Network Tomography From Measured End-to-End Delay Covariance

Article *in* IEEE/ACM Transactions on Networking · January 2005

DOI: 10.1109/TNET.2004.838612 · Source: IEEE Xplore

Nick Duffield
Texas A&M University
221 PUBLICATIONS 12,248 CITATIONS
SEE PROFILE

Francesco Lo Presti
University of Rome Tor Vergata
106 PUBLICATIONS 4,009 CITATIONS
SEE PROFILE

Network Tomography From Measured End-to-End Delay Covariance

N. G. Duffield, *Senior Member, IEEE,* and Francesco Lo Presti

***Abstract*—End-to-end measurement is a common tool for network performance diagnosis, primarily because it can reflect user experience and typically requires minimal support from intervening network elements. However, pinpointing the site of performance degradation from end-to-end measurements is a challenging problem. In this paper, we show how end-to-end delay measurements of multicast traffic can be used to infer the under-lying logical multicast tree and the packet delay variance on each of its links. The method does not depend on cooperation from intervening network elements; multicast probing is bandwidth efficient. We establish desirable statistical properties of the estimator, namely consistency and asymptotic normality. We evaluate the approach through simulations, and analyze its failure modes and their probabilities.**

***Index Terms*—End-to-end measurement, multicast, packet delay, statistical inference, topology discovery.**

I. Introduction

A. Background and Motivation

Monitoring the performance of large communications networks and diagnosing the causes of its degradation is a challenging problem. There are two broad approaches. In the *internal* approach, direct active or passive measurements are made at or between network elements. This approach has a number of potential limitations: 1) it may not be available for general users; 2) coverage may not span paths of interest; 3) measurements may be disabled during period of high load; 4) there are issues of scale gathering and correlating the measurements in large networks; and 5) composing per-hop measurements to form an end-to-end view is a challenge.

This motivates *external* approaches, diagnosing problems through end-to-end measurements, without necessarily assuming the cooperation of network elements. There has been much recent experimental work to understand the phenomenology of end-to-end performance (e.g., see [2], [9], [21], [23], [25]–[27]). There are presently several measurement infrastructure projects (including CAIDA[1], IPMA[2], NIMI [24], Surveyor[3]) that collect and analyze end-to-end measurements across a mesh of paths between a number of hosts. The `ping` and `traceroute` diagnostic tools are widely used to determine connectivity, round-trip loss, and delay in IP networks. `pathchar` [10] extends the approach of `traceroute` to estimate hop-by-hop link capacities, packet delay, and loss rates. These approaches have several potential drawbacks: 1) delays may not be representative of regular traffic, since their generation of Internet Control Message Protocol (ICMP) packets can have low priority in routers; 2) round-trip reporting and possibly asymmetric paths hinder the unambiguous attribution of delays to specific link directions; and 3) ICMP traffic can be disabled by network administrators (a recent study [1] found that more then 50% of probed nodes did not reply to ICMP echo messages because of ICMP filtering along the path).

In response to some of these concerns, a multicast-based approach to active measurement has been proposed in [3]. The idea is that correlation in performance seen on *intersecting* end-to-end paths can be used to draw inferences about the performance characteristics of their common portion, without cooperation from the network. Multicast traffic is well suited for this since a given packet only occurs once per link in the multicast tree. End-to-end characteristics seen at different endpoints are then highly correlated. In [3], it was shown how to exploit these correlations in order to determine the per link loss rates in the underlying logical multicast tree. Variations of the basic idea enable the estimation of the packet delay distribution [18] and the underlying multicast topology [15].

Another advantage of using multicast is scalability. Suppose packets are exchanged on a mesh of paths between a collection of N measurement hosts stationed in a network. With unicast, the probe load may grow proportionally to N^2 in some links of the network. With multicast, the load grows proportionally to N. Offsetting these advantages, multicast is not currently widely deployed, which limits the coverage of the methods. In response to this limitation, unicast variations of the inference methods have been proposed; see Section I-D.

B. Contribution

This paper describes a method to infer the variance of internal link delays from measured end-to-end delays of multicast probe packets. Furthermore, this data can be used to determine the logical multicast topology if it is not supplied in advance. The method rests on (generalizations of) the following observation. Assume first that link delays are independent random

Manuscript received April 18, 2000; approved by IEEE/ACM Transactions on Networking Editor E. Zegura. This work was sponsored in part by the Defense Advanced Research Projects Agency and the Air Force Research Laboratory under Agreement F30602-98-2-0238.

N. G. Duffield is with AT&T Labs-Research, Florham Park, NJ 07932 USA (e-mail: duffield@research.att.com).

F. Lo Presti is with Dipartimento di Informatica, Università dell'Aquila, 67010 Coppito (AQ), Italy (e-mail: lopresti@di.univaq.it).

Digital Object Identifier 10.1109/TNET.2004.838612

[1] CAIDA: Cooperative Association for Internet Data Analysis. [Online]. Available: http://www.caida.org

[2] IPMA: Internet Performance Measurement and Analysis. [Online]. Available: http://www.merit.edu/ipma

[3] Surveyor: An Infrastructure for Internet Performance Measurements. [Online]. Available: http://www.isoc.org/inet99/proceedings/4h/4h_2.htm

1063-6692/04$20.00 © 2004 IEEE

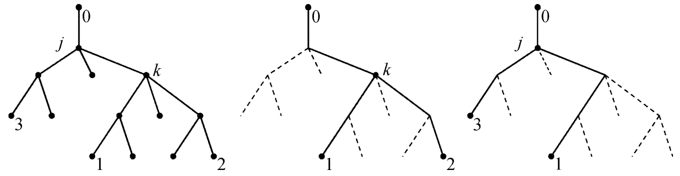


Fig. 1. Logical multicast tree (left) and two embedded two receivers trees (center and right).

variables, both spatially (i.e., between different links) and temporally (i.e., between different packets). Consider the logical multicast topology of the left side of Fig. 1, in which packets are multicast from the root 0 to receivers at leaf nodes. Let D_i be the delay experienced by packets on link i, and let X_i be the cumulative delay experienced along the path from the root 0 to node i. Focus on the embedded two-leaf tree formed by the root 0, leaf nodes 1 and 2, and their nearest common ancestor k; see the center of Fig. 1. From the independence of link delays, we have

$$\mathrm{Var}(X_k) = \mathrm{Cov}(X_1, X_2). \tag{1}$$

A more formal proof is given later. Similarly, consider the tree formed by the root 0, the leaf nodes 1 and 3, and their nearest common ancestor j; see the right-hand side of Fig. 1. Then $\mathrm{Var}(X_j) = \mathrm{Cov}(X_1, X_3)$. Observe that $X_k = X_j + D_k$, and X_j and D_k are independent. Therefore,

$$\mathrm{Var}(D_k) = \mathrm{Var}(X_k) - \mathrm{Var}(X_j) = \mathrm{Cov}(X_1, X_2) - \mathrm{Cov}(X_1, X_3). \tag{2}$$

This expresses the variance of the packet delay on the internal link from node j to node k in terms of the covariances of source-to-leaf delays. An unbiased estimate of the latter can be formed directly from end-to-end measurements, yielding unbiased estimators for (1) and (2). Section II specifies the delay model, and these basic estimators for a known topology.

In a general topology, there exists a convex family of unbiased delay variance estimators based on (1) and (2). Each is consistent, i.e., it converges almost surely to the true value. Section III presents estimators for cumulative and link delay variance that have the fastest asymptotic rate of convergence as the number of probes increases. Packet loss reduces the number of packets available for delay estimation, hence slowing convergence rates. We quantify this and describe a version of our estimators that makes maximal use of information from surviving packets. This requires the inversion of an empirical covariance matrix whose dimension grows rapidly with the number of leaf nodes of the tree. In the case of a binary tree, we are able to make use of the natural recursive structure of the tree to simplify the calculation. An algorithm for this is provided in Section IV.

In Section V, we extend the approach to infer the logical multicast topology when this is not supplied in advance. This is based upon the observation that, when link delays are independent, the cumulative delay variance is increasing along paths from the root. According to (1), a sibling pair can be identified by the criterion that their delay covariance is maximal. Repeated application of this criterion allows any binary tree to be identified from the measured delay covariances. This approach is inspired by a related method for the inference of binary trees from end-to-end multicast loss [15], [28]. The method here extends to general trees. We prove that the resulting topology estimator is consistent and evaluate it through model-based simulations in Section VI. A closer analysis of the modes of failure, and their asymptotic probabilities, is made in Section VII. We conclude in Section VIII. Proofs of the theorems are deferred to the Appendix.

C. *Implementation Requirements and Applications*

Realization of multicast inference in the Internet requires the availability of participating end-hosts and the transmission of measurement data from the end-hosts to a common location for inference. To date, loss multicast inference has been deployed on a measurement infrastructure comprising a number of hosts, either dedicated or co-opted for measurements. Scheduling and coordination of the measurement and data transmission functions is managed using the National Internet Measurement Infrastructure (NIMI) [20]. We plan to supplement these with delay-based variance inference. Physical topology is currently laid out using the `mtrace`[4] measurement tool. `mtrace` reports the route from a multicast source to a receiver, along with other information about that path such as per-hop loss and rate. Presently it does not support delay measurements. A potential drawback for larger topologies is that `mtrace` does not scale to large numbers of receivers because it needs to run once for each receiver to cover the entire multicast tree. In addition, it relies on multicast routers responding to explicit measurement queries; the feature that can be administratively disabled. As an alternative, we propose topology changes could be detected from ongoing measurements using the methods presented here. Changes in the logical multicast topology would then trigger appropriate `mtrace` measurements to determine changes in the physical topology. A similar approach could be adopted by multicast applications which benefit from and/or require knowledge of the multicast topology. Several reliable multicast protocols rely on logical hierarchies based on the underlying topology if possible; see, e.g., [24]. Other applications attempt to group receivers that share the same network bottleneck [28].

While there are benefits to using a fixed infrastructure (e.g., reliability), administrative issues and the use of specialized hosts limits the ubiquity of multicast inference. In addition, transmission of measured data to a common point for inference suffers from the scaling implosion problem, because the aggregate traffic volume is proportional to the number of receivers.

To address these potential limitations, a solution which does not require specialized hosts and that controls the amount of measurement data transmitted has been recently proposed in [5]. This approach exploits the Real-Time Transport Protocol (RTP) and its control protocol (RTCP) [30]. RTP is used to carry multicast audio and video over the Internet. Receivers use RTCP to periodically multicast reports to the group for transmission control. In this application, any ongoing RTP transmission (e.g., an RTP audio feed) is regarded as a measurement probe stream, with extended RTCP reporting detailed per-packet measurements [17]. Any third-party host joining the multicast group

[4] [Online]. Available: ftp://ftp.parc.xerox.com/pub/net-research/ipmulti

of ongoing RTCP sessions can monitor RTCP reports of these sessions, collect measurements, and perform inference.

Accuracy of the inferred network characteristics depends on the stability over time of network characteristics and, especially in the in the case of RTP measurements, on the relative stability of the multicast tree during the measurement period. Consider, for example, a 10-kByte/s probe packet stream comprising one 200-byte packet every 20 ms, equivalent to a compressed audio transfer. Ten thousand such packets (even if we often observe convergence with many fewer probes) would require a measurement period of 200 s for transmission, i.e., just over 3 min. Recent measurements show this to be shorter than the typical timescales of changes in end-to-end delay behavior, at least for unicast traffic. In [31], round-trip delay statistics were found to be reasonably constant over timescales of 10–30 min. Earlier measurements [7], [23] also exhibited level shifts dividing periods of relative constancy in delay statistics over a timescale of hours. Thus, it is reasonable to expect that, in practice, measurements could usually be completed over periods of relative constancy. We also expect user participation in such sessions to last longer than the 3-min figure given above, in which case relatively stable measurement topology would be covered. This is not a problem in the case of a fixed infrastructure, e.g., NIMI. In case of monitoring an ongoing RTP session, the stability of user membership could affect the inference, depending on the membership dynamics. This will be the subject of future study.

All of the proposed techniques can be also effectively combined with direct measurement approaches to overcome the potential limitations of the latter. Although disablement of ICMP is not currently widespread, nevertheless, direct one-way delay measurement techniques that depend upon them have been found to have limitations when resolution down to individual links is required. Indeed, a recent proposal [1] combines ICMP measurements with tomographic delay inference with the aim of resolving delays at a finer spatial granularity.

The delay-variance estimates themselves can be used to detect links of higher delay variance. The variance of the packet delay (on a link or path) can be used to estimate or bound the variance of the interpacket delay variation. Let D^i be the delay encountered by packet i on a given link. The interpacket delay variation (or jitter) between packets i and $i+1$ on the link is $J^i = D^{i+1} - D^i$. Observe $\mathrm{Var}(J^i) = \mathrm{Var}(D^i) + \mathrm{Var}(D^{i+1}) - 2\mathrm{Cov}(D^i, D^{i+1})$. Assuming stationarity and independence, $\mathrm{Var}(J^i) = 2\mathrm{Var}(D^i)$. Measurements of end-to-end delays in the Internet [2] show that end-to-end delays in successive packets are only slightly dependent when the interpacket time is longer than the typical queueing timescales. The dependence is stronger at shorter timescales: successive packets are more likely to queue together. With positive correlation between successive probe delays, $\mathrm{Cov}(D^i, D^{i+1}) > 0$; in this case, $\mathrm{Var}(J^i)$ is bounded above by $2\mathrm{Var}(D^i)$, a quantity that we can estimate.

D. Related Work on Unicast Measurements

There has been increasing interest in methodologies for characterizing link-level behavior from end-to-end measurements. In particular, methods to extend the inference techniques to unicast measurements have been recently proposed in [8] and [14] for the inference of loss rates and [8] and [12] for delay distributions and covariance. The premise is that unicast measurements could be used to complement multicast measurements for those portions of the network which do not support multicast.

Use of end-to-end measurements of packet pairs in a tree connecting a single sender to several receivers for estimation of the link delay has been first considered in [8]. The inference of the link-delay distribution is formulated as a maximum-likelihood estimation problem. In [11], we extend the results in [8] and describe techniques to infer the variance of internal link delays from end-to-end measurements of packet pairs in a tree which generalize the methods we present in this paper.

II. Delay Trees and Nonparametric Estimation

A. Tree Model

The physical multicast tree comprises network elements (the nodes) and the communication links that join them. The logical multicast tree comprises the branch points of the physical tree and the logical links between them. A logical link comprises a chain of one or more physical links. Thus, each node in the logical tree, except the leaf nodes and possibly the root, have two or more children. We can construct the logical tree from the physical tree by deleting all links with one child and adjusting the links accordingly by directly joining its parent and child.

Let $\mathcal{T} = (V, L)$ denote a logical multicast tree with nodes V and links L. We identify one node, the root 0, with the source of probes, and $R \subset V$ will denote the set of leaf nodes (identified as the set of receivers). A link is internal if neither of its endpoints is the root or a leaf node. W will denote $V \setminus (\{0,1\} \cup R)$, where 1 denotes the child node of 0, the set of nodes terminating internal nodes. The set of children of node $j \in V$ is denoted by $d(j)$. Each node, k, apart from the root, has a parent $f(k)$ such that $(f(k), k) \in L$; for simplicity, we shall refer to this link as link k. Define recursively the compositions $f^n = f \circ f^{n-1}$ with $f^1 = f$. Nodes are said to be siblings if they have the same parent. If $k = f^m(j)$ for some $m \in \mathbb{N}$, we say that j is descended for k (or, equivalently, that k is an ancestor of j) and write the corresponding partial order in V as $j \prec k$. $i \vee j$ will denote the nearest (i.e., $\preceq$-minimal) common ancestor of i and j.

B. Delay-Variance Tree Model

The delay on link k is a random variable D_k taking values in the extended positive real line $\overline{\mathbb{R}} = \mathbb{R}_+ \cup \{\infty\}$. By convention, $D_0 = 0$. The value $D_k = \infty$ indicates the packet is lost on the link; $\alpha_k = \mathrm{P}[D_k < \infty]$ is the probability of successful transmission across the link. We assume the D_k are independent random variables. The delay on the path from the root 0 to a node k is $X_k = \sum_{j \succeq k} D_j$; thus, the value $X_k = \infty$ indicates that the packet was lost somewhere on the path from 0 to k.

Denote the conditional link and cumulative delay variances by $r_k = \mathrm{Var}(D_k | D_k < \infty)$ and $s_k = \mathrm{Var}(X_k | X_k < \infty)$. By the assumption of link delay independence, $s_k = \sum_{\succeq k} r_j$. We write $r = (r_k)_{k \in V}$ and call the pair $(\mathcal{T}, r)$ a **delay-variance tree**. It is called **canonical** if $r_k > 0, \forall k \in V \setminus \{0\}$. This implies that $s_i > s_j$ when $i \prec j$. Any noncanonical delay-variance tree $(\mathcal{T}, r)$ can be reduced to canonical form by removing zero variance links and identifying their endpoints. Henceforth, we assume that the underlying delay-variance tree is canonical.

C. Cumulative Delay Variance Estimation

Consider first a logical subtree of a logical multicast tree $\mathcal{T}$ formed by the root 0 and a nonleaf node k with two descendants 1 and 2 that are leaf nodes; see Fig. 1 (center). We assume initially that all delays are finite $\mathrm{P}[D_k = \infty] = 0$. Then

$$\begin{aligned}\mathrm{Cov}(X_1, X_2) &= \mathrm{Cov}\left(X_k + (X_1 - X_k), X_k + (X_2 - X_k)\right)\\ &= \mathrm{Cov}(X_1 - X_k, X_k) + \mathrm{Cov}(X_2 - X_k, X_k)\\ &\quad + \mathrm{Cov}(X_1 - X_k, X_2 - X_k) + \mathrm{Var}(X_k)\\ &= \mathrm{Var}(X_k) \end{aligned} \tag{3}$$

since, by assumption of mutual independence of the link delays D_k, the random variables X_k, $X_1 - X_k$, and $X_2 - X_k$ are mutually independent. Hence, any unbiased estimator of $\mathrm{Cov}(X_1, X_2)$ is also an unbiased estimator of $\mathrm{Var}(X_k)$. Let $X_1^{(i)}$, $X_2^{(i)}$, $i = 1, 2 \ldots n$, be measured end-to-end delays between the root 0 and leaf nodes 1 and 2, respectively. Abbreviate $\mathrm{Cov}(X_j, X_k)$ by s_{jk} and write s_{kk} as s_k. We estimate s_k by the unbiased estimator of s_{12}, namely $\widehat{s}_{12}$ where

$$\widehat{s}_{ij} = \frac{1}{n-1}\left(\sum_{m=1}^{n} X_i^{(m)} X_j^{(m)} - \frac{1}{n}\sum_{m,m'=1}^{n} X_i^{(m)} X_j^{(m')}\right). \tag{4}$$

D. Link Delay-Variance Estimation

By the independence assumption on the link delays $r_k = s_k - s_{f(k)}$. Thus, any family of (unbiased) estimators $(\widehat{s}_k)_{k\in V}$ of the s_k yields unbiased estimators $\widehat{r}_k = \widehat{s}_k - \widehat{s}_{f(k)}$. Generalizations for estimating higher order joint moments are given in [12].

III. Delay-Variance Estimation on General Trees

A. Unbiased Delay-Variance Estimators

In a general tree, let $Q(k) = \{\{i, j \subset R | i \vee j = k, \}$ be the set of distinct pairs of leaf nodes whose $\prec$ least-common ancestor is k. Any convex combination $\sum_{\{i,j\}\in Q(k)} \mu_{ij}\widehat{s}_{ij}$ (i.e., with the $\mu_{ij} \geq 0$ and summing to 1) is also an unbiased estimator of s_k. An example is the **uniform estimator**

$$\frac{1}{\#Q(k)}\sum_{\{i,j\}\in Q(k)} \widehat{s}_{ij}. \tag{5}$$

A disadvantage with the uniform estimator is that high variance of one of the summands may lead to high estimator variance overall. This motivates choosing coefficients μ_{ij} in order to reduce variance. In this section, we assume all delays to be finite with bounded fourth moments. Later we shall relax the finiteness assumption.

We formalize the notion of (possibly random) convex combinations of $\widehat{s}_{ij}$ as a **covariance aggregator**. This is a sequence $\mu = (\mu(n))_{n\in\mathbb{N}}$ (here n labels the number of probes) of random weights $\{\mu_{ij}(n) : \{i,j\} \in Q(k); k \in V \setminus R\}$ with $0 \leq \mu_{ij}(n) \leq 1$ and $\sum_{\{i;j\}\in Q(k)} \mu_{ij}(n) = 1$ for each $k \in V \setminus R$, and with the property that each $\mu(n)$ is a function of the end-to-end delays $(X_k)_{k\in R}$ of the first n probes.

Let $\widehat{s} = \{\widehat{s}_{ij}(n) : \{i,j\} \in Q(k); k \in V \setminus R\}$ be a family of estimators ($\widehat{s}_{ij}(n)$ estimates s_{ij}), each $\widehat{s}_{ij}(n)$ being a function of the end to end delays $(X_k)_{k\in R}$ of the first n probes. Given a covariance aggregator μ, we can estimate $\mathrm{Var}(X_k)$ by

$$V_k(\mu, \widehat{s}) = \sum_{\{i,j\}\in Q(k)} \mu_{ij}\widehat{s}_{ij}. \tag{6}$$

A covariance aggregator is called **deterministic** if it does not depend on the $X^{(i)}$. We denote the set of such aggregators with indices in $Q(k)$ by $\mathcal{D}_k$. An example is the **uniform** aggregator that was used in the uniform estimator (5): $\mu_{ij} = (\#Q(k))^{-1}$.

B. Minimum Variance Estimation of Cumulative and Link Delays

Define the covariances $C_{(ij),(\ell m)} = \mathrm{Cov}(Z_i Z_j, Z_\ell Z_m)$, where $Z_i = X_i - E[X_i]$. We will use $C_{(k)} = [C_{(ij),(\ell m)}]_{\{i,j\},\{\ell,m\}\in Q(k)}$ to denote the matrix obtained by letting the indices (ij) and (ℓm) run over $Q(k)$.

In the next theorem we characterize the asymptotic distribution of the $\widehat{s}_{ij}$ as $n \to \infty$, and give a form for the estimator $V_k(\mu, \widehat{s})$ of cumulative variance that has minimum variance.

Theorem 1: (i) For each $k \in V \setminus R$, the random variables $\{\sqrt{n}(\widehat{s}_{ij} - s_k) | \{i,j\} \in Q(k)\}$ converge in distribution as $n \to \infty$ to a multivariate Gaussian random variable with mean 0 and covariance matrix $C(k)$. The $\widehat{s}_{ij}$ are consistent estimators of s_k, as is $V(\mu, \widehat{s})$. For a deterministic covariance aggregator $\mu \in \mathcal{D}_k$, $\sqrt{n}(V_k(\mu, \widehat{s}) - s_k)$ converges in distribution as $n \to \infty$ to a Gaussian random variable of mean 0 and variance $\mu \cdot C(k) \cdot \mu$.

(ii) The minimal asymptotic variance $\inf_{\mu\in\mathcal{D}_k} \mu \cdot C(k) \cdot \mu$ is achieved when

$$\mu_{ij} = \mu^*_{ij}\left(C(k)\right) := \frac{\left(C(k)^{-1} \cdot 1\right)_{(ij)}}{1 \cdot C(k)^{-1} \cdot 1} \tag{7}$$

where $C(k)^{-1}$ denotes the inverse matrix of $C(k)$ and $\mathbf{1}_{(ij)} = 1, \{i,j\} \in Q(k)$. The corresponding asymptotic variance of the variance estimator is $(\mathbf{1} \cdot C(k)^{-1} \cdot \mathbf{1})^{-1}$.

Operationally, the coefficients μ_{ij} of the minimum variance estimator of Theorem 1 are to be calculated from an *estimate* of the covariance matrix $C(k)$. Let $Z_i^{(m)} = X_i^{(m)} - (1/n)\sum_{m=1}^{n} X_i^{(m)}$. Let $\widehat{C}(k)$ denote the empirical covariance matrix with entries

$$\begin{aligned}\widehat{C}(k)_{(ij),(i'j')} = \frac{n^2}{(n-1)^3}\Bigg(&\sum_{m=1}^{n} Z_i^{(m)} Z_j^{(m)} Z_{i'}^{(m)} Z_{j'}^{(m)}\\ &- \frac{1}{n}\sum_{m=1}^{n} Z_i^{(m)} Z_j^{(m)} \sum_{m=1}^{n} Z_{i'}^{(m)} Z_{j'}^{(m)}\Bigg). \end{aligned} \tag{8}$$

$\widehat{C}(k)$ is an unbiased estimator of $C(k)$. Estimating $\mu^*(C(k))$ by $\mu^*(\widehat{C}(k))$ and s_k by $V_k(\mu^*(\widehat{C}), \widehat{s})$ potentially introduces bias and increases variance in the estimation of the s_k. However, the following Theorem shows that $\mu^*(\widehat{C}(k))$ is consistent and has the same asymptotic variance as $V_k(\mu^*(C), \widehat{s})$.

Theorem 2: $V_k(\mu^*(\widehat{C}(k)), \widehat{s})$ is a consistent estimator of s_k. $\sqrt{n}(V_k(\mu^*(\widehat{C}(k)), \widehat{s}) - s_k)$ converges in distribution to a Gaussian random variable of mean zero and variance $(\mathbf{1} \cdot C(k)^{-1} \cdot \mathbf{1})^{-1}$.

Given a pair $\mu = (\mu(k), \mu(f(k))) \in \mathcal{D}_k \times \mathcal{D}_{f(k)}$ of deterministic covariance aggregators with indices in $Q(k)$ and $Q(f(k))$, respectively, form an unbiased estimate of r_k as

$$W_k(\mu, \widehat{s}) := V_k\left(\mu(k), \widehat{s}\right) - V_{f(k)}\left(\mu\left(f(k)\right), \widehat{s}\right). \qquad (9)$$

Let $C'(k)$ denote the $\#Q(k) + \#Q(f(k))$-dimensional matrix written in block form as

$$C' = \begin{pmatrix} C(k) & C\left(k, f(k)\right) \\ C\left(k, f(k)\right)^T & C\left(f(k)\right) \end{pmatrix} \qquad (10)$$

where $C(k, f(k))$ is the $\#Q(k) \times \#Q(f(k))$ matrix of covariances $[C_{(ij),(\ell m)}]_{(ij) \in Q(k), (\ell m) \in Q(f(k))}$. Then statements analogous to Theorem 1(ii) follow straightforwardly, using parallel arguments. We state without proof the following.

Theorem 3: (*i*) For each deterministic covariance aggregator $\mu = (\mu(k), \mu(f(k))) \in \mathcal{D}_k \times \mathcal{D}_{f(k)}$, $\sqrt{n}(W_k(\mu, \widehat{s}) - r_k)$ converges to a Gaussian random variable of mean 0 and variance $\mu \cdot C'(k)^{-1}\mu$.

(*ii*) The minimal asymptotic variance of deterministic aggregators $\inf_{\mu \in \mathcal{D}_k \times \mathcal{D}_{f(k)}} \mu \cdot C'(k) \cdot \mu$ is achieved when

$$\begin{pmatrix} \mu(k) \\ \mu\left(f(k)\right) \end{pmatrix} = \left(c_1 c_2 - c_3^2\right)^{-1} \left(C'\right)^{-1}(k) \begin{pmatrix} (c_2 + c_3)\mathbf{1}_k \\ -(c_1 + c_3)\mathbf{1}_{f(k)} \end{pmatrix} \qquad (11)$$

and takes the value $(c_1 + c_2 + 2c_3)/(c_1 c_2 - c_3^2)$ where $c_1 = \mathbf{1}_k \cdot C(k)^{-1} \cdot \mathbf{1}_k$, $c_2 = \mathbf{1}_{f(k)} \cdot C(f(k))^{-1} \cdot \mathbf{1}_{f(k)}$, and $c_3 = \mathbf{1}_{f(k)} \cdot C(k, f(k))^{-1} \cdot \mathbf{1}_k$. Here, the subscripts on $\mathbf{1}_k$, $\mathbf{1}_{f(k)}$ distinguish the subspaces in which these vectors live.

C. Impact of Loss on Estimator Variance

Although lost packets do not yield delay samples at receivers descended from a link where loss occurred, the foregoing still applies to estimation of the delay variance based on received packets. For nodes $U \subset V$, define $I_n(U)$ as those packets $\{1, \ldots, n\}$ that reach all nodes in U; the number of such packets is $N_n(U) = \#I_n(U)$. The probability of a packet reaching all nodes in $U \subset V$ is $B(U) = \prod_{\{j \succeq u | u \in U\}} \alpha_j$, where α_j is the probability of successful transmission over link j. Clearly $n^{-1}N_n(U)$ converges almost surely to $B(U)$ as $n \to \infty$.

We can adapt the foregoing approach using an estimator $\widehat{u}_{ij}$ of the variance of the cumulative delay of packets reaching k, analogous to $\widehat{s}_{ij}$, by using only those packets in $I_n(R(k))$. In the notation of (4), this amounts to the replacements $n \mapsto N(R(k))$ and $\sum_{m=1}^{n} \mapsto \sum_{m \in I_n(R(k))}$. It is straightforward to show that all statements of Theorems 1 and 2 hold under the replacements: $\widehat{s} \mapsto \widehat{u}$, $C(k) \mapsto C(k)/B(R(k))$, and $n \mapsto N_n(R(k))$ and $\sum_{m=1}^{n} \mapsto \sum_{m \in I_n(R(k))}$ in (8). Thus, when sampling only probes received at all leaves descended from k, the minimal variance estimator of s_k is $V_k(\mu^*(C(k)/B(R(k))))$, convergence being slowed relative to the lossless case, with convergence rates multiplied by $B(R(k)) < 1$.

However, this approach does not scale well as the topology grows. Assuming link loss rates to be bounded away from zero, the proportion of packets reaching all receivers in a tree, namely $B(R)/n$, decays geometrically fast in the number of links in the tree. An alternative that wastes less data, and hence reduces estimator variance, is to use all packets received at i and j, i.e., in the $I_n(\{i,j\})$, not just those in $I_n(R(k))$. Define

$$\widehat{v}_{ij} = \frac{1}{N_n\left(\{i,j\}\right) - 1} \times \left(\sum_m X_i^{(m)} X_j^{(m)} - \frac{1}{N_n(\{i,j\})} \sum_{m,m'} X_i^{(m)} X_j^{(m')} \right) \qquad (12)$$

where the sums m and m' run over $I_n(\{i,j\})$. $\widehat{v}_{ij}$ is an unbiased estimate on s_{ij}. The asymptotic variance of these estimators is given in the following.

Theorem 4: (*i*) For each $k \in V \setminus R$, the random variables $\{\sqrt{n}\ (\widehat{v}_{ij} - s_k) | \{i,j\} \in Q(k)\}$ converge in distribution as $n \to \infty$ to a multivariate Gaussian random variable with mean 0 and covariance matrix $G(k)_{(ij),(\ell m)} = C(k)_{(ij),(\ell m)} B(i,j,\ell,m)/(B(i,j)B(\ell,m))$. Hence, the $\widehat{v}_{ij}$ are consistent estimators of s_k and so is $V_k(\mu, \widehat{v})$ for any deterministic covariance aggregator μ. For any deterministic covariance aggregator μ, $\sqrt{n}(V_k(\mu, \widehat{v}) - s_k)$ converges in distribution as $n \to \infty$ to a Gaussian random variable of mean zero and variance $\mu \cdot G(k) \cdot \mu$.

(*ii*) The minimal asymptotic variance $\inf_{\mu \in \mathcal{D}_k} \mu \cdot G(k) \cdot \mu$ is achieved when $\mu = \mu^*(G)$; the corresponding minimal asymptotic variance is $(\mathbf{1} \cdot G(k)^{-1} \cdot \mathbf{1})^{-1}$.

D. Effect of Model Violation

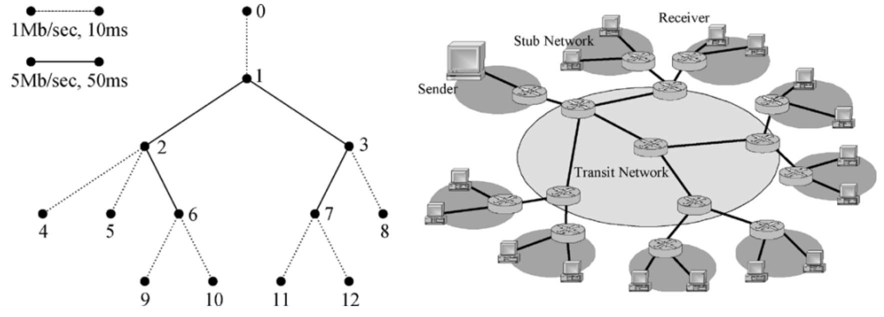
The fundamental assumption underlying our analysis is the independence of probes delays. In practice, network delays display both spatial dependence (i.e., dependence between delays on different links) and temporal dependence among successive probes on the same link which violates the independence assumption.

With spatial correlation, the covariance among delays on different links is nonzero. From (3), we then have that in general $\mathrm{Cov}(X_1, X_2) \neq \mathrm{Var}(X_k)$. Estimation of $\mathrm{Var}(X_k)$ via unbiased estimators of $\mathrm{Cov}(X_1, X_2)$ is thus biased, the amount of bias depending on the degree of correlations among delays. We expect small correlations in large networks due to traffic and link diversity. In this case, we can approximate $\mathrm{Var}(X_k) \approx \mathrm{Cov}(X_1, X_2)$ so that estimation via (6) should still yield reasonable results.

With temporal correlation, delay between consecutive probes is not independent. This is to be expected as consecutive probes are expected to experience a similar level of congestion in a link unless large interarrival are used. This poses no problem in presence of short-term correlation, since the estimator still converges provided the underlying process is still stationary and ergodic. The price of correlation, however, is that the convergence rate is slower than when the delays are independent.

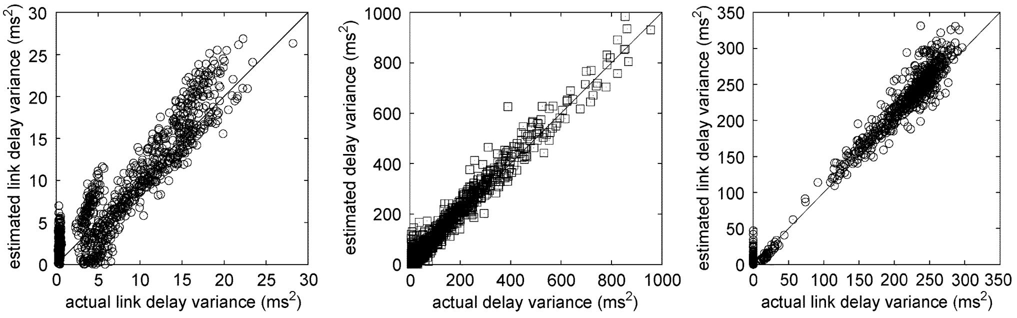
E. Inference Accuracy

We investigated the accuracy of the delay-variance estimators through simulations. We conducted model simulations using pseudorandom link delays conforming to the independence assumptions. The delay-variance estimators converged to their true values at the rates predicted by the results of this section. Details of these simulations can be found in [12]. Here we



(a) (b)

Fig. 2. Topologies used in simulations. (a) Small tree network comprising 12 nodes. (b) Diagram of the large network comprising 156 nodes.



(a) (b) (c)

Fig. 3. Simulation results. Scatter plots for link delay inference. Small network: (a) nodes have a four–packet buffer and (b) a 20-packet buffer. (c) Large network.

also show results from network-level simulations using ns.[5] These allowed us to simulate probe and background traffic at the packet level, with packet delay and loss occurring through queueing and buffer overflow. In what follows, unless otherwise stated, we focus on the behavior of the minimum variance estimator.

We first considered small-scale network simulations using the topology shown in Fig. 2(a). We arranged for some heterogeneity with the interior of the tree having higher capacity (5 Mb/s) and latency (50 ms) than at the edge (1 Mb/s and 10 ms). Each node had a finite buffer capacity; packet losses were due to drops for the tail of the buffer. We used buffer capacities of 4 and 20 packets in two different sets of experiments. The cross traffic comprised 66 FTP sessions over TCP and 29 UDP traffic sources following an exponential ON–OFF model; there were, on average, around eight background traffic sources per link. In each simulation, we use the source-to-leaf delays of probes as data to infer delay variance per internal link by and from the source to a given internal node. Since the simulations exhibit packet loss, the inference was performed using the algorithms described in Section III-C.

Fig. 3 shows scatter plots of 1200 pairs of (inferred, actual) link delay variance, based on 1000 probes, with a buffer capacity of four packets [in Fig. 3(a)] and with a buffer capacity of 20 packets [in Fig. 3(b)].

Comparing the plots, we see that inference is more accurate for the simulated network with larger buffer capacities, particularly for small delay variances. We attribute the bias of inference to departures of the delay process from the independence assumption of the model. We calculated the off-diagonal elements of the correlation matrix of the actual link delays. For a buffer of size 4, the mean value was 0.071. For a buffer of size 20, the mean was 0.021. Thus correlations were more pronounced for the smaller buffer size, leading to greater inference inaccuracy.

In order to quantify the accuracy of inference, we define a metric for evaluating estimator accuracy. If w and $\widehat{w}$ are the actual and inferred delay variances (either cumulative to a link or at the link itself), we form their error factor $F(\widehat{w}, w) = \max\{\widehat{w}/w, w/\widehat{w}\}$. For example, if $\widehat{w}$ is either twice or half w, their error factor is 2. As a robust summary statistic to capture the center of the distribution of error factors, we use the two-sided quartile-weighted median (QWM) $(Q_{.25} + 2Q_{.5} + Q_{.75})/4$, where Q_p denotes the pth quantile of a given set of error factors.

With the minimum variance estimator, the estimated and actual delay variance differed by QWM of the error factors of about 1.3 for small correlations (buffer size = 20), rising to about a factor 2 for larger correlations (buffer size = 4). We found no great advantage in increasing the number of probes to 10 000 since bias becomes a larger part of the errors.

In order to investigate the dependence of estimator performance upon the underlying topology, we simulated larger topologies generated by the gt-itm topology generator.[6] We conducted experiments across the multicast logical tree spanning a source and a set of receivers. Here we show the results for a hierarchical transit-stub network where 24 stub networks are connected via a 12-node transit network (the topology can be found in [19, Fig. 17]). The entire network comprises 156

[5] ns: Network Simulator. [Online]. Available: http://www-mash.cs.berkeley.edu/ns/ns.html

[6] GT-ITM: Georgia Tech Internetwork Topology Models. [Online]. Available: http://www.cc.gatech.edu/fac/Ellen.Zegura/graphs.html

nodes. Links between transit nodes have 50-Mb/s capacity and 10-ms propagation delay; the other links have a 10-Mb/s capacity and 5-ms delay. Each link buffer can hold 100 packets.

We selected one source and 38 receivers for the multicast measurements. The logical multicast tree spanning the source and the receivers comprises 62 nodes [see Fig. 2(a) for a sketch of the logical topology]. We note that, in this case, each logical link encompasses one or more physical links. The number of hops between the source and a receiver ranges between 5–11. The source generates probes as a 20-Kb/s stream of 40-byte UDP packets according to a Poisson process and mean interarrival time of 16 ms; in the worst case, this represents 0.2% of the link capacity. Background traffic comprises 1276 TCP sessions and 48 exponential ON–OFF UDP sources. Averaged over 100 simulations, link variance ranged from as little as 0.01 ms^2 for some backbone links to 300 ms^2 for some receiver links.

For this topology, Fig. 3(c) shows the inferred to the actual link delay variance based on 1000 probes. The QWM of the error factors is 2.09. Inference is more accurate for link variances with values larger than 1 ms^2. For these values the QWM of error factors is only 1.05. For the smaller variances, results are less accurate; for these values QWM of the error factors grows to 3.53. These large errors are essentially due to: 1) the larger topology which negatively affect the estimator variance and 2) wide link delay-variance spread (remember that link-variance estimation is carried out by difference of cumulative link-delay variances). Nevertheless, estimation errors do not impair identification of those links with largest delay variance.

IV. Computation for Large Topologies

Computation of a general estimator of s_k or the form $V_k(\mu, \widehat{s})$ requires computation of $\#Q(k)$ covariances $\widehat{s}_{ij}$. Computation of the minimum variance estimator $V_k(\mu^*, \widehat{s})$ further requires inversion of the $\#Q(k)$-dimensional matrix $\widehat{C}$. Growth of dimensionality with larger topologies is rapid and the computational cost may be prohibitive. For example, in a perfectly balanced tree of depth m and branching ratio r, the number of covariances calculated in estimating all of the s_k's grows proportionately to r^{mr} for large m. This motivates the use of estimates for the s_k, which, although potentially suboptimal in their variance, are less computationally intensive. We now describe a class of estimators that achieve this by taking advantage of the tree structure.

A. Capitalizing on the Tree Structure

For $i \prec k$, let $d(k,i) = \{j \in d(k) | i \preceq j\}$, i.e., the unique child j of k that is an ancestor of (or equal to) i. A covariance aggregator is called **local** if it has the following form:

$$\mu_{ij} = \phi_{d(i\vee j,i)d(i\vee j,i)} \left(\psi_i \psi_{f(i)} \cdots \psi_{d(i\vee j,i)}\right) \times \left(\psi_j \psi_{f(j)} \cdots \psi_{d(i\vee j,j)}\right) \quad (13)$$

where ϕ, ψ are two families of (possibly random) elements of [0, 1] with the following properties:

$$\{\phi_{ij} | \{i,j\} \subset d(k); k \in V \setminus R\}, \text{ with } \sum_{\{i,j\}\subset d(k)} \phi_{ij} = 1 \quad (14)$$

along with ϕ_{ij} depending on the first n delays $\{X_k : k \in R(i \vee j)\}$ and

$$\{\psi_j | j \in V\} \text{ with } \sum_{j \in d(k)} \psi_j = 1, \quad \forall k \in V \setminus R \quad (15)$$

and ψ_j depending on the first n delays $\{X_k : k \in R(f(j))\}$.

The significance of this form becomes apparent after we define node-averaged delays recursively through $Y_k = \sum_{j\in d(k)} Y_j \psi_j$ with $Y_k = X_k$, $k \in R$; each Y_k is an average of the end-to-end delays seen at the receivers in $R(k)$. We associate estimators $\widehat{w}_{ij}$ of $s_{i\vee j}$ through $\widehat{w}_{ij} = (1/N_n(R(k)))(\sum_{m\in I_n(R(k))} Y_i^{(m)} Y_j^{(m)}$ $-(1/N_n(R(k))) \sum_{m\in I_n(R(k))} Y_i^{(m)} \sum_{m\in I_n(R(k))} Y_j^{(m)})$.

If we now use a convex combination of $\widehat{w}_{ij}$ [instead of $\widehat{v}_{ij}$ in (6)], we obtain

$$V_k(\mu, \widehat{w}) = \sum_{\{i,j\}\in Q(k)} \mu_{ij} \widehat{w}_{ij} = \sum_{\{i,j\}\subset d(k)} \phi_{ij} \widehat{w}_{ij}. \quad (16)$$

Observe the reduced number of covariances to be calculated in the right-hand side of (16). Using a local covariance aggregator to combine the $\widehat{w}_{ij}$ allows us to take advantage of the inherent recursive structure of the tree. For the perfectly balanced tree of depth m and branching ratio r, the number of covariances to be calculated to estimate all s_k grows as r^m, compared with r^{mr} in the general case.

B. Minimal Variance Estimators on Binary Trees

An example of a local aggregator is the **uniform local aggregator** in which it averages uniformly across siblings with $\psi_i = 1/\#d(f(i))$ and $\phi_{ij} = 2/(\#d(k)(\#d(k)-1))$. However, it is natural to optimize the variance over all local aggregators. Since $\mathrm{Var}(V_k(\mu, \widehat{w})) \geq \mathrm{Var}(V_k(\mu, \widehat{v}))$, such an estimator may not be optimal over the set of all covariance aggregators; put another way, μ^* in (7) may not be local. However, we show now that μ^* *is* local for binary trees. This result appears restrictive at first, since not all multicast trees are binary. However, any tree can be extended to a binary tree by the insertion of links with zero delay variance. Since $\mathrm{Var}(\mu^*, \widehat{w})$ is consistent, the estimated delay variance for these links converges to 0 as $n \to \infty$: these can then be removed at the end of the calculation. We use this approach when we address topology inference in Section V.

Let S_k denote the $\#R(k)$-dimensional matrix with entries $s_{i\vee j}$, and U_k denote the $R(k)$-dimensional matrix with all entries equal 1. In a binary tree, let i^* be the unique sibling of a node i. The following theorem (the proof of which can be found in [13] and [32]) holds.

Theorem 5: $\mu^*(C(k))$ is local in a binary tree, with $\phi = 1$, as follows:

$$\psi_i = \frac{\delta(i^*)}{\delta(i) + \delta(i^*)} \quad \text{where} \quad \delta(i) = \det\left(S_i - s_{f(i)} U_i\right). \quad (17)$$

$V_k(\mu^*(C(k)), \widehat{w})$ has asymptotic variance $\det(C)/ \sum_{\{i,j\}\in Q(k)} \eta_i \eta_j$ where $\eta_i = \prod_{r=0}^{q_i-1} \delta(f^r(i)^*)$.

V. Topology Inference Through Delay-Variance Estimation

In this section, we adapt the foregoing work to infer the underlying tree $\mathcal{T}$ when it is not known in advance. The key observation is that $s_j > s_k$ when j is a descendent node of k. Consider a binary tree. By the assumption of independent link delays, $s_j = s_k + \sum_{k \prec i \preceq j} r_i > s_k$. Thus, the cumulative delay $s_{\ell \vee \ell'}$ is maximized when receivers ℓ and ℓ' are siblings. If not, then one of the receivers would have a sibling and the cumulative delay from the root to their ancestor would be greater. Since $s_{\ell \vee \ell'} = s_{\ell \ell'}$, the siblings can be identified on the basis of receiver measurements alone. Substituting a composite node that represents their parent and iterating then reconstructs the binary tree. In this section, we formalize this argument and extended it to reconstruct arbitrary canonical delay variance trees.

A. Deterministic Reconstruction of Delay-Variance Trees

We show that canonical delay-variance trees with receiver set R are in a one–one correspondence with the set of receiver covariances $(s_{ij})_{i,j \in R}$. We do this by formulating an algorithm to reconstruct the former from the latter. The next subsection adapts the algorithm to infer the tree from measured covariances.

We start with the special case of binary trees. The Deterministic Binary Delay-Variance Tree (DBDT) Classification Algorithm is shown in Fig. 4; it works as follows. R' denotes the current set of nodes from which a pair of siblings will be chosen, initially equal to the receiver set R. We first find the pair $U = \{u, v\}$ that maximizes s_{uv}; U is identified with the pair's parent and replaces u and v in R' (line 6). Correspondingly, we adjoin a row and column for U to the matrix s (line 8). Links (U, u) and (U, v) are added to the tree, and their link variances are calculated (line 9). This process is repeated until all sibling pairs have been identified (loop at line 4). If the last parent U identified has variance $s_U = 0$, then, since the tree is canonical, it is the root. Otherwise, we adjoin the root node and link joining it to its single child (line 13). We remark that the u and v row and column of the matrix s could be deleted after line 8 since they are not used after this point.

```
1.  Input: The set of receivers $R$ and the delay covariance matrix $s = (s_{jk})_{j,k \in R}$
2.  $R' := R; V' := R'; L' = \emptyset$ ;
3.  foreach $k \in R$ { $s_k := s_{kk}$ ; }
4.  while $|R'| > 1$ do
5.      select $U = \{u, v\} \subseteq R'$ with maximal $s_{uv}$;
6.      $V' := V' \cup \{U\}; R' := (R' \setminus U) \cup \{U\}$;
7.      $s_U := s_{uv}; s_{UU} := s_{uv}$ ;
8.      foreach $k \in R'$ do $s_{Uk} := s_{uk}; s_{kU} := s_{ku}$ ; enddo
9.      foreach $k \in U$ do $L' = L' \cup \{(U, k)\}$ ; $r_k := s_k - s_U$ ; enddo
10. enddo
11. if $s_U > 0$ do
13.     $V' := V' \cup \{0\}$ ; $L' = L' \cup \{(0, R')\}$ ;
14. enddo
15. Output: binary delay-variance tree $((V', L'), r)$ ;
```

Fig. 4. Deterministic Binary Delay-Variance Tree Classification Algorithm (DBDT).

We say that the algorithm reconstructs the binary delay-variance tree $((V, L), r)$ if, given R and the $s_{uv} = r_{u \vee v}$, $u, v \in R$, it produces $((V, L), r)$ as its output. Clearly this happens if and only if, before each iteration of the while loop 4 in Fig. 4, (V', L') can be decomposed in terms of disjoint subtrees $V' = \sum_{k \in R'} V(k)$ and $L' = \sum_{k \in R'} L(k)$. These subtrees may just be trivial ones $\mathcal{T}(k) = (\{k\}, \emptyset)$ comprising a root node k. We note also that these trees cover R, i.e., $R = \cup_{k \in R'} R(k)$. These properties hold before the first while loop and hold subsequently, since each loop of a successful reconstruction amalgamates binary subtrees rooted at siblings.

Theorem 6: DBDT reconstructs any binary canonical delay-variance tree.

In a general tree, then s_{uv} is the same for any pair $\{u, v\}$ in a sibling set U and takes the value $s_{f(U)}$. This suggests an extension to DBDT to reconstruct general canonical delay-variance trees, namely in line 5 to find instead the maximal subset $U \subseteq R'$ such that, for each $u, v \in U$, $s_{uv} = \max_{jk \in R'} s_{jk}$. It can be shown that this does reconstruct in the general case. However, we adopt a slightly different approach that is better adapted to inferring the tree from measured data. We use a two-stage approach. We first apply DBDT to an arbitrary tree and observe that the effect is to reconstruct a noncanonical binary tree in which siblings may be separated by links with zero delay variance. In the second stage, we obtain the underlying general tree by pruning, i.e., removing the zero delay-variance links and identifying their endpoints. For later use, we find it useful to specify a generalization of this procedure. For each $\varepsilon > 0$, the Tree Pruning Algorithm TP(ε) acts on a delay-variance tree by pruning all links whose delay variance is less than or equal to ε. The pruning operation described above is then TP(0). We specify TP in Fig. 5.

```
Input: a delay-variance tree $(\mathcal{T}, r)$;
Parameter: a threshold $\varepsilon \geq 0$;
$V' := \{0\} \cup d_{\mathcal{T}}(0); L' := \{(0, k) : k \in d_{\mathcal{T}}(0)\}$;
$U := d_{\mathcal{T}}(0)$;
while $U \neq \emptyset$ do
    select $j \in U$;
    $U := U \setminus \{j\} \cup d_{\mathcal{T}}(j)$;
    if $(r_k \leq \varepsilon) \vee (j \neq R)$ then
        $L' := (L' \cup \{(f_{\mathcal{T}'}(j), k) : k \in d_{\mathcal{T}}(j)\}) \setminus \{(f_{\mathcal{T}'}(j), j)\}$;
        $V' := V'/\{j\} \cup d_{\mathcal{T}}(j)$;
    else
        $L' := L' \cup \{(j, k) : k \in d_{\mathcal{T}}(j)\}$;
        $V' := V' \cup d_{\mathcal{T}}(j)$;
    endif;
enddo
Output: $(\mathcal{T}', r')$
```

Fig. 5. Tree pruning algorithm TP(ε).

The algorithm reconstructs the tree if, for each node $k \in V$ having n children, there is a run of $n - 1$ while loops in DBDT that identify binary nodes $U^{(1)}, U^{(2)}, \ldots U^{(n-1)}$ that include all of the children. We call this run of binary groupings an **outer loop**. $U^{(1)}$ is a binary subset of $R^{(1)} = d(k)$, while, for $m = 2, \ldots n - 1$, each $U^{(m)}$ is a binary subset of $R^{(m)} = (R^{(m-1)} \setminus U^{(m-1)}) \cup U^{(m-1)}$. We assume that a tie-breaking rule is specified for line 4 of Fig. 4 when there is more than one maximizer. For example, select the maximizing pair $\{u, v\}$ for which u most recently included in V' and v the next most recently included, using an arbitrary initial order for R. In this case, $d(k)$ can be written as $\{u_1, \ldots, u_n\}$ with $U^{(1)} = \{u_1, u_2\}$ and $U^{(m)} = \{U^{(m-1)}, u_{m+1}\}$ for $m = 2, \ldots n - 1$. The outer loop produces the subtree shown in Fig. 6.

Theorem 7: The **Deterministic Delay-Variance Tree** algorithm DDT $=$ TP$(0) \circ$ DBDT reconstructs any canonical delay-variance tree $((V, L), r)$.

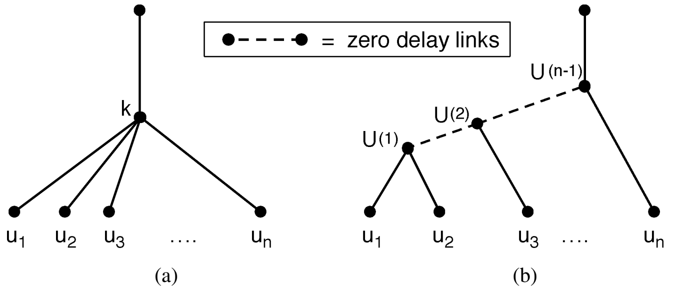


Fig. 6. (a) General node with n children. (b) Example of corresponding binary tree with zero delay links.

```
Input: Receivers set $R$, number of probes $n$, receiver traces $(X_k^{(i)})_{k\in R}^{i=1,2,\dots n}$ ;
$R' := R, V' := R;\ L' = \emptyset$ ;
foreach $k \in R$ do
    $s_k := w(k,k)$;
    foreach $i = \{1,\dots,n\}$ do $Y_k^{(i)} = X_k^{(i)}$ ; enddo
enddo
while $|R'| > 1$ do
    select $\{u,v\} \subset R'$ that maximizes $s_{\{u,v\}} := w(u,v)$;
    $V' := V' \cup \{\{u,v\}\}$; $R' := (R' \setminus \{u,v\}) \cup \{\{u,v\}\}$;
    foreach $(k \in \{u,v\})$ do
        $r_k := s_{\{u,v\}} - s_k$; $L' := L' \cup \{(\{u,v\},k)\}$ ;
        foreach $(\ell,\ell' \in R(k))$ do $S_{k,\ell\ell'} := s_{\ell\vee\ell'}$ ; enddo
        $\delta(k) := \det(S_k - \widehat{s}_{\{u,v\}}U_k)$ ;
    enddo
    foreach $(m \in \{1,\dots n\})$ do $Y_{\{u,v\}}^{(m)} := \frac{\left(\delta(u)Y_u^{(m)}+\delta(v)Y_m^{(m)}\right)}{(\delta(u)+\delta(v))}$ ; enddo
enddo
Output: delay-variance tree $((\{0\} \cup V', \{(0,R')\} \cup L'), \{0\} \cup r)$
procedure $w(i,j)$ {
    return $\frac{(\sum_{m=1}^n Y_i^{(m)}Y_j^{(m)} - n^{-1}\sum_{m=1}^n Y_i^{(m)}\sum_{m=1}^n Y_j^{(m)})}{n-1}$ }
```

Fig. 7. Binary delay-variance tree classification algorithm (BDT). The functions $\vee$ and $R(\cdot)$ return ancestors and leaf nodes, respectively, from the current (V', L'). U_k is the $\#R(k)$-dimensional matrix with all unit entries.

B. Inference of Loss Tree From Measured Delay Covariances

We present stochastic versions of the above algorithms that estimate topology based on *estimated* delay covariances. We adapt the minimum variance approach of Section III. Given a pair of nodes $\{k,\ell\}$, we can estimate $\mathrm{Cov}(X_k, X_\ell)$ by

$$V_{k,\ell}(\mu,\widehat{s}) = \sum_{\{i,j\}\in R(k)\times R(\ell)} \mu_{ij}\widehat{s}_{ij} \tag{18}$$

where $\widehat{s} = \{\widehat{s}_{ij}\}_{i,j\in R}$ and μ is a covariance aggregator. For this estimator, properties analogous to those established in Section III directly follow. In particular, $\sqrt{n}(V_{k,\ell}(\mu,\widehat{s}) - s_{k\ell})$ converges to a Gaussian random variable of mean zero and variance $\mu \cdot C(k,\ell) \cdot \mu$, where $C(k,\ell) = [C_{(ij)(i'j')}]_{\{i,j\},\{i'j'\}\in R(k)\times R(\ell)}$; moreover, the minimum variance estimator is achieved when $\mu = \mu^*(C(k,\ell)) = (C(k,\ell)^{-1}\cdot\mathbf{1})/\mathbf{1}\cdot C(k,\ell)^{-1}\cdot\mathbf{1}$. Denote by $\widehat{C}(k,\ell)$ the empirical version of $C(k,\ell)$, i.e., the covariance matrix with entries given by (8). Similar to Theorem 2 we have the following theorem.

Theorem 8: $V_{k,\ell}(\mu^*(\widehat{C}(k,\ell)),\widehat{s})$ is a consistent estimator of s_{kl}. $\sqrt{n}(V_{k,\ell}(\mu^*(\widehat{C}(k,\ell)),\widehat{s}) - s_{kl})$ converges in distribution to a Gaussian random variable of mean 0 and variance $(\mathbf{1}\cdot C(k,\ell)^{-1}\cdot\mathbf{1})^{-1}$.

Inference of Binary Trees From Measurements. Inference of binary trees from measured receiver delays is performed by the Binary Delay-Variance Tree Classification Algorithm (BDT); see Fig. 7. This combines DBDT with the minimum variance estimator from (18), taking advantage of the tree structure and the optimality of the local aggregator for binary trees. In distinction with DBDT, we exclude the test to see if the last U identified is the root, since the event $s_U = 0$ happens with probability zero for continuous delay distributions.

In the following, we will use the notation $(\widehat{\mathcal{T}}, \widehat{r})$ to denote an inferred delay-variance tree; sometimes we will use $\widehat{\mathcal{T}}_X$ to distinguish the topology inferred by a particular algorithm X. P_X^f will denote the probability of false identification of topology $\mathcal{T}$ of the delay-variance tree $(\mathcal{T}, r)$, i.e., $P_X^f = \mathrm{P}_{\mathcal{T},r}[\widehat{\mathcal{T}}_X \neq \mathcal{T}]$.

Theorem 9: Let $(\mathcal{T}, r)$ be a binary canonical delay variance tree. $\lim_{n\to\infty} P_{\mathrm{BDT}}^f = 0$.

Inference of General Trees From Measurements. The adaptation of DDT to the classification of general loss trees is more complicated than the binary case. In DDT, s_{jk} takes the same value for any two nodes $\{j,k\}$ in a sibling set, giving rise to zero loss links between the nodes grouped in an outer loop, which are then pruned by TP(0). However, using a measured delay, the corresponding estimates will not be equal for finitely many probes. In order to group nodes appropriately, we apply a threshold $\varepsilon > 0$ while pruning, so that links are pruned if the estimated link delay variance does not exceed ε. For each $\varepsilon > 0$, the **Delay-Variance Tree Classification Algorithm** is $\mathrm{DT}(\varepsilon) = \mathrm{TP}(\varepsilon)\circ\mathrm{BDT}$. Since link delay-variance estimates become accurate as the number of probes grows to infinity, all links with delay variance greater that ε should be correctly classified. The proof of the following is similar to that of Theorem 9.

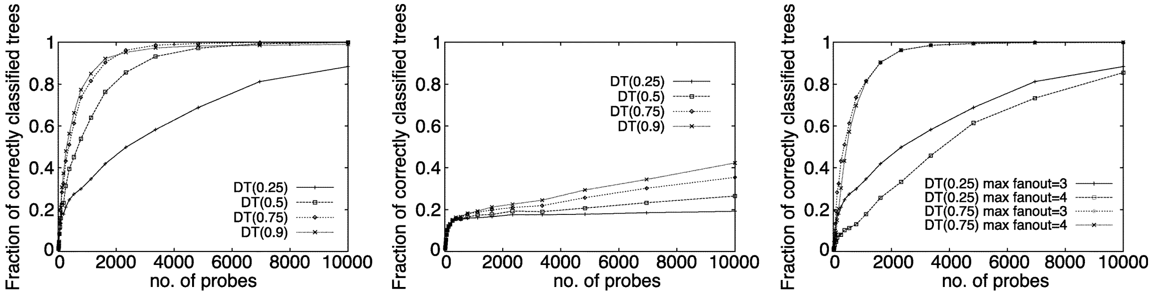
Theorem 10: Let $(\mathcal{T}, r)$ be a canonical delay-variance tree in which all link variances $r_k > \varepsilon'$ for some $\varepsilon' > 0$. For each $\varepsilon \in (0, \varepsilon')$, $\lim_{n\to\infty} P_{\mathrm{DT}(\varepsilon)}^f = 0$.

Convergence to the true topology requires ε to be smaller than the internal links delay variance, which are typically not known in advance. A very small value of ε is more likely to satisfy the above condition but at the cost, as shown in Section VI, of slower classifier convergence. A large value of ε, on the other hand, is more likely to result in systematically removing links with small delay variance.

In practice, we believe that the choice of ε does not pose a problem: for many applications, while it is important to correctly identify links with a high loss rate, failure to detect links with small loss rates would be acceptable. In this case, it could be sufficient if the convergence of the inferred topology to $\mathcal{T}^\varepsilon = \mathrm{TP}(\varepsilon)(\mathcal{T})$ was obtained from $\mathcal{T}$ by ignoring links whose loss rates fell below some specific value ε, which would be regarded as some application-specific minimum delay variance of interest.

The results below establish the desired convergence to $\mathcal{T}^\varepsilon$ for any $\varepsilon \geq 0$ provided $\varepsilon \neq r_k$, $k \in V \setminus R$. The key observation is that, since the deterministic versions of the algorithm reconstruct $\mathcal{T}^\varepsilon$, so does BDT, as the number of probes grows. Denote $P_{\mathrm{BDT}}^f(\varepsilon) = \mathrm{P}_{\mathcal{T},r}[\widehat{\mathcal{T}}_{\mathrm{BDT}} \neq \mathcal{T}^\varepsilon]$. Without further proof we have, we have the following theorem.

Theorem 11: Let $(\mathcal{T}, r)$ be a canonical delay-variance tree. For each $\varepsilon \geq 0$, such that $\varepsilon \neq r_k$, $k \in V \setminus R$, $\lim_{n\to\infty} P_{\mathrm{BDT}}^f(\varepsilon) = 0$.



(a) (b) (c)

Fig. 8. Dependence of the accuracy on the threshold ε and the topology. A fraction of trees correctly classified by DT(ε) in 1000 simulations over randomly generated 15 nodes tree: link variance is uniformly distributed in the interval (a) [1,10] and (b) [1,100]. (c) Different maximum fanout.

C. Effect of Spatial Correlation

In case of spatial correlation, the estimates $V_{k,\ell}(\mu,\widehat{s})$ are biased and do not necessarily converge to $s_{k,\ell}$. As a consequence, the results shown above do not apply in general and the behavior of the algorithm greatly depends on the amount of bias. In case of small correlation, i.e., when $V_{k,\ell}(\mu,\widehat{s}) \approx s_{k,\ell}$, we can distinguish two cases. In the case that, despite the presence of correlation, $V_{k,\ell}(\mu,\widehat{s}) > V_{k,\ell'}(\mu,\widehat{s})$ iff $s_{k,\ell} > s_{k,\ell'}$ for all k, ℓ, and ℓ' (i.e., when the ordering between covariance and their estimates is preserved), then all results still apply. Indeed, by similar arguments to that of Theorem 10, we can recover $\mathcal{T}$ (and $\mathcal{T}^{\varepsilon}$) by DT($\varepsilon$) with probability 1 as n grows to infinity. Instead, when the conditions above are not met, the inferred topology will converge to a topology $\mathcal{T}'$, which, in general, will differ from $\mathcal{T}$. ($\mathcal{T}'$ is the unique canonical delay tree with independent delays which is compatible with the end-to-end covariances $s_{k,\ell}$.) In this case, while it may not be possible to recover $\mathcal{T}$, we might still correctly recover the pruned topology $\mathcal{T}^{\varepsilon}$ as long as the condition TP(ε)($\mathcal{T}'$) $= \mathcal{T}^{\varepsilon}$ is verified. We expect this to be the case when the bias is small compared to the chosen threshold ε, which in this case also has a role of tolerance parameter.

VI. Simulation Evaluation of Topology Inference

We evaluated the accuracy of the classification algorithms through simulation. We first performed model-based simulations, where link delays are independent distributed random variables. We used exponential distributed delays and, unless otherwise stated, we assumed no packet loss. Then we investigated the behavior of the estimator via ns simulations. The ns simulations allow us to investigate the performance of the classification algorithms in a more realistic setting in which the model assumptions can be violated.

A. Model-Based Simulation

Dependence of Accuracy on Threshold ε. We conducted 1000 simulations over randomly generated trees of 15 nodes and a maximum branching ratio of 3. Link variance was randomly chosen in the interval [1,10]. Convergence of the estimated topology to the true topology is assured by choosing $\varepsilon < 1$. In Fig. 8(a), we plot the fraction of correctly classified trees for $\varepsilon = 0.25$, 0.5, 0.75, and 0.9. Except with a small numbers of probes, accuracy is best for $\varepsilon = 0.75$. Smaller values of ε result in stricter grouping criteria, and thus statistical fluctuations of the estimates lead to erroneous exclusion of nodes from groups. Increasing ε initially decreases the probability of such events, but, as ε approaches the smallest link delay variance $r_{\min}$, the probability of falsely including nodes in a group increases. When ε increases beyond $r_{\min}$, this link is effectively ignored and so the probability of correct classification drops to zero.

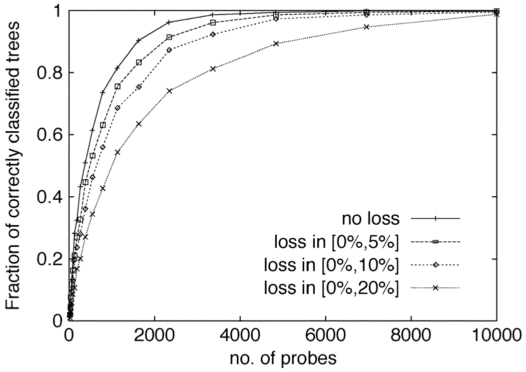


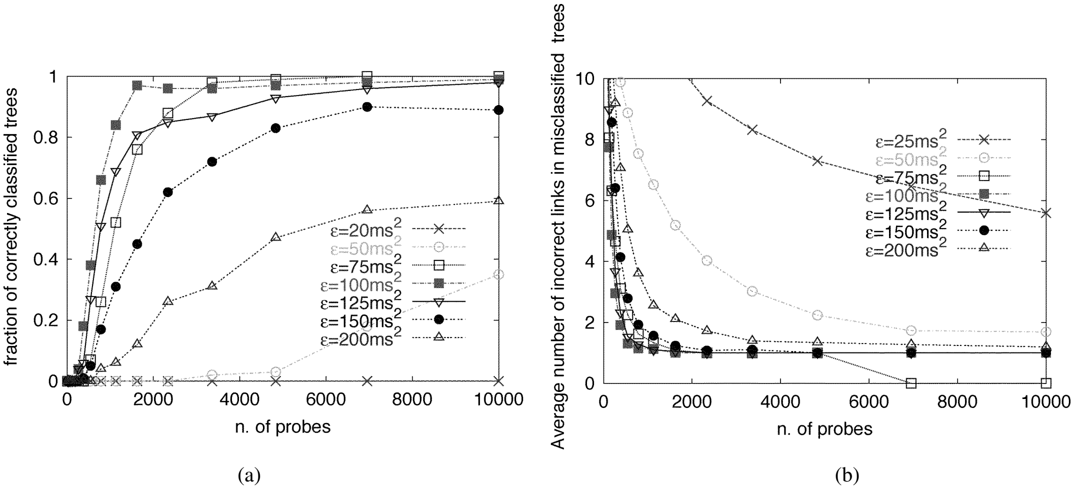
Fig. 9. Dependence of the accuracy on loss rate. A fraction of trees correctly classified by DT(ε) in 1000 simulations over a randomly generated 15 nodes for different loss-rate intervals.

Accuracy Versus Variance Spread and Topology. Accuracy decreases noticeably when the range of link variance is expanded to [1,100]; with 1000 probes and ε =0.75, only about 35% of the trees were correctly classified, see Fig. 8(b). The corresponding proportion was 100% for variances in [1,10]. This occurs because large delay variance leads to larger estimator variances, and hence mistaken pairing of nonsibling nodes or erroneous inclusion or exclusion of nodes in a group is more likely to occur. Accuracy decreases for a larger branching ratio. Fig. 8(c) compares accuracy for maximum branching ratios 3 and 4 and $\varepsilon = 0.5$, 0.75, and delay variances in [1,10]. Larger branching ratios require more pruning operations, thus affording more opportunities for misclassification.

Dependence on Loss. Packet loss increases estimator variance, and hence decreases inference accuracy; see Section III-C. This is evident in Fig. 9, which displays a fraction of correctly classified trees decreases for various ranges of randomly selected loss rates. Link variance is randomly chosen in the interval [1,10] and $\varepsilon = 0.75$.

B. ns *Simulation*

Here we report the results for the ns topology already considered in Section III-E. We conducted 100 ns simulations of the



(a)

(b)

Fig. 10. ns simulation: convergence to the pruned topology $\mathcal{T}^{\varepsilon}$. (a) A fraction of experiments in which $\mathcal{T}^{\varepsilon}$ was correctly identified for different values of ε. (b) Average number of incorrect links in the inferred topology $\widehat{\mathcal{T}}$ when $\mathcal{T}^{\varepsilon}$ was misclassified (i.e., when $\widehat{\mathcal{T}} \neq \mathcal{T}^{\varepsilon}$).

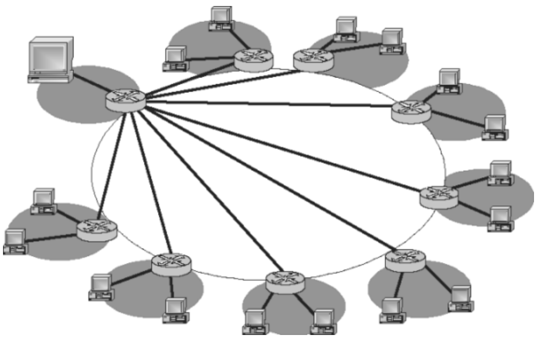
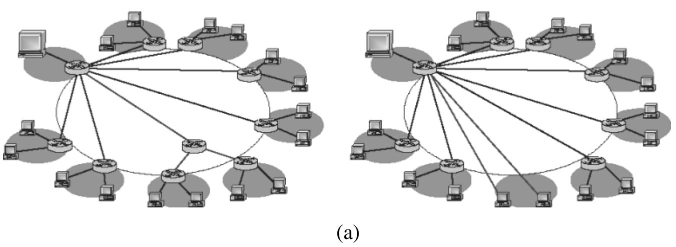


Fig. 11. ns simulation. Diagram of the logical topology $\mathcal{T}^{\varepsilon}$ obtained by ignoring links with delay variance smaller than ε (for any $\varepsilon \in [1, 200]$ (ms^2)).



(a)

Fig. 12. ns simulation. Most likely error in the reconstructed topology. (a) For smaller values of ε, failing to prune one or more links in the reconstructed topology. (b) Or, for larger ε, excessive pruning of one or more links.

classification algorithm for different values of ε. In these simulations, DT(ε) was never able to correctly classify the topology $\mathcal{T}$ for as many as 10 000 probes. We found this to be typical for large topologies of this type. The reasons are twofold. First, in a large network, link delay varies a great deal between backbone and access links. The results is a large spread in link delay variances which negatively affects the inference accuracy. On the one hand, most of the choice of threshold values ε exceeded the variance of the faster links, thus resulting in erroneous pruning. The choice of very small ε, on the other hand, always resulted in the erroneous exclusion of some node from the relative group in access networks. Second, in such a network scenario, even a small correlation has no negligible effect since estimator bias can easily become of the same order—if not even larger—than the variance of the faster links. Nevertheless, the algorithm performs quite well if we ignore failures to detect links with small delay variance. In Fig. 10(a), we plot the fraction of experiments in which the topology $\mathcal{T}^{\varepsilon}$ was correctly identified for selected values of ε in the interval [1,200] ms^2.

The topology $\mathcal{T}^{\varepsilon}$ itself is sketched in Fig. 11. (The same topology was obtained for any value ε chosen in the interval $\varepsilon \in [1, 200]$ ms^2.) We observe that the topology is obtained by collapsing all links in the backbone. In this case, correct classification corresponds to correctly identifying which receivers are in the same stub network.

Accuracy depends on the relative values of ε and the internal link delay variances (considering only links with variance larger than ε), and it is better for intermediate values. This behavior is identical to that previously observed for the model simulations. It is interesting to observe that misclassifaction was largely due to very few errors. In Fig. 10(b), we plot the average number of "incorrect" links in the reconstructed topology. Informally, these are the links which should be pruned or added to obtain $\mathcal{T}^{\varepsilon}$ from $\widehat{\mathcal{T}}$. This number thus represents a measure of how the reconstructed topology differs from the (pruned) actual one. For more than 2000 probes, we have that, but for $\varepsilon = 25$ ms^2 (a small threshold resulting in too few pruned links), the reconstructed topology differed from $\mathcal{T}^{\varepsilon}$ by one link only. We observed that, in fact, most of the time misclassification was only due to either failing to prune one link in the reconstructed topology (smaller value of ε) or excessive pruning of one link (larger ε). In the first case, receivers in the same stub network were still correctly grouped [see Fig. 12(a)]; this was not the possible in the second case for some receivers [see Fig. 12(b)].

VII. Topology Misclassification

We analyze in detail the modes of failure of DT and estimate the convergence rates for the probability of correct classification as the number of probes grows. We analyze topology misclassification by focusing on how sets of receivers can be misgrouped in the estimated topology $\widehat{\mathcal{T}}$. We formalize the notion of correct receiver grouping as follows. Let $R_{\mathcal{T}}$ denote the set of receivers in the logical multicast topology $\mathcal{T}$.

Definition 1: Let $(\mathcal{T} = (V, L), r)$ be a delay-variance tree and denote $(\widehat{\mathcal{T}} = (\widehat{V}, \widehat{L}), \widehat{r})$ an inferred delay-variance tree. The receivers $R_{\mathcal{T}}(k)$ descended from a node $k \in W$ are said to be **correctly grouped** in $\widehat{\mathcal{T}}$ if there exists a node $\widehat{k} \in \widehat{V}$ such that $R_{\mathcal{T}}(k) = R_{\widehat{\mathcal{T}}}(\widehat{k})$. In this case, we shall say also that node k is correctly classified in $\widehat{\mathcal{T}}$.

The notion of correct grouping allows the trees rooted at k and $\widehat{k}$ to be different; it only requires the sets of receivers descended from k and $\widehat{k}$ be equal. Correct receiver grouping and correct topology classification are related. In the case of binary trees, the topology is correctly classified if and only if so is every interior node. This property allows us to study topology misclassification by looking at receiver misgrouping. To this end, we consider more general convex combinations of the delay covariances than those expressed by (18) to take into account groups of nodes which may result from nodes misgrouping. For two disjoint subsets of R, S_1 and S_2, $S_1, S_2 \neq \emptyset$, set

$$V_{S_1,S_2}(\mu, \widehat{s}) := \sum_{\{i,j\} \in S_1 \times S_2} \mu_{ij} \widehat{s}_{ij} \tag{19}$$

where μ is any suitable covariance aggregator. Properties similar to those established in Section V-B hold for these convex combinations. In particular, $\sqrt{n}(V_{S_1,S_2}(\mu^*(\widehat{C}(S_1 \times S_2)), \widehat{s})$ $-V_{S_1,S_2}(\mu^*(C(S_1 \times S_2)), s)$ converges to a Gaussian random variable of mean zero and variance $(\mathbf{1} \cdot C(S_1 \times S_2)^{-1} \cdot \mathbf{1})^{-1}$, where $C(S_1 \times S_2) = [C_{(ij)(\ell m)}]_{\{i,j\},\{\ell,m\} \in S_1 \times S_2}$.

A. Misgrouping and Misclassification of Binary Trees

We start by studying misgrouping in BDT. Denote by G_k the event that BDT correctly groups nodes in $R(k)$. This happens if

$$\widehat{D}(S_1, S_2, S_3) := V_{S_1,S_2}(\mu^*, \widehat{s}) - V_{S_1,S_3}(\mu^*, \widehat{s}) > 0 \tag{20}$$

for all $(S_1, S_2, S_3) \in \mathcal{S}(k)$ where $\mathcal{S}(k) = \{S_1, S_2 \subset R(k), S_3 \subseteq R \setminus R(k), S_k \neq \emptyset, S_1 \cap S_2 = \emptyset\}$. Equation (20) ensures that, for all possible ways to reconstruct the tree, proper subsets of $R_{\mathcal{T}}(k)$ are never grouped with receivers not in $R_{\mathcal{T}}(k)$, which in turn guarantees that receivers in $R_{\mathcal{T}}(k)$ are first all grouped together. By construction, in $\widehat{\mathcal{T}}$, there is a node $\widehat{k}$ such that $R_{\mathcal{T}}(k) = R_{\widehat{\mathcal{T}}}(k)$. Let $Q(S_1, S_2, S_3)$ denote the event that (20) holds; then, $G_k \supseteq Q_k \overset{\text{def}}{=} \cap_{(S_1,S_2,S_3) \in \mathcal{S}(k)} Q(S_1, S_2, S_3)$. This provides the following upper bound for the misgrouping probability, denoted by P_k^f, as

$$P_k^f := \mathrm{P}\left[G_k^e\right] \le \sum_{(S_1,S_2,S_3) \in \mathcal{S}(k)} \mathrm{P}\left[Q^c(S_1, S_2, S_3)\right]. \tag{21}$$

Normal Approximations. We derive the asymptotic behavior of P_k^f for many probes as follows.

Theorem 12: Let $(\mathcal{T}, r)$ be a canonical delay-variance tree. For each $k \in W$, $\sqrt{n}(\widehat{D}(S_1, S_2, S_3) - D(S_1, S_2, S_3))$, $(S_1, S_2, S_3) \in \mathcal{S}(k)$, converges in distribution, as the number of probes $n \to \infty$, to a Gaussian random variable with mean 0 and variance $\sigma_D^2(S_1, S_2, S_3) = \sum_{\{i,j\},\{\ell,m\} \in S_1 \times S_2 \cup S_1 \times S_3}$ $(\partial D(S_1, S_2, S_3)/\partial s_{ij}) C_{(ij),(\ell m)} (\partial D(S_1, S_2, S_3)/\partial s_{\ell m})$, where $D(S_1, S_2, S_3) = V_{S_1,S_2}(\mu^*, s) - V_{S_1,S_3}(\mu^*, s)$. Moreover, $\inf_{(S_1,S_2,S_3) \in \mathcal{S}(k)} D(S_1, S_2, S_3) = r_k$.

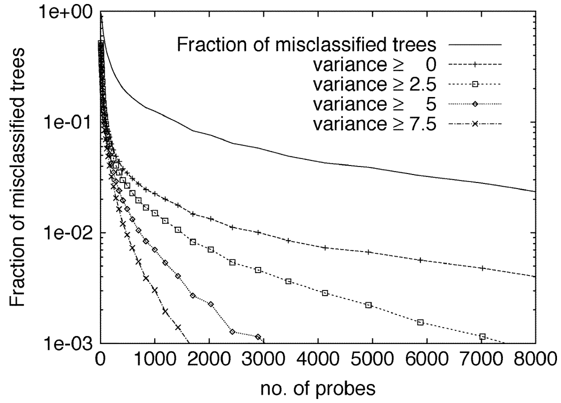


Fig. 13. Misclassification and misgrouping in BDT. Fraction of links misclassified with $\mathrm{variance} \ge \phi$, for $\phi = 0$, 2.5,5,7.5%. Link variance in [1,100].

Theorem 12 suggests that we approximate $\mathrm{P}[Q^c(S_1, S_2, S_3)] = \mathrm{P}[\widehat{D}(S_1, S_2, S_3) < 0]$ by $\Psi(-\sqrt{n} \cdot D(S_1, S_2, S_3)/\sigma_D(S_1, S_2, S_3))$, where Ψ is the cdf of a standard normal distribution. For large n, we can approximate to leading exponential order as $\mathrm{P}[Q^c(S_1, S_2, S_3)] \approx e^{-(n/2)(D(S_1,S_2,S_3)^2/\sigma_D^2(S_1,S_2,S_3))}$. Since the largest term over $\mathcal{S}(i)$ should dominate all others for large n, we have

$$\mathrm{P}_k^f \approx e^{-\left(\frac{n}{2}\right) \inf_{(S_1,S_2,S_3) \in \mathcal{S}(k)} \frac{D(S_1,S_2,S_3)^2}{\sigma_D^2(S_1,S_2,S_3)}}. \tag{22}$$

In the case of binary trees, when all groups are correctly formed, so is the topology; therefore, we have that $P_{\mathrm{BDT}}^f \le \sum_{k \in W} P_k^f \approx \max_{k \in W} P_k^f$ which suggest that $\log \mathrm{P}_{\mathrm{BDT}}^f$ versus n is asymptotically linear with slope

$$\frac{1}{2} \inf_{k \in W} \inf_{(S_1,S_2,S_3) \in \mathcal{S}(k)} \frac{D(S_1, S_2, S_3)^2}{\sigma_D^2(S_1, S_2, S_3)}. \tag{23}$$

Modes of Misclassification by BDT in Experiments. Calculation of the infimum in (23) is in general quite difficult since $\sigma_D^2(S_1, S_2, S_3)$ is a complex function of both the topology and the link variances. Here we use experience from experiments to identify the dominant modes of misclassification and misgrouping. For the binary trees used in Section VI, we plot in Fig. 13 the proportion of links that had variances greater than or equal to a given threshold ϕ and were still misclassified by BDT, along with a proportion of experiments in which BDT incorrectly identifies the topology. Observe that errors are dominated by misclassification of low variance links. This suggests that, for large n, $\mathrm{P}_{\mathrm{BDT}}^f \approx \mathrm{P}_j^f$, where $j = \arg\min_{k \in W} r_k$, i.e., the most likely way to misclassify a tree is by not correctly grouping receivers that share the link with smallest variance.

B. Misgrouping and Misclassification by DT(ε)

We now turn our attention to the errors in classifying general trees by DT(ε). In the following, without loss of generality, we will study the errors in the classification of the pruned tree $(\mathcal{T}^\varepsilon, r^\varepsilon) = \mathrm{BDT}(\varepsilon)(\mathcal{T}, r)$, under the assumption that $\varepsilon \neq r_k$, $k \in W$. W^ε will denote the set of nodes in $\mathcal{T}^\varepsilon$ terminating internal links.

Let $(\widehat{\mathcal{T}}', \widehat{r}')$ denote the tree produced by BDT. Then, the final estimate is $(\widehat{\mathcal{T}}^{\varepsilon}, \widehat{r}^{\varepsilon}) = \mathrm{TP}(\varepsilon)(\widehat{\mathcal{T}}', \widehat{r}')$. In distinction with the binary case, incorrect grouping by BDT is sufficient but not necessary for the misclassification. For $\mathrm{DT}(\varepsilon)$, incorrect classification occurs if any of the following holds:

1) at least one node in $\mathcal{T}^{\varepsilon}$ is misclassified in $\widehat{\mathcal{T}}'$;
2) $\mathrm{TP}(\varepsilon)$ prunes links from $\widehat{\mathcal{T}}'$ that are present in $\mathcal{T}^{\varepsilon}$;
3) $\mathrm{TP}(\varepsilon)$ fails to prune links from $\widehat{\mathcal{T}}'$ that are not present in $\mathcal{T}^{\varepsilon}$.

Observe that 1) implies that a node i such that $r_i \le \varepsilon$ can be misclassified and still $\widehat{\mathcal{T}}^{\varepsilon} = \mathcal{T}^{\varepsilon}$ provided that all of the resulting erroneous links are pruned.

We have already analyzed errors of type 1) in the analysis of BDT. Errors of type 2) are excluded if for all W^{ε}

$$\widehat{E}(S_1, S_2, S_3) := V_{S_1,S_2}(\mu^*, \widehat{s}) - V_{S_1 \cup S_2, S_3}(\mu^*, \widehat{s}) > \varepsilon \quad (24)$$

for all $(S_1, S_2, S_3) \in \mathcal{S}(i)$, since this condition implies that all estimated loss rates of links in the actual tree are greater than ε. Errors of type 3) are excluded if $(D(S_1, S_2, S_3) < 0 \wedge E(S_1, S_2, S_3) \ge \varepsilon) \vee (D(S_1, S_2, S_3) \ge 0 \wedge E(S_1, S_2, S_3) < \varepsilon)$ for all $(S_1, S_2, S_3) \in \mathcal{S}(\varepsilon)$ where $\mathcal{S}(\varepsilon) = \{(S_1, S_2, S_3) : S_j \subset R; S_j \ne \emptyset; S_j \cap S_k = \emptyset, j \ne k; (S_1 \cup S_2 \cup S_3) \cap R_{\mathcal{T}}(i) = \emptyset \vee (S_1 \cup S_2 \cup S_3) \subseteq R_{\mathcal{T}}(i) \vee \exists j \in \{1, 2, 3\} R_{\mathcal{T}}(i) \subseteq S_j, i \in W^{\varepsilon}\}$. The latter conditions ensure that all of the links in the binary tree produced by BDT, which are either results of node misgrouping or corresponding to fictitious links due to binary reconstruction, have estimated variance less than ε and are hence pruned.

Computation of the misclassification probability $P^f_{\mathrm{BDT}(\varepsilon)}$ follows the same lines of the previous section. Here we summarize the results regarding the asymptotic behavior of $\mathrm{P}^f_{\mathrm{DT}(\varepsilon)}$ which are based on the the following result, the proof of which, being similar to that of Theorem 12, is omitted.

Theorem 13: Let $(\mathcal{T}, r)$ be a canonical delay-variance tree. For each $\varepsilon > 0$, $(S_1, S_2, S_3) \in \cup_{k \in W^{\varepsilon}} \mathcal{S}(k) \cup \mathcal{S}(\varepsilon)$, $\sqrt{n} \cdot (\widehat{E}(S_1, S_2, S_3) - E(S_1, S_2, S_3))$, converges in distribution, as the number of probes $n \to \infty$, to a Gaussian random variable with mean 0 and variance $\sigma^2_E(S_1, S_2, S_3)$.

For large n, we expect the logarithms of the probabilities of errors of types 1), 2), and 3) to be asymptotically linear in n, with slopes, respectively, $c^{(i)} = (1/2) \inf_{k \in W^{\varepsilon}} \inf_{(S_1,S_2,S_3) \in \mathcal{S}(k)} (D(S_1, S_2, S_3)^2 / \sigma^2_D(S_1, S_2, S_3))$, $c^{(ii)} = (1/2) \inf_{k \in W^{\varepsilon}} \inf_{(S_1,S_2,S_3) \in \mathcal{S}(k)} ((E(S_1, S_2, S_3) - \varepsilon)^2 / \sigma^2_E(S_1, S_2, S_3))$ and $c^{(iii)} = (1/2) \inf_{(S_1,S_2,S_3) \in \mathcal{S}(\varepsilon)} ((E(S_1, S_2, S_3) - \varepsilon)^2) / \sigma^2_E(S_1, S_2, S_3))$. The dominant mode of misclassification is that with the lowest slope n. Hence, we approximate the misclassification probability to leading exponential order by

$$P^f_{\mathrm{BDT}(\varepsilon)} \approx e^{-\left(\frac{n}{2}\right) \min\left\{c^{(i)}, c^{(ii)}, c^{(iii)}\right\}}. \quad (25)$$

VIII. Conclusion

This paper analyzed a novel technique for the inference from end-to-end measurements of the variance of the delay encountered by multicast packets on an internal link. We constructed a convex family of variance estimators and found the estimator of minimal asymptotic variance. Furthermore, the underlying multicast topology can be estimated if it is not known in advance.

We investigated the modes of topology misclassification. We found that misgrouping (i.e., incorrect identification of ancestors) is far less frequent than misclassification for other reasons (false inclusion or exclusion of a link). Errors of the latter type typically apply predominantly to links with small delay variances. The consequences of such errors are expected to be small in measurement infrastructure application in which it is desired to locate the worst link, i.e., that with highest delay variance. Likewise, the algorithms are very accurate at inferring the descendency structure of the tree. This is a useful property if the information obtainable by these methods is to be used, e.g., for grouping receivers for flow control. Errors of inclusion and exclusion apply to links of smallest delay variance.

The model assumes that link delays are independent for different packets and links. Concerning the former, we observe that temporal correlations of a sufficiently short range will not impair the consistency of the estimator, although they will slow down its convergence. Concerning the latter, random early detection (RED) [16] policies in Internet routers may help reduce dependence; evidence for this comes from related work on internal link loss inference [4]. The introduction of RED was found to increase accuracy of inference relative to networks with a Drop from Tail packet discard mechanism.

Appendix
Proofs of Theorems

Proof of Theorem 1

(*i*) The proof follows from standard results in multivariate analysis; convergence to the stated Gaussian random variable follows by [22, Corollary 1.2.18].

(*ii*) Since the μ_{ij} sum to 1, the proof follows by considering the constrained minimization of $\mu \cdot C(k) \cdot \mu - 2\lambda\mu \cdot \mathbf{1}$ with Lagrange multiplier λ. As a covariance matrix, $C(k)$ is positive definite and hence invertible; minimization of the convex function of μ takes place at the the stationary point $\mu = \lambda C(k)^{-1} \cdot \mathbf{1}$. This yields $\mu^*(C(k))$ upon normalization. The corresponding minimal asymptotic variance is $(\mathbf{1} \cdot C(k)^{-1} \cdot \mathbf{1})^{-1}$. ■

Proof of Theorem 2

Clearly $\widehat{C}(k)$ converges almost surely to $C(k)$ as $n \to \infty$. Since matrix inversion is continuous on the set of strictly positive definite matrices, $\mu^*(\widehat{C}(k))$ converges almost surely (to $\mu^*(C(k))$); since each $\widehat{s}_{ij}$ converges to $s_{ij} = s_k$, $V_k(\mu^*(\widehat{C}(k)), \widehat{s})$ is consistent.

By the δ-method (see, e.g., [29]), $\sqrt{n}(V(\mu^*(\widehat{C}(k)), \widehat{s}) - s_k)$ converges to a Gaussian random variable with mean 0 and variance $\alpha \cdot C(k) \cdot \alpha$, where, for $(\ell, m) \in Q(k)$, $\alpha_{\ell m} = (\partial / \partial s_{\ell m}) \sum_{\{i,j\} \in Q(k)} \mu^*_{ij}(C(k)) s_{ij}$.

Differentiating, $\alpha_{\ell m} = \mu^*_{\ell m}(C(k)) + \sum_{\{i,j\} \in Q(k)} s_{ij} (\partial / \partial s_{\ell m}) \mu^*_{ij}(C(k))$. But $s_{ij} = s_k$ for $\{i, j\}$ in $Q(k)$ and so is constant in the sum. Since the μ^*_{ij}'s sum to 1, the sum in the above is zero. Hence $\alpha = \mu^*(C(k))$, from which the result directly follows. ■

Proof of Theorem 4

Convergence to some random Gaussian random variable in (i) is immediate from [22, Corollary 1.2.18]. It remains only to calculate the covariance matrix. Since $\widehat{v}_{ij}$ is invariant with respect to shifts in the mean of X_i or X_j, we can without loss of generality take $E[X_i] = 0$. We analyze $\mathrm{Cov}(\widehat{v}_{ij}, \widehat{v}_{\ell m})$ by expanding $\sum_{m \in I_n(\{i,j\})}$ in (12) as $\sum_{m \in I_n(\{i,j,k,\ell\})} + \sum_{m \in I_n(\{i,j\}) \setminus I_n(\{\ell,m\})}$ and similarly with $I_n(\{k,\ell\})$. Picking out the dominant terms, one finds that, conditioned on $N_n(\{i,j\})$, $N_n(\{\ell,m\})$, and $I_n(\{i,j,\ell,m\})$ all being greater than 1, $\mathrm{Cov}(\widehat{v}_{ij}, \widehat{v}_{\ell m}) = (1/N_n(\{i,j\})N_n(\{\ell,m\}))(N_n(\{i,j,\ell,m\})\mathrm{Cov}(X_iX_j, X_\ell X_m) + O(1))$. Since $n^{-1}N_n(\{i_1,\dots,i_p\})$ converges as $n \to \infty$ to $B(i_1,\dots i_p)$, the distribution of $\sqrt{n}\widehat{v}_{ij}$ has the stated property. The proofs of (ii) is analogous to that of Theorems 1 and 2. ■

Proof of Theorem 6

Suppose the algorithm does not reconstruct the tree. Then there must be an iteration of the while loop for which u and v are not siblings. Consider R', V' at the start of the first loop that this occurs. Let w be the sibling of u. $w \notin R'$ since $u \vee w \prec u \vee v$ implies $s_{uw} > s_{uv}$, contradicting the maximality of s_{uv}. Since the subtrees comprising (V', L') are disjoint, no ancestor of u (or hence of w) can lie in R'. Since the tree is binary, w must have at least two descendants t_1, t_2 in R', since otherwise $\cup_{k \in R'} R(k)$ would not cover R. Since $t_1 \vee t_2 \prec w$, then $s_{t_1t_2} > s_w > s_{u \vee v} = s_{uv}$, contradicting the maximality of s_{uv}. ■

Proof of Theorem 7

If the algorithm does not reconstruct, consider the first node k for which the outer loop fails to execute, as described above. Consider R', V' at the start of this loop, and assume that s_k is unique. Failure can happen as follows.

1) *If the first pair grouped by* DBDT *in the outer loop are not siblings*. This is excluded by Theorem 6.
2) *If* $d(k) \not\subseteq R'$. Suppose $d(k) \ni v \notin R'$. Similarly to 1), since v has siblings in R', it can have no ancestors in R' and hence has at least two descendants in R', contradicting the maximality of $s_{u_1u_2}$.
3) *If not all members are* $d(k)$ *are included in* V' *during the execution of the outer loop*. From line 8 of Fig. 4, we have $s_{U^{(m-1)}u_{m+1}} = s_{U^{(m-2)}u_{m+1}} = \dots = s_{U^{(1)}u_{m+1}} = s_k$. Hence, each u_m enters into V' during execution of the outer loop. This also show that $r_{U^{(m)}} = 0$ for $m = 1,\dots n-2$ and hence that all links $(U^{(m+1)}, U^{(m)})$, $m = 1,\dots,n-2$ are pruned by TP(0).
3) *If a nonchild node of* k *enters* V' *during the outer loop*. Since the tree is canonical, $s_{j\ell} < s_k$ for $j \notin V(k)$. Hence, such a node cannot enter into V' before the children of k.

Finally, if $s_{k_1} = \dots = s_{k_m}$ for $k_1,\dots,k_m \in R'$, then, by the tie-breaking rules, the outer loops for each k_i are performed separately, with the k_i going first for which $d(k_i)$ contains member of $\cup_i d(k_i)$ most recently added to V'. ■

Proof of Theorem 9

Consider DBDT applied to the same canonical delay-variance tree. Denote by $U = \{k,\ell\}$ the generic binary subset of S that maximizes $s_{k\ell}$ in line $5'$ of DBDT. Assume initially that U is unique. Since the delay-variance tree is canonical, $s_{k\ell} > s_{k'\ell'}$ for any other candidate binary set $\{k',\ell'\}$. By the convergence property of Theorem 8, $\mathrm{P}[V_{k,\ell}(\mu,\widehat{s}) > V_{k',\ell'}(\mu',\widehat{s})] \to 1$ as $n \to \infty$, and hence $\lim_{n\to\infty} P^f_{\mathrm{BDT}} = 0$.

If U is not unique, then there is a set $\mathcal{U}$ of pairs $U_{(j)} = \{k_{(j)}, \ell_{(j)}\}$, $j = 1,\dots m$ (some $m > 1$), each with maximal covariance. Since the tree is canonical, then, after each $U_j \in \mathcal{S}$ has been grouped in DBDT, the remaining pairs are still maximizers amongst all pairs of the reduced set $(S \setminus U_j) \cup \{U_j\}$ in line 10 of Fig. 7. Hence, the minimizing pairs in $\mathcal{U}$ are grouped successively. In BDT, the strict equality of the covariances no longer holds for finitely many probes n. However, by Theorem 8, the probability that pairs in $\mathcal{U}$ will yield the smaller m values—and so will be grouped successively—converges to 1 as $n \to \infty$. Hence, $\lim_{n\to\infty} P^f_{\mathrm{BDT}} = 0$. ■

Proof of Theorem 12

Convergence to a Gaussian random variable follows from the asymptotic normality of each term. The expression for the variance then follows from application of the δ-method. For the second statement, observe that, for $(S_1,S_2,S_3) \in \mathcal{S}(i)$, since for any $\{i,j\} \in S_1 \times S_2 s_{i\vee j} \geq s_k$ it follows that $V_{S_1,S_2}(\mu^*,s) \geq s_k$. Similarly, we have that $V_{S_1,S_3}(\mu^*,s) \leq s_{f(k)}$ for any $\{i,j\} \in S_1 \times S_3$. Therefore, $D_k(S_1,S_2,S_4) \geq r_i$. The equality is attained for $S_1 \subseteq R(h(k))$, $S_2 \subseteq R(h^*(k))$ and $S_3 \subseteq R(k^*)$ for which, for any μ, $V_{S_1,S_2}(\mu,s) = s_k$ and $V_{S_1,S_3}(\mu,s) = s_{f(k)}$. ■

ACKNOWLEDGMENT

The authors would like to thank J. Horowitz and D. Towsley for useful comments and suggestions.

REFERENCES

[1] K. G. Anagnostakis, M. B. Greenwald, and R. S. Ryger, "Cing: measuring network-internal delays using only existing infrastructure," in *Proc. IEEE INFOCOM*, San Francisco, CA, Apr. 2003, pp. 2112–2121.

[2] J. Bolot, "Characterizing end-to-end packet delay and loss in the Internet," *J. High-Speed Networks*, vol. 2, no. 3, pp. 289–298, Dec. 1993.

[3] R. Caceres, N. G. Duffield, J. Horowitz, and D. Towsley, "Multicast-based inference of network internal loss characteristics," *IEEE Trans. Inform.Theory*, vol. 45, pp. 2462–2480, Nov. 1999.

[4] R. Caceres, N. G. Duffield, J. Horowitz, D. Towsley, and T. Bu, "Multicast-based inference of network internal loss characteristics: accuracy of packet estimation," in *Proc. IEEE INFOCOM*, New York, Mar. 1999, pp. 371–379.

[5] R. Caceres, N. G. Duffield, and T. Friedman, "Impromptu measurement infrastructures using RTP," in *Proc. IEEE INFOCOM*, New York, June 2002, pp. 1490–1499.

[6] K. Claffy, G. Polyzos, and H.-W. Braun, "Measurements considerations for assessing unidirectional latencies," *Internetworking: Res. Experience*, vol. 4, no. 3, pp. 121–132, Sept. 1993.

[7] M. J. Coates and R. Nowak, "Network loss inference using unicast end-to-end measurement," in *Proc. ITC Conf. IP Traffic, Modeling and Management*, Monterey, CA, Sept. 2000, pp. 28-1–28-9.

[8] M. J. Coates and R. Nowak, "Network delay distribution inference from end-to-end unicast measurement," in *IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, Salt Lake City, UT, May 2001.

[9] R. L. Carter and M. E. Crovella, "Measuring bottleneck link speed in packet-switched networks," in *Proc. PERFORMANCE '96*, Lausanne, Switzerland, Oct. 1996.

[10] A. Downey, "Using pathchar to estimate Internet link characteristics," in *Proc. ACM SIGCOMM*, Cambridge, MA, 1999.

[11] N. G. Duffield, J. Horowitz, F. Lo Presti, and D. Towsley, "Network delay tomography from end-to-end unicast measurements," in *Proc. Tyrrhenian Int. Workshop Digital Communications*, Taormina, Italy, Sept. 2001.

[12] N. G. Duffield and F. Lo Presti, "Multicast inference of packet delay variance at interior network links," in *Proc. IEEE INFOCOM*, Tel Aviv, Israel, Mar. 2000, pp. 1351–1360.

[13] N. G. Duffield and F. Lo Presti, "Network Tomography From Measured End-to-End Delay Covariance," Computer Science Dept., Università dell'Aquila, Tech. Rep. 010-2004, Mar. 2004.

[14] N. G. Duffield, F. Lo Presti, V. Paxson, and D. Towsley, "Inferring link loss using striped unicast probes," in *Proc. IEEE INFOCOM*, Anchorage, AK, Apr. 2001, pp. 915–923.

[15] N. G. Duffield, J. Horowitz, F. Lo Presti, and D. Towsley, "Multicast topology inference from measured end-to-end loss," *IEEE Trans. Inform. Theory*, vol. 8, pp. 26–45, Jan. 2002.

[16] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Trans. Networking*, vol. 1, pp. 397–413, Aug. 1993.

[17] K. Almeroth, R. Cole, N. G. Duffield, K. Hedayat, K. Sarac, and M. Westerlund, "RTP Control Protocol Extended Reports (RTCP XR),", RFC 3611, T. Friedman, R. Caceres, and A. Clark, Eds., Nov. 2003.

[18] F. Lo Presti, N. G. Duffield, J. Horowitz, and D. Towsley, "Multicast-based inference of network-internal delay distributions," *IEEE/ACM Trans. Networking*, vol. 10, pp. 761–775, Dec. 2002.

[19] F. Lo Presti, N. G. Duffield, J. Horowitz, and D. Towsley, "Multicast-based inference of network-internal delay distributions," UMass, CMPSCI Tech. Rep. 99-55, Sept. 1999.

[20] V. Paxson, J. Mahdavi, A. Adams, and M. Mathis, "An architecture for large-scale Internet measurement," *IEEE Trans. Commun.*, vol. 36, pp. 48–54, Aug. 1998.

[21] M. Mathis and J. Mahdavi, "Diagnosing Internet congestion with a transport layer performance tool," in *Proc. INET* , Montreal, ON, Canada, June 1996.

[22] R. J. Muirhead, *Aspects of Multivariate Statistical Theory*. New York: Wiley, 1982.

[23] A. Mukherjee, "On the dynamics and significance of low frequency components of Internet load," *Internetworking: Res. Experience*, vol. 5, pp. 163–205, Dec. 1994.

[24] S. Paul *et al.*, "Reliable multicast transport protocol (RMTP)," *IEEE J. Select. Areas Commun.*, vol. 15, pp. 407–421, Apr. 1997.

[25] V. Paxson, "End-to-end routing behavior in the Internet," in *Proc. ACM SIGCOMM*, Stanford, CA, Aug. 1996.

[26] V. Paxson, "End-to-end Internet packet dynamics," in *Proc. ACM SIGCOMM*, Cannes, France, Sept. 1997, pp. 139–152.

[27] V. Paxson, "Automated packet trace analysis of TCP implementations," in *Proc. ACM SIGCOMM*, Cannes, France, Sept. 1997, pp. 167–179.

[28] S. Ratnasamy and S. McCanne, "Inference of multicast routing tree topologies and bottleneck bandwidths using end-to-end measurements," in *Proc. IEEE INFOCOM*, New York, Mar. 1999, pp. 353–360.

[29] M. J. Schervish, *Theory of Statistics*. New York: Springer, 1995.

[30] *RTP: A Transport Protocol for Real-Time Applications*, January 1996.

[31] Y. Zhang, N. G. Duffield, V. Paxson, and S. Shenker, "On the constancy of Internet path properties," in *Proc. ACM SIGCOMM Internet Measurement Workshop*, San Francisco, CA, Nov. 2001.

[32] N. G. Duffield and F. Lo Presti, "Multicast inference of packet delay variance at interior network links," in *Proc. IEEE INFOCOM*, Tel Aviv, Israel, Mar. 2000, pp. 1351–1360.

N. G. Duffield (M'97–SM'01) received the B.A. degree in natural sciences and the Certificate of Advanced Study in Mathematics from the University of Cambridge, Cambridge, U.K., in 1982 and 1983, respectively, and the Ph.D. degree in mathematical physics from the University of London, London, U.K., in 1987.

He subsequently held postdoctoral and faculty positions in Heidelberg, Germany and Dublin, Ireland. He is currently a Technology Leader with the Internet and Networking Research Group, AT&T Labs-Research, Florham Park, NJ. His research focuses on Internet performance measurement, inference, and analysis.

Francesco Lo Presti received the Laurea degree in electrical engineering and the Ph.D. degree in computer science from the University of Rome "Tor Vergata," Rome, Italy, in 1993 and 1997, respectively.

He subsequently held a postdoctoral position with the Computer Science Department, University of Massachusetts, Amherst. Since 2001, he has been an Assistant Professor with the Computer Science Department, Università dell'Aquila, Coppito, Italy. His research interests include measurements, modeling, and performance evaluation of computer and communications networks.