Multicast-Based Inference of Network-Internal Delay Distributions*

F. Lo Presti[†,§] N.G. Duffield[†] J. Horowitz[‡] D. Towsley[§]

[†]AT&T Labs–Research
180 Park Avenue
Florham Park, NJ 07932, USA
{lopresti,duffield}@research.att.com

[‡]Dept. Math. & Statistics
University of Massachusetts
Amherst, MA 01003, USA
joeh@math.umass.edu

[§]Dept. of Computer Science
University of Massachusetts
Amherst, MA 01003, USA
towsley@gaia.cs.umass.edu

May 14, 2001

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 [4, 5], 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.

Keywords. End-to-end measurements, queueing delay, estimation theory, multicast tree, network tomography.

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

1 Introduction

Background and Motivation. Monitoring the performance of large communications networks is essential for diagnosing the causes of performance degradation. There are two broad approaches to monitoring. In the *internal* approach, direct measurements are made at or between network elements, e.g., of packet loss or delay. In the *external* approach, 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 the in the gathering of measurement data, and joining data collected from a large number of elements in order to form a composite view of end-to-end performance.

This motivates external approaches, network diagnosis 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 [3, 9, 18, 25, 26, 28]). Several research efforts are developing measurement infrastructures (Felix [12], IPMA [14], NIMI [17] and Surveyor [34]) 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 roundtrip 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` [16], 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 [10]. 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 [4, 5]. 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 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.

Contribution The work of [4, 5] 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 non-parametric discrete distributions. The choice of non parametric distributions 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 [2, 23, 26]), to the best of our knowledge there is no general model for per link delays. The use of a non-parametric 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. Formally, there is no difficulty in formulating a continuous version of the inference algorithm. However, it proceeds via inversion of Laplace transforms, a procedure that is in practice implemented numerically. In the discrete approach we can explicitly trade-off the detail of the distribution with the cost of calculation; the cost is inversely proportional to the bin widths of the discrete distribution.

The principle 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 in two sets of experiments. The first set used a 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.

In the second set of experiments we conducted an `ns` simulation of packets on a multicast tree. 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. In any case, by explicitly introducing spatial correlations into the model simulations, we were able to show that small violations of the independence assumption lead to only small inaccuracies of the estimated distribution. This continuity property of the deformation in inference due to correlations is also to be expected on theoretical grounds.

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.

Implementation Requirements Since the data for delay inference comprises one-way packet delays, the primary requirement is the deployment of measurement hosts with synchronized clocks. Global Positioning System (GPS) systems afford one way to achieve a 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) [19]. However, this provides accuracy only on the order of milliseconds at best, a resolution at least as coarse as the queueing delays in practice. An alternative approach that could supplement delay measurement from unsynchronized or coarsely synchronized clocks has been developed in [27, 29, 20]. These authors propose algorithms to detect clock adjustments and rate 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 respond-

ing 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 [4, 5] is to infer the logical multicast topology directly from measured probe statistics; see [30] and [7]. This method does not require cooperation from the network.

Structure of the Paper. The remaining sections of the paper are organized as follows. In Section 2 we describe the delay model and in Section 3 we derive the delay estimator. In Section 4 we describe the algorithm used to compute the estimator from data. In Section 5 we present the model and network simulations used to evaluate our approach. Section 6 concludes the paper.

2 Model & Framework

2.1 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. The logical multicast tree comprises the branch points of the physical tree, and the logical links between them. The logical links comprise one or more physical links. Thus each node in the logical tree, except for the leaf nodes and the root, must have 2 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 and receivers, and the set of links L, which are ordered pairs (j, k) of nodes, indicating a link from j to k. We will let $U = V \setminus \{0\}$. 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 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 by $f^n(k) = f(f^{n-1}(k))$ 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 non-negative 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.

2.2 Modeling Delay and Loss of Probe Packets

Probe packets are sent down the tree from the root node 0. 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 0 to a node k is $Y_k = \sum_{j \succeq k} D_j$. Note that $Y_k = \infty$ iff $D_j = \infty$ for some $j \succeq k$, i.e. if the packet was lost on some link between node 0 and k.

Unless otherwise stated, we will discretize each link delay D_k to a set $\{0, q, 2q, \ldots, i_{\max}q, \infty\}$. Here q is the bin width, $i_{\max} + 1$ is the number of bins, and the point ∞ is interpreted as "packet lost" or "encountered delay greater than $i_{\max}q$". The distribution of D_k is denoted by α_k, where $\alpha_k(i) = \mathsf{P}[D_k = iq]$ with $\alpha_k(\infty)$ the probability that $D_k = \infty$. For each link, we denote u_k 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 $k \in V$, the cumulative delay process Y_k, $k \in V$, takes values in $\{0, q, 2q, \ldots, i_{\max}q\ell(k), \infty\}$, i.e., it supports addition in the ranges of the constituent D_j. We set $A_k(i) = \mathsf{P}[Y_k = iq]$ with $A_k(\infty)$ the probability that $Y_k = \infty$. 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 non-zero probability that a probe experiences no delay in traversing each link.

3 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 $(Y_{k,l})_{k \in R, l=1,\ldots,n}$. 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 3.1 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 3.2 we present the proof of the main result which also provides the construction of the algorithm to compute the estimator we present in Section 4. In Section 3.4 we analyze the rate of convergence of the estimator as the number of probes increases.

3.1 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 $\{\min_{j \in R(k)} Y_j \leq iq\}$ that the end-to-end delay is no greater than iq for at least least one receiver in $R(k)$. Let $\gamma_k(i) = \mathsf{P}[\Omega_k(i)]$ denote its probability. Finally let Γ denote the mapping associating the link distributions $(\alpha_k(i))_{k \in U, i \in \{0,\ldots,i_{\max}\}}$ 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* $\mathcal{A} = \{\alpha = (\alpha_k(i))_{k \in U, i \in \{0,\ldots,i_{\max}\}} : \alpha_k(0) > 0, \ \sum_{i \leq i_{\max}} \alpha_k(i) \leq 1\}$ *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 bijection from* $\mathcal{A}$ *to* $\mathcal{G}$ *which is continuously differentiable and has a continuously differentiable inverse.*

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

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

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

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

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

Let $\mathcal{A}^{(1)} = \{\alpha = (\alpha_k(i))_{k \in U, i \in \{0,\ldots,i_{\max}\}} : \alpha_k(0) > 0, \ \sum_{i \leq i_{\max}} \alpha_k(i) < 1\}$ 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 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 $n \to \infty$. ∎

3.2 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.

3.2.1 Relating γ to α

Let $\beta_k(i) = \mathsf{P}[\min_{j \in R(k)} Y_j - Y_{f(k)} \leq iq]$, $i = 0, \cdots, i_{\max}$, so $\beta_k(i)$ obeys the recursion

$$\begin{array}{ll} \beta_k(i) = \sum_{j=0}^{i} \alpha_k(j) \left[1 - \prod_{d \in d(k)} (1 - \beta_d(i-j))\right] & k \in U \setminus R \\ \beta_k(i) = \sum_{j=0}^{i} \alpha_k(j) & k \in R. \end{array} \tag{3}$$

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

$$\begin{array}{ll} \gamma_k(i) = \sum_{j=0}^{i} A_k(j) \left[1 - \prod_{d \in d(k)} (1 - \beta_d(i-j))\right] & k \in U \setminus R \\ \gamma_k(i) = \sum_{j=0}^{i} A_k(j) & k \in R \end{array} \tag{5}$$

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 [4] (by identifying the event *no delay* with the event *no loss*).

3.2.2 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). We postpone to Appendix A the proof that the inverse is unique and continuously differentiable. 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

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

and

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

These equations are formally identical to those for the loss estimator [4]. From [4], 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 expression (5) we replace $\beta_d(i)$ with $\frac{\gamma_d(i) - \sum_{j=1}^{i-1} \beta_d(i-j)A_k(j) - \beta_d(0)A_k(i)}{A_k(0)}$ (from (4)) and obtain the following equation

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

(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 Appendix A we consider 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 $\beta_k(i) = \frac{\gamma_k(i) - \sum_{j=1}^{i} A_{f(k)}(j)\beta_k(i-j)}{A_{f(k)}(0)}$.

Computing α

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

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

3.3 Example: the Two-leaf Tree

In this section we illustrate the application of the results of Section 3.1 to the two-leaf tree of Figure 1. We assume that on each link, a probe either suffers no delay, a unit amount of delay, or is otherwise lost; for each link, therefore, the delay takes values in $\{0, 1, \infty\}$.

For this example, equations (6) and (8) can be solved explicitly. Combined (6) and (8) with (9)

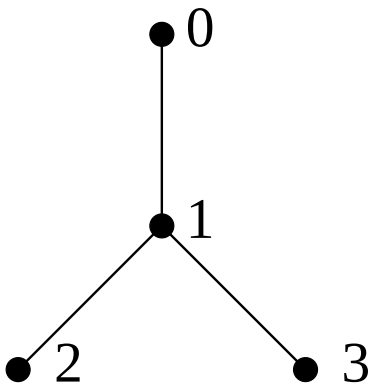


Figure 1: TWO-LEAF MULTICAST TREE.

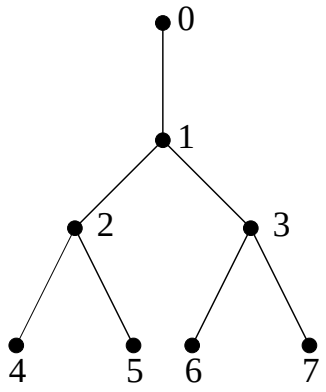


Figure 2: FOUR-LEAF MULTICAST TREE.

we obtain the estimates

$$
\begin{aligned}
\widehat{\alpha}_1(0) &= \frac{\widehat{\gamma}_2(0)\widehat{\gamma}_3(0)}{\widehat{\gamma}} \\
\widehat{\alpha}_2(0) &= \frac{\widehat{\gamma}}{\widehat{\gamma}_3(0)} \\
\widehat{\alpha}_3(0) &= \frac{\widehat{\gamma}}{\widehat{\gamma}_2(0)} \\
\widehat{\alpha}_1(1) &= \frac{1}{2}\frac{\widehat{\gamma}_2(0)\widehat{\gamma}_3(0)}{\widehat{\gamma}}\left(\frac{\widehat{\gamma}_2(1)}{\widehat{\gamma}_2(0)}+\frac{\widehat{\gamma}_3(1)}{\widehat{\gamma}_3(0)}-1-\sqrt{\left(\frac{\widehat{\gamma}_2(1)}{\widehat{\gamma}_2(0)}+\frac{\widehat{\gamma}_3(1)}{\widehat{\gamma}_3(0)}-1\right)^2-4\frac{\widehat{\gamma}_2(1)\widehat{\gamma}_3(1)}{\widehat{\gamma}_2(0)\widehat{\gamma}_3(0)}+4\frac{\widehat{\gamma}_2(1)+\widehat{\gamma}_3(1)-\widehat{\gamma}_1(1)}{\widehat{\gamma}}}\right) \\
\widehat{\alpha}_2(1) &= \frac{\widehat{\gamma}_2(1)-\widehat{\gamma}_2(0)-\widehat{\alpha}_1(1)\widehat{\gamma}/\widehat{\gamma}_3(0)}{\widehat{\gamma}_2(0)\widehat{\gamma}_3(0)}\widehat{\gamma} \\
\widehat{\alpha}_3(1) &= \frac{\widehat{\gamma}_3(1)-\widehat{\gamma}_2(0)-\widehat{\alpha}_1(1)\widehat{\gamma}/\widehat{\gamma}_2(0)}{\widehat{\gamma}_2(0)\widehat{\gamma}_3(0)}\widehat{\gamma}
\end{aligned}
$$

where $\widehat{\gamma} = \widehat{\gamma}_2(0) + \widehat{\gamma}_3(0) - \widehat{\gamma}_1(0)$.

3.4 Rate of Convergence of the Delay Distribution Estimator

3.4.1 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 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 $\sigma_{(k_1,i)(k_2,j)} = \lim_{n\to\infty} n\mathsf{Cov}(\widehat{\gamma}_{k_1}(i), \widehat{\gamma}_{k_2}(j))$, for $k_1, k_2 \in U$, $i, j \in \{0, \ldots, i_{\max}\}$, $D_{(k_1,i)(k_2,j)}(\alpha) = \frac{\partial \Gamma_{k_1}^{-1}(i)}{\partial \gamma_{k_2}(j)}(\Gamma(\alpha))$ 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

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

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 existst 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 Chapter 7 of [33]) yields that $\widehat{\alpha} = \Gamma^{-1}(\widehat{\gamma})$ is also asymptotically Gaussian as $n \to \infty$:

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

■

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: (i) the number of probes required to obtain a desired level of accuracy of the estimate; (ii) 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

$$\alpha_k(i) \pm z_{\delta/2} \sqrt{\frac{\nu_{(k,i)(k,i)}}{n}}, \tag{10}$$

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

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

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

$$\alpha_k(i) \pm z_{\delta/2} \sqrt{\frac{\widehat{\nu}_{(k,i)(k,i)}}{n}}. \tag{11}$$

3.4.2 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

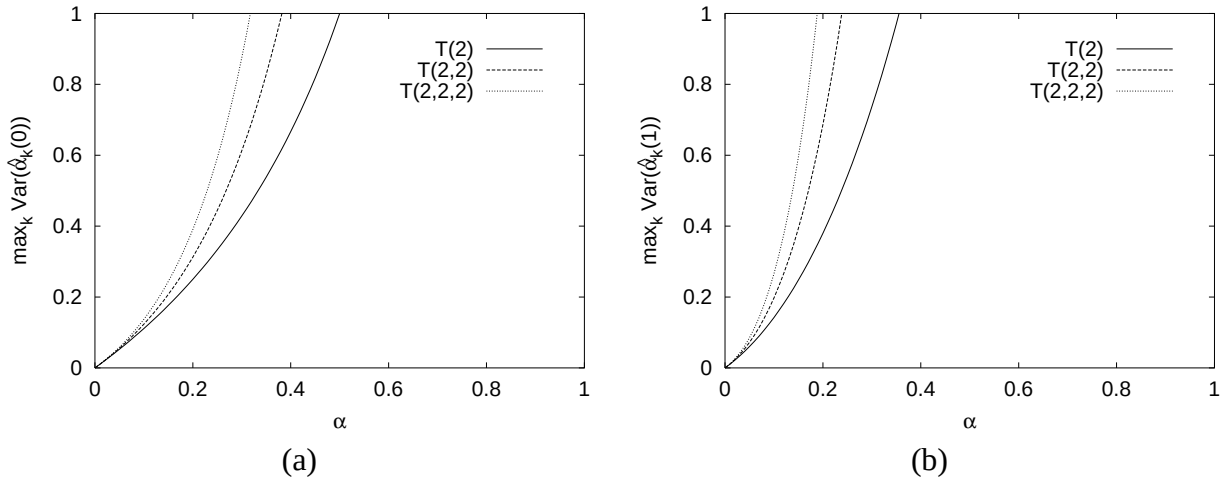


Figure 3: ASYMPTOTIC ESTIMATOR VARIANCE AND TREE DEPTH. Binary trees of depth 2, 3 and 4. Left: Minimum and Maximum Variance of the estimates $\widehat{\alpha}_k(0)$ (a) and $\widehat{\alpha}_k(1)$ (b) over all links.

parameters, namely, the delay distributions and the multicast tree topology. The following Theorem, the proof of which we postpone to Appendix C, characterizes the behavior of the variance for small delays. Set $\| \alpha \| = \max_{k \in U, i>0} \alpha_k(i)$.

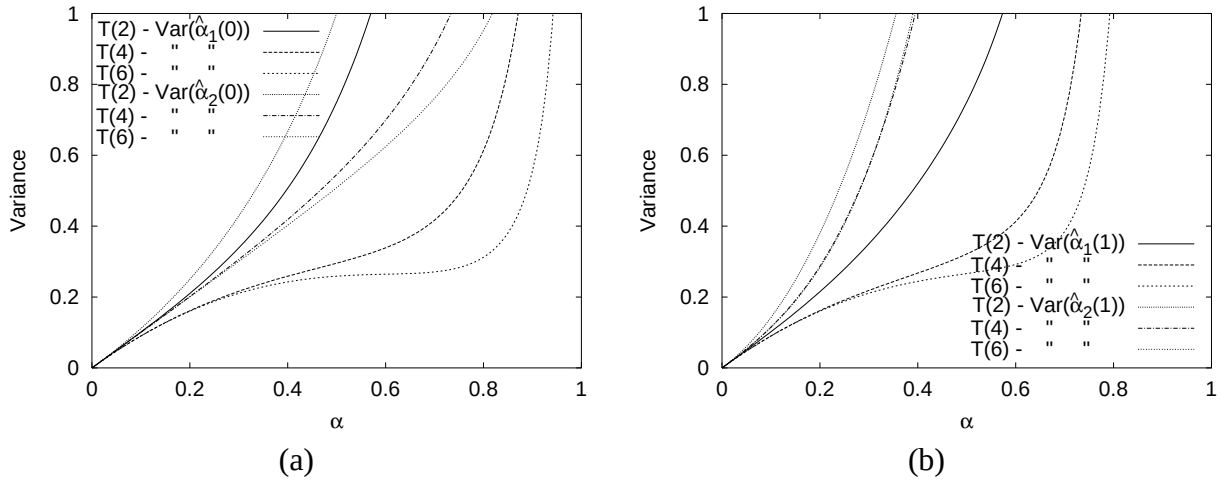
Theorem 4 *As* $\| \alpha \| \to 0$,

$$\nu = \begin{pmatrix} \nu_{k_1 k_1} & 0 & 0 & \dots & 0 \\ 0 & \nu_{k_2 k_2} & 0 & \dots & 0 \\ 0 & 0 & \nu_{k_3 k_3} & \dots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \dots & \nu_{k_{\#U} k_{\#U}} \end{pmatrix} + O(\| \alpha \|^2) \text{ with } \nu_{kk} = \begin{pmatrix} \sum_{i>0} \alpha_k(i) & \alpha_k(1) & \alpha_k(2) & \dots & \alpha_k(i_{\max}) \\ \alpha_k(1) & \alpha_k(1) & 0 & \dots & 0 \\ \alpha_k(2) & 0 & \alpha_k(2) & \dots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ \alpha_k(i_{\max}) & 0 & 0 & \dots & \alpha_k(i_{\max}) \end{pmatrix} \tag{12}$$

.

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 $T(r_1, \dots, r_m)$ to denote a tree of $m+1$ levels where, 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 function of $\alpha_k(1) = \alpha$.

In Figure 3 we show the dependence on tree depth for binary trees of depth 2, 3 and 4. We plot (a) the maximum value $\max_k \mathsf{Var}(\widehat{\alpha}_k(0))$ of the variance over the links, and (b) $\max_k \mathsf{Var}(\widehat{\alpha}_k(1))$ (b). In these examples, the variance increases with the tree depth. In Figure 4 we show the dependence on branching ratio for a tree of level 2. We plot the estimator variance for both link 1 (the



(a) (b)

Figure 4: ASYMPTOTIC ESTIMATOR VARIANCE AND BRANCHING RATIO. Binary tree with depth 2 and 2, 4 or 6 receivers. Left: Variance of $\widehat{\alpha}_k(0)$ (a) and $\widehat{\alpha}_k(1)$ (b) for link 1 (common link) and 2 (generic receiver).

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. The first can be explained in terms of the increased number of measurements available for the estimation as the number of receivers sharing a given link increases; the second appears to be the effect of cumulative errors that accrue as the number of links along a path increases (α is computed iteratively on the tree). We also observe that the variance increases with the delay lag; this appears to be caused by the iterative computation on the number of bins that progressively cumulate errors.

4 Computation of the Delay Distribution Estimator

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 also discuss its suitability for distributed implementation and how to adapt the computation to handle different bin sizes.

We assume we have the experimental data of source-to-leaf delays $(Y_{k,m})_{k \in R, m=1,\dots n}$ 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: (i) removal of fixed delays and (ii) choosing a bin size q and computing the estimate $\widehat{\gamma}$.

The first step is necessary since 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 in three unknowns). Thus we normalize each measurement by subtracting the minimum delay seen at the leaf. Observe that to interpret the observed minimum delay as the transmission delay assumes that at least one probe has experienced no queuing delay along the path.

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 trade-off 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 $(Y_k)_{k\in R}$, we recursively construct the auxiliary vector process $\widehat{Y} = (\widehat{Y}_k)_{k\in V}$:

$$\widehat{Y}_{k,l} = Y_{k,l} - \min_{m\in\{1,\ldots,n\}} Y_{k,m},\ k \in R \tag{13}$$

$$\widehat{Y}_{k,l} = \min_{j\in d(k)} \widehat{Y}_{j,l},\ k \in V \setminus R. \tag{14}$$

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

$$\widehat{\gamma}_k(i) = n^{-1} \sum_{m=1}^{n} \mathbf{1}_{\{\widehat{Y}_{k,m} \le iq+q/2]\}},\ i = 0, \ldots, i_{\max}, \tag{15}$$

with

$$i_{\max} = \left\lceil \frac{\max_{k\in R} \max_{m\in N_k(n)} \widehat{Y}_{k,m}}{q} \right\rceil .$$

Here $\lceil x \rceil$ denotes the smallest integer greater than x and $N_k(n) = \{m \in \{1,\ldots,n\} | Y_{k,m} < \infty\}$. Observe that i_{max} represents the largest value at which the estimate $\widehat{\alpha}(i_{\max})$ is non zero.

The estimate can be computed iteratively over the delay lag and recursively over the tree. The pseudo code for carrying out the computation is found in Figure 5. The procedure `find_y` calculates $\widehat{Y}_k$ and $\widehat{\gamma}_k$, with $\widehat{Y}_{k,l}$ initialized to $Y_{k,l} - \min_{m\in\{1,\ldots,n\}} Y_{k,m}$ for $k \in R$ and ∞ (a value larger than any observed delay suffices) otherwise. The procedure `infer_delay` calculates $\widehat{\alpha}_k(i)$

```
procedure main {
    find_y ( 1 ) ;
    foreach ( i ∈ {0, ··· , i_max} )
        infer_delay ( 1 , i ) ;
}

procedure find_y ( k ) {
    foreach ( j ∈ d(k) ) {
        Ŷ_j = find_y ( j ) ;
        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}};
    return Ŷ_k ;
}

procedure infer_delay ( k, i ) ;
    if ( i == 0 ) {
        Â_k[i] = solvefor1 ( Â_k[i] , (1 − γ̂_k[i]/Â_k[i] == ∏_{d∈d(k)} 1 − γ̂[i]/Â[i] ) ;
    } else {
        Â_k[i] = solvefor2 ( Â_k[i] , 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 ) ;
    }
    β̂_k[i] = (γ̂_k[i] − Σ_{j=1}^{i} Â_{f(k)}[j] β̂_k[i−j]) / Â_{f(k)}[0] ;
    α̂_k[i] = (Â_k[i] − Σ_{j=1}^{i} Â_{f(k)}[j] α̂_k[i−j]) / Â_{f(k)}[0] ;
    foreach ( j ∈ d(k) )
        infer_delay ( j , i ) ;
}
```

Figure 5: PSEUDOCODE FOR INFERENCE OF DELAY DISTRIBUTION.

for a fixed i recursively on the tree, with $\widehat{A}_k[i]$, $k \in V$, $i = 0, \cdots, i_{\max}$ initialized to 0, except for $\widehat{A}_0[0]$ set to 1. The output of the algorithm is the estimates $\widehat{\alpha}_k$, $k \in U$.

Within the code, an empty product (which occurs when the first argument of `infer` is a leaf) is assumed to be zero. The routines `solvefor1` and `solvefor2` return the value of the first symbolic argument that solves the equation in the second argument. `solvefor1` returns a solution in $(0, 1)$; from Lemma 1 in [4] this is known to be unique. `solvefor2` returns the unique solution if the second argument is linear in $\widehat{A}_k(i)$ (this happen only if k is a leaf-node), otherwise it returns the second largest solution.

4.1 Distributed Implementation

As with the loss estimator [4] the algorithm is recursive on trees. In particular, observe that the computation of $\widehat{\gamma}$ and $\widehat{A}_k$ only requires the knowledge of $(\widehat{Y}_{j,m})_{j\in d(k),m=1,\ldots,n}$; these are computed recursively on the the tree starting from the receivers. Therefore it is possible to distribute the computation among the nodes of the tree (or representative nodes of subtrees), with each node k being responsible for the aggregation of the measurements of its child nodes through (14) and for the computation of $\widehat{A}_k$.

4.2 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. A drawback of this approach is that each distribution is computed for all the different bin sizes. The distributed nature of the algorithm suggests we can do better; indeed, since A_k, $k \in U$, can be computed independently from one another, it is possible to compute each link delay distribution only for the bin size best suited to its delay characteristics. More precisely, let q_k denote the bin size adopted for link k. In order to compute $\widehat{\alpha}_k$ with bin size q_k we need to compute both $\widehat{A}_k$ and $\widehat{A}_{f(k)}$ with bin size q_k. Thus, the overall computation requires calculating each cumulative distribution $\widehat{A}_k$ only for the bin sizes q_j, $j \in d(k) \cup \{k\}$, *i.e.*, only for the bin sizes adopted for the links terminating at node k and at all its child nodes rather than for bin sizes adopted for all links.

In an implementation, we envision that a fixed value for all links is 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.

5 Experimental Evaluation

We evaluated our delay estimator through extensive simulation. Our first set of experiments focuses 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. In

our second set of experiments 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 queuing delay and queue overflows at network nodes as multicast probes compete with traffic generated by TCP/UDP traffic sources.

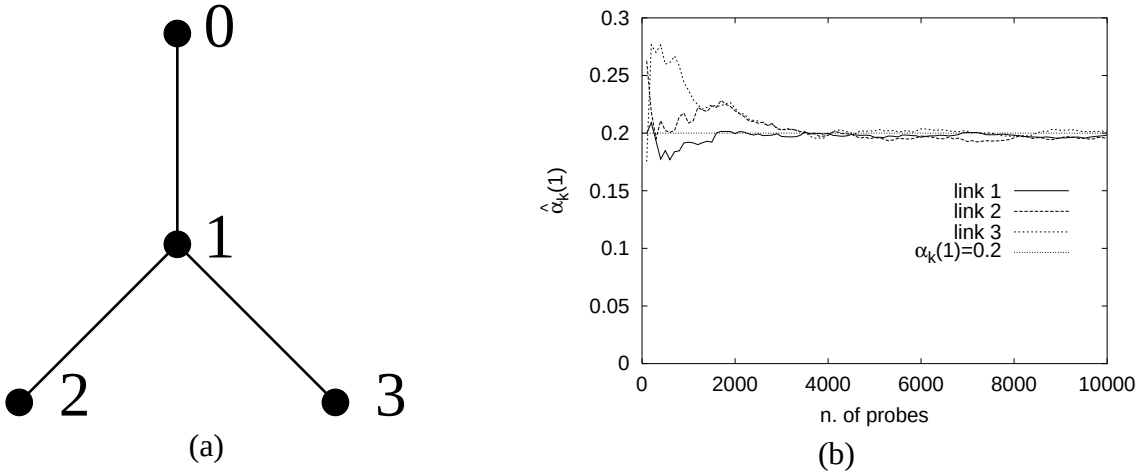
5.1 Comparing Inferred vs. Sample Distributions

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 $k \in V$, the inferred distribution $\widehat{\alpha}_k$ ($\widehat{A}_k$) is computed from the end-to-end measurements using the algorithm described in Section 4. Its accuracy must be measured against the actual data, represented by a finite sequence of delays $\{D_{k,m}\}_{m=1}^{n}$ ($\{Y_{k,m}\}_{m=1}^{n}$), 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.

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.

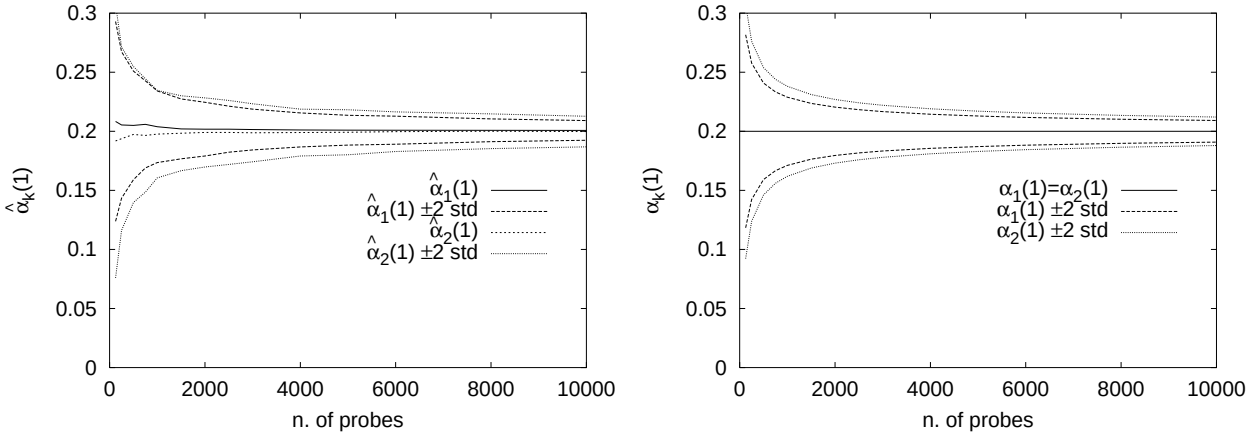
We compute the sample distributions $\tilde{\alpha}$ and $\tilde{A}$ using the same bin size q of the estimator. More precisely, we compute $(\tilde{\alpha}_k)_{k\in V}$ and $(\tilde{A}_k)_{k\in V}$ as $\tilde{\alpha}_k(i) = \#N_{f(k)}(n)^{-1} \sum_{j\in N_{f(k)}(n)} \mathbf{1}_{\{D_{k,j}\in(id-d/2,id+d/2]\}}$, $i = 0, \ldots, i_{\max}$ ($\tilde{\alpha}_k(\infty) = \#N_{f(k)}(n)^{-1} \sum_{j\in \#N_{f(k)}(n)} \mathbf{1}_{\{D_{k,j}=\infty\}}$) and $\tilde{A}_k(i) = n^{-1} \sum_{j=1}^{n} \mathbf{1}_{\{Y_{k,j}\in(id-d/2,id+d/2]\}}$, $i = 0, \ldots, i_{\max}$ ($\tilde{A}_k(\infty) = n^{-1} \sum_{j=1}^{n} \mathbf{1}_{\{Y_{k,j}=\infty\}}$). (Observe that in computing $(\tilde{\alpha}_k)_{k\in V}$, the sum is carried out only over $N_{f(k)}(n) = \{j \in \{1, \ldots, n\} | Y_{f(k),j} < \infty\}$, 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 $\sum_{j\le i} \tilde{\alpha}_k(i)$ lies between $\sum_{j\le i} \widehat{\alpha}_k(i) \pm \Delta_k$ $i = 0, \ldots, i_{\max}$. The same result holds for the tail probabilities, $\sum_{j>i} \alpha_k(i)$.



(a) (b)

Figure 6: TWO RECEIVERS TREE. (a): Simulation topology. (b): Convergence of $\widehat{\alpha}_k(1)$ to $\alpha_k(1)$. $\alpha_k(0) = 0.79$, $\alpha_k(1) = 0.2$ and $\alpha_k(\infty) = 0.01$ for all links.



(a) (b)

Figure 7: AGREEMENT BETWEEN SIMULATED AND THEORETICAL CONFIDENCE INTERVALS. (a): Results from 100 model simulations. (b): Prediction from (10). The graphs show two-sided confidence interval at 2 standard deviation for link 1 and 2. Parameters are $\alpha_k(1) = 0.2$ and $\alpha_k(\infty) = 0.01$ for all links.

5.2 Model Simulation

We first consider the two-leaf topology of Figure 6(a), with source 0 and receivers 2 and 3. Link delays are independent, taking values in $\{0, 1, \infty\}$; if a probe is not lost it experiences either no delay or unit delay. In Figure 6(b) we plot the estimate $\widehat{\alpha}_k(1)$ versus the model values for a run comprising 10000 probes. The estimate converges within 2% of the model value within 4000 probes. In Figure 7 we compare the empirical and theoretical 95% confidence intervals. The theoretical intervals are computed from (10). The empirical intervals are computed over 100 independent simulations. The agreement between simulation and theory is close: the two sets of curves are almost indistinguishable.

Next we consider the topology of Figure 8. Delays are independently distributed according to a truncated geometric distribution taking values in $\{0, 1, \ldots, 40, \infty\}$ (in ms) . This topology is also used in subsequent TCP/UDP simulations, and the link average delay and loss probability are chosen to match the values obtained from these. The average delay ranges between 1 and 2ms for the slower *edge* links and between 0.2 and 0.5ms for the *interior* faster links; the link losses range from 1% to 11%. In Figure 9 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 Figure 10 where we plot the minimum, median and the maximum largest absolute deviation in 100 simulations computed over all links as a function of n (a) and link by link for $n = 10000$ (b). 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 $n = 10000$, 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 3.4 and is explained in terms of the larger inferred variances of downstream with respect to upstream nodes.

5.3 TCP/UDP Simulations

We used the topology shown in Figure 8. To capture the heterogeneity between edges and core of a WAN, interior links have higher capacity (5Mb/sec) and propagation delay (50ms) than those at

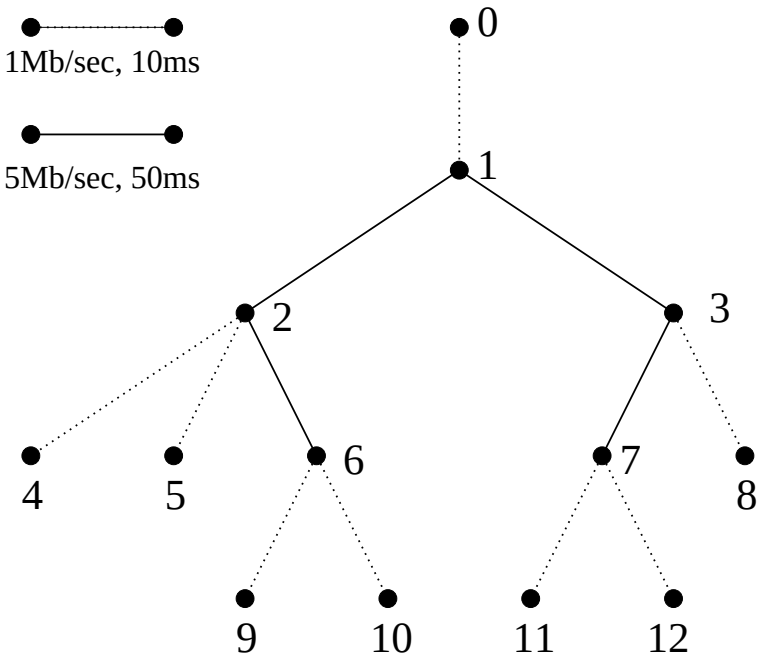
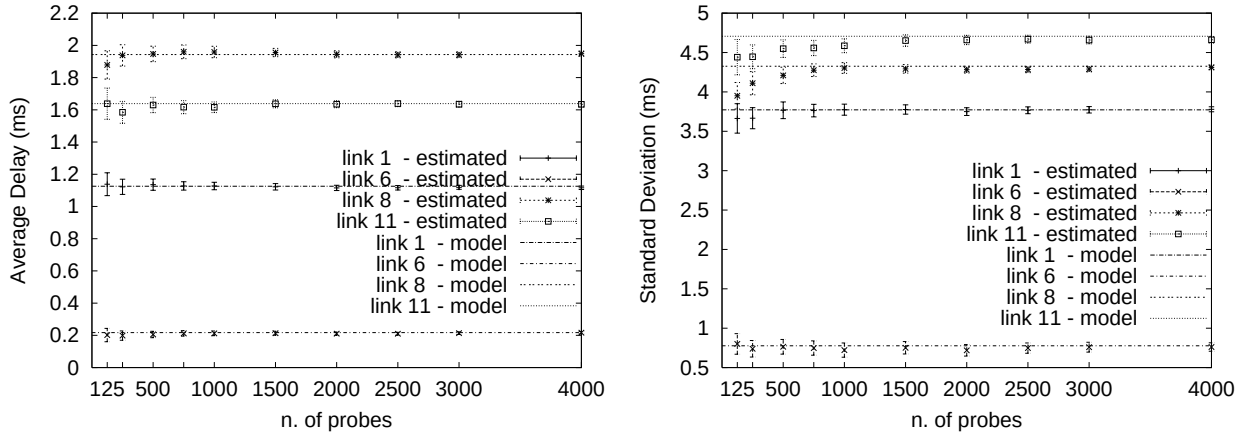


Figure 8: Simulation Topology: Link are of two types: *edge* links of 1MB/s capacity and 10ms latency, and *interior* links of 5Mb/s capacity and 50ms latency.



(a) (b)

Figure 9: MODEL SIMULATION: TOPOLOGY OF FIGURE 8. ESTIMATED VERSUS THEORETICAL DELAY AVERAGE AND STANDARD DEVIATION WITH 95% CONFIDENCE INTERVAL COMPUTED OVER 100 MODEL SIMULATIONS.

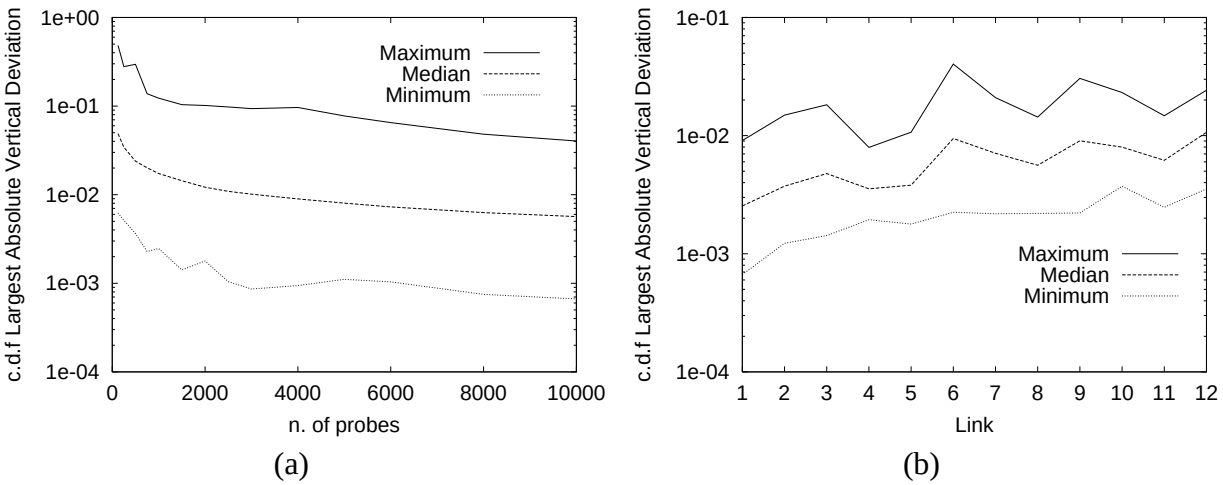


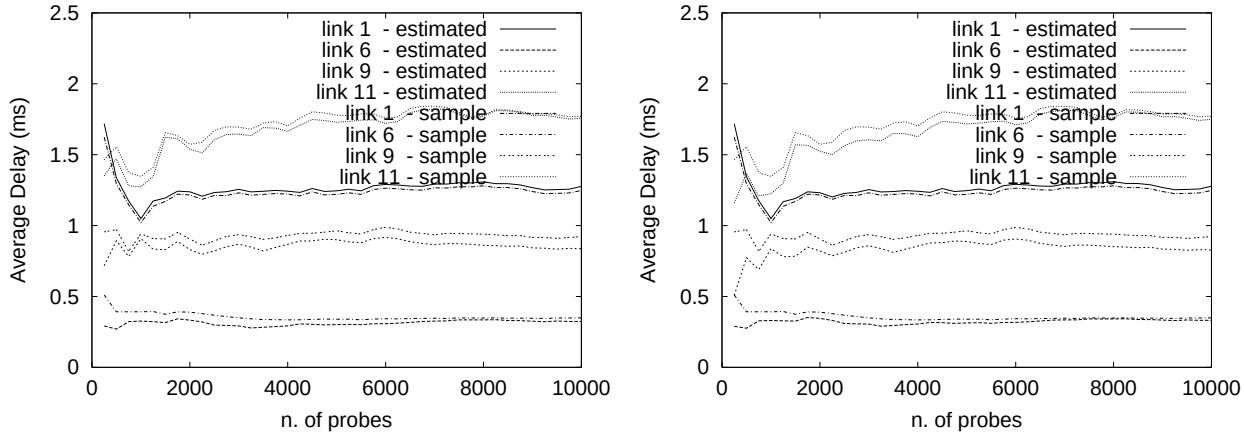
Figure 10: MODEL SIMULATION: TOPOLOGY OF FIGURE 8. ACCURACY OF THE ESTIMATED DISTRIBUTION. LARGEST VERTICAL ABSOLUTE DEVIATION BETWEEN ESTIMATED AND SAMPLE C.D.F. Minimum, median and the maximum largest absolute deviation in 100 simulations computed over all links as function of n (a) and link by link for $n = 10000$ (b).

the edge (1Mb/sec and 10ms). Each link is modeled as a FIFO queue with a 4-packet capacity.

Node 0 generates probes as a 20Kbit/s stream comprising 40 byte UDP packets according to a Poisson process with a mean interarrival time of 16ms; this represents 2% of the smallest link capacity. Observe that, even for this simple topology with 8 end-points, a mesh of unicast measurements with the same traffic characteristics would require an aggregate bandwidth of 160Kbit/s at the root. The background traffic comprises a mix of infinite data source TCP connections (FTP) and exponential on-off sources using UDP. Averaged over the different simulations, the link loss ranges between 1% and 11% and link utilization ranges between 20% and 60%.

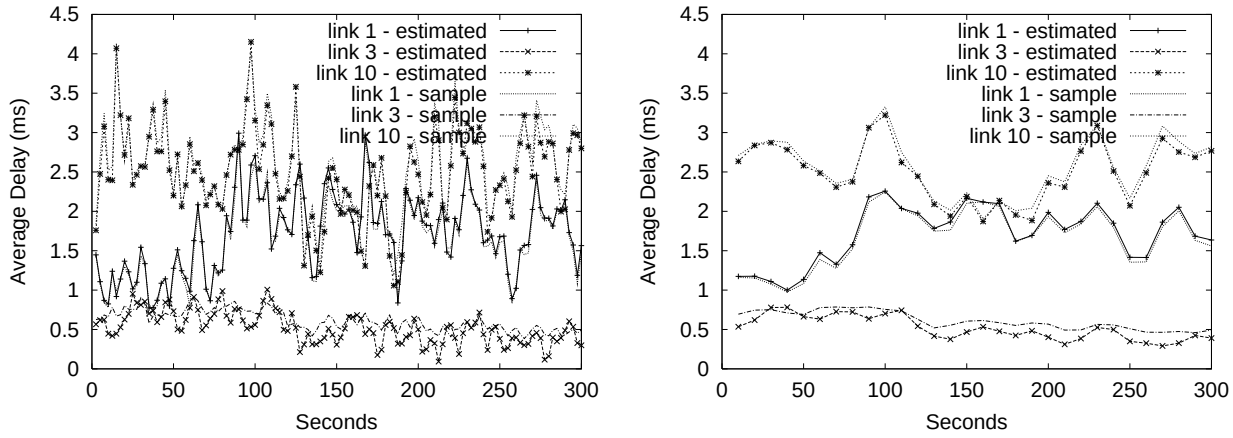
For a single experiment, Figure 11 compares the estimated versus the sample average delay for representative selected links. The analysis has been carried out using $q = 1ms$ (a) and $q = 0.1ms$ (b). In this example, we obtain practically the same accuracy despite a tenfold difference in resolution. (Observe that $q = 1ms$ is the same order of magnitude as the average delays.) The inferred 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 Figure 12 we plot the inferred versus the sample average delay for 3 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



(a) (b)

Figure 11: CONVERGENCE OF INFERRED VERSUS SAMPLE AVERAGE LINK DELAY IN TCP/UDP SIMULATIONS. (a): bin-size $q = 1ms$. (b): bin size $q = 0.1ms$. The graphs shows how the inferred values closely track the sample average delays.



(a) (b)

Figure 12: DYNAMIC ACCURACY OF INFERENCE. Sample and Inferred average delay on links 1, 3 and 10 of the multicast tree in Figure 8. (a): 5 seconds window. (b): 20 second windows. Background traffic has random start and stop times.

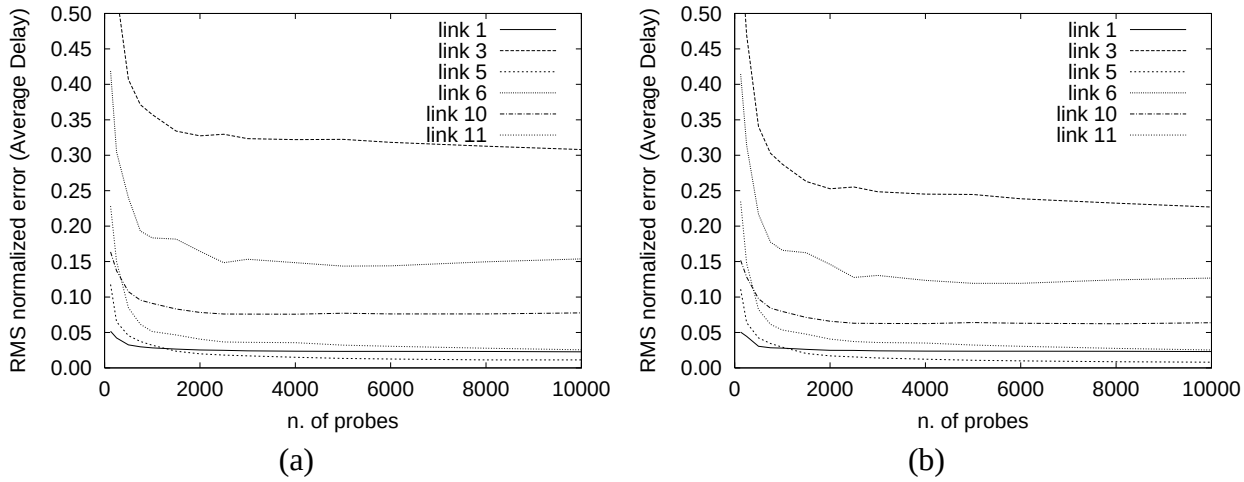


Figure 13: ACCURACY OF INFERENCE: AVERAGE DELAY. Left: $q = 1ms$. Right: $q = 0.1ms$. The graphs show the normalized Root Mean Square error between the estimated and sample average delay over 100 simulations.

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 second windows.

For a selection of links, in Figure 13 we plot the Root Mean Square (RMS) normalized error between the estimated and sample average delays calculated over 100 simulations using $q = 1ms$ and $q = 0.1ms$. The two plots demonstrate that the error drops significantly as the number of probes increases 2000, after which it becomes almost constant. In this example, increasing the resolution by a factor of ten somewhat improves the overall accuracy of the estimates, especially for those links that enjoy smaller delays. After 10000 probes the relative error ranges from 1% to 23%. The higher values occur when link average delays are small due to the fact that for these links the same absolute error results in a more pronounced relative error.

The persistence of systematic errors we observe in Figure 13 is due to the presence of spatial correlation. In our simulations, a multicast probe is more likely to experience 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. From Figure 13 we verify that, not surprisingly node 5 enjoys the smallest relative error. We believe that these high correlations are a result of the small scale of the simulated network. We have observed smaller correlations in large simulations as would be expected in real networks because of the wide traffic and link diversity.

The autocorrelation function rapidly decreases and can be considered negligible for a lag larger than 30 (approximatively 2 seconds). 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 18000 probes were generated, we plot the complementary c.d.f. conditioned on the delay being finite in Figure 14. In Figure 15 we also plot the complementary c.d.f of the node cumulative delay. (We show only the internal links as $\widehat{A}_k(i) = \widehat{\gamma}_k(i), k \in R$.) Here $q = 1ms$.

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 capture the relationship between the actual distributions correctly. We verified this by convolving α_k and $A_{f(k)}$ and comparing the result with A_k; as expected, in the presence of strong local correlation, the results exhibit significant differences that account for the discrepancies of the inferred distributions. Nevertheless, the results should be affected in a continuous way with small violations leading to small inaccuracies. Indeed, we have good agreement for the inferred distributions of links 4, 5, 10 and 12 that are the nodes with smallest spatial correlations. Unfortunately, it is not easy to determine whether the correlations are strong and therefore to assess the expected accuracy of the estimates, even though pathological shapes of the inferred distributions could provide evidence of strong local correlations[1]. A solution to this problem could be the extension of the model to explicitly account for the presence of spatial correlation in the analysis. This will be the focus of future research.

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

[1]To this end, we observed that under, strong spatial correlation, inaccuracies of the estimator $\widehat{\alpha}$ are often associated with the existence of significant increasing behavior in portions of the complementary c.d.f. that leads to negative inferred probabilities with possibly non negligible absolute values.

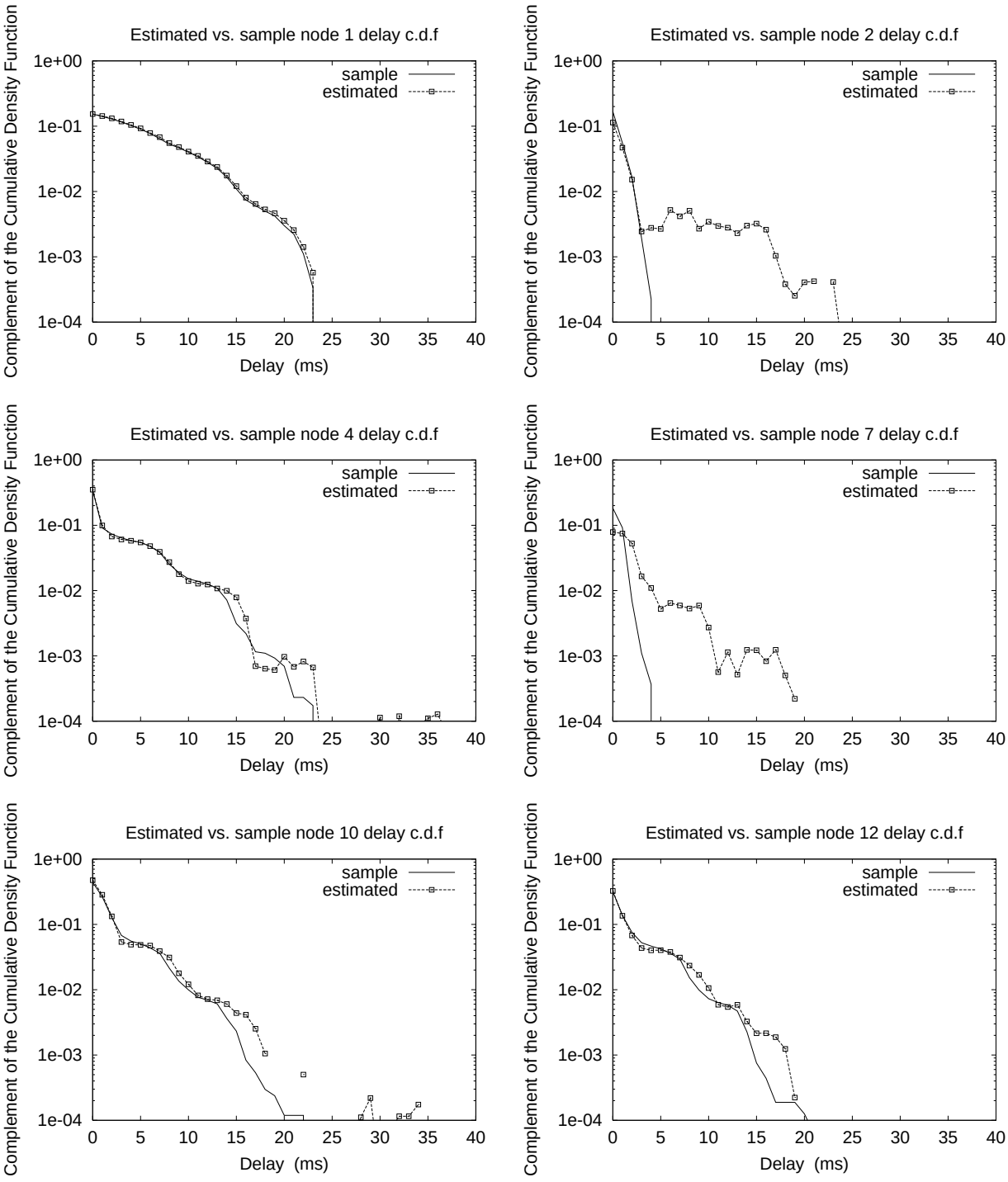


Figure 14: Sample vs. Estimated Delay complementary c.d.f. for selected links.

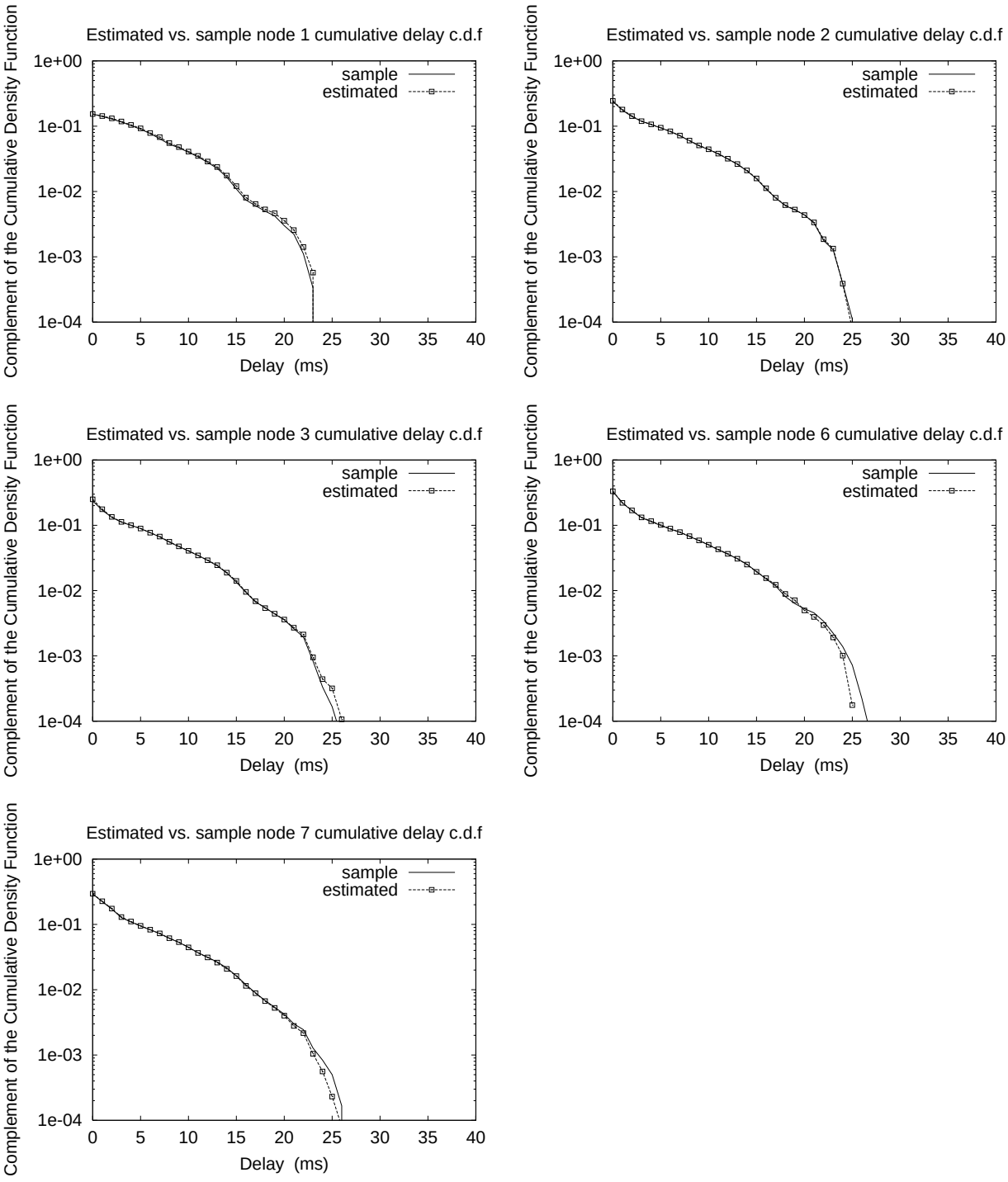


Figure 15: Sample vs. Estimated cumulative delay complementray c.d.f.

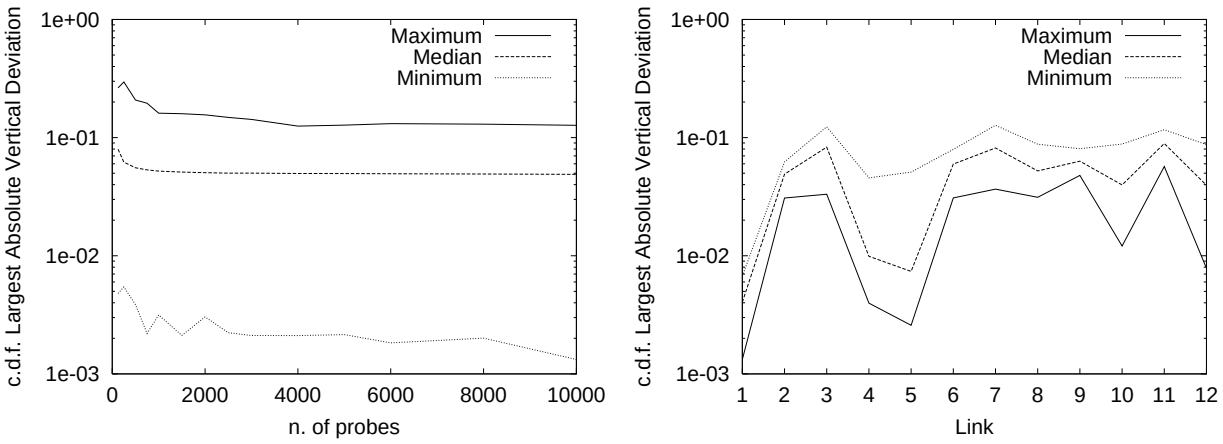


Figure 16: TCP/UDP SIMULATION: TOPOLOGY OF FIGURE 8. ACCURACY OF THE ESTIMATED DISTRIBUTION. LARGEST VERTICAL ABSOLUTE DEVIATION BETWEEN ESTIMATED AND THEORETICAL C.D.F. Minimum, median and maximum largest absolute deviation in 100 simulations computed over all links as function of n (a) and link by link for $n = 10000$ (b).

pendence, 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.

Finally in Figure 16 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 (left) and link by link for $n = 10000$ (right). 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.

6 Conclusions 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 directly 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. To deal with continuously distributed delays, we havederived a continuous version of the inference algorithm, which we are currently investigating.

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.

References

[1] R. Bellmann and R. Roth, "The Laplace Transform", World Scientific, Singapore 1984.

[2] 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.

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

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

[5] 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. of IEEE Infocom '99*, New York, NY, 23-25 March, 1999.

[6] R. Caceres, N.G. Duffield, S. Moon, and D. Towsley, "Inferring Link-Level Performance from End-to-End Measurements", in Proc. IEEE/ISOC Global Internet '99, December 1999.

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

[8] 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.

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

[10] A. Downey, "Using `pathchar` to estimate Internet link characteristics, *Proc. SIGCOMM 1999*, Cambridge, MA, pp. 241-250, Sept. 1999.

[11] N. Duffield, F. Lo Presti, "Multicast Inference of Packet Delay Variance at Interior Networks Links", in Proc. IEEE Infocom 2000, Tel Aviv, Israel, March 26-30, 2000.

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

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

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

[15] IP Performance Metrics Working Group. For more information see `http://www.ietf.org/html.charters/ippm-charter.html`

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

[17] J. Mahdavi, V. Paxson, A. Adams, M. Mathis, "Creating a Scalable Architecture for Internet Measurement," *Proc. of INET '98*, Geneva, Switzerland, July 1998.

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

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

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

[21] S. Moon, J. Kurose, P. Skelly and D. Towsley, "Correlation of Packet Delay and Loss in the Internet" *Tech. Report University of Massachussets at Amherst*, 1999.

[22] `mtrace` – Print multicast path from a source to a receiver. For more information see `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. For more information see `http://www-mash.cs.berkeley.edu/ns/ns.html`

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

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

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

[28] V. Paxson, "Automated Packet Trace Analysis of TCP Implementations," *Proc. SIGCOMM 1997*, Cannes, France, pp. 167-179, Sept. 1997.

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

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

[31] D. Sanghi, A. Agrawala and B. Jain, "Experimental assessment of end-to-end behavior on Internet", *Proc. IEEE Infocom '93*, San Francisco, CA, pp. 867-874, Mar. 1993.

[32] D. Sanghi, O. Gudnumdsson, A. K. Agrawala, "Study of network dynamics", *Proc. 4th Joint European Networking Conference*, Trondheim, Norway, pp. 241-249, May 1993.

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

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

A Uniqueness and Continuous Differentiability of the Inverse

The algorithm presented in Section 3.2.2 computes a solution of the system of equations (5) in the unknown $A = (A_k(i))_{k\in V, i\in\{0,\ldots,i_{\max}\}}$ given $\gamma = (\gamma_k(i))_{k\in V, i\in\{0,\ldots,i_{\max}\}}$. By deconvolution we then compute $\alpha = (\alpha_k(i))_{k\in V, i\in\{0,\ldots,i_{\max}\}}$.

We now show that the solution so computed is the unique solution of the equation $\gamma = \Gamma(\alpha)$, i.e. that it uniquely defines the inverse $\alpha = \Gamma^{-1}(\gamma)$. To this end we rewrite the mapping $\alpha = \Gamma(\gamma)$ as $\gamma = \chi \circ \psi(\alpha)$ where $A = \psi(\alpha)$ is clearly a bijection. It remains to show that χ is also a bijection. To prove this consider α, A and γ such that $A = \psi(\alpha)$ and $\gamma = \chi(A)$. We first show that $A_k(i)$, $i = 1, \ldots, i_{\max}$ is the second largest solution of (8).

In the binary case we can directly solve (8) to obtain the two solutions

$$A_k(i) + A_k(0)\left(\frac{\beta_{d_1}(i)}{\beta_{d_1}(0)} + \frac{\beta_{d_2}(i)}{\beta_{d_2}(0)} - 1\right) \geq A_k(i) + A_k(0)$$

and $A_k(i)$. For the general case we have the following Lemma.

Lemma 1 *Let $x_1 \geq x_2 \geq \ldots \geq x_m$, $m \leq \#d(k)$ denote the real solutions of the equation*

$$\begin{gathered}\gamma_k(i) + A_k(0)\left\{\prod_{d\in d(k)}\left[1 - \frac{\gamma_d(i) - \sum_{j=1}^{i-1}\beta_d(i-j)A_k(j) - \beta_d(0)x}{A_k(0)}\right] - 1\right\} + \\ \sum_{j=1}^{i-1} A_k(j)\left\{\prod_{d\in d(k)}\left[1-\beta_d(i-j)\right] - 1\right\} + x\left\{\prod_{d\in d(k)}\left[1-\beta_d(0)\right] - 1\right\} = 0.\end{gathered} \tag{16}$$

Then $x_2 = A_k(i)$.

Proof: Substitute $x = A_k(i) + yA_k(0)$ in equation (16) obtaining

$$\begin{gathered}\gamma_k(i) + A_k(0)\left\{\prod_{d\in d(k)}\left[1-\beta_d(i)+\beta_d(0)y\right] - 1\right\} + \\ \sum_{j=1}^{i-1} A_k(j)\left\{\prod_{d\in d(k)}\left[1-\beta_d(i-j)\right] - 1\right\} + (A_k(i) + yA_k(0))\left\{\prod_{d\in d(k)}\left[1-\beta_d(0)\right] - 1\right\} = 0.\end{gathered} \tag{17}$$

To prove the lemma we simply need to show that $y = 0$ is the second largest solution of (17). Expanding the product in the second term we get

$$\begin{gathered}\gamma_k(i) + A_k(0)\left\{\prod_{d\in d(k)}\left[1-\beta_d(i)\right] - 1\right\} + A_k(0)\left\{\sum_{b\in B}\prod_{m\in\{1,\ldots,\#d(k)\}}(1-\beta_{d_m}(i))^{b_m}\beta_{d_m}(0)^{1-b_m}y^{1-b_m}\right\} + \\ \sum_{j=1}^{i-1} A_k(j)\left\{\prod_{d\in d(k)}\left[1-\beta_d(i-j)\right] - 1\right\} + (A_k(i) + yA_k(0))\left\{\prod_{d\in d(k)}\left[1-\beta_d(0)\right] - 1\right\} = 0,\end{gathered}$$

where $b = \{b_1, \ldots, b_{\#d(k)}\}$, $B = \{0,1\}^{\#d(k)} \setminus \{0\}^{\#d(k)}$. Observing that the constant terms sum to 0 (by equation (5) and dividing by $A_k(0)$) we find that this reduces to

$$\sum_{b\in B}\prod_{m\in\{1,\ldots,\#d(k)\}}(1-\beta_{d_m}(i))^{b_m}\beta_{d_m}(0)^{1-b_m}y^{1-b_m} + y\left\{\prod_{d\in d(k)}\left[1-\beta_d(0)\right] - 1\right\} = 0.$$

Grouping with respect to y^l, we obtain

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

The coefficients of the polynomial 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 (17) greater than zero. ∎

From the uniqueness of the solution of (6) for canonical delay trees and by induction on i, it then follows that χ is a bijection; thus, the inverse is uniquely defined.

To prove that the inverse is continuously differentiable we proceed as follows. Let $\theta_{k,i}(\gamma, A)$, $k \in U$, $i = 0, \ldots, i_{\max}$, denote the left hand side of (8). Define the function $H(\gamma, A) = (\theta_{k,j}(\gamma, A))_{k\in U, j=0,\ldots,i_{\max}}$. $H(\gamma, A) = 0$ is the system of equations to be solved to compute A given γ. Denote by $A(\gamma)$ the unique solution to $H(\gamma, A(\gamma)) = 0$. The proof that the inverse is continuously differentiable amounts to showing that so is $A = \chi^{-1}(\gamma) = A(\gamma)$ (as ψ and its inverse clearly are). For canonical trees, $A_k(0) > 0$, $k \in U$, and therefore H is continuously differentiable. Then, by the Implicit Function Theorem, so is $A(\gamma)$, provided that the determinant of the Jacobian $\frac{\partial H}{\partial A}\big|_{A=A(\gamma)}$ is different from zero. To see this, observe that $\frac{\partial \theta_{k_1,i_1}(\gamma,A)}{\partial A_{k_2,i_2}} = 0$ if $k_1 \neq k_2$ or $i_1 < i_2$; hence the Jacobian matrix is always triangular. The diagonal elements are $\frac{\partial \theta_{k,i}(\gamma,A)}{\partial A_{k,i}}\Big|_{A=A(\gamma)} = \tilde{\theta}'_{k,i}(0)/A_k(0) < 0$.

B The Continuous Model: Delay Distribution Analysis

In this Appendix we formulate the delay analysis for continuous delay distributions, rather than for discrete distributions. We assume now that $D_k \in \mathbb{R}_+ \cup \{\infty\}$ and that the distribution is absolutely continuous w.r.t Lebesgue measure on $\mathbb{R}_+$ with density α_k, together with an atom at ∞ of mass $\alpha_k(\infty) = 1 - \int_0^\infty \alpha_k(x)dx$, the probability that a packet is lost traversing the link terminating at k. (For simplicity of notation, here we do not consider the atom at 0 representing the probability that a probe experiences minimum delay.) A_k is defined similarly for the source to root delays Y_k. Similar to the discrete case we define $\Omega_k(x)$ as the event $\{\min_{j\in R(k)} Y_j \leq x\}$ and $\gamma_k(x) = \mathsf{P}[\Omega_k(x)]$, $k \in U$. Finally, let $\beta_k(x) = \mathsf{P}[\min_{j\in R(k)} Y_j - Y_{f(k)} \leq x]$, $k \in U$. From the above definitions, the following relations hold:

$$A_k(x) = \int_0^x \alpha_k(y) A_{f(k)}(x-y)dy, \; k \in U,$$

where we set $A_0(x) = \delta(x)$,

$$\beta_k(x) \;=\; \int_0^x \alpha_k(y)(1 - \textstyle\prod_{d\in d(k)} [1 - \beta_d(x-y)])dy,\ k \in U \tag{18}$$

and

$$\gamma_k(x) \;=\; \int_0^x A_k(y)(1 - \textstyle\prod_{d\in d(k)} [1 - \beta_d(x-y)])dy,\ k \in U, \tag{19}$$

which is the continuous version of (5). Empty products are assumed to be equal to zero.

The above equation can be rewritten in more convenient form using the Laplace transform

$$\widetilde{\gamma}_k(s) = \widetilde{A}_k(s)\left[\frac{1}{s} - *_{d\in d(k)}\left(\frac{1}{s} - \frac{\widetilde{\gamma}_d(s)}{\widetilde{A}_k(s)}\right)\right]\ k \in U \tag{20}$$

or

$$\frac{1}{s} - \frac{\widetilde{\gamma}_k(s)}{\widetilde{A}_k(s)} = *_{d\in d(k)}\left[\frac{1}{s} - \frac{\widetilde{\gamma}_d(s)}{\widetilde{A}_k(s)}\right],\ k \in U \tag{21}$$

where, $\widetilde{f}(s) = \int_0^\infty f(x)e^{-sx}dx$ is the Laplace transform of $f(x)$, $*$ is the convolution operator in the domain s, $\widetilde{f}(s) * \widetilde{g}(s) = \int_{a-i\infty}^{a+i\infty} f(p)g(s-p)dp$, and $\widetilde{\beta}_d(s) = \widetilde{\gamma}_d(s)/\widetilde{A}_k(s)$.

Given $\widetilde{\gamma}_k(s)$, $k \in U$, (21) represents a system of $\#U$ independent equations in the unknowns $\widetilde{A}_k(s)$, $k \in U$. $\widetilde{\alpha}_k(s)$ can then be computed by the quotients

$$\widetilde{\alpha}_k(s) = \frac{\widetilde{A}_k(s)}{\widetilde{A}_{f(k)}(s)},\ k \in U.$$

Solving equation (21) is not trivial, especially because of the convolution on the right hand side which, in general, can be computed only numerically. Furthermore, we have not yet been able to establish whether the solution is unique. The inversion of the Laplace transform poses other challenges. It is well known, indeed (see for example [1]), that the inverse Laplace transform is an unbounded operator. In other words arbitrarily small changes in the transform will produce arbitrarily large changes in the original function. Therefore it may not be easy to control the accuracy of the results obtained with such an approach. All these issues will be subject of further investigation. The estimator $\widehat{\widetilde{\alpha}}(s)$ can be computed in the same manner from the estimates $\widehat{\widetilde{\gamma}}(s)$ obtained from the measurements. (21) can be written as a fixed point equation for the $\widetilde{A}_k(s)$; this suggests the possible use of the contraction mapping theorem in order to establish existence and uniqueness of solutions.

C Proof of Theorem 4

The proof proceeds by establishing a number of subsidiary results.

C.1 Limit Behavior of A, β and γ

As $\| \alpha \| \to 0$,

(i)

$$A_k(i) = \begin{cases} 1 - s_k(0) + O(\| \alpha \|^2) & i = 0 \\ \sum_{l=0}^{\ell(k)} \alpha_{f^l(k)}(i) + O(\| \alpha \|^2) & i > 0 \end{cases} \tag{22}$$

(ii)

$$1 - \beta_k(i) = \sum_{j>i} \alpha_k(i) + O(\| \alpha \|^2) \tag{23}$$

(iii)

$$\gamma_k(i) = 1 - s_k(i) + O(\| \alpha \|^2) \tag{24}$$

where

$$s_k(i) = \sum_{l=0}^{\ell(k)} \sum_{j>i} \alpha_{f^l(k)}(j) \tag{25}$$

The relation (i) is clear for $i = 0$ by expanding $A_k(0) = \prod_{l=0}^{\ell(k)} \alpha_k(0)$; for $i > 0$, it follows by an inductive argument on k and i: it is true for $k = 1$ and $i = 1$; if it is true for $j \succ k$ and for $0 \leq i' \leq i - 1$, then

$$A_k(i) = \sum_{j=0}^{i} A_{f(k)}(j)\alpha_k(i-j) = A_{f(k)}(0)\alpha_k(i) + A_{f(k)}(l)\alpha_k(0) + O(\| \alpha \|^2) = \alpha_k(i) + \sum_{l=0}^{\ell(f(k))} \alpha_{f^l(k)}(i) + O(\| \alpha \|^2).$$

Also (ii) follows by an inductive argument. Observe from (3) that if (ii) holds for all j in $d(k)$ and $i' \leq i$ then $1 - \beta_k(i) = 1 - \sum_{j=0}^{i} \alpha_k(j) + O(\| \alpha \|^2)$. Since $\beta_k(i) = \sum_{j=0}^{i} \alpha_k(j)$ for $k \in R$, $i = 1, \ldots, i_{\max}$, (ii) holds for all k and i. (iii) then follows by expanding $\gamma_k(i) = \sum_{j=0}^{i} A_k(j)[1 - \prod_{d \in d(k)} \beta_d(i-j)]$ (Observe that the terms within square brackets are always of the form $1 - O(\| \alpha \|)$).

C.2 Limit Behavior of the Covariance Matrix σ

As $\| \alpha \| \to 0$,

$$\mathsf{Cov}(\widehat{\gamma}_{k_1}(i_1), \widehat{\gamma}_{k_2}(i_2)) = s_{k_1 \wedge k_2}(\max(i_1, i_2)) + O(\| \alpha \|^2) \tag{26}$$

where $k_1 \wedge k_2$ denotes the minimal common ancestor of k_1 and k_2 with respect to $\prec$.

To see this, we write $\mathsf{Cov}(\widehat{\gamma}_{k_1}(i_1), \widehat{\gamma}_{k_2}(i_2)) = \mathsf{E}[\widehat{\gamma}_{k_1}(i_1)\widehat{\gamma}_{k_2}(i_2)] - \mathsf{E}[\widehat{\gamma}_{k_1}(i_1)]\mathsf{E}[\widehat{\gamma}_{k_2}(i_2)]$. By definition, $\mathsf{E}[\widehat{\gamma}_{k_1}(i_1)] = \gamma_{k_1}(i_1)$ and $\mathsf{E}[\widehat{\gamma}_{k_1}(i_1)\widehat{\gamma}_{k_2}(i_2)] = \mathsf{P}[\min_{m \in R(k_1)} Y_m \leq i_1 q \cap \min_{m \in R(k_2)} Y_m \leq i_2 q]$ [2]. We have three cases:

[2] Since probes are assumed independent, it suffices to evaluate all random quantities for $n = 1$ probes.

(i) $k_1 \succeq k_2$**,** $i_1 \geq i_2$**.**

In this case $\{\min_{m\in R(k_2)} Y_m \leq i_2 q\} \subseteq \{\min_{m\in R(k_1)} Y_m \leq i_1 q\}$ and $\mathsf{E}[\widehat{\gamma}_{k_1}(i_1)\widehat{\gamma}_{k_2}(i_2)] = \widehat{\gamma}_{k_2}(i_2)$. Thus,

$$\mathsf{Cov}(\widehat{\gamma}_{k_1}(i_1), \widehat{\gamma}_{k_2}(i_2)) = (1-\gamma_{k_1}(i_1))\gamma_{k_2}(i_2) = s_{k_1}(i_1) + O(\| \alpha \|^2).$$

(26) follows as $i_1 = \max(i_1, i_2)$ and $k_1 = k_1 \wedge k_2$.

(ii) $k_1 \succ k_2$**,** $i_1 < i_2$**.**

Write $\mathsf{P}[\min_{m\in R(k_1)} Y_m \leq i_1 q \cap \min_{m\in R(k_2)} Y_m \leq i_2 q] = \mathsf{P}[\min_{m\in R(k_1)} Y_m \leq i_1 q] + \mathsf{P}[\min_{m\in R(k_2)} Y_m \leq i_2 q] - \mathsf{P}[\min_{m\in R(k_1)} Y_m \leq i_1 q \cup \min_{m\in R(k_2)} Y_m \leq i_2 q]$. The first two terms are $\gamma_{k_1}(i_1)$ and $\gamma_{k_2}(i_2)$. Then, as $\| \alpha \| \to 0$

$$\mathsf{E}[\widehat{\gamma}_{k_1}(i_1)]\mathsf{E}[\widehat{\gamma}_{k_2}(i_2)] = 1 - s_{k_1}(i_1) - s_{k_2}(i_2) + O(\| \alpha \|^2), \tag{27}$$

it readily follows that

$$\mathsf{Cov}(\widehat{\gamma}_{k_1}(i_1), \widehat{\gamma}_{k_2}(i_2)) = 1 - \mathsf{P}[\min_{m\in R(k_1)} Y_m \leq i_1 q \cup \min_{m\in R(k_2)} Y_m \leq i_2 q] + O(\| \alpha \|^2). \tag{28}$$

To compute $\mathsf{P}[\min_{m\in R(k_1)} Y_m \leq i_1 q \cup \min_{m\in R(k_2)} Y_m \leq i_2 q]$ we need to define some additional quantities. Denote by $W = \{w_1, \ldots, w_l\} \subset V$ a set of nodes that induces a partition on R, *i.e.*, W is such that $\cup_{h=1}^{l} R(w_h) = R$ and $R(w_h) \cap R(w_{h'}) = \emptyset$, $1 \leq h, h' \leq l$, $h \neq h'$. Associate to W a set of delay values $\Delta = \{j_1, \ldots, j_l\}$. The quantities we introduce below are a generalization of β and γ, where we use different delays, $j_1, \ldots, j_l$ for different sets of receivers, namely $R(w_1), \ldots, R(w_l)$. For $k \in V$, define

$$\chi_{k,W}(\Delta) = \mathsf{P}[\cup_{h=1}^{l} \min_{w\in R(w_h)\cap R(k)} Y_m - Y_{f(k)} \leq i_h q]. \tag{29}$$

Then, $\chi_{k,W}$ obeys the recursion

$$\chi_{k,W}(\Delta) = \sum_{j=0}^{\max_{j_{k,W}}} \alpha_k(j) \left[1 - \prod_{d\in d(k)} 1 - \chi_{d,W}(\Delta - j)\right], \; k \in U \setminus R \tag{30}$$

$$\chi_{k,W}(\Delta) = \beta_k(j_{k,W}), \; k \in R. \tag{31}$$

where $\Delta - j = \{j_1 - j, \ldots, j_l - j\}$ and $j_{k,W} = \max_{l': R(k)\cap R(w_{l'})\neq\emptyset} j_{l'}$. Probabilities with negative index are assumed to be equal to zero.

For a given node k, define now

$$\eta_{k,W}(\Delta) = \mathsf{P}[\cup_{h=1}^{l} \min_{w\in R(w_h)\cap R(k)} Y_m \leq j_h q]. \tag{32}$$

The following, which can be regarded as a generalization of (5), holds

$$\eta_{k,W}(\Delta) = \sum_{j=0}^{j_{k,W}} A_k(j) \left[1 - \prod_{d \in d(k)} 1 - \chi_{d,W}(\Delta - j) \right], \; k \in U \setminus R \tag{33}$$

$$\eta_{k,W}(\Delta) = \sum_{j=0}^{j_{k,W}} A_k(j), \; k \in R. \tag{34}$$

For $\| \alpha \| \to 0$, it is easy to verify that $\eta_{k,W}(\Delta)$ behaves as $\gamma_k(j_{k,W})$ (the terms within square bracket are always of the form $1 - O(\| \alpha \|)$). In other words, $\eta_{k,W}(\Delta) = 1 - s_k(j_{k,W}) + O(\| \alpha \|^2)$.

With the definitions above, we can now write

$$\mathsf{P}[\min_{m \in R(k_1)} Y_m \leq i_1 q \cup \min_{m \in R(k_2)} Y_m \leq i_2 q] = \eta_{k_1,W}(\Delta) \tag{35}$$

where $W = \{w_1, \ldots, w_l\}$, $w_1 = k_2$, and $\Delta = \{j_1, \ldots, j_l\}$, with $j_1 = i_2$ and $j_{l'} = i_1$, $2 \leq l' \leq l$. Then, as $(\| \alpha \|) \to \infty$,

$$\mathsf{P}[\min_{m \in R(k_1)} Y_m \leq i_1 q \cup \min_{m \in R(k_2)} Y_m \leq i_2 q] = 1 - s_{k_1}(i_2) + O(\| \alpha \|^2). \tag{36}$$

Thus,

$$\mathsf{Cov}(\widehat{\gamma}_{k_1}(i_1), \widehat{\gamma}_{k_2}(i_2)) = s_{k_1}(i_2) + O(\| \alpha \|^2). \tag{37}$$

(26) follows as $k_1 = k_1 \wedge k_2$ and $i_2 = \max(i_1, i_2)$.

(iii) $k_1 \wedge k_2 \succ k_1, k_2$**.**

We proceed as for **(ii)**. In this case, we can write

$$\mathsf{P}[\min_{m \in R(k_1)} Y_m \leq i_1 q \cup \min_{m \in R(k_2)} Y_m \leq i_2 q] = \eta_{k_1 \wedge k_2,W}(\Delta) \tag{38}$$

where $W = \{w_1, \ldots, w_l\}$, $w_1 = k_1$ and $w_2 = k_2$, and $\Delta = \{j_1, \ldots, j_l\}$, with $j_1 = i_1$, $j_2 = i_2$ and $j_{l'} = -1$, $3 \leq l' \leq l$ (for (38) to hold, we need to set j_l equal to -1, $l \neq 1, 2$, or any other negative number, to insure that all events regarding receivers different from $R(k_1)$ and $R(k_2)$ have probability zero). Thus, as $\| \alpha \| \to \infty$,

$$\mathsf{P}[\min_{m \in R(k_1)} Y_m \leq i_1 q \cup \min_{m \in R(k_2)} Y_m \leq i_2 q] = 1 - s_{k_1 \wedge k_2}(\max(i_1, i_2)) + O(\| \alpha \|^2). \tag{39}$$

Therefore,

$$\mathsf{Cov}(\widehat{\gamma}_{k_1}(i_1), \widehat{\gamma}_{k_2}(i_2)) = s_{k_1 \wedge k_2}(\max(i_1, i_2)) + O(\| \alpha \|^2). \tag{40}$$

C.3 Limit Behavior of the Jacobian $D(\alpha)$

As $\|\alpha\| \to 0$,

$$D(\alpha) = B \otimes D + O(\|\alpha\|) \text{ where } D_{k_1 k_2} = \begin{cases} 1 & k_2 = k_1 \\ -1 & k_2 = f(k_1) \\ 0 & \text{otherwise} \end{cases}, \tag{41}$$

where B is a $(i_{\max}+2)\times(i_{\max}+2)$ matrix with entries $B_{ii'} = \delta_{ii'} - \delta_{ii'+1}$, $\otimes$ denotes the Knonecker product, and $\delta_{ii'} = 1$ if $i = i'$ and 0 otherwise.

To establish this, we first show that the inverse $D^{-1}(\alpha)$, whose elements are $(D^{-1}(\alpha))_{(k_1,i_1)(k_2,i_2)} = \partial\gamma_{k_1}(i_1)/\partial\alpha_{k_2}(i_2)$, has the following form for $\|\alpha\| \to 0$,

$$D^{-1}(\alpha) = L \otimes \tilde{D} + O(\|\alpha\|) \text{ where } \tilde{D}_{k_1 k_2} = \begin{cases} 1 & k_1 \preceq k_2 \\ 0 & \text{otherwise} \end{cases}, \tag{42}$$

where L is a unit lower triangular matrix, *i.e.*, $L_{ii'} = \mathbf{1}_{\{i \le i'\}}$. To this end we rewrite $\gamma_k(i)$ as $\sum_{j=0}^{i} A_k(j)[1 - \prod_{d\in d(k)}(1-\beta_d(i-j))]$. We have the following three cases:

(i) $k_1 \preceq k_2$**,** $i_1 \ge i_2$**.**

Let l' be such that $f^{l'}(k_1) = k_2$. Then, for $\|\alpha\| \to 0$

$$\begin{aligned}
\frac{\partial\gamma_{k_1}(i_1)}{\partial\alpha_{k_2}(i_2)} &= \sum_{j=i_2}^{i_1} \frac{\partial A_{k_1}(j)}{\alpha_{k_2}(i_2)}[1 - \prod_{d\in d(k_1)}(1-\beta_d(i_1-j))] && (43)\\
&= \sum_{j=i_2}^{i_1} \frac{\partial}{\alpha_{k_2}} \left(\sum_{\sum_{l=0}^{\ell(k_1)} j_l = j} \prod_{l=0}^{\ell(k_1)} \alpha_{f^l(k_1)}(j_l) \right) [1 - \prod_{d\in d(k_1)}(1-\beta_d(i_1-j))] && (44)\\
&= \prod_{l=0, l\neq l'}^{\ell(k_1)} \alpha_{f^l(k_1)}(0)[1 - \prod_{d\in d(k_1)}(1-\beta_d(i_1-i_2))] + \\
&\quad \sum_{j=i_2+1}^{i_1} \sum_{\sum_{l=0}^{\ell(k_1)} j_l = j, j_{l'}=i_2} \prod_{l=0, l\neq l'}^{\ell(k_1)} \alpha_{f^l(k_1)}(j_l)[1 - \prod_{d\in d(k_1)}(1-\beta_d(i_1-j))] && (45)\\
&= 1 + O(\|\alpha\|) && (46)
\end{aligned}$$

as the first term of (45) goes to 1 while the second goes to 0 because in any product there is at least one l such that $j_l > 0$.

(ii) $k_1 \succ k_2$**,** $i_1 \geq i_2$**.**

Let d' be such that there exists an l such that $f^l(k_2) = d'$. Then,

$$\begin{aligned}
\frac{\partial \gamma_{k_1}(i_1)}{\partial \alpha_{k_2}(i_2)} &= \sum_{j=0}^{i_1-i_2} A_{k_1}(j) \frac{\partial[1 - \prod_{d\in d(k_1)}(1-\beta_d(i_1-j))]}{\alpha_{k_2}(i_2)} && (47)\\
&= \sum_{j=0}^{i_1-i_2} A_{k_1}(j) \frac{\partial \beta_{d'}(i_1-j)}{\alpha_{k_2}(i_2)} [1 - \prod_{d\in d(k_1), d\neq d'} (1-\beta_d(i_1-j))] && (48)\\
&= O(\| \alpha \|) && (49)
\end{aligned}$$

as each product goes to 0, as $\| \alpha \| \to 0$.

(iii) $i_1 < i_2$**.**

In this last case, $\gamma_{k_1}(i_1)$ does not depend on $\alpha_{k_2}(i_2)$ and the derivative is 0.

Since matrix inversion is continuous in an open neighborhood on non-singular matrices, then (41) follows since D and $\tilde{D}$ are inverses (see Section 10 of [4]) as also are L and B (trivial) and since for invertible square matrices F and G, $(F \otimes G)^{-1} = F^{-1} \otimes G^{-1}$.

C.4 Proof

From Theorem 3, (26), (41) and continuity of finite dimensional matrix products, we have for $\| \alpha \| \to 0$ that

$$\nu_{(k_1,i_1)(k_2,i_2)} = \sum_{k_1',k_2',i_1',i_2'} D_{(k_1,i_1)(k_1',i_1')} s_{k_1',k_2'}(\max(i_1',i_2')) D_{(k_2,i_2)(k_2',i_2')} + O(\| \alpha \|^2). \qquad (50)$$

It remains to evaluate

$$\begin{aligned}
\sum_{k_1',k_2',i_1',i_2'} D_{(k_1,i_1)(k_1',i_1')} s_{k_1',k_2'}(\max(i_1',i_2')) D_{(k_2,i_2)(k_2',i_2')} &= \sum_{i_1',i_2'} B_{i_1 i_1'} [s_{k_1\wedge k_2}(\max(i_1',i_2')) - s_{f(k_1)\wedge k_2}(\max(i_1',i_2')) - \\
& s_{k_1\wedge f(k_2)}(\max(i_1',i_2')) + s_{f(k_1)\wedge f(k_2)}(\max(i_1',i_2'))] B_{i_2 i_2'} \qquad (51)
\end{aligned}$$

When $k_1 \neq k_2$, (51) yields 0. Indeed, if $k_2 \succ k_1$, $k_1 \wedge k_2 = f(k_1) \wedge k_2 = k_2$, while $k_1 \wedge f(k_2) = f(k_1) \wedge f(k_2) = f(k_2)$, and hence (51) is zero. Similarly for $k_1 \succ k_2$. In all other cases, $k_1 \wedge k_2 \succ k_1, k_2$ and so $k_1 \wedge k_2 = f(k_1) \wedge k_2 = k_1 \wedge f(k_2) = f(k_1) \wedge f(k_2) = f(k_2)$ and (51) is again zero.

When $k_1 = k_2$, $k_1 \wedge k_2 = k_1$, $k_1 \wedge f(k_2) = f(k_1) \wedge k_2 = f(k_1) \wedge f(k_2) = f(k_1)$ and (51) reduces to

$$\sum_{i'_1, i'_2} B_{i_1 i'_1} [s_{k_1}(\max(i'_1, i'_2)) - s_{f(k_1)}(\max(i'_1, i'_2))] B_{i_2 i'_2} \tag{52}$$

Substituting $B_{ij} = \delta_{ij} - \delta_{ij+1}$ in (52), it is easy to verify the following:

$$\sum_{i'_1, i'_2} B_{i_1 i'_1} [s_{k_1}(\max(i'_1, i'_2)) - s_{f(k_1)}(\max(i'_1, i'_2))] B_{i_2 i'_2} = \begin{cases} \sum_{j>0} \alpha_{k_1}(j) & i_1 = i_2 = 0 \\ \alpha_{k_1}(i_1) & i_1 \neq 0,\ i_2 = 0 \\ \alpha_{k_1}(i_2) & i_2 = 0,\ i_2 \neq 0 \\ 0 & i_1 \neq i_2,\ i_1, i_2 \neq 0 \end{cases} \tag{53}$$