Unicast Inference of Network Link Delay Distributions from Edge Measurements

Meng-Fu Shih

Advisor: Professor A. Hero

May 1st, 2001

Abstract

Inference of network internal link characteristics has become an increasingly important issue for operating and evaluating large telecommunication networks. Since it is usually impractical to directly monitor each link along a specific path, end-to-end probes are sometimes used to collect link characteristic information at edge nodes of the network. This report deals with unicast probing methods for estimation of link delay characteristics. Link delay is usually considered as an important QoS parameter. Unicast traffic is easy to generate and is supported by almost every network currently in operation.

In the first part of the report, we assume that link delays are spatially and temporally independent. We propose a bias corrected estimator for the internal link delay cumulant generating function (CGF). Through computer simulations we show that the proposed estimator attains a level of mean squared error comparable to link delay CGF estimates obtained from directly measurements. By estimating the link delay level exceedance probabilities using the CGF estimates, we are also able to locate the bottleneck link within the network.

The second part of the report considers the link delays along a certain path as a spatial first order Markov chain. Under the assumption of discrete link delay and temporal independence, we generalize the EM algorithm of link delay probability mass function (pmf) estimation into spatially dependent delay network. The computer simulations show that the proposed EM algorithm provides satisfying pmf estimates under both independent and dependent delay networks.

1 Introduction

Network monitoring, prediction and diagnosis are very important issues for network operators, designers, or even end users. Useful information of network behavior includes packet drop rate, packet delay distribution or average, bandwidth utilization, traffic intensity, packet length distribution, and so on. Consider the following examples.

(1) Network administrator can use traffic information to identify when the network has to be modified. One modification that can relieve an Ethernet network that is approaching its capacity is to partition it into subnetworks connected by bridges. The subnetworks should connect nodes that communicate frequently in a way that limits the traffic between different subnetworks. To accomplish this, network administrator needs to know, or at least be able to estimate the traffic intensities between each possible source-destination node pairs.

(2) Another example is to improve the routing performance within the network. Current routing protocols use simple metrics to specify the cost of reaching a destination through a certain link. RIP uses hop counts and most routers implementing OSPF use link bandwidths. If the routers can be provided with information about link delay and packet drop rate, they will be able to include such information in the metrics and make routing more efficient.

(3) Packet delay and drop rate are also important parameters of quality of service (QoS), especially for video and voice data. Network users can use them to judge if they get the requested service level, while network administrators can upgrade and maintain the network according to these parameters.

However, monitoring and diagnosing network internal behavior is a challenging problem due to several factors:

(1) Direct measurement of packet transport statistics are usually impossible - internal nodes may not support such diagnostics or these diagnostics may be disabled to minimize overhead;

(2) The internal parameters of service provider controlled links are usually inaccessible to outsiders.

Current monitoring schemes, like Remote Monitering ($RMON$), require dispatching network monitoring devices at all the nodes of interest. The console manager has to exchange data with every monitoring device in order to collect all the required information. This may cause some problems. First, dispatching monitors at each node of interest may be costly. Second, data exchange between the console and each monitor may create heavy loads to the network. Consider a full binary tree network with N nodes. If each node is installed with a monitor, there are totally N monitoring data flows in the network. On the other hand, if only root node and leaf nodes are to be monitored, they create $(N+3)/2$ data flows instead. When N goes large, the latter case will have approximately half the number of data flows in the former case. This has prompted several groups to investigate methods for inferring internal network behavior based on end-to-end measurements; the so called *network tomography* problem.

Network tomography algorithms can be categorized into two different groups: passive and active methods. Active methods send special packets, called probes, over known routes through the network to measure end-to-end delay, packet error rates, or packet loss probabilities. This gives the network monitoring people more control of the characteristics of the statistics that they want to collect. However, the actively sent probes will purturb the network and hence influence the true underlying traffic behavior which is to be estimated. Moreover, there is a risk that for a substantial network the insertion of a large number of probes required to accurate estimates might significantly impact the performance of the network. Passive methods collect those information by directly monitoring the existing flows. This kind of methods will neither purturb the network nor increase the traffic load. However, a long time may be required to collect enough information before doing any estimation, and the nature of the particular flow to be monitored may cause the information unreliable.

Several passive methods have been developed. Vardi [1] proposed a network tomography method to estimate the traffice intensity between all source-destination directed pairs of nodes of a network from repeated measurements of traffic flow. He also extend this problem to random (Markovian) routing netowrks. This problem was further investigated by Cao, et al [2] for the case of time varying network transport characteristics. Tebaldi and West [5] also developed Bayesian inference techiques for this problem. Tsang, Coates, and Nowak [3, 4] proposed passive unicast network loss tomography based on monitoring existing TCP flows.

There are two kinds of active probing schemes: unicast and multicast probing. Each unicast probe is sent from the source to only one destination, while a multicast probe can have multiple destinations (which means the internal network nodes will replicate the probe in order to send to different receivers). The idea behind multicast-based approach is that correlation in performance seen on intersecting end-to-end paths can be used to draw inferences about the performance characteristics of the common portion of the paths, without the cooperation of network elements on the path. Multicast traffic is particularly well suited for this since a given packet only occurs once on a given link in the multicast tree. Suppose packets are exchanged on a mesh of paths between a collection of N measurement nodes in a network. If the packets are unicast, then the load on the network may grow proportioinally to N^2 in some parts of the network, depending on the topology. For multicast traffic the load grows proportionally only to N. One of the most promising proposals is MINC (Multicast Inference of Network Characteristics) [6], which proposed strategies for estimating link packet loss rate, link packet delay distribution and variance, network topology and other internal network behaviors [7, 8, 9, 10, 11, 12, 13]. Ziotopoulos, Hero, and Wasserman also presented an estimation algorithm of network link loss rates via chaining in multicast trees.

Although the performance of those multicast-based algorithms is impressive, but there are two serious deficiencies in the methodology. Firstly, multicast protocols are not supported by significant portions of the Internet. Secondly, the internal performance measured by active multicast probes often differs significantly from that encountered by unicast packets, which comprise by far the most substantial component of internet traffic [15] Several papers on link parameter measurement propose unicast probing methods. Jacobson [16] designed a new tool, called `pathchar`, to infer the characteristics of individual links along an internet path by measuring the round trip time of packets sent from a single host. Downey [17] further developed techniques to improve the accuracy of `pathchar`'s estimate and proposed an adaptive form of data collection that reduces the number of measurements required. Lai and Baker [18] presented a *packet-tailgating* technique to measure link bandwidths by packet delays. INCITE (InterNet Control and Inference Tools at the Edge) [19] proposed several promising techniques on unicast network tomography, including both loss rate and delay distribution estimation [20, 21, 22]. Shih and Hero [23] developed a unicast technique for inference of link delay cumulant generating function.

In chapter 1, we focus on estimation of the internal link delay cumulant generating function (CGF). Packet delays are due to three factors:(1) queueing delays, which depend on service times and buffer occupancy;(2) transmission delays, which depend on packet size and link data rate; (3) propagation delays, which depend on the transmission medium. The sum of these delays over a route can be measured by end-to-end delays of unicast probes sent over the route. After collecting a sufficient number of these probes, an overdetermined system of equations is constructed for the link delay CGFs. Based on a least-square approximation, we propose a bias corrected estimator for each internal link delay CGF. We evaluate performance of the algorithm using the `ns` network simulation program [24]. Several measures of performance are investigated, including overall mean-square goodness of fit of the estimated CGF to the empirical CGF. We also apply the estimated CGF to bound the delay level exceedance probability and localize the bottleneck link.

In chapter 2, we consider a different model for internal link delays. As in [9, 22], we discretize the link delay into finite levels. Then we introduce spatial dependency of link delays into our network model. We discuss the reason to adopt spatial dependency and restrict it to the case of first order Markov chain. An EM algorithm is proposed to estimate the conditional link delay probability mass functions (pmf) using unicast packet measurements, which is the modification of the algorithm proposed in [22]. The performance of EM algorithms with and without spatial dependency assumption are both evaluated by computer simulations. The goodness of fit of estimated delay pmfs to the true pmfss is also presented for comparison.

2 Estimation of Network Internal Link Delay Cumulant Generating Functions

2.1 Network Delay Model

Let a communication network consist of m internal links. Identical probe packets are sent through n different paths across the network. Suppose we know the routing of each of the probes, i.e. for a specific pair of source and destination nodes of the probes, we can identify those internal links which the probes will go through. This specifies a nxm probe routing matrix A. A has elements a_{ij} equal to 1 when probe path i intersects link j, and equal to 0 otherwise. Let M_i denote the set of link indices which compose the ith probe path, $i = 1, \ldots, n$. Then $Y_i = \sum_{j \in M_i} X_{ij}$ is the measured end-to-end delay of a probe transmitted along the ith path where X_{ij} is the delay encountered by probe i across link j and $i = 1, \ldots, n$. Define the end-to-end probe delay CGF $K_{Y_i}(t) = \log E[e^{tY_i}]$ and the link delay CGF of the jth link $K_{X_{ij}}(t) = \log E[e^{tX_{ij}}]$, $j \in M_i$, with CGF parameter t, $t \in (-\infty, \infty)$. We make the following temporal/spatial independence and stationary assumptions, respectively:

A1) The link delays X_{ij} are mutually independent, $j \in M_i$, $i = 1, \ldots, n$. This means for a given path j, the delays on different links along that path are independent, which gives us spatial independence. Moreover, for a given link i, packet delays on that link for probes along different paths are also independent, because of the assumption that packets entering the same queue at different times have independent queueing delays and probes along different paths always enter the same link at different times. This specifies the temporal independence.

A2) If paths of probe i and probe k both contain a common link j, then X_{ij} and X_{kj} have identical CGF denoted K_{X_j}. This assumes that link delay distribution are stationary over the observation period.

The CGF of Y can therefore be expressed as

$$\begin{aligned} K_{Y_i}(t) &= \log E\left[e^{tY_i}\right] \\ &= \log E\left[e^{t\left(\sum_{j \in M_i} X_{ij}\right)}\right] \\ &= \log\left\{\prod_{j \in M_i} E\left[e^{tX_{ij}}\right]\right\} \\ &= \sum_{j \in M_i} \log E\left[e^{tX_{ij}}\right] \\ &= \sum_{j=1}^{m} a_{ij} \cdot K_{X_j}(t) \\ &= A_{(i)} \cdot K_X(t) \end{aligned} \tag{1}$$

where $A_{(i)}$ denotes the ith row of the routing matrix A and $K_X(t) = [K_{X_1}(t), \ \ldots, K_{X_m}(t)]^T$ (^{T} denotes transpose). Thus we can express the vector of end-to-end CGFs $K_Y(t) = [K_{Y_1}(t),$ $\ldots, K_{Y_n}(t)]^T$ by the linear relation

$$K_Y(t) = A \cdot K_X(t). \tag{2}$$

When $n \geq m$ and A is full rank, the relation (2) is invertible and thus $K_X(t)$ can be determined from $K_Y(t)$ by the formula $K_X(t) = (A^T A)^{-1} A^T K_Y(t)$. Let $B = (A^T A)^{-1} A^T$, then we have

$$K_{X_j}(t) = \sum_{i=1}^{n} b_{ji} \cdot K_{Y_i}(t). \tag{3}$$

for $j = 1 \ldots m$. A full rank matrix A can be ensured by making $n \geq m$, and selecting distinct probe paths which cover the network, i.e., every link is contained in some path. When A is not full rank, only linear combinations of those link CGFs lying outside of the null space of A can be determined from (2).

2.2 Estimation of CGF

Let N_i be the number of probes collected for a given path i, $i = 1, \ldots, n$. Define

$$\hat{M}_{Y_i}(t) = \frac{1}{N_i} \sum_{k=1}^{N_i} e^{tY_{ik}}, \tag{4}$$

where Y_{ik} is the measured end-to-end delay of the kth received probe along path i. We obtain estimates of the vector $K_X(t)$ from $\hat{M}_Y(t) = [\hat{M}_{Y_1}(t) \ldots \hat{M}_{Y_n}(t)]^T$ by the method of least-squares (LS). Note that as $\hat{M}_{Y_i}(t)$ is an unbiased estimate of the moment generating function $M_{Y_i}(t) = e^{K_{Y_i}(t)}$, a plausible estimator for $K_{X_j}(t)$ in (3) would be the method-of-moments estimate (MOM):

$$\hat{K}'_{X_j} = \sum_{i=1}^{n} b_{ji} \cdot \log\left(\hat{M}_{Y_i}(t)\right) \tag{5}$$

Unfortunately, this estimator is biased due to non-linearity of the log. In order to obtain a bias corrected estimator for $K_{X_j}(t)$, we apply a technique similar to that of Gibbens [25]. In [25] linearization was used to derive bias corrected estimators for *effective bandwidth*, which is of similar mathematical form as the cumulant generating function. Observe that as $\log(1+u) = u - \frac{u^2}{2} + H.O.T.$

$$
\begin{aligned}
\hat{K}'_{X_j}(t) &= \log\left\{\prod_{i=1}^{n}\left(\hat{M}_{Y_i}(t)\right)^{b_{ji}}\right\} \\
&= \log\left\{\prod_{i=1}^{n} E^{b_{ji}}\left[\hat{M}_{Y_i}(t)\right] - \left(\prod_{i=1}^{n} E^{b_{ji}}\left[\hat{M}_{Y_i}(t)\right] - \prod_{i=1}^{n}\left(\hat{M}_{Y_i}(t)\right)^{b_{ji}}\right)\right\} \\
&= \log\left\{\prod_{i=1}^{n} E^{b_{ji}}\left[\hat{M}_{Y_i}(t)\right]\left(1 - \left(1 - \frac{\prod_{i=1}^{n}\left(\hat{M}_{Y_i}(t)\right)^{b_{ji}}}{\prod_{i=1}^{n} E^{b_{ji}}\left[\hat{M}_{Y_i}(t)\right]}\right)\right)\right\} \\
&\approx K_{X_j}(t) - w_j - \frac{1}{2}w_j^2, \qquad (6)
\end{aligned}
$$

where $w_j = 1 - \frac{\prod_{i=1}^{n}\left(\hat{M}_{Y_i}(t)\right)^{b_{ji}}}{\prod_{i=1}^{n}\left(E\left[\hat{M}_{Y_i}(t)\right]\right)^{b_{ji}}}$. This suggests that a reasonable way to correct the bias is to use (6) with an estimate of w_j:

$$
\hat{K}_{X_j}(t) = \sum_{i=1}^{n} b_{ji}\log\left(\hat{M}_{Y_i}(t)\right) + \hat{E}[w_j] + \frac{1}{2}\hat{E}[w_j^2], \qquad (7)
$$

where $\hat{E}[\cdot]$ denotes empirical average for which we use MOM estimates

$$
\begin{aligned}
\hat{E}[w_j] &= 1 - \frac{\prod_{i=1}^{n}\hat{E}\left[\left(\hat{M}_{Y_i}(t)\right)^{b_{ji}}\right]}{\hat{M}_{X_j}(t)} \qquad (8) \\
\hat{E}[w_j^2] &= 1 - \frac{2\prod_{i=1}^{n}\hat{E}\left[\left(\hat{M}_{Y_i}(t)\right)^{b_{ji}}\right]}{\hat{M}_{X_j}(t)} + \\
&\quad \frac{\prod_{i=1}^{n}\hat{E}\left[\left(\hat{M}_{Y_i}(t)\right)^{2b_{ji}}\right]}{\hat{M}^2_{X_j}(t)}. \qquad (10)
\end{aligned}
$$

$\hat{M}_{X_j}(t)$ is an estimate of the moment generating function of link delay at link j, which can be obtained from

$$
\hat{M}_{X_j}(t) = \prod_{i=1}^{n}\left(\hat{M}_{Y_i}(t)\right)^{b_{ji}}. \qquad (11)
$$

In order to obtain the empirical average $\hat{E}\left[\left(\hat{M}_{Y_i}(t)\right)^{b_{ji}}\right]$, we propose the following two methods:

1 **Sliding Window Method** We define a sliding window with window size W and sliding step size S. Define the number $N_w = \lfloor\frac{N_i - W}{S}\rfloor$ of window increments. Then we

have

$$\hat{E}\left[\left(\hat{M}_{Y_i}(t)\right)^{b_{ji}}\right] = \sum_{l=1}^{N_w} \frac{1}{N_w} \left(\frac{1}{W} \sum_{k=(l-1)S+1}^{(l-1)S+W} e^{t y_{ik}} \right)^{b_{ji}} . \tag{12}$$

2 **Monte Carlo Bootstrap Method** The Monte Carlo bootstrap method described in [26] is quite useful in estimating statistics given a set of i.i.d. observations. The steps are described as follows:

(a) Fit the nonparametric MLE of the distribution of Y_i,

$$\hat{F}: \text{ mass } \frac{1}{N_i} \text{ at } Y_{ik}, \qquad k = 1, \cdots, N_i$$

(b) Draw a "bootstrap sample" from $\hat{F}$,

$$Y_{i1}^*, Y_{i2}^*, \cdots, Y_{iN_i}^* \overset{iid}{\sim} \hat{F}$$

and calculate

$$\hat{M}_{Y_i}^{b_{ji}*}(t) = \frac{1}{N_i} \sum_{k=1}^{N_i} e^{tY_{ik}^*}$$

(c) Independently repeat the previous step a large number L times, obtaining "bootstrap replications" $\hat{M}_{Y_i}^{b_{ji}*}(t)_1, \hat{M}_{Y_i}^{b_{ji}*}(t)_2, \cdots, \hat{M}_{Y_i}^{b_{ji}*}(t)_L$, and calculate

$$\hat{E}\left[\left(\hat{M}_{Y_i}(t)\right)^{b_{ji}}\right] = \frac{1}{L} \sum_{l=1}^{L} \hat{M}_{Y_i}^{b_{ji}*}(t)_l$$

We obtain the empirical average $\hat{E}\left[\left(\hat{M}_{Y_i}(t)\right)^{2b_{ji}}\right]$ in a similar manner using either methods above.

2.3 Experimental Results

We used the `ns` network simulator program to perform a *TCP/UDP* simulation of the network in Fig. 1. Probes were sent through 5 different paths in order to estimate delay CGF for 4 links. The topology is shown in Fig. 1, and the corresponding routing matrix A is

$$A = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \end{pmatrix} \tag{13}$$

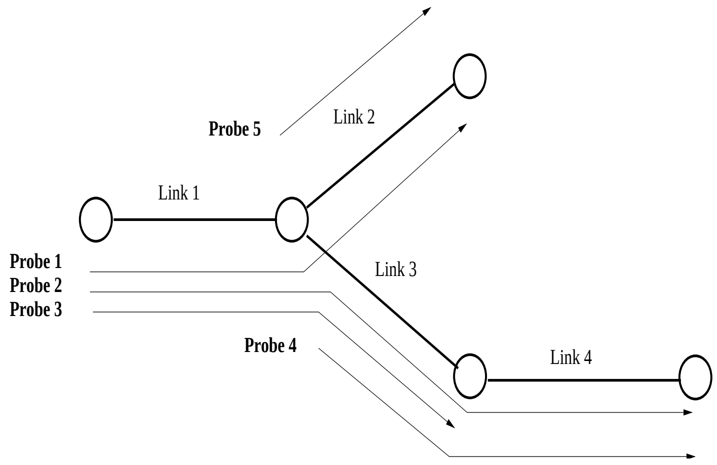

Figure 1: Probe routing paths for the experiment described in Section 2.3

We set up a similar test environment to that reported in Presti and Duffield [9]. All the links to be estimated had bandwidth 4Mb/sec with latency 50ms. Each link was modeled as a Drop-Tail queue (FIFO queue with finite buffer). The queue buffer sizes were 50 packets. We generated probes as 40 byte UDP packets. The probe transmissions were generated independently at each source node according to a Poisson process with mean interarrival time being 16ms and rate 20Kb/sec. The background traffic consisted of both Exponential on-off UDP traffic and FTP traffic.

N probes were collected for each path for a total of 5xN probes. We estimated each probe queueing delay by substracting the minimum probe delay over the N trials. This provides a biased estimate of queueing delay across the probe path since the minimum probe delay is a biased estimator of transmission delay plus latency. However, the bias decreases as $1/N$. Here we use the moving window method to estimate expected values in (9) and (10). We set the window size W to be 2/3 of N, and the window shift step size S to be 10 probe delay samples.

We compared the proposed bias corrected estimator to the biased estimator (5) for $K_X(t)$. We evaluated the CGFs over the range $t = -50$ to $t = 50$. Comparing the estimates of CGF of sampled link delays with and without bias correction in Table 1, we can see that the proposed estimator achieves lower MSE, especially for link 3, which is the link with heaviest load. The corresponding estimated delay CGFs for all the links are shown in Fig. 2. These results were obtained by using $N = 1500$ probe observations per route.

Table 1: MSE of $\hat{K}_{X_j}$ (bias corrected) and $\hat{K}'_{X_j}$ (no bias correction)

Link	1	2	3	4
MSE of $\hat{K}_{X_j}$	0.00033	0.00013	0.00045	0.00013
MSE of $\hat{K}'_{X_j}$	0.00031	0.00015	0.00132	0.00017

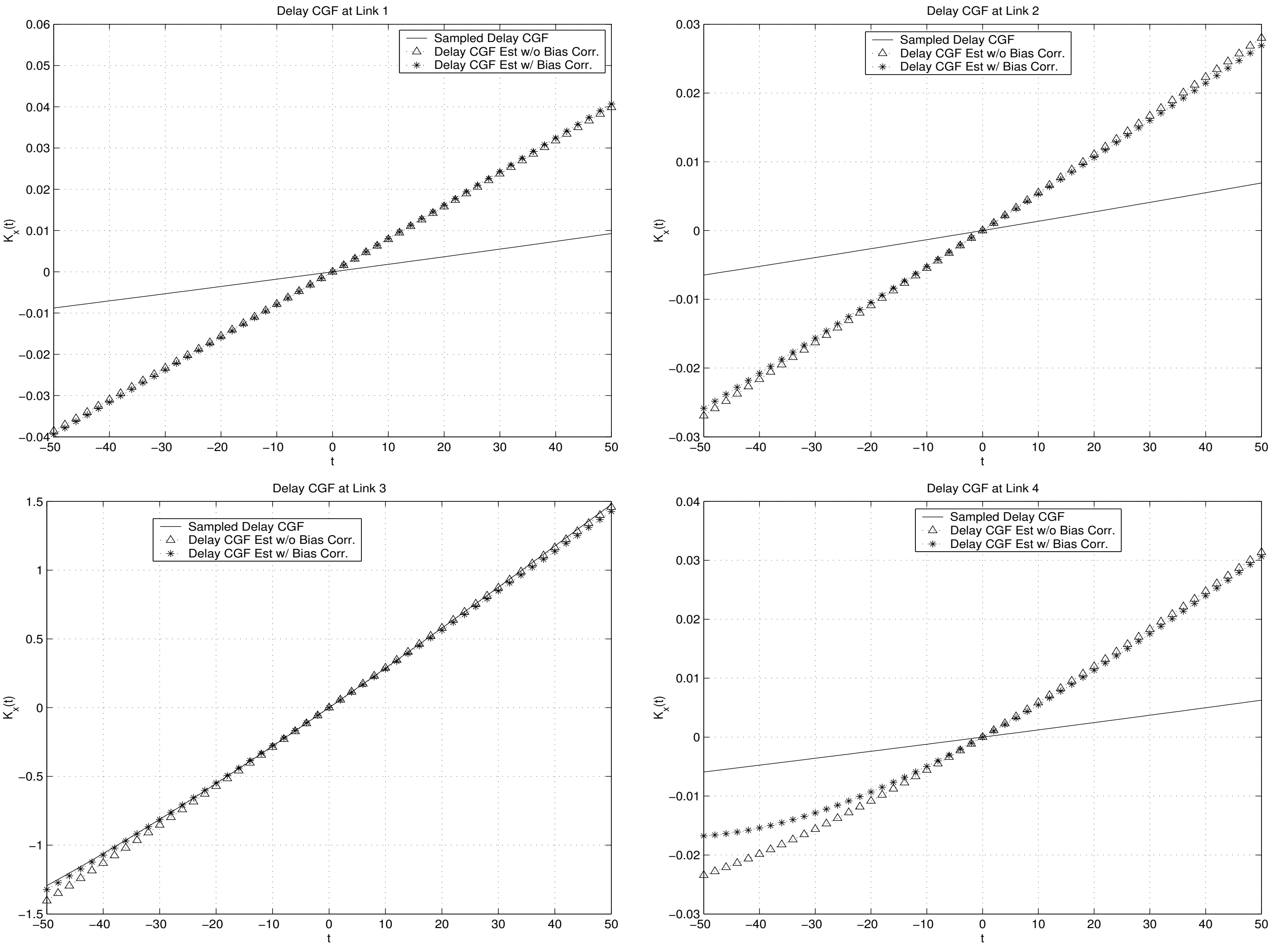

Figure 2: Link Delay CGFs in the ns simulation

2.4 Applications and Extensions

Each link delay CGF preserves all the statistical information of the delay since it is the log of the Fourier transform of the link delay probability density function. We can accurately estimate many features of the delay distribution from the delay CGF.Here we give results for *Bottleneck link detection* We define a bottleneck as the event that the probability of a link delay exceeding some delay threshold δ exceeds a prespecified threshold P. By the Chernoff bound,

$$P(X_j \geq \delta) \leq e^{-t\delta} E\left[e^{tX_j}\right] = P_j, \tag{14}$$

for $t > 0$. By appropriately selecting the threshold δ and a threshold P close to 1, we can detect a bottleneck link by testing whether $\max_{j=1,\ldots,m} P_j > P$ In Table 2, we show the Chernoff bounds for $P(X_j \geq \delta = 0.002s)$ which were estimated from the computer simulation in Section 4. By setting threshold P to be 0.95, we can identify link 3 as the bottleneck link.

Table 2: Chernoff bound and empirical estimate of $P(X_j \geq 0.02)$ for each link delay in Section 4

Link	1	2	3	4
P_j	0.9424	0.9295	1.0025	0.9329
$P(X_j \geq \delta)$	0.0014	0.0002	0.9932	0.0003

2.5 Conclusion

We proposed a unicast method to perform inference on internal link delay characteristics. We derived a bias corrected estimator for internal link delay cumulant generating functions based on LS approximation. The proposed estimator was evaluated by `ns` simulations with TCP/UDP background traffic and FIFO finite buffer link queues. The MSE of the proposed estimator is generally lower than that of the direct biased sample mean estimator. One application of estimated link delay CGF is to localize the bottleneck link. We estimated the Chernoff bound of the delay exceedence probability for each link and identify the bottleneck link of which the bound exceeds the threshold.

3 Estimation of Discretized Internal Link Delay Distributions

3.1 Network Delay Model

Here we consider the network as a logical tree $T = (V, E)$, where V denotes the set of nodes and E denotes the set of links. Suppose totally we have L links in the tree. We label them with $1, \ldots, L$ and associate the link connecting the root node as link 1. Each node in the network has only one parent node, except the root, which has no parnet. In addition, each node has two or more child nodes, except the leaf nodes and possibly the root node. If there is a node with only one child in the physical network, we then just delete that node and all the links attached to it, then add a new link connecting its parent and child node to the tree.

The unicast probe packets are sent from the root node to one of the leaf nodes. The leaf nodes, considered as packet receivers, collect end-to-end probe delays for internal link delay distribution inference. Suppose we have R receivers in the network, then we need to send probe packets to each of the receivers in order to estimate delay distributions at all links, i.e., we will have R different probe paths. Note that all the probe paths include link 1.

The packet link delays, $d^l, l \in E$ are modeled as discrete random variables [9, 22], which take values from the set $\{0, 1, \ldots, D\}$ time units. Let l_r denote the number of links consisting of the path from root to receiver r. The end-to-end packet delay along path r, z_r, then falls in the range $\{0, 1, \ldots, l_r D\}$ time units. For packet delays at each link, we have the following temporal independence and stationary assumption:

A1) Different packets going through a link in the network will have i.i.d. link delays at that link.

Two different models for link delay probability distributions are investigated here :

A2.1) **Spatial Independence Model** With spatial independence assumption, packet delays at different links are considered as independent random variables. We associate each link l a probability mass function (pmf) $\{p_d^l : d = 0, \ldots, D\}$.

A2.2) **First Order Markov Spatial Dependency Model** This model assumes a path in the delay network to be a spatial first order Markov chain. Suppose $\{d_1, \cdots, d_n\}$ are ordered link delays along a path, then the following holds:

$$P(d_i | d_{i-1}, \cdots, d_1) = P(d_i | d_{i-1}),\ i = 2, \cdots, n$$

Let l^F denote the parent link of l. When $l \neq 1$, we assign the packet delay at link l, d^l, a conditional pmf $\{p_{d_1,d_2}^l : d_1, d_2 = 0, \ldots, D\}$, where $p_{d_1,d_2}^l = P(d^l = d_1 | d^{l^F} = d_2)$. For delays at link 1, the pmf remains $\{p_d^1 : d = 0, \ldots, D\}$.

Spatial Independence Model is used in many internal delay analysis/estimation papers, e.g. [8, 9, 22]. However, this is not true in practical situations. Link delay spatial dependency is usually caused by the following facts:

1) The interaction between different data flows along the same path. Consider a path of connected links (queues) and several data flows traversing it. If a packet in one of the flows experiences large delay at a link, which means there are many packets preceeding it in the link buffer, then it is very likely that it will also experience large delay in the next link, since those preceeding packets may again cause a large queueing delay for it in the next link.

2) The correlation of background traffic. Correlated background traffics may exist in many different paths within the network, for instance, the same news article downloaded by users from different places in Michigan. The spatially dependent background traffics will cause delay dependency among the links which they traverse.

3) If link level ARQ (Automatic Retransmission Request) is carried out in the network, heavy delay at a link may propogate backward along the path which the data flow comes from. For example, if the buffer occupancy level at a link is continuously high, most of the packets through that link will experience large delay. Moreover, it is also very likely to drop the packets due to buffer overflow. This will cause retransmissions over the previous link and hence effectively increase the delay at the previous link.

Appendix A and B derive the Fisher information matrix of the pmf parameters for the two models above, respectively.

3.2 EM Algorithm for Spatial Indepedence Delay Networks

This EM algorithm is derived by M.Coates and R.Nowak in [22]. Here we present a version without probe packet pair observations.

Let $y^l_{r,n}$ denote the delay on link l for the nth probe packet sent from root to receiver r. let $\mathbf{y}^l_r = \{y^l_{r,n}\}$ be the set of delays at link l for all probes sent to receiver i and $\mathbf{y}^l = \{\mathbf{y}^l_r\}$ be the set of delays at link l for all the probe packets. Then the set $\mathbf{y} = \{\mathbf{y}^l\}$ contains all the internal link delays for probe packets, which are not observable. Hence $\mathbf{y}$ is called the *unobserved data*. Along with the end-to-end probe delays $\mathbf{z} = \{z_r\}$, which are the *observed data*, we define the *complete data* $\mathbf{x} \equiv \{\mathbf{z}, \mathbf{y}\}$. Let the parameter vector $\mathbf{p} = \{p^l_d\}$ and the complete data likelihood function can be factorized as follows:

$$f(\mathbf{x}|\mathbf{p}) = g(\mathbf{z}|\mathbf{y})h(\mathbf{y}|\mathbf{p}), \tag{15}$$

where g is the conditional pmf of $\mathbf{z}$ given $\mathbf{y}$, which is a kronecker delta function since $\mathbf{y}$ determines $\mathbf{z}$, and h is the likelihood of $\mathbf{y}$. Since $g(\mathbf{z}|\mathbf{y})$ does not depend on p, we can only consider $h(\mathbf{y}|\mathbf{p})$, which is proportional to $f(\mathbf{x}|\mathbf{p})$.

Let $m_{l,d}$ be the number of probe packets which experience a delay of d at link l. Define N_r as the number of probes received at leaf node r and L_r as the set of links in the path from root to leaf node r. Then $m_{l,d} = \sum_{r:l\in L_r} \sum_{n=1}^{N_r} I_{(y^l_{r,n}=d)}$, where I denotes the indicator function. Hence we have

$$h(\mathbf{y}|\mathbf{p}) = \prod_{l=1}^{L}\prod_{d=0}^{D}(p^l_d)^{m_{l,d}} \tag{16}$$

and the log likelihood is

$$\log h(\mathbf{y}|\mathbf{p}) = \sum_{l=1}^{L}\sum_{d=0}^{D} m_{l,d}\log p^l_d \tag{17}$$

Thus, the EM algorithm is

1) **E-Step** In the E-step we need only compute the expectation of $m_{l,d}$. Let $\mathbf{p}^{(i)}$ be the estimated parameter vector after the ith iteration. Then

$$\begin{aligned} E_{\mathbf{p}}\left[m_{l,d}|\mathbf{z}\right] &= E_{\mathbf{p}}\left[\sum_{r:l\in L_r}\sum_{n=1}^{N_r} I_{(y^l_{r,n}=d)}|\mathbf{z}\right] \\ &= \sum_{r:l\in L_r}\sum_{n=1}^{N_r} E_{\mathbf{p}}\left[I_{(y^l_{r,n}=d)}|z_{r,n}\right] \\ &= \sum_{r:l\in L_r}\sum_{n=1}^{N_r} p^{(i)}(y^l_{r,n}=d|z_{r,n}) \end{aligned}$$

Note here the conditional probability

$$p(y^l_{r,n}=d|z_{r,n}) = p^l_d \cdot q_{r,l}(z_{r,n}-d)/q_r(z_{r,n}),$$

where $q_{r,l}$ and q_r are defined in (A.1).

2) **M-Step** To maximize the expectation above, we have

$$p^{l(i+1)}_d = \frac{E_{\mathbf{p}}\left[m_{l,d}|\mathbf{z}\right]}{\sum_{d=0}^{D} E_{\mathbf{p}}\left[m_{l,d}|\mathbf{z}\right]}.$$

3.3 EM Algorithm for First Order Markov Spatial Dependency Delay Networks

We have the same sets of *observed* and *unobserved data* as those in previous section, but here our parameter vector becomes $\mathbf{p} = \{p^1_d\} \cup \{p^l_{d_1,d_2}\}^L_{l=2}$.

Let m^1_d be the number of probe packets which experience delay of d at link 1. Also let $m^l_{d1,d2}, l \neq 1$ denote the number of probe packets which have delay $d1$ at link l and $d2$ at link

l^F, the previous link. Then $m_d^1 = \sum_{r=1}^{R} \sum_{n=1}^{N_r} I_{(y_{r,n}^1 = d)}$ and $m_{d1,d2}^l = \sum_{r:l\in L_r} \sum_{n=1}^{N_r} I_{(y_{r,n}^l = d_1, y_{r,n}^{l^F} = d_2)}$. By using the chain rule, we have the new likelihood function

$$h_d(\mathbf{y}|\mathbf{p}) = \prod_{d=0}^{D} (p_d^1)^{m_d^1} \times \prod_{l=2}^{L} \prod_{d1=0}^{D} \prod_{d2=0}^{D} (p_{d_1,d_2}^l)^{m_{d1,d2}^l}, \tag{18}$$

and the log likelihood becomes

$$\log h_d(\mathbf{y}|\mathbf{p}) = \sum_{d=0}^{D} m_d^1 \log(p_d^1) + \sum_{l=2}^{L} \sum_{d_1=0}^{D} \sum_{d_2=0}^{D} m_{d_1,d_2}^l \log(p_{d_1,d_2}^l) \tag{19}$$

So the EM algorithm can be formulated as follows:

1) **E-Step** As in the spatial independence case, we need only compute the expectation of m_d^1 and m_{d_1,d_2}^l. Let $\mathbf{p}^{(i)}$ be the estimated parameter vector after the ith iteration. Then

$$\begin{aligned} E_{\mathbf{p}}\left[m_d^1|\mathbf{z}\right] &= \sum_{r=1}^{R} \sum_{n=1}^{N_r} p^{(i)}(y_{r,n}^1 = d|z_{r,n}) \\ E_{\mathbf{p}}\left[m_{d_1,d_2}^l|\mathbf{z}\right] &= \sum_{r:l\in L_r} \sum_{n=1}^{N_r} p^{(i)}(y_{r,n}^l = d_1, y_{r,n}^{l^F} = d_2|z_{r,n}),\ l \neq 1 \end{aligned}$$

The conditional probabilities become

$$\begin{aligned} p(y_{r,n}^1 = d|z_{r,n}) &= p_d^1 \cdot q_{r,1}(z_{r,n} - d|d)/q_r(z_{r,n}) \\ p(y_{r,n}^l = d_1, y_{r,n}^{l^F} = d_2|z_{r,n}) &= p_{d_1,d_2}^l \cdot q_{r,l}(z_{r,n} - d_1|d_1, d_2)/q_r(z_{r,n}),\ l \neq 1 \end{aligned}$$

where q_r, $q_{r,1}$ and $q_{r,l}$ are defined in (B.1).

2) **M-Step** To maximize the expectation in E-step, we have

$$\begin{aligned} p_d^{1(i+1)} &= \frac{E_{\mathbf{p}}\left[m_d^1|\mathbf{z}\right]}{\sum_{d=0}^{D} E_{\mathbf{p}}\left[m_d^1|\mathbf{z}\right]} \\ p_{d_1,d_2}^{l(i+1)} &= \frac{E_{\mathbf{p}}\left[m_{d_1,d_2}^l|\mathbf{z}\right]}{\sum_{d_1=0}^{D} E_{\mathbf{p}}\left[m_{d_1,d_2}^l|\mathbf{z}\right]},\ l \neq 1 \end{aligned}$$

3.4 Experimental Results

To assess the performance of our algorithm, we use MATLAB to simulate the simple two-leaf network depicted in Fig.3.

In each of the following experiments, we generate 1500 independent probe packet measurements for each source-receiver path and use both EM algorithms to estimate internal link delay distributions. We discretize the link delays into 11 bins, i.e. $D = 10$. We run the simulation independently for 50 times and then average the result.

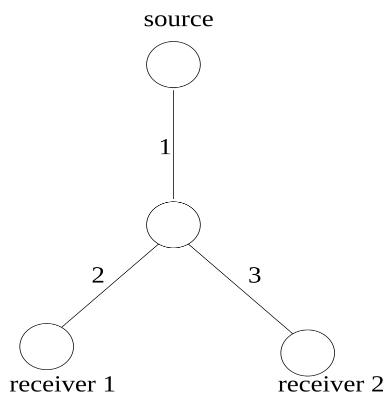

Figure 3: The two-leaf network in MATLAB simulation

Experiment 1: We generate spatially independent link delays in this experiment and estimate link delay pmfs with both EM algorithms. Pmf estimates are plotted in comparison with true pmfs in Fig.4. It shows that although EM alogrithm performs better when it assumes spatial independence, both algorithms capture the profiles of true internal link delay pmfs.

Experiment 2: Here we use the First Order Markov Dependency Model for our delay network and generate spatially dependent link delays. The probe delays at link 2 and 3 are generated according to the conditional pmfs shown in Fig.5(a), given delays at link 1. Fig.6 depicts the true and estimated marginal internal link delay pmfs. It shows that with spatial dependency assumption, the EM algorithm works well and has close estimates to true pmfs. The estimated conditional pmfs are also shown in Fig. 5(b). However, if the EM algorithm assumes spatial independence, it fails to estimate pmfs at link 2 and 3, which have dependency on delays at link 1.

3.5 Conclusion

We introduce spatial dependency in the modeling of network internal link delays. We formulate this property as a spatial first order Markov chain for each probe path. Fisher information matrices of the link delay pmfs given a set of end-to-end probe delay observations are derived for both the spatially independent and dependent link delays. We modify the EM algorithm used in spatial independence model to estiamte the conditional delay pmfs in spatial dependence model. We conduct two computer simulations in MATLAB to investigate the performance of these EM algorithms. We find the modified EM algorithm performs well no matter the true link delays are spatially dependent or not, while the original algorithm fails to estimate the marginal pmfs on links which has dependent delay on that of the previous link.

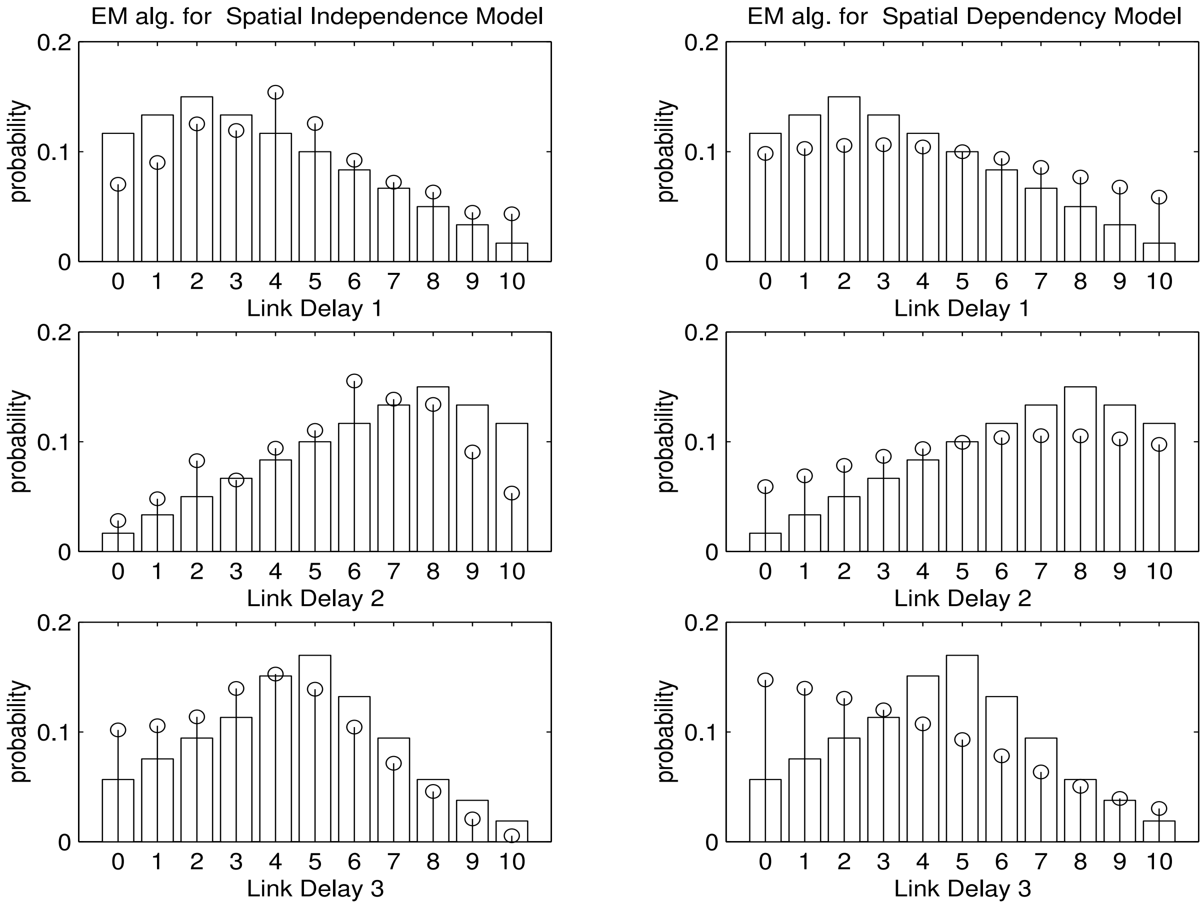

Figure 4: True (bar) and estimated (stem) link delay pmfs in spatially independent delay experiment by using both EM algorithms

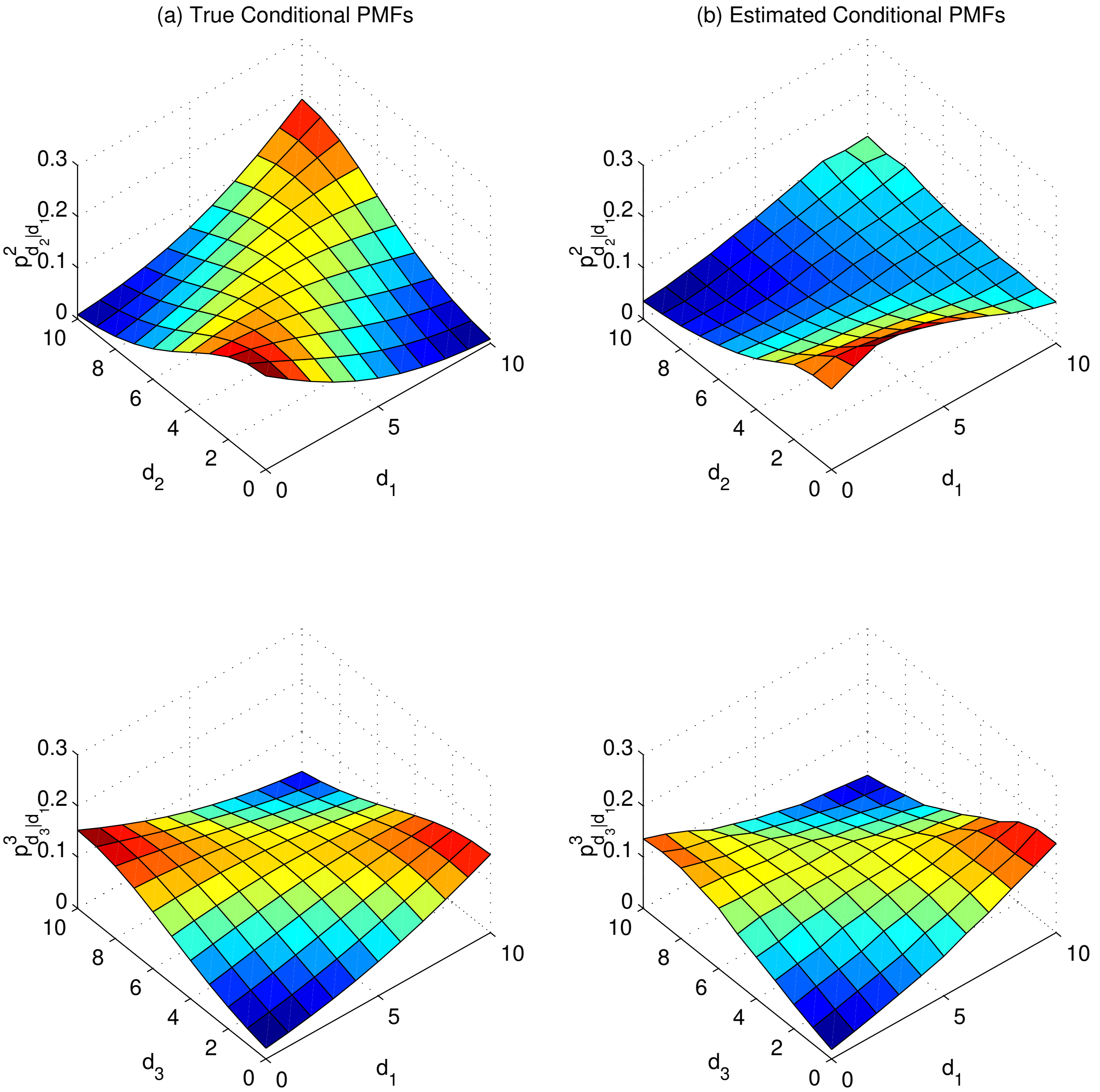

Figure 5: True and estimated conditional link delay pmfs $\{p^2_{d_2,d_1}\}$ and $\{p^3_{d_3,d_1}\}$ in spatially dependent delay experiment

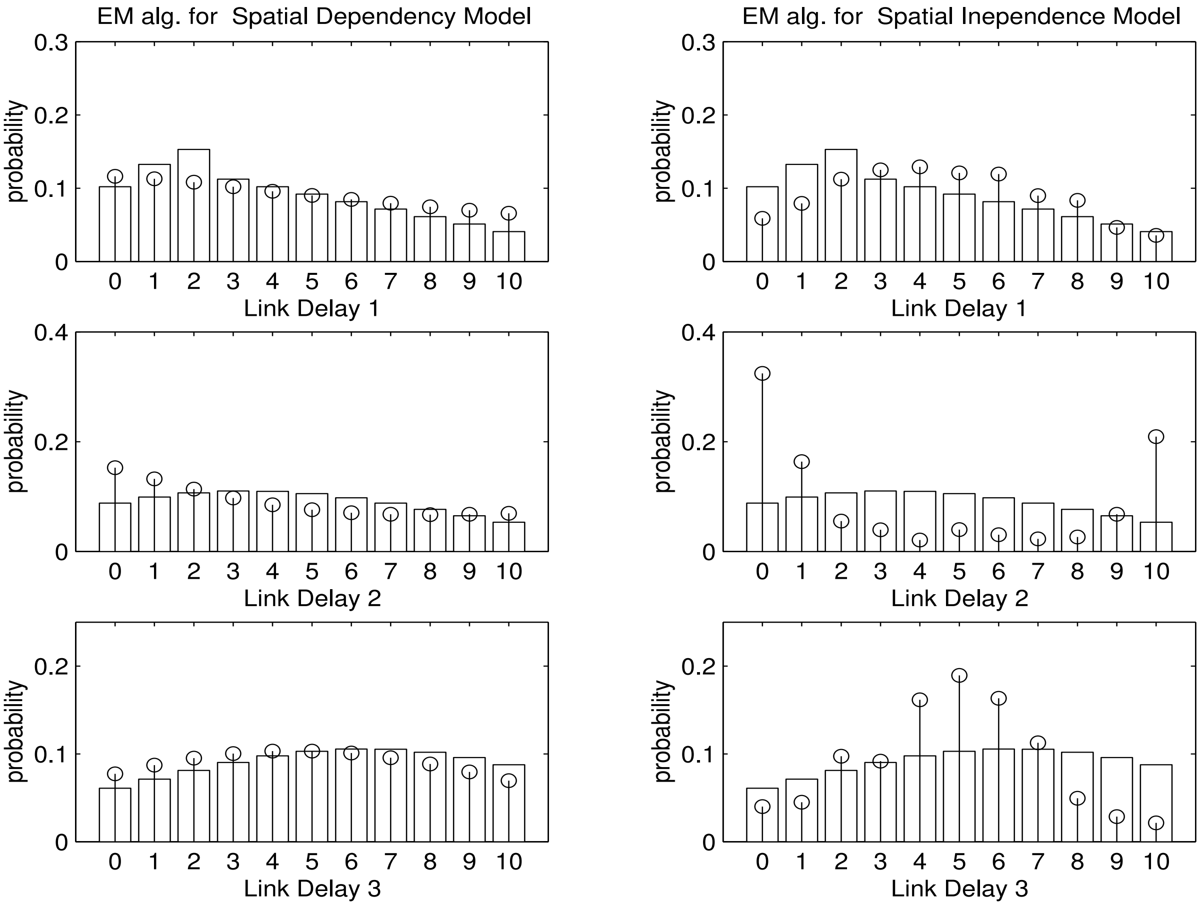

Figure 6: True (bar) and estimated (stem) link delay pmfs in spatially dependent delay experiment by using both EM algorithms

4 Discussion and Future Work

The bias-corrected estimators for link delay CGF using unicast end-to-end measurements are proved to achieve lower MSE than direct biased LS estimators. However, with some type of background traffics, like Pareto On-Off traffics, the proposed estimator may get worse estimates sometimes, i.e., higher MSEs. This means the robustness of the estimator need to be further investigated. As we discussed in chapter 3, internal link delays exhibit spatial correlation. A more sophisticated model must be adopted for link delay CGFs in order to take this into account.

The delay pmf estimation shown in chapter 3 has a major drawback. The bin size used in discretizing delays has to be the same for each internal link. In practice, delays at different links may differ by several orders, which means one fixed bin size may be too fine for some of the links but too coarse for some others. Estimation of delay ranges at each link may be needed before the EM algorithm is performed. In addition, EM algorithm also needs to be modified to allow different bin size for different link. Then `ns` simulations can help us to validate the delay spatial dependency and examine the performance of the proposed EM algorithm in a more realistic environment.

Future works for both proposed schemes in chapter 2 and 3 may include the following:

1) Through `ns` or MATLAB simulations, investigate the rate of convergence of the estimator covariance matrix to the Cramèr-Rao bound.

2) Instead of sending probe packets, these algorithms can also be applied by monitoring existing unicast TCP traffics. Simulations should be conducted to test this validity.

3) Investigate more general setups for delay spatial dependency instead of restricting it to the first order Markov chain case. This may include identification of link delay dependency area within the network and estimation of its intensity.

4) Both chapter 2 and 3 assume temporal independence and stationarity over the probing period, which may be violated in real applications. Some adaptive estimation must be done in order to track the true link delay distributions.

5) Packet loss during observation period shall be taken into account when estimating link delay distributions. A lost packet can be associated with infinite delay.

6) Without discretizing link delays, we also need to explore link delay probability density function (pdf) estimation, which can avoid quantization errors. Unlike chapter 2 in this report, we shall look into tree topologies of the network and deal with underdetermined systems.

7) Our work shall include more network internal statistic estimation and event detection. One good example is the distributed spatio-temporal change detection, which is an important topic in distributed denial-of-service attack prevention.

A The Fisher Information Matrix of End-to-End Packet Delays with Spatial Independence Assumption

A.1 Definitions and Notations

Suppose we have a logical tree network topology. Here we assume the packet delays at one link are independent of those in any other links. The following defines the variables we used in the Fisher information matrix for end-to-end packet delays observed at all receivers.

(1) d^l : Link delay at link l. We discretize the link delay into $D+1$ time units, i.e., $d^l = 0, \cdots, D$.

(2) R : The number of receivers (leaves) in the binary tree network topology. We number those receivers with $r = 1, \cdots, R$.

(3) L : The number of links in the network. We number the links with $l = 1, \cdots, L$. The link connecting the root will always be labeled with 1.

(4) l_r : The number of links in the path from the root to receiver r, $r = 1, \cdots, R$.

(5) L_r : The set of links in the path from the root to receiver r. $L_r = \{e_{r,1}, e_{r,2}, \cdots, e_{r,l_r}\}$, $r = 1, \cdots, R$. Note that $e_{r,1} = 1$ for each L_r.

(6) N_r : The number of packets sent from the root and collected at receiver r, $r = 1, \cdots, R$.

(7) $z_{r,n}$: The end-to-end delay of the n-th packet sent from the root to receiver r, $n = 1, \cdots, N_r$, $r = 1, \cdots, R$. Here we assume the temporal independence, i.e., $\{z_{r,n}\}_{n=1}^{N_r}$ are i.i.d. for a given receiver r.

(8) $\underline{z}$: The vector of observations. $\underline{z} = [z_1, z_2, \cdots, z_R]^T$, where $z_r = [z_{r,1}, \cdots, z_{r,N_r}]$ for $r = 1, \cdots, R$.

(9) p_d^l : The probability mass function(PMF) of packet delay at link l, $d = 0, \cdots, D$, $l = 1, \cdots, L$.

(10) $\underline{\theta}$: The parameter vector. $\underline{\theta} = \left[p_0^1 \cdots p_D^1, p_0^2 \cdots p_D^2, \cdots, p_0^L \cdots p_D^L\right]^T$

(11) $q_r(k)$: The PMF of Z_r, which is the convolution of PMFs of packet delays at links in L_r, i.e.,

$$\begin{aligned} q_r(k) &= \sum_{d_1=max(0,k-(l_r-1)D)}^{min(D,k)} p_{d_1}^1 \times \sum_{d_2=max(0,k-d_1-(l_r-2)D)}^{min(D,k-d_1)} p_{d_2}^{e_{r,2}} \times \cdots \times \\ & \sum_{d_{l_r-1}=max(0,k-d_1-\cdots-d_{l_r-2}-D)}^{min(D,k-d_1-\cdots-d_{l_r-2})} p_{d_{l_r-1}}^{e_{r,l_r-1}} \times p_{k-d_1-\cdots-d_{l_r-1}}^{e_{r,l_r}}, \end{aligned}$$

where $k = 0, \cdots, l_r \times D$.

(12) $q_{r,l}(k)$: The PMF of $(Z_r - d^l)$, $l \in L_r$, which is the convolution of PMFs of packet delays at links in L_r except link l, i.e.,

$$
\begin{aligned}
q_{r,l}(k) \quad = \quad & \sum_{d_1=max(0,k-(l_r-2)D)}^{min(D,k)} p^1_{d_1} \times \sum_{d_2=max(0,k-d_1-(l_r-3)D)}^{min(D,k-d_1)} p^{e_{r,2}}_{d_2} \times \cdots \times \\
& \sum_{d_{i-1}=max(0,k-d_1-\cdots-d_{i-2}-(l_r-i)D)}^{min(D,k-d_1-\cdots-d_{i-2})} p^{e_{r,i-1}}_{d_{i-1}} \times \sum_{d_{i+1}=max(0,k-d_1-\cdots-d_{i-1}-(l_r-i-1)D)}^{min(D,k-d_1-\cdots-d_{i-1})} p^{e_{r,i+1}}_{d_{i+1}} \times \cdots \times \\
& \sum_{d_{l_r-1}=max(0,k-d_1-\cdots-d_{i-1}-d_{i+1}-\cdots-d_{l_r-2}-D)}^{min(D,k-d_1-\cdots-d_{i-1}-d_{i+1}-\cdots-d_{l_r-2})} p^{e_{r,l_r-1}}_{d_{l_r-1}} \cdot p^{e_{r,l_r}}_{k-d_1-\cdots-d_{i-1}-d_{i+1}-\cdots-d_{l_r-1}},
\end{aligned}
$$

where $e_{r,i} = l$ and $k = 0, \cdots, (l_r - 1) \times D$.

(13) $q_{r,l_1,l_2}(k)$: The PMF of $(Z_{r,n} - d^{l_1} - d^{l_2})$, $l_1, l_2 \in L_r$ and $l_1 \neq l_2$, which is the convolution of PMFs of delays at links in L_r except link l_1 and l_2, i.e.,

$$
\begin{aligned}
q_{r,l_1,l_2}(k) \quad = \quad & \sum_{d_1=max(0,k-(l_r-3)D)}^{min(D,k)} p^1_{d_1} \times \sum_{d_2=max(0,k-d_1-(l_r-4)D)}^{min(D,k-d_1)} p^{e_{r,2}}_{d_2} \times \cdots \times \\
& \sum_{d_{i-1}=max(0,k-d_1-\cdots-d_{i-2}-(l_r-i-1)D)}^{min(D,k-d_1-\cdots-d_{i-2})} p^{e_{r,i-1}}_{d_{i-1}} \times \sum_{d_{i+1}=max(0,k-d_1-\cdots-d_{i-1}-(l_r-i-2)D)}^{min(D,k-d_1-\cdots-d_{i-1})} p^{e_{r,i+1}}_{d_{i+1}} \\
& \times \cdots \times \sum_{d_{j-1}=max(0,k-d_1-\cdots-d_{i-1}-d_{i+1}-\cdots-dj-2-(l_r-j)D)}^{min(D,k-d_1-\cdots-d_{i-1}-d_{i+1}-\cdots-d_{j-2})} p^{e_{r,j-1}}_{d_{j-1}} \times \\
& \sum_{d_{j+1}=max(0,k-d_1-\cdots-d_{i-1}-d_{i+1}-\cdots-d_{j-1}-(l_r-j-1)D)}^{min(D,k-d_1-\cdots-d_{i-1}-d_{i+1}-\cdots-d_{j-1})} p^{e_{r,j+1}}_{d_{j+1}} \times \cdots \times \\
& \sum_{d_{l_r-1}=max(0,k-d_1-\cdots-d_{i-1}-d_{i+1}-\cdots-d_{j-1}-d_{j+1}-\cdots-d_{l_r-2}-D)}^{min(D,k-d_1-\cdots-d_{i-1}-d_{i+1}-\cdots-d_{j-1}-d_{j+1}-\cdots-d_{l_r-2})} p^{e_{r,l_r-1}}_{d_{l_r-1}} \times \\
& p^{e_{r,l_r}}_{k-d_1-\cdots-d_{i-1}-d_{i+1}-\cdots-d_{j-1}-d_{j+1}-\cdots-d_{l_r-1}},
\end{aligned}
$$

where $e_{r,i} = l_1, e_{r,j} = l_2$ and $k = 0, \cdots, (l_r - 2) \times D$.

A.2 The Fisher Information Matrix

First we form the likelihood function for the observations:

$$f(\underline{z}; \underline{\theta}) = \prod_{r=1}^{R} \prod_{n=1}^{N_r} q_r(z_{r,n}). \tag{20}$$

Then we can get the log likelihood as follows:

$$\log f(\underline{z};\underline{\theta}) = \sum_{r=1}^{R}\sum_{n=1}^{N_r} \log q_r(z_{r,n}). \tag{21}$$

The Fisher information matrix can be obtained by taking the following expectation:

$$-E\left[\nabla^2_{\underline{\theta}} \sum_{r=1}^{R}\sum_{n=1}^{N_r} \log q_r(z_{r,n})\right] = -\sum_{r=1}^{R} N_r \cdot E\left[\nabla^2_{\underline{\theta}} \log q_r(z_r)\right]. \tag{22}$$

Lets examine the matrix $E\left[\nabla^2_{\underline{\theta}} \log q_r(z_r)\right]$:

(1) If $l \in L_r$,

$$\begin{aligned} E\left[\frac{\partial^2}{\partial p^l_{d_1}\partial p^l_{d_2}} \log q_r(z_r)\right] &= \sum_{k=max(d_1,d_2)}^{min(d_1,d_2)+(l_r-1)D} \left[\frac{\partial^2}{\partial p^l_{d_1}\partial p^l_{d_2}} \log q_r(k)\right] \times q_r(k) \\ &= \sum_{k=max(d_1,d_2)}^{min(d_1,d_2)+(l_r-1)D} -\frac{q_{r,l}(k-d_1)\cdot q_{r,l}(k-d_2)}{q_r(k)}. \end{aligned}$$

(2) If $l_1, l_2 \in L_r$ and $l_1 \neq l_2$,

$$\begin{aligned} E\left[\frac{\partial^2}{\partial p^{l_1}_{d_1}\partial p^{l_2}_{d_2}} \log q_r(z_r)\right] &= \sum_{k=max(d_1,d_2)}^{min(d_1,d_2)+(l_r-1)D} \left[\frac{\partial^2}{\partial p^{l_1}_{d_1}\partial p^{l_2}_{d_2}} \log q_r(k)\right] \times q_r(k) \\ &= \sum_{k=d_1+d_2}^{d_1+d_2+(l_r-2)D} \frac{q_{r,l_1,l_2}(k-d_1-d_2)\cdot q_r(k) - q_{r,l_1}(k-d_1)\cdot q_{r,l_2}(k-d_2)}{q_r(k)} \\ &\quad - \left[\sum_{k=min(d_1,d_2)}^{d_1+d_2-1} + \sum_{d_1+d_2+(l_r-2)D+1}^{min(d_1,d_2)+(l_r-1)D}\right] \frac{q_{r,l_1}(k-d_1)\cdot q_{r,l_2}(k-d_2)}{q_r(k)}. \end{aligned}$$

(3) For other cases, the double partial derivative will be zero and so is the expectation.

B The Fisher Information Matrix of End-to-End Packet Delays with First Order Markov Spatial Dependency Assumption

B.1 Definitions and Notations

Suppose we have a logical tree network topology. Here we assume the delay of a packet at one link depends only on its delay at previous link. It is independent of the delays in any other links. The following defines the variables we used in the Fisher information matrix for end-to-end packet delays observed at all receivers.

(1) d^l : Link delay at link l. We discretize the link delay into $D+1$ time units, i.e., $d^l = 0, \cdots, D$.

(2) R : The number of receivers (leaves) in the binary tree network topology. We number those receivers with $r = 1, \cdots, R$.

(3) L : The number of links in the network. We number the links with $l = 1, \cdots, L$. The link connecting the root will always be labeled with 1.

(4) l_r : The number of links in the path from the root to receiver r, $r = 1, \cdots, R$.

(5) l^F : The parent link of link l.

(6) L_r : The set of links in the path from the root to receiver r in descending order. $L_r = \{e_{r,1}, e_{r,2}, \cdots, e_{r,l_r}\}$, $r = 1, \cdots, R$. Note that $e_{r,1} = 1$ and $e_{r,i-1} = e^F_{r,i}$, $i = 2, \cdots, l_r$ for each L_r.

(7) $A(l)$: The set of "ancestor" links of link l. If $l \in L_r$ and $l = e_{r,i}$, $A(l) = [e_{r,1}, \cdots, e_{r,i-1}]$.

(8) N_r : The number of packets sent from the root and collected at receiver r, $r = 1, \cdots, R$.

(9) $z_{r,n}$: The end-to-end delay of the n-th packet sent from the root to receiver r, $n = 1, \cdots, N_r$, $r = 1, \cdots, R$. Here we assume the temporal independence, i.e., $\{z_{r,n}\}_{n=1}^{N_r}$ are i.i.d. for a given receiver r.

(10) $\underline{z}$: The vector of observations. $\underline{z} = [z_1, z_2, \cdots, z_R]^T$, where $z_r = [z_{r,1}, \cdots, z_{r,N_r}]$ for $r = 1, \cdots, R$.

(11) p^1_d : The probability mass function(PMF) of packet delay at link 1, $d = 0, \cdots, D$.

(12) $p^l_{d,c}$: The conditional probability mass function(PMF) of packet delay at link l given $d^{l^F} = c$, where $d, c = 0, \cdots, D$, and $l = 2, \cdots, L$.

(13) $\underline{\theta}$: The parameter vector. $\underline{\theta} = \left[p^1_0 \cdots p^1_D, p^2_{0,0} \cdots p^2_{D,D}, \cdots, p^L_{0,0} \cdots p^L_{D,D}\right]^T$

(14) $q_r(k)$: The PMF of Z_r, which will be the convolution of PMFs of packet delays at links in L_r, for $r = 1, \cdots, R$, i.e.,

$$\begin{aligned} q_r(k) &= \sum_{d_1=max(0,k-(l_r-1)D)}^{min(D,k)} p^1_{d_1} \times \sum_{d_2=max(0,k-d_1-(l_r-2)D)}^{min(D,k-d_1)} p^{e_{r,2}}_{d_2,d_1} \times \cdots \times \\ & \sum_{d_{l_r-1}=max(0,k-d_1-\cdots-d_{l_r-2}-D)}^{min(D,k-d_1-\cdots-d_{l_r-2})} p^{e_{r,l_r-1}}_{d_{l_r-1},d_{l_r-2}} \times p^{e_{r,l_r}}_{k-d_1-\cdots-d_{l_r-1},d_{l_r-1}}, \end{aligned}$$

where $k = 0, \cdots, l_r \times D$.

(15) $q_{r,1}(k|k_1)$: The conditional PMF of $(Z_{r,n} - d^1)$ given $d^1 = k_1$, which is the convolution of all the conditional PMFs of delays at links in L_r except link 1 given $d^1 = k_1$.

$$\begin{aligned} q_{r,1}(k|k_1) &= \sum_{d_2=max(0,k-(l_r-2)D)}^{min(D,k)} p^{e_{r,2}}_{d_2,k_1} \times \cdots \times \\ & \sum_{d_{l_r-1}=max(0,k-d_2-\cdots-d_{l_r-2}-D)}^{min(D,k-d_2-\cdots-d_{l_r-2})} p^{e_{r,l_r-1}}_{d_{l_r-1},d_{l_r-2}} \times p^{e_{r,l_r}}_{k-d_2-\cdots-d_{l_r-1},d_{l_r-1}}, \end{aligned}$$

where $k = 0, \cdots, (l_r - 1) \times D$.

(16) $q_{r,l}(k|k_1, c_1)$: This is the partial derivative $\frac{\partial}{\partial p^l_{k_1,c_1}} q_r(k + k_1)$, where $l \in L_r.l \neq 1$. It is the convolution of all the (conditional) PMFs of packet delays at links in L_r except link l, given $d^l = k_1$ and $d^{l^F} = c_1$, i.e.,

$$\begin{aligned} q_{r,l}(k|k_1, c_1) &= \sum_{d_1=max(0,k-c_1-(l_r-3)D)}^{min(D,k-c_1)} p^1_{d_1} \times \sum_{d_2=max(0,k-c_1-d_1-(l_r-4)D)}^{min(D,k-c_1-d_1)} p^{e_{r,2}}_{d_2,d_1} \times \cdots \times \\ & \sum_{d_{i-2}=max(0,k-c_1-d_1-\cdots-d_{i-3}-(l_r-i)D)}^{min(D,k-c_1-d_1-\cdots-d_{i-3})} p^{e_{r,i-2}}_{d_{i-2},d_{i-3}} \times p^{e_{r,i-1}}_{c_1,d_{i-2}} \times \\ & \sum_{d_{i+1}=max(0,k-c_1-d_1-\cdots-d_{i-2}-(l_r-i-1)D)}^{min(D,k-c_1-d_1-\cdots-d_{i-2})} p^{e_{r,i+1}}_{d_{i+1},k_1} \times \cdots \times \\ & \sum_{d_{l_r-1}=max(0,k-c_1-d_1-\cdots-d_{i-2}-d_{i+1}-\cdots-d_{l_r-2}-D)}^{min(D,k-c_1-d_1-\cdots-d_{i-2}-d_{i+1}-\cdots-d_{l_r-2})} p^{e_{r,l_r-1}}_{d_{l_r-1},d_{l_r-2}} \times \\ & p^{e_{r,l_r}}_{k-c_1-d_1-\cdots-d_{i-2}-d_{i+1}-\cdots-d_{l_r-1},d_{l_r-1}}, \end{aligned}$$

where $e_{r,i} = l$ and $k = c_1, \cdots, c_1 + (l_r - 2) \times D$.

(17) $q_{r,1,e_{r,2}}(k|k_2)$: This is the double partial derivative $\frac{\partial^2}{\partial p_d^1 \partial p_{k_2,d}^{e_{r,2}}} q_r(k+k_1+d)$, for $d = 0, \cdots, D$. It is the convolution of PMFs of packet delays at links in L_r except link 1 and $e_{r,2}$, given $d^{e_{r,2}} = k_2$, i.e.,

$$
\begin{aligned}
q_{r,1,e_{r,2}}(k|k_2) \;=\; & \sum_{d_3=max(0,k-(l_r-3)D)}^{min(D,k)} p_{d_3,k_2}^{e_{r,3}} \times \cdots \times \\
& \sum_{d_{l_r-1}=max(0,k-d_3-\cdots-d_{l_r-2}-D)}^{min(D,k-d_3-\cdots-d_{l_r-2})} p_{d_{l_r-1},d_{l_r-2}}^{e_{r,l_r-1}} \cdot p_{k-d_3-\cdots-d_{l_r-1},d_{l_r-1}}^{e_{r,l_r}},
\end{aligned}
$$

where $k = 0, \cdots, (l_r-2) \times D$.

(18) $q_{r,1,l}(k|k_1,k_2,c_2)$: This is the double partial derivative $\frac{\partial^2}{\partial p_{k_1}^1 \partial p_{k_2,c_2}^l} q_r(k+k_1+k_2)$, where $l \in L_r, l \neq e_{r,2}$. It is the convolution of all the conditional PMFs of packet delays at links in L_r except link 1 and l given $d^1 = k_1$, $d^l = k_2$ and $d^{l^F} = c_2$, i.e.,

$$
\begin{aligned}
q_{r,1,l}(k|k_1,k_2,c_2) \;=\; & \sum_{d_2=max(0,k-c_2-(l_r-4)D)}^{min(D,k-c_2)} p_{d_2,k_1}^{e_{r,2}} \times \cdots \times \\
& \sum_{d_{i-2}=max(0,k-c_2-d_2-\cdots-d_{i-3}-(l_r-i)D)}^{min(D,k-c_1-d_1-\cdots-d_{i-3})} p_{d_{i-2},d_{i-3}}^{e_{r,i-2}} \times p_{c_2,d_{i-2}}^{e_{r,i-1}} \times \\
& \sum_{d_{i+1}=max(0,k-c_2-d_1-\cdots-d_{i-2}-(l_r-i-1)D)}^{min(D,k-c_2-d_1-\cdots-d_{i-2})} p_{d_{i+1},k_2}^{e_{r,i+1}} \times \cdots \times \\
& \sum_{d_{l_r-1}=max(0,k-c_2-d_1-\cdots-d_{i-2}-d_{i+1}-\cdots-d_{l_r-2}-D)}^{min(D,k-c_2-d_1-\cdots-d_{i-2}-d_{i+1}-\cdots-d_{l_r-2})} p_{d_{l_r-1},d_{l_r-2}}^{e_{r,l_r-1}} \times \\
& p_{k-c_2-d_1-\cdots-d_{i-2}-d_{i+1}-\cdots-d_{l_r-1},d_{l_r-1}}^{e_{r,l_r}},
\end{aligned}
$$

where $e_{r,i} = l$ and $k = c_2, \cdots, c_2 + (l_r-3) \times D$.

(19) $q_{r,l_1,l_2}(k|c_1,k_2)$: This is the double partial derivative $\frac{\partial^2}{\partial p_{d,c_1}^{l_1} \partial p_{k_2,d}^{l_2}} q_r(k+d+k_2)$ for $d = 0, \cdots, D$, where $l_1, l_2 \in L_r, l_1 = l_2^F \neq 1$. It is the convolution of all the (conditional) PMFs of packet delays at links in L_r except link l_1 and l_2 given $d^{l_1^F} = c_1$ and $d^{l_2} = k_2$,

i.e.,

$$
\begin{aligned}
q_{r,l_1,l_2}(k|c_1,k_2) \;=\; & \sum_{d_1=max(0,k-c_1-(l_r-4)D)}^{min(D,k-c_1)} p^1_{d_1} \times \sum_{d_2=max(0,k-c_1-d_1-(l_r-5)D)}^{min(D,k-c_1-d_1)} p^{e_{r,2}}_{d_2,d_1} \times \cdots \times \\
& \sum_{d_{i-2}=max(0,k-c_1-d_1-\cdots-d_{i-3}-(l_r-i-1)D)}^{min(D,k-c_1-d_1-\cdots-d_{i-3})} p^{e_{r,i-2}}_{d_{i-2},d_{i-3}} \times p^{e_{r,i-1}}_{c_1,d_{i-2}} \times \\
& \sum_{d_{i+2}=max(0,k-c_1-d_1-\cdots-d_{i-2}-(l_r-i-2)D)}^{min(D,k-c_1-d_1-\cdots-d_{i-2})} p^{e_{r,i+2}}_{d_{i+2},k_2} \times \cdots \times \\
& \sum_{d_{l_r-1}=max(0,k-c_1-d_1-\cdots-d_{i-2}-d_{i+2}-\cdots-d_{l_r-2}-D)}^{min(D,k-c_1-d_1-\cdots-d_{i-2}-d_{i+2}-\cdots-d_{l_r-2})} p^{e_{r,l_r-1}}_{d_{l_r-1},d_{l_r-2}} \times \\
& p^{e_{r,l_r}}_{k-c_1-d_1-\cdots-d_{i-2}-d_{i+2}-\cdots-d_{l_r-1},d_{l_r-1}},
\end{aligned}
$$

where $e_{r,i} = l_1$, $e_{r,i+1} = l_2$ and $k = c_1, \cdots, c_1 + (l_r - 3) \times D$.

(20) $q_{r,l_1,l_2}(k|k_1,c_1,k_2,c_2)$: This is the partial derivative $\frac{\partial^2}{\partial p^{l_1}_{k_1,c_1} \partial p^{l_2}_{k_2,c_2}} q_r(k+k_1+k_2)$, where $l_1, l_2 \in L_r, l_1 \in A(l_2), l_1 \neq l_2^F, l_1 \neq 1$. It is the convolution of all the (conditional) PMFs of delays at links in L_r except link l_1 and l_2, given $d^{l_1} = k_1$, $d^{l_1^F} = c_1$, $d^{l_2} = k_2$ and $d^{l_2^F} = c_2$, i.e.,

$$
\begin{aligned}
q_{r,l_1,l_2}(k|k_1,c_1,k_2,c_2) \;=\; & \sum_{d_1=max(0,k-c_1-c_2-(l_r-5)D)}^{min(D,k-c_1-c_2)} p^1_{d_1} \times \sum_{d_2=max(0,k-c_1-c_2-d_1-(l_r-6)D)}^{min(D,k-c_1-c_2-d_1)} p^{e_{r,2}}_{d_2,d_1} \times \cdots \times \\
& \sum_{d_{i-2}=max(0,k-c_1-c_2-d_1-\cdots-d_{i-3}-(l_r-i-2)D)}^{min(D,k-c_1-c_2-d_1-\cdots-d_{i-3})} p^{e_{r,i-2}}_{d_{i-2},d_{i-3}} \times p^{e_{r,i-1}}_{c_1,d_{i-2}} \times \\
& \sum_{d_{i+1}=max(0,k-c_1-c_2-d_1-\cdots-d_{i-2}-(l_r-i-3)D)}^{min(D,k-c_1-c_2-d_1-\cdots-d_{i-2})} p^{e_{r,i+1}}_{d_{i+1},k_1} \times \cdots \times \\
& \sum_{d_{j-2}=max(0,k-c_1-c_2-d_1-\cdots-d_{i-2}-d_{i+1}-\cdots-dj-3-(l_r-j)D)}^{min(D,k-c_1-c_2-d_1-\cdots-d_{i-2}-d_{i+1}-\cdots-d_{j-3})} p^{e_{r,j-2}}_{d_{j-2},d_{j-3}} \times p^{e_{r,j-1}}_{c_2,d_{j-2}} \times \\
& \sum_{d_{j+1}=max(0,k-c_1-c_2-d_1-\cdots-d_{i-2}-d_{i+1}-\cdots-d_{j-2}-(l_r-j-1)D)}^{min(D,k-c_1-c_2-d_1-\cdots-d_{i-2}-d_{i+1}-\cdots-d_{j-1})} p^{e_{r,j+1}}_{d_{j+1},k_2} \times \cdots \times \\
& \sum_{d_{l_r-1}=max(0,k-c_1-c_2-d_1-\cdots-d_{i-2}-d_{i+1}-\cdots-d_{j-2}-d_{j+1}-\cdots-d_{l_r-2}-D)}^{min(D,k-c_1-c_2-d_1-\cdots-d_{i-2}-d_{i+1}-\cdots-d_{j-2}-d_{j+1}-\cdots-d_{l_r-2})} p^{e_{r,l_r-1}}_{d_{l_r-1},d_{l_r-2}} \times \\
& p^{e_{r,l_r}}_{k-c_1-c_2-d_1-\cdots-d_{i-2}-d_{i+1}-\cdots-d_{j-2}-d_{j+1}-\cdots-d_{l_r-1},d_{l_r-1}},
\end{aligned}
$$

where $e_{r,i} = l_1, e_{r,j} = l_2$ and $k = c_1 + c_2, \cdots, c_1 + c_2 + (l_r - 4) \times D$.

B.2 The Fisher Information Matrix

First we form the likelihood function for the observations:

$$f(\underline{z};\underline{\theta}) = \prod_{r=1}^{R}\prod_{n=1}^{N_r} q_r(z_{r,n}). \tag{23}$$

Then we can get the log likelihood as follows:

$$\log f(\underline{z};\underline{\theta}) = \sum_{r=1}^{R}\sum_{n=1}^{N_r} \log q_r(z_{r,n}). \tag{24}$$

The Fisher information matrix can be obtained by taking the following expectation:

$$-E\left[\nabla^2_{\underline{\theta}} \sum_{r=1}^{R}\sum_{n=1}^{N_r} \log q_r(z_{r,n})\right] = -\sum_{r=1}^{R} N_r \cdot E\left[\nabla^2_{\underline{\theta}} \log q_r(z_r)\right]. \tag{25}$$

Lets examine the matrix $E\left[\nabla^2_{\underline{\theta}} \log q_r(z_r)\right]$:

(1) For each z_r, link 1 is always included in the path. Hence

$$\begin{aligned} E\left[\frac{\partial^2}{\partial p^1_{d_1}\partial p^1_{d_2}} \log q_r(z_r)\right] &= \sum_{k=max(d_1,d_2)}^{min(d_1,d_2)+(l_r-1)D} \left[\frac{\partial^2}{\partial p^1_{d_1}\partial p^1_{d_2}} \log q_r(k)\right] \times q_r(k) \\ &= \sum_{k=max(d_1,d_2)}^{min(d_1,d_2)+(l_r-1)D} -\frac{q_{r,1}(k-d_1|d_1)\cdot q_{r,1}(k-d_2|d_2)}{q_r(k)}. \end{aligned}$$

(2) If $l \in L_r, l \neq 1$,

$$\begin{aligned} E\left[\frac{\partial^2}{\partial p^l_{d_1,c_1}\partial p^l_{d_2,c_2}} \log q_r(z_r)\right] &= \sum_{k=max(d_1+c_1,d_2+c_2)}^{min(d_1+c_1,d_2+c_2)+(l_r-2)D} \left[\frac{\partial^2}{\partial p^l_{d_1,c_1}\partial p^l_{d_2,c_2}} \log q_r(k)\right] \times q_r(k) \\ &= \sum_{k=max(d_1+c_1,d_2+c_2)}^{min(d_1+c_1,d_2+c_2)+(l_r-2)D} -\frac{q_{r,l}(k-d_1|d_1,c_1)\cdot q_{r,l}(k-d_2|d_2,c_2)}{q_r(k)}. \end{aligned}$$

(3) For each z_r,

$$E\left[\frac{\partial^2}{\partial p^1_{d_1}\partial p^{e_{r,2}}_{d_2,c}} \log q_r(z_r)\right] = \sum_{k=max(d_1,d_2+c)}^{min(d_1+D,d_2+c)+(l_r-2)D} \left[\frac{\partial^2}{\partial p^1_{d_1}\partial p^{e_{r,2}}_{d_2,c}} \log q_r(k)\right] \times q_r(k) =$$

$$\begin{cases} \sum_{k=max(d_1,d_2+c)}^{min(d_1+D,d_2+c)+(l_r-2)D} -\frac{q_{r,1}(k-d_1|d_1)\cdot q_{r,e_{r,2}}(k-d_2|d_2,c)}{q_r(k)}, & \text{for } c \neq d_1 \\ \sum_{k=d_1+d_2}^{d_1+d_2+(l_r-2)D} \frac{q_{r,1,e_{r,2}}(k-d_1-d_2|d_2)\cdot q_r(k)}{q_r(k)} - \sum_{k=max(d_1,d_2+c)}^{min(d_1+D,d_2+c)+(l_r-2)D} \frac{q_{r,1}(k-d_1|d_1)\cdot q_{r,e_{r,2}}(k-d_2|d_2,d_1)}{q_r(k)}, \\ \text{for } c = d_1 \end{cases}$$

(4) If $l \in L_r, l \neq 1$ and $l \neq e_{r,2}$,

$$\begin{aligned}
E\left[\frac{\partial^2}{\partial p^1_{d_1} \partial p^l_{d_2,c}} \log q_r(z_r)\right] &= \sum_{k=max(d_1,d_2+c)}^{min(d_1+D,d_2+c)+(l_r-2)D} \left[\frac{\partial^2}{\partial p^1_{d_1} \partial p^l_{d_2,c}} \log q_r(k)\right] \times q_r(k) \\
&= \sum_{k=d_1+d_2+c}^{d_1+d_2+c+(l_r-3)D} \frac{q_{r,1,l}(k-d_1-d_2|d_1,d_2,c)\cdot q_r(k)}{q_r(k)} - \\
&\quad \sum_{k=max(d_1,d_2+c)}^{min(d_1+D,d_2+c)+(l_r-2)D} \frac{q_{r,1}(k-d_1|d_1)\cdot q_{r,l}(k-d_2|d_2,c)}{q_r(k)},
\end{aligned}$$

(5) If $l_1, l_2 \in L_r$ and $l_1 = l_2^F \neq 1$

$$E\left[\frac{\partial^2}{\partial p^{l_1}_{d_1,c_1} \partial p^{l_2}_{d_2,c_2}} \log q_r(z_r)\right] = \sum_{k=max(d_1+c_1,d_2+c_2)}^{min(d_1+c_1,d_2+c_2)+(l_r-2)D} \left[\frac{\partial^2}{\partial p^{l_1}_{d_1,c_1} \partial p^{l_2}_{d_2,c_2}} \log q_r(k)\right] \times q_r(k) =$$

$$\begin{cases} \sum_{k=max(d_1+c_1,d_2+c_2)}^{min(d_1+c_1,d_2+c_2)+(l_r-2)D} -\frac{q_{r,l_1}(k-d_1|d_1,c_1)\cdot q_{r,l_2}(k-d_2|d_2,c_2)}{q_r(k)}, \qquad \text{for } c_2 \neq d_1 \\ \sum_{k=d_1+c_1+d_2}^{d_1+c_1+d_2+(l_r-3)D} \frac{q_{r,l_1,l_2}(k-d_1-d_2|c_1,d_2)\cdot q_r(k)}{q_r(k)} - \sum_{k=max(d_1+c_1,d_2+c_2)}^{min(d_1+c_1,d_2+c_2)+(l_r-2)D} \frac{q_{r,l_1}(k-d_1|d_1,c_1)\cdot q_{r,l_2}(k-d_2|d_2,d_1)}{q_r(k)}, \\ \text{for } c_2 = d_1 \end{cases}$$

(6) If $l_1, l_2 \in L_r, l_1 \in A(l_2), l_1 \neq l_2^F$ and $l_1 \neq 1$,

$$\begin{aligned}
E\left[\frac{\partial^2}{\partial p^{l_1}_{d_1,c_1} \partial p^{l_2}_{d_2,c_2}} \log q_r(z_r)\right] &= \sum_{k=max(d_1+c_1,d_2+c_2)}^{min(d_1+c_1,d_2+c_2)+(l_r-2)D} \left[\frac{\partial^2}{\partial p^{l_1}_{d_1,c_1} \partial p^{l_2}_{d_2,c_2}} \log q_r(k)\right] \times q_r(k) \\
&= \sum_{k=d_1+c_1+d_2+c_2}^{d_1+c_1+d_2+c_2+(l_r-4)D} \frac{q_{r,l_1,l_2}(k-d_1-d_2|d_1,c_1,d_2,c_2)\cdot q_r(k)}{q_r(k)} - \\
&\quad \sum_{k=max(d_1+c_1,d_2+c_2)}^{min(d_1+c_1,d_2+c_2)+(l_r-2)D} \frac{q_{r,l_1}(k-d_1|d_1,c_1)\cdot q_{r,l_2}(k-d_2|d_2,c_2)}{q_r(k)},
\end{aligned}$$

(7) For other cases, the double partial derivative will be zero and so is the expectation.

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] J. Cao, D. Davis, S.V. Wiel, B. Yu, "Time-varing network tomography: router link data," *lucent Bell Lab Technical Memo*, Feb. 2000. `http://plan9.bell-labs.com/cm/ms/departments /sia/cao/htmls/pub.html`

[3] M. Coates and R. Nowak, "Network inference from passive unicast measurements," Technical Report TR-0002, Rice Univ., ECE Dept., Jan. 2000.

[4] Y. Tsang, M. Coates and R. Nowak, "Passive unicast network tomography based on TCP monitoring," Technical Report TR-0005, Rice Univ., ECE Dept., Nov. 2000.

[5] C. Tebaldi and M. West, "Bayesian inference on network traffic using link count data," *J.Amer.Statist.Assoc.*, vol.93, no.442, pp.557-573, June 1998.

[6] Multicast-based inference of network-internal characteristics (MINC). See gaia.cs.umass.edu/minc.

[7] 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, pp.2462 - 2480, Nov. 1999.

[8] N.G. Duffield, F.F. Presti, "Multicast inference of packet delay variance at interior network links," *Proceedings of INFOCOM'00.*

[9] F.L. Presti, N.G. Duffield, J. Horowitz, D. Towsley, "Multicast-based inference of network-internal delay distributions," UMass Computer Science TR99-55, Nov. 1999.

[10] R. Caceres, N.G. Duffield, J. Horowitz, D. Towsley and T. Bu, "Multicast-Based Inference of Network-Internal Characteristics: Accuracy of Packet Loss Estimation," *Proceedings of INFOCOM'99.*

[11] R. Caceres, N.G. Duffield, S.B. Moon, D. Towsley, "Inferring Link-level Performance from End-to-End Multicast Measurements," *Proceedings of CDC'99.*

[12] R. Caceres, N.G. Duffield, S.B. Moon, D. Towsley, "Inferring Link-level Performance from End-to-End Multicast Measurements," *Proc. IEEE/ISOC Global Internet '99*, December 1999.

[13] A. Adams, T. Bu, R. Caceres, N. Duffield, T.Friedman, J. Horowitz, F. Lo Presti, S.B. Moon, V. Paxson, D. Towsley, "The Use of End-to-end Multicast Measurements for Characterizing Internal Network Behavior." *IEEE Communications Magazine*, May 2000.

[14] A. Ziotopoulos, A. Hero, W. Wasserman, "Estimation of Network Link Loss Rates Via Chaining In Multicast Trees," *ICASSP 2001*, to appear.

[15] N. Duffield, F. Presti, V. Paxson, D. Towsley, "Inferring link loss using striped unicast probes," *Proceedings of INFOCOM'01*, to appear.

[16] Jacobson. "`pathchar`-a tool to infer characteristics of Internet paths," Presented at the *Mathematical Sciences Research Institute*(MSRI), Apr. 1997.

[17] A. Downey, "Using `pathchar` to estimate Internet link characteristics," Proceedings of *ACM SIGCOMM '99*.

[18] K. Lai, M. Baker, "Measuring link bandwidths using a deterministic model of packet delay" Proceedings of *ACM SIGCOMM 2000*.

[19] InterNet Control and Inference Tools at the Edge (INCITE). See www.dsp.rice.edu/incite.

[20] M. Coates and R. Nowak, "Network loss inference using unicast end-to-end measurement," In *Proc. ITC Seminar on IP Traffic, Measurement and Modelling*, pp.28-1-28-9, Monterey, CA, Sep. 2000.

[21] M. Coates and R. Nowak, "Unicast network tomography using EM algorithms," Technical Report TR-0004, Rice University, Sep. 2000.

[22] M. Coates and R. Nowak, "Network tomography for internal delay estimation," *ICASSP 2001*, to appear.

[23] M. Shih and A. Hero, "Unicast inference of network link delay distributions from edge measurements," *ICASSP 2001*, to appear.

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

[25] R.J. Gibbens, "Traffic characterisation and effective bandwidths for broadband network traces," *Stochastic Networks, Theory and Applications*, pp. 169-179, Oxford Science Pub., 1996.

[26] B. Efron, "The Jackknife, the bootstrap and other resampling plans," *Society for Industrial and Applied Math*, 1982.