Multiple Source Internet Tomography

Mark J. Coates, Michael G. Rabbat, and Robert D. Nowak

***Abstract*— Information about the topology and link-level characteristics of a network is critical for many applications including network diagnostics and management. However, this information is not always directly accessible; subnetworks may not cooperate in releasing information and widespread local measurement can be prohibitively expensive. Network tomographic techniques obviate the need for network cooperation, but the majority assume probing from a single source, which imposes scalability limitations because sampling traffic is concentrated on network links close to the source. We describe a multiple source, end-to-end sampling architecture that uses coordinated transmission of carefully engineered multi-packet probes to jointly infer logical topology and estimate link-level performance characteristics. We commence by demonstrating that the general multiple source, multiple destination tomography problem can be formally reduced to the two source, two destination case, allowing the immediate generalization of any sampling techniques developed for the simpler, smaller scenario. We then describe a method for testing whether links are shared in the topologies perceived by individual sources, and describe how to fuse the measurements in the shared case to generate more accurate estimates of the link-level performance statistics.**

***Index Terms*— Internet tomography, end-to-end measurements, active probing, topology discovery, loss rate estimation.**

I. Introduction

Efficient and effective sampling of a network plays a vital role in network performance monitoring, providing information that can be used to improve the performance of overlay network applications and security systems. Network sampling is not limited to the periodic measurement of traffic loads, flow characteristics, link losses or delays. Richer information may be derived by carefully engineering end-to-end multi-packet probes, coordinating their transmission from multiple sources in the network, and measuring the order of packet arrivals or delay differences. In this paper, we address spatially-distributed end-to-end sampling; we develop probing methodologies and describe methods for inferring topological information and link-level performance statistics.

A robust monitoring architecture must operate over networks consisting of multiple domains and transparent or uncooperative switching elements; it must negotiate the challenge of restricted access to portions of the network. End-to-end sampling, coupled with statistical inference to form tomographic techniques, becomes attractive because it avoids reliance on network cooperation. The architecture can provide topological information to supplement that provided by more direct approaches, and can generate estimates of link-level performance metrics, offering an efficient alternative to link-level measurement. In order to be scalable to larger networks, the architecture must not overload the network with probes nor concentrate its traffic on any one link or region. This is one of the main shortcomings of many of the previously proposed link-level tomographic techniques (see the summaries in [3], [4]); network sampling is performed from a single source to multiple destinations, so all probing traffic flows across the egress link from the sampling source. The obvious way to distribute the sampling load more evenly is to perform probing from multiple sources. Although this may at first seem like a simple extension of single source sampling, the challenge of coordinating spatially distributed sampling and fusing the information is far from trivial. New sampling strategies are required to determine how the topologies perceived by the different sources overlap and which measurements can be combined.

In this paper, we describe a multiple source end-to-end sampling architecture that addresses the joint problem of monitoring link-level performance and identifying the *logical* topology of generalized networks of M sources transmitting to N receivers (an M-by-N network). We require that the subnetwork between any given source and the receivers forms a tree, and we call the combined network a “multiple-tree” network. We restrict our attention to identification of the logical topology (specified by the branching and joining points in the network) because the end-to-end measurements do not provide sufficient information to identify the physical topology. We address the important question of how to determine which sampled measurements from different sources can be fused to infer the performance characteristics of shared links with greater accuracy. Throughout, we strive to develop a robust architecture that does not rely on unrealistic assumptions or conditions that are difficult to achieve (such as precise synchronization).

A. Related Work and Contributions

In contrast to most previous work, the multiple source sampling architecture we propose strives to jointly identify logical topology and estimate link-level characteristics. In this section we briefly review previous approaches to the individual taks. We indicate the unique aspects of our methodology, but also highlight how it can act in a complementary role to some existing techniques. We also discuss the relationship between our architecture and methods for inferring shared bottleneck links.

Topology identification techniques fall into three broad categories: (i) `traceroute`-based approaches that identify

This research was supported by the NSERC AAPN Research Network, and by Applied Signal Technology, Inc.

Portions of this work were presented at the ACM Internet Measurement Conference, 2003 [1], and at IEEE Infocom, 2004 [2].

M.J. Coates is with the Department of Electrical and Computer Engineering, McGill University, 3480 University St, Montreal, QC, Canada H3A 2A7. Email coates@ece.mcgill.ca

M.G. Rabbat and R.D. Nowak are with the Department of Electrical and Computer Engineering, University of Wisconsin, 1415 Engineering Drive Madison, WI 53706. Email rabbat@cae.wisc.edu, nowak@engr.wisc.edu

the network layer (ISO layer 3) topology [5]–[13]; (ii) layer-2 topology identification approches based on SNMP MIB (Simple Network Management Protocol Management Information Base) information [14]–[17]; and (iii) tomographic techniques that identify the logical topology [18]–[27]. The major challenges are how to probe the network efficiently, how to combine (potentially inconsistent) information from different measurements, and how to address unresponsive or anonymous routers. The limitations of the `traceroute`- and SNMP-based approaches are that they rely on substantial cooperation from network elements; the `traceroute`-based approaches also fail to capture the complex interconnections between layer-2 network elements in Ethernet LANs and ATM networks. The tomographic techniques are limited in that they can only identify a logical topology and they are less robust, but often they can perform a complementary role to the more direct techniques, filling in missing information. Our approach is tomographic in nature; it extends prior work in that it identifies a multiple source logical topology.

The articles [4], [28] provide an overview of tomographic approaches for inferring link-level performance metrics. In the field of link-level tomography, the most closely related work to that described here is the multiple-source network tomography for link-level performance metric inference proposed by Bu et al. [29]. In contrast to our technique, this method requires a known topology and does not exploit any additional information which can be obtained if sources probe cooperatively.

Other techniques that involve multiple source sampling focus on the problem of identifying shared bottleneck links. The method of Harfoush et al. [30] addresses only the case of a single source and two receivers (the *inverted Y*-topology) and determines whether losses occur predominantly on the shared portion of paths to two different receivers. Rubenstein et al. address both this case and the case where there are two sources and a single destination (the *Y*-topology) [31]. Katabi et al. [32] describe a similar approach based using existing traffic between multiple sources and single destination, assuming a known tree topology. More recently, Cui et al. have addressed the same problem but also considered the two-source two-receiver network (assuming that there is a common branching point). These techniques address a different problem to that addressed by our methods; they do not attempt to identify topology nor do they strive to generate estimates of link-level performance metrics (although it is an ingredient in [30]). Our architecture strives to identify shared links (be they bottleneck or not) and this identification is formulated as a hypothesis test (or detection problem), similar in nature to the detection formulations in [31], [33].

This paper unifies and extends techniques that we have described previously [1], [2]. In [1], we described a method for merging two known single-source tree topologies into a single multiple-tree network. The method combines a simple, robust multiple source probing method and hypothesis tests based solely on packet order-of-arrival information garnered from the probes. In [2], we improved upon this work by establishing the equivalence between the hypothesis test and a model-order identification problem. We also performed a more complete analysis of the timing and synchronization requirements of the probing methodology and described how multiple-source trees could be used to perform more efficient tomographic network monitoring.

There are three major contributions of the paper. The first is a proof of the equivalence of inference based on measurements over an M-by-N multiple-tree network and inference based on measurements over the set of the component 2-by-2 networks. This equivalence is important because it indicates that sampling techniques can be developed for the simpler 2-by-2 case and they can be immediately generalized to M-by-N multiple-tree networks. The second contribution addresses the scalability of our multiple source sampling architecture: we describe how to incorporate the packet stripe approach of Duffield et al. [19] to probe larger networks, reducing the probing complexity from $O(M^2N^2)$ to $O(M^2N)$. In the third contribution, we establish a test for the *identifiability* of an M-by-N network (whether the logical topology can be identified from the available ordering and metric measurements).

B. Structure of the Paper

The rest of the paper is organized as follows. In Section II we prove that any M-by-N multiple tree network can be accurately described in terms of all 2-by-2 subnetwork components, thereby reducing the general M-by-N network tomography to a collection of 2-by-2 subproblems. Then Section III describes and analyzes an architecture for active sampling on 2-by-2 components. We show that precise time synchronization is not necessary and describe an extension of the scheme for efficiently probing more than two destinations simultaneously to improve scalability. Section IV formally develops a statistical procedure for simultaneously inferring link-level performance parameters and characterizing network topology based on measurements from the multiple source sampling architecture. This procedure is flexible, allowing for various types of performance metrics to be easily incorporated. Section V then presents an algorithm for merging two single-source topologies using 2-by-2 topological characterizations produced by the statistical test. This section also establishes a test for uniqueness of the resulting multiple-source topology. Finally, in Section VI we present simulation results, and we conclude in Section VII.

II. Components of General Networks

In this section we demonstrate that under general assumptions about routing behavior, any M-by-N network can be equivalently described in terms of a collection of *2-by-2 component networks*. This reduction allows us to focus our analysis in following sections to 2-by-2 components.

A natural way to represent an M-by-N network component is to use a graph to describe the network topology. In a sense, end-to-end measurements provide information about paths through the network. Special structure in the probes transmitted in single-source tomographic techniques induces correlations between the measurements observed at different destinations [3]. This correlation allows one to identify how much paths from the source to the destinations overlap. Network structure inferred in this manner is generally referred

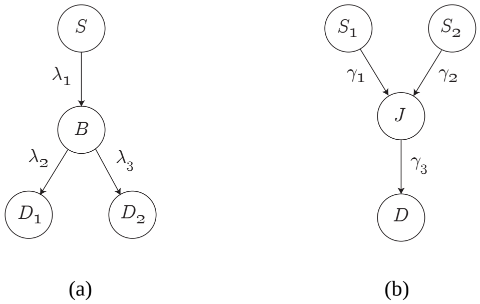


Fig. 1. The (a) “inverted-Y” and (b) “Y” topologies which we assume to be characteristic of all 1-by-2 and 2-by-1 components. Variables λ_i and γ_i correspond to a performance measure on each link (e.g., delay variance). Nodes B and J indicate where two paths branch or join.

to as a *logical routing topology*; i.e., nodes in the inferred network correspond to sources, destinations, or routers in the physical network where two or more paths join or branch. This is in contrast to a *physical routing topology*, such as one discovered using SNMP techniques or `traceroute`, which contains a node for every router encountered along the path between each source and destination, regardless of whether paths join or branch there.

Typically, single-source topology identification schemes assume the underlying logical topology is a tree rooted at the source, with each receiver being a leaf node. Generalizing this concept, we define an M-by-N logical routing topology (i.e., a multiple tree topology) as a directed acyclic graph along with a function which maps each source-destination pair to the route from the source to the destination. We make the following assumptions on routing behavior.

A1 There is a unique path from each source to each destination.

A2 Two paths from the same source to different receivers take the same route until they branch, so that all 1-by-2 components have the “inverted Y” structure depicted in Figure 1(a).

A3 Two paths from different sources to the same receiver use exactly the same set of links after they join, so that all 2-by-1 components have the “Y” structure depicted in Figure 1(b).

These assumptions are motivated by the shortest-path nature of routing in the Internet, where the next hop taken by a packet is determined according to a routing table lookup on the destination address. Together, they imply that internal nodes in the inferred network (i.e., nodes which are not sources or receivers) have degree at least three, and both the in-degree and out-degree are at least one. This is typical characteristic of logical topologies. When certain types of load-balancing are used in the network, A1-A3 may be violated. We elaborate more on this situation later.

Under the assumptions above, we more formally define our notion of an M-by-N network.

Definition 1 (M-by-N Network Component): The portion of a network connecting M sources to N destinations is described in terms of paths between each source and destination, along with a performance function defined on all subsets of these paths. Given a set of sources $\mathcal{S}$ and a set of destinations $\mathcal{D}$ with $M = |\mathcal{S}|$ and $N = |\mathcal{D}|$, an M-by-N network component is characterized by the pair $(\mathcal{P}, \theta)$, where θ is a real-valued function defined on portions of paths through the network, and $\mathcal{P}[a, b]$ denotes the path from a to its descendent b. The set of all possible M-by-N network components is denoted by $\mathcal{G}_{M,N}$.

In what follows, we use the notation $B(S_1; D_1, D_2)$ to refer to the internal node where paths from S_1 to D_1 and to D_2 branch. Similarly, $J(S_1, S_2; D_1)$ denotes the internal node where the paths from S_1 and from S_2 to D_1 join.

Notice that the assumptions above imply that each internal node is a joining point or a branching point. Therefore all internal nodes can be enumerated via the functions B and J. As an aside, note that B and J are symmetric functions in the sense that $B(S_1; D_1, D_2) = B(S_1; D_2, D_1)$ and $J(S_1, S_2; D_1) = J(S_2, S_1; D_1)$. Also note that link-level performance parameters are deterministic quantities which typically govern the distribution of some other performance-related quantity. In the following sections we will be estimating these parameters from random (noisy) measurements. We assume the performance parameters obey the following properties on individual links and over portions of a path.

A4 Either $\theta(i) \geq 0$ for all links i, or $\theta(i) \leq 0$ for all i.

A5 Furthermore, suppose that links $i_1, \ldots, i_n$ are a (not necessarily contiguous) subset of a path through the network. Then the performance measure across this entire portion of the path, $\theta(\{i_1, \ldots, i_n\})$, is related to the link-level performance values by $\theta(\{i_1, \ldots, i_n\}) = \sum_{j=1}^{n} \theta(i_j)$.

These two properties are equivalent to the monotonicity and separability properties assumed in [23]. Many of the performance measures we are interested in obey A4 and A5. For example, delay variance is a non-negative quantity so A4 is satisfied, and assuming that queueing events on different links are independent, delay variances add up along a path. Packet drop probabilities can also be handled by working with the logarithm of the success probability as a surrogate. This quantity is strictly negative (the success probability lies in $(0, 1]$) and under the same independence assumption, log success probabilities also add up along paths.

The five assumptions together define the class of multiple-tree topologies. Describing and analyzing measurement schemes on a general multiple-tree topology can be cumbersome. Instead, we prefer to analyze a fundamental building block of any M-by-N network component and then extend results for the smaller component to the general case. The main result of this section establishes that any M-by-N network can be decomposed into 1-by-2 and 2-by-1 components in such a way that no information about the network is lost. According to assumption A2 any 1-by-2 component is completely characterized by the three performance parameters, λ_1, λ_2, and λ_3 as depicted in Figure 1(a). We will use the triple $(\lambda_1, \lambda_2, \lambda_3)$ to denote a 1-by-2 component. Similarly, based on A3, let $(\gamma_1, \gamma_2, \gamma_3)$ denote the parameters of a 2-by-1 component. Also, let $C^{S}_{D_1,D_2} \in \mathcal{G}_{1,2}$ denote the 1-by-2 component from S to D_1 and D_2, and let $C^{S_1,S_2}_{D} \in \mathcal{G}_{2,1}$ denote the 2-by-1 component from S_1 and S_2 to D.

In addition to decomposing an M-by-N network component into 1-by-2 and 2-by-1 components, we want to show that the collection of 1-by-2 and 2-by-1 components can be used to reconstruct the M-by-N network. However, this is not true for any collection of 1-by-2 and 2-by-1 components. We need to ensure that certain regularity conditions hold across the collection of components. Specifically, for a given source S and destination D, the performance measure over the end-to-end path between S and D needs to be the same in all components involving S and D. We refer to this condition as *component consistency* and define it formally now. In the definition below we use superscripts to indicate that $\lambda_i^{(j)}$ and $\gamma_i^{(j)}$ are the λ_i and γ_j values associated with component j.

Definition 2 (Component Consistency): Two components are consistent if the performance measure on entire paths between common sources and destinations is the same in each component. Let $C_{D_1,D_2}^{S_1}, C_{D_1,D_3}^{S_1} \in \mathcal{G}_{1,2}$ be two 1-by-2 components and let $C_{D_1}^{S_1,S_2}, C_{D_1}^{S_1,S_3} \in \mathcal{G}_{2,1}$ be two 2-by-1 components all defined on a common set of sources and destinations. Components $C_{D_1,D_2}^{S_1} = (\lambda_1^{(1)}, \lambda_2^{(1)}, \lambda_3^{(1)})$ and $C_{D_1,D_3}^{S_1} = (\lambda_1^{(2)}, \lambda_2^{(2)}, \lambda_3^{(2)})$ are said to be consistent if $\lambda_1^{(1)} + \lambda_2^{(1)} = \lambda_1^{(2)} + \lambda_2^{(2)}$. Similarly, $C_{D_1}^{S_1,S_2} = (\gamma_1^{(3)}, \gamma_2^{(3)}, \gamma_3^{(3)})$ and $C_{D_1}^{S_1,S_3} = (\gamma_1^{(4)}, \gamma_2^{(4)}, \gamma_3^{(4)})$ are said to be consistent if $\gamma_1^{(3)} + \gamma_3^{(3)} = \gamma_1^{(4)} + \gamma_3^{(4)}$. Finally, $C_{D_1,D_2}^{S_1}$ and $C_{D_1}^{S_1,S_2}$ are said to be consistent if $\lambda_1^{(1)} + \lambda_2^{(1)} = \gamma_1^{(3)} + \gamma_3^{(3)}$.

The following theorem establishes that any M-by-N network can be constructed from a collection of 1-by-2 and 2-by-1 components.

Theorem 1 (Decomposition of M-by-N Components): Assume A1-A5 and fix a set of sources $\mathcal{S}$ and destinations $\mathcal{D}$ with $|\mathcal{S}| = M$ and $|\mathcal{D}| = N$. Let $\left(\mathcal{G}_{1,2}\right)^{M\binom{N}{2}}$ denote the collection of sets of consistent 1-by-2 components on $\mathcal{S}$ and $\mathcal{D}$ such that if $C_1 \in \left(\mathcal{G}_{1,2}\right)^{M\binom{N}{2}}$ then there is a 1-by-2 component in C_1 for every source and pair of destinations. Similarly, let $\left(\mathcal{G}_{2,1}\right)^{\binom{M}{2}N}$ denote the collection of sets of consistent 2-by-1 components on $\mathcal{S}$ and $\mathcal{D}$ such that if $C_2 \in \left(\mathcal{G}_{2,1}\right)^{\binom{M}{2}N}$ then there is a 2-by-1 component in C_2 for each destination and pair of sources. Let

$$\mathcal{C}_{M,N} = \left(\mathcal{G}_{1,2}\right)^{M\binom{N}{2}} \times \left(\mathcal{G}_{2,1}\right)^{\binom{M}{2}N} \tag{1}$$

denote the product of these two collections. That is, each $C \in \mathcal{C}_{M,N}$ is a collection of 1-by-2 and 2-by-1 components with one 1-by-2 component for each source and pair of destinations and one 2-by-1 component for each destination and pair of sources, such that these components are pair-wise consistent. There is a one-to-one correspondence between $\mathcal{G}_{M,N}$ and $\mathcal{C}_{M,N}$.

Proof: We will construct a bijection between $\mathcal{G}_{M,N}$ and $\mathcal{C}_{M,N}$. To simplify the discussion, for $C \in \mathcal{C}_{M,N}$ and $S_1, S_2 \in \mathcal{S}$, $D_1, D_2 \in \mathcal{D}$, we use the notation $C_{D_1,D_2}^{S_1}$ to refer to the 1-by-2 component in C corresponding to S_1, D_1, and D_2. Similarly, let $C_{D_1}^{S_1,S_2}$ denote the 2-by-1 component in C for S_1, S_2, and D_1.

Given $G \in \mathcal{G}_{M,N}$, obtaining all 1-by-2 and 2-by-1 components for any source destination is relatively straightforward. For each source $S \in \mathcal{S}$, and pair of destinations $D_1, D_2 \in \mathcal{D}$ let $b = B(S; D_1, D_2)$ and set

$$C_{D_1,D_2}^{S} = \big(\theta(\mathcal{P}[S,b]),\, \theta(\mathcal{P}[b,D_1]),\, \theta(\mathcal{P}[b,D_2])\big). \tag{2}$$

Likewise, for each pair of sources $S_1, S_2 \in \mathcal{S}$ and destination $D \in \mathcal{D}$ let $j = J(S_1, S_2; D)$ and set

$$C_{D}^{S_1,S_2} = \big(\theta(\mathcal{P}[S_1,j]),\, \theta(\mathcal{P}[S_2,j]),\, \theta(\mathcal{P}[j,D])\big). \tag{3}$$

This completely determines the mapping from $\mathcal{G}_{M,N}$ to $\mathcal{C}_{M,N}$, and it is clear that two networks $G, \tilde{G} \in \mathcal{G}_{M,N}$ can map to the same $C \in \mathcal{C}_{M,N}$ only if $\theta(\mathcal{P}[S,a]) = \tilde{\theta}(\tilde{\mathcal{P}}[S,a])$ and $\theta(\mathcal{P}[a,D]) = \tilde{\theta}(\tilde{\mathcal{P}}[a,D])$ for all sources S, destinations D, and internal nodes a. However, if this is the case then G and $\tilde{G}$ must be identical.

Next, we construct a mapping from each element $C \in \mathcal{C}_{M,N}$ to an M-by-N component $G \in \mathcal{G}_{M,N}$ and verify its uniqueness. Let $C \in \mathcal{C}_{M,N}$ be given. A key observation is that describing the internal structure of G amounts to identifying the locations of all branching and joining points. Consider the path from a source S to a destination D. Two internal nodes a and b along this path, e.g., two branching points, are identical if $\theta(\mathcal{P}[S,a]) = \theta(\mathcal{P}[S,b])$ or equivalently, $\theta(\mathcal{P}[a,D]) = \theta(\mathcal{P}[b,D])$. Thus, having the quantities $\theta(\mathcal{P}[S,a])$ and $\theta(\mathcal{P}[a,D])$ for every internal node (branching and joining point) and every source and destination is equivalent to identifying the M-by-N network. Observe that the branching point $B(S; D_1, D_2)$ appears in C_{D_1,D_2}^{S}, and the joining point $J(S_1, S_2; D)$ appears in $C_{D}^{S_1,S_2}$, so C contains information about every internal node. We can completely specify θ, and thus G, from the elements of C in the following fashion. For a given source $S \in S$ and pair of destinations $D_1, D_2 \in \mathcal{D}$ let $C_{D_1,D_2}^{S} = (\lambda_1, \lambda_2, \lambda_3)$. Take $b = B(S; D_1, D_2)$ to be the branching point for this pair of paths and set

$$\begin{aligned} \theta(\mathcal{P}[S,b]) &= \lambda_1 \qquad (4)\\ \theta(\mathcal{P}[b,D_1]) &= \lambda_2 \qquad (5)\\ \theta(\mathcal{P}[b,D_2]) &= \lambda_3. \qquad (6) \end{aligned}$$

Similarly, given a pair of sources $S_1, S_2 \in \mathcal{S}$ and destination $D \in \mathcal{D}$ let $C_{D}^{S_1,S_2} = (\gamma_1, \gamma_2, \gamma_3)$. Take $j = J(S_1, S_2; D)$ to be the joining point for these two paths and set

$$\begin{aligned} \theta(\mathcal{P}[S_1,j]) &= \gamma_1 \qquad (7)\\ \theta(\mathcal{P}[j,D_1]) &= \gamma_2 \qquad (8)\\ \theta(\mathcal{P}[j,D_2]) &= \gamma_3. \qquad (9) \end{aligned}$$

Repeating these steps for all possible combinations of sources and destinations defines a unique θ and completes our construction of G from C. Moreover, it is clear that two collections of components $C, \tilde{C} \in \mathcal{C}_{M,N}$ can both map to the same network $G \in \mathcal{G}_{M,N}$ only if $C_{D_1,D_2}^{S} = \tilde{C}_{D_1,D_2}^{S}$ and $C_{D}^{S_1,S_2} = \tilde{C}_{D}^{S_1,S_2}$ for all source and destination combinations, in which case $C = \tilde{C}$. ■

An immediate consequence of this result is that an appropriate collection of 2-by-2 components suffices to reconstruct any M-by-N network component.

Corollary 1: Under assumptions A1-A5, for sources $\mathcal{S}$ and destinations $\mathcal{D}$, there is a one-to-one correspondence between $\mathcal{G}_{M,N}$ and

$$C^2_{M,N} = \left(\mathcal{G}_{2,2}\right)^{\binom{M}{2}\binom{N}{2}}, \tag{10}$$

the space of collections of consistent 2-by-2 components with one 2-by-2 component for each pair of sources and pair of destinations.

From Theorem 1 it is clear that each 2-by-2 component gives us two 2-by-1's and two 1-by-2's, and the complete collection of 2-by-2 components provides 1-by-2 and 2-by-1 components for every combination of sources and destinations. Existing single source probing techniques typically employ back-to-back packet probes to identify parameters on each of the 1-by-2 components [19], [30], [34]–[36]. Consider the inverted-Y topology shown in Figure 1(a). The source transmits one packet to destination D_1 and then immediately transmits another packet to D_2. Before these packets reach the branching point they will have very similar experiences in terms of queues encountered along the way because they are closely spaced. After the branching point their experiences are independent, as they traverse through completely different sets of queues. Correlations in the measurements obtained at the destinations reflect information about the shared portion of the path (up to the branching point) which is then used to estimate parameters of 1-by-2 components and reconstruct 1-by-N topologies.

In light of Theorem 1, we would also like to measure 2-by-1 components, however the back-to-back packet probe methodology fails. Clearly, transmitting packets through the wild Internet from different sources so that they consistently arrive at a joining point back-to-back would be quite a feat. We can still obtain information about multiple source topologies without complete 2-by-1 component information. In particular, Corollary 1 suggests that 2-by-2 components are also useful.

Figure 2 depicts some examples of 2-by-2 components. The component depicted in Figure 2(a) stands out from the rest, as the paths to both receivers join at the same node. The branching point is also the same node for both of the underlying 1-by-2 components, and so the links from the branching point to each destination are identical in each 1-by-2 component. In principle, we should be able to take advantage of this information to improve our estimates of the performance parameters on these links. Knowing that paths from both sources to both destinations join at a *common node* and that this node is *above* the common branching point is useful for gleaning further information about the M-by-N network topology. Because of these distinguishing features over other 2-by-2 components, we refer to the component depicted in Figure 2(a) as the *shared* 2-by-2 component and focus our attention on it. Any 2-by-2 component which is not shared is called non-shared.

Finally, note that in general the complete collection of 2-by-2 components is sufficient but not necessary for obtaining all 1-by-2 and 2-by-1 components, since each 1-by-2 will appear in multiple 2-by-2 components (fix one source and a pair of destinations, and enumerate over all other sources). Thus, while measuring all 2-by-2 components may seem like a heavy load, in practice we can get away with much less probing.

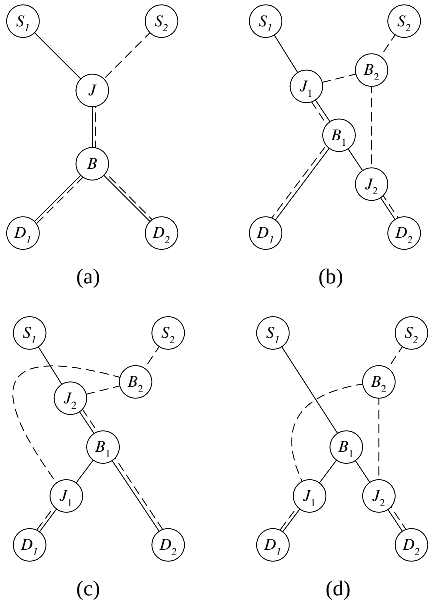


Fig. 2. Examples of 2-by-2 components. The shared component depicted in figure (a) has special structure (a "shared" joining point) which is can be exploited to improve parameter estimates and learn partial information about the structure of larger networks.

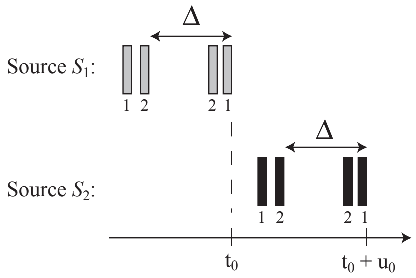


Fig. 3. A multi-destination probe, comprised of four back-to-back packet pairs. Each source transmits two back-to-back packet pairs with constant inter-pair spacing Δ. The number beneath each packet indicates its destination, and the offset u_0 between each source's transmit time is randomized.

III. Multiple Source Sampling Architecture

This section develops a methodology for taking measurements from two sources to two destinations which can simultaneously be used to distinguish whether the underlying topology is shared and to characterize network performance on logical links. A single multi-destination probe is shown in Figure 3. Each source transmits two back-to-back packet pairs spaced by a pre-determined (constant) amount of time, Δ. The packets within each back-to-back pair go do different destinations, as indicated by the number directly under each packet. The offset, u_0, between sending times at the two sources is randomized. Destinations simply record the order in which packets arrive ("from S_1 first" or "from S_2 first"). Because these operations do not require precise time synchronization the algorithm is easy to implement and offers reliable performance in practice.

Back-to-back packet pairs are commonly used to infer internal performance characteristics from end-to-end measurements. Our probes are novel in the way we structure the back-to-back packet pairs transmitted by two sources to measure whether the underlying topology is shared. In order to facilitate explaining and analyzing the architecture, we initially assume the sources are precisely synchronized. This assumption is relaxed later. We also assume that packets are not reordered within the network and discuss this assumption further at the end of the section.

A. Packet Arrival Order

To motivate our probe design we first discuss the packet arrival order metric. Consider the 2-by-1 component depicted in Figure 1(b). Suppose the sources simultaneously transmit a packet to the destination. Under the assumptions listed above, the packets will arrive at the destination in the same order they arrive at the joining point. That is, assuming the packets are not reordered after they pass through the joining point, their arrival order is uniquely determined *at* the joining point. The shared component is unique in that there is one joining point for paths to both destinations, thus we expect arrival order properties to be the same at each destination. Non-shared topologies, on the other hand, have distinct joining points for each destination. The design of our multiple source sampling architecture leverages this difference to characterize the underlying topology.

B. Multi-Destination Probes

To analyze and understand the multi-destination probe design we begin by considering what happens to the first packet in each back-to-back packet pair. The corresponding sequence of first packets for three consecutive multi-destination probes is depicted in Figure 4. Source S_1 transmits a packet to destination D_1 at time t_0 and then transmits a packet to D_2 after Δ seconds. The spacing Δ is chosen to be sufficiently large so that the two packets are never in the same queue. Specifically,

$$\Delta > \frac{\text{packet size}}{\text{min. bandwidth}}, \tag{11}$$

where “min. bandwidth” refers to the minimum bandwidth of all links in the 2-by-2 network. In practice this bandwidth can be estimated using one of a number of tools, or Δ can simply be chosen reasonably large (on the order of 10 milliseconds). By imposing this spacing the measurements from these two packets are independent.

Source S_2 transmits packets in a similar configuration, but with a random offset between the transmission of its first packet and the time S_1 makes its initial transmission. That is, if S_1 transmits packets at times t_0 and $t_0+\Delta$ then S_2 transmits at times t_0+u and $t_0+u+\Delta$, where u is a uniform random variable on the interval $[-R, R]$ and R is much larger than Δ. We will describe how to choose R more precisely later, after clarifying its role. These four packets – two from each source – constitute a single probe. In subsequent probes the values of Δ and R remain fixed, but a new offset u is drawn, independently, at each repetition. Successive probe transmission times t_i are spaced by at least $2R$ so that the resulting measurements are independent.

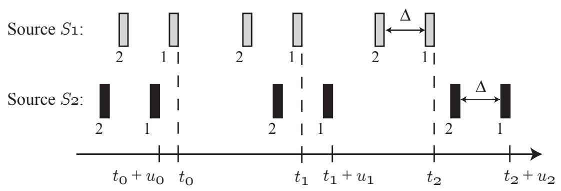


Fig. 4. Focusing on the initial packet of each back-to-back pair, to analyze arrival order measurements. The offset u_i is an i.i.d. random variable, uniformly distributed on $[-R, R]$, where $R \gg Delta$. The time between probes, $t_{i+1} - t_i$, is at least $2R$ to ensure probes are independent.

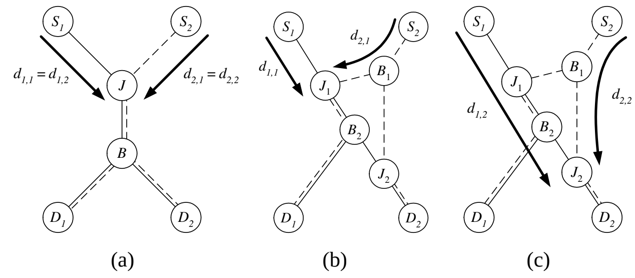


Fig. 5. Packet arrival order at D_i is the same as the order in which they arrive at J_i. Arrival order is determined by the delays incurred by packets travelling from the sources to the joining point. Shared and non-shared topologies are depicted with delays to each joining point labelled. In the shared topology there is a common joining point. The collaborative multiple source probing algorithm leverages this idea to identify whether a topology is shared or not.

Let $d_{1,1}$ and $d_{2,1}$ denote the delays incurred by packets in a particular probe travelling from sources S_1 and S_2 to joining point $J_1 = J(S_1, S_2; D_1)$ as depicted in Figure 5. Consider the first packet transmitted by each source to D_1 in the case where $u = 0$ so that both sources transmit simultaneously. The packet arrival order at D_1 indicates whether $d_{1,1} < d_{2,1}$ or vice versa. If the packet from S_1 arrives first then $d_{1,1} < d_{2,1}$ and if the packet from S_2 arrives first then $d_{2,1} < d_{1,1}$. Equivalently, the arrival order is given by the sign of the quantity $\delta_1 \equiv d_{2,1} - d_{1,1}$.

If there is a non-zero offset u then the arrival at D_1 is a function of

$$(t_0 + u + d_{2,1}) - (t_0 + d_{1,1}) = \delta_1 + u. \tag{12}$$

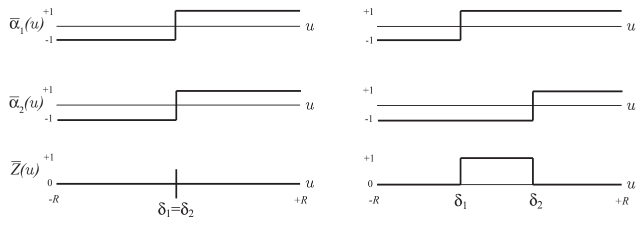
Setting $\alpha_1 = \text{sign}(\delta_1 + u)$, we have that $\alpha_1 = +1$ when the packet from S_1 arrives before the packet from S_2 at D_1, and $\alpha_1 = -1$ when the packet from S_2 arrives first. Thus, α_1 indicates arrival order at destination 1. Defined in a similar fashion, $\alpha_2 = \text{sign}(\delta_2 + u)$ reflects arrival order at D_2.

Because of cross traffic in the network, we regard the delays $d_{i,j}$ as independent random variables. Notice that for the shared topology shown in Figure 5(a),

$$\bar{\delta}_1 = \mathbb{E}[d_{2,1} - d_{1,1}] = \mathbb{E}[d_{2,2} - d_{1,2}] = \bar{\delta}_2, \tag{13}$$

since there is a unique joining point. Thus, if the 2-by-2 component is shared then on average, the arrival orders at both receivers will be the same. Define the arrival order statistic

$$Z = \mathbb{I}\{\alpha_1 \neq \alpha_2\}, \tag{14}$$



(a) Shared (b) Non-shared

Fig. 6. Displaying $\bar{\alpha}_1$, $\bar{\alpha}_2$, and $\bar{Z}$ as functions of the random offset u for both (a) shared and (b) non-shared topologies. Because $\delta_1 \neq \delta_2$ in the non-shared case, there is a range of offsets where we will observe different arrival orders at each destination.

where $\mathbb{I}\{\cdot\}$ is the indicator function. The random variable Z takes value 1 when the arrival order at each destination is different, and is zero otherwise. We treat Z as a Bernoulli random variable with parameter $\rho \in [0,1]$ quantifying the probability of there being different arrival orders at each destination. If the 2-by-2 component we are probing is shared then on average $Z = 0$, and so $\rho \approx 0$.

On the other hand, if the component is not shared it is unlikely that the delay differences, δ_1 and δ_2, to each destination are the same. For a subset of offset values, as illustrated in Figure 6, we expect the arrival order to be different at each destination. The random offset, u, provides a mechanism for exploring the behavior of an unknown 2-by-2 network. Let $\bar{\alpha}_1(u) = \mathbb{E}[\alpha_1|u]$, $\bar{\alpha}_2(u) = \mathbb{E}[\alpha_2|u]$, and $\bar{Z}(u) = \mathbb{E}[Z|u] = P(\alpha_1 \neq \alpha_2|u)$. Figure 6 shows these quantities for shared and non-shared topologies.

We estimate $\rho = \mathbb{E}[Z]$ via Monte Carlo integration, by transmitting a sequence of n probes with offsets $u_1, u_2, \ldots, u_n$ drawn i.i.d. on $[-R, R]$, and recording arrival orders at each destination. Ignoring the effects of cross traffic, to compute the expectation we need to ensure that R is chosen in such a way that the entire range where $Z = 1$ is contained in the interval $[-R, R]$. After all n probes have been transmitted, the arrival order outcomes are collected at a central location and we compute $Z_1, \ldots, Z_n$. Recall that Δ and the spacing $t_{i+1} - t_i$ are designed so that arrival order measurements from different probes are independent. We then estimate by computing $\hat{\rho} = \frac{1}{n}\sum_{i=1}^{n} Z_i$. If any packets within a probe are dropped, we discard the arrival order information for that probe and adjust the estimate accordingly. In Section IV we discuss how loss information is more explicitly incorporated into the procedure to improve performance.

C. Cross Traffic and Single-Destination Probes

Now, our goal is to determine whether the 2-by-2 component we are probing is shared or non-shared. As described above, if it is shared then we expect the arrival order to always be the same at each destination. However, bursts of cross traffic can cause different arrival order events. To quantify cross-traffic effects we transmit probes with the same structure described above but with *all* packets transmitted to the same destination. We call these *single-destination* probes. By forcing all of packets to go through the same joining point we mimic conditions of the shared component. Any time the arrival order is different for a single-destination probe (if the first pair of packets arrives in a different order at the destination than the second pair), it must be due to cross traffic. As above, we transmit many single-destination probes to D_1 with different offsets. Then we average the resulting arrival order measurements to obtain $\hat{\rho}_1$, which characterizes the amount of cross-traffic along the path to J_1. A similar sequence of single-destination probes are transmitted to D_2 to yield $\hat{\rho}_2$.

Regardless of whether the underlying topology is shared or not, we always expect to have $\hat{\rho}_1 \approx 0$ and $\hat{\rho}_2 \approx 0$, since cross traffic is the only mechanism which can cause different arrival orders with all packets going to the same destination. When the underlying topology is shared, we additionally expect all three setups (multi-dest. probes, and single-dest. to D_1 or D_2) to give similar results since packets pass through the same joining point.

When the topology is not shared there are two factors affecting the arrival order statistics, Z_i, for multi-destination probes. In addition to cross traffic, different arrival orders can occur because the delays to each joining point, δ_1 and δ_2, are not equal. Thus, if the underlying 2-by-2 component is not shared, $\hat{\rho}$ should be significantly larger than $\hat{\rho}_1$ and $\hat{\rho}_2$. We develop a formal procedure for deciding whether a 2-by-2 component is shared or not from the measurements in Section IV.

D. Synchronization

One attractive feature of packet arrival order measurements is that they do not require precise timing infrastructure. Destinations only record the order in which packets arrive. It is not practical to assume that sources are precisely synchronized either. However, it is reasonable to assume that the sources can achieve a coarse level of synchronization, e.g., via a crude handshaking mechanism. We expect that sources will be able to reliably synchronize to within a few milliseconds at the beginning of an experiment.

We characterize the discrepancy between the two source clocks in terms of a constant offset and a difference in rate. Letting $\tau_1(t)$ and $\tau_2(t)$ denote each source's perception of time, set $\tau_2(t) = \beta\tau_1(t) + \kappa$ for constants β and κ. Without loss of generality, let $\tau_1(t) = t$. Suppose that probes are sent every $T \geq 2R$ seconds so that S_1 begins transmitting at times $t_0,\ t_0{+}T,\ t_0{+}2T, \ldots$, and so on. Recalling (12), the quantity determining packet arrival order, we find that the expression for the arrival order of the kth probe at D_2 is

$$\begin{aligned} \alpha_2(k) &= \operatorname{sign}(d_{2,2} - d_{1,2} + u + \kappa + k\beta T) &(15)\\ &= \operatorname{sign}(d_{2,2} - d_{1,2} + \tilde{u}_k), &(16) \end{aligned}$$

where $\tilde{u}_k = u + \kappa + k\beta T$. From this perspective, we can consider clock differences at the sources in terms of how they affect the distribution of the random offset. The constant offset, κ, acts as an initial offset so that for the first probe ($k = 0$), $\tilde{u}_0$ is distributed uniformly on the interval $[-R + \kappa, R + \kappa]$. Then the rate β shifts this interval by β for each subsequent probe. Note that β need not be known precisely. As long as

$$\delta_1, \delta_2 \in [-R + \kappa + k\beta T,\ R + \kappa + k\beta T] \qquad (17)$$

for each k then the probability of observing different arrival orders on each individual trial is the same, and our computation is not effected. Clock drift equally effects measurements to both destinations in the shared case. The major concern that the transmission times at each source may become so disparate that transition region of an unshared component may fall outside the probing window $[-R, R]$. By choosing R sufficiently large it is clear that coarse synchronization between sources suffices. If β is very large relative to R, it may be necessary to reestablish the coarse level of synchronization periodically. However, we do not need precise synchronization over the course of the entire experiment.

E. On Packet Reordering and Load Balancing

In their 2002 study on packet reordering, Bellardo and Savage conclude that the probability of two packets travelling along the same network path being reordered is highly correlated with the time-spacing between them as they traverse the network [37]. The probability of reordering decreases dramatically as the space between packets increases. Their empirical results indicate that packets travelling more than 200 microseconds apart are reordered with probability less than 0.01. Iannaccone et al. have studied packet reordering in Sprint's network [38]. Similarly, they observe that between 1% and 2% of packets traversing the network are reordered, however their study focuses on traffic in TCP flows in which packets are transmitted in back-to-back clusters. For certain offsets, u, in our multiple source probing algorithm packets will occasionally arrive at a joining point very close to each other, making them susceptible to reordering. These are precisely the same offsets for which cross traffic may cause different arrival orders. Thus, we can group the effects of reordering with random queueing delay effects and treat these together as a source of noise.

It is also possible that load balancing may be employed within the network we are probing. This situation violates our assumption that there is a unique path from each node to any destination. Often, in order to reduce packet reordering within TCP flows, load balancing systems distribute packets over multiple paths using a source/destination-based hash. Then packets with the same source and destination get routed along the same path, but packets with different sources or different destinations potentially get routed down different paths. In this case, the underlying topology can be much more complicated than the 2-by-2 components depicted in Figure 2 and assumptions A2 and A3 stated in Section II may be violated. That is, the paths from two sources to a destination could join and branch multiple times. Our procedure simply determines the *existence* of a shared joining point before the final branching point in the 2-by-2 component. Regardless of what happens above the joining point (e.g., joining and branching before joining again), the measurements will indicate that the topology is shared. If this type of structure does not occur then the measurements will reflect a non-shared topology.

F. Incorporating Performance Measurements

Recall the multi-destination probe structure depicted in Figure 3. Our discussion thus far has focused on arrival orders of the first probe in each back-to-back packet pair. Existing unicast single source network tomography techniques use back-to-back packet probes for estimating link-level performance. In fact, if multicast packets are being used then we do not even need the back-to-back packets to infer performance characteristics. For unicast measurements, individual back-to-back packet pairs can be used to assess performance characteristics while arrival order measurements in conjunction with the multi-destination probe design are used to characterize the topology. The procedure described in Section IV combines performance and arrival order measurements to jointly infer the topology (shared/non-shared) and link-level performance parameters.

G. Scaling to Larger Networks

A common criticism of active probing techniques – those based on sending traffic into the network as opposed to passively observing existing traffic – is their inability to scale to many sources and destinations. If experiments are performed for each pair of destinations and there are N destinations then the number of experiments grows like $\binom{N}{2} \propto N^2$. Each experiment translates to more traffic being transmitted over the network, which is undesirable. The results presented in Section VI indicate that accurate estimates can be achieved using roughly 1000 probes per pair of destinations. If these measurements are made over the course of five minutes, assuming a probe size of 70 bytes, the average load on the network is less than 2kbps. However, with a network of M sources and N destinations this amount of traffic is multiplied by a factor of M^2N^2, which is unacceptable. This scaling factor can be decreased by simultaneously performing experiments to multiple destinations, using groups of many back-to-back packets similar to the technique described in [19]. Instead of sending groups of two back-to-back packets, each source transmits groups of N back-to-back packets, one for each destination. Then the scaling factor is reduced to M^2N. Linear growth in the number of receivers is a huge improvement, since typically there may be many receivers but only a handful of sources will be used.

In comparison to single-source active probing techniques, there are two additional advantages, from a scalability standpoint, to using multiple sources in a cooperative fashion. First, the probing load is distributed among the sources. In single source schemes all of the probe traffic concentrates on the initial link leaving the source. By using multiple sources this load is more evenly distributed across the network. Additionally, by jointly incorporating measurements from multiple sources in our statistical inference we can obtain high accuracy estimates of internal performance characteristics using fewer probes than one would need if the measurements from each source were analyzed independently.

IV. Statistical Test for Shared Topologies

In this section we address the problem of deciding whether a 2-by-2 component is shared given a set of multiple source measurements. Couched in decision theory, our procedure is flexible and can accommodate arrival order, loss, and delay variance measurements, or any combination thereof. When both arrival measurements are used in conjunction with one of the performance modalities the procedure jointly solves for the topology characterization and performance estimates. In consideration of space limitations on this paper we focus on the case where arrival order measurements and loss measurements are available. For a complete outline of the framework please see [39].

Suppose that the sources have carried out a set of experiments. Let $\boldsymbol{z}$ denote the set of arrival order measurements and let $\boldsymbol{y}$ denote the set of loss measurements from the experiments. The paths from each source to the two destinations each form a 1-by-2 component. Using the notation introduced in Section II – specifically, in Figure 1 – each 1-by-2 component is characterized by three link-level performance parameters, $\lambda_1^{(i)}, \lambda_2^{(i)}, \lambda_3^{(i)}$, where we use the superscript i to index which source's 1-by-2 topology we are referring to.

We will construct a hypothesis test to determine whether the underlying 2-by-2 component is shared or not. 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 $\boldsymbol{\lambda} = (\lambda_1^{(1)}, \lambda_2^{(1)}, \lambda_3^{(1)}, \lambda_1^{(2)}, \lambda_2^{(2)}, \lambda_3^{(2)})$ denote the six dimensional vector of loss rates, and let $\boldsymbol{\rho} = (\rho, \rho_1, \rho_2)$ denote the three dimensional vector of different arrival order probabilities. The key difference between each hypothesis is the number of free parameters. Under the non-shared hypothesis we make the approximation that there is no correspondence between the two 1-by-2 components, and so all nine variables in $(\boldsymbol{\lambda}, \boldsymbol{\rho})$ are allowed to vary. However, under the shared hypothesis we impose the restrictions $\lambda_2^{(1)} = \lambda_2^{(2)}$, $\lambda_3^{(1)} = \lambda_3^{(2)}$, and $\rho = \rho_1 = \rho_2$ based on characteristics of the joining and branching points of the shared 2-by-2 topology. Thus, under the shared hypothesis there are only five degrees of freedom.

Taking the standard decision-theoretic approach, we calculate the likelihood of our data under each hypothesis, $p(\boldsymbol{y}, \boldsymbol{z} | H_i, \boldsymbol{\lambda}, \boldsymbol{\rho})$. By assumption, the spacing Δ is large enough so that each back-to-back packet pair in the probes are independent, and since dropped packets are not included in the arrival order measurements $\boldsymbol{y}$ and $\boldsymbol{z}$ are independent. Thus, we can factor the likelihood function according to

$$p(\boldsymbol{y}, \boldsymbol{z} | H_i, \boldsymbol{\lambda}, \boldsymbol{\rho}) = p(\boldsymbol{y} | H_i, \boldsymbol{\lambda}) p(\boldsymbol{z} | H_i, \boldsymbol{\rho}). \tag{18}$$

For this specific example, both the loss measurements and arrival order measurements are Bernoulli distributed.

Since the parameters $\boldsymbol{\lambda}$ and $\boldsymbol{\rho}$ are unknown, we take the generalized likelihood ratio test (GLRT) approach to solving this composite hypothesis problem. In the GLRT, the unknown parameters $\boldsymbol{\lambda}$ and $\boldsymbol{\rho}$ are replaced with their maximum likelihood estimates under each model. The generalized likelihood ratio is given by

$$\Lambda(\boldsymbol{y}, \boldsymbol{z}) = \frac{\max_{\boldsymbol{\lambda} \in [0,1]^6,\ \boldsymbol{\rho} \in [0,1]^3} p(\boldsymbol{y} | H_N, \boldsymbol{\lambda}) p(\boldsymbol{z} | H_N, \boldsymbol{\rho})}{\max_{\boldsymbol{\lambda} \in [0,1]^4,\ \boldsymbol{\rho} \in [0,1]} p(\boldsymbol{y} | H_S, \boldsymbol{\lambda}) p(\boldsymbol{z} | H_N, \boldsymbol{\rho})}. \tag{19}$$

Then a decision is made according to

$$\Lambda(\boldsymbol{y}, \boldsymbol{z}) = \quad \underset{H_S}{\overset{H_N}{\gtrless}} \quad \eta, \tag{20}$$

for a threshold η. When the likelihood ratio is greater than the threshold, the test declares that the topology is non-shared. Otherwise the test declares it is shared.

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. If the threshold is too large then we will be too aggressive in declaring topologies to be shared, and conversely if it is too small then we run the risk of not identifying shared topologies at all. 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 [40]. Under mild assumptions on the regularity of the likelihood functions $p(\boldsymbol{y} | H_i, \boldsymbol{\lambda})$ and $p(\boldsymbol{z} | H_i, \boldsymbol{\rho})$ which are satisfied for this setup, Wilks' Theorem states that under the shared (i.e., null or restricted) hypothesis, $2 \log \Lambda(\boldsymbol{y}, \boldsymbol{z}) \xrightarrow{d} \chi_\nu^2$, where ν is the difference in the number of degrees of freedom under each hypothesis. In other words, if we are using loss and arrival order measurements then under the shared hypothesis, $2 \log \Lambda(\boldsymbol{y}, \boldsymbol{z})$ converges in distribution to a chi-squared random variable with four degrees of freedom. By knowing the distribution of the log likelihood ratio statistic under the shared hypothesis we can fix a threshold based on a desired probability of mistakenly declaring that the topology is not shared when it is really shared (referred to as a Type I error). For example, to have a Type I error rate of approximately 25%, set $\eta = 0.429$.

A. Decision Performance

The previous section described how to set a threshold in the statistical test by using Wilks' Theorem to control the Type I error rate. To precisely quantify the Type II error rate one must adopt a model for the process generating arrival order observations. Because of the complicated interplay between queueing, background traffic, and characteristics of the underlying topology (e.g., propagation delays), any model we use here will probably not be of much use in practice. Instead, we build an intuition for how the problem parameters R, $|\delta_1 - \delta_2|$, and the number of samples effect the overall system performance.

Intuitively, the test developed in the previous section is simultaneously performing two tasks. It's primary function is to determine whether the number of different arrival orders is statistically significant, indicating a non-shared topology. Statistical significance is measured by the ability of the measurements to resolve the region $[\delta_1, \delta_2]$ where arrival orders transition in the non-shared case (see Fig. 6(b)). Thus, the quantities at play are the width of the probing interval, $[-R, R]$, the target region $[\delta_1, \delta_2]$, and the level of "noise" due to background traffic. The larger the size of the probing interval is very large relative to the target region (i.e., $2R$ vs. $|\delta_1 - \delta_2|$) the more probes are needed to resolve the target region. Similarly, if variability in arrival orders due to queueing is large relative to systematic arrival orders (attributable to a non-shared topology), then more measurements must be taken to average away the effects of queueing. Since the size of the target region is not known ahead of time these tradeoffs are not easily quantified. However, we emphasize that system accuracy can always be improved by collecting more measurements, at the cost of increased bandwidth usage.

In the experiments and simulations reported we choose R to be the maximum round trip time, motivated by the following sequence of bounds:

$$|\delta_i| < \max(d_{1,i}, d_{2,i}) < \max(RTT_{1,i}, RTT_{2,i}), \qquad (21)$$

where i ranges over the destinations. This guarantees that if the topology is not shared then the target region is included in $[-R, R]$. Although this approach may seem too conservative, we have found that this setting produces accurate inferences using a reasonable number of probes (roughly 1000).

Placing meaningful distributions on the size $|\delta_1 - \delta_2|$ of the target region, or the level of background traffic is not a simple task since these quantities may vary greatly depending on the setting. When probes are transmitted across the Internet at large we expect that propagation delays will dominate queueing delay, due to geographically long links and over-provisioned network infrastructure. This scenario is advantageous in our setup, since longer propagation delays correspond to larger target regions. When the network of interest spans a smaller geographic area (e.g., campus or metropolitan networks) or involves wireless links, the target region may not be as well pronounced, making the decision task more challenging. Recent studies of RTT characteristics suggest that both of these scenarios are plausible [41], [42]. We emphasize that variability due to background traffic can always be overcome by taking more measurements.

V. Merging Single-Source Topologies

Next, we will show how knowledge of whether each 2-by-2 component is shared or not can be used to infer characteristics of the M-by-N network component. More specifically, we consider the problem of merging two 1-by-N topologies on the same set of destinations. The challenge in this problem stems from the fact that logical topologies do not come equipped with labels for internal nodes. Rather, as we saw in Section II, locations of internal nodes are determined by their relative distance (in terms of a performance metric) along the path from a source to a destination.

Single source topologies only contain branching points since all paths originate at the same source. Thus, the problem of merging two single source topologies amounts to identifying the locations of joining points. Without measuring the parameters of 2-by-1 components, the best we can hope to do is to identify the relative order of joining and branching points along every path. This section describes an algorithm for estimating the possible range of values which could be taken by $\theta(\mathcal{P}[S,j])$ and $\theta(\mathcal{P}[j,D])$ for joining points j. We also describe necessary conditions for shared/unshared results obtained using the techniques described in the previous section to uniquely identify a ordering of joining and branching points.

In what follows we assume that the 1-by-N tree topology from one source to all receivers is known. There are a number of methods for identifying single-source logical tree topologies using end-to-end measurements, including those described in [18], [20], [24], [26], [43]. Our goal is to locate, with respect to the known topology, where the paths from a second source to the same set of destinations join in this topology. Let S_1 and S_2 be two sources and let $G_1 \in \mathcal{G}_{1,N}$ be the logical tree topology from source S_1 to N destinations. In Section II we stated assumption A3, that the paths for every pair of sources to a given receiver join at one particular point. Based on this assumption, the path from S_2 to a destination D must join the path from S_1 to D somewhere before D. Without any other information, we have no way of knowing where the joining point lies along this path. However, if the 2-by-2 component from S_1 and S_2 to two destinations D_1 and D_2 is shared, then we know that the joining point lies somewhere above (i.e., closer to S_1) the branching point to D_1 and D_2 in G_1. Thus, information about shared topologies can be used to narrow the range where a joining point could possibly lie with respect to the G_1 topology.

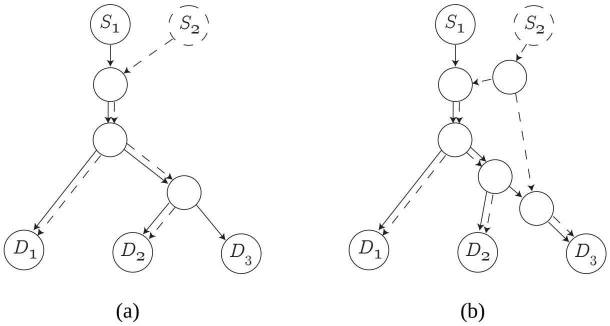


Fig. 7. Locating joining points from S_2 with respect to S_1's topology (solid edges) using information about shared and non-shared 2-by-2 subcomponents. (a) If the 2-by-2 component for (D_1, D_2) is shared then the joining point to D_1 and D_2 must lie on the logical link directly below S. (b) If the components for (D_1, D_3) and (D_2, D_3) are not shared then the joining point to D_3 must be located on the link immediately before D_3.

Information about non-shared topologies can also be useful, when used in conjunction with knowledge of shared topologies. Suppose there are three destinations D_1, D_2, and D_3, such that $B(S_1; D_1, D_2)$ is closer to S_1 than the branching point $B(S_1; D_2, D_3)$ to D_2 and D_3, as depicted in Figure 7. Also, suppose that the 2-by-2 component $(S_1, S_2; D_1, D_2)$ is shared, but $(S_1, S_2; D_1, D_3)$ and $(S_1, S_2; D_2, D_3)$ are not shared. From the shared information we know that the joining points $J(S_1, S_2; D_1)$ and $J(S_1, S_2; D_2)$ lie above the branching point $B(S_1; D_1, D_2)$. The non-shared information also implies that the third joining point $J(S_1, S_2; D_3)$ must lie along the logical link from $B(S_1; D_2, D_3)$ to D_3. If it were any closer to S_1 then the 2-by-2 component to D_2 and D_3 would have to be shared.

By locating joining points using the logic described above, we are essentially limiting the range of values which the performance metric $\theta(\mathcal{P}[J_i, D_i])$ can take, where J_i denotes the joining point from source S_1 and S_2 to destination D_i. Without information about which 2-by-2 components are shared, the location is unconstrained and can take values in the interval $\big[0, \theta(\mathcal{P}[S_1, D_i])\big)$. Let $B_{i,j} = B(S_1; D_i, D_j)$ denote the branching point from S_1 to destinations D_i and D_j. If we know that the 2-by-2 component to D_i and D_j is shared, then we have lower bounds

$$\theta(\mathcal{P}[J_i, D_i]) \geq \theta(\mathcal{P}[B_{i,j}, D_i]) \quad (22)$$
$$\theta(\mathcal{P}[J_j, D_j]) \geq \theta(\mathcal{P}[B_{i,j}, D_j]). \quad (23)$$

Similarly, information about non-shared 2-by-2 components, in conjunction with information about shared components can be used to upper bound $\theta(\mathcal{P}[J_i, D_i])$.

The algorithm described in Figure 8 computes these bounds in a systematic fashion. Step 1 of the algorithm uses information about shared 2-by-2 components to compute the tightest possible lower bound on $\theta(J_j)$ for each destination. Then Step 2 uses shared and non-shared components to tighten the upper bounds. In Step 2, the set $\mathcal{I}$ corresponds to indices of destinations for which the branching points $B_{i,j}$ are closer to S_1 than the current lower bound on $\theta(J_j)$. If $i \in \mathcal{I}$ then we know that the 2-by-2 component for D_i and D_j is not shared. This step then checks whether the path from S_2 to D_i passes through $B_{i,j}$ by using other shared information. If this is the case then we can tighten the upper bound, since otherwise the 2-by-2 component for D_i and D_j must also be shared.

The algorithm is not iterative, so we do not need to worry about convergence. When run to completion, the algorithm produces bounds on the locations of joining points to each destination with respect to the S_1 tree topology, G_1. The best we could hope to do is to isolate these joining points to a single logical link in G_1, between two branching points[1]. When this is possible we say that the joining points are *identifiable with respect to* G_1.

Definition 3 (Identifiability): Given the single source topology, $G_1 \in \mathcal{G}_{1,N}$, from S_1 to N destinations and intervals $[a_1, b_1), \ldots, [a_n, b_n)$ bounding the locations where the paths from S_1 and S_2 to D_i join, we say that the joining points are identifiable with respect to G_1 if for every destination D_i, there is no branching point $B_{i,j}$ with $\theta(\mathcal{P}[B_{i,j}, D_i]) \in \big(a_i, b_i\big)$.

Note that the lower bounds returned by our algorithm correspond to nodes in G_1 (either a destination, or a branching point). The definition for identifiability simply implies that there is no other branching point between the node at the lower bound, and the upper bound b_i. The following theorem gives necessary and sufficient conditions for identifiability of joining points with respect to a topology G_1, given whether each 2-by-2 component is shared or not.

Theorem 2 (Test for Identifiability): Let $G_1 \in \mathcal{G}_{1,N}$ and for each pair of receivers $i, j \in \{1, \ldots, N\}$, $i \neq j$, let $s(i, j)$, the indicator of whether the 2-by-2 component from S_1 and S_2 to D_i and D_j, be given. Identify a binary variable $m_{i,j}$ with each branching point $B_{i,j}$ in G_1. For each pair of destinations D_i, D_j, if $s(i, j) = 1$ set $m_{i,j} = 1$, set $m(i, k) = 1$ for all $B_{i,k}$ which are descendants of $B_{i,j}$ in G_1, and similarly set $m(j, k) = 1$ for all $B_{j,k}$ which are descendants of $B_{j,k}$ in G_1. The joining points are identifiable with respect to G_1 if an only if $m(i, j) = 1$ for all pairs i, j.

Proof: To see why having every $m(i, j) = 1$ is sufficient condition for identifiability, consider the fact that if all $m(i, j) = 1$ then every branching point in G_1 appears in a path from S_2 to some destination. In this manner, every logical link in G_1 is isolated, and the paths from S_2 to each destination are resolved to a single link. This the joining point for destination D_j either lies on the logical link entering D_j or on the link just above the highest shared branching point, and no higher.

To prove the opposite direction, assume that the joining points are identifiable and suppose that there is a branching point $B_{i,j}$ for which $m(i, j) = 0$. This implies that $B_{i,j}$ is neither a shared branching point nor does it appear in any shared path from S_2 to D_i or to D_j. On the other hand, since $B_{i,j}$ is an internal node in the logical tree topology, it has at least two descendants. In particular, there is an outgoing link in the path to D_i and a different outgoing link in the path to D_j, and the lower bounds on joining points J_i and J_j lie below $B_{i,j}$. However, since $m(i, j) = 0$ one of these joining points could potentially lie above $B_{i,j}$, and this contradicts the assumption that the joining points are identifiable. ■

VI. Simulations

Our previous papers [1] and [2] describes a series of experiments conducted using techniques described in this paper. Among those results, using arrival order measurements we successfully identified shared and non-shared components in an Internet experiment involving two sources and seven destinations. The sources were both located in North America. Destinations were located in N. America and Europe. In the Internet experiment we validated our results using `traceroute`. Another experiment was reported where our technique was used to successfully characterize 2-by-2 components for 2 sources and 18 destinations located in a university LAN. For this experiment we confirmed that the inferred components were correct with the help of the IT department. Finally, we conducted a set of simulations illustrating that joint inference using loss and arrival order measurements could significantly improve performance.

For this paper we have performed another set of simulations on a larger network topology, mimicking real-world conditions. The results reinforce our belief that the techniques developed in this paper, arrival order measurements in particular, are robust in a variety of conditions. The simulated network was composed of 318 nodes, modelled after the Abilene multicast topology [44]. Measurements were made from two sources, located off the Chicago and Indiana nodes, to nine destinations, each positioned off one of the other core network nodes. The resulting logical topology is depicted in Figure 9. In total there were 22 shared 2-by-2 components and 14 non-shared. Cross traffic was produced by a collection of web servers and clients located throughout the network. Typical packet drop rates on each link in the network ranged between 0 and 1.5%. In these experiments each source transmitted a total of 1000 probes. The simulation was repeated 100 times.

Figure 10 depicts the receiver operator characteristic for our decision scheme, one minus the Type II error probability versus the Type I error. A Type I error is one where we

[1] Recall that each logical link in the single-source topology G_1 may be a concatenation of physical links involving routers or switches which are not branching points. However, in the merged logical topology one of these nodes may appear as a joining point. Hence, the second tree joins “between” two branching points in the first tree. E.g., see Fig. 7.

Merging Algorithm

Inputs: Sources S_1 and S_2; a set of destinations $\mathcal{D}$ with $|\mathcal{D}| = N$; the 1-by-N tree topology $G_1 \in \mathcal{G}_{1,N}$ from S_1 to all destinations; and results from tests for shared components with $s(i,j) = 1$ if the 2-by-2 component to destinations D_i and D_j is shared, and $s(i,j) = 0$ otherwise

Output: Intervals $I_j \subset \mathbb{R}$ which bound the range taken by $\theta(\mathcal{P}[J_j, D_j])$, with respect to G_1.

Initialization: For each destination D_j, set $I_j = \big[0, \theta(\mathcal{P}[S_1], D_j)\big)$.

Step 1 (Lower Bounds): For each pair of destinations D_i, D_j, let our current bounds on joining point location be $I_i = (a_i, b_i)$ and $I_j = (a_j, b_j)$. If $s(i,j) = 1$, update

$$\begin{aligned} I_i &= \big[\max\{a_i, \theta(\mathcal{P}[B_{i,j}, D_i])\}, b_i\big) && (24) \\ I_j &= \big[\max\{a_j, \theta(\mathcal{P}[B_{i,j}, D_j])\}, b_j\big). && (25) \end{aligned}$$

Step 2 (Upper Bounds): For each destination D_j, we now have $I_j = [a_j, b_j)$. Let

$$\mathcal{I} = \{i \in \{1, \ldots, N\}, i \neq j \ : \ \theta(\mathcal{P}[B_{i,j}, D_j]) > a_j\} \tag{26}$$

denote the set of indices of other destinations for which the branching points $B_{i,j}$ are closer to S_1 than the current lower bound on J_j. For each $i \in \mathcal{I}$, if for any other destination $k = 1, \ldots, N$, $k \neq j$ we have $B_{i,j} \in \mathcal{P}[S_1, D_k]$ and $s(i,k) = 1$, then update

$$I_j = \big[a_j, \min\{b_j, \theta(\mathcal{P}[B_{i,k}, D_j])\}\big). \tag{27}$$

Fig. 8. Our algorithm for merging two 1-by-N single source logical topologies.

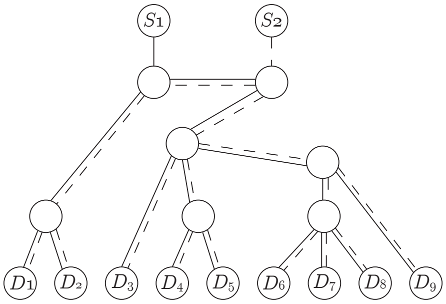


Fig. 9. The two source, nine destination logical topology from the simulations. The simulated physical topology contained 318 nodes, modelled after the Abilene multicast network.

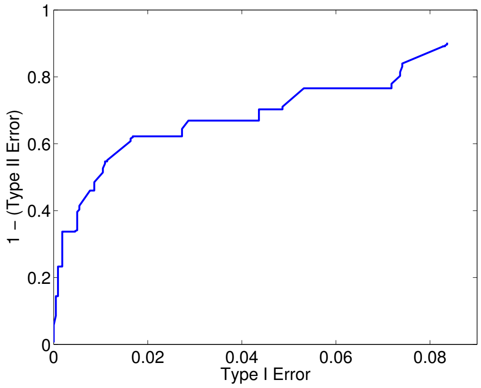


Fig. 10. A plot of one minus the Type II error versus the Type I error. By choosing a threshold, η, for the hypothesis test, we fix an operating point along the horizontal axis, and performance reflected in the score along the vertical axis.

mistakenly declare that a shared topology is non-shared. A Type II error is the opposite, identifying a non-shared topology as shared. The ideal operating point is in the upper left-hand corner of the figure, where both the Type I and Type II errors are zero. In practice, choosing a particular decision threshold, η, determines the operating position along the horizontal axis, and the value along the vertical axis is an indication of performance. Note that in this figure the Type I error only ranges over $[0, 0.09]$ as opposed to $[0, 1]$. So, if we are willing to operate at a Type I of roughly 0.1 we will achieve a Type II error of roughly 0.1 also. In this experiment, only arrival order measurements were used. However, we can improve performance by incorporating loss measurements into the decision scheme when the number of observed packet drops is significant. Performance can be improved by taking more measurements.

Observe that in the simulated topology, knowing whether each 2-by-2 component is shared or not is nearly sufficient to identify joining point locations with respect to the S_1 topology. All joining points will fall either immediately to the left or to the right of the first branching point after S_1. There will be one branching point with value $m(i,j)$, as defined in Theorem 2, will be equal to zero. The same is true for identifiability with respect to the S_2 topology.

VII. Conclusions

This paper presented theoretical results unifying our previous work. By proving that M-by-N network components can be decomposed into a collection of 2-by-2 components we reduced the general multiple source multiple receiver network tomography problem to a simpler case. We then developed a novel measurement scheme and testing procedure which can be used to distinguish between the shared and non-shared classes of 2-by-2 component networks. Our procedure jointly estimates link-level performance parameters and classifies topology. We illustrated how this information can be used to merge two single sources trees and established a test for the identifiability of multiple source tree topologies from measurements. Simulations illustrate the efficacy of our procedure.

To our knowledge, it is not possible to completely characterize 2-by-1 components using current tomographic techniques. This is an interesting open question and its solution would enable a framework for completely discovering and characterizing general network topologies.

REFERENCES

[1] M. Coates, M. Rabbat, and R. Nowak, “Merging logical topologies using end-to-end measurements,” in *Proc. ACM SIGCOMM Conf. on Internet Measurement*, Miami, FL, October 2003.

[2] M. Rabbat, R. Nowak, and M. Coates, “Multiple source, multiple destination network tomography,” in *Proc. IEEE Infocom*, Hong Kong, March 2004.

[3] M. Coates, A. Hero, R. Nowak, and B. Yu, “Internet tomography,” *IEEE Signal Processing Magazine*, May 2002.

[4] R. Castro, M. Coates, G. Liang, R. Nowak, and B. Yu, “Internet tomography: Recent developments,” *Statistical Science*, vol. 52, no. 3, pp. 499–517, Aug. 2004.

[5] A. Clauset and C. Moore, “Why mapping the Internet is hard,” e-print: arXiv:cond-mat/0407339, 2004.

[6] B. Yao, R. Viswanathan, F. Chang, and D. Waddington, “Topology inference in the presence of anonymous routers,” in *Proc. IEEE Infocom*, San Francisco, CA, March 2003.

[7] B. Donnet, T. Friedman, and M. Crovella, “Improved algorithms for network topology discovery,” in *Proc. Passive and Active Measurements Workshop*, Boston, MA, Mar. 2005.

[8] L. Dall’Asta, I. Alvarez-Hamelin, A. Barrat, A. Vazquez, and A. Vespignani, “A statistical approach to the traceroute-like exploration of networks: theory and simulations,” *Lecture Notes in Computer Science*, vol. 3405, p. 140, 2005.

[9] `traceroute` – a tool for printing the route packets take to a network host., http://ee.lbl.gov/traceroute.tar.Z.

[10] B. Cheswick, H. Burch, and S. Branigan, “Mapping and visualizing the Internet,” in *Proc. USENIX Annual Tech. Conf.*, Monterey, CA, June 2000.

[11] B. Huffaker, D. Plummer, D. Moore, and k. claffy, “Topology discovery by active probing,” in *Proc. Symposium on Applications and the Internet*, Nara City, Japan, Jan. 2002.

[12] R. Govindan and H. Tangmunarunkit, “Heuristics for Internet map discovery,” in *Proc. IEEE Infocom*, Tel Aviv, Israel, Mar. 2000.

[13] D. Wetherall, N. Spring, and R. Mahajan, “Measuring ISP topologies with Rocketfuel,” in *Proc. ACM Sigcomm*, Pittsburgh, PA, Aug. 2002.

[14] Y. Bejerano, Y. B. M. Garofalakis, and R. Rastogi, “Physical topology discovery for large multi-subnet networks,” in *Proc. IEEE Infocom*, San Francisco, CA, April 2003.

[15] Y. Breitbart, M. Garofalakis, C. Martin, R. Rastogi, S. Seshadri, and A. Silberschatz, “Topology discovery in heterogeneous IP networks,” in *Proc. IEEE Infocom*, Tel Aviv, Israel, March 2000.

[16] Y. Breitbart, M. Garofalakis, B. Jai, C. Martin, R. Rastogi, and A. Silberschatz, “Topology discovery in heterogeneous IP networks: the NetInventory system,” *IEEE/ACM Trans. Networking*, vol. 12, no. 3, pp. 401–414, 2004.

[17] B. Lowekamp, D. R. OHallaron, and T. R. Gross, “Topology discovery for large Ethernet networks,” in *Proc. ACM Sigcomm*, San Diego, CA, Aug. 2001.

[18] S. Ratnasamy and S. McCanne, “Inference of multicast routing trees and bottleneck bandwidths using end-to-end measurements,” in *Proc. IEEE Infocom*, New York, NY, March 1999.

[19] N. Duffield, F. Lo Presti, V. Paxson, and D. Towsley, “Inferring link loss using striped unicast probes,” in *Proc. IEEE Infocom*, Anchorage, Alaska, April 2001.

[20] N. Duffield, J. Horowitz, and F. Lo Presti, “Adaptive multicast topology inference,” in *Proc. IEEE Infocom*, Anchorage, Alaska, April 2001.

[21] N. Duffield, J. Horowitz, F. Lo Presti, and D. Towsley, “Multicast topology inference from measured end-to-end loss,” *IEEE Trans. Information Theory*, pp. 26–45, Jan. 2002.

[22] A. Bestavros, J. Byers, and K. Harfoush, “Inference and labeling of metric-induced network topologies,” in *Proc. IEEE Infocom*, New York, NY, June 2002.

[23] ——, “Inference and labeling of metric-induced network topologies,” *IEEE Trans. on Parallel and Distributed Systems*, vol. 16, no. 10, Oct. 2005.

[24] M. Coates, R. Castro, M. Gadhiok, R. King, Y. Tsang, and R. Nowak, “Maximum likelihood network topology identification from edge-based unicast measurements,” in *Proc. ACM Sigmetrics*, Marina Del Rey, CA, June 2002.

[25] R. Castro, M. Coates, and R. Nowak, “Likelihood based hierarchical clustering,” *IEEE Trans. Signal Processing*, vol. 42, pp. 2308–2321, Aug. 2004.

[26] M. Shih and A. Hero, “Network topology discovery using finite mixture models,” in *Proc. IEEE ICASSP*, Montreal, May 2004.

[27] Y. Tsang, M. Yildiz, R. Nowak, and P. Barford, “Network radar: Tomography from round trip time measurement,” in *Proc. ACM SIGCOMM Conf. on Internet Measurement*, Taromina, Sicily, Italy, Oct. 2004, pp. 175–180.

[28] M. Coates and R. Nowak, “Sequential monte carlo inference of internal delays in nonstationary communication networks,” *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 366–376, February 2002.

[29] T. Bu, N. Duffield, F. Lo Presti, and D. Towsley, “Network tomography on general topologies,” in *Proc. ACM Sigmetrics*, Marina Del Rey, CA, June 2002.

[30] K. Harfoush, A. Bestavros, and J. Byers, “Robust identification of shared losses using end-to-end unicast probes,” in *Proc. IEEE Conf. Network Protocols*, Osaka, Japan, Nov. 2000.

[31] D. Rubenstein, J. Kurose, and D. Towsley, “Detecting shared congestion of flows via end-to-end measurement,” in *Proc. of ACM SIGMETRICS 2000*, San Jose, CA, June 2000.

[32] D. Katabi, I. Bazzi, and X. Yang, “A passive approach for detecting shared bottlenecks,” in *Proc. IEEE Conf. on Comp. Comm. and Networks*, Arizona, Oct. 2001.

[33] W. Cui, S. Machiraju, R. Katz, and I. Stoica, “SCONE: A tool to estimate shared congestion among internet paths,” EECS Department, University of California, Berkeley, Tech. Rep. UCB/CSD-04-1320, 2004.

[34] R. Cáceres, N. Duffield, J. Horowitz, and D. Towsley, “Multicast-based inference of network-internal loss characteristics,” *IEEE Transactions on Information Theory*, vol. 45, pp. 2462–2480, November 1999.

[35] M. Coates and R. Nowak, “Network loss inference using unicast end-to-end measurements.” in *ITC Seminar on IP Traffic, Measurement, and Modeling*, Monterey, CA, Sep. 2000.

[36] N. Duffield and F. Lo Presti, “Multicast inference of packet delay variance at interior network links,” in *Proc. of IEEE Infocom 2000*, Tel Aviv, Israel, March 2000.

[37] J. Bellardo and S. Savage, “Measuring packet reordering,” in *Proc. of the ACM Sigcomm Internet Measurement Workshop*, Marseille, France, Nov. 2002.

[38] G. Iannaccone, S. Jaiswal, and C. Diot, “Packet reordering inside the Sprint backbone,” Sprint Labs, Technical Report TR01-ATL-062917, June 2001.

[39] M. Rabbat, “Multiple source network tomography,” Master’s thesis, Rice University, Houston, TX, May 2003.

[40] S. Wilks, “The large-sample distribution of the likelihood ratio for testing composite hypotheses,” *Annals of Math. Stat.*, March 1938.

[41] J. Aikat, J. Kaur, F. Smith, and K. Jeffay, “Variability in tcp round-trip times,” in *Proc. ACM SIGCOMM Conf. on Internet Measurement*, Miami, FL, October 2003.

[42] Y. Zhang, N. Duffield, V. Paxson, and S. Shenker, “On the constancy of internet path properties,” in *Proc. ACM SIGCOMM Internet Measurement Workshop*, San Francisco, CA, November 2001.

[43] N. Duffield, J. Horowitz, F. Lo Presti, and D. Towsley, “Multicast topology inference from end-to-end measurements,” in *ITC Seminar on IP Traffic, Measurement, and Modeling*, Monterey, CA, September 2000.

[44] Abilene Multicast Map, http://www.abilene.iu.edu/images/ab-mcast.pdf.