Unicast-Based Inference of Network Link Delay Distributions with Finite Mixture Models (Revision)

Meng-Fu Shih, Alfred O Hero

Submitted to IEEE Transactions on Signal Processing,
Special Issue on Signal Processing in Networking - Aug. 2003.

October 2002

Corresponding author: Meng-Fu Shih,
Dept of EECS,
Univ. of Michigan,
Ann Arbor, MI 48109-2122
734-763-0564 (Tel)
`mfshih@umich.edu`

Abstract

Providers of high quality-of-service over telecommunication networks require accurate methods for remote measurement of link-level performance. Recent research in network tomography has demonstrated that it is possible to estimate internal link characteristics, e.g., link delays and packet losses, using unicast probing schemes in which probes are exchanged between several pairs of sites in the network. In this paper we present a new method for estimation of internal link delay distributions using the end-to-end packet delay statistics gathered by such unicast probes. Our method is based on a variant of the penalized maximum likelihood expectation-maximization (PML-EM) algorithm applied to an additive finite mixture model for the link delay probability density functions. The mixture model incorporates a combination of discrete and continuous components and we use a minimum message length (MML) penalty for selection of model order. We present results of `matlab` and `ns-2` simulations to illustrate the promise of our network tomography algorithm for light cross-traffic scenarios.

Keywords: network tomography, additive-convolutive mixture models, EM algorithm, MML penalties, signal processing in networking.

1 Introduction

Herein we address the problem of determining internal link-delay distributions from multiple end-to-end unicast packet probes that are sent across the network from many different edge node pairs. This is a sub-problem of network tomography [1, 2] which is concerned with reconstruction of unobserved states of a network from a set of indirect measurements of these states. For example, end-to-end network tomography allows a few agents at the edge of a network to gain important information on global network behavior without cooperation of internal nodes. This is especially useful when link parameters are inaccessible or when direct measurement of data traffic statistics are not supported by internal switches and routers [1, 2, 3, 4, 5]. The unicast tomography methods we present in this paper are applicable to this situation and are designed to perform well when the network is lightly loaded. While modification of our methods may also be applicable to wireless networks, here we focus on the simpler case of wired networks, such as the Internet.

The causes of delays along a packet probe's path through the network can be separated into the sum of two types of delays: constant link transmission delays and time-varying link processing delays. Link transmission delays are due to the propagation delays through the physical medium, e.g., a wire or optical fiber. Link processing delays are due to a combination of router queueing, buffering and servicing delays that depend on factors such as: the amount of cross-traffic at the router, the number of retransmits required over the link, and the integrity of router equipment and associated software. While transmission delays usually remain constant over a probing interval, processing delays are highly variable and are thus commonly modelled as random variables. Thus it is generally impossible to recover the actual internal link delays that probes encounter along their end-to-end path. However, the determination of the statistical distribution of the internal link delays from multiple end-to-end delay measurements can be formulated as a statistical inverse problem whose solution yields estimates of the internal delay distribution [2, 6]. These estimates can be used by an autonomous system (AS), e.g. an Internet service provider (ISP), to evaluate its average quality of service (QoS) or to assess link performance of other, perhaps competing, AS's. When acquired over large portions of the network, link delay estimates can also be used for detecting network anomalies such as imminent link failures or coordinated denial of service (DoS) attacks.

The problem of empirically characterizing Internet link delay distributions has been looked at by several groups, see for example [7, 8, 9, 10]. A common observation is that when the link is lightly loaded, such as in the early morning, link delay scatterplots appear stationary. Furthermore, while much of the scatter appears spread out over a continuum of delay values, a non-negligable proportion of the delays appear to concentrate at one or more discrete values, see for example [10, Fig. 4]. This implies the existence of point masses in the time-averaged link delay distribution. The positions of these point masses vary according to factors such as: length of packet; incoming and outgoing queue sizes of routers on the link; router configuration; deployment of firewalls; and the physical distance between routers [10].

In this paper we propose to capture these empirically observed features by fitting hybrid continuous/discrete finite mixture models to the link delay distributions. While our algorithms are easily generalizable to multiple discrete point masses, for simplicity we focus here on the case where the discrete component is a single point mass. Unlike purely continuous models the hybrid continuous/discrete model is identifiable and is justified for the lightly loaded scenario. In this scenario there is a non-zero probability that a packet will encounter an empty queue in which case the packet delay is non-random, being due to fixed propagation and processing delays. While this is unlikely in a congested network, the model is valid for a number of common monitoring situations such as service and performance verification and detection of onset congestion. Moreover, we would like to point out that the delay point mass is implicit in canonical delay trees, used in discrete delay tomography, for which there is a non-zero probability that a packet traverses each link without any delay (see, e.g., [6]).

Lo Presti *et. al* [6] were the first to propose estimating internal delay distributions from end-to-end measurements. These authors uniformly discretized the internal delay distributions and derived an algorithm based on empirical histogram estimation. Their method uses multicast probes, which requires cooperation of the network to run a multicast session such as RTP during the probing interval. To overcome this restriction, Coates and Nowak [5] developed an internal delay histogram estimator based on an alternative unicast probing scheme in which edge sites exchange a succession of closely spaced packet pairs. Their estimator is based on a statistical inverse problem formulation and used an iterative maximum likelihood via expectation maximization (ML-EM) approach. In related work these authors also developed a sequential Monte-Carlo method for tracking changes in non-stationary networks [11]. The principal restriction of the approaches [5, 6, 11] is the requirement of discrete link delays. Overly coarse discretization, or binning, of the link delays leads to excessive model approximation error and causes bias in derived estimates such as delay mean and variance. At the opposite extreme, excessively fine discretization leads to high runtime complexity of these algorithms. Furthermore, the determination of the appropriate number and size of the bins requires tight bounds on link delay characteristics which are usually unknown such as maximum and minimum processing delay.

Several alternatives to the fixed and uniform binning scheme of Lo Presti *et. al* [6] have been studied. Duffield *et. al* considered a variable bin size model, where smaller bins are used to describe probability mass concentrations for small delays [12]. Tseng, Coates, and Nowak [13] proposed a nonparametric algorithm where the number of bins for internal link delays is adapted to the number of measurements. They use a wavelet-based penalized maximum likelihood estimator to smooth the estimates. A non-parameteric method was proposed for unicast probing by Shih and Hero [14] in which the statistical inverse problem is formulated in the cumulant generating function (CGF) domain. By sampling the CGF's arising from the least squares solution of the inverse problem, a set of continuous (un-binned) link delay density estimates can be obtained.

Herein we propose a new method for estimation of internal delay distributions from unicast end-to-end measurements which is based on additive mixture models for the internal link delays. As the end-to-end delay measurements sums the (assumed independent) internal link delays over the probe path, the densities of the measurements are convolutive mixtures of these additive mixture models. This makes our estimation problem more challenging than the usual standard mixture model estimation problem which has received much attention in both the statistical and engineering literature. Additional issues which we address are: 1) the additive mixture model orders are unknown in practice; and 2) the internal link delay distributions are composed of a combination of continuous and a discrete components. We handle the convolutive mixture complication by adopting an iterative ML-EM formulation of the estimation problem using an enlarged complete data space. We handle the problem of unknown model order by adapting the unsupervised minimum-message-length (MML) approach used in Figueiredo and Jain [15]. Specifically, we add an information theoretic order selection penalty to the log-likelihood to which a penalized ML-EM (PML-EM) algorithm is applied. We handle the presence of both discrete and continuous link delay components by the following simple additive mixture model: the delay density is a (unknown) convex combination of a point mass positioned at the (unknown) transmission delay and a (unknown) number of Gaussian components with (unknown) means and variances. We adopted Gaussian continuous components to simplify the implementation but heavy-tailed densities can also be easily accommodated in our framework.

The outline of the paper is as follows. In Sec. 2 we give the main assumptions underlying our work. In Sec. 3 we review the continuous and discrete delay models for unicast network delay tomography and discuss model identifiability. In Sec. 4 we introduce the hybrid mixture models for delays of the internal links and present the ML-EM algorithm for estimating the parameters of these models from end-to-end delay measurements. In Sec. 4.3 we present the PML-EM algorithm with MML penalty to control excessive order estimation of the unpenalized ML-EM algorithm. In Sec. 5 we illustrate the performance of the ML-EM and PML-EM algorithms on simulated data using `matlab` and `ns-2` [16] simulators.

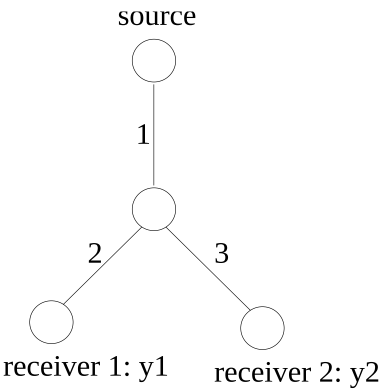


Figure 1: Two-leaf network.

2 Network Model and Main Assumptions

As in [5] we represent the network topology as a directed logical tree $T = (V, E)$ where V is the set of nodes, e.g. routers and terminals, and E is the set of links. Let there be a total of L links in the network and number them from 1 to L. The logical tree representation has a single root node, serving as a source, several internal nodes having degree at least 2, and several leaf nodes, containing receivers. If there are a total of R leaf nodes then there are R possible paths on which the source can send packets to the receivers. The edge of the network is the set of leaf nodes plus the root node. Each node in the tree has two or more child nodes, except the root and leaf nodes. A network with three links is shown in Fig. 1 showing a single root node, a single internal node, and two leaf nodes.

In a unicast probing session a leaf node is (randomly) selected by the source and a time stamped packet, called a (unicast) probe, is sent to it. The leaf node records the time the packet is received. Subtracting this number from the packet's time stamp gives the end-to-end delay of the packet. End-to-end delays on the same probe path are random variables due to the random ambient cross traffic through links along the path. Unicast probing is repeated until the session is over or enough packets are received by each leaf node to perform the next step: network delay tomography. The aim of network delay tomography is to identify the packet delay distribution for each individual internal link from the end-to-end delays observed by the receivers. Network tomography is possible since the end-to-end delay is a sum of the internal link delays encountered along the probe path and all of the R paths cross at common links.

Let X_l be the packet delay encountered by a probe at link l, $l = 1, \ldots, L$, and let Y_i be the end-to-end packet delay along the i-th path, $i = 1, \ldots, R$. We make the following independence and stationarity assumptions: (A1) *Spatial Independence*: packet delays at different links are statistically independent, i.e., X_i and X_j are independent for $i \neq j$. (A2) *Temporal Independence and Stationarity*: for a given link, the delays encountered by different packets at that link are statistically independent and identically distributed (i.i.d.). It is important to point out that while (A1) and (A2) are normally not satisfied in practice (see, e.g., [17]), these are commonly assumed in order to permit tractable analysis. An example where spatial independence (A1) is violated is when there is interaction among different data flows along the same path. In (A2) temporal independence fails when Internet traffic is bursty or the network has a long latency time which correlates different probe delays. Stationarity fails when the unicast probing session is long relative to the stationarity time of the network. However, experiments have shown that the performance of network tomography is remarkably insensitive to violations of (A1) and (A2) [4, 5, 6, 11, 18].

3 Unicast Network Delay Tomography

3.1 Discrete Delay Model

In the widely adopted discrete link delay model [5, 6, 11] a universal bin size q is used to discretize link delays X_l at each link $l = 1, \ldots, L$. The time intervals $(iq, (i+1)q]$, $i = 0, \ldots, D$, are called the delay bins. Here D is a positive integer and $D = \infty$ can be used to account for lost probe packets or large delays which are out of range. Discretization produces the discretized delay value i when X_l falls in the i-th bin. A probability mass function (pmf), or histogram, $P_l = \{p_{l,d} : d = 0, \ldots, D\}$ is then associated with the discretized delays over link l, where the probability $p_{l,d} = P(X_l \in (dq, (d+1)q])$ is an unknown to be estimated and $\sum_{d=0}^{D} p_{l,d} = 1$. For a probe path containing j links, the discretized end-to-end packet delay varies over the range $0, \ldots, j \cdot D \cdot q$.

The identifiability of the link delay pmf's from end-to-end delays is easily studied for the simple case of the two-leaf tree network shown in Fig.1 when we discretize to only two delay bins ($D = 1$). We say that the link pmf's are not identifiable if the Fisher information matrix (FIM) for the unknown parameters, i.e. the pmf's P_1, P_2 and P_3, is singular. This is a local measure of identifiability which is necessary, but not always sufficient for global identifiability [19].

Let the true delay pmf's for link 1 to 3 be $P_1 = \{p_1, 1 - p_1\}$, $P_2 = \{p_2, 1 - p_2\}$, and $P_3 = \{p_3, 1 - p_3\}$, respectively. Define the parameter vector $\theta = [p_1, p_2, p_3]^T$. Let $\mathbf{Y}_1 = \{Y_1^1, \ldots, Y_1^n\}$ and $\mathbf{Y}_2 = \{Y_2^1, \ldots, Y_2^n\}$ be the sets of end-to-end packet delays computed at receivers 1 and 2, respectively, after reception of n unicast probes. The joint pmf $P(\mathbf{Y}_1, \mathbf{Y}_2;\ \theta)$ can be simply expressed as the product of two binomial pmfs. The FIM $\mathbf{A}$ is obtained from the expectation of the Hessian matrix whose elements are the second order partials of the log-likelihood function: $\mathbf{A} = -E[\nabla_\theta^2 \log P(\mathbf{Y}_1, \mathbf{Y}_2;\ \theta)]$. The FIM is easily computed and is given below in

(1), where $Q_1 = p_1(1-p_2) + p_2(1-p_1)$, and $Q_2 = p_1(1-p_3) + p_3(1-p_1)$.

$$\mathbf{A}(p_1, p_2, p_3) = \\ n \begin{pmatrix} \frac{p_2+p_3}{p_1} + \frac{2-p_2-p_3}{1-p_1} + \frac{(1-2p_2)^2}{Q_1} + \frac{(1-2p_3)^2}{Q_2} & \frac{1}{Q_1} & \frac{1}{Q_2} \\ \frac{1}{Q_1} & \frac{p_1}{p_2} + \frac{1-p_1}{1-p_2} + \frac{(1-2p_1)^2}{Q_1} & 0 \\ \frac{1}{Q_2} & 0 & \frac{p_1}{p_3} + \frac{1-p_1}{1-p_3} + \frac{(1-2p_1)^2}{Q_2} \end{pmatrix}. \quad (1)$$

This is non-singular except when $p_1 = p_2 = p_3$.

The discrete delay model has two main drawbacks. First, the proper bin size needs to be carefully selected. Second, a universal bin size may not be suitable due to large variation of packet delay ranges over different links. Although in [6] it was proposed to adopt different bin sizes for different links, those bin sizes still need to be chosen in advance.

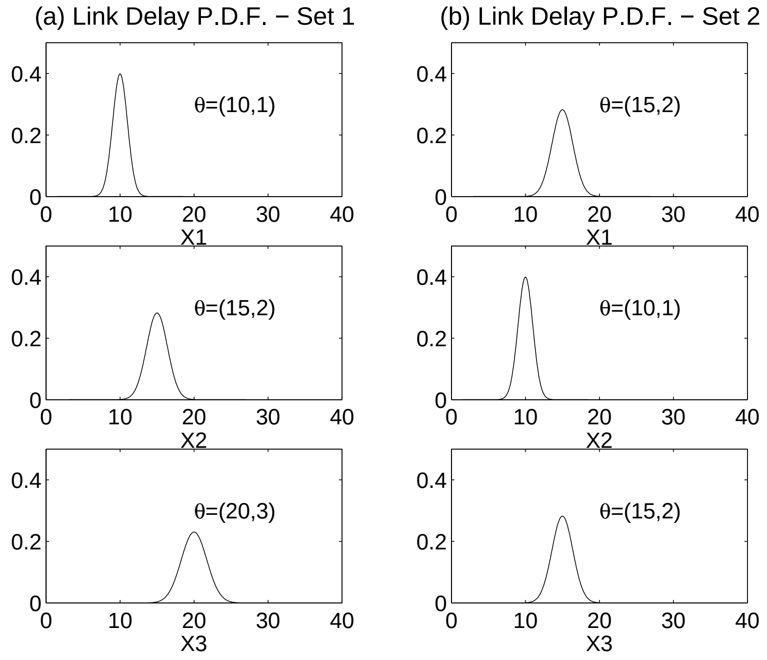
3.2 Continuous Delay Model

One way to avoid the pitfalls of binning is to use a flexible continuous link delay model. For example, closed form expressions for the probability density function (pdf) of queueing delay have been derived for simple queueing models such as M/M/1. These expressions could possibly be extended to a network of queues but it is well known that the M/M/1 model is an inadequate model for Internet traffic [20]. An alternative is to approximate each link delay density by a finite mixture which, with sufficiently large number of components, can describe any continuous density function [21]. Let $f_l(x)$ be the link delay pdf at link l. A finite mixture model for this pdf is

$$f_l(x) = \sum_{m=1}^{k_l} \alpha_{l,m} \phi(x; \theta_{l,m}), \quad l = 1, \ldots, L \tag{2}$$

where k_l denotes the number of components, $\alpha_{l,m}$, $m = 1, \ldots, k_l$, denotes the mixing parameter for the m-th component $\left(0 < \alpha_{l,m} < 1, \sum_{m=1}^{k_l} \alpha_{l,m} = 1\right)$, and $\phi(x; \theta_{l,m})$ is a density function over the x-domain parameterized by the parameter vector $\theta_{l,m}$. Many different choices for $\phi(x; \theta)$ are possible including: Gaussian; generalized Gaussian; exponential; or uniform. For the case of a Gaussian mixture $\theta_{l,m} = [\mu_{l,m}, \sigma^2_{l,m}]$ is the vector specifying the position (mean) and width (sqrt(variance)) of the m-th mixture component at the l-th link.

However, the use of pure continuous mixture density functions can cause more serious identifiability problems than the purely discrete model discussed in the previous subsection. To illustrate consider again the simple two-leaf tree of Fig. 1. Assume that all link delays are Gaussian, i.e., $k_1, k_2, k_3 = 1$ (single component mixtures) $\phi(x; \theta) = \exp(-(x-\mu)^2/(2\sigma^2))/(\sqrt{2\pi}\sigma)$. The end-to-end delays Y_1 and Y_2 are also Gaussian, as they are sums of independent Gaussian random variables, with joint pdf $f(Y_1, Y_2) = \phi(Y_1;\ \mu_1 + \mu_2, \sigma_1^2 + \sigma_2^2)\ \phi(Y_2;\ \mu_1 + \mu_3, \sigma_1^2 + \sigma_3^2)$. This joint pdf is completely described by the 4 parameters $E[Y_1] = \mu_1 + \mu_2$,


(a) Link Delay P.D.F. − Set 1

(b) Link Delay P.D.F. − Set 2

(c) End−to−End Delay P.D.F.

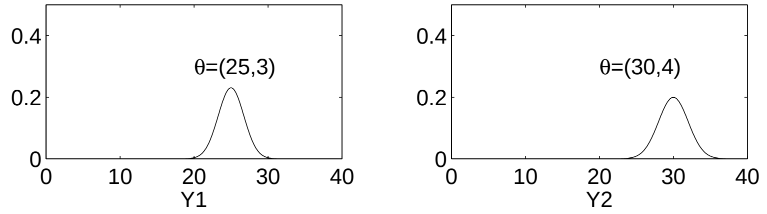


Figure 2: Example of two sets (a) and (b) of Gaussian internal link delay densities along the two probe paths in the network in Fig. 1. The Gaussian end-to-end delay densities shown in (c) are parameterized by only 4 location and width parameters which is insufficient to recover the 6 location and width parameters in (a) and (b).

$E[Y_2] = \mu_1 + \mu_3$, $\mathrm{var}(Y_1) = \sigma_1^2 + \sigma_2^2$ and $\mathrm{var}(Y_2) = \sigma_1^2 + \sigma_3^2$. This gives only 4 equations for the 6 unknowns parameters $\Theta = [\mu_1, \mu_2, \mu_3, \sigma_1^2, \sigma_2^2, \sigma_3^2]$ so the simple Gaussian model is not identifiable for any value of the parameters. An example is shown in Fig. 2 where (a) and (b) are two different sets of internal link delay distributions for the network in Fig.1.

One can also consider the packet-pair/stripe schemes suggested in [5] and [22], in which a pair or a stripe of closely spaced unicast packets with distinct destinations are sent back-to-back from the root node. These packets encounter virtually the same delays on shared links along their paths. As shown in [18], packet-pair probing allows identification of higher order moments of internal link delays when the branching ratio is larger than 2. However, the link delay means cannot be uniquely identified.

4 Hybrid Finite Mixture Approach

4.1 Hybrid Finite Mixture Model

In analysis of a queueing system, the utilization factor ρ is an important parameter for describing system behavior. The parameter ρ denotes the probability that the system is busy serving customers and, for a stable system, ρ must satisfy $0 \leq \rho < 1$ [23]. A lightly loaded link satisfies $\rho << 1$, i.e. there is a non-negligible probability that a packet encounters an empty queue, i.e. an idle router, and passes without delay. This suggests placing a point mass component with weight $1-\rho$ in the link delay mixture model. If this point mass is included in addition to the continuous components the link delay pdf becomes a hybrid discrete/continuous finite mixture model. Hence, similar to (2), we obtain

$$f_l(x) = \alpha_{l,0}\delta(x - x_{l,0}) + \sum_{m=1}^{k_l} \alpha_{l,m}\phi(x;\theta_{l,m}). \tag{3}$$

Here $\alpha_0 = 1 - \rho$, $\delta(x)$ is a point mass (dirac delta function) at zero and $x_{l,0}$ is the pure (non-random) transmission delay experienced by the packet. All other parameters are defined as in (2), except now the α's must satisfy $\sum_{m=0}^{k_l} \alpha_{l,m} = 1$, $\alpha_{l,m} \geq 0$. The discrete mass component $\delta(x)$ not only makes the delay distribution more precisely model the behavior of a link queue, but as shown below also buys us identifiability of the link delay distribution parameters.

For any probe path the distributions of the end-to-end probe packet delays will be the convolution of the link distributions, which are also hybrid mixtures. Now, similarly to the previous section, let's assume that the continuous mixture component is a single Gaussian pdf. Due to the point mass in (3) the end-to-end delay distribution is no longer Gaussian as long as $\alpha_{l,0} \neq 0$. Indeed, the end-to-end delay distribution includes shifted versions of each Gaussian component from which one can extract the parameters $\theta_{l,m}$ by fitting the density translates, see Fig. 3. It might seem strange that the addition of a point mass allows one to uniquely identify the set of parameters of the internal link components from a single probe path. However, one still needs multiple paths to assign these parameters to specific links.

For a general logical tree network topology, if the packets are sent through all the R possible routes, a simple sufficient condition for identifiability of the hybrid link mixtures is that: (1) $\alpha_{l,0} > 0$ for all l; and (2) all the continuous components in each link delay distribution have distinct parameters.

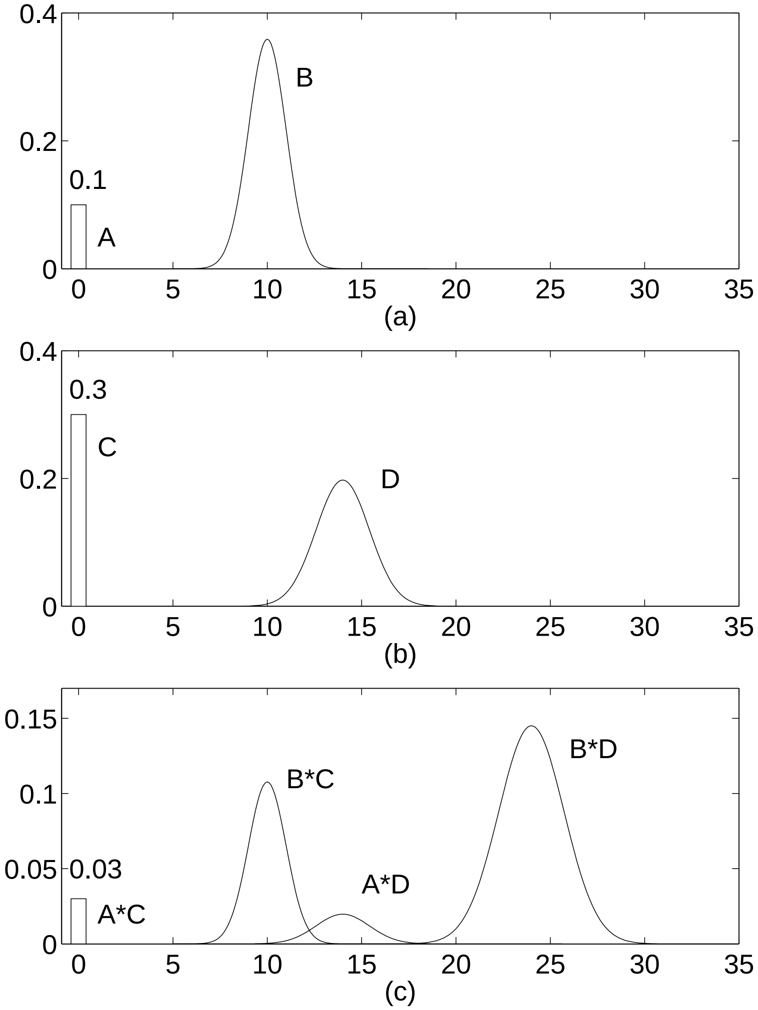


Figure 3: Example of hybrid internal link delay mixtures (a) and (b) over a two link probe path. The non-random minimum delays for both the links are set to 0. End-to-end delay distribution is a hybrid mixture from which all parameters can be identified. Here $B*C$ denotes convolution of the internal link components labelled B and C.

4.2 ML-EM Algorithm

Here we present an ML-EM algorithm for approximating the maximum likelihood estimates of the internal link mixture model parameters from end-to-end measurements. Let U be a finite mixture random variable with k components and pdf of the form $f(U) = \sum_{m=1}^{k} \alpha_m \phi_m(U)$ where $\sum_{m=1}^{k} \alpha_m = 1$. An example of a Gaussian mixture with 3 components is given in Fig.4. The solid line depicts the density function and the dashed line shows each component. There are two different interpretations of finite mixture models which will be useful. The first one is simply that $f(U)$ is a multi-component pdf for U. The second interpretation is that U is selected at random from a pool of hidden random variables $U_1, \ldots, U_k$ with selection probabilities $\alpha_1, \ldots, \alpha_k$, respectively. Define the binary random selection vector $\mathbf{Z} = [Z_1, \ldots, Z_k]^T$ where $Z_m = 1$ if and only if the m-th variable U_m is selected and assign to this event probability α_m. U can be expressed as

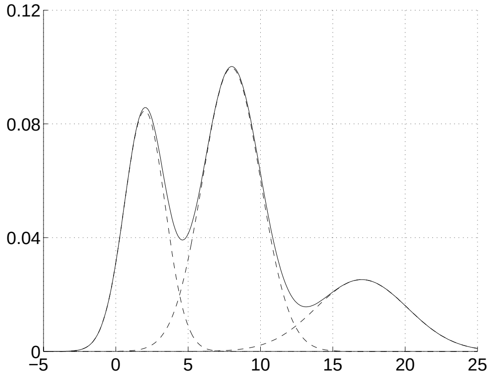


Figure 4: Gaussian mixture example. $f_U(u) = 0.3N(u; 2, 2) + 0.5N(u; 8, 4) + 0.2N(u; 17, 10)$

$U = \sum_{m=1}^{k} Z_m U_m$. Thus, if U_m has pdf $\phi_m(U_m)$ then this is identically the conditional pdf $f(U|Z_m = 1)$. Thus $f(U) = \sum_{m=1}^{k} \alpha_m \phi_m(U)$, which is the mixture model for U that we started out with. The second interpretation is critical for development of the ML-EM algorithm which we address below.

Assume that we have prior knowledge of all the link mixture orders $\{k_l\}_{l=1}^{L}$. We will relax this assumption in the next section. Let N_i be the number of packets sent from the source to receiver i and let M_i be the set of links along that path. Define $X_l^{(i,n)}$ the delay at link l encountered by the n-th packet sent to receiver i. Let $\mathbf{Z}_l^{(i,n)} = \left[Z_{l,0}^{(i,n)}, \ldots, Z_{l,k_l}^{(i,n)}\right]$ be the selection vector for $X_l^{(i,n)}$.

With these definitions, maximum likelihood (ML) estimation of the set of internal link mixture densities can be formulated as a missing data problem. The Expectation Maximization (EM) algorithm has been extensively applied to approximate ML and penalized ML (PML) estimates for mixture models [15, 29, 30]. Let $\mathbf{X} = \{X_l^{(i,n)}\}$ and $\mathbf{Z} = \{\mathbf{Z}_l^{(i,n)}\}$ for all l, i, n. $\{\mathbf{X}, \mathbf{Z}\}$ is called *missing data* or *hidden data*. Define $Y^{(i,n)}$ as the end-to-end delay of the n-th packet received by receiver i. The observables $\mathbf{Y} = \{Y^{(i,n)}\}_{i,n}$ are called the *incomplete data* and the set $\{\mathbf{X}, \mathbf{Z}, \mathbf{Y}\}$ is said to be the *complete data*. The EM algorithm generates a sequence of estimates of the unknown parameters $\mathbf{\Theta}$ which have the property that the likelihood sequence $\mathcal{L}(\mathbf{\Theta}) = f(\mathbf{Y}|\mathbf{\Theta})$ is nondecreasing.

It is easily shown that the likelihood of the *complete data* can be factorized as

$$\mathcal{L}_c(\mathbf{\Theta}) \overset{\text{def}}{=} f(\mathbf{X}, \mathbf{Y}, \mathbf{Z}|\mathbf{\Theta}) = f(\mathbf{Y}|\mathbf{X}) f(\mathbf{X}, \mathbf{Z}|\mathbf{\Theta}),$$

and thus maximization of $\mathcal{L}_c(\mathbf{\Theta})$ is equivalent to maximization of the likelihood function $\mathcal{L}(\mathbf{\Theta}) \overset{\text{def}}{=} f(\mathbf{X}, \mathbf{Z}|\mathbf{\Theta})$. For a specific link l, X_l is a mixed random variable with density function f_l given by (3) and therefore, up to a constant, the complete data log-likelihood function is:

$$\log \mathcal{L}(\mathbf{\Theta}) = \sum_{l=1}^{L} \sum_{i:l \in M_i} \sum_{n=1}^{N_i} \left\{ Z_{l,0}^{(i,n)} \log \alpha_{l,0} + \sum_{m=1}^{k_l} Z_{l,m}^{(i,n)} \left(\log \alpha_{l,m} + \log \phi(X_l^{(i,n)}; \theta_{l,m}) \right) \right\}. \tag{4}$$

The EM algorithm updates parameter estimates by applying two steps at each iteration. At the t-th iteration, the E-step computes conditional expectation of complete data log-likelihood given observations $\mathbf{Y}$ and current parameter estimates $\hat{\mathbf{\Theta}}^{(t)}$

$$Q(\mathbf{\Theta}, \hat{\mathbf{\Theta}}^{(t)}) = E\left[\log \mathcal{L}(\mathbf{\Theta}) | \mathbf{Y}; \hat{\mathbf{\Theta}}^{(t)}\right]. \tag{5}$$

The M-step maximizes the Q function computed in the E step with respect to $\mathbf{\Theta}$ to produce

$$\hat{\mathbf{\Theta}}^{(t+1)} = \underset{\mathbf{\Theta}}{\operatorname{argmax}}\, Q(\mathbf{\Theta}, \hat{\mathbf{\Theta}}^{(t)}). \tag{6}$$

Derivation of the E and M steps for the hybrid mixture model is similar to that for a single Gaussian mixture model [30] and is given in Appendix A.

4.3 PML-EM Algorithm with MML Penalty

When the number of link components k_l's is unknown the ML-EM algorithm is not guaranteed to converge. This is due to a fundamental ambiguity of unknown model order. To illustrate, consider the estimation of a k-component mixture having the form of (2) with parameters $\mathbf{\Theta} = \{\alpha_1, \ldots, \alpha_k, \theta_1, \ldots, \theta_k\}$. These parameters have the same likelihood as the $k+1$ component mixture $\mathbf{\Theta}' = \{\alpha_1, \ldots, \alpha_{k-1}, (1-\beta)\alpha_k, \beta\alpha_k, \theta_1, \ldots, \theta_k, \theta_k\}$ for any $0 < \beta < 1$. One of the most effective ways to eliminate this ambiguity is to add a penalty to the log-likelihood function which penalizes the addition of more components to the mixture.

Many model order penalties have been proposed including: Akaike Information Criterion(AIC) [24], Minimum Description Length(MDL) [25] and Minimum Message Length(MML) [26]. Figueiredo and Jain [15] applied the MML penalty to finite mixture models by introducing a prior to the parameters and an information theoretic penalty depending on quantization of parameter space. They developed an unsupervised method for simultaneously selecting model order and estimating parameters. The incomplete data penalized log-likelihood is expressed as

$$\tilde{\mathcal{L}}(\mathbf{\Theta}) \overset{\text{def}}{=} \log f(\mathbf{\Theta}) + \log f(\mathbf{Y}|\mathbf{\Theta}) - \frac{1}{2}\log|\mathbf{I}(\mathbf{\Theta})| - \frac{c}{2}(1 + \log \kappa_c), \tag{7}$$

where $\mathbf{I}(\mathbf{\Theta})$ is the Fisher information matrix associated with the incomplete data $\mathbf{Y}$, $|\mathbf{A}|$ denotes the determinant of square matrix $\mathbf{A}$, c is the dimension of $\mathbf{\Theta}$, and κ_c is the so-called *optimal quantizing lattice constant for* $\Re^c$.

To apply the MML algorithm [15] of Figueiredo and Jain to our network delay tomography problem their method has to be extended to another layer of hidden data. More specifically, while in [15] the realizations from the mixture model were observed directly, in our application only sums of these realizations (along probe paths) are observed. In other words, the end-to-end delays are themselves convolutive mixtures of the additive mixtures describing the link delays. In addition to the approximations made in [15], including $\kappa_c \approx 1/12$, several other approximations were made to make this extension.

The standard incomplete data Fisher Information matrix $\mathbf{I}(\mathbf{\Theta})$ is not closed form, even for a directly observed finite mixture [27]. Therefore, we replace it by the complete data Fisher information matrix which in the network tomography setting is

$$\tilde{\mathbf{I}}(\mathbf{\Theta}) \;=\; -E\left[\nabla^2_{\mathbf{\Theta}} \log f(\mathbf{X}, \mathbf{Z}|\mathbf{\Theta})\right] = \text{block-diag}\left\{n_l \mathbf{I}_1(\mathbf{\Theta}_l)\right\}_{l=1}^{L},$$

where $\mathbf{I}_1(\mathbf{\Theta}_l)$ is the Fisher information matrix associated with the complete data at link l, $\mathbf{\Theta}_l$ denotes the parameter set of the l-th link, and $n_l = \sum_{i:l\in M_i} N_i$ is the total number of packets passing through the l-th link. $\mathbf{I}_1$ itself has block-diagonal structure

$$\mathbf{I}_1(\mathbf{\Theta}_l) = \text{block-diag}\left\{\mathbf{A}_l, \alpha_{l,1}\mathbf{I}_2(\theta_{l,1}), \ldots, \alpha_{l,k_l}\mathbf{I}_2(\theta_{l,k_l})\right\},$$

where $\mathbf{I}_2(\theta_{l,m})$ is the Fisher information matrix associated with the hidden m-th component delay variable $X_{l,m}$ on link l, and $\mathbf{A}_l = \text{diag}\left\{\alpha_{l,m}^{-1}\right\}_{m=0}^{k_l}$. If any one of the $\alpha_{l,m}$'s is zero, it is removed from $\mathbf{A}_l$ and k_l is decreased by 1.

The prior on the parameter set was taken as

$$f(\mathbf{\Theta}) = \prod_{l=1}^{L}\left\{f(\alpha_{l,0}, \ldots, \alpha_{l,k_l}) \prod_{m=1}^{k_l} f(\theta_{l,m})\right\}.$$

where $f(\alpha_{l,0}, \ldots, \alpha_{l,k_l})$ and $f(\theta_{l,m})$ are the non-informative Jeffreys' priors [28],

$$\begin{aligned} f(\alpha_{l,0}, \ldots, \alpha_{l,k_l}) &\propto \sqrt{|\mathbf{A}|} = (\alpha_{l,0}\alpha_{l,1}\ldots\alpha_{l,k_l})^{-1/2} \\ f(\theta_{l,m}) &\propto \sqrt{|\mathbf{I}_2(\theta_{l,m})|} \end{aligned}$$

for $\sum_{m=0}^{k_l} \alpha_{l,m} = 1$ and $0 < \alpha_{l,m} < 1$. This yields the MML penalized likelihood function

$$\tilde{\mathcal{L}}(\mathbf{\Theta}) = \log f(\mathbf{Y}|\mathbf{\Theta}) - \frac{d}{2}\sum_{l=1}^{L}\sum_{m=1}^{k_l} \log \alpha_{l,m} - \sum_{l=1}^{L} \frac{k_l(d+1)+1}{2}\left(\log\frac{n_l}{12} + 1\right), \qquad (8)$$

where d is the dimension of $\theta_{l,m}$, e.g., $d = 2$ for a Gaussian component mixture.

To derive the E step of the EM algorithm applied to maximizing (8) we adopt the same complete data as in the previous section. With this it is easy to see that the E step is a modification of (5) where $Q(\mathbf{\Theta}, \hat{\mathbf{\Theta}}^{(t)})$ now has an additional penalty given by the second and third additive terms on the RHS of (8). The modified M-step gives the following updates for the mixing parameters

$$\alpha_{l,m}^{(t+1)} = \frac{\max\left\{\left(\sum_{i:l\in M_i}\sum_{n=1}^{N_i}\omega_{l,m}^{(i,n)}\right) - \frac{d}{2}, 0\right\}}{\sum_{m=1}^{k_l}\max\left\{\left(\sum_{i:l\in M_i}\sum_{n=1}^{N_i}\omega_{l,m}^{(i,n)}\right) - \frac{d}{2}, 0\right\} + \sum_{i:l\in M_i}\sum_{n=1}^{N_i}\omega_{l,0}^{(i,n)}} \quad m = 1, \ldots, k_l. \qquad (9)$$

The M-step for the remaining parameters depends on the specific form of the mixture density components.

The algorithm uses the following strategy to select the number k_l of components at the l-th link. It starts with all the k_l set to same user-specified upper bound and annihilates components as follows. If $\alpha_{l,m}^{(t+1)} = 0$, component m is removed from f_l and its probability mass is redistributed over the other non-zero-probability components at the next iteration. Note this procedure is myopic since it does not allow components to come back to life and therefore the estimate is not guaranteed to converge to the MML estimate. However, we restrict our implementation to this myopic strategy due to implementation complexity constraints.

5 Experimental Results

5.1 Model Simulation: ML-EM for Known Model Order

We simulated a small network with the simple virtual tree topology shown in Fig.5. Throughout this experiment the number of components k_l's are known to the estimator. We specialized the EM algorithm to a Gaussian continuous component mixture (see Appendix). 2 or 3 Gaussian components were assigned to each link in addition to a point mass. These simulations were implemented in `matlab` and we generated 15000 i.i.d. end-to-end probe delays for each of the four probe paths. The ML-EM algorithm was applied to estimate the Gaussian components, their mixing parameters and the weight of the point mass at zero. Convergence was achieved after 4000 iterations, or approximately 90 iterations per parameter. Table 5.1 lists the number of Gaussian mixture components for each link and the true/estimated probabilities $\alpha_{l,0}$ of the probe encountering empty queue on link l. Fig. 6 compares the estimated Gaussian mixture components to true Gaussian mixture components. These results illustrate high accuracy for this simulation where there is no model error and the number of components is known.

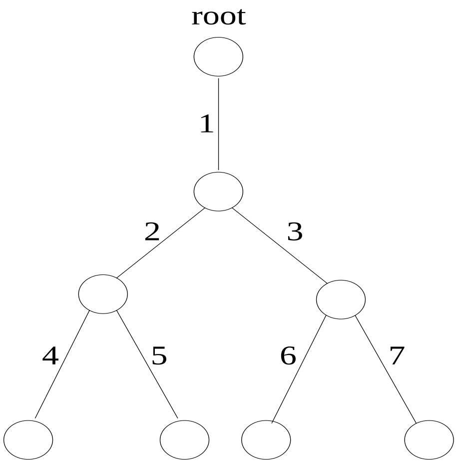


Figure 5: Network topology used in simulations.

Link(l)	1	2	3	4	5	6	7
k_l	3	2	2	2	2	2	2
$\alpha_{l,0}$	0.25	0.3	0.1	0.2	0.15	0.3	0.2
$\hat{\alpha}_{l,0}$	0.253	0.304	0.099	0.199	0.152	0.313	0.201

Table 1: List of numbers of Gaussian mixture components and true/estimated $\alpha_{l,0}$ for `matlab` simulation in Sec. 5.1

5.2 NS Simulation: MML for Unknown Model Order

For a more realistic simulation we used `ns-2` [16] to simulate the network shown in Fig.5 with a variety of cross traffic types and router configurations. The links were assigned 4Mb bandwidth and 50ms latency. The `ns` parameters for each link were set to a Drop-Tail queue (FIFO queue with finite buffer). The queue buffer sizes were 50 packets long. Probe packets were defined as 40 byte UDP packets. They are generated independently and sent along each of the 4 paths according to a Poisson process with mean interarrival time 16ms and rate 20Kb/sec. Cross traffic was also generated in each link by `ns` and consisted of 11 Pareto On-Off UDP streams and 11 Exponential On-Off TCP flows. A total of $N = 20000$ probe packets are collected at each receiver node. We estimated each probe queueing delay by subtracting the minimum probe delay over the N samples for the same path.

The MML algorithm was implemented with Gaussian continuous mixtures and the number of mixture components at each link was initialized to $k_l = 10$. The convergence curve of the penalized likelihood function is shown in Fig.7. The vertical lines indicate the iterations when at least one component is annihilated and the numbers above them show the corresponding links. Link delay pdf estimates are shown in Fig.8 for 4 representative links. The true internal link delay distributions are estimated empirically from the `ns` simulated data. They are also assumed to be mixed distributions with an atom at origin and a continuous portion. The mass of the atom, which is denoted as “True $\alpha_{l,0}$” in the figure, is the empirically estimated probability of an empty queue at link l calculated from sample averages. The continuous portion of the true distribution is estimated by the histogram of non-zero link delay samples and normalized to have mass $1 -$ (True $\alpha_{l,0}$). The estimated Gaussian mixtures are shown along with the normalized histogram for comparison. Note that the probability mass of the mixture is $1 - \hat{\alpha}_{l,0}$ for delays at link l. For link 1,2,4, and 5, the Gaussian mixture components capture the profile of the empirical continuous portion of the density and form a good estimate. However, for some other links, especially for link 7 as shown in Fig.9, the Gaussian mixture with a single point mass does a poor job at capturing the spike in the internal link histogram. This error is probably due to the limitation of the 10 Gaussian + 1 point mass component model. For a better fit to the internal delay histograms it may be necessary to assign more point masses and other flatter density models. This is a topic for future study. Other sources of error might include: violation of the spatial or temporal independence assumptions; insufficient number of

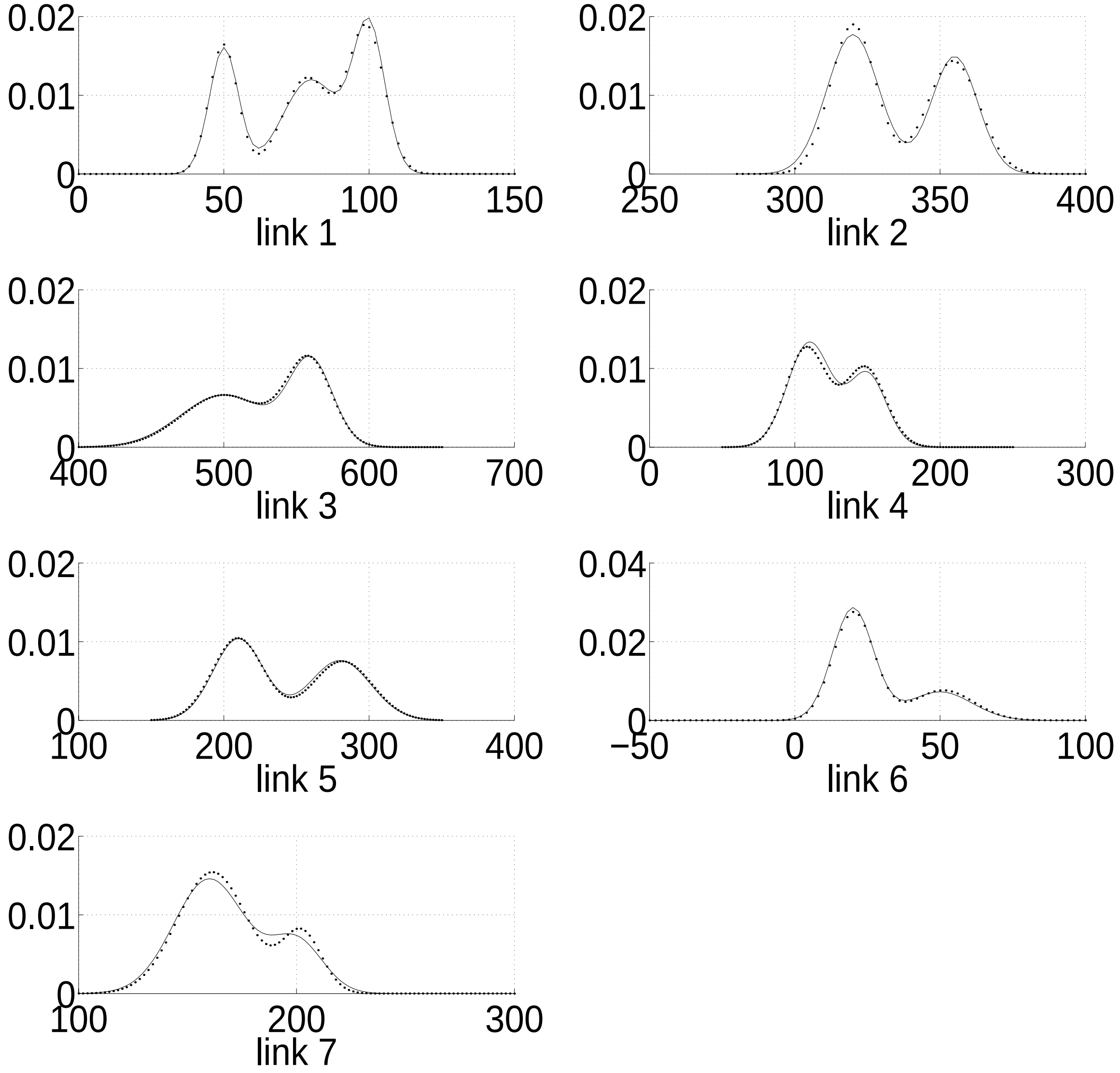


Figure 6: True(solid) and estimated(dotted) Gaussian mixture components for model simulation. The horizontal axes denote link packet delays. Here the EM algorithm is used to estimate the mixed Gaussian mixture parameters in addition to the empty queue probabilities $\{\alpha_{l,0}\}$ for simulated measurements obeying a true mixed Gaussian mixture with known numbers of components.

probe samples to resolve link densities; insufficient number of iterations of the MML algorithm; existence of local maxima in the likelihood function; and burstiness (non-stationarity) of the traffic. These are topics worthy of additional investigation.

6 Conclusion and Future Work

This paper focuses on the estimation of internal link delay distributions from end-to-end unicast packet delay measurements when there is a positive probability of zero queueing delay. We proposed a new hybrid discrete-continuous finite mixture model which circumvents the difficulties of link delay discretization. For the case

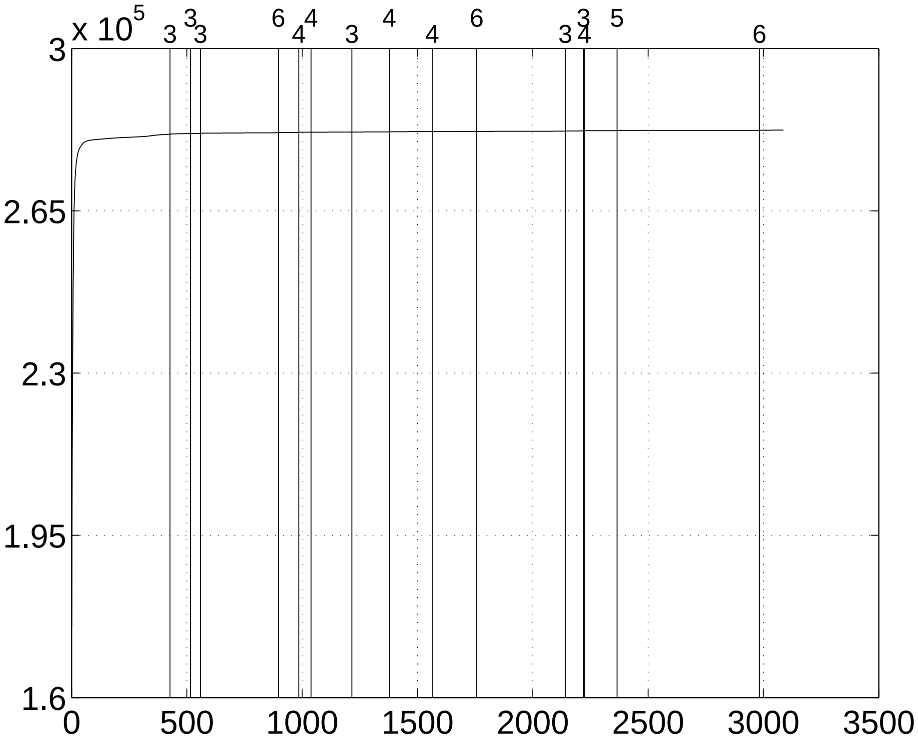


Figure 7: Convergence curve of the unsupervised MML penalized likelihood function obtained during the `ns` simulation. The horizontal axis shows the number of iterations and the vertical lines denote the iteration number where the number of Gaussian mixture components (initialized at 10 for each link) is reduced. Numbers at the top of the graph denote the particular link affected by this reduction.

that mixture model orders are known, we derived an EM algorithm for approximating the ML estimates. Model simulation showed that when all the assumptions hold the EM algorithm can accurately estimate the delay distributions for each internal link. When the model orders are unknown, we implemented an MML order selection penalty and derived an unsupervised algorithm for estimating both the number of mixture components and the continuous density parameters. Although the estimates obtained at the convergence of this algorithm are not necessarily MML estimates, results of `ns-2` simulation showed that reasonably accurate estimates of link delay distributions are possible.

Future work includes finding ways to accelerate convergence of the ML-EM and PML-EM algorithms so that real network data can be applied. The `matlab` implementations are quite slow and this makes it difficult to perform extensive comparisons. Another direction is extension of our model to include spatial dependencies of link delays among different links, especially the links along the same path. For time-varying scenarios adaptive schemes need to be developed in order to capture possible changes in the traffic statistics and the network environment. Another direction is to apply these methods to detecting abnormal changes in link delay distributions. This may help early detection of possible network failures and/or malicious network activities.

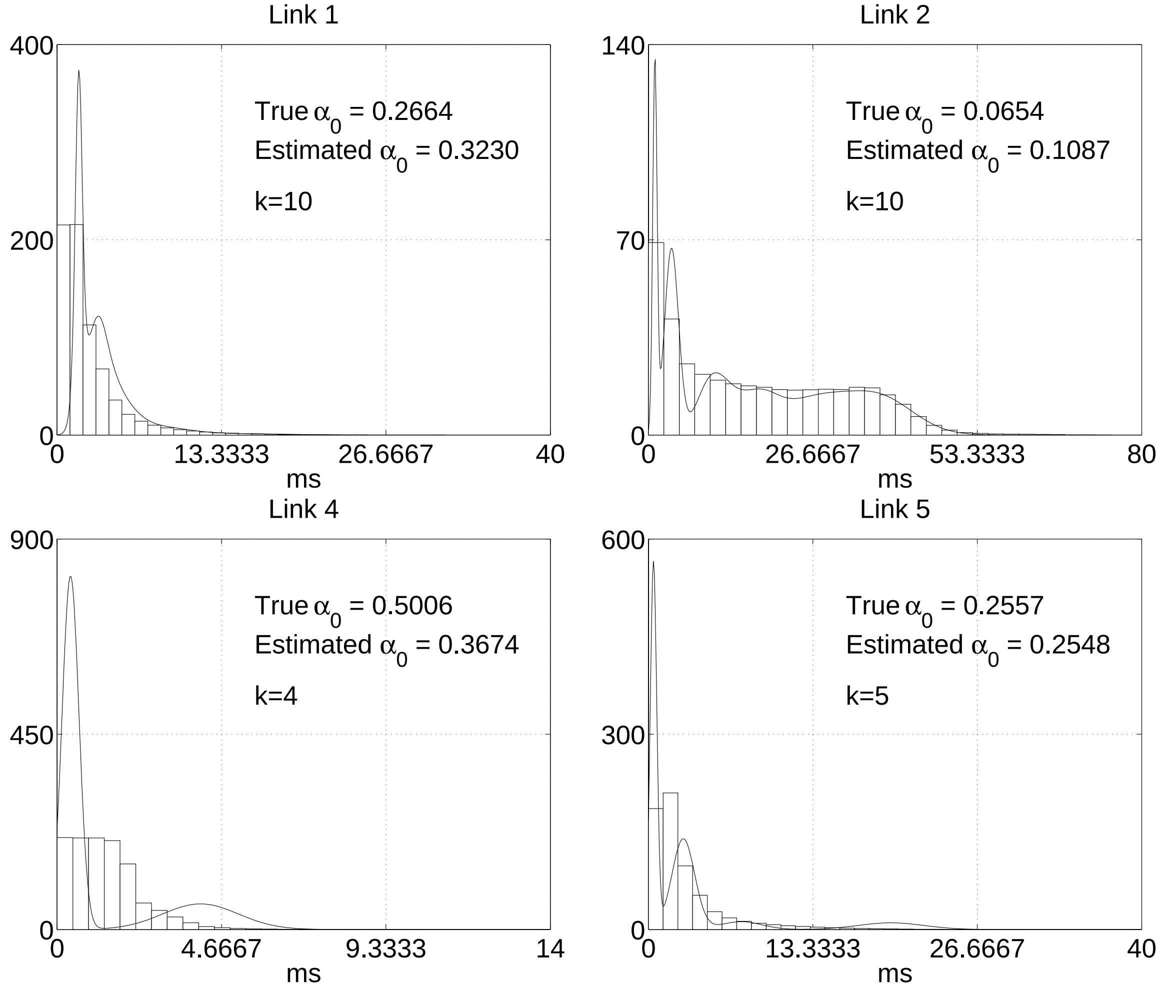


Figure 8: Normalized ns-derived histograms for non-zero link delays and estimated Gaussian mixture density for indicated links. The horizontal axes denote link packet delays in milliseconds. Here the MML algorithm was initialized with 10 Gaussian components for each link, and these plots show estimated mixture densities after 3000 iterations.

A Derivation of EM algorithm

Throughout the derivation, we assume the point mass is located at zero delay for all the links. Before showing the details, we define some additional notations as follows.

1. $g_i(y;\mathbf{\Theta})$ denotes the end-to-end packet delay pdf from root node to receiver i. $g_i(y;\mathbf{\Theta})$ is the convolution of f_l's for $l \in M_i$, which is also a hybrid mixture. The discrete component at $y = 0$ has mass $\prod_{l\in M_i} \alpha_{l,0}$.

2. $g_{i,(l,m)}(y;\mathbf{\Theta})$ is similar to $g_i(y;\mathbf{\Theta})$, except in the convolution f_l is replaced by its m-th component, which is $\alpha_{l,0}$ when $m = 0$ or $\alpha_{l,m}\phi(y;\theta_{l,m})$ when $m \neq 0$. This is the likelihood of an end-to-end delay along path i being equal to y when the delay at link l is contributed by the hidden component random variable $X_{l,m}$.

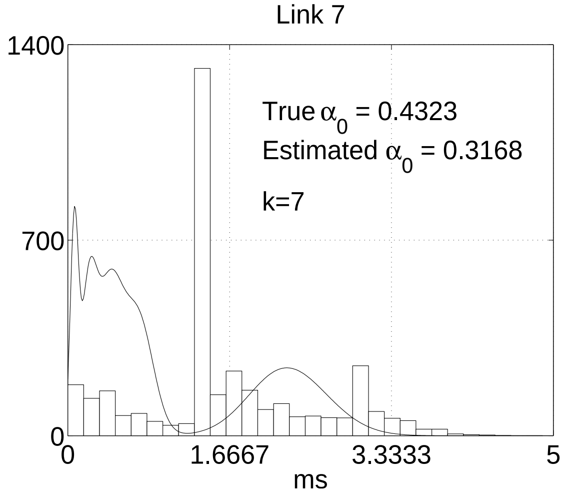


Figure 9: Same as in Fig.8 except that a poor fit occurs for link 7 which has a large point mass near 1.6ms delay. It is mismatched to the single point mass (at zero delay) model used for this experiment.

3. $h_{i,l}(y;\mathbf{\Theta})$ is similar to $g_i(y;\mathbf{\Theta})$, except the l-th link is removed from the convolution.

E-step

To compute the conditional expectation of complete data log-likelihood in (5) in the t-th iteration, let

$$\begin{aligned}\omega_{l,m}^{(i,n)} &= E\left[Z_{l,m}^{(i,n)}|Y^{(i,n)};\hat{\mathbf{\Theta}}^{(t)}\right]\\ &= \frac{P(Z_{l,m}^{(i,n)}=1,Y^{(i,n)};\hat{\mathbf{\Theta}}^{(t)})}{P(Y^{(i,n)};\hat{\mathbf{\Theta}}^{(t)})}\\ &= \frac{g_{i,(l,m)}(y^{(i,n)};\hat{\mathbf{\Theta}}^{(t)})}{g_i(y^{(i,n)};\hat{\mathbf{\Theta}}^{(t)})}\end{aligned}$$

for $m=0,\ldots,k_l$ and $l=1,\ldots,L$. Note that $\omega_{l,m}^{(i,n)}=\begin{cases}1, & m=0\\ 0, & m\neq 0\end{cases}$ when $Y^{(i,n)}=0$ since all the internal link delays must be zero (contributed by the discrete component) when there is no path delay. Also let

$$\begin{aligned}Q_{l,m}^{(i,n)}(\theta_{l,m}) &= E\left[Z_{l,m}^{(i,n)}\log\phi(X_l^{(i,n)};\theta_{l,m})|Y^{(i,n)};\hat{\mathbf{\Theta}}^{(t)}\right]\\ &= \int \log\phi(x;\theta_{l,m})\cdot\frac{\hat{\alpha}_{l,m}^{(t)}\phi(x;\hat{\theta}_{l,m}^{(t)})h_{i,l}(y^{(i,n)}-x;\hat{\mathbf{\Theta}}^{(t)})}{g_i(y^{(i,n)};\hat{\mathbf{\Theta}}^{(t)})}dx\end{aligned}$$

for $m=1,\ldots,k_l$ and $l=1,\ldots,L$. $Q_{l,m}^{(i,n)}(\theta_{l,m})=0$ when $Y^{(i,n)}=0$. The expected log likelihood function in (5) becomes

$$Q(\mathbf{\Theta},\hat{\mathbf{\Theta}}^{(t)})=\sum_{l=1}^{L}\sum_{i:l\in M_i}\sum_{n=1}^{N_i}\left\{\sum_{m=0}^{k_l}\omega_{l,m}^{(i,n)}\log\alpha_{l,m}+\sum_{m=1}^{k_l}Q_{l,m}^{(i,n)}(\theta_{l,m})\right\} \tag{10}$$

The conditional expectation in (10) holds for any choices of ϕ. If Gaussian density functions are used as mixture components (as in our simulations), $Q^{(i,n)}_{l,m}(\theta_{l,m})$ can be further elaborated as

$$
\begin{aligned}
Q^{(i,n)}_{l,m}(\theta_{l,m}) &= \int \left[-\frac{1}{2}\log 2\pi - \log \sigma_{l,m} - \frac{(x-\mu_{l,m})^2}{2\sigma^2_{l,m}} \right] \cdot \frac{\hat{\alpha}^{(t)}_{l,m}\phi(x;\hat{\theta}^{(t)}_{l,m})h_{i,l}(y^{(i,n)}-x;\hat{\mathbf{\Theta}}^{(t)})}{g_i(Y^{(i,n)};\hat{\mathbf{\Theta}}^{(t)})}dx \\
&= -\omega^{(i,n)}_{l,m}\left(\frac{\log 2\pi}{2} + \log \sigma_{l,m} + \mu^2_{l,m}\right) - \frac{\int (x^2 - 2\mu_{l,m}x)\hat{\alpha}^{(t)}_{l,m}\phi(x;\hat{\theta}^{(t)}_{l,m})h_{i,l}(y^{(i,n)}-x;\hat{\mathbf{\Theta}}^{(t)})dx}{2\sigma^2_{l,m}g_i(y^{(i,n)};\hat{\mathbf{\Theta}}^{(t)})}.
\end{aligned}
$$

To evaluate the integral terms in $Q^{(i,n)}_{l,m}$, first let $\kappa = \prod_{\{l' \in M_i, l' \neq l\}}(k_{l'}+1)$ be the model order of $h_{i,l}(y;\hat{\mathbf{\Theta}}^{(t)})$. Also define $\hat{\beta}_j$, $\hat{\rho}_j$ and $\hat{\gamma}^2_j$ ($j = 1,\ldots,\kappa-1$) the mixing probability, mean and variance of the j-th component, respectively. Note that the 0-th component is the atom at zero delay with weight $\hat{\beta}_0 = \prod_{\{l' \in M_i, l' \neq l\}} \hat{\alpha}^{(t)}_{l',0}$. Then $h_{i,l}(y;\hat{\mathbf{\Theta}}^{(t)}) = \hat{\beta}_0\delta(y) + \sum_{j=1}^{\kappa-1}\hat{\beta}_j\phi(y;\hat{\rho}_j,\hat{\gamma}^2_j)$ and it is straight forward to compute the integrals

$$
\begin{aligned}
\int x^2\hat{\alpha}^{(t)}_{l,m}\phi(x;\hat{\theta}^{(t)}_{l,m})h_{i,l}(y^{(i,n)}-x;\hat{\mathbf{\Theta}}^{(t)})dx &= \hat{\beta}_0 \cdot (y^{(i,n)})^2 \cdot \phi(y^{(i,n)};\hat{\theta}^{(t)}_{l,m}) + \\
&\quad \sum_{j=1}^{\kappa-1}\left(\hat{\zeta}^2_j + \hat{\eta}^2_j\right) g_{i,(l,m)}(y^{(i,n)};\hat{\mathbf{\Theta}}^{(t)}) \\
\int x \cdot \hat{\alpha}^{(t)}_{l,m}\phi(x;\hat{\theta}^{(t)}_{l,m})h_{i,l}(y^{(i,n)}-x;\hat{\mathbf{\Theta}}^{(t)})dx &= \hat{\beta}_0 \cdot y^{(i,n)} \cdot \phi(y^{(i,n)};\hat{\theta}^{(t)}_{l,m}) + \sum_{j=1}^{\kappa-1}\hat{\eta}_j g_{i,(l,m)}(y^{(i,n)};\hat{\mathbf{\Theta}}^{(t)})
\end{aligned}
$$

where $\hat{\eta}_j = (\hat{\gamma}^2_j \cdot \hat{\mu}^{(t)}_{l,m} + \hat{\sigma}^{2(t)}_{l,m}(y^{(i,n)} - \hat{\rho}_j))/(\hat{\sigma}^{2(t)}_{l,m} + \hat{\gamma}^2_j)$ and $\hat{\zeta}^2_j = (\hat{\sigma}^{2(t)}_{l,m} \cdot \hat{\gamma}^2_j)/(\hat{\sigma}^{2(t)}_{l,m} + \hat{\gamma}^2_j)$.

M-step

To update the parameter estimates $Q(\mathbf{\Theta},\hat{\mathbf{\Theta}}^{(t)})$ is to be maximized over $\mathbf{\Theta}$ as shown in (6). The updated estimates are

$$
\begin{aligned}
\hat{\alpha}^{(t+1)}_{l,m} &= \frac{\sum_{i:l\in M_i}\sum_{n=1}^{N_i}\omega^{(i,n)}_{l,m}}{\sum_{i:l\in M_i}N_i} \quad m = 0,\ldots,k_l \\
\hat{\theta}^{(t+1)}_{l,m} &= \operatorname*{argmax}_{\theta} \sum_{i:l\in M_i}\sum_{n=1}^{N_i} Q^{(i,n)}_{l,m}(\theta) \quad m = 1,\ldots,k_l,
\end{aligned}
$$

In the case of ϕ's being Gaussian, $\hat{\theta} = (\hat{\mu},\hat{\sigma}^2)$ and

$$
\begin{aligned}
\hat{\mu}^{(t+1)}_{l,m} &= \frac{1}{\sum_{i:l\in M_i}\sum_{n=1}^{N_i}\omega^{(i,n)}_{l,m}} \cdot \sum_{i:l\in M_i}\sum_{n=1}^{N_i} \frac{\int x \cdot \hat{\alpha}^{(t)}_{l,m}\phi(x;\hat{\theta}^{(t)}_{l,m})h_{i,l}(y^{(i,n)}-x;\hat{\mathbf{\Theta}}^{(t)})dx}{g_i(y^{(i,n)};\hat{\mathbf{\Theta}}^{(t)})} \\
\hat{\sigma}^{2\ (t+1)}_{l,m} &= \frac{1}{\sum_{i:l\in M_i}\sum_{n=1}^{N_i}\omega^{(i,n)}_{l,m}} \cdot \sum_{i:l\in M_i}\sum_{n=1}^{N_i} \frac{\int (x-\hat{\mu}^{(t+1)}_{l,m})^2\hat{\alpha}^{(t)}_{l,m}\phi(x;\hat{\theta}^{(t)}_{l,m})h_{i,l}(y^{(i,n)}-x;\hat{\mathbf{\Theta}}^{(t)})dx}{g_i(y^{(i,n)};\hat{\mathbf{\Theta}}^{(t)})}
\end{aligned}
$$

References

[1] Y. Vardi, “Network tomography: Estimating source-destination traffic intensities from link data,” *J. Amer. Statist. Assoc.*, vol.91, pp.365-377,1996.

[2] M. Coates, A. Hero, R. Nowak, B. Yu, ”Internet tomography,” *IEEE Signal Processing Magazine*, Vol. 19, No. 3, pp. 47-65, May 2002.

[3] J. Cao, D. Davis, S.V. Wiel, B. Yu, “Time-varing network tomography: router link data,” *J. Amer. Statist. Assoc.*, vol. 95, 1063-1075, 2000.

[4] R. Caceres, N.G. Duffield, J. Horowitz, D.F. Towsley, “Multicast-based inference of network-internal loss characteristics,” *IEEE Trans. on Information Theory*, vol.45, no.7, Nov. 1999.

[5] M. Coates and R. Nowak, “Network tomography for internal delay estimation,” *ICASSP 2001*, Salt Lake City, May 2001.

[6] F. Lo Presti, N.G. Duffield, J. Horowitz, D. Towsley, “Multicast-based inference of network-internal delay distributions,” *UMass CMPSCI*, pp.99-55, 1999.

[7] National Laboratory for Applied Network Research (NLNAR). See `http://moat.nlanr.net/index.html`.

[8] The Surveyor Project. See `http://www.advanced.org/surveyor/`.

[9] W. Jiang and H. Schulzrinne, “QoS measurement of Internet real-time multimedia services,” *Proceedings NOSSDAV*, Chapel Hill, NC, June 2000.

[10] K. Mochalski and J. Micheel and S. Donnelly, ”Packet delay and loss at the Auckland Internet Access Path,” *Proceedings PAM2002 Passive & Active Measurement Workshop*, Fort Collins, Colorado, pp. 46-55, Mar. 2002.

[11] M. Coates and R. Nowak, “Sequential Monte Carlo inference of internal delays in nonstationary communication networks,” *IEEE Trans. on Signal Processing*, Vol.50, no.2, pp.366 -376, Feb. 2002.

[12] N.G. Duffield, J. Horowitz, F. Lo Presti, and D. Towsley, “Network delay tomography from end-to-end unicast measurements,” *Proc. of the 2001 International Workshop on Digital Communications 2001- Evolutionary Trends of the Internet*, Taormina, Italy, Sep., 2001

[13] Y. Tsang, M. Coates, and R. Nowak, “ Nonparametric internet tomography,” *ICASSP 2002*, Orlando, May 2002.

[14] M. Shih and A. Hero, “Unicast inference of network link delay distributions from edge measurements,” *ICASSP 2001*, Salt Lake City, May 2001.

[15] Figueiredo M., Jain, A.K.,“Unsupervised learning of finite mixture models,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol.24, pp.381–396, Mar. 2002.

[16] *ns* - Network Simulator. See `http://www-mash.cs.berkeley.edu/ns/ns.html`.

[17] Y. Zhang, N.G. Duffield, V. Paxson, S. Shenker, “On the constancy of internet path properties,” *ACM SIGCOMM Internet Measurement Workshop*, San Francisco, CA, Nov., 2001.

[18] N.G. Duffield, F. Lo Presti, “Multicast inference of packet delay variance at interior network links,” Preprint from MINC, `http://www-net.cs.umass.edu/minc/`.

[19] Thomas J. Rothenberg, “Identification in parametric models,” *Econometrica*, vol. 29(3), pp.557-591, May 1971.

[20] V. Paxson, S. Floyd, “Wide area traffic: the failure of Poisson modeling,” *IEEE/ACM Trans. on Networking*, 3(3), pp.226-244, June 1994.

[21] J. S. Marron, M. P. Wand, “Exact mean integrated squared error,” *Annals of Statistics*, vol.20, pp.712-736, 1992.

[22] N.G. Duffield, F. Lo Presti, V. Paxson, D. Towsley, “Inferring link loss using striped unicast probes,” *Proc. IEEE Infocom*, Apr. 2001.

[23] D. Bertsekas, R. Gallager, “Data networks,” Prentice-Hall, 1992.

[24] M. Whindham and A. Cutler, “Information ratios for validating mixture analysis,” *Journal of the American Statistical Association*, vol.87, pp.1188-1192, 1992.

[25] J. Rissanen, “Stochastic Complexity in Statistical Inquiry,” Singapore: World Scientific, 1989.

[26] C. Wallace and P. Freeman, “Estimation and inference via compact coding,” *Journal of the Royal Statistical Society (B)*, vol.49, no.3, pp.241-252, 1987.

[27] G. McLachlan and D. Peel, “Finite mixture models,” New York: John Wiley & Sons, 2000.

[28] J. Bernardo and A. Smith, *Bayesian Theory*. Chichester, UK: J. Wiley & Sons, 1994.

[29] G. Celeux, S. Chrètien, F. Forbes, and A. Mkhadri, “A component-wise EM algorithm for mixtures,” Tech. Rep. 3746, INRIA Rhône-Alpes, France, 1999.

[30] L. Xu, M. Jordon, “On convergence properties of the EM algorithm for Gaussian mixtures,” *Neural Computation*, vol.8, pp.129-151, 1996.