Identifying Lossy Links in Wired/Wireless Networks by Exploiting Sparse Characteristics

Hyuk Lim and Jennifer C. Hou

Abstract

In this paper, we consider the problem of estimating link loss rates based on end-to-end path loss rates in order to identify lossy links on the network. We first derive a maximum likelihood estimate for the problem and show that the problem boils down to the matrix inversion problem for an under-determined system of linear equations. Without any prior knowledge of the statistics of packet loss rates, most of the existing work uses the minimum norm solution for the under-determined linear system. We devise, under the assumption that link failures are abnormal events in real networks and lossy links are sparse among all the internal links, an iterative algorithm to identify non-lossy links and to remove the corresponding terms from the under-determined linear system. To identify non-lossy links, we propose to use three different criteria (and a combination thereof): the criterion determined by a basis selection technique, that obtained by sorting path loss rates, and that determined by the minimum norm least square solution. We show via simulation and empirical studies on the MIT *Roofnet* traces that the computational complexity of the iterative algorithm is comparable to that of the minimum norm least square approach, and that the solution obtained under the iterative algorithm achieves high coverage of lossy links, while incurring only a small number of false positives in various network scenarios.

Index Terms

Network tomography, packet loss rate, and sparse distribution

H. Lim is with the Department of Information and Communications, Gwangju Institute of Science and Technology, Gwangju 500-712, Korea. Email: hlim@gist.ac.kr

J. C. Hou is with the Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA. Email: jhou@cs.uiuc.edu.

I. Introduction

Fundamental network characteristics such as the network delay and the packet loss rate provide important information on the operational conditions of a network. They are also highly correlated to events perceived by network operators and end users. Acquiring network characteristics for internal links is, however, quite difficult, due to the fact that most core routers and switches do not support direct access to the links. They are usually inferred by observing and analyzing end-to-end measurements, the techniques of which are usually termed as *network tomography*. For example, the network delay experienced at each router can be inferred by using the routing information (available in the routing table) and the end-to-end measurements of paths delays. Indeed most network tomography techniques are grounded on mathematical inference that estimates the source signal of interest (i.e., network characteristics of internal links) with a set of sampled observations (i.e. end-to-end measurement data).

Network tomography has received significant attention. According to the way in which measurements are made, existing work can be roughly classified into those that send multicast packets to probe the network [2], [14], those that send unicast packets to probe the network [3], [6], [8], [10], [13], [16], and those that employ passive approaches to monitor data traffic (without sending probing packets) [12], [13], [16]. Various mathematical techniques such as maximum likelihood estimation (MLE) [2], [7], expectation-maximization (EM) [6], [16], and Bayesian inference [12], [13], [15] have been exploited as the inference base for network tomography.

One of the major challenges in all the aforementioned network tomography problems is that the mapping from the observed measurements to the corresponding link-level characteristics cannot in general be uniquely determined [5]. One has to utilize *additional* statistical information on the link-level characteristics. For example, in the case that unicast is used to probe the network, the number of measurement paths is generally smaller than that of internal links. As a result, the statistics obtained by measuring end-to-end path characteristics is not sufficient to uniquely determine the link-level network characteristics. To obtain additional statistical information, several measurement methods, such as those in [6], [10], send back-to-back packet pairs and exploit the conditional probability of packet transmission. Zhang *et al.* [23], on the other hand, exploit statistical independence to estimate the traffic matrix between origin and destination pairs.

In this paper, we consider the problem of inferring packet loss rates of internal links and identifying lossy links given a set of end-to-end measurements of path loss rates (and the routing information). We first derive a simple maximum likelihood estimate (MLE) to this problem, and show that the problem essentially boils down to solving an under-determined system of linear equations (that characterizes the relationship between link loss rates and path loss rates). We would like to stress that, although similar network tomography problems have been considered in the form of an under-determined system, this is the first effort to show that MLE is *implied* in the under-determined linear system. While most existing work [3], [13] uses the minimum norm solution as the solution to this under-determined linear system, we show that there is room for further improvement. We exploit the statistical observation that lossy links are sparse in real operational networks and formulate a new optimization problem that minimizes the number of lossy links, subject to satisfying the under-determined linear system. This statistical observation has been corroborated by empirical studies in [3], [21], in which the authors showed the distribution of the packet loss rate is centered near zero even though it has a wide range. The sparsity property of lossy links has also been *implicitly* used in [8], [13] in its design of the inference algorithm (in a different context).

Finding an optimal solution to the optimization problem that minimizes the number of lossy links, subject to the under-determined linear system is NP hard and requires combinational search [4]. We propose an iterative algorithm to solve the optimization problem by identifying non-lossy links and pruning columns that correspond to non-lossy links from the matrix that characterizes the under-determined linear system. The process of identifying non-lossy links is performed using three different criteria (and a combination thereof) determined by three different methods: basis selection, sorting of path loss rates, and solving the minimum norm least square problem. We show via simulation and empirical studies on the MIT *Roofnet* traces that the sparse solution obtained under the iterative algorithm achieves high *coverage* and incurs a small number of *false positives* under various network scenarios. Here the coverage is defined as the ratio of the number of links correctly identified to be lossy to that of real lossy links, and a false positive occurs when a non-lossy link is incorrectly identified to be lossy.

The rest of the paper is organized as follows. In Section II, we infer link loss rates based on maximum likelihood estimation (MLE) and show that the problem essentially boils down to solving an under-determined linear system. In Section III, we give a summary of related work in the literature, and motivate the need for our work. In Section IV, we validate the assumption that lossy links are sparse in real operational networks, and propose an iterative algorithm to solve the new optimization problem that minimizes the number of lossy links, satisfying the under-determined linear system. Following that, we present in Section V and VI our simulation and experiment studies that evaluate the proposed iterative algorithm in terms of computational overhead, coverage (i.e., the number of correctly inferred lossy links), and false positives (i.e., the number of incorrectly inferred lossy links). Finally, we conclude the paper in Section VII.

II. Preliminaries

The network tomography problem considered in the paper is to infer the link loss rates in a network by observing the end-to-end path loss rates between end hosts. The packet loss rate is defined as the ratio of the number of successfully transmitted packets to the total number of packets during a measurement interval. Note that packet loss rates dynamically change over time and cannot be simultaneously measured at all the links, and hence for problem tractability, we focus on the first-order statistics of packet loss rates. The assumption of stationary behavioral characteristics over a measurement interval has been corroborated by rigorous, empirical studies given in [3], [13].

A. Problem Formulation Based on Maximum Likelihood Estimation

Consider a network $\mathcal{N}$ that consists of a set L of unidirectional links indexed by $l = 1, \cdots, |L| \triangleq n$ and a set S of source-destination directed paths indexed by $s = 1, \cdots, |S| \triangleq m$. Here $|\cdot|$ denotes the cardinality of a set.

Each path s is composed of a set $L(s) \subset L$ of links. The sets $L(s)$, $s \in S$ define an m-by-n routing matrix $\mathbf{A}$ whose elements are

$$a_{sl} = \begin{cases} 1, & \text{if } l \in L(s), \\ 0, & \text{otherwise}. \end{cases}$$

This routing matrix $\mathbf{A}$ can be obtained by several techniques based on 'traceroute' and is assumed to be given in this paper.

Under this network model, we infer link loss rates based on maximum likelihood estimation (MLE) and show that the problem essentially boils down to solving an under-determined linear system. Let t_s and f_s denote the numbers of received and lost packets along a path s, respectively.

Then, the observation data set is defined as $O = \bigcup_{s \in S} (t_s, f_s)$. Let the set of link loss rates to be estimated be denoted as $P = \{p_1, \cdots, p_n\}$, where p_l is the loss rate at link l.

Under the assumption of a Bernoulli loss process, the likelihood function for a single observation can be written as

$$\Pr(O|P) = \prod_{s \in S} (1 - p_s)^{t_s} p_s^{f_s}, \tag{1}$$

where p_s is the loss rate of path s and can be expressed as

$$p_s = 1 - \prod_{l \in L(s)} (1 - p_l). \tag{2}$$

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

$$\begin{aligned} \ln \Pr(O|P) &= \sum_{s \in S} [t_s \ln(1 - p_s) + f_s \ln(p_s)] \\ &= \sum_{s \in S} \left[t_s \sum_{l \in L(s)} x_l + f_s \ln \left(1 - \prod_{l \in L(s)} e^{x_l} \right) \right], \end{aligned}$$

where $x_l = \ln(1 - p_l)$. In the matrix form, the log-likelihood function can be expressed as

$$\ln \Pr(O|P) = \mathbf{t}^T \mathbf{A} \mathbf{x} + \mathbf{f}^T \ln \left(\mathbb{1}_m - e^{\mathbf{A} \mathbf{x}} \right), \tag{3}$$

where the column vectors of $\mathbf{x}$, $\mathbf{t}$, and $\mathbf{f}$ are defined as $\mathbf{x} = [x_1, \cdots, \ \ x_n]^T$, ,$\mathbf{t} = [t_1, \cdots, t_m]^T$, and $\mathbf{f} = [f_1, \cdots, f_m]^T$, respectively, and $\mathbb{1}_m \in \mathbb{R}^m$ is the 1's column vector .

By replacing the variable $\mathbf{y} = \mathbf{A}\mathbf{x}$ in Eq. (3) and differentiating it with respect to $\mathbf{y}$, we have

$$\frac{\partial \ln \Pr(O|P)}{\partial y_s} = t_s - f_s \frac{e^{y_s}}{1 - e^{y_s}}.$$

Setting the above equation to zero, we obtain the value of y_s that maximizes $\ln \Pr(O|P)$:

$$y_s = \ln\left(\frac{t_s}{t_s + f_s}\right) = \ln(1 - p_s) \quad \text{for} \quad s \in S.$$

Thus the maximum likelihood estimate of $\mathbf{x}$ is the solution of the following linear equation:

$$\mathbf{y} = \mathbf{A}\mathbf{x}. \tag{4}$$

Alternatively, the system of linear equations in Eq. (4) can be obtained by taking logarithms on Eq. (2):

$$\ln(1 - p_s) = \sum_{l \in L(s)} \ln(1 - p_l) = \sum_{l \in L} a_{sl} \ln(1 - p_l),$$

and

$$y_s = \sum_{l \in L} a_{sl} x_l,$$

which leads to $\mathbf{y} = \mathbf{A}\mathbf{x}$.

In summary, the problem of inferring link loss rates, $\{p_l \mid l \in L\}$, based on maximum likelihood estimation with the likelihood function Eq. (1), eventually boils down to solving the system of linear equations in Eq. (4). If the system of linear equations is under-determined, MLE has an infinite number of solutions.

B. Solving Under-Determined Linear Equations

The tomography problem can be stated as follows: Given a routing matrix $\mathbf{A}$ and the packet loss rate, $\mathbf{y} \in \mathbb{R}^m$, measured by end hosts on an end-to-end basis, infer the packet loss rate of each link $\mathbf{x} \in \mathbb{R}^n$ such that $\mathbf{y} = \mathbf{A}\mathbf{x}$. As a special case, if $\mathbf{A}$ is a square matrix with full rank, then there exists an inverse matrix of $\mathbf{A}$, and we have a unique solution of $\mathbf{x} = \mathbf{A}^{-1}\mathbf{y}$.

In general, the number, n, of links is larger than the number, m, of paths measured end-to-end, and the number of unknown variables is larger than that of linear constraints. That is, $\mathbf{y} = \mathbf{A}\mathbf{x}$ is under-determined, and may have an infinite number of solutions. For example, in a server-client measurement scheme, a tree rooted at the server is constructed with all the clients as leaves. The number of links is usually larger than that of paths in the tree. In an overlay network of k nodes, although the number of paths, k^2, is larger than the number of links, as reported in [3], the rank of $\mathbf{A}$ is not proportional to k but is much smaller than the number of links. As a result, the linear system is usually reduced to a smaller under-determined linear system (with full ranks) by selecting measurements on *independent* paths.

To solve such an under-determined linear system, it is necessary to impose additional conditions for selecting a solution. One possible criterion (that has been widely used) is to select the solution that renders a minimal L_p norm. For example, if L_2 norm is used as the criterion, the problem can be written as

$$\begin{aligned} &\text{minimize } \mathbf{x}^T\mathbf{x} \\ &\text{subject to } \mathbf{A}\mathbf{x} = \mathbf{y}, \end{aligned} \tag{5}$$

and the optimizer x^* is called the *minimum norm solution* of the undetermined system.

If $\mathbf{A}$ is full rank (i.e., $m = \text{rank}(\mathbf{A})$ and $m < n$), there exists an inverse matrix of $\mathbf{A}\mathbf{A}^T$, and the optimizer can be shown to be $x^* = \mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1}\mathbf{y}$. Here, $\mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1}$ is the (right) pseudo-inverse matrix of $\mathbf{A}$. On the other hand, if $\mathbf{A}$ is rank deficient (i.e., $m > \text{rank}(\mathbf{A})$), $\mathbf{A}\mathbf{A}^T$ is singular, and the *minimum norm least square solution* can be obtained by singular value decomposition (SVD) [11]. Let the singular value decomposition of $\mathbf{A}$ be expressed as

$$\mathbf{A} = \mathbf{U} \cdot \begin{bmatrix} \mathbf{\Sigma} & 0 \\ 0 & 0 \end{bmatrix} \cdot \mathbf{V}^T, \tag{6}$$

where $\mathbf{U}$ and $\mathbf{V}$ are column and row orthogonal matrices:

$$\begin{aligned} \mathbf{U} &= [\mathbf{u}_1, \cdots, \mathbf{u}_m], \\ \mathbf{V} &= [\mathbf{v}_1, \cdots, \mathbf{v}_n], \end{aligned}$$

$\mathbf{\Sigma}$ is a diagonal matrix:

$$\mathbf{\Sigma} = \text{diag}(\sigma_1, \cdots, \sigma_W),$$

the subscript *w* is the rank of the matrix $\mathbf{A}$, and σ_i's are singular values of $\mathbf{A}$ in the decreasing order (i.e., $\sigma_1 \geq \cdots \geq \sigma_W > 0$). Then the matrix $\mathbf{A}^+$ is defined as

$$\begin{aligned} \mathbf{A}^+ &= \mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1} = \mathbf{V} \cdot \begin{bmatrix} \mathbf{\Sigma}^{-1} & 0 \\ 0 & 0 \end{bmatrix} \cdot \mathbf{U}^T \\ &= \sum_{i=1}^{W} \frac{1}{\sigma_i} \mathbf{v}_i \mathbf{u}_i^T. \end{aligned} \tag{7}$$

With the use of $\mathbf{A}^+$, the solution to Eq. (5) is simply expressed as $\mathbf{x}^* = \mathbf{A}^+\mathbf{y}$ in the rank deficient case.

In summary, as the problem in Eq. (4) has infinitely many solutions, the minimum norm solution has been widely used. However, we will show in subsequent sections that with exploitation of the sparsity characteristics of lossy links, it is possible to devise better solutions than the minimum norm solution.

III. Related Work

Network tomography has recently received significant attention, and can be broadly classified into two categories according to the network characteristics to be inferred and estimated [5]: *(i)* link-level parameters, such as the link delay and the link loss rate, based on end-to-end, path-level measurement [2], [3], [6], [8], [10], [13], [14], [16], and *(ii)* path-level traffic intensity based on link-level measurements [15], [17], [22], [23]. Research efforts in the former category can be further classified, based on the mechanisms with which measurements are made, into multicast probing-based [2], [14], unicast probing-based [3], [6], [8], [10], [13], [16], and passive monitoring [12], [13], [16]. They can also be further classified based on the structure of paths that connect the probe packet senders and receivers, such as trees and/or meshes.

The network tomography problem considered in this paper is to estimate the packet loss rates of internal links based on end-to-end, path-level measurements and does not make any explicit assumptions on the measurement method or the connection structure. In what follows, we summarize several works that are most closely related to ours.

Coates and Nowak [6] proposed a back-to-back packet pair measurement scheme to collect more informative statistics for link loss inference in a single source, multiple-receiver network. The method exploits the conditional success probability of the second packet as the additional constraints in the loss inference problem. Harfoush [10] also used the packet pair probing technique for determining whether a pair of connections from the same source experience shared losses.

Padmanabhan *et al.* [13] presented a server-based inference framework that does not inject active probing packets into the network, but instead exploits existing traffic traces observed at a HTTP server to infer link characteristics. To infer the link loss rates on the paths between the server and its clients, they proposed three methods: random sampling, linear optimization, and Bayesian inference using Gibbs sampling. The latter two methods are reported to give better performance but are computationally more expensive. The linear optimization method minimizes the cost function of $||\mathbf{x}||_1$ subject to $\mathbf{A}\mathbf{x} = \mathbf{y}$. An optimization problem of this type is called the *minimum fuel* problem, and the minimum norm solution obtained under this formulation usually gives a solution with a smaller number of non-zero entries than that obtained in Eq. (5) (based on the L_2 norm) [4]. With the use of the likelihood function of (1), the Bayesian inference method uses Markov Chain Monte Carlo (MCMC) with Gibbs sampling to solve the problem and outperforms the others. However, due to its computation complexity, it is practically impossible to apply the Bayesian inference method with Gibbs sampling to a large-scale tomography problem.

Chen *et al.* [3] focused on how to reduce the measurement overhead in a monitoring system for overlay networks. An algebraic approach has been proposed to select and monitor only k

linearly independent paths in an overlay network with n end hosts, while fully describing all the n^2 paths ($k \approx n \log(n)$ for a reasonably large n). The loss rates on the paths not selected can be inferred by the estimates for the links on the selected and independent paths. While the overlay network is characterized by an over-determined linear system, the linear system obtained by pruning linearly dependent paths becomes under-determined with full rank. The authors then used the QR decomposition to compute the minimum norm solution.

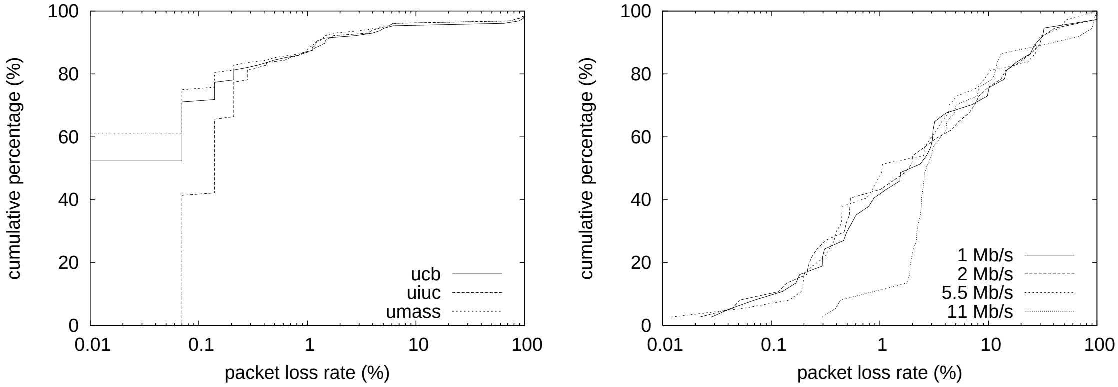
By exploiting the sparsity characteristics of lossy links, Duffield [8] proposed a simple rule-based algorithm to infer the link that performs worst. The proposed *smallest consistent failure set* (SCFS) rule designates as lossy only those links that are nearest the root and consistent with the observed pattern of lossy paths. SCFS incurs much less computational complexity, and yet renders performance comparable to that of the linear optimization method given in [13].

Zhang *et al.* [23] considered the network tomography problem that estimates and infers the traffic matrix (i.e., the volume of traffic between several origin and destination pairs) from link measurements. They introduced the notion of *regularization of ill-posed problems* (i.e., solving under-determined linear equations such as that defined in Eq. (4)), and proposed to exploit statistical independence of network traffic between origin and destination pairs and to minimize mutual information between them. They proposed an estimation method, called *MMI*, and showed that as compared to the minimum norm solution, MMI renders more accurate and robust estimates.

In summary, while the minimum norm solution has been usually used (such as in [3], [13]) as a solution in the absence of other additional information, it has been shown in [8], [23] in a different problem setting that with additional information as side constraints, the resulting solution could be made more accurate and robust. In the next section, we will leverage the fact that lossy links are usually sparse in real networks, and devise a solution algorithm to the under-determined linear system.

IV. Estimation of Link Loss Rates Using the Sparsity Characteristics of Lossy Links

In this section, we first investigate whether or not the assumption that lossy links are sparse is reasonable. Then we exploit the sparsity characteristics of lossy links (i.e., there exist only a small number of x_l's whose values differ significantly from zero in real networks) to reduce the



(a) Path loss rate - Internet (NLANR)

(b) Link loss rate - Wireless mesh network (Roofnet)

Fig. 1. Cumulative percentages of packet loss rates in Internet and wireless mesh network.

number of unknown variables in $\mathbf{x}$, and devise (under the sparsity assumption) an algorithm that computes $\mathbf{x}$ in Eq. (4). By identifying links whose loss rates are likely to be zero and forcing the corresponding estimates to be zero, we obtain a new system of linear equations with matrix $\mathbf{A}_r$, where $\mathbf{A}_r$ is reduced from the m-by-n matrix $\mathbf{A}$.

A. Validation of the Sparsity Characteristics of Lossy Links

The first issue we address is whether or not lossy links are indeed sparse. Many empirical studies have showed that packet loss rates of end-to-end paths are usually close to zero even though their distribution has a wide range. For example, Zhang *et al.* [21] obtained packet loss traces between 31 hosts by using the NIMI measurement infrastructure in 2000. An analysis on these traces showed that 11-15 % of the traces incurred no loss, 47-52% incurred loss rates of 0.1%, 21-24% incurred loss rates of 0.1-1.0%, 12-15% incurred loss rates of 1.0-10%, and only 0.5-1% incurred loss rates exceeding 10%. Chen *et al.* [3] measured packet loss rates between 51 hosts in the PlanetLab testbed in 2003 and reported that 95.9% of paths had loss rates of 0-5%.

We obtain the data sets of loss rates (measured on July 29, 2004) from the Active Measurement Project (AMP) at National Laboratory for Applied Network Research (NLANR). Fig. 1 (a) depicts loss rates measured at three monitors located at Univ. of California at Berkeley, University of Illinois at Urbana Champaign, and Univ. of Massachusetts, to the other 128 AMP monitors. In line with the reports given in [3], [21], we observe that 88% of the 384 paths incur small loss rates (less than 1%).

As the loss rates of links on a path are not larger than that of the path (i.e., for $k \in L(s)$, $p_k - p_s = -(1-p_k)(1-\prod_{l \in L(s), l \neq k}(1-p_l) \leq 0$), a non-lossy path contains no lossy link. As a result, based on the reports given in [3], [21] and our own analysis (of data traces available at NLANR), we conclude that a majority of links have nearly zero loss rates and lossy links are quite sparse in real wired networks.

We also obtain data traces measured on a 38-node urban 802.11b mesh network *Roofnet* [1]. With the packet loss traces measured at different transmission rates of 1, 2, 5.5, and 11 Mb/s, we construct network topologies spanning the whole wireless nodes with the use of the traces of received signal strengths. Fig. 1 (b) depicts the loss rates incurred on the links of the network topologies, which are constructed at different transmission rates. We observe that, although the link loss rates are much larger than those incurred in wired networks (Fig. 1 (a)), still 50 % of the logical links incur loss rates of less than 1 %, and 80 % of them incur loss rates of less than 10 %. That is, even in a typical wireless mesh network, the link loses are sparse *enough.* We will exploit this characteristics in estimating link loss rates.

B. Overview of Our Solution Approach

Based on the sparsity characteristic of lossy links, we propose to compute $\mathbf{x}$ by solving the following optimization problem:

$$\begin{aligned} &\text{minimize} \quad \mathbb{1}_n^T \,\text{sign}(-\mathbf{x}) \\ &\text{subject to } \mathbf{A}\mathbf{x} = \mathbf{y}, \quad x_l \leq 0 \quad \text{for} \quad l \in L. \end{aligned} \tag{8}$$

The cost function in Eq. (8) gives the number of nonzero entries of $\mathbf{x}$. While the minimum norm solution of Eq. (5) has the tendency to spread the loss rates among a number of links, the sparse solution obtained in Eq. (8) will only assign non-zero terms to a small number of links, subject to the constraint.

Consider the scenario in which there exists one lossy link along a path. The minimum norm solution would assign non-zero and (relatively) small loss rates to *several* non-lossy links on the lossy path. This leads to an increase in the number of incorrectly inferred lossy links (termed as *false positives*). On the other hand, if there exist two or more lossy links on a path, the solution given in Eq. (9) may assign a high loss rate to only one of them. This leads to a decrease in the number of correctly inferred lossy links (termed as *coverage*). With the sparsity characteristic of lossy links, we expect that such cases occur rarely.

The optimization problem given in Eq. (8) is closely related to the best basis selection problem, in which a proper subset of vectors is chosen from the over-complete representation of a signal [4]. In other words, the best basis selection problem is to select a few columns $\mathbf{a}_i$'s of the matrix $\mathbf{A}$ that best represent the measurement vector $\mathbf{y}$ in Eq. (4). Finding a smallest basis set of vectors is NP hard, and requires combinatorial search [4]. To this end, we propose a suboptimal method to compute a sparse solution for the optimization problem in Eq. (8). *The key operation is to identify non-lossy links and to reduce in a step-by-step manner the dimension of the system of linear equations, by eliminating identified non-lossy links until* $\|\mathbf{Ax}-\mathbf{y}\|$ *reaches its minimum value.* The process stops when reducing the dimension of the set of linear equations does not further minimize $\|\mathbf{Ax}-\mathbf{y}\|$.

Fig. 2 gives the proposed algorithm. It starts after pruning $(n-m)$ non-lossy links because the rank of $\mathbf{A}$ is less than or equal to m. Then in each iteration, Δi more non-lossy links are inferred. The procedure of assigning weights and identifying non-lossy links in steps 1–2 is carried out by using the three criteria to be given in Section IV-C. Once a link is identified as a non-lossy link, its loss rate is set to zero, regardless of its computed loss rate. The loss rates of the reduced set of linear equations are then computed by the truncated SVD technique given in Section IV-D.

There are several possible refinements that can be made to further improve the efficiency of the algorithm given in Fig. 2: *(i)* once a link is identified to be non-lossy, it is excluded in subsequent iterations; and *(ii)* the rankings of links are not recomputed in every iteration; only those of remaining links are required to be updated.

C. Classifying Links into Lossy and Non-lossy

We consider three methods in selecting non-lossy links (i.e., the links whose loss rates are likely to be zero). Based on the outcomes of these methods, we than assign rankings to links, with lower rankings assigned to non-lossy links. The rankings are integer values in the range of 1 and n. Given the rankings of a link computed in the three methods, w_1, w_2, and w_3, we then assign $w = \max(w_1, w_2, w_3)$ as the final ranking for the link. The values of w's are used to determine whether or not the corresponding links are non-lossy. If a link is identified as a lossy

// The number of inferred non-lossy links is increased by Δi in each iteration

$i \leftarrow (n - m)$ // i is the number of non-lossy links

min_result $\leftarrow \infty$

WHILE $\quad i < n$

1. Compute the rankings of links according to the criteria to be discussed in Section IV-C.
2. Identify non-lossy links whose rankings are in the range of $[1, \cdots, i]$ and assign zero loss rates to them.
3. Obtain the reduced m-by-($n - i$) matrix $\mathbf{A_r}$ by taking columns corresponding to the remaining links.
4. Compute the solution for the reduced linear system using the truncated SVD technique given in Section IV-D.
5. Assign the solution obtained in step 4 to the remaining links.
6. Compute $\|\mathbf{Ax} - \mathbf{y}\|$. If $\|\mathbf{Ax} - \mathbf{y}\| <$ *min_result*, *result* $\leftarrow \mathbf{x}$.
7. $i \leftarrow i + \Delta i$.

END

result contains the link loss rates $\mathbf{x}$ that gives

the minimum $\|\mathbf{Ax} - \mathbf{y}\|$.

Fig. 2. Iterative algorithm that computes the sparse solution for the optimization problem in Eq. (8).

link based on a criterion, it is identified as a lossy link regardless of the decisions based on the other criteria because the final ranking is determined by the maximum value of the rankings.

Criterion I — Selecting Bases*:* We use the solution to the best basis selection problem [18], [24], [25], which aims to find a sparse solution with less than n nonzero entries. Note that $\mathbf{y}$ can be expressed by using the column vectors of $\mathbf{A}$:

$$\mathbf{y} = \sum_{l=1}^{n} x_l \mathbf{a}_l = x_1 \mathbf{a}_1 + \cdots + x_n \mathbf{a}_n,$$

where $\mathbf{a}_l$ is the l^{th} column vector of $\mathbf{A}$. By solving the best basis selection problem, we can select the non-zero column vectors of $\mathbf{A}$ to best represent the measured vector $\mathbf{y}$ in Eq. (4). The selected column vectors correspond to lossy links with non-zero loss rate,

Among a number of algorithms for best basis selection, we use the simplest method based on the projection technique [18], [24], [25]. Suppose that only the k^{th} link among n links has a non-zero loss rate, i.e., $x_k \neq 0$ and $x_l = 0$ for $l \neq k$. Then, $\mathbf{y}$ is collinear with the k^{th} column of $\mathbf{A}$, and we have $x_k = \mathbf{a}_k^T \mathbf{y} / \mathbf{a}_k^T \mathbf{a}_k$. Even though x_l's are not exactly zero for $l \neq k$, x_k will have a larger value than the others with a high probability. Therefore, in the case that only one lossy link exists, we can identify a lossy link as the one that maximizes the normalized projection on each column of $\mathbf{A}$, i.e., the k^{th} link is identified as a lossy link, if

$$k = \arg\max_i \frac{|\mathbf{a}_i^T \mathbf{y}|}{\mathbf{a}_i^T \mathbf{a}_i} \text{ for } i = 1, \cdots, n. \tag{9}$$

When p lossy links are to be identified (Fig. 2), we retain p links whose corresponding column vectors of $\mathbf{A}$ render the first p largest normalized projections of $\mathbf{y}$, and declare the other links as non-lossy. The reduced matrix $\mathbf{A}_r$ is thus composed of the p columns of $\mathbf{A}$ associated with the selected p links.

Criterion II — Sorting Path Loss Rates*:* If a path contains a lossy link, then its path loss rate is likely to be larger than that of paths without any lossy link. On the other hand, if there is no lossy link on a path, the loss rate of the path would be zero. In such a case, we can identify the non-lossys links on the path and prune them in the next iteration. Thus, we determine the ranking of a link based on the loss rate of the path(s) that the link belongs to.

First, the rankings of paths are computed according ot the path loss rates. We assign a (low) ranking of ”1” to the links on a path with the minimum path loss rate. Then, the ranking of a link is determined by that of tha path that the link belongs to. As a link may belong to multiple paths, the ranking of a link is computed by taking the minimum value of rankings obtained at multiple paths. Note that at least one path including a specific link has almost zero loss rate of path, it means the link is not a lossy link. Specifically, let the set of linear equations of Eq. (4) be sorted in an ascending order with respect to $\mathbf{y}$, (i.e., $y_s < y_{s+1} \leq 0$ for $1 \leq s < m$). We compute $\alpha(l) \triangleq \min S(l)$ for $l \in L$, where $S(l)$ is the set of the indices of the paths which link l traverses. For example, if a link l belongs to the ordered, multiple paths of 1, 3, and 5, $S(l) = \{1, 3, 5\}$ and $\alpha(l) = 1$. The rankings of links w_2's can then be determined according to $\alpha(l)$'s. Note that once the values of $\alpha(l)$ are computed, they need not be computed in subsequent iterations. Instead, only the rankings (for the lossy links that remain in each iteration) have to be updated.

In spite of its low computational overhead, this ranking mechanism does not reflect whether or not a link is lossy under several cases. Even though a link exists on a path with the minimum path loss rate, it may be lossy. Consider, for example, a tree-like network topology with a lossy link connected to its root node (i.e., measurement server). As every path includes the link, the link has the ranking of ”1” and could be identified to be non-lossy. For this reason, the rankings w_2 will be used in conjunction with those obtained in the other methods. As we use the $\max$ function to calculate the final composite rankings, the above false positive case can be readily avoided.

Criterion III — Using the Minimum Norm Least Square Solution*:* Recall that in each iteration in Fig. 2, we compute the solution for the reduced linear system (Step 4). We may use the solution obtained in Step 4 to determine the link rankings w_3 to be used in the next iteration. The links with small loss rates are assigned low rankings.

Summary*:* Use of different combinations of the criteria helps to identify the combinations that are robust and less susceptible to measurement errors and numerical computation errors. Through our extensive studies, we find that the combination of Criteria II and III gives the best results, and will henceforth use it in the simulation study. However, as the computation of SVD is quite expensive (N^3), we use the combination of Criteria I and II to obtain the reduced m-by-m matrix $\mathbf{A}_r$ in the first iteration of Fig. 2 without computing SVD of the original and large matrix m-by-n matrix $\mathbf{A}$.

D. Computing Link Loss Rates

As the reduced matrix $\mathbf{A}_r$ is usually rank deficient, we estimate the loss rates of lossy links by computing the minimum norm least square solution via the SVD technique. In this subsection, we elaborate on the SVD technique used to compute the pseudo-inverse of $\mathbf{A}$ in Section II-B. One issue that has to be carefully addressed is that the measurement of path loss rates is highly susceptible to noises resulting from various factors (such as the burstiness in packet losses), and the noises thus introduced in the measurement may seriously impair the accuracy of the calculation results. This is due to the fact that the terms in Eq. (7) contains reciprocals of small, near-zero singular values.

To reduce such undesirable effects, we use the *truncated pseudo-inverse* method described in [9], in which small singular values are simply discarded by truncating the terms at an earlier index $\gamma < w$. That is, instead of using $\mathbf{\Sigma}$ in Eq. (7), we use

$$\mathbf{\Sigma}_\gamma = \mathrm{diag}(\sigma_1, \cdots, \sigma_\gamma),$$

and the truncated pseudo-inverse of $\mathbf{A}$ can be written as

$$\mathbf{A}_\gamma^+ = \sum_{i=1}^{\gamma} \frac{1}{\sigma_i} \mathbf{v}_i \mathbf{u}_i^T. \tag{10}$$

While truncating higher terms in Eq. (7) reduces the adverse effect of measurement noises, it may also result in loss of accuracy. To determine an adequate truncating index γ, we use the following criterion: the percentage accounted for by the first k singular values is defined as

$$\tau_k = \frac{\sum_{i=1}^{k} \sigma_i}{\sum_{i=1}^{W} \sigma_i}.$$

One may pre-determine a cut-off value, τ^* of cumulative percentage of singular values, and calculate γ to be the smallest integer such that $\tau_\gamma \geq \tau^*$. We usually set τ^* to 0.98.

V. Simulation Results

To evaluate the performance of the proposed algorithm in estimating packet loss rates, we have conducted both a simulation study and an empirical study based on the MIT *Roofnet* traces. We report the simulation results in this section, and will defer the discussion on the empirical study in Section VI. We compare the following methods with respect to their capability of inferring packet loss rates:

(i) Minimum norm least square solution (MNLS): as the routing matrix $\mathbf{A}$ is in general a flat matrix (i.e., $m < n$) with rank deficiency, we compute the minimum norm least square solution with the use of SVD in Eq. (7).

(ii) Linear optimization (LINOPT): The linear optimization method proposed in [13] is used. (Recall that it minimizes the cost function $\|\mathbf{x}\|_1$ subject to $\mathbf{Ax} = \mathbf{y}$.) We obtain its solution by using the 'fminsearch' function in Matlab. The maximum number of iterations is restricted to half of the default value[1], due to the prohibitively long time incurred in computation. We set the starting value as the minimum norm least square solution obtained

[1]The default value in Matlab is $(20 \times n)$.

in (i). If the starting value is randomly selected, it usually leads to inaccurate estimates given the specified number of iterations.

(iii) Sparse solutions obtained by the proposed algorithm (SS): We compute two sparse solutions using the algorithm in Fig. 2, based on the combination of the Criteria II and III used to assign link rankings (Section IV-C). In Fig. 2, the number of iterations is equal to $(n - m)/\Delta i$, and its default value is set to twenty. We can control the granularity of solutions by changing Δi.

We evaluate the above algorithms with respect to the *coverage*, the *false positive rate*, and the *similarity* under different network topologies. The coverage is defined as the ratio of the number of links correctly identified to be lossy to that of real lossy links, and the false positive rate is defined as the ratio of the number of links incorrectly identified to be lossy to that of the links identified to be lossy. To compute the coverage and the false positive rate, we select c links with the largest estimated loss rates. Here, c is set to be the number of actual lossy links in the simulation setting. As we are interested in identifying the most lossy links, it is appropriate to select the c links with high loss rates and count the numbers of correctly and falsely identified links for comparing performances between under MNLS, LINOPT, and SS. If an inference algorithm tends to give high loss rates to a number of links, it results in both high coverage and a high false positive rate. On the other extreme, it gives in low coverage and a low false positive rate. A good inference algorithm should achieve both high coverage and low false positive rates.

On the other hand, the similarity measure between the actual loss rate $\mathbf{x}$ and its estimate $\hat{\mathbf{x}}$ obtained by a given algorithm is defined by the *cosine distance* as follows:

$$cos(\mathbf{x}, \hat{\mathbf{x}}) = \frac{\mathbf{x} \cdot \hat{\mathbf{x}}}{\sqrt{(\mathbf{x} \cdot \mathbf{x})(\hat{\mathbf{x}} \cdot \hat{\mathbf{x}})}}.$$

The cosine distance is considered as the angle between two vectors $\mathbf{x}$ and $\hat{\mathbf{x}}$. If the vectors are identical, the cosine distance becomes 1. One may use other measures such as the Euclidean distance to evaluate the similarity performances. However, as the actual/estimated rates contain a number of 0's and only a few non-zero values, a simple sum of estimation errors cannot effectively reflect the estimation performance.

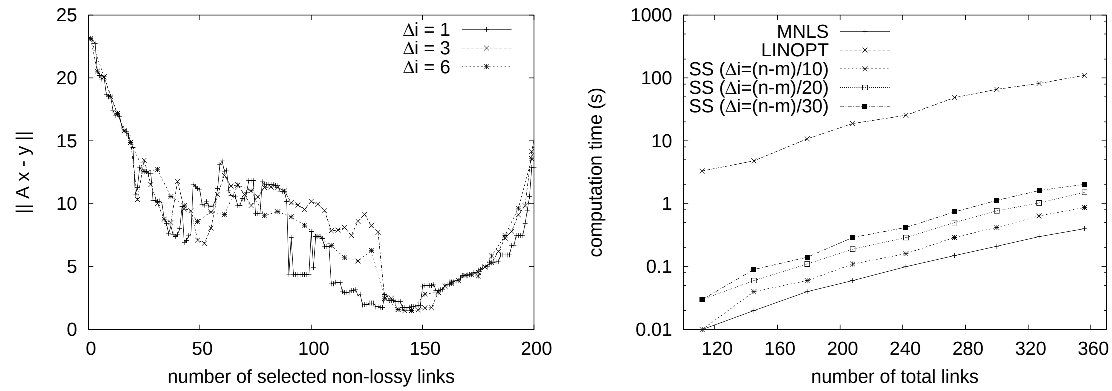
The topologies considered in the simulation study are a transit-stub topology and a random topology obtained by the GT-ITM topology generator [20] and the *ns*-2 simulator [19]. As

TABLE I

SIMULATION ENVIRONMENTS FOR WIRED NETWORKS

ID	network			measurement	
	topology	nodes	links	paths	links
0	transit-stub	100	374	50	77
1	transit-stub	1,010	8,068	100	208
2	transit-stub	1,010	8,068	200	356
3	transit-stub	5,025	83,106	200	407
4	transit-stub	5,025	83,106	400	726
5	random	100	1,058	50	65
6	random	1,000	10,008	100	221
7	random	1,000	10,008	200	365
8	random	5,000	50,184	200	539
9	random	5,000	50,184	400	916

listed in Table I, each topology has 100 – 5,000 nodes including both internal nodes (e.g., routers and switches) and end hosts. In each topology, several routing tables are constructed with measurement paths of different sizes. The transit-stub topology and the random topology have 374 – 83,106 and 1,058 – 50,184 unidirectional links, respectively. A fraction f of the total links are non-lossy, where the value of f varies in the range of 0.5 – 0.9. The loss rates for both lossy and non-lossy links are assigned according to the following loss rate distribution used in [3], [13]: *(i)* LRD_1: if a link is classified as a non-lossy link in the simulation, its loss rate is randomly selected from the range of 0 – 0.01, and if a link is classified as lossy, its lossy rate is randomly selected from the range of 0.05 – 0.1, *(ii)* LRD_2: the loss rates for non-lossy and lossy links are selected from the ranges of 0 – 0.01 and 0.01 – 1, respectively. After assigning the loss rates to links, we obtain the path loss rates by a Gilbert model, in which packets traveling along a path are independently dropped at each link according to its assigned loss rate. For each configuration, we repeat the simulation runs twenty times and report the average of similarity, coverage, and false positive rate.



(a) Search for the minimum $\|\mathbf{Ax}-\mathbf{y}\|$ (b) Computational costs

Fig. 3. Computation of the sparse solution (Topology 1 and LRD_1).

A. Computing the Sparse Solution

We demonstrate how the proposed algorithm computes the sparse solution. Recall that in Fig. 2 the number of selected non-lossy links is increased from $(n-m)$ to n with an increment step of Δi in every iteration. If a link is classified to be non-lossy, its loss rate is set to zero, and the corresponding column is removed from the matrix. After the iterations, we choose the solution that gives the minimum value of $\|\mathbf{Ax}-\mathbf{y}\|$ as a sparse solution.

Fig. 3 (a) shows the changes of $\|\mathbf{Ax}-\mathbf{y}\|$ under Topology 1 as the iterations proceed. (The details on the topology are given in Table I). To help understand the process of computing the sparse solution, the iteration starts at $i=0$ instead of 108 (= $n-m$). In the course of identifying and pruning non-lossy links, the value of $\|\mathbf{Ax}-\mathbf{y}\|$ gradually decreases to a point (i.e., $i^* \approx 140$) and increases again. This implies that selecting more non-lossy links does not improve to reduce the error of $\|\mathbf{Ax}-\mathbf{y}\|$ for $i > i^*$, and we take the solution at i^* as a sparse solution. Moreover, i^* does not change significantly under different incremental values of Δi.

Next, we investigate the computational complexity of the proposed algorithm. Although the minimum norm least square solution is repeatedly computed in Fig 2 (Step 4), the complexity is *not* the number of iterations times higher than that for the minimum norm least square solution. This is because the dimension of the set of linear equations decreases as the iterations proceed. To compare the computational overheads incurred by the various algorithms considered in this simulation study, we have implemented them with Matlab functions, and made the measurement using the ’cputime’ function on an IBM Thinkpad T30 (with a single 1.8 GHz Pentium IV processor and 512 MBytes main memory) that runs Microsoft Windows XP. Fig. 3 (b) shows the average CPU times incurred in computing the solution using MNLS, LINOPT, and SS (with rankings calculated by criteria II and III). Note that SS with rankings calculated by criteria I and II incurs less computational overhead, because it does not compute SVD of the original matrix $\mathbf{A}$.

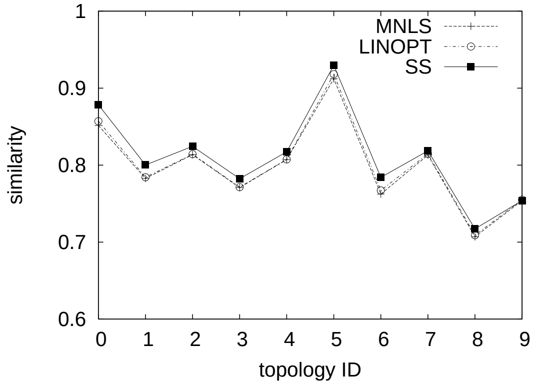
We observe that the computation time of SS with $\Delta = (n-m)/10$ is only twice as much as that of MNLS, even though SS requires SVD be computed ten times. As Δi decreases, the computation time of SS also linearly increases. As compared with LINOPT, SS with $\Delta = (n-m)/10$ incurs approximately two orders of magnitude smaller computation overhead than LINOPT. As reported in [13], although the Markov Chain Monte Carlo (MCMC) with Gibbs sampling outperforms LINOPT, it is also computationally much more expensive than LINOPT. Hence, we do not consider MCMC with Gibbs sampling technique in the simulation study.

B. Comparison w.r.t. Similarity

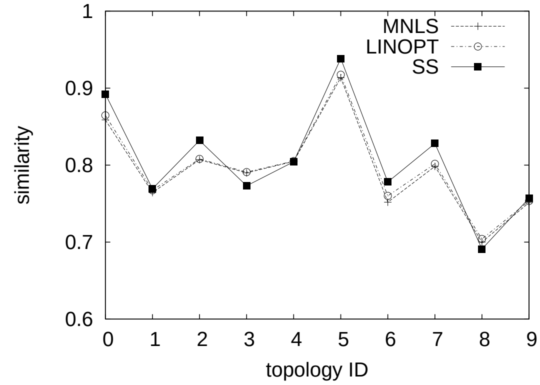
Fig. 4 gives the cosine distances for two possible fractions of non-lossy links f = 0.9 and 0.7 under two loss rate distributions LRD_1 and LRD_2 in 10 topologies (Topologies 0-9). When f = 0.9, only 10 % of links are lossy, and the assumption of sparse lossy links holds. On the other hand, when f = 0.7, the assumption does not really hold, and we can study whether or not, and to what extent, the proposed algorithm renders reasonable results.

Several observations are in order. First, SS gives a larger cosine distance than both MNLS and LINOPT under most cases, and achieves the best performance. Note that the obtained cosine distance varies from 6.54 (Topology 8 in Fig. 4 (c)) and to 9.32 (Topology 5 in Fig. 4 (b)) and highly depends on the network topology used in simulation. Under LRD_2 (Fig. 4 (c) and (d)) where the link loss rates are distributed in a larger range, the lossy links can be more clearly separated and SS gives better improvement than it does under LRD_1 (Fig. 4 (a) and (b)). Second, MNLS and LINOPT give almost the same coverage and false positive rates. Recall that MNLS minimizes the 2-norm of $\mathbf{x}$ while LINOPT does the 1-norm. Based on this observation, we conclude that it is not necessary to solve the linear optimization problem, as it is computationally more expensive and yields approximately the same results.

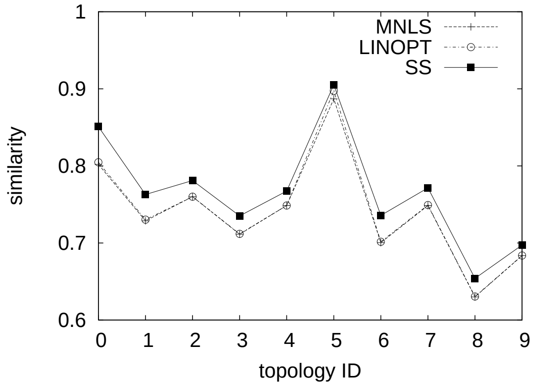
Fig. 5 (a) gives actual and estimated loss rates under the loss rate distributions LRD_1 in



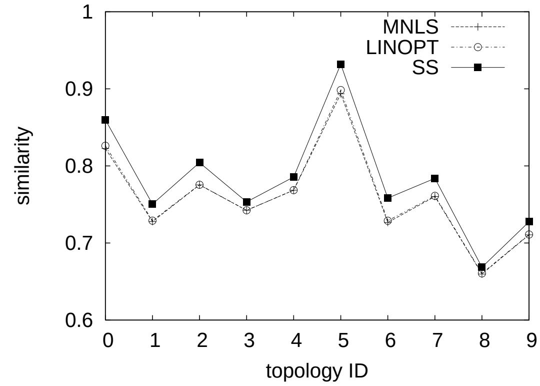
(a) f = 0.9 and LRD_1



(b) f = 0.7 and LRD_1



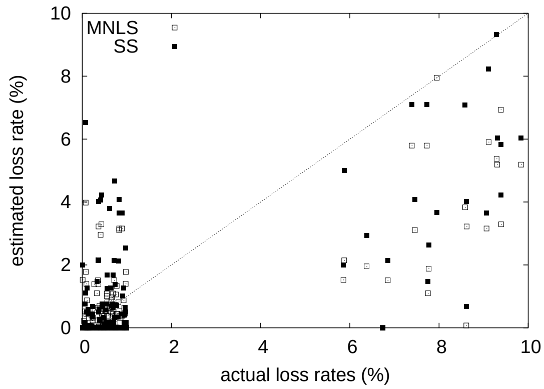
(c) f = 0.9 and LRD_2



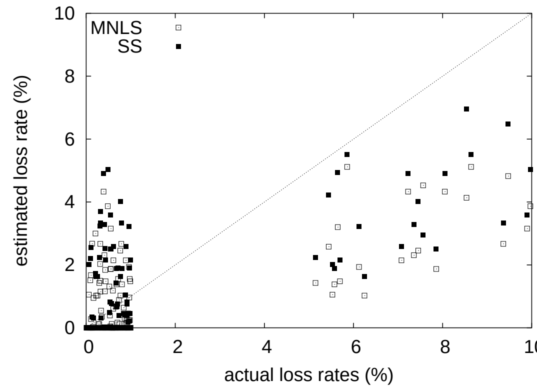
(d) f = 0.7 and LRD_2

Fig. 4. Similarity measure for MNLS, LINOPT, and SS in a variety of topologies.

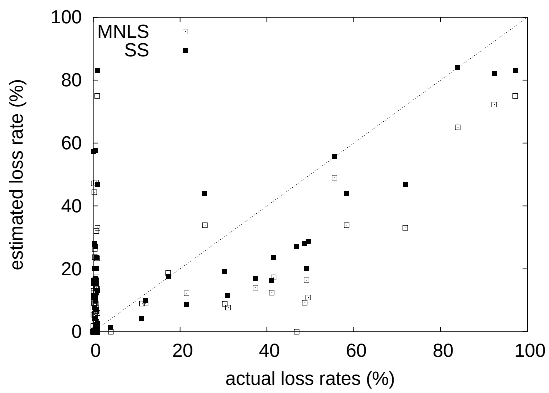
Topology 1. Recall that lossy links have loss rates in the range of 0.05 – 0.1, and non-lossy links have loss rates in the range of 0 – 0.01. If the estimation is perfect, the loss rates lie on the straight line from the origin to (10,10). We observe that most of the estimates are below the line, which implies the loss rates are under-estimated under both methods. However, as compared to MNLS, SS gives the larger estimate that is close to the ideal line in most cases. The reason is that *the sparse solution selects a small number of links and assigns comparatively larger loss rates on them, while the minimum norm least square solution has the tendency to spread small loss rates to a number of links*. In general, SS gives more accurate estimates under the different topologies as shown in Fig. 5 (b) – (d).



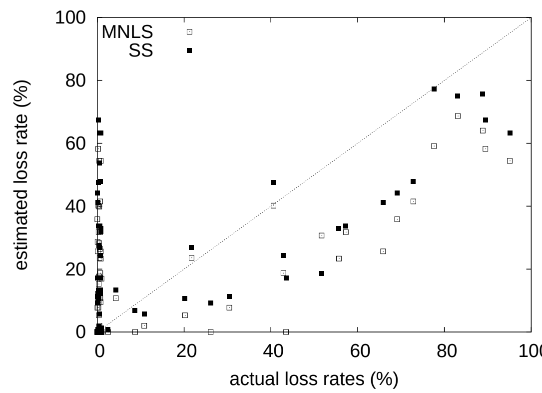
(a) Topology 1 and LRD_1



(b) Topology 6 and LRD_1



(c) Topology 1 and LRD_2



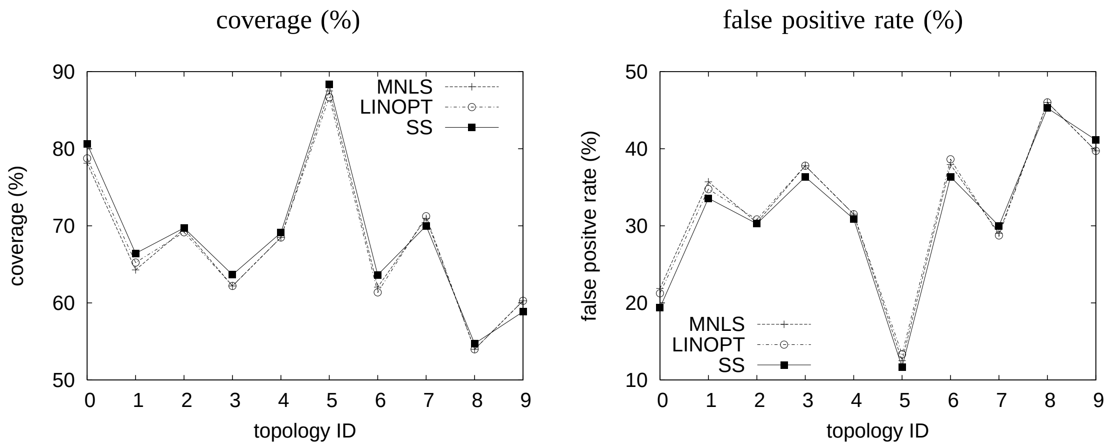
(d) Topology 6 and LRD_2

Fig. 5. Real and estimated loss rates under the minimum norm least square solution (MNLS) and the sparse solution (SS) in the case of $f = 0.9$.

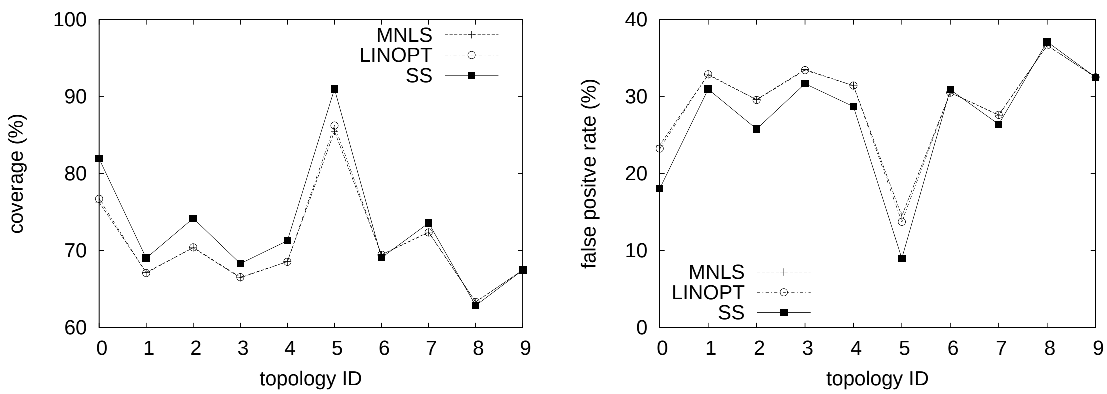
C. Comparison w.r.t. Coverage and False Positive Rate

Fig. 6 and 7 show the performances of loss rate estimation in terms of the coverage and the false positive rate under the loss rate distributions LRD_1 and LRD_2, respectively. Overall, the performance trends with respect to the coverage and the false positive rate shown in Figs. 6 and 7 match that of the cosine distance in Fig. 4 because the performance metrics highly depend on network topologies. For instance, the algorithms achieve better performance for comparatively small networks, e.g., Topologies 0 and 5 in Table I.

Among all the algorithms, SS achieves the highest coverage while keeping the lowest false positive rate for most cases. This result implies that it is possible to identify more accurately lossy links by exploiting the sparse distribution of lossy links. On the other hand, because the



(a) fraction of non-lossy links: $f = 0.9$



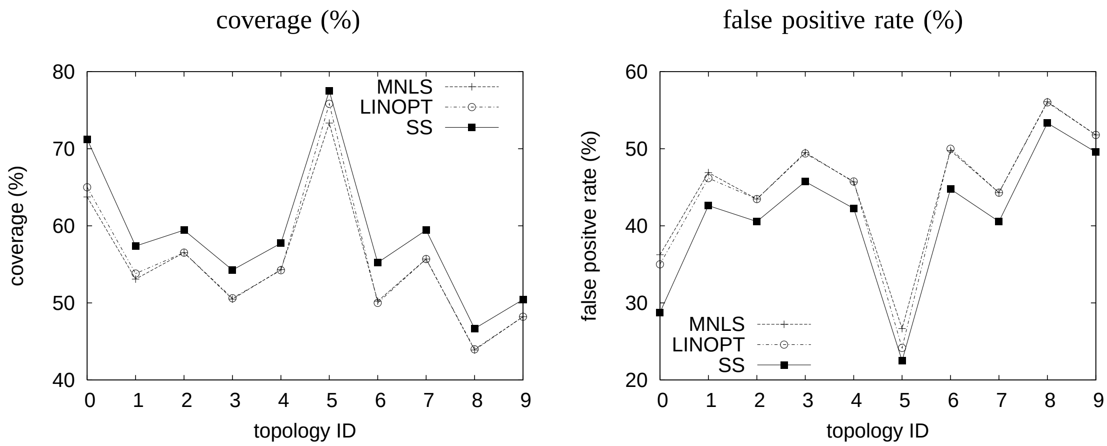
(b) fraction of non-lossy links: $f = 0.7$

Fig. 6. Performance (in terms of the coverage and the false positive rate) of MNLS, LINOPT, and SS for loss rate distribution: LRD_1 in a variety of topologies.

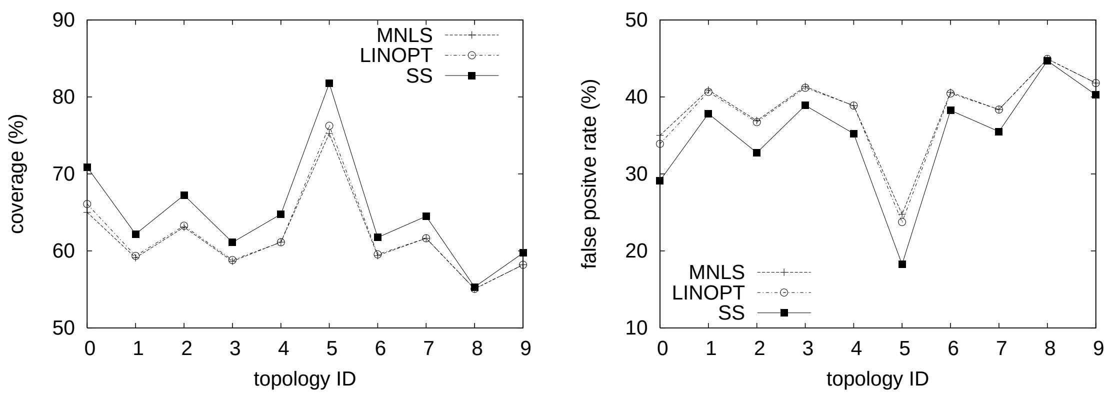
performance dramatically changes according to the simulation setting, we carry out an empirical study in Section VI to further investigate the performance based on real-life traces.

VI. Experiments Based on the *Roofnet* Data Traces

To evaluate the proposed algorithm in real environments, we leverage the empirical traces available in the MIT Roofnet project [1]. As discussed in Section IV-A, we construct the logical links (and the corresponding topology) for each transmission rate with the use of received signal strengths. The data traces provide us with the link loss rates, and we compute the loss rate of a path using Eq. (2). With the “derived” topology and the path loss rates, we compute the estimated link loss rate using MNLS, LINOPT, and SS, and compare them with the actual link



(a) fraction of non-lossy links: $f = 0.9$



(b) fraction of non-lossy links: $f = 0.7$

Fig. 7. Performance (in terms of the coverage and the false positive rate) of MNLS, LINOPT, and SS for loss rate distribution: LRD_2 in a variety of topologies.

loss rates. Note that with the path loss rates measured between end-nodes on the Internet it is not possible to directly validate/invalidate algorithms, because the link loss rates at intermediate nodes are not available.

Table II gives the cosine distances under MNLS, LINOPT, and SS, at the transmission rate of 1, 2, 5.5, and 11 Mb/s. Although the link loss rates in wireless network are believed to be much higher than those in wired networks (i.e., lossy links are *not* sparse), SS still achieves the best performance with respect to the similarity under all the cases. As a matter of fact, the performance discrepancy is much more salient between SS and MNLS/LINOPT.

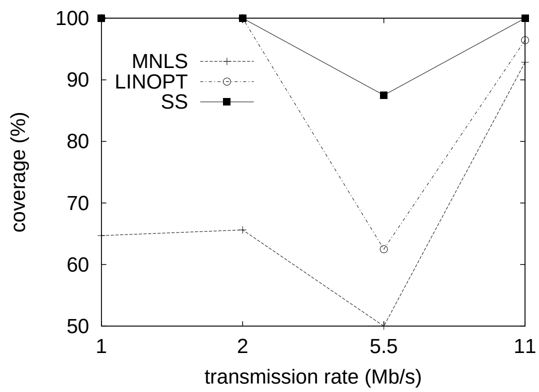
Fig. 8 (a) – (d) show the coverage and the false positive rate when we take 2 or 6 links as the

TABLE II

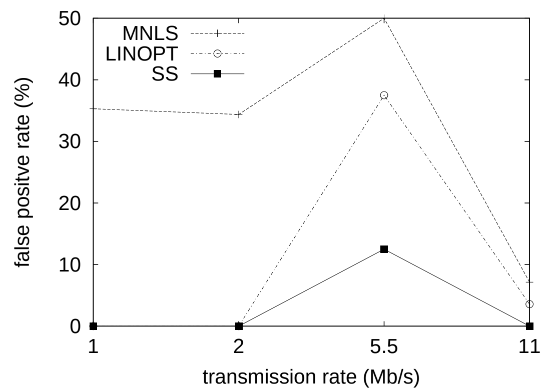
SIMILARITY MEASURE IN THE WIRELESS MESH NETWORK.

bandwidth (Mb/s)	MNLS	LINOPT	SS
1	0.747	0.875	0.959
2	0.718	0.845	0.951
5.5	0.645	0.785	0.909
11	0.741	0.742	0.814

2 lossiest links (c = 2)

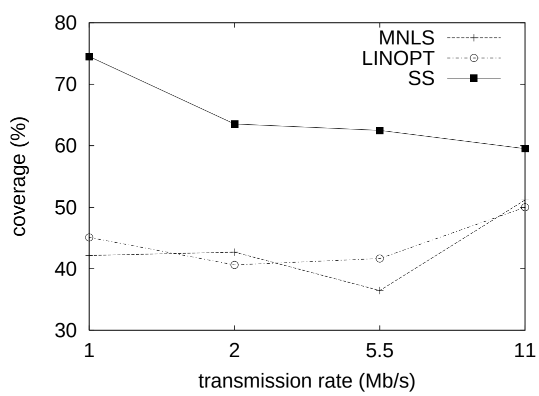


(a) coverage (%)

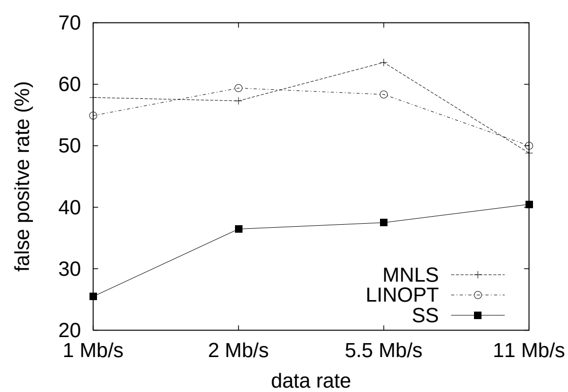


(b) false positive rate (%)

6 lossiest links (c = 6)



(c) coverage (%)



(d) false positive rate (%)

Fig. 8. Performance (in terms of the coverage and the false positive rate) in the wireless mesh network.

TABLE III

MEAN AND MEDIAN OF ABSOLUTE ESTIMATION ERRORS IN THE WIRELESS MESH NETWORK.

bandwidth (Mb/s)	mean of absolute errors (%)			median of absolute errors (%)		
	MNLS	LINOPT	SS	MNLS	LINOPT	SS
1	9.6	7.38	4.58	5.26	4.10	2.50
2	11.03	8.66	5.10	9.10	6.24	1.97
5.5	11.57	9.44	5.21	8.29	8.05	4.53
11	13.22	13.06	11.30	6.79	6.39	5.06

most lossy links depending on the estimated loss rates. SS gives the largest coverage and the smallest false positive rate in all the cases. In particular, as shown in Fig. 8 (a) and (b), if 2 links with the largest link loss estimates under SS are selected at the transmission rates of 1, 2, and 11 Mb/s, they are actually the most lossy links in the wireless network with a probability of 1. In Fig. 8 (c) and (d), the coverage and the false positive rate gets smaller and larger, respectively, as the transmit rate increases. This is due to the fact that the links incur higher loss rates at a high transmission rate and the distribution of the link loss rates is less sparse (Fig. 1 (b)).

In the course of carrying out the experiment, we also find that even though the topology is constructed using the traces of received signal strengths, a few links still incur packet loss rates close to 1. In this case, SS successfully identifies the links with high lossy rates and assign them high loss estimates while MNLS and LINOPT give inaccurate estimates.

Table III shows the statistics of absolute estimation errors. In all the cases, SS gives the smallest mean and median errors. Although the link loss rates in wireless network are believed to be much higher than those in wired networks (i.e., lossy links are *not* sparse), SS still achieves the best estimation performance under most cases.

VII. CONCLUSION

In this paper, we have considered the problem of estimating, based on end-to-end path loss rates, loss rates of internal links. It has been shown that the maximum likelihood estimate (MLE) for link loss rates leads to an under-determined system of linear equations. Although most existing work uses the minimum norm solution as the solution to this under-determined linear system, we show that there is room for further improvement. We exploit the statistical knowledge that lossy links are sparse in real operating networks and formulate a new optimization problem that minimizes the number of lossy links, subject to satisfying the set of linear equations. We then propose an iterative algorithm to solve the optimization problem by identifying non-lossy links and pruning columns that correspond to non-lossy links from the matrix that characterizes the set of linear equations. The process of identifying non-lossy links is performed using three different criteria (and a combination thereof) determined by three different methods: basis selection, sorting of path loss rates, and solving the minimum norm least square problem. We show via simulation and empirical studies on the Roofnet traces that (i) the computational complexity of the iterative algorithm is comparable to that of MNLS (and two orders of magnitude smaller than LINOPT); and (ii) the sparse solution obtained under the iterative algorithm achieves high coverage and incurs a small number of false positives under various network scenarios.

As part of our future work, we will further investigate the issues of i) how to improve the iteration procedure by using the rank information of the reduced matrix, and ii) how to predict the identification performance in advance for a given routing matrix $\mathbf{A}$. We will also carry out empirical experiments on traces obtained on large-scale networks to further validate and evaluate the proposed iterative algorithm.

REFERENCES

[1] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris. Link-level measurements from an 802.11b mesh network. In *Proceedings of ACM SIGCOMM*, 2004.

[2] R. Caceres, N. Duffield, J. Horowitz, and D. Towsley. Multicast-based inference of network-internal loss characteristics. *IEEE Trans. on Information Theory*, 45:2462–2480, 2002.

[3] Y. Chen, D. Bindel, H. Song, and R. Katz. An algebraic approach to practical and scalable overlay network monitoring. In *Proceedings of ACM SIGCOMM*, 2004.

[4] A. Cichocki and S. Amari. *Adaptive blind signal and image processing*, chapter 2.5. John Wiley and Sons, 2002.

[5] M. J. Coates, A. O. Hero, R. Nowak, and B. Yu. Internet tomography. *IEEE Signal Processing Magazine*, pages 47–65, 2002.

[6] M. J. Coates and R. Nowak. Network loss inference using unicast end-to-end measurement. In *Proceedings of ITC Conference on IP Traffic, Modelling and Management*, 2000.

[7] M. J. Coates and R. Nowak. Sequential Monte Carlo inference of internal delays in non-stationary data networks. *IEEE Transactions on Signal Processing*, 50(2):366–376, 2002.

[8] N. Duffield. Simple network performance tomography. In *Proceedings of ACM Internet Measurement Conference*, 2003.

[9] N. G. Gencer and S. J. Williamson. Differential characterization of neural sources with the bimodal truncated SVD pseudo-inverse for EEG and MEG measurements. *IEEE Trans. on Biomedical Engineering*, 45(7), 1998.

[10] K. Harfoush, A. Bestavros, and J. Byers. Robust identification of shared losses using end-to-end unicast probe. In *Proceedings of the 2000 International Conference on Network Protocols*, 2000.

[11] S. Haykin. *Adaptive Filter Theory*. Prentice Hall, 3 edition, 1996.

[12] V. N. Padmanabhan, L. Qiu, and H. J. Wang. Passive network tomography using Bayesian inference. In *Proceedings of ACM SIGCOMM Workshop on Internet measurement*, 2002.

[13] V. N. Padmanabhan, L. Qiu, and H. J. Wang. Server-based inference of Internet link lossiness. In *Proceedings of IEEE INFOCOM*, 2003.

[14] F. Presti, N. G. Duffield, J. Horowitz, and D. Towsley. Multicast-based inference of network-internal delay distribution. Technical report 99-55, Univ. Massachusetts, Amherst, MA, 1999.

[15] C. Tebaldi and M. West. Bayesian inference on network traffic using link count data. *Journal of American Statistical Association*, 93(442):557–576, 1998.

[16] Y. Tsang, M. J. Coates, and R. Nowak. Passive network tomography using EM algorithms. In *Proceedings of IEEE International Conference Acoustic and Signal Processing*, 2001.

[17] Y. Vardi. Network tomography: Estimating source-destination traffic intensities from link data. *Journal of American Statistical Association*, 91(433):365–377, 1996.

[18] L. Vielva, D. Erdogmus, and J. C. Principe. Under-determined blind source separation using a probabilistic source sparsity model. In *Proceedings of International Conference on Independent Component Analysis and Blind Signal Separation*, 2001.

[19] VINT Project. The network simulator - ns-2. http://www.isi.edu/nsnam/ns.

[20] E. W. Zegura, K. Calvert, and S. Bhattacharjee. How to model an Internetwork. In *Proceedings of IEEE INFOCOM*, 1996.

[21] Y. Zhang, N. Duffield, V. Paxson, and S. Shenker. On the constancy of Internet path properties. In *Proceedings of ACM SIGCOMM Workshop on Internet Measurement*, 2001.

[22] Y. Zhang, M Roughan, N. Duffield, and A. Greenberg. Fast accurate computation of large-scale IP traffic matrices from link loads. In *Proceedings of ACM SIGMETRICS*, 2003.

[23] Y. Zhang, M. Roughan, C. Lund, and D. Donoho. An information-theoretic approach to traffic matrix estimation. In *Proceedings of ACM SIGCOMM*, 2003.

[24] J. Adler, B. Rao, and K. Kreutz-Delgado. Comparision of basis selection methods. In *Proceedings of Asilomar Conf. Signal, Sys., Comp.*, 1996.

[25] R. Coifman and M. Wickerhaser. Entropy-based algorithms for best basis selection. *IEEE Trans. Info. Theory*, IT-38(2):713-18, 1992.