Network Tomography on General Topologies *

Tian Bu
Department of Computer Science
University of Massachusetts
Amherst, MA 01003, USA
tbu@cs.umass.edu

Nick Duffield
AT&T Labs-Research
180 Park Avenue
Florham Park, NJ 07932, USA
duffield@research.att.com

Francesco Lo Presti
Dipartimento di Informatica
Università dell'Aquila
Via Vetoio, Coppito (AQ), Italy
lopresti@univaq.it

Don Towsley
Department of Computer Science
University of Massachusetts
Amherst, MA 01003, USA
towsley@cs.umass.edu

ABSTRACT

In this paper we consider the problem of inferring link-level loss rates from end-to-end multicast measurements taken from a collection of trees. We give conditions under which loss rates are identifiable on a specified set of links. Two algorithms are presented to perform the link-level inferences for those links on which losses can be identified. One, the *minimum variance weighted average (MVWA) algorithm* treats the trees separately and then averages the results. The second, based on *expectation-maximization (EM)* merges all of the measurements into one computation. Simulations show that EM is slightly more accurate than MVWA, most likely due to its more efficient use of the measurements. We also describe extensions to the inference of link-level delay, inference from end-to-end unicast measurements, and inference when some measurements are missing.

1. INTRODUCTION

As the Internet grows in size and diversity, its internal behavior becomes ever more difficult to characterize. Any one organization has administrative access to only a small fraction of the network's internal nodes, whereas commercial factors often prevent organizations from sharing internal performance data. Thus it is important to characterize internal performance from end-to-end measurements.

One promising technology that avoids these problems uses end-to-end multicast measurements from a single tree to infer link-level loss rates and delay statistics [1] by exploiting the inherent correlation in performance observed by multicast receivers. A shortcoming of this technology is that it is usually impossible to include all links of interest in any one tree. Consider the network in Figure 1(a) as an example. In this network, end-hosts 0 and 1 are sources, end-hosts 4 and 5 are receivers, and the set of links of interest is $\{(2,5) \quad (3,2)\}$. It is observed that both tree 1 and tree 2 are needed to cover the set of links of interest as illustrated in Figure 1(b) and 1(c). Therefore, in order to characterize the behavior of a network (or even a portion of it), it is necessary to perform measurements on multiple trees. Inferring link-level performance from measurements taken from several trees poses a challenging problem that is the focus of this paper.

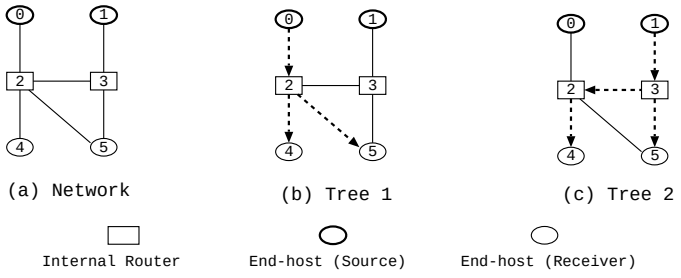


Figure 1: Single tree can not characterize a network

In this paper we address the following two problems. Given a collection of multicast trees, can we infer the performance of all of the links (or a specified subset) that are contained by the trees? Second, when the performance of the links of interest can be identified, how do we obtain accurate estimates of their performance? Focusing on loss rate as the performance metric, we introduce and evaluate two algorithms. The first, the *minimum variance weighted average (MVWA) algorithm*, performs inference on each tree separately and, for each link, returns a weighted average of the estimates taken from the different trees. This procedure may not always be able to infer the behavior of links whose loss rates are, nevertheless, identifiable. The loss rates for these links are obtained as a solution to a set of linear equations involving the inferred loss rates from individual trees. The second algorithm, the *expectation-maximization (EM) algorithm*, on the other hand, applies the standard expectation-maximization technique [15] to the measurement data taken from all of the trees. It returns estimates

*This work was supported in part by DARPA under contract F30602-00-2-0554 and F30602-98-2-0238, and by the National Science Foundation under Grant EIA-0080119.

of the loss rates of all identifiable links. We evaluate the two algorithms through simulation studying their convergence rates and relative performance. We find that EM estimates are at least as accurate than those produced by MVWA. The improvement is more pronounced when either the number or measurements is small or the distribution of measurements among the various trees is skewed.

Although the focus here is on link-level loss rates, we give extensions to EM to handle link delay. In addition, we show how MVWA and EM can be applied when end-to-end multicast measurements are not available, or when some measurements are missing.

There is a related problem of how to choose the set of trees so as to cover all of the links in the network (or subset of interest) in an efficient manner. This question has been dealt with elsewhere, [2] and is not considered here. We take as given the set of trees and observations from which we are to draw inferences.

Network tomography from end-to-end measurements has received considerable attention recently. In the context of multicast probing, the focus has been on loss, delay, and topology identification. Extensions to unicast probing can be found in [6, 7, 8, 11, 13]. However, these have treated only individual trees. There are techniques for round trip metrics such as loss rate and delay [14], based on measurements taken from a single node. Last, linear algebraic methods have been proposed for estimating link-level average round trip delays [19] and one-way delays, [12] . Neither of these extend to other metrics. Furthermore, the latter only yields biased estimates of average delays.

The remainder of the paper is organized as follows. Section 2 presents the model for a "multicast forest" (set of multicast trees). In Section 3 we present necessary and sufficient conditions for when the loss probabilities can be inferred from end-to-end multicast measurements. The MVWA and EM algorithms are presented in Section 4 along with convergence properties of the latter. Section 5 presents the results of simulation experiments. Extensions to delay inference, the use of unicast, and missing data are found in Section 6. Last, Section 8 concludes the paper.

2. NETWORK AND LOSS MODEL

Let $N = (V(N), E(N))$ denote a network with sets of nodes $V(N)$ and links $E(N)$. Here $(i,j) \in E(N)$ denotes a directed link from node i to node j in the network. Let Ψ denote a set of multicast trees embedded in N, i.e., $\forall T \in \Psi$, $V(T) \subseteq V(N)$ and $E(T) \subseteq E(N)$. We denote $\cup_{T\in\Psi} V(T)$ by $V(\Psi)$ and $\cup_{T\in\Psi} E(T)$ by $E(\Psi)$. Note that $(i,j) \in E(\Psi)$ can appear in more than one tree. For $(i,j) \in E(N)$, we denote $\Psi_{i,j} \subseteq \Psi$ the set of trees which include link (i,j). Consider a tree $T \in \Psi$. Each node i in T, apart from the root $\rho(T)$, has a parent in T, $f(i,T)$, such that $(f(i,T),i) \in E(T)$. The set of children of i in tree T is denoted by $d(i,T)$. Let $\tau_{i,T}$ denote the subtree of T rooted at node i. Let $R(\tau_{i,T})$ denote the receivers in subtree $\tau_{i,T}$. We denote the path from node i to j, $i,j \in V(T)$ in tree T by $p_T(i,j)$. Define a segment in T to be a path between either the root and the closest branch point, two neighboring branch points, or a branch point and a leaf. We represent a segment by the set of links that comprises it.

For $T \in \Psi$, we identify the root $\rho(T)$ with the source of probes, and the set of leaves $R(T)$ with the set of receivers. For a tree T, a probe is sent down the tree starting at the root. If it reaches node $j \in V(T)$, a copy of the probe is produced and sent down the tree toward each child of j. As a packet traverses link (i,j), it is lost with probability $1-\alpha_{i,j}$ and arrives at j with probability $\alpha_{i,j}$. We denote $1-\alpha_{i,j}$ by $\overline{\alpha}_{i,j}$. Let $\alpha = (\alpha_{i,j})_{(i,j)\in E(\Psi)}$. We assume losses of the same probe on different links and of different probes on the same link are independent, and that losses of probes sent from the different sources $\rho(T)$, $T \in \Psi$ are independent.

We describe the passage of probes down each tree T by a stochastic process $X_T = (X_{k,T})_{k\in V(T)}$ where $X_{k,T} = 1$ if the probe reaches node k, 0 if does not. By definition $X_{\rho(T),T} = 1$. If $X_{i,T} = 0$ then $X_{j,T} = 0$ for all $j \in d(i,T)$. If $X_{i,T} = 1$ then for $j \in d(i,T)$, $X_{j,T} = 1$ with probability $\alpha_{i,j}$ and $X_{j,T} = 0$ with probability $\overline{\alpha}_{i,j}$. We assume that the collection of trees is in *canonical form*, namely that $0 < \alpha_{i,j} < 1, \forall (i,j) \in E(\Psi)$. An arbitrary collection of trees can be transformed into one with canonical form.

In an experiment, a set of probes is sent from the multicast tree sources $\rho(T)$, $T \in \Psi$. For each $T \in \Psi$, we can think of each probe as a trial, the outcome of which is a record of whether or not the probe was received at each receiver in $R(T)$. In terms of the random process X_T, the outcome is a configuration $X_{R(T)} = (X_{i,T})_{i\in R(T)}$ of zeros and ones at the receivers. Notice that only the values of X_T at the receivers are observable; the values at the internal nodes are unknown. Each outcome is thus an element of the space $\Omega_{R(T)} = \{0,1\}^{\#R(T)}$. For a given set of link probabilities α the distribution of $X_{R(T)}$ on $\Omega_{R(T)}$ will be denoted $\mathsf{P}_{\alpha,T}$. The probability of a single outcome $x \in \Omega_{R(T)}$ is $p(x;\alpha) = \mathsf{P}_{\alpha,T}[X_{R(T)} = x]$.

3. IDENTIFIABILITY

In order to perform tomography from measurements on the tree set Ψ, we require that the link probabilities are determined from the set leaf probabilities that are measured directly. We phrase this in terms *identifiability*, which captures the property that link probabilities can be distinguished by measurements from an infinite sequence of probes. We say that $\{\mathsf{P}_{\alpha,T}\}_{T\in\Psi}$ identifies α if for any α', $\{\mathsf{P}_{\alpha,T}\}_{T\in\Psi} = \{\mathsf{P}_{\alpha',T}\}_{T\in\Psi}$ implies $\alpha = \alpha'$. In this section, we establish necessary and sufficient conditions for identifiability.

We are given a set of canonical trees Ψ with an associated link success probability vector $\alpha = (\alpha_{i,j})_{(i,j)\in E(\Psi)}$. Let S be the set of all segments within the trees contained in Ψ. Define β_s to be the logarithm of the probability that a packet successfully traverses segment $s \in S$ given that it reached the start of that segment, $\beta_s = \log(\prod_{(i,j)\in s} \alpha_{i,j}) = \sum_{(i,j)\in s} \log \alpha_{i,j}$. We introduce the $\#S \times \#E(\Psi)$ matrix A where $A_{s,(i,j)} = 1$ if link (i,j) belongs to segment s and 0 otherwise. Using the sets of trees in Figure 1 as an example, if we order the links as $(0,2)$ $(2,4)$ $(2,5)$ $(1,3)$ $(3,5)$ $(3,2)$ and the segment as $\{(0,2)\}$ $\{(2,4)\}$ $\{(2,5)\}$ $\{(1,3)\}$ $\{(3,5)\}$ $\{(3,2),(2,4)\}$, the matrix A is

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}$$

If we define $z_{(i,j)} = \log \alpha_{i,j}, \forall (i,j) \in E(\Psi)$, we then have the following equation

$$Az = \beta \tag{1}$$

Here the components of z are $z_{(i,j)}$ and the components of β are β_s. Note that A needs not be a square matrix in general.

Before stating and proving results on identifiability, we note that for a given set of link probabilities α, there exists at least one solution, namely $z = \log \alpha$, to (1). Let A^T denote the matrix transpose of A.

THEOREM 1. *Let Ψ be a set of canonical loss trees. Then the following are equivalent:*

(i) For some α, $\{\mathsf{P}_{\alpha,T}\}_{T\in\Psi}$ identifies α.

(ii) Equation (1) has a unique solution $z = (A^T A)^{-1} A^T \beta$.

(iii) $Az = 0$ iff $z = 0$.

(iv) For all α, $\{\mathsf{P}_{\alpha,T}\}_{T\in\Psi}$ identifies α.

Proof. (i)⇔(ii). First, we note that β is identifiable from $\{\mathsf{P}_{\alpha,T}\}_{T\in\Psi}$ (Theorem 3 in [4]). Suppose that $\{\mathsf{P}_{\alpha,T}\}_{T\in\Psi}$ cannot identify α, i.e., there are at least two sets of link probabilities, α and α' that are consistent with $\{\mathsf{P}_{\alpha,T}\}_{T\in\Psi}$. Based on the derivation of (1) there cannot exist an unique solution to (1). Similarly, if α is identifiable, it is obtained by solving (1). Suppose that (1) does not have a unique solution. Then, from the derivation of (1) it follows that there exist multiple values of α that can give rise to $\{\mathsf{P}_{\alpha,T}\}_{T\in\Psi}$. Suppose that there exists a unique solution to (1). It is easy to show by contradiction that necessarily there is only one value of α that can give rise to $\{\mathsf{P}_{\alpha,T}\}_{T\in\Psi}$. For (ii) ⇔(iii), observe that (1) has a unique solution if and only if the nullspace of A is in $\{0\}$. In this case $A^T A$ is invertible, and the expression for z then follows on pre-multiplying (1) by A^T. $(A^T A)^{-1} A^T$ is the generalized inverse of A; see [16]. Furthermore, solutions of (1) must be unique for all α, and hence (ii)⇔(iv). □

It should be clear from this theorem that identifiability is a topological property, i.e., not dependent on the values α. We can use this fact to select β at our convenience. Suppose we are interested in identifying a set of links a set of links $C \subset E(\Psi)$. Choosing $\alpha_{i,j} = e^{-1}, \forall (i,j) \in E(\Psi)$ results in $\beta_s = \#s$. Hence we have:

THEOREM 2. *Let Ψ be a set of canonical loss trees. $\{\mathsf{P}_{\alpha,T}\}_{T\in\Psi}$ identifies $(\alpha_{i,j})_{(i,j)\in C}$ iff there is a unique value of $\{z_{(i,j)} : (i,j) \in C\}$ that satisfies equation (1) for $\beta_s = \#s, \forall s \in S$.*

4. LOSS INFERENCE

In this section, we describe two algorithms for loss inference in a collection of multicast trees. In the first algorithm we perform inference on each tree separately, and then we take the weighted average of the different estimates so obtained. In the second algorithm we perform inference on the entire set of measurement from all of the trees using the Expectation-Maximization (EM) algorithm.

4.1 Measurement Experiment

A measurement experiment for a collection of multicast trees Ψ consists of sending n_T probes from $\rho(T)$, $T \in \Psi$. For each $T \in \Psi$, we denote by $\mathbf{x}_{R(T)} = (x^1_{R(T)}, \ldots, x^{n_T}_{R(T)})$, (with $x^m_{R(T)} = (x^m_{k,T})_{k\in R(T)}$) the set measured of end-to-end loss down T. $\mathbf{x}_R = (\mathbf{x}_{R(T)})_{T\in\Psi}$ will denote the complete set of measurements.

4.2 Minimum Variance Weighted Average

A technique for loss inference for a single tree has been proposed in [4]. For a given set of trees Ψ, we can proceed as follows: (1) consider each tree $T \in \Psi$ separately, by using the algorithm provided in [4] on the measurements $\mathbf{x}_{R(T)}$; this yields estimates for all segments in T; (2) combine the estimates from the different trees.

We first consider the problem of combining estimators of segment transmission probabilities. Let s be a segment, and $\Psi_s \in \Psi$ the maximal set of topologies that include s as segment. Inference on each logical topology $T \in \Psi_s$ provides us with an estimate $\hat{q}_{s,T}$ of the transmission probability $q_s = e^{\beta_s}$ across the segment s. How should the $\hat{q}_{s,T}$ be combined to form a single estimate of q_s?

We consider convex combinations of the form

$$\hat{q}_s = \sum_{T\in\Psi_s} \lambda_T \hat{q}_{s,T}, \quad \lambda_T \in [0,1]; \sum_{T\in\Psi_s)} \lambda_T = 1. \tag{2}$$

We propose to select the minimum variance combination as the single estimator. By assumption, the $\hat{q}_{s,T}$ are independent, and so

$$\mathsf{Var}(\hat{q}_s) = \sum_{T\in\Psi_s} \lambda_T^2 \mathsf{Var}(\hat{q}_{s,T}). \tag{3}$$

$\mathsf{Var}(\hat{q}_{s,T})$ is clearly jointly convex in the $(\lambda_T)_{T\in\Psi_s}$, and by explicit differentiation under the constraint $\sum_{T\in\Psi_s} \lambda_T = 1$, the minimum for $\mathsf{Var}(\hat{q}_s)$ occurs when

$$\lambda_T = \frac{\mathsf{Var}(\hat{q}_{s,T})^{-1}}{\sum_{T'\in\Psi_s} \mathsf{Var}(\hat{q}_{s,T'})^{-1}} \tag{4}$$

Now, in general, $\mathsf{Var}(\hat{q}_{s,T})$ depends on the topology T. But it follows from Theorem 5 in [4] that the asymptotic variance $n_T \mathsf{Var}(\hat{q}_{s,T})$ converges to $\overline{q}_s + O(\|\overline{\alpha}\|^2)$ as $n_T \to \infty$. Thus, for small loss probabilities, we can use the approximation $\mathsf{Var}(\hat{q}_{s,T}) \approx n_T^{-1} \overline{q}_s$. In this approximation, the coefficients $\lambda_T \approx n_T / \sum_{T'\in\Psi(T)} n_{T'}$. We will use this approximation in (2) as our minimum variance weighted average algorithm (MVWA) algorithm, i.e.,

$$\hat{q}_s = \frac{\sum_{T\in\Psi_s} n_T \hat{q}_{s,T}}{\sum_{T\in\Psi_s} n_T} \tag{5}$$

We note two special cases: (i) s comprises a single link (i,j), in which case the estimate is for the link rate $\alpha_{i,j}$; (ii) only one tree contains s, in which case the sums in (5) trivially have one term.

It remains to recover link probabilities from the $\hat{q}_s$. Following Theorem 1, identifiable link probabilities $\alpha_{i,j}$ are estimated by

$$\log \hat{\alpha}_{i,j} = \sum_s A^*_{(i,j),s} \log \hat{q}_s \tag{6}$$

A simple example is when two segments s, s' are such that s is obtained by appending the link (i,j) to s'. Clearly $A^*_{(i,j),s} = 1 - A^*_{(i,j),s'}$ with (6) reducing to taking quotients: $\hat{\alpha}_{i,j} = \hat{q}_s / \hat{q}_{s'}$.

4.3 EM Algorithm

Here we turn to a more direct approach to inference, namely, we use the Maximum Likelihood Estimator to estimate α from the set of measurements $\mathbf{x}_R$, i.e., we estimate α by the value $\widehat{\alpha}$ which maximizes the probability of observing $\mathbf{x}_R$.

Let $n_T(x_{R(T)})$ denote the number of probes for which the outcome $x_{R(T)} \in \Omega_{R(T)}$ is obtained, $T \in \Psi$. The probability of the n_T

independent observations $\mathbf{x}_{R(T)}$ is then

$$p(\mathbf{x}_{R(T)};\alpha) = \prod_{m=1}^{n_T} p(x^m_{R(T)};\alpha) = \prod_{x_{R(T)}\in\Omega_{R(T)}} p(x_{R(T)};\alpha)^{n_T(x_{R(T)})}$$

and the probability of the complete set of measurement $\mathbf{x}_R$ at the receivers is

$$p(\mathbf{x}_R;\alpha) = \prod_{T\in\Psi} p(\mathbf{x}_{R(T)};\alpha). \quad (7)$$

Our goal is to estimate α by the maximizer of (7), namely,

$$\widehat{\alpha} = \arg\max p(\mathbf{x}_R;\alpha). \quad (8)$$

In [4], a direct expression for $\widehat{\alpha}$ are obtained for the case of a single tree, *i.e.*, when $\#\Psi = 1$. For the general case, unfortunately, we have been unable to obtain a direct expression for $\widehat{\alpha}$. Instead, we follow the approach in [7, 8], and employ the EM algorithm to obtain an iterative approximation $\widehat{\alpha}^{(\ell)}$, $\ell = 0, 1, \ldots,$ to $\widehat{\alpha}$. To understand the idea behind the EM algorithm, assume that we can observe the entire loss process at each node, i.e., assume knowledge of the values $\mathbf{x}_T = (x^1_T, \ldots, x^m_T)$, (with each $x^m_T = (x^m_{k,T})_{k\in V(T)}$), $T \in \Psi$. In this case estimation of α becomes trivial: with complete data knowledge it is easy to realize that the MLE estimate of the success probability $\alpha_{i,j}$ along link (i,j), $\widehat{\alpha}_{i,j}$, is just the fraction of probes successfully transmitted along (i,j), $(i,j) \in E(\Psi)$, i.e.,

$$\widehat{\alpha}_{i,j} = \frac{\sum_{T\in\Psi_{i,j}} n_{j,T}}{\sum_{T\in\Psi_{i,j}} n_{i,T}} \quad (i,j) \in E(\Psi), \quad (9)$$

where $n_{k,T} = \sum_{m=1}^{n_T} x^m_{k,T}$ is the number of probes sent from $\rho(T)$ which arrived to node $k \in V(T)$, $T \in \Psi$.

The EM algorithm assumes complete knowledge of the loss process such that the resulting likelihood has a simple form. Since the complete data, and thus the counts $n_{k,T}$ (except for the leaves nodes) are not known, the EM algorithm proceeds iteratively to augment the actual observations with the unobserved observation at the interior links. Below we briefly describe the algorithm and the intuition behind it. We spell out the detail in Section7.

- *Step 1.* Select an initial link loss rate $\widehat{\alpha}^{(0)}$. The simulation study suggests the values that the algorithm converges to are independent of $\widehat{\alpha}^{(0)}$.
- *Step 2.* Estimate the (unknown) counts $n_{k,T}$ by $\hat{n}_{k,T} = \mathsf{E}_{\widehat{\alpha}^{(\ell)}}[n_{k,T}|\mathbf{x}_R]$. In other words, we estimate the counts by their conditional expectation given the observed data $\mathbf{x}_R$ under the probability law induced by $\widehat{\alpha}^{(\ell)}$.
- *Step 3.* Compute the new estimate $\alpha^{(\ell+1)}$ via (9), using the estimated counts $\hat{n}_{k,T}$ computed in the previous step in place of the actual (unknown) counts $n_{k,T}$. In other words, we set

$$\widehat{\alpha}^{(\ell+1)}_{i,j} = \frac{\sum_{T\in\Psi_{i,j}} \hat{n}_{j,T}}{\sum_{T\in\Psi_{i,j}} \hat{n}_{i,T}} \quad (i,j) \in E(\Psi). \quad (10)$$

- *Step 4.* Iterate steps 2 and 3 until some termination criterion is satisfied. Set $\widehat{\alpha} = \widehat{\alpha}^{(\ell)}$, where ℓ is the terminal number of iterations.

Tree	Source	Receivers
1	0	12 13 14 15 16 17 18 19
2	1	12 13 14 15 16 17 18 19
3	2	12 13 14 15
4	25	16 17 18 19

Table 1: Tree layout for model simulation

As shown in Section7, the EM iterates converges to a local (but not necessarily) global maximizer of (7). However, our simulation results suggests it always converge to the global maximizer $\widehat{\alpha}$ and the convergence does not depend on the initial values.

5. SIMULATION EVALUATION

We evaluate our loss inference algorithms using the ns [18] simulator. This work has two parts: model simulation and network simulation. In the model simulation, losses are determined by time-invariant Bernoulli processes. In the network simulation, losses are due to congestion as probes compete with other background traffic. The majority of the background traffic in the network simulation is produced by TCP flows. However, we do include some on-off flows where the on and off periods have either a Pareto or an exponential distribution. We chose such a mix because TCP is the dominant transport protocol on the Internet.

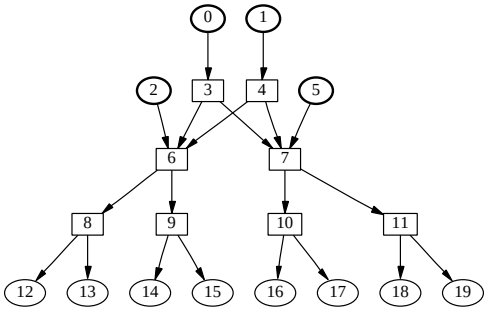


Figure 2: Model simulation topology: Nodes are of three types; bold ellipse: potential sender, ellipse: potential receivers, and box: internal nodes.

5.1 Comparing loss probability

Our approach for comparing two sets of loss probabilities was first introduced in [5]. Assume that we want to compare two loss probabilities p and q. For example p could be an inferred probability on a link, q the corresponding actual probability. For some **error margin** $\varepsilon > 0$ we define the **error factor**

$$F_\varepsilon(p,q) = \max\left\{\frac{p(\varepsilon)}{q(\varepsilon)}, \frac{q(\varepsilon)}{p(\varepsilon)}\right\} \quad (11)$$

where $p(\varepsilon) = \max\{\varepsilon, p\}$ and $q(\varepsilon) = \max\{\varepsilon, q\}$. Thus, we treat p and q as being not less than ε, and having done this, the error factor is the maximum ratio, upwards or downwards, by which they differ. Unless otherwise stated, we used the default value $\varepsilon = 10^{-3}$ in this paper. This choice of metric is motivated by the desire to estimate the relative magnitude of loss ratios on different links in order to distinguish those which suffer higher loss.

5.2 Model simulation

The topology for model simulation is presented in Figure 2. A total of four trees are embedded in the topology as described in Table 1. A time-invariant Bernoulli loss processes is associated with each link. In the simulation, uniform loss rates are assigned to all links.

We use loss rates of 2% and 4% on each link and let each source send equal numbers of probes down to the trees. For each loss rate, we vary the total number of probes sent by all sources from 50 to 1600. Each setting is simulated ten times with different random seeds. For each simulation, we use both the MVWA and EM to estimate loss rates and compare with the actual simulation loss rates.

Figure 3 shows box-plots[1] of error factors between inferred loss and simulated loss over all links and all runs. In the figure, error factors are displayed as a function of number of probes and one graph is for each loss rate. (Note that the total number of probes increase exponentially). In each graph, we plot error factors for both MVWA (abbreviated as WA) algorithm and EM algorithm. Observed from graph that the estimates produced by EM algorithm show greater accuracy and less variability than these produced by MVWA algorithm under both loss rates we simulate when the number of probes are small. However, as the number of probes increases, the estimates yielded by both algorithm become more accurate, the difference between two algorithm become less, and their variability reduces. The same set of simulations were done when the numbers of probes in each tree are different. The results are very close to the case where the numbers of probes are equal.

Note that every link in the topology described in Figure 2 is a segment in at least one of the trees. We also simulated a network embedded by a collection of trees where some links are not a segment in any trees even they are identifiable. The error factors we observed are very similar to those presented in Figure 3.

Since the EM algorithm is more accurate and of less variability than MVWA algorithm, we focus on evaluating EM algorithm in next subsection.

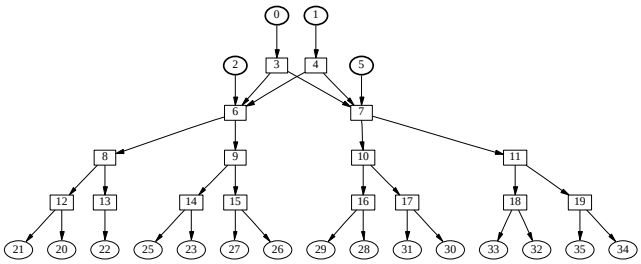


Figure 4: Small network simulation topology: Nodes are of three types; bold ellipse: potential sender, ellipse: potential receivers, and box: internal nodes.

5.3 Network simulation

In this section, we simulate two topologies, a small network in Figure 4 and a multicast topology based on the Abilene network. In both topologies, background traffic is generated by infinite TCP and on-off UDP flows. All the routers in the network are configured to be droptail routers since the droptail routers are prevalent in the Internet.

[1] In a box-plot, the box has lines at the lower quartile, median, and upper quartile values. The whiskers are lines extending from each end of the box to show the extent of the rest of the data. Outliers are data with values beyond the ends of the whiskers.

Tree	Source	Receivers
1	0	20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35
2	1	20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35
3	2	20 21 22 23 24 25 26 27
4	5	28 29 30 31 32 33 34 35

Table 2: Tree layout for small network simulation

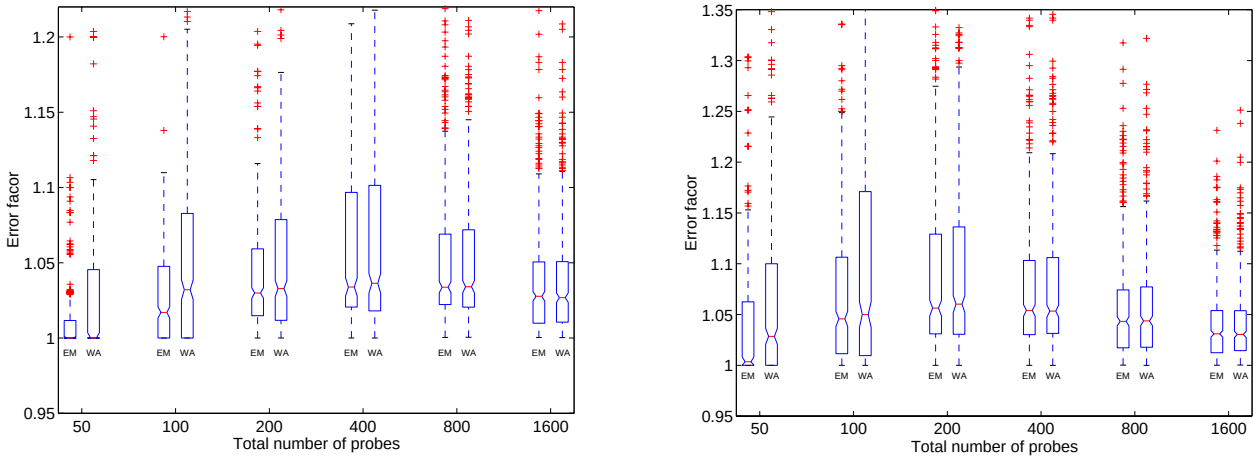
Small network. The tree layout of the small network is described in Table 2. We use constant bit rate probes and the interval between probes is $100ms$. We conducted a total of 7 simulations which differ according to the duration of the measurement. We start with an initial duration of 2 seconds and double it each time until reaching 128 seconds. Each of these simulations is run 10 times with different random seeds. For each simulation, we calculate the loss rates using the EM algorithm.

The link losses in the set of simulations are due to all flows competing for bandwidth. Since different types of flows may exhibit different behavior, the probe flow does not necessarily suffer the same loss rate as the background flows do. Therefore, the error of using inferred loss to estimate the link loss may due to one of the two possibilities. Either probe traffic loss rate differ from all traffic loss rate or the estimates yielded by the EM algorithm do not agree with the probe loss rate. In order to distinguish them, we compare the inferred results to both probe loss rate and all traffic loss rate.

Figure 5 illustrates box plots of error factors for all links and all simulation runs. The error factors are plotted as a function of measurement time. On the left we show the error factor between inferred and simulated all traffic loss; on the right between inferred and simulated probe loss. We observe from both graphs in the figure that both the error factors and their variabilities decrease as the number of probes increase. The improvements are more significant for short measurements.

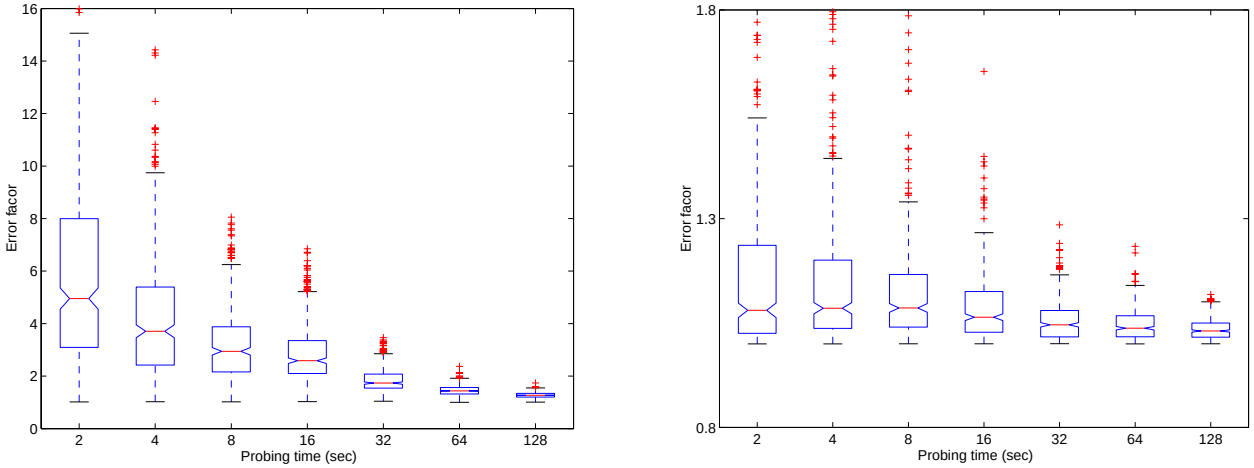
We present scatter plots for the all traffic loss vs. inferred loss on the left and probe traffic loss vs. inferred loss on the right in Figure 6 when the measurement duration is 128 seconds. We make two observations. First, the inferred loss rate almost always overestimates the link loss rate. Second, the inferred loss rate provides a very good estimate of the probe traffic loss rate. The difference between the inferred loss rates and all traffic loss rates is due to that the probe traffic endures a higher loss rate than the rest of traffic. We conjecture that this is because the majority of the background traffic come from infinite TCP flows. TCP reduces its sending rate when the losses are detected. Therefore, fewer TCP packets will suffer loss. However, the CBR source sends probes at a constant rate which is not affected by congestion. We expect the algorithm to be more accurate in the Internet since the Internet contains many short lived TCP flows and many of them complete transmission before they respond to losses.

Abilene network. Abilene [21] is an advanced backbone network that supports the work of Internet2 universities as they develop advanced Internet applications. One major goal of Abilene is to provide a separate network to enable the testing of advanced network capabilities prior to their introduction into the application develop-



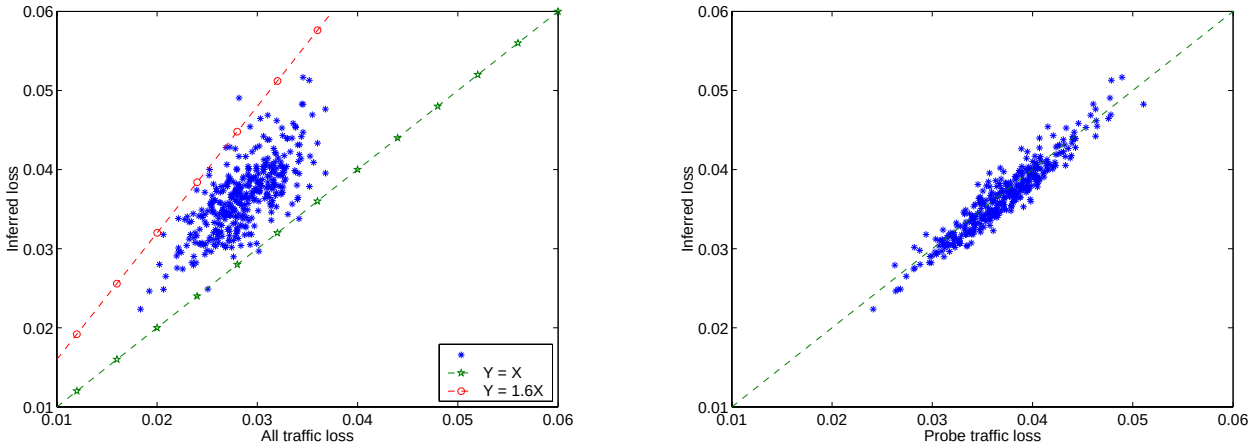
(a) loss = 2% (b) loss = 4%

Figure 3: Accuracy of MVWA(WA) algorithm vs. EM: Box-plot of error factors over all links and all runs for loss rate 2%(left) and 4%(right).



(a) All loss vs. inferred loss (b) Probe loss vs. inferred loss

Figure 5: Accuracy of EM algorithm vs. probing time: Error factor over all links and all runs



(a) All loss vs. inferred loss (b) Probe loss vs. inferred loss

Figure 6: Small network scatter plot: inferred loss vs. all loss, inferred loss vs. probe loss

ment network. Multicast is one among all such services. Abilene supports native and sparse mode multicast. As of October 01, 2001, Multicast protocols, PIM-sparse, MBGP and MSDP have been deployed in the backbone. The Abilene multicast logical topology is illustrated in [22]. It consists of 159 nodes and 165 edges. Each node in the graph represents a physical location and each link represents a physical interconnection between some two routers from different locations. Because the more detailed topology within each physical location is not available to us, we treat each node as a router and focus on the logical topology in our experiments. There are three types of links in Abilene backbone, OC3 (155M), OC12 (622M) and OC48 (2.5G). The type of the links that connect participants to backbone are not labeled and we assume they are T3 (45M). Since the ns simulator does not allow us to simulate enough number of flows to fill up such high bandwidth links and generate losses, we scale down the bandwidth proportionally by 10^8 times. Last, we assume that only the leaves in the topology (i.e., node of degree one) are senders or receivers.

We lay out a total of eight trees that can identify 41 links. An equal number of probes is sent by each source and the interval between probes is $200ms$. We conducted a simulation of duration 256 seconds and ran it ten times with different random seeds. For each simulation, we estimate the loss rates using the EM algorithm and compare them to the simulated loss rates. Figure 7 illustrates scatter plots for inferred loss vs. all loss (left) and inferred loss vs. probe loss (right). Similar to what we observed in small network simulation, the EM algorithm provides accurate estimates of probe loss rates. However, the inferred loss rates are almost always higher than the simulated all traffic loss rates.

6. EXTENSIONS

In this section, we first extend the EM algorithm to infer the distribution of links delay. Second, since multicast is not supported everywhere in the Internet and internal performance observed by multicast packets may differ from that observed by unicast packets, it is important to show our algorithms for inferring a set of multicast trees can be applied to unicast measurements. Last, the algorithms we presented so far rely on the availability of complete information from the receivers. However, this may pose a serious problem in their deployment. We demonstrate the use of our algorithms to handle incomplete observations from receivers.

6.1 Delay inference

We now illustrate the use of end-to-end measurements from a collection of multicast trees Ψ to estimate the delay characteristics of internal links.

We associate with each link (i,j) a random variable $D_{i,j}$ which represents the queueing delay that would be encountered by packets traversing link (i,j). For the analysis, we quantize the queueing delay to a finite set of values $\mathcal{Q} = \{0, q, 2q, \ldots, Bq, \infty\}$, where q is a suitable fixed bin size. A queueing delay equal to ∞ indicates that the packet is lost on the link. We define the bin associated to $iq \in \mathcal{Q}$ to be the interval $[iq - q/2, iq + q/2), i = 1, \ldots, B$, and $[Bq + q/2, \infty)$ the one associated to the value ∞. Because delay is non-negative, we associate with 0 the bin $[0, q/2)$. We thus model the link queueing delay by a nonparametric discrete distribution that we can regard as a discretized version of the actual delay distribution. We denote the distribution of $D_{i,j}$ by $\alpha_{i,j} = (\alpha_{i,j}(d))_{d\in\mathcal{Q}}$, where $\alpha_{i,j}(d) = P[D_{i,j} = d], d \in \mathcal{Q}$. We will denote $\alpha = (\alpha_{i,j})_{(i,j)\in E(\Psi)}$. We assume that queueing delays are independent between different packets, and for the same packets on different links. Thus the progress of each probe down the tree T is described by an independent copy of a stochastic process $Y_T = (Y_{k,T})_{k\in V(T)}$ which represents the accrued queueing delay of packets. The queueing delay experienced by a packet from $\rho(T)$ to node i is $Y_{i,T} = \sum_{(m,n)\in p_T(\rho(T),i)} D_{m,n}$ where $p_T(\rho(T), i)$ denote the path on tree T from source to node i.

In an experiment, a set of probes is sent from the multicast tree sources $\rho(T)$, $T \in \Psi$. For each $T \in \Psi$, we can think of each probe as a trial, the outcome of which is a configuration of source to receivers queueing delays $Y_{R(T)} = (Y_{k,T})_{k\in R(T)}$ we also discretize to the set $\mathcal{Q}$. Each outcome is thus an element of the space $\Omega_{R(T)} = \mathcal{Q}^{\#R(T)}$.

As with loss estimation, we use maximum likelihood estimation based on measurements across the multicast tress $T \in \Psi$. Let us dispatch n_T probes from $\rho(T)$, $T \in \Psi$, and let $n_T(y_{R(T)})$ denote the number of probes for which the outcome $y_{R(T)} \in \Omega_{R(T)}$ is obtained. The probability of the n_T independent observations $\mathbf{y}_{R(T)} = (y^1_{R(T)}, \ldots, y^{n_T}_{R(T)})$, (with each $y^m_{R(T)} = (y^m_{k,T})_{k\in R(T)}$), is then

$$\begin{aligned} p(\mathbf{y}_{R(T)};\alpha) &= \prod_{m=1}^{n_T} p(y^m_{R(T)};\alpha) \\ &= \prod_{y_{R(T)}\in\Omega_{R(T)}} p(y_{R(T)};\alpha)^{n_T(y_{R(T)})} \end{aligned}$$

where $p(y;\alpha) = \mathsf{P}_\alpha[Y_T = y_T]$. The probability of the complete set of measurements $\mathbf{y}_R = (\mathbf{y}_{R(T)})_{T\in\Psi}$ at the receivers is

$$p(\mathbf{y}_R;\alpha) = \prod_{T\in\Psi} p(\mathbf{y}_{R(T)};\alpha). \tag{12}$$

Our goal is to estimate α by the maximizer of (12), namely,

$$\widehat{\alpha} = \arg\max p(\mathbf{y}_R;\alpha). \tag{13}$$

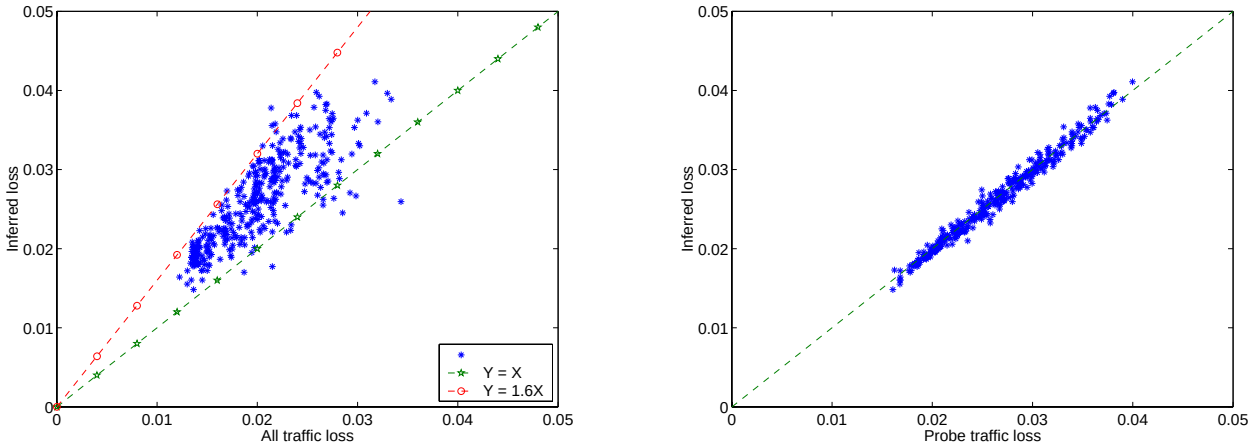
As with loss inference, we resort to the EM algorithm to obtain an iterative solution $\widehat{\alpha}^{(\ell)}$, $\ell = 0, 1, \ldots$, to a (local) maximizer of the likelihood (12). Assume complete knowledge of the delay process at each link, namely the values $\mathbf{y}_T = (y^1_T, \ldots, y^m_T)$, (with each $y^m_T = (y^m_{k,T})_{k\in V(T)}$), $T \in \Psi$. Denote by $n_{i,j,T}(d)$ the total number of packets sent by $\rho(T)$ that experienced a delay equal to d along link (i,j). It is easy to verify that with complete data, the MLE estimate of $\alpha_{i,j}(d)$ is

$$\widehat{\alpha}_{i,j}(d) = \frac{\sum_{T\in\Psi_{i,j}} n_{i,j,T}(d)}{\sum_{T\in\Psi_{i,j}}\sum_{d\in\mathcal{Q}} n_{i,j,T}(d)} \quad \forall (i,j) \in E(\Psi). \tag{14}$$

Thus, with complete knowledge, the MLE estimate of $\alpha_{i,j}(d)$ is simply the fraction of the probes traversing link (i,j) which encountered a delay equal to d.

For delay inference the EM algorithm proceeds as for the loss case. Below we briefly describe the algorithm and intuition behind it. Details can be found in [3].

1. *Step 1.* Select the initial link delay distribution $\widehat{\alpha}^{(0)}$.
2. *Step 2.* Given the current estimate $\widehat{\alpha}^{(\ell)}$, Estimate the (unknown) counts $n_{i,j,T}(d)$ by $\hat{n}_{i,j,T}(d) = \mathsf{E}_{\widehat{\alpha}^{(\ell)}}[n_{i,j,T}(d)|\mathbf{y}_R]$. In other words, we estimate the counts by their conditional expectation given the observed data $\mathbf{y}_R$ under the probability law induced by $\widehat{\alpha}^{(\ell)}$.



(a) All loss vs. inferred loss

(b) Probe loss vs. inferred loss

Figure 7: Abilene scatter plot: inferred loss vs. all loss, inferred loss vs. probe loss

3. *Step 3.* Compute the new estimate $\alpha^{(\ell+1)}$ via (14), using the estimated counts $\hat{n}_{i,j,T}(d)$ computed in the previous step in place of the actual (unknown) counts $n_{i,j,T}(d)$.

4. *Iteration.* Iterate steps 2 and 3 until some termination criterion is satisfied. Set $\widehat{\alpha} = \widehat{\alpha}^{(\ell)}$, where ℓ is the terminal number of iterations.

Complexity
The complexity of the algorithm is dominated by the computation the conditional expectations which can be accomplished in time linear with $\#V(T) \times \#\mathcal{Q}$, $T \in \Psi$. The computation can be done by extending the approach for computing loss conditional probability and is described in [3].

Convergence
The conditions for convergence can be established similarly as for loss inference.

6.2 Inference with unicast measurement

So far we have presented inference algorithms for a collection of trees based on end-to-end multicast measurements. These techniques can be extended to work with unicast measurements from multiple sources as well.

The rationale behind unicast based inference is that: (1) measurement domain is limited because large portions of the Internet do not support network-level multicast, and that (2) the internal performance observed by multicast packets may differs from that observed by unicast packets. Techniques for unicast measurements and inference have been recently proposed in [6, 11] for the inference of loss rates and [7, 8, 9] for delay distributions. However, these works only handle the inference of a *single* source with multiple pairs of receivers and thus may pose severe limitations in scope.

The key idea behind unicast inference is to design unicast measurement whose correlation properties closely resemble those of multicast traffic, so that it becomes possible to use the inference techniques developed for multicast inference; the closer the correlation properties are to that of multicast traffic, the more accurate the results.

A basic approach for unicast inference is to dispatch two back-to-back packets (a packet pair) from a probe source to a pair of distinct receivers. For each such packet pair, the two packets traverse a common set of links down a node where their paths diverge to the two receivers. By choosing *multiple* sources and pairs of receivers, it is possible to cover a more significant portion of a network than with a single source. The inference for the link loss probability and link delay distribution from a set of packet pair measurements is formulated as a maximum likelihood estimation problem which is then solved using the algorithms we presented earlier in the paper. The idea, is that treat the unicast packet pair measurements as statistically equivalent to a notion multicast packet that descends the same tree. The entire set of measurements is thus considered equivalent to a set of multicast measurements down a collection of 2 leaf trees. The analysis then follows the same approach for a collection of trees detailed in Sections 4 and 6.1.

6.3 Inference with missing data

The algorithms presented in the paper so far rely on the availability of complete information from the receivers. However, as described in [10], this may pose a serious problem in their deployment. For example, the loss reports from receives may be delivered unreliably and there may be bandwidth constraints for transmitting loss reports. Therefore, it is important to extend the algorithms to handle incomplete data sets. An algorithm has been proposed in [10] to handle incomplete data for a single tree. The goal of this section is to extend the algorithms we proposed earlier in the paper to handle incomplete data for a collection of trees.

The basic idea is first to convert each tree $T \in \Psi$ with incomplete observations to multiple sub-trees sharing the same source but with complete observations. For tree T with incomplete data in a collection of tree Ψ, assume that the outcomes of the kth probe sent by $\rho(T)$ are only observable by $R_k(T) \subseteq R(T)$. With probe k, we associate the multicast tree T_k that spans the root $\rho(T)$ and $R_k(T)$. This is obtained by finding the spanning tree of $\rho(T)$ and $R_k(T)$ in T. Therefore, the tree T with incomplete observation can be treated as a set of trees $\{T_k\}_{k=1,\ldots,n_T}$, each of which is with complete observation. Note that the same tree may appear many times in $\{T_k\}_{k=1,\ldots,n_T}$ and can be merged as one tree with multiple probes. For each tree with incomplete data in Ψ, we replace it with the set of its subtrees with complete data and add these trees

to Ψ. We then have a set of trees each of which has complete data and the algorithms described in Sections 4 and 6.1 can be applied to the inference of loss rate and delay distribution.

7. EM ALGORITHM FOR LOSS INFERENCE

We find convenient to work with the log-likelihood function

$$\mathcal{L}^{\rm inc}(\mathbf{x}_R;\alpha) = \sum_{T\in\Psi} \mathcal{L}^{\rm inc}{}_T(\mathbf{x}_{R(T)};\alpha) \tag{15}$$

where
$\mathcal{L}^{\rm inc}{}_T(\mathbf{x}_{R(T)};\alpha) = \sum_{x_{R(T)}\in\Omega_{R(T)}}(n_T(x_{R(T)})\log p(x_{R(T)};\alpha))$ is the log-likelihood of the the measurement down the tree $T\in\Psi$. We estimate α by the maximizer of the likelihood (15), namely $\widehat{\alpha} = \arg\max \mathcal{L}^{\rm inc}{}_T(\mathbf{x}_{R(T)};\alpha)$. We follow the approach in [7, 8] and employ the EM algorithm to obtain an iterative approximation to the maximizer of (15). The basic idea is that rather than performing a complicated maximization, we "augment" the observed data with *unobserved* or *latent* data so that the resulting log-likelihood has a simpler form. Following [8], we augment the actual observations with the *unobserved* observations at the interior links. In other words, we assume complete knowledge of the loss process. The log-likelihood for the *complete data* $\mathbf{x} = (\mathbf{x}_T)_{T\in\Psi}$ is

$$\mathcal{L}(\mathbf{x};\alpha) = \sum_{T\in\Psi} \mathcal{L}(\mathbf{x}_T;\alpha) \tag{16}$$

where $\mathcal{L}(\mathbf{x}_T;\alpha) = \log p(\mathbf{x}_T;\alpha)$ is the log-likelihood of the complete set data for T. It is easy to realize that $p(x_T^1,\ldots,x_T^{n_T};\alpha) = \prod_{(i,j)\in E(T)} \alpha_{i,j}^{n_{j,T}}\overline{\alpha}_{i,j}^{n_{i,T}-n_{j,T}}$ and that

$$\begin{aligned}\mathcal{L}(\mathbf{x};\alpha) &= \sum_{(i,j)\in E(\Psi)}\Big(\sum_{T\in\Psi_{i,j}} n_{j,T}\log\alpha_{i,j} \\ &\quad + \Big(\sum_{T\in\Psi_{i,j}} n_{i,T} - \sum_{T\in\Psi_{i,j}} n_{j,T}\Big)\log\overline{\alpha}_{i,j}\Big).\end{aligned} \tag{17}$$

Maximization of (17) is trivial, as the stationary point conditions

$$\frac{\partial\mathcal{L}(\mathbf{x};\alpha)}{\partial\alpha_{i,j}} = 0 \quad (i,j)\in E(\Psi) \tag{18}$$

immediately yield

$$\widehat{\alpha}_{i,j} = \frac{\sum_{T\in\Psi_{i,j}} n_{j,T}}{\sum_{T\in\Psi_{i,j}} n_{i,T}} \quad (i,j)\in E(\Psi). \tag{19}$$

Since $\mathbf{x}$ and thus the counts except for leaves are not known, the EM algorithm uses the complete log-likelihood $\mathcal{L}(\mathbf{x};\alpha)$ to iteratively find $\widehat{\alpha}$ as follows:

1. *Initialization.* Select the initial link loss rate $\widehat{\alpha}^{(0)}$. The simulation study suggests the values that the algorithm converges to are independent of initial values.

2. *Expectation.* Given the current estimate $\widehat{\alpha}^{(\ell)}$, compute the conditional expectation of the log-likelihood given the observed data $\mathbf{x}$ under the probability law induced by $\widehat{\alpha}^{(\ell)}$,

$$\begin{aligned}Q(\alpha';\widehat{\alpha}^{(\ell)}) &= E_{\widehat{\alpha}^{(\ell)}}[\mathcal{L}(\mathbf{x};\alpha')|\mathbf{x}_R] \\ &= \sum_{(i,j)\in E(\Psi)}\Bigg\{\sum_{T\in\Psi_{i,j}}\widehat{n}_{j,T}\log\alpha'_{i,j} \\ &\quad + \Big(\sum_{T\in\Psi_{i,j}}\widehat{n}_{i,T} - \sum_{T\in\Psi_{i,j}}\widehat{n}_{j,T}\Big)\log\overline{\alpha}'_{i,j}\Big)\Bigg\}\end{aligned} \tag{20}$$

where $\hat{n}_{k,T} = \mathsf{E}_{\widehat{\alpha}^{(\ell)}}[n_{k,T}|\mathbf{x}_R]$. $Q(\alpha';\widehat{\alpha}^{(\ell)})$ has the same expression as $\mathcal{L}(\mathbf{x};\alpha')$ but with the actual *unobserved* counts $n_{k,T}$ replaced by their conditional expectations $\widehat{n}_{k,T}$. To compute $\hat{n}_{k,T}$, remember that $n_{k,T} = \sum_{m=1}^{n_T} x_{k,T}^m$. Thus, we have

$$\begin{aligned}\widehat{n}_{k,T} &= \sum_{m=1}^{n_T}\mathsf{P}_{\widehat{\alpha}^{(\ell)}}[X_{k,T}=1|X_{R(T)}=x_{R(T)}^m] \\ &= \sum_{x_{R(T)}\in\Omega_{R(T)}} n_T(x_{R(T)})\mathsf{P}_{\widehat{\alpha}^{(\ell)}}[X_{k,T}=1|X_{R(T)}=x_{R(T)}]\end{aligned} \tag{21}$$

3. *Maximization.* Find the maximizer of the conditional expectation $\alpha^{(\ell+1)} = \arg\max_{\alpha'} Q(\alpha',\widehat{\alpha}^{(\ell)})$. The maximizer is given by (19) with the conditional expectation $\hat{n}_{k,T}$ in place of $n_{k,T}$.

4. *Iteration.* Iterate steps 2 and 3 until some termination criterion is satisfied. Set $\widehat{\alpha} = \widehat{\alpha}^{(\ell)}$, where ℓ is the terminal number of iterations.

Complexity

The complexity of the algorithm is dominated by computation of the conditional expectation $\widehat{n}_{k,T}$. This can be accomplished in linear time with $\#V(T)$, $T\in\Psi$. The algorithm is described in [3].

Convergence

We establish conditions for convergence of estimated parameters and likelihood under the EM algorithm for loss inference. Observe that the complete data log-likelihood function (17) can be written

$$\mathcal{L}(\mathbf{x};\alpha) = \sum_{T\in\Psi}\sum_{i\in V(T)\setminus\{\rho(T)\}} n_{i,T}\phi_{i,T}(\alpha) \tag{22}$$

where

$$e^{\phi_{i,T}(\alpha)} = \frac{\alpha_{f(i,T),i}}{\overline{\alpha}_{f(i,T),i}}\prod_{j\in d(i,T)}\overline{\alpha}_{i,j} \tag{23}$$

(Here the empty product when $d(i,T)=\emptyset$ is taken as 1). Thus the log likelihood comes from an exponential family with sufficient statistics $(n_{i,T})_{T\in\Psi,\ i\in V(T)}$ and parameters α. The exponential family is *regular*, since we take α in the convex set $\mathcal{A} = (0,1)^{\times_{T\in\Psi}V(T)}$. Note that the map $\alpha\mapsto\phi$ is invertible: $e^{\phi_{i,T}} = \alpha_{f(i,T),i}/\overline{\alpha}_{f(i,T),i}$ for a receiver i in $R(T)$. Invertibility then follows by induction: if we know all the $(\alpha_{i,j})_{j\in d(i,T)}$ then we can recover $\alpha_{f(i,T),i}$ from ϕ_i. It follows that the exponential family is *curved*: the $\phi_{i,T}$ are constrained to some $\#V$-dimensional smooth submanifold of $\mathbb{R}^{\times_{T\in\Psi}V(T)\setminus\{\rho(T)\}}$ through the constraint that the link probabilities α calculated from ϕ_T on different trees T must agree on common links.

The following convergence results for the sequence of EM iterates $\hat{\alpha}^{(\ell)}$ follow from the regular exponential family property; see Theorem 6 in [20].

THEOREM 3. *(i) $\mathcal{L}^{\rm inc}(\mathbf{x}_R;\hat{\alpha}^{(\ell)})$ converges to some limit L.*

(ii) If $\{\alpha\in\mathcal{A}\mid\mathcal{L}^{\rm inc}(\mathbf{x}_R;\alpha)=L\}$ is discrete, $\hat{\alpha}^{(\ell)}$ converges to some α^* that is a stationary point of $\mathcal{L}^{\rm inc}(\mathbf{x}_R;\alpha)$.

(iii) If $\mathcal{L}i(\mathbf{x}_R;\alpha)$ is unimodal, $\hat{\alpha}^{(\ell)}$ converges to the incomplete data MLE, namely, $\hat{\alpha} = \arg\max_\alpha\mathcal{L}^{\rm inc}(\mathbf{x}_R;\alpha)$

The theorem implies that when there are multiple stationary points, *e.g.* local maxima, the EM iterates may not converge to the global maximizer. Unfortunately, we were not able to establish whether there is a unique stationary point or conditions under which unicity holds.

8. SUMMARY

In this paper, we focused on inferring network internal link-level performance from end-to-end multicast measurements taken from a collection of trees. We addressed two questions:

- Given a collection of multicast trees, whether all of the links (or a specified subset) are identifiable.
- If a set of links of interest are identifiable, how do we obtain accurate estimates of their performance.

With loss rates as performance metrics, we established necessary and sufficient conditions for identifiability; and proposed two algorithms, MVWA algorithm and EM algorithm for inferring a set of links of interests. The algorithms are evaluated through model simulation and network simulation. The model simulation suggests that the EM algorithm is more accurate and of less variability. In the network simulation, we observe that EM algorithm can provide accurate estimate to the probe traffic loss whereas over-estimate all traffic loss slightly. Moreover, we extend the EM algorithm infer link delays, and demonstrate how to use our algorithms when only unicast measurement are available or some of the observations made at end-hosts are missing.

9. REFERENCES

[1] A. Adams, T. Bu, R. Caceres, N.G. Duffield, T. Friedman, J. Horowitz, F. Lo Presti, S.B. Moon, V. Paxson, and D. Towsley. "The Use of End-to-End Multicast Measurements for Characterizing Internal Network Behavior", *IEEE Communications Magazine*, May 2000.

[2] M. Adler, T. Bu, R. Sitaraman, and D. Towsley. "Tree Layout for Internal Network Characterizations in Multicast Networks", *Proc. NGC'01*, London, UK, Nov. 2001.

[3] T. Bu, N.G. Duffield, F. Lo Presti, and D. Towsley. "Network Tomography on General Topologies". *UMass CMPSCI Technique Report*.

[4] R. Cáceres, N.G. Duffield, J.Horowitz, and D. Towsley. "Multicast-Based Inference of Network Internal Loss Characteristics" IEEE Trans. on Information Theory, vol. 45, pp. 2462-2480, 1999.

[5] R. Caceres, N.G. Duffield, J. Horowitz, D. Towsley, and T. Bu. "Multicast-Based Inference of Network-Internal Characteristics: Accuracy of Packet Loss Estimation". *Proceedings of INFOCOM'99*.

[6] M. Coates and R. Nowak. "Network loss inference using unicast end-to-end measurement", *Proc. ITC Conf. IP Traffic, Modeling and Management*, Monterey, CA, September 2000.

[7] M. Coates and R. Nowak. "Sequential Monte Carlo Inference of Internal Delays in Nonstationary Communication Networks," submitted for publication, Jan 2001.

[8] M.J. Coates and R. Nowak. "Network Delay Distribution Inference from End-to-end Unicast Measurement," Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing, May 2001.

[9] N.G. Duffield, J. Horowitz, F. Lo Presti, and D. Towsley. "Network Delay Tomography from End-to-End Unicast Measurements", Proc. of the 2001 International Workshop on Digital Communications 2001 Evolutionary Trends of the Internet, Taormina, Italy, September 2001.

[10] N.G. Duffield, J. Horowitz, D. Towsley, W. Wei, and T. Friedman. "Multicast-based loss inference with missing data", to appear in IEEE Journal of Selected Areas in Communications

[11] N.G. Duffield, F. Lo Presti, V. Paxson, and D. Towsley. "Inferring Link Loss Using Striped Unicast Probes", Proc. IEEE INFOCOM 2001, Anchorage, AK, April 2001.

[12] Omer Gurewitz and Moshe Sidi. "Estimating One-way Delays from Cyclic-Path Delay Measurements", Proc. IEEE INFOCOM 2001, Anchorage, AK, April 2001.

[13] Khaled Harfoush, Azer Bestavros, and John Byers. "Robust Identification of Shared Losses Using End-to-End Unicast Probes", Proc. IEEE ICNP 2000, Osaka, Japan.

[14] K. Lai and M. Baker. "Measuring link bandwidths using a deterministic model of packet delay," *Proc. SIGCOMM 2000*, Sweden, August 2000.

[15] Geoffrey J. McLachlan and Thriyambakam Krishnan. The EM algorithm and extensions. John Wiley, New York (1997)

[16] R. Penrose. "A Generalized Inverse for Matrices." Proc. Cambridge Phil. Soc. 51, 406-413, 1955.

[17] F. Lo Presti, N.G. Duffield, J.Horowitz, and D. Towsley. "Multicast-Based Inference of Network-Internal Delay Distributions", submitted for publication, September 1999.

[18] `ns` – Network Simulator. See http://www-mash.cs.berkeley.edu/ns/ns.html

[19] Y. Shavitt, X. Sun, A. Wool, and B. Yener. "Computing the unmeasured: an algebraic approach to mapping the Internet," Proc. IEEE INFOCOM 2001, Anchorage, AK, April 2001.

[20] C.F. Jeff Wu. "On the convergence properties of the EM algorithm", Annals of Statistics, vol. 11, pp. 95-103, 1982.

[21] Abilene Network Operations Center. http://www.abilene.iu.edu/

[22] The Abilene network multicast deployment. http://www.abilene.iu.edu/images/ab-mcast.pdf