Tomography-based Overlay Network Monitoring

Yan Chen, David Bindel, Randy H. Katz
Computer Science Division
University of California at Berkeley
Berkeley, CA 94720-1776, USA
{yanchen, dbindel, randy}@cs.berkeley.edu

ABSTRACT

Overlay network monitoring enables distributed Internet applications to detect and recover from path outages and periods of degraded performance within seconds. For an overlay network with n end hosts, existing systems either require $O(n^2)$ measurements, and thus lack scalability, or can only estimate the latency but not congestion or failures. Unlike other network tomography systems, we characterize end-to-end losses (this extends to any additive metrics, including latency) rather than individual link losses. We find a minimal basis set of k linearly independent paths that can fully describe all the $O(n^2)$ paths. We selectively monitor and measure the loss rates of these paths, then apply them to estimate the loss rates of all other paths. By extensively studying synthetic and real topologies, we find that for reasonably large n (e.g., 100), k is only in the range of $O(n \log n)$. This is explained by the moderately hierarchical nature of Internet routing.

Our scheme only assumes the knowledge of underlying IP topology, and any link can become lossy or return to normal. In addition, our technique is tolerant to topology measurement inaccuracies, and is adaptive to topology changes.

Categories and Subject Descriptors

C.2.3 [**Network Operations**]: Network monitoring

General Terms

Measurement, Algorithms

Keywords

Overlay networks, Network measurement and monitoring, Network tomography, Numerical linear algebra

1. INTRODUCTION

With the rapid growth of the Internet, new large-scale globally distributed network services and applications have emerged, such as overlay routing and location systems, application-level multicast, and peer-to-peer file sharing. As these systems have flexibility in choosing their communication paths and targets, they can benefit significantly from dynamic network distance prediction (e.g., latency and loss rate).

Existing network distance estimation systems can be grouped into two categories: *static estimation* [18, 23] and *dynamic monitoring* [13, 8, 3]. Previous static estimation systems, such as Global Network Positioning (GNP) [18], achieve a high level of accuracy, but also incur high overhead for continuously updating the estimates.

Dynamic monitoring can detect path outages and periods of degraded performance within seconds. However, existing schemes either require pair-wise measurements for all end hosts, and thus lack scalability [3]; or they can only estimate latency, but not congestion or failures [13, 8]. Existing scalable systems, such as [13, 8], cluster end hosts based on their network proximity or latency similarity under normal conditions. However, end hosts in the same cluster may not have similar losses, especially when the losses happen in the last mile.

In this paper, we describe a scalable overlay network congestion/failure monitoring system which is highly accurate and incrementally deployable. Consider an overlay network of n end hosts; we define a path to be a routing path between a pair of end hosts, and a link to be an IP link between routers. A path is a concatenation of links. There are $O(n^2)$ paths among the n end hosts, and we wish to select a minimal subset of paths to monitor so that the loss rates and latencies of all other paths can be inferred. The loss rates are used to estimate the congestion/failures on the overlay paths.

To this end, we propose a tomography-based overlay network monitoring system in which we selectively monitor a *basis set* of k paths (typically $k \ll n^2$). Any end-to-end path can be written as a unique linear combination of paths in the basis set. Consequently, by monitoring loss rates for the paths in the basis set, we infer loss rates for all end-to-end paths. This can also be extended to other additive metrics, such as latency. The end-to-end path loss rates can be computed even when the paths contain *unidentifiable links* for which loss rates cannot be computed. We provide an intuitive picture of this characterization process in terms of *virtual links*.

Although congestion outbursts within seconds are hard to detect and bypass, the delay in Internet inter-domain path failovers averages over three minutes [16]. Our loss rate estimation will filter out measurement noise with smoothing

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
IMC'03, October 27–29, 2003, Miami Beach, Florida, USA.
Copyright 2003 ACM 1-58113-773-7/03/0010 ...$5.00.

techniques, such as exponentially-weighted moving average (EWMA), and detect these path failovers quickly to have applications circumvent them.

Our key observation is that k grows relatively slowly as a function of n. The dimension k is bounded by the number of links in the subgraph induced by the routing paths. In an Internet-like topology with a power-law degree distribution, there are $O(N)$ links, where N is the total number of end hosts in the network. This is because a small number of nodes have high degree and the links between them are heavily used [12]. Consequently, if $n = O(N)$, then $k < O(n)$. However, even when $n \ll N$, the moderately hierarchical structure of the network causes many routing paths to overlap [26], so that the number of links in the routing path subgraph grows much slower than $O(n^2)$. Our extensive study of both synthetic and real Internet topologies suggests that for a randomly selected subset of n end hosts, k grows like $O(n \log n)$ when n is sufficiently large (say 100).

Furthermore, our technique is tolerant to topology measurement inaccuracies, and is adaptive to topology changes.

Besides simulating our system with various synthetic and real topologies, we implemented our system on the PlanetLab testbed [22]. We deployed it on 51 global hosts (each from a different organization) and ran the experiments over four weekdays with a total of 76.5M UDP packets. Both simulation and implementation results show we achieve high accuracy when estimating path loss rates with k measurements. For example, the average absolute error of loss rate estimation for the Internet experiments is only 0.0027 with average $k = 872$ out of a total of $51 \times 50 = 2550$ paths. On average, for 248 of the 2550 paths, the routing information obtained via traceroute is unavailable or incomplete, which shows that our technique is robust against topology measurement errors. See our tech report [7] for details on both simulation and experiments on PlanetLab.

The rest of the paper is organized as follows. We survey related work in Sec. 2, describe our model and basic theory in Sec. 3 and present algorithms in Sec. 4. Finally, we discuss the generalization of our framework in Sec. 5 and conclude in Sec. 6.

2. RELATED WORK

Network tomography has been extensively studied ([10] provides a good survey). Most existing systems assume that limited measurement information is available (often in a multicast tree-like structure), and they try to infer the characteristics of the links [1, 2, 6, 20] or shared congestion [24] in the middle of the network.

In many cases, these inferences are limited due to limited measurement and the irregularity of Internet topologies. In contrast, we do not care about the characteristics of *individual* links. Furthermore, we do not have any restriction on the paths to measure. Our goal is to selectively measure a small subset of paths so that we can infer the loss rates of all other paths.

As the closest work to ours, Shavitt *et al.* also use algebraic tools to compute the distances that are not explicitly measured [25]. Given certain "Tracer" stations deployed and some direct measurements among the Tracers, they search for path or path segments whose loss rates can be inferred from these measurements. Thus their focus is not on Tracer/path selection. Neither do they examine the topology measurement errors or the topology change problems.

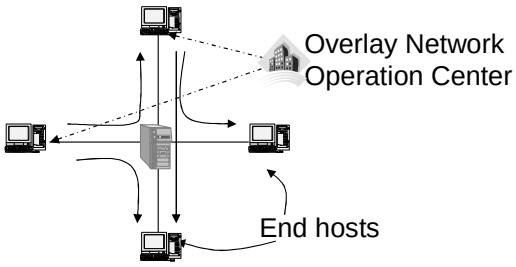

Figure 1: Architecture of a tomography-based overlay network monitoring system

Recently, Ozmutlu *et al.* selected a minimal subset of paths to cover all links for monitoring, assuming link-by-link latency is available via end-to-end measurement [19]. Their approach has the following three limitations. 1) Traceroute cannot give accurate link-by-link latency. Many routers in the Internet hide their identities. Besides, traceroute uses the ICMP protocol for measurement, and routers often treat ICMP packet differently from TCP/UDP packets. Therefore, latency data is not representative. 2) It is not applicable for loss rate, because it is difficult to estimate link-by-link loss rates from end-to-end measurements. Loss rate is often more important for applications than latency. 3) It assumes static routing paths and does not consider topology changes.

Many of the previous findings can be leveraged to refine loss rate prediction. For example, [20] finds that the end-to-end losses are dominated by a small number of lossy links. Thus, the path space to be monitored can be reduced to those paths that include lossy links. Consequently, the basis set and the amount of measurement will be reduced.

3. THE MODEL

In this section, we develop the model for tomography-based overlay monitoring.

Given n end hosts to be monitored, we assume that they belong to an overlay network (such as a virtual private network), or that they cooperate to share the monitoring services. Thus, we can measure the routing topology and loss rate of any path. The end hosts are under the control of a central authority (e.g., an overlay network operation center (ONOC)) to measure the topology and loss rates of paths, though in the future we plan to investigate techniques to distribute the work of the central authority.

For simplicity, we mostly assume symmetric routing and undirectional links in the paper. But our techniques work without changes for asymmetric routing, as used in the Internet experiments. Fig. 1 shows an example where there are four end hosts on the overlay network. There are six paths and four links. The end hosts measure the topology and report to the ONOC, which selects four paths and instruments two of the end hosts to measure the loss rates of the four paths. The end hosts periodically report the loss rates measured to the ONOC. Then the ONOC infers the loss rates of every link, and consequently the loss rates of the other two paths. Applications can query the ONOC for the loss rate of any path, or they can set up triggers to receive alerts when the loss rates of paths of interest exceed a certain threshold.

The path loss rates can be measured by either passive observation of normal traffic to estimate packet drop rate [20] or active measurement. The measurements of selected paths

do not have to be taken at exactly the same time because Zhang *et al.* report that the loss rate remains operationally stable in the time scale of an hour [27]. The network topology can be measured via traceroute or other advanced tools [15, 9]. We discuss topology changes in Sec. 4.4.

3.1 Theory and Notations

Symbols	Meanings
M	total number of nodes
N	number of end hosts
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
t	number of identifiable links
$G \in \{0,1\}^{r \times s}$	original path matrix
$\bar{G} \in \{0,1\}^{k \times s}$	reduced path matrix
$k \leq s$	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
$\mathcal{N}(G)$	null space of G
$\mathcal{R}(G^T)$	row(path) space of G (== range(G^T))

Table 1: Table of notations

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 probability p of packet loss on the path represented by v is given by

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

By taking logarithms on both sides of (1), we have

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

If we define a column vector $x \in \mathbb{R}^s$ with elements $x_j := \log(1 - l_j)$, and write v^T for the row vector which is the transpose of v, we can rewrite (2) in the following dot product form:

$$\log(1 - p) = \sum_{j=1}^{s} v_j x_j = v^T x \tag{3}$$

Considering all $r = O(n^2)$ paths in the overlay network, there are r linear equations of the form (3). Putting them together, we form a rectangular matrix $G \in \{0,1\}^{r \times s}$ to represent 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 probability of packet loss during transmission on 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 system of equations relating the link losses to path losses as

$$Gx = b \tag{4}$$

In general, the measurement matrix G may be rank deficient: i.e., $k = \text{rank}(G)$ and $k < s$. If G is rank deficient, we will be unable to determine the loss rate of some links from (4). We call these links *unidentifiable* as in [6].

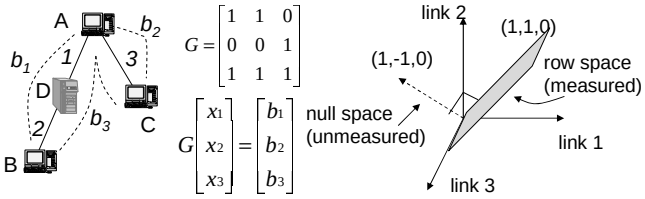

Figure 2: Sample overlay network.

We illustrate how rank deficiency can occur in Fig. 2. There are three end hosts (A, B, and C) on the overlay, three links (1, 2 and 3) and three paths between the end hosts. Because links 1 and 2 always appear together, their individual loss rates cannot be computed from the measurements. For example, suppose that $x_1 + x_2 = b_1 = -0.06$ and $x_3 = b_2 = -0.01$. We know that $x_1 = -0.03 + \alpha$ and $x_2 = -0.03 - \alpha$ for some α, but the value of α cannot be determined from the end-to-end measurements. The set of vectors $\alpha \begin{bmatrix} 1 & -1 & 0 \end{bmatrix}^T$ which are not defined by (4) can be added to x without affecting b. This set of vectors is the *null space* of G.

To separate the identifiable and unidentifiable components of x, we write x as $x = x_G + x_N$, where $x_G \in \mathcal{R}(G^T)$ is in the *row space* of G and $x_N \in \mathcal{N}(G)$ is in the orthogonal *null space* of G (i.e. $Gx_N = 0$). The vector x_G contains all the information we can know from (4) and the path measurements. For instance, we can determine $x_1 + x_2$ in Fig. 2, but not $x_1 - x_2$. Intuitively, links 1 and 2 together form a single *virtual link* with an identifiable loss rate $x_1 + x_2$. All end-to-end paths can be written in terms of such *virtual links*, as we describe in more details in Sec. 3.3. So x_G involves all the links, while x_N only involves unidentifiable links. The decomposition of x for the sample overlay network is shown below.

$$x_G = \frac{(x_1 + x_2)}{2} \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} + x_3 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} b_1/2 \\ b_1/2 \\ b_2 \end{bmatrix} \tag{5}$$

$$x_N = \frac{(x_1 - x_2)}{2} \begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix} \tag{6}$$

Because x_G lies in the k-dimensional space $\mathcal{R}(G^T)$, only k independent equations of the r equations in (4) are needed to uniquely identify x_G. By measuring k independent paths, we can compute x_G. Since $b = Gx = Gx_G + Gx_N = Gx_G$, we can compute all elements of b from x_G, and thus obtain the loss rate of all other paths. For example, in Fig. 2, we only need to measure b_1 and b_2 to compute x_G, from which we can calculate b_3. Detailed algorithms are described in Sec. 4.

3.2 Dimension Analysis of Path Space ($\mathcal{R}(G^T)$)

In this section, we will examine asymptotically how big k is in terms of n.

THEOREM 1. *Given a power-law degree network topology of M nodes, the frequency f_d of nodes with outdegree d is*

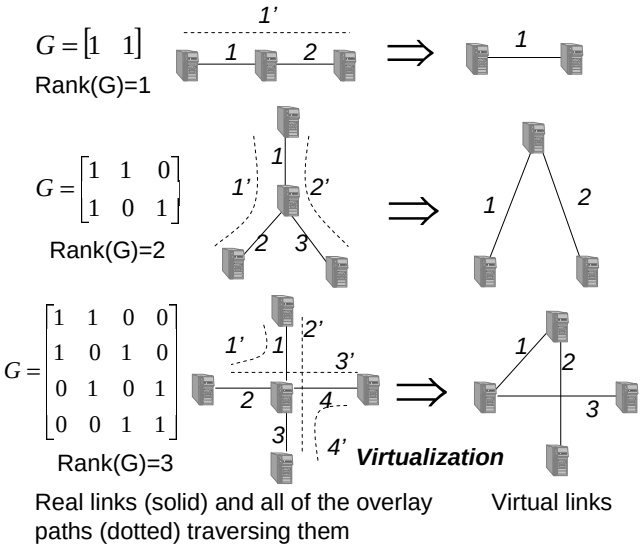

Figure 3: Sample parts of IP network and overlay paths.

proportional to d^c, where c is the outdegree exponent constant (i.e., $f_d \propto d^c$). With $d \geq 1$ and $c < -2$ (as found in [12]), the number of end hosts N is at least $M/2$.

See the Appendix for the proof. It also follows the intuition that the number of end hosts should be more than the number of routers in the Internet.

Meanwhile, Faloutsos *et al.* prove that such a topology has only $O(M)$ links (Lemma 2 in [12]). Combining the two facts, given N end hosts, there are at most $O(N)$ links in the topology. Thus, if the majority of the end hosts are on the overlay network ($n = O(N)$), the dimension of $\mathcal{R}(G^T)$ is $O(n)$.

What about if only a small portion of the end hosts are on the overlay? Tangmunarunkit *et al.* found that the power-law degree Internet topology has moderate hierarchy due to the heavy-tailed degree distribution [26]. Because G is an r by s matrix, k is bounded by the number of links s. If it is a strict hierarchy like a tree, $s = O(n)$, thus $k = O(n)$. But if there is no hierarchy at all (e.g., clique), $k = O(n^2)$ because all the $O(n^2)$ paths are linearly independent. Moderate hierarchy should fall in between. We found that for reasonably large n (e.g, 100), $k = O(n \log n)$. Refer to our tech report [7] for full regression analysis and results.

3.3 Intuition through Virtual Links

In Sec. 3.1, we explain in algebraic terms how to compute all end-to-end path loss rates from only k path measurements. Our actual computations are based completely on this algebraic picture; however, these formulas may not seem intuitive. We now describe a more intuitive picture using the notion of *virtual links*. The key idea is that although the loss rates of some individual links are incomputable (unidentifiable links), each of them is covered by some path segment whose loss rate is computable, and the loss rates of these path segments are sufficient to compute the path loss rates in which we are interested.

We choose a minimal set of such path segments that can fully describe all end-to-end paths, and refer to them as *virtual links*. If a link is identifiable, the link itself is a virtual link.

Fig. 3 illustrates some examples. In the top figure, the virtual link is a concatenation of two sequential physical links as we discussed before. In the middle figure, there are three links, but only two paths traverse these links. Thus, rank(G) = 2 and none of the links are identifiable. In the bottom figure, there are four links, and a total of four paths traversing them. But the four paths are linearly dependent, so rank(G) = 3, and none of the link loss rate are computable. We can use any three out of the four paths as virtual links, and the other one can be linearly represented by the virtual links. For example, path $4'$ can be described as virtual links 2+3-1.

Since the dimension of $\mathcal{R}(G^T)$ is k, the minimum number of virtual links which can fully describe $\mathcal{R}(G^T)$ is also k. x_G is a linear combination of the vectors representing the virtual links. Since virtual links are identifiable, x_G is also computable. From x_G, we can compute the loss rates of all end-to-end paths as we can do with virtual links.

4. ALGORITHMS

In this section, we describe implementation techniques.

4.1 Selecting Measurement Paths

To characterize all $O(n^2)$ end-to-end paths, we monitor k linearly independent end-to-end paths and form a reduced system

$$\bar{G} x_G = \bar{b} \tag{7}$$

where $\bar{G} \in \{0,1\}^{k \times s}$ and $\bar{b} \in \mathbb{R}^k$ consist of k rows of G and b, respectively. Linearly independent sets of rows and columns in rank-deficient problems are usually computed using *rank-revealing decompositions* [14]. For a dense r by s matrix with rank k, common rank-revealing decompositions include Gaussian elimination with complete pivoting (as used in [25]), QR with column pivoting, and the singular value decomposition (SVD). The former two cost $O(rks)$, and the SVD costs $O(rs^2)$. Our G matrix is very sparse; that is, there are only a few nonzeros per row. Rank-revealing decompositions for many sparse problems can be computed much more quickly than in the dense case. However, the exact cost depends strongly on the structure of the problem, and efficient computation rank-revealing decompositions of sparse matrices is an open area of research [17], [21].

We select rows using Algorithm 1, which is a variant of the QR procedure [14, p.223]. The procedure incrementally builds a decomposition

$$\bar{G}^T = QR \tag{8}$$

where $Q \in \mathbb{R}^{s \times k}$ is a matrix with orthonormal columns and $R \in \mathbb{R}^{k \times k}$ is upper triangular. We do not store Q explicitly; instead, we write Q as $R^{-1}\bar{G}^T$. The idea is the same as the classical Gram-Schmidt algorithm: as each row is inspected, we subtract off any components in the space spanned by the previous rows, so that the remainder is orthogonal to all previous rows. If the remainder is zero, then the row was linearly dependent upon the previous rows; otherwise, we extend the factorization.

In practice, we use a variant of Algorithm 1 which uses optimized routines from the LAPACK library [4] and inspects several rows at a time. The time complexity of processing each vector is dominated by the solution of a triangular system to compute $\hat{R}_{12}$, which costs $O(k^2)$. The total cost of the algorithm is $O(rk^2)$ and the constant in the bound is

```
   procedure SelectPath(G)
1  for every row v in G do
2      R̂12 := R^{-T} Ḡ v^T = Q^T v^T
3      R̂22 := ||v||^2 − ||R̂12||^2
4      if R̂22 ≠ 0 then
5          Mark v as a measurement path
6          Ḡ := [Ḡ; v]
7          R := [R  R̂12; 0  R̂22]
       end
   end
```

Algorithm 1: Path (row) selection algorithm

modest: on a Pentium 4 running at 1.5 GHz, our code takes just over ten minutes to process a problem with $n = 350$ ($r = 61075$) and $k = 2958$. The memory cost is roughly $k^2/2$ single-precision floating point numbers for storing the R factor.

When k exceeds 10000 the $O(k^2)$ memory requirement becomes too onerous. We note that dense factorization methods may still be feasible if the number of overlay end-hosts is small or if we relax our original problem statement.

4.2 Path Loss Calculations

The QR decomposition which we use to select measurement paths is also used to compute a solution to the underdetermined system (7). To choose a unique solution x_G to $\bar{G}x_G = \bar{b}$, we impose the additional constraint that $x_G = \bar{G}^T y$. We can then compute

$$\begin{aligned} y &:= R^{-1}R^{-T}\bar{b} \\ x_G &:= \bar{G}^T y. \end{aligned}$$

This is a standard method for finding the minimum norm solution to an underdetermined system (see [14], [11]). The dominant cost in the computation is the solution of two triangular linear systems for y, which costs $O(k^2)$. Once we have computed x_G, we can compute $b := Gx_G$, and from there infer the loss rates of the remaining paths.

4.3 Topology Measurement Error Tolerance

Our technique is tolerant to network topology measurement errors because our goal is to estimate the end-to-end path loss rate instead of any interior link loss rate. For example, poor alias resolution of routers may present one physical link as several links. At worst, our failure to recognize the links as the same will result in a few more path measurements because the rank of G is higher. But we can still get accurate path loss rate estimation as verified by Internet experiments in [7].

4.4 Topology Changes

During normal operation, new links may appear or disappear, routing paths between end hosts may change, and hosts may enter or exit the overlay network. These changes may cause rows or columns to be added to or removed from G, or entries in G may change. We designed a set of efficient algorithms to add/remove end hosts and to handle routing changes. We incrementally add/remove paths from G and $\bar{G}$, and each path change takes at most $O(k^2)$ time (see [7]).

4.5 Robustness and Real-time Response

There are some scenarios such that the overlay monitoring system can fail to provide real-time loss rate estimation for some paths. This can happen when a routing change is just detected, or the measurement node(s) crash, or some node(s) just join or leave the overlay network. Before we incrementally set up new measurement path(s) and collect results, for a short period, there are some paths for which we can not compute loss rates. However, we can still return bounds on the computed loss rate (see Sec. 5). For example, we can check whether all the links on the incomputable path are covered by $\bar{G}$, and if so, yield an upper bound (though possibly a pessimistic one) quickly. Furthermore, such bounds may be already sufficient for some applications.

5. DISCUSSION

In this section, we generalize our framework to infer the path loss rate bound when we have only restricted measurements.

We note that, in addition to the equations (4), the unknown x_j must satisfy the inequalities $x_j \leq 0$. While we do not make use of them in our current work, these inequalities can be used in conjunction with (4) to bound failure probabilities, both from below and from above. For example, the loss probability l_j is bounded above by the loss probability of the least lossy path that includes link j. More generally, we have the following theorem:

THEOREM 2. *Let $v \in \{0,1\}^s$ represent a network path with loss probability p, and let $w = G^T c$ for some $c \in \mathbb{R}^r$ (i.e. $w \in \mathcal{R}(G^T)$). Then*

1. *If $v \leq w$ elementwise, then $\log(1-p) \geq c^T b$*
2. *If $v \geq w$ elementwise, then $\log(1-p) \leq c^T b$*

PROOF. In the first case, $v \leq w$ so that $v - w \leq 0$ elementwise. Since $x \leq 0$ elementwise, $(v-w)^T x \geq 0$, or $v^T x \geq w^T x$. We know $\log(1-p) = v^T x$ from (3), and $w^T x = c^T G x = c^T b$. By substitution, we have $\log(1-p) \geq c^T b$. The second case is nearly identical. □

In principle, we can compute good upper and lower bounds on path loss rates by solving two linear programming problems:

1. Maximize $c_u^T b$ subject to $G^T c_u \geq v$,
2. Minimize $c_l^T b$ subject to $G^T c_l \leq v$.

Then $1-\exp(c_l^T b) \leq p \leq 1-\exp(c_u^T b)$. When $v \in \mathcal{R}(G^T)$, we have $v = G^T c_u = G^T c_l$, and the bound is tight. While this approach seems to offer bounds on path loss probabilities that are possibly optimal given the measured data, we have not yet applied the technique in practice.

6. CONCLUSIONS

In this paper, we present a tomography-based overlay network monitoring system. For an overlay of n end hosts, the space of $O(n^2)$ paths can be characterized by a basis of $O(n \log n)$ paths. We selectively monitor these basis paths, then use the measurements to infer the loss rates of all other

paths. Both simulation and real implementation on the Internet show that our techniques achieve accurate loss rate estimation.

For more efficient monitored path selection, we plan to investigate the use of iterative methods [5], [17] such as CGNE or GMRES both to select rows and to compute loss rate vectors. In our preliminary experiments, the path matrix G has been well-conditioned, which suggests that iterative methods may converge quickly. We are also applying the inequality bounds in Sec. 5 for diagnostics, to detect which links or path segments fail when end-to-end congestion occurs.

7. REFERENCES

[1] Adams, A., et al. The use of end-to-end multicast measurements for characterizing internal network behavior. In *IEEE Communications* (May, 2000).

[2] Adler, M., et al. Tree layout for internal network characterizations in multicast networks. In *3rd International Workshop on Networked Group Communication (NGC)* (2001).

[3] Andersen, D. G., Balakrishnan, H., Kaashoek, M. F., and Morris, R. Resilient overlay networks. In *Proc. of ACM SOSP* (2001).

[4] Anderson, E., et al. *LAPACK Users' Guide*, third ed. Society for Industrial and Applied Mathematics, Philadelphia, PA, 1999.

[5] Barrett, R., et al. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods, 2nd Edition*. SIAM, Philadelphia, PA, 1994.

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

[7] Chen, Y., Bindel, D., Song, H., and Katz, R. H. Tomography-based overlay network monitoring. Tech. Rep. UCB//CSD-03-1252, University of California, Berkeley, 2003.

[8] Chen, Y., Lim, K., Overton, C., and Katz, R. H. On the stability of network distance estimation. In *ACM SIGMETRICS Performance Evaluation Review (PER)* (Sep. 2002).

[9] Coates, M., Castro, R., and Nowak, R. Maximum likelihood identification of network topology from edge-based unicast measurements. In *ACM SIGMETRICS* (2002).

[10] Coates, M., Hero, A., Nowak, R., and Yu, B. Internet Tomography. *IEEE Signal Processing Magazine 19*, 3 (2002), 47–65.

[11] Demmel, J. *Applied Numerical Linear Algebra*. SIAM, 1997.

[12] Faloutsos, M., Faloutsos, P., and Faloutsos, C. On power-law relationship of the Internet topology. In *ACM SIGCOMM* (1999).

[13] Francis, P., et al. IDMaps: A global Internet host distance estimation service. *IEEE/ACM Trans. on Networking* (Oct. 2001).

[14] Golub, G., and Loan, C. V. *Matrix Computations*. The Johns Hopkins University Press, 1989.

[15] Govindan, R., and Tangmunarunkit, H. Heuristics for Internet map discovery. In *IEEE INFOCOM* (2000).

[16] Labovitz, C., Ahuja, A., Abose, A., and Jahanian, F. An experimental study of delayed Internet routing convergence. In *Proc. of ACM SIGCOMM* (2000).

[17] Meyer, C., and Pierce, D. Steps toward an iterative rank-revealing method. Tech. Rep. ISSTECH-95-013, Boeing Information and Support Services, 1995.

[18] Ng, T. S. E., and Zhang, H. Predicting Internet network distance with coordinates-based approaches. In *Proc.of IEEE INFOCOM* (2002).

[19] Ozmutlu, H. C., Gautam, N., and Barton, R. Managing end-to-end network performance via optimized monitoring strategies. *Journal of Network and System Management, Special Issue on Management of Converged Networks 10*, 1 (2002).

[20] Padmanabhan, V., Qiu, L., and Wang, H. Server-based inference of Internet performance. In *IEEE INFOCOM* (2003).

[21] Pierce, D., and Lewis, J. Sparse multifrontal rank revealing QR factorization. *SIAM Journal on Matrix Analysis and Applications 18*, 1 (January 1997).

[22] PlanetLab. http://www.planet-lab.org/.

[23] Ratnasamy, S., et al. Topologically-aware overlay construction and server selection. In *Proc. of IEEE INFOCOM* (2002).

[24] Rubenstein, D., Kurose, J. F., and Towsley, D. F. Detecting shared congestion of flows via end-to-end measurement. *IEEE/ACM Transactions on Networking 10*, 3 (2002).

[25] Shavitt, Y., Sun, X., Wool, A., and Yener, B. Computing the unmeasured: An algebraic approach to Internet mapping. In *IEEE INFOCOM* (2001).

[26] Tangmunarunkit, H., et al. Network topology generators: Degree-based vs structural. In *ACM SIGCOMM* (2002).

[27] Zhang, Y., et al. On the constancy of Internet path properties. In *Proc. of SIGCOMM Internet Measurement Workshop* (2001).

APPENDIX

Proof for Theorem 1

Proof. Given that the power-law distribution topology has out-degree exponent: the frequency f_d of an outdegree d is proportional to the outdegree to the power of a constant, i.e., $f_d = Nd^c$, where N is the proportion constant. Assume that end hosts have degree 1, then the number of end hosts is N.

If $c < -1$, then

$$M = N \sum_{d=1}^{M-1} d^c \tag{9}$$

$$\leq N \left(1 + \int_1^{M-1} x^c \, dx\right) \tag{10}$$

$$\leq N \left(1 + \int_1^{\infty} x^c \, dx\right) \tag{11}$$

$$= N \left(1 - \frac{1}{1+c}\right) \tag{12}$$

$$= N \frac{c}{1+c} \tag{13}$$

Therefore, the fraction $\frac{N}{M}$ is at least $\frac{1+c}{c} = 1 + \frac{1}{c}$. If $c \leq -2$ then $\frac{N}{M} \geq \frac{1}{2}$. □