

# Local Area Network Analysis Using End-to-End Delay Tomography

Earl Lawrence<sup>1</sup>

George Michailidis and Vijay N. Nair<sup>2 3</sup>

## Abstract

There has been considerable interest over the last few years in collecting and analyzing internet traffic data in order to estimate quality of service parameters such as packet loss rates and delay distributions. In this paper, we focus on fast and efficient estimation methods for network link delay distributions based on end-to-end measurements obtained by probing the underlying. We introduce a rigorous statistical framework for designing the necessary probing experiments and examine the properties of the proposed estimators. The proposed framework and the resulting methodology are validated using data collected on the University of North Carolina (UNC) local area network.

## 1 Introduction and Motivation

Over the last decade, computer networks have experienced an exponential growth in terms of the number of users, the amount of traffic, and the number and complexity of the applications. Another important feature of such networks is their multi-layered structure and the lack of centralized control; the latter, has enabled service providers to develop and offer a rich variety of applications and services at different quality-of-service levels. On the other hand, the decentralized nature of the environment makes it very difficult to assess network performance. Furthermore, traditional queuing and traffic models do a poor job of capturing the complexity and characteristics of network behavior. This has led to the emergence of *network tomography* – an area that uses *active* and *passive* traffic measurement schemes to quantify the performance of large-scale networks. An increasing body of literature has appeared that deals with tomography related problems [1, 2, 3, 4, 5, 6].

In many situations, it is impractical to monitor a large num-

ber of links directly, due to the fact that it may affect network performance and the difficulty of handling the massive amounts of generated data. Furthermore, if the network of interest spans several subnets, assistance from one provider is not enough. For these reasons, active network probing offers a good alternative. This means injecting small amounts of traffic on a network at easily accessible endpoints (sources) and monitoring its end-to-end performance as it travels to other endpoints (receivers). These active measurement schemes allow one to monitor network performance characteristics, while at the same time minimally interfering with the network. The challenge is to *deconvolve* the end-to-end information to make inferences about individual links in the network.

In this paper, we introduce a discrete statistical model for estimating link delay distributions based on active probing of the network. We consider several techniques for fitting this model. In particular, we focus on the following two issues: (i) conditions required for the probing experiments that guarantee a *unique solution* to the deconvolution problem and (ii) fast estimation techniques that allow frequent monitoring in short intervals and that also can estimate the probabilities of large link delays. Our main goal will be to evaluate the model and estimation scheme using real data collected at the UNC using Voice-over-IP equipment.

## 2 Framework

In this section, we discuss the modeling framework used for solving the active delay tomography problem. Specifically, we introduce the logical topology used for conducting a tomography experiment, the probing schemes employed for obtaining the path level delay information, and the stochastic assumptions underlying the obtained measurements. For ease of presentation, we focus on single source tomography experiments, although the proposed framework can easily handle multi-source experiments.

### 2.1 Topology

We describe a network as a graph with computers, routers, and hubs serving as nodes and the links between them serving as edges. Because of our focus on active probing from a single source, we focus exclusively on representing the network as a *tree*: a directed, acyclic graph in which each node except the source has a single parent. See Figure 1 for an example. Formally, let  $\mathcal{T} = \{\mathcal{V}, \mathcal{E}\}$  be a tree with node set  $\mathcal{V}$  and link set  $\mathcal{E}$ . The nodes will follow a canonical numbering scheme starting from the root node 0. Each link will be named after

<sup>1</sup>Statistical Sciences Group, Los Alamos National Laboratory, Los Alamos NM 87545. earl@lanl.gov

<sup>2</sup>Department of Statistics, University of Michigan, Ann Arbor MI 48109. {gmichail, vnn}@umich.edu

<sup>3</sup>This research was supported in part by NSF grants CCR-0325571 and DMS-0204247. The authors would like to thank: Jim Landwehr, Lorraine Denby, and Jean Meloche of Avaya Labs for making their ExpertNet tool available for VoIP data collection and for many useful discussions on network monitoring; Yinghan Yang for assistance with data collection; Jim Gogan and his team from the IT Division at UNC for their technical support in deploying the ExpertNet tool on their campus network, for troubleshooting hardware problems and for providing information about the structure and topology of the network; Don Smith of the CS Department at UNC for helping us establish the collaboration with the UNC IT group; and Steve Marron for many helpful comments during the course of this research.

its terminal node.

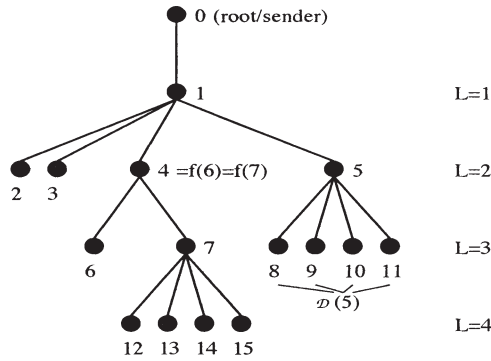


Figure 1: Tree network with notation.

## 2.2 Experimentation

The data collection is based upon *flexicast* probing [2]. This methodology derives from the desire to combine the identifiability guaranteed by multicast schemes with the more simplistic data and algorithm complexity arising in smaller experiments. The main idea is to probe the receiver set in groups. In this paradigm, multicast schemes are a special case. A flexicast scheme consists of experiments involving groups of receivers of different sizes. Each experiment consists of a series of probes simultaneously sent to each member of the group using multicast probing or back-to-back unicast probing. The observed data for an experiment with  $k$  receivers are a set of  $k$ -tuples of end-to-end delays where each end-to-end delay is the sum of the delays occurring on each link. The simultaneous probing induces correlations in the end-to-end measurements due to the portion of the delay that results from the shared path.

## 2.3 Stochastic Assumptions

Consider the traffic traces in Figure 2. One of the striking features of the data is its heavy-tailed nature, as demonstrated in the corresponding CCDF plot 2. This feature is often seen in other types of Internet related traffic such as packet counts or flow durations. Since delay is linked to packet counts, the heavy-tailed nature of the distribution does not come as a surprise. Nevertheless, this must be taken into account when choosing an appropriate model. This type of behavior is consistent across different links and times. Links and times may exhibit different amounts and magnitudes of ‘spikiness,’ but the overall pattern is very consistent. Our stochastic model, introduced in [1], will be based on discretizing the continuous delays based on a common bin size  $q$ . Let  $X_k \in \{0, q, 2q, \dots, bq\}$  be the discretized delay accumulated on link  $k$  where  $b$  is some maximum (for the remainder, we will drop the notational dependence on  $q$ ). Let  $\alpha_k(i) = P\{X_k = i\}$ . We make two independence assumptions. First, we assume that consecutive probes are independent of each other. Second, we assume that the delay a probe experiences along one link is independent of the delay it experiences along any other link. This model makes no shape assumptions making it ideal for the type of data that we see.

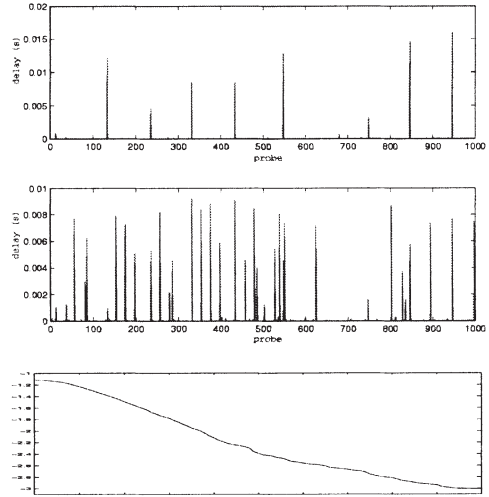


Figure 2: Traffic traces of end-to-end delay collected on the UNC LAN and a corresponding CCDF plot.

Despite this nonparametric component, each link delay still follows a multinomial distribution, which in turn allows standard estimation tools to be extended in a fairly straightforward manner.

The actual observation made on some receiver node  $r$  is  $Y_r = X_1 + \dots + X_r$ . This is the sum of the link delays along the path from the source to the receiver. The goal is the estimate  $\bar{\alpha}$  based upon these end-to-end measurements.

## 2.4 Identifiability

With our modeling framework set, we can turn to our first result which describes how to design probing experiments on a tree in order to guarantee the estimation of the discrete delay model.

**Proposition 1** *Let  $\mathcal{C}$  be a collection of probing experiments and  $\mathcal{T}$  be a general tree network topology. For every internal node  $s \in \mathcal{T}$ , let there be at least one  $c \in \mathcal{C}$  such that  $c$  probes at least two receivers and  $s$  is a branching node for  $c$ . For each receiver  $r$ , let there be at least one  $c \in \mathcal{C}$  such that  $c$  covers  $r$ . These conditions are necessary and sufficient to guarantee that  $\mathcal{C}$  identifies the discrete delay distribution for every link in  $\mathcal{T}$ .*

A detailed proof is given in [2]. The key concept is that the grouped probing leads to correlated end-to-end results. The correlations provided information about the shared paths. By using all internal nodes as branching points, we get shared-path information from the source to every internal node.

## 3 Estimation Methodology

This section will describe procedures for fitting the above model. We will start with a brief description of maximum

likelihood estimation (MLE) and then move on to two extensions of this procedure: local MLEs and large delay estimation.

### 3.1 Maximum Likelihood

The discrete, nonparametric modeling framework results in multinomial outcomes for the path-level experiments. Our observations consist of the number of times that each individual outcome  $\vec{y}$  was observed from the set of outcomes  $\mathcal{Y}^c$  for a given experiment. We denote these counts  $N_{\vec{y}}^c$ . Consider the likelihood equation:

$$l(\vec{\alpha}; \mathbf{Y}) = \sum_{c \in \mathcal{C}} \sum_{\vec{y} \in \mathcal{Y}^c} N_{\vec{y}}^c \log[\gamma_c(\vec{y}; \vec{\alpha})], \quad (1)$$

where  $\gamma_c(\vec{y}; \vec{\alpha})$  is a path-level probability. This equation is difficult to maximize directly. However, due to the inverse nature of the problem, it corresponds to a missing data problem; if the counts for the unobserved link delays were known, the maximization would be fairly straightforward as the link outcomes are also simple multinomial experiments. As a result, the expectation-maximization (EM) algorithm is a natural candidate in the present setting for computing the maximum likelihood estimates. Only the the sufficient statistics for each link need to be computed: the counts for the number of times that  $X_k$  took on each value. The E-step can be broken into two parts. Assume that we have some parameter vector  $\vec{\alpha}^{(z-1)}$ . First, we compute the expected number of times each link delay vector,  $\vec{x}$ , occurred.

$$N_{\vec{x}}^c(z) = \frac{P\{\vec{X}^c = \vec{x}\}^{(z-1)}}{P\{\vec{Y}^c = \vec{y}(\vec{x})\}^{(z-1)}} N_{\vec{y}}^c \quad (2)$$

Then, we use these values to compute the number of times that probes on link  $k$  had a delay of  $i$  units.

$$M_{k,i}^{(z)} = \sum_{c \in \mathcal{C}: k \in \mathcal{T}^c} \sum_{\vec{x} \in \mathcal{X}^c: x_k = i} N_{\vec{x}}^c(z) \quad (3)$$

We also need to keep track of  $m_k$  which is the total number of probes that crossed link  $k$ . The M-step is quite simple once the sufficient statistics have been imputed.

$$\alpha_k(i)^{(z)} = \frac{1}{m_k} M_{k,i}^{(z)} \quad (4)$$

The computationally challenging aspect in our setting is to partition the observed end-to-end delays into the set of possible link delay combinations. The details of this procedure are covered in [2]. We note that this differs from previous estimators in two ways. First, the estimator is built for the flexicast probing of which multicast is a special case. Further, the maximum likelihood approach has advantages with regard to efficiency. Thus the MLE presented here results in more precise estimates for multicast data than previous procedures.

The maximum likelihood estimates can be shown to possess all the desirable statistical properties; namely, they are strongly consistent, asymptotically normal and fully efficient.

We consider next the complexity of an EM iteration. We start by looking at a single experiment. There are  $b^{|\mathcal{T}^c|}$  link delay outcomes for this experiment. For each of these, there are  $|\mathcal{T}^c|$  multiplications to compute the probability of the link delay outcome. There is also a single addition to tally up the end-to-end probabilities and a single division to compute the conditional probability of each outcome given the end-to-end outcome. Finally, there are  $|\mathcal{T}^c|$  additions to tally up the sufficient statistics. Overall, this gives us  $\mathcal{O}\{b^{|\mathcal{T}^c|}\}$  operations. The largest subtree sets the complexity for the E-step at  $\mathcal{O}\{b^{|\mathcal{T}^l|}\}$  where  $l$  is the largest experiment. The M-step is trivial consisting of  $|\mathcal{E}b|$  divisions.

Given the exponential complexity of this algorithm, another approach is desirable, especially when one is interested in monitoring the quality of several links in real time. In the next subsection, we consider a faster algorithm based upon estimating the delay distributions through maximum likelihood of appropriately chosen sub-trees and then combining the results.

### 3.2 Grafting

We consider an alternative to the full EM: computing the local MLE on the subtrees and combining the results in a process called peeling (so called because we will use a path-level distribution and the distribution of a subpath to solve for, or peel away, the distribution for the other subpath). We call this combination of trees grafting. In essence, we treat each experiment as a multicast experiment on the probing subtree. We use the EM algorithm to solve for the MLE of the logical links on this subtree and then peel to get estimates for individual links. For collections of bicast (where probing packets are sent to two receiver nodes) and unicast (one receiver node) experiments, this scheme scales very well because the EM algorithm is applied to a series of three-link, two-layer trees. Based on numerical experiments considered in [2], increasing the number of bins on a the links increases the average iterations approximately linearly while adding links increases the required iterations exponentially. This local scheme takes advantage of this fact by trading links for bins.

We will explain the details using just bicast and unicast experiments. First, consider a bicast experiment and the subtree that it probes. Let the trunk have  $t$  links and the branches have  $l_1$  and  $l_2$  links respectively. The subtree has just three logical links with varying numbers of bins on each: the trunk has  $t(b+1)$  bins and the branches have  $l_1(b+1)$  and  $l_2(b+1)$  bins. We apply the EM algorithm to this logical subtree and solve for its MLE. This is done for all of the experiments.

Individual links can be identified in several ways. We will discuss top-down peeling. Because the same identifiability conditions apply, at least one pair must split at node 1 and at least one of the local MLEs must give us an estimate for link 1. At least one experiment gives us the local MLE for the path from the root node to every child of node 1. The individual links can then be identified through peeling. This process continues down the tree identifying each link. The re-

ceivers covered by bicast experiments can be identified as the branches in a subtree or by peeling from the branches. The receivers covered by only unicast experiments can be identified by peeling.

For this scheme, we favor a more sophisticated peeling mechanism than previously discussed. A better fixed-point type algorithm arises from postulating an EM algorithm using imaginary data. Imagine that we send  $n$  probes across the path. Form data by setting  $n_d = n\pi_{0,2}(d)$ . The data are counts of the number of times delay  $d$  was observed on the path for all possible  $d$ . In the E step, we want to compute  $M_i$ , the expected number of times that delay  $i$  was seen on the unknown link. After the  $z$ -th iteration, this is given by:

$$M_i^{(z+1)} = \sum_{j=0}^b \frac{\alpha_2^{(z)}(i)\alpha_1(j)}{\pi_{0,2}^{(z)}(i+j)} n_{i+j}, \quad (5)$$

where  $\bar{\pi}_{0,2}^{(z)}$  is updated with each update of  $\bar{\alpha}_2^{(z)}$ . Note that this is not the quantity used to generate the data. Since we obtain our estimates by dividing  $M_i$  by  $n$ , we get the following equation:

$$\alpha_2(i)^{(z+1)} = \sum_{j=0}^b \frac{\alpha_2^{(z)}(i)\alpha_1(j)}{\pi_{0,2}^{(z)}(i+j)} \pi_{0,2}(i+j). \quad (6)$$

This equation is no longer based on our imaginary data and can be run until  $\bar{\alpha}_2$  approaches its fixed point. Unlike the previously mentioned scheme, this peeling function uses all of the information from the two known distributions.

If multiple estimates are available for a link delay distribution, they can be combined in various ways. Simple averaging is one. A better method is weighted averaging based on the number of observations. Thus, if we have two estimates of  $\bar{\alpha}_1$  from experiments  $c_1$  and  $c_2$ , we can combine them to get

$$\bar{\alpha}_1 = \frac{n^{c_1}\bar{\alpha}^{c_1} + n^{c_2}\bar{\alpha}^{c_2}}{n^{c_1} + n^{c_2}} \quad (7)$$

The grafting procedure also has several desirable asymptotic properties. It is consistent and asymptotically normal, but it is not as efficient as the complete MLE. A more thorough treatment of both the MLE and the grafting procedure is contained in [2]. This paper also includes comparisons with other work.

### 3.3 Large Delay Estimation

Among the advantages of the discrete delay model and its estimation schemes is the ability to quickly estimate the probability of large delays. This information is very important for capacity planning and performance estimation for applications like Voice-Over-IP (VoIP) telephony. By setting the bin size  $q$  to be some large delay of interest, such as a detrimental threshold for VoIP quality, this framework can be easily used to estimate the probability of a deteriorating quality of service on each link. This model can usually be quickly estimated since most of the mass is expected to be in the first

(or zero) bin. This process is equivalent in some ways to tail estimation.

In addition to solving a specific problem, using this procedure improves speed in two ways. First, it reduces the empirical estimate of the max delay  $b$ : larger bin sizes lead to fewer bins. Additionally, this procedure takes advantage of the above complexity calculation. The complexity of the EM algorithm in the worst-case scenario is based upon observing at least one value for every possible end-to-end delay. Using a large bin size collapses the observations. For example, under a certain bin size, the bulk of observations at some pair might range from (0,0) to (5,5) thus requiring the estimation scheme to operate on 36 bins. Increasing the bin size by a factor of six collapses all of those bins into a single (0,0) observation while likely leaving the extreme delay from the tails as individual observations. Thus, we obtain significant computational gains.

## 4 Design of Probing Experiments

The importance of the identifiability condition previously mentioned is that it provides us with a set of conditions to *construct* a collection of probing experiments that *provably* identifies the individual link delay distributions. We describe next such a procedure, based on the set covering problem. For ease of presentation we restrict attention to a collection comprised of bicast and unicast schemes. Let  $\mathcal{B}$  and  $\mathcal{U}$  denote the set of *all* possible bicast and unicast schemes, respectively, for a given topology  $\mathcal{T}$  with edge set  $\mathcal{E}$ . Every scheme  $\delta \in \mathcal{B} \cup \mathcal{U}$  can be represented by a vector  $x^\delta$  of length  $|\mathcal{V}| - 1$  with elements  $x_i^\delta \in \{0, 1\}$ , where a 1 indicates that the scheme traverses node  $i$  and a 0 that it does not. For example, the corresponding  $x$  vector for the bicast scheme  $\langle 2, 3 \rangle$  for the topology shown in Figure 1 is

$$x^{\langle 2,3 \rangle} = [1110000000000000]'$$

We can then posit the following integer program, with decision variables  $\xi$

$$\text{minimize} \quad \sum_{\delta \in \mathcal{B} \cup \mathcal{U}} \xi^\delta \quad (8)$$

$$\text{s.t.} \quad \sum_{\delta \in \mathcal{B}} x_j^\delta \xi^\delta \geq 1, \quad j \in \mathcal{I} \quad (9)$$

$$\sum_{\delta \in \mathcal{B} \cup \mathcal{U}} x_j^\delta \xi^\delta \geq \kappa, \quad j \in \mathcal{V} - \{0\} \quad (10)$$

$$\xi^\delta \in \{0, 1\}$$

The first constraint captures the condition in Proposition 1 that every internal node needs to correspond to the splitting node of some bicast scheme, while the second constraint indicates that every node in the tree must be covered at least  $\kappa \geq 1$  times. In the statement of the Proposition  $\kappa = 1$ , but for various reasons we may want to allow each link to be covered by multiple probing experiments. Notice that we can modify the

last constraint and can require differential coverage for different links of the tree. Finally, it is worth noting that similar ideas can be used for multi-source network topologies.

## 5 Analysis of UNC Network Traffic

### 5.1 Large Delay Analysis

The University of North Carolina is currently engaged in testing their network for Voice-Over-IP readiness. As part of this test, Avaya Labs has supplied them with a system for simulating phone calls over their computer network. The system is able to collect packet level information to synchronize machine clocks and compute one-way delays. This provides a unique opportunity to implement the above methods while helping UNC to evaluate their network. The system has 15 endpoints. For our proof-of-concept tests, we organize them into the tree shown in Figure 3. Node 1 is the main campus router and it connects to the University gateway. Nodes 2, 3, and 9 are also large routers responsible for different portions of the campus. The accessible nodes are all located in dorms and other university buildings. We root the tree at Sitterson Hall which houses the computer science department.

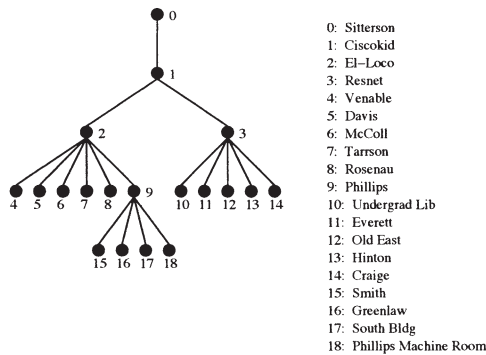


Figure 3: UNC Local Area Network.

The present study involves the use of a single source for presentation purposes. The methodology presented can be easily extended to multi-source probing and this is the subject of current investigation. A more complete analysis of this network would naturally involve probing the network from each of the endpoints.

The testing equipment simulates phone calls between endpoints. Although we cannot use the multicast protocol, simulating simultaneous phone calls creates an appropriate back-to-back mechanism: the two packets making up the back-to-back pair are sent within nanoseconds of each other. Propagation delay was removed by subtracting the minimum observed delay along each path from all of the delays collected along that path. As a general experimental protocol, we probe the network in pairs using the following experiments:

$$C = \{\langle 4, 5 \rangle, \langle 6, 7 \rangle, \langle 8, 10 \rangle, \langle 11, 12 \rangle, \langle 13, 14 \rangle, \langle 15, 16 \rangle, \langle 17, 18 \rangle\}$$

This set of experiments satisfies the posited identifiability condition. A single probing session consists of two passes

through the collection of experiments sending about 500 probes to each pair in a single pass. We conducted experiments over the course of several days in order to evaluate both the network and the methodology.

The experiments presented here were conducted on March 1, 17, and 21 of 2005 (there exists a significantly larger collection of similar data, whose analysis confirms the consistency of our results). The first date should be a fairly typical Tuesday. The second date is during Spring Break. The last date is the first Monday after Spring Break. On 3/1 and 3/17 we have collections at 9:00am, 12:00pm, 3:00pm, 6:00pm, and 9:00pm. On 3/21, we have data for 8:00am, 10:00am, 12:00pm, 2:00pm, 4:00pm, 6:00pm, 8:00pm, 10:00pm, and 12:00am. For all three days, we choose a bin size of  $q = .0001s$ . This bin size, while somewhat arbitrary, was selected since it is a large delay on this network and several delays of this magnitude can lead to a noticeable degradation in call quality. This is an example of choosing a large bin size in order to estimate the probability of a troublesome delay. The goal is assess the quality of the network upgrade it where necessary. While a more detailed analysis, using a finer grained bin would provide better estimates of the mean and variance of link delays, the most detrimental factor is large delay so we make that our focus. We expect most of the mass to fall on the zero bin, but are interested in how much lies beyond. From this analysis, we construct a picture of the probability of large delay (greater than or equal to one unit) throughout the course of the day: Figures 4, 5, and 6.

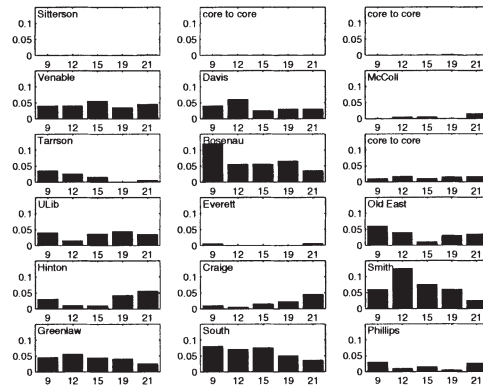


Figure 4: Probability of large delay on 3/1/2005.

This analysis is revealing in a number of ways. First, many buildings (Venable, Davis, Rosenau, Smith, Greenlaw, and South) show a typical diurnal pattern. Each of these buildings is administrative or departmental meaning that the majority of users follow a regular 9 to 5 schedule. Other buildings show a somewhat different pattern; either, a more uniform throughout the day or even an opposite one. Hinton, for example, is a large freshman dorm and thus the drop during the day and increase at night are expected as the residents return from classes and other activities in the evening. This is instructive for several reasons. It provides an indirect method of validation. The estimation procedure is locating patterns that we expect to see. This is a strong indication that the methodol-

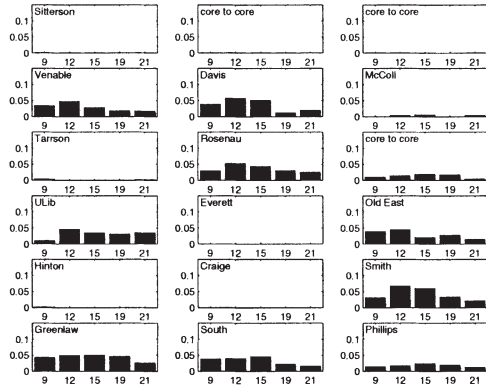


Figure 5: Probability of large delay on 3/17/2005.

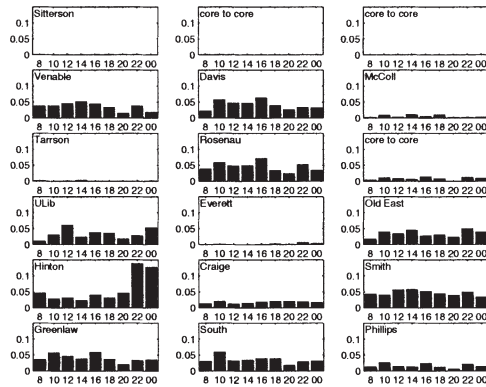


Figure 6: Probability of large delay on 3/21/2005.

ogy uncovers patterns witnessed in other studies of local area networks.

A second measure of validation is found by comparing the dorm activity before, during, and after Spring Break. Everett, Old East, Hinton, and Craige are dorms. The second collection of data taken during Spring Break reveals almost no large delays in three out of four of these buildings. This is an expected outcome as most of the dorm population was absent during that period. The other buildings also show reduced large delays. The Hinton dorm is especially interesting, since it experienced very little congestion over the break, but a significant increase to pre-break levels on the first day back. The ability to track a known campus event again provides a strong indication of the methodology’s success. Finally, the results exhibit a strong degree of consistency. Many of the monitored buildings show similar patterns during the three collection times. Furthermore, these patterns are in agreement with previous data collections examined elsewhere [2].

Active probing also allows us to evaluate the quality of the network. Many of the building links would probably require upgrades in order to support delay sensitive applications. Some of the departmental and administration buildings (Smith and South) show a tendency to have large delays even without additional traffic. Further, one of the interior core-to-core links shows some propensity to large delays, which may

propagate to large sections of the campus network.

## 5.2 Detailed Analysis

Capacity planning problems also rely on good knowledge of sample summaries like the mean, median, and variance. To estimate these well, a finer grained solution is required. To demonstrate the effectiveness of the methodology for producing this type of solution, we repeated the analysis for 3/1 using a bin size 10 times smaller than above. Figure 7 contains the partial results of this analysis. These figures have the first 20 bins of the distribution for links 9, 13, and 17 corresponding to a core-to-core link, a dorm link, and a departmental link. The more detailed distributions can be used to compute means and variances. The median can be located or bounded quite easily by simply examining the distributions. Further, the complete distributions demonstrate the same diurnal patterns seen above. For example, the distributions for link 17 are much more clustered at the lower levels after the work day is complete.

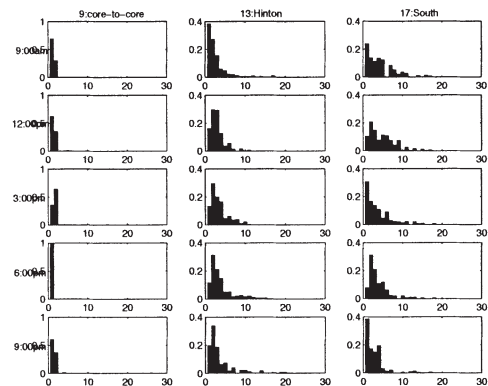


Figure 7: Distribution for 3 links throughout 3/1 using a bin size .00001s.

## 6 Concluding Remarks

We have discussed several aspects of the network tomography problem. First, we have introduced a suitable modeling framework that is appropriate for capturing the unique aspects of bursty network traffic. Additionally, several estimation schemes are presented that focus on fast estimation and estimation for large delays. Finally, as a demonstration of these techniques we have presented a real data analysis. Such analysis is still fairly novel in the area of active tomography as the tools for collection are not yet widely available. Based on this analysis, we have strong reason to believe that the methodology presented is a useful tool for assessing and monitoring network performance.

## References

- [1] R. Cáceres, N. G. Duffield, J. Horowitz, and D. F. Towsley. Multicast-based inference of network-internal loss

characteristics. *IEEE Transactions on Information Theory*, 45(7):2462–2480, November 1999.

[2] E. Lawrence, G. Michailidis, and V. N. Nair. Flexicast delay tomography. *Journal of the Royal Statistical Society Series B*, 2005. Submitted.

[3] G. Liang and B. Yu. Maximum pseudo likelihood estimation in network tomography. *IEEE Transactions on Signal Processing*, 51(8):2043–2053, August 2003.

[4] F. Lo Presti, N. G. Duffield, J. Horowitz, and D. Towsley. Multicast-based inference of network-internal delay distributions. *IEEE Transactions on Networking*, 10(6):761–775, December 2002.

[5] Y. Tsang, M. Coates, and R. D. Nowak. Network delay tomography. *IEEE Transactions on Signal Processing*, 51(8):2125–2135, August 2003.

[6] Y. Zhang, M. Roughan, C. Lund, and D. Donoho. An information-theoretic approach to traffic matrix estimation. In *ACM SIGCOMM 2003*, 2003.