Network Tomography from Measured End-to-End Delay Covariance*

N.G. Duffield[1]
[1]AT&T Labs–Research
180 Park Avenue
Florham Park, NJ 07932, USA
duffield@research.att.com

F. Lo Presti[1,2]
[2]Dept. of Computer Science
University of Massachusetts
Amherst, MA 01003, USA
lopresti@research.att.com

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 underlying 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.

Keywords: multicast, end-to-end measurment, packet delay, statistical inference, topology discovery.

1 Introduction

1.1 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 to performance diagnosis. In the *internal* approach, direct measurements are made at or between network elements, e.g. of packet loss or delay, involving possibly both active and passive measurements. This approach has a number of potential limitations: (i) it may not be available for general users; (ii) coverage may not span paths of interest; (iii) measurements may be disabled during period of high load; (iv) there are issues of scale gathering and correlating the measurements in large networks; (v) how should per hop measurements be composed to form an end-to-end view?

This motivates *external* approaches, diagnosing the network through end-to-end measurements, without necessarily assuming the cooperation of network elements on the path. There has been much recent experimental work to understand the phenomenology of end-to-end performance (e.g., see [1, 2, 8, 16, 21, 24,

*This work was supported in part by DARPA and the Air Force Research Laboratory under agreement F30602-98-2-0238.

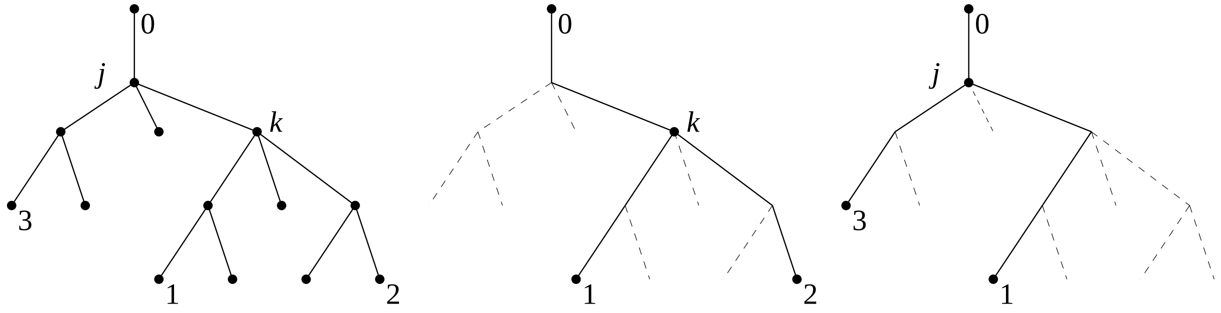


Figure 1: Logical Multicast Tree (left) and two embedded two receivers trees (center and right).

25, 27]); There are presently several measurement infrastructure projects (including CAIDA [6], Felix [11], IPMA [13], NIMI [15], Surveyor [31]) 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, roundtrip loss and delay in IP networks. `pathchar` [9] extends the approach of `traceroute` to estimate hop-by-hop link capacities, packet delay and loss rates. These approaches have several potential drawbacks: (i) delays may not be representative of regular traffic, since their generation of Internet Control Message Protocol (ICMP) packets can have low priority in routers; (ii) roundtrip reporting and possibly asymmetric paths hinder the unambiguous attribution of delays to specific link directions; (iii) encapsulation may hide the TimeToLive (TTL) field in the IP header from higher layers, and hence approaches that depend on TTL manipulation–such as `traceroute` and `pathchar`–may only see single composite hops between tunnel endpoints.

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. 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 on the network may grow proportionally to N^2 in some links of the network. With multicast, the load grows proportionally only to N.

1.2 Contribution

In this paper we describe 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. We assume first that link delays are independent random variables, both spatially (i.e. between different links) and temporally (i.e. between different packets). Consider the logical multicast topology of Figure 1(left), 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 2 leaf tree formed by the root 0, leaf nodes 1 and 2 and their nearest common ancestor k; see Figure 1(center). From the independence of link delays, it follows that

$$\mathsf{Var}(X_k) = \mathsf{Cov}(X_1, X_2); \qquad (1)$$

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 Figure 1(right). Then $\mathsf{Var}(X_j) = \mathsf{Cov}(X_1, X_3)$. Observe that $X_k = X_j + D_k$, and X_j and D_k are independent. Therefore,

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

This expresses the variance of the packet delay on the internal link from node i to node k, in terms of the covariances of source-to-leaf delays. We can form an unbiased estimate of the latter directly from end-to-end measurements, from which we obtain unbiased estimators for (1) and (2). In Section 2 we specify the delay model, and these basic estimators for a known topology. We also give a generalization for estimating higher order delay moments in certain topologies.

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 3 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. The formulation of our minimum variance estimators 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 binary tree we are able to make use of the natural recursive structure of the tree to simplify the calculation. We provide an algorithm for this in Section 4.

In Section 5 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 [5, 29]. The method here extends to general trees. We prove the resulting topology estimator is consistent, and evaluate it through model-based simulations in Section 6. A closer analysis of the modes of failure, and their asymptotic probabilities, is made in Section 7. We conclude in Section 8. The proofs of the theorems are deferred to Section 9. Some of the results from Section 3 were announced by us previously in [10].

1.3 Implementation Requirements

Since the data for inference comprises one-way packet delays, we require source and receiver clocks to be sufficiently synchronized over a measurement period during which a given set of probes is dispatched. Since delay variance calculations are insensitive to absolute time shifts, it is important to control only the relative clock drift. Sufficiently small clock drift may be corrected for; see [18, 26, 28]. We note that several of the measurement infrastructures mentioned earlier use Global Positioning System (GPS) for synchronization; this enables absolute one-way delay measurements accurate to within tens of microseconds or better. The Network Time Protocol (NTP) [17] is more widely deployed, but provides accuracy in only the order a few tens of milliseconds.

1.4 Applications and Related Work

Multicast-based network inference tools based on loss measurements have been deployed in NIMI. We plan to supplement these with delay-based variance inference. Physical topology is currently laid out using the `mtrace` [19] 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. Knowledge of the multicast topology can be helpful to multicast applications. Several reliable multicast protocols rely on logical hierarchies based on the underlying topology if possible; see, e.g., [22]. Other applications attempt to group receivers that share the same network bottleneck, [29].

The delay variance estimates themselves can be used to detect links of higher delay variance. Since the performance of delay sensitive applications may degraded on traversing such a link, such information may be used to control routing in order that the traffic passes over other links. 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 $\mathsf{Var}(J^i) = \mathsf{Var}(D^i) + \mathsf{Var}(D^{i+1}) - 2\mathsf{Cov}(D^i, D^{i+1})$. Assuming stationarity and independence, this yields $\mathsf{Var}(J^i) = 2\mathsf{Var}(D^i)$. Measurements of end-to-end delays in the Internet [1] show that end-to-end delays successive packets are only slightly dependent when the interpacket time is longer than the typical queueing timescales. Stronger dependence is found at shorter timescales: successive packets are more likely to queue together. With positive correlation between successive probe delays $\mathsf{Cov}(D^i, D^{i+1}) > 0$; in this case $\mathsf{Var}(J^i)$ is bounded above by $2\,\mathsf{Var}(D^i)$, a quantity that we can estimate.

2 Tree and Delay Models and Non-Parametric Estimation

Tree Model. We identify the physical multicast tree as comprising actual network elements (the nodes) and the communication links than 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 2 or more children.

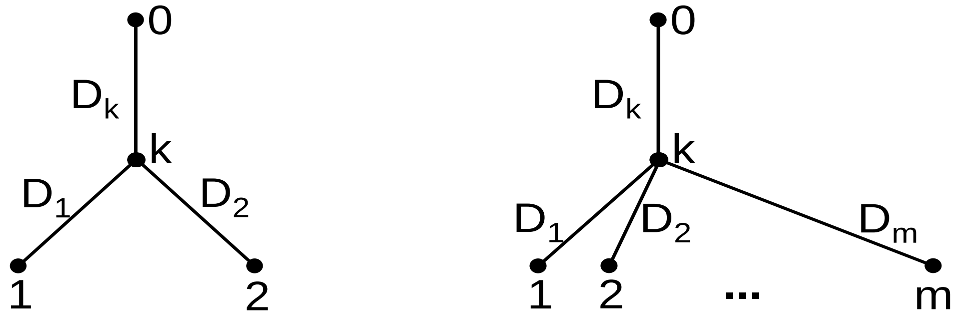


Figure 2: LEFT: Two leaf tree. RIGHT: m-leaf tree.

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). 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.

Delay-Variance Tree Model. The delay on link k is 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; $\alpha_k = \mathsf{P}[D_k < \infty]$ is the probability of successful transmission across the link. We assume the D_k are independent random variables. The delay experienced 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 $r_k = \mathsf{Var}(D_k | D_k < \infty)$ and $s_k = \mathsf{Var}(X_k | X_k < \infty)$. By the assumption of link delay independence, $s_k = \sum_{j \succeq k} r_j$. 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 condition implies that $s_i > s_j$ when $i \prec j$. Any delay-variance tree $(\mathcal{T}, r)$ not in canonical form can be reduced to one in canonical form by removing zero variance links and identifying their endpoints. Henceforth, we assume that the underlying tree is a canonical delay-variance trees.

Cumulative Delay Variance Estimation. Consider first a logical subtree of a logical multicast tree $\mathcal{T}$ formed by the root 0, and a non-leaf node k with two descendents 1 and 2 that are leaf nodes; see Figure 2(left). We assume initially that all delays are finite $\mathsf{P}[D_k = \infty] = 0$. Then:

$$\mathsf{Cov}(X_1, X_2) \quad = \quad \mathsf{Cov}(X_k + (X_1 - X_k), X_k + (X_2 - X_k))$$

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

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 $\mathsf{Cov}(X_1, X_2)$ is also an unbiased estimator of $\mathsf{Var}(X_k)$. Let $X_1^{(i)}, X_2^{(i)}$, $i = 1, 2, \dots n$ be measured end-to-end delays between the root 0 and leaf nodes 1 and 2 respectively. Abbreviate $\mathsf{Cov}(X_j, X_k)$ 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

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

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 of the r_k through $\widehat{r}_k = \widehat{s}_k - \widehat{s}_{f(k)}$.

General Delay Moment Estimation. This approach generalizes to nodes with branching ratio $m > 2$; see Figure 2(right). Denote the joint cumulants of the end-to-end delays $X_1, \dots, X_m$ by

$$K^{j_1,\dots,j_m}(\boldsymbol{X_1,\dots,X_m}) = \left(\prod_{i=1}^{m} \frac{\partial^{j_i}}{\partial\theta_i^{j_i}}\right) \log \mathsf{E}[\exp(\sum_{i=1}^{m}\theta_i X_i)]\Big|_{\theta_i=\mathbf{0}} \tag{5}$$

These have the property that $K(\boldsymbol{X}+\boldsymbol{Y}) = K(\boldsymbol{X}) + K(\boldsymbol{Y})$ whenever $\boldsymbol{X}$ and $\boldsymbol{Y}$ are independent vectors of random variables. Hence

$$K^{1,\dots,1}(X_1,\dots,X_m) = K^{1,\dots,1}(X_k,\dots,X_k) = K^m(X_k), \tag{6}$$

i.e., the m^{th} cumulant of the delay on the common link equals the joint cumulant of the end-to-end delays.

3 Delay Variance Estimation on General Trees

3.1 Unbiased Delay Variance Estimators

In a general tree let $Q(k) = \{\{i,j\} \subset R \mid i \vee j = k, \}$ 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 the **uniform estimator**

$$\frac{1}{\#Q(k)} \sum_{\{i,j\}\in Q(k)} \widehat{s}_{ij}. \tag{7}$$

One potential 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} that are functions of the end-to-end delays themselves in order to reduce the estimator variance. In this section we shall assume that all delays are finite and have bounded fourth moments. Later we shall relax the finiteness assumption.

We formalize the notion of (possibly random) convex combinations of $\widehat{s}_{ij}$ through a **covariance aggregator**. For $S \subset R$, let $\mathcal{F}_n(S)$ denote the σ-algebra generated by the end-to-end delays $(X_k)_{k\in S}$ (i.e. the set of events that can be determined from knowing $(X_k)_{k\in S}$). A covariance aggregator μ is sequence $(\mu(n))_{n\in\mathbb{N}}$ of random vectors $\{\mu_{ij}(n) : \{i,j\} \in Q(k);\ k \in V \setminus R\}$ with $0 \leq \mu_{ij}(n) \leq 1$ and $\sum_{\{i,j\}\in Q(k)} \mu_{ij}(n) = 1$ for each $k \in V \setminus R$. We assume each $\mu(n)$ to be $\mathcal{F}_n(R)$-measurable, i.e., that it is a function of the measured delays of the first n probes. We will usually suppress the explicit dependence on the number of probes n.

Let $\widehat{s} = \{\widehat{s}_{ij}(n) : \{i,j\} \in Q(k);\ k \in V \setminus R\}$ be a family of estimators, each $\widehat{s}_{ij}(n)$ being an $\mathcal{F}_n(\{i,j\})$-measurable unbiased estimator of $\mathsf{Var}(X_k)$. Given a covariance aggregator μ, we can estimate $\mathsf{Var}(X_k)$ by

$$V_k(\mu, \widehat{s}) = \sum_{\{i,j\}\in Q(k)} \mu_{ij}\widehat{s}_{ij} \tag{8}$$

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 (7): $\mu_{ij} = (\#Q(k))^{-1}$.

3.2 Minimum Variance Estimation of Cumulative and Link Delays

Define the covariance matrix

$$C_{(ij),(\ell m)} = \mathsf{Cov}\ (\ Z_i Z_j\ ,\ Z_\ell Z_m\ )\ , \tag{9}$$

where $Z_i = X_i - \mathsf{E}[X_i]$. We will use $C(k) = \left[C_{(ij),(\ell m)}\right]_{\{i,j\},\{\ell,m\}\in Q(k)}$ to denote the matrix obtained by letting the indices (ij) and (ℓm) in (9) run over $Q(k)$; this is a submatrix of the matrix $C^0(k)$ obtained by taking the indices unrestricted over the set $Q^0(k)$ of binary subsets of $R(k)$.

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 $\{\sqrt{n}\ (\widehat{s}_{ij} - s_k) \mid \{i,j\} \in Q(k)\}$ converge in distribution as $n \to \infty$ to a multivariate Gaussian random variable with mean 0 and covariance matrix $C(k)$. Hence the $\widehat{s}_{ij}$ are consistent estimators of s_k, and so is $V(\mu, \widehat{s})$. For any 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 zero 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

$$\mu_{ij} = \mu^*_{ij}(C(k)) := \left(C(k)^{-1} \cdot \mathbf{1}\right)_{(ij)} \Big/ \mathbf{1} \cdot C(k)^{-1} \cdot \mathbf{1} \tag{10}$$

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 $\left(\mathbf{1} \cdot C(k)^{-1} \cdot \mathbf{1}\right)^{-1}$.

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

$$\widehat{C}(k)_{(ij),(i'j')} \quad = \quad \frac{n^2}{(n-1)^3}\left(\sum_{m=1}^n Z_i^{(m)} Z_j^{(m)} Z_{i'}^{(m)} Z_{j'}^{(m)} - \frac{1}{n}\sum_{m=1}^n Z_i^{(m)} Z_j^{(m)} \sum_{m=1}^n Z_{i'}^{(m)} Z_{j'}^{(m)}\right) \tag{11}$$

$\widehat{C}(k)$ 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 $\left(\mathbf{1}\cdot C(k)^{-1}\cdot\mathbf{1}\right)^{-1}$.*

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 a unbiased estimate of r_k as

$$W_k(\mu,\widehat{s}):=V_k(\mu(k),\widehat{s})-V_{f(k)}(\mu(f(k)),\widehat{s}) \tag{12}$$

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

$$C'(k)=\begin{pmatrix} C(k) & C(k,f(k)) \\ C(k,f(k))^T & C(f(k)) \end{pmatrix}, \tag{13}$$

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

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

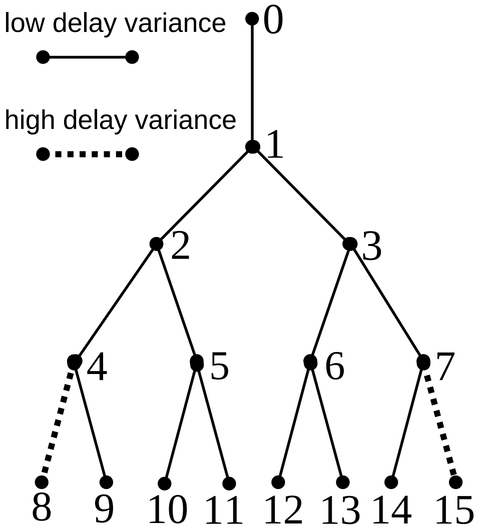
$$\begin{pmatrix} \mu(k) \\ \mu(f(k)) \end{pmatrix}=(c_1c_2-c_3^2)^{-1}(C')^{-1}(k)\begin{pmatrix} (c_2+c_3)\mathbf{1}_k \\ -(c_1+c_3)\mathbf{1}_{f(k)} \end{pmatrix} \tag{14}$$

and takes the value $(c_1+c_2+2c_3)/(c_1c_2-c_3^2)$ 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.

3.3 Example of Minimum Variance Estimator

The difference between uniform and minimum variance delay estimated is more marked when the link delay variances are more heterogeneous. We illustrate this in the 8-leaf binary tree of Figure 3(left). Consider, for example, the case that the delay variance on links 8 and 15 is 100 times that on all other links, i.e. the delays are scaled by a factor 10. In the minimum variance estimator, the weighting μ_{ij} is reduced when i or j is descended through a high delay variance link. In this topology, this occurs when estimating delays to nodes 1, 2, or 3. As an example we tabulate the weights $\mu_{ij}(C(1))$ in the table in Figure 3(right). The weight for the pair $(8,15)$ of high variance links is 10^{-4} times the highest weight, that for pair $(9,14)$.

We compare the variance of the uniform and minimum variance estimators. From Theorem 1, the minimum variance of estimated cumulative delay variance to node k is $\left(\mathbf{1}\cdot C(k)^{-1}\cdot\mathbf{1}\right)^{-1}$, while that of the uniform estimator is $\left(\mathbf{1}\cdot C(k)\cdot\mathbf{1}\right)/\#Q(k)^2$. For $k=1$, estimator variance is reduced by a factor of



Weight μ_{ij}	Link pairs (i,j)
0.000018	(8,15)
0.001213	(8,12) (8,13) (10,15) (11,15)
0.001811	(8,14) (9,15)
0.081286	(10,12) (10,13) (11,12) (11,13)
0.121322	(9,12) (9,13) (10,14) (11,14)
0.181077	(9,14)

Figure 3: MINIMUM VARIANCE ESTIMATION: LEFT: 8 leaf binary tree; Links 8 and 15 have delay variance 100 times that of others. RIGHT: Weights for Minimum Variance Estimator.

approximately 8.8; for $k = 2, 3$ by a factor approximately 5.7. All other nodes k have only two descendants, one of which may terminate a high variance link; there is no flexibility to avoid the any high variance V_k, and hence the factor is 1.

3.4 Impact of Loss on Estimator Variance

Although lost packets clearly will not provide 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 $n \to \infty$.

Then we can adapt the approach of the foregoing theory by forming an estimator $\widehat{u}_{ij}$ of the variance of the cumulative delay of packets reaching k, analogously 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 following replacements: $\widehat{s} \mapsto \widehat{u}$, $C(k) \mapsto C(k)/B(R(k))$, and in the definition (11) of $\widehat{C}(k)$ replacing $n \mapsto N_n(R(k))$ and $\sum_{m=1}^{n} \mapsto \sum_{m \in I_n(R(k))}$. Summarizing, 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 no loss case as convergence rates are multiplied by a factor $B(R(k)) < 1$.

A disadvantage of this approach is that is 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

$$\widehat{v}_{ij} = \frac{1}{N_n(\{i,j\})-1}\left(\sum_m X_i^{(m)}X_j^{(m)} - \frac{1}{N_n(\{i,j\})}\sum_{m,m'} X_i^{(m)}X_j^{(m')}\right) \tag{15}$$

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

Theorem 4 *(i) For each $k \in V \setminus R$ the random variables $\{\sqrt{n}\ (\widehat{v}_{ij} - s_k) \mid \{i,j\} \in Q(k)\}$ 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 $\left(\mathbf{1} \cdot G(k)^{-1} \cdot \mathbf{1}\right)^{-1}$.

(iii) $V_k(\mu^*(\widehat{G}), \widehat{v})$ has the same asymptotic properties as $V_k(\mu^*(G), \widehat{v})$ where the estimated covariance $\widehat{G}$ is defined by

$$\frac{N_n(\{i,j\})N_n(\{k,\ell\})}{N_n(\{i,j,k,\ell\})}\widehat{G}_{(ij),(k\ell)} = \sum_m Z_i^{(m)}Z_j^{(m)}Z_k^{(m)}Z_\ell^{(m)} - \frac{1}{N_n(\{i,j,k,\ell\})}\sum_{m,m'} Z_i^{(m)}Z_j^{(m)}Z_k^{(m')}Z_\ell^{(m')} \tag{16}$$

where the sums run over $I_n(\{i,j,k,\ell\})$.

3.5 Inference Accuracy

Some of the results of this section have been previously announced by us, without proof, in [10]. There, we also investigated the experimental accuracy of the delay variance estimators. 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. We also conducted network-level simulations using ns [23]. These simulated probe and background traffic at the packet level, with packet delay and loss occurring through simulated queueing and buffer overflow. These simulations test the robustness of the method to violations of the delay independence assumption. Delay correlations were larger when smaller buffers were used. With the minimum variance estimator, the estimated and actual delay variance differed by a median factor of about 1.3 for small correlations, rising to about a factor 2 for larger correlations. The uniform estimator had noticeably higher error factors in the latter case.

4 Computational Approaches 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 may preclude practical calculations due to the computational cost. For example, in a perfectly balanced tree of depth m and branching ratio r, the number of covariances to be calculated for estimation of all s_k 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.

4.1 Capitalizing on the Tree Structure

For $i \prec k$ let $d(k,i) = \{j \in d(k) \mid i \preceq j\}$, 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 whose significance we explain shortly:

$$\mu_{ij} = \phi_{d(i\vee j,i)\ d(i\vee j,j)} \left(\psi_i \psi_{f(i)} \dots \psi_{d(i\vee j,i)}\right) \left(\psi_j \psi_{f(j)} \dots \psi_{d(i\vee j,j)}\right), \tag{17}$$

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

$$\{\phi_{ij} \mid \{i,j\} \subset d(k);\ k \in V \setminus R\} \quad \text{with} \quad \sum_{\{i,j\}\subset d(k)} \phi_{ij} = 1 \tag{18}$$

and ϕ_{ij} being $\mathcal{F}_n(R(i \vee j))$-measurable; and

$$\{\psi_j \mid j \in V\} \quad \text{with} \quad \sum_{j \in d(k)} \psi_j = 1,\ \forall k \in V \setminus R \tag{19}$$

and ψ_j being $\mathcal{F}_n(R(f(j)))$-measurable.

The significance of this form becomes apparent after we define node averaged delays recursively through

$$Y_k = \sum_{j\in d(k)} Y_j \psi_j \quad \text{with} \quad Y_k = X_k,\ k \in R; \tag{20}$$

each Y_k is an average of the end-to-end delays seen at the receivers descended from k. Using the Y_k, we associate estimators $\widehat{w}_{ij}$ of $s_{i\vee j}$ through

$$\widehat{w}_{ij} = \frac{1}{N_n(R(k)) - 1} \left(\sum_{m\in I_n(R(k))} Y_i^{(m)} Y_j^{(m)} - \frac{1}{N_n(R(k))} \sum_{m\in I_n(R(k))} Y_i^{(m)} \sum_{m \in I_n(R(k))} Y_j^{(m)} \right). \tag{21}$$

Note than only probes in $I_m(R(k)$, i.e. those received at *all* nodes in $R(k)$, are used in (21).

If we now use a convex combination of the $\widehat{w}_{ij}$ (instead of the $\widehat{v}_{ij}$ in (8)), we obtain

$$V_k(\mu, \widehat{w}) = \sum_{\{i,j\}\in Q(k)} \mu_{ij} \widehat{w}_{ij} = \sum_{\{i,j\}\subset d(k)} \phi_{ij} \widehat{w}_{ij}. \tag{22}$$

Observe the reduced number of covariances to be calculated in the RHS of (22). Using a local covariance aggregator to combine the $\widehat{w}_{ij}$ allows us to take advantage of the inherent recursive structure of the tree through (20). For the perfectly balance 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} is the general case. However, since we use only packets in $I_n(R(k))$ to estimate s_k there is a trade-off between this computational reduction and the increase of variance due to the reduced number of packets.

4.2 Minimal Variance Estimators on Binary Trees

An example of a local aggregator is the **uniform local aggregator** in which averages uniformly across siblings with $\psi_i = 1/\#d(f(i))$ and $\phi_{ij} = 2/(\#d(k)(\#d(k)-1))$. But it is natural to optimize the variance over all local aggregators. Since $\mathsf{Var}(V_k(\mu, \widehat{w})) \geq \mathsf{Var}(V_k(\mu, \widehat{v}))$ such an estimator may not be optimal over the set of all covariance aggregators; put another way, μ^* in (10) 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 $\mathsf{Var}(\mu^*, \widehat{w})$ is consistent, the estimated delay variance for these links converges to 0 as $n \to \infty$: these inserted can then be removed at the end of the calculation. Indeed, we shall use this approach when we address topology inference in Section 5.

Let S_k denote the $\#R(k)$-dimensional matrix with entries $s_{i\vee j}$, and U_k the $R(k)$-dimensional matrix with all entries equal 1. In a binary tree let i^* be the unique sibling of a node i (except the root and its unique descendant).

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

$$\psi_i = \frac{\delta(i^*)}{\delta(i) + \delta(i^*)} \qquad \text{where} \qquad \delta(i) = \det\left(S_i - s_{f(i)}U_i\right). \tag{23}$$

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

5 Topology Inference Through Delay Variance Estimation

In this section we show how the foregoing approach can be adapted to infer the underlying tree $\mathcal{T}$ when it is not known in advance. The key observation underlying the approach 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, should then reconstruct the binary tree. In this section we formalize the foregoing approach and show how it can be extended to reconstruct arbitrary canonical delay variance trees.

5.1 Deterministic Reconstruction of Delay-Variance Trees

We now show that canonical delay-variance trees with receiver set R are in 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. In the next subsection this algorithm is adapted to estimated 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 Figure 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

```
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 $((V', L'), r)$ ;
```

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

$U = \{u, v\}$ 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.

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 Figure 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 the effect is to reconstruct a non-canonical 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

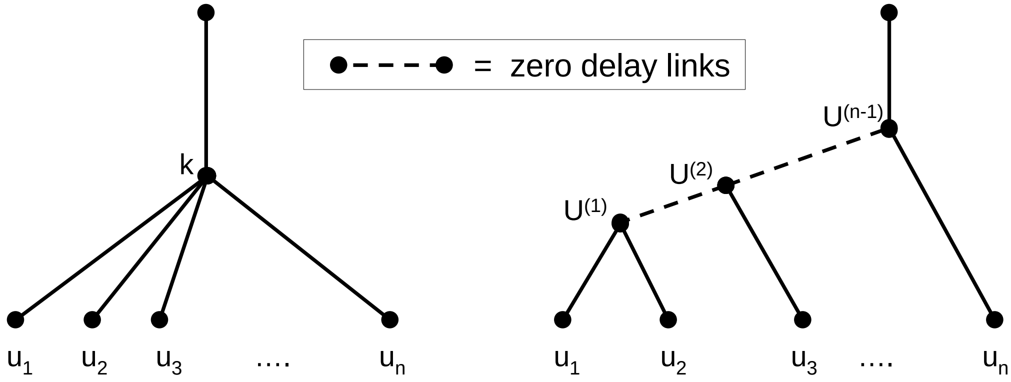


Figure 5: LEFT: General node with n children RIGHT: Example of corresponding binary tree with zero delay links.

useful to specify a generalization of this procedure. For each $\varepsilon > 0$, the Tree Pruning Algorithm $\mathsf{TP}(\varepsilon)$ 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 $\mathsf{TP}(0)$. We specify TP in Figure 6.

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 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 Figure 4 when there is more than one maximizer. One example is to select the maximizing pair $\{u, v\}$ for which u most recently included in V' and v the next most recently included, and 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 Figure 5.

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

5.2 Inference of Loss Tree from Measured Leaf Delay Covariances

We now present stochastic versions of the above algorithms that estimate topology based on *estimated* delay covariances. We adapt the minimum variance approach of Section 3 as follows. Given a pair of nodes $\{k, \ell\}$ we can estimate $\mathsf{Cov}(X_k, X_\ell)$ by

$$V_{k,\ell}(\mu, \widehat{s}) = \sum_{\{i,j\} \in R(k) \times R(\ell)} \mu_{ij} \widehat{s}_{ij} \tag{24}$$

where $\widehat{s} = \{\widehat{s}_{ij}\}_{i,j \in R}$ and μ is a covariance aggregator. This estimator obeys analogous properties to established in Section 3 directly follows. 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 $\mu = \mu^*(C(k, \ell)) = \left(C(k, \ell)^{-1} \cdot \mathbf{1}\right) / \mathbf{1} \cdot C(k, \ell)^{-1} \cdot \mathbf{1}$.

```
*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:* $(\mathcal{T}', r')$
```

Figure 6: Tree Pruning Algorithm $\mathsf{TP}(\varepsilon)$

.

Denote by $\widehat{C}(k,\ell)$ the empirical version of $C(k,\ell)$, i.e., the covariance matrix with entries given by (11). Similar to Theorem 2 we have:

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 Figure 7. This combines DBDT with the minimum variance estimator from (24), taking advantage of the tree structure and the optimality of the local aggregator for binary trees. Note that, 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 $\widehat{\mathcal{T}}_{\mathsf{X}}$ to distinguish the topology inferred by a particular algorithm X. P^f_{X} will denote the probability of false identification of topology $\mathcal{T}$ of the delay-variance tree $(\mathcal{T}, r)$ i.e. $P^f_{\mathsf{X}} = \mathsf{P}_{\mathcal{T},r}[\widehat{\mathcal{T}}_{\mathsf{X}} \neq \mathcal{T}]$.

Theorem 9 *Let $(\mathcal{T}, r)$ be a binary canonical delay variance tree.* $\lim_{n\to\infty} P^f_{\mathsf{BDT}} = 0$.

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). But using 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

```
Input: The set of receivers $R$, 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 $Y_{\{u,v\}}^{(m)} := \Big(\delta(u)Y_u^{(m)} + \delta(v)Y_m^{(m)}\Big) \Big/ (\delta(u)+\delta(v))$ ; enddo
enddo
Output: delay-variance tree $((\{0\} \cup V', \{(0,R')\} \cup L'), \{0\} \cup r)$
procedure $w(i,j)$ {return $(n-1)^{-1}(\sum_{m=1}^{n} Y_i^{(m)}Y_j^{(m)} - n^{-1}\sum_{m=1}^{n} Y_i^{(m)} \sum_{m=1}^{n} Y_j^{(m)})$ };
```

Figure 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.

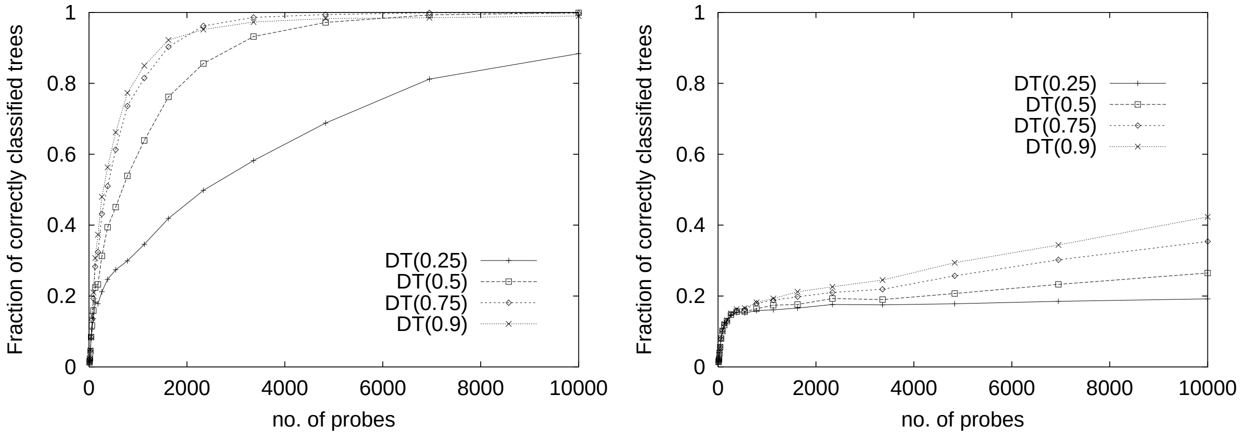
are pruned if the estimated link delay variance does not exceed ε. For each $\varepsilon > 0$ the **Delay-Variance Tree Classification Algorithm** is $\mathsf{DT}(\varepsilon) = \mathsf{TP}(\varepsilon) \circ \mathsf{BDT}$. Since link delay variance estimates become accurate as the number of probes grows 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 $\varepsilon \in (0, \varepsilon')$, $\lim_{n\to\infty} P^f_{\mathsf{DT}(\varepsilon)} = 0$.*

6 Simulation Evaluation of Topology Inference

We evaluated the accuracy of the classification algorithms in a number of model-based simulations in which the link delays are independent exponentially distributed random variable. Unless otherwise stated, we assume no packet loss.

Dependence of Accuracy on Threshold ε. We conducted 1000 simulations over randomly generated trees of 15 nodes and maximum branching ratio 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 Figure 8(a) we plot the fraction of correctly classified trees for the different general tree classification algorithms and $\varepsilon = 0.25, 0.5, 0.75, 0.9$. Except with small numbers of probes, accuracy is best for $\varepsilon = 0.75$. Smaller



(a) Variance in $[1, 10]$

(b) Variance in $[1, 100]$

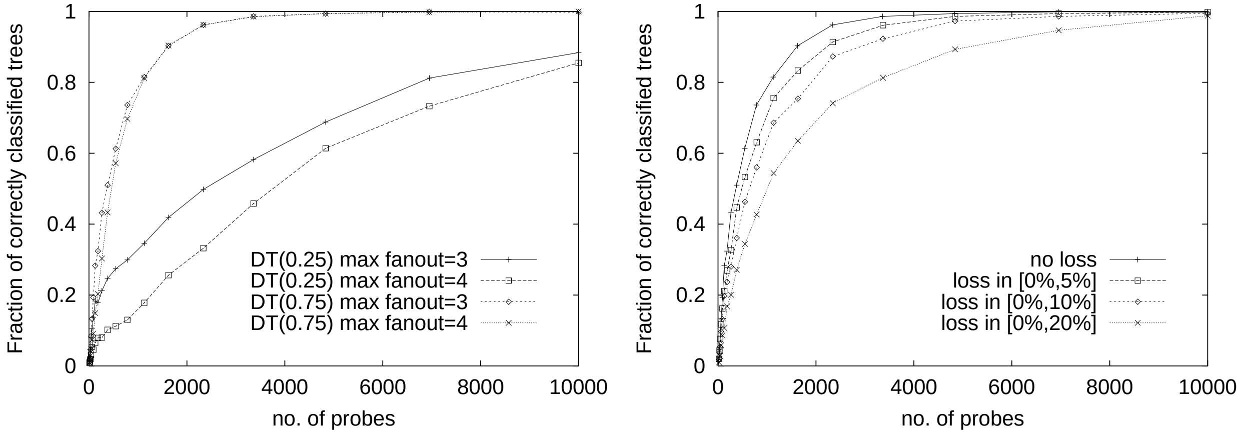
Figure 8: DEPENDENCE OF THE ACCURACY ON THE THRESHOLD ε. Fraction of trees correctly classified by $\mathsf{DT}(\varepsilon)$ in 1000 simulations over randomly generated 15 nodes tree for $\varepsilon = 0.25, 0.5, 0.75, 0.9$. Link variance is uniformly distributed in the interval [1,10] (a) [1,100] (b).

values of ε result in stricter grouping criteria and so 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 would rapidly drop to zero.

Dependence of Accuracy on Variance Spread and Topology. Accuracy decreases noticeably when the range of possible variance is expanded to $[1, 100]$; with 1000 probes and $\varepsilon = 0.75$ only about 35% of the trees were correctly classified, see Figure 8(b). The corresponding proportion was 100% for variances in $[1, 10]$. This occurs because large delay variance lead to larger estimator variances, and hence mistaken pairing of non-sibling nodes, or erroneous inclusion or exclusion of nodes in a group, is more likely to occur. In this example, the algorithm performs poorly because the largest delay variance possible 100, is much larger than the smallest, 1, and so any threshold $\varepsilon < 1$ represents a grouping criterion that is difficult to attain with accuracy. Indeed, we verified that misclassification was caused mostly by false exclusion from groups of nodes that terminated smaller variance links.

Algorithm accuracy decreases for larger branching ratio; see Figure 9(a), which 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. The difference is evident for smaller value of ε because of the higher probability of falsely excluding a node from a group.

Dependence on Loss. As described in Section 3.4 packet loss increases estimator variance, and hence decreases inference accuracy. This is evident in Figure 9(b) which displays 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 $\varepsilon = 0.75$.



(a) (b)

Figure 9: DEPENDENCE OF THE ACCURACY ON TOPOLOGY AND LOSS. Fraction of trees correctly classified by $\mathsf{DT}(\varepsilon)$ in 1000 simulations over randomly generated 15 nodes: maximum fanout 3 and 4 (a); different loss rate intervals (b).

7 Topology Misclassification

We analyze the modes of failure of DT, and estimate the convergence rates for the probability of successful 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 $k \in V \setminus R_{\mathcal{T}}$ 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 need to consider more general convex combinations of the delay covariances than those expressed by (24) to take into account groups of nodes which may result from nodes misgrouping. Given two disjoint subsets of R, S_1 and S_2, $S_1, S_2 \neq \emptyset$, we denote

$$V_{S_1,S_2}(\mu, \widehat{s}) := \sum_{\{i,j\} \in S_1 \times S_2} \mu_{ij} \widehat{s}_{ij} \tag{25}$$

where μ is any suitable covariance aggregator. Properties similar to those established in Section 5.2 hold for these convex combinations. In particular, $\sqrt{n}(V_{S_1,S_2}(\mu^*(\widehat{C}(S_1 \times S_2)), \widehat{s}) - V_{S_1,S_2}(\mu^*(C(S_1 \times S_2)), s)$, converge 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 $C(S_1 \times S_2) = [C_{(ij)(\ell m)}]_{\{i,j\},\{\ell,m\} \in S_1 \times S_2}$.

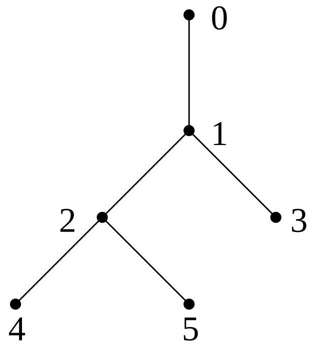


Figure 10: THE THREE-RECEIVER BINARY TREE.

7.1 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

$$\widehat{D}_k(S_1, S_2, S_3) := V_{S_1,S_2}(\mu^*, \widehat{s}) - V_{S_1,S_3}(\mu^*, \widehat{s}) > 0 \tag{26}$$

for all $(S_1, S_2, S_3) \in \mathcal{S}(k)$ where $\mathcal{S}(k) = \{S_1, S_2 \subset R(k), S_3 \subseteq R \setminus R(k), S_k \neq \emptyset, S_1 \cap S_2 = \emptyset\}$. (26) 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, this ensures that in $\widehat{\mathcal{T}}$ there is a node $\widehat{k}$ such that $R_{\mathcal{T}}(k) = R_{\widehat{\mathcal{T}}}(k)$. Let $Q_k(S_1, S_2, S_3)$ denote the event that (26) holds; then, $G_k \supseteq \cap_{(S_1,S_2,S_3)\in\mathcal{S}(k)} Q_k(S_1, S_2, S_3)$. This provides the following upper bound for the misgrouping probability, denoted by P_k^f, as

$$P_k^f := \mathsf{P}[G_k^c] \leq \sum_{(S_1,S_2,S_3)\in\mathcal{S}(k)} \mathsf{P}[Q_k^c(S_1, S_2, S_3)] \tag{27}$$

Normal Approximations. We now consider the asymptotic behavior of P_k^f for large numbers of probes.

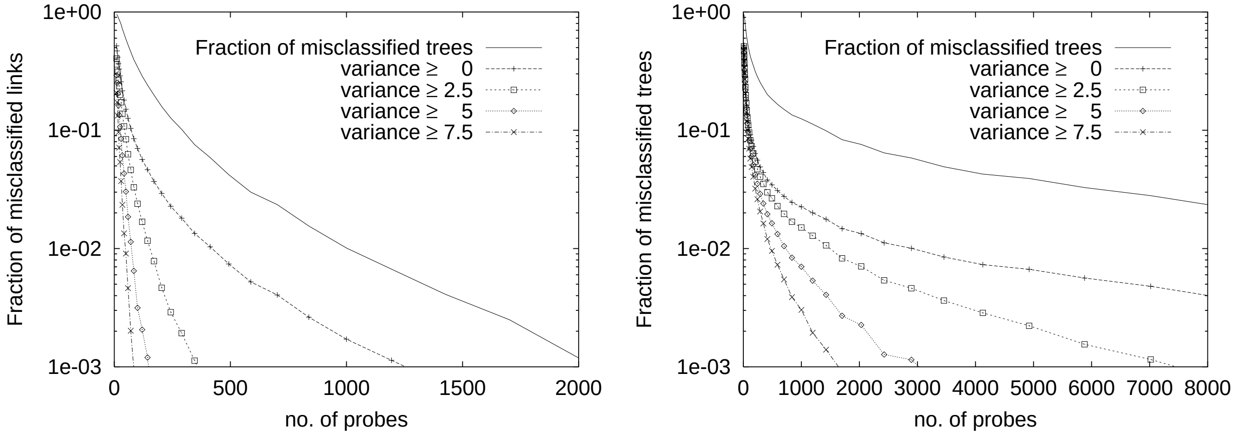
Theorem 11 *Let $(\mathcal{T}, r)$ a canonical delay-variance tree. For each $k \in V \setminus R$, $\sqrt{n}(\widehat{D}_k(S_1, S_2, S_3) - D_k(S_1, S_2, S_3))$, $(S_1, S_2, S_3) \in \mathcal{S}(k)$, converges in distribution, as the number of probes $n \to \infty$, to a Gaussian random variable with mean 0 and variance*

$$\sigma^2_{D_k}(S_1, S_2, S_3) = \sum_{\{i,j\},\{\ell,m\}\in S_1\times S_2\cup S_1\times S_3} \frac{\partial D_k(S_1, S_2, S_3)}{\partial s_{ij}} C_{(ij),(\ell m)} \frac{\partial D_k(S_1, S_2, S_3)}{\partial s_{\ell m}}, \tag{28}$$

where $D_k(S_1, S_2, S_3) = V_{S_1,S_2}(\mu^*, s) - V_{S_1,S_3}(\mu^*, s)$. Moreover, $\inf_{(S_1,S_2,S_3)\in\mathcal{S}(k)} D_k(S_1, S_2, S_3) = r_k$.

Theorem 11 suggests that we can approximate $\mathsf{P}[Q_k^c(S_1, S_2, S_3)] = \mathsf{P}[\widehat{D}_k(S_1, S_2, S_3) < 0]$ by $\Psi(-\sqrt{n}\cdot D_k(S_1, S_2, S_3)/\sigma_{D_k}(S_1, S_2, S_3))$, where Ψ is the cdf of a standard normal distribution. For large n, we can approximate to leading exponential order as

$$\mathsf{P}[Q_k^c(S_1, S_2, S_3)] \approx e^{-(n/2)\frac{D_k(S_1,S_2,S_3)^2}{\sigma^2_{D_k}(S_1,S_2,S_3)}}. \tag{29}$$



(a) Variance in $[1, 10]$

(b) Variance in $[1, 100]$

Figure 11: MISCLASSIFICATION AND MISGROUPING IN BDT. Fraction of links misclassified with variance $\geq \phi$, for $\phi = 0, 2.5, 5, 7.5\%$. Link variance is uniform in $[1, 10]$ (a) and in $[1, 100]$ (b).

Since the largest term over $\mathcal{S}(i)$ should dominate all others for large n, we have

$$\mathsf{P}^f_k \approx e^{-(n/2)\inf_{(S_1,S_2,S_3)\in\mathcal{S}(k)} \frac{D_k(S_1,S_2,S_3)^2}{\sigma^2_{D_k}(S_1,S_2,S_3)}}. \tag{30}$$

In the case of binary trees, when all groups are correctly formed so is the topology; therefore, we have that $P^f_{\mathsf{BDT}} \leq \sum_{k\in V\setminus R} P^f_k \approx \max_{k\in V\setminus R} P^f_k$ which suggest that $\log \mathsf{P}^f_{\mathsf{BDT}}$ vs. n is asymptotically linear with slope

$$\frac{1}{2} \inf_{k\in V\setminus R} \inf_{(S_1,S_2,S_3)\in\mathcal{S}(k)} \frac{D_k(S_1,S_2,S_3)^2}{\sigma^2_{D_k}(S_1,S_2,S_3)} \tag{31}$$

Example: The Three-Receiver Binary Tree. To illustrate the results, we consider the simplest case of a binary with three receivers in Figure 10. The topology is correctly inferred by BDT when nodes 4 and 5 are grouped which happens when both $\widehat{s}_{45} > \widehat{s}_{43}$ and $\widehat{s}_{54} > \widehat{s}_{53}$. Misclassification requires either inequality to be false. Consider the first; we have that $\mathsf{P}[\widehat{s}_{45} \leq \widehat{s}_{43}] \approx e^{-(1/2)\frac{(s_{45}-s_{43})^2}{\mathsf{Var}[\widehat{s}_{45}-\widehat{s}_{43}]}}$, where $\mathsf{Var}[\widehat{s}_{45} - \widehat{s}_{43}] = \mathsf{Var}[\widehat{s}_{45}] + \mathsf{Var}[\widehat{s}_{43}] - 2\,\mathsf{Cov}[\widehat{s}_{45}, \widehat{s}_{43}] = (C_{(45),(45)} + C_{(43),(43)} - 2C_{(42),(43)})/n$. Then by expanding the terms $C_{(ij),(\ell m)} = K^4(X_{i\vee j\vee\ell\vee m}) + s_{i\vee m}s_{j\vee\ell} + s_{i\vee\ell}s_{j\vee m}$, we readily obtain $\mathsf{P}[\widehat{s}_{45} \leq \widehat{s}_{43}] \approx e^{-(n/2)\frac{r_2^2}{K^4(D_1)+r_2^2+(r_1+r_2+r_4)(r_2+r_3+r_5)}}$ Similarly, $\mathsf{P}[\widehat{s}_{54} \leq \widehat{s}_{53}] \approx e^{-(n/2)\frac{r_2^2}{K^4(D_1)+r_2^2+(r_1+r_2+r_5)(r_2+r_3+r_4)}}$, which yields

$$\mathsf{P}^f_{\mathsf{BDT}} \approx e^{-(n/2)\frac{r_2^2}{K^4(D_1)+r_2^2+\max\{(r_1+r_2+r_4)(r_2+r_3+r_5),(r_1+r_2+r_5)(r_2+r_3+r_4)\}}}.$$

To verify the accuracy of this approximation, we conducted 10000 experiments with all link delays exponentially distributed: link variance was 10 for all links but link 2 the variance of which was 1. We obtained an approximate slope of 4.84×10^{-4} in good agreement with the experimental value of 4.96×10^{-4}.

Modes of Misclassification by BDT in Experiments. Calculation of the infimum in (31) is in general quite difficult since $\sigma^2_{D_k}(S_1, S_2, S_3)$ is a complex function of both the topology and the links variances. Here we use experience from experiments to identify the dominant modes of misclassification and misgrouping. For the binary trees used in Section 6, we plot in Figure 11 the proportion of links that had variances greater than or equal to a given threshold ϕ and were still misclassified by BDT, along with 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, $\mathsf{P}^f_{\mathsf{BDT}} \approx \mathsf{P}^f_j$, where $j = \text{argmin}_{k \in V \setminus R} r_k$, i.e., the most likely way to misclassify a tree is by not correctly grouping receivers that share the link with smallest variance. Moreover, we found out in our experiments that as the number of probes increases, the most likely way to misgroup such link occurs by mistakenly pairing one of its child nodes with its sibling, i.e., when $S_1 = R(h(j))$, $S_2 = R(h^*(j))$ and $S_3 = R(j^*)$. This suggests that we can use the following approximation

$$\mathsf{P}^f_{\mathsf{BDT}} \approx \mathsf{P}^f_j \approx e^{-(n/2)\frac{r_j^2}{\sigma^2_{D_k}(R(h(j)),R(h^*(j)),R(j^*))}}. \tag{32}$$

7.2 Misgrouping and Misclassification by $\mathsf{DT}(\varepsilon)$

We now turn our attention to the errors in classifying general trees by $\mathsf{DT}(\varepsilon)$. In the following, we let $(\widehat{\mathcal{T}}', \widehat{r}')$ denote the tree produced by BDT, the final estimate $\widehat{\mathcal{T}}$ is obtained from $\widehat{\mathcal{T}}'$ by pruning links whose inferred delay variance is smaller than ε, i.e., $(\widehat{\mathcal{T}}, r) = \mathsf{TP}(\varepsilon)(\widehat{\mathcal{T}}', \widehat{r}')$. In distinction with the binary case, incorrect grouping by BDT is sufficient but not necessary for the misclassification. For $\mathsf{DT}(\varepsilon)$, incorrect classification occurs in any of the following holds:

(i) at least one node in $\mathcal{T}$ is misclassified in $\widehat{\mathcal{T}}'$; or

(ii) $\mathsf{TP}(\varepsilon)$ prunes links from $\widehat{\mathcal{T}}'$ that are present in $\mathcal{T}$; or

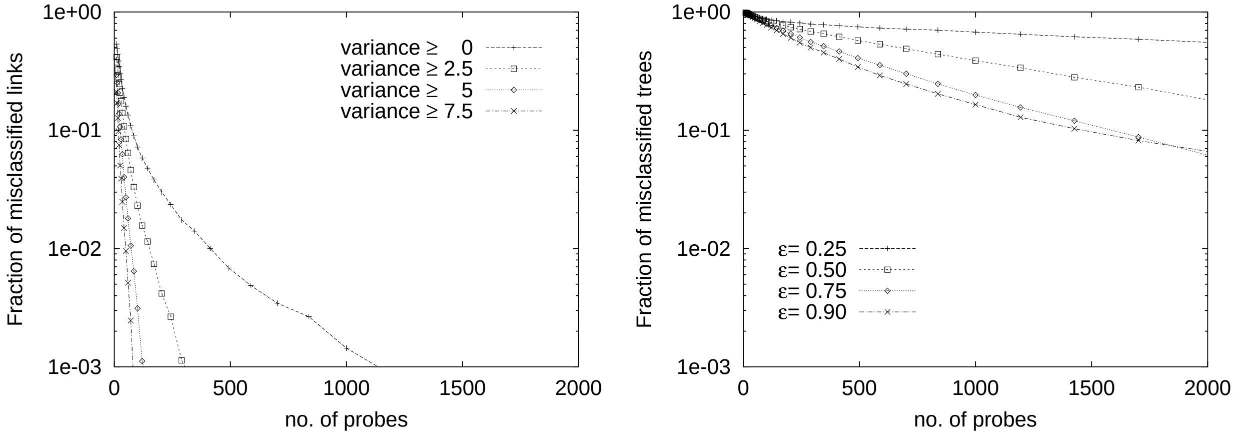
(iii) $\mathsf{TP}(\varepsilon)$ fails to prune links from $\widehat{\mathcal{T}}'$ that are not present in $\mathcal{T}$.

We now analyze misclassification in $\mathsf{DT}(\varepsilon)$. Let G denote the event that the topology is correctly classified. We have that $G \supseteq \cap_{k \in V \setminus R}(G_k \cap H_k(\varepsilon) \cap K_k(\varepsilon))$, where $H_k(\varepsilon) = \cap_{(S_1,S_2,S_3) \in \mathcal{S}(k)} H(S_1, S_2, S_3, \varepsilon)$, and for $(S_1, S_2, S_3) \in \mathcal{S}(k)$, $H(S_1, S_2, S_3, \varepsilon)$ is the event that

$$\widehat{E}_k(S_1, S_2, S_3) := V_{S_1,S_2}(\mu^*, \widehat{s}) - V_{S_1 \cup S_2, S_3}(\mu^*, \widehat{s}) > \varepsilon, \tag{33}$$

and $K_k(\varepsilon) = \cap_{S_1,S_2,S_3 \in \mathcal{K}(k)} H^c_k(S_1, S_2, S_3, \varepsilon)$ where $\mathcal{K}(k) = \{S_1, S_2, S_3 \subset R(k) : S_i \neq \emptyset; S_i \cap S_j = \emptyset, l \vee m = k, l \in S_i, m \in S_j, i \neq j\}$. When G_k holds, $H_k(\varepsilon)$ ensures that for all possible ways to reconstruct the tree, the inferred loss rate on link $\widehat{k}$ is larger than ε; $\cap_{k \in V \setminus R} K_k(\varepsilon)$ ensures that all the links in $\mathcal{T}'$ which are not present in $\mathcal{T}$ have inferred link variance smaller than ε. Thus, we obtain the following upper bound on the misclassification probability

$$P^f_{\mathsf{DT}(\varepsilon)} \leq \sum_{i \in V \setminus R} \left(\sum_{(S_1,S_2,S_3) \in \mathcal{S}(i)} \mathsf{P}[Q^c_i(S_1, S_2, S_3)] + \mathsf{P}[H^c_i(S_1, S_2, S_3, \varepsilon)] + \sum_{(S_1,S_2,S_3) \in \mathcal{K}(i)} \mathsf{P}[H^c_i(S_1, S_2, S_3, \varepsilon)] \right) \tag{34}$$



(a) (b)

Figure 12: MISGROUPING AND MISCLASSIFICATION IN $\mathsf{DT}(\varepsilon)$. Link variance distributed in $[1, 10]$. Fraction of misclassified links (a) and trees (b).

Normal Approximations. We now consider the asymptotic behavior of $\mathsf{P}^{f}_{\mathsf{DT}(\varepsilon)}$. The proof of the following result is similar to that of Theorem 11; it is omitted.

Theorem 12 *Let $(\mathcal{T}, r)$ be a canonical delay-variance tree. For each $k \in V \setminus R$, and $(S_1, S_2, S_3) \in \mathcal{S}(k) \cup \mathcal{K}(k)$, $\sqrt{n} \cdot (\widehat{E}_k(S_1, S_2, S_3) - E_k(S_1, S_2, S_3))$, converges in distribution, as the number of probes $n \to \infty$, to a Gaussian random variable with mean 0 and variance $\sigma^2_{E_k}(S_1, S_2, S_3)$. Moreover $\inf_{(S_1,S_2,S_3)\in\mathcal{S}(k)} E_k(S_1, S_2, S_3) = r_k$, and $E_k(S_1, S_2, S_3) = 0$ for $(S_1, S_2, S_3) \in \mathcal{K}(k)$.*

Theorem 12 suggests that for large n, we can use the following approximations

$$\begin{aligned} \mathsf{P}[H^c_k(S_1, S_2, S_3, \varepsilon)] &\approx e^{-(n/2)\frac{(E_k(S_1,S_2,S_3)-\varepsilon)^2}{\sigma^2_{E_k}(S_1,S_2,S_3)}} && (S_1, S_2, S_3) \in \mathcal{S}(k) \\ \mathsf{P}[H^c_k(S_1, S_2, S_3, \varepsilon)] &\approx e^{-(n/2)\frac{\varepsilon^2}{\sigma^2_{E_k}(S_1,S_2,S_3)}} && (S_1, S_2, S_3) \in \mathcal{K}(k) \end{aligned} \tag{35}$$

Since the largest term should dominate for large n, we expect the curve of $\log \mathsf{P}_{\mathsf{DT}(\varepsilon)}$ vs. n be asymptotically linear with negative slope

$$\frac{1}{2} \inf_{k\in V\setminus R} \left\{ \inf_{(S_1,S_2,S_3)\in\mathcal{S}(k)} \frac{D_k(S_1, S_2, S_3)^2}{\sigma^2_{D_k}(S_1, S_2, S_3)}, \inf_{(S_1,S_2,S_3)\in\mathcal{S}(k)} \frac{(E_k(S_1, S_2, S_3) - \varepsilon)^2}{\sigma^2_{E_k}(S_1, S_2, S_3)}, \inf_{(S_1,S_2,S_3)\in\mathcal{K}(k)} \frac{\varepsilon^2}{\sigma^2_{E_k}(S_1, S_2, S_3)} \right\} \tag{36}$$

The three possible dominating exponents in (36) correspond to the three possible modes of misclassification (i), (ii) and (iii) respectively, as listed above near the start of Section 7.2.

Modes of Misclassification by DT in Experiments. For the examples of Section 6 with link variances random in $[1, 10]$, Figure 12(a) displays the fraction of links that were misclassified by $\mathsf{DT}(0)$ and had link variance larger than a given value ϕ; Figure 12(b) displays the fraction of trees misclassified for $\varepsilon = 0.25, 0.5, 0.75, 0.9$. The difference in slopes between the two set of curves shows that for large numbers of probes, receivers become correctly grouped, leaving errors in tree misclassification to be dominated by pruning errors. Eq. (36) then indicates that the most likely way to misclassify under $\mathsf{DT}(\varepsilon)$ is for smaller ε by not completely grouping a set of sibling and, for larger ε, by pruning link with small link variances. Accuracy is best for intermediate value when both types of error have the same probability.

For larger delay variances in $[1, 100]$, misclassification is always dominated by node exclusion. In this case, from (36) we expect that for large n, $\log \mathsf{P}_{\mathsf{DT}(\varepsilon)} \approx -(n/2) \inf_{k \in V \setminus R} \inf_{(S_1,S_2,S_3)\in\mathcal{K}(k)} \frac{\varepsilon^2}{\sigma^2_{E_k}(S_1,S_2,S_3)}$ and hence that the ratio $\log \mathsf{P}_{\mathsf{DT}(\varepsilon')}/\log \mathsf{P}_{\mathsf{DT}(\varepsilon)} \approx \varepsilon'^2/\varepsilon^2$. To test this, we compared the experimental and theoretical ratios. For the pairs $(\varepsilon, \varepsilon') = (0.5, 0.75)$ and $(0.75, 0.9)$, and obtained the values of 0.4712 and 0.7144 in good agreement with the theoretical values which are 0.444 and 0.6944.

8 Conclusions

In this paper we have 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 that 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 located 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) [12] 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.

Acknowledgement

We thank Joseph Horowitz and Don Towsley for useful comments and suggestions.

9 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 Corollary 1.2.18 in [20]

(ii) Since the μ_{ij} sum to 1, the proof follows by considering the constrained minimization of $\mu \cdot C(k) \cdot \mu - 2k\mu \cdot \mathbf{1}$ with Lagrange multiplier k. 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 $\mu = kC(k)^{-1} \cdot \mathbf{1}$. This yields $\mu^*(C(k))$ upon normalization. The corresponding minimal asymptotic variance is $\mu^*(C(k)) \cdot C(k) \cdot \mu^*(C(k)) = \left(\mathbf{1} \cdot C(k)^{-1} \cdot \mathbf{1}\right)^{-1}$. ∎

Proof of Theorem 2: Comparing with (9), then 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. [30]), $\sqrt{n}\ (V(\mu^*(\widehat{C}(k)), \widehat{s}) - s_k)$ converges to a Gaussian random variable with mean 0 and variance $\alpha \cdot C^0(k) \cdot \alpha$, where for $(\ell, m) \in Q^0(k)$ (recalling the definitions of C^0 and Q^0 from below (9)),

$$\alpha_{\ell m} = \frac{\partial}{\partial s_{\ell m}} \sum_{\{i,j\} \in Q(k)} \mu^*_{ij}(C(k)) s_{ij}. \tag{37}$$

Differentiating,

$$\alpha_{\ell m} \;=\; \mu^*_{\ell m}(C(k)) \chi_{Q(k)}(\{\ell, m\}) + \sum_{\{i,j\} \in Q(k)} s_{ij} \frac{\partial}{\partial s_{\ell m}} \mu^*_{ij}(C(k)), \tag{38}$$

where $\chi_{Q(k)}$ denotes the indicator function of the set $Q(k)$. But $s_{ij} = s_{i \vee j} = s_k$ for $\{i,j\}$ in $Q(k)$ and so is constant in the sum. Since the μ^*_{ij} sum to 1, the sum in (38) is zero. Hence $\alpha \cdot C^0(k) \cdot \alpha = \mu^*(C(k)) \cdot C(k) \cdot \mu^*(C(k))$. ∎

Proof of Theorem 4: Convergence to some random Gaussian random variable in (i) is immediate from Corollary 1.2.18 in [20]. 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 $\mathsf{E}[X_i] = 0$. We analyze $\mathsf{Cov}(\widehat{v}_{ij}, \widehat{v}_{\ell m})$ by expanding $\sum_{m \in I_n(\{i,j\})}$ in (15) 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,

$$\mathsf{Cov}(\widehat{v}_{ij}, \widehat{v}_{\ell m}) = \frac{1}{N_n(\{i,j\}) N_n(\{\ell, m\})} \left(N_n(\{i,j,\ell,m\}) \mathsf{Cov}(X_i X_j, X_\ell X_m) + O(1) \right). \tag{39}$$

Since $n^{-1} N_n(\{i_1, \ldots, i_p\})$ converges as $n \to \infty$ to $B(i_1, \ldots i_p)$, the distribution of $\sqrt{n}\ \widehat{v}_{ij}$ has the stated property. The proofs of (ii) and (iii) are analogous to those of Theorems 1 and 2. ∎

The proof of Theorem 5 uses the following supplementary result.

Lemma 1 *Let M be an $m \times m$ matrix, and $M(t)$ the matrix with elements $M_{ij}(t) = M_{ij} + t$. Let $T = \{t \in \mathbb{R} \mid \det M(t) \neq 0\}$. Then for each $1 \leq i \leq m$, $\det(M(t)) \sum_j (M^{-1}(t))_{ij}$ is independent of $t \in T$.*

Proof of Lemma 1: With no loss of generality, take the case $i = 1$. $\det(M(t)) \sum_j (M^{-1}(t))_{1j}$ is equal to the determinant of the matrix

$$M' = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ M_{21} + t & M_{22} + t & \cdots & M_{2m} + t \\ \vdots & \vdots & \ddots & \vdots \\ M_{m1} + t & M_{m2} + t & \cdots & M_{mm} + t \end{pmatrix}. \tag{40}$$

The matrix M'' formed by subtracting t times the first row of M' from all other rows has the same determinant as M'. But M'' does not depend on t, so neither does $\det M' = \det M''$. ∎

Proof of Theorem 5: For $\{i,j\}$ and $\{\ell,m\}$ in $Q(k)$, (9) can be written as in terms of cumulants as

$$C_{(ij),(\ell m)} = K^{(1,1,1,1)}(X_i,X_j,X_\ell,X_m)+K^{(1,1)}(X_i,X_m)K^{(1,1)}(X_j,X_\ell)+K^{(1,1)}(X_i,X_\ell)K^{(1,1)}(X_j,X_m). \tag{41}$$

Let $\{h,h^*\}$ denote the set $d(k)$ of children of k, and without loss of generality assume $i,\ell \in R(h)$ and $j,m \in R(h^*)$. Setting $A_{i\ell} = X_{i\vee\ell} - X_k$ and $A'_{i\ell} = X_i - X_{i\vee\ell}$, the first term of (41) is $K^{(1,1,1,1)}(X_i,X_j,X_\ell,X_m) = K^{(1,1,1,1)}(X_k + A_{i\ell} + A'_{i\ell}, X_k + A_{jm} + A'_{jm}, X_k + A_{i\ell} + A'_{\ell i}, X_k + A_{jm} + A'_{mj})$. By the assumption of independence of per links delays, distinct X, A and A' are mutually independent, and hence this reduces to $K^{(1,1,1,1)}(X_k + A_{i\ell}, X_k + A_{jm}, X_k + A_{i\ell}, X_k + A_{jm}) = K^{(2,2)}(X_k + A_{i\ell}, X_k + A_{jm}) = K^4(X_k)$. The second and third terms of (9) are together equal to $s_{i\vee m}\ s_{j\vee\ell} + s_{i\vee\ell}\ s_{j\vee m}$ and since $s_{i\vee m} = s_{j\vee\ell} = s_k$ the first two terms of (41) depend only on $k = i\vee j\vee\ell\vee m$. We can rewrite (41) as $C = (K^4(X_k)+s_k^2)U_h\otimes U_{h^*} + S_h\otimes S_{h^*}$, where the first component of the tensor product carries the indices i,ℓ, the second j,m. By Lemma 1, $\sum_{\ell m} C^{-1}_{(ij),(\ell m)}$ is equal to a $\det(C-(K^4(X_k)+s_k^2)U_h\otimes U_{h^*}))/\det(C)$ times

$$\sum_{\ell\in R(h), m\in R(h^*)} (S_h\otimes S_{h^*})^{-1}_{(ij),(\ell m)} = \sum_{\ell\in R(h)} (S_h)^{-1}_{i\ell} \sum_{m\in R(h^*)} (S_{h^*})^{-1}_{jm} \tag{42}$$

Note for any $k \in V$ with offspring $\{h,h^*\}$ that $(S_k)_{ij} = s_k$ if $i\in R(h)$ (resp. $R(h^*)$) and $j \in R(h^*)$ (resp. $R(h)$). Thus $(S_k - s_kU_k)$ has a block diagonal structure

$$(S_k - s_kU_k) = (S_h - s_kU_h)\oplus(S_{h^*} - s_kU_{h^*}) \tag{43}$$

and hence, when $i\in R(h)$,

$$\sum_{\ell\in R(k)} (S_k - s_kU_k)^{-1}_{i\ell} = \sum_{\ell\in R(h)} (S_h - s_kU_h)^{-1}_{i\ell}, \tag{44}$$

because $(S_k - s_kU_k)_{i\ell} = 0$ when $i\in R(h)$ and $\ell\in R(h^*)$. When q_i is such that $f^{q_i+1}(i) = k$, and writing $j_r = f^r(j)$, then repeated use of Lemma 1 gives

$$\sum_{\ell\in R(h)} (S_h)^{-1}_{i\ell} = \frac{1}{\det(S_h)}\prod_{r=1}^{q}\frac{\det(S_{j_r} - s_{j_r}U_{j_r})}{\det(S_{j_{r-1}} - s_{j_r}U_{j_{r-1}})} = \frac{1}{\det(S_h)}\prod_{r=0}^{q-1}\delta(f^r(i)^*). \tag{45}$$

Here we used the factoring property $\det(S_k - s_kU_k) = \det(S_h - s_kU_h)\det(S_{h^*} - s_kU_{h^*})$ that follows from the form (43). The locality on μ and form of ψ then follows from the fact that the RHS of (45) is a product over the nodes $i, f(i), f^2(i),\ldots,f^{q-1}(i), f^q(i) = h$. ∎

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 descendents 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 $s_{t_1 t_2} > s_w > s_{u\vee v} = s_{uv}$, 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 for the following reasons:

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 (i), 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 $s_{u_1 u_2}$.
3. *If not all members are $d(k)$ are included in V' during the execution of the outer loop.* From line 8 of Figure 4 we have $s_{U^{(m-1)}u_{m+1}} = s_{U^{(m-2)}u_{m+1}} = \ldots = s_{U^{(1)}u_{m+1}} = s_k$. Hence each u_m enters into V' during execution of the outer loop. This also show that $r_{U^{(m)}} = 0$ for $m = 1, \ldots n-2$ and hence that all links $(U^{(m+1)}, U^{(m)})$, $m = 1, \ldots, n-2$ are pruned by TP(0).
4. *If a non-child 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 $s_{k_1} = \ldots = s_{k_m}$ for $k_1, \ldots, k_m \in R'$, 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', etc. ■

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 the maximizing U is unique. Since the delay variance tree is canonical, $s_{k\ell} > s_{k'\ell'}$ for any other candidate binary set $U' = \{k', \ell'\}$. by the convergence property of Theorem 8, $\mathsf{P}[V_{k,\ell}(\mu, \widehat{s}) > V_{k',\ell'}(\mu', \widehat{s})] \to 1$ as $n \to \infty$, and hence $\lim_{n\to\infty} P^f_{\mathsf{BDT}} = 0$.

If the minimizing U is not unique, then there is a set $\mathcal{U}$ of pairs $U_{(j)} = \{k_{(j)}, \ell_{(j)}\}$, $j = 1, \ldots m$ (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 Figure 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. But 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 $\lim_{n\to\infty} P^f_{\mathsf{BDT}} = 0$. ■

Proof of Theorem 11: Convergence to a Gaussian random variable follows from the asymptotic normality of each term. The expression for the variance then follows from a direct 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, since for any $\{i, j\} \in S_1 \times S_3$ $s_{i\vee j} \leq s_{f(k)}$ we have that $V_{S_1,S_3}(\mu^*, s) \leq s_{f(k)}$. 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 $V_{S_1,S_3}(\mu, s) = s_{f(k)}$. ■

References

[1] J. Bolot, "Characterizing End-to-End Packet Delay and Loss in the Internet." *Journal of High-Speed Network*, vol. 2 n. 3, pp. 289-298, Dec. 1993.

[2] J. Bolot and A. Vega Garcia "The case for FEC-based error control for packet audio in the Internet" *to appear in ACM Multimedia Systems*.

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

[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" *Proc. IEEE Infocom '99*, New York, NY, Mar. 1999.

[5] R. Caceres, N.G. Duffield, J .Horowitz, F. Lo Presti and D. Towsley, "Loss-Based Inference of Multicast Network Topology" IEEE Conference on Decision and Control, Phoenix, AZ, Dec. 1999.

[6] CAIDA: Cooperative Association for Internet Data Analysis. For more information see http://www.caida.org

[7] K. Claffy, G. Polyzos and H-W. Braun, "Measurements Considerations for Assessing Unidirectional Latencies", *Internetworking: Research and Experience*, Vol. 4, no. 3, pp. 121-132, Sept. 1993.

[8] R. L. Carter and M. E. Crovella, "Measuring Bottleneck Link Speed in Packet-Switched Networks," *PERFORMANCE '96*, Oct. 1996.

[9] A. Downey, "Using pathchar to estimate Internet link characteristics", Proc. ACM SIGCOMM'99, Cambridge, MA.

[10] N.G. Duffield and F. Lo Presti, "Multicast Inference of Packet Delay Variance at Interior Network Links", Proceedings IEEE Infocom 2000, Tel Aviv, March 2000.

[11] Felix: Independent Monitoring for Network Survivability. For more information see ftp://ftp.bellcore.com/pub/mwg/felix/index.html

[12] S. Floyd and V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance," *IEEE/ACM Transactions on Networking*, Vol. 1, no. 4, August 1993.

[13] IPMA: Internet Performance Measurement and Analysis. For more information see http://www.merit.edu/ipma

[14] V. Jacobson, Pathchar - A Tool to Infer Characteristics of Internet paths. For more information see ftp://ftp.ee.lbl.gov/pathchar

[15] V. Paxson, J. Mahdavi, A. Adams, M. Mathis, "An Architecture for Large-Scale Internet Measurement," *IEEE Communications*, Vol. 36, No. 8, pp. 48-54, August 1998.

[16] M. Mathis and J. Mahdavi, "Diagnosing Internet Congestion with a Transport Layer Performance Tool," *Proc. INET '96*, Montreal, June 1996.

[17] D. Mills, "Network Time Protocol (Version 3): Specification, Implementation and Analysis", *RFC 1305*, Network Information Center, SRI International, Menlo Park, CA, Mar. 1992.

[18] S. Moon, P. Skelly and D. Towsley, "Estimation and Removal of Clock Skew from Network Delay Measurements" *Proc. IEEE Infocom '99*, New York, NY, Mar. 1999.

[19] mtrace – Print multicast path from a source to a receiver. For more information see ftp://ftp.parc.xerox.com/pub/net-research/ipmulti

[20] R.J. Muirhead, "Aspects of Multivariate Statistical Theory", Wiley, New York, 1982.

[21] A. Mukherjee, "On the Dynamics and Significance of Low Frequency Components of Internet Load", *Internetworking: Research and Experience*, Vol. 5, pp. 163-205, Dec. 1994.

[22] S. Paul, et al. "Reliable multicast transport protocol (RMTP)", *IEEE JSAC*, Vol. 15, No. 3, pp. 407–421, April 1997.

[23] ns – Network Simulator. For more information see http://www-mash.cs.berkeley.edu/ns/ns.html

[24] V. Paxson, "End-to-End Routing Behavior in the Internet," *Proc. ACM SIGCOMM '96*, Stanford, Aug. 1996.

[25] V. Paxson, "End-to-End Internet Packet Dynamics," *Proc. ACM SIGCOMM 1997*, Cannes, France, pp. 139-152, Sept. 1997.

[26] V. Paxson, "Measurements and Analysis of End-to-End Internet Dynamics," Ph.D. Dissertation, University of California, Berkeley, Apr. 1997.

[27] V. Paxson, "Automated Packet Trace Analysis of TCP Implementations," *Proc. ACM SIGCOMM 1997*, Cannes, France, 167–179, Sept. 1997.

[28] V. Paxson, "On calibrating measurements of Packet Transit Times", *Proc. ACM SIGMETRICS '98*, Madison, June 1998.

[29] S. Ratnasamy and S. McCanne, "Inference of Multicast Routing Tree Topologies and Bottleneck Bandwidths using End-to-end Measurements", Proc. IEEE Infocom' 99, New York, NY, Mar. 1999.

[30] M.J. Schervish, "Theory of Statistics", Springer, New York, 1995.

[31] Surveyor. For more information see http://io.advanced.org/surveyor/