Large scale inference and tomography for network monitoring and diagnosis.

Mark Coates Alfred Hero Robert Nowak Bin Yu

June, 2001

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 anamolous/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, pattern recognition, system identification, and array processing all have interesting interpretations in the networking context. This article introduces the new field of large-scale network inference, a field which we believe will benefit greatly from the wealth of signal processing research 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 decentralized multi-layered collection of heterogeneous terminals, routers and other platforms that we encounter today when surfing the web. Unlike, for example, the US 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 has allowed Internet service providers (ISP)'s to develop a rich variety of user-services at different quality-of-service (QoS) levels. However, in such a decentralized environment quantitative assessment of network performance is difficult. One cannot depend on the cooperation of individual servers and routers to freely transmit vital network statistics such as traffic rates, link delays, and dropped packet rates. Indeed, an ISP may regard such information as highly confidential. On the other hand, sophisticated methods of active probing and/or passive monitoring can be used to extract useful statistical quantities that can reveal hidden network structure and detect and isolate congestion, routing faults, and anomalous traffic. The problem of extracting such 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. In particular, it is likely that the solution of such network inverse problems will benefit from signal processing know-how acquired in areas such as image reconstruction, pattern recognition, system identification, and sensor array signal processing.

This article deals with large scale network monitoring and inference for wired networks such as the Internet. Researchers in this area have taken an approach which is different from previous approaches relying on detailed queueing and traffic models or approaches relying on closely cooperating nodes. The problems involve estimating a potentially very large number of simple spatially distributed parameters, e.g., single link loss rates, delay distributions, connectivity, and traffic flow. To tackle such large tasks, researchers adopt the simplest possible models for network traffic and ignore many intricacies of packet transport such as feedback and latency. Focus is shifted from detailed mathematical modeling of network dynamics to careful handling of measurement and probing strategies, large scale computations, and model validation. 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 computation science involves: complexity reducing hierarchical statistical models; moment-based estimation; EM algorithms; Monte-Carlo Markov Chain 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 `ns`; and application to real network trace data. However, it need be emphasized that while simpler models may enable inference of gross-level performance characteristics, they may not be sufficient for fine-grain analysis of individual queueing mechanisms and network traffic behavior.

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: 1) development of software tools to monitor/probe the network; 2) probabilistic modeling of networks of queues; and 3) 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 special 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 [1]. Active probing tools such as `ping`, `pathchar` (`pchar`), `clink`, and `tracerout` measure and report packet transport attributes of the route along which a probe makes a round trip from source to destination and back to source. A survey of these and other Internet measurement software tools can be found on the CAIDA (Cooperative Association for Internet Data Analysis) web site `http://www.caida.org/Tools/`. Trajectory sampling measurement packets [2] 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.

Queueing networks offer a rich mathematical framework which can be useful for analyzing small scale networks with a few interconnected servers. See the recent edited books by Kelly and others for a comprehensive overview of this area [3, 4]. The limitations of queueing 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 qualititatively different from what is observed at a microscopic scale with a few isolated atomic nuclei. Furthermore, detailed information on queueing 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 queueing network analysis becomes necessary when addressing a different set of problems that can be solved offline. Such problems include calculating accurate estimates of fine grain network behavior, e.g. the dynamics of node traffic rates, service times, and buffer states.

The area of statistical modeling of network traffic is a mature and active field [5, 6, 7, 8, 9]. Sophisticated fractal and multifractal models of single traffic streams can account for long range dependency, heavy tailed distributions, and other peculiar behaviors. Such self similar behavior of traffic rates has been validated for heavily loaded wired networks [10]. 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 solve the inverse problem arising in each of these applications. In many cases, algorithms requiring moderate computation can obtain accurate tomographic reconstructions of internal network parameters despite ignoring effects such as long range dependency. While some progress has been made on incorporating simple first order spatio-temporal dependency models into large scale network inference problems much work remains to be done.

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 then turn to origin-destination traffic matrix inference from link measurements in the context of both stationary and non-stationary traffic.

2 Network Tomography

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. 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 (i.e., routers).

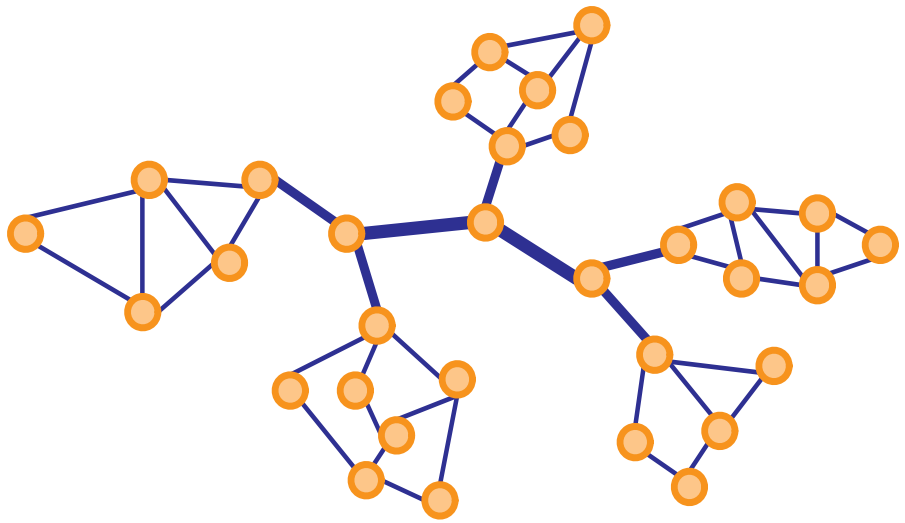


Figure 1: An arbitrary network topology. Each node represents a computer or a cluster of computers or a router. Each edge in the graph represents a direct link between two nodes. The topology here depicts "clusters" corresponding to local area networks or other subnetworks connected together via the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

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* [11] 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 [12, 13, 14, 15, 16, 17, 18, 19, 20, 21] and (ii) sender-receiver path-level traffic intensity estimation based on link-level traffic measurements [22, 11, 23, 24, 25].

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 queueing delay

at the link's source node. A packet is dropped at a link 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 at the link's source node, but may also be caused by equipment downtime due to maintenance or power failures. Random link delays and packet losses become particularly significant 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 moreso 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 it 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 [26], statistics [27], and in applied mathematics [28]. 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 of 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 paper. 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.

2.1 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, source-to-destination (SD) counts of packet losses[1]. Let $\boldsymbol{\theta}$ denote the collection of log success probabilities for each link. The SD log success probability is simply $\boldsymbol{A\,\theta}$. Assuming a known number of packets sent from each source to destination, a binomial measurement model can be adopted [14]. 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, SR delays are measured and the goal is estimation of the delay probability distributions along each link. In 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 [20], $\boldsymbol{y}$ is again related to $\boldsymbol{\theta}$ according to the linear model (1). Third, in the origin-destimation (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 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

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

behavior which involve analogs of classical Kalman-filtering methods [23, 15]. 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 discovery" problem [19, 29], 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 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 [12] pioneered the use of multicast probing for network tomography, and stimulated much of the current work in this area [12, 13, 14, 30, 15, 16, 18, 19, 20, 31, 21].

We now outline a general framework for the link-level tomography problems considered in this section. Consider 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 lost along the way and determine the transmission delay by way of some form of acknowledgement from the receiver to the sender upon successful packet reception. It is assumed that the senders 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 r nodes connected by m links, labeled $j = 1, \ldots, m$. Potentially, a logical link connecting two nodes represents many routers and the physical links between them. Let there be n distinct measurement paths (from a sender to a receiver) through the network, labeled $i = 1, \ldots, n$. 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 $(0, 1)$ 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}.

Figure 2 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

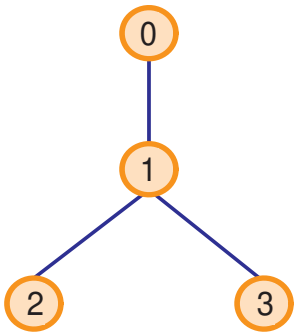


Figure 2: Tree-structured topology.

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 the matrix A has the form:

$$A = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \tag{3}$$

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

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 synchronised 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.

There are two 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 receivers. At internal routers where branching occurs, e.g., node 1 in Figure 2, each multicast packet is duplicated 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 [13]. Consider loss rate tomography for the network depicted in Figure 2. 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 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. This ratio 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 [13, 32].

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 the 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 [18, 32].

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 2, 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 [14] and [17] 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 describes two packets that are 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 droptail policy[2], if two back-to-back packets are sent across a common link and one of the pair is sucessfully 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 has been verified experimentally in real networks [34, 16]. If one assumes that the probability of success for one packet conditional 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 [16].

In the case of bandwidth tomography, the authors of [35] 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; when the counter reaches zero, the next router discards the packet. The `nettimer` program described in [35] 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 2 for illustration, `nettimer` might send packet-pairs form node 0 along path 0-1-2. If the TTL of the large packet is set to one, the tailgating smaller packet measures the propagation delay on link 0-1.

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 sessions). Loss rate and delay distribution tomography methods have been developed specifically for unicast packet pairs in [14, 17, 36, 37]. Unicast packet *stripes* (triples, quadruples, etc.) have also been investigated for loss rate tomography [16].

[2] A droptail queueing 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 queueing strategies, such as random-early-detection (RED) [33], packets can be dropped even if they have already entered the queue

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 [14, 15].

The successful traversal of a single packets 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 α_j, $j \in 1, \ldots, m$ form 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 reached the destination, denoted as m_k. Determination of the success of a given packet is handled by an acknowledgement sent from the receiver back to the sender. For example, such acknowledgements 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 [14, 17, 16].

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

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

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 $\beta_j \approx 1$ 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 [38, 34].

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

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

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 [14].

The link-level loss inference framework is evaluated in [37] using the `ns-2` network simulation environment [39]. Measurements were collected by passively monitoring existing TCP connections[3]. The experiments involved simulation of the 12-node network topology shown in Figure 1. 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 utilisation of the network is in all cases relatively high; otherwise, few packet drops occur and loss estimation is of little interest. All the TCP connections flowing from the sender to the receivers are used when collecting packet and packet-pair measurements (see [37] for details on the data collection process). Measurements were collected over a 300 second interval.

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 1-2 and 2-5 experience substantial losses, thereby testing the framework's ability to separate cascaded losses. In Scenario 2, links 1-2 and 4-8 experience substantial loss,

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

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

(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 3 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 [18, 15, 20]; estimation of low order moments such as link delay variance [32] from end-to-end delay variances which are additive over the probe paths; 3) estimation of the link delay cumulant generating functions (CGF) [20, 31] 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.

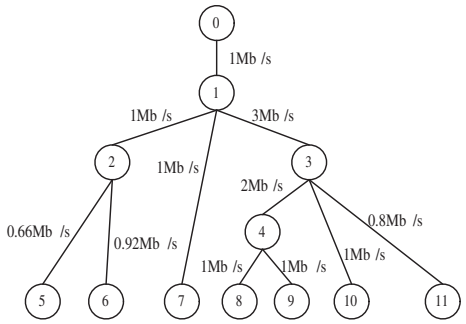
The CGF of a random variable Y is defined as $K_Y(t) = \log E[e^{tY}]$ where t is a variable which can be real or complex depending on the application. When Y is a sum of a set $\{X_j\}_{j=1}^{m}$ of statistically independent random variables the CGF satisfies the additive property $K_Y(t) = \sum_{j=1}^{m} K_{X_j}(t)$. 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

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

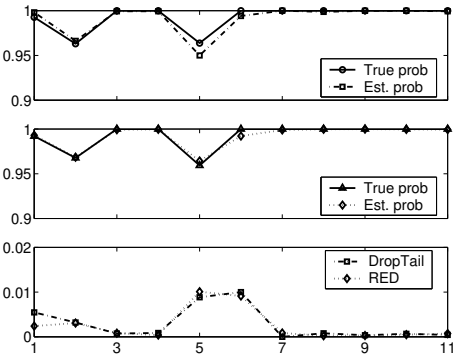
where $\mathbf{K}_Y(t) = [K_{Y_1}(t), \ldots, K_{Y_n}(t)]^T$ and $\mathbf{K}_X(t) = [K_{X_1}(t), \ldots, K_{X_m}(t)]^T$ are n-element and m-element vector functions of t, respectively.

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, $i = 1, \ldots, n$.

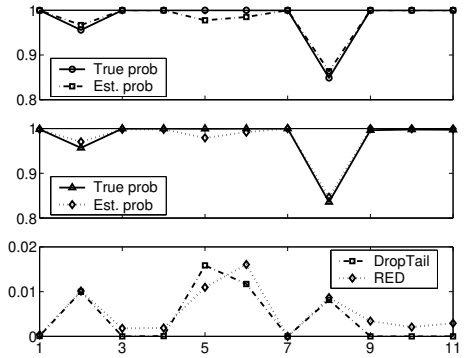
When A is *not full rank*, only linear combinations of those link CGFs lying outside of the null space of A can be determined from (9). We call such a linear combination an *identifiable subspace* of CGFs. Depending on the routing matrix A, identifiable subspaces can correspond to weighted averages of CGFs $\sum_{j=1}^{N} \alpha_j K_{X_j}(t)$ 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.



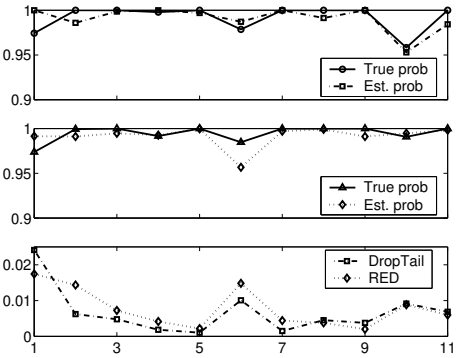
(a) A network consisting of a single sender (node 0), four internal routers (nodes 1-4), and seven receivers (nodes 5-11). The bandwidth in megabits/second (Mb/s) is indicated for each link.



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



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



(d) Scenario 3: Traffic mixture - medium losses

Figure 3: Performance of the link-level loss tomography framework examined through ns-2 simulation of the network in (a). Panels (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 two panels display for each link 1-11 (horizontal axis): (top) an example of true and estimated success rates and (bottom) mean absolute error between estimated and true success rates over 10 trials.

If one of the regions was identified as a potential site of anomalous behavior a similar probing process is 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):

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

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 by assuming different models for the statistical distribution of the error residual. One model, discussed in [20], 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 $\mathbf{K}_X$ is 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 4. All four links had `ns-2` bandwidth set equal to 4Mb/sec and latency set equal

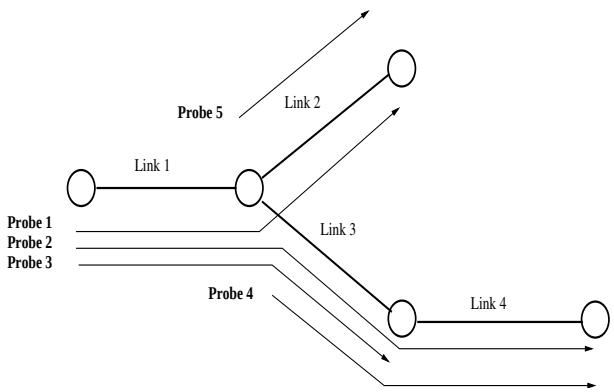


Figure 4: Unicast delay estimation probe routing used in `ns-2` simulation. Tailgating can be used to emulate the internal probes 3,4,5.

to 50ms. Each link was modeled as a Drop-Tail queue (FIFO queue with finite buffer) with queue buffer size of 50 packets. 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 20Kb/sec. The background traffic consisted of both Exponential on-off UDP traffic and FTP traffic. The background traffic on link 3 was approximatly 10 times higher than the background traffic on the other links to simulate a "bottleneck" link.

Table 1 shows the integrated mean squared error $\int_0^\infty |K_{X_j}(t) - \hat{K}_{X_j}(t)|^2 dt$ based on LS estimators with bootstrap bias correction (first row) and no bias correction (row 2).

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

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

We next illustrate the application of the CGF 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 Chernoff bound P_j on the bottleneck probability $P(X_j \geq \delta = 0.02s)$ (s denotes "seconds") which were estimated by plugging bias corrected CGF estimates into the right hand side of (11). Also shown are the "true" link exceedance probabilities which were empirically estimated from directly measured single-link loss statistics from an independent `ns-2` simulation. Note that while the Chernoff bounds are sometimes greater than one, the amplitude of the bound follows the trend of the true bottleneck probabilities. In particular if we set as our criterion for detection of a bottleneck as: "the probability that X_j exceeds $0.02s$ is at least 0.95", we see that the estimated Chernoff bound correctly identifies link 3 as the bottleneck link.

Table 2: Chernoff bound P_j and empirical estimate of $P(X_j \geq 0.02)$. A bottleneck at link 3 is correctly identified if one selects a 95 percentile bottleneck detection level ($P_j \geq 0.95$).

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

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]

[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.

(OSPF) routing protocol are related to the traffic on the paths. Ideally, a data-driven OD matrix should be central to the routing optimization program.

There are currently two ways to obtain OD traffic counts. Indirect methods are considered in [11, 22, 25, 23]; a direct method via software such as *NetFlow* supported by Cisco routers is described in [23, 40]. 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 $\boldsymbol{x}$); thirdly, even though $\boldsymbol{A}$ is singular as in other cases discussed, these techniques use statistical means to induce a regularization enabling the recovery of the whole $\boldsymbol{x}$ (or the traffic intensities underlying $\boldsymbol{x}$). Moreover, the most recent work [23] on this also deals with the time-varying aspect of the data.

Vardi was the first to investigate this problem. In [11] he studies a network with a general topology, using an iid 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 [25]. A Bayesian formulation and Markov Chain Monte Carlo estimation technique has also been proposed [22].

Cao *et al.* [23] 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 Traffic Control Protocol (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 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, can reduce packet forwarding efficiency, and involve a significant administrative burden when used on a large scale.

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 and let $\boldsymbol{x} = (x_1, \ldots, x_n)^T$ denote the *unobserved* vector of corresponding byte counts for all OD pairs 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 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

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} \tag{12}$$

where $\boldsymbol{A}$ is a $m \times n$ *routing matrix* of 0's and 1's that is determined by the routing scheme of the network. The work of [23] only considers *fixed routing*, i.e. there is only one route from an origin to a destination. In [23] the unobserved OD byte counts are modeled as

$$x_i \sim \text{normal}(\lambda_i, \phi\lambda_i), \text{ independently} \tag{13}$$

and this implies

$$\boldsymbol{y} \sim \text{normal}(\boldsymbol{A}\lambda,\ \boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{A}'), \tag{14}$$

where

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

Here $\lambda > \mathbf{0}$ s the vector of OD mean rates and c is a fixed power constant (both $c = 1$ and 2 work well with the Lucent network data as shown in [23, 24]). $\phi > 0$ 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.* [23] address the non-stationarity in the data using a local likelihood model (cf. [41]); 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 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 iid assumption is imposed (as a simplified and yet practical way to treat the approximately stationary observations within the window). And 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 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. [42, 43]) 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 iid statistical model. To smooth the parameter estimates, a random walk model is applied to the logarithm of the parameters λ's and ϕ over the time windows.

4.1 Example: Time-varying OD Traffic Matrix Estimation

Figure 5 is a network at Lucent Technologies considered in [23, 24]. Figures 6 and 7 are taken from [23]. They show the validation (via *NetFlow*) and estimated OD traffic based on the link traffic for the subnetwork around Router 1 with 4 origins/destinations in Figure 5. Figure 6 gives the full scale and Figure 7 is the zoomed-in scale ($20\times$). It is obvious that the estimated

OD traffic 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 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.

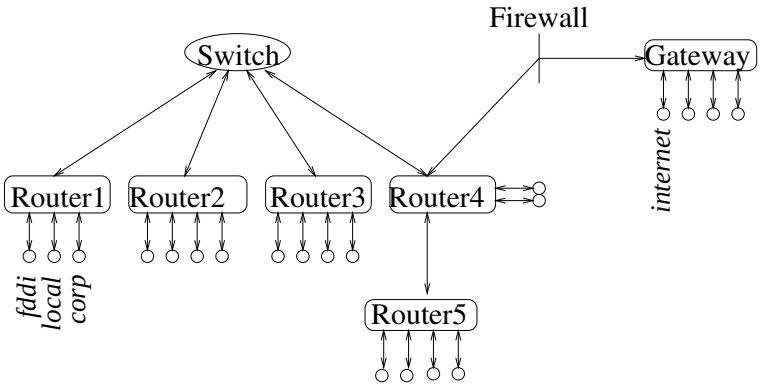


Figure 5: A network at Lucent Technologies

Even though the method described in [23] 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 [24].

5 Conclusion and Future Directions

This paper has provided an overview of the emerging area of large scale inference and tomography in communications networks. 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 model has 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 [20] and multicast [13] 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, nonsta-

tionary 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 [36]. Sequential Monte Carlo inference techniques are employed in [36] 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 8 illustrates this scenario and displays the estimated delay distributions at different time instances (see [36] for further details).

There is also an accelerating trend toward network security 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 [37]. 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 [31, 21] 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 service.

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?

References

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

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

[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] 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.

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

[7] 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.

[8] 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.

[9] 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.

[10] 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.

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

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

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

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

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

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

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

[29] R. Cáceres, N. Duffield, J. Horowitz, F. Lo Presti, and D. Towsley. Loss-based inference of multicast network topology. In *Proc. IEEE Conf. Decision and Control*, Dec. 1999.

[30] 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.

[31] 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.

[32] 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.

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

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

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

[36] M. Coates and R. Nowak. Sequential Monte Carlo inference of internal delays in non-stationary communication networks. to appear in *IEEE Trans. Signal Processing, Special Issue on Monte Carlo Methods for Statistical Signal Processing*, 2001.

[37] 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.

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

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

[40] 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.

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

[42] W. E. Deming and F. F. Stephen. On a least squares adjustment of a sampled frequency table when the expected marginal totals are known. *Ann. Math. Statist.*, 11:427–444.

[43] I. Csiszár. I-divergence geometry of probability distributions and minimization problems. *Ann. Probab.*, 3(1):146–158, 1975.

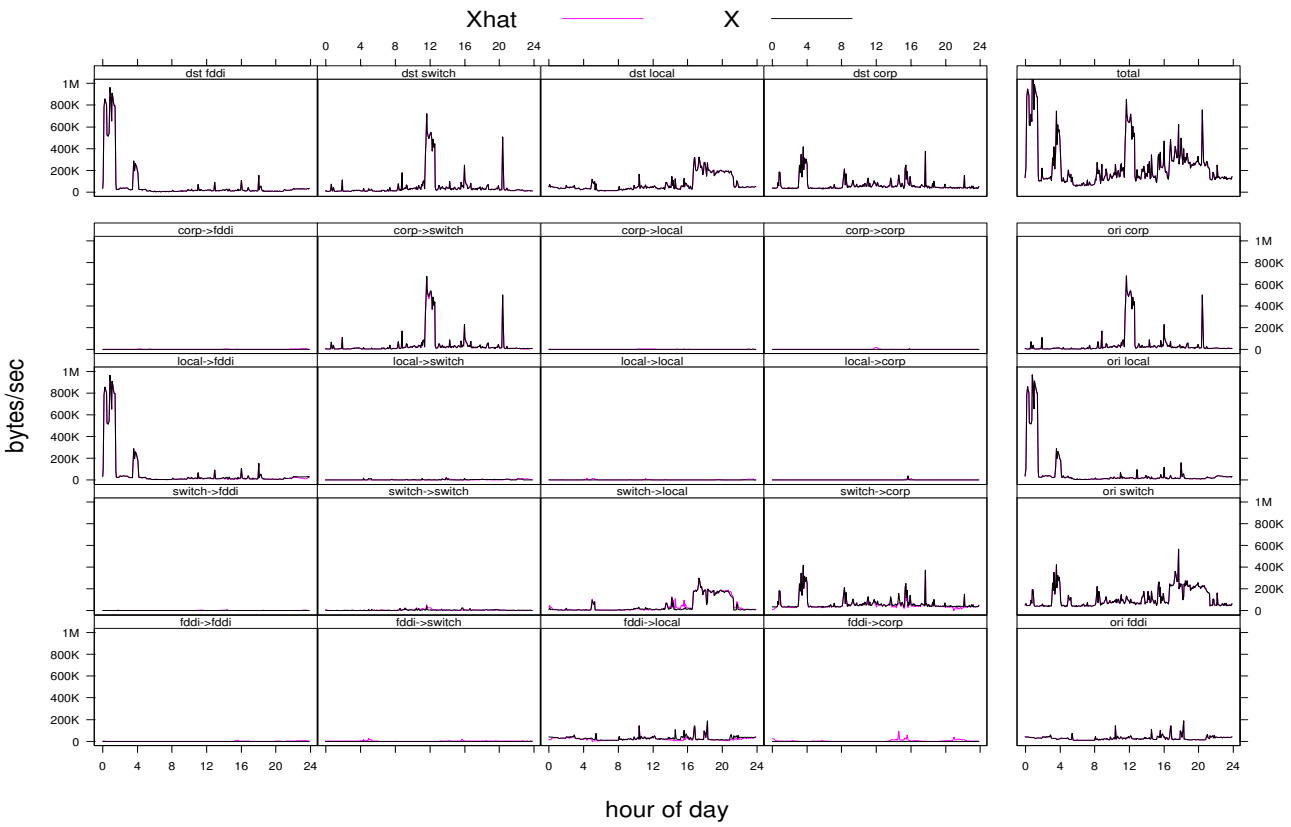


Figure 6: Full-scale time series plots of OD traffic on Feb. 22, 1999 for Router 1 sub-network with 4 origins/destinations. In the lower-left 4×4 matrix, the rows (from TOP down) correspond to corp, local, switch and fddi and the columns (from RIGHT to LEFT) correspond to corp, local, switch and fddi. 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.

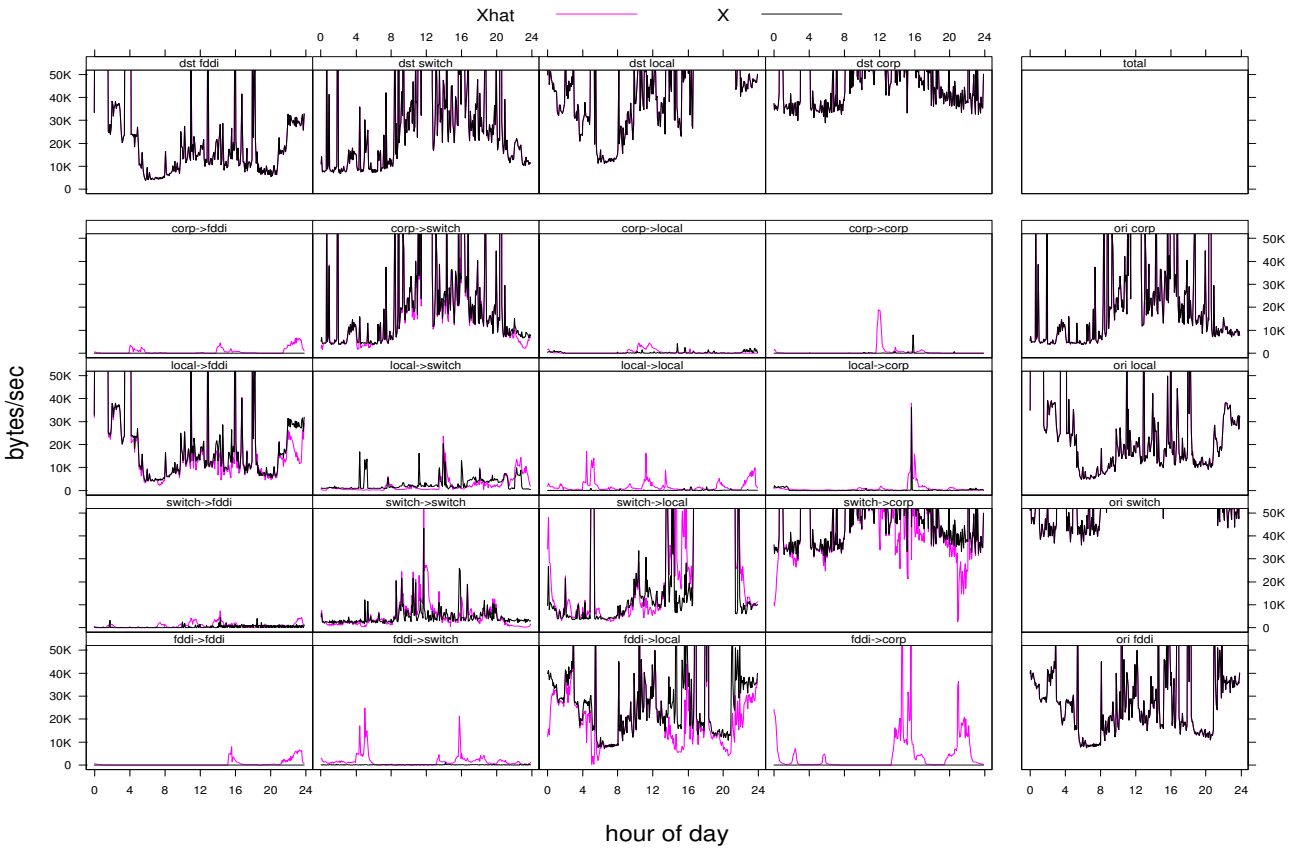
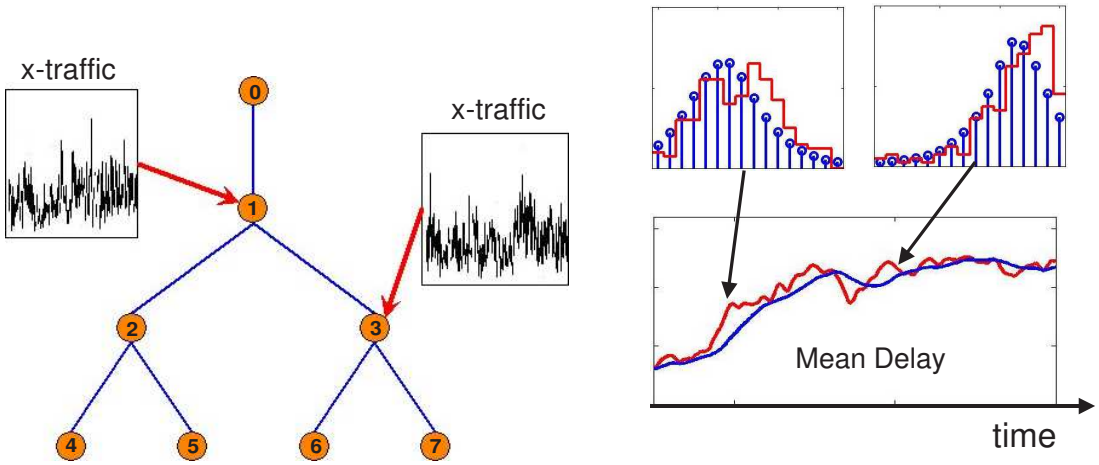


Figure 7: Zoomed-in time series plots of OD traffic on Feb. 22, 1999 for Router 1 sub-network with 4 origins/destinations. In the lower-left 4×4matrix, the rows (from TOP down) correspond to corp, local, switch and fddi and the columns (from RIGHT to LEFT) correspond to corp, local, switch and fddi. 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 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 8: 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.