MSMD Network Tomography Based on Metrics

Hong Fei¹, Bai Yu² and Zhao Zhuang³

¹School of Computer Science, BeiHang University
No.37, XueYuan Road, HaiDian District, Beijing, P.R.China (100083)
hongfei@buaa.edu.cn

²School of Mathematical Sciences, Peking University
HaiDian District, Beijing, P.R.China (100871),
pkuby@math.pku.edu.cn

³College of Software Engineering, Graduate School of the CAS
No.19, Yu Quan Road, Beijing China, (100049)
zhaozh@gscas.ac.cn

Abstract: Today’s Internet is a massive, distributed network which continues to explode in size as e-commerce and related activities grow. 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. Recently, researchers present a collaborative framework for performing network tomography on topologies with multiple sources and multiple destinations (MSMD), without assuming the topology to be known. With introduction of useful metrics, we give a statistical test which quantifies the tradeoff between network topology complexity and performance estimation, and identifies when measurements made by the two sources can be combined to achieve reduced variance performance estimates. The performance and efficacy of the algorithm are assessed through ns-2 simulations.

Keywords: Network tomography, Metrics, Topology identification.

1. INTRODUCTION

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

Usually we cannot directly measure the aspects of the system that we need in order to make informed decisions. However, we can frequently make useful measurements that do not require special cooperation from internal network devices and do not inordinately impact network load. Sophisticated methods of active network probing or passive traffic monitoring can generate network statistics that indirectly relate to the performance measures we require. Subsequently, we can apply inference techniques, derived in the context of other statistical inverse problems, to extract the hidden information of interest. Firstly, there gives surveys on the field of inferential network monitoring or network tomography, highlighting challenges and open problems, and identifying key issues that must be addressed. It builds upon the signal processing survey paper [1] and focuses on recent developments in the field[2]. The task of inferential network monitoring demands the estimation of a potentially very large number of spatially distributed parameters. To successfully address such large-scale estimation tasks, researchers adopt models that are as simple as possible but do not introduce significant estimation error. Such models are not suitable for intricate analysis of network queuing dynamics and fine time-scale traffic behavior, but they are often sufficient for inference of performance characteristics. The approach shifts the focus from detailed queuing analysis and traffic modeling [3] to careful design of measurement techniques and large-scale inference strategies.

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 [5] 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 [6, 7, 8, 9, 10, 11, 12, 13, 14, 15] and ii) sender-receiver path-level traffic intensity estimation based on link-level traffic measurements [5, 16, 17, 18, 19, 20].

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:

$$Y_t = AX_t + \epsilon$$

where: $Y_t$ is a vector of measurements, e.g., packet counts or end-to-end delays, recorded at a given time $t$ at a number of different measurement sites, $A$ is a routing matrix, $\epsilon$ is a noise vector, and $X_t$ is a vector of time-dependent packet parameters, e.g. mean delays, logarithms of packet transmission probabilities over a link, or the random origin-destination traffic vector. Typically, but not always, $A$ is a binary matrix (the $i,j$th element is equal to
This paper mainly based on the study of the multiple source, multiple destination network tomography problem [21]. Using multiple sources in the context of network tomography, it is possible to identify segments within a network shared by the paths connecting multiple sources and destinations. This information may be useful for identifying potential bottlenecks. Sharing statistics between sources may also be useful for optimizing the use of network resources when transferring large amounts of data. Additionally, in some cases it is possible to fuse information gleaned from multiple sources to get a more accurate and refined network characterization.

The majority of work in network tomography has revolved on active probing from a single source. Also, it is typical to focus on either (step 1) identifying the topology, or (step 2) estimating link-level performance parameters in which case it is assumed that the topology is known. This paper presents a multiple source active measurement procedure and a statistical framework enabling the joint characterization of topology and link-level performance. Jointly solving for performance parameters and topology leverages on the close coupling between link-level characteristics, routes derived from the network topology, and end-to-end measurements.

The remainder of the paper is organized as follows. Section 2 describes the useful metrics. These metrics are the foundation for the novel MSMD algorithm aimed at characterizing the multiple source topology. Then a statistical framework allowing the joint characterization of topology and performance is presented in Section 3. Section 4 gives results from simulation and experiments on NS-2 and we conclude in Section 5.

2. METHODOLOGY

To facilitate the development of statistical methods geared at analyzing data from networks, it is important to have metrics which define and measure distances between a network’s links, its paths, and also between different networks [22].

Consider a strongly connected directed network, where there exists a directed path between any pair of nodes in the network. An edge between two immediately connected nodes is referred to as a "link", $L(i)$, and is indexed by $i = (i \rightarrow i')$, $i = 1,...,I$. A set of connected links leading from a source-node $j_1$ to a destination-node $j_2$ is referred to as a "path" or "trail", $T(j)$, with $j = (j_1, j_2)$, $j = 1,...,J$, indexing the source-destination (SD) pairs.

Consider now a fixed-routing protocol: for each SD pair $j$, the same unique path, $T(j)$, always carries the traffic from $j_1$ to $j_2$. The routing protocol is described by a “routing matrix” $A$, with entries $a_{i,j} = 1$ if $L(i)$ is in $T(j)$ and $a_{i,j} = 0$ if $L(i)$ is not in $T(j)$, $i = 1,...,I$, $j = 1,...,J$. We now define distance functions for links and paths of such networks.

**Distance function for links:**

...
\[ d_L(i, i') \equiv d(L(i), L(i')) \equiv \sum_j a_{i,j} + \sum_j a_{i',j} - 2\sum_j a_{i,j}a_{i',j} \quad i \text{ and } i' \text{ in } \{1, \ldots, I\} \]

**Distance function for paths:**
\[ d_T(j, j') \equiv d(T(j), T(j')) \equiv \sum_i a_{i,j} + \sum_i a_{i,j'} - 2\sum_i a_{i,j}a_{i,j'} \quad j \text{ and } j' \text{ in } \{1, \ldots, J\} \]

**Lemma:** The distance functions above satisfy non-negativity, symmetry, and the triangle inequality.

**Comments, interpretations, and properties:**

1) Denote: \( W(i) = \{ \text{all paths passing through } L(i) \} \), \( W(i, i') = W(i) \cap W(i') \), and let \( w(i) \) and \( w(i, i') \), respectively, be the cardinality of these sets. Then:
\[ d_L(i, i') = w(i) + w(i') - 2w(i, i') = w(i) + w(i', i') - 2w(i, i') \]

Similarly, denote: \( M(j) = \{ \text{all links in the path } T(j) \} \), \( M(j, j') = M(j) \cap M(j') \), and let \( m(j) \) and \( m(j, j') \), respectively, be the cardinality of these sets. Then
\[ d_T(j, j') = m(j) + m(j') - 2m(j, j') = m(j) + m(j', j') - 2m(j, j') \]

2) Note:
\[ \sum_i w(i) = \sum_j m(j) = \sum_{i,j} a_{i,j} \]

3) Imagine that each link has a dedicated ‘lane’ for each SD path passing through it. Then \( w(i) \) is the number of lanes in \( L(i) \), and can be thought of as the “width” (“band-width”, “capacity”, etc) of \( L(i) \). Since \( w(i, i') \) is a count of the lanes that are common to both link \( L(i) \) and \( L(i') \), it can be thought of as their common width. With similar interpretation, think of \( m(j, j') \) as the common length of \( T(j) \) and \( T(j') \). Using this terminology we note that the distance between two links/paths is the sum of their norms (width/length) minus twice their common norm (width/length), respectively.

4) When two links/paths are close in distance, they likely share many common paths/links, respectively, and therefore will likely fail or operate flawlessly together. For instance, \( d_L(i, i') = 0 \) holds if, and only if, the traffic-flows on \( L(i) \) and \( L(i') \) are identical. Thus, the proposed metrics are akin to measures of pair-wise statistical dependency on the network’s components.

**Measuring distance between topologically similar networks:**

It is natural to compare traffic flows on topologically similar networks by comparing their routing matrices. Toward this end, for two networks with the same number of nodes (and hence pairs of nodes) and links, but possibly different routing matrices, say \( A \) and \( A' \) respectively, let \( B = \{ \text{all } I \times J \text{ matrices } B \text{ derived from } A' \text{ by row- and column } - \)
permutations}. Define:

\[ d_x(A, A') \equiv \min \{ \| A - B \| : B \text{ in } B \} \]

= \text{distance between } A \text{ (strung out as } \alpha \text{ vector) and the set } B .

**Lemma:** \( \arg\min \{ \| A - B \| : B \text{ in } B \} = \arg\max \{ \sum_{ij} a_{ij} b_{ij} : B \text{ in } B \} . \)

**Note:** \( d_x(A, A') \) is invariant under rows and columns permutations, so when two networks differ only in labels of links or nodes but otherwise have the same flow pattern, the distance between them is zero.

**Rescaling:**

It might be desirable in certain applications to ‘rescale’ a distance function so it lies in a predetermined finite interval. This could be useful, for instance, if one is to bring into a common denominator the comparisons of networks of significantly different dimensions across different applications.

### 3. MSMD IN METRICS

This section describes the multiple source measurement algorithm developed in this work. Measurements are developed to exploit differences between the shared and non-shared topologies. Sources transmit packets in a semi-randomized fashion, and destinations record packet arrival order. Because neither of these operations require precise time synchronization between any of the participating hosts the algorithm is easy and practical to implement. Algorithm in detail refer to [21].

Initially, to facilitate in explaining the algorithm we make the following idealistic assumptions:

1) There is no cross-traffic in the network so that there is no variability in delay along any link,
2) Sources are precisely synchronized,
3) Routes between end-hosts are unique, and
4) Packets do not get reordered within the network.

The framework is flexible, taking as inputs either arrival order measurements, delay variance measurements, loss measurements, or any combination thereof. When multiple sets of measurements are used (e.g. arrival order and loss), the test jointly solves for the topology characterization and performance estimates. Due to constraints on the length of this paper, we limit our discussion to the case where arrival order measurements and loss measurements are both used. For a complete outline of the framework please see [21].

Suppose the sources send \( N \) probes. Each destination keeps track of packet arrival order and loss. Let \( z \) denote the set of arrival order measurements and let \( y \) denote the set of loss measurements for an experiment. Denote by \( \theta_1, \ldots, \theta_6 \) the link-level loss rates, corresponding to links as depicted in the two 1-by-2 networks in Figure 1.
Figure 1 Two 1-by-2 components which comprise the 2-by-2 problem.

Let $H_S$ denote the hypothesis that the 2-by-2 topology is shared, and let $H_N$ denote the hypothesis that the topology is not shared. Let $\theta = \{\theta_1, \ldots, \theta_6\}$ denote the general six-dimensional vector of loss rates, and let $\rho = (\rho_1, \rho_2, \rho_3)$ denote the three-dimensional vector of different arrival order probabilities.

Under each hypothesis the joint likelihood function is written as $p(y, z|H_i, \theta, \rho)$. A decision is made by choosing the hypothesis which maximizes the likelihood given the observations. We factor the likelihood function into

$$p(y, z|H_i, \theta, \rho) = p(y|H_i, \theta) p(z|H_i, \rho)$$

Implying that the loss measurements and arrival order measurement are statistically independent. Independence follows from the assumption that the inter packet-pair spacing, $\Delta$, is large enough that queuing effects experienced by the first and second back-to-back packet probes sent from each source are independent.

Now, the true parameters $\rho, \theta$ are unknown variables. We take the Generalized Likelihood Ratio Test (GLRT) approach to solving this composite hypothesis problem. In the GLRT, the unknown distribution parameters $\theta$ and $\rho$ are replaced with their maximum likelihood estimates under each model. Under $H_N$, we have $\theta \in [0,1]^6$ and $\rho \in [0,1]^3$. On the other hand, under $H_S$, the model order of the network is reduced. Consequently, the parameter space is restricted so that $\theta_2 = \theta_3$ and $\theta_4 = \theta_5$. Thus, under $H_S$, with metric introduced last section, The GLRT can be written as:
In general, setting a threshold for the GLRT is a difficult task when no uniformly most powerful test exists and when a priori probabilities are not available for each hypothesis. However, for the composite hypothesis test as formed above, a threshold can be set using Wilks’ Theorem for the asymptotic behavior of the log likelihood ratio statistic. In general, it is difficult to precisely quantify the error rate when the true topology is not shared (Type II error) because it depends on the magnitude of diversity between the delay differences on links, and these are parameters we do not know. In this section we offer an intuitive explanation of the characteristics of Internet topologies and traffic which will affect performance in the non-shared scenario. In the next section we further evaluate the performance through simulation.

We begin by relating the problem when the true topology is shared to the classic signal-in-noise detection problem. We would like to decide whether or not the topology is not-shared given a set of noisy measurements. The signal is the “bump” region of offsets between $d_1$ and $d_2$ where different arrival order events are observed. Noise takes the form of queuing due to cross-traffic which both cause different arrival order events and same arrival can order events where they would otherwise not occur. In such a problem, the error rate is usually parameterized by a signal-to-noise ratio, with performance improving as this ratio increases.

4. SIMULATION RESULTS

Next, we evaluate our multiple source algorithm using the ns-2 simulator [23]. Both loss and arrival order measurements were used in the simulation. Packet delays and losses are due to congestion as probes compete with cross-traffic. Infinite TCP flows produce the majority of the background traffic, as TCP is the dominant transport protocol on the Internet. A few exponential on-off flows are also included, with the over all mix of background traffic such that link-level loss rates vary between 0.01% and 2%. Probes in the simulation are composed of multicast packets.

The simulated topology is depicted in Figure 2. Note the flow of probe traffic on one link. The 2-by-2 networks for destination pairs (S,R) are shared, and those for all other pairs of destinations are non-shared. The simulation was repeated 500 times, with different random seeds. Each trial consists of 1000 probes transmitted over 200 simulated seconds. All settings were chosen to reflect a realistic scenario.
Next we assess the performance of our algorithm. Figure 3 shows a plot of the Type I error rate versus one minus the Type II error rate. This type of plot is sometimes referred to as Receiver-Operator Characteristics, or ROC curves. Note that the origin is in the upper left-hand corner of the figure, and that the indices on each axis range between 0 and 0.5. Three curves are shown for the cases when only arrival order measurements are used, only loss measurements are used, and both arrival order and loss measurements are used. In order to set a threshold for the statistical test, we choose the Type I error rate (along the x-axis). The resulting Type II error rate is depicted along the y-axis. Ideally, we would like these curves to go through the top left corner, where there is no error of either type. Note that the detector using combined arrival order and loss measurements outperforms both the loss only and arrival order only detectors.

Figure 3 Type II versus Type I error rates

Type II versus Type I error rates for detectors using both loss and arrival order measurements, only loss measurements, and only arrival order measurements. We choose a value for the Type I error in order to set the threshold, and the resulting Type II error is depicted along the y-axis. The joint detector (using arrival order and loss measurements) exhibits the best performance.
Each of the curves depicted in Figure 3 show results for when 1000 probes are used. Next, we analyze the performance by varying the number of measurements fed in to the algorithm. Fig 4. depicts the ROC curve for the joint detector, varying the number of probes used by the algorithm. As expected, the Type II error rate decreases quickly as the number of probes increases. When 1000 probes are used, it is possible to achieve a Type II error rate as low as 10% with the Type I error rate at 5%. Thus, it is possible to achieve desirable performance using a moderate number of probes.

![ROC curve for the joint detector, varying the number of probes used by the algorithm.](image)

Figure 4 ROC curve for the joint detector, varying the number of probes used by the algorithm.

5. CONCLUSION AND DISCUSSION

Networks are complicated mathematical structures. They are abundant around us: social, behavioral, traffic, communication, information, and data networks are some of the examples where network structures and data are common. Many tools typically depend on having a metric which quantifies notions of “similarity”, “closeness”, and facilitates ordered-comparisons. The metric is a ground-level component of any statistical study, as it determines the statistical properties (robustness, efficiency, etc.) of the analysis. With this in mind, we introduce metrics into MSMD network tomography. The performance and efficacy of the algorithm are assessed through ns-2 simulations.

REFERENCES


