Statistical Science
2004, Vol. 19, No. 3, 499–517
DOI 10.1214/088342304000000422

Network Tomography: Recent Developments

Rui Castro, Mark Coates, Gang Liang, Robert Nowak and Bin Yu

Abstract. Today's Internet is a massive, distributed network which continues to explode in size as e-commerce and related activities grow. The heterogeneous and largely unregulated structure of the Internet renders tasks such as dynamic routing, optimized service provision, service level verification and detection of anomalous/malicious behavior extremely challenging. The problem is compounded by the fact that one cannot rely on the cooperation of individual servers and routers to aid in the collection of network traffic measurements vital for these tasks. In many ways, network monitoring and inference problems bear a strong resemblance to other "inverse problems" in which key aspects of a system are not directly observable. Familiar signal processing or statistical problems such as tomographic image reconstruction and phylogenetic tree identification have interesting connections to those arising in networking. This article introduces network tomography, a new field which we believe will benefit greatly from the wealth of statistical theory and algorithms. It focuses especially on recent developments in the field including the application of pseudo-likelihood methods and tree estimation formulations.

Key words and phrases: Network tomography, pseudo-likelihood, topology identification, tree estimation.

1. INTRODUCTION

No network is an island, entire of itself; every network is a piece of an internetwork, a part of the main (with apologies to John Donne, *Devotions XVII. Meditation*). Although administrators of small-scale networks can monitor local traffic conditions and identify congestion points and performance bottlenecks, very few networks are completely isolated. The user-perceived performance of a network thus depends heavily on the performance of an internetwork, and monitoring this internetwork is extremely challenging. Diverse subnetwork ownership and the decentralized, heterogeneous and unregulated nature of the extended internetwork combine to render a coordinated measurement framework infeasible. There is no real incentive for individual servers and routers to collect and freely distribute vital network statistics such as traffic rates, link delays and dropped packet rates. Collecting all pertinent network statistics imposes an impracticable overhead expense in terms of added computational, communication, hardware and maintenance requirements. Even when data collection is possible, network owners generally regard the statistics as highly confidential. Finally, the task of relaying measurements to the locations where decisions are made consumes exorbitant bandwidth and presents scheduling and coordination nightmares.

Rui Castro is a Ph.D. student, DSP Group, Department of Electrical and Computer Engineering, Rice University, Houston, Texas 77005, USA (e-mail: rcastro@rice.edu). Mark Coates is Assistant Professor, Department of Electrical and Computer Engineering, McGill University, Montreal, Quebec, Canada H3A 2A7 (e-mail: coates@ece.mcgill.ca). Gang Liang is a Ph.D. student, Department of Statistics, University of California, Berkeley, California 94720, USA (e-mail: liang@stat.berkeley.edu). Robert Nowak is Associate Professor, Department of Electrical and Computer Engineering, University of Wisconsin, Madison, Wisconsin 53706, USA. Bin Yu is Professor, Department of Statistics, University of California, Berkeley, California 94720, USA.

Despite this state of affairs, accurate, timely and localized estimates of network performance characteristics are vital ingredients in efficient network operation. With performance estimates in hand, more sophisticated and ambitious traffic control protocols and dynamic routing algorithms can be designed. Quality-of-service guarantees can be provided if available bandwidth can be gauged; the resulting service-level agreements can be verified. Detecting anomalous or malicious behavior becomes a more achievable task.

Usually we cannot directly measure the aspects of the system that we need to make informed decisions. However, we can frequently make useful measurements that do not require special cooperation from internal network devices and do not inordinately impact network load. Sophisticated methods of active network probing or passive traffic monitoring can generate network statistics that indirectly relate to the performance measures we require. Subsequently, we can apply inference techniques, derived in the context of other statistical inverse problems, to extract the hidden information of interest.

This article surveys the field of inferential network monitoring or *network tomography*, highlighting challenges and open problems, and identifying key issues that must be addressed. It builds upon the signal processing survey paper by Coates, Hero, Nowak and Yu (2002b) and focuses on recent developments in the field. The task of inferential network monitoring demands the estimation of a potentially very large number of spatially distributed parameters. To successfully address such large-scale estimation tasks, researchers adopt models that are as simple as possible but do not introduce significant estimation error. Such models are not suitable for intricate analysis of network queuing dynamics and fine time-scale traffic behavior, but they are often sufficient for inference of performance characteristics. The approach shifts the focus from detailed queuing analysis and traffic modeling (Kelly, Zachary and Ziedins, 1996; Chao, Miyazawa and Pinedo, 1999) to careful design of measurement techniques and large-scale inference strategies.

Measurement may be passive (monitoring traffic flows and sampling extant traffic) or active (generating probe traffic). In either case, statistical models should be developed for the measurement process, and the temporal and spatial dependence of measurements should be assessed. These are active areas of research in network tomography that we do not directly address in this paper (see Section 5 for a summary of future directions). If existing traffic is being used to sample the state of the network, care must be taken that the temporal and spatial structure of the traffic process does not bias the sample. If probes are used, then the act of measurement must not significantly distort the network state. Design of the measurement methodology must take into account the limitations of the network. As an example, the clock synchronization required for measurement of one-way packet delay is extremely difficult.

Once measurement has been accomplished, statistical inference techniques can be applied to determine performance attributes that cannot be directly observed. When attempting to infer a network performance measure, measurement methodology and statistical inference strategy must be considered *jointly*. In work thus far in this area, a broad array of statistical techniques has been employed: complexity-reducing hierarchical statistical models; moment- and likelihood-based estimation; expectation–maximization and Markov chain Monte Carlo algorithms. However, the field is still in the embryonic phase, and we believe that it can benefit greatly from the wealth of extant statistical theory and algorithms.

In this article, we focus exclusively on inferential network monitoring techniques that require minimal cooperation from network elements that cannot be directly controlled. Numerous tools exist for active and passive measurement of networks (see *http://www.caida.org/tools* for a survey). The tools measure and report internetwork attributes such as bandwidth, connectivity and delay, but they do not attempt to use the recorded information to infer any performance attributes that have not been directly measured. The majority of the tools depend on accurate reporting by all network elements traversed during measurement.

The article commences by reviewing the area of internetwork inference and tomography, and provides a simple, generalized formulation of the network tomography problem. In Section 3 we describe a pseudo-likelihood approach to network tomography that addresses some of the scalability limitations of existing techniques. We consider the problem of determining the connectivity structure or topology of a network and relate this task to the problem of hierarchical clustering. We introduce new likelihood-based hierarchical clustering methods and results for identifying network topology. Finally, we identify open problems and provide our vision of future challenges.

2. NETWORK TOMOGRAPHY

2.1 Network Tomography Basics

Large-scale network inference problems can be classified according to the type of data acquisition and the performance parameters of interest. To discuss these distinctions, we require some basic definitions. Consider the network depicted in Figure 1. Each node represents a computer terminal, router or subnetwork (consisting of multiple computers/routers). A connection between two nodes is called a *path*. Each path consists of one or more *links*—direct connections with no intermediate nodes. The links may be unidirectional or bidirectional, depending on the level of abstraction and the problem context. Each link can represent a chain of *physical* links connected by intermediate routers. Messages are transmitted by sending *packets* of bits from a *source* node to a *destination* node along a path which generally passes through several other nodes.

Broadly speaking, large-scale network inference involves estimating network performance parameters based on traffic measurements at a limited subset of the nodes. Vardi (1996) was one of the first researchers to rigorously study this sort of problem and he coined the term *network tomography* due to the similarity between network inference and medical tomography. Two forms of network tomography have been addressed in the recent literature: (1) link-level parameter estimation based on end-to-end, path-level traffic measurements (*http://gaia.cs.umass.edu/minc*; Cáceres, Duffield, Horowitz and Towsley, 1999; Ratnasamy and McCanne, 1999; Coates and Nowak, 2000; Harfoush, Bestavros and Byers, 2000; Duffield, Lo Presti, Paxson and Towsley, 2001; Shih and Hero, 2001; Ziotopolous, Hero and Wasserman, 2001; Lo Presti, Duffield, Horowitz and Towsley, 2002; Tsang, Coates and Nowak, 2003) and (2) sender–receiver path-level traffic intensity estimation based on link-level traffic measurements (Vanderbei and Iannone, 1994; Vardi, 1996; Tebaldi and West, 1998; Cao, Davis, Vander Wiel and Yu, 2000a; Cao, Vander Wiel, Yu and Zhu, 2000b; Liang and Yu, 2003b).

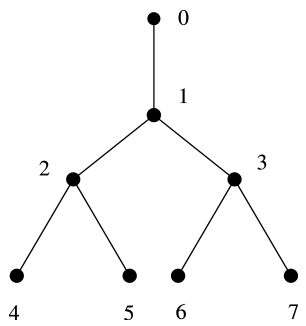

FIG. 1. *An arbitrary virtual multicast tree with four receivers.*

In link-level parameter estimation, the traffic measurements typically consist of counts of packets transmitted and/or received between source and destination nodes or time delays between packet transmissions and receptions. The goal is to estimate the loss rate or the queuing delay on each link. The measured time delays are due to both propagation delays and router processing delays along the path. The path delay is the sum of the delays on the links that comprise the path; the link delay comprises both the propagation delay on that link and the queuing delay at the routers that lie along that link. A packet is dropped if it does not successfully reach the input buffer of the destination node. Link delays and occurrences of dropped packets are inherently random. Random link delays can be caused by router output buffer delays, router packet servicing delays and propagation delay variability. Dropped packets on a link are usually due to overload of the finite output buffer of one of the routers encountered when traversing the link, but may also be caused by equipment downtime due to maintenance or power failures. Random link delays and packet losses become particularly substantial when there is a large amount of cross-traffic competing for service by routers along a path.

In path-level traffic intensity estimation, the measurements consist of counts of packets that pass through nodes in the network. In privately owned networks, the collection of such measurements is relatively straightforward. Based on these measurements, the goal is to estimate how much traffic originated from a specified node and was destined for a specified receiver. The combination of the traffic intensities of all these origin–destination pairs forms the *origin–destination traffic matrix*. In this problem not only are the node-level measurements inherently random, but the parameter to be estimated (the origin–destination traffic matrix) must itself be treated not as a fixed parameter, but as a random vector. Randomness arises from the traffic generation itself, rather than perturbations or measurement noise.

The inherent randomness in both link-level and path-level measurements motivates the adoption of statistical methodologies for large-scale network inference and tomography. Many network tomography problems can be roughly approximated by the (not necessarily Gaussian) linear model

$$\mathbf{Y}_t = \mathbf{A}\mathbf{X}_t + \boldsymbol{\varepsilon}, \tag{1}$$

where $\mathbf{Y}_t$ is a vector of measurements (e.g., packet counts or end-to-end delays) recorded at a given time t at a number of different measurement sites, $\mathbf{A}$ is a *routing matrix*, $\boldsymbol{\varepsilon}$ is a noise vector and $\mathbf{X}_t$ is a vector of time-dependent packet parameters (e.g., mean delays, logarithms of packet transmission probabilities over a link or the random origin–destination traffic vector). In some cases the vector $\mathbf{X}_t$ is a random vector with an underlying parameterized distribution $f(\mathbf{X}_t|\boldsymbol{\theta}_t)$ (see the example in Section 3.1), and it is the parameters $\boldsymbol{\theta}_t$ that interest us. Typically, but not always, $\mathbf{A}$ is a binary matrix (the i, jth element is equal to 1 or 0) that captures the topology of the network. In this paper, we consider the problems of using the observations $\mathbf{Y}_t$ to estimate $\boldsymbol{\theta}_t$ (see Section 3.1), $\mathbf{X}_t$ (see Section 3.2) or $\mathbf{A}$ (see Section 4).

What sets the large-scale network inference problem (1) apart from other network inference problems is the potentially very large dimension of $\mathbf{A}$ which can range from a half a dozen rows and columns for a few packet parameters and a few measurement sites in a small local area network, to thousands or tens of thousands of rows and columns for a moderate number of parameters and measurements sites in the Internet. The associated high-dimensional problems of estimating $\mathbf{X}_t$ are specific examples of *inverse problems*. Inverse problems have a very extensive literature (O'Sullivan, 1986). Solution methods for such inverse problems depend on the nature of the noise $\boldsymbol{\varepsilon}$ and the $\mathbf{A}$ matrix, and typically require iterative algorithms since they cannot be solved directly. In general, $\mathbf{A}$ is not full rank, so that identifiability concerns arise. Either one must be content to resolve only linear combinations of the parameters or one must employ statistical means to introduce regularization and induce identifiability. Both tactics are utilized in the examples in later sections of the article. In most of the large-scale Internet inference and tomography problems studied to date, the components of the noise vector $\boldsymbol{\varepsilon}$ are assumed to be approximately independent Gaussian, Poisson, binomial or multinomial distributed. When the noise is Gaussian distributed with covariance independent of $\mathbf{AX}_t$, methods such as recursive linear least squares can be implemented using conjugate gradient, Gauss–Seidel and other iterative equation solvers. When the noise is modeled as Poisson, binomial or multinomial distributed, more sophisticated statistical methods, such as reweighted nonlinear least squares, maximum likelihood via expectation–maximization (EM) and maximum a posteriori via Markov chain Monte Carlo (MCMC) algorithms, become necessary.

3. PSEUDO-LIKELIHOOD APPROACHES

In developing methods to perform network tomography, there is a trade-off between statistical efficiency (accuracy) and computational overhead. In the past, researchers have addressed the extreme computational burden posed by some of the tomographic problems, developing suboptimal but lightweight algorithms, including a fast recursive algorithm for link delay distribution inference in a multicast framework (Lo Presti et al., 2002) and a method-of-moments approach for origin–destination matrix inference (Vardi, 1996). More accurate but computationally burdensome approaches have also been explored, including maximum-likelihood methods (Coates and Nowak, 2000; Tsang, Coates and Nowak, 2003; Cao et al., 2002a), but in general they are too intensive computationally for any network of reasonable scale.

More recently, we proposed a unified pseudo-likelihood approach (Liang and Yu, 2003a, b) that eases the computational burden but maintains good statistical efficiency. The idea of modifying likelihood is not new, and many modified likelihood models have been proposed, for example, pseudo-likelihood for Markov random fields by Besag (1974, 1975), partial likelihood for hazards regression by Cox (1975) and quasi-maximum likelihood for finance models by White (1994). In this section, we decribe the pseudo-likelihood approach. We explore two concrete examples: (1) internal link delay distribution inference through multicast end-to-end measurements and (2) origin–destination (OD) matrix inference through link traffic counts (the OD matrix specifies the volume of traffic between a source and a destination).

The network tomography model we consider in this section is a special case of (1), in which the error term $\boldsymbol{\varepsilon}$ is omitted for further simplification. Hence the model can be rewritten as

$$\mathbf{Y} = \mathbf{AX}, \tag{2}$$

where $\mathbf{X} = (X_1, \ldots, X_J)'$ is a J-dimensional vector of network dynamic parameters (e.g., link delay, traffic flow counts at a particular time interval), $\mathbf{Y} = (Y_1, \ldots, Y_I)'$ is an I-dimensional vector of measurements and $\mathbf{A}$ is an $I \times J$ routing matrix.

As mentioned before, $\mathbf{A}$ is not full rank in a general network tomography scenario, where typically $I \ll J$; hence, constraints have to be introduced to ensure the identifiability of the model. A key assumption is that all components of $\mathbf{X}$ are independent of each other. Such an assumption does not hold strictly in a real network

due to the temporal and spatial correlations between network traffic, but it is a good first-step approximation. Furthermore, we assume that

$$X_j \sim f_j(\theta_j), \quad j = 1, \ldots, J, \tag{3}$$

where f_j is a density function and θ_j is its parameter. Then the parameter of the whole model is $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_J)$. In our first network tomography example, that of link-level delay distribution estimation, the goal is estimation of $\boldsymbol{\theta}$; in the second example, it is estimation of the actual $\mathbf{X}_t$.

The main idea of the pseudo-likelihood approach is to decompose the original model into a series of simpler subproblems by selecting pairs of rows from the routing matrix $\mathbf{A}$ and to form the pseudo-likelihood function by multiplying the marginal likelihoods of such subproblems. Let S denote the set of subproblems by selecting all possible pairs of rows from the routing matrix $\mathbf{A}$: $S = \{s = (i_1, i_2) : 1 \le i_1 < i_2 \le I\}$. Then for each subproblem $s \in S$, we have

$$\mathbf{Y}^s = \mathbf{A}^s \mathbf{X}^s, \tag{4}$$

where $\mathbf{X}^s$ is the vector of network dynamic components involved in the given subproblem s, $\mathbf{A}^s$ is the corresponding subrouting matrix and $\mathbf{Y}^s = (Y_{i_1}, Y_{i_2})'$ is the observed measurement vector of s. Let θ^s be the parameter of s and let $p^s(\mathbf{Y}^s; \theta^s)$ be its marginal likelihood function. Usually subproblems are dependent, but ignoring such dependencies, the pseudo-likelihood function can be written as the product of marginal likelihood functions of all subproblems, that is, given observation $y_1, \ldots, y_T$, the pseudo-log-likelihood function is defined as

$$L^p(y_1, \ldots, y_T; \theta) = \sum_{t=1}^{T} \sum_{s \in S} l^s(y_t^s; \theta^s), \tag{5}$$

where $l^s(\mathbf{Y}^s; \theta^s) = \log p^s(\mathbf{Y}^s; \theta^s)$ is the log-likelihood function of subproblem s. Maximizing the pseudo-log-likelihood function L^p gives the maximum-pseudo-likelihood estimate (MPLE) of parameter θ. Maximizing the pseudo-likelihood is not an easy task because $L^p(y_1, \ldots, y_T; \theta)$ is a summation of many functions. Since the maximization of the pseudo-likelihood function is a typical missing value problem, a pseudo-EM algorithm (a variant of the EM algorithm; Liang and Yu, 2003a, b), is employed to maximize the function $L^p(y_1, \ldots, y_T; \theta)$. Let $l^s(\mathbf{X}^s; \theta^s)$ be the log-likelihood function of a subproblem s given the complete data $\mathbf{X}^s$ and let $\theta^{(k)}$ be the estimate of θ obtained in the kth step. The objective function $Q(\theta, \theta^{(k)})$ to be maximized in the $(k+1)$st step of the pseudo-EM algorithm is defined as

$$Q(\theta, \theta^{(k)}) = \sum_{s \in S} \sum_{t=1}^{T} \mathrm{E}_{\theta^{s(k)}}\big(l^s(x_t^s; \theta^s) | y_t^s\big), \tag{6}$$

which is obtained by assuming the independence of subproblems in the expectation step. The starting point of the pseudo-EM algorithm can be arbitrary, but just as in the EM algorithm, care needs to be taken to ensure that the algorithm does not converge to a local maximum.

There are several points worth noting in constructing the pseudo-likelihood function:

1. Selecting three or more rows each time may also be reasonable to construct a pseudo-likelihood function, but there is a trade-off between the computational complexity incurred and the estimation efficiency achieved by taking more dependence structures into account. The experience with the two examples we discuss later shows that selecting two rows each time gives satisfactory estimation results while keeping the computational cost within a reasonable range.
2. Currently all possible pairs are selected to construct the pseudo-likelihood function, but a subset can be judiciously chosen to reduce the computation. The pseudo-likelihood is obtained by assuming all subproblems to be independent. Although this assumption is frequently violated, we obtain, under mild conditions, the consistency and asymptotic normality of maximum pseudo-likelihood estimates (Liang and Yu, 2003a). Furthermore, the performances of the full- and pseudo-likelihood approaches are comparable at least in the two examples below.

In summary, the pseudo-likelihood approach keeps a good balance between the computational complexity and the statistical efficiency of the parameter estimation. Even though the basic idea of divide-and-conquer is not new, it is very powerful when combined with pseudo-likelihood for large network problems.

3.1 Example: Multicast Delay Distribution Inference

The Multicast-Based Inference of Network-Internal Characteristics (MINC) project (*http://gaia.cs.umass.edu/minc*) pioneered the use of multicast probing for network link-level queuing delay distribution estimation. A similar approach through unicast end-to-end measurements can be found in Tsang, Coates and Nowak (2003). Consider a general multicast tree, as depicted in Figure 1. Each node is labeled with a number

and we adopt the convention that link i connects node i to its parental node. Each probing packet with a time stamp sent from root node 0 will be received by all end receivers 4–7. For any pair of receivers, each packet experiences the same amount of delay over the common path. For instance, copies of the same packet received at receiver 4 and 5 experience the same amount of delay on the common links 1 and 2. Measurements are made at end receivers, so only the aggregated delays over the paths from root to end receivers are observed.

Due to the aggregation of the measured delays, model (2) can be naturally applied to the problem of the multicast internal link (queuing) delay distribution inference. For each probing packet, $\mathbf{X}$ is the vector of unobserved delays over each link and $\mathbf{Y}$ is the vector of observed path-level delays at each end receiver. Vector $\mathbf{A}$ is an $I \times J$ routing matrix determined by the multicast spanning tree, where I is the number of end receivers and J is the number of internal links. For the multicast tree depicted in Figure 1, (2) can be written as

$$\begin{pmatrix} Y_1 \\ Y_2 \\ Y_3 \\ Y_4 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X_1 \\ X_2 \\ \vdots \\ X_7 \end{pmatrix},$$

where $Y_1, \ldots, Y_4$ are the measured delays at end receivers $4, \ldots, 7$ and $X_1, \ldots, X_7$ are the delays over internal links ending at nodes $1, \ldots, 7$.

Each link has a certain amount of minimal delay (the propagation delay on the link), which is assumed to be known beforehand. After compensating for the minimal delay of each link, a discretization scheme is imposed on link-level delay by Lo Presti et al. (2002) such that X_j takes finite possible values $\{0, q, 2q, \ldots, mq, \infty\}$, where q is the bin width and m is a constant. Therefore, each X_j is a discrete random variable whose possible values are $\{0, q, 2q, \ldots, mq, \infty\}$ with respective probabilities $\theta_j = (\theta_{j0}, \theta_{j1}, \ldots, \theta_{jm}, \theta_{j\infty})$. When the delay is infinite, it implies the packet is lost during the transmission.

As discussed by Lo Presti et al. (2002), the bin size q is chosen beforehand and then the delay measurements are discretized accordingly. The bin size and the maximum observed queuing delay provide an indication of the required value of m. This process introduces a quantization error such that the equation $\mathbf{Y} = A\mathbf{X}$ does not hold exactly: the error diminishes as q is reduced. The choice of q thus represents a trade-off between the accuracy of estimation and cost of computations, because a smaller bin size entails higher dimension of delay distributions. In experiments and simulations (Lo Presti et al., 2002; Liang and Yu, 2003a) it has been observed that the parameter estimation has similar accuracy over a significant range of q (from very small bin size to bin size of the same order as the mean link delays). In practice, we choose a reasonable q based on the spread of the delay measurements and prior knowledge of network topology and network traffic. If the resultant distributions appear too coarse, we repeat the inference with a finer bin size.

To ensure identifiability, we consider only canonical multicast trees (Lo Presti et al., 2002), defined as those that satisfy

$$\theta_{j0} = P(X_j = 0) > 0, \quad j = 1, \ldots, J,$$

that is, each individual packet has a positive probability to have zero delay over any internal link. The goal of the multicast delay distribution inference is to estimate the delay distribution parameters θ_j.

For the problem of multicast internal delay inference, the maximum-likelihood method is usually infeasible for networks of realistic size, because the likelihood function involves finding all possible internal delay vectors $\mathbf{X}$ which can account for each observed delay vector $\mathbf{Y}$. We can show that the computational complexity grows at a nonpolynomial rate. Lo Presti et al.'s (2002) recursive algorithm is a computationally efficient method for estimating internal delay distributions by solving a set of convolution equations. Our pseudo-likelihood approach is motivated by the decomposition of multicast spanning trees depicted in Figure 2. A virtual two-leaf subtree is formed by considering only two receivers R_1 and R_3 in the original multicast tree. The marginal likelihood function of the virtual two-leaf subtree is tractable because of its simple structure. For a multicast tree with I end receivers, there is a total of $I(I-1)/2$ subtrees: different subtrees contain delay distribution information on

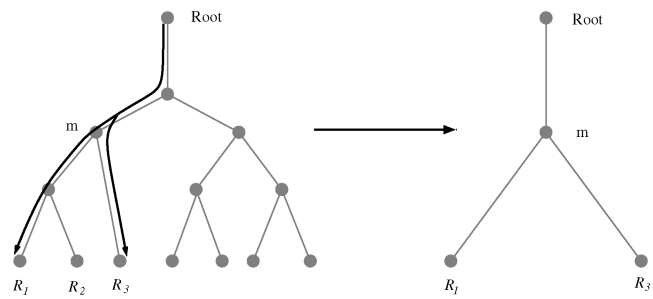

FIG. 2. *Pseudo-likelihood*: *subtree decomposition*.

different virtual links. Combining all subproblems by ignoring their dependencies enables us to recover link delay distributions. Since forming the subtree is equivalent to selecting two rows from the routing matrix $\mathbf{A}$, the pseudo-likelihood method is applicable to the general network tomography model (2).

Given multiple observed end-to-end multicast measurements $\{y_1, \ldots, y_T\}$, the pseudo-log-likelihood function can be written as

$$L^p(y_1, \ldots, y_T; \theta) = \sum_{s \in S} \sum_{t=1}^{T} \log p(\mathbf{Y}^s = y_t^s | \theta^s),$$

where $p(\mathbf{Y}^s = y_t^s | \theta^s)$ is the probability of the delay measurement $\mathbf{Y}^s$ of subtree s being y_t^s when its link delay distributions are θ^s. The pseudo-log-likelihood function is maximized in an EM fashion with small variations (Liang and Yu, 2003a).

We evaluate the performance of the pseudo-likelihood methodology by model simulations carried out on the four-leaf multicast tree depicted in Figure 1. Due to the small size of the multicast tree, the maximum-likelihood estimation (MLE) method can be implemented, and so we can compare the performance of maximum-pseudo-likelihood estimation (MPLE) with that of MLE and also with that of the recursive algorithm of Lo Presti et al. (2002). For each link the bin size $q = 1$ and the number of bins m is set to be 14. During each simulation 2000 i.i.d. multicast delay measurements are generated, with each internal link having an independent discrete delay distribution. Figure 3 shows the delay distribution estimates of three arbitrarily selected links along with their true delay distributions in one such experiment. The plot shows that both MPLE and MLE capture most of the link delay distributions and their performance is comparable, whereas the recursive algorithm sometimes gives estimates far from the truth.

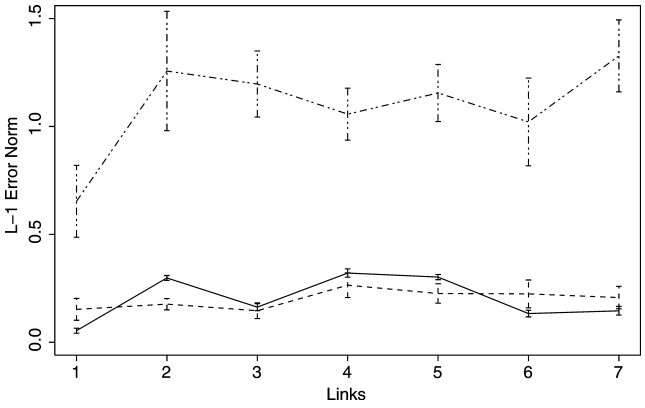

FIG. 4. *Link L_1 error norm averaged over* 30 *simulations. The solid line is the MPLE, the dashed line is the MLE and the dotted line is the recursive algorithm of Lo Presti et al.* (2002). *For each link the vertical bar shows the standard deviation of the L_1 error norm for the given link.*

A further comparison is illustrated in Figure 4, which shows the L_1 error norm of MLE and MPLE for each link, as averaged over 30 independent simulations. For each link the L_1 error norm is simply the sum of the absolute differences between probability estimates and the true probabilities. As a common measure of the performance of density estimates, the L_1 error norm enjoys several theoretical advantages as discussed by Scott (1992). The plot shows that MLE and MPLE have comparable estimation performance for tracking link delay distributions, while the recursive algorithm has much larger L_1 errors on all links. Meanwhile, we can see that MPLE has smaller standard deviation

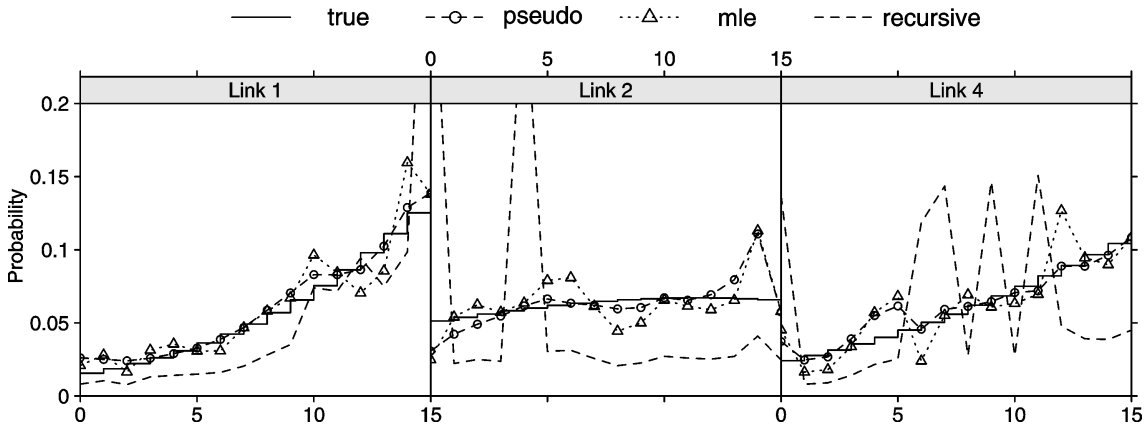

FIG. 3. *Delay distribution estimates of three arbitrarily selected internal links*: *link* 1, *link* 2 *and link* 4. *The solid step function is the true distribution, the dashed line with circles is the MPLE, the dotted line with triangles is the MLE and the dashed line is the recursive estimate.*

on L_1 error norm than MLE on all links, implying that MPLE is more robust than MLE. This is because the pseudo-likelihood function, which is a product of less complex likelihood functions on subproblems, has a nicer surface than the full-likelihood function (Blackwell, 1973).

3.2 Example: Origin–Destination Traffic Matrix Inference

Vardi (1996) was the first researcher to study the problem of inferring the origin–destination (OD) traffic matrix from link traffic counts at router interfaces (his work originated in 1993, but appeared in 1996). In this problem the observations are the link counts at router interfaces and the OD traffic variables to be estimated are linear aggregations of these link counts. Assuming i.i.d. Poisson distributions for the OD traffic byte counts on a general network topology, Vardi demonstrated the identifiability of the Poisson model and developed an EM algorithm to estimate Poisson parameters in both deterministic and Markov routing schemes. To reduce the computational complexity of the EM algorithm, he proposed a moment estimation method and briefly discussed the normal model as an approximation to the Poisson model. Follow-up works treated the special case involving a single set of link counts: Vanderbei and Iannone (1994) applied the EM algorithm and Tebaldi and West (1998) presented a Bayesian perspective and a Markov chain Monte Carlo implementation.

Cao et al. (2000a) used real data to revise the Poisson model and to address the nonstationary aspect of the problem. They represented link count measurements as summations of various OD counts that are modeled as independent random variables. Even though the transmission control protocol (TCP), which governs the flow of the majority of Internet traffic, generates feedback that creates dependence, direct measurements of OD traffic indicate that the dependence between traffic in opposite directions is weak. This renders the independence assumption a reasonable approximation. Time-varying traffic matrices estimated from a sequence of link counts are validated by comparing the estimates with actual OD counts that were collected by running Cisco's NetFlow software on a small network depicted in Figure 5b. Such direct point-to-point measurements are often not available because they require additional router CPU resources, can reduce packet forwarding efficiency and involve a significant administrative burden when used on a large scale.

The network tomography model specified by (2) is applicable to the OD matrix inference through link traffic counts since the observed link traffic counts are linear aggregations of the unobserved OD variables to be estimated. Here $\mathbf{Y} = (Y_1, Y_2, \ldots, Y_I)'$ is the vector of observed traffic byte counts measured on each link interface during a given time interval and $\mathbf{X} = (X_1, X_2, \ldots, X_J)'$ is the corresponding vector of unobserved true OD traffic byte counts at the same time period. Vector $\mathbf{X}$ is called the OD traffic matrix, even though it is arranged as a column vector for notational convenience. Under a fixed routing scheme, $\mathbf{Y}$ is determined uniquely by $\mathbf{X}$ through the $I \times J$ routing matrix $\mathbf{A}$, in which I is the number of measured incoming/outgoing unidirectional links and J is the number of possible OD pairs. In contrast to multicast delay inference, the ultimate goal of the OD traffic matrix inference is to estimate the underlying random OD traffic $\mathbf{X}$ given the observed link traffic $\mathbf{Y}$. To achieve this goal, we first estimate the mean of the traffic vector, as described below.

Each component of $\mathbf{X}$ is assumed to be independent normally distributed and to satisfy the mean–variance relationship $X_j \sim N(\lambda_j, \phi\lambda_j^c)$ independently, where ϕ is a positive scalar applicable to all OD pairs and c is a power constant. For the examples below, our exploratory data analysis has shown that the Gaussian

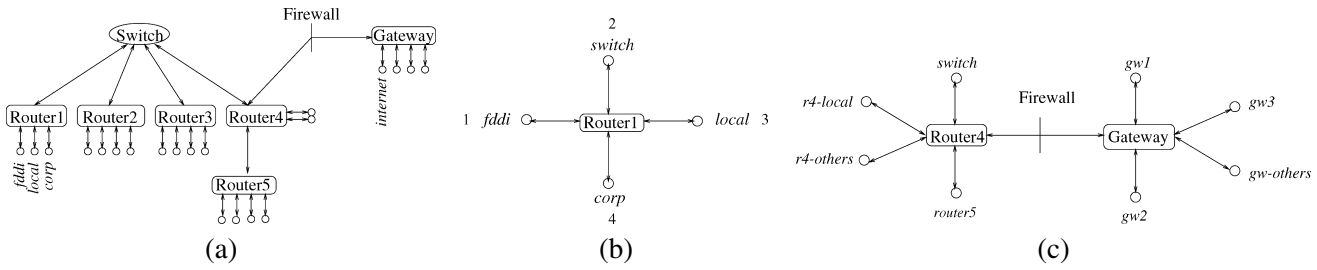

FIG. 5. (a) *A router network at Lucent Technologies*. (b) *Network topology around router* 1. (c) *A two-router network around router* 4 *and gateway.*

distribution does capture the characteristics of OD traffic flows well. As a second-order approximation to real network traffic, the mean–variance relationship is critical in the Gaussian model. It is well-known that real network traffic exhibits strong long range dependence (Leland, Taqqu, Willinger and Wilson, 1994), which is in general incompatible with the generation of normal distributions. Despite this phenomenon, several researchers have suggested that the power law describes well the mean–variance relationship for a large load of aggregated network traffic (Rolls, 2003; Morris and Lin, 2000).

The assumption implies that

$$\mathbf{Y} = \mathbf{AX} \sim N(\mathbf{A}\lambda, \mathbf{A}\Sigma\mathbf{A}'), \tag{7}$$

where $\lambda = (\lambda_1, \ldots, \lambda_J)$ and $\Sigma = \phi\,\mathrm{diag}(\lambda_1^c, \ldots, \lambda_J^c)$, so the parameter of the full model is $\theta = (\phi, \lambda)$. The mean–variance relationship is a key assumption to ensure the identifiability of the normal model. It implies that an OD pair with large traffic byte counts tends to have large variance with the same scale factor ϕ. For the power constant c, both $c = 1$ and 2 work well with the Lucent network data as shown by Cao et al. (2000a, b). Because $c = 1$ or $c = 2$ give similar results, in this paper, we use $c = 1$ as in Cao et al. (2000b), but note that the pseudo-likelihood method can deal with $c = 2$ without any additional technical difficulties. Then given observed link traffic count vectors $\{y_1, \ldots, y_T\}$, the pseudo-log-likelihood function can be written as

$$L^p(\lambda, \Sigma) \propto -\frac{1}{2}\sum_{s\in S}\sum_{t=1}^{T}\{-\log|\mathbf{A}^s\Sigma_s\mathbf{A}^{s\prime}| + (y_t^s - \mathbf{A}^s\lambda^s)'(\mathbf{A}^s\Sigma_s\mathbf{A}^{s\prime})^{-1} \cdot (y_t^s - \mathbf{A}^s\lambda^s)\},$$

where for a subproblem s, λ^s is its mean traffic vector, Σ_s is its covariance matrix and $\mathbf{A}^s$ is the subrouting matrix. The maximization of the pseudo-log-likelihood function is realized by the pseudo-EM algorithm as well (Liang and Yu, 2003a).

Cao et al. (2000a) addressed the nonstationarity of the data using a local likelihood model. For any given time interval t, analysis is based on a likelihood function derived from the observations within a symmetric window of size w around t (e.g., in the experiments described below, $w = 11$ corresponds to observations within about an hour in real time). Within this window, an i.i.d. assumption is imposed (as a simplified and yet practical way to treat the approximately stationary observations within the window). Maximum-likelihood estimation is carried out for the parameter estimation via a combination of the EM algorithm and a second-order global optimization routine. The componentwise conditional expectations of the OD traffic, given the link traffic, the estimated parameters and the positivity constraints on the OD traffic, are used as the initial estimates of the OD traffic. The linear equation $\mathbf{y} = \mathbf{Ax}$ is enforced via the iterative proportional fitting algorithm (Cao et al., 2000a; Csiszár, 1975) to obtain the final estimates of the OD traffic. The positivity and the linear constraints are very important final steps to get reliable estimates of the OD traffic, in addition to the implicit regularization introduced by the i.i.d. statistical model. To smooth the parameter estimates, a random walk model also was applied by Cao et al. (2000a) to the logarithm of the parameters λ and ϕ over the time windows.

Even though the full-likelihood method described by Cao et al. (2000a) uses all available information to estimate parameter values and the OD traffic vector $\mathbf{X}$, it does not computationally scale to networks with many nodes. In general, if there are N_e edge nodes, the number of floating point operations needed to compute the MLE is at least proportional to N_e^5 after exploiting sparse matrix calculation in each iteration. Assuming that the average number of links between an OD pair is $O(\sqrt{N_e})$, it can be shown that the overall computational complexity of each iteration of the pseudo-EM algorithm is $O(N_e^{3.5})$. Compared with the complexity of the full-likelihood $O(N_e^5)$, the pseudo-likelihood approach reduces the computational complexity considerably. Moreover, the pseudo-likelihood approach fits into the framework of the distributed computing, which is beneficial to realistic applications.

First, to compare with results presented by Cao et al. (2000a) we analyzed the same raw network OD traffic data collected on February 22, 1999 for the *Router* 1 network depicted in Figure 5b. Figures 6 and 7 show the estimated OD traffic from MPLE and MLE based on the link traffic for the subnetwork along with the validation OD traffic via NetFlow. Figure 6 gives the full scale plot and Figure 7 is the zoomed-in scale (20×). From the plot we can see that estimated OD traffic from both MPLE and MLE agrees well with the NetFlow measured OD traffic for large measurements, but not so well for small measurements where the Gaussian model is a poor approximation. From the point of view of origin–destination traffic engineering, it is adequate that the large traffic flows are inferred accurately. For tasks such as planning and provisioning activities, OD traffic estimates can then be used as

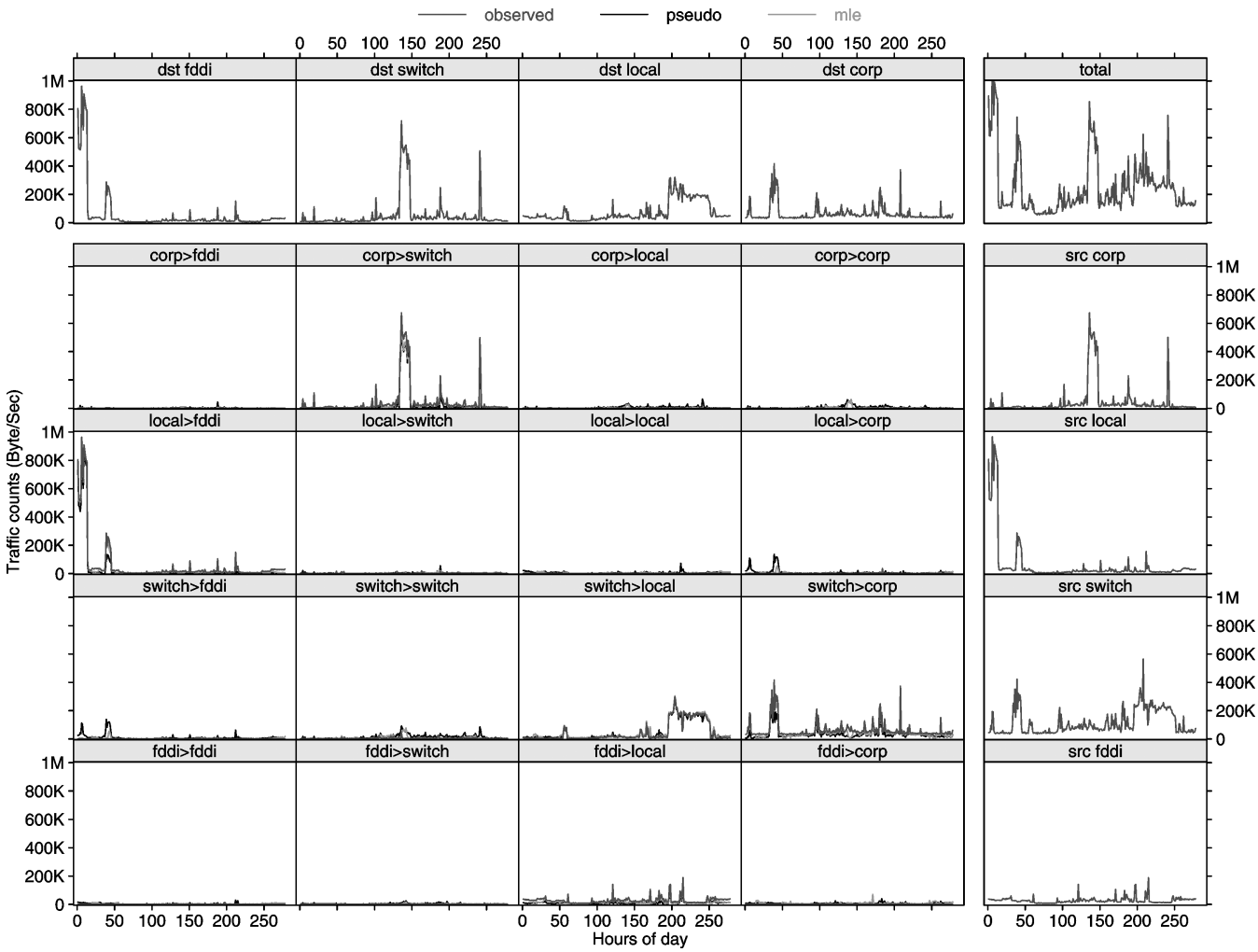

FIG. 6. *Full scale OD traffic count estimates $\hat{x}_t$ obtained from pseudo- and full-likelihood methods against the true OD traffic counts for four node network around router* 1.

inexpensive substitutes for direct measurements. The performances of MPLE and MLE are comparable in this case, but the computation of the MPLE is faster than MLE. For this example, the computations are carried out using R 1.5.0 (Ihaka and Gentleman, 1996) on a 1-GHz laptop: it takes about 12 s to compute the MPLE and about 49 s to compute the MLE in producing Figure 6.

Second, to assess the performance of MPLE more thoroughly, simulations were carried out on some larger networks through the network simulator `ns-2` (*http://www.isi.edu/nsnam/ns*). The experimental network topologies are (i) the two-router network depicted in Figure 5c and (ii) the Lucent network illustrated in Figure 5a, which comprises 21 end nodes and 27 links. From the simulation results (plots not shown), we see that both pseudo- and full-likelihood methods capture the dynamics of the simulated OD traffic under the zoomed-in scale. Table 1 summarizes the execution time for both pseudo- and full-likelihood approaches under the three different settings. From the table we can see that the pseudo-likelihood approach speeds up the computation without losing much estimation performance, so it is more scalable to larger networks.

TABLE 1
Execution times of MPLE and MLE on router networks of different sizes

Network topology	Number of edge nodes	MPLE time (s)	MLE time (s)
Figure 5b	4	12	49
Figure 5c	8	18	88
Figure 5a	21	151	2395

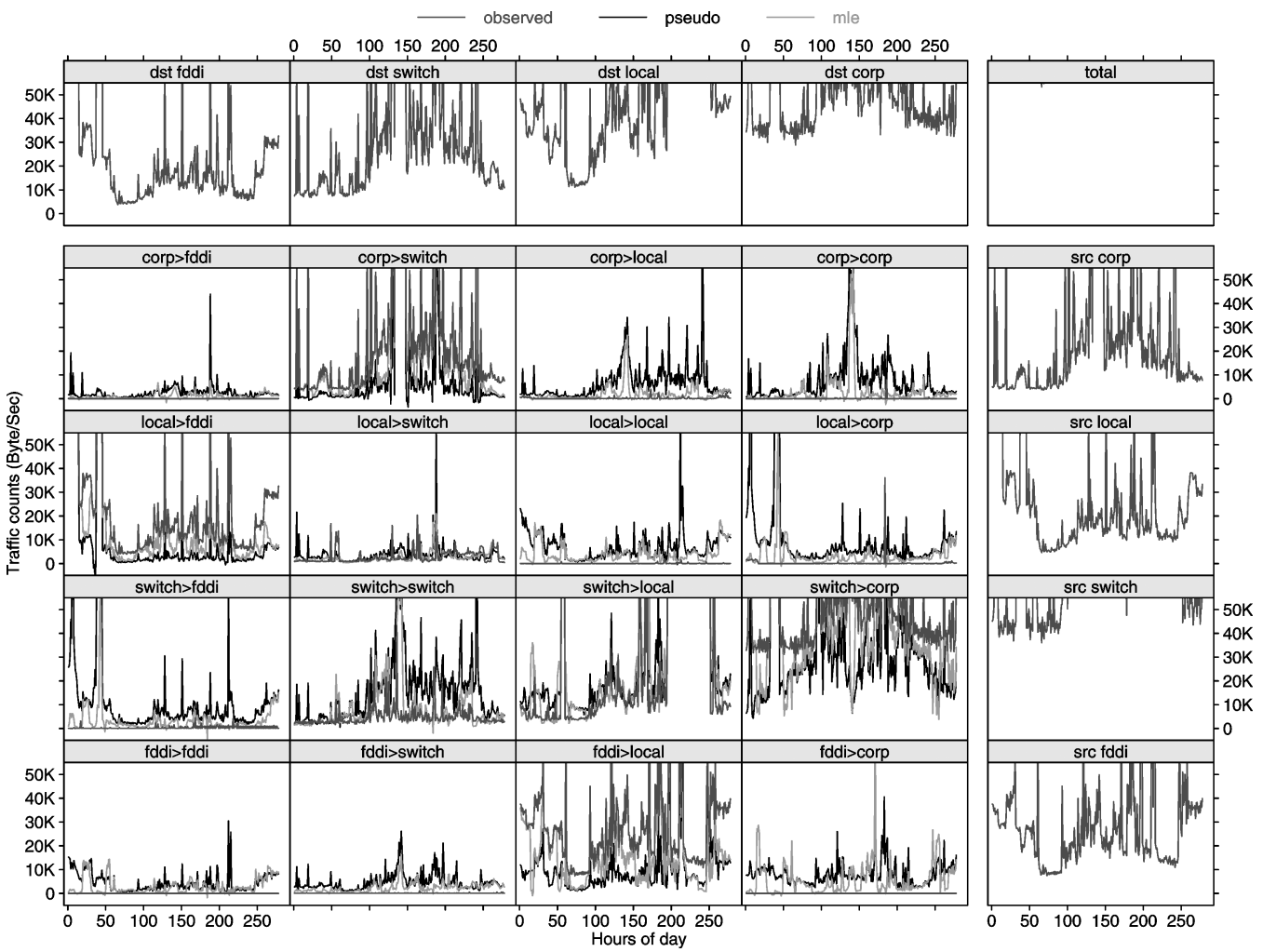

FIG. 7. *OD traffic count estimates $\hat{x}_t$ obtained from pseudo- and full-likelihood methods against the true OD traffic counts for four node network around router* 1. *The plot has been zoomed-in* 20× *to illustrate the detailed features.*

4. TOPOLOGY IDENTIFICATION

In the previous section it was assumed that the network topology was known; this knowledge is essential for successful application of the techniques described. When the topology is unknown, tools such as `traceroute` (see *http://www.caida.org/tools*) can be used in an attempt to identify it. However, these tools rely on close cooperation from the network internal devices and are incapable of detecting certain types of devices. The tools can thus determine the topology only if the network is functioning properly and network elements are prepared to cooperate and reveal themselves. These conditions are often not met and are becoming more uncommon as the Internet grows in size and speed; there is little motivation for extremely high-speed or heavily loaded switches to spend time processing requests that are not central to the process of communication. Also, the fear of malicious attacks (such as denial of service attacks) forces network administrators to block access to some diagnosis tools on routers (such as `ping` or the ability to respond to ICMP packets), preventing their use for legitimate purposes.

It is therefore desirable to develop a method for estimating topology that uses only measurements taken at the network edge, obtained without cooperation from internal devices. We consider a single source that is communicating with multiple receivers (denote the set of receiver nodes by R). The physical network topology can be represented as a directed graph, where each vertex represents a physical device (e.g., a router or a switch) and the edges correspond to the connections between those devices. In our approach we use only end-to-end measurements and do not use any network

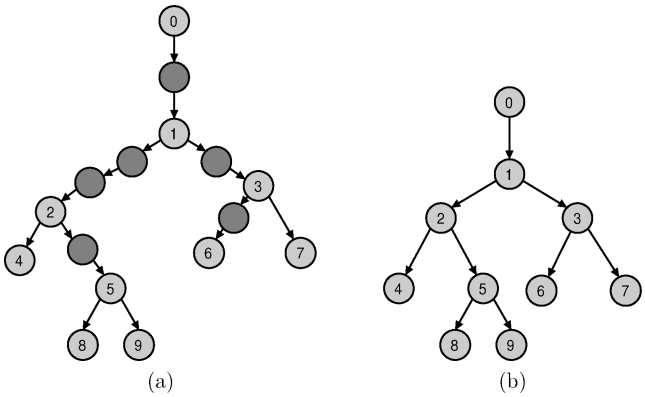

FIG. 8. (a) *Physical topology and* (b) *corresponding logical topology. The darker unnumbered nodes are devices where no branching of traffic occurs and therefore do not appear in the logical topology.*

device information, which forces us to rely solely on traffic and queueing characteristics. With this limited knowledge, it is only possible to identify the so-called logical topology (see Figure 8 for an illustration of the distinction between logical and physical topologies). In the logical topology, each vertex represents a physical network device where traffic branching occurs, that is, where two or more source–destination paths diverge. The set of vertices thus corresponds to a subset of the traversed physical devices. An edge is included between two vertices if traffic travels between the corresponding network devices and does not pass through any other devices in the included subset. Each edge corresponds to a connection between two physical devices, but the connection may include several network devices where no traffic branching occurs. We assume that the routes from the sender to the receivers are fixed during the measurement period, in which case the topology is a tree-structured graph, as in Figure 8. Every node has at least two children, apart from the root node (which has one) and the leaf nodes (which have none). If all internal nodes have exactly two children, then the tree is called binary.

Ratnasamy and McCanne (1999) first demonstrated that observations of correlations in end-to-end (multicast) loss measurements could be used to reconstruct the logical topology. Duffield, Horowitz and Lo Presti (2001) and Duffield, Horowitz, Lo Presti and Towsley (2002) then rigorously established the correctness of the proposed algorithm and developed a more general framework in which other measurements, such as delay variance, could be used. This work was extended to unicast scenarios by Bestavros, Byers and Harfoush (2002), Coates et al. (2000a) and Castro, Coates and Nowak (2004). In the following discussion, we focus on the unicast measurement procedure we proposed (Coates et al., 2002a) and the hierarchical clustering interpretation of the topology identification problem expounded by Castro, Coates and Nowak (2004).

Recall equation (1). In the topology identification problem the quantity of interest is $\mathbf{A}$, the routing matrix. Note that the entries of this matrix are only 0 or 1. The measurements $\mathbf{Y}_t$ are obtained through special measurement techniques described below and the *partial ordering* of $\mathbf{Y}_t$ can be used to determine $\mathbf{A}$. The matrix estimation formulation above is not well suited to the topology identification problem, so we formulate it below as a tree estimation exercise. One can also regard the topology discovery problem as hierarchical clustering. Within such a framework one wants to identify clusters of receivers that share certain properties. In particular, we want to identify the clusters of receiver nodes whose paths from the source node are the same up to a certain point.

Our goal is to identify the logical topology. With each internal node in a tree we associate a metric value γ_k. We consider only metrics that have a monotonic property: An internal node has a smaller metric value than any of its descendants (e.g., in Figure 8 $\gamma_5 > \gamma_2$). Examples of such metrics in networking are the average delay or delay variance experienced by a packet traveling from the source to node k or the bandwidth-related metric we describe in Section 4.2.1.

Since we do not know the topology, we cannot estimate the metric values directly, but it is possible to estimate them indirectly. Let $a(i, j)$ denote the nearest common ancestor of a given receiver pair $i, j \in R$ [e.g., $a(4, 9) = 2$]. Define $\gamma_{ij} \equiv \gamma_{a(i,j)}$. The value γ_{ij} can be regarded as a characterization of the shared portion of the paths from the root to i and j. The shared path for a pair of nodes (i, j) is the path from the root to node $a(i, j)$. In the context of hierarchical clustering, the γ_{ij} can be interpreted as *similarity* values. Note that there is an enforced symmetry in this model: $\gamma_{ij} = \gamma_{ji}$. Knowledge of the pairwise metric values and the monotonicity property suffices to completely identify the logical topology (Duffield et al., 2002).

For example, referring to Figure 8, the metric γ_{67} is greater than γ_{i7} for all $i \in R \setminus \{6, 7\}$, revealing that nodes 6 and 7 have a common parent in the logical tree. This property can be exploited recursively to devise a simple and effective bottom-up merging algorithm that identifies the complete, logical topology (Duffield et al., 2002; Castro, Coates and Nowak,

2004). These same techniques are used in agglomerative hierarchical clustering methods (Ward, 1963; Willet, 1988; Fasulo, 1999).

4.1 Likelihood Formulation and Optimization Strategies

In general, we do not have access to the exact pairwise metric values and can only observe a noisy and distorted version of them, usually obtained by actively probing the network. If we have a statistical model that relates the underlying (unknown) metric values and the measurements, we can formulate the topology identification problem as a maximum-likelihood estimation exercise.

For a given unknown tree $\mathcal{T}$ with a receiver set R, let X_{ij} be a random variable parameterized by γ_{ij} for any $i, j \in R$, $i \neq j$, and let $\boldsymbol{\gamma} = \{\gamma_{ij}\}$. Let $p(\mathbf{x}|\boldsymbol{\gamma})$ denote the joint probability density function of those random variables. A sample $\mathbf{x} \equiv \{x_{ij} : i, j \in R,\ i \neq j\}$ of the random variables X_{ij} is observed. These are the pairwise measurements recorded through a probing process such as the one described in Section 4.2.1. The maximum-likelihood tree estimate is then given by

$$\mathcal{T}^*(\mathbf{x}) = \arg\max_{\mathcal{T} \in \mathcal{F}} \sup_{\boldsymbol{\gamma} \in \mathcal{G}(\mathcal{T})} p(\mathbf{x}|\boldsymbol{\gamma}), \tag{8}$$

where $\mathcal{F}$ denotes the *forest* of all possible trees with leaves R, and $\mathcal{G}(\mathcal{T})$ is the set of all $\boldsymbol{\gamma}$'s that satisfy the monotonicity property for the tree $\mathcal{T}$. In many situations we are not interested in estimating $\boldsymbol{\gamma}$; hence, we can regard $\boldsymbol{\gamma}$ as nuisance parameters. In that case, (8) can be interpreted as a maximization of the profile likelihood (Berger, Liseo and Wolpert, 1999)

$$\mathcal{L}(\mathbf{x}|\mathcal{T}) \equiv \sup_{\boldsymbol{\gamma} \in \mathcal{G}(\mathcal{T})} p(\mathbf{x}|\boldsymbol{\gamma}). \tag{9}$$

The solution of (8) is referred to as the maximum-likelihood tree (MLT).

Under reasonable modeling assumptions the random variables X_{ij} are independent. Taking this into account yields a useful factorization of the log-likelihood. Assume that the random variables X_{ij} have densities $p(x_{ij}|\gamma_{ij})$, $i, j \in R$, $i \neq j$, with respect to a common dominating measure. Let $f_{ij}(x_{ij}|\gamma_{ij}) = \log p(x_{ij}|\gamma_{ij})$. The log-likelihood is then

$$\log p(\mathbf{x}|\boldsymbol{\gamma}) = \sum_{i \in R} \sum_{j \in R \setminus \{i\}} f_{ij}(x_{ij}|\gamma_{ij}). \tag{10}$$

The optimization problem in (8) is quite formidable. We are not aware of any method for computation of the global maximum except by a brute force examination of each tree in the forest. Consider a tree with N leaves. A very loose lower bound on the size of the forest $\mathcal{F}$ is $N!/2$. For example, if $N = 10$, then there are more than 1.8×10^6 trees in the forest. Moreover, the computation of the profile likelihood (9) is nontrivial because it involves a constrained optimization over $\mathcal{G}(\mathcal{T})$. Castro, Coates and Nowak (2004) showed that if the functions f_{ij} are concave, it is not necessary to perform the constrained optimization, since the maximum-likelihood metric value estimate for the MLT is always in the interior of the set $\mathcal{G}(\mathcal{T})$. Hence one can just compute an unconstrained optimization and subsequently check if the resulting maximizer lies in the set $\mathcal{G}(\mathcal{T})$. However, even with this simplification, it is still infeasible to search exhaustively over all candidate trees. In the following subsections we briefly describe two alternative algorithms that return tree estimates that are an approximation to the MLT.

4.1.1 *Bottom-up agglomerative procedure.* In a scenario where one can determine the true pairwise similarity metrics $\boldsymbol{\gamma}$, it is possible to reconstruct the tree topology using a simple agglomerative bottom-up procedure (Willet, 1988; Duffield et al., 2002). When we only have access to the measurements $\mathbf{x}$, conveying indirect information about $\boldsymbol{\gamma}$, we can still develop a bottom-up agglomerative clustering algorithm to estimate the true topology. This method follows the same conceptual framework as many hierarchical clustering techniques, and proceeds by repeatedly applying four steps:

1. Choose the pair of nodes with the highest similarity.
2. Merge the pair into a new node/cluster.
3. Update the similarities between the new node and the former existing nodes.
4. Repeat the procedure until only one node is left.

The crucial step is the update of the similarity values, and in many hierarchical clustering algorithms the update procedure is chosen via application-dependent heuristics (Fasulo, 1999). In our model-based approach, which relates $\boldsymbol{\gamma}$ to $\mathbf{x}$, the appropriate update of similarities arises naturally from the likelihood formulation and leads to the agglomerative likelihood tree (ALT) algorithm (Castro, Coates and Nowak, 2004).

The algorithm commences by considering a set of nodes $\mathcal{S}$, initialized to the receiver set R, and forming the estimates of the pairwise similarity metrics for each pair of nodes in the set $\mathcal{S}$, given by

$$\widehat{\gamma}_{ij} = \arg\max_{\gamma \in \mathbb{R}} \big(f_{ij}(x_{ij}|\gamma) + f_{ji}(x_{ji}|\gamma) \big), \qquad i, j \in \mathcal{S}, i \neq j. \tag{11}$$

One expects the above estimated pairwise similarities to be reasonably close to the true similarities $\boldsymbol{\gamma}$. Consider the pair of nodes such that $\widehat{\gamma}_{ij}$ is greatest, that is,

$$\widehat{\gamma}_{ij} \geq \widehat{\gamma}_{lm}, \quad \forall l, m \in \mathcal{S}.$$

We infer that i and j are the most similar nodes, implying that they have a common parent k in the tree.

Assuming that our decision is correct, the tree structure and the likelihood impose some structure on the true similarities, providing a logical way to perform the merging of similarities (see Castro, Coates and Nowak, 2004, for details). The algorithm proceeds by replacing nodes i and j with their parent k in $\mathcal{S}$. For a given node k, we denote by R_k the set of receivers which are descendants of k in the tree. Thus, at the initial stage of the algorithm $R_i = \{i\}$, and after the update step, $R_k = R_i \cup R_j$. We update the similarity estimates in $\mathcal{S}$ according to

$$\widehat{\gamma}_{kl} = \widehat{\gamma}_{lk} \equiv \arg\max_{\gamma \in \mathbb{R}} \sum_{r \in R_k} f_{rl}(x_{rl}|\gamma) + f_{lr}(x_{lr}|\gamma), \tag{12}$$

where $l \in \mathcal{S} \setminus \{k\}$.

These two steps, selecting the pair of nodes with maximum estimated similarity for merger and updating the similarities, are repeated until there is a single node in $\mathcal{S}$. Castro, Coates and Nowak (2004) formalized the concepts behind this algorithm and showed that if the underlying tree is binary and the estimated pairwise similarities are sufficiently close to the true similarities, then the ALT algorithm is equivalent to the MLT and identifies the true topology.

4.1.2 *Markov chain Monte Carlo approach.* Despite the simplicity of the ALT algorithm, it is a greedy procedure based on local decisions that involve the estimated pairwise similarities. If an incorrect local decision is made at some stage in the algorithm, then it cannot be reversed. In the topology estimation problem the measurement process is generally distributed, relying on clocks and counters at numerous network sites. It is frequently the case that several of the measurements are substantially more inaccurate than the rest. The ALT algorithm compares pairwise similarity estimates, each of which is formed from only a subset of the available measurements and is thus vulnerable to the effect of the local inaccuracies. Unlike the ALT, the MLT estimator takes a global approach: the expression to be optimized in (8) involves a contribution from all of the measurements, and identification of the MLT requires a simultaneous consideration of all the pairwise similarities. The price to pay is that identification of the MLT involves a search over the entire forest $\mathcal{F}$. In this section we propose a random search technique that efficiently searches the forest of trees and, most importantly, focuses on the likely regions of the forest.

Recall the profile likelihood defined in (9) and note that the maximum likelihood tree is the tree that maximizes $\mathcal{L}(\mathbf{x}|\mathcal{T})$. For a given set of measurements $\mathbf{x}$ we can regard the profile likelihood $\mathcal{L}(\mathbf{x}|\mathcal{T})$ as a discrete distribution over the set of possible tree topologies $\mathcal{F}$ (up to a normalizing factor). One way to search the set $\mathcal{F}$ is to sample it according to this distribution. The more likely trees are sampled more often than the less likely trees, making the search more efficient. The sampling can be implemented using the Metropolis–Hastings algorithm (Coates et al., 2002a; Hastings, 1970). For this we need to construct a Markov chain with state space $\mathcal{F}$. We allow only certain transitions. For a given state (a tree) $s_i \in \mathcal{F}$ we can move to another state (tree) using "birth moves" and "death moves" as illustrated in Figure 9. Details of the entire procedure can be found in Castro, Coates and Nowak (2004). The Metropolis–Hastings algorithm is a basic sampling approach, which, despite its simplicity, results in improved performance compared to ALT; the incorporation of more sophisticated sampling strategies is an avenue for developing improved topology identification procedures.

To achieve our (approximate) solution of (8), we simulate the constructed chain and keep track of the tree we visit that has the largest likelihood; the longer

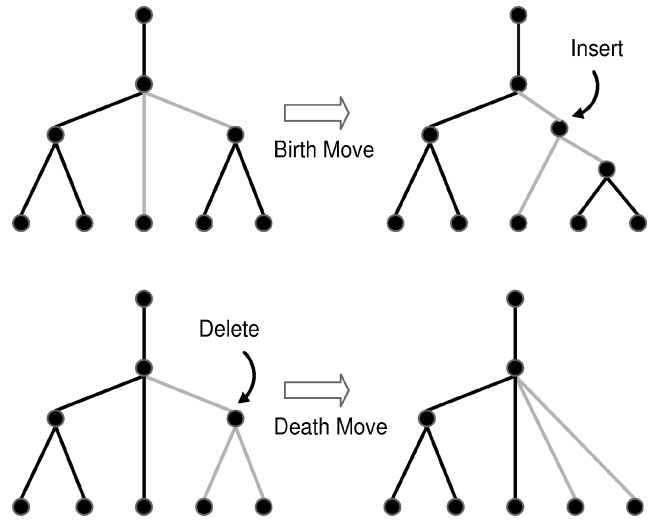

FIG. 9. *Illustration of the birth and death moves in the MCMC search algorithm. The birth move selects a node with more than two children, chooses two of these children and inserts an extra node as the new parent of these children. The death move chooses a node with two children and deletes that node.*

the chain is simulated, the higher the chance of visiting the MLT at least once. Although theoretically the starting point (initial state) of the chain is not important, provided that the chain is simulated for long enough, starting at a reasonable point improves the chance of visiting the MLT in a reasonable simulation period. Starting the chain simulation from the tree obtained using the ALT algorithm is a reasonable approach, since this is a consistent estimator and so one expects the resulting tree to be "close" (in terms of the number of MCMC moves) to the actual MLT. This is the major reason the simple Metropolis–Hastings sampling procedure works reasonably well. Although inefficiencies can prevent it from visiting more than a small region of the forest, it does visit much of the region near the MLT early in its evolution and can thus "correct" local errors in the ALT.

One drawback to the likelihood criterion is that it places no penalty on the number of links in the tree. As a consequence, trees with more links can have higher likelihood values (since the extra degrees of freedom they possess allow them to fit the data more closely). This is an instance of the classic "overfitting" problem associated with model estimation (Rissanen, 1989) and can be remedied by applying regularization, that is, by replacing the simple likelihood criterion with a *penalized* likelihood criterion,

$$\widehat{\mathcal{T}}_\lambda = \arg\max_{\mathcal{T}\in\mathcal{F}} \log \mathcal{L}(\mathbf{x}|\mathcal{T}) - \lambda n(\mathcal{T}), \tag{13}$$

where $n(\mathcal{T})$ is the number of links in the tree $\mathcal{T}$ and $\lambda \geq 0$ is a parameter, chosen by the user, to balance the trade-off between fitting to the data and controlling the number of links in the tree. We can use an MCMC method in a similar fashion as before to approximately find the solution of (13). Minimum description length principles (Rissanen, 1989) motivate a penalty that is dependent on the size of the network (in terms of the number of receivers). However, other model selection techniques lead to choices of different penalties (Robert and Casella, 1999).

4.2 Experimental Results

4.2.1 *Probing techniques and modeling.* There are several possible choices for similarity metrics in the topology identification problem; the only constraints are that the metric obey the monotonicity property and is measurable in a practical setting. It is possible to devise similarity metrics that rely on packet losses (e.g., average loss on a shared path). Although these are appealing because they are very simple to measure, losses are relatively rare in a properly functioning network (generally less than 2% for an end-to-end path), so these metrics have poor discrimination properties. Metrics that use delay/timing measurements offer better discrimination (Duffield, Horowitz and Lo Presti, 2001), but their estimation often requires clock synchronization between various physical points in the network, a rather difficult task (Pásztor and Veitch, 2002). In earlier work we proposed a topology identification method based on delay differences (Coates et al., 2002a). The measurement technique overcomes the clock synchronization issues without impairing the good discrimination of delay-based metrics and hence it is easily deployed in practice. The method relies on a measurement scheme called sandwich probing, details of which can be found in Coates et al. (2002a); here we present a brief overview.

Each sandwich probe consists of three packets and gives information about the shared path between two receivers. Figure 10 illustrates the probing scheme. The large packet is destined for node 2; the small packets are destined for node 3. The black circles on the links represent physical queues where no branching occurs. The initial spacing between the small probes d is increased along the shared path from nodes 0 to 1 because the second small probe p_2 queues behind the large packet (due to the bandwidth limitations of each link). The measurement collected for each receiver pair is the extra delay difference Δd between the two small packets. This extra delay is due to the queueing of the second small packet behind the large one, for all links in the shared portion of the path. The metric used for each pair is the mean delay difference. In idealized network conditions (Coates et al., 2002a), the contribution

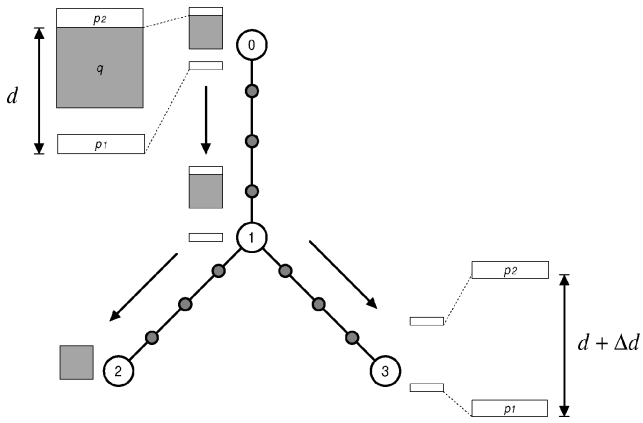

FIG. 10. *Example of sandwich probe measurement.*

from each link in the shared path to the mean delay difference is inversely proportional to its bandwidth and is always positive, so the metric satisfies the monotonicity property.

The observed mean delay differences are noisy versions of the underlying metrics, primarily because of the influence of background traffic in the network. Let x_{ij} be the sample mean of repeated delay difference measurements for pair $i, j \in R$. We assume that the cross-traffic is stationary over the measurement interval and the initial spacing of the two small packets d is large enough so that neither the large packet nor the second small packet queues behind the first small packet at any time. We send each probe far apart in time, so we can assume that the outcomes of different measurements are independent. Under these and other mild assumptions, the measurements are statistically independent and have finite variance; hence, according to the central limit theorem, the distribution of each empirical mean tends to a Gaussian. This motivates the (approximate) model

$$x_{ij} \sim \mathcal{N}(\gamma_{ij}, \sigma_{ij}^2), \tag{14}$$

where σ_{ij}^2 is the sample variance of the measurements, divided by the number of measurements x_{ij} is the sample mean of the measurements and $\mathcal{N}(\gamma, \sigma^2)$ denotes the Gaussian density with mean γ and variance σ^2. Notice that we are not assuming that the delay differences are normally distributed, but only their empirical means. Under the above assumptions, as the number of measurements increases, the model accuracy increases. We also assume that the measurements for the different receiver pairs are statistically independent, which is a reasonable assumption due to generally weak spatial correlation between traffic on different links.

4.2.2 *Internet experiments.* We have implemented a software tool called `nettomo` that performs sandwich probing measurements and estimates the topology of a tree-structured network. We conducted Internet experiments using several hosts in the United States and abroad. The topology inferred from `traceroute` is depicted in Figure 11a. Often the `traceroute` tool cannot be used to determine the topology, but it does work in this measurement scenario and thus provides a useful ground truth for validation (even here, `traceroute` fails to detect one network element). The source for the experiments was located at Rice University. There are 10 receiver clients, 2 located on different networks at Rice, 2 at separate hosts in Portugal and 6 located at four other U.S. universities.

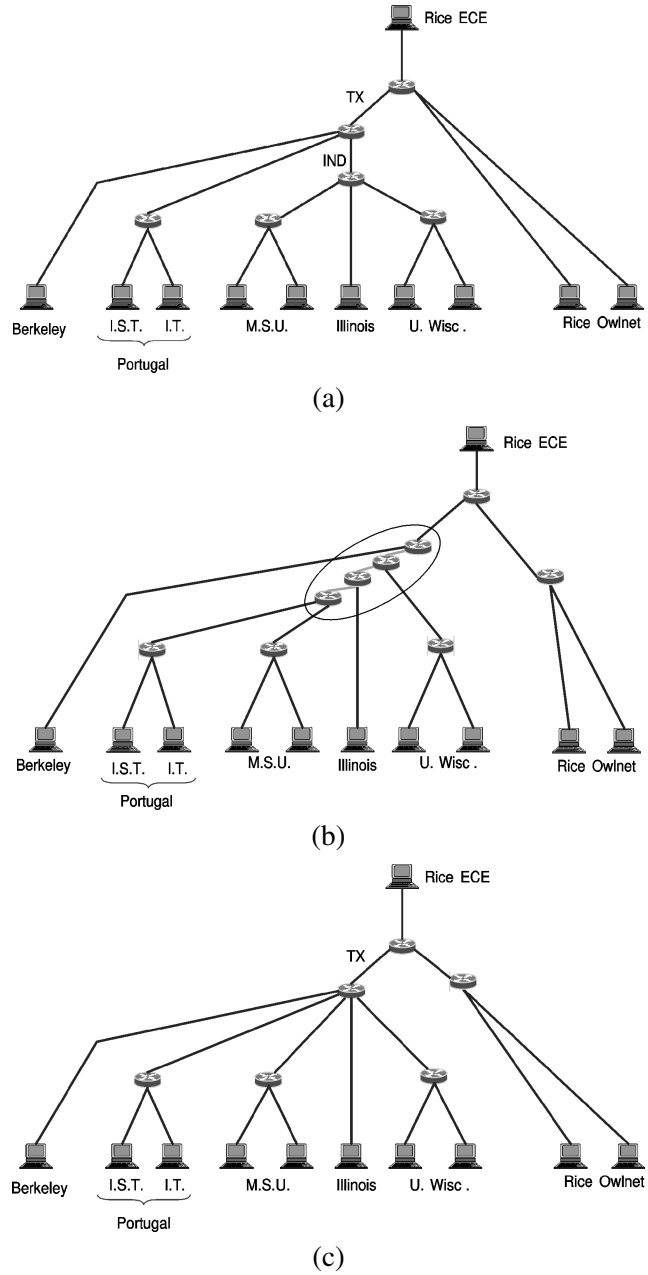

FIG. 11. (a) *The topology of the network used for Internet experiments obtained using* `traceroute`. (b) *Estimated topology using the ALT algorithm. The three links inside the ellipse have link-parameter values* $\gamma_k - \gamma_{f(k)}$ *that are one order of magnitude smaller than all the other links.* (c) *The estimated topology obtained using the MCMC method with a penalized likelihood criterion.*

The experiment was conducted for a period of 8 min, during which a sandwich probe was sent to a randomly chosen receiver pair once every 50 ms. Without any loss, the maximum number of probes available is 8600. This corresponds to less than 200 probes per pair; hence the traffic overhead on any link is very low.

We applied the ALT algorithm to the measurements collected and the result is depicted in Figure 11b. Since the procedure is suited only for binary trees, it adds some extra links with small link-level metric value (i.e., $\gamma_k - \gamma_{f(k)} \approx 0$). The extra links are an artifact of our model and are essentially overfitting the data. Using the maximum penalized likelihood approach, we obtain the result depicted in Figure 11c (see Coates et al., 2002a, for details of the penalty selection procedure). Notice that this is very close to the `traceroute` topology, but it fails to detect the backbone connection between Texas and Indianapolis. We expect that the latter connection is very high speed and that the queuing effects on the constituent links are too minor to influence measurements sufficiently for its detection. The estimated topologies also place an extra shared element between the Rice computers. This element is not a router and hence is not shown in the topology returned by `traceroute`, but it corresponds to a real physical device and branching point. To the best of our knowledge, the detected element is a bandwidth limitation device.

5. CONCLUSION AND FUTURE DIRECTIONS

This article has provided an overview of the area of large-scale inference and tomography in communication networks. As is evident from the limited scale of the simulations and experiments discussed in this article, the field is emerging. Deploying measurement/probing schemes and evaluating inference algorithms for larger networks is the next key step. Statistics will continue to play an important role in this area and in this section we attempt to stimulate the reader with an outline of some of the many open issues. These issues can be divided into extensions of the theory and potential networking application areas.

The spatiotemporally stationary and independent traffic and network transport models that currently dominate network tomography research have limitations, especially in tomographic applications that involve heavily loaded networks. Since one of the principal applications of network tomography is to detect heavily loaded links and subnets, relaxation of these assumptions continues to be of great interest. Some recent work on relaxing spatial dependence and temporal independence has appeared in unicast (Shih and Hero, 2001) and multicast (Cáceres et al., 1999) settings. However, we are far from the point of being able to implement flexible yet tractable models which simultaneously account for long time traffic dependence, latency, dynamic random routing and spatial dependence. As wireless links and ad hoc networks become more prevalent, accounting for spatial dependence and routing dynamics will become increasingly important.

Recently there have been some preliminary attempts to deal with the time-varying, nonstationary nature of network behavior. In addition to the estimation of time-varying OD traffic matrices discussed in Section 3.2, other researchers have adopted a dynamical systems approach to handle nonstationary link-level tomography problems (Coates and Nowak, 2002). Sequential Monte Carlo inference techniques were employed by Coates and Nowak (2002) to track time-varying link delay distributions in nonstationary networks. One common source of temporal variability in link-level performance is the nonstationary characteristics of cross-traffic.

There is also an accelerating trend toward network security that will create a highly uncooperative environment for active probing—firewalls designed to protect information may not honor requests for routing information, special packet handling (multicast, TTL, etc.) and other network transport protocols required by many current probing techniques. This has prompted investigations into more passive traffic monitoring techniques, for example, based on sampling TCP traffic streams (Padmanabhan, Qiu and Wang, 2002; Tsang, Coates and Nowak, 2001). Furthermore, the ultimate goal of carrying out network tomography on a massive scale poses a significant computational challenge. Decentralized processing and data fusion will probably play an important role in reducing both the computational burden and the high communication overhead of centralized data collection from edge nodes.

The majority of work reported to date has focused on reconstruction of network parameters which may be only indirectly related to the decision-making objectives of the end-user regarding the existence of anomalous network conditions. An example of this is bottleneck detection considered by Shih and Hero (2001) and Ziotopolous, Hero and Wasserman (2001) as an application of reconstructed delay or loss estimation. Other important decision-oriented applications may be detection of coordinated attacks on network resources, network fault detection and verification of services.

Finally, the impact of network monitoring, which is the subject of this article, on network control and provisioning could become the application area of most practical importance. Admission control, flow control,

service level verification, service discovery and efficient routing could all benefit from up-to-date and reliable information about link and router level performances. The big question is, Can statistical methods be developed which ensure accurate, robust and tractable monitoring for the development and administration of the Internet and future networks?

ACKNOWLEDGMENTS

This work was supported by National Science Foundation Grants MIP–97-01692, ANI-00-99148, FD-01-12731 and ANI-97-34025, Office of Naval Research Contract N00014-00-1-0390, Army Research Office Grants DAAD19-99-1-0290, DAAD19-01-1-0643 and DAAH04-96-1-0337, Science and Engineering Research Canada, and Department of Energy Grant DE-FC02-01ER25462. We acknowledge the invaluable contributions of J. Cao, D. Davis, M. Gadhiok, R. King, E. Rombokas, Y. Tsang and S. Vander Wiel to the work described in this article.

REFERENCES

BERGER, J. O., LISEO, B. and WOLPERT, R. L. (1999). Integrated likelihood methods for eliminating nuisance parameters (with discussion). *Statist. Sci.* **14** 1–28.

BESAG, J. (1974). Spatial interaction and the statistical analysis of lattice systems (with discussion). *J. Roy. Statist. Soc. Ser. B* **36** 192–236.

BESAG, J. (1975). Statistical analysis of non-lattice data. *The Statistician* **24** 179–195.

BESTAVROS, A., BYERS, J. and HARFOUSH, K. (2002). Inference and labeling of metric-induced network topologies. In *Proc. IEEE INFOCOM 2002* **2** 628–637. IEEE Press, New York.

BLACKWELL, D. (1973). Approximate normality of large products. Technical report, Dept. Statistics, Univ. California, Berkeley.

CÁCERES, R., DUFFIELD, N., HOROWITZ, J. and TOWSLEY, D. (1999). Multicast-based inference of network-internal loss characteristics. *IEEE Trans. Inform. Theory* **45** 2462–2480.

CAO, J., DAVIS, D.,VANDER WIEL, S. and YU, B. (2000a). Time-varying network tomography: Router link data. *J. Amer. Statist. Assoc.* **95** 1063–1075.

CAO, J., VANDER WIEL, S., YU, B. and ZHU, Z. (2000b). A scalable method for estimating network traffic matrices. Technical report, Bell Labs.

CASTRO, R., COATES, M. and NOWAK, R. (2004). Likelihood based hierarchical clustering. *IEEE Trans. Signal Process.* **52** 2308–2321.

CHAO, X., MIYAZAWA, M. and PINEDO, M. (1999). *Queueing Networks*: *Customers*, *Signals and Product Form Solutions*. Wiley, New York.

COATES, M., CASTRO, R., NOWAK, R., GADHIOK, M., KING, R. and TSANG, Y. (2002a). Maximum likelihood network topology identification from edge-based unicast measurements. In *Proc. ACM SIGMETRICS 2002* 11–20. ACM Press, New York.

COATES, M., HERO, A., NOWAK, R. and YU, B. (2002b). Internet tomography. *IEEE Signal Processing Magazine* **19**(3) 47–65.

COATES, M. and NOWAK, R. (2000). Network loss inference using unicast end-to-end measurement. In *Proc. ITC Seminar on IP Traffic*, *Measurement and Modelling* 28-1–28-9. Available at citeseer.ist.psu.edu/context/1699850/514748.

COATES, M. and NOWAK, R. (2002). Sequential Monte Carlo inference of internal delays in nonstationary communication networks. *IEEE Trans. Signal Process.* **50** 366–376.

COX, D. R. (1975). Partial likelihood. *Biometrika* **62** 269–276.

CSISZÁR, I. (1975). I-divergence geometry of probability distributions and minimization problems. *Ann. Probab.* **3** 146–158.

DUFFIELD, N., HOROWITZ, J. and LO PRESTI, F. (2001). Adaptive multicast topology inference. In *Proc. IEEE INFOCOM 2001* **3** 1636–1645. IEEE Press, New York.

DUFFIELD, N., HOROWITZ, J., LO PRESTI, F. and TOWSLEY, D. (2002). Multicast topology inference from measured end-to-end loss. *IEEE Trans. Inform. Theory* **48** 26–45.

DUFFIELD, N., LO PRESTI, F., PAXSON, V. and TOWSLEY, D. (2001). Inferring link loss using striped unicast probes. In *Proc. IEEE INFOCOM 2001* **2** 915–923. IEEE Press, New York.

FASULO, D. (1999). An analysis of recent work on clustering algorithms. Technical Report 01-03-02, Dept. Computer Science and Engineering, Univ. Washington. Available at citeseer.nj.nec.com/fasulo99analysi.html.

HARFOUSH, K., BESTAVROS, A. and BYERS, J. (2000). Robust identification of shared losses using end-to-end unicast probes. In *Proc. IEEE International Conference on Network Protocols* 22–33. IEEE Press, New York. Errata available as Technical Report 2001-001, Dept. Computer Science, Boston Univ.

HASTINGS, W. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika* **57** 97–109.

IHAKA, R. and GENTLEMAN, R. (1996). R: A language for data analysis and graphics. *J. Comput. Graph. Statist.* **5** 299–314.

KELLY, F. P., ZACHARY, S. and ZIEDINS, I., eds. (1996). *Stochastic Networks*: *Theory and Applications*. Oxford Univ. Press.

LELAND, W., TAQQU, M., WILLINGER, W. and WILSON, D. (1994). On the self-similar nature of Ethernet traffic. *IEEE/ACM Transactions on Networking* **2** 1–15.

LIANG, G. and YU, B. (2003a). Maximum pseudo likelihood estimation in network tomography. *IEEE Trans. Signal Process.* **51** 2043–2053.

LIANG, G. and YU, B. (2003b). Pseudo likelihood estimation in network tomography. In *Proc. IEEE INFOCOM 2003* **3** 2101–2111. IEEE Press, New York.

LO PRESTI, F., DUFFIELD, N., HOROWITZ, J. and TOWSLEY, D. (2002). Multicast-based inference of network-internal delay distributions. *IEEE/ACM Transactions on Networking* **10** 761–775.

MORRIS, R. and LIN, D. (2000). Variance of aggregated web traffic. In *Proc. IEEE INFOCOM 2000* **1** 360–366. IEEE Press, New York.

O'SULLIVAN, F. (1986). A statistical perspective on ill-posed inverse problems (with discussion). *Statist. Sci.* **1** 502–527.

PADMANABHAN, V. N., QIU, L. and WANG, H. (2002). Passive network tomography using Bayesian inference. In *Proc. ACM SIGCOMM Workshop on Internet Measurement* 93–94. ACM Press, New York.

PÁSZTOR, A. and VEITCH, D. (2002). PC based precision timing without GPS. In *Proc. ACM SIGMETRICS 2002* 1–10. ACM Press, New York.

RATNASAMY, S. and MCCANNE, S. (1999). Inference of multicast routing trees and bottleneck bandwidths using end-to-end measurements. In *Proc. IEEE INFOCOM 1999* **1** 353–360. IEEE Press, New York.

RISSANEN, J. (1989). *Stochastic Complexity in Statistical Inquiry.* World Scientific, Singapore.

ROBERT, C. and CASELLA, G. (1999). *Monte Carlo Statistical Methods.* Springer, New York.

ROLLS, D. (2003). Limit theorems and estimation for structural and aggregate teletramc models. Ph.D. dissertation, Queen's Univ., Kingston, Ontario, Canada.

SCOTT, D. (1992). *Multivariate Density Estimation*: *Theory*, *Practice and Visualization.* Wiley, New York.

SHIH, M. and HERO, A. (2001). Unicast inference of network link delay distributions from edge measurements. In *Proc. IEEE International Conference on Acoustics*, *Speech*, *and Signal Processing* **6** 3421–3424. IEEE Press, New York.

TEBALDI, C. and WEST, M. (1998). Bayesian inference on network traffic using link count data (with discussion). *J. Amer. Statist. Assoc.* **93** 557–576.

TSANG, Y., COATES, M. and NOWAK, R. (2001). Passive network tomography using EM algorithms. In *Proc. IEEE International Conference on Acoustics*, *Speech*, *and Signal Processing* **3** 1469–1472. IEEE Press, New York.

TSANG, Y., COATES, M. J. and NOWAK, R. (2003). Network delay tomography. *IEEE Trans. Signal Process.* **51** 2125–2136.

VANDERBEI, R. J. and IANNONE, J. (1994). An EM approach to OD matrix estimation. Technical Report SOR 94-04, Princeton Univ.

VARDI, Y. (1996). Network tomography: Estimating source-destination traffic intensities from link data. *J. Amer. Statist. Assoc.* **91** 365–377.

WARD, J. H. (1963). Hierarchical grouping to optimize an objective function. *J. Amer. Statist. Assoc.* **58** 236–245.

WHITE, H. (1994). *Estimation*, *Inference and Specification Analysis.* Cambridge Univ. Press, New York.

WILLET, P. (1988). Recent trends in hierarchical document clustering: A critical review. *Information Processing and Management* **24** 577–597.

ZIOTOPOLOUS, A., HERO, A. and WASSERMAN, K. (2001). Estimation of network link loss rates via chaining in multicast trees. In *Proc. IEEE International Conference on Acoustics*, *Speech*, *and Signal Processing* **4** 2517–2520. IEEE Press, New York.