, v_2, v_3, v_4, v_5, v_6]^T \in \mathbb{R}^6$  in  $\mathcal{R}(G^T)$  satisfies  $v_1 + v_2 + v_3 = v_4 + v_5 + v_6$  because any path in  $G$  has one incoming link *and* one outgoing link. Vectors like  $[1\ 0\ 0\ 0\ 0\ 0]^T$  do not belong to  $\mathcal{R}(G^T)$ , as they do not satisfy that condition. This example illustrates the intuition of Theorem 1 below, which shows that, in a directed graph, each path itself is a MILS—i.e., it is the *minimal identifiable consecutive path segment*.

**Theorem 1:** In a directed graph, no end-to-end path contains an identifiable subpath except loops.

*Proof:* For any interior node  $N$  in the network, define vectors  $u^N \in \{-1, 0, 1\}^s$  such that  $u_i^N = 1$  if link  $i$  is an incoming link for node  $i$ , and  $u_i^N = -1$  if link  $i$  is an outgoing link for node  $i$ . For any path with vector  $v$ ,  $v^T u^N$  is nonzero iff the path begins or ends at  $N$ . Because  $N$  is an internal node and the rows of  $G$  all correspond to end-to-end paths,  $G u^N = 0$ .

Any identifiable link sequence in the network can be represented by a vector  $x$  such that  $x = G^T z$  for some  $z$ ; for such a link sequence,  $x^T u^N = z^T G u^N = 0$ ; therefore,  $x$  cannot correspond to a link sequence with an endpoint at  $N$ . Thus, no identifiable link sequence may have an endpoint at an interior network node. This means that the only identifiable link sequences are loops and end-to-end paths.  $\square$

Routing loops are rare in the Internet; thus, Theorem 1 says that each path is a MILS and there are no others. This means that there are no individual links or subpaths whose loss rates can be exactly determined from end-to-end measurements. Next, we will discuss some practical methods to get finer level unbiased inference on directed graphs, such as the Internet.

2) *Practical Inference Methods for Directed Graphs:* Considering the simple directed graph in Fig. 5, the problem of determining link loss rates is similar to the problem of breaking a deadlock: If any of the individual links can be somehow measured, then loss rates of all other links can be determined through end-to-end measurements. Since link loss rates cannot be negative, for a path with zero loss rate, all the links on that path must

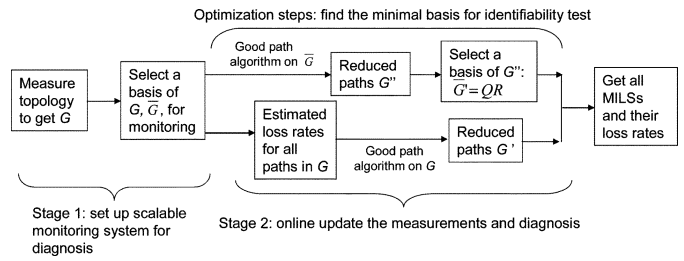


Fig. 6. The operational flowchart of the LEND system architecture.

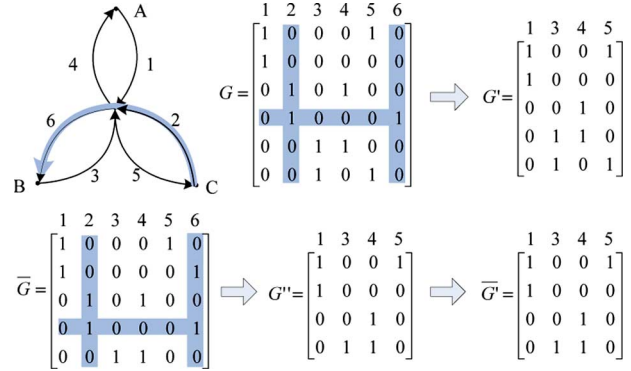


Fig. 7. Examples showing all the matrices in the flowchart.

also have zero loss rates. This can break the deadlock and help solve the link loss rate of other paths. We call this inference approach the *good path algorithm*. Note that this is a *fact* instead of an extra assumption. Our PlanetLab experiments as well as [20], show that more than 50% of paths in the Internet have no loss.

In addition, we can relax the definition of “good path” and allow a negligible loss rate of at most  $\sigma$  (e.g.,  $\sigma = 0.5\%$ , which is the threshold for “no loss” in [20]). Then, we again have a tradeoff between accuracy and diagnosis granularity, as depicted in our framework. Note that although the strict good path algorithm cannot be applied to other metrics such as latency, such bounded inference is generally applicable.

As illustrated in the second stage of Fig. 6, we identify MILSs in directed graphs in two steps. First, we find all the good paths in  $G$  and thus establish some good links. We remove these good links and good paths from  $G$  to get a submatrix  $G'$ . Then, we apply Algorithm 1 to  $G'$  to find all lossy MILSs and their loss rates in  $G$ . For the good links that are in the middle of lossy MILSs identified, we add them back so that MILSs are consecutive. In addition, we apply the following optimization procedures to get  $Q$  quickly for the identifiability test (step 10 of Algorithm 1).

We remove all the good links from  $\bar{G}$  and get a smaller submatrix  $G''$  than  $G'$ . By necessity,  $G''$  contains a basis of  $G'$ . We can then use the small matrix  $G''$  to do  $QR$  decomposition and thus get  $Q'$ . Since  $G''$  is usually quite small even for  $G$  from a reasonably large overlay network, this optimization approach makes LEND very efficient for online diagnosis. In Fig. 7, we use a simple topology to show the matrices computed in the whole process. The path from  $C$  to  $B$  is a good path, and thus links 2 and 6 are good links.

#### D. Dynamic Update for Topology and Link Property Changes

During monitoring, good links may become lossy and vice-versa, routing paths between end-hosts may change, and hosts may enter or exit the overlay network. These changes may result in changes to the reduced matrix  $G'$ , forcing us to recompute the MILSs and their loss rates. We perform this recomputation in two steps: We first incrementally update the decomposition of the  $G'$  matrix, and then we compute the MILSs and their properties using the algorithm described in Section IV-B.

We express changes to  $G$  and  $G'$  in terms of four kinds of primitive updates: adding a bad path, deleting a bad path, adding a good path, and deleting a good path. Any more complicated change can be expressed in terms of these four operations. For example, if the routing tables changes so that some bad paths are rerouted, we would delete the original bad paths from the system and add the routes for the new good paths. When a bad path is added or deleted, there may be one row that is added to or removed from  $G'$ ; similarly, when a good path is added or deleted, the set of links identified as good by the good path algorithm may change, so that a few columns are added to or removed from  $G'$ . To update a QR decomposition of  $G'$  after one column or row update costs time proportional to the size of the matrix, or  $O(k' \times s')$  time (see the discussion in [21, Section 4.3]), and since at most  $l$  rows or columns are affected by one of our primitive updates, the total cost of such updates is at most  $O(l \times k' \times s')$ . This cost is much less expensive than the initial QR factorization of  $G'$ , which costs  $O(r' \times k' \times s')$ .

In Section VII-B4, we show that it takes only a few seconds to complete an incremental update to  $Q'$  and  $R'$  and reidentify the MILSs. Given that end-to-end Internet paths tend to be stable on the time scale of a day [22] and link loss rates remain operationally stable on the time scale of an hour [20], our algorithm should suffice for online updates and diagnosis.

#### E. Combining With Statistical Diagnosis

As discussed before, the linear system is underconstrained, and so there exist some unidentifiable links. With MILSs, we attempt to discover the smallest path segments for which properties can be uniquely identified. However, there are various statistical methods that produce estimates of properties at a finer granularity, e.g., at the virtual link level (see Section II-A for definition). Essentially, these methods use statistical assumptions to resolve the likely behavior in the unmeasured space discussed in Section III-A and therefore provide only possible estimates as shown in Fig. 2 [5].

Because of this, our LEND approach and other statistical methods can complement each other nicely. For example, we can discover some links or link segments that are lossy by the least-unbiased approach. If the user wants to make predictions at a finer level of granularity with potential degradation of accuracy, we can further apply the statistical algorithms *on the lossy MILSs*. In comparison with the traditional statistical tomography that has to consider the whole path, our scheme can help significantly reduce *complexity* without losing inference accuracy by considering a subset of the links. Our MILSs are vectors in  $\mathcal{R}(G^T)$ , and the MILS set contains a basis of  $\mathcal{R}(G^T)$ . Thus, inference with MILSs is equivalent to inference with the whole end-to-end paths.

Take the linear optimization and Bayesian inference using Gibbs sampling introduced in [5], for example; these algorithms

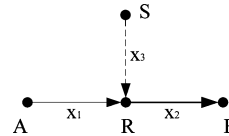


Fig. 8. IP spoofing example.

can be used without modification on our MILS set rather than on the original end-to-end paths. Section VI-C6 shows that Gibbs sampling inference combined with our least-unbiased approach improves its accuracy. In addition, the computational complexity of Gibbs sampling inference based on the MILS set is dramatically reduced because the input “paths” are much shorter than the whole end-to-end paths.

#### V. DIAGNOSIS VALIDATION THROUGH IP SPOOFING

Internet diagnosis systems are difficult to evaluate because of the general lack of ground truth—it is very hard, if not virtually impossible, to obtain the link level performance from the ISPs. We will first evaluate the system through simulations in Section VI. Then, we test LEND on the real Internet in Section VII. For validation on the real Internet, in addition to the classical cross validation, we need a more powerful approach. As shown in Section II, existing router-based diagnosis tools like Tulip are neither very accurate nor scalable and, therefore, do not suit our needs. In this section, we propose an IP-spoofing-based mechanism for link-level diagnosis validation.

Though IP spoofing is usually used by malicious hackers to hide their identities, it also is a useful tool to cope with the rigid routers. For example, IP spoofing is used to help measure ICMP generation time in routers [23]. We use IP spoofing to obtain a limited source routing, which helps validate the accuracy of MILSs. With this technique, we can measure the properties of new paths that we could not normally probe. These additional measurements are then used to validate the inferred loss rates of MILSs.

Fig. 8 shows an example of how to use IP spoofing to “create” a new path. Each line in the figure can be a single link or a sequence of links. For simplicity, we just call it a link in this section. Assume router  $R$  is on the path from the node  $A$  to node  $B$ , and the path from  $S$  to  $B$  does not go via  $R$ . To create a new path  $S \rightarrow R \rightarrow B$ ,  $S$  sends an ICMP ECHO request packet to  $R$  with spoofed source IP as  $B$ . When the packet reaches router  $R$ ,  $R$  will generate an ICMP ECHO reply packet and send it to  $B$ . Thus, we get a path from  $S$  to  $B$  via router  $R$ . Assume  $x_i$  is the logarithm of the success rate of link  $i$  as defined before and  $b_B$  is the logarithm of the success rate of path  $S \rightarrow R \rightarrow B$ . Thus, we have  $x_2 + x_3 = b_B$ . Since  $x_3 \leq 0$ , we get a lower bound of  $x_2$ , i.e.,  $x_2 \geq b_B$ . For validation, we use the source routing capability we have created to measure some new paths and check whether they are consistent with the MILSs and their inferred loss rates obtained from normal non-IP-spoofed measurements. For example, normal measurements on path  $A \rightarrow B$  reveal that there is a single lossy MILS  $l$  on  $R \rightarrow B$ , then the logarithm of  $l$ 's success rate should be bounded by  $b_B$  as discussed before. See details in Section VII-B2, where the consistency checking idea is also used in cross validation.

The principle of IP-spoofing-based source routing is simple. However, many practical problems need to be addressed.

- First, most edge routers check outgoing packets and disable IP spoofing from the internal networks. In addition, all PlanetLab hosts are disabled from IP spoofing. However, we were able to get one host, our institution, exempted from such filtering.
- Second, as with other router-based diagnosis approaches [12], our scheme is subject to ICMP rate-limiting on routers for measuring the loss rates. We filter those routers with strict ICMP rate-limiting.

## VI. EVALUATION WITH SIMULATION

In this section, we present our evaluation metrics, simulation methodology, and simulation results.

### A. Metrics

The metrics we have used to evaluate our algorithms include the granularity of diagnosis, MILS loss rate estimation accuracy, and the speed of setup and online diagnosis.

Of these metrics, the first one, diagnosis granularity, is particularly important. For diagnosis, we focus on the lossy paths and examine how many links we suspect could be the cause of the network congestion/failures. We define the diagnosis granularity of a path as the average of the lengths of all the lossy MILSs contained in the path. The diagnosis granularity of an overlay network is defined as the average diagnosis granularity of all the lossy paths in the overlay. For example, suppose an overlay network has only two lossy paths: one path has two lossy MILSs of length 2 and 4 separately, and the other lossy path consists of only one lossy MILS of length 3. Then, the diagnosis granularity for the overlay is  $((2 + 4) / 2 + 3) / 2 = 3$ . We measure lengths in terms of physical links, except when we compare to other approaches for which the natural unit of length is a virtual link.<sup>4</sup>

We call a MILS as lossy (or bad) if its loss rate exceeds 3%, which is the threshold between “minor loss” and “perceivable loss” (like “tolerable loss” and “serious loss”) as defined in [20]. As we mentioned in Section VI-B, we call a path “good” if it has less than 0.5% loss, which is the threshold for “no loss” in [20]. Because we allow “good” paths to have nonzero loss, the good path algorithm introduces some error. The question is whether this error will accumulate in a serious way when we compute the loss rates for MILSs. If the error does not accumulate, we can simply adjust the threshold of what we consider to be a “good” path in order to trade higher accuracy for better granularity and faster computations.

For each MILS, we evaluate the error of the inferred loss rate  $\hat{p}$  compared to the real loss rate  $p$  by analyzing both the absolute error  $|p - \hat{p}|$  and the error factor  $F_\varepsilon(p, \hat{p})$ , which is defined in [4] to be

$$F_\varepsilon(p, \hat{p}) = \max \left\{ \frac{p(\varepsilon)}{\hat{p}(\varepsilon)}, \frac{\hat{p}(\varepsilon)}{p(\varepsilon)} \right\} \quad (4)$$

where  $p(\varepsilon) = \max(\varepsilon, p)$  and  $\hat{p}(\varepsilon) = \max(\varepsilon, \hat{p})$ . Thus,  $p$  and  $\hat{p}$  are treated as no less than  $\varepsilon$ , and the error factor is the maximum ratio, upward or downward, by which they differ. We use the default value  $\varepsilon = 0.002$ , which is consistent with the link loss rate distribution selected in simulation (see Section VI-B). If the estimate is perfect, the error factor is one.

<sup>4</sup>As defined before, a network is composed of virtual links after merging consecutive links without a branching point.

The LEND system operates in two stages: setup and monitoring. In the first phase, we select  $O(n \log n)$  paths to monitor, while in the second phase, we monitor these paths and use our measurements to diagnose any congestion/failure locations among all  $O(n^2)$  paths in the system. The setup phase takes only a few minutes even for a reasonably large overlay network of several hundred hosts, as shown in [8]. In this paper, we evaluate the speed of the second phase.

### B. Simulation Methodology

We consider the following dimensions for simulation.

- Topology type: We experiment with three types of BRITE [24] router-level topologies—Barabasi–Albert, Waxman, and hierarchical models—as well as with a real router topology with 284 805 nodes [9].
- Topology size: the number of nodes ranges from 1000 to 20 000. This node count includes both internal nodes (i.e., routers) and end-hosts.
- Fraction of end-hosts on the overlay network: We define end-hosts to be the nodes with the least degree. We then choose 50 to 300 end-hosts at random to be on the overlay network. We prune the graphs to remove the nodes and links that are not referenced by any path on the overlay network.
- Link loss rate distribution: 95% of the links are classified as “good” and the rest as “bad.” We focus on directed graphs, so the bidirectional links between a pair of nodes are assigned separate loss rates. We use two different models for assigning loss rate to links, as in [5]. In the first model (LLRD<sub>1</sub>), the loss rate for good links is selected uniformly at random in the 0%–0.2% range and the rate for bad links is chosen in the 5%–10% range. In the second model (LLRD<sub>2</sub>), the loss rate ranges for good and bad links are 0%–0.2% and 0.2%–100%, respectively. Given space limitations, most results discussed are under model LLRD<sub>1</sub> except for Section VI-C4.
- Loss model: After assigning each directional link a loss rate, we use either a Bernoulli or Gilbert model to simulate the loss processes at each link in the same manner as in [5] and [8]. The Gilbert model is more likely to generate bursty losses than the Bernoulli model. The state transition probabilities are selected so that the average loss rate matches the loss rate assigned to the link. We found that the results for the Bernoulli and the Gilbert models are similar. Since the Gilbert loss model is more realistic, all results presented in the paper are based on this model.

We repeated our experiments five times for each simulation configuration unless noted otherwise, where each repetition has a new topology and new loss rate assignments. The path loss rate is simulated based on the transmission of 10 000 packets. Using the loss rates of selected paths as input, we compute  $x_G$  then the loss rates of all the MILSs.

### C. Simulation Results

In this section, we discuss the results of our simulations. For all three types of synthetic topologies in our study, we found our system had similar accuracy. Barabasi–Albert topologies have the smallest MILSs, but also the shortest path lengths, while hierarchical topologies have both the longest MILSs and path lengths. Barabasi–Albert topologies have the largest ratios of



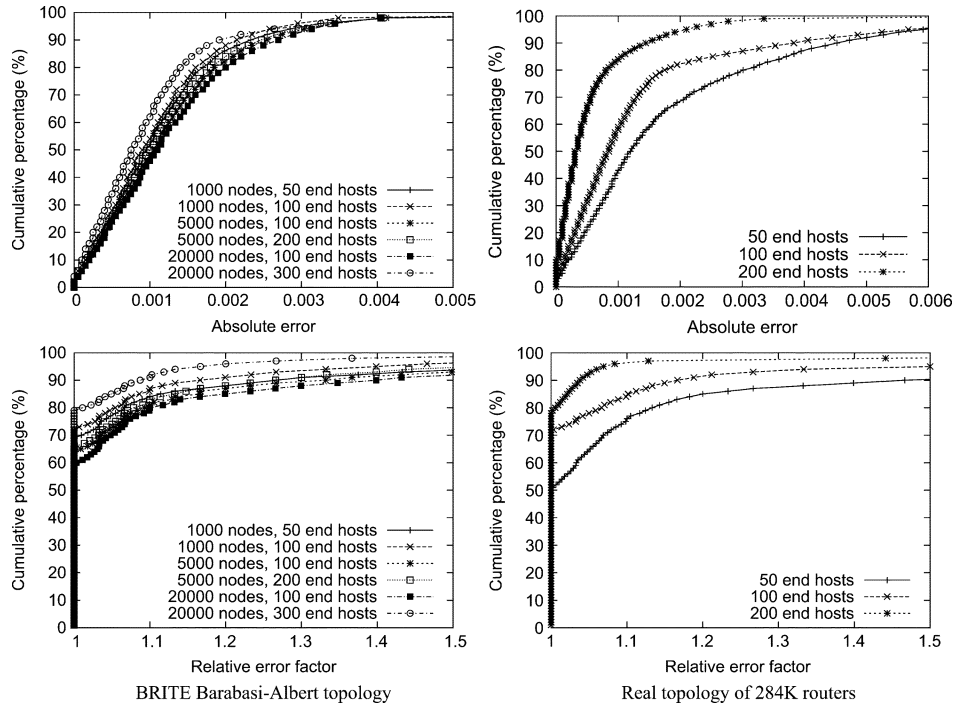


Fig. 9. Accuracy of MILSs on lossy paths: (top) cumulative distribution of absolute errors and (bottom) error factors under the Gilbert model for various topologies.

diagnosis granularity to average path length, while hierarchical topologies have the smallest such ratios. Thus, we only show the Barabasi–Albert and hierarchical topology results and omit the results of Waxman topology because of limited space.

1) *Accuracy of MILSs*: For all topologies in Section VI-B, we achieved highly accurate estimates of the MILS loss rates. Since our goal is to diagnose lossy paths, we evaluate the accuracy of the estimates of loss rates only for MILSs on the lossy paths. The results are even better when we consider the MILSs on all paths.

We plot the cumulative distribution functions (cdfs) of absolute errors and error factors with the Gilbert model in Fig. 9. The results on Waxman and hierarchical topologies are similar to those on Barabasi–Albert topologies, and so we omit them in the interest of space.

The errors come from the measurement noise and the approximation of the good path algorithm. The accumulated error is a potential problem for inference in a large network. However, our simulation results show it is not severe at all in our system. For all the configurations, 90% of the absolute errors are less than 0.006, and 90% of the error factors are less than 1.6. This shows that errors introduced by the good path algorithm and measurements do not accumulate in the matrix computations.

2) *Granularity of MILSs*: Table II shows the granularity of MILSs and related statistics under hierarchical BRITE topologies and the real-world Mercator topology. We first prune the topology so that it only contains the links on the paths among the random selected end-hosts. Then, we merge the links without branching points into one virtual link. We select a basis set  $\bar{G}$  for monitoring, which is again much smaller than the total number of paths. After that, we remove the good paths and good links inferred from these good paths from  $G$  and obtain  $G'$ . The number of lossy paths and the number of links in the lossy paths gives the size of  $G'$ , as shown in this table. The loss rate esti-

mation of MILSs is actually based on  $\bar{G}'$ , of which the size is about 30%–50% of the size of  $G'$  for the loss rate distribution of  $LLRD_1$ .

The MILS identification and loss rate calculation are based on virtual links to reduce the computational cost. Thus, the length of lossy paths and MILSs in the rightmost two columns of Table II are computed based on virtual links. The average lengths in terms of physical links are given in parentheses. The average length of the MILSs is quite small, less than two virtual links and less than three physical links. The last column of Table II shows the diagnosis granularity in length of both virtual links and links. Most diagnosis granularity is less than two virtual links, which is quite close to the diagnosis upper bound of pure end-to-end approaches (i.e., diagnosing every virtual link). Clearly, the diagnosis granularity becomes finer as more hosts are employed. This shows that *the granularity of MILSs is very small, and we can effectively locate the congestion/failure points*.

3) *Influencing Factors of the MILS Granularity*: In this subsection, we study two factors that influence MILS length: the size of overlay network and loss rate distributions of links.

Fig. 10 (top) shows the length of MILSs with different sizes of overlay network under the Mercator topology and  $LLRD_1$  loss rate distribution. *Link merging* in the figure means to merge consecutive link sequences without branching into virtual links. When the overlay network size is very small, less than 50, there is not much path sharing, so the MILSs are long. With more hosts and paths, sharing becomes significant, and the MILSs are dramatically shorter.

Fig. 10 (bottom) shows the lengths of MILSs for an overlay of 100 end-hosts under the Mercator topology with different percentage of links to be lossy links. Again, the loss rate distribution is  $LLRD_1$ . The lengths of the MILSs almost grow linearly with the percentage of lossy links. Usually the percentage of lossy

TABLE II  
SIMULATION RESULTS FOR HIERARCHICAL BRITE TOPOLOGY AND A REAL ROUTER TOPOLOGY. OL MEANS THE OVERLAY NETWORK. PL MEANS THE PATH LENGTH. NUMBER OF LINKS SHOWS THE NUMBER OF LINKS AFTER PRUNING (I.E., REMOVING THE NODES AND LINKS THAT ARE NOT ON THE OVERLAY PATHS). NUMBER OF VLS (VIRTUAL LINKS) GIVES THE NUMBER OF LINKS AFTER MERGING CONSECUTIVE LINKS WITHOUT BRANCHING POINT. LP STANDS FOR LOSSY PATHS. THE RIGHTMOST FOUR COLUMNS ARE COMPUTED USING THE VIRTUAL LINKS AFTER MERGING. THE CORRESPONDING LENGTH VALUES BEFORE MERGING ARE GIVEN IN THE PARENTHESES

Hierarchical model										
# of nodes	# of end host	# of paths	Avg PL	# of links	# of VLS	Rank ( $k$ )	# of LP	# of links in LP	Avg MILS length	Avg diagnosis granularity
1000	50	2450	9.73	1026	781	498	1009	122	1.83(2.52)	1.80(2.09)
	100	9900	9.59	1753	1480	1040	3233	174	1.45(1.87)	1.26(1.41)
5000	100	9900	14.57	2898	2026	1182	5459	386	2.30(3.41)	1.87(2.56)
	200	39800	14.44	5132	3979	2587	18317	531	1.71(2.34)	1.46(1.76)
20000	100	9900	18.84	4567	2782	1492	6254	577	3.04(4.69)	2.10(3.28)
	300	89700	18.80	10669	7494	4497	57442	1401	2.68(3.80)	2.07(2.67)

A real-router topology of 284,805 nodes

# of end host on OL	# of paths	Avg PL	# of links	# of VLS	Rank ( $k$ )	# of LP	# of links in LP	Avg MILS length	Avg diagnosis granularity
50	2450	8.86	3798	2774	1921	1042	903	2.23(3.03)	2.24(3.07)
100	9900	8.80	9802	7782	5879	3551	1993	1.71(2.27)	2.05(2.95)
200	39800	8.80	22352	18545	14811	14706	4335	1.49(1.92)	1.77(2.38)

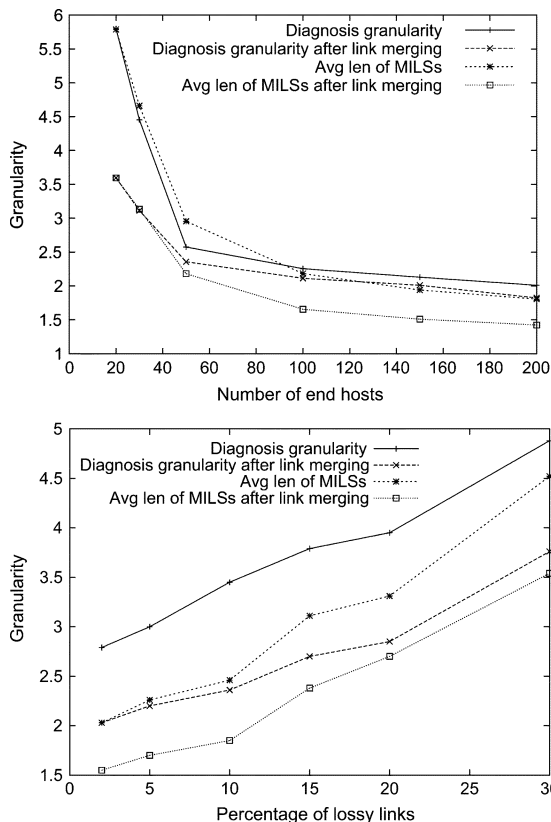


Fig. 10. (top) Granularity of MILSs with different network sizes and (bottom) different percentage of links as lossy links.

links in the Internet is very small—2% or even less. Therefore, the lengths of the MILSs are very small, which we also verify in the Internet experiment described in Section VII.

The average length of lossy MILSs is always higher than that of good MILSs. This is not surprising because the longer the

MILS is, the more likely it is to be lossy. Thus, the diagnosis granularity may be larger than the average length of all MILSs.

4) *Results for Different Link Loss Rate Distribution and Running Time:* We have run all the simulations above not only with the LLRD<sub>1</sub> model, but also with the LLRD<sub>2</sub> model. The results in the latter case are very similar to those of LLRD<sub>1</sub>, except that with larger loss rates and the same percentage of lossy links, the length of MILSs on the lossy paths has been increased by a bit. Given space limitations, we only show the lossy path inference with the Barabasi–Albert topology model and the Gilbert loss model in Table III.

The running times for LLRD<sub>1</sub> and LLRD<sub>2</sub> are similar, as in Table III. All speed results in this paper are based on a 3.2-GHz Pentium 4 machine with 2 GB memory. Note that it takes about 45 min to setup (select the measurement paths) for an overlay of 300 end-hosts, but less than 1 min for an overlay of size 100. Note that the setup only needs to run once, and there are efficient schemes to incrementally update  $\bar{G}$  when there are routing changes or adding/removing links [8]. Meanwhile, the continuous monitoring, inference, and diagnosis are very fast, for all cases. Even for the large overlay with 300 end-hosts, 89 700 paths, and more than 20 000 links, we can diagnose *all* trouble spots within 1 min. This shows that *we can achieve near real-time diagnosis*.

5) *Results for Dynamic Changes:* Because routes change in the Internet, and because end-hosts may join or leave the overlay network, our monitoring system must allow for dynamic updates. In this section, we describe two common scenarios: a change to the routing or a new host that joins the overlay. We show that the updates required by these changes can be expressed in terms of four primitive types of updates provided by the LEND system. In Section IV-D, we analyze the complexity of these updates. Here, we describe the performance of the update algorithms in simulations with a real Internet topology [9].

*Adding Nodes:* We start with an overlay network of 90 random end-hosts. Then, we randomly add an end-host to join the overlay, and repeat the process until the size of the overlay

TABLE III  
SIMULATION RESULTS WITH MODEL LLRD<sub>2</sub> USING BARABASI-ALBERT TOPOLOGIES

# of nodes	# of end hosts		Avg PL	# of LP	# of links in LP	Avg MILS length	Avg diagnosis granularity	Speed (second)	
	total	overlay						setup	update
1000	506	50	4.49	481	117	1.457(2.062)	1.476(1.656)	0.83	0.39
		100	4.42	1815	191	1.266(1.818)	1.169(1.259)	2.91	0.69
5000	2489	100	5.19	2046	587	1.384(2.027)	1.247(1.402)	19.8	0.93
		200	5.13	9028	1124	1.326(1.938)	1.187(1.271)	329	4.2
20000	10003	100	5.63	2232	1261	1.57(2.44)	1.491(1.688)	47.0	3.9
		300	5.62	23337	3692	1.321(2.051)	1.147(1.256)	2626	48.2

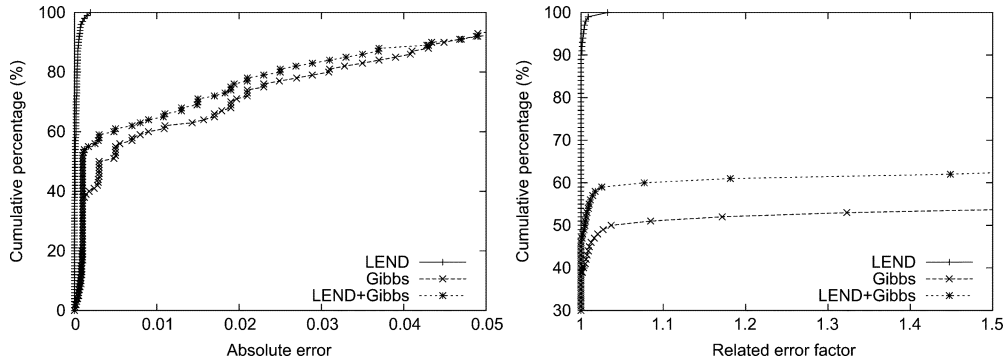


Fig. 11. (left) Absolute error and (right) relative error factor of Gibbs sampling and LEND.

reaches 100. Averaged over three runs, the average running time for adding a node is 0.21 s.

**Routing Changes:** Changes in network routing tables can substantially change the paths between hosts in the overlay network. To study the cost to update the LEND system after a routing change, we created an overlay network with 100 randomly chosen end-hosts. Then, we removed one of the links used by the overlay and recomputed the routes. In order to update the LEND system in this instance, we may require all four of the primitive updates described in Section IV-D. Averaged over three runs, the time to change a routing path (deleting one path and adding a new one) is about 1.2 s. This time is comparable to the time to recompute all the matrices from scratch, which is about 2.3 s. This is because the topology is relatively small (only 100 end-hosts) and because we tuned the performance of the initialization algorithm.

**6) Comparison With Gibbs Sampling:** In [5], Padmanabhan *et al.* propose three statistical methods to infer the loss rate of links using end-to-end measurement. We also implemented the Gibbs sampling algorithm, which was shown to be the most accurate approach in [5]. Note that in [5], the object is only to find out which virtual links are lossy; the method does not give an estimate of the loss rate. We modified the algorithm to use the average loss rate of all the samplings as the estimate of the loss rate of the virtual links.

Fig. 11 shows the absolute and relative errors of the inference of virtual links or MILSs. In this experiment, we use the real Mercator topology measured in [9] with Gilbert loss model and LLRD<sub>1</sub> distribution. There are 50 end-hosts and, thus, 4950 paths in total. Fig. 11 clearly shows that the accuracy of MILSs is much better than that of Gibbs sampling on virtual links. We note that the false positives and false negatives in Gibbs sampling are relatively frequent (about 10% in total), and thus for some virtual links, the absolute error is quite high ( $\geq 5\%$ ).

Fig. 11 also shows that Gibbs sampling inference based on our MILSs is more accurate than that based on end-to-end paths. This may be because MILSs have finer granularity and reduce the interaction between identified MILSs in the inference. The relative error factor results in Fig. 11 show the same trends we see in the absolute errors. As for running speed, Gibbs sampling based on the whole paths takes about five times longer than Gibbs sampling based on the MILS set when using the same running environment (i.e., the same machine and Matlab tool).

## VII. INTERNET EXPERIMENTS

Shortest path routing is often violated in the Internet, a phenomenon known as *path inflation* [25]. In addition, the loss behavior of real links may be more complicated than the behavior in synthetic models. Therefore, we deployed and evaluated our LEND system on the PlanetLab [26], and we discuss our results in this section.

### A. Methodology

We deployed our monitoring system on 135 PlanetLab hosts around the world (see Table IV). Each host is from a different institution. About 60% of the hosts are in US, and the others are located mostly in Europe and Asia. There are altogether  $135 \times 134 = 18\,090$  end-to-end paths among these end-hosts. In our experiments, we measured all the paths for validation. However, in practice, we only need to measure the basis set of on average 5706 end-to-end paths. The measurement load can be evenly distributed among the paths with the technique in [8] so that each host only needs to measure about 42 paths.

First, we measured the topology among these sites by simultaneously running “traceroute” to find the paths from each host to all others. Each host saves its destination IP addresses for sending measurement packets later. Then, we measured the loss rates between each pair of hosts. Our measurement consists of

TABLE IV  
DISTRIBUTION OF SELECTED PLANETLAB HOSTS

Areas and Domains		# of hosts
US (77)	.edu	50
	.org	14
	.net	2
	.com	10
	.us	1
Inter-national (58)	Europe	25
	Asia	25
	Canada	3
	South America	3
	Australia	2

TABLE V  
INTERNET EXPERIMENT RESULTS. THE LAST TWO ROWS ARE COMPUTED USING THE VIRTUAL LINKS. THE CORRESPONDING LENGTH VALUE USING PHYSICAL LINKS ARE GIVEN IN THE PARENTHESIS

End-to-end path	18,090
Avg path length	15.2
# of MILSs	1009
Avg length of MILSs	2.3(3.9)
Avg diagnosis granularity	2.3(3.8)

300 trials, each of which lasts 300 ms. During a trial, each host sends a 40-byte UDP packet to every other host. The packet consists of 20-byte IP header, 8-byte UDP header, and 12-byte data on sequence number and sending time. For each path, the receiver counts the number of packets received out of 300 to calculate the overall loss rate. We used the sensitivity test similar to that of [8] to choose these parameters so that measurement packets will not cause additional congestion.

To prevent any host from receiving too many packets simultaneously, each host sends packets to other hosts in a different random order. Furthermore, any single host uses a different permutation in each trial so that each destination has equal opportunity to be sent later in each trial. This is because when sending packets in a batch, the packets sent later are more likely to be dropped than received. Such random permutations are pre-generated by each host. To ensure that all hosts in the network take measurements at the same time, we set up sender and receiver daemons, then use a well-connected server to broadcast a “START” command.

## B. Experiment Results

In April 2005, we ran the experiments 10 times at different times of night and day. We report the average results from the 10 experiments.

1) *Granularity of MILSs and Diagnosis*: Of the  $135 \times 134 = 18090$  end-to-end paths, 65.5% were good paths, and these paths contained about 70.5% of the links. After removing good paths, only 6450 paths remained. The average length of lossy MILSs on these bad paths is 3.9 links or 2.3 virtual links (see Table V).

The diagnosis granularity of lossy paths is a little high: 3.8. However, we believe this is reasonable and acceptable for the following two reasons. First, it is well known that many packet losses happen at edge networks. In the edge networks, the paths usually have a long link chain without branches. For example, all paths starting from planetlab1.cs.northwestern.edu go through the same five first hops. If we use virtual link as the unit, we find the granularity is reduced to about 2.3 virtual

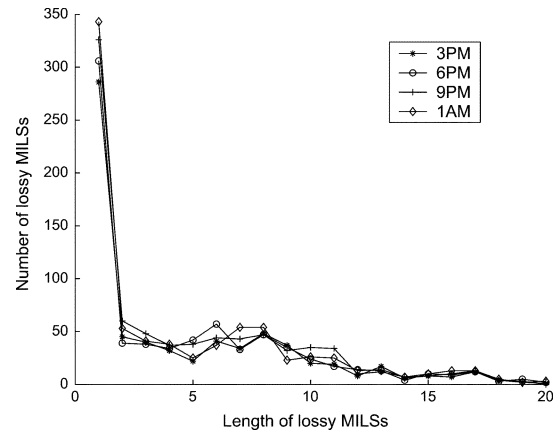


Fig. 12. Length distribution of lossy MILSs in physical links.

links. This shows our LEND approach can achieve good diagnosis granularity comparable to other more biased tomography approaches while achieving high accuracy.

Second, we find that there are some very long lossy MILSs. In our experiments, some MILSs were longer than 10 hops. Such long lossy MILSs occur in relatively small overlay networks because some paths do not overlap any other paths. Fig. 12 shows the distribution of the length in physical links of lossy MILSs during different times in a day (US Central Standard Time).

As shown in Section VI-C6, we can further apply the Gibbs sampling approach [5] based on the MILSs found and obtain a lower bound on the diagnosis granularity, which is 1.9 physical links (and one virtual link, obviously). However, accuracy will be sacrificed to some extent as shown in Section VI-C6. Nevertheless, by combining Gibbs sampling with our LEND system, we have the flexibility to trade off between granularity and accuracy.

2) *Accuracy Validation Results*: We apply the two schemes in Section V to validate our results: cross validation and consistency checking with IP-spoof-based source routing.

*Cross Validation*: We split the paths in the basis  $\vec{G}$  into two sets. We use the first set as the input  $\vec{G}$  to the LEND system to generate MILSs and to infer their loss rates. Then, we use the measurements of the second part to test the inferred link loss rates for cross validation. The basic idea is that if a path  $p$  in the second validation set contains some nonoverlapped MILSs  $v_i, i = 1, \dots, n$  obtained by the inference on the first set, then the loss rate of  $p$  should be no less than the total loss rate of these MILSs because  $p$  may have some additional lossy links that are not covered by these MILSs. Assuming the loss rate of  $p$  is measured to be  $l$  and the calculated loss rate of each MILS  $v_i$  is  $l_i$ , we check whether the following inequality holds:

$$(1 - l) < \prod_{i=1}^n (1 - l_i) + \varepsilon. \quad (5)$$

$\varepsilon$  shows the tolerable value of errors. In our experiments,  $\varepsilon$  is chosen to be 0.5%. We run the cross validation for each Internet experiment. The detailed numbers in different experiments are slightly different, but the cross-validation results are similar. Therefore, we take one experiment as an example. In this experiment, we have 5720 paths in  $\vec{G}$ , and we choose 2860 of them to identify 571 MILSs and infer their loss rates. Then, we validate the loss rates against the other 2860 paths. Of 571 MILSs,

TABLE VI  
MILS-TO-AS PATH LENGTH

	1 AS	2 ASs	3 ASs	> 3ASs
Len 1 MILSs (33.6%)	6.1%	27.5%	0	0
Len 2 MILSs (9.8%)	2.6%	5.8%	1.3%	0
Len > 2 MILSs (56.6%)	6.8%	17.8%	21.8%	10.2%

320 are on the paths of the second set and, thus, verified by 2200 paths. The result shows that more than 99.0% paths in the second set are consistent with MILSs computed by the first set. This shows that the loss rate inference of the MILSs is accurate.

*IP-Spoof-Based Consistency Checking:* For validation, we started the loss rate measurements and sent IP spoof packets at the same time. To reduce the overhead introduced by IP spoofing, we intentionally select the spoofed IP addresses to only infer the path segments that are more likely to be lossy based on some previous experiments. We applied the method introduced in Section V to measure 1000 path segments. Then, similar to the cross validation, we adopted (5) for matching validation. Out of a total of 1664 lossy MILSs, 361 are on the 1000 new paths and thus validated. When using the same parameter  $\varepsilon = 0.005$ , 93.5% of the loss rates of the new spoofed paths are consistent with the loss rate of these MILSs. Note that Internet routing changes may affect the validation results because once the path routing is changed, the reflecting router may no longer be on the original path, making the validation inapplicable. Fortunately, Internet routing is quite stable, and the IP-spoof-based consistency checking thus demonstrates that the MILS loss rate inference is very accurate.

3) *MILS-to-AS Mapping:* After we identify the lossy MILSs, we can locate and study the distribution of the lossy links. For example, are the lossy links usually within an AS or between two ASs?

To study this problem, we first need to obtain an accurate IP-to-AS mapping. A complete IP-to-AS mapping can be constructed from BGP routing tables by inspecting the last AS (the origin AS) in the AS path for each prefix. Mao *et al.* show that the IP-to-AS mapping extracted from BGP tables can lead to accurate AS-level forwarding path identification by changing about 3% assignment of the original IP-to-AS mapping [27]. However, their available IP-to-AS mapping result was obtained from measurement in 2003, and it is incomplete—we found that 1/4 of routers on our measurement paths are not mapped to any AS. Thus, we derive the IP-to-AS mapping from BGP tables directly, using the BGP tables published in Route Views [28] on March 2, 2005. The mapping is quite complete, and only 1.6% IPs involved (end-hosts and internal routers) cannot be mapped to ASs.

Ignoring these unmapped nodes, we map MILSs to their AS sequences, then analyze the relationship between lossy links and ASs. Table VI shows the length of AS paths of the lossy MILSs. Since it is impossible to infer which link or links are lossy in a long MILS, we only consider the short MILSs with length 1 or 2, which consist of about 44% of all lossy MILSs. It is obvious that most lossy links are connecting two different ASs. For example, most length 1 MILSs (27.5% of all MILSs) are connecting two ASs. This observation is consistent with common belief that the links connecting two ASs are more likely to be congested than those within an AS.

4) *Speed Results:* The LEND system is very fast in our Internet experiments. After topology measurement, the average setup (monitoring path selection, i.e., stage 1 in Fig. 6) time is 109.3 s, and the online diagnosis (stage 2 in Fig. 6) of the 3714 lossy paths for altogether 18 090 paths takes only 4.2 s.

## VIII. DISCUSSION

### A. Link Properties for Diagnosis

In this paper, we focus on inferring link loss rates, but a natural question is whether the algorithms are applicable to other link properties. One difficulty is that the linear algebraic model only works for additive metrics, such as loss rate and latency. Link bandwidth and capacity are not additive metrics, and therefore do not fit the linear algebraic model. Another difficulty is that the good path algorithm will not work if the path properties are always positive; thus, our approach does not apply to latency directly (in the directed graph model). However, latency can be decomposed into propagation delay, transmission delay, and queuing delay. In diagnosis, people usually care about the queuing delay, which is zero when the network traffic is light and becomes very large when congestion happens. Therefore, the good path algorithm may apply when most paths are over-provisioned, so queuing delay is negligible for most paths. It is a future work to conduct experiments to evaluate the applicability of our methods in this case.

### B. Justifying the MILS Definition

In this section, we discuss other possible identifiable units for diagnosis. Currently, MILSs have very strong conditions:

- MILSs consist of *consecutive* sequences of links.
- Each MILS is a *subpath* of one end-to-end path.
- The MILSs are *minimal*, i.e., one MILS cannot be expressed as a sum of other MILSs.
- Each MILS is *identifiable*, i.e., their properties can always be uniquely determined by end-to-end properties.

By defining MILSs in this way, we ensure that there are not too many of them: There may be at most  $l$  MILSs for each path of length  $l$ , and because of path overlap, there will typically be far fewer than  $r \times l_{\max}$  MILSs overall. By requiring that MILSs be identifiable, we also ensure that we can cheaply compute their loss rates using Algorithm 1. However, it is possible to relax the conditions in our definition, and we consider the effect that more relaxed definitions would have below.

As we have shown with Theorem 1, these conditions are so strong that, in a directed graph, the only MILSs are end-to-end paths. In order to localize the physically lossy links, we therefore used *bounds* on link loss rates at a granularity finer than that of the MILSs, which we obtain through the good path algorithm. With sufficiently tight bounds on a few physical links, we are able to get tight bounds on loss rates for link sequences that are shorter than end-to-end paths. This suggests that one could relax the definition of a MILS to allow some *unidentifiable* MILSs with loss rates that, though not uniquely identifiable, might still be known to lie within a small interval. The drawback to allowing unidentifiable MILSs is that it is impossible to tell which unidentifiable link sequences might have tightly bounded loss rates from the topology alone.

As we see in the bottom example of Fig. 3, MILSs can overlap each other. In this example, the sum of the loss rates on physical

links 1 and 4 is uniquely determined by end-to-end measurements, but because those links are nonconsecutive, we add link 3 in order to get the MILS  $b$ . Therefore, if we allowed nonconsecutive link sequences to be MILSs, we could conceivably get more detailed information. For example, if path  $3'$  were lossy, we might be able to tell that the fault was in link 3 and not in links 1 or 4. However, the number of possible nonconsecutive link sequences in each path grows exponentially with the path length  $l$ , and so it would be infeasible to test every nonconsecutive link sequence to see if it is identifiable. While it seems better to relax the requirement of continuity in MILSs, *identifiable* nonconsecutive link sequences are quite rare in practice. We did try several different topologies (synthetic topologies and Planetlab topology) and found only a small number of identifiable nonconsecutive sequences existed. Compared to the number of identifiable MILSs, they can be ignored. The experimental results suggest that our MILS definition is quite efficient and it represents the identifiable sequences in the paths.

## IX. CONCLUSION

In this paper, we advocate a nonbiased end-to-end network diagnosis paradigm that gives a smooth tradeoff between accuracy and diagnosis granularity when combined with various statistical assumptions. We introduce the concept of minimal identifiable link sequence and propose the good path algorithms to leverage measurement snapshots to effectively diagnose for directed graphs. Both simulation and PlanetLab experiments show that we can achieve fine level diagnosis with high accuracy in near real time. We further design a novel IP spoofing based scheme to validate Internet experiments.

## REFERENCES

- [1] "Technology Overview," Akamai Inc. [Online]. Available: <http://www.akamai.com/en/html/technology/overview.html>
- [2] M. Coates, A. Hero, R. Nowak, and B. Yu, "Internet tomography," *IEEE Signal Process. Mag.*, vol. 19, no. 3, pp. 47–65, May 2002.
- [3] A. Adams *et al.*, "The use of end-to-end multicast measurements for characterizing internal network behavior," *IEEE Commun. Mag.*, vol. 38, no. 5, pp. 152–159, May 2000.
- [4] T. Bu, N. Duffield, F. Presti, and D. Towsley, "Network tomography on general topologies," in *Proc. ACM SIGMETRICS*, 2002, pp. 21–30.
- [5] V. Padmanabhan, L. Qiu, and H. Wang, "Server-based inference of Internet link lossiness," in *Proc. IEEE INFOCOM*, 2003, pp. 145–155.
- [6] D. Rubenstein, J. F. Kurose, and D. F. Towsley, "Detecting shared congestion of flows via end-to-end measurement," *IEEE/ACM Trans. Netw.*, vol. 10, no. 3, pp. 381–395, Jun. 2002.
- [7] N. Duffield, "Simple network performance tomography," in *Proc. ACM SIGCOMM IMC*, 2003, pp. 210–215.
- [8] Y. Chen, D. Bindel, H. Song, and R. H. Katz, "An algebraic approach to practical and scalable overlay network monitoring," in *Proc. ACM SIGCOMM*, 2004, pp. 55–66.
- [9] R. Govindan and H. Tangmunarunkit, "Heuristics for Internet map discovery," in *Proc. IEEE INFOCOM*, 2000, pp. 1371–1380.
- [10] R. Caceres, N. Duffield, J. Horowitz, D. Towsley, and T. Bu, "Multicast-based inference of network-internal characteristics: Accuracy of packet loss estimation," in *Proc. IEEE INFOCOM*, 1999, pp. 371–379.
- [11] N. G. Duffield, F. L. Presti, V. Paxson, and D. Towsley, "Inferring link loss using striped unicast probes," in *Proc. IEEE INFOCOM*, 2001, pp. 915–923.
- [12] R. Mahajan, N. Spring, D. Wetherall, and T. Anderson, "User-level Internet path diagnosis," in *Proc. ACM SOSP*, 2003, pp. 106–119.
- [13] K. Anagnostakis, M. Greenwald, and R. Ryger, "Cing: Measuring network-internal delays using only existing infrastructure," in *Proc. IEEE INFOCOM*, 2003, pp. 2112–2121.
- [14] Y. Shavitt, X. Sun, A. Wool, and B. Yener, "Computing the unmeasured: An algebraic approach to Internet mapping," in *Proc. IEEE INFOCOM*, 2001, pp. 1646–1654.

- [15] G. H. Golub and C. F. Van Loan, *Matrix Computations*. Baltimore, MD: Johns Hopkins Univ. Press, 1989.
- [16] R. Caceres, N. Duffield, J. Horowitz, and D. Towsley, "Multicast-based inference of network-internal loss characteristics," *IEEE Trans. Inf. Theory*, vol. 45, no. 7, pp. 2462–2480, Nov. 1999.
- [17] N. Duffield, J. Horowitz, D. Towsley, W. Wei, and T. Friedman, "Multicast-based loss inference with missing data," *IEEE J. Sel. Areas Commun.*, vol. 20, no. 4, pp. 700–713, May 2002.
- [18] C. Tang and P. McKinley, "On the cost-quality tradeoff in topology-aware overlay path probing," in *Proc. IEEE ICNP*, 2003, pp. 268–279.
- [19] R. A. Brualdi, A. Pothen, and S. Friedland, "The sparse basis problem and multilinear algebra," *SIAM J. Matrix Anal. Appl.*, vol. 16, pp. 1–20, 1995.
- [20] Y. Zhang *et al.*, "On the constancy of Internet path properties," in *Proc. ACM SIGCOMM IMW*, 2001, pp. 197–211.
- [21] G. W. Stewart, *Matrix Algorithms: Basic Decompositions*. Philadelphia, PA: SIAM, 1998.
- [22] V. Paxson, "End-to-end routing behavior in the Internet," *IEEE/ACM Trans. Netw.*, vol. 5, no. 5, pp. 601–615, Oct. 1997.
- [23] R. Govindan and V. Paxson, "Estimating router ICMP generation delays," in *Proc. PAM*, 2002, pp. 1–8.
- [24] A. Medina, I. Matta, and J. Byers, "On the origin of power laws in Internet topologies," *Comput. Commun. Rev.*, pp. 18–28, Apr. 2000.
- [25] N. Spring, R. Mahajan, and T. Anderson, "Quantifying the causes of path inflation," in *Proc. ACM SIGCOMM*, 2003, pp. 113–124.
- [26] PlanetLab, [Online]. Available: <http://www.planet-lab.org/>
- [27] Z. M. Mao *et al.*, "Scalable and accurate identification of AS-level forwarding paths," in *Proc. IEEE INFOCOM*, 2004, pp. 1605–1615.
- [28] University of Oregon Route Views archive project, [Online]. Available: <http://www.routeviews.org/>



**Yao Zhao** received the Master's degree in computer science from Tsinghua University, Beijing, China, in 2001, and is pursuing the Ph.D. degree in the Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, IL.

He joined Bell Labs in June 2009. His research interests include network measurement, monitoring and security, and wireless ad hoc and sensor networks.



**Yan Chen** received the Ph.D. degree in computer science from the University of California at Berkeley in 2003.

He is an Assistant Professor with the Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, IL. His research interests include network security, network measurement, and diagnosis for both wired and wireless networks.

Dr. Chen won the Department of Energy (DoE) Early CAREER Award in 2005, the Department of Defense (DoD) Young Investigator Award in 2007, and the Microsoft Trustworthy Computing Awards in 2004 and 2005 with his colleagues.



**David Bindel** received the Ph.D. degree in computer science from the University of California at Berkeley in 2006.

He was a Courant Instructor of Mathematics at New York University, New York, and joined Cornell University, Ithaca, NY, as an Assistant Professor in Computer Science in Fall 2009. His research interests include numerical linear algebra, numerical software design, and modeling microsystems, musical instruments, human bones, and computer networks.

Dr. Bindel's dissertation was awarded the Householder Prize for most outstanding dissertation in numerical linear algebra over the past three years in 2008.