Temporal Delay Tomography

Vijay Arya
National ICT Australia (NICTA)
The University of Melbourne, Australia
Email: vijay.arya@nicta.com.au

N. G. Duffield
AT&T Labs–Research
Florham Park, NJ, USA
Email: duffield@research.att.com

Darryl Veitch
CUBIN[1], Dept. of EEE
The University of Melbourne, Australia
Email: d.veitch@ee.unimelb.edu.au

***Abstract*—Multicast-based network tomography enables inference of average loss rates and delay distributions of internal network links from end-to-end measurements of multicast probes. Recent work showed that this method, based on correlating observations of multicast receivers, also supports the inference of *temporal* loss characteristics of network links. In this paper, we show that temporal characteristics can, in fact, be estimated even for link delay processes. Knowledge of temporal delay characteristics has applications for delay sensitive services such as VOIP as well as for characterizing the queueing behavior of bottleneck links. By assuming mutually independent, but arbitrary link delay processes, we develop estimators which can infer, in addition to delay distributions, the probabilities of arbitrary patterns of delay, means and full distributions of delay-run periods at chosen delay levels, for each link in the multicast tree. By applying the recently proposed principle of subtree-partitioning, the estimator is made scalable to multicast trees of large degree. Estimation error and convergence rates are evaluated using simulations.**

I. INTRODUCTION

Multicast-based services form an increasingly important part of Internet Service Providers' offerings; see [1], [2] for example. Generally, multicast Virtual Private Networks may be used for distribution of time-sensitive financial data, or corporate video broadcasts; IP-based video distribution to the home may involve multicast groups with large and geographically distributed membership. Recent growth in the use of such applications has renewed interest in scalable methods for multicast performance measurement. However, direct measurement of all network links of interest remains a challenge of scale. Whereas ISPs do conduct active measurements between hosts located in major (regional or city-level) router centers, pushing these measurements out to the customer edge of the network would involve instrumenting a far larger number of access points.

The challenge of scale motivates the use of tomographic methods to infer performance of internal network links. Network performance tomography rests on the principle that performance measurements on intersecting paths can be correlated to infer performance of the common path portion. End-to-end measurements over multicast trees are well suited for this, due the inherent correlation between different receivers' experience of the same packet. A body of work on Multicast Inference of Network Characteristics (MINC) has shown how to infer average packet loss rates [3] and delay distributions [4] of network links, and even the network topology itself [5]. More recently, [6] showed that tomographic methods can also be used to monitor VPNs.

This paper aims to expand the capabilities of delay tomography, which currently only provides a means of estimating delay distributions of network links. Up until now, the literature is centered around a key assumption on the link delay process: *temporal independence* i.e., independence of packet queueing delays over time. This assumption only supports the measurement of statistics concerning the link delay distribution. It is well known however that packet traffic is bursty and exhibits temporal dependence, and performance of network applications depends on the durations of congestion events. Hence such statistics cannot provide a finer grained view of link delays, in particular, temporal characteristics such as the probability of two consecutive packets encountering delays above a given level, or mean durations of packet runs at or above a given delay level. In this paper, we remove the assumption of temporal independence and show how it is possible to recover, for each logical link in a multicast tree, temporal properties of link delay processes based on end-to-end delay observations made at receivers.

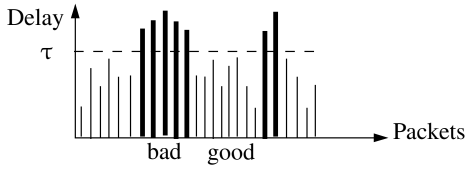

Fig. 1. Link delay process transitioning between good and bad states.

One of the outcomes of our work is being able to infer the average duration of high delay bursts on internal network links. Consider a sample path of a link delay process (Fig 1) which we regard as transitioning between two states, good and bad. During the good state, the packet queueing delays remain below a certain acceptable level τ. When the link becomes congested and the delays exceed this level, the process transitions into the bad state. In this setting, our work shows how one can infer, in addition to the delay distribution (e.g. the proportion of packets in the bad state), temporal properties such as the delay-run distribution (for any k, the proportion of packet runs in the bad state exceeding length k) and its mean (average length of such bad delay runs).

One of the main application areas for temporal information of this nature is real-time services such as VOIP. End-to-end

[1] ARC Special Research Centre for Ultra-Broadband Information Networks. CUBIN is an affiliated program of NICTA.

VoIP performance is sensitive to changes in delay and therefore to the duration of periods where delay is at a given level. Knowing typical durations of high delay bursts on internal links enables intelligent path selection. Another application is in the detection and characterization of links causing service degradation. This is a more general concept and problem than traditional 'bottleneck' detection. For example, average loss rates and delays may be relatively high, yet this could be acceptable to applications provided loss runs and runs of high delay are short. The estimation of temporal parameters is essential to make these deeper distinctions and thereby to localize and quantify the root causes of service degradation. This is of particular interest to network operators but end user application can also benefit, for example in distributed gaming.

II. Related Work

A body of work on network delay tomography [4], [7]–[11] allows the inference of delay distribution and delay variance of internal network links based on end-to-end multicast and unicast packet-pair measurements. See [12] for a survey. All of the above work is based on the assumption of temporal independence.

We know of no prior work concerned with estimation of temporal delay characteristics of network links. Recently, our work [13] showed how temporal loss characteristics (such as mean duration of loss-runs on internal links) can be estimated using end-to-end multicast measurements. In this work, we achieve the same for delay. We also employ another technique described in that paper, *subtree partitioning*. This is a technique to reduce the computational complexity of MINC based estimation, by transforming any multicast tree into a virtual binary tree for estimation purposes. Technically it results in estimators which only require the solution of either linear or quadratic equations to recover the probabilities of shared paths, avoiding the need for root finding of higher order polynomials.

The problem of delay estimation is much more complex than that of loss. Whereas the link loss model is binary (packet is either transmitted or lost), the (discretized) link delay model is multi-state in which loss appears as a special case of infinite delay. By assuming *independence* between links and packets, Lopresti et. al. [4] first showed how the discretized link delay distributions can be estimated. Their estimator uses a recursive computation or 'deconvolution', and builds the delay distribution from the lowest delay bin upwards. Our work generalizes this work where we assume temporal *dependence* and move from estimating probabilities about single packets (probability that a packet encounters delay d) to a temporal case of estimating probabilities about arbitrary group of packets (e.g. the probability that two consecutive packets encounter delays d and v respectively).

The delay distribution estimator found in [4] is not in general the Maximum Likelihood Estimator (MLE), and indeed direct calculation of the MLE appears infeasible in general. A different approach was taken by Liang and Yu [8], in which a pseudo-likelihood function is maximized. Our work takes a different direction and is not obviously related to the MLE.

Note that all temporal estimation methods of this paper may, in principle, be extended to groups of unicast packets emulating multicast packets, in the same manner as [14], [15] extend [3] for inferring average loss rates. Furthermore, the estimation methods can also be used to infer temporal properties of jitter based on measurements of inter-packet times.

III. Model

In this section we justify and describe how we model the delay processes over a tree, and derive key consequences.

Tree Model Let $\mathcal{T} = (V, L)$ denote the logical multicast tree consisting of a set of nodes V and links L. Let $0 \in V$ denote the root node and let $R \subset V$ be the set of leaf nodes. A link is an ordered pair $(k, j) \in \{V \times V\}$ representing a logical link from node k to node j. The set of children of a node k is denoted by $c(k) = \{j \in V : (k, j) \in L\}$. All nodes have at least two leaves, except the root (just one) and the leaves (none), see Fig 2(a). Let $U = V \setminus \{0\}$ denote the set of all non-root nodes of $\mathcal{T}$. For each node $k \in U$ there is a unique node $j = f(k)$, the father of k, such that $(k, j) \in L$. We define $f^{\ell}(k)$ recursively by $f^{\ell}(k) = f(f^{\ell-1}(k))$, with $f^0(k) = k$. Now $\forall k \in U$, the set of ancestors of k is $a(k) = \{j \in U : f^{\ell}(k) = j, \ell \geq 0\}$. Note $k \in a(k)$. We define $a(0) = 0$. For convenience, we refer to link $(f(k), k)$ simply as link k.

Modelling Link Delay In our model, an infinite stream of probe packets indexed by $i \in \mathbb{Z}$ is dispatched from the root node 0. Each probe that arrives at a node k results in a copy being sent to each of its children. When a probe attempts to traverse a link k, it encounters a random delay on the link and may even be lost. The passage of probes down the tree is modeled by two stochastic processes: $X_k = \{X_k(i)\}$ and $Z_k = \{Z_k(i)\}$ for each node k. The process Z_k is the *discrete time delay process* which determines the delay encountered by probes and also if probes are lost as they attempt to traverse link k. The process X_k acts as a *bookkeeping process* and records the cumulative delay encountered by each probe on the path from the root to node k.

Ideally, the state of links over the tree could be characterized by a set of underlying continuous time random processes $\{\mathcal{Z}_k(t) : t \in \mathbb{R}\}$, $k \in U$. If a packet is transmitted on link k at time t, it encounters a delay determined by the value $\mathcal{Z}_k(t) \in \mathbb{R}_+ \cup \{\infty\}$, taken by the link process, where ∞ accommodates packet loss. We consider an abstracted form of the problem, where the probe index i not only indexes probes, but also plays the role of discrete time. In this setting, the delay sampling processes are well defined for all i, with the interpretation that $Z_k(i)$ determines the delay that *would* be experienced by probe i, had it been present.

We discretize each link delay to the set $\{0, b, 2b, \ldots, mb, \infty\}$, where b is the bin width and m is any predefined threshold. The symbol ∞ is interpreted as "packet lost or encountered delay exceeding mb". Thus for each link k, the discrete-time discrete-state link delay process $Z_k(i)$ takes values from the state space $\mathcal{D} = \{0, 1, 2, \ldots, m, \infty\}$.

The bookkeeping process at a node k is denoted by $\{X_k(i) : i \in \mathbb{Z}\}$, and takes values $X_k(i) \in \{0, 1, 2, \dots, m\ell(k), \infty\}$ where $\ell(k)$ is the level or height of node k (hence $f^{\ell(k)}(k)$ is the root node). The link delay process acts deterministically on the node bookkeeping process at each probe index i as follows:

$$X_k(i) \;=\; Z_k(i) + X_{f(k)}(i)\ . \tag{1}$$

Fig 2(b) shows some examples. For the root node we have simply $X_0(i) = 0, \forall i$. We have the following probability model for $d, q \in \mathcal{D}$:

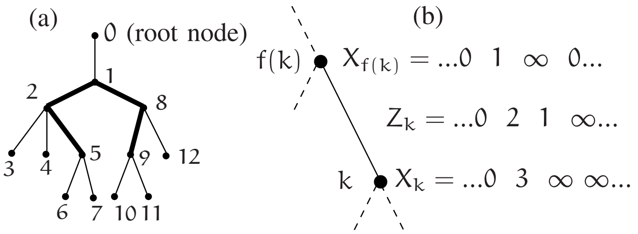
(a)
0 (root node)
1
2
8
3
4
5
9
12
6
7
10
11
(b)
f(k)
$X_{f(k)} = \dots 0 \;\; 1 \;\; \infty \;\; 0\dots$
$Z_k = \dots 0 \;\; 2 \;\; 1 \;\; \infty\dots$
k
$X_k = \dots 0 \;\; 3 \;\; \infty \;\; \infty\dots$

Fig. 2. (a) Example of a logical tree, (b) Delay model: the Link delay process $Z_k(i)$ on link k acts deterministically on input process $X_{f(k)}$ to produce process X_k.

$$\Pr[X_k(i) = d \,|\, X_{f(k)}(i) = q] = \begin{cases} 0 & \text{if } d < q \\ 1 & \text{if } d = q = \infty \\ \Pr[Z_k(i) = d - q] & \text{otherwise.} \end{cases} \tag{2}$$

We assume the following dependence structure:
SPATIAL: The processes $\{Z_k, k \in U\}$ are mutually independent.
TEMPORAL: For each $k \in U$, Z_k is stationary and ergodic.

Thus we assume that packets encounter delays independently across links, but that within each link, delays encountered by packets can be correlated, with parameters that in general depend on the link. Although potential mechanisms for spatial correlations have been identified (e.g. TCP synchronization [16]), we believe that diversity of link losses and round-trip times will prevent such mechanisms from operating in links which carry large numbers of flows; see [3]. The assumption of ergodicity means that empirical time averages of functions of the process will converge to the true expected value almost surely, ensuring our estimators have desirable properties.

A. Examples and Consequences of Delay Processes

Equation (1) implies that $X_k(i)$ is the sum of delay processes over its ancestor nodes, each also at index i:

$$X_k(i) = \sum_{j \in a(k)} Z_j(i)\ . \tag{3}$$

It follows that X_k inherits the stationarity and ergodicity properties of link delay processes. We now consider how more detailed properties of X_k follow from those of the Z_k in three classes of examples.

Bernoulli Scheme: For each link k the $Z_k(i)$ are i.i.d., so the delays encountered by packets are independent of each other. Such a process is simply characterized by the delay probabilities $\Pr[Z_k(i) = p], p \in \mathcal{D}$. This is the model used in prior work on delay tomography [4].

Stationary Ergodic Semi-Markov Process: This is a generalization of the well known On-Off process, whereby Z_k sojourns in delay state i for a period whose duration is distributed according to independent copies L_i of a random variable. At the end of the sojourn it jumps into state j with probability p_{ij} where $p_{ii} = 0$ and $\sum_j p_{ij} = 1$. Jumps occur independently of each other and of the sojourn durations. In general, the convolution of semi-Markov processes are not semi-Markov. However, the class with geometrically distributed sojourn in the minimal delay state is closed under convolution. This is because minimal path delay requires minimal delay on all constituent links, and once all links enter the minimal delay state, all memory of previous states is lost. Such models are interesting in practice because measurements over wide area networks have shown that packets carrying delays which spike above ambient levels arrive approximately as a Poisson process [17].

Stationary Ergodic Markov Process of Order r: For each link k, $Z_k(i)$ is a Markov process of order r as determined by the transition probabilities $\Pr[Z_k(i) = d | Z_k(i-1), \dots, Z_k(i-r) = d']$. Similarly to above, as a projection of the product process $\otimes_{j \in a(k)} Z_k$, X_k is not Markov in general. The case $r = 1$ is a Markov process with geometrically distributed sojourn times L_i, and so X_k also has geometrically distributed times in the minimal delay state.

B. Delay Preprocessing

The delay on a network link consists of a fixed propagation component, and a variable component due to queueing in buffers and router processing. We are concerned with the variable component only. Hence, following the approach of Lo Presti et. al [4], we remove the fixed component by subtracting from each source-to-leaf delay measurement, the minimum delay observed at that leaf. It is the resulting excess end-to-end delays that are discretized with bin size b, modeled using the Z_k as described above, and thereby used as input for estimation.

IV. Temporal Characteristics of Delay Processes

Before embarking on estimation, we need to establish which temporal characteristics of the processes Z_k one would wish to estimate. We choose an ambitious objective, the joint probability of a group of probes observing an arbitrary pattern of delay states on a link in a general, non-parametric setting. We show how to extract these in the next section. Here we show that such joint probabilities can be used to recover temporal statistics of practical interest.

Delay-run distributions and Mean delay-run lengths: Let H be any subset of the full state space $\mathcal{D}$. L_k^H be a random variable which indicates the lengths of runs of Z_k in states from subset H. The distribution of L_k^H and its mean μ_k^H are related to the joint distribution of the process Z_k as follows:

$$\begin{aligned}
&\Pr[L_k^H \geq j] \\
&= \Pr[Z_k(j)\in H, \dots, Z_k(1)\in H \mid Z_k(0) \notin H, Z_k(1)\in H] \\
&= \frac{\Pr[Z_k(j)\in H, \dots, Z_k(1)\in H, Z_k(0) \notin H]}{\Pr[Z_k(0) \notin H, Z_k(1) \in H]} \\
&= \frac{\Pr[Z_k(j)\in H, \dots, Z_k(1)\in H] - \Pr[Z_k(j)\in H, \dots, Z_k(0)\in H]}{\Pr[Z_k(1)\in H] - \Pr[Z_k(0)\in H, Z_k(1)\in H]}
\end{aligned}$$

and hence:

$$\begin{aligned}
\mu_k^H &= E[L_k^H] = \sum_{j\geq 1} j\Pr[L_k^H = j] = \sum_{j\geq 1} \Pr[L_k^H \geq j] \\
&= \frac{\Pr[Z_k(1)\in H]}{\Pr[Z_k(1)\in H] - \Pr[Z_k(0)\in H, Z_k(1)\in H]} \qquad (4)
\end{aligned}$$

The last sum is absolutely convergent because μ_k^H is finite. This formula makes intuitive sense as the ratio of the expected proportion of time spent in runs in the subset H (per time index) divided by the expected number of transitions into H (per time index).

Thus, the mean delay-run length of any subset H is accessible even without parametric help, provided we can estimate the simplest joint probabilities with respect to that subset, those for a single: $\Pr[Z_k(i) \in H]$ and a successive pair: $\Pr[Z_k(i) \in H, Z_k(i+1) \in H]$, of probes. The tail probability of runs in subset H, $\Pr[L_k^H \geq j]$ can be obtained from the joint probabilities of H for one, two, j, and $j+1$ probes.

The prime application of Eq. (4) is to partition link states into two classes, which we call good and bad. The bad class could be the subset of states with delay greater than some level τ (Fig. 3). The above formula yields the mean duration of runs in good and bad classes in terms of respective single and successive joint probabilities, i.e., we can recover the mean duration of runs in which the delay exceeds τ (or remains at least τ).

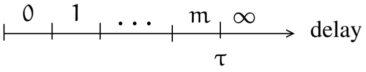

Fig. 3. Partition of delay states into good and bad classes, based on the threshold τ. In this example, good $= \{0, \dots, m\}$, bad $= \{\infty\}$.

V. Estimation of Temporal Delay Characteristics

Recall that the source at root node 0 sends a stream of n multicast probe packets. The outcome of the i-th probe is the set of discretized source-to-receiver delays $\{X_r(i), r \in R\}$. The goal is to infer temporal parameters of processes $Z_k, k \in U$ solely from the n receiver outcomes. In the previous section we discussed what these temporal parameters could be, and highlighted the importance of joint probabilities of delay patterns. In this section we show how to estimate these joint probabilities from receiver data.

A. Link Probabilities to Path Probabilities

To manipulate joint probabilities, we define the probe index set $I = \{i_1, i_2, \dots, i_s\}$ (not necessarily contiguous) and random vectors

$$\mathbf{X}_k(I) = [X_k(i_1), \dots, X_k(i_s)], \quad \mathbf{Z}_k(I) = [Z_k(i_1), \dots, Z_k(i_s)]$$

We show the estimation of joint probabilities involving delay values from partial state space $\mathcal{D}' = \mathcal{D} \setminus \{\infty\}$ and extend it to full space $\mathcal{D}$ in section V-E. Let $\mathbf{d}, \mathbf{q} \in \mathcal{D}'^{|I|}$ be vectors of delay values, for e.g. we write $\mathbf{d} = [d_1, \dots, d_s]$. The notation $\mathbf{d} \leq \mathbf{q}$ is used to mean $d_j \leq q_j, \forall j$. Let $\mathbf{m} = [m, \dots, m]$, $\mathbf{0} = [0, \dots, 0]$. Observe that $\forall\ \mathbf{d} \in \mathcal{D}'^{|I|}, \mathbf{0} \leq \mathbf{d} \leq \mathbf{m}$.
We define the *link pattern probability* $\alpha_k(I, \mathbf{d})$ as

$$\alpha_k(I, \mathbf{d}) = \Pr[\mathbf{Z}_k(I) = \mathbf{d}] = \Pr[\mathbf{X}_k(I) - \mathbf{X}_{f(k)}(I) = \mathbf{d}],$$

$\{\mathbf{Z}_k(I) = \mathbf{d}\}$ denotes the event $\{Z_k(i_1) = d_1, \dots, Z_k(i_s) = d_s\}$. $\alpha_k(I, \mathbf{d})$ denotes the probability that a pattern of probes given by index set I encounter a pattern of delays given by $\mathbf{d}$, on link k. By stationarity, $\alpha_k(I, \mathbf{d}) = \alpha_k(i + I, \mathbf{d})$, where $i + I \equiv \{i + i_1, \dots, i + i_s\}$. For example, $\alpha_k(\{1\}, [d]) = \alpha_k(\{i\}, [d])$ (the probability that a probe encounters delay d on link k), and $\alpha_k(\{1, 2\}, [d_1, d_2]) = \alpha_k(\{i, i+1\}, [d_1, d_2])$ (the probability that two consecutive probes encounter delays d_1 and d_2 on link k).

Because of temporal dependence, $\alpha_k(I, \mathbf{d})$ is not in general the product $\prod_{j=1}^{s} \Pr[Z_k(i_j) = d_j]$. However, thanks to spatial independence, $\forall k \in V$ the *path pattern probability* $A_k(I, \mathbf{d})$ can be expressed as a convolution of joint link and subpath probabilities as follows:

$$\begin{aligned}
&A_k(I, \mathbf{d}) = \Pr[\mathbf{X}_k(I) = \mathbf{d}] \\
&= \sum_{\mathbf{0} \leq \mathbf{q} \leq \mathbf{d}} \Pr[\mathbf{X}_k(I) - \mathbf{X}_{f(k)}(I) = \mathbf{q}]\ \Pr[\mathbf{X}_{f(k)}(I) = \mathbf{d} - \mathbf{q}] \\
&= \sum_{\mathbf{0} \leq \mathbf{q} \leq \mathbf{d}} \alpha_k(I, \mathbf{q})\ A_{f(k)}(\mathbf{d} - \mathbf{q}) \qquad (5)
\end{aligned}$$

The goal is to estimate the link probabilities $\alpha_k(I, \mathbf{d}), \mathbf{0} \leq \mathbf{d} \leq \mathbf{m}$ for each link $k \in U$. From Eq. (5), these can be recursively deconvolved if we know the path probabilities $A_k(I, \mathbf{d}), \mathbf{0} \leq \mathbf{d} \leq \mathbf{m}, \forall k \in V$. We therefore proceed to derive expressions for $A_k(I, \mathbf{d})$ (deconvolution appears later in section V-E).

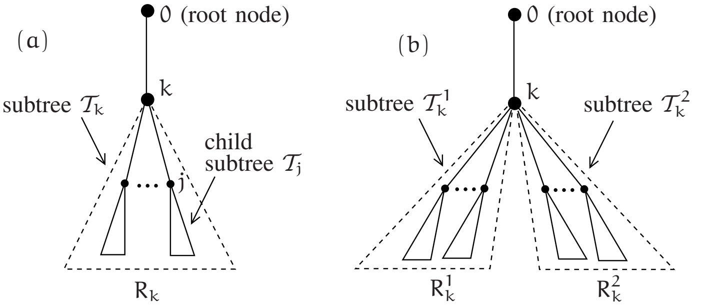
0 (root node)
(a)
k
subtree $\mathcal{T}_k$
child subtree $\mathcal{T}_j$
j
R_k
0 (root node)
(b)
k
subtree $\mathcal{T}_k^1$
subtree $\mathcal{T}_k^2$
R_k^1
R_k^2

Fig. 4. The subtree $\mathcal{T}_k$ with root at node k. (a): node k has a number of child subtrees rooted at the child nodes $\{c(k)\}$. (b): subtree partitioning of $\mathcal{T}_k$. The child subtrees of k are partitioned into two virtual subtrees $\mathcal{T}_k^1$, $\mathcal{T}_k^2$, anchored at k, with receivers R_k^1, R_k^2.

B. Estimation of Path Pattern Probabilities

Consider a branch node $k \in U$ in the tree in figure 4(a). It is the root of the subtree $\mathcal{T}_k$ of $\mathcal{T}$ which has receivers R_k. For a probe i sent from the source, we define the following random variables and corresponding random vectors:

$$Y_k(i) = \min_{r \in R_k} X_r(i), \quad \mathbf{Y}_k(I) = [Y_k(i_1), \dots, Y_k(i_s)] \qquad (6)$$

$$\widetilde{Y}_k(i,d) = \begin{cases} 1 \text{ if } Y_k(i) - X_{f(k)}(i) \le d \\ 0 \text{ if } Y_k(i) - X_{f(k)}(i) > d \end{cases}$$

$$\widetilde{\mathbf{Y}}_k(I,\mathbf{d}) = [\widetilde{Y}_k(i_1,d_1),\ldots,\widetilde{Y}_k(i_s,d_s)] \tag{7}$$

Let $\mathbf{B} \in \{0,1\}^{|I|}$. We define $\forall k \in U$,

$$\begin{aligned} \gamma_k(I,\mathbf{d}) &= \Pr[\mathbf{Y}_k(I) \le \mathbf{d}] \\ \beta_k(I,\mathbf{d},\mathbf{B}) &= \Pr[\widetilde{\mathbf{Y}}_k(I,\mathbf{d}) = \mathbf{B}] \end{aligned} \tag{8}$$

where $\gamma_k(I,\mathbf{d})$ is the probability that, for each probe index $i_j \in I$, the minimum delay on any path from source S to receivers in R_k, does not exceed $d_j \in \mathbf{d}$. On the other hand, $\beta_k(I,\mathbf{d},\mathbf{B})$ is the probability that, for each probe index $i_j \in I$, the minimum delay on any path from node $f(k)$ to receivers in R_k is either $\le d_j$ or $> d_j \in \mathbf{d}$, depending on whether $b_j \in \mathbf{B}$ is 1 or 0. The $\gamma_k(I,\mathbf{d})$'s are directly observable from receiver data and we wish to use them to recover $A_k(I,\mathbf{d})$'s. Let $\mathbf{1} = [1,\ldots,1]$. The following convolution links A, β, and γ:

$$\begin{aligned} &\gamma_k(I,\mathbf{d}) = \Pr[\mathbf{Y}_k(I) \le \mathbf{d}] \\ &= \sum_{0 \le \mathbf{q} \le \mathbf{d}} \Pr[\mathbf{X}_{f(k)}(I) = \mathbf{q}]\ \Pr[\mathbf{Y}_k(I) - \mathbf{X}_{f(k)}(I) \le \mathbf{d} - \mathbf{q}] \\ &= \sum_{0 \le \mathbf{q} \le \mathbf{d}} A_{f(k)}(I,\mathbf{q})\ \beta_k(I,\mathbf{d}-\mathbf{q},\mathbf{1}) \end{aligned} \tag{9}$$

So far, the generalisation from single probes to probe patterns has been remarkably straightforward. This is not true for the next two properties, which give expressions for $\beta_k(I,\mathbf{d},\mathbf{B})$ in the cases $\mathbf{B}=\mathbf{1}$ and $\mathbf{B}\neq\mathbf{1}$.

Property 1 (Parent-Child) The following relationship holds between β_k and the $\{\beta_j, j \in c(k)\}$ of the children of k . For convenience we relabel the children as $j = 1, 2 \cdots c_k$ where $c_k = |c(k)|$. We have

$$\beta_k(I,\mathbf{d},\mathbf{1}) = \sum_{0 \le \mathbf{q} \le \mathbf{d}} \alpha_k(I,\mathbf{q})\ \eta_k(I,\mathbf{d}-\mathbf{q}) \tag{10}$$

where

$$\begin{aligned} \eta_k(I,\boldsymbol{w}) &= \sum_{\substack{\{\mathbf{B}_1,\ldots,\mathbf{B}_{c_k}\} \\ \text{s.t.} \vee_j \mathbf{B}_j = \mathbf{1}}} \prod_{j=1}^{c_k} \beta_j(I,\boldsymbol{w},\mathbf{B}_j) = 1 - \prod_{j=1}^{c_k} \big(1 - \beta_j(I,\boldsymbol{w},\mathbf{1})\big) \\ &+ \mathbf{1}_{|I|>1} \Big(\sum_{\substack{\{\mathbf{B}_1 \neq \mathbf{1},\ldots,\mathbf{B}_{c_k} \neq \mathbf{1}\} \\ \text{s.t.} \vee_j \mathbf{B}_j = \mathbf{1}}} \prod_{j=1}^{c_k} \beta_j(I,\boldsymbol{w},\mathbf{B}_j) \Big) \end{aligned} \tag{11}$$

In Eq. (11), we have first used the fact that OR over a subtree can be decomposed as a OR over child subtrees, and then the property that such child subtrees are mutually independent. The 'first' $(1 - \prod_j)$ term in Eq. (11) corresponds to all child subtree receiver events $\{\mathbf{B}_j\}$ such that $\mathbf{B}_j = \mathbf{1}$ for at least one j. In the traditional case where $I = \{i\}$, this is the only term, since it is the same event as $\vee_j \mathbf{B}_j = \mathbf{1}$. In the temporal case for arbitrary I this is not true, since which receivers see the minimum delay can be different for each i in I. As a result, there are extra terms with $\beta_j(I,\boldsymbol{w},\mathbf{B}_j)$ factors with $\mathbf{B}_j \neq \mathbf{1}$.

Property 2 (Recursion over index sets with $\mathbf{B} = \mathbf{1}$) One can express $\beta_k(I,\mathbf{d},\mathbf{B} \neq \mathbf{1})$ in terms of $\beta_k(I',\mathbf{d}',\mathbf{1})$, where $I' \subseteq I$. For instance, if $\mathbf{B} = [b_1 = 0, b_2, \ldots, b_s]$, and $I' = \{i_2,\ldots,i_s\}$, $\mathbf{B}' = [b_2,\ldots,b_s]$, $\mathbf{d}' = [d_2,\ldots,d_s]$, then

$$\begin{aligned} &\beta_k(I,\mathbf{d},\mathbf{B}) = \Pr[\widetilde{\mathbf{Y}}_k(I,\mathbf{d}) = \mathbf{B}] \\ &= \Pr[\widetilde{Y}_k(i_1,d_1) = 0, \widetilde{Y}_k(i_2,d_2) = b_2,\ldots,\widetilde{Y}_k(i_s,d_s) = b_s] \\ &= \Pr[\widetilde{Y}_k(i_2,d_2) = b_2,\ldots,\widetilde{Y}_k(i_s,d_s) = b_s] \\ &\quad - \Pr[\widetilde{Y}_k(i_1,d_1) = 1, \widetilde{Y}_k(i_2,d_2) = b_2,\ldots,\widetilde{Y}_k(i_s,d_s) = b_s] \\ &= \beta_k(I',\mathbf{d}',\mathbf{B}') - \beta_k(I,\mathbf{d},[1,b_2,\ldots,b_s]) \end{aligned}$$

which has eliminated 0 at i_1. The above can be applied recursively to eliminate all zeroes, resulting in terms of the form $\beta_k(I',\mathbf{d}',\mathbf{1})$, $I' \subseteq I$, $|I| - z(\mathbf{B}) \le |I'| \le |I|$, where $z(\mathbf{B})$ denotes the number of zeroes in $\mathbf{B}$. In general

$$\beta_k(I,\mathbf{d},\mathbf{B} \neq \mathbf{1}) = (-1)^{z(\mathbf{B})} \beta_k(I,\mathbf{d},\mathbf{1}) + \delta_k(I,\mathbf{d},\mathbf{B}) \tag{12}$$

where $\delta_k(I,\mathbf{d},\mathbf{B})$ is the appropriate summation of β_k's for index sets $I' \subset I$. For e.g., if $I = \{1,2\}$, $\mathbf{B} = \{0,1\}$, $\mathbf{d} = [d_1,d_2]$, then $\beta_k(I,\mathbf{d},\mathbf{B}) = -\beta_k(I,\mathbf{d},\mathbf{1}) + \big(\beta_k(\{2\},[d_2],\mathbf{1})$

Eq. (12) can be used in (11) to remove all $\mathbf{B}_j \neq \mathbf{1}$ terms, leaving only terms of $\mathbf{B}_j = \mathbf{1}$ type, giving

$$\begin{aligned} \eta_k(I,\boldsymbol{w}) &= 1 - \prod_{j=1}^{c_k} \big(1 - \beta_j(I,\boldsymbol{w},\mathbf{1})\big) \\ &+ \mathbf{1}_{|I|>1} \Big(\sum_{\substack{\{\mathbf{B}_1 \neq \mathbf{1},\ldots,\mathbf{B}_{c_k} \neq \mathbf{1}\} \\ \text{s.t.} \vee_j \mathbf{B}_j = \mathbf{1}}} \prod_{j=1}^{c_k} \ \big\{(-1)^{z(\mathbf{B}_j)} \beta_j(I,\boldsymbol{w},\mathbf{1}) \\ &\qquad + \ \delta_j(I,\boldsymbol{w},\mathbf{B}_j)\big\} \Big) \end{aligned} \tag{13}$$

By using (10) in (9) and simplifying the convolution, we get

$$\gamma_k(I,\mathbf{d}) = \sum_{0 \le \mathbf{q} \le \mathbf{d}} A_k(I,\mathbf{q})\ \eta_k(\mathbf{d}-\mathbf{q}) \tag{14}$$

Rewriting Eq. (9) for the $\{\beta_j(I,\mathbf{d}), j \in c(k)\}$, we get

$$\gamma_j(I,\mathbf{d}) = \sum_{0 \le \mathbf{q} \le \mathbf{d}} A_k(I,\mathbf{q})\ \beta_j(I,\mathbf{d}-\mathbf{q},\mathbf{1}) \tag{15}$$

Using the above three equations, the desired path pattern probabilities for node k, $A_k(I,\mathbf{d}), 0 \le \mathbf{d} \le \mathbf{m}$, can be computed using the observables $\gamma_k(I,\mathbf{d})$ and $\{\gamma_j(I,\mathbf{d}), j \in c(k)\}, 0 \le \mathbf{d} \le \mathbf{m}$. This is one of our main results. We have obtained a complete generalization from the case of single probes $I = \{i\}$ and estimating only the distribution $A_k(i,[d]), d \in \mathcal{D}'$, to a temporal case for patterns of probes I, and estimating $A_k(I,\mathbf{d}), \mathbf{d} \in \mathcal{D}'^{|I|}$. For $I = \{i\}$, the above equations reduce to the equations of Lopresti et.al. [4] for delay distribution estimation.

Recovery of $A_k(I,\mathbf{d})$ from the above equations involves two levels of recursion: (*i*) over delay vectors, which arises due to convolution, (*ii*) over index sets which arises due to summation term involving δ in Eq. (13). Note that $\delta(I,.,.)$ only contains terms involving $I' \subset I$ and therefore *does not* contain $A_k(I,.)$. Thus estimation can be performed recursively starting from $I = \{i\}$ when the summation term with δ vanishes and $\mathbf{d} = \mathbf{0}$ when the convolution vanishes. Each step of recursion involves solving polynomials of degree c_k (due to the $\prod_j$ terms in (13)) in the unknown A_k .

C. Example: Binary tree

We show the computation of $A_k(I, \mathbf{d})$ in a binary tree for pairs of consecutive probes i.e. $I = \{1, 2\}$. Consider a branch node k and its two children $j = \{1, 2\}$. Due to recursion over index sets, we start with the case of $I = \{1\}$.

Single probes $I = \{1\}$: The base case of recursion occurs for $I = \{i\}$ and $\mathbf{d} = [0]$. To simplify notation, we drop the index set I, $\mathbf{B} = \mathbf{1}$, and vector notation for delays. For example, $\beta_j(I, [d_1], \mathbf{1}) = \beta_j(d_1)$. Writing out Eqs. (14) and (15),

$$\gamma_k(0) = A_k(0)\{1 - (1 - \beta_1(0))(1 - \beta_2(0))\}$$
$$\gamma_j(0) = A_j(0)\beta_j(0) \tag{16}$$

from which $A_k(0)$ is recovered by solving a linear equation as $A_k(0) = (\gamma_1(0)\gamma_2(0))/(\gamma_1(0)+\gamma_2(0)-\gamma_k(0))$. Substituting back $A_k(0)$ gives the $\beta_j(0)$ for use in the next step. Assuming that A_k and β_j's have been computed $\forall\ q_1 < d_1$, $A_k(d_1)$ is recovered using (14) and (15) which take the form

$$\gamma_k(d_1) = A_k(0)\underset{*}{\eta_k(d_1)} + \underset{*}{A_k(d_1)}\eta_k(0) + \sum_{0<q_1<d_1} A_k(q_1)\eta_k(d_1 - q_1)$$

where $\eta_k(w) = \{1 - (1 - \beta_1(w))(1 - \beta_2(w))\}$

$$\gamma_j(d_1) = A_k(0)\underset{*}{\beta_j(d_1)} + \underset{*}{A_k(d_1)}\beta_j(0) + \sum_{0<q_1<d_1} A_k(q_1)\beta_j(d_1 - q_1) \tag{17}$$

The unknown terms are marked by a "*". $A_k(d_1)$ is recovered by solving a quadratic equation and substituting back $A_k(d_1)$ gives $\beta_j(d_1)$'s.

Pairs of consecutive probes $I = \{1, 2\}$: Again, to simplify the notation, we drop the index set I, $\mathbf{B} = \mathbf{1}$, and vector notation for delays. For e.g., $\beta_j(I, [d_1, d_2], \mathbf{1}) = \beta_j(d_1, d_2)$. The estimation proceeds from delay vector $[0, 0]$ until $[m, m]$. Assuming that A_k and β_j's have been computed for the set $\{[q_1, q_2] : q_1 \le d_1, q_2 \le d_2\} \setminus \{[d_1, d_2]\}$, $A_k(d_1, d_2)$ is recovered as follows. We expand Eqs. (14) and (15)

$$\gamma_k(d_1, d_2) = A_k(0, 0)\ \underset{*}{\eta_k(d_1, d_2)} + \underset{*}{A_k(d_1, d_2)}\ \eta_k(0, 0)$$
$$+ \sum_{\substack{q_1 \le d_1,\\ (q_1,q_2)\neq(0,0),}} \sum_{\substack{q_2 \le d_2 \\ (q_1,q_2)\neq(d_1,d_2)}} A_k(q_1, q_2)\ \eta_k(d_1 - q_1, d_2 - q_2)$$
$$\gamma_j(d_1, d_2) = A_k(0, 0)\ \underset{*}{\beta_j(d_1, d_2)} + \underset{*}{A_k(d_1, d_2)}\ \beta_j(0, 0)$$
$$+ \sum_{\substack{q_1 \le d_1,\\ (q_1,q_2)\neq(0,0),}} \sum_{\substack{q_2 \le d_2 \\ (q_1,q_2)\neq(d_1,d_2)}} A_k(q_1, q_2)\ \beta_j(d_1 - q_1, d_2 - q_2)$$
$$\eta_k(w_1, w_2) = 1 - \prod_{j=1}^{2}(1 - \beta_j(w_1, w_2))$$
$$+ \prod_{j=1}^{2}(\beta_j(w_1) - \beta_j(w_1, w_2)) + \prod_{j=1}^{2}(\beta_j(w_2) - \beta_j(w_1, w_2)) \tag{18}$$

The unknown terms are marked by a "*" and $A_k(d_1, d_2)$ is obtained by solving a quadratic equation.

D. Subtree Partitioning

In practice, solving polynomials arising from (13) implies numerical root finding for each node, each $I' \in I$, and each $\mathbf{d} \in \mathcal{D}'^{|I|}$, which can be computationally slow. We now present a computationally effective method of estimating the path pattern probabilities on trees of arbitrary order which only needs solutions of quadratic equations.

We use a technique called subtree-partitioning recently proposed in [13] for loss inference. The idea is to combine child subtrees of node k into two sets, indexed by $j \in \{1, 2\}$, corresponding to two virtual subtrees $\mathcal{T}_k^1$ and $\mathcal{T}_k^2$ of $\mathcal{T}_k$ with receivers R_k^j (details of the allocation do not affect what follows), see Fig. 4(b). We therefore define additional quantities corresponding to the two virtual subtrees

$$Y'^j_k(i) = \min_{r \in R_k^j} X_r(i), \quad \mathbf{Y}'^j_k(I) = [Y'^j_k(i_1), \dots, Y'^j_k(i_s)] \tag{19}$$

$$\widetilde{Y}'^j_k(i, d) = \begin{cases} 1 \text{ if } Y'^j_k(i) - X_k(i) \le d \\ 0 \text{ if } Y'^j_k(i) - X_k(i) > d \end{cases}$$

$$\widetilde{\mathbf{Y}}'^j_k(I, \mathbf{d}) = [\widetilde{Y}'^j_k(i_1, d_1), \dots, \widetilde{Y}'^j_k(i_s, d_s)] \tag{20}$$

and the corresponding probabilities:

$$\gamma'^j_k(I, \mathbf{d}) = \Pr[\mathbf{Y}'^j_k(I) \le \mathbf{d}]$$
$$\beta'^j_k(I, \mathbf{d}, \mathbf{B}) = \Pr[\widetilde{\mathbf{Y}}'^j_k(I, \mathbf{d}) = \mathbf{B}]$$

Since sets of disjoint child subtrees are still mutually independent, β_k can be related to β'^1_k and β'^2_k (previously this was to $\{\beta_j, j \in d(k)\}$). Then following the same sequence of steps as before, we get a simpler version of Eq. (13) and an equivalent of Eq. (15)

$$\eta_k(I, \boldsymbol{w}) = 1 - \prod_{j=1}^{2}\left(1 - \beta'^j_k(I, \boldsymbol{w}, \mathbf{1})\right)$$
$$+ \mathbf{1}_{|I|>1}\Bigg(\sum_{\substack{\{\mathbf{B}_1 \neq \mathbf{1}, \mathbf{B}_2 \neq \mathbf{1}\} \\ \text{s.t.} \mathbf{B}_1 \vee \mathbf{B}_2 = \mathbf{1}}} \prod_{j=1}^{2} \quad \{(-1)^{z(\mathbf{B}_j)}\beta'^j_k(I, \boldsymbol{w}, \mathbf{1})$$
$$+ \quad \delta^j_k(I, \boldsymbol{w}, \mathbf{B}_j)\}\Bigg) \tag{21}$$
$$\gamma'^j_k(I, \mathbf{d}) = \sum_{\mathbf{0} \le \mathbf{q} \le \mathbf{d}} A_k(I, \mathbf{q})\ \beta'^j_k(I, \mathbf{d} - \mathbf{q}, \mathbf{1}) \tag{22}$$

Using these Eqs. along with (14) leads to a quadratic expression in $A_k(I, \mathbf{d})$ for all nodes irrespective of their degree, which can be solved explicitly. By making arbitrary trees appear as binary in this way, numerical root finding is eliminated.

E. Deconvolving α's : Path probabilities to link probabilities

After having obtained $A_k(I, \mathbf{d})$, for all $k \in U$, for $\mathbf{0} \le \mathbf{d} \le \mathbf{m}$, $\alpha_k(I, \mathbf{d})$ for $\mathbf{d} \ge \mathbf{0}$, are recursively deconvolved as follows:

(i) For $\mathbf{d} = \mathbf{0}$:

$$\alpha_k(I, \mathbf{0}) = \frac{A_k(I, \mathbf{0})}{A_{f(k)}(I, \mathbf{0})} \tag{23}$$

(ii) For $0 < \mathbf{d} \leq \mathbf{m}$:

$$\alpha_k(I, \mathbf{d}) = \frac{A_k(I, \mathbf{d}) - \sum_{0<\mathbf{q}\leq\mathbf{d}} A_{f(k)}(I, \mathbf{q})\alpha_k(I, \mathbf{d}-\mathbf{q})}{A_{f(k)}(I, 0)} \tag{24}$$

(iii) When $\mathbf{d} \in \mathcal{D}^{|I|}$, $\mathbf{d} \leq \mathbf{m}$ does not hold, i.e., at least one element of $\mathbf{d}$ could be ∞. In this case, $\alpha_k(I, \mathbf{d})$ is obtained using $\alpha_k(I', \mathbf{q})$'s, $\mathbf{q} \leq \mathbf{m}$ where $I' \subseteq I$. For instance, consider a vector $\mathbf{d} = [d_1 = \infty, d_2, \ldots, d_s]$. Then $\alpha_k(I, \mathbf{d})$ can be expressed as

$$\begin{aligned} \alpha_k(I, \mathbf{d}) = \alpha_k(\{i_2, \ldots, i_s\}, [d_2, \ldots, d_s]) \\ - \sum_{q \leq m} \alpha_k(I, [q, d_2, \ldots, d_s]) \end{aligned} \tag{25}$$

For e.g., for the case of a single probes $I = \{1\}$, $\alpha_k(I, [\infty]) = 1 - \sum_{q\leq m} \alpha_k(I, [q])$.

F. Estimator Definitions

We now complete the definition of the estimators. $\gamma_k(I, \mathbf{d})$ can be estimated using the empirical frequencies as

$$\widehat{\gamma}_k(I, \mathbf{d}) \quad = \quad \frac{\sum_{i=0}^{n-|I|-1} \mathbf{1}_{\{\mathbf{Y}_k(i+I)\leq\mathbf{d}\}}}{n - |I| - 1} \tag{26}$$

Next, the $\widehat{\gamma}_k(I, \mathbf{d})$ and $\widehat{\gamma}_j(I, \mathbf{d}), j \in c(k)$ are used to obtain the estimator $\widehat{A}_k(I, \mathbf{d})$ for $A_k(I, \mathbf{d})$. Finally, following equations (23-25), $\widehat{\alpha}_k(I, \mathbf{d})$ are recursively deconvolved. In practice, the estimators are modified to ensure that solutions make physical sense, for example that $\widehat{\alpha}_k(I, \mathbf{d}) \in [0, 1]$.

The mean delay-run length of a subset of one or more states $H \subset \mathcal{D}$, μ_k^H is estimated using the respective single and successive joint link probabilities of states in the subset. Let $\alpha_k(\{i_1\}, [d]) = \alpha_k(d)$ and $\alpha_k(\{1, 2\}, [d, q]) = \alpha_k(d, q)$, then

$$\widehat{\mu}_k^H = \frac{\sum_{d\in H} \widehat{\alpha}_k(d)}{\sum_{d\in H} \widehat{\alpha}_k(d) - \sum_{d\in H} \sum_{q\in H} \widehat{\alpha}_k(d, q)} \tag{27}$$

VI. Analysis of Delay Estimator Properties

In this Section we outline the statistical properties of the estimators from Section V: consistency and the asymptotic variance.

All the estimators $\widehat{A}_k(I, \mathbf{d})$ are constructed as follows. The true joint transmission probabilities $A_k(I, \mathbf{d})$ obey a relation $A_k(I, \mathbf{d}) = g(\boldsymbol{\gamma})$ where $\boldsymbol{\gamma}$ is a set of probabilities of leaf events, for some function g. The form of g is relatively involved, being a composition of expressions for quadratic roots and iterations; however it is clearly a continuous function, in particular at $\boldsymbol{\gamma}$. $\widehat{A}_k(I, \mathbf{d})$ is then the plug-in estimator $\widehat{A}_k(I, \mathbf{d}) = f(\widehat{\boldsymbol{\gamma}})$, where $\widehat{\boldsymbol{\gamma}}$ are the empirical frequencies of the events that define $\boldsymbol{\gamma}$. Now by virtue of the stationary and ergodic assumption on the Z_k, $\widehat{\boldsymbol{\gamma}}$ converges almost surely to $\boldsymbol{\gamma}$ as the number of probes grows, i.e., the estimator is consistent, as asserted.

For estimators of this type, a function g of the empirical averages $\boldsymbol{\gamma}$ that, with g being also differentiable, one can determine that $\sqrt{n} \cdot (\widehat{A}_k(I, \mathbf{d}) - A_k(I, \mathbf{d}))$ is asymptotically Gaussian with zero mean as $n \to \infty$, with a covariance matrix of the form $\nabla g \cdot h \nabla g$ where ∇g is the matrix of partial derivatives of g with respect to the coordinates $\boldsymbol{\gamma}$, and h is the asymptotic covariance matrix between the $\boldsymbol{\gamma}$ [18].

VII. Simulation Experiments

In this section we illustrate the performance of the estimators using simulations. We conducted two classes of experiments: (*i*) Monte-Carlo simulations which obey the model assumptions from Section III, and (*ii*) TCP simulations using ns-2. We use the estimators from Section V-B for single and two packet indices, and measure the following parameters of link delay processes. For each state p, we estimate three parameters: $\alpha_k(p)$, $\alpha_k(p, p)$, and the mean run length μ_k^p. To assess the accuracy of estimates, we compute their relative errors (this combines bias and variance) as: $re(\theta) = |(\theta - \widehat{\theta})/\theta|$. In addition, we average respective relative errors over all states of a link to obtain the "per link" errors: $e_k(1) = \frac{\sum_p re(\alpha_k(p))}{m+1}$, $e_k(11) = \frac{\sum_p re(\alpha_k(p,p))}{m+1}$, and $e_k(\mu) = \frac{\sum_p re(\mu_k^p)}{m+1}$.

Model Carlo Simulations We experiment with two different families of discrete-time discrete-state link delay processes: (*a*) *Markov chains* (1-st order) to provide an example of a gentle departure from Bernoulli assumptions. In this family, the run or burst length of each delay state is geometrically distributed, and (*b*) *semi-Markov processes* with Zipf distributed run lengths, to provide much stronger temporal dependence. We use a Zipf with a finite support over $[1, 1000]$, a distribution which has a power-law shaped tail over a wide range of values, but which has all moments finite. The choice of run-length distribution determines the burstiness of delay states. For the marginal distribution of delay processes, we consider: (*i*) *Uniform* where the mass of the distribution is spread uniformly over all states. (*ii*) *Mixture distribution* inspired by queueing models, with a point mass at zero state followed by a geometric decay (discretized exponential) over remaining states.

Rows 1 through 4 of Fig 5 shows the results for model simulations. Rows 1 and 2 provide benchmark results for a two-receiver tree, for Markovian and Zipf semi-Markov delay processes respectively, with uniform delay distributions. The delay processes on links are identically distributed and the delay states on links have same mean run length. As results for each of the three links are similar since the tree is small, we show results for the shared link k only, and drop the subscript k. Plots (a)-(c) show errors $e(1)$, $e(11)$, and $e(\mu)$ respectively, plotted as a function of the mean run length for a fixed number of probes (30k). Plots show the median errors calculated over 256 independent repetitions of the experiments (the grey confidence intervals show 5th and 95th percentiles and give an idea of the variation over these). We see in general that performance degrades both with increasing number of states and the mean run length of states. Plot (a) reports error in estimation of probabilities for a single packet index, and (b) reports the error in estimation of joint probabilities for two packet indices. The performance is satisfying, with errors being only slightly larger than those for (a), despite the more

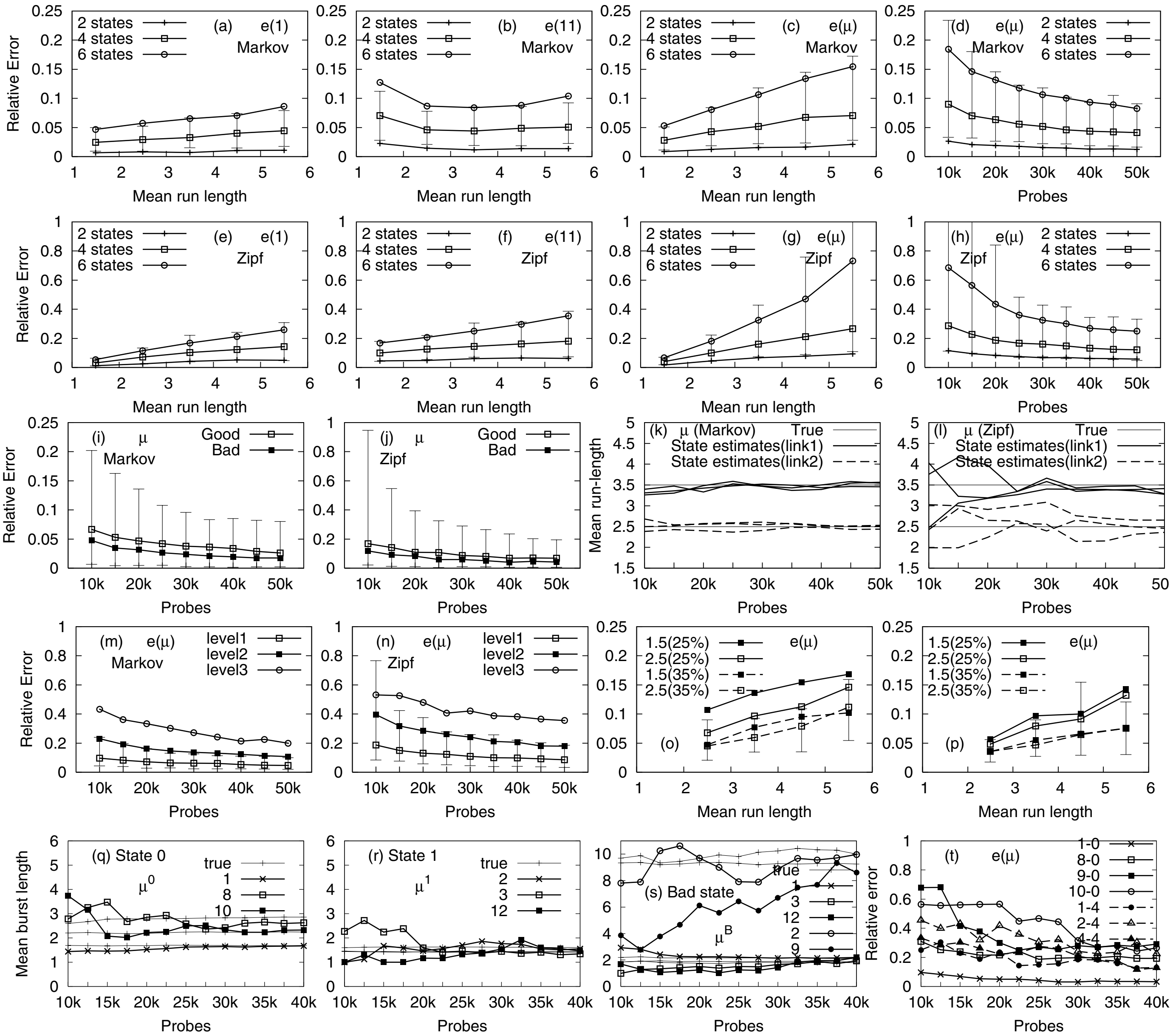

Fig. 5. Simulation results: Model simulations Rows 1-4, ns-2 simulations Row 5.

challenging, temporal target. Plots (c) reports errors in the estimation of mean run lengths of delay states. Because the computation of run lengths involve a subtraction and a division of estimated quantities ($\mu_k^p = \alpha_k(p)/(\alpha_k(p) - \alpha_k(p,p))$, we expect its variance to be greater that that of its components, resulting in an increased error. This is indeed what we see. Plot (d) shows relative errors for estimates of mean run length as the number of probes grow (10k-50k) for a fixed mean run length of 3.5. Plots (e)-(h) tell a similar story for Zipf semi-Markov processes, but as expected with an increased error due to the high burstiness of the process.

Plots (i), (j) focus on good and bad states, reporting relative errors in the estimation of their mean run lengths $e(\mu_k^G)$ and $e(\mu_k^B)$ for Markov and semi-Markov processes, in experiments with 4-state ($\{0,1,2,\infty\}$) link delay processes. The good state comprises delay states $\{0,1\}$ and the bad states $\{2,\infty\}$. Thus mean run length in the bad state corresponds to the mean duration of runs in which the delay is at least 2 units. The performance is satisfying because, in spite of aggregating states, the error in estimation compares well with the average of errors of mean run length of individual states.

Plots (k), (l) focus on a finer grained view of per-link behaviour, by showing the evolution of μ_k^p estimates with the number of probes for each state, and each link, in the case of a two-receiver tree with different link parameters. In this experiment, the link delay process had three states ($\{0,1,\infty\}$) and $\mu_1^p = 3.5$ and $\mu_2^p = 2.5$ for all p. We see that as the number of probes grow, estimates converge to the true values. The estimates of the Zipf case are much slower to converge than those of the Markov case.

Plots (m) and (n) focus on larger trees: tertiary trees with 3 levels, 9 receivers, and 4 delay states with mean run lengths of 2.5. Here subtree-partitioning is used. Estimated run length $e(\mu_k)$ is shown for links averaged over successive levels in the tree. As we go down the tree the variance increases.

Plots (o) and (p) examine the effect on $e(\mu_k)$ of a non-uniform delay distribution. Results for the shared link between two receivers are shown in the Markov case, for two different (see caption) 'mean of geometric decay(mass at zero)' combinations, for 20k probes. All states had the same mean run lengths and estimation was performed for delay states in $\{0, 1, 2, 3, \infty\}$.

We see that higher masses at zero state implies lower errors since estimation is performed recursively from the zero state. Also, when the geometric decay has a low mean, mass at higher states is too low to be accurately estimated, resulting in higher average error $e(\mu_k)$ over delay states. When the mean increases, some mass shifts to higher states, improving their estimates and reducing the average error. Plot (p) shows the effect of increasing the mean run length. This decreases burstiness of higher delay states which reduces estimation error.

TCP *Simulations* Plots (q)-(t) show the results of TCP simulations using a realistic tree topology (Fig 2(a)) with high bandwidth interior links (bold lines, 10Mpbs, propagation delay 50ms) and low bandwidth exterior links (thin lines, 2.5Mps, 10ms). The background traffic on each link is comprised of infinite source TCP connections. Multicast probes of 40 bytes were sent from the root node 0 using a Poisson process with mean inter-packet time of 16ms. The end-to-end probe delays were discretized using a bin size of 2ms into the state space $\{0, \ldots, m = 4, \infty\}$ and used for estimation. In these experiments, ∞ was considered as bad state. The mean run length of states $0..m$ varied between 1.5-3.5. On an average link delay distributions had a point mass of about 35% at state 0 and the mass decayed approximately geometrically over higher states. Mean run lengths of states decreased as we move from state 0 to m. In our experiments probes experienced higher queueing delays on interior links where buffer sizes were larger. Loss rates were low and remained below 10%.

Figures (q)-(s) show true and estimated mean run lengths of selected states over selected links. Error increases both when we move down the tree and towards higher delay states. Fig (q) shows how the mean run length of state 0 on various links converge to the true values. In (r) and (s) we show the same for state 1 and the bad state ∞. In our experiments links 2 and 9 experienced significant delays larger than m units, and hence have longer mean run lengths in the 'bad' state. Lastly, (t) shows average relative error to estimate mean run length for state 0 (links 1, 8, 9, 10) and state 4 (links 1, 2, 4). In general, error in TCP simulations remains higher than in model simulations due to the bias introduced by binning.

VIII. Conclusion

Packet traffic is bursty, and the performance of network applications depends on temporal metrics such as the duration of congestion events. In this paper we showed how tomography techniques can be extended to include the estimation of temporal parameters of link delay processes, which is of interest to delay sensitive applications, and application aware link and path health monitoring.

We derived an estimator capable of estimating temporal parameters such as the probability of arbitrary patterns of delays, and the mean run lengths of various delay states, for each link in the multicast tree. Run lengths can also be measured for sets of 'bad' and 'good' delay states.

The methods are non-parametric, working in a general context where the true nature of delays is unknown. We show how subtree-partitioning can be applied to make delay estimation scalable to large degree trees. We show that the estimators are consistent, and provide simulation illustrations of their relative errors as a function of parameters, and the number of probes, using Markovian and semi-Markovian delay processes (with Zipf tails) and ns-2 simulations.

In future work, the variance of the estimators presented here will be investigated in more detail, including the impact larger and less homogeneous trees, as well as real network traffic conditions.

References

[1] "AT&T Business Service Guide: AT&T VPN Service," February 2007, See: http://new.serviceguide.att.com/avpn.pdf.

[2] "AT&T U-verse," June 2007, https://uverse1.att.com/launchAMSS.do.

[3] R. Caceres, N. G. Duffield, J. Horowitz, D. Towsley, and T. Bu, "Multicast-Based Inference of Network-Internal Characteristics: Accuracy of Packet Loss Estimation," in *IEEE INFOCOM*, March 1999.

[4] F. Lo Presti, N. G. Duffield, J. Horowitz, and D. Towsley, "Multicast-based Inference of Network Internal Delay Distributions," *IEEE/ACM Transactions on Networking*, vol. 10, no. 6, pp. 761–775, 2002.

[5] N. G. Duffield, J. Horowitz, F. Lo Presti, and D. Towsley, "Multicast Topology Inference from Measured End-to-end Loss," *IEEE Transactions on Information Theory*, vol. 48, no. 1, pp. 26–45, 2002.

[6] Y. Gu, L. Breslau, N. G. Duffield, and S. Sen, "GRE Encapsulated Multicast Probing: A Scalable Technique for Measuring One Way Loss," in *ACM Sigmetrics*, June 2007.

[7] N. G. Duffield and F. Lo Presti, "Multicast Inference of Packet Delay Variance at Interior Network Links," in *IEEE INFOCOM*, March 2000.

[8] G. Liang and B. Yu, "Maximum Pseudo Likelihood Estimation in Network Tomography," *IEEE Trans. on Signal Processing (Special Issue on Data Networks)*, vol. 51, no. 8, pp. 2043–2053, 2003.

[9] Y. Tsang, M. Yildiz, P. Barford, and R. Nowak, "Network Radar: Tomography from Round Trip Time Measurements," in *ACM SIGCOMM Internet Measurement Conference*, October 2004.

[10] E. Lawrence, G. Michailidis, and V. Nair, "Flexicast Delay Tomography," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 68, no. 5, pp. 785–813, 2006.

[11] A. Chen, J. Cao, and T. Bu, "Network Tomography: Identifiability and Fourier Domain Estimation," in *IEEE INFOCOM*, May 2007.

[12] R. Castro, M. Coates, G. Liang, R. Nowak, and B. Yu, "Network Tomography: Recent Developments," *Statistical Science*, vol. 19, no. 3, pp. 499–517, 2004.

[13] V. Arya, N. G. Duffield, and D. Veitch, "Multicast Inference of Temporal Loss Characteristics," *Perform. Eval.*, vol. 64, no. 9-12, pp. 1169–1180, 2007.

[14] N. G. Duffield, F. Lo Presti, V. Paxson, and D. Towsley, "Inferring Link Loss using Striped Unicast Probes," in *IEEE INFOCOM*, April 2001.

[15] M. Coates and R. Nowak, "Network Loss Inference using Unicast End-to-end Measurement," in *ITC Seminar on IP Traffic, Measurement and Modeling*, September 2000.

[16] V. Jacobson, "Congestion Avoidance and Control," in *ACM SIGCOMM*, August 1988.

[17] Y. Zhang, N. Duffield, V. Paxson, and S. Shenker, "On the constancy of internet path properties," in *ACM SIGCOMM Internet Measurement Workshop*, November 2001.

[18] M. J. Schervish, *Theory of Statistics*. Springer, 1995.