Multicast-Based Inference of Network-Internal Delay Distributions

Francesco Lo Presti, N. G. Duffield, *Senior Member, IEEE*, Joe Horowitz, and Don Towsley, *Fellow, IEEE*

***Abstract*—Packet delay greatly influences the overall performance of network applications. It is therefore important to identify causes and locations of delay performance degradation within a network. Existing techniques, largely based on end-to-end delay measurements of unicast traffic, are well suited to monitor and characterize the behavior of particular end-to-end paths. Within these approaches, however, it is not clear how to apportion the variable component of end-to-end delay as queueing delay at each link along a path. Moreover, there are issues of scalability for large networks.**

In this paper, we show how end-to-end measurements of multicast traffic can be used to infer the packet delay distribution and utilization on each link of a logical multicast tree. The idea, recently introduced in [3] and [4], is to exploit the inherent correlation between multicast observations to infer performance of paths between branch points in a tree spanning a multicast source and its receivers. The method does not depend on cooperation from intervening network elements; because of the bandwidth efficiency of multicast traffic, it is suitable for large-scale measurements of both end-to-end and internal network dynamics. We establish desirable statistical properties of the estimator, namely consistency and asymptotic normality. We evaluate the estimator through simulation and observe that it is robust with respect to moderate violations of the underlying model.

***Index Terms*—End-to-end measurements, estimation theory, multicast tree, network tomography, queueing delay.**

I. Introduction

A. Background and Motivation

MONITORING the performance of large communications networks is essential for diagnosing the causes of performance degradation. Two broad approaches to monitoring performance currently exist: the *internal* approach, where direct measurements are made at or between network elements, e.g., of packet loss or delay, and the *external* approach, where measurements are made across a network on end-to-end or edge-to-edge paths.

The internal approach has a number of potential limitations. Due to the commercial sensitivity of performance measurements and the potential load incurred by the measurement process, it is expected that measurement access to network elements will be limited to service providers and, possibly, selected peers and users. The internal approach assumes sufficient coverage, i.e., that measurements can be performed at all relevant elements on paths of interest. In practice, not all elements may possess the required functionality or such functionality may be disabled at heavily utilized elements in order reduce CPU load. On the other hand, arranging for complete coverage of larger networks raises issues of scale, both in gathering measurement data, and merging data collected from a large number of elements in order to form a composite view of end-to-end performance. Last, internal measurements usually focus on average behavior (e.g., throughput, average queue length) but not on other statistics.

This motivates the need for external approaches, where no assumption is made regarding 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 [8], [19], and [25]–[27]). Several research efforts are developing measurement infrastructures (Felix [12], IPMA [14], NIMI [18], and Surveyor [32]) with the aim of collecting and analyzing end-to-end measurements across a mesh of paths between a number of hosts. Standard diagnostic tools for IP networks, ping and traceroute, report round-trip loss and delay, the latter incrementally along the IP path by manipulating the time-to-live (TTL) field of probe packets. A recent refinement of this approach, pathchar [15], estimates hop-by-hop link capacities, packet delay, and loss rates. Pathchar is still under evaluation; initial experience indicates many packets are required for inference, leading to either high load of measurement traffic or long measurement intervals, although adaptive approaches can reduce this [9]. More broadly, measurement approaches based on TTL expiry require the cooperation of network elements in returning Internet Control Message Protocol (ICMP) messages. Finally, the success of active measurement approaches to performance diagnosis may itself cause increased congestion if intensive probing techniques are widely adopted.

In response to some of these concerns, a multicast-based approach to active measurement has been proposed recently in [3] and [4]. The idea behind the approach is that correlation in performance seen on *intersecting* end-to-end paths can be used to draw inferences about the performance characteristics of the common portion (the intersection) of the paths, without the cooperation of network elements on the path. Multicast traffic is particularly well suited for this since a given packet only occurs once on a given link in the (logical) multicast tree. Thus,

Manuscript received September 18, 1999; revised January 12, 2001; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor E. Zegura. This work was supported in part by DARPA and the Air Force Research Laboratory under agreement F30602-98-2-0238.

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

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

J. Horowitz is with the Mathematics and Statistics Department, University of Massachusetts, Amherst, MA 01003 USA (e-mail:joeh@math.umass.edu).

D. Towsley is with the Computer Science Department, University of Massachusetts, Amherst, MA 01003 USA (e-mail:towsley@cs.umass.edu).

Digital Object Identifier 10.1109/TNET.2002.805026

characteristics such as loss and end-to-end delay of a given multicast packet as seen at different endpoints are highly correlated. Another advantage of using multicast traffic is scalability. Suppose packets are exchanged on a mesh of paths between a collection of N measurement hosts stationed in a network. If the packets are unicast, then the load on the network may grow proportionally to N^2 in some parts of the network, depending on the topology. For multicast traffic, the load grows proportionally only to N.

B. Contribution

The work of [3] and [4] showed how multicast end-to-to measurements can be used to infer per-link loss rates in a logical multicast tree. In this paper we extend this approach to infer the probability distribution of the per-link variable delay. Thus, we are not concerned with propagation delay on a link, but rather the distribution of the additional variable delay that is attributable to either queuing in buffers or other processing in the router. A key part of the method is an analysis that relates the probabilities of certain events visible from end-to-end measurements (end-to-end delays) to the events of interest in the interior of the network (per-link delays). Once this relation is known, we can estimate the delay distribution on each link from the measured distributions of end-to-end delays of multicast packets.

For a glimpse of how the relations between end-to-end delay and per-link delays could be found, consider a multicast tree spanning a source of multicast probes (identified as the root of the tree) and a set of receivers (one at each leaf of the tree). We assume the packets are potentially subject to queuing delay and even loss at each link. Focus on a particular node k in the interior of the tree. If, for a given packet, the source-to-leaf delay does not exceed a given value on any leaf descended from k, then clearly the delay from the root to the node k was less than that value. The stated desired relation between the distributions of per-link and source-to-leaf delays is obtained by a careful enumeration of the different ways in which end-to-end delay can be split between the portion of the path above or below the node in question, together with the assumption that per-link delays are *independent* between different links and packets. We shall comment later upon the robustness of our method to violation of this independence assumption.

We model link delay by a nonparametric discrete distribution. The choice of a nonparametric distribution rather than a parameterized delay model is dictated by the lack of knowledge of the distribution of link delays in networks. While there is significant prior work on the analysis and characterization of end-to-end delay behavior (see [1], [23], and [26]), to the best of our knowledge there is no general model for per-link delays. The use of a nonparametric model provides the flexibility to capture broadly different delay distributions, albeit at the cost of increasing the number of quantities to estimate (i.e., the weights in the discrete distribution). Indeed, we believe that our inference technique can shed light on the behavior and dynamics of per-link delays and so provide useful results for analysis and modeling; this we will consider in future work.

The discrete distribution can be regarded as a binned or discretized version of the (possibly continuous) true delay distribution. Use of a discrete rather than a continuous distribution allows us to perform the calculations for inference using only algebra. Moreover, we can explicitly tradeoff the detail of the distribution with the cost of calculation; the cost is inversely proportional to the bin widths of the discrete distribution.

The principal results of the analysis are as follows. Based on the independent delay model, we derive an algorithm to estimate the per-link discrete delay distributions and utilization from the measured end-to-end delay distributions. We investigate the statistical properties of the estimator and show it to be strongly consistent, i.e., it converges, with probability 1, to the true distribution as the number of probes grows to infinity. We show that the estimator is asymptotically normal; this allows us to compute the rate of convergence of the estimator to its true value and to construct confidence intervals for the estimated distribution based on a given number of probes. This is important because the presence of large-scale routing fluctuation (e.g., as seen in the Internet; see [25]) sets a timescale within which measurement must be completed and hence limits the accuracy that can be obtained when sending probes at a given rate.

We evaluated our approach through extensive simulation. We first used model simulation in which packet delays obeyed the independence assumption of the model. We applied the inference algorithm to the end-to-end delays generated in the simulation and compared the results with the (true) model delay distribution. We verified the convergence to the model distribution, and also the rate of convergence, as the number of probes increased.

We then conducted network level simulation with ns [24]. Packet delays and losses were entirely due to queueing and packet discard mechanisms, rather than model driven. The bulk of the traffic in the simulations was background traffic due to TCP and UDP traffic sources; we compared the actual and predicted delay distributions for the probe traffic. Here we found rapid convergence, although with some persistent differences with respect to the actual distributions.

These differences appear to be caused by violation of the model due to the presence of spatial dependence (i.e., dependence between delays on different links). In our simulations we find that, when this type of dependence occurs, it is usually between the delays on child and parent links. However, it can extend to entire paths. As far as we know, there are no experimental results concerning the magnitude of such dependence in real networks.

We also verified the presence of temporal dependence, i.e., dependence between the delays between successive probes on the same link. This is to be expected from the phenomenology of queueing: when a node is idle, many consecutive probes can experience constant delay; during congestion, probes can experience the same delay if their interarrival time is smaller than the congestion timescale. This poses no difficulty as all that is required for consistency of the estimator is ergodicity of the delay process, a far weaker assumption than independence. However, dependence can decrease the rate of convergence of the estimators. In our experiments, inferred values closely tracked the actual ones despite the presence of temporal dependence.

C. Implementation and Requirements

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, multicast inference has been deployed on the NIMI measurement infrastructure comprising a number of hosts (approximately 50 as of mid–2001). Another approach to multicast measurement has been recently proposed in [2]. This approach, which does not require specialized end hosts, relies on the use the Real-Time Transport Protocol (RTP) and its control protocol (RTCP) [31]. RTP is used to carry multicast audio and video over the Internet. RTCP is used by the receivers to periodically multicast reports to the group for transmission control. The idea is to regard any ongoing RTP transmission, e.g.,, an RTP audio feed, as a measurement probe stream (observe, though, that the traffic characteristic may violate the model assumption) and to extend RTCP reports to include per-packet measurements. Then, any third-party host may, by joining the multicast group of ongoing RTCP sessions, monitor RTCP reports of these sessions, collect measurements, and perform inference. Thus, in practice, it would be possible to perform network measurements by exploiting existing Internet RTP traffic.

Since the data for delay inference comprises one-way packet delays, the primary requirement is the ability to measure the variable component of the one-way delay accurately. There are two aspects to this measurement. First, it is necessary to identify the variable component. This can be done provided that a sufficient number of packets incur the minimum one-way delay. Link utilizations are low enough that this appears to be generally true [28], [21]. Second, it is necessary to accurately measure one-way delays so as to be able to subtract the minimum delay. This is easily done if the measurement hosts are deployed with synchronized clocks. Global Positioning System (GPS) systems afford one way to achieve synchronization to within tenths of microseconds; it is currently used or planned in several of the measurement infrastructures mentioned earlier. More widely deployed is the Network Time Protocol (NTP) [20]. However, this provides accuracy only on the order of milliseconds at best, a resolution at least as coarse as the queueing delays in practice. Alternative approaches that can supplement delay measurement from unsynchronized or coarsely synchronized clocks have been developed in [28] and [21]. These authors propose algorithms to detect and clock mismatches and to calibrate the delay measurements.

Another requirement is knowledge of the multicast topology. There is a multicast-based measurement tool, mtrace [22], already in use in the Internet. mtrace reports the route from a multicast source to a receiver, along with other information about that path such as per-hop loss 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 as it needs to run once for each receiver to cover the entire multicast tree. In addition, mtrace relies on multicast routers responding to explicit measurement queries, a feature that can be administratively disabled. An alternative approach that is closely related to the work on multicast-based loss inference [3], [4] is to infer the logical multicast topology directly from measured probe statistics; see [29] and [5]. This method does not require cooperation from the network.

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 [6] and [11] for the inference of loss rates and [7] and [10] for delay distributions[1] . The premise is that unicast measurements could be used to complement multicast measurements for those portions of the network which do not support multicast. The approach is to design unicast measurement whose correlation properties closely resemble those of multicast traffic. By doing so, it is then possible to use the inference techniques developed for multicast inference such as those described in this paper as well as in [3]; the closer the correlation properties are to that of multicast traffic, the more accurate the results. The basic approach, which has been further refined in [11] for the estimation of the loss rates, is to dispatch two back-to-back packets (a packet pair) from a probe source to a pair of distinct receivers. The premise is that, when the duration of network congestion events exceeds the temporal width of the packets, packets experience very similar behavior when they traverse common portions of their paths. If the experience were identical, the two packets would indeed have the same statistical properties as a notional multicast packet that followed the same path.

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 [7]. The inference of the link delay distribution is formulated as a maximum likelihood estimation problem which is solved using the Expectation Maximization (EM) algorithm. The results have been extended in [10]. Preliminary results on these methods reported in these papers show promise, but reveal the inherent following limitations: (1) poor scalability because of the higher traffic load and (2) potential accuracy degradation since the imperfect correlation of unicast traffic results in estimators bias, the extent of which cannot be typically determined.

E. Structure of the Paper

The remaining sections of the paper are organized as follows. In Section II, we describe the delay model and in Section III we derive the delay estimator. In Section IV, we describe the algorithm used to compute the estimator from data. In Section V, we present the model and network simulations used to evaluate our approach. Section VI concludes the paper.

II. Model and Framework

A. Description of the Logical Multicast Tree

We identify the physical multicast tree as comprising actual network elements (the nodes) and the communication links that join them.

[1]These results on unicast measurements came after the work of this paper was submitted.

The logical multicast tree comprises the root, the leaves, the branch points of the physical tree (i.e., those nodes with two or more outgoing links), and "logical" links corresponding to the paths between them (within the physical tree). Thus, each node in the logical tree, except for the leaves and the root, must have two or more children. We can construct the logical tree from the physical tree by deleting all links with one child (except for the root) and adjusting the links accordingly by directly joining its parent and child.

Let $\mathcal{T} = (V, L)$ denote the *logical* multicast tree, consisting of the set of nodes V, including the source (root) and receivers, and the set of links L, which are ordered pairs (j, k) of nodes, indicating a link from j to k. The set of *children* of node j is denoted by $d(j)$; these are the nodes whose parent is j. Nodes are said to be siblings if they have the same parent. For each node j, other than the root, labeled 0, there is a unique node $f(j)$, the *parent* of j, such that $(f(j), j) \in L$. Each link can therefore be also identified by its "child" endpoint. We shall define $f^n(k)$ recursively as $f^n(k) = f(f^{n-1}(k))$ with $f^1 = f$. We say that j is a descendant of k if $k = f^n(j)$ for some integer $n > 0$, and write the corresponding partial order in V as $j \prec k$. For each node j, we define its *level* $\ell(j)$ to be the nonnegative integer such that $f^{\ell(j)}(j) = 0$. The root $0 \in V$ represents the source of the probes and the set of *leaf* nodes $R \subset V$ (i.e., those with no children) represents the receivers. We will let $U = V \setminus \{0\}$.

B. Modeling Delay and Loss of Probe Packets

Probe packets are sent down the tree from the root. Each probe that arrives at node k results in a copy being sent to every child of k. We associate with each node k a random variable D_k taking values in the extended positive real line $\mathbb{R}_+ \cup \{\infty\}$. By convention $D_0 = 0$. D_k is the random delay that would be encountered by a packet attempting to traverse the link $(f(k), k) \in L$. The value $D_k = \infty$ indicates that the packet is lost on the link. We assume that the D_k are independent. The cumulative delay experienced on the path from the root to a node k is $Z_k = \sum_{j \preceq k} D_j$.

We discretize each link delay D_k to a set $\{0, q, 2q, \ldots, i_{\max}q, \infty\}$. Here q is the bin width, $i_{\max} + 2$ is the number of bins, and the point ∞ is interpreted as "packet lost" or "encountered delay greater than $i_{\max}q$." We define the bin associated with iq to be the interval $[iq - q/2, iq + q/2)$, $i = 1, \ldots, i_{\max}$ and $[i_{\max}q - q/2, \infty)$ the one associated with the value ∞. Because delay is nonnegative, we associate with 0 the bin $[0, q/2)$.

The distribution of D_k is denoted by α_k, where $\alpha_k(i) = P[D_k = iq]$ with $\alpha_k(\infty)$ the probability that $D_k = \infty$. For each link, we denote u_k as the *link utilization*; then, $u_k = 1 - \alpha_k(0)$, the probability that a packet experiences delay or is lost in traversing link k.

For each $k \in V$, we also discretize the cumulative delay Z_k, $k \in V$, to the set $\{0, q, 2q, \ldots, i_{\max}q, \infty\}$. We set $A_k(i) = P[Z_k = iq]$ with $A_k(\infty)$ the probability that $Z_k = \infty$. Because of delay independence, for finite i, $A_k(i) = \sum_{j=0}^{i} \alpha_k(j) A_{f(k)}(i - j)$, $k \in U$; by convention $A_0(0) = 1$.

We consider only **canonical delay trees**. A delay tree consists of the pair $(\mathcal{T}, \alpha)$, $\mathcal{T} = (V, L)$, $\alpha = (\alpha_k(i))_{k \in U, i \in \{0, \ldots, i_{\max}\}}$. A delay tree is said to be *canonical* if $\alpha_k(0) > 0, \forall k \in U$, i.e., if there is a nonzero probability that a probe experiences no delay in traversing each link.

III. Delay Distribution Estimator and its Properties

Consider an experiment in which n probes are sent from the source node down the multicast tree. As result of the experiment, we collect the set of source-to-leaf delays $(Z_{k,m})_{k \in R, m=1,\ldots,n}$. Our goal is to infer the internal delay characteristics solely from the collected end-to-end measurements.

In this section, we state the main analytic results on which inference is based. In Section III-A, we establish the key property underpinning our delay distribution estimator, namely the one-to-one correspondence between the link delay distributions and the probabilities of a well-defined set of observable events. Applying this correspondence to measured leaf delays allows us to obtain an estimate of the link delay distribution. We show that the estimator is strongly consistent and asymptotically normal. In Section III-B, we present the proof of the main result which also provides the construction of the algorithm to compute the estimator we present in Section IV. In Section III-C, we analyze the rate of convergence of the estimator as the number of probes increases.

A. The Delay Distribution Estimator

Let $\mathcal{T}(k) = (V(k), L(k))$ denote the subtree rooted at node k and $R(k) = R \cap V(k)$ the set of receivers which descend from k. Let $\Omega_k(i)$ denote the event $\{\min_{j \in R(k)} Z_j \leq iq\}$ that the (discretized) end-to-end delay is no greater than iq for at least least one receiver in $R(k)$. Let $\gamma_k(i) = P[\Omega_k(i)]$ denote its probability. Finally, let Γ denote the mapping associating the link distributions $\alpha = (\alpha_k(i))_{k \in U, i \in \{0, \ldots, i_{\max}\}}$ to the probabilities of the events $\Omega_k(i)$, $\gamma = (\gamma_k(i))_{k \in U, i \in \{0, \ldots, i_{\max}\}}$. The proof of the next result is given in the following section.

Theorem 1: Let

$$\mathcal{A} = \left\{ \alpha = (\alpha_k(i))_{k \in U, i \in \{0, \ldots, i_{\max}\}} : \alpha_k(0) > 0, \sum_{i \leq i_{\max}} \alpha_k(i) \leq 1 \right\}$$

and $\mathcal{G} = \{\gamma = (\gamma_k(i))_{k \in U, i \in \{0, \ldots, i_{\max}\}} : \exists \alpha \in \mathcal{A} | \gamma = \Gamma(\alpha)\}$. Γ is a continuously differentiable bijection from $\mathcal{A}$ to $\mathcal{G}$ having a continuously differentiable inverse.

Estimate γ by the empirical probabilities $\widehat{\gamma}$, where

$$\widehat{\gamma}_k(i) = n^{-1} \sum_{m=1}^{n} \mathbf{1}_{\{\widehat{Y}_{k,m} \leq iq + q/2\}}. \tag{1}$$

Here $\mathbf{1}_S$ denotes the indicator function of the set S and $(\widehat{Y}_{k,m})_{k \in U, m=1,\ldots,n}$ are the subsidiary quantities

$$\widehat{Y}_{k,m} = \min_{d \in R(k)} Z_{d,m}, \quad k \in U. \tag{2}$$

Our estimate of $\alpha_k(i)$ is $\widehat{\alpha}_k(i) = (\Gamma^{-1}(\widehat{\gamma}))_k(i)$. We estimate the link k utilization by $\widehat{u}_k = 1 - \widehat{\alpha}_k(0)$.

Let

$$\mathcal{A}^{(1)} = \left\{ \alpha = (\alpha_k(i))_{k\in U, i\in\{0,\ldots,i_{\max}\}} : \alpha_k(0) > 0, \sum_{i\le i_{\max}} \alpha_k(i) < 1 \right\}$$

denote the open interior of $\mathcal{A}$. The following holds.

Theorem 2: When $\gamma \in \Gamma(\mathcal{A}^{(1)})$, as $n \to \infty$, $\widehat{\alpha} = \Gamma^{-1}(\widehat{\gamma})$ converges almost surely to α, i.e., the estimator is strongly consistent.

Proof: Since Γ^{-1} is continuous on $\Gamma(\mathcal{A}^{(1)})$ and $\mathcal{A}^{(1)}$ is open in $\mathcal{A}$, it follows that $\Gamma(\mathcal{A}^{(1)})$ is an open set in $\Gamma(\mathcal{A})$. By the Strong Law of Large Numbers, since $\widehat{\gamma}$ is the mean of n independent, identically distributed (i.i.d.) random variables, $\widehat{\gamma}$ converges to γ almost surely for $n \to \infty$. Therefore, when $\gamma \in \Gamma(\mathcal{A}^{(1)})$, there exists n_0 such that $\widehat{\gamma} \in \Gamma(A^{(1)})$, $n > n_0$. Then, the continuity of Γ^{-1} insures that $\widehat{\alpha}$ converges almost surely to α as $n \to \infty$. ∎

B. Proof of Theorem 1

To prove the theorem, we first express γ as function of α and then show that the mapping from $\mathcal{A}$ to $\mathcal{G}$ is injective.

1) Relating γ to α: Let $\beta_k(i) = P[\min_{j\in R(k)} Z_j - Z_{f(k)} \le iq]$, $i = 0, \ldots, i_{\max}$, so $\beta_k(i)$ obeys the recursion

$$\beta_k(i) = \sum_{j=0}^{i} \alpha_k(j) \left[1 - \prod_{d\in d(k)} (1 - \beta_d(i-j)) \right] \quad k \in U \setminus R$$

$$\beta_k(i) = \sum_{j=0}^{i} \alpha_k(j) \; k \in R. \tag{3}$$

Then, by observing that

$$\gamma_k(i) = \sum_{j=0}^{i} \beta_k(i-j) A_{f(k)}(j) \tag{4}$$

$k \in U \setminus R$, we readily obtain

$$\gamma_k(i) = \sum_{j=0}^{i} A_k(j) \left[1 - \prod_{d\in d(k)} (1 - \beta_d(i-j)) \right] \quad k \in U \setminus R$$

$$\gamma_k(i) = \sum_{j=0}^{i} A_k(j) \; k \in R. \tag{5}$$

The set of equations (5) completely identifies the mapping Γ from $\mathcal{A}$ to $\mathcal{G}$. The mapping is clearly continuously differentiable. Observe that the above expressions can be regarded as a generalization of those derived for the loss estimator in [3] (by identifying the event *no delay* with the event *no loss*).

2) Relating α to γ: It remains to show that the mapping from $\mathcal{A}$ to $\mathcal{G}$ is injective. To this end, below we derive an algorithm for inverting (5). Given its length, we will omit the proof of the uniqueness and continuous differentiability of the inverse. The details can be found in [17].

For the sake of clarity we separate the algorithm into two parts. In the first we derive the cumulative delay distributions A from γ; then, we deconvolve A to obtain α.

Computing A:

Step 0: Solve (5) for $i = 0$. This amounts to solving the equations

$$\left(\frac{1 - \gamma_k(0)}{A_k(0)} \right) = \prod_{d\in d(k)} \left(\frac{1 - \gamma_d(0)}{A_k(0)} \right), k \in U \setminus R \tag{6}$$

and

$$\gamma_k(0) = A_k(0), \; k \in R. \tag{7}$$

These equations are formally identical to those for the loss estimator [3]. From [3], we have that the solution of (6) exists and is unique in (0,1) provided that $0 < \gamma_k(0) < \sum_{d\in d(k)} \gamma_d(0)$, which holds for canonical delay trees. We then compute $\beta_k(0) = \gamma_k(0)/A_{f(k)}(0)$, $k \in U$.

Step i: Given $A_k(j)$ and $\beta_k(j)$, $k \in U$, $j = 0, \ldots, i-1$, in this step we compute $A_k(i)$ and $\beta_k(i)$, $k \in U$. For $k \in U \setminus R$, in (5) we replace $\beta_d(i)$ with $(\gamma_d(i) - \sum_{j=1}^{i-1} \beta_d(i-j) A_k(j) - \beta_d(0) A_k(i))/A_k(0)$ [from (4)]) and obtain (8), shown at the bottom of the page, where the unknown term $A_k(i)$ is highlighted in boldface. This is a polynomial in $A_k(i)$ of degree $\#d(k)$. As shown in the Appendix , we recover $A_k(i)$ from the second largest solution of (8).

For $k \in R$, we directly compute $A_k(i)$ from (5), $A_k(i) = \gamma_k(i) - \sum_{j=0}^{i-1} A_k(j)$. Then we compute $\beta_k(i)$, $k \in U$, as $\beta_k(i) = (\gamma_k(i) - \sum_{j=1}^{i} A_{f(k)}(j) \beta_k(i-j))/A_{f(k)}(0)$

Computing α: Once step $i_{\max}$ is completed, we compute $\alpha_k(i)$, $k \in U$ as follows:

$$\alpha_k(i) = \begin{cases} \frac{A_k(0)}{A_{f(k)}(0)} & i = 0 \\ \frac{(A_k(i) - \sum_{j=1}^{i} A_{f(k)}(j) \alpha_k(i-j))}{A_{f(k)}(0)} & i > 0. \end{cases} \tag{9}$$

$$\gamma_k(i) + \sum_{j=1}^{i-1} A_k(j) \left\{ \prod_{d\in d(k)} [1 - \beta_d(i-j)] - 1 \right\} + \boldsymbol{A_k(i)} \left\{ \prod_{d\in d(k)} [1 - \beta_d(0)] - 1 \right\} + A_k(0) \left\{ \prod_{d\in d(k)} \left[1 - \frac{\gamma_d(i) - \sum_{j=1}^{i-1} \beta_d(i-j) A_k(j) - \beta_d(0) \boldsymbol{A_k(i)}}{A_k(0)} \right] - 1 \right\} = 0 \tag{8}$$

C. Rate of Convergence of the Delay Distribution Estimator

1) Asymptotic Behavior of the Delay Distribution Estimator: In this section, we study the rate of convergence of the estimator. Theorem 2 states that $\widehat{\alpha}$ converges to α with probability 1 as n grows to infinity, but it provides no information on the rate of convergence. Because of the mild conditions satisfied by Γ^{-1}, we can use the Central Limit Theorem to establish the following asymptotic result.

Theorem 3: When $\gamma \in \Gamma(\mathcal{A}^{(1)})$, as $n \to \infty$, $\sqrt{n}(\widehat{\alpha}-\alpha)$ converges in distribution to a multivariate normal random variable with mean vector 0 and covariance matrix $\nu = D(\alpha)\cdot\sigma\cdot D^T(\alpha)$ where $\sigma_{(k_1,i)(k_2,j)} = \lim_{n\to\infty} n \cdot \mathrm{Cov}(\widehat{\gamma}_{k_1}(i), \widehat{\gamma}_{k_2}(j))$, for $k_1, k_2 \in U$, $i,j \in \{0,\ldots,i_{\max}\}$, $D_{(k_1,i)(k_2,j)}(\alpha) = (\partial\Gamma^{-1}_{k_1}(i))/(\partial\gamma_{k_2}(j))(\Gamma(\alpha))$ and D^T denotes the transpose.

Proof: By the Central Limit Theorem, it follows that the random variables $\widehat{\gamma}$ are asymptotically Gaussian as $n \to \infty$ with

$$\sqrt{n}(\widehat{\gamma}-\gamma)\xrightarrow{\mathcal{D}}\mathcal{N}(0,\sigma). \tag{10}$$

Here $\mathcal{D}$ denotes convergence in distribution. Following the same lines as in the proof of Theorem 1, when $\gamma \in \Gamma(\mathcal{A}^{(1)})$, there exists n_0 such that $\widehat{\gamma} \in \Gamma(\mathcal{A}^{(1)})$, $n > n_0$. Then, Since Γ^{-1} is continuously differentiable on $\mathcal{G}$, the Delta method (see [30, Ch. 7]) yields that $\widehat{\alpha} = \Gamma^{-1}(\widehat{\gamma})$ is also asymptotically Gaussian as $n \to \infty$:

$$\sqrt{n}(\Gamma^{-1}(\widehat{\gamma})-\alpha)\xrightarrow{\mathcal{D}}\mathcal{N}(0,\nu). \tag{11}$$

■

Theorem 3 allows us to compute confidence intervals for the parameters, and allows us to estimate their rate of convergence to the true values as n grows. This is relevant in assessing: 1) the number of probes required to obtain a desired level of accuracy of the estimate and 2) the likely accuracy of the estimator from actual measurements by associating confidence intervals to the estimates.

For large n, the estimator $\widehat{\alpha}_k(i)$ will lie in the interval

$$\alpha_k(i) \pm z_{\delta/2}\sqrt{\frac{\nu_{(k,i)(k,i)}}{n}} \tag{12}$$

where $z_{\delta/2}$ is the $1-\delta/2$ quantile of the standard distribution and the interval estimate is a $100(1-\delta)\%$ confidence interval.

To obtain the confidence interval for $\widehat{\alpha}$ derived from measured data from n probes, we estimate ν by $\widehat{\nu} = D(\widehat{\alpha})\cdot\widehat{\sigma}\cdot D^T(\widehat{\alpha})$ where

$$\widehat{\sigma}_{(k_1,i)(k_2,j)} = \frac{1}{n-1}\left(\sum_{l=1}^{n}\mathbf{1}_{\{\widehat{Y}_{k_1,l}\le iq+q/2\ \bigwedge\ \widehat{Y}_{k_2,l}\le jq+q/2\}} - \frac{1}{n}\sum_{l=1}^{n}\mathbf{1}_{\{\widehat{Y}_{k_1,l}\le iq+q/2\}}\sum_{l=1}^{n}\mathbf{1}_{\{\widehat{Y}_{k_2,l}\le jq+q/2\}}\right)$$

and $D(\widehat{\alpha})$ is the Jacobian of the inverse map Γ^{-1} computed for $\alpha = \widehat{\alpha}$. We then use confidence intervals of the form

$$\widehat{\alpha}_k(i) \pm z_{\delta/2}\sqrt{\frac{\widehat{\nu}_{(k,i)(k,i)}}{n}}. \tag{13}$$

2) Dependence of the Delay Distribution Estimator on Topology: The estimator variance determines the number of probes required to obtain a given level of accuracy. Therefore, it is important to understand how the variance is affected by the underlying parameters, namely, the delay distributions and the multicast tree topology. The following Theorem characterizes the behavior of the variance for small delays. Set $\|\alpha\| = \max_{k\in U, i>0}\alpha_k(i)$.

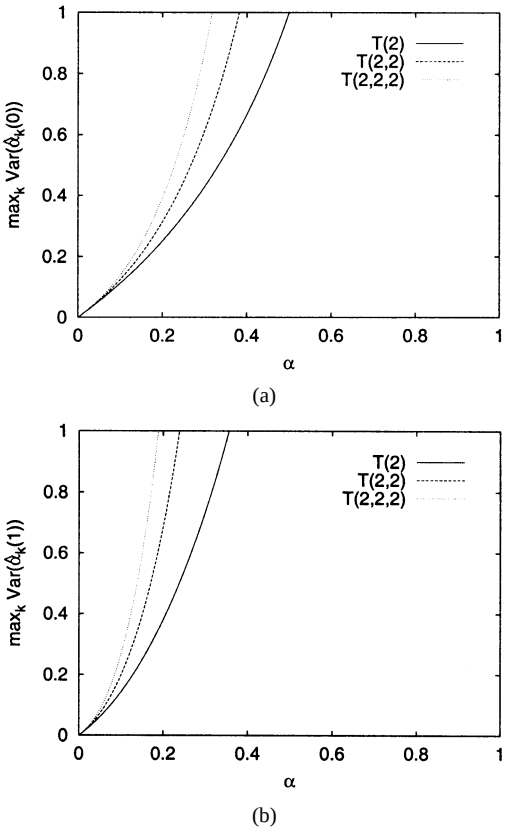


Fig. 1. Asymptotic estimator variance and tree depth. Binary trees of depths 2, 3, and 4. Maximum Variance of the estimates $\widehat{\alpha}_k(0)$ (left) and $\widehat{\alpha}_k(1)$ (right) over all links.

Theorem 4: As $\|\alpha\| \to 0$, then

$$\nu = \begin{pmatrix} \nu_{k_1k_1} & 0 & 0 & \ldots & 0 \\ 0 & \nu_{k_2k_2} & 0 & \ldots & 0 \\ 0 & 0 & \nu_{k_3k_3} & \ldots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \ldots & \nu_{k_{\#U}k_{\#U}} \end{pmatrix} + O(\|\alpha\|^2)$$

with

$$\nu_{kk} = \begin{pmatrix} \sum_{i>0}\alpha_k(i) & \alpha_k(1) & \alpha_k(2) & \ldots & \alpha_k(i_{\max}) \\ \alpha_k(1) & \alpha_k(1) & 0 & \ldots & 0 \\ \alpha_k(2) & 0 & \alpha_k(2) & \ldots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ \alpha_k(i_{\max}) & 0 & 0 & \ldots & \alpha_k(i_{\max}) \end{pmatrix}.$$

Theorem 4 can be derived by direct computation of the element of ν and by taking the limits for $\|\alpha\| \to 0$. Given the length of the proof, it is omitted. The details can be found in [17].

Theorem 4 states that the estimator variance is, to first order, independent of the topology. To explore higher order dependencies, we computed the asymptotic variance for a selection of trees with different depths and branching ratio. We use the notation $T(r_1,\ldots,r_m)$ to denote a tree of $m+1$ levels where,

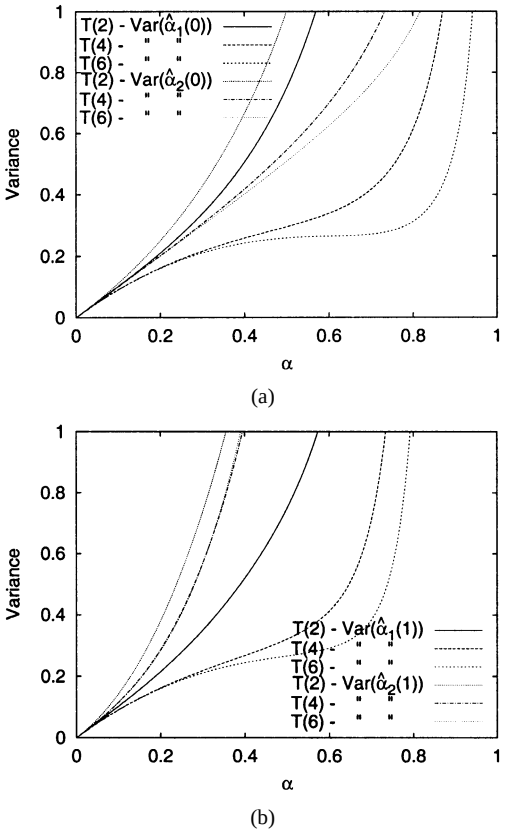


Fig. 2. Asymptotic estimator variance and branching ratio. Binary tree with depth 2 and 2, 4, or 6 receivers. Variance of (a) $\widehat{\alpha}_k(0)$ and (b) $\widehat{\alpha}_k(1)$ for link 1 (common link) and link 2 (generic receiver).

apart from node 0, which has one descendant, nodes at level j have exactly r_j children. For simplicity, we consider the case when link delay takes values in {0,1}, i.e.,, we consider no loss and study the behavior as a function of $\alpha_k(1) = \alpha$.

In Fig. 1, we show the dependence on tree depth for binary trees of depths 2, 3, and 4. We plot the maximum value $\max_k \mathrm{Var}(\widehat{\alpha}_k(0))$ of the variance over the links, and $\max_k \mathrm{Var}(\widehat{\alpha}_k(1))$. In these examples, the variance increases with the tree depth. In Fig. 2 we show the dependence on branching ratio for a tree of level 2. We plot the estimator variance for both link 1 (the common link) and link 2 (a generic receiver). In these examples, increasing the branching ratio decreases the variances, especially those of the common link estimates, which increase less than linearly for α up to 0.7 when the branching ratio is larger than 3. In all cases, the variance of $\widehat{\alpha}_k(1)$ is larger than that of $\widehat{\alpha}_k(0)$.

As predicted by Theorem 4, the estimator variance is always asymptotically linear in α independently of the topology as $\alpha \to 0$. As α increases, the behavior is affected by different factors: increasing the branching ratio results in a reduction of the variance, while increasing the tree depth results in variance increase.

IV. Computation of the Delay Distribution Estimate

In this section, we describe an algorithm for computing the delay distribution estimate from measurements based on the results presented in the previous section.

We assume we have the experimental data of source-to-leaf delays $(Z_{k,m})_{k\in R, m=1,\ldots n}$ from n probes, as collected at the leaf nodes $k \in R$. Two steps must be initially performed to render the data into a form suitable for the inference algorithms: 1) removal of fixed delays and 2) choosing a bin size q and computing the estimate $\widehat{\gamma}$.

The first step is necessary since here we are interested in the variable portion of the link delay. (Observe also that it is generally not possible to apportion the deterministic component of the source-to-leaf delays between interior links. To see this, it is sufficient to consider the case of the two receiver tree; expressing the link fixed delays in terms of the source-to-leaf fixed delays results in two equations with three unknowns). Thus, we normalize each measurement by subtracting the minimum delay seen at the leaf which we assume to be equal to the end-to-end fixed transmission delay. In other words, we assume that at least one probe has experienced no queueing delay along the path. We note that this assumption may not always hold in practice, especially for a small number of probes and/or when there is congestion along one or more links. In these cases, the algorithm yields biased estimates as the error in the normalized measurements results in the estimates $\widehat{\gamma}$ being biased. Nevertheless, since Γ^{-1} is continuous, we expect to have accurate results as long as the end-to-end minimum delay is very close to the actual transmission delay.

The second step is to choose the bin size q and discretize the delay measurements accordingly. This introduces a quantization error which affects the accuracy of the estimates. As our results have shown, the accuracy improves as q decreases (we have obtained accurate results over a significant range of values of q up to the same order of magnitude as the average link delay). The choice of q represents a tradeoff between accuracy and cost of the computation as a smaller bin size entails a higher computational cost due to the higher dimensionality of the binned distributions.

These two steps are carried out as follows. From the measured data, we recursively construct the auxiliary vector process $\widehat{Y} = (\widehat{Y}_{k,l})_{k\in U, m=1,\ldots,n}$

$$\widehat{Y}_{k,l} = Z_{k,l} - \min_{m\in\{1,\ldots,n\}} Z_{k,m},\ k \in R \tag{14}$$

$$\widehat{Y}_{k,l} = \min_{j\in d(k)} \widehat{Y}_{j,l},\ k \in U \setminus R. \tag{15}$$

The binned estimates $\widehat{\gamma}$ are

$$\widehat{\gamma}_k(i) = n^{-1} \sum_{m=1}^{n} \mathbf{1}_{\{\widehat{Y}_{k,m} \le iq+q/2]\}},\ i = 0, \ldots, i_{\max}$$

with $i_{\max} = \left\lceil (\max_{k\in R} \max_{m\in N_k(n)} \widehat{Y}_{k,m})/q \right\rceil$. Here $\lceil x \rceil$ denotes the smallest integer greater than x and $N_k(n) = \{m \in \{1,\ldots,n\} | Z_{k,m} < \infty\}$.

The pseudocode for carrying out the computation is found in Fig. 3. The procedure find_y_γ calculates $\widehat{Y}$ and $\widehat{\gamma}$, with $\widehat{Y}_{k,l}$ initialized to $Z_{k,l} - \min_{m\in\{1,\ldots,n\}} Z_{k,m}$ for $k \in R$ and ∞ (a value larger than any observed delay suffices) otherwise. The procedure infer_A calculates the cumulative distributions $\widehat{A}_k$, for a given node $k \in U$ ($\widehat{A}_k[i]$, $k \in V$, $i = 0, \ldots, i_{\max}$ are initialized to 0, except for $\widehat{A}_0[0]$ which is set to 1). The link distributions $\widehat{\alpha}_k$, $k \in U$ are then recovered by deconvolution. The routines solvefor1 and solvefor2 solve (6) and (8), respectively, with respect to $\widehat{A}_k[i]$. solvefor1 returns a solution in (0,1); from

```
procedure main {
    find_y_γ ( 1 , q ) ;
    foreach ( k ∈ U )
        infer_A ( k , q ) ;
    foreach ( k ∈ U )
        foreach ( i ∈ {0, ···, i_max} )
            α̂_k[i] = (Â_k[i] − Σ_{j=1}^{i} Â_{f(k)}[j] α̂_k[i−j]) / Â_{f(k)}[0] ;
    }

procedure find_y_γ ( k , q ) {
    foreach ( j ∈ d(k) ) {
        find_y_γ ( j , q ) ;
        foreach ( m ∈ {1, ..., n} )
            Ŷ_k[m] = min{Ŷ_k[m], Ŷ_j[m]};
    }
    foreach ( i ∈ {0, ···, i_max} )
        γ̂_k[i] = n^{-1} Σ_{j=1}^{n} 1_{{Ŷ_k[j] ≤ iq+q/2}};
}

procedure infer_A ( k , q ) {
    if ( k ∈ R )
        Â_k = γ̂_k ;
    else
        foreach ( i ∈ {0, ···, i_max} ) {
            if ( i == 0 )
                Â_k[i] = solvefor1 ( (1 − γ̂_k[i]/Â_k[i] == ∏_{d∈d(k)} 1 − γ̂[i]/Â[i] ) ;
            else
                Â_k[i] = solvefor2 ( γ̂_k[i] + Â_k[0] { ∏_{d∈d(k)} [1 − (γ̂_d[i] − Σ_{j=1}^{i} β̂_d[i−j] Â_k[j]) / Â_k[0])] − 1 } +
                Σ_{j=1}^{i} Â_k[j] { ∏_{d∈d(k)} [1 − β̂_d[i − j]] − 1 } == 0 ) ;
            foreach ( d ∈ d(k) )
                β̂_d[i] = (γ̂_d[i] − Σ_{j=1}^{i} Â_k[j] β̂_d[i−j]) / Â_k[0] ;
        }
}
```

Fig. 3. Pseudocode for inference of delay distribution.

[3, Lemma 1] this is known to be unique. solvefor2 returns the second largest solution.

A. Adopting Different Bin Sizes

Following the results of the previous section, we presented the algorithm using a fixed value of q for all links. This can be quite restrictive in a heterogeneous environment, where links may differ significantly in terms of speed and buffer sizes; a single value of q could be at the same time too coarse-grained for describing the delay of a high-bandwidth link but too fine-grained to efficiently capture the essential characteristics of the delay experienced along a low bandwidth link.

A simple way to overcome this limitation is to run the algorithm for different values of q, each best suited for the behavior of a different group of links, and retain each time only the solutions for those links. This approach, though, ends up computing all distributions for all bin sizes. A more efficient solution can take advantage of the fact that the cumulative delay distributions $\widehat{A}_k$ are computed independently as follows. Denote by q_k the bin size adopted for link k. For each link $k \in U$, compute $\widehat{A}_k$ and $\widehat{A}_{f(k)}$ using q_k as bin size and then deconvolve the two distributions to obtain $\widehat{\alpha}_k$ (with bin size q_k). Thus, overall, we need to compute $\widehat{A}_k$, $k \in U$, only for $q \in \{q_j | j \in \{k\} \cup d(k)\}$. (Observe that since the computation of $\widehat{A}_k$ with a given bin size requires the estimates $\widehat{\gamma}_j$, $j \in \{k\} \cup d(k)$, to be computed with the same bin size, we need to precompute the estimates $\widehat{\gamma}_k$ for $q \in \{q_j | j \in \{f(k)\} \cup d(f(k)) \cup d(k)\}$).

In an implementation, we envision that a fixed value for all links be used first. This can be chosen based on the measurements spread and the tree topology or delay past history. Then, with a better idea of each link delay spread, it would be possible to refine the value of the bin size on a link by link basis.

B. Computing $\widehat{\alpha}$ via Linear Least Square

The algorithm we described computes $\widehat{\alpha}$ from $\widehat{A}$ by simply inverting the convolution

$$A_k(i) = \sum_{j=0}^{i} \alpha_k(j) A_{f(k)}(i-j) \tag{16}$$

by

$$\alpha_k(i) = \frac{A_k(i) - \sum_{j=1}^{i} A_{f(k)}(j)\alpha_k(i-j)}{A_{f(k)}(0)} \tag{17}$$

$k \in U$, $i = 0, \ldots, i_{\max}$. In our experiments, we found out that direct application of (17) yields in some cases poor results, especially for links with very small delay where most of the probabilities $\alpha_k(i)$ are 0. In this case, we often obtain negative estimates with non negligible absolute values. Causes of this behavior are: 1) the statistical variability of the estimates $\widehat{A}$ and 2) correlation among delays on different links (in this case the use of (16) becomes an approximation). To improve the accuracy of the estimates, we use a least square error with linear inequality constraints estimate (LSI) approach to estimate α. The idea is to compute the best "linear" estimate of α_k given $\widehat{A}_k$ and $\widehat{A}_{f(k)}$. This amounts to determining the best (in least square error sense) approximation of the distribution α_k that satisfies the convolution (16). The use of linear inequality constraints allows us to impose nonnegativity to each probability.

To formulate the estimation of $\widehat{\alpha}_k$ as a LSI, we write (16) in matrix form

$$Ex = b$$

where

$$E = \begin{bmatrix} \widehat{A}_{f(k)}(0) & 0 & \cdots & 0 \\ \widehat{A}_{f(k)}(1) & \widehat{A}_{f(k)}(0) & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ \widehat{A}_{f(k)}(i_{\max}-1) & \widehat{A}_{f(k)}(i_{\max}-2) & \cdots & 0 \\ \widehat{A}_{f(k)}(i_{\max}) & \widehat{A}_{f(k)}(i_{\max}-1) & \cdots & \widehat{A}_{f(k)}(0) \end{bmatrix}$$

and $x^T = [\widehat{\alpha}_k(0), \ldots, \widehat{\alpha}_k(i_{\max})]^T$, $b^T = [\widehat{A}_k(0), \ldots, \widehat{A}_k(i_{\max})]^T$.

The LSI can be formulate as follows.

Minimize $||Ex - b||$ subject to

$$\sum_{i=0}^{i_{\max}} x(i) \leq 1, x(i) \geq 0, \quad i = 0, \ldots, i_{\max}$$

where $||y|| = \sum_i y^2(i)$. The solution of the LSI problem can be carried out using standard techniques involving the solution of a finite sequence of ordinary least square problems. Details can be found [16, Ch. 23].

V. Experimental Evaluation

We evaluated our delay estimator through extensive simulation. Here, we first focus on the statistical properties of the estimator. We perform *model simulation*, where delay and loss are determined by random processes that follow the model on which we based our analysis. Then we investigate the behavior of the estimators in a more realistic setting where the model assumption of independence may be violated. To this end, we perform *TCP/UDP simulation*, using the ns simulator. Here delay and loss are determined by queueing delay and queue overflows at network nodes as multicast probes compete with traffic generated by TCP/UDP traffic sources.

A. Comparing Inferred Versus Sample Distributions

Before examining the results of our experiments, we describe our approach to assessing the accuracy of the inferred distributions. Given an experiment in which n probes are sent from the source to the receivers, for $k \in U$, the inferred distribution $\widehat{\alpha}_k(\widehat{A}_k)$ is computed from the end-to-end measurements using the algorithm described in Section IV. Its accuracy must be measured against the actual data, represented by a finite sequence of delays $\{D_{k,m}\}_{m=1}^n$ ($\{Z_{k,m}\}_{m=1}^n$), experienced by the probes in traversing (reaching) that link. For simplicity of notation, we assume, hereafter, that each set of data has been already normalized by subtracting the minimum delay from the sequence.

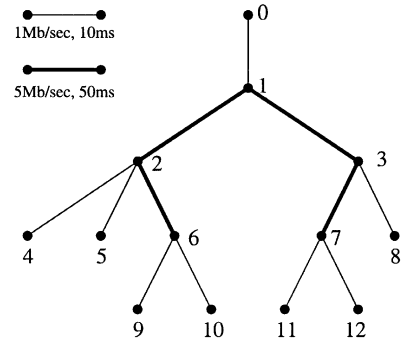


Fig. 4. Simulation topology. Link are of two types: *edge* links of 1 MB/s capacity and 10 ms latency and *interior* links of 5 Mb/s capacity and 50 ms latency.

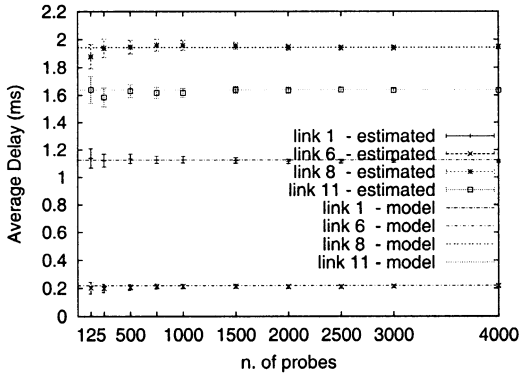
We compare summary statistics of link delay, namely the mean and the variance. A finer evaluation of the accuracy lies in a direct comparison of the inferred and sample distributions. To this end, we also compute the largest absolute deviation between the inferred and sample c.d.f.s. This measure is used in statistics for the Kolmogoroff–Smirnoff test for goodness of fit of a theoretical with a sample distribution. A small value for this measure indicates that the theoretical distribution provides a good fit to the sample distribution; a large value leads to the rejection of the hypothesis. We cannot directly apply the test as we deal with an inferred rather than a sample c.d.f.; however, we will use the largest absolute deviation as a global measure of accuracy of the inferred distributions.

We compute the sample distributions $\tilde{\alpha}_k$ and $\tilde{A}_k$ using the same bin size q of the estimator. More precisely, we compute $\tilde{\alpha}_k$, $k \in U$ and $\tilde{A}_k$, $k \in U$ as $\tilde{\alpha}_k(i) = \#N_{f(k)}(n)^{-1} \sum_{l \in N_{f(k)}(n)} \mathbf{1}_{\{D_{k,l} \in (iq-q/2, iq+q/2]\}}$, $i = 0, \ldots, i_{\max}$, and $\tilde{A}_k(i) = n^{-1} \sum_{l=1}^n \mathbf{1}_{\{Z_{k,l} \in (iq-q/2, iq+q/2]\}}$, $i = 0, \ldots, i_{\max}$. (Observe that in computing $\tilde{\alpha}_k$, the sum is carried out only over $N_{f(k)}(n)$, i.e., the set of probes with finite cumulative delay to $f(k)$.)

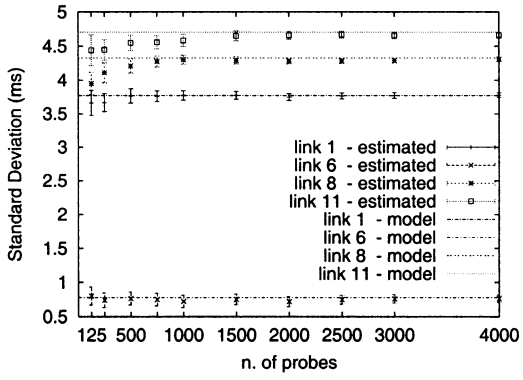
The largest absolute deviation between the inferred and sample c.d.f.s is, then, $\Delta_k = \max_{l=0,\ldots,i_{\max}} |\sum_{i=0}^l \tilde{\alpha}_k(i) - \sum_{i=0}^l \widehat{\alpha}_k(i)|$. In other words, Δ_k is the smallest nonnegative number such that $\sum_{j \leq i} \tilde{\alpha}_k(i)$ lies between $\sum_{j \leq i} \widehat{\alpha}_k(i) \pm \Delta_k$ $i = 0, \ldots, i_{\max}$. The same result holds for the tail probabilities, $\sum_{j>i} \alpha_k(i)$.

B. Model Simulation

We consider the topology of Fig. 4. Delays are independently distributed according to a truncated geometric distribution taking values in $\{0, 1, \ldots, 40, \infty)$ (in milliseconds) . This topology is also used in subsequent TCP/UDP simulations,

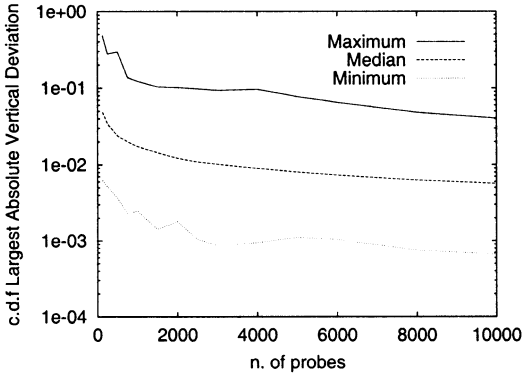


(a)

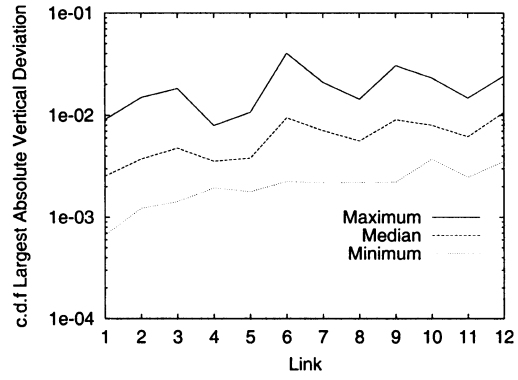


(b)

Fig. 5. Model simulation. Estimated versus (a) theoretical delay average and (b) standard deviation with 95% confidence interval computed over 100 model simulations.



(a)



(b)

Fig. 6. Model simulation. Accuracy of the estimated distribution. Largest vertical absolute deviation between estimated and sample c.d.f. Minimum, median, and maximum largest absolute deviation in 100 simulations computed (a) over all links as function of n and (b) link by link for $n = 10\,000$.

and the link average delay and loss probability are chosen to match the values obtained from these. The average delay ranges between 1 and 2 ms for the slower *edge* links and between 0.2 and 0.5 ms for the *interior* faster links; the link losses range from 1% to 11%. In Fig. 5, we plot the estimated average link delay and standard deviation with the empirical 95% confidence interval computed over 100 simulations. The results are very accurate even for several hundred probes: the theoretical average delay always lies within the confidence interval and the standard deviation does so for 1500 or more probes.

To compare the inferred and sample distributions, we computed the largest absolute deviation between the inferred and sample c.d.f.s. The results are summarized in Fig. 6 where we plot the minimum, median, and the maximum largest absolute deviation in 100 simulations computed over all links as a function of n and link by link for $n = 10\,000$. The accuracy increases with the number of probes as $1/\sqrt{n}$ with a spread of two orders of magnitude between the minimum and maximum. For more than 3000 probes, the average largest deviation over all links is less than 1%. The accuracy varies from link to link: when the number of probes is $n = 10\,000$, then at one extreme we have link 4 with $0.18\% \leq \Delta_4 \leq 0.8\%$ and at the other extreme link 6 with $0.3\% \leq \Delta_6 \leq 4\%$ over 100 simulations. We observe that the inferred distributions are less accurate as we go down the tree. This is in agreement with the results of Section III-C and is explained in terms of the larger inferred variances of downstream with respect to upstream nodes.

C. TCP/UDP Simulations

We conducted several sets of ns simulations. Here, we first use a simple topology to study in detail the accuracy and convergence properties of the inference algorithm. We will then consider a larger topology to study how the algorithm performs in a more realistic setting.

We first consider topology shown in Fig. 4. To capture the heterogeneity between edges and core of a WAN, interior links have higher capacity (5 Mb/s) and propagation delay (50 ms) than those at the edge (1 Mb/s and 10 ms). Each link is modeled as a FIFO queue with a four-packet capacity. Observe that for this topology logical and physical links coincide.

Node 0 generates probes as a 20-Kb/s stream comprising 40-byte UDP packets according to a Poisson process with a mean interarrival time of 16 ms; this represents 2% of the smallest link capacity. The background traffic comprises 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. Averaged over the different simulations, the link loss ranges between 1%–11% and link utilization ranges between 20%–60%.

For a single experiment, Fig. 7 compares the estimated versus the sample average delay for representative selected links. The analysis has been carried out using $q = 1$ ms and $q = 0.1$ ms. In this example, we obtain practically the same accuracy despite a tenfold difference in resolution. (Observe that $q = 1$ ms is the same order of magnitude as the average delays.) The inferred

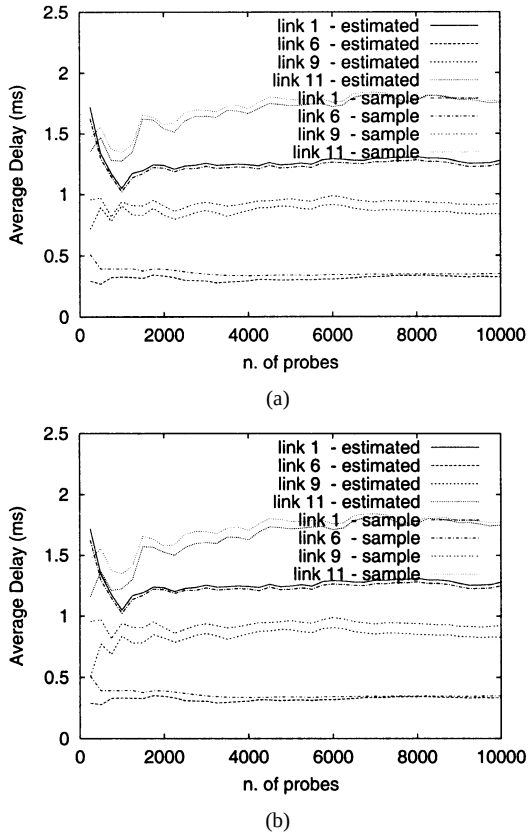


Fig. 7. Convergence of inferred versus sample average link delay in TCP/UDP simulations. (a) Bin-size $q = 1$ ms. (b) Bin size $q = 0.1$ ms. The graphs shows how the inferred values closely track the sample average delays.

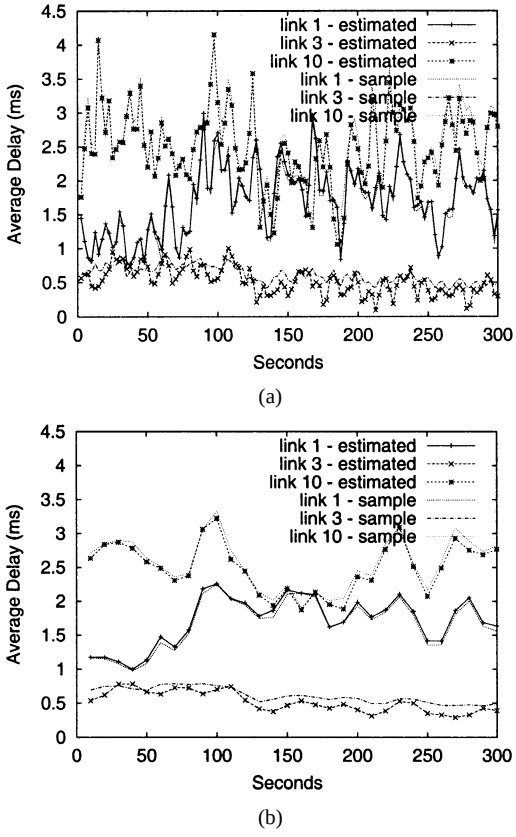


Fig. 8. Dynamic accuracy of inference. Sample and inferred average delay on links 1, 3, and 10 of the multicast tree in Fig. 4. (a) Five-second window. (b) Twenty-second windows. Background traffic has random start and stop times.

averages rapidly converge to the sample averages even though we have persistent systematic errors in the inferred values due to consistent spatial correlation. We shall comment upon this later.

In order to show that the inferred values converge quickly and exhibit good dynamical tracking, in Fig. 8 we plot the inferred versus the sample average delay for three links (1, 3, and 10) computed over a moving window of two different sizes with jumps of half its width. To allow greater dynamics, here we arranged background sources with random start and stop times. Under both window sizes (approximately 300 and 1200 probes are used, respectively), the estimates of the average delays of links 1 and 10 show good agreement and a quick response to delay variability revealing a good convergence rate of the estimator. For link 3 with a smaller average delay, the behavior is rather poor, especially for 5-s windows.

The persistence of systematic errors we observe in Fig. 7 is due to the presence of spatial correlation. In our simulations, a multicast probe is more likely to experience a similar level of congestion on consecutive links or on sibling links than is dictated by the independence assumption. We also verified the presence of temporal correlation among successive probes on the same link caused by consecutive probes experiencing the same congestion level at a node.

To assess the extent to which our real traffic simulations violate the model assumptions, we computed the delay correlation between links pairs and among packets on the same link. The analysis revealed the presence of significant spatial correlations up to $0.3 \sim 0.4$ between consecutive links. The smallest values are observed for link 5 which always exhibits a correlation of less than 0.1 with its parent node.

The autocorrelation function rapidly decreases and can be considered negligible for a lag larger than 30 (approximatively 2 s). The presence of short-term correlation does not alter the key property of convergence of the estimator as it suffices that the underlying processes be stationary and ergodic (this happens for example, when recurrence conditions are satisfied). The price of correlation, however, is that the convergence rate is slower than when the delays are independent.

Now we turn our attention to the inferred distributions. For an experiment of 300 seconds during which approximately 18 000 probes were generated, we plot the complementary c.d.f. conditioned on the delay being finite in Fig. 9. In Fig. 10, we also plot the complementary c.d.f of the node cumulative delay. (We show only internal links as $\widehat{A}_k(i) = \widehat{\gamma}_k(i)$, $k \in R$.) Here $q = 1$ ms.

From these two sets of plots, it is striking to note the differences between the accuracy of the estimated cumulative delay distributions $\widehat{A}_k$ and the estimated link delay distributions $\widehat{\alpha}_k$: while the former are all very close to the actual distributions, the latter are inaccurate in many cases. This is explained by observing that, in the presence of significant correlations, the convolution among A_k, α_k, and $A_{f(k)}$ used in the model does not accurately capture the relationship between the actual distributions.

The accuracy of the inferred cumulative delay distributions, on the other hand, derives from the fact that even in presence of significant local correlations, (8), which assumes indepen-

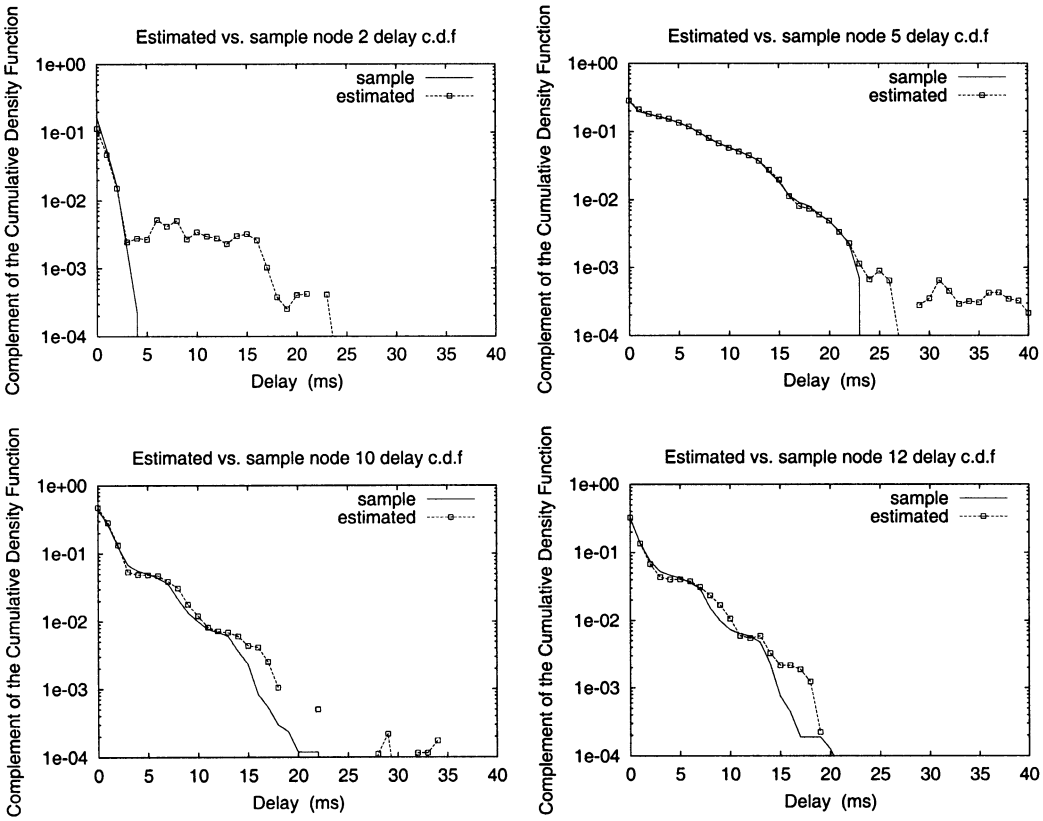
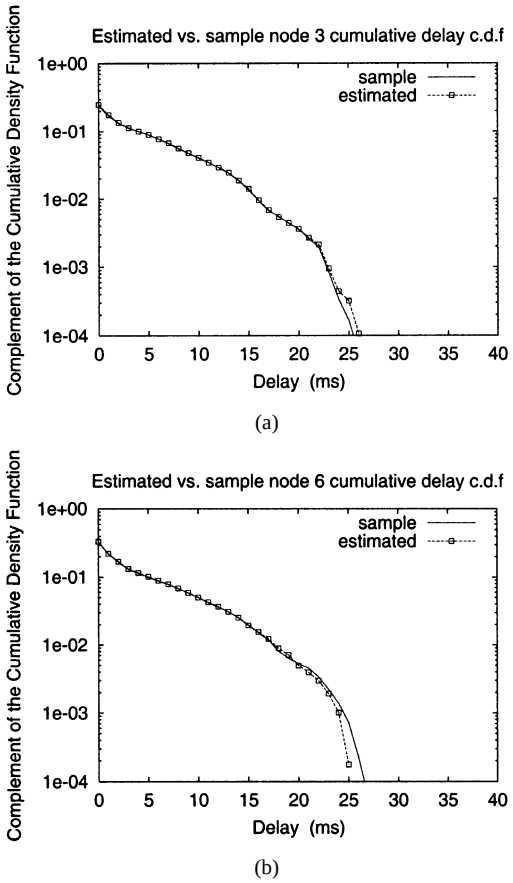


Fig. 9. Sample versus estimated delay complementary c.d.f. for selected links.



(a)

(b)

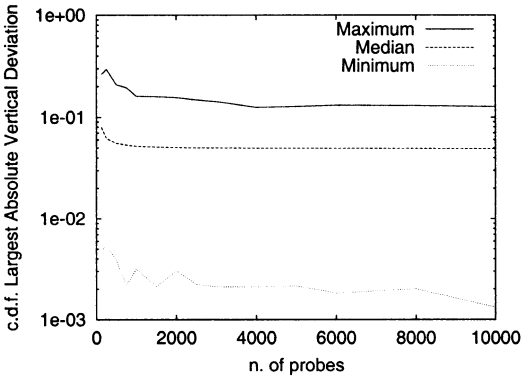
Fig. 10. Sample versus estimated cumulative delay complementary c.d.f. for selected links. (a) Node 3. (b) Node 6.

dence, is still accurate. This can be explained by observing that (8) is equivalent to (4) which consists of a convolution between $A_{f(k)}$ and β_k; we expect the correlation between the delay accrued by a probe in reaching node $f(k)$ and the minimum delay accrued between node $f(k)$ and any receiver to be rather small, especially as the tree size grows, as these delays span the entire multicast tree.

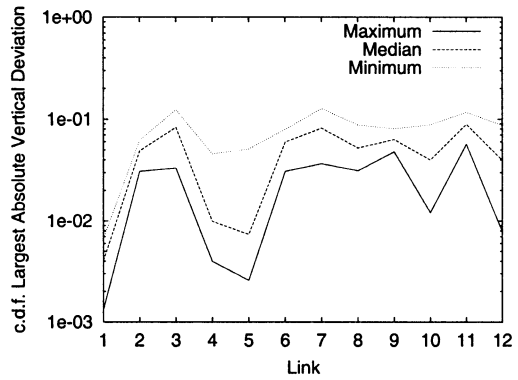
In Fig. 11, we plot the minimum, median, and maximum largest deviation between inferred and theoretical c.d.f. over 100 simulations computed over all links as function of n and link by link for $n = 10\,000$. Due to spatial correlation, the largest deviation level off after the first 2000 probes, with the median stabilizing at 5%. The accuracy again exhibits a negative trend as we descend the tree.

We also simulated larger networks to assess how the estimator would perform under realistic conditions. In these experiments, the network topologies were generated using the gt-itm topology generator [13]. For each topology, we fixed a source and a set of receivers and conducted experiments across the multicast logical tree spanning those nodes. Below we will summarize the results for the largest topology we considered, a hierarchical transit-stub network where 24 stub networks are interconnected via a 12-node transit network (the topology can be found in [17, Fig. 17]). The entire network comprises 156 nodes overall. Links between transit nodes have a 50-Mb/s capacity and a 10-ms propagation delay; the other links have a 10-Mb/s capacity and a 5-ms delay. The link buffer on each link accommodates 100 packets.

We selected one source and 38 receivers for the multicast measurements. The logical multicast tree spanning the probes

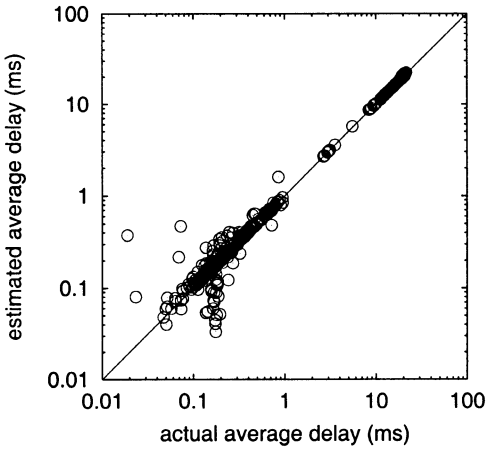


(a)

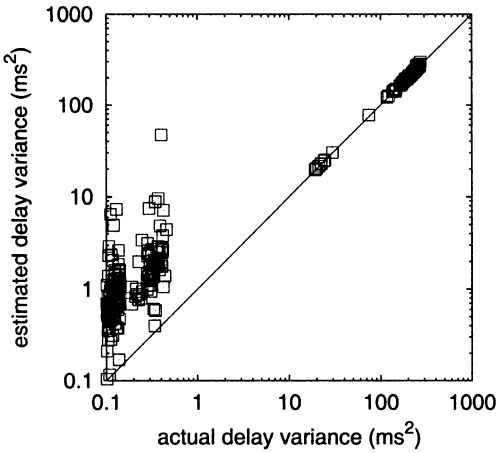


(b)

Fig. 11. TCP/UDP simulation: Topology of Fig. 4. Accuracy of the estimated distribution. Largest vertical absolute deviation between estimated and theoretical c.d.f. Minimum, median, and maximum largest absolute deviation in 100 simulations computed (a) over all links as function of n and (b) link by link for $n = 10\,000$.



(a)



(b)

Fig. 12. Inferred vs. actual average and variance of link delay in simulations. Scatter plot for 10 experiments: (a) average link delay and (b) link delay variance.

source and the receivers comprises 62 nodes. 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 and 11; the average source-receiver path length is 7.34 hops. As in the previous example, the source generates probes as a 20-kb/s stream of 40-byte UDP packets according to a Poisson process and mean intearrival 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 different simulations, loss ranged between 0%–3.4% and link utilization between 5%–93%.

Given the large number of links in the tree, here we focus on summary statistics from 10 simulations of 300 s each. In Fig. 12, we display scatter plots of inferred versus actual link delay mean and variance (using $q = 1$ ms for the analysis). Accuracy increases for link with higher delays as exhibited by the clustering about the line $y = x$. In order to quantify the accuracy of the estimates, we computed the median of the relative absolute error of the estimates of the link delay mean and variance. The median was 1.98% for the mean but 312% for the delay variance. The latter value is due to the significant relative errors in the estimation of very small delay variances (observe that the variance values span almost five orders of magnitude). Estimates were more accurate for larger delays: if we consider delay means larger than 1 ms or delay variances larger than 1 ms^2, the median of the relative absolute error fell to 0.93% and 1.57%, respectively.

VI. Conclusion and Future Work

In this paper, we introduced the use of end-to-end multicast measurements to infer network internal delay in a logical multicast tree. Under the assumption of delay independence, we derived an algorithm to estimate the per link discrete delay distributions and utilization from the measured end-to-end delay distributions. We investigated the statistical properties of the estimator, and showed it to be strongly consistent and asymptotically normal.

We evaluated our estimator through simulation. Using model simulations we verified the accuracy and convergence of the inferred to the actual values as predicted by our analysis. In real traffic simulations, we found rapid convergence, although some persistent differences from the actual distributions because of spatial correlation.

We are extending our delay distribution analysis in several directions. First we plan to do more extensive simulations, exploring larger topologies, different node behavior, background traffic and probe characteristics. Moreover, we are exploring how probe delay is representative of the delay suffered by other applications and protocols, for example TCP.

Second, we are analyzing the effect of spatial correlations among delays and we are planning to extend the model by di-

$$\gamma_k(i)+\sum_{j=1}^{i-1}A_k(j)\left\{\prod_{d\in d(k)}[1-\beta_d(i-j)]-1\right\}+x\left\{\prod_{d\in d(k)}[1-\beta_d(0)]-1\right\}$$
$$+A_k(0)\left\{\prod_{d\in d(k)}\left[1-\frac{\gamma_d(i)-\sum_{j=1}^{i-1}\beta_d(i-j)A_k(j)-\beta_d(0)x}{A_k(0)}\right]-1\right\}=0 \tag{18}$$

rectly taking into account the presence of correlation. Moreover, we are studying the effect of the choice of the bin size on the accuracy of the results.

Finally, we believe that our inference technique can shed light on the behavior and dynamics of per link delay and so allow us to develop accurate link delay models. This will be also object of future work.

We feel that multicast based delay inference is an effective approach to perform delay measurements. The techniques developed are based on rigorous statistical analysis and, as our results show, yield representative delay estimates for all traffic that experiences the same per node behavior as multicast probes. The approach does not depend on cooperation from network elements and, because of bandwidth efficiency of multicast traffic, it is well suited to cope with the growing size of today's networks.

APPENDIX

The following Lemma shows that we recover $A_k(i)$ from the second largest solution of (8).

Lemma 1: Let $x_1 \geq x_2 \geq \cdots \geq x_m$, $m \leq \#d(k)$ denote the real solutions of (18), shown at the top of the page.Then $x_2 = A_k(i)$.

Proof: Substitute $x = A_k(i) + yA_k(0)$ into (18) to obtain

$$\gamma_k(i)+A_k(0)\left\{\prod_{d\in d(k)}[1-\beta_d(i)+\beta_d(0)y]-1\right\}$$
$$+\sum_{j=1}^{i-1}A_k(j)\left\{\prod_{d\in d(k)}[1-\beta_d(i-j)]-1\right\}$$
$$+(A_k(i)+yA_k(0))\left\{\prod_{d\in d(k)}[1-\beta_d(0)]-1\right\}=0. \tag{19}$$

To prove the lemma, we simply need to show that $y = 0$ is the second largest solution of (19). Expanding the product in the second term, after some algebra we obtain

$$\tilde{\theta}_{k,i}(y)=\sum_{l=1}^{\#d(k)}y^l\sum_{b\in B_l}\prod_{m=1}^{\#d(k)}(1-\beta_{d_m}(i))^{b_m}\beta_{d_m}(0)^{1-b_m}$$
$$+y\left\{\prod_{d\in d(k)}[1-\beta_d(0)]-1\right\}=0$$

where $b=\{b_1,\ldots,b_{\#d(k)}\}$ and $B_l=\{b\in\{0,1\}^{\#d(k)}\setminus\{0\}^{\#d(k)}|\sum b_m=\#d(k)-l\}$. The coefficients of the polynomial $\tilde{\theta}_{k,i}(y)$ are all positive but the last which is negative. The proof follows by observing that, since $\tilde{\theta}_{k,i}(0)=0$, $\tilde{\theta}'_{k,i}(0)<0$ and $\tilde{\theta}''_{k,i}(y)>0$, $y\geq 0$, there is one and only one solution of (19) greater than zero. ■

REFERENCES

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

[2] R. Caceres, N. G. Duffield, and T. Friedman, "Impromptu measurement infrastructures using RTP," *Proc. IEEE Infocom '02*, pp. 23–27, June 2002.

[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," *Proc. IEEE Infocom '99*, pp. 23–25, Mar. 1999.

[5] R. Caceres, N. G. Duffield, J. Horowitz, F. Lo Presti, and D. Towsley, "Loss-based inference of multicast network topology," *Proc. 1999 IEEE Conf. Decision and Control*, pp. 3065–3070, Dec. 1999.

[6] M. 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.

[7] M. J. Coates and R. Nowak, "Network tomography for Internet delay estimation," *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, pp. 3409–3412, May 2001.

[8] R. L. Carter and M. E. Crovella, "Measuring bottleneck link speed in packet-switched networks," *Performance Evaluation*, vol. 27-28, pp. 297–318, Oct. 1996.

[9] A. Downey, "Using pathchar to estimate internet link characteristics," in *Proc. SIGCOMM 1999*, Cambridge, MA, Sept. 1999, pp. 241–250.

[10] N. Duffield, J. Horowitz, F. Lo Presti, and D. Towsley, "Network delay tomography from end-to-end unicast measurments," in *Proc. 2001 Tyrrhenian Int. Workshop on Digital Communications*, Taormina, Italy, Sept. 2001, pp. 576–595.

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

[12] Felix: Independent Monitoring for Network Survivability [Online]. Available: ftp://ftp.bellcore.com/pub/mwg/felix/index.html

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

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

[15] V. Jacobson. Pathchar – A Tool to Infer Characteristics of Internet Paths. [Online]. Available: ftp://ftp.ee.lbl.gov/pathchar

[16] C. Lawson and R. Hanson, *Solving Least Squares Problems*. Philadelphia, PA: SIAM, 1995.

[17] F. Lo Presti, N. G. Duffield, J. Horowitz, and D. Towsley, "Multicast-Based Inference of Network-Internal Delay Distributions," UMass CMPSCI, 99–55, 1999.

[18] 'J. Mahdavi, V. Paxson, A. Adams, and M. Mathis, "Creating a scalable architecture for internet measurement," in *Proc. INET '98*, Geneva, Switzerland, July 1998.

[19] M. Mathis and J. Mahdavi, "Diagnosing internet congestion with a transport layer performance tool," in Proc. INET '96, Montreal, ON, Canada, June 1996, pp. 86–91.

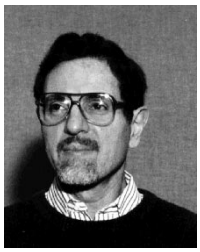
[20] D. Mills, "Network Time Protocol (Version 3): Specification, Implementation and Analysis," SRI International, Network Information Center, Menlo Park, CA, RFC 1305, 1992.
[21] S. Moon, P. Skelly, and D. Towsley, "Estimation and removal of clock skew from network delay measurements," in *Proc. Infocom '99*, New York, Mar. 1999, pp. 227–234.
[22] mtrace – Print Multicast Path from a Source to a Receiver [Online]. Available: ftp://ftp.parc.xerox.com/pub/net-research/ipmulti
[23] 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.
[24] ns – Network Simulator [Online]. Available: http://www-mash.cs.berkeley.edu/ns/ns.html
[25] V. Paxson, "End-to-end routing behavior in the internet," in *Proc. SIGCOMM '96*, Stanford, CA, Aug. 1996, pp. 601–615.
[26] ——, "End-to-end internet packet dynamics," in *Proc. SIGCOMM 1997*, Cannes, France, Sept. 1997, pp. 139–152.
[27] ——, "Automated packet trace analysis of TCP implementations," in *Proc. SIGCOMM 1997*, Cannes, France, Sept. 1997, pp. 167–179.
[28] ——, "On calibrating measurements of packet transit times," in *Proc. SIGMETRICS '98*, Madison, WI, June 1998.
[29] S. Ratnasamy and S. McCanne, "Inference of multicast routing tree topologies and bottleneck bandwidths using end-to-end measurements," *Proc. Infocom' 99*, pp. 353–360, Mar. 1999.
[30] M. J. Schervish, *Theory of Statistics*. New York: Springer, 1995.
[31] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A transport protocol for real-time applications," IETF RFC 1889, Jan. 1996.
[32] Surveyor [Online]. Available: http://io.advanced.org/surveyor/

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

He subsequently held a post-doctoral position in the Computer Science Department at the University of Massachusetts, Amherst. Since 2001, he has been an Assistant Professor in the Computer Science Department, Universitat dell'Aquila, Coppito, Italy. His research interests include measurements, modeing, and performance evaluation of computer and communication networks.

N. G. Duffield (M'97–SM'01) received the B.A. degree in natural sciences and the Certificat of Advanced Study in Mathematics from Cambridge University, 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 post-doctoral and faculty positions in Heidelberg, Germany, and Dublin, Ireland. He is currently a Technology Leader in the Internet and Networking Research group at AT&T Labs—Research, Florham Park, NJ. His current research focuses on Internet performance and measurement, inference and analysis.



Joe Horowitz received the B.S. degree from the Massachusetts Institute of Technology, Cambridge, in 1962 and the M.S. and Ph.D. degrees from the University of Michigan, Ann Arbor, in 1963 and 1967, respectively, all in mathematics.

He joined the Department of Mathematics and Statistics at the University of Massachusetts, Amherst, in 1969, where he has been a Professor since 1980. He has been a Visiting Faculty Member at Stanford University, Stanford, CA, the University of Strasbourg, Germany, ETH-Zurich, Zurich, Switzerland, and the Technion, Haifa, Israel, and a Fulbright Research Fellow (1988 and 1992) at the Indian Statistical Institute, New Delhi. His current research interests include statistical modeling and analysis of stochastic processes arising in communications networks, image analysis (primarily medical imaging), time series analysis, and decision analysis using Bayesian networks.

Don Towsley (M'78–SM'93–F'95) received the B.A. degree in physics and the Ph.D. degree in computer science from the University of Texas, Austin, in 1971 and 1975, respectively.

From 1976 to 1985, he was a Member of the Faculty of the Department of Electrical and Computer Engineering, University of Massachusetts, Amherst, where he is currently a Distinguished Professor in the Department of Computer Science. He has held visiting positions at IBM T.J. Watson Research Center, Yorktown Heights, NY (1982–1983), Laboratoire MASI, Paris, France (1989–1990), INRIA, Sophia-Antipolis, France (1996), and AT&T Labs—Research, Florham Park, NJ (1997). His research interests include networks, multimedia systems, and performance evaluation. He currently serves on the editorial boards of *Performance Evaluation* and *Journal of the ACM*.

Dr. Towsley has served on the editorial boards of the IEEE Transactions on Communications and the IEEE/ACM Transactions on Networking. He received the 1998 IEEE Communications Society William Bennet Paper Award and three Best Conference Paper Awards from ACM SIGMETRICS. He was a Program Co-Chair of the joint ACM SIGMETRICS and PERFORMANCE'92 Conference. He is a Member of ORSA and Chair of IFIP Working Group 7.3. He is a Fellow of the ACM.