Internet Tomography

Mark Coates Alfred Hero Robert Nowak Bin Yu

January, 2002

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 increasingly challenging tasks. 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 problems such as tomographic image reconstruction, system identification, and array processing all have interesting interpretations in the networking context. This article introduces the new field of network tomography, a field which we believe will benefit greatly from the wealth of signal processing theory and algorithms.

1 Introduction

The Internet has evolved from a small tightly controlled network serving only a few users in the late 1970's to the immense multi-layered collection of heterogeneous terminals, routers and other platforms that we encounter today when web-surfing. Unlike, for example, the telephone network which evolved in a slower and more controlled manner, the Internet has evolved very rapidly in a largely unregulated and open environment. The lack of centralized control and the heterogeneous nature of the Internet leads to a very important problem: mapping network connectivity, bandwidth, and performance as functions of space and time. A wide variety of Internet maps have been produced using existing networking tools such as `ping` and `traceroute`. Information on these tools, along with a collection of interesting Internet mapping projects, can be found on the CAIDA (Cooperative Association for Internet Data Analysis) [1]. A survey of many Internet mapping projects and results is found in the popular-science book *Atlas of Cyberspace* [2]. The mapping techniques described in the references above, however, usually provide only a partial picture of the Internet because they do not produce quantitative performance information. The decentralized nature of the Internet makes quantitative assessment of network performance very difficult. One cannot depend on individual servers and routers to freely transmit vital network statistics such as traffic rates, link delays, and dropped packet rates. The collection of network statistics at servers and internal routers can impose an impracticable overhead expense in terms of added computing, communication, and hardware requirements. Even if such statistics can be collected, an Internet Service Provider (ISP) may regard such information as highly confidential. Moreover, the transmission of statistics to a central processing point may consume considerable bandwidth, adding to network load and congestion.

In certain cases, however, useful network statistics can be indirectly acquired without special-purpose cooperation from servers and routers and with little or no impact on network load. These statistical quantities can reveal hidden network structure and help to detect and isolate congestion, routing faults, and anomalous traffic. The acquisition of the statistics relies on the application of sophisticated methods of active network probing or passive traffic monitoring. These methods do not directly provide the desired information. The problem of extracting the hidden information from active or passive traffic measurements falls in the realm of statistical inverse problems, an area which has long been of interest to signal and image processing researchers. Signal processing know-how, acquired in areas such as image reconstruction, pattern recognition, system identification, and sensor array signal processing, can provide tremendous insight into networking inverse problems.

This article deals with network monitoring and inference for wired networks such as the Internet. The word "inference" is intended to more sharply delineate the field of study addressed in the article, precluding approaches that directly measure network statistics or rely on complete cooperation from the network. The task of inferential network monitoring gives rise to problems that involve the estimation of a potentially very large number of spatially distributed parameters, e.g., single link loss rates, delay distributions, connectivity, and traffic flow. To tackle such large estimation problems, researchers adopt the simplest possible models for network traffic and ignore many intricacies of packet transport such as feedback and latency. These simpler models, although not suitable for fine-grain analysis of individual queuing mechanisms and network traffic behavior, are generally adequate for the inference of gross-level performance characteristics. Focus is shifted from detailed mathematical modeling of network dynamics [3, 4] to careful handling of measurement and probing strategies, large scale computations, and model validation. The measurement methodologies require: software tools for monitoring traffic flow and generating probe traffic; statistical modeling of the measurement process; sampling strategies for online data collection. The underlying computational science involves: complexity reducing hierarchical statistical models; moment and likelihood based estimation; expectation-maximization algorithms; Markov Chain Monte Carlo algorithms; and other iterative optimization methods. Model validation includes: study of parameter identifiability conditions; feasibility analysis via Cramér-Rao bounds and other bounding techniques; implementation of network simulation software such as the `ns-2` network simulation environment [5]; and application to real network data.

Many in the network community have long been interested in measuring internal network parameters and in mathematical and statistical characterization of network behavior. Researchers in the fields of computer science, network measurement and network protocols have developed software for measuring link delays, detecting intruders and rogue nodes, and isolating routing table inconsistencies and other faults. Researchers from the fields of networking, signal processing, automatic control, statistics, and applied mathematics have been interested in modeling the statistical behavior of network traffic and using these models to infer data transport parameters of the network. Previous work can be divided into three areas: i) development of software tools to monitor/probe the network; ii) probabilistic modeling of networks of queues; and iii) inference from measurements of single stream or multiple streams of traffic.

Computer scientists and network engineers have developed many tools for active and passive measurement of the network. These tools usually require extra cooperation (in addition to the basic cooperation required for routine packet transmission) amongst the nodes of the network. For example, in sessions running under RTCP (Real Time Control Protocol), summary sender/receiver reports on packet jitter and packet losses are distributed to all session participants [6]. Active probing tools such as `ping`, `pathchar` (`pchar`), `clink`, and `traceroute` measure and report packet transport attributes of the round-trip path (from sender to receiver and back) of a probe (see [1] for a survey of these and other measurement tools). Trajectory sampling [7] is another example of an active probing software tool. These methods depend on accurate reporting by all nodes along the route and many require special assumptions, e.g., symmetric forward/reverse links, existence of store-and-forward routers, non-existence of fire-walls. As the Internet evolves towards decentralized, uncooperative, heterogeneous administration and edge-based control these tools will be limited in their capability. In the future, large-scale inference and tomography methods such as those discussed in this article will become of increasing importance due to their ability to deal with uncooperative networks.

Network queueing theory offers a rich mathematical framework which can be useful for analyzing small scale networks with a few interconnected servers. See the recent books [3, 4] for overviews of this area. The limitations of queuing network models for analyzing real, large-scale networks can be compared to the limited utility of classical Newtonian mechanics in complex large scale interacting particle systems: the macroscopic behavior of an aggregate of many atoms appears qualitatively different from what is observed at a microscopic scale with a few isolated atomic nuclei. Furthermore, detailed information on queuing dynamics in the network is probably unnecessary when, by making a few simple approximations, one can obtain reasonably accurate estimates of average link delays, dropped packet probabilities, and average traffic rates directly from external measurements. The much more computationally demanding queuing network analysis becomes necessary when addressing a different set of problems that can be solved off-line. Such problems include calculating accurate estimates of fine grain network behavior, e.g., the dynamics of node traffic rates, service times, and queue lengths.

The area of statistical modeling of network traffic is a mature and active field [8, 9, 10, 11, 12]. Sophisticated fractal and multifractal models of single traffic streams can account for long range dependency, non-Gaussian distributions, and other peculiar behaviors. Such self similar behavior of traffic rates has been validated for heavily loaded wired networks [13]. For a detailed overview of these and other statistical traffic models we refer the reader to the companion article(s) in this special issue. To date these models are overly complicated to be incorporated into the large scale network inference problems discussed in this article. Simplifying assumptions such as spatial and temporal independence are often made in order to devise practical and scalable inference algorithms. By making these assumptions, a fundamental linear observation model can be used to simplify the inference process. While some progress has been made on incorporating simple first order spatio-temporal dependency models into large scale network inference problems [14] much work remains to be done.

This article attempts to be fairly self-contained; only a modest familiarity with networking principles is required and basic concepts are defined as necessary. For more background information, the a number of recent textbooks [15, 16, 17, 18, 19, 20, 21] provide an excellent introductions to the field of networking. The article is organized as follows. First we briefly review the area of large scale network inference and tomography. We then discuss link-level inference from path measurements and focus on two examples; loss rate and delay distribution estimation. We consider the problem of determining the connectivity structure or topology of a network and then turn to origin-destination traffic matrix inference from link measurements in the context of both stationary and non-stationary traffic.

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. Y. Vardi was one of the first to rigorously study this sort of problem and coined the term *network tomography* [22] due to the similarity between network inference and medical tomography. Two forms of network tomography have been addressed in the recent literature: i) link-level parameter estimation based on end-to-end, path-level traffic measurements [23, 24, 25, 26, 27, 28, 29, 30, 31, 32] and ii) sender-receiver path-level traffic intensity estimation based on link-level traffic measurements [33, 22, 34, 35, 36].

In link-level parameter estimation, the traffic measurements typically consist of counts of packets transmitted and/or received between nodes or time delays between packet transmissions and receptions. The time delays are due to both propagation delays and router processing delays along the path. The measured path delay is the sum of the delays on the links comprising the path; the link delay comprises both the propagation delay on that link and the queuing delay at the routers lying 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

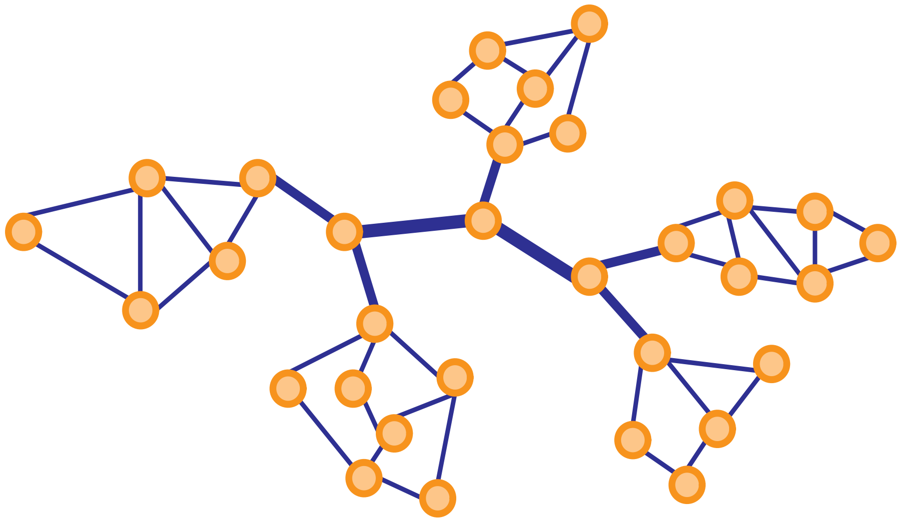


Figure 1: An arbitrary network topology. Each node represents a computer or a cluster of computers or a router. Each edge in the graph represents a direct link between two nodes. The topology here depicts "clusters" corresponding to local area networks or other subnetworks connected together via the network "backbone". The width of each edge reflects the bandwidth of the corresponding connection (thicker edge implies higher bandwidth).

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 down-time 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

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{\theta} + \boldsymbol{\epsilon}, \tag{1}$$

where: $\boldsymbol{y}$ is a vector of measurements, e.g., packet counts or end-to-end delays, taken at a number of different measurement sites; $\boldsymbol{A}$ is a *routing matrix*; $\boldsymbol{\theta}$ is a vector of packet parameters, e.g. mean delays, logarithms of packet transmission probabilities over a link, or the random origin-destination traffic vector; $\boldsymbol{\epsilon}$ is a noise term which can result from random perturbations of $\boldsymbol{\theta}$ about its mean value and possibly also additive noise in the measured data $\boldsymbol{y}$ (in the origin-destination traffic matrix estimation problem $\boldsymbol{\epsilon}$ is generally assumed to be zero. Typically, but not always, $\boldsymbol{A}$ is a binary matrix (the i, j-th element is equal to '1' or '0') that captures the topology of the network. The problem of large scale network inference refers to the problem of estimating the network parameters $\boldsymbol{\theta}$ given $\boldsymbol{y}$ and either a set of assumptions on the statistical distribution of the noise $\boldsymbol{\epsilon}$ or the introduction of some form of regularization to induce identifiability. Specific examples are discussed below.

What sets the large scale network inference problem (1) apart from other network inference problems is the potentially very large dimension of $\boldsymbol{A}$ 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 $\boldsymbol{\theta}$ are specific examples of *inverse problems*. Inverse problems have a very extensive literature both in signal processing [37], statistics [38], and in applied mathematics [39]. Solution methods for such inverse problems depend on the nature of the noise $\boldsymbol{\epsilon}$ and the $\boldsymbol{A}$ matrix and typically require iterative algorithms since they cannot be solved directly. In general, $\boldsymbol{A}$ is not full-rank, so that identifiability concerns arise. Either one must be content to resolve 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 $\boldsymbol{\epsilon}$ are assumed to be approximately independent Gaussian, Poisson, binomial or multinomial distributed. When the noise is Gaussian distributed with covariance independent of $\boldsymbol{A\theta}$, 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 non-linear least squares, maximum likelihood via expectation-maximization (EM), and maximum a posteriori (MAP) via Monte Carlo Markov Chain (MCMC) algorithms can be used. These approaches will be illustrated in Sections 3 and 4.

2.2 Examples of Network Tomography

Let us consider three concrete examples of the linear model (1). First, consider the problem of estimating the packet success probability on each link given end-to-end, origin-to-destination (OD) counts of packet losses[1]. Let $\boldsymbol{\theta}$ denote the collection of log success probabilities for each link. The OD log success probability is simply $\boldsymbol{A}\,\boldsymbol{\theta}$, where $\boldsymbol{A}$ is the binary routing matrix described above. Assuming a known number of packets sent from each source to destination, a binomial measurement model can be adopted [25]. When the number of packets sent and received are large, then the binomial model can be approximated with a Gaussian likelihood, leading to the classical linear model above (1). Second, suppose that end-to-end, OD delays are measured and the goal is estimation of the delay probability distributions along each link. In

[1]The loss probabilities or "loss rates" are simply one minus the probability of successful transmission.

this case, let $\boldsymbol{\theta}$ be a vector composed of the cumulant generating functions of the delay densities on each link. Then, with appropriate approximation arguments [31], $\boldsymbol{y}$ is again related to $\boldsymbol{\theta}$ according to the linear model (1). Third, in the OD traffic matrix estimation case, $\boldsymbol{y}$ are link-level packet count measurements and $\boldsymbol{\theta}$ are the OD traffic intensities. Gaussian assumptions are made on the origin-destination traffic with a mean-variance relationship in high count situations in [17] leading to the linear equation (1) without the error term $\boldsymbol{\epsilon}$. In each of these cases, the noise $\boldsymbol{\epsilon}$ may be correlated and have a covariance structure depending on $\boldsymbol{A}$ and/or $\boldsymbol{\theta}$, leading to less than trivial inference problems. Moreover, in many cases the limited amount of data makes Gaussian approximations inappropriate and discrete observation models (e.g., binomial) may be more accurate descriptions of the discrete, packetized nature of the data. These discrete observation models necessitate more advanced inference tools such as the Expectation-Maximization (EM) algorithm and Monte Carlo simulation schemes (more on this in Section 3).

Let us consider two further embellishments of the basic network inference problem described by the linear model (1). First, all quantities may, in general, be time-varying. For example, we may write

$$\boldsymbol{y}_t \;=\; \boldsymbol{A}_t\,\boldsymbol{\theta}_t \;+\; \boldsymbol{\epsilon}_t, \tag{2}$$

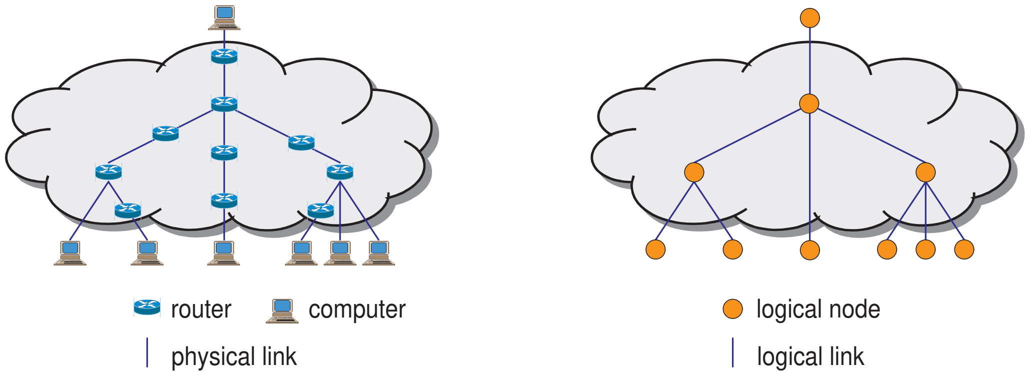
where t denotes time. The estimation problems now involve tracking time varying parameters. In fact, the time-varying scenario probably more accurately reflects the dynamical nature of the true networks. There have been several efforts aimed at tracking nonstationary network behavior which involve analogs of classical Kalman-filtering methods [34, 26]. Another variation on the basic problem (1) is obtained by assuming that the routing matrix $\mathbf{A}$ is not known precisely. This leads to the so-called "topology identification" problem [30, 40, 41, 42, 43, 44, 45], and is somewhat akin to blind deconvolution or system identification problems.

3 Link-Level Network Inference

Link-level network tomography is the estimation of link-level network parameters (loss rates, delay distributions) from path-level measurements. Link-level parameters can be estimated from direct measurements when all nodes in a network are cooperative. Many promising tools such as `pathchar (pchar), traceroute, clink, pipechar` use Internet Control Message Protocol (ICMP) packets (control packets that request information from routers) in order to estimate link-level loss, latencies and bandwidths. However, many routers do not respond to or generate ICMP packets or treat them with very low priority, motivating the development of large-scale link-level network inference methods that do not rely on cooperation (or minimize cooperation requirements).

In this article we discuss methods which require cooperation between a subset of the nodes in the network, most commonly the edge nodes (hosts or ingress/egress routers). Research to date has focused on the parameters of delay distributions, loss rates and bandwidths, but the general problem extends to the reconstruction of other parameters such as available bandwidths and service disciplines. The Multicast-based Inference of Network-internal Characteristics (MINC) Project at the University of Massachusetts [23] pioneered the use of multicast probing for network tomography, and stimulated much of the current work in this area [23, 24, 25, 46, 26, 27, 29, 30, 31, 47, 32].

We now outline a general framework for the link-level tomography problems. Consider network depicted in Figure 2(a). This illustrates the scenario where packets are sent from a set of sources to a number of destinations. The end-to-end (path-level) behavior can be measured via a coordinated measurement scheme between the sender and the receivers. The sender can record whether a packet successfully reached its destination or was dropped/lost and determine the transmission delay by way of some form of acknowledgment from the receiver to the sender upon successful packet reception. It is assumed that the sender cannot directly determine the specific link on which the packet was dropped nor measure delays or bandwidths on individual links within paths. A network can be logically represented by a graph consisting of nodes connected by links. Potentially, a logical link connecting two nodes represents many routers and the physical links between them, as depicted in Figure 2.



(a) Physical network

(b) Logical tree topology

Figure 2: Physical and logical networks. The "cloud" indicates portions of the network that are inaccessible by direct measurement. (a) Physical structure for single sender multiple receiver network. (b) Logical topology.

Each node is numbered $j = 0, \ldots, m$, and each link is assigned the number of the connected node below it. Let there be n distinct measurement paths (from a sender to a receiver) through the network, enumerated $i = 1, \ldots, n$. Define a_{ij} to be the probability that the i-th measurement path contains the j-th link. In most cases a_{ij} will take values of 0 or 1, but it is useful to maintain a level of generality which can handle random routing. A is the routing matrix having ij-th element a_{ij}. The rows of A correspond to paths from the sender to the receivers and the columns correspond to individual links in those paths. Figure 3 illustrates a simple network consisting of a single sender (node 0), two receivers (the leaves of the tree, nodes 2 and 3) and an internal node representing a router at which the two communication paths diverge (node 1). Only end-to-end measurements are possible, i.e., the paths are (0,2), and (0,3), where (s, t) denotes the path between nodes s and t. There are 3 links and 2 paths/receivers, and therefore the matrix A is 2×3 dimensional and has the form:

$$A = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \quad (3)$$

Note that in this example, A is not full rank. We discuss the ramifications in later sections.

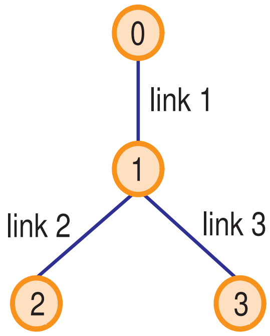


Figure 3: Tree-structured topology.

A number of key assumptions underpin current link-level network tomography techniques, determining measurement frameworks and mathematical models. The routing matrix is usually assumed to be known and constant throughout the measurement period. Although the routing tables in the Internet are periodically updated, these changes occur at intervals of several minutes. However, the dynamics of the routing matrix may restrict the amount of data that can be collected and used for inference. Most current methodologies usually assume that that performance characteristics on each link are statistically independent of all other links, however this assumption is clearly violated due to common cross-traffic flowing through the links. Assumptions of temporal stationarity are also made in many cases. In link-level delay tomography, it is generally assumed that synchronized clocks are available at all senders and receivers. Although many of the simplifying assumptions do not strictly hold, such "first-order" approximations have been shown to be reasonable enough for the large-scale inference problems of interest here [23, 24, 25, 26, 27, 28, 29, 30, 31, 32].

There are two common modes of communication in networks: multicast and unicast. In unicast communication, each packet is sent to one and only one receiver. In multicast communication, the sender effectively sends each packet to a group of subscribing receivers. At internal routers where branching occurs, e.g., node 1 in Figure 3, each multicast packet is replicated and sent along each branching path. We now overview the different approaches to link-level network tomography that are enabled by the two modes of communication. Subsequently, we provide two detailed examples of link-level network tomography applications.

3.1 Multicast Network Tomography

Network tomography based on multicast probing was one of the first approaches to the problem [24]. Consider loss rate tomography for the network depicted in Figure 3. If a multicast packet is sent by the sender and received by node 2 but not by node 3, then it can be immediately determined that loss occurred on link 3 (successful reception at node 2 implies that the multicast packet reached the internal node 1). By performing such measurements repeatedly, the rate of loss on the two links 2 and 3 can be estimated; these estimates and the measurements enable the computation of an estimate for the loss rate on link 1.

To illustrate further, let θ_1, θ_2, and θ_3 denote the log success probabilities of the three links in the network, where the subscript denotes the lower node attached to the link. Let $\widehat{p}_{2|3}$ denote the ratio of the number of multicast packet probes simultaneously received at both nodes 2 and 3 relative to the total number received at node 3. Thus, $\widehat{p}_{2|3}$ is the empirical probability of success on link 2 conditional upon success on link 3, which provides a simple estimate of θ_2. Define $\widehat{p}_{3|2}$ in a similar fashion and also let $\widehat{p}_i$, $i = 2, 3$, denote the ratio of the total number of packets received at node i over the total number of multicast probes sent to node i. We can then write

$$\begin{pmatrix} \log \widehat{p}_2 \\ \log \widehat{p}_3 \\ \log \widehat{p}_{2|3} \\ \log \widehat{p}_{3|2} \end{pmatrix} \approx \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{pmatrix}. \tag{4}$$

A least squares estimate of $\{\theta_i\}$ is easily computed for this overdetermined system of equations. Sophisticated and effective algorithms have been derived for large-scale network tomography in [24, 48, 25, 49].

Similar procedures can be conducted in the case of delay distribution tomography. There is a certain minimum propagation delay along each link, which is assumed known. Multicast a packet from node 0 to nodes 2 and 3, and measure the delay to each receiver. The delay on the first link from 0 to 1 is identical for both receivers, and any discrepancy in the two end-to-end delay measurements is solely due to a difference in the delay on link 1 to 2 and the delay link 1 to 3. This observation allows us to estimate the delay distributions on each individual link. For example, if the measured end-to-end delay to node 2 is equal to the known minimum propagation delay, then any extra delay to node 3 is incurred on link 1 to 3. Collecting delay measurements from repeated experiments in which the end-to-end delay to node 2 is minimal allows construction of a histogram estimate of the delay distribution on link 1 to 3. In larger and more general trees, the estimation becomes more complicated. Advanced algorithms have been developed for multicast-based delay distribution tomography on arbitrary tree-structured networks [29, 48].

3.2 Unicast Network Tomography

Alternatively, one can tackle loss rate and delay distribution tomography using unicast measurements. Unicast measurements are more difficult to work with than multicast, but since many networks do not support multicast, unicast-based tomography is of considerable practical interest. The difficulty of unicast-based tomography is that although single unicast packet measurements allow one to estimate end-to-end path loss rates and delay distributions, there is not a unique mapping of these path-level parameters to the corresponding individual link-by-link parameters. For example, referring again to Figure 3, if packets are sent from node 0 to nodes

2 and 3 and n_k and m_k denote the numbers of packets sent to and received by receiver node k, $k = 2, 3$, then

$$\begin{pmatrix} \log \widehat{p}_2 \\ \log \widehat{p}_3 \end{pmatrix} \approx \underbrace{\begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}}_{A} \begin{pmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{pmatrix} \tag{5}$$

where $\widehat{p}_k = m_k/n_k$ and θ_j, $j = 1, 2, 3$ denotes the log success probability associated with each link. Clearly, a unique solution for $\{\theta_j\}$ does not exist since A is not full rank.

To address this challenge in unicast loss tomography, the authors of [25] and [28] independently proposed methodologies based on measurements made using unicast, back-to-back packet pairs. These measurements provide an opportunity to collect more informative statistics that can help to resolve the link-level loss rates and delay distributions. A packet pair consists of two packets sent one after the other by the sender, possibly destined for different receivers, but sharing a common set of links in their paths. In networks whose queues obey a standard droptail policy,[2] if two back-to-back packets are sent across a common link and one of the pair is successfully transmitted across the link, then it is highly probable that the other packet is also successful. Similarly, the two packets in each pair will experience roughly the same delay through shared links. These observations have been verified experimentally in real networks [51, 27]. If one assumes that the probability of success for one packet conditioned on the success of the other is approximately unity, then the same methodology developed for multicast-based tomography (as described above) can be employed with unicast, packet-pair measurements [27].

In the case of bandwidth tomography, the authors of [52] addressed the challenge of non-uniqueness through clever use of the header fields of unicast packets. The time-to-live (TTL) field in each packet header indicates how many hops the packet should traverse. At each router the packet encounters the TTL counter is decremented by one, and when the counter reaches zero the next router discards the packet. The `nettimer` program described in [52] uses "tailgating" to collect measurements: many packet-pairs are sent from the source, each consisting of a large packet followed by a small packet. The TTL field of the large packet is varied during the measurement period so that it is propagated through only a portion of the path. The end-to-end delay measured by the small packet (in a relatively uncongested network) is primarily comprised of the propagation delay experienced by the large packet, enabling inference of the bandwidth of the subpath traversed by the large packet. Referring to the simple triad network in Figure 3 for illustration, `nettimer` might send packet-pairs from node 0 along links 1 and 2. If the TTL of the large packet is set to one, the tailgating smaller packet measures the propagation delay on link 1.

Unicast measurement can be conducted either actively or passively. In the case of active measurement, probe packets are sent by the senders specifically for the purpose of estimation. In passive monitoring, the sender extracts data from existing communications (e.g., records of TCP[3] sessions) [49, 53]. Loss rate and delay distribution tomography methods have been

[2]A droptail queuing policy means that a packet is dropped by a queue only if it reaches the queue and there is insufficient space in the buffer. In active queuing strategies, such as random-early-drop (RED) [50], packets can be dropped (with a certain probability) even if they have already entered the queue.

[3]Data transmission in the Internet is primarily handled by the Transmission Control Protocol (TCP) and

developed specifically for unicast packet pairs in [25, 28, 14, 49]. Unicast packet *stripes* (triples, quadruples, etc.) have also been investigated for loss rate tomography [27].

3.3 Example: Unicast Inference of Link Loss Rates

Link loss rates can be inferred from end-to-end, path-level unicast packet measurements using the approximate linear model given in equations (1) when the numbers packet counts are large; refer to Section 3.2. However, as stated earlier the discrete process of counting the number of sent and received packets suggests the use of a discrete probability distribution in our modeling and analysis. We give a brief introduction and example of this approach here, and for more details the interested reader is referred to related papers [25, 26, 54].

The successful traversal of a single packet across a link can be reasonably modeled as a sequence of Bernoulli events. Associate with the j-th link in the network a single parameter governing the Bernoulli model. This parameter is the probability (rate) of successful transmission on the link α_j. The complete set for all m logical links in the network is denoted by $\boldsymbol{\alpha} \equiv \{\alpha_j\}_{j=1}^m$, which comprise the success rates that network loss tomography strives to identify.

Measurements are collected by sending n_k single packets along the path to receiver k and recording how many successfully reach the destination, denoted as m_k. Determination of the success of a given packet is handled by an acknowledgment sent from the receiver back to the sender. For example, such acknowledgments are a built-in feature of TCP. The likelihood of m_k given n_k is binomial (since Bernoulli losses are assumed) and is given by

$$l(m_k \,|\, n_k, p_k) \quad = \quad \binom{n_k}{m_k} p_k^{m_k} (1 - p_k)^{n_k - m_k}, \tag{6}$$

where $p_k = \prod_{j \in \mathcal{P}(0,k)} \alpha_j$ and $\mathcal{P}(0,k)$ denotes the sequence of nodes in the path from the sender 0 to receiver k.

If the routing matrix A is full rank, then unique maximum likelihood estimates of the loss rates can be formed by solving a set of linear equations. If A is not full rank, then there is no unique mapping of the path success probabilities to the success probabilities on individual links (between routers) in the path. To overcome this difficulty, measurements are made using back-to-back packet pairs or sequences of packets as discussed above [25, 28, 27].

If two back-to-back packets are sent to node j from its parent node $\rho(j)$, then define the conditional success probability as

$$\beta_j \;\equiv\; \Pr(\text{1st packet } \rho(j) \to j \mid \text{2nd packet } \rho(j) \to j\ ),$$

where $\rho(j) \to j$ is shorthand notation denoting the successful transmission of a packet from $\rho(j)$ to j. That is, given that the second packet of the pair is received, then the first packet is received with probability β_j and dropped with probability $1 - \beta_j$. It is anticipated that $\beta_j \approx 1$

Internet Protocol (IP). TCP/IP were developed by a Department of Defense to allow cooperating computers to share resources across a network. IP is responsible for moving packets of data from node to node and TCP coordinates the delivery between the sender and receiver (server and client).

for each j, since knowledge that the second packet was successfully received suggests that the queue for link j was not full when the first packet arrived. Evidence for such behavior has been provided by observations of the Internet [55, 51]. Denote the complete set of conditional success probabilities by $\boldsymbol{\beta} \equiv \{\beta_j\}_{j=1}^m$.

Suppose that each sender sends a large number of back-to-back packet pairs in which the first packet is destined for one of its receivers k and the second for another of its receivers l. The time between pairs of packets must be considerably larger than the time between two packets in each pair. Let $n_{k,l}$ denote the number of pairs for which the second packet is successfully received at node l, and let $m_{k,l}$ denote the number of pairs for which both the first and second packets are received at their destinations. With this notation, the likelihood of $m_{k,l}$ given $n_{k,l}$ is binomial and is given by

$$l(m_{k,l}\,|\,n_{k,l}, p_{k,l}) \;\; = \;\; \binom{n_{k,l}}{m_{k,l}} p_{k,l}^{m_{k,l}} \, (1-p_{k,l})^{n_{k,l}-m_{k,l}},$$

where $p_{k,l}$ is a product whose factors are $\boldsymbol{\beta}$ elements on the shared links and $\boldsymbol{\alpha}$ elements on the unshared links. The overall likelihood function is given by

$$l(m|n,p) \;\equiv\; \prod_k l(m_k|n_k,p_k) \times \prod_{k,l} l(m_{k,l}|n_{k,l},p_{k,l}) \tag{7}$$

The goal is to determine the vectors α and β that maximize (7). Maximizing the likelihood function is not a simple task because the individual likelihood functions $l(m_k\,|\,n_k, p_k)$ or $l(m_{k,l}\,|\,n_{k,l}, p_{k,l})$ involve products of the $\boldsymbol{\beta}$ and/or $\boldsymbol{\alpha}$ probabilities. Consequently, numerical optimization strategies are required. The Expectation-Maximization (EM) algorithm is an especially attractive option that offers a stable, scalable procedure whose complexity grows linearly with network dimension [25]. An closely-related EM algorithm can be employed in link-level delay density tomography [26, 56].

The link-level loss inference framework is evaluated in [49, 54] using the `ns-2` network simulation environment [5]. Measurements were collected by passively monitoring existing TCP connections. The experiments involved simulation of the 12-node network topology shown in Figure 4(a), and the estimated success probabilities determined using the network tomography algorithm above are depicted in Figure 4. This topology reflects the nature of many networks — a slower entry link from the sender, a fast internal backbone, and then slower exit links to the receivers.

In the simulations, numerous short-lived TCP connections were established between the source (node 0) and the receivers (nodes 5-11). In addition, there is cross-traffic on internal links, such that in total there are approximately thirty TCP connections and thirty User Datagram Protocol (UDP)[4] connections operating within the network at any one time. The average utilization of the network is in all cases relatively high All the TCP connections flowing from the sender to the receivers are used when collecting packet and packet-pair measurements (see [49] for details on the data collection process). Measurements were collected over a 300 second interval.

[4]UDP is a simpler protocol than TCP. UDP simply sends packets and does not receive an acknowledgment from the receiver.

The experiments were designed to ascertain whether the unicast link-level loss tomography framework is capable of discerning where significant losses are occurring within the network. They assess its ability to determine how extensive the heavy losses are and to provide accurate estimates of loss rates on the better performing links. Three traffic scenarios were explored. In Scenario 1, links 2 and 5 experience substantial losses, thereby testing the framework's ability to separate cascaded losses. In Scenario 2, links 2 and 8 experience substantial loss, (testing the ability to resolve distributed losses in different branches of the network). In Scenario 3, many more on-off UDP and on-off TCP connections were introduced throughout the topology. Figure 4 displays the simulation results for each of the different traffic scenarios.

3.4 Example: Unicast Inference of Link Delay Distributions

When the link delays along a path are statistically independent the end-to-end delay densities are related to the link delay densities through a convolution. Several methods for unraveling this convolution from the end-to-end densities are: 1) transformation of the convolution into a more tractable matrix operator via discretization of the delays [29, 26, 31]; 2) estimation of low order moments such as link delay variance [48] from end-to-end delay variances which are additive over the probe paths; 3) nonparametric density estimation methods in combination with EM tomography algorithms [56]; 4) estimation of the link delay cumulant generating function (CGF) [31, 47] from the end-to-end delay CGF's which are also additive over the probe paths. Here we discuss the CGF estimation method from which any set of delay moments can be recovered.

Let Y_i denote the total end-to-end delay of a probe sent along the i-th probe path. Then

$$Y_i = a_{i1}X_{i1} + \cdots + a_{im}X_{im}, \qquad i = 1, \ldots n \tag{8}$$

where X_{ij} is the delay of the i-th probe along the j-th link in the path and $a_{ij} \in \{0,1\}$ are elements of the routing matrix A. Here $\{X_{ij}\}_{i=1}^{n}$ are assumed to be i.i.d. realizations of a random variable X_j associated with the delay of the j-th link.

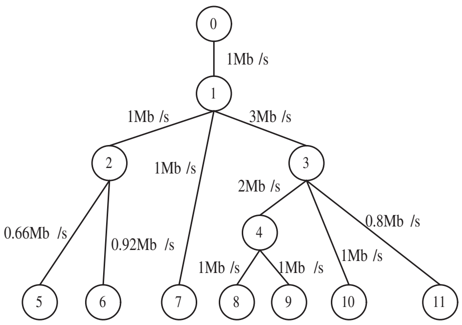
The CGF of a random variable Y is defined as $K_Y(t) = \log E[e^{tY}]$ where t a real variable. When Y is a sum of a set $\{X_j\}_{j=1}^{m}$ of statistically independent random variables the CGF satisfies the additive property $K_Y(t) = \sum_{j=1}^{m} K_{X_j}(t)$. Therefore, in view of the end-to-end delay representation (8), and assuming independent $X_{i1}, \ldots, X_{im}$ (spatial independence), the vector of CGFs of the end-to-end probe delays $\{Y_i\}_{i=1}^{m}$ of the i-th probe satisfies the linear system of equations

$$\mathbf{K}_Y(t) = A\mathbf{K}_X(t), \tag{9}$$

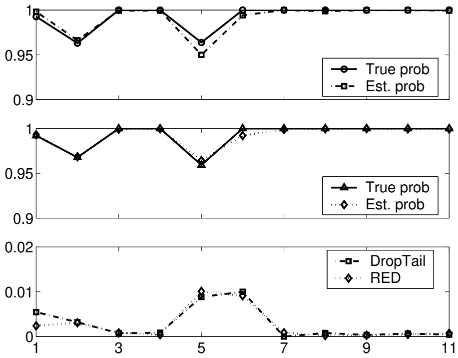
where $\mathbf{K}_Y(t) = [K_{Y_1}(t), \ldots, K_{Y_n}(t)]^T$ and $\mathbf{K}_X(t) = [K_{X_1}(t), \ldots, K_{X_m}(t)]^T$ are n-element and m-element vector functions of t, respectively.

The linear equation (9) raises two issues of interest: 1) conditions on A for identifiability of $\mathbf{K}_X(t)$ from $\mathbf{K}_Y(t)$; and 2) good methods of estimation of $\mathbf{K}_X(t)$ from end-to-end delay measurements Y_i, $i = 1, \ldots, n$.

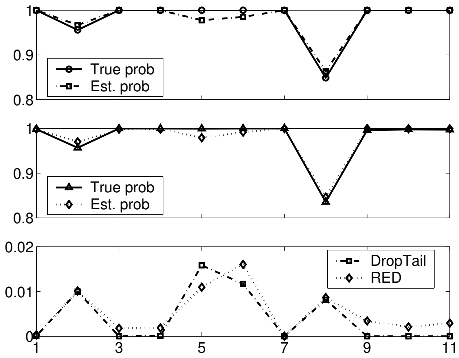
When A is *not full rank*, only linear combinations of those link CGFs lying outside of the null space of A can be determined from (9). We call such a linear combination an *identifiable*



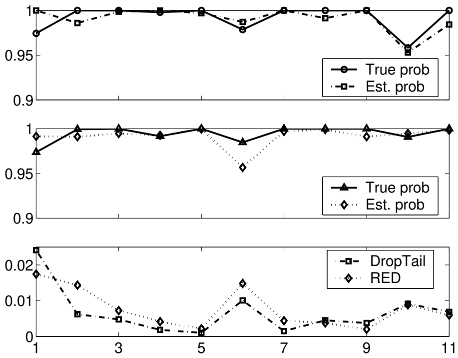
(a) A network consisting of a single sender (node 0), four internal routers (nodes 1-4), and seven receivers (nodes 5-11). The bandwidth in megabits/second (Mb/s) is indicated for each link.



(b) Scenario 1: Heavy losses on links 2 and 5



(c) Scenario 2: Heavy losses on links 2 and 8



(d) Scenario 3: Mixed traffic and medium losses

Figure 4: Performance of the link-level loss tomography framework examined through ns-2 simulation of the network in (a). Subfigures (b)-(d) show true and estimated link-level success rates of TCP flows from the source to receivers for several traffic scenarios, as labeled above. In (b)-(d), the three panels in each display for the success probability (vertical axis) versus link 1-11 (horizontal axis): (top) an example of true and estimated success rates with droptail queues, (middle) true and estimated success rates with RED queues, and (bottom) mean absolute error between estimated and true success rates over 10 independent trials of a 300 second observation interval.

subspace of CGFs. Depending on the routing matrix A, identifiable subspaces can correspond to weighted averages of CGFs $\sum_{j=1}^{m} \alpha_j K_{X_j}(t)$ over a region of the network. This motivates a multi-resolution successive refinement algorithm for detecting and isolating bottlenecks, faults, or other spatially localized anomalies. In such an algorithm large partially overlapping regions of the network are probed with a small number of probes just sufficient for each of the CGF linear combinations to be sensitive to anomalous behavior of the aggregate regional delay distributions. An example of the anomalous behaviors of interest is a sudden shift of the mass of the delay distribution towards larger delay values, possibly indicating an emerging region of congestion. If one of the regions is identified as a potential site of anomalous behavior, a similar probing process can be repeated on subregions of the suspected region. This process continues down to the single link level within a small region and requires substantially fewer probe paths than would be needed to identify the set of all link delay CGF's.

Estimation of the CGF vector $\mathbf{K}_X(t)$ from an i.i.d. sequence of end-to-end probe delay experiments can be formulated as solving a least squares problem in a linear model analogous to (1):

$$\hat{\mathbf{K}}_Y(t) = A\mathbf{K}_X(t) + \boldsymbol{\epsilon}(t). \tag{10}$$

where $\hat{\mathbf{K}}_Y$ is an empirical estimate of the end-to-end CGF vector and $\boldsymbol{\epsilon}$ is a residual error. Different methods of solving for $\mathbf{K}_X$ result from assuming different models for the statistical distribution of the error residual. One model, discussed in [31], is obtained by using a method-of-moments (MOM) estimator for $\mathbf{K}_Y$ and invoking the property that MOM estimators are asymptotically Gaussian distributed as the number of experiments gets large. The bias and covariance of $\hat{\mathbf{K}}_Y$ can then be approximated via bootstrap techniques and an approximate maximum likelihood estimate of $\mathbf{K}_X$ may be generated by solving (10) using iteratively reweighted least squares (LS). Using other types of estimators of $\mathbf{K}_Y$, e.g. kernel based density estimation or mixture models with known or approximatable bias and covariance, would lead to different LS solutions for $\mathbf{K}_X$.

The `ns`-2 network simulator was used to perform a simulation of the 4 link network shown in Figure 5.

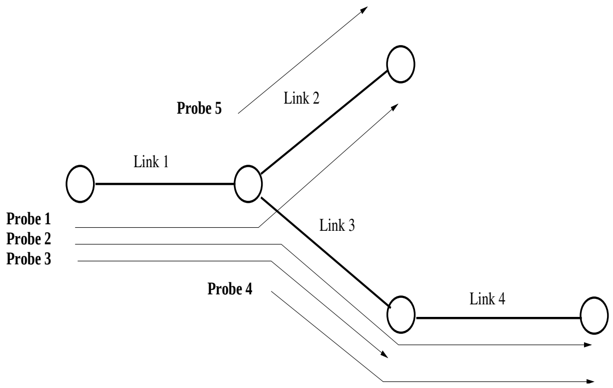


Figure 5: Unicast delay estimation probe routing used in `ns-2` simulation. Tailgating can be used to emulate the internal probes 3,4,5.

Each link was a Drop-Tail queue with buffer size of 50 packets. The internal "bottleneck" link, link 3 in Fig. 5, was assigned bandwidth 5Mbps with latency 50ms. Links 1, 2 and 4 were assigned bandwidths 1Mbps and latency of 10ms. The background traffic consisted of both Exponential on-off UDP traffic and TCP traffic (links 1-4 were assigned different numbers of background UDP and TCP traffic sources in UDP/TCP proportions 6/3, 5/2, 8/4, and 4/2, respectively). Probes were generated as 40 byte UDP packets at each sender node according to a Poisson process with mean interarrival time being 16ms and rate being 20Kb/sec. The number of probes per path was 3000. Probe-derived link CGF estimators with and without bias correction were computed and compared with the true link CGF's (computed from direct link measurements of background traffic alone). Differences between the true CGF's and the estimated CGF's can be attributed to both statistical estimation error and systematic bias due to probe-induced perturbations of background traffic. The link CGF estimate without bias correction was obtained by finding the LS fit to the vector $\mathbf{K}_X(t)$ in relation (10) with $\hat{\mathbf{K}}_Y(t)$ obtained by straight empirical averaging over the $N = 3000$ measured probe delays. Specifically, the i-th element of $\hat{\mathbf{K}}_Y(t)$ is the raw sample average $\hat{K}_{Y_i}(t) = N^{-1}\sum_{k=1}^{N} e^{tY_{ik}}$, where $\{Y_{ik}\}_{k=1}^{N}$ are the probe delays along the i-th probe path among those indicated in Fig. 5. The bias corrected link CGF was estimated using the bootstrap procedure described in [31]. In this procedure we aggregated 40 separate estimates of $\hat{\mathbf{K}}_Y(t)$ each computed over a randomly selected subset of 2800 probe delays taken from the 3000 measured probe delays.

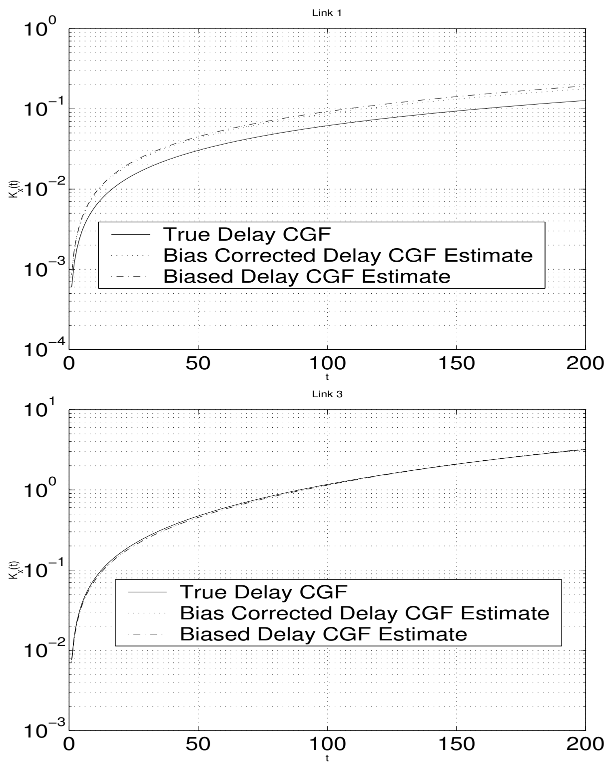


Figure 6: Estimates of the CGF function $K_{X_j}(t)$, $t \geq 0$, for links 1 and 3 compared to the true CGF function.

Figure 6 shows the trajectories of the CGF estimates with and without bias correction in addition to the true CGF for links 1 and 3. Table 1 shows the average squared error per unit t of the link CGF estimates over the range $t \in [-200, 200]$. These estimates were based on applying ordinary LS to (10) with and without bootstrap bias correction. Note from the table that the average MSE of the bias corrected CGF estimate is almost 9% lower than the average MSE incurred by the raw CGF estimate.

Link	1	2	3	4
MSE of $\hat{K}_{X_j}$	0.000909	0.000421	0.000974	0.000325
MSE of $\hat{K}'_{X_j}$	0.001171	0.000327	0.001026	0.000363

Table 1: MSE of $\hat{K}_{X_j}$ (bias correction) and $\hat{K}'_{X_j}$ (no bias correction) for estimated link CGF's. Link 3 is bottleneck link.

We next illustrate the application of the CGF estimation technique to bottleneck detection. Define a bottleneck as the event that a link delay exceeds a specified delay threshold. The Chernoff bound specifies an upper bound on the probability of bottleneck in the j-th link in terms of the CGF

$$P(X_j \geq \delta) \leq \min_{t>0} \left(e^{-t\delta} e^{t\mathbf{K}_{X_j}(t)} \right). \tag{11}$$

In Table 2, we show the estimated Chernoff bounds P_j on the bottleneck probability $P(X_j \geq \delta)$. These were estimated by plugging bias corrected CGF estimates into the right hand side of (11). Here $\delta = 0.005$sec. Note that the estimated Chernoff bounds correctly identify the bottleneck link (link 3) as that link having probability close to 1. In particular if we set the following criterion for detection of a bottleneck: "the probability that X_j exceeds 0.005sec" is at least 0.5, we see that the estimated Chernoff bound correctly identifies link 3 as the bottleneck link.

Link	1	2	3	4
P_j	0.439	0.415	0.964	0.392

Table 2: Estimated Chernoff bounds P_j on $P(X_j \geq 0.005sec)$. Bottleneck at link 3 is correctly identified by its high probability of large delay.

3.5 Example: Topology Identification

Most of the network tomography problems addressed in earlier sections dealt with the identification of network performance parameters, with full knowledge of the network (routing) topology. The network topology is expressed by the matrix $\mathbf{A}$ in equation (1). Knowledge of $\mathbf{A}$ is crucial for most network tomography problems, however such knowledge is not always readily available. Most existing tools for network topology mapping, such as `traceroute`, rely on the cooperation of routers and thus can only reveal those portions of the network that are functioning properly and wish to be known. These cooperative conditions are often not met in practice, and may be increasingly uncommon as the network grows and privacy and proprietary concerns increase.

For situations in which common tools such as `traceroute` are not applicable, a number of methods have been proposed for the identification of network (routing) topology based on end-to-end measurements that measure the degree of correlation between receivers [30, 40, 41, 43, 44, 45]. Most of these approaches have concentrated on identifying the tree structured topology connecting a single sender to multiple receivers. It is assumed that the routes from the sender to the receiver are fixed. With only end-to-end measurements, it is only possible to identify the logical topology defined by the branching points between paths to different receivers.

The key idea in most of the existing topology identification methods is to collect measurements at pairs of receivers that behave (in an average sense) as a monotonic, increasing function of the number of shared links or queues in the paths to the two receivers. A simple example is the case of delay covariance. If two receivers share some portion of their paths, then the covariance between the end-to-end delays to the two receivers is reflective of the sum of the variances on the shared links (assuming the delays are not correlated on unshared links). The more shared links (larger shared portion of their paths), the larger the covariance between the two.

Metrics possessing this type of monotonicity property can be estimated from a number of different end-to-end measurements including counts of losses, counts of zero delay events (utilization), delay correlations, and delay differences [30, 40, 41, 42, 43, 45, 44]. Using such metrics, topology identification can be cast as a Maximum Likelihood estimation problem as follows. The estimated metrics $\boldsymbol{x} \equiv \{x_{i,j}\}$, where the indices i, j refer to different pairs of receivers, can be interpreted as observations of the true metric values $\boldsymbol{\gamma} \equiv \{\gamma_{i,j}\}$ contaminated by some randomness or noise. The estimated metrics are randomly distributed according to a density (whose precise form depends on the contamination model) that is parameterized by the underlying topology $\mathcal{T}$ and the set of true metric values, written as $p(\boldsymbol{x}|\boldsymbol{\gamma}, \mathcal{T})$. The estimated metrics $\boldsymbol{x}$ are fixed quantities and hence when $p(\boldsymbol{x}|\boldsymbol{\gamma}, \mathcal{T})$ is viewed as a function of $\mathcal{T}$ and $\boldsymbol{\gamma}$ it is called the likelihood of $\mathcal{T}$ and $\boldsymbol{\gamma}$. The maximum likelihood tree is given by

$$\mathcal{T}^* = \arg\max_{\mathcal{T}\in\mathcal{F}} \max_{\boldsymbol{\gamma}\in\mathcal{G}} p(\boldsymbol{x}|\boldsymbol{\gamma}, \mathcal{T}), \tag{12}$$

where $\mathcal{F}$ denotes the *forest* of all possible tree topologies connecting the sender to the receivers and $\mathcal{G}$ denotes the set of all metrics satisfying the monotonicity property.

The likelihood optimization in (12) is quite formidable and we are not aware of any method for computing the global maximum except by a brute force examination of each tree in the forest. Consider a network with N receivers. 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. This explosion of the search space precludes the brute force approach in all but very small (logical) networks. While determining the globally optimal tree is prohibitive in most cases, suboptimal algorithms based on deterministic and Monte Carlo optimization methods can provide good estimates of the topology. As far as deterministic algorithms are concerned, the Deterministic Binary Tree (DBT) classification algorithm proposed in [40] is a representative example. The DBT algorithm is a recursive selection and merging/aggregation process that generates a binary tree from the bottom-up (receivers to sender). The greedy nature of the DBT algorithm can lead to very suboptimal results. To avoid this pitfall, a Markov Chain Monte Carlo (MCMC) procedure has been proposed to quickly search through the "topology space," concentrating on regions with the highest likelihood [44]. The most advantageous attribute of the MCMC procedure is that it attempts to identify the topology *globally*, rather than incrementally (and suboptimally) a small piece at a time.

To illustrate the topology identification problem, consider the network topology depicted in Figure 7(a). This is the true topology connecting a sender (at Rice University) to a number of other computers in North America and a couple in Europe. In this case, `traceroute` was used to obtain the true topology (in many cases this may not be possible, but here it provides a convenient "ground-truth" for our experiment). End-to-end measurements using a special-purpose unicast probes called "sandwich" probes were used to obtain a set of metrics satisfying the monotonicity property [44]. The sandwich probing scheme is delay-based, but it measures only *delay differences*, so that no clock synchronization is required. Figure 7(b) depicts the most commonly identified topology (over many different experiments on different days and at different times of day). The identified topology generally agrees with the true topology.

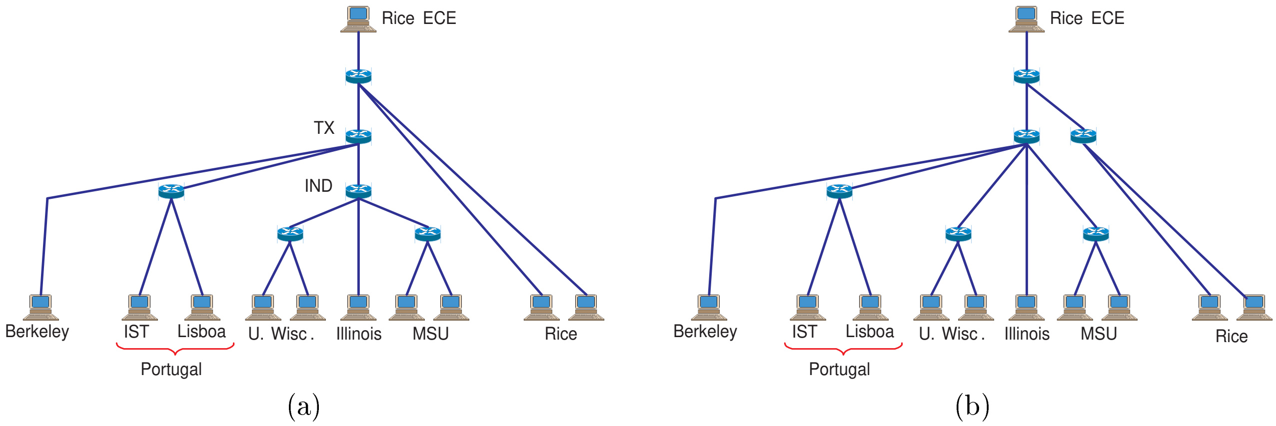


Figure 7: (a) The topology of the network used for Internet experiments. (b) Most commonly estimated topology using the MPL criterion. The link between TX and IND is not detected and an extra common link is associated with the Rice clients, but otherwise the estimated topology is a faithful representation of the true topology.

4 Origin-Destination Tomography

Origin-destination tomography is essentially the antithesis of link-level network tomography: the goal is the estimation of path-level network parameters from measurements made on individual links. By far the most intensively studied origin-destination network tomography problem is the estimation of origin-destination (OD) traffic from measurable traffic at router interfaces. In privately-owned networks, the collection of link traffic statistics at routers within the network is often a far simpler task than performing direct measurement of OD traffic. The OD traffic matrix, which indicates the intensity of traffic between all origin-destination pairs in a network, is a key input to any routing algorithm, since the link weights of the Open Shortest Path First[5] (OSPF) routing protocol are related to the traffic on the paths. Ideally, a data-driven OD

[5]Open Shortest Path First (OSPF) is a routing protocol developed for IP networks. OSPF is a *link-state* routing protocol that calls for the sending of *link-state advertisements* to all other routers in the same hierarchical area. A link state takes the form of a weight, effectively the cost of routing via that link.

matrix should be central to the routing optimization program.

There are currently two ways to obtain OD traffic counts. Indirect methods collect sums of OD traffic counts and are considered in [22, 33, 36, 34]; a direct method to measure OD traffic counts via software such as *NetFlow* supported by Cisco routers is described in [34, 57]. Both approaches need the cooperation of the routers in the network, but this is not problematic for privately-owned networks. The link traffic counts at routers are much easier to collect relative to the direct approach via *NetFlow* and lead to a linear inverse problem. There are noticeable features about this particular inverse problem worthy of elaboration. Firstly, the OD traffic vector to be estimated is not a fixed parameter vector, but a random vector, denoted by $\boldsymbol{x}$; secondly, the linear equation (1) is used without the error term $\boldsymbol{\epsilon}$ (stochastic variability is captured in $\boldsymbol{x}$). Although $\boldsymbol{A}$ is singular as in other cases discussed, the techniques in [22, 33, 36, 34] use statistical means to induce a regularization enabling the recovery of the entire $\boldsymbol{x}$ (or the traffic intensities underlying $\boldsymbol{x}$). Moreover, the most recent work [34] addressing this problem also deals with the time-varying or nonstationary aspect of the data.

Vardi was the first to investigate the OD network tomography problem. In [22] he studies a network with a general topology, using an independent and identically distributed (i.i.d.) Poisson model for the OD traffic byte counts. He specifies identifiability conditions under the Poisson model and develops a method that uses the EM algorithm on link data to estimate Poisson parameters in both deterministic and Markov routing schemes. To mitigate the difficulty in implementing the EM algorithm under the Poisson model, he proposes a moment method for estimation and briefly discusses the normal model as an approximation to the Poisson. Related work treated the special case involving a single set of link counts and also employed an EM algorithm [36]. A Bayesian formulation and Markov Chain Monte Carlo estimation technique has also been proposed [33].

Cao *et al.* [34] use real data to revise the Poisson model and to address the non-stationary aspect of the problem. Their methodology is validated through comparison with direct (but expensive) collection of OD traffic. Cao *et al.* represent link count measurements as summations of various OD counts that were modeled as independent random variables. (Even though TCP feedback 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 (or non-stationary) traffic matrices estimated from a sequence of link counts were validated on a small subnetwork with 4 origins/destinations by comparing the estimates with actual OD counts that were collected by running Cisco's *NetFlow* software on the routers. Such direct point-to-point measurements are often not available because they require additional router CPU resources, which can reduce packet forwarding efficiency, and involve a significant administrative burden when used on a large scale.

Let $\boldsymbol{x} = (x_1, \ldots, x_n)^T$ denote the *unobserved* vector of corresponding byte counts for all OD pairs during a given time interval in the network. Here ^{T} indicates transpose and $\boldsymbol{x}$ is the 'traffic matrix' even though it is arranged as a column vector for convenience. One natural way to enumerate all the OD variables into a vector is to first enumerate all the routers and then the end nodes or origin-destination nodes by 1 through, say, I, and make these indices blocked by routers: the end nodes connected to the first router in the first block, and those connected to the second router in the second block, and so forth. Then, to form the OD vector, we put the OD traffic accounts in the order $(1,1),(1,2),...,(1,I),(2,1),(2,2),...,(2,I),...,(I,1),(I,2),...,(I,I)$, where (i,j) is the index of the OD traffic from the ith end node to the jth end node. Let $\boldsymbol{y} = (y_1,\ldots,y_m)^T$ denote the *observed* column vector of incoming/outgoing byte counts measured on each router link interface during a given time interval, again blocked into first the link measurements on the interfaces of the first router and so on. One element of $\boldsymbol{x}$, for example, corresponds to the number of bytes originating from a specified origin node to a specified destination node, whereas one element of $\boldsymbol{y}$ corresponds to bytes sent from the origin node regardless of their destination. Thus each element of $\boldsymbol{y}$ is a sum of selected elements of $\boldsymbol{x}$, so

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} \tag{13}$$

where $\boldsymbol{A}$ is defined as before, an $m \times n$ *routing matrix* of 0's and 1's that is determined by the routing scheme of the network. The orders of elements in $\boldsymbol{x}$ and $\boldsymbol{y}$ determine the positions of the 0's and 1's of $\boldsymbol{A}$ accordingly. The work of [34] only considers *fixed routing*, i.e. there is only one route from an origin to a destination. The unobserved OD byte counts are modeled as

$$x_i \sim \text{normal}(\lambda_i, \phi\lambda_i^c), \text{ independently,} \tag{14}$$

where c is a fixed power constant (its specification is found to be robust in the sense that both $c = 1$ and $c = 2$ work well with the Lucent network data as shown in [34, 35]). This implies

$$\boldsymbol{y} \sim \text{normal}(\boldsymbol{A}\lambda,\ \boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{A}^T), \tag{15}$$

where

$$\lambda = (\lambda_1,\ldots,\lambda_n)^T, \text{ and } \boldsymbol{\Sigma} = \phi\ \text{diag}(\lambda_1^c,\ldots,\lambda_n^c).$$

Here $\lambda > \mathbf{0}$ is the vector of OD mean rates. $\phi > 0$ is a scale parameter that relates the variance of the counts to their mean, since usually larger counts have larger variance. The mean-variance relationship is necessary to ensure the identifiability of the parameters in the model. Heuristically, under this constraint, the covariances between the y's give the identifiability of the parameters up to the scale parameter ϕ which can be determined from the expectation of a y.

Cao *et al.* [34] address the non-stationarity in the data using a local likelihood model (cf. [58]); that is, for any given time t, analysis is based on a likelihood function derived from the observations within a symmetric window of size $w = 2h + 1$ 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 (MLE) is carried out for the parameter estimation via a combination of the EM algorithm and a second-order global optimization routine. The component-wise conditional expectations of the OD traffic, given the link traffic, estimated parameters, and the positivity constraints on the OD traffic, are used as the initial estimates of the OD traffic. The linear equation $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ is enforced via the iterative proportional fitting algorithm (cf. [59, 60]) 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 state-space model is imposed in [34] on the logarithm of the parameters λ's and ϕ over the time windows of size $w = 2h + 1$ (in our implementation for the simple network of Router 1, we use h=5 or w=11). Let $\theta_t = (\lambda_t, \phi_t)$ be the parameter vector for the tth time window. We assume the following random walk model for the evolution of the log parameters:

$$\log(\theta_t) = \log(\theta_{t-1}) + \mathbf{v}_t,$$

where $\mathbf{v}_t \sim \text{normal}(\mathbf{0},\ D)$, independent for different t, and D is a diagonal matrix obtained using estimates of θ_t in the MLE approach described earlier. Given the parameters, the link counts are assumed i.i.d. as before:

$$(Y_{t-h}, ..., Y_t, ..., Y_{t+h})^T | \theta_t \sim \text{i.i.d. Normal}(\mathbf{A}\lambda_t,\ \mathbf{A}\mathbf{\Sigma}_t\mathbf{A}^T).$$

This leads to a two-pass algorithm on the data. For the second pass, inference at time t is carried out in a sequential manner. We first obtain the posterior probability density $p(\theta_{t-1})$ based on the first $t-1$ windows of data, then the prior probability density $\pi(\theta_t)$ is updated via the random walk equation, and then the maximum a posterior estimate of θ_t via numerical optimization using the observations in the tth time window and the prior.

This state-space model does improve on the parameter estimates, but not so much on the estimated OD traffic $\boldsymbol{x}_t$, which implies an insensitivity of the final OD traffic estimates. This insensitivity or robustness to changes in parameter estimates is probably due to the fact that even in the MLE approach, positivity and linear constraints are imposed on the OD estimates, and these constraints override the improvements brought about by the state-space model.

4.1 Example: Time-varying OD Traffic Matrix Estimation

Figure 8 is a network at Lucent Technologies considered in [34, 35]. Figures 9 and 10 are taken from [34]: traffic plots only for the subnetwork around Router 1 with four origin-destination end nodes. These plots show the validation (via *NetFlow*) and estimated OD traffic based on the link traffic. Figure 9 gives the full scale and Figure 10 is the zoomed-in scale (20×). It is obvious that the estimated OD traffic agrees well with the *NetFlow* measured OD traffic for large measurements ($>$ 50 K bytes/sec), but not so well for small measurements ($<$ 20 K bytes/sec) where the Gaussian model is a poor approximation. From the point of view of traffic engineering, it is adequate that the large traffic flows are inferred accurately. Hence for some purposes such as planning and provisioning activities estimates of OD traffic could be used as inexpensive substitutes for direct measurements.

Even though the method described in [34] uses all available information to estimate parameter values and the OD traffic vector $\boldsymbol{x}$, it does not 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. A scalable algorithm that relies on a divide-and-conquer strategy to lower the computational cost without losing much of the estimation efficiency is proposed in [35].

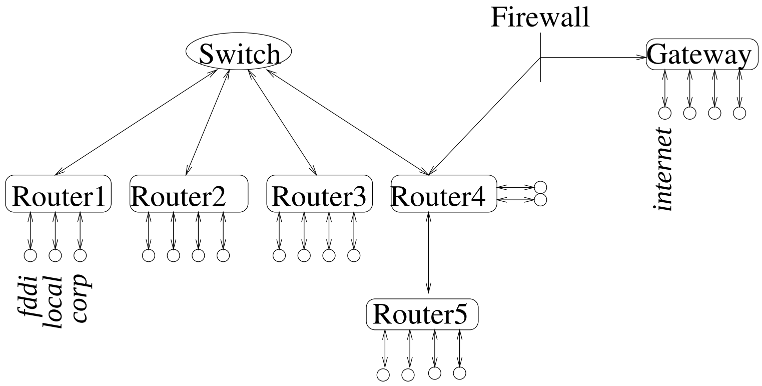


Figure 8: A network at Lucent Technologies

5 Conclusion and Future Directions

This article has provided an overview of the area of large scale inference and tomography in communications networks. As is evident from the limited scale of the simulations and experiments discussed in this article, the field is only just emerging. Deploying measurement/probing schemes and inference algorithms in larger networks is the next key step. Statistical signal processing will continue to play an important role in this area and here 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 applications areas.

The spatio-temporally stationary and independent traffic and network transport models have limitations, especially in tomographic applications involving 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 [31] and multicast [24] 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 spatial dependence and routing dynamics will become dominant.

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 4, others have adopted a dynamical systems approach to handle nonstationary link-level tomography problems [14]. Sequential Monte Carlo inference techniques are employed in [14] 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. Figure 11 illustrates this scenario and displays the estimated delay distributions at different time instances (see [14] for further details).

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 [49]. 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 communications overhead of centralized data collection from edge-nodes.

The majority of work reported to date has focused on reconstruction of network parameters which may only be 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 which has been considered in [47, 32] as an application of reconstructed delay or loss estimation. However, systematic development of large scale hypothesis testing theory for networks would undoubtedly lead to superior detection performance. 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 signal processing 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 the National Science Foundation, grant nos. MIP–9701692, ANI-0099148, FD01-12731, and ANI-9734025, the Office of Naval Research, grant no. N00014-00-1-0390, the Army Research Office, grant nos. DAAD19-99-1-0290, DAAD19-01-1-0643, and DAAH04-96-1-0337, and the Department of Energy, grant no. DE-FC02-01ER25462. The authors would also like to acknowledge the invaluable contributions of J. Cao, R. Castro, D. Davis, M. Gadhiok, R. King, E. Rombokas, C. Shih, Y. Tsang, and S. Vander Wiel to the work described in this article.

References

[1] CAIDA: Cooperative Association for Internet Data Analysis. http://www.caida.org/Tools/.

[2] M. Dodge and R. Kitchin. *Atlas of Cyberspace*. Pearson Education, 2001.

[3] F. P. Kelly, S. Zachary, and I. Ziedins. *Stochastic networks: theory and applications*. Royal Statistical Society Lecture Note Series. Oxford Science Publications, Oxford, 1996.

[4] X. Chao, M. Miyazawa, and M. Pinedo. *Queueing networks: customers, signals and product form solutions*. Systems and Optimization. Wiley, New York, NY, 1999.

[5] UCB/LBNL/VINT network simulator ns (version 2). URL: http://www.isi.edu/nsnam/ns/.

[6] RTP: A transport protocol for real-time applications, Jan. 1996. IETF Internet Request For Comments: RFC 1889.

[7] N. Duffield and M. Grossglauser. Trajectory sampling for direct traffic observation. In *Proc. ACM SIGCOMM 2000*, Stockholm, Sweden, Aug. 2000.

[8] W. Leland, M. Taqqu, W. Willinger, and D. Wilson. On the self-similar nature of Ethernet traffic (extended version). *IEEE/ACM Trans. Networking*, pages 1–15, 1994.

[9] V. Paxson. End-to-end internet packet dynamics. In *Proc. ACM SIGCOMM*, 1997.

[10] R. Riedi, M. S. Crouse, V. Ribeiro, and R. G. Baraniuk. A multifractal wavelet model with application to TCP network traffic. *IEEE Trans. Info. Theory, Special issue on multiscale statistical signal analysis and its applications*, 45:992–1018, April 1999.

[11] A. Feldmann, A. C. Gilbert, P. Huang, and W. Willinger. Dynamics of IP traffic: a study of the role of variability and the impact of control. In *Proc. ACM SIGCOMM*, pages 301–313, Cambridge, MA, 1999.

[12] A. C. Gilbert, W. Willinger, and A. Feldmann. Scaling analysis of conservative cascades, with applications to network traffic. *IEEE Trans. Info. Theory*, IT-45(3):971–991, Mar. 1999.

[13] A. Veres, Z. Kenesi, S. Molnár, and G. Vattay. On the propagation of long-range dependence in the Internet. In *Proc. ACM SIGCOMM 2000*, Stockholm, Sweden, Aug. 2000.

[14] M. Coates and R. Nowak. Sequential Monte Carlo inference of internal delays in nonstationary communication networks. *IEEE Trans. Signal Processing, Special Issue on Monte Carlo Methods for Statistical Signal Processing*, 2002.

[15] J. Kurose and K. Ross. *Computer Networking: A top-down approach featuring the Internet.* Addison-Wesley, 2001.

[16] C. Huitema. *Routing in the Internet.* Prentice-Hall, 2000.

[17] S. Keshav. *An Engineering Approach to Computer Networking.* Addison-Wesley, 1997.

[18] L. Peterson and B. Davie. *Computer Networks: A Systems Approach.* Morgan Kaufmann, 2000.

[19] W. Stevens. *UNIX Network Programming, vol. 1: Networking APIs: Sockets and XTI.* Prentice-Hall, 1997.

[20] A. Tanenbaum. *Computer Networks.* Prentice-Hall, 1996.

[21] W. Stevens and G. Wright. *TCP/IP Illustrated.* Addison-Wesley, 2002.

[22] Y. Vardi. Network tomography: estimating source-destination traffic intensities from link data. *J. Amer. Stat. Assoc.*, pages 365–377, 1996.

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

[24] R. Cáceres, N. Duffield, J. Horowitz, and D. Towsley. Multicast-based inference of network-internal loss characteristics. *IEEE Trans. Info. Theory*, 45(7):2462–2480, November 1999.

[25] M. Coates and R. Nowak. Network loss inference using unicast end-to-end measurement. In *ITC Seminar on IP Traffic, Measurement and Modelling*, Monterey, CA, Sep. 2000.

[26] M. Coates and R. Nowak. Network delay distribution inference from end-to-end unicast measurement. In *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Proc.*, May 2001.

[27] N.G. Duffield, F. Lo Presti, V. Paxson, and D. Towsley. Inferring link loss using striped unicast probes. In *Proceedings of IEEE INFOCOM 2001*, Anchorage, Alaska, April 2001.

[28] A. Bestavros K. Harfoush and J. Byers. Robust identification of shared losses using end-to-end unicast probes. In *Proc. IEEE Int. Conf. Network Protocols*, Osaka, Japan, Nov. 2000. *Errata* available as Boston University CS Technical Report 2001-001.

[29] F. Lo Presti, N.G. Duffield, J. Horowitz, and D. Towsley. Multicast-based inference of network-internal delay distributions. Technical report, University of Massachusetts, 1999.

[30] S. Ratnasamy and S. McCanne. Inference of multicast routing trees and bottleneck bandwidths using end-to-end measurements. In *Proceedings of IEEE INFOCOM 1999*, New York, NY, March 1999.

[31] M.F. Shih and A.O. Hero. Unicast inference of network link delay distributions from edge measurements. Technical report, Comm. and Sig. Proc. Lab. (CSPL), Dept. EECS, University of Michigan, Ann Arbor, May 2001.

[32] A.-G. Ziotopolous, A.O. Hero, and K. Wasserman. Estimation of network link loss rates via chaining in multicast trees. In *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Proc.*, May 2001.

[33] C. Tebaldi and M. West. Bayesian inference on network traffic using link count data (with discussion). *J. Amer. Stat. Assoc.*, pages 557–576, June 1998.

[34] J. Cao, D. Davis, S. Vander Wiel, and B. Yu. Time-varying network tomography: router link data. *J. Amer. Statist. Assoc*, 95:1063–1075, 2000.

[35] J. Cao, S. Vander Wiel, B. Yu, and Z. Zhu. A scalable method for estimating network traffic matrices from link counts. URL: http://www.stat.berkeley.edu/~binyu/publications.html, 2000.

[36] R.J. Vanderbei and J. Iannone. An EM approach to OD matrix estimation. Technical Report SOR 94-04, Princeton University, 1994.

[37] R. J. Mammone. Inverse problems and signal processing. In *The Digital Signal Processing Handbook*, chapter VII. CRC Press, Boca Raton, FL, 1998.

[38] Finbarr O'Sullivan. A statistical perspective on ill-posed inverse problems. *Statistical Science.*, 1(4):502–527, 1986.

[39] F. Natterer. *The Mathematics of Computerized Tomography.* Wiley, New York, 1986.

[40] N.G. Duffield, J. Horowitz, F. Lo Presti, and D. Towsley. Multicast topology inference from end-to-end measurements. In *ITC Seminar on IP Traffic, Measurement and Modelling*, Monterey, CA, Sep. 2000.

[41] N.G. Duffield, J. Horowitz, F. Lo Presti, and D. Towsley. Multicast topology inference from measured end-to-end loss. *IEEE Trans. Information Theory*, 2002.

[42] N. G. Duffield, J. Horowitz, F. Lo Presti, and D. Towsley. Multicast topology inference from measured end-to-end loss. *IEEE Trans. Info. Theory*, 48(1):26–45, January 2002.

[43] A. Bestavros, J. Byers, and K. Harfoush. Inference and labeling of metric-induced network topologies. Technical Report BUCS-2001-010, Computer Science Department, Boston University, June 2001.

[44] R. Castro, M.J. Coates, M. Gadhiok, R. King, R. Nowak, E. Rombokas, and Y. Tsang. Maximum likelihood network topology identification from edge-based unicast measurements. Technical Report TREE0107, Department of Electrical and Computer Engineering, Rice University, Oct. 2001.

[45] R. Castro, M. Coates, and R. Nowak. Maximum likelihood identification of network topology from end-to-end measurement. In *DIMACS Workshop on Internet and WWW Measurement, Mapping and Modeling, DIMACS Center, Rutgers University, Piscataway, New Jersey*, February 2002.

[46] M. Coates and R. Nowak. Networks for networks: Internet analysis using Bayesian graphical models. In *IEEE Neural Network for Signal Processing Workshop*, Sydney, Aust., Dec. 2000.

[47] M.F. Shih and A.O. Hero. Unicast inference of network link delay distributions from edge measurements. In *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Proc.*, May 2001.

[48] N. Duffield and F. Lo Presti. Multicast inference of packet delay variance at interior network links. In *Proceedings of IEEE INFOCOM 2000*, Tel Aviv, Israel, Mar. 2000.

[49] Y. Tsang, M. Coates, and R. Nowak. Passive network tomography using EM algorithms. In *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Proc.*, May 2001.

[50] Recommendations on queue management and congestion avoidance in the Internet, Apr. 1998. IETF Internet Request For Comments: RFC 2309.

[51] V. Paxson. End-to-end Internet packet dynamics. *IEEE/ACM Trans. Networking*, 7(3):277–292, June 1999.

[52] K. Lai and M. Baker. Measuring link bandwidths using a deterministic model of packet delay. In *Proc. ACM SIGCOMM 2000*, Stockholm, Sweden, Aug. 2000.

[53] Y. Tsang, M. Coates, and R. Nowak. Passive unicast network tomography based on tcp monitoring. Technical Report TREE-05, Rice University, 2000.

[54] R. Nowak and M. Coates. Unicast network tomography using the EM algorithm. *submitted to IEEE Trans. Info. Th.* (also see Rice University Tech. Report TREE-0108), 2002.

[55] J-C. Bolot. End-to-end packet delay and loss behaviour in the Internet. In *Proc. ACM SIGCOMM 1993*, pages 289–298, Sept. 1993.

[56] Y. Tsang, M. Coates, and R. Nowak. Nonparametric internet tomography. In *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Proc.*, May 2002.

[57] A. Feldmann, A. Greenberg, C. Lund, N. Reingold, J. Rexford, and F. True. Deriving traffic demands for operational IP networks: methodology and experience. In *Proc. ACM SIGCOMM 2000*, Stockholm, Sweden, Aug. 2000.

[58] C. Loader. *Local regression and likelihood.* Springer, New York, NY, 1999.

[59] W.E. Deming and F.F. Stephan. On a least squares adjustment of a smapled frequency table when the expected marginal totals are known. *Annals of Mathematical Statistics*, 11:427–444, 1940.

[60] I. Csiszár. Divergence geometry of probability distributions and minimization problems. *Ann. Prob.*, 3(1):146–158., 1975.

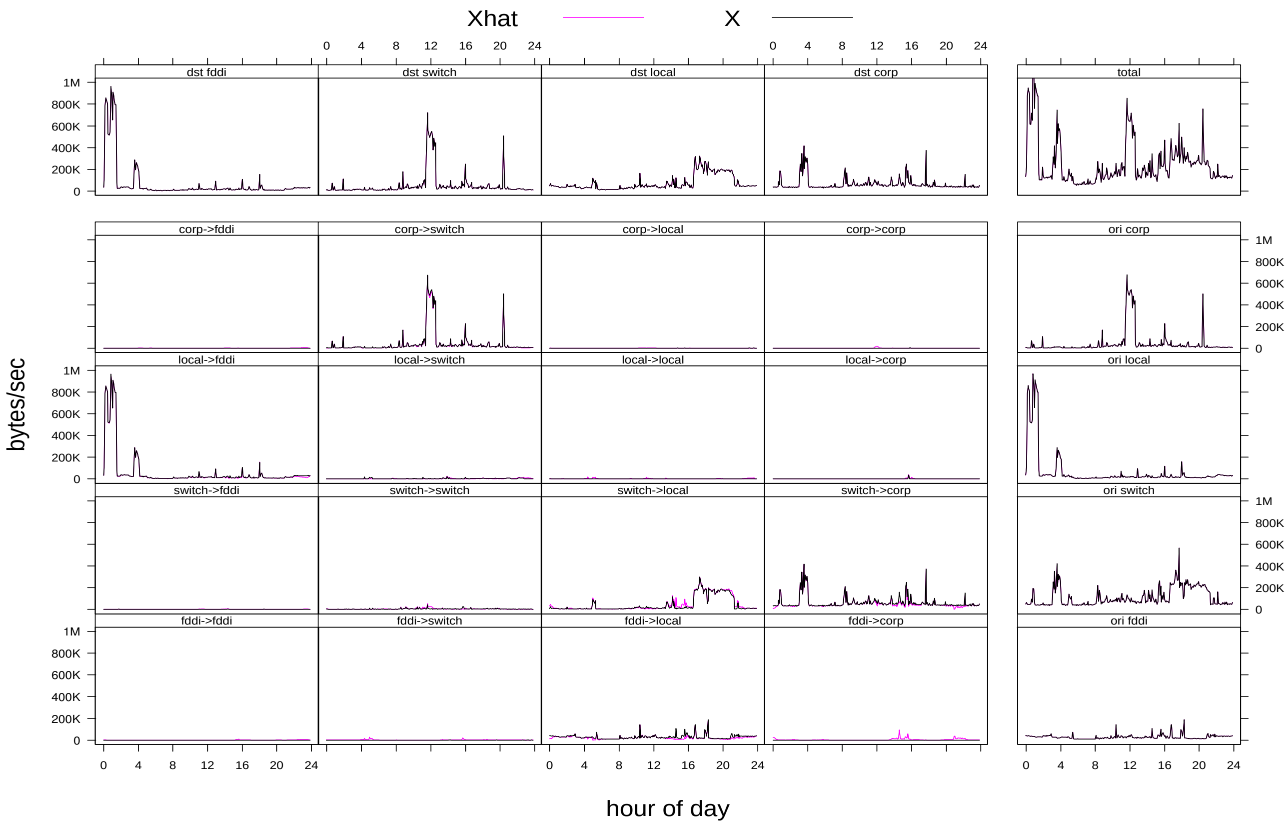


Figure 9: Full-scale time series plots of OD traffic on Feb. 22, 1999 for Router 1 sub-network with 4 origins/destinations. In the lower-left 4×4 matrix, the rows (from TOP down) correspond to corp, local, switch and fddi and the columns (from RIGHT to LEFT) correspond to corp, local, switch and fddi. These 4×4 main panels correspond to the 16 OD pairs. For example, the (1,2) panel is corp $\rightarrow$ switch. The 8 marginal panels (above and to the right of the main matrix) are the observed link traffic used to infer the 16 OD traffic pairs. The top-right corner shows the total observed link traffic. Xhat is the estimated OD traffic and X is the observed OD traffic. At this time-scale it is impossible to differentiate between estimated and observed OD traffic in most panels of the matrix.

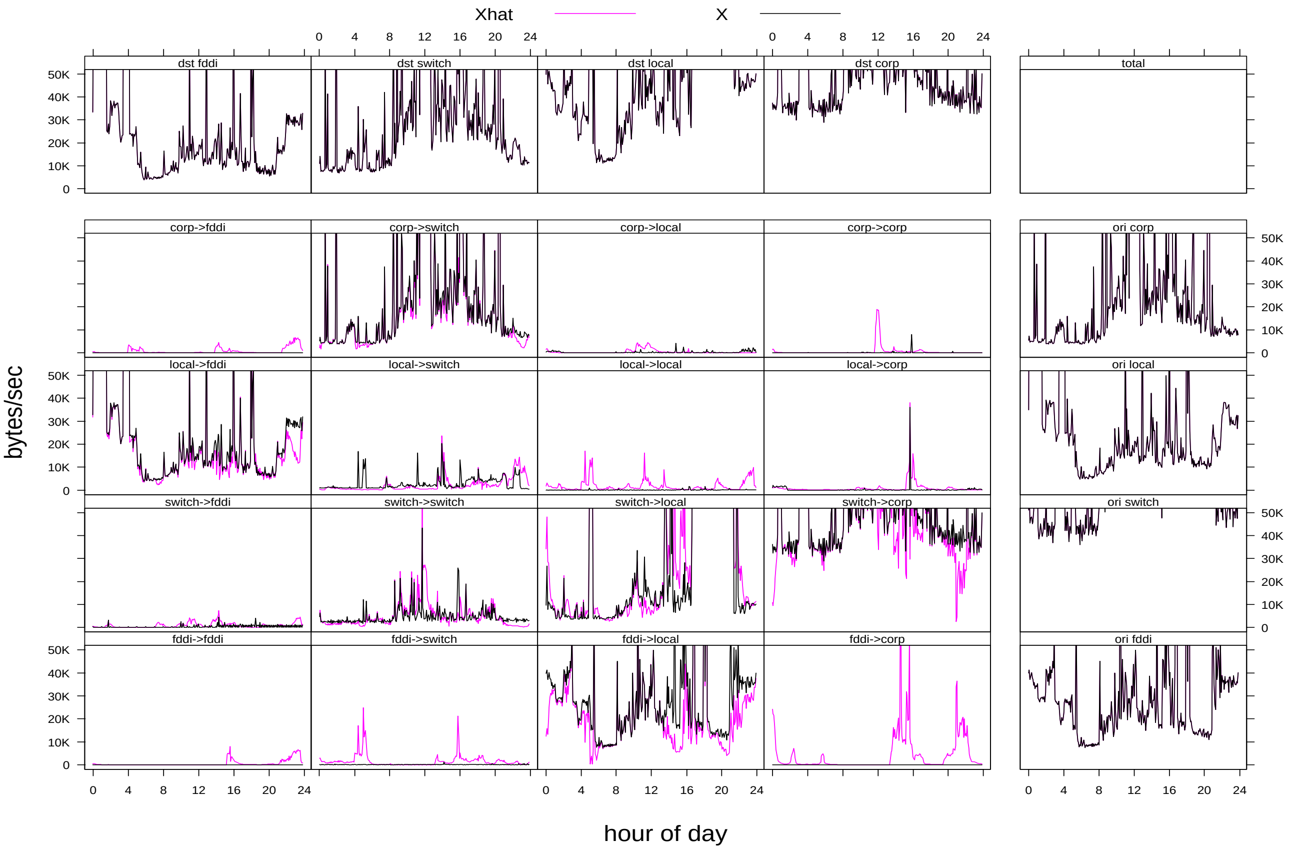
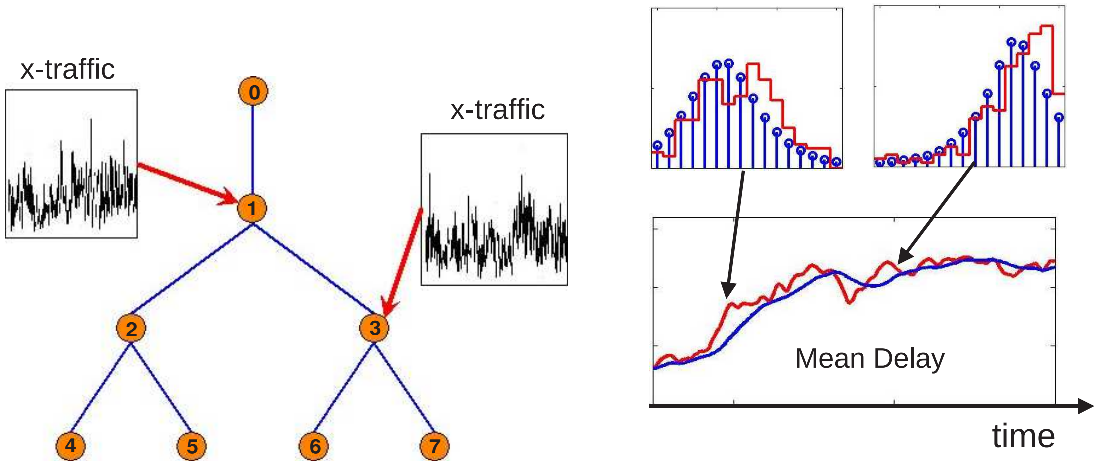


Figure 10: Time series plots of OD traffic like in Fig. 6, except that the scale is zoomed in. At this zoomed-in time-scale it is easier to differentiate between estimated and observed OD traffic in most panels, particularly when there is a small traffic load.



(a) Network with nonstationary cross-traffic

(b) Tracking delay characteristics

Figure 11: Performance of the sequential Monte Carlo tracking of time-varying link delays from end-to-end measurements. (a) Single source, four receiver simulated network with nonstationary cross-traffic. (b) True delay distributions (red) and estimates (blue) as a function of time.