Network Delay Tomography

Y. Tsang[1], M. Coates[2], R. Nowak[1]
Department of Electrical and Computer Engineering,
[1] Rice University, Houston, Texas, USA. E-mail: {ytsang, nowak}@rice.edu,

[2] McGill University, Montreal, Quebec, Canada. E-mail: coates@tsp.ece.mcgill.ca.

Abstract

The substantial overhead of performing internal network monitoring motivates techniques for inferring spatially localized information about performance using only end-to-end measurements. In this paper, we present a novel methodology for inferring the queuing delay distributions across internal links in the network based solely on unicast, end-to-end measurements. The major contributions are: (1) we formulate a measurement procedure for estimation and localization of delay distribution based on end-to-end packet pairs; (2) we develop a simple way to compute MLEs using the Expectation-Maximization (EM) algorithm; (3) we develop a new estimation methodology based on recently proposed nonparametric, wavelet-based density estimation method; and (4) we optimize the computational complexity of the EM algorithm by developing a new fast Fourier transform implementation. Realistic network simulations are carried out using `ns-2` to demonstrate the accuracy of the estimation procedure.

1 Introduction

Spatially localized information about network performance, such as link loss rates, queuing delays and available bandwidths, plays an important role in isolation of network congestion and detection of performance degradation. Routing algorithms, servicing strategies, security programs and performance verification can benefit from monitoring techniques that report such information. Monitoring can be performed internally, but it is impractical to directly measure traffic characteristics at all internal devices for a number of reasons [1]. This has prompted several groups to investigate methods for inferring internal network behavior based on "external" end-to-end network measurements [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]. This problem is often referred to as *network tomography*; see [11] for an overview of work in this area.

Queuing delays are one of the most critical performance characteristics. Optimizing communication network routing and service strategies requires knowledge of the queuing delay at different points in the network. Measuring end-to-end (source to receiver) delays using timestamps [8, 12, 13] is relatively easy and inexpensive in comparison to internal measurements, although there are of course measurement issues that must be addressed.

In this paper, we introduce a new methodology for network tomography, specifically, estimating the probability distribution of the queuing delay on each link based on end-to-end unicast packet pair measurements. Our approach employs unicast, end-to-end measurement of back-to-back packets. By back-to-back packets we mean two packets that are sent simultaneously by the source, possibly destined for different receivers, but sharing a common set of links in their paths.

1.1 Contribution

Earlier inference methodologies focused on multicast routing. In multicast routing, packets are delivered from sender to the receivers in one send operation. Along the path, probe packets are duplicated as needed as the paths diverge [2, 6]. Although multicast methods show promise for network performance inference, these techniques are often impractical in real networks. Many routers do not support multicast traffic and, if they do, they treat the packets differently from the majority of the traffic which is based on unicast routing. Therefore, inferences drawn from multicast routing may poorly reflect the actual network performance as observed by most traffic. However, the use of single unicast packets does not provide correlated measurements as do multicast packets. This motivates the use of back-to-back (closely time-spaced) unicast packets, which mimic the behavior of multicast packets to some degree.

Moreover, in this paper, we describe a *nonparametric* framework for the inference of internal delay distributions based on unicast end-to-end measurement. By nonparametric we mean that the number of parameters or the degrees of freedom diverges as a function of the number of delay measurements [14]. Most work to date in network tomography is based on *parametric* models. Parametric models assume that the measured traffic data depends on a finite number of parameters. For example, earlier work in delay distribution estimation has been based on discretized (or quantized) delay measurements, with internal delay distributions modeled as discrete probability mass functions (pmfs) [1, 4, 5]. In this context, the parameters are simply the probabilities associated with each pmf. It has been our experience, as well as that of others [15, 16], that no sufficiently simple parametric model is capable of portraying the wide variety of internal delay distributions observed in practice, thus motivating the consideration of nonparametric or continuous models. The complex nature of network delay distributions is evident in the simulated network measurements and estimates depicted in Figure 1.

Our methodology offers several significant advantages over existing methods: (1) it utilizes unicast measurement so that inferred performance reflects the experience of the majority of network traffic; (2) the estimation procedure is nonparametric and very flexible, in that it is capable of recovering densities from a broad range of function spaces including Bounded Variation (BV) functions and Besov spaces, which include both smooth and piecewise smooth densities; (3) the use of a Multiscale Maximum Penalized Likelihood Estimator (MMPLE) provides a computationally fast method for balancing the bias-variance trade-off and has been shown to be nearly optimal for density estimation in the above-mentioned function spaces [17, 18]; (4) we develop a new fast Fourier

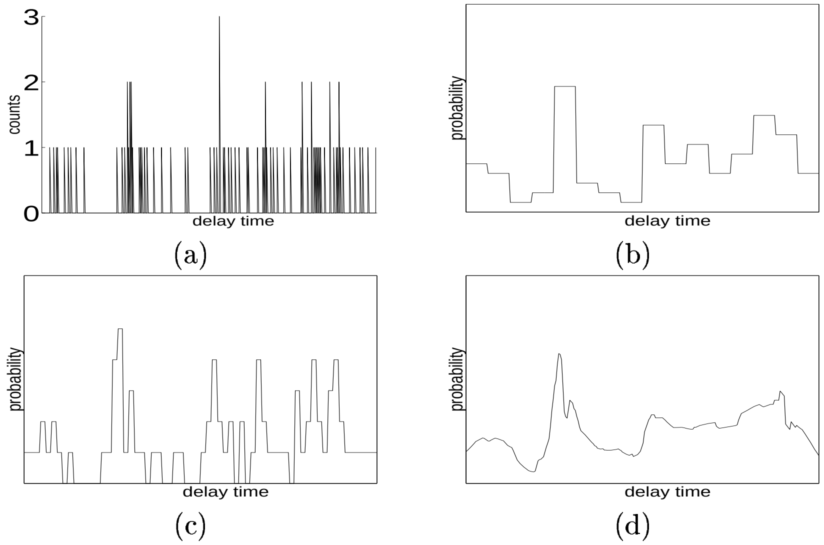


Figure 1: (a) ns-2 delay measurements using 170 packets on link 9 for network depicted in Fig. 2. Horizontal axis is (discretized) delay time and vertical axis denotes the number of occurrences of a particular delay measurement. (b) Discretized pmf with 16 equal-width bins. (c) Discretized pmf with 64 bins. (d) Nonparametric density estimate proposed in this paper obtained by direct estimation using link delays.

transform based implementation of the Expectation-Maximization (EM) algorithm for the network tomography problem which, in combination with MMPLE, leads to a worst-case overall complexity of $O(MN^2 \log N)$ where M is the number of links in the network and N is the number of packet pair measurements. In general, the complexity is substantially less than this (see Section 3.4 for clarification). We demonstrate the flexibility and accuracy of the nonparametric approach through `ns-2` [19] simulation.

1.2 Related Work

Lo Presti *et al.* have outlined a framework for the inference of internal queuing delay distributions based on multicast end-to-end measurement [1]. Multicast-based procedures for estimating low order moments such as link delay variances have also been developed [20]. The multicast framework has the advantage of scalability (each measurement probe provides some information about all links in the considered network) and guaranteed, structured correlation between the delay measurements at different receivers. However, multicast is not supported by all networks and there is evidence that routers treat multicast packets differently from the unicast packets that make up the majority of network traffic [6]. These concerns motivate the development of an inference framework based on unicast measurement. However, an important new consideration arises in the unicast setting. For a fixed measurement overhead, multicast measurement provides much more data than unicast. This means that if the framework of [1] were adapted to unicast measurement, as suggested in [6], it would need to perform with significantly less information available.

Lai and Baker [8] have implemented **`nettimer`**, a procedure that estimates link-level bandwidth.

Similar in nature to pathchar [21], it exploits the time-to-live field of packets to collect informative measurements. nettimer generates accurate estimates of bandwidths (particularly when they are small), although it requires a relatively large number of measurement packets. Theoretically, it could be used to estimate queuing delays, but to our knowledge there has been no experimental work exploring its performance. The number of measurement packets needed for estimation may prove prohibitive given the short duration over which delay distributions are generally stable. It would seem that network utilization would need to be low in order to achieve reliable estimates.

Shih and Hero have developed a method for estimation of the link delay cumulant generating functions (CGFs) [22, 23]. The CGFs have the advantage of being additive over a path of several links in contrast to the convolutional way in which link delay pmfs combine to form end-to-end delay pmfs. Based on the disentangled CGFs, it is straightforward to reconstruct the delay distributions. This technique has the benefit of imposing no discretization, but does not impose smoothness constraints, leading to an ill-posed problem when data is limited. The chief disadvantage of the technique is that in order for all links to be resolved, internal measurements must be available or a tool such as nettimer must be used.

Coates and Nowak have described a sequential Monte Carlo-based internal delay estimation framework in [4, 5]. This framework directly addresses the time-varying nature of network delay behavior. In this approach, a fine-level of quantization can be imposed, and smoothness is incorporated through the adoption of a slowly-varying time-dependent Bayesian prior. However, the parameters associated with the prior introduce a potentially undesirable parametric nature to the estimation task.

Anagnostakis and Greenwald have explored the feasibilities of using existing network infrastructure in making delay measurements [24, 25]. They have also studied the differences in direct measurements and indirect inference for determining the internal delays. The direct measurements depend on the Timestamping mechanism of the ICMP protocol [26]. However, they did not evaluate the inaccuracy in ICMP timestamping mechanism and they have assumed both sender and receivers are synchronized.

Recently, several studies have explored other forms of delay models [27, 16, 15]. The accuracy complexity trade-off is the motivation for all these researches. Duffield *et al.* [16] have described a varying bin size model for estimating the link delay distribution where the delay bin size is a composition of fixed bin size models. The idea is that the smaller bins are used to capture the small delay values. The larger bins are used to prevent explosion of the numbers of parameters and to capture the delays experienced by slower links. The authors then relate the varying bin size model to the fixed bin size model where the analysis takes place. The construction of varying bin size is chosen a priori or based on the measurements.

In a recent paper by Shih and Hero [15], a finite mixture model is proposed to estimate the link delay probability distribution functions. They model the delay with continuous Gaussian mixture components and assume that the components in the link delay distribution have distinct means and variances.

1.3 Paper Structure

The remainder of the paper is structured in the following manner. In Section 2 we describe the measurement framework, modeling assumptions and implementation requirements. In Section 3 we describe the inference methodology, detailing the MMPLE procedure and EM algorithm. In Section 4 we describe the results of `ns-2` experiments designed to explore the performance of the methodology. In Section 5, we make some concluding remarks.

2 Measurement Framework

Throughout this paper, we concentrate on networks comprised of a single source transmitting measurement probes to multiple receivers. There is no difficulty extending the approach to measurements made at multiple sources, although care must be taken that measurements are sufficiently separated for independence assumptions to hold. We assume that the topology is fixed throughout the measurement period, but straightforward extensions can account for changes in topology over coarse time scales. The assumption of fixed topology implies every probe packet sent to a specific receiver traverses the same path; i.e., the routes are unique, there are no route change during the measurement period nor load-balancing in the routers.

For the networks we consider, standard network routing protocols force packets to follow a specific route indicated by the routing table and they produce a tree-structured topology, with the *source* at the root and the *receivers* at the leaves. A network with six receivers is depicted in Fig. 2. The nodes between the source and receivers represent internal devices (e.g., *routers*, *switches*, or other buffering elements). For simplicity, we will refer to all internal nodes as "routers." Connections between the source, routers, and receivers are called *links*. Each link between routers may be a direct connection, or there may be "hidden" routers (where no branching occurs) along the link that are not explicit in our representation. We adopt the notation that link i connects node i (below) to its parent node (above). We consider the situation where measurements can only be made at the edge of the network and assume that the routing table (and thus topology) is fixed and known for the duration of the measurement.

The basic measurement and inference idea is quite straightforward. Suppose two closely time-spaced (back-to-back) packets are sent from the source to two different receivers. The paths to these receivers traverse a common set of links, but at some point the two paths diverge (as the tree branches). The two packets should experience approximately the same delay on each shared link in their path. This facilitates the estimation of the delays occurring on each link.

We collect measurements of the end-to-end delays from source to receivers, and we index the packet pair measurements by $k = 1, \ldots, N$. For the k-th packet pair measurement, let $y_1(k)$ and $y_2(k)$ denote the two end-to-end delays measured. The ordering 1 and 2 is *arbitrary*; the indices are randomly selected with no dependence on the order in which the packets were sent from the source. This will be important in dealing with discrepancies between the delays experienced by the two

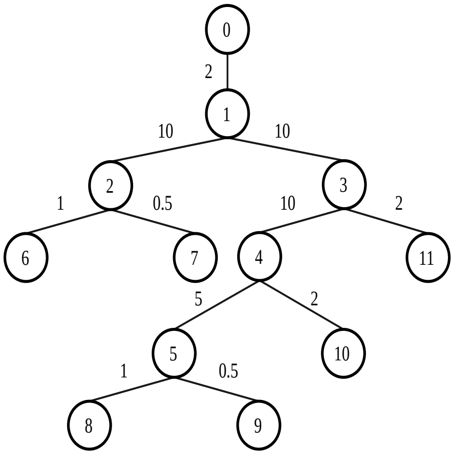


Figure 2: Tree-structured network topology used for ns-2 simulation experiments. Source (node 0) transmits to 6 receivers (nodes 6-11). Link speeds in Mb/s are shown next to the links. Link i connects node i to its parent node, e.g. link 9 connects nodes 5 and 9.

packets on shared links, which will be discussed in greater detail in Section 2.1. In this paper, we do not consider the case in which one or both of the packets is dropped (lost). We simply discard packet pairs in which a loss occurs. However, it is possible to extend our approach to include losses. Since we are interested in inferring queuing delay, our first step is to extract what we perceive as the minimum delay (propagation + transmission) on each measurement path. The minimum delay corresponds to the case in which all queues in the path are empty (i.e., no queuing delay). This is estimated as the smallest delay measurement we acquire on the path during the measurement period. We assume that the true minimum delay is observed over the measurement period. If this is not the case, then queuing delay is systematically underestimated for links on the affected path.

Our goal is a nonparametric estimate of the delay distributions on each link. Clearly it is impossible to completely determine an infinite dimensional density function from a finite number of delay measurements, but we require that as the number of delay measurements increases so does the accuracy of our estimation procedure. Thus, we adopt the following procedure. The end-to-end delay measurements are binned, *but* the number of bins is chosen to be equal to or greater than the number of delay measurements. We stress that this is not a parametric step. This means that there is less than one measurement per bin, on average, and hence we do not lump or group delays in an artificial, prescribed fashion. Thus, we place no prior restriction on the form of the density estimator; the more measurements one has, the more one can resolve the structural nuances of the delay densities.

In practice we choose the number of bins to be the smallest power of two greater than or equal to the number of measurements (facilitating certain processing steps to be described later). We upper bound the maximum delay on any one link by the maximum end-to-end delay along the path(s) that include the link. Let $d_{\max}$ denotes the maximum path delay on any links, this upper bound for a particular link and let K be the smallest power of 2 that is greater than or equal to the number of measurement packets N. The bin width for the link is then set at $d_{\max}/(K-1)$. This procedure is conservative, in that the estimated $d_{\max}$ may be substantially larger than the true maximum queuing delay. It may be preferable to use previous link-delay estimates or bandwidth estimates from a procedure such as `nettimer` [8] to gauge the maximum delay on any link.

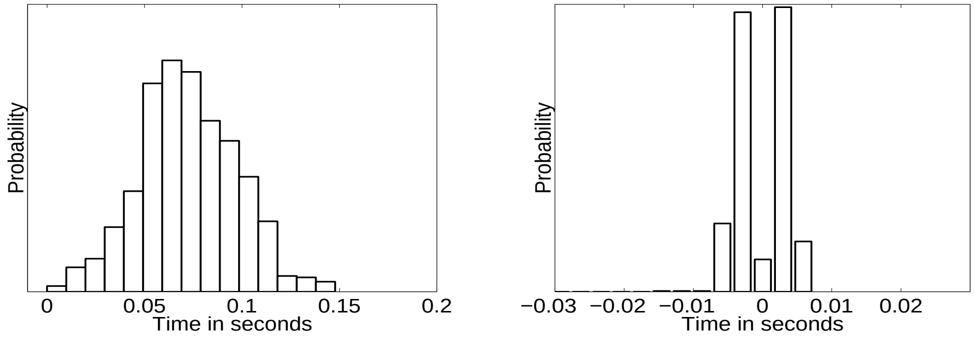
At this stage, each end-to-end measurement has been ascribed a discrete number between 0 and $(K-1)$. To illustrate our inference methodology in its simplest form, suppose that we send many packet pairs to receivers 6 and 7 in Fig. 2 and measure the delays experienced by each packet. Each measurement consists of a pair of delays, one being the delay to receiver 6 and the other the delay to receiver 7. From these measurements, collect events where '0' delay (a delay in bin zero) is measured at receiver 6. Now, assuming that the delay is the same for both packets on the common links (1 and 2 in this case), any "additional" delay observed to the receiver at 7 can be attributed to link 7 alone. We can then build a histogram estimate of the delay pmf for link 7. This simple idea can be extended and improved to obtain estimators for the delay distributions on all links which take advantage of *all* the measured data (not just special cases like the one above). In Section 3 , we describe the large-scale inference procedure in detail.

The basic inference idea is simple. Suppose the network is stationary over each measurement period, the delays are identical on shared links, and The true delay pmfs are strictly positive and canonical (there is some mass in the zero delay bin). This implies the first packet has left the queue before the second packet enters. The delay experienced by the second packet will not be dependent on the delay of the first one. In addition, suppose that the link delays experienced by an individual packet are independent of one another as in the multicast scenario. Then, based on the identifiability analysis carried out for the multicast case [1], one can easily show that the true distributions can be uniquely identified from such end-to-end measurements (as the number of measurements tends to infinity). The issue about slightly different delays on shared link in practice will be addressed in the following section. It is important to point out that unique identification is not possible (in general) using single packet delay measurements; there are ambiguous cases that cannot be resolved without multiple packet correlations [3].

2.1 Model Assumptions

There are several assumptions in the framework that are worthy of discussion. Firstly, we assume spatial independence of delay. Delay on neighboring links is generally correlated to a greater or lesser extent depending on the amount of shared traffic. In the `ns-2` [19] experiments discussed in Section 4, weak correlation of delays is observed. In the presence of weak correlation, our framework is able to derive good estimates of the delay distributions. As the correlation grows stronger, we see a gradual increase of bias in the estimates. We also assume temporal independence (successive probes across the same link experience independent delays). Temporal dependence was observed in [1] and in our experiments; indeed it is exploited in [5]. As in [1], the maximum likelihood estimator we employ remains consistent in the presence of temporal dependence, but the convergence rate slows. It practical situations, dependencies are usually weak and do not have a dramatic effect on the performance of the estimator. Ignoring dependencies can also be interpreted and analyzed as a case of Besag's *pseudolikelihood* approach [28].

Finally, our framework hinges on an assumption that packets in a pair experience a common delay on shared links. If the delays are identical on shared links, then the difference between the two delay measurements can be attributed solely to the delays experienced on unshared links in the two paths. This is the key to uniquely determining the delays on a link-by-link basis. However, in practice the two packets may experience slightly different delays on shared links due to the fact that one packet precedes the other in the common queues and additional packets may intervene between the two. The nature of this delay differential is exposed in Fig. 3, which shows the histogram of the difference between the end-to-end delays of two closely-spaced packets sent to the same receiver over the Internet. This histogram is constructed from back-to-back packet pair measurements using the `netdyn` tool [13]. Ideally, the delays should be identical, but we see a small discrepancy between the two. The second packet in the pair typically experiences a slightly greater delay. However, recall that the ordering of the packets was arbitrary in our recording process. In effect then, the discrepancies between the delays on shared links adds an approximately zero mean error to the difference between the two end-to-end measurements. We clearly see the symmetric zero-mean nature in the empirical data shown in Fig. 3, and we have observed similar behavior in all our measurements and simulations. This "noise" produces a smoothing (or blurring) in the inferred delay pmfs. Nonetheless, because the errors are roughly zero mean, we can still use the estimated delay pmfs to obtain approximately unbiased estimates of the expected delay (Fig. 3b) on each link or the locations of modes in the density, for example. The errors could also be directly modeled, but our experimentation suggests that these errors are relatively insignificant in the overall process, due to the greater variability caused by the limited number of probes that can be used in practical situations.



(a) (b)

Figure 3: (a) End-to-end delay histogram (packets sent from Rice University to Michigan State University). (b) Difference between delays of the two packets in packet pairs. Measurements were made using the *netdyn* tool.

2.2 Measurement Requirements

The delay inference framework requires knowledge of the (logical) topology of the network and the capability to perform one-way delay measurements. We perform the construction of the topology using a modified, lightweight version of `traceroute` [29, 30]. Alternatively, it is possible to determine the topology using the end-to-end unicast measurement and inference procedure we recently proposed in [31]. Collection of one-way delay measurements requires that the receivers cooperate with the source and the precision of the system timing [32].

We do not necessarily require that clocks at the source and receivers be synchronized, but we do require that the disparity between clocks remain very nearly constant over the measurement period. In this way we can be sure that subtracting the estimated minimum delay does not induce bias in our estimates. A further difficulty lies in clock resolution. Clocks must be precise enough to ensure that time measurement errors are insignificant relative to the scale of the time delays of interest. Deployment of Global Positioning System (GPS) devices allows these clock difficulties to be avoided, as it provides synchronized measurements to within tenths of microseconds. Alternatively, delay measurements can be adjusted using algorithms developed to detect and compensate for clock adjustments and rate discrepancies [33, 34, 32]. In this paper, we assume that synchronized measurements are available.

3 Delay Distribution Inference

We commence with the description of our inference framework by formalizing our measurement and modeling notation. Let $p_i = \{p_{i,0} \ldots, p_{i,K-1}\}$ denote the probabilities of a delay of $0, 1, \ldots, K-1$ time units, respectively, on link i. We denote the packet pair measurements $\boldsymbol{y} \equiv \{y_1(k), y_2(k)\}_{k=1}^{N}$.

In general, only a relatively small amount of data can be collected over the period when delay distributions can be assumed approximately stationary. A natural estimate would be the maximum likelihood estimates (MLEs) of $\boldsymbol{p} \equiv \{p_i\}$, the collection of all delay pmfs. However, if a large number of bins is used (i.e., high-resolution delay estimates), then the problem is ill-posed and the MLE tends to overfit to the probe data (see Fig. 1(a)), producing highly variable estimates that do not accurately reflect the delay distribution of the traffic at large. High variance manifests itself in irregular, noisy-looking estimates [35]. One way to reduce this irregularity is to maximize a penalized likelihood (see Fig. 1(d)). We replace the maximum (log) likelihood objective function $L(\boldsymbol{p}) = \log l(\boldsymbol{y}|\boldsymbol{p})$ with an objective function of the form:

$$L(\boldsymbol{p}) - \text{pen}(\boldsymbol{p}) \tag{1}$$

where $\text{pen}(\boldsymbol{p})$ is a non-negative real-valued functional that penalizes the irregularity (or *complexity*) of $\boldsymbol{p}$. A small value of $\text{pen}(\boldsymbol{p})$ indicates that $\boldsymbol{p}$ is a smooth, regular function; a large value indicates that $\boldsymbol{p}$ is irregular and complex function. The maximization of this penalized log-likelihood involves a trade-off between fidelity to the data (large $L(\boldsymbol{p})$) and smoothness or simplicity (small $\text{pen}(\boldsymbol{p})$).

We will describe a specific choice of penalty functional in Section 3.2. Before moving to that, however, we will quickly formulate the basic likelihood function and motivate the adoption of an EM algorithm for optimization.

3.1 Likelihood Function

Under the assumption of spatial independence, the likelihood of each delay measurement $\{y_1(k), y_2(k)\}$ is parameterized by a convolution of the pmfs in the path from the source to receiver. With our modeling constraint that packets in a pair experience the same delay on shared links, the likelihood of the two measurements made by the k-th packet pair is:

$$\begin{aligned} l(y_1(k), y_2(k)|\boldsymbol{p}) &= \\ &\sum_j \rho_{c,k}(j)\rho_{1,k}\left(y_1(k)-j\right)\rho_{2,k}\left(y_2(k)-j\right). \end{aligned} \tag{2}$$

The range of the summation is determined by the ranges of the pmfs $\rho_{c,k}$, $\rho_{1,k}$ and $\rho_{2,k}$. The pmf $\rho_{c,k}$ is the convolution of the pmfs of the links on the shared path of the two packets, e.g. $\rho_{c,k} = p_1 * p_2$ for a 6-7 pair in Fig. 2 (with $*$ denoting convolution). The pmf $\rho_{1,k}$ (resp. $\rho_{2,k}$) is the convolution of the pmfs on the links traversed only by the packet that measures $y_1(k)$ (resp. $y_2(k)$). The joint likelihood $l(\boldsymbol{y}|\boldsymbol{p})$ of all measurements is equal to a product of the individual likelihoods:

$$l(\boldsymbol{y}|\boldsymbol{p}) = \prod_{k=1}^{N} l(y_1(k), y_2(k)|\boldsymbol{p}). \tag{3}$$

The presence of convolved link pmfs in the likelihood of each measurement (2) results in an objective function that cannot be maximized analytically. The maximization of the likelihood function requires numerical optimization, and an Expectation-Maximization (EM) algorithm [36] is an attractive strategy for this purpose. Before giving the details of the algorithm, we briefly review the multiscale maximum penalized likelihood estimate (MMPLE) nonparametric density estimation procedure employed in our framework.

3.2 MMPLE Density Estimation

Here we briefly outline the MMPLE density estimation procedure developed in [17, 18]. To introduce the idea, we consider a case where the link delays have been directly measured. Let $z_i(k)$, $k = 1, \ldots, N_i$, denote a set of delay measurements for a particular link i. We assume that these measurements are independent and identically distributed according to a continuous delay density $p(t)$, where without loss of generality we assume that $t \in [0, 1]$ (for convenience of exposition we take the maximum delay to be unity). Define a discrete pmf via $p_{i,j} = \int_{(j)/K}^{(j+1)/K} p(t)dt$, $j = 0, \ldots, K-1$, where K is the smallest power of two greater than or equal to N_i. It follows that the number of measurements falling in the interval $[\frac{j}{K}, \frac{(j+1)}{K}]$, denoted $m_{i,j}$, is multinomially distributed [14], i.e., $\{m_{i,j}\} \sim \text{Multinomial}(N_i; \{p_{i,j}\})$. The MMPLE estimator maximizes the following criterion with respect to $\{p_{i,j}\}$:

$$\log \text{Multinomial}(N_i; \{p_{i,j}\}) - \text{pen}(\{p_{i,j}\}), \tag{4}$$

where

$$\text{pen}(\{p_{i,j}\}) \equiv \frac{1}{2}\log(N_i) \times \#_i, \tag{5}$$

where $\#_i$ is the number of non-zero coefficients in the discrete Haar wavelet transform of the pmf $\{p_{i,j}\}$. This number reflects the irregularity and complexity of the pmf — the larger the value of $\#_i$, the more "bumps" in the pmf. There are two important features of the MMPLE: (1) the global maximizer can be computed in $O(K)$ operations; (2) the MMPLE is nearly minimax optimal in the rate of convergence over a broad class of function spaces [17, 18].

Computing the MMPLE is very similar to standard wavelet denoising methods. Finding the optimal solution to (4) involves computing the Haar wavelet transform of the pmf and thresholding ("keeping" or "killing") each Haar wavelet coefficient according to a generalized likelihood ratio test (GLRT). Due to the multinomial form of the likelihood, the GLRTs involve binomial statistics (instead of the usual Gaussian statistics involved in standard wavelet denoising problems). The physical interpretation of each GLRT is simple: if the magnitude of the wavelet coefficient is sufficiently large, then that coefficient is left unaltered, otherwise it is set to zero. In detail, the MMPLE estimator is computed according to the four steps below.

i. Compute the (unnormalized) Haar scaling coefficients of the sequence $\{m_{i,j}\}$ as follows. For scales $\ell = 0, \ldots, \log_2 N_i$

$$s^\ell_{i,j} \;=\; \sum_{j=1}^{2^\ell} m_{i,j+k2^\ell}, \quad k = 0, \ldots, N_i/2^\ell - 1.$$

Note that $\{m_{i,j}\}$ the scaling coefficients at scale $\ell = 0$.

ii. Form the "multiscale coefficients"

$$\rho^\ell_{i,j} \;=\; s^{\ell-1}_{i,2j}/(s^{\ell-1}_{i,2j} + s^{\ell-1}_{i,2j+1}).$$

Note that $s^\ell_{i,j} = s^{\ell-1}_{i,2j} + s^{\ell-1}_{i,2j+1}$. Therefore, the scaling coefficients at scale $\ell - 1$ can be constructed from the scaling coefficients at scale ℓ along with the multiscale coefficients at scale ℓ according to

$$s^{\ell-1}_{i,2j} = \rho^\ell_{i,j} s^\ell_{i,j} \quad \text{and} \quad s^{\ell-1}_{i,2j+1} = (1-\rho^\ell_{i,j}) s^\ell_{i,j}. \tag{6}$$

The multiscale coefficients are closely related to the usual Haar wavelet coefficients. Specifically, the (unnormalized) Haar wavelet coefficient

$$\begin{aligned} \omega^\ell_{i,j} &= s^{\ell-1}_{i,2j} - s^{\ell-1}_{i,2j+1} \\ &= (2\rho^\ell_{i,j} - 1) s^\ell_{i,j} \;. \end{aligned}$$

Note, in particular, that if $\rho_{i,j}^{\ell} = 1/2$, then $\omega_{i,j} = 0$.

iii. Compute the test statistic

$$t_{i,j}^{\ell} \;=\; s_{i,2j}^{\ell-1}\left[\log(\rho_{i,j}^{\ell}) - \log(1/2)\right] + s_{i,2j+1}^{\ell-1}\left[\log(1-\rho_{i,j}^{\ell}) - \log(1/2)\right] \;,$$

and "threshold" the multiscale coefficients according to

$$\delta(\rho_{i,j}^{\ell}) = \begin{cases} 1/2, & \text{if } t_{i,j}^{\ell} < 1/2 \log N_i \\ \rho_{i,j}^{\ell}, & \text{if } t_{i,j}^{\ell} \geq 1/2 \log N_i \end{cases} .$$

iv. Construct the MMPLE estimate by recursively applying (6) beginning with $s_{i,1}^{\log_2 N_i} = 1$ and using the thresholded multiscale coefficients $\{\delta(\rho_{i,j}^{\ell})\}$ in place of the original coefficients. The resulting scale $\ell = 0$ scaling coefficients are the desired elements of the MMPLE estimator $\{\widehat{p}_{i,j}\}$.

The near minimax optimality implies that the rate at which the estimator converges to the true continuous density (as a function of the number of measurements N_i) cannot be significantly improved upon. More complicated and computationally intensive procedures will not significantly outperform the MMPLE. The optimization is carried out by performing a set of K independent generalized likelihood ratio tests. In all results in this paper we employ a *translation-invariant* version of the MMPLE, in which multiple MMPLEs are computed with K different shifted versions of the Haar wavelet basis and the resulting estimates are averaged. This produces a slight improvement over the basic MMPLE and can be efficiently computed in $O(K \log K)$ operations.

3.3 EM Algorithm

The MMPLE methodology can be employed in the tomographic delay estimation case by simply adopting the penalized likelihood criterion:

$$\log l(\boldsymbol{y}|\boldsymbol{p}) - \sum_i \frac{1}{2}\log(N_i) \times \#_i, \tag{7}$$

where N_i denotes the number probe packets passing through link i and $\#_i$ denotes the number of non-zero Haar wavelet coefficients in the delay pmf p_i of link i. Unfortunately, the penalized likelihood function cannot be maximized analytically due to the convolutional relationship between link delay pmfs and end-to-end measurements $\boldsymbol{y}$.

The first step in developing an EM algorithm is to propose a suitable *complete data* quantity that simplifies the likelihood function. Let $z_i(k)$ denote the delay on link i for the packets in the k-th pair. Let $\boldsymbol{z}_i = \{z_i(k)\}$ and $\boldsymbol{z} = \{\boldsymbol{z}_i\}$. The link delays $\boldsymbol{z}$ are not observed, and hence $\boldsymbol{z}$ is called the *unobserved data.* Define the *complete data* $\boldsymbol{x} \equiv \{\boldsymbol{y}, \boldsymbol{z}\}$. Note that the complete data likelihood may be factorized as follows:

$$l(\boldsymbol{x}|\boldsymbol{p}) = f(\boldsymbol{y}|\boldsymbol{z})g(\boldsymbol{z}|\boldsymbol{p}),$$

where f is the conditional pmf of $\boldsymbol{y}$ given $\boldsymbol{z}$ (which is a point mass function since $\boldsymbol{z}$ determines $\boldsymbol{y}$), and g is the likelihood of $\boldsymbol{z}$. The factorization shows that $l(\boldsymbol{x}|\boldsymbol{p}) \propto g(\boldsymbol{z}|\boldsymbol{p})$, since $f(\boldsymbol{y}|\boldsymbol{z})$ does not depend on the parameters $\boldsymbol{p}$. Next note that the likelihood

$$g(\boldsymbol{z}|\boldsymbol{p}) = \prod_{i,j} p_{i,j}^{m_{i,j}},$$

where $m_{i,j} \equiv \sum_{k=1}^{N} \mathbf{1}_{z_i(k)=j}$ is the number of packets (out of all the packet pair measurements) that experienced a delay of j on link i; here $\mathbf{1}_A$ denotes the *indicator function* of the event A. Therefore, we have

$$l(\boldsymbol{x}|\boldsymbol{p}) \propto \prod_{i,j} p_{i,j}^{m_{i,j}},$$

and, if the $m_{i,j}$ were available, then the MLE of $p_{i,j}$ would be simply

$$\widehat{p}_{i,j} = \frac{m_{i,j}}{\sum_{k=0}^{K-1} m_{i,k}}. \tag{8}$$

Similarly, given the $m_{i,j}$ we could directly apply the MMPLE described above (see [17, 18, 37] for implementation details).

The EM algorithm is an iterative method that constructs and utilizes a *complete data* likelihood function to maximize the original likelihood function. By suitable modification, it can be used to maximize a penalized log-likelihood objective function like (7), whilst preserving the advantage of the O(K) computational simplicity of the MMPLE technique.

When a modified EM algorithm is used to maximize a penalized log-likelihood function, it alternates between computing the conditional expectation of the complete data log likelihood given the observations $\boldsymbol{y}$ and maximizing the sum of this expectation and the imposed complexity penalty ($-$pen($\boldsymbol{p}$)) with respect to $\boldsymbol{p}$. Notice that, ignoring constant terms, the complete data log likelihood is linear in $\boldsymbol{m}$:

$$\log l(\boldsymbol{x}|\boldsymbol{p}) \propto \sum_{i,j} m_{i,j} \log p_{i,j}.$$

Thus, in the E-Step we need only compute the expectation of $\boldsymbol{m} = \{m_{i,j}\}$.

E-Step: Let $\boldsymbol{p}^{(r)}$ denote the value of $\boldsymbol{p}$ after the r-th iteration. Then

$$\begin{aligned} \widehat{m}_{i,j}^{(r)} &\equiv \boldsymbol{E}_{\boldsymbol{p}^{(r)}}[m_{i,j}|\boldsymbol{y}], \\ &= \boldsymbol{E}_{\boldsymbol{p}^{(r)}}\left[\sum_{k=1}^{N_i} \mathbf{1}_{\{z_i(k)=j\}}|\boldsymbol{y}\right], \\ &= \sum_{k=1}^{N_i} \boldsymbol{E}_{\boldsymbol{p}^{(r)}}\left[\mathbf{1}_{\{z_i(k)=j\}}|\boldsymbol{y}\right], \end{aligned}$$

$$\begin{aligned} &= \sum_{k=1}^{N_i} \boldsymbol{E}_{\boldsymbol{p}^{(r)}} \left[\mathbf{1}_{\{z_i(k)=j\}} | y_1(k), y_2(k) \right], \\ &= \sum_{k=1}^{N_i} p^{(r)} \left(z_i(k) = j | y_1(k), y_2(k) \right). \end{aligned} \tag{9}$$

Thus, the conditional expectation of $\boldsymbol{m}$ can be computed by determining the conditional probabilities above for each packet pair measurement. A fast message-passing algorithm for this calculation is described in the next section.

M-Step: In the penalized case (7), apply the MMPLE algorithm described in Section 3.2 with the conditional expectation $\{\widehat{m}_{i,j}^{(r)}\}$ in place of $\{m_{i,j}\}$. In the case of unpenalized maximum likelihood estimation, simply substitute $\{\widehat{m}_{i,j}^{(r)}\}$ in place of $\{m_{i,j}\}$ in equation (8).

3.4 Fast Fourier Transform based EM Algorithm

The expectation step of the EM algorithm poses the major portion of the computational burden of the optimization task. It can be performed using a message passing (or upward-downward) procedure [38]. The message passing procedure is based on a factorization of the likelihood function. According to (9), our task for each measurement in the r-th iteration of the EM algorithm is to compute $\boldsymbol{p}^{(r)}\left(z_i = j | y_1, y_2\right)$ (we have dropped the measurement index k for notational ease). In the 1980's, Pearl [39] and Spiegelhalter [40] independently developed the message passing methodology, an exact probability propagation algorithm for inferring the distributions of individual variables in singly-connected graphical models (factor graphs). The basic idea of the algorithm is that each node in the graph propagates its information (a measurement or current pmf estimate in this case) to every other node. Each node then combines all the messages it receives to compute the distribution of its variable.

Figure 4 depicts an example of the type of graphical model that arises in the delay inference procedure. This factor graph is used for evaluation of the pmf estimates in the $r+1$-th iteration of the EM algorithm. In this factor graph, the nodes labeled d_i correspond to the nodes of the tree that are involved in measurement to nodes 6 and 7 in the example network. The nodes $p_i^{(r)}$ contain the delay pmf estimates that were generated in the previous iteration of the algorithm. The nodes labeled z represent the *complete* data, that is, the unobserved individual link delays.

We will briefly illustrate the operation of message passing algorithm by considering how it behaves when acting on a measurement made by a packet pair destined for nodes 6 and 7 in the example network. The message passing algorithm can be divided into two stages. In the upward stage, starting at the leaves, information is passed via messages from node to node until the root is reached. In the downward stage, information from the root is passed via messages from node to node until the leaves are reached. Individual nodes then combine the upward and downward messages they received to generate marginal pmfs for their values.

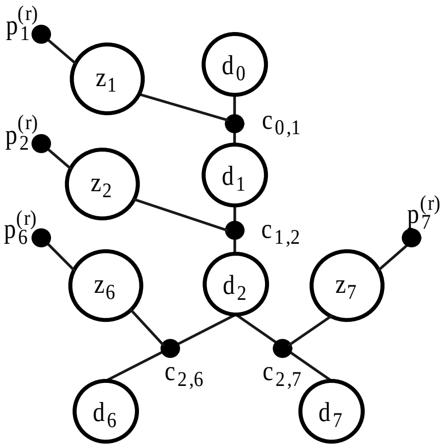


Figure 4: The factor graph used in the message-passing algorithm for a measurement made by a packet pair sent to nodes 6 and 7 in the network of Fig. 2. Measurements are available at nodes 6 and 7; the nodes $p_i^{(r)}$ contain current pmf estimates; and node $c_{a,b}$ indicates the convolutional relationship between nodes d_a, d_b and z_b.

At a leaf node (d_6 or d_7) in Figure 4, the upward message is simply a delay pmf that has a one in the bin of the delay measurement being processed and zeros everywhere else. The upward message from z_6 is the previous pmf estimate for link 6. At node $c_{2,6}$ this message is convolved with the message from the leaf node d_6, and the result is passed up to the branching point d_2. A similar process occurs from leaf node 7. At node d_2, the upward messages from the two lower branches are multiplied together and the resultant message is passed up. The convolution procedure continues up the shared branch until the root node is reached. In the downward stage, the initial message from the root contains the information that the delay at the root is zero: it is a delay pmf with one in the zero bin and zeros elsewhere. Messages are passed down, with convolution exactly as before. At the branching node d_2, the message passed down to node $c_{2,6}$ is the product of the downward message from $c_{1,2}$ and the upward message from $c_{2,7}$. At the end of the two stages, the each node z_i multiplies the upward message, the downward message, and its distribution from the previous EM iteration to obtain $p_i^{(r+1)}(z_i = j | y_1(k), y_2(k))$.

A straightforward implementation of this message passing procedure, as first proposed in [4], has a computational complexity of $O(LK^2)$ *per measurement and iteration of EM*, where L is the maximum path length in the network and K is the number of bins. Recall that K is the smallest power of two greater than or equal to N. For each measurement, the act of passing a message within the algorithm involves the evaluation of a number of summations, which can be cast as convolutions. These convolutions involve vectors of maximum length LK, where L is the maximum path length in the network. Implementation of the convolutions in the Fourier domain reduces the computational complexity from $O(LK^2)$ to $O(LK \log K)$ *per measurement and iteration of EM.* This reduction can be substantial when N (and hence K) is reasonably large.

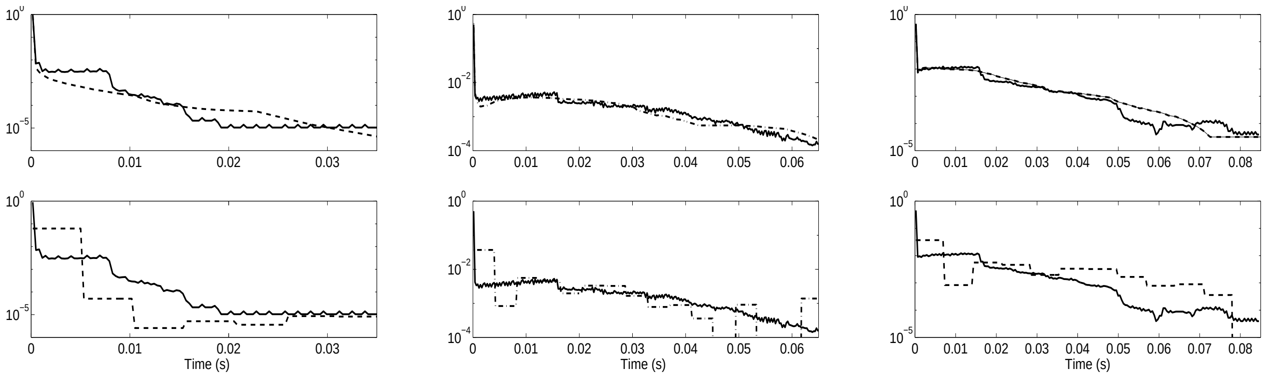
4 Simulation Experiments

In order to verify the performance of our estimation methodology, we conducted `ns-2` [19] simulation experiments using the network depicted in Fig. 2. Interior links in the network have higher capacity (5-10 Mb/sec) and propagation delay (50 ms) than the edge links (0.5-2 Mb/sec and 10 ms). Queues are FIFO (droptail) with space for 35 packets. Node 0 generates a 19.2 Kbit/s probing stream comprised of UDP packet-pair probes (60 bytes each). Packet-pair sending times are generated according to a Poisson process; the mean time-spacing is 50 ms. The probe-stream requires less than 1% of any link's capacity. Background traffic is composed of a mixture of long-lived data-source TCP (FTP) connections, exponential on-off sources using UDP, and multiple short-duration TCP connections. Averaged over the simulations, link utilization ranges between 10 and 60 percent, and loss rates ranged from 0 to 2 percent; typical values for certain real networks.

The network was simulated for multiple two minute measurement periods; from within each measurement period, 25 seconds (inference period) was isolated for analysis. This time duration corresponds to 500 packet-pairs (assuming no probes are lost). Throughout the inference period, queue lengths in the network were determined at a fine time scale by monitoring the arrivals of every packet at each queue. A "true" pmf for each link was formed by calculating delays from queue lengths and link capacities, quantizing and forming a histogram. When generating this true pmf, so much data is available that the quantization can be very fine (constructing an excellent estimate of the delay density) without affecting estimation stability.

In Fig. 5, we show the results of one experiment, comparing the true pmfs to the nonparametric MMPLE estimator and the MLE estimator of [4] using a 16-bin discretized pmf (16 bins was found to give the best performance among unpenalized estimators; see discussion below). We display results for the lower bandwidth links because for our experimental set-up, queuing delay was concentrated in these links. We display the results of representative links that provide a meaningful indication of performance. There is substantial mass in the tails of these pmfs and we can evaluate how well the pmf estimates generated by our proposed methodology estimates match the tails; network performance hinges critically on the tail probabilities of queues [41, 42]. In the higher bandwidth links, there is much less mass in the pmf tails 8(a). For these links, both the MLE and MMPLE estimates match the true pmf where probability mass is concentrated, but there is insufficient information to closely match the tails. We calculated the MLE for a variety of bin sizes, but show the bin size that achieved the best fit to the true pmf (in this case 16 bins). The nonparametric estimator was calculated from $K = 512$ bins.

In Fig. 6, we plot the magnitude of the L_1 error norm between the true pmf and the MMPLE for the links in the network, as averaged over 25 simulations. Also shown are the results for the MLE for medium (64 bins) and large (16 bins) bin sizes. The L_1 error norm is simply the sum of the absolute difference between the estimated pmf and the true pmf over the K bins. As discussed in [14, 43], the L_1 error criterion is a common measure of the performance of a density estimate. The advantage of such as a measure as opposed to a mean-squared error criterion is that more



(a) (b) (c)

Figure 5: Comparison between true pmfs (solid) and estimated pmfs (dashed). Top panel shows true pmf and MMPLE (calculated using 512 bins); bottom panel shows true pmf and MLE (calculated using 16 bins). 16 bins is determined as the bin size at which the MLE obtains the best fit. (a) Link 5. (b) Link 7. (c) Link 9.

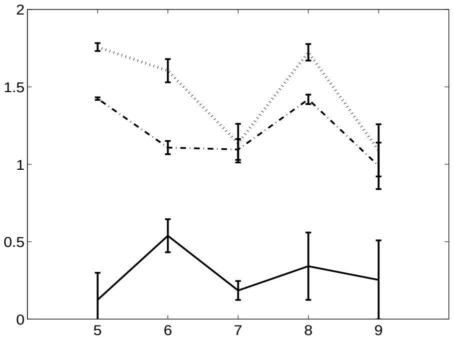


Figure 6: L_1 error criterion averaged over 25 simulations (means and standard deviation) for link 5, 7 and 9. Solid line is MMPLE, dashed line is MLE (16 bins), dotted line MLE (64 bins).

attention is paid to the tails of the distributions. It also enjoys several theoretical advantages over other measures [43].

As is evident from the two figures, the MMPLE technique generates estimates which are smooth, close fits to the true pmfs. In order to introduce some degree of smoothness, MLE estimates must be calculated using a large bin size, resulting in an inability to capture the finer details of a pmf.

In order to illustrate the performance of the algorithm in a larger network, we also simulate a 20-receivers scenario as shown in Fig. 7. The packet probing rate from the source, as well as the composition of background traffic remains the same as in the first scenario. The link loss rates range from 0 to 2 percent, and the link utilization varies between 0 and 60 percent, averaged over 20 simulations. We use the same inference window of 25 seconds. If we assume there is no packet loss, then there are a total of 500 packet pairs. However as the number of total measurements remains unchanged while the number of receivers increases, the number of measurements obtained for each link reduces. In Fig. 8 and in Fig. 9, we show the results and performance of the algorithm. Fig. 10 compares the delay cumulative distribution function (cdf) obtained by estimation based on direct measurement with the delay cdf estimated using the MMPLE technique and the probe measurements for a representative link in the network.

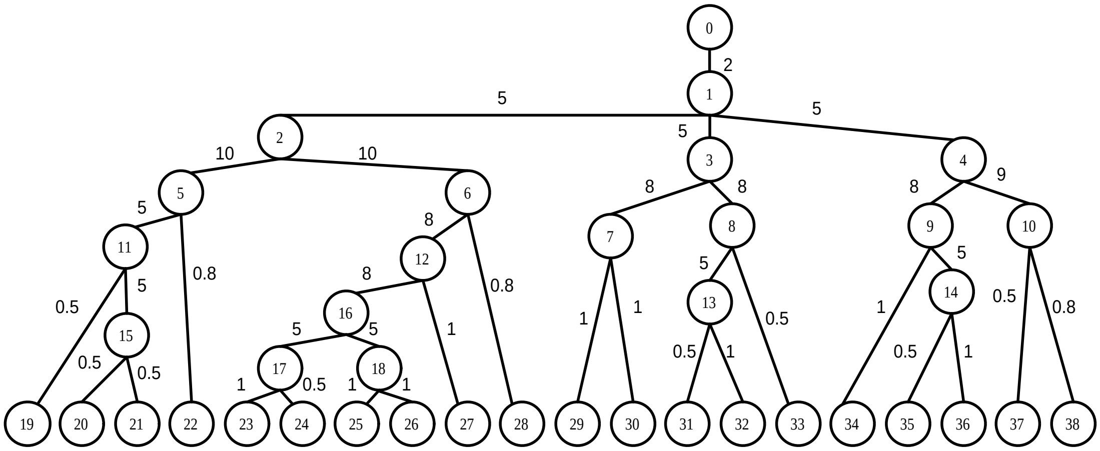
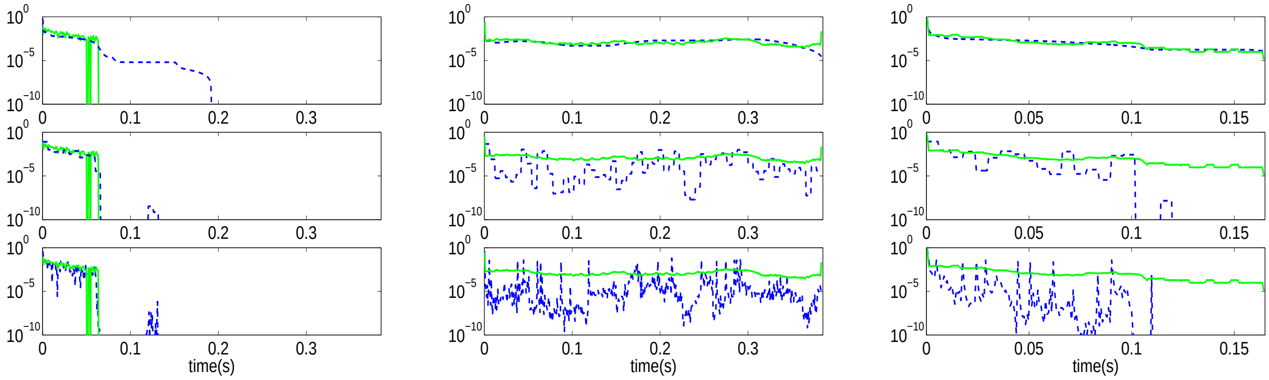


Figure 7: A larger tree-structured network topology used for ns-2 simulation experiments. Source (node 0) transmits to 20 receivers (nodes 19-38). Link speeds in Mb/s are shown next to the links.



(a) (b) (c)

Figure 8: Comparison between true pmfs (solid) and estimated pmfs (dashed). Top panel shows true pmf and MMPLE (calculated using 512 bins); middle panel shows true pmf and MLE (calculated using 64 bins); bottom panel shows true pmf and MLE (calculated using 512 bins). 512 bins is determined as the bin size at which the MLE obtains the best fit. (a) Link 1. (b) Link 20. (c) Link 31.

When the amount of probing that can be performed is limited, we believe that the most substantial source of error is the intrinsic variability in probe measurements. Another potential source of error is the discrepancy between the delays experienced by the two packets in each pair on their common path. We therefore examined the extent and effect of the delay discrepancy; with 512 bins, the overwhelming majority of the discrepancy was concentrated in 0-3 bins, with a maximum value of 16 bins. The effect of these discrepancies on the quality of the estimates is relatively minor when such a small amount of data is available for inference. If we directly measure the delays experienced by probes on each link (which can be done in our simulation), the estimates we obtain

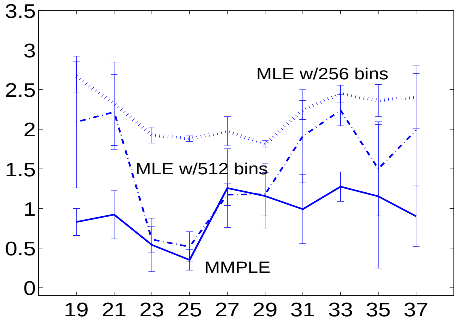


Figure 9: L_1 error criterion averaged over 20 simulations (means and standard deviation) for some terminating links. Solid line is MMPLE, dash-dot line is MLE (512 bins), and dotted line is MLE (256 bins).

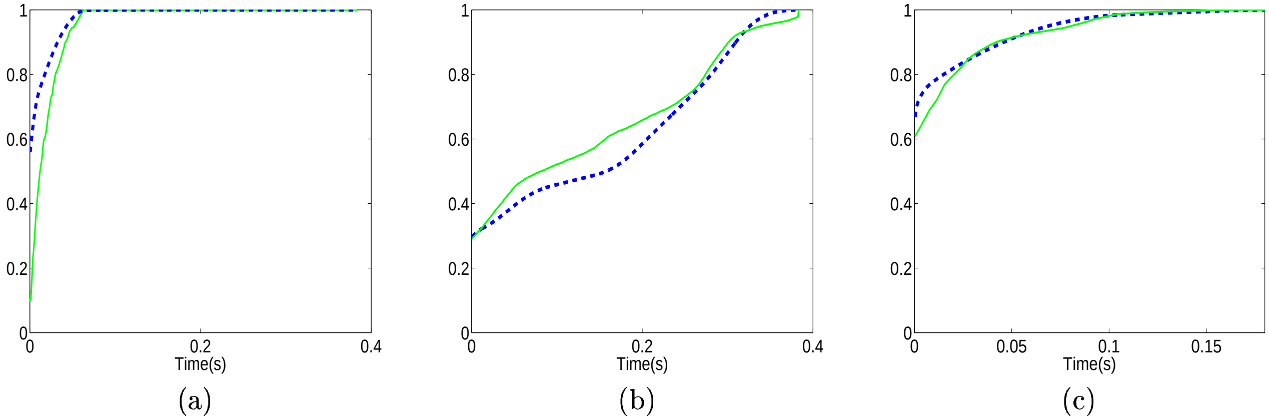


Figure 10: The cdf estimates obtained from direct measurement (solid) to the tomographic one (dotted). (a) Link 1. (b) Link 20. (c) Link 31.

are very similar to those obtained by our tomographic procedure.

5 Conclusions

In this paper, we introduce a new nonparametric methodology for network delay tomography based on unicast end-to-end measurement. Our approach takes advantage of the correlation between the delay experienced by back-to-back packet pairs. We pose the network tomography problem as a maximum penalized likelihood estimation, and develop a fast Fourier Transform based EM algorithm for computing our estimates. The complexity is reduced to $O(MN^2 \log N)$, where M is the number of links in the tree and N is the number of probes.

One of the key features of the framework are its flexibility (the ability to capture fine details and smooth regions) and the introduction of a complexity penalization that allows smooth, accurate estimates to be generated even when the amount of data is very small. The basic MMPLE framework developed here could be extended to the multicast approach suggested in [6] and may also be applicable in time-varying contexts like those considered in [4, 5]. We demonstrate the accuracy of the estimation procedure using network-level simulator `ns-2`.

References

[1] F. Lo Presti, N.G. Duffield, J. Horowitz, and D. Towsley, "Multicast-based inference of network-internal delay distributions," Tech. Rep., University of Massachusetts, 1999.

[2] R. Cáceres, N. Duffield, J. Horowitz, and D. Towsley, "Multicast-based inference of network-internal loss characteristics," *IEEE Trans. Info. Theory*, vol. 45, no. 7, pp. 2462–2480, November 1999.

[3] M. Coates and R. Nowak, "Network loss inference using unicast end-to-end measurement," in *ITC Seminar on IP Traffic, Measurement and Modelling*, Monterey, CA, Sep. 2000.

[4] M. Coates and R. Nowak, "Network delay distribution inference from end-to-end unicast measurement," in *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Proc.*, May 2001.

[5] M. Coates and R. Nowak, "Sequential Monte Carlo inference of internal delays in nonstationary communication networks," in *IEEE Trans. Signal Processing, Special Issue on Monte Carlo Methods for Statistical Signal Processing*, 2001.

[6] N.G. Duffield, F. Lo Presti, V. Paxson, and D. Towsley, "Inferring link loss using striped unicast probes," in *Proceedings of IEEE INFOCOM 2001*, Anchorage, Alaska, April 2001.

[7] K. Harfoush A. Bestavros and J. Byers, "Robust identification of shared losses using end-to-end unicast probes," in *Proc. IEEE Int. Conf. Network Protocols*, Osaka, Japan, Nov. 2000, *Errata* available as Boston University CS Technical Report 2001-001.

[8] K. Lai and M. Baker, "Measuring link bandwidths using a deterministic model of packet delay," in *Proc. ACM SIGCOMM 2000*, Stockholm, Sweden, Aug. 2000.

[9] S. Ratnasamy and S. McCanne, "Inference of multicast routing trees and bottleneck bandwidths using end-to-end measurements," in *Proceedings of IEEE INFOCOM 1999*, New York, NY, March 1999.

[10] D. Rubenstein, J. Kurose, and D. Towsley, "Detecting shared congestion of flows via end-to-end measurement," in *Proc. ACM SIGMETRICS 2000*, Santa Clara, CA, June 2000.

[11] M. Coates, A. Hero, R. Nowak, and B. Yu, "Internet tomography," *IEEE Signal Processing Magazine*, May 2002.

[12] J. Kurose and K. Ross, *Computer Networking*, Addison Wesley, 2001.

[13] *Netdyn*, http://www.cs.umd.edu/projects/netcalliper/NetDyn.html.

[14] D.W. Scott, *Multivariate Density Estimation: Theory, Practice and Visualization*, Wiley, New York, 1992.

[15] M.F. Shih and A.O. Hero, "Unicast-based inference of network link delay distributions using mixed finite mixture models," in *IEEE Int. Conf. Acoust., Speech, and Signal Proc.*, Orlando, Florida, May 2002.

[16] N.G. Duffield, J. Horowitz, F. Lo Presti, and D. Towsley, "Network delay tomography from end-to-end unicast measurements," in *Proc. of the 2001 International Workshop on Digital Communications 2001-Evoluntionary Trends of the Internet*, Taormina, Italy, Sept. 2001.

[17] R. Nowak and E. Kolaczyk, "Multiscale maximum penalized likelihood estimators," in *Proceedings. 2002 IEEE International Symposium on Information Theory*, 2002, p. 156.

[18] E. Kolaczyk and R. Nowak, "Multiscale likelihood analysis and complexity penalized estimation," submitted to *A*nnals of Statistics, 2000.

[19] "UCB/LBNL/VINT network simulator ns (version 2)," URL: http://www.isi.edu/nsnam/ns/.

[20] N. Duffield and F. Lo Presti, "Multicast inference of packet delay variance at interior network links," in *Proceedings of IEEE INFOCOM 2000*, Tel Aviv, Israel, Mar. 2000.

[21] V. Jacobson, "pathchar," 1997, ftp://ftp.ee.lbl.gov/pathchar/msri-talk.ps.gz.

[22] M.F. Shih and A.O. Hero, "Unicast inference of network link delay distributions from edge measurements," in *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Proc.*, May 2001.

[23] M.F. Shih and A.O. Hero, "Unicast inference of network link delay distributions from edge measurements," Tech. Rep., Comm. and Sig. Proc. Lab. (CSPL), Dept. EECS, University of Michigan, Ann Arbor, May 2001.

[24] K. Anagnostakis and M. Greenwald, "On the feasibility of network delay tomography using only existing infrastructure," in *Tech. Rep., MS-CIS-02-XX, CIS Department, University of Pennsylvania*, July 2002.

[25] K. Anagnostakis and M. Greenwald, "Direct measurement versus indirect inference for determining network-internal delays," in *Proceedings of Performance*, Rome, Italy, Sept. 2002.

[26] J. Postel, "Internet control message protocol," Sept. 1981, IETF Internet Request For Comments: RFC 792.

[27] Y. Tsang, M. Coates, and R. Nowak, "Nonparametric internet tomography," in *IEEE Int. Conf. Acoust., Speech, and Signal Proc.*, Orlando, Florida, May 2002.

[28] J. Besag, "On the statistical analysis of dirty pictures," *J. Roy. Stat. Soc. B*, vol. 48, pp. 259–302, 1986.

[29] V. Jacobson, "traceroute," 1989, available as ftp://ftp.ee.lbl.gov/traceroute.tar.Z.

[30] M. Coates, M. Gadhiok, R. King, and R. Nowak, "`nettomo`: A tool for unicast network tomography," Tech. Rep. TREE-05, Rice University, Jun. 2001.

[31] M. Coates, R. Castro, Y. Tsang, and R. Nowak, "Maximum likelihood network topology identification from edge-based unicast measurements," in *ACM SIGMETRICS*, Marina Del Rey, Los Angeles, June 2002.

[32] A. Pásztor and D. Veitch, "Precision based precision timing without gps," in *ACM SIGMETRICS*, Marina Del Rey, Los Angeles, June 2002.

[33] S. Moon, P. Skelly, and D. Towsley, "Estimation and removal of clock skew from network delay measurements," in *Proc. IEEE Infocom*, 1999.

[34] V. Paxson, "On calibrating measurements of packet transit times," in *Proc. ACM SIGMETRICS 1998*, Madison, 1998.

[35] P.J. Green, "Penalized likelihood," *Encyclopedia of Statistical Sciences, update volume*, vol. 3, pp. 578–586, 1999.

[36] G. McLachlan and T. Krishnan, *The EM Algorithm and Extensions*, Wiley, New York, 1997.

[37] R. Willett and R. Nowak, "Multiresolution nonparametric intensity and density estimation," in *IEEE Int. Conf. Acoust., Speech, and Signal Proc.*, Orlando, Florida, May 2002.

[38] B. Frey, *Graphical Models for Machine Learning and Digital Communication*, MIT Press, Cambridge, 1998.

[39] J. Pearl, "Fusion, propagation, and structuring in belief networks," *Artificial Intelligence*, vol. 29, pp. 245–257, 1986.

[40] D.J. Spiegelhalter, "Probabilistic reasoning in predictive expert systems," in *Uncertainty in Artificial Intelligence*, L.N. Kanal and J.F. Lemmer, Eds., pp. 47–68. North Holland, Amsterdam, 1986.

[41] V. Ribeiro, R. Riedi, M. Crouse, and R. Baraniuk, "Multiscale queuing analysis of long-range-dependent network traffic," in *Proceedings of IEEE INFOCOM*, Tel Aviv, Israel, March 2000.

[42] A. Erramilli, O. Narayan, and W. Willinger, "Experimental queueing analysis with long-range dependent traffic," *IEEE/ACM Transactions on Networking*, vol. 4, no. 2, pp. 209–223, April 1996.

[43] L. Devroye and G. Lugosi, *Combinatorial Methods in Density Estimation*, Springer-Verlag, New York, 2001.