

Fast, Moment-Based Estimation Methods for Delay Network Tomography

Earl Lawrence *

Statistical Sciences, Los Alamos National Laboratory

P.O. Box 1663, MS F600

Los Alamos NM 87545

P: 505-665-0898, F: 505-667-4470, E: earl@lanl.gov

George Michailidis

Department of Statistics, University of Michigan

453 West Hall

Ann Arbor MI 48109

P: 734-763-3498, F: 734-763-4676, E: gmichail@umich.edu

Vijayan N. Nair

Department of Statistics, University of Michigan

277 West Hall

Ann Arbor MI 48109

P: 734-763-8018, F: 734-763-4676, E: vnn@umich.edu

Abstract

Consider the delay network tomography problem where the goal is to estimate distributions of delays at the link-level using data on end-to-end delays. These measurements are obtained using probes that are injected at nodes located on the periphery of the network and sent to other nodes also located on the periphery. Much of the previous literature deals with discrete delay distributions by discretizing the data into small bins. This paper considers more general models with a focus on computationally efficient estimation. The moment-based schemes presented here are designed to function well for larger networks and for applications like monitoring that require speedy solutions.

EDICS: SSP-DECO, SSP-HIER, SSP-NGAU, SSP-PARE

Index Terms

network tomography, Internet, traffic, Gauss-Newton, non-linear least squares, moment estimation, monitoring

I. INTRODUCTION

Digital communications networks have become vital resources in today's information society. As such, network engineers and service providers are keenly interested in continuously assessing and monitoring the quality of service of these networks. The quality of service (QoS) characteristics include loss rates, delays, bandwidth utilization, and other measures of throughput and performance. There are, however, several challenges in collecting and analyzing data on the performance of computer and communication networks. One of the biggest challenges stems from the fact that the Internet is a collection of subnets that are controlled by many different entities which do not typically collaborate, resulting in restricted access to the internal nodes of the network.

To deal with this problem, network engineers send probe packets (ghost packets that mimic the application of interest such as VOIP) from a source node to several receiver nodes, all located on the periphery of the network of interest, and measure end-to-end delay or loss characteristics. The inverse problem is to recover the node-level delay and loss characteristics from the end-to-end measurements. This is referred to as the active network tomography problem in the literature, with the term "active" relating to the fact that the data are collected by actively probing the network.

Probing experiments consist of sending ghost packets from a single source to a group of receivers. This is done using the *multicast* protocol: a single packet originates at the source and is duplicated at each branching point on the way to the receivers. This results in correlated observations of end-to-end packet loss and delay. The goal is the recovery of the loss rates and delay distributions associated with the interior network nodes. This paper focuses on delay estimation. This is a deconvolution problem embedded in a network structure.

The delay tomography problem has been discussed by several authors. The work in [1] develops a fast heuristic algorithm for network delay modeling. Their work, like much that follows, uses a discrete delay model in which continuous measurements are binned. Although their model is fast, it can have extremely poor statistical efficiency. A straight-forward EM implementation is computationally intensive due to the large number of parameters (the number of bins multiplied by the number of nodes in the tree). Some efforts have been made to solve the problems of statistical and computational efficiency including pseudo-likelihood methods [2], optimized EM algorithms [3], and clever probing with local computation [4]. Although these solutions somewhat reduce the computational burden, the problem still remains that statistical efficient algorithms are not useful in moderate- to large-sized networks or for applications that require fast estimation.

One such application is network monitoring. A typical scenario is as follows. For a given network of interest, a number of probe packets (on the order of a few thousand over a few minutes) are sent from the source node to the receiver nodes at regular time periods. The end-to-end statistics are measured and then deconvolved to estimate the link-level quantities of interest. This deconvolution algorithm needs to satisfy two requirements. First, it must be able to solve the inverse problem in roughly the same amount of time as the sampling. Estimation that is not complete before the next sample is ready is effectively useless. Second, the algorithm must be statistically efficient so that we inject as few probes as possible and avoid affecting the network traffic.

Some recent work attempts to address this problem. For packet loss monitoring, the work of [6] introduce extend the multicast probing idea to obtain a scalable system for estimating one-way packet loss. The work of [5] uses Fourier characteristic functions to model delays in a computationally efficient manner. Their work uses mixture distributions to model the delays, thus providing a different kind of flexibility from the models described here that focus on moments.

This paper seeks to develop statistically efficient tomographic estimation for link-level QoS that scales well to time-dependent applications and large networks. There are two aspects to this development. The first is a set of models that facilitate fast estimation and have small storage requirements. The second aspect is an algorithm with improved computational complexity over the estimation schemes of previous network tomography solutions. The resulting estimation algorithm will be fast enough to accommodate accurate real-time monitoring of a network using the specified model.

The paper is organized as follows. Section 2 describes network tomography in mathematical detail. Section 3 discusses on the identifiability of models that are useful in this context. Section 4 lays out the estimation scheme in detail. Sections 5 and 6 test the algorithm and a specific model in a wide variety of simulation scenarios and a specific monitoring scenario.

II. THE INVERSE PROBLEM

We can represent the topology of a network with a single source node and many receiver nodes as a tree (see Figure 1 for simple examples that will be used later to illustrate the ideas). In general, we will denote the tree as $\mathcal{T} = \{\mathcal{V}, \mathcal{E}\}$ where \mathcal{V} is the set of nodes and \mathcal{E} is the set of links that connect the nodes. The node set \mathcal{V} consists of a source node 0, receiver nodes \mathcal{R} , and interior nodes (that are inaccessible) \mathcal{I} . Let $\mathcal{P}_{i,j}$ denote the path between any two connected nodes i and j . Links will share a name with the node at their terminus (thus, a name k may refer to a link or node depending on context and there is no link 0).

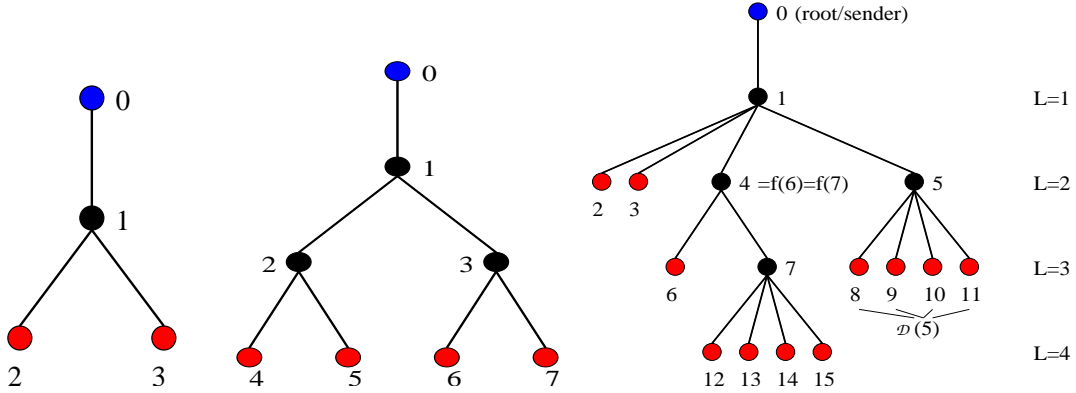


Fig. 1. Left: Simple two-layer, two-leaf topology. Middle: Three-layer binary tree. Right: Tree topology with notation.

Given a network of interest, suppose we send a probe packet from the source node to subset of the receiver nodes. Let $\mathbf{Y} = \{Y_{r_1}, \dots, Y_{r_m}\}^T$ (where m is the number of the receivers in the experiment) denote the end-to-end delays that are observed. Let $\mathcal{P}_{0,r} = \{1, i_1, \dots, r\}$ denote the links between the source node 0 and receiver node r . Let X_1, X_{i_1}, \dots, X_r be the internal link-level delays. Then, $Y_r = X_1 + X_{i_1} + \dots + X_r$. A similar set of relationships hold for the other receiver nodes and corresponding internal nodes. We can write these relationships, in general, as the matrix equation

$$\mathbf{Y} = \mathbf{A}\mathbf{X} \quad (1)$$

where \mathbf{X} is the set of all internal node delays and \mathbf{A} is the routing matrix: the matrix with a column for each link and row for each path and $\mathbf{A}_{i,j} = 1$ if path i includes link j . The tree topologies and the inaccessibility of the internal nodes imply that \mathbf{A} is not invertible and the above equation is cannot be directly solved for \mathbf{X} .

Example: Consider the simple topology on the left in Fig. 1 where node 1 is inaccessible, and nodes 2 and 3 are probed from node 0. We will observe delays on the path from 0 to 2 and the path from 0 to 3 and we wish to make inference on the delays accumulated between nodes 0 and 1, nodes 1 and 2, and nodes 2 and 3. This problem has two paths and three links. The routing matrix is given by

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}.$$

Because \mathbf{A} cannot generally be inverted, we will have to conduct experiments in special ways to ensure that the deconvolution problem can be solved. Most of the traffic on the Internet uses a *unicast* transmission scheme: a single source transmits packets to a single receiver. Unless we make strong

assumptions, this type of transmission cannot be used by itself in order to recover the link-level delay distributions. As mentioned above, the multicast protocol probes multiple receivers from a single source, resulting in correlated delays. This correlation adds information that can be used to solve the deconvolution problem. The *flexicast* probing method in [4] refers to the grouping of receivers in such a way as to guarantee identifiability while reducing the computational burden. This method breaks the total set of receivers into groups that are probed separately from each other. When chosen properly, the observed end-to-end data can be used to estimate all of the internal link-level parameters.

III. MODELS AND IDENTIFIABILITY

The focus of this section is the development of models that are appropriate for fast estimation in applications such as monitoring. We consider a general framework for modeling delays as follows. First, we assume that the internal delay random variables X are spatially and temporally independent. Specifically, delays on different links are independent and delays at different times over a single link are independent. This is a common assumption in the tomography literature (see the references above). The framework for the distribution of a delay on a single link k can be written as

$$\begin{aligned} \mathbb{P}\{X_k < \infty\} &= \alpha_k, \\ \mathbb{P}\{X_k = 0 | X_k < \infty\} &= p_k, \\ X_k | \{0 < X_k < \infty\} &\sim F_k(x; \theta_k). \end{aligned} \tag{2}$$

Each packet is successfully transmitted across link k with probability α_k . Given transmission, the packet experiences an empty queue, and thus zero delay, with probability p_k . Finally, given transmission and a non-empty queue, the delay in transmission across link k follows some continuous distribution indexed by parameter θ_k .

This general framework is appealing for several reasons. First, it combines loss and delay modeling under one umbrella so that separate analyses are not required. Second, the model explicitly accounts for two important network engineering characteristics: transmission/loss probability and link utilization (one minus the empty-queue probability). Finally, it allows flexibility in the choice of F_k . We can exploit this flexibility to choose models that meet the requirements of fast estimation.

The following proposition presents conditions under which models embedded in the general framework can be estimated.

Proposition 1: Let \mathcal{T} be a general tree network. Let \mathcal{C} be a collection of flexicast receiver groups \mathcal{C}_j , $j = 1, \dots, C$. Let each link delay distribution be given by Equation 2. The parameters of the link-

level delay distributions are estimable if (a) $\alpha_k, p_k > 0$ for all $k \in \mathcal{V}$, (b) for every internal node $s \in \mathcal{I}$, there is at least one flexicast scheme $\mathcal{C}_j \in \mathcal{C}$ with at least two receivers that are descendants of s (s is a branching node for \mathcal{C}_j), and (c) every receiver $r \in \mathcal{R}$ is included in at least one $\mathcal{C}_j \in \mathcal{C}$.

The proof is given in the Appendix.

Proposition 1 presents very general conditions under which a wide variety of models can be estimated. However, busy networks may experience link utilizations that are at or very near one, violating one of the conditions given above. For these instances, we present a simple sufficiency condition that covers numerous appropriate parametric distributions or parametric mean-variance relationships. This type of modeling, although approximate, is ideally suited for time-dependent applications because they result in simple parameter-based formulas for quantities of interest like moments and quantiles. The key feature here is that central moments, $E(X_k - \mu_k)^\lambda$, are estimable for $\lambda > 1$ on all links in the tree. Thus we can estimate any distribution or set of moment relationships that is identifiable from these moments.

Proposition 2: Let \mathcal{T} be a general tree network. Let \mathcal{C} be a collection of flexicast receiver groups \mathcal{C}_j , $j = 1, \dots, C$. Let each link delay distribution be given by $F_k(x; \theta_k)$. The parameters of the link-level delay distributions are identifiable if (a) for every internal node $s \in \mathcal{I}$, there is at least one flexicast scheme $\mathcal{C}_j \in \mathcal{C}$ with at least two receivers that are descendants of s (s is a branching node for \mathcal{C}_j), (b) every receiver $r \in \mathcal{R}$ is included in at least one $\mathcal{C}_j \in \mathcal{C}$, and (c) for $k \in \mathcal{I}$, θ_k is estimable based on central moments of order two and higher and for $k \in \mathcal{R}$, θ_k is estimable from the ordinary or central moments of any order.

Remark 1: This proposition gives identifiability for many useful parametric distributions including exponential, gamma, log-normal, Weibull, and others. The key is that each of these distributions has higher order moments that are functions of or provide information about the mean.

Remark 2: This proposition allows the estimation of different distributions on different links. This provides a great deal of modeling flexibility. Large capacity links can be given one distributional form and smaller links another. Further specification can be considered according to prior knowledge as long as estimability is ensured.

The proof of this proposition can be found in the Appendix. See [7] for a discussion of parametric delay estimation and examples. The proof also demonstrates the ability to simply track raw central moments without specifying a parametric form. For example, one could simply estimate the variance for each link and use this statistic to make decisions about the network (see [8] for detailed work on variance tomography).

Specifying a complete distribution can be quite restrictive and there may be no obvious choice. As a

result, we also consider semiparametric model fitting in which moments are only specified as desired. These can be used to characterize the distributions under study or as a guide toward choosing appropriate parametric models. In particular, they allow the researcher to identify and monitor only desired quantities without further restriction. This reduces the storage and estimation requirements as well, resulting in lighter computational burden that still meets that demand of the user. As a useful example, we will consider the following model that will be explored further in the latter parts of the paper:

$$E(X_k) = \mu_k, \quad \text{Var}(X_k) = \phi \mu_k^\gamma. \quad (3)$$

This model, partially inspired by generalized linear modeling and quasi-likelihood [9] modeling, has two appealing features: it is fairly simple with a form that is similar to many common parametric distributions including exponential, gamma, and log-normal. The work of [10] in the passive tomography setting (estimating origin-destination traffic intensity based on link-level packet counts) is also a source of inspiration. In their work, the packet count data are modeled as normally distributed with variance proportional to the mean raised to a power. Accordingly, the model is appropriate for delays since they can act as a proxy for packet counts. This model is easily embedded in the general framework Eq. 2. Under the general framework, the means are identifiable by Proposition 1. The parameters ϕ and γ are not strictly identifiable in all situations since they are common across links. It is easy to show that these parameters cannot be identified for a network with common continuous link means on all links. In this case, the parameter γ and the function $\log(\phi)$ are perfectly correlated. Nevertheless, they can be considered nuisance parameters. Specifying the second moment adds information for estimating the mean even when the parameters cannot be estimated because the relationship between the log of the variance and the mean is unchanged.

IV. ESTIMATION

This section develops a scheme for estimating link-level delay distributions based on end-to-end measurements. The goal is a scheme that scales well to large networks and is fast enough for realistic monitoring. The approach we take is based on moment matching. As we will see, this only requires storage of the required end-to-end moments and the parameters can be estimated using matrix inversions.

Before we begin, we mention briefly that maximum likelihood estimation presents several difficulties. First, the likelihood functions are large and unwieldy for even simple distributions and moderately sized trees. Some discussion of this issue can be found in [7]. Second, fitting procedures like EM can be quite computationally intensive and do not lend themselves to the fast implementation required by many

applications. Finally, we have presented several types of models and maximum likelihood may not be appropriate for all them, especially when no particular likelihood is assumed. Here we develop a unified estimation procedure that applies quite generally and is computationally efficient. Some of the following is developed in [7] in the context of parametric modeling. This presentation is more general and more rigorous.

A. Problem Formulation

We focus on moment estimation. The goal will be to choose parameter values so that our moment formulas match the observed moments as closely as possible based on minimizing some loss function. Although there are many choices, we favor squared-error loss. The choice is attractive for two reasons. First, for large samples, the moments should be approximately normal, thus squared-error loss is appropriate for approximating a normal likelihood. Second, we can borrow from the well-developed literature on optimization in the squared-error setting.

The focus will be on minimizing the squared difference between observed and expected moments. The expected moments are in terms of the parameters that we are attempting to estimate. The moments may take any number of forms. They may be an end-to-end means, variances, or other higher order central moments. They may also be probabilities, possibly on the log or logit scale, of seeing zero delay or loss along a path or set of paths. All that is required in much of the subsequent development is that each be some observed moment from end-to-end measurements and that each converges in probability to its true value and converges in distribution to a Gaussian variate (after appropriate scaling).

We consider estimation for some set of flexicast receiver groups $\mathcal{C} = \{\mathcal{C}_j, j = 1, \dots, C\}$. The monitoring will be done based on a series of probes collected over small intervals. For the present development, we will consider a single interval to avoid additional notation. Let M_i^j be the observed i -th moment for the j -th flexicast receiver group. Let $\mathcal{M}_i^j(\theta)$ be the functional form of the i -th moment from the j -th flexicast receiver group. The loss function is given by:

$$Q(\theta; \mathbf{M}) = \sum_{j=1}^C \sum_i \left[M_i^j - \mathcal{M}_i^j(\theta) \right]^2.$$

Example: As an example, consider the observed values that could be used to fit the following exponential-like model, a special case of the model in Equation 3 and estimable under Proposition 2:

$$\begin{aligned} \mathbf{E}(X_k) &= \theta_k \\ \mathbf{Var}(X_k) &= \theta_k^2. \end{aligned}$$

Estimation for the simplest topology in Fig. 1 can be performed based on the following end-to-end statistics:

$$\begin{aligned} \mathbb{E}(Y_r) &= \theta_1 + \theta_r, \quad r \in \{2, 3\}, \\ \text{Cov}(Y_2, Y_3) &= \theta_1^2, \\ \text{Var}(Y_r) &= \theta_1^2 + \theta_r^2, \quad r \in \{2, 3\}. \end{aligned}$$

Note that point masses at zero and infinity can be added by treating all of the above end-to-end statistics as conditional on being positive and finite and adding log-probability-based moments to include empty queues and transmission success. In this example, transmission probabilities could be modeled with equations like

$$\log(\mathbb{P}\{Y_r < \infty\}) = \log(\alpha_1) + \log(\alpha_r), \quad r \in \{2, 3\}. \quad (4)$$

We note that this setting allows for quite general estimation. Parametric distributions are of course easily fit with this method, as in [7]. Later, we will focus on a semiparametric approach by specifying the functional form for any desired set of moments. Further shape restrictions can be imposed as desired. Finally, as we have seen, this procedure allows us to easily mix a continuous distribution with point masses at zero and infinity. This is appealing for network tomography since it allows one to estimate the probability of an empty queue and a full queue along with the queuing distribution for the intermediate case.

Example: As a further example of the generality of this method, consider the simplest topology in Fig. 1 with the simplest version of the discrete delay model studied in [4] and elsewhere. Let each link delay have two possible values: $X_k \in \{0, 1\}$ and let $\alpha_k(i) = \mathbb{P}\{X_k = i\}$. Note that identifiability for this model can be shown using Proposition 2. Using this model, we can form the following moments to be used for estimation:

$$\begin{aligned} \text{Cov}(Y_2, Y_3) &= \alpha_1(1) - \alpha_1^2(1), \\ \mathbb{E}(Y_2) &= \alpha_1(1) + \alpha_2(1), \\ \mathbb{E}(Y_3) &= \alpha_1(1) + \alpha_3(1). \end{aligned}$$

Similar moments can be computed for discrete distributions with more bins and for networks with more complicated topologies, but the equations quickly become extremely cumbersome.

B. Fitting

Because of the similarity to traditional nonlinear least squares, we consider model fitting using the Gauss-Newton procedure (see [11], for example). The algorithm is attractive because of its simplicity. We develop it here in the context of the moment estimation.

Rewrite the loss function as one sum over all the moments and consider its derivatives:

$$Q(\theta; \mathbf{M}) = \sum_i [M_i - \mathcal{M}_i(\theta)]^2,$$

$$\frac{\partial Q(\theta; \mathbf{M})}{\partial \theta_j} = -2 \sum_i [M_i - \mathcal{M}_i(\theta)] \frac{\partial \mathcal{M}_i(\theta)}{\partial \theta_j}.$$

In order to minimize the least-squares criterion, we need to find θ such that the above derivative equation is equal to a vector of zeros (allowing us to drop the constant). We can consider all of the derivatives in matrix form:

$$\left[\frac{\partial Q(\theta; \mathbf{M})}{\partial \theta} \right] = D' [M - \mathcal{M}(\theta)],$$

where

$$D_{i,j} = \frac{\partial \mathcal{M}_i(\theta)}{\partial \theta_j}.$$

Example: Consider again the simplest topology in Fig. 1 with the exponential-like semiparametric model on each link. The model yields the following vectorized end-to-end moments:

$$\mathcal{M}(\theta) = \begin{bmatrix} \theta_1 + \theta_2 \\ \theta_1 + \theta_3 \\ \theta_1^2 \\ \theta_1^2 + \theta_2^2 \\ \theta_1^2 + \theta_3^2 \end{bmatrix}.$$

The entries are the means, the covariance, and the variances of the end-to-end values.

For this simple example, we have the following matrix of partial derivatives:

$$D = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 2\theta_1 & 0 & 0 \\ 2\theta_1 & 2\theta_2 & 0 \\ 2\theta_1 & 0 & 2\theta_3 \end{bmatrix}.$$

We can approximate the moments at the true value using a Taylor expansion around some initial guess $\theta^{(0)}$:

$$\mathcal{M}(\theta_0) \approx \mathcal{M}(\theta^{(0)}) + D(\theta_0 - \theta^{(0)}),$$

Forming the residuals and replacing the true value with the observed moments gives us an updating scheme based on solving a linear system. Thus at some iteration q , we have the following linear system:

$$M - \mathcal{M}(\theta^{(q)}) = D\beta,$$

where β is the unknown offset between the current estimate of θ and the true value θ_0 . We can solve the system and get the next iteration given by $\theta^{(q+1)} = \theta^{(q)} + \hat{\beta}$.

In general each iteration should be closer to the minimizer of the loss function. However, there may be situations where the step increases the sum of squares. To avoid this, we use the modified Gauss-Newton in which the next iteration is given by $\theta^{(q+1)} = \theta^{(q)} + r\hat{\beta}$ where $0 < r \leq 1$. This fraction can be chosen adaptively at each step. If the full step reduces the sum of squares, then it is taken. Otherwise, set $r = .5$. If the half step fails to reduce the sum of squares, then it is halved again. This guarantees that the loss function is reduced with every step and gives convergence to a stationarity point. Examination of the derivatives will indicate if the point is a minimum.

The algorithm has useful complexity properties in terms of both memory and computation. Since the estimation is based only on the moments, it requires very little storage. This is a vast improvement over algorithms that require all of the data or the counts of the binned data, like those referenced in Section 1. Further, the efficient implementation of the algorithm, involving a QR factorization and one matrix inversion gives computational complexity of $\mathcal{O}(m^3)$ for each iteration, where m is the number of required moments. Again, this is a large improvement over other methods that have exponential complexity, such as the EM algorithm for the discrete delay model [4]. Further improvement can be gained in many cases by using sparse matrix techniques, as will be the case with large trees and small flexicast probing groups, such as bicasts. These two points make the current modeling paradigm ideal for application requiring fast estimates.

C. Large Sample Behavior

The procedure has desirable asymptotic properties that allow for inference in the large sample setting.

Proposition 3: Let \mathcal{T} be a general tree network. Let \mathcal{C} be a collection of flexicast receiver groups \mathcal{C}_j , $j = 1, \dots, C$. Let each link delay distribution be given by $F_k(x, \theta_k)$. Assume that the model specification is correct and that the F_k are identifiable from \mathcal{C} . If the observed end-to-end moments obey the Strong Law of Large Numbers and the Central Limit Theorem, then the least-squares moment

estimator is consistent and asymptotically normal:

$$\begin{aligned}\hat{\theta} &\rightarrow \theta_0 \text{ w.p. } 1 \\ \sqrt{\mathbf{n}^\dagger}(\hat{\theta} - \theta_0) &\Rightarrow Z \sim N(\vec{0}, \Sigma),\end{aligned}$$

where $\sqrt{\mathbf{n}^\dagger} = \sqrt{\mathbf{n}}D$, \mathbf{n} is a diagonal matrix with the appropriate number of observations for each moment, D is the matrix of partial derivatives, and Σ is the covariance of the observed moments.

The proof can be found in the Appendix. The term \mathbf{n}^\dagger can be viewed as the number of observations related to each estimated parameter. With these facts, we can compute approximate confidence regions and hypothesis tests, key components for monitoring applications.

D. Weighting

In the traditional nonlinear least squares setting, observations with different variances can be weighted to assist estimation. The same procedure can be used here in the current moment matching setting, particularly as the variance for moments is often well-known. Two procedures can be used depending on the setting. If an entire distribution is specified and the variance of each moment can be calculated in terms of the parameters, then the weights can be updated with each new iteration of the parameters. Alternatively, the variance of each moment can be computed based on the observations and held constant throughout the estimating procedure.

A second method computes weights based on the general formulas for the variances of the moments. For example, the variance of an end-to-end mean is just the end-to-end variance divided by the number of observations. More complicated moments have more complicated variances, but empirical estimates can be computed directly from the observations.

This procedure is easily extended to larger numbers of moments. It also extends to flexicast probing in which separate groups of receivers are probed simultaneously. In this case, the above procedure is applied to each flexicast probing group individually. As probing groups are independent, the overall weight matrix is block-diagonal.

Weighting the observations can also play a useful numerical role when some of the moments are large in magnitude compared to others. The weighting generally helps to speed convergence and does not affect the other properties of the algorithm. See [7] for a limited comparison in a parametric case.

V. SIMULATION STUDY

In this section we examine the results of a large simulation study in order to assess the performance of the algorithm both generally and across many specific scenarios. We also compare the efficiency of

the proposed procedure with a simpler variance-only estimation procedure.

The data is simulated from the simple two-layer tree from Fig. 1. We fit the semiparametric model with point masses at zero and infinity that is obtained from embedding Eq. 3 in Eq. 2. Each parameter is assigned a high and low value and data is generated at each combination. The transmission probabilities can be either 0.9 or 0.999. The empty-queue probabilities can be either 0.1 or 0.5. The trunk link means can be either 2 or 10 and the leaf link means can be either 3 or 11. The overdispersion parameter ϕ can take values 3 or 9. Finally, the power parameter γ can take values of 2 and 3. The parameter γ is manipulated by generating data using the log-normal distribution for $\gamma = 2$ or the inverse Gaussian distribution for $\gamma = 3$. For each set of parameters, there are 100 data sets with 100,000 observations.

A. Testing the Semi-Parametric Model

We analyze the data using the Gauss-Newton procedure with weights based on the computed variance of each observation.

The maximum RMSE proportion for any of the three transmission parameters across all of the simulations is 0.0013 and the maximum proportion absolute bias is 0.0004. These parameters are independent of the other parameters and are easily estimated from their sufficient statistics: counts of the individual path success and a count of the joint path successes. These statistics allow a matrix equation on the log scale that can be solved directly without iteration. With the large number of observations, these sufficient statistics have small variances and the resulting estimates are very good.

The maximum RMSE proportion for any of the empty-queue parameters is 0.13 and the maximum proportion absolute bias is 0.037. Although this parameter is not independent of the mean and variance parameters, the bulk of the information comes from sufficient statistics similar to those used for the transmission parameters. Once again, these statistics have small variances leading to good parameter estimates.

The mean parameters are more difficult. The average RMSE proportion across all scenarios for each of the mean parameters is around 0.13 and the average proportion absolute bias across all scenarios is around 0.025. These averages conceal some large values. The maximum absolute proportional biases for each of the links is in the 0.40 to 0.44 range. In the cases where this occurs (biases of greater than 0.2 occur in about 4% of the scenarios), it is almost always related to the disparity in mean parameters. Either the trunk mean is small and both leaf means are large or vice versa. These parameter settings can produce unusual results. When the trunk is an order of magnitude larger than the leaf means, the observed end-to-end covariance can be larger than one of the observed end-to-end variances. If the reverse

is true and the trunk mean is an order of magnitude smaller than the leaf means, then the end-to-end covariance can be very small relative to the end-to-end variances. In both cases, the algorithm accounts for the resulting variation by inflating or shrinking at least one of the mean parameters while trying to maintain the end-to-end means. In these cases, the mean-variance relationship that makes estimation possible actually causes trouble as the random fluctuations in the variance-covariance moments lead to untenable settings that are resolved as best as possible.

All of these estimates would appear to have good properties for tracking network behavior and detecting change.

B. Comparing with Variance-Only Estimation

Each of the scenarios was also considered using an algorithm that estimates only the transmission probabilities and the variances. In this case, the variance estimation uses a single matrix inversion to estimate the link variances from the observed end-to-end covariance matrix. The variances are not restricted to any parametric form.

Figure 2 shows histograms the log of the ratios of the RMSE for the semi-parametric variances to the RMSE for the variance-only variances for each of the 2048 scenarios. As the histograms indicate, the semi-parametric model is competitive on variance estimation for the vast majority of the cases. This should not be surprising as the data were generated from an appropriate model. The cases for which the variance-only estimation is superior roughly correspond to those cases for which the data has the largest variance. In these cases, the moments show greater variability and the semi-parametric model with its common variance parameters may not fit as well.

Although the unrestricted variance modeling may be more flexible and have a slight advantage even when the semi-parametric model is appropriate, it does not necessarily more useful for decision-making. The average RMSE for these statistics is very large under both estimation schemes (often several times the true value) since these statistics can vary considerably. Using the semi-parametric model, with its ability to track the much better-behaved link means, would appear to provide a much better model for network monitoring.

VI. NETWORK MONITORING

In this section, we investigate simulated monitoring scenarios to assess the effectiveness of our models and algorithms using *ns-2*, a network simulator [12].

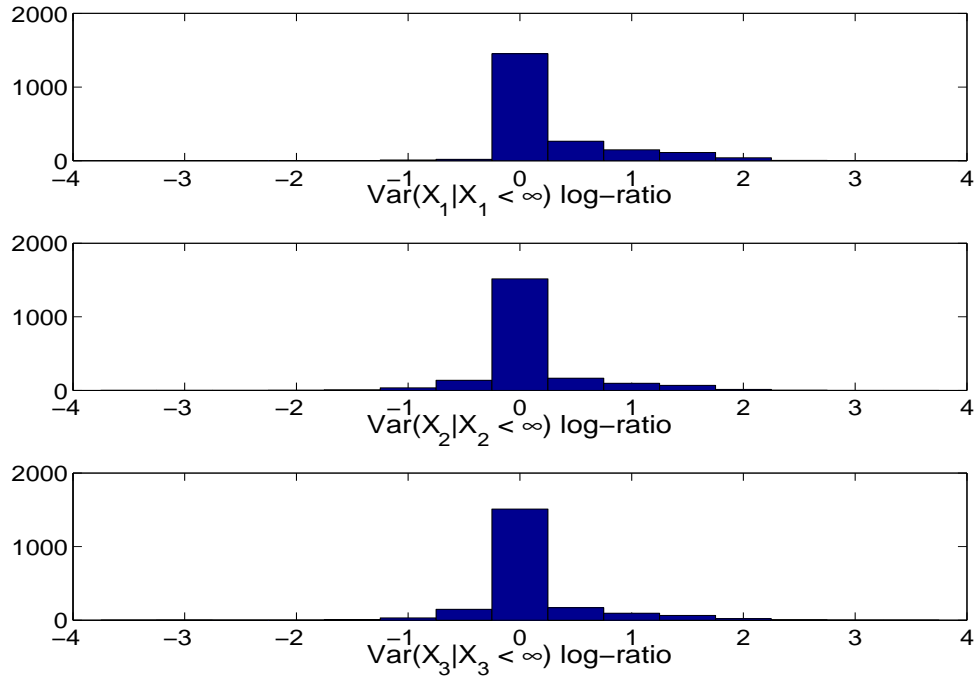


Fig. 2. Log of the ratio of RMSE of the variance from the semi-parametric model to the variance from the variance-only estimation.

A. Monitoring Technique

We consider monitoring a network using the semiparametric model (Eq. 3) embedded in the general framework (Eq. 2). This model allows one to track the probabilities of loss and empty queue as well as the mean of the delay distribution.

We will use two kinds of exponentially-weighted, moving average (EWMA) charts to monitor the link delay distributions. For the individual links, EWMA charts will be computed where the observations are estimated quantities for each time interval. Monitoring these alone can sometimes be misleading as the estimates of the means are correlated. As a result, we will use a multivariate EWMA (MEWMA) chart to monitor the whole network at once. This chart is an extension of the EWMA in which the observations are a vector. First we compute the vector moving average for the current time period H : $Z_H = \lambda \vec{\mu}_H + (1 - \lambda)Z_{H-1}$, where $\vec{\mu}_H$ is our estimated vector of means at the current time period. Then we compute the quadratic form: $T^2 = Z_H' \Sigma^{-1} Z_H$, where Σ is the estimated covariance of Z_H computed during a control period. We chart and monitor T^2 to evaluate the network as a whole. See [13] for more

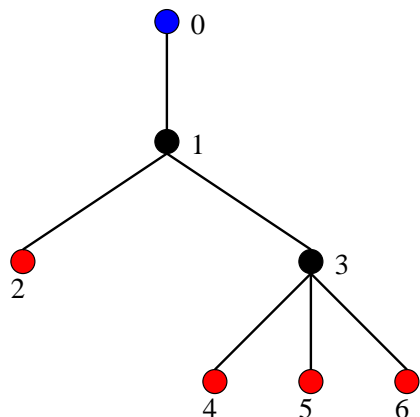


Fig. 3. Small network used in the monitoring example.

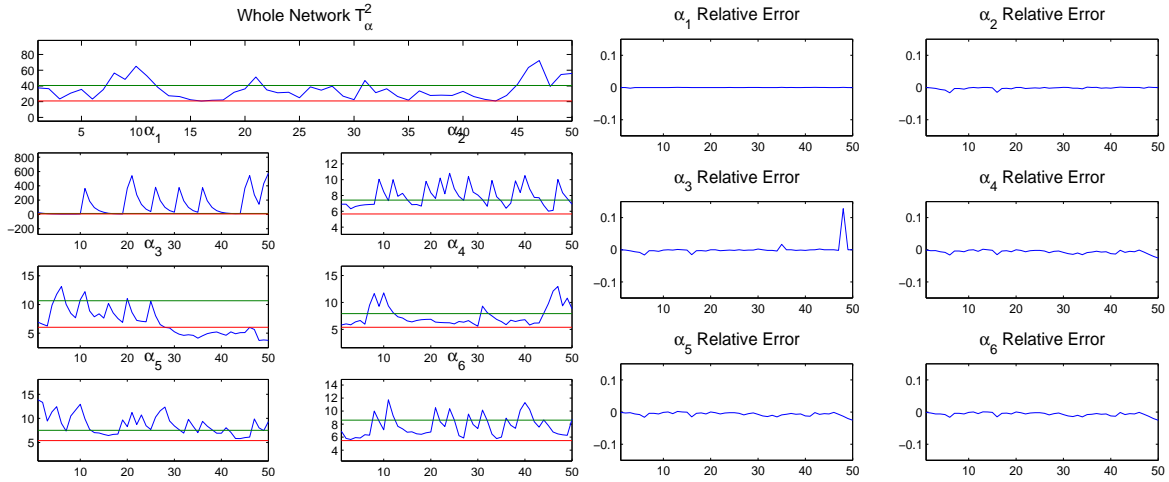
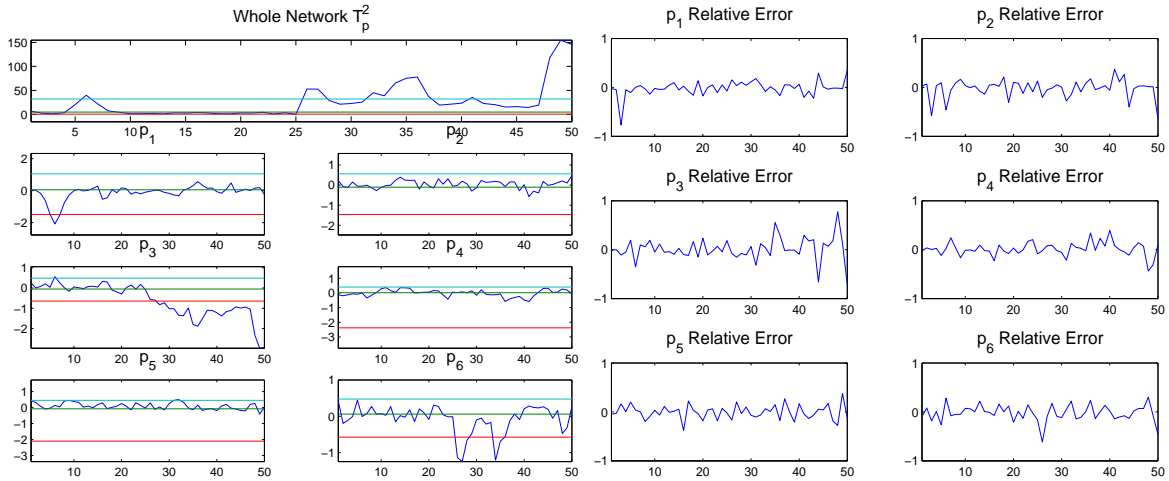
background on this issue.

B. Monitoring Scenario in NS-2

The *ns-2* network simulator allows us to simulate network traffic using realistic network protocols. Topologies are constructed in terms of endpoints, routers (both of these are nodes), and connections (links). Parameters such as transmission protocols and bandwidths can be specified to mimic real networks. This simulator also allows us to assess the performance of our methods in the presence of violations of our spatio-temporal assumptions.

We simulate a monitoring scenario using the topology shown in Fig. 3. Each link has 10Mb of bandwidth. The background traffic on each link consists of TCP and UDP connections with exponential interarrival times and pareto lifetimes. The UDP connections are arranged to be fewer in number and much longer in length than the TCP connections.

In the first phase, we simulate a period in which the network is in its normal state to obtain control parameters. In the next phase, we simulate a period in which the network goes out of control. At the halfway point, we double the number of TCP arrivals on the interior link 3. We generate 10000 seconds of data for the control period and 5000 seconds of data for the monitoring period, both at 10 probes per second. We will use 100 seconds of data for each monitoring period, so the estimates at each time point will be based on the previous 1000 probes.

Fig. 4. Control charts and relative accuracy for $\vec{\alpha}$ Fig. 5. Control charts and relative accuracy for \vec{p}

C. Results

The results of the simulation are control charts for the mean, the logit of the transmission probability, and the logit of the empty-queue probability at both the link-level and the whole-network T^2 level. The control charts and relative errors can be seen in Fig. 4, 5, and 6.

First we consider the question of efficiency. For the 50 monitoring periods, the longest estimation time for a Matlab implementation on a 3 GHz Xeon processor was about 11 seconds. With monitoring periods of 100 seconds, we easily have the solution for a particular time period before the data from the next

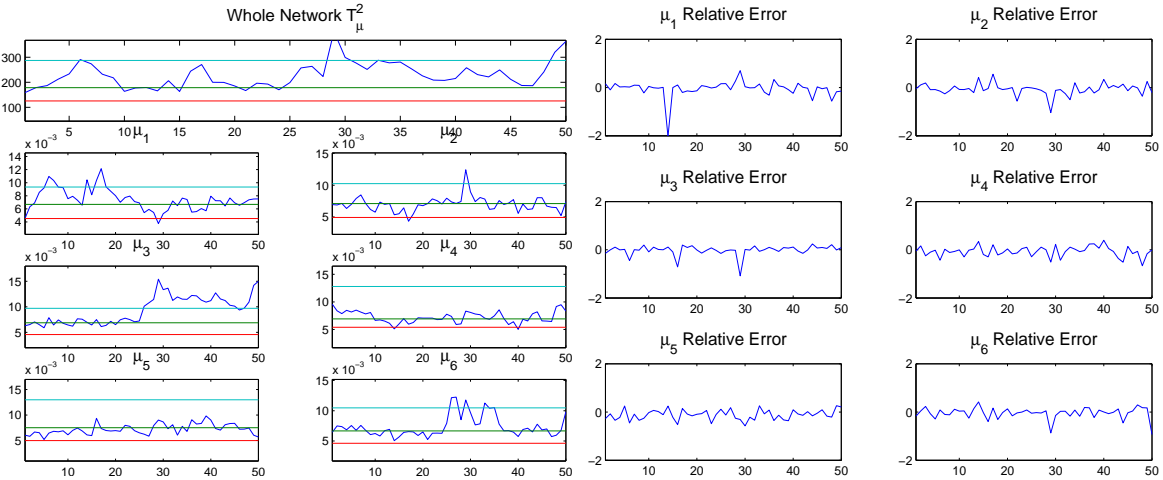


Fig. 6. Control charts and relative accuracy for $\vec{\mu}$

time period are available.

Next, we examine the accuracy of the method. The right panels of Fig. 4, 5, and 6 show the relative accuracy in terms of percentage for each parameter. These are computed using the observed link-level output available from the simulation. With few exceptions, the values are quite small. There is almost no error in the estimation of the transmission probability, largely because it is always close to one. The errors in empty-queue probabilities are all less than 1%. For both of the probability moments, the data provide quite a bit of information about the parameter values and both could be estimated using a single matrix inversion (the transmission moments are essentially estimated this way). The errors in the mean parameters are less than 2%. All of the link means are generally of the same magnitude, a scenario identified in the simulation study as being fairly easy for the algorithm.

The charts show a fairly dramatic change. Depending on the rule, the problem should be diagnosed quickly with the trouble assigned to link 3. The bounds here are chosen to give the same probability coverage as plus or minus three standard deviations when monitoring a Gaussian variate. Something more appropriate could be chosen along with more sophisticated stopping rules; this is the subject of ongoing research [14].

VII. CONCLUSION

This paper has extended the active network tomography paradigm to include practical continuous estimation. In addition to parametric models, we consider semiparametric models that only require the user to specify a few moments. This allows flexible choices to be made depending on goals. Second, we

develop a procedure for moment estimation using the Gauss-Newton search. The procedure is capable of handling a wide class of models and is computationally efficient enough to scale well to large networks. We have considered an example in which the model tracks the quantities of interest very closely and does so quickly enough to monitor changes in the traffic.

In general, the models that we consider here are approximate; a trade we have made in order to be efficient. There is further work to be done in the development of flexible models that are capable of modeling network traffic to any desired accuracy. In the area of monitoring, further work needs to be done in determining appropriate bounds and decision rules declaring a network to be in or out of control.

APPENDIX A

PROOF OF PROPOSITION 1

We begin with $\vec{\alpha}$. Estimation and identifiability for these parameters can also be found in Xi *et al.* [15]. By assumption, there is a probing scheme \mathcal{C}_j with receivers k and ℓ whose branching point is node 1. Consider the data arising from this scheme in the following linear model:

$$\begin{pmatrix} \log \mathbf{P}\{Y_k, Y_\ell < \infty\} \\ \log \mathbf{P}\{Y_k < \infty\} \\ \log \mathbf{P}\{Y_\ell < \infty\} \end{pmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \begin{pmatrix} \log(\alpha_1) \\ \log(\pi_{1,k}) \\ \log(\pi_{1,\ell}) \end{pmatrix},$$

where $\pi_{1,k}$ is the path-level transmission probability from node 1 to node k . The routing matrix in this equation is invertible, meaning we can estimate α_1 . The remaining internal links proceed by induction. A similar equation can be formulated for any internal node s using $\pi_{0,s}$ in place of α_1 and then using the transmission probability estimates for each link in the path up to link s to calculate α_s . The transmission probabilities for the receivers are easily obtained in a similar manner using path-level estimates and the estimates for the ancestor probabilities.

We now turn to the remaining parameters. By assumption, there is a group with receivers k and ℓ that split at node 1. Consider data from this group in which $y_k = y_\ell < \infty$. With probability one $x_1 = y_k = y_\ell$. Thus we have direct observation from link 1. We can estimate p_1 as the proportion of these observations that is equal to 0. The remainder of these observations, those greater than zero, can be used to estimate the continuous queuing distribution. For example, these direct observations can be used to estimate the Moment Generating Function (MGF) and compute the density.

Further link parameters for descendants of node 1 can be estimated using path-level moments and probabilities and the estimates for the moments and probabilities of the node's ancestors. Again, the MGF can be used to compute the density. \square

APPENDIX B
PROOF OF PROPOSITION 2

The proposition is easily shown by demonstrating that the required moments are available under this probing scheme.

Consider link 1. By assumption, there is some receiver group \mathcal{C}_j with receivers k and ℓ that split at node 1. Under our assumptions, the covariance of the end-to-end delays observed at these nodes is given by the variance of the delay distribution on link 1. Note that we also get the variances of the path-level distributions on $\mathcal{P}_{1,k}$ and $\mathcal{P}_{1,\ell}$.

We proceed by induction. Assume that we have estimated the central moments of orders two through $\lambda - 1$ for link 1 and the paths to the receivers, $\mathcal{P}_{1,k}$ and $\mathcal{P}_{1,\ell}$. Consider the quantity:

$$\mathbb{E}(Z_k^{\lfloor \lambda/2 \rfloor} Z_\ell^{\lambda - \lfloor \lambda/2 \rfloor}),$$

where Z is a centered end-to-end measurement. This estimable quantity is a function of the λ -th order central moment of link 1 and the lower order moments of link 1 and the receiver paths. Thus central moments of all order two and higher are estimable. Thus, we can estimate all the components of θ_1 .

The rest of the links in \mathcal{I} are handled through induction. For any node s , there is once again some pair k and ℓ that splits at s . If we have estimated the required moments for all ancestors of s , then the moments for s are obtained by estimating the central moments of the distribution on $\mathcal{P}_{0,s}$ (in the same fashion that we used for link 1) and subtracting off the moments of each ancestor link.

Receiver links are also similar except that we can use the first order moment as well. Clearly we can estimate the mean of the end-to-end path. Since we have completely characterized the distributions of all the links in \mathcal{I} , we can estimate their means and subtract them from the end-to-end mean. \square

APPENDIX C
PROOF OF PROPOSITION 3

We begin with consistency. Consider the vector of partial derivatives of the loss function set equal to zero:

$$\left[\frac{\partial Q(\theta; \mathbf{M})}{\partial \theta} \right] = \vec{0}.$$

If the model is correct, then a solution of this equation is θ_0 and $\mathcal{M}(\theta_0)$. Further, the Hessian at this value is given by: $H = D'D$ where D is the matrix of partial derivatives. If the model is identifiable, then the columns of D are linearly independent and H is strictly positive-definite. By the implicit function theorem,

there is a neighborhood U around $\mathcal{M}(\theta_0)$, a neighborhood V around θ_0 , and a unique, continuously differentiable function $\varphi : U \rightarrow V$ such that

$$\left[\frac{\partial Q(\theta; \mathbf{M})}{\partial \theta} \right]_{\theta=\varphi(\mathbf{M})} = \vec{0}, \forall \mathbf{M} \in U.$$

By the Strong Law of Large Numbers, \mathbf{M} will be in U almost surely. Thus the minimizer $\hat{\theta}$ that makes $Q(\theta, \mathbf{M}) = 0$ is consistent.

Given consistency, we can asymptotic normality. Consider the Taylor expansion of the observed moments around the true value. Note that for large samples, $\hat{\theta}$ sets the loss function equal to zero. In the following, h_i is the Hessian for the i -th moment evaluated at a point between $\hat{\theta}$ and θ_0 in order to give equality. Here \mathbf{n} is a diagonal matrix formed from the sample size for each scheme.

$$\begin{aligned} \mathbf{M} &= \mathcal{M}(\theta_0) + D(\hat{\theta} - \theta_0) \\ &+ \frac{1}{2} \begin{pmatrix} (\hat{\theta} - \theta_0)' h_1 (\hat{\theta} - \theta_0) \\ \vdots \\ (\hat{\theta} - \theta_0)' h_m (\hat{\theta} - \theta_0) \end{pmatrix} \\ \sqrt{\mathbf{n}}[\mathbf{M} - \mathcal{M}(\theta_0)] &= \sqrt{\mathbf{n}}D(\hat{\theta} - \theta_0) \\ &+ \frac{1}{2}\sqrt{\mathbf{n}} \begin{pmatrix} (\hat{\theta} - \theta_0)' h_1 (\hat{\theta} - \theta_0) \\ \vdots \\ (\hat{\theta} - \theta_0)' h_m (\hat{\theta} - \theta_0) \end{pmatrix} \end{aligned}$$

As the sample size goes to infinity, the term on the left is distributed as normal variate with mean $\vec{0}$ and covariance Σ . The Hessians are bounded on the closure of the above neighborhood V because they are continuous. As a result, the last term on the right goes to zero as $\hat{\theta}$ goes to θ_0 . Substituting terms gives us

$$\sqrt{\mathbf{n}}\dagger(\hat{\theta} - \theta_0) \Rightarrow N(\vec{0}, \Sigma). \square$$

REFERENCES

- [1] F. Lo Presti, N. G. Duffield, J. Horowitz, and D. Towsley, "Multicast-based inference of network-internal delay distributions," *IEEE Transactions on Networking*, vol. 10, no. 6, pp. 761–775, December 2002.
- [2] G. Liang and B. Yu, "Maximum pseudo likelihood estimation in network tomography," *IEEE Transactions on Signal Processing*, vol. 51, no. 8, pp. 2043–2053, August 2003.
- [3] Y. Tsang, M. Coates, and R. D. Nowak, "Network delay tomography," *IEEE Transactions on Signal Processing*, vol. 51, no. 8, pp. 2125–2135, August 2003.
- [4] E. Lawrence, G. Michailidis, and V. N. Nair, "Network delay tomography using flexicast experiments," *Journal of the Royal Statistical Society, Series B.*, vol. 68, no. 5, pp. 785–813, 2006.

- [5] A. Chen, J. Cao, and T. Bu, "Network tomography: Identifiability and fourier domain estimation," *INFOCOM 2007. 26th IEEE International Conference on Computer Communications*, 2007.
- [6] Y. Gu, L. Breslau, N. G. Duffield, and S. Sen, "Gre encapsulated multicast probing: A scalable technique for measuring one-way loss," *INFOCOM 2008. 27th IEEE Conference on Computer Communications*, 2008.
- [7] E. Lawrence, G. Michailidis, and V. N. Nair, "Statistical inverse problems in active network tomography," *A Festschrift in Memory of Yehuda Vardi. IMS Lecture Notes-Monograph Series*, 2006.
- [8] N. G. Duffield and F. Lo Presti, "Network tomography from measured end-to-end delay covariance," *IEEE/ACM Transactions on Networking*, vol. 12, no. 6, pp. 978–99, December 2004.
- [9] P. McCullagh and N. J. A., *Generalized Linear Models*. Chapman & Hall, 1989.
- [10] J. Cao, D. Davis, S. Vander Wiel, and B. Yu, "Time-varying network tomography: Router link data," *Journal of the American Statistical Association*, vol. 95, no. 452, pp. 1063–1075, December 2000.
- [11] D. Bates and D. Watts, *Nonlinear Regression Analysis and Its Applications*. New York: John Wiley & Sons, 1988.
- [12] *The Network Simulator*, Information Sciences Institute, 2004, <http://www.isi.edu/nsnam/ns/>.
- [13] D. Montgomery, *Introduction to Statistical Quality Control*. John Wiley & Sons, 1997.
- [14] X. Yang, "Design of probing experiments and online monitoring of network performance," Ph.D. dissertation, University of Michigan, 2007.
- [15] B. Xi, G. Michailidis, and V. N. Nair, "Estimating network loss rates using active tomography," *Journal of the American Statistical Association*, vol. 101, no. 476, pp. 1430–1448, 2006.