Towards Unbiased End-to-End Network Diagnosis

Yao Zhao, Yan Chen
Northwestern University
{yzhao, ychen}@cs.northwestern.edu

David Bindel
University of California at Berkeley
dbindel@eecs.berkeley.edu

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 MILSes 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 MILSes, 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.

Categories & Subject Descriptors

C.2.3 [Computer-Communication Networks]: Network Operations - Network monitoring

General Terms

Measurement, Experimentation

Keywords

Internet diagnosis, Network measurement, Linear algebra

1. 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 Committees 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 Internet 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.

Though several router-based Internet diagnosis tools have been proposed [2, 3], these tools generally need special support from routers. For example, Tulip [2] requires the routers to support continuous IP-ID for generated ICMP packets. Also these ICMP-based tools are subject to ICMP rate limiting, are sensitive to cross-traffic, and are un-scalable (see Section 2).

In contrast, many recently-developed tools for *Internet Tomography* use signal processing and statistical approaches to infer link level properties [4–7] or shared congestion [8] based on end-to-end measurements of IP routing paths. Here we define that the paths are composed of links, which are IP connections between routers. The relation between path and link properties can be written as a large linear system; however, as we observed in [9, 10], the linear system is *fundamentally underconstrained*: there exist *unidentifiable links* with properties that cannot be uniquely determined from path measurements.

To overcome this challenge to infer the property of each physical link or virtual link (a consecutive subpath of an IP path with no branches [7]), existing tomography approach have to make ceratin assumptions. These assumptions may not always hold in the Internet, which will cause systematic inference errors with non-zero expected value. In other words, such errors cannot converge to zero even with infinite amount of measurements. We call such error *bias* and those statistical assumptions *biased assumptions*.

In this paper, we advocate a different paradigm for network diagnosis: unbiased diagnosis (*i.e.*, with zero bias). Note that there are two *fundamental* statistical assumptions for *any* end-to-end network diagnosis approaches as follows.

- End-to-end measurement can infer the end-to-end properties accurately.

SIGCOMM'06, September 11–15, 2006, Pisa, Italy.

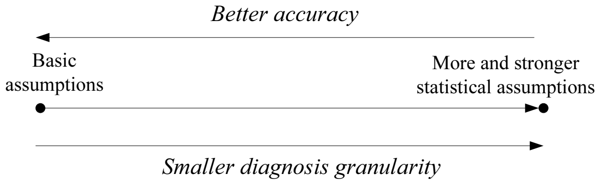

Better accuracy
Basic assumptions
More and stronger statistical assumptions
Smaller diagnosis granularity

Figure 1: The spectrum of network diagnosis methods.

- The linear system between path- and link- level properties assumes independence between link-level properties.

Although these two assumptions have been proved to work extremely well in practice (see Section 3.1), they may still introduce some bias. However, these are 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* and hence, *the most accurate*, diagnosis. We call it *Least-biased End-to-End Network Diagnosis* (in short, LEND) system.

When combined with statistical inference, this paradigm gives a full spectrum of network diagnosis methods with smooth tradeoff between accuracy and diagnosis granularity as shown in Figure 1. Here, we define the *diagnosis granularity* as the length of the smallest consecutive link sequences whose properties are inferred. Given Internet being an underconstrained system, 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.

Moreover, the LEND system desires the following properties:

- *Scalability*: Both the measurement and the inference computation impose low overhead even for large networks.
- *No special router support needed.*

In LEND system, 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. Then we apply an algebraic approach to separate the identifiable and unidentifiable components of each path to find the MILSes. For networks modeled as undirected graphs, this is relatively easy. We can use routing information to get the MILSes which are uniquely defined by the inherent path sharing of the Internet; and we propose efficient algorithms to find all such MILSes. However, the real Internet has asymmetric link properties (*e.g.*, loss rate), and so must be modeled as a directed graph. But to find the MILSes in a directed graph is significantly more challenging. In this paper, we make the following contributions.

- 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 which 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 such observation, we propose a "good path" algorithm, which uses both topology and measurement snapshots to find MILSes with the finest granularity.
- We design efficient algorithms to incrementally update the MILSes 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.
- To validate our estimates, we propose a novel method of link-level loss rate inference based on IP spoofing which enables a limited form of source routing.

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 lengths of all the lossy MILSes 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 [11]. This suggests we can do very fine-level accurate diagnosis with reasonably large overlay networks.

In addition, the loss rate inference on the MILSes 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 seconds, and the online diagnosis of 18,090 paths, 3,714 of which are lossy, takes only 4.2 seconds.

For the rest of the paper, we first survey related work in the next section. Then we define MILS in Section 3, present its discovery algorithms in Section 4, and validate in Section 5. Evaluations are described in Sections 6 and 7. Finally, we conclude in Section 8.

2. RELATED WORK

The algebraic approach was also used recently for scalable overlay network monitoring to infer the end-to-end *path* properties [10]. By computing the loss rates of some "virtual links" from a subset of the $O(n^2)$ paths, the loss rates of the remaining paths can be inferred. However, a "virtual link" defined in [10] is not a physical link or a subpath (different definition to this paper and other tomography papers). It is much more challenging to infer the properties on the *link* level, as we show in this paper. Nevertheless, for overlay diagnosis, we naturally inherit the scalability and load balancing from [10]. 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.

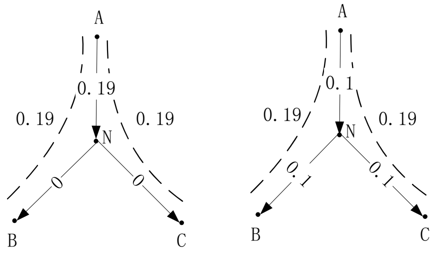
Ping and traceroute are the earliest Internet diagnosis tools, and they are still widely used. However, the asymmetry of Internet routing and of link properties makes it difficult to use these tools to infer properties of individual links. The latest work on network diagnosis can be put into two categories: *pure end-to-end approaches* [4–7, 9, 12, 13] and *router response based approaches* [2, 3, 14].

2.1 Pure End-to-end Approach

Most end-to-end tomography tools fall in one of two classes: tools which are based on temporal correlations among multiple receivers in a multicast-like environment [4–6, 12, 13], and tools which impose additional statistical assumptions beyond the linear loss model described in Section 3.1. As we discuss below, none of these tools provide *unbiased diagnosis* as defined in Section 1. As evidence of the utility of least-unbiased diagnosis, we show in Section 6 that our inference is much more accurate than the inference of one statistical tool based on Gibbs sampling.

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 does not exist 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 [7] and [9] 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, Figure 2. In this tree, we can only measure 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 [7] and SCFS [9] 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 [7], 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, and the LEND system will determine that none of the individual links are identifiable, and will get MILSes $A \rightarrow N \rightarrow B$ and $A \rightarrow N \rightarrow C$.



(a) (b)

Figure 2: Example of an underconstrained system.

Gurewitz *et al.* use linear algebra to estimate one-way delay with the measurement of delay of cyclic paths which are hard to obtain in the real Internet [15].

2.2 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 [2, 3] 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 [2]. Tulip, the latest representative of this router-based approach [2], 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 about 40% overall loss rates [2].

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

3. ALGEBRAIC MODEL AND MILS

In this section, we briefly describe the system architecture and the algebraic model of the LEND system. The algebraic model is widely used in Internet tomography. But the techniques for diagnosis require significant amount of extra design over this framework, as we will describe in the paper, *e.g.*, the MILSes introduced in Section 3.2.

3.1 Algebraic Model

Here we briefly introduce the algebraic model that is widely used in network diagnosis. Suppose an overlay network spans s IP links. We represent a path by a column vector $v \in \{0,1\}^s$, where the jth 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

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

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.* 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]. In addition to [16], formula (1) has also proven useful in many other link/path loss inference works, such as [6, 7, 17, 18]. Our Internet experiments also show that the link loss dependence has little effect on the accuracy of (1).

We take logarithms on both sides of (1). Then by defining a column vector $x \in \mathbb{R}^s$ with elements $x_j = \log(1 - l_j)$, and writing v^T as the transpose of the row vector v, we can rewrite (1) as follows:

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

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 ith 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 equations in form (2) as

$$Gx = b \tag{3}$$

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 $k < s$ [10]. 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 [9]. Figure 2 shows an example in which no link is identifiable.

3.2 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

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 ith link
p_i	loss rate on ith 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 (= 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 $\bar{G}'^T$. $\bar{G}'^T = Q'R'$

Table 1: Table of notation

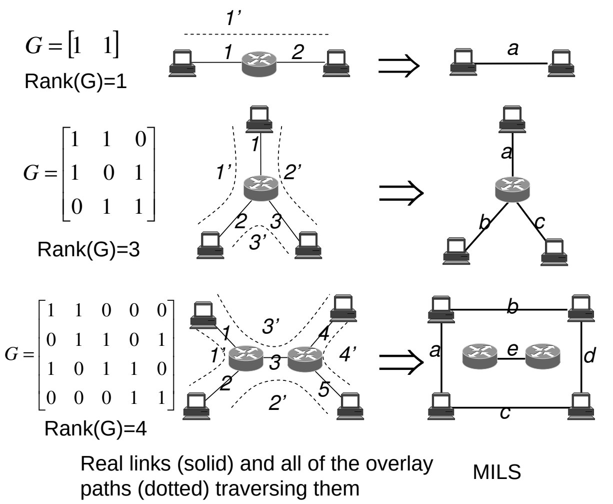

$G = [1 \quad 1]$
Rank(G)=1
1'
1
2
a
$G = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$
Rank(G)=3
1
1'
2'
2
3
3'
a
b
c
$G = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}$
Rank(G)=4
1
1'
3'
4
4'
3
2
2'
5
a
b
e
d
c
Real links (solid) and all of the overlay paths (dotted) traversing them
MILS

Figure 3: Sample topologies and MILSes.

end-to-end path measurements. We introduce *minimal identifiable link sequence* or *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.

Figure 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 MILSes. But link 3 can be presented as $(2' + 3' - 1' - 4')/2$, which means link 3 is identifiable, and there are *five* MILSes. These examples show three features of the MILS set:

- The MILSes 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 Figure 3. But it is helpful to keep such links for diagnosis.
- Some MILSes may contain other MILSes. For instance, MILS e is contained in MILSes 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 MILSes 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 MILSes 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 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 which is as small as possible. Thus, we want to have vectors which correspond to consecutive link sequences. If we did not make this assumption, there could exist an exponentially large number of MILSes.

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 MILSes 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 related work [10], of which the goal is to find a basis of $\mathcal{R}(G^T)$ made of end-to-end paths, identifying MILSes is a more challenging task.

4. IDENTIFYING MILSES

The LEND system consists of two stages, shown in Figure 4. In the first stage, we infer the loss rates of all end-to-end paths; this can be done with $O(n \log n)$ path measurements, as described in [10]. In this paper, we focus on the second stage: finding MILSes and inferring their loss rates. 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.

4.1 MILSes in Undirected Graphs

As we have defined them, MILSes satisfy two properties: they are minimal, *i.e.* they cannot be decomposed into shorter MILSes; and they are identifiable, *i.e.* they can be expressed as linear combinations of end-to-end paths. Algorithm 1 finds all possible MILSes 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

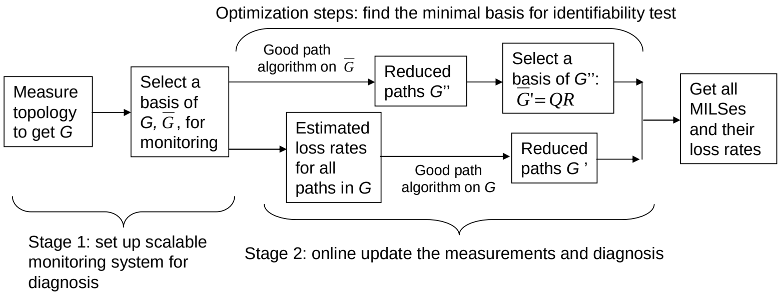

Optimization steps: find the minimal basis for identifiability test
Measure topology to get G
Select a basis of G, $\bar{G}$, for monitoring
Good path algorithm on $\bar{G}$
Reduced paths G''
Select a basis of G'': $\bar{G}'=QR$
Estimated loss rates for all paths in G
Good path algorithm on G
Reduced paths G'
Get all MILSes and their loss rates
Stage 1: set up scalable monitoring system for diagnosis
Stage 2: online update the measurements and diagnosis

Figure 4: The operational flowchart of the LEND system architecture

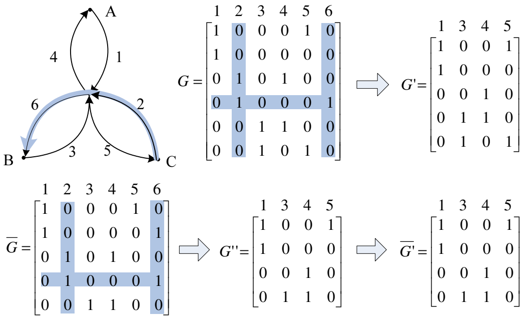


Figure 5: Examples showing all the matrices in the flowchart

will lie in the path space if and only if $\|v\| = \|Q^T v\|$. If the 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 $O(i \times k)$.

```
procedure Seek_MILS
Let $Q$ be an orthonormal basis of $\mathcal{R}(G^T)$ which is pre-
computed as in [10] ;
foreach path $p$ in $G$ do
    start_mils := logical array of length($p$) ;
    Clear start_mils to all false ;
    for $i := 1$ to length(p) do
        foreach segment $S = p_k \ldots p_l$ of length $i$ do
            if start_mils($k$) then
                continue ;
            else
                Let $v$ be the corresponding vector of $S$ ;
                if $\|Q^T v\| = \|v\|$ then
                    start_mils($k$) := true ;
                    $S$ is a MILS ;
                else
                    $S$ is not a MILS ;
                end
            end
        end
    end
end
```

Algorithm 1: Seeking all MILSes in an undirected graph

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 MILSes 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 MILSes are obtained from scanning all r end-to-end paths than from scanning only the k end-to-end paths which are directly monitored. Furthermore, each physical link used by the network will be used by one of the k monitored paths, so the MILSes obtained from this smaller set of paths do cover every physical link. Therefore 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.

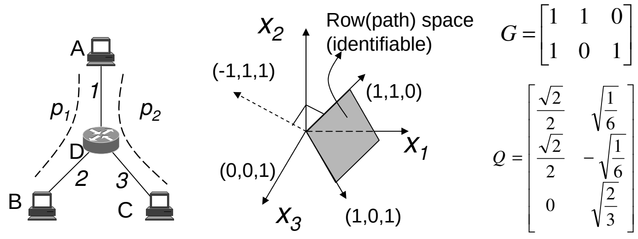

A
B
C
D
1
2
3
p_1
p_2
x_2
x_1
x_3
Row(path) space (identifiable)
(-1,1,1)
(1,1,0)
(0,0,1)
(1,0,1)
$G=\begin{bmatrix}1&1&0\\1&0&1\end{bmatrix}$
$Q=\begin{bmatrix}\frac{\sqrt{2}}{2}&\sqrt{\frac{1}{6}}\\\frac{\sqrt{2}}{2}&-\sqrt{\frac{1}{6}}\\0&\sqrt{\frac{2}{3}}\end{bmatrix}$

Figure 6: MILSes in undirected graph.

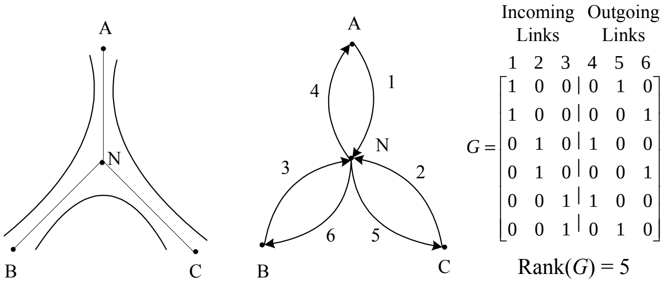


Figure 7: Undirected graph vs Directed graph.

Once we have identified all the MILSes, 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 [10]). For example in Figure 6, $x_G = (\frac{2x_1+x_2+x_3}{3}, \frac{x_1+2x_2-x_3}{3}, \frac{x_1-x_2+2x_3}{3})^T$. Obviously, x_G shows some identifiable vectors in $\mathcal{R}(\bar{G})$, however, they may not be MILSes. 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 [21] with the same order of computational complexity. However, focus of this paper is on the directed graph which is ignored in [21]. Furthermore, compared to [21], our approach inherits the key feature of measurement efficiency of [10] (*i.e.* requiring only $O(n \log n)$ measurements of end-to-end paths instead of n^2 paths), and reuses the computational output of [10] such as x_G and Q.

4.2 MILSes in Directed Graphs

4.2.1 Special Properties for Directed Graphs

Surprisingly, our MILS algorithm cannot be extended to directed graphs directly. We found that no path can be decomposed into more than one MILS, *i.e.*, each path itself is a MILS. Figure 7 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, 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 Figure 7, we can explain the lack of identifiable links as follows. We can split G into two sub-matrices, 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 $u^N \in \{0,1\}^s$ and $w^N \in \{0,1\}^s$ such that $u_i^N = 1$ if link i is an incoming link for node i, and $w_i^N = 1$ if link i is an outgoing link for node i. For any path with vector v, $v^T u^N$ is the count of the number of links going into N which appear on the path, and $v^T w^N$ is the count of the links exiting N. If v corresponds to an end-to-end routing path, either N is traversed exactly once and $v^T u^N = v^T w^N = 1$, or N is not traversed at all and $v^T u^N = v^T w^N = 0$. Since every row in G represents an end-to-end path, we have $Gu^N = Gw^N$.

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,

$$x^T u^N = z^T G u^N = z^T G w^N = x^T w^N$$

Therefore, if the link sequence includes an incoming link for node N, it must also include an outgoing link. 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. □

Routing loops are rare in the Internet, thus given Theorem 1, 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.

4.2.2 Practical Inference Methods for Directed Graphs

Considering the simple directed graph in Figure 7, 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 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 call relax the definition of “good path” and allow a negligible loss rate of at most σ (*e.g.*, $\sigma = 0.5\%$, which is the threshold for “no loss” in [20]). Then again it becomes 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 Figure 4, there are two steps for identifying MILSes under directed graphs. 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 MILSes and their loss rates in G. For the good links which are in the middle of lossy MILSes identified, we add them back so that MILSes 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, such optimization approach makes the LEND very efficient for online diagnosis. In Figure 5, 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.

4.3 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 re-compute the MILSes and their loss rates. We perform this re-computation in two steps: we first incrementally update the decomposition of the G' matrix, and then compute the MILSes and their properties using the algorithm described in Section 4.1.

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 which 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 [22, 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 7.2.4, we show that it takes only a few seconds to complete an incremental update to Q' and R' and re-identify the MILSes. Given that end-to-end Internet paths tend to be stable on the time scale of a day [23] and link loss rates remain operationally stable on the time scale of an hour [20], our algorithm should suffice for online updates and diagnosis.

4.4 Combining with Statistical Diagnosis

As discussed before, the linear system is under-constrained, and so there exist some unidentifiable links. With MILSes, we attempt to discover the smallest path segments for which properties can be uniquely identified. However, there are various statistical methods which produce estimates of properties at a finer granularity, *e.g.* at the virtual link level (see Section 2.1 for definition). Essentially, these methods try to use statistical assumptions to resolve the likely behavior in the unmeasured space discussed in Section 3.1, and therefore provide only possible estimates as shown in Figure 2 [7].

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 MILSes*. In comparison with the traditional statistical tomography which 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 MILSes are vectors in $\mathcal{R}(G^T)$, and the MILS set contains a basis of $\mathcal{R}(G^T)$. Thus, inference with MILSes is equivalent to inference with the whole end-to-end paths.

Take the linear optimization and Bayesian inference using Gibbs sampling introduced in [7], for example; these algorithms can be used without modification on our MILS set rather than on the original end-to-end paths. Section 6.3.6 shows that combined with our least-unbiased approach, Gibbs sampling inference improves its accuracy. In addition, the computational complexity of Gibbs sampling inference based on the MILS set is dramatically reduced because the input "paths" is much shorter than the whole end-to-end paths.

5. 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 6. Then we test LEND on the real Internet in Section 7. For validation on the real Internet, in addition to the classical cross validation, we need a more powerful approach. As shown in Section 2, existing router-based diagnosis tools like Tulip are neither very accurate nor scalable, and so 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 [24]. We use IP spoofing to obtain a limited source routing, which helps validate the accuracy of MILSes. With this technique, we can measure the properties of new paths which we could not normally probe. These additional measurements are then used to validate the inferred loss rates of MILSes.

Figure 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. Assuming 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 MILSes 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 7.2.2 where the consistency checking idea is also used in cross-validation.

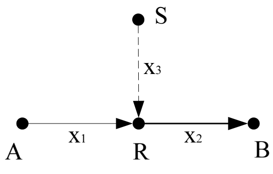


Figure 8: IP spoofing example.

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. We managed to get one host in our institute exempted from such filtering.
- Second, as with other router-based diagnosis approaches [2], our scheme is subject to ICMP rate-limiting on routers for measuring the loss rates. We filter those routers with strict ICMP rate-limiting.

6. EVALUATION WITH SIMULATION

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

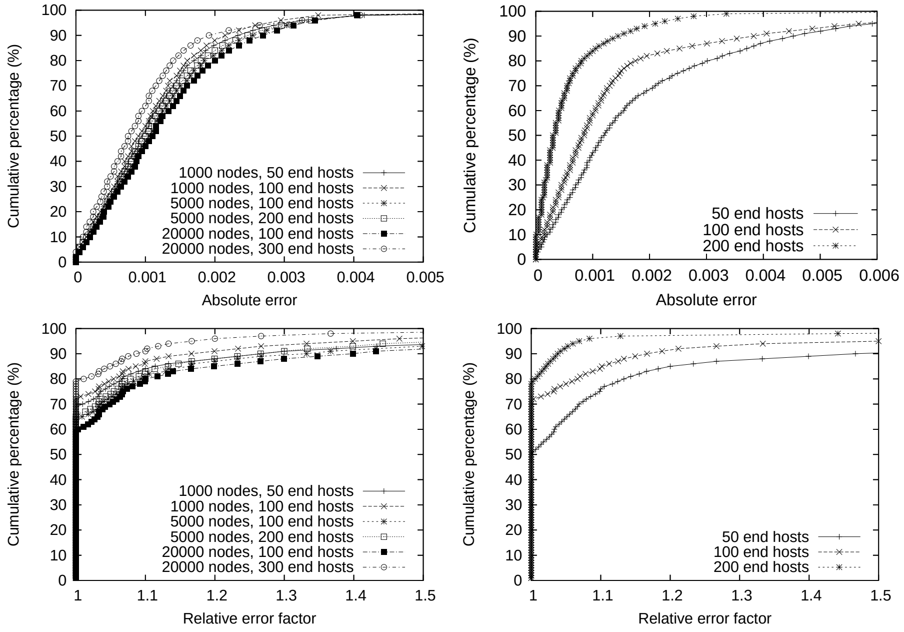
6.1 Metrics

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 to what range we can locate the cause of network congestion/failures. We define the diagnosis granularity of a path as the average of the lengths of all the lossy MILSes 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, if an overlay network has only two lossy paths: one path has two lossy MILSes 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 $((2 + 4) /2 + 3) /2 = 3$. The granularity indicates the range of congestion/failure locations when they occur. We represent the granularity with both physical link and virtual link [1] as the length unit. In this paper, we use physical link as the default unit, and use the virtual link as unit only when specifically comparing with the optimal lower bound of end-to-end approaches which have the diagnosis granularity as each virtual link.

Throughout this paper, we classify 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 6.2a good path has less than 0.5% loss rate, the threshold for "no loss" in [20], and thus the good path algorithm introduces certain errors (or bias). The question is whether the error introduced by the good path algorithm will be accumulative or not in the matrix computations. If the error is not accumulative, we can simply adjust the threshold of the good path for desirable accuracy and best diagnosis granularity.

[1] As defined before, a network is composed of virtual links after merging consecutive links without branching point.



BRITE Barabasi-Albert topology

Real topology of 284K routers

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

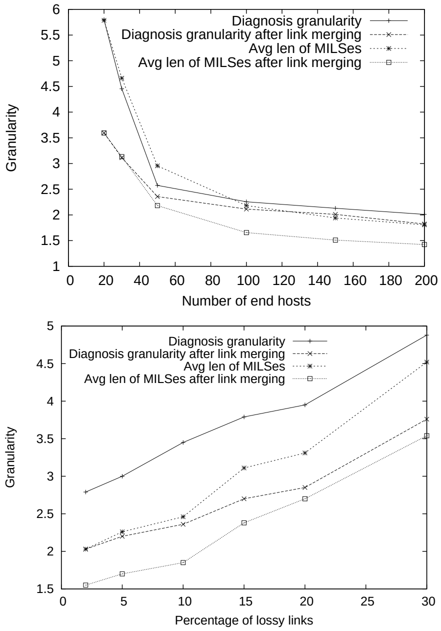


Figure 10: Granularity of MILSes with different network sizes (top) and different percentage of links as lossy links (bottom).

To compare the inferred loss rate $\hat{p}$ with the real loss rate p of MILSes, we analyze both the absolute error and the error factor. The absolute error is $|p-\hat{p}|$. We adopt the error factor $F_\varepsilon(p,\hat{p})$ defined in [6] as follows:

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

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 ε, and thus the error factor is the maximum ratio, upwards or downwards, by which they differ. We use the default value $\varepsilon = 0.003$, as is consistent with the link loss rate distribution selected in simulation (See Section 6.2). If the estimation is perfectly on target, the error factor is one.

Operation of the LEND system requires two steps: setup, and monitoring and diagnosis. In the first step we select $O(n \log n)$ paths to measure, while in the second step we monitor these paths and diagnose the congestion/failure locations of all the $O(n^2)$ paths. The running time for the first step is only a few minutes even for a reasonably large overlay network of several hundred hosts, as shown in [10]. Thus in this paper, we focus on evaluating the speed of the second step.

6.2 Simulation Methodology

We consider the following dimensions for simulation.

- Topology type: We experiment with three types of BRITE [25] router-level topologies - Barabasi-Albert, Waxman and hierarchical models - as well as with a real router topology with 284,805 nodes [11].
- Topology size: the number of nodes ranges from 1000 to 20000. 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 randomly choose from 50 to 300 end hosts 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, thus 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 [7]. In the first model ($LLRD_1$), the loss rate for good links is selected uniformly at random in the 0-0.3% range and the rate for bad links is chosen in the 5-10% range. In the second model ($LLRD_2$), the loss rate ranges for good and bad links are 0-0.3% and 0.3-100% respectively. Given space limitations, most results discussed are under model $LLRD_1$ except for Section 6.3.4.
- 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 [7, 10]. 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 10000 packets. Using the loss rates of selected paths as input, we compute x_G, then the loss rates of all the MILSes.

6.3 Simulation Results

In this section, we discuss the evaluation results. Our experiments show that the three synthetic topologies have similar results for the accuracy. For the diagnosis granularity, Barabasi-Albert topologies have the largest ratios of diagnosis granularity vs. the average path length. Thus we only show the Barabasi-Albert topology results because it gives the most conservative results on fault localization.

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)

Table 2: Simulation results for 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 mainly computed using the virtual links after merging. The corresponding length values before merging are given in the parenthesis.

# 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

Table 3: Simulation results with model $LLRD_2$ using Barabasi-Albert topologies

6.3.1 Accuracy of MILSes

For all topologies in Section 6.2, we achieved high loss rate estimation accuracy. Since our goal is to diagnose lossy paths, we evaluate the accuracy of the estimates of loss rates only for MILSes on the lossy paths. The results are even better when we consider the MILSes on all paths.

We plot the cumulative distribution functions (CDFs) of absolute errors and error factors with the Gilbert model in Figure 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 when computing large matrix. 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 is due to the least-unbiased qualities of our diagnosis algorithms.

6.3.2 Granularity of MILSes

Table 2 shows the granularity of MILSes and related statistics under the real-world Mercator topology. The granularity results from other BRITE topologies are similar, and we omit the exact tables for the limited space of the paper. 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 estimation of MILSes is actually based on $\bar{G}'$, of which the size is about 30% to 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 MILSes in the rightmost two columns of Table 2 is computed based on virtual links. After that, we recover each virtual link to its original link segments and give the length value in parenthesis of the table. The average length of MILSes is quite small, mostly less than 2 when considering virtual links, and mostly less than 3 without such link merging. The last column of Table 2 shows the diagnosis granularity in length of both virtual links and links. Most diagnosis granularity is less than 2 virtual links, which is quite close to the diagnosis upper bound of pure end-to-end approaches (*i.e.*, diagnosing every virtual link). This shows that *the granularity of MILSes is very small and we can effectively locate the congestion/failure points*.

6.3.3 Influencing Factors of the MILS Granularity

In this subsection, we study two such influencing factors: the size of overlay network and loss rate distributions of links.

Figure 10 (top) shows the granularity of MILSes 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 sequence without branching into virtual link. When the overlay network size is very small, less than 50, there is not much path sharing, so the MILS lengths are long. With more hosts and paths, sharing becomes significant, and the MILS lengths are reduced dramatically. Such sharing growth becomes slower and slower when the network size is bigger than 100.

Figure 10 (bottom) shows the granularity of MILSes 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 granularity of MILSes almost grows linearly to the percentage of lossy links. Usually the percentage of lossy links in the Internet is very small, like 2% of even smaller. So the granularity of the MILSes is very small, which is also verified through the Internet experiment described in Section 7.

The average length of lossy MILSes is always higher than that of good MILSes. 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 MILSes.

6.3.4 Results for Different Link Loss Rate Distribution and Running Time

We have also run all the simulations above with model $LLRD_2$. The results are very similar to those of $LLRD_1$ except that with larger loss rates and the same percentage of lossy links, the length of MILSes 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 3.

The running time for $LLRD_1$ and $LLRD_2$ are similar, as in Table 3. All speed results in this paper are based on a 3.2GHz Pentium 4 machine with 2GB memory. Note that it takes about 45 minutes to setup (select the measurement paths) for an overlay of 300 end hosts, but less than one minute 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 [10]. 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 one minute. This shows that *we can achieve near real-time diagnosis*.

6.3.5 Results for Dynamic Changes

Because of the change of Internet and Overlay network, our monitoring system has to dynamically update according to the changes. In this section, we study two common scenarios: end hosts joining as well as routing changes. In Section 4.3, we analyze the computation complexity of four primitive updates to our LEND system. We use the real topology [11] in simulation to show the efficiency of the dynamic update of our LEND system.

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 reaches 100. Averaged over three runs, the average running time for adding a node is 0.21 second. Notice that we add a block of paths together to speedup adding node.

Routing changes: Routing changes influence the link sequences in the paths, and as a result the loss rate of the paths may also changes a lot. We first create an overlay network with 100 random end hosts on the real router topology. Then we simulate topology changes by randomly choosing a link that is on some path of the overlay and removing of such a link will not cause disconnection for any pair of overlay end hosts. Then we assume that the link is broken, and re-route the affected path(s). The changed paths may actually trigger all the four basic changes we described in Section 4.3. Average over three runs, the average running time for changing a routing path (delete the original and then add a new one) is about 1.2 seconds. This time is comparable to the time of re-computing all the matrixes from scratch, which is about 2.3 seconds. This is because the block algorithm of path adding speedup much and the topology is not very large (only 100 end hosts).

6.3.6 Comparison with Gibbs Sampling

In [7], V. Padmanabhan *et al.* proposed three statistical approaches 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 [7]. Note that in [7], the object is only to find out which virtual links are lossy, which does not give an inference on the value of loss rate. By modifying the algorithm a little bit, we use the average loss rate of all the samplings as the inferred loss rate of virtual links.

Figure 11 shows the absolute and relative errors of the inference of virtual links or MILSes. Here we select the real Mercator topology measured in [11] with Gilbert loss model and $LLRD_1$ distribution. There are 50 end hosts, and thus 4950 paths in total. Figure 11 clearly shows that the accuracy of MILSes is much better than that of Gibbs Sampling on virtual links. It is worth mentioning that the false positives and false negatives of Gibbs Sampling are relatively high (about 10% in total), and thus for some virtual links the absolute error is quite high ($\geqslant 5\%$). Figure 11 also shows that Gibbs sampling inference based on our MILSes has higher accuracy than that based on end-to-end paths. This may be because MILSes have finer granularity and reduce the interaction between identified MILSes in the inference. The relative error factor results in Figure 11 confirm the result of absolute errors. As for running speed, Gibbs sampling based on the whole paths takes about 5 times more running time than that based on MILS set when using the same running environment (*i.e.*, the same machine and Matlab tool).

US (77)	# of hosts	International (58)	# of hosts
.edu	50	Europe	25
.org	14	Asia	25
.net	2	Canada	3
.com	10	South America	3
.us	1	Australia	2

Table 4: Distribution of selected PlanetLab hosts.

7. INTERNET EXPERIMENTS

Shortest path routing is often violated in the Internet, a phenomenon known as *path inflation* [26]. In addition, the behavior of lossy links may be more complicated than those of synthetic models. Therefore, we deployed and evaluated our LEND system on the PlanetLab [27] and discuss the results in this section.

7.1 Methodology

We deployed our monitoring system on 135 PlanetLab hosts over the world (See Table 4). Each host is from a different institute. About 60% of hosts are in US and others are distributed 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. But in practice, we only need to measure the basis set of on average 5,706 end-to-end paths. The measurement load can be evenly distributed among the paths with the technique in [10] 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 300 trials, each of which lasts 300 msec. 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 [10] 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 per-

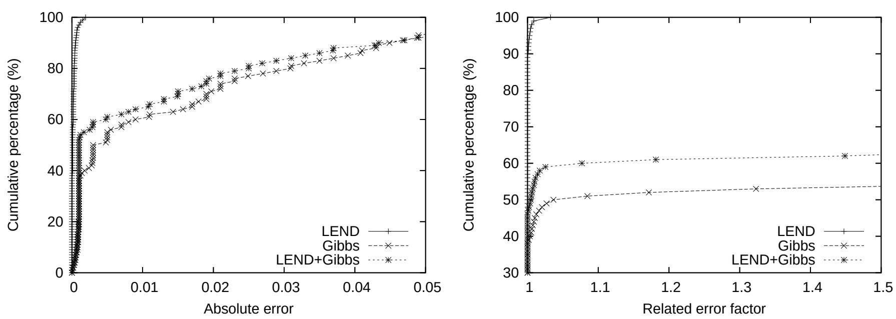


Figure 11: Absolute error (left) and relative error factor (right) of Gibbs Sampling and LEND.

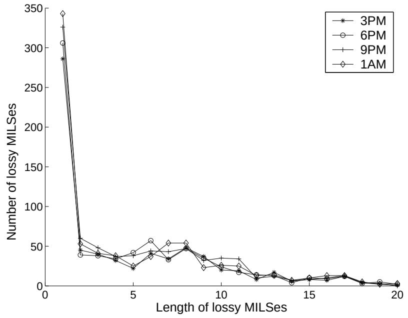


Figure 12: Length distribution of lossy MILSes in physical links.

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

Table 5: 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.

mutations 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.

7.2 Experiment Results

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

7.2.1 Granularity of MILSes and Diagnosis

For the total of $135 \times 134 = 18,090$ end-to-end paths, after removing about 65.5% good paths containing about 70.5% good links, there are only 6450 paths remaining. The average length of lossy MILSes on bad paths is 3.9 links or 2.3 virtual links.

The diagnosis granularity of lossy paths is a little high: 3.8. But we believe it 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 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 exist some very long lossy MILSes as illustrated in Figure 12, which shows the distribution of the length in physical links of lossy MILSes measured in different time periods of a day (US Central Standard Time). For example, some MILSes are longer than 10 hops. Such long lossy MILSes occur in relatively small overlay networks because some paths do not overlap any other paths.

As shown in Section 6.3.6, we can further apply Gibbs sampling approach [7] based on the MILSes found and obtain the lower bound on the diagnosis granularity, which is 1.9 physical links and obviously one hop with respect to virtual links. However, accuracy will be sacrificed to some extent as shown in Section 6.3.6. Nevertheless, by combining both statistic approaches and our LEND system, we provide the full flexibility to trade off between granularity and accuracy.

7.2.2 Accuracy Validation Results

We apply the two schemes in Section 5 to validate our results: cross-validation and consistency checking with IP spoof-based source routing.

7.2.2.1 Cross Validation.

We split the paths in the basis $\bar{G}$ into two sets. The first set serves as the input $\bar{G}$ to the LEND system to generate a MILS set and 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 non-overlapped MILSes $v_i, i = 1, ..., n$ 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 MILSes, because p may have some additional lossy links that are not covered by these MILSes. Assume 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:

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

ε shows the tolerable value of errors. In our experiments, ε is chosen as 1%. Take one experiment for example, we have 5720 paths in $\bar{G}$ and we choose only 2860 of them to identify 571 MILSes and infer their loss rates. Then we validate the loss rates by the other 2860 paths. 320 out of 571 MILSes are on the paths of the second set, and thus verified by 2200 paths. The result shows that more than 99.1% paths in the second set are consistent with MILSes computed by the first set. This shows that the loss rate inference of the MILSes is accurate.

7.2.2.2 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 which are more likely to be lossy based on some previous experiments. We applied the method introduced in Section 5 to measure 1000 path segments. Then, similar to the cross validation, we adopted Eq. (5) for matching validation. 361 lossy MILSes out of a total of 1664 lossy MILSes are on the 1000 new paths, and thus validated. When using the same parameter $\varepsilon = 0.01$, 94.1% of the loss rates of the new spoofed paths are consistent with the loss rate of these MILSes. 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 thus the IP spoof based consistency checking demonstrates that the MILS loss rate inference is very accurate.

7.2.3 MILS to AS Mapping

After we identify the lossy MILSes, we can locate and study the distribution of the lossy links. For example, are the lossy links usually within an AS or between two ASes?

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 [28]. However, their available IP-to-AS mapping result was obtained from measurement in 2003 and it is incomplete somehow – 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 [29] on March 2nd, 2005. The mapping is quite complete and only 1.6% IPs involved (end hosts and internal routers) cannot be mapped to ASes.

Ignoring these unmapped nodes, we map MILSes to their AS sequences, and then analyze the relationship between lossy links and ASes. Table 6 shows the length of AS paths of the lossy MILSes. Since it is impossible to infer which link or links are lossy in a long MILS, we only consider the short MILSes with length 1 or 2 which consist of about 44% of all lossy MILSes. It is obvious that most lossy links are connecting two different ASes. For example, most length 1 MILSes (27.5% of all MILSes) are connecting two ASes. This observation is consistent with common belief that the links connecting two ASes are more likely to be congested than those within an AS.

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

Table 6: MILS-to-AS path length

7.2.4 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 Figure 4) time is 109.3 seconds, and the online diagnosis (stage 2 in Figure 4) of the 3714 lossy paths for altogether 18,090 paths takes only 4.2 seconds.

8. CONCLUSIONS

In this paper, we advocate the non-biased end-to-end network diagnosis paradigm which gives 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.

9. REFERENCES

[1] Akamai Inc., "Technology overview," `http://www.akamai.com/en/html/technology/overview.html`.

[2] R. Mahajan, N. Spring, D. Wetherall, and T. Anderson, "User-level internet path diagnosis," in *ACM SOSP*, 2003.

[3] K. Anagnostakis, M. Greenwald, and R. Ryger, "cing: Measuring network-internal delays using only existing infrastructure," in *IEEE INFOCOM*, 2003.

[4] M. Coates, A. Hero, R. Nowak, and B. Yu, "Internet Tomography," *IEEE Signal Processing Magazine*, vol. 19, no. 3, pp. 47–65, 2002.

[5] A. Adams et al., "The use of end-to-end multicast measurements for characterizing internal network behavior," in *IEEE Communications*, May, 2000.

[6] T. Bu, N. Duffield, F. Presti, and D. Towsley, "Network tomography on general topologies," in *ACM SIGMETRICS*, 2002.

[7] V. Padmanabhan, L. Qiu, and H. Wang, "Server-based inference of Internet link lossiness," in *IEEE INFOCOM*, 2003.

[8] D. Rubenstein, J. F. Kurose, and D. F. Towsley, "Detecting shared congestion of flows via end-to-end measurement," *ACM Transactions on Networking*, vol. 10, no. 3, 2002.

[9] N. Duffield, "Simple network performance tomography," in *ACM SIGCOMM Internet Measurement Conference (IMC)*, 2003.

[10] Y. Chen, D. Bindel, H. Song, and R. H. Katz, "An algebraic approach to practical and scalable overlay network monitoring," in *ACM SIGCOMM*, 2004.

[11] R. Govindan and H. Tangmunarunkit, "Heuristics for Internet map discovery," in *IEEE INFOCOM*, 2000.

[12] R. Caceres, N. Duffield, J. Horowitz, D. Towsley, and T. Bu, "Multicast-based inference of network-internal characteristics: Accuracy of packet loss estimation," in *IEEE INFOCOM*, 1999.

[13] N.G. Duffield, F.L. Presti, V. Paxson, and D. Towsley, "Inferring link loss using striped unicast probes," in *IEEE INFOCOM*, 2001.

[14] Stefan Savage, "Sting: a TCP-based network measurement tool," in *Proceedings of USITS*, October 1999.

[15] O. Gurewitz and M. Sidi, "Estimating one-way delays from cyclic-path delay measurements," in *IEEE Infocom*, 2001.

[16] R. Caceres, N. Duffield, J. Horowitz, and D. Towsley, "Multicast-based inference of network-internal loss characteristics," *IEEE Transactions in Information Theory*, vol. 45, 1999.

[17] N. Duffield, J. Horowitz, D. Towsley, W. Wei, and T. Friedman, "Multicast-based loss inference with missing data," *IEEE Journal of Selected Areas of Communications*, vol. 20, no. 4, 2002.

[18] C. Tang and P. McKinley, "On the cost-quality tradeoff in topology-aware overlay path probing," in *IEEE ICNP*, 2003.

[19] R.A.Brualdi, A. Pothen, and S. Friedland, "The sparse basis problem and multilinear algebra," *SIAM Journal of Matrix Analysis and Applications*, vol. 16, pp. 1–20, 1995.

[20] Y. Zhang et al., "On the constancy of Internet path properties," in *Proc. of SIGCOMM IMW*, 2001.

[21] Y. Shavitt, X. Sun, A. Wool, and B. Yener, "Computing the unmeasured: An algebraic approach to Internet mapping," in *IEEE INFOCOM*, 2001.

[22] G. W. Stewart, *Matrix Algorithms: Basic Decompositions*, Society for Industrial and Applied Mathematics, 1998.

[23] V. Paxon, "End-to-end routing behavior in the Internet," *IEEE/ACM Transactions on Networking*, vol. 5, no. 5, 1997.

[24] R. Govindan and V. Paxson, "Estimating router icmp generation delays," in *Passive & Active Measurement (PAM)*, 2002.

[25] A. Medina, I. Matta, and J. Byers, "On the origin of power laws in Internet topologies," in *ACM Computer Communication Review*, Apr. 2000.

[26] N. Spring, R. Mahajan, and T. Anderson, "Quantifying the causes of path inflation," in *Proceedings of ACM SIGCOMM*. ACM, 2003.

[27] PlanetLab, "http://www.planet-lab.org/," .

[28] Z. M. Mao and et. al., "Scalable and accurate identification of as-level forwarding paths," in *IEEE Infocom*, 2004.

[29] University of Oregon Route Views Archive Project, "`http://www.routeviews.org/`," .