

Network Routing Topology Inference from End-to-End Measurements

Jian Ni Haiyong Xie Sekhar Tatikonda Yang Richard Yang
Yale University, New Haven, Connecticut, USA

Abstract—Inference of the routing topology and link performance from a node to a set of other nodes is an important component of network monitoring and application design. In this paper we propose a general framework for designing topology inference algorithms based on additive metrics. Our framework allows the integration of both end-to-end packet probing measurements and traceroute type measurements. Based on this framework we design several computationally efficient topology inference algorithms. In particular, we propose a novel sequential topology inference algorithm to address the probing scalability problem and handle dynamic node joining and leaving. We provide sufficient conditions for the correctness of our algorithms and derive lower bounds on the probability of correct topology inference. We conduct Internet experiments to evaluate and demonstrate the effectiveness of our algorithms.

I. INTRODUCTION

A scalable tool to infer the routing topology and link performance from a node to a set of other nodes can be a particularly useful tool. In *network monitoring*, this tool can help a network operator to obtain routing information and network internal characteristics (e.g., loss rate, delay, utilization) from its network to a set of other collaborating networks that are separated by non-participating autonomous networks. In *application design*, this tool can be particularly useful for peer-to-peer (P2P) style applications where a node communicates with a set of other nodes (called *peers*) for file sharing and multimedia streaming. For example, a node may want to know the routing topology to other nodes so that it can select peers with low or no route overlap to improve resilience against network failures (e.g., [2]). As another example, a streaming node using multi-path may want to know both the routing topology and link loss rates so that the selected paths have low loss correlation (e.g., [3]).

There are two primary approaches to infer the routing topology and link performance of a communication network. Both have their limitations. One is to use tools based on measurements or feedback messages of the internal nodes (e.g., routers). Such an approach is limited as today's communication networks (e.g., the Internet) are evolving towards more decentralized and private administration. For example, a common approach to infer the routing topology from a source node to a destination node in the Internet is to use *traceroute*. Traceroute relies on internal routers responding to ICMP (Internet Control Message Protocol) messages. However, some routers in the Internet do not return ICMP messages or simply discard ICMP messages (e.g., many enterprise networks disable traceroute-like probing due to privacy con-

cerns). These routers are known as *anonymous routers* [24] and their existence makes the routing topology inferred by traceroute-like tools inaccurate. Furthermore, traceroute-like tools cannot discover layer-2 switches or MPLS (Multiprotocol Label Switching) paths that are increasingly being deployed.

Not depending on cooperation from the internal nodes, the *network tomography* approach utilizes end-to-end packet probing measurements (such as packet loss and delay measurements) conducted by the end hosts to infer the routing topology and link performance. Due to its flexibility and reliability, network tomography has attracted many recent studies (e.g., [8], [11]). Many previous network tomography studies are based on multicast probing because of its effectiveness and probing efficiency (e.g., [7], [13], [16], [19], [20]). Since IP multicast is not widely deployed in the Internet, unicast network tomography approaches based on back-to-back unicast packet pairs or strings were investigated (e.g., [10], [14], [22]). The main challenges of network tomography include *computational complexity* and *probing scalability* (especially for unicast probing), which limits the number of destination nodes that a source node can infer. In addition, the focus of previous studies is on a relatively stable set of nodes, while in many applications (e.g., P2P file sharing and streaming applications) nodes may join or leave a session frequently. This places extra challenge for efficient network inference.

In this paper we study the problem of inferring the network routing topology from a source node to a set of destination nodes¹, where the set can be dynamic. We propose a general framework for designing topology inference algorithms based on additive metrics. We show how to construct additive metrics from end-to-end packet probing measurements and traceroute type measurements. Since a linear combination of different additive metrics is still an additive metric, the framework can flexibly utilize all information available from different measurements to achieve best accuracy.

Based on the framework we design several computationally efficient topology inference algorithms. In particular, we propose a novel sequential topology inference algorithm that significantly reduces the probing scalability problem. In addition, our algorithm can handle dynamic node joining and leaving, and thus is particularly desirable for applications where node dynamics are prevalent. We demonstrate the efficiency and effectiveness of the proposed topology inference algorithms

¹We use destination nodes for simplicity, which could be relay nodes or peer nodes of the source node in real applications.

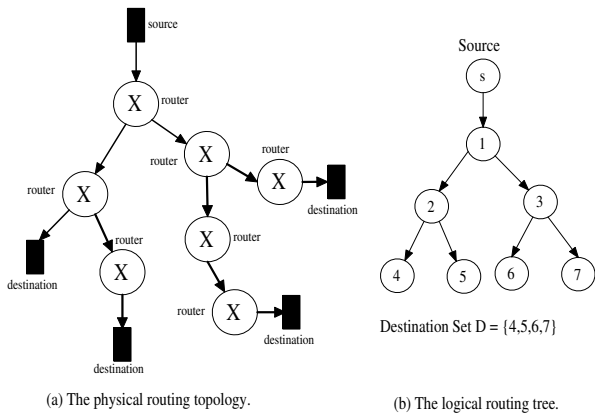


Fig. 1. Single source and multiple destinations: the physical routing topology and the associated logical routing tree topology.

via rigorous analysis and Internet experiments.

The rest of the paper is organized as follows. In Section II we introduce the network model and inference problems. In Section III we discuss how to construct additive metrics from end-to-end measurements. In Section IV and V we propose and analyze a neighbor-joining based topology inference algorithm and a sequential topology inference algorithm which can be applied to any additive metric. We design Internet routing tree topology inference schemes and evaluate their performance via Internet experiments in Section VI. The paper is concluded in Section VII.

II. NETWORK MODEL AND INFERENCE PROBLEMS

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denote the topology of the network, which is a directed graph with node set \mathcal{V} (end systems, internal switches and routers, etc.) and link set \mathcal{E} (communication links that join the nodes). For any nodes i and j in the network, if the underlying routing algorithm returns a sequence of links that connect j to i , we say j is *reachable* from i . We call this sequence of links a *path* from i to j , denoted by $\mathcal{P}(i, j)$. We assume that during the measurement period, the underlying routing algorithm determines a unique path from a node to another node that is reachable from it.

Hence the *physical routing topology* from a source node to a set of destination nodes is a (directed) tree. From the physical routing topology, we can derive a *logical routing tree* which consists of the source, the destinations, and the branching nodes (internal nodes with at least two outgoing links) of the physical routing tree [7], [13]. Note that a logical link may comprise more than one consecutive physical links. An example is shown in Fig. 1. For simplicity we use routing tree to express logical routing tree unless otherwise noted.

Suppose s is a source node in the network, and D is a set of destination nodes that are reachable from s . Let $T = (V, E)$ denote the routing tree from s to nodes in D , with node set V and link set E . Let $U = s \cup D$ be the set of terminal nodes which are nodes of degree one (i.e., end systems). Each node $k \in V$ has a *parent* $f(k) \in V$ such that $(f(k), k) \in E$, and a set of children $c(k) = \{j \in V : f(j) = k\}$,

except that the source (root of the tree) has no parent and the destinations (leaves of the tree) have no children. For notational simplification, we use e_k to denote link $(f(k), k)$.

Each link $e \in E$ is associated with a parameter θ_e (either a scalar or a vector). The network inference problems involve using measurements taken at the terminal nodes to infer (1) topology of the routing tree; (2) link parameters θ_e of links on the routing tree. In this paper we focus on routing tree topology inference. Link parameter inference with known tree topology was studied in [7], [10], [17], [19], [22].

A. Probing Model

A probe from s to D can be a multicast packet sent from s to all nodes in D . By multicast we mean that when an internal node on the routing tree receives the packet, it will duplicate the packet and send a copy to all its children on the tree. Since (IP) multicast is not widely deployed in the Internet, a method to mimic the transmission of a multicast packet is to use back-to-back unicast packet pair or string, in which a source node sends k back-to-back unicast packets to k different destination nodes respectively (we call it $1 \times k$ packet string probing). Since the packets are very close to each other, we assume that the back-to-back packets sent by the source node to different destination nodes have the same network experience (loss, delay, etc.) in the shared links.

For a probe sent by the source node, we define a set of *link state variables* Z_e for all $e \in E$. Z_e takes value in a state set \mathcal{Z} . The distribution of Z_e is parameterized by θ_e , e.g., $\mathbb{P}(Z_e = i) = \theta_e(i)$ for $i \in \mathcal{Z}$.

The transmission of a probe from s to nodes in D will induce a set of *outcome variables* on the routing tree T . For each node $k \in V$, we use X_k to denote the (random) outcome of the probe at node k . X_k takes value in an outcome set \mathcal{X} . The outcome of the probe at node k (i.e., X_k) is determined by the outcome of the probe at node $f(k)$ (i.e., $X_{f(k)}$) and the link state of e_k (i.e., Z_{e_k}):

$$X_k = g(X_{f(k)}, Z_{e_k}). \quad (1)$$

Assumption 1. *The link states are independent from link to link (spatial independence) and are stationary during the measurement period.*

Under Assumption 1 we can show that the outcome variables X_k 's induced by the transmission of a probe on the routing tree form a Markov random field [16]. In addition, under mild conditions, the link parameters of all links on the routing tree as well as the tree topology can be identified (uniquely determined) by the joint distribution of the outcome variables at the terminal nodes [9], [16].

In actual network inference problems, the joint distribution of the outcome variables normally is not given. We need to estimate the joint distribution based on measurements taken at the terminal nodes. Specifically, the source node will send a sequence of n probes, and there are totally n outcomes $X_V^{(t)} = (X_k^{(t)} : k \in V)$, $t = 1, 2, \dots, n$, one for each probe. For the t -th probe, only the outcomes $X_U^{(t)} = (X_k^{(t)} : k \in U = s \cup D)$ at the terminal nodes can be measured and observed. We can

estimate the joint distribution of the outcome variables using the observed empirical distribution, which converges to the stationary distribution almost surely if the link state processes are stationary and ergodic during the measurement period.

B. Network Tomography Examples

Example 1: Link Loss Inference [7]. In this case, the link state variable Z_e is a Bernoulli random variable which takes value 1 with probability α_e if the probe can go through link e , and takes value 0 with probability $1 - \alpha_e \triangleq \bar{\alpha}_e$ if the probe will be lost on the link. α_e is called the *success rate* of link e and $\bar{\alpha}_e$ is called the *loss rate* of link e . The outcome variable L_k is also a Bernoulli random variable, which takes value 1 if the probe successfully reaches node k . It is clear that for link loss inference

$$L_k = L_{f(k)} \cdot Z_{e_k} = \prod_{e \in \mathcal{P}(s,k)} Z_e. \quad (2)$$

Example 2: Link Utilization Inference [12]. In this case, the link state variable Z_e is a Bernoulli random variable which takes value 1 with probability γ_e if the probe will not experience any queueing delay on link e , and takes value 0 with probability $1 - \gamma_e \triangleq \bar{\gamma}_e$ if the probe will experience some queueing delay on the link. $\bar{\gamma}_e$ can be viewed as the utilization of link e . The outcome variable U_k is also a Bernoulli random variable, which takes value 1 if the packet will reach node k with no queueing delay. For this example we also have

$$U_k = U_{f(k)} \cdot Z_{e_k} = \prod_{e \in \mathcal{P}(s,k)} Z_e. \quad (3)$$

Example 3: Link Delay Inference [19]. In this case, the link state variable Z_e is a random variable denoting the random (queueing) delay of link e . θ_e can be a certain moment of Z_e , e.g., $\theta_e = \text{var}(Z_e)$; or the distribution of Z_e is parameterized by θ_e , e.g., $\theta_e(i) = \mathbb{P}(Z_e = i)$, $i \in \mathcal{Z}$. The outcome variable T_k denotes the cumulative (queueing) delay experienced by the probe from s to node k . For link delay inference

$$T_k = T_{f(k)} + Z_{e_k} = \sum_{e \in \mathcal{P}(s,k)} Z_e. \quad (4)$$

III. CONSTRUCT ADDITIVE METRICS

Let $T = (V, E)$ be a routing tree with source node s and destination nodes D . We say d is an *additive metric* on T if

- (a) $0 < d(e) < \infty$, $\forall e \in E$;
- (b) $d(i, j) = \sum_{e \in \mathcal{P}(i,j)} d(e)$, $\forall i, j \in V$.

$d(e)$ can be viewed as the *length* of link e and $d(i, j)$ can be viewed as the *distance* between nodes i and j . Let $U = s \cup D$ be the set of terminal nodes on the tree. We use $d(U^2) = \{d(i, j) : i, j \in U\}$ to denote the distances between the terminal nodes. It is known that the topology and link lengths of a tree are uniquely determined by the distances between the terminal nodes under an additive metric [6].

Suppose the source node s is fixed, for any destination node $i \in D$, let $\rho(i) = d(s, i)$ be the *path length* from s to i (under additive metric d). For any pair of destination nodes $i, j \in D$, let \underline{ij} denote their *nearest common ancestor* (i.e., the ancestor of both i and j that is farthest from root s on the tree). Let $\rho(i, j) = d(s, \underline{ij})$ be the *shared path length* from s to i and j .

Let $\rho(s, D) = \{\rho(i) : i \in D\}$ denote the path lengths from s to nodes in D , $\rho(s, D^2) = \{\rho(i, j) : i, j \in D\}$ denote the shared path lengths from s to pairs of nodes in D . Note that

$$\rho(i, j) = \frac{d(s, i) + d(s, j) - d(i, j)}{2}, \quad \forall i, j \in D. \quad (5)$$

Hence there is a 1-1 mapping between $d(U^2)$ and $\rho(s, D) \cup \rho(s, D^2)$. We can recover the topology of the routing tree if we know either $d(U^2)$ or $\rho(s, D) \cup \rho(s, D^2)$. The key thing is to construct an additive metric for which we can derive/estimate $d(U^2)$ or $\rho(s, D) \cup \rho(s, D^2)$ from end-to-end measurements.

A. Additive Metric Based on Traceroute-like Measurements

Using traceroute-like measurements, the source node s can obtain the unique labels (IP addresses) of the internal nodes (routers) in the path from the source node to any destination node (provided that the internal nodes respond to traceroute-like measurements). We can construct an additive metric d_r by defining the link length $d_r(e)$ to be the number of hops (physical links) contained in logical link e . The path length $\rho_r(i)$ is the number of hops contained in the path from s to i , and the shared path length $\rho_r(i, j)$ is the number of hops contained in the shared portion of the paths from s to i and j . The shared portion of two paths can be determined by comparing the labels of the internal nodes in the two paths.

If some internal nodes do not respond to traceroute-like measurements (e.g., anonymous routers, layer-2 switches, MPLS switches), then the derived path lengths and shared path lengths can be distorted. We use $\hat{\rho}_r(s, D)$ and $\hat{\rho}_r(s, D^2)$ to denote the measured path lengths and shared path lengths with possible measurement errors.

B. Additive Metrics Based on Multicast Probing

For a (multicast) probe sent by the source node, let $X_V = (X_k : k \in V)$ be the outcome Markov random field on T . For each link $(i, j) \in E$ we can define an $M \times M$ (assume $|\mathcal{X}| = M$) *forward link transition matrix* P_{ij} and an $M \times M$ *backward link transition matrix* P_{ji} with entries $P_{ij}(x_i, x_j) = \mathbb{P}(X_j = x_j | X_i = x_i, x_i, x_j \in \mathcal{X})$. If $0 < |P_{ij}|, |P_{ji}| < 1$ for all links, then we can construct an additive metric d_0 with link length [4]:

$$d_0(e) = -\log |P_{ij}| - \log |P_{ji}|, \quad \forall e = (i, j) \in E.$$

For any pair of nodes $i, j \in U$, $d_0(i, j)$ can be computed by

$$d_0(i, j) = -\log |P_{ij}| - \log |P_{ji}|, \quad i, j \in U. \quad (6)$$

There are other choices of the additive metric for the specific network inference problem.

1) *Loss-Based Additive Metric*: For Example 1 (link loss inference) in Section II.B, if $0 < \alpha_e < 1$ for all links, then we can construct an additive metric d_l with link length $d_l(e) = -\log \alpha_e, \forall e \in E$. Under the spatial independence assumption that the link states are independent from link to link, $\rho_l(s, D) \cup \rho_l(s, D^2)$ can be obtained by

$$\begin{aligned} \rho_l(i) &= -\log \mathbb{P}(L_i = 1), \quad i \in D; \\ \rho_l(i, j) &= -\log \frac{\mathbb{P}(L_i = 1)\mathbb{P}(L_j = 1)}{\mathbb{P}(L_i = 1, L_j = 1)}, \quad i, j \in D. \end{aligned} \quad (7)$$

2) *Utilization-Based Additive Metric*: Similarly for Example 2 (link utilization inference), if $0 < \beta_l < 1$ for all links, then we can construct an additive metric d_u with link length $d_u(e) = -\log \beta_e, \forall e \in E$. Under the spatial independence assumption, $\rho_u(s, D) \cup \rho_u(s, D^2)$ can be obtained by

$$\begin{aligned} \rho_u(i) &= -\log \mathbb{P}(U_i = 1), \quad i \in D; \\ \rho_u(i, j) &= -\log \frac{\mathbb{P}(U_i = 1)\mathbb{P}(U_j = 1)}{\mathbb{P}(U_i = 1, U_j = 1)}, \quad i, j \in D. \end{aligned} \quad (8)$$

3) *Delay-Based Additive Metric*: For Example 3 (link delay inference), if $0 < \text{var}(Z_e) < \infty$ for all links, then we can construct an additive metric d_v with link length $d_v(e) = \text{var}(Z_e), \forall e \in E$. Under the spatial independence assumption, $\rho_v(s, D) \cup \rho_v(s, D^2)$ can be obtained by

$$\begin{aligned} \rho_v(i) &= \text{var}(T_i), \quad i \in D; \\ \rho_v(i, j) &= \text{cov}(T_i, T_j), \quad i, j \in D. \end{aligned} \quad (9)$$

As in (6), (7), (8), (9), if we know the pairwise joint distributions of the outcome variables at the terminal nodes, then we can construct an additive metric and derive $\rho(U^2)$ or $\rho(s, D) \cup \rho(s, D^2)$. In actual network inference problems we are not given such distributions. We can use measurements taken at the terminal nodes to estimate the distributions (e.g., using empirical distributions).

Let s send a sequence of n probes to (a subset of) destination nodes in D . For any probed node i , let $T_i^{(t)}$ be the measured (one-way) delay of the t -th probe from s to i , with $T_i^{(t)} = \infty$ means t -th probe lost. We use $T_i^{\min} = \min_t T_i^{(t)}$ to approximate the propagation delay from s to i .

The loss outcomes can be derived as follows: $L_i^{(t)} = 1$ if $T_i^{(t)} < \infty$, and $L_i^{(t)} = 0$ if $T_i^{(t)} = \infty$ (i.e., probe lost). As in [12], the utilization outcomes can be derived as follow: $U_i^{(t)} = 1$ if $T_i^{(t)} - T_i^{\min} < \epsilon$ (probe experiences no queuing delay, where ϵ is a small value, e.g., 0.1ms, to account for possible measurement noise) and $U_i^{(t)} = 0$ otherwise. Then we can construct explicit estimators for the path lengths and shared path lengths in (7), (8), (9) as follows:

$$\hat{\rho}_l(i) = -\log \bar{L}_i, \quad \hat{\rho}_l(i, j) = -\log \bar{L}_i \bar{L}_j / \bar{L}_{ij}; \quad (10)$$

$$\hat{\rho}_u(i) = -\log \bar{U}_i, \quad \hat{\rho}_u(i, j) = -\log \bar{U}_i \bar{U}_j / \bar{U}_{ij}; \quad (11)$$

$$\hat{\rho}_v(i) = \text{var}(T_i), \quad \hat{\rho}_v(i, j) = \text{cov}(T_i, T_j). \quad (12)$$

$\bar{L}_i = \sum_{t=1}^n L_i^{(t)} / n$ (resp., \bar{U}_i) is the *sample mean* of $L_i^{(t)}$'s (resp., $U_i^{(t)}$'s). $\bar{L}_{ij} = \sum_{t=1}^n L_i^{(t)} L_j^{(t)} / n$ (resp., \bar{U}_{ij}) is the *sample mean* of $L_i^{(t)} L_j^{(t)}$'s (resp., $U_i^{(t)} U_j^{(t)}$'s). $\text{var}(T_i)$ is the

sample variance of $T_i^{(t)}$'s (not counting ∞ 's), and $\text{cov}(T_i, T_j)$ is the *sample covariance* of $T_i^{(t)}$'s and $T_j^{(t)}$'s (not counting ∞ 's). Note that possible time asynchronization between the destination nodes and the source node will not affect our estimators in (10), (11), (12).

A nice property of additive metrics is that a linear combination of several additive metrics is still an additive metric. In order to utilize all information collected from different measurements, we can construct a new additive metric using a linear (convex) combination of $\hat{d}_l, \hat{d}_u, \hat{d}_v$: $\hat{d}_t = a_l \hat{d}_l + a_u \hat{d}_u + a_v \hat{d}_v$ with $a_l + a_u + a_v = 1$. The (estimated) path lengths and shared path lengths under the new additive metric can be easily computed: $\hat{\rho}_t = a_l \hat{\rho}_l + a_u \hat{\rho}_u + a_v \hat{\rho}_v$. In practice we can select the coefficients empirically based on the current network state or to minimize the variance of the constructed estimator $\hat{\rho}_t$.

C. Additive Metrics Based on Unicast Packet Pair Probing

The validity of (6), (7), (8), (9) depends on the assumption that the packets (of the same probe) sent to different destination nodes have the same network experience (loss, delay, etc.) in the shared links. This assumption is certainly true for multicast probes, but it may not hold for unicast packet pair/string probes. Can we still construct additive metrics from unicast probing? The answer is yes, under certain conditions.

Suppose the source node s sends two back-to-back packets to destination nodes i and j , for which the first packet (denoted by a) is sent to node i and the second packet (denoted by b) is sent to node j . Let Z_e^a and Z_e^b be the link state variables experienced by packet a and packet b in link e , respectively.

First consider link loss (or utilization) inference. Let $\alpha_e = \mathbb{P}(Z_e^x = 1)$ for $x = a, b$ be the *marginal* link success rate of link e . Let $\beta_e = \mathbb{P}(Z_e^b = 1 | Z_e^a = 1)$ be the *conditional* link success rate of link e , i.e., β_e is the conditional probability of the second packet b successfully goes through link e given that the first packet a successfully goes through link e .

If $0 < \alpha_e < \beta_e \leq 1$ for all links, then $0 < \frac{\alpha_e}{\beta_e} < 1$, and we can construct an additive metric d_l^b with link length $d_l^b(e) = -\log \frac{\alpha_e}{\beta_e}, \forall e \in E$. In real networks, we would expect $\alpha_e < \beta_e$, because the fact that the first packet successfully goes through a link indicates that the link is in good state and the second packet, which closely follows the first packet, can also go through the link. This phenomenon was observed in real Internet measurements (e.g., [5], [23]).

Let L_i^a be the loss outcome variable of packet a at node i , L_j^b be the loss outcome variable of packet b at node j . Under the spatial independence assumption, $\rho_l^b(s, D) \cup \rho_l^b(s, D^2)$ can be obtained by

$$\rho_l^b(i) = -\log \frac{\mathbb{P}(L_i^a = 1)\mathbb{P}(L_i^b = 1)}{\mathbb{P}(L_i^a = 1, L_i^b = 1)}, \quad i \in D;$$

$$\rho_l^b(i, j) = -\log \frac{\mathbb{P}(L_i^a = 1)\mathbb{P}(L_j^b = 1)}{\mathbb{P}(L_i^a = 1, L_j^b = 1)}, \quad i, j \in D. \quad (13)$$

Now consider link delay inference. If $\text{cov}(Z_e^a, Z_e^b) > 0$ for all links (which we would expect to hold in real networks because the two back-to-back packets are very close hence their

experienced delays in the same link are positively correlated), then we can construct an additive metric d'_v with link length $d'_v(e) = \text{cov}(Z_e^a, Z_e^b), \forall e \in E$.

Let T_i^a be the delay outcome variable of packet a at node i , T_j^b be the delay outcome variable of packet b at node j .

$$\begin{aligned} T_i^a &= \sum_{e \in \mathcal{P}(s, \underline{ij})} Z_e^a + \sum_{e \in \mathcal{P}(\underline{ij}, i)} Z_e^a, \\ T_j^b &= \sum_{e \in \mathcal{P}(s, \underline{ij})} Z_e^b + \sum_{e \in \mathcal{P}(\underline{ij}, j)} Z_e^b. \end{aligned}$$

Under the spatial independence assumption, $\rho'_v(s, D) \cup \rho'_v(s, D^2)$ can be obtained by

$$\begin{aligned} \rho'_v(i) &= \text{cov}(T_i^a, T_i^b), \quad i \in D; \\ \rho'_v(i, j) &= \text{cov}(T_i^a, T_j^b), \quad i, j \in D. \end{aligned} \quad (14)$$

Similarly as in (10), (11), (12), we can construct explicit estimators for the path lengths and shared path lengths in (13) and (14) using measured outcomes at the terminal nodes.

IV. TREE TOPOLOGY INFERENCE BASED ON NEIGHBOR JOINING

We first propose a tree topology inference algorithm using (estimated) path lengths and shared path lengths as the input based on the idea of *neighbor joining*. The algorithm begins with a leaf set including all destination nodes. In each step it selects a group of nodes that are likely to be *neighbors* (i.e., *siblings*, nodes with the same parent on the tree), deletes them from the leaf set, creates a new node as their parent and adds that node to the leaf set. The whole process is iterated until only one node left in the leaf set, which will be the child of the root (source node). To avoid trivial cases, we assume $|D| \geq 2$.

Algorithm 1 (Neighbor-Joining Tree Topology Inference)

Input: Source s , Destinations D , $\hat{\rho}(s, D)$, $\hat{\rho}(s, D^2)$, $\Delta > 0$.

1. $V = \{s\}$, $E = \emptyset$.
- 2.1 Find $i^*, j^* \in D$ with the largest $\hat{\rho}(i, j)$ (break the tie arbitrarily). Create a node f as the parent of i^* and j^* .
 $D = D \setminus \{i^*, j^*\}$, $V = V \cup \{i^*, j^*\}$, $E = E \cup \{(f, i^*), (f, j^*)\}$.
 (+) $\hat{d}(f, i^*) = \hat{\rho}(i^*) - \hat{\rho}(i^*, j^*)$, $\hat{d}(f, j^*) = \hat{\rho}(j^*) - \hat{\rho}(i^*, j^*)$.
- 2.2 For each $k \in D$, if $\hat{\rho}(i^*, j^*) - \hat{\rho}(i^*, k) \leq \frac{\Delta}{2}$:
 $D = D \setminus k$, $V = V \cup k$, $E = E \cup (f, k)$.
 (+) $\hat{d}(f, k) = \hat{\rho}(k) - \hat{\rho}(i^*, j^*)$.
- 2.3 For each $k \in D$, compute: $\hat{\rho}(k, f) = \frac{1}{2}(\hat{\rho}(k, i^*) + \hat{\rho}(k, j^*))$.
 $D = D \cup f$. $\hat{\rho}(f) = \hat{\rho}(i^*, j^*)$.
3. If $|D| = 1$, for the $k \in D$: $V = V \cup k$, $E = E \cup (s, k)$.
 Otherwise, repeat Step 2.

Output: Tree $\hat{T} = (V, E)$, and link length $\hat{d}(e)$ for all $e \in E$.

Note that Algorithm 1 only requires (estimated) shared path lengths between the source and pairs of the destinations, $\hat{\rho}(s, D^2)$, to infer the tree topology (steps without (+)). If the (estimated) path lengths $\hat{\rho}(s, D)$ are also available, then Algorithm 1 can also infer the link lengths (steps with (+)). We can use the link lengths returned by Algorithm 1 to infer the link performance parameters (e.g., link loss, utilization, and delay variance parameters in Section III.B).

The neighbor joining idea was widely used in *clustering* for building cluster trees [15] and in *evolutionary biology* for building phylogenetic trees [21]. This idea was applied in [13], [20] to infer the topology of multicast routing trees based on shared losses observed at the destination nodes. Compared with the algorithms in [13], [20], Algorithm 1 only requires (estimated) shared path lengths between pairs of the destination nodes which can be collected from both *multicast* probing and *unicast* packet pair probing as we described in Section III. In addition, Algorithm 1 is *computationally efficient* due to the simplicity of additive metrics. For a general routing tree with N destination nodes, the computational complexity of Algorithm 1 is $O(N^3)$. The algorithms in [13], [20] have an $O(N^3)$ complexity only for binary trees. For general trees one needs to search among all subsets of the destination nodes (# of searches is on the order of 2^N), and numerical root finding procedure is required when the degree of internal nodes is greater than five [13].

A. Condition for Correct Topology Inference

Let T be the true topology of the routing tree, $d(e)$'s be the true link lengths, and $\rho(s, D^2)$ be the true shared path lengths under additive metric d .

Proposition 1. Let $\Delta = \min_{e \in E} d(e)$ be the minimum link length on the routing tree. A sufficient condition for Algorithm 1 to return the correct tree topology is:

$$|\hat{\rho}(i, j) - \rho(i, j)| < \frac{\Delta}{4}, \quad \forall i, j \in D. \quad (15)$$

Therefore, if the estimated shared path lengths $\hat{\rho}(s, D^2)$ are close enough to the true values, then Algorithm 1 will return the correct tree topology. We can derive exponential error bounds for the shared path length estimators in (10) and (11) under Assumption 1 [17]. Formally, for a sample size n (number of probes) and a small $\epsilon > 0$:

$$\mathbb{P}\{|\hat{\rho}_l(i, j) - \rho_l(i, j)| \geq \epsilon\} \leq e^{-c_{ij}(\epsilon)n}.$$

Let \hat{T}_n be the inferred tree topology returned by Algorithm 1 with sample size n . Let $P_n = \mathbb{P}\{\hat{T}_n = T\}$ denote the probability of correct topology inference of Algorithm 1.

Proposition 2. Let $\Delta = \min_{e \in E} d(e)$. If $\mathbb{P}\{|\hat{\rho}(i, j) - \rho(i, j)| \geq \frac{\Delta}{4}\} \leq e^{-c_{ij}(\Delta)n}$ for all $i, j \in D$ where n is the sample size and $c_{ij}(\Delta)$ is a constant determined by i, j , and Δ , then for a routing tree with N destination nodes:

$$P_n \geq 1 - N^2 e^{-c(\Delta)n}, \quad (16)$$

i.e., the probability of correct topology inference of Algorithm 1 goes to 1 exponentially fast in the sample size.

The proofs are omitted due to space limitation, which can be found in [18].

V. DYNAMIC TREE TOPOLOGY INFERENCE

Algorithm 1 in Section IV may have some limitations in practice. First, it requires estimated shared path lengths from the source to all pairs of the destination nodes as the input. If multicast probing is not supported by the network, and the

Procedure: $\text{add_node}(T, k, j, \Delta)$

- IF k is a leaf node on the tree $T = (V, E)$,
 j is sibling (neighbor) of k on the updated tree:
 1. Create a new node p as their parent:
 $V = V \cup \{p, j\}$,
 $E = E \setminus (f(k), k) \cup \{(f(k), p), (p, k), (p, j)\}$.
- ELSE
 Suppose k has l children c_1, \dots, c_l .
 2. c_i selects a destination node d_i descended from it.
 3. Measure/estimate $\hat{\rho}(d_1, d_2)$ and $\hat{\rho}(j, d_i)$ for $i = 1, \dots, l$.
 4. Find d_{i^*} with the largest $\hat{\rho}(j, d_i)$.
- case (a) $\hat{\rho}(d_1, d_2) - \hat{\rho}(j, d_{i^*}) \geq \frac{\Delta}{2}$: j is sibling of k
 5. create a new node p as their parent:
 $V = V \cup \{p, j\}$,
 $E = E \setminus (f(k), k) \cup \{(f(k), p), (p, k), (p, j)\}$.
- case (b) $|\hat{\rho}(d_1, d_2) - \hat{\rho}(j, d_{i^*})| < \frac{\Delta}{2}$: j is child of k
 6. $V = V \cup j$, $E = E \cup (k, j)$.
- case (c) $\hat{\rho}(j, d_{i^*}) - \hat{\rho}(d_1, d_2) \geq \frac{\Delta}{2}$: j is sibling/descendant of c_{i^*}
 7. $\text{add_node}(T, c_{i^*}, j, \Delta)$.

Fig. 2. Procedure to add a new destination node j to routing tree T .

number of destination nodes N is large, then it is difficult to obtain $\hat{\rho}(s, D^2)$ using a single $1 \times N$ (unicast) packet string probing without violating the assumption that the string of packets have the same or positively correlated network experiences in the shared links. If the source node uses back-to-back (unicast) packet pair probings, then it requires $\binom{N}{2} = O(N^2)$ 1×2 probings. If these probings are conducted in parallel, then this will quickly use up the outgoing bandwidth of the source node; on the other hand if these probings are conducted in sequence, then it will take a long time to obtain the measurements and it is more likely that the network state will change during the measurement period which will violate the stationarity assumption (Assumption 1). We tested Algorithm 1 using Internet experiments and we found that it only works well for a small number of destination nodes (6 or less).

Second, in real applications (e.g., P2P applications), the destination nodes that a source node communicates with often change over time. Hence the routing tree topology will also change over time. When a destination node leaves, it is relatively easy to derive the updated routing tree topology from the previous one. When a new destination node joins, we could run Algorithm 1 over the new set of destination nodes to infer the updated routing tree topology. However, this is not an efficient solution when nodes join and leave frequently. Therefore we are motivated to design the following procedure to add a new destination node to the exiting routing tree.

$\text{add_node}(T, k, j, \Delta)$ is a recursive procedure that adds a new destination node j to routing tree $T = (V, E)$ via a node k on the tree. Let $f(k)$ be the parent of k on the (old) tree T . The procedure for add_node is described in Fig. 2.

By running $\text{add_node}(T, s, j, \Delta)$, we add a new destination node j to the routing tree T rooted at s . Note that in Step 3 in order to estimate the shared path lengths, s only needs to send probes to $l + 1$ nodes, where l is the internal node degree. For an l -ary tree with N destination nodes, in the worst case, the source node requires $O(l \log_l N)$ unicast

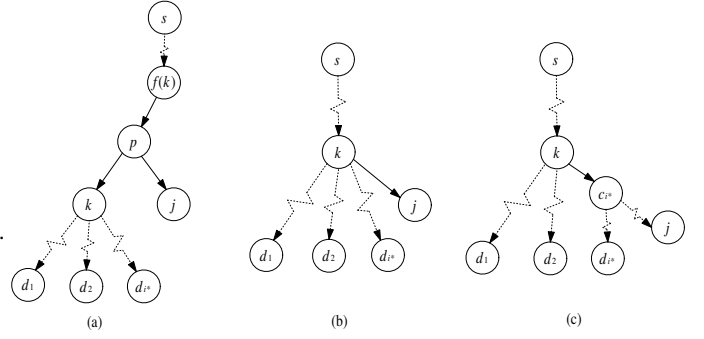


Fig. 3. 3 cases of adding a new node j to the tree via a node k on the tree.

packet pair probings where the tree depth is $O(\log_l N)$. While if we apply Algorithm 1 to infer the topology of the new tree, the source node requires $O(N^2)$ unicast packet pair probings!

A. Apply add_node for Sequential Tree Topology Inference

For a source node s and a set of destination nodes D , we can also apply procedure add_node over the nodes in D in sequence to construct the tree topology incrementally. This is described in the following algorithm.

Algorithm 2 (Sequential Tree Topology Inference)

Input: Source s , Destinations $D = \{1, 2, \dots, N\}$, $\Delta > 0$.

1. $V_0 = \{s\}$, $E_0 = \emptyset$, $T_0 = (V_0, E_0)$.
2. For $j = 1$ to N : $T_j = \text{add_node}(T_{j-1}, s, j, \Delta)$.

Output: Tree $\hat{T} = T_N$.

A comparison between Algorithm 1 and Algorithm 2 is shown in Table I. Note that for multicast probing, Algorithm 1 is more efficient; while for unicast packet pair probing, Algorithm 2 is more efficient.

B. Condition for Correct Topology Inference

If the estimated shared path lengths measured in Step 3 are close enough to the true values, then $\text{add_node}(T, s, j, \Delta)$ will correctly add a new destination node to the tree. Formally,

Proposition 3. *Let Δ be the minimum link length on the updated tree topology including existing destination nodes and the new destination node j . A sufficient condition for the recursive procedure $\text{add_node}(T, s, j, \Delta)$ to return the correct tree topology (after adding node j) is that for all the nodes k visited by the recursive procedure*

$$|\hat{\rho}(d_1, d_2) - \rho(d_1, d_2)| < \frac{\Delta}{4},$$

$$|\hat{\rho}(j, d_i) - \rho(j, d_i)| < \frac{\Delta}{4}, \quad i = 1, 2, \dots, l. \quad (17)$$

Proposition 4. *Let Δ be the minimum link length on the updated tree topology. If for all the nodes k visited by the recursive procedure $\text{add_node}(T, s, j, \Delta)$, we have $\mathbb{P}\{|\hat{\rho}(d_1, d_2) - \rho(d_1, d_2)| \geq \frac{\Delta}{4}\} \leq e^{-c_{d_1 d_2}(\Delta)^n}$ and $\mathbb{P}\{|\hat{\rho}(j, d_i) - \rho(j, d_i)| \geq \frac{\Delta}{4}\} \leq e^{-c_{j d_i}(\Delta)^n}$ for $i = 1, \dots, l$, where n is the sample size and $c_{d_1 d_2}(\Delta)$ and $c_{j d_i}(\Delta)$'s are constants, then the probability of correct topology inference of*

TABLE I
COMPARISON BETWEEN ALGORITHM 1 (NEIGHBOR-JOINING) AND ALGORITHM 2 (SEQUENTIAL)

N Destination Nodes, l -ary Tree with Depth $O(\log_l N)$, Sample Size n with Sample Interval T_0					
		Multicast Probing		Unicast Packet Pair Probing	
		Probing Traffic Overhead	Probing Time Complexity	Probing Traffic Overhead	Probing Time Complexity
Add One Node	Algorithm 1 (NJ)	$O(nN)$	$O(nT_0)$	$O(nN^2)$	$O(nT_0N^2)$
	Algorithm 2 (Sequential)	$O(nl \log_l N)$	$O(nT_0 \log_l N)$	$O(nl \log_l N)$	$O(nT_0l \log_l N)$
Build Whole Tree	Algorithm 1 (NJ)	$O(nN)$	$O(nT_0)$	$O(nN^2)$	$O(nT_0N^2)$
	Algorithm 2 (Sequential)	$O(nNl \log_l N)$	$O(nT_0N \log_l N)$	$O(nNl \log_l N)$	$O(nT_0Nl \log_l N)$

$\text{add_node}(T, s, j, \Delta)$ for an l -ary tree with N destination nodes satisfies:

$$P_n \geq 1 - (l + 1) \log_l N e^{-c(\Delta)n}. \quad (18)$$

The proofs can be found in [18].

VI. SCHEMES FOR INTERNET ROUTING TREE TOPOLOGY INFERENCE

In this section we design schemes for Internet routing tree topology inference using algorithms we have developed so far. We consider the following schemes:

1. *Traceroute-based inference scheme (TR)*: we use traceroute measurements to construct additive metric \hat{d}_r and derive the shared path lengths $\hat{\rho}_r(s, D^2)$ as described in Section III.

2. *Tomography-based inference scheme (Tomo)*: we use unicast packet pair/string measurements to construct additive metrics $\hat{d}_l, \hat{d}_u, \hat{d}_v$ and estimate the shared path lengths $\hat{\rho}_l(s, D^2), \hat{\rho}_u(s, D^2), \hat{\rho}_v(s, D^2)$ as described in Section III. We construct a new additive metric using a convex combination of the additive metrics to utilize all information: $\hat{d}_t = a_l \hat{d}_l + a_u \hat{d}_u + a_v \hat{d}_v$ with $a_l + a_u + a_v = 1$.

We have shown that if the estimated shared path lengths are close enough to the true values (e.g., condition (15) or (17)), then both Algorithms 1 and 2 will return the correct routing tree topology.

For traceroute measurements, the measured shared path lengths can be distorted due to the existence of anonymous routers, layer-2 switches, and MPLS switches. For network tomography measurements, the assumption of *independent* and *stationary* link states can be violated, hence a large sample size with long measurement period may not return more accurate estimation of shared path lengths. Hence the conditions for correct topology inference (15) or (17) may not hold for both type of measurements.

In order to utilize information collected from both traceroute measurements and network tomography measurements to achieve best accuracy, we propose the following hybrid scheme for Internet topology inference:

3. *Traceroute+Tomography inference scheme (TRTomo)*: we use both traceroute measurements and network tomography measurements to construct additive metrics \hat{d}_r and \hat{d}_t , respectively, and we construct a new additive metric $\hat{d}_{rt} = A\hat{d}_r + \hat{d}_t$ with a large A which makes $A\hat{d}_r$ dominate \hat{d}_t . The motivation for selecting a large A is because that traceroute measurements

could be distorted but are nevertheless *consistent*. An anonymous router will affect all the paths passing that router (i.e., the path lengths of those paths are all reduced by 1). Hence if $\hat{\rho}_r(i, j) > \hat{\rho}_r(i, k)$, then we know for sure that j is closer to i than k on the tree. The reverse is not true: even if j is closer to i than k , we may have $\hat{\rho}_r(i, j) = \hat{\rho}_r(i, k)$ because of anonymous routers, hence network tomography measurements are required.

For a large number of destination nodes, we propose to infer the routing tree topology using a two-step procedure: first use traceroute measurements ($\hat{\rho}_r$) (or other heuristics, e.g., round trip times, AS information) to build a skeleton of the tree, then add tomography measurements ($\hat{\rho}_t, \hat{\rho}_{rt}$) on subtrees (with relatively small number of destination nodes) to construct the topology of the subtrees. We find this approach significantly reduces the probing scalability problem of the pure network tomography approach while improve the accuracy of pure traceroute-based approach.

We refer to the above schemes as *TR*, *Tomo* and *TRTomo* for short hereafter. We evaluate their performance via Internet experiments.

A. Experiment Setup and Evaluation Methodology

Experiment Setup: We choose an idle host in our local network as the source node, and two sets of PlanetLab [1] nodes as the destination nodes. We have implemented a *sender utility* program that can send probing packet pairs or strings, and a *receiver utility* program to receive the probing packets and measure the one-way delays of the probing packets. The size of the probing packets is 80 bytes. We collect the measured one-way delays from the receivers through the sender utility program.

The first destination node set, referred to as *US nodes*, consists of 30 hosts in the US (most of them are located in US universities). The second set, referred to as *International nodes*, consists of 30 international nodes (10 in North America, 10 in Europe, 10 in East Asia). Note that the reliability of the chosen nodes are important to our measurements, hence we choose nodes that have low CPU load and long running time.

We run the sender utility on the source and the receiver utility on the two sets of PlanetLab nodes. Each probing from the source to a subset of the destinations consists of 1200 packet strings. The interval between consecutive strings is set to 10 milliseconds (contributing to a probing rate of 64 kbps per destination node).

Evaluation methodology: We evaluate the performance of the three topology inference schemes by varying the *anonymization ratio*, the level of the underlying routers discarding traceroute ICMP probing. For each anonymization ratio, we test the topology inference schemes 20 rounds.

In each round, we first obtain the sequence of underlying routers from the source to each destination using traceroute. The destination nodes we choose have the property that the paths from the source to them contain no or very few anonymous routers so we can obtain the *ground-truth topology* in order to test the topology inference schemes. We then count the total number of unique routers we have seen for all destinations, and compute how many of them in total should be anonymized according to the anonymization ratio. We then iteratively choose a destination randomly, anonymize the last m routers along its route², where m is computed as the anonymization ratio times the route length; we also keep track of the number of unique routers we anonymized in each iteration, and terminate the anonymization procedure once the total number of unique anonymized routers reaches the number we computed *a priori*.

We use two metrics to evaluate the performance of the topology inference schemes. The first metric is *correctness ratio*, defined as the average percentage of internal nodes in the ground-truth topology that are correctly inferred over all rounds. An internal node in the ground-truth topology is *correctly inferred* iff there is an internal node in the inferred topology with the same set of destination nodes descending from it. The second metric is *node ratio*, defined as the average ratio of the number of internal nodes in the inferred topology and in the ground-truth topology over all rounds.

B. Experimental Results

We run experiments using the US nodes and International nodes, and refer to them as US experiments and International experiments, respectively. We plot the correctness ratios (Fig. 4 and 5) and node ratios (Fig. 6 and 7) with varying levels of underlying routers being anonymized, for US and International experiments respectively.

1) *Correctness Ratio:* As shown in Fig. 4 and 5, both TR and TRTomo schemes can correctly infer most of the internal nodes in the ground-truth topology when the anonymization ratio is small. As the anonymization ratio increases, the correctness ratio of the TR scheme decreases to 0; while the correctness ratio of the TRTomo scheme stabilizes around 0.5. This is because the TR scheme heavily rely on routers' support for traceroute probing, while the TRTomo scheme can improve its accuracy by using both traceroute measurements and tomography measurements. When the anonymization ratio is 1 (no routers response to traceroute probing), the TRTomo

²When choosing stable PlanetLab nodes, we find that a lot of nodes are behind routers that do not respond to traceroute probing. Most of these routers are edge routers or access routers of the network in which the destination nodes are located in. This suggests that traceroute probings are likely to be discarded in enterprise networks to protect their internal hosts; hence, the routers in the last few hops to a destination are more likely to be anonymous routers.

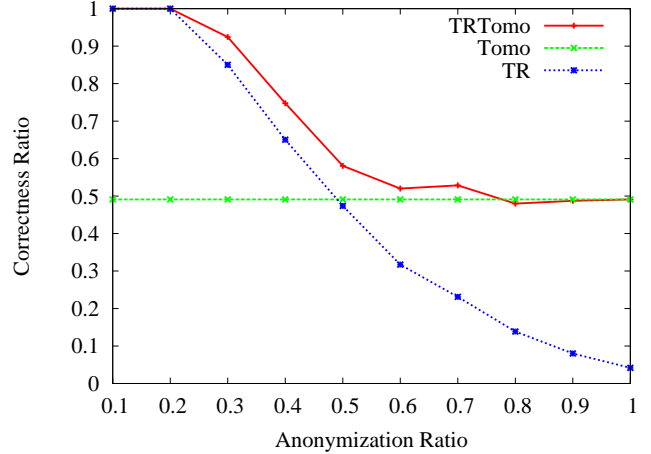


Fig. 4. US-experiment: correctness ratio of inferred topology.

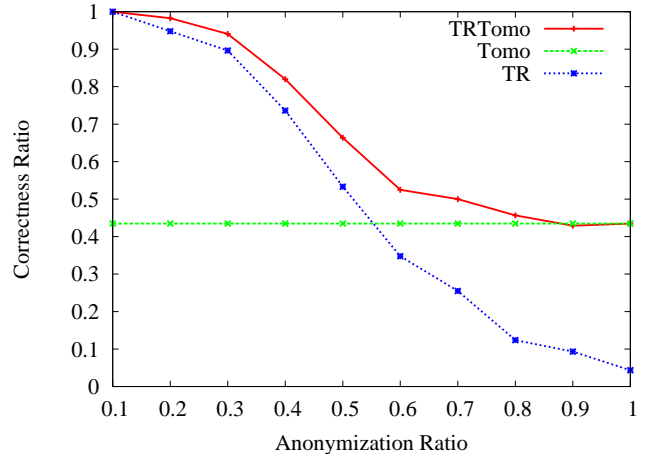


Fig. 5. International-experiment: correctness ratio of inferred topology.

scheme is just the Tomo scheme, so we determine the correctness ratio of Tomo using the correctness ratio of TRTomo at anonymization ratio 1, which is around 0.5.

From our experiences we would like to comment on why the pure Tomo scheme alone can only infer 50% of the internal nodes but cannot infer all the internal nodes in our experiments. First, the link states may be *time-varying* instead of *stationary* during the measurement period. Second, there are several limitations of the PlanetLab testbed. We observed that the network connections from the source to the PlanetLab nodes are pretty good in most of the time, hence the shared path lengths derived from loss and delay metrics are quite small and can be easily distorted by measurement noises. In addition, most PlanetLab nodes are often running multiple applications and processes. This introduces non-negligible node delays to the delay measurements which will affect the delay and utilization metrics.

2) *Node Ratio:* As shown in Fig. 6 and 7, the node ratio of the TR scheme is close to 1 when the anonymization ratio is

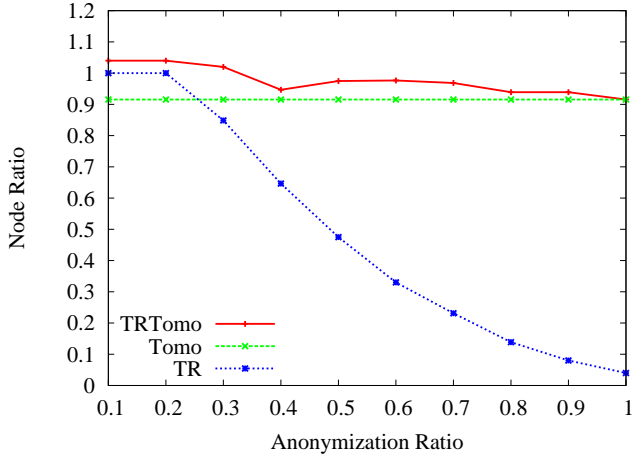


Fig. 6. US-experiment: node ratio of inferred topology.

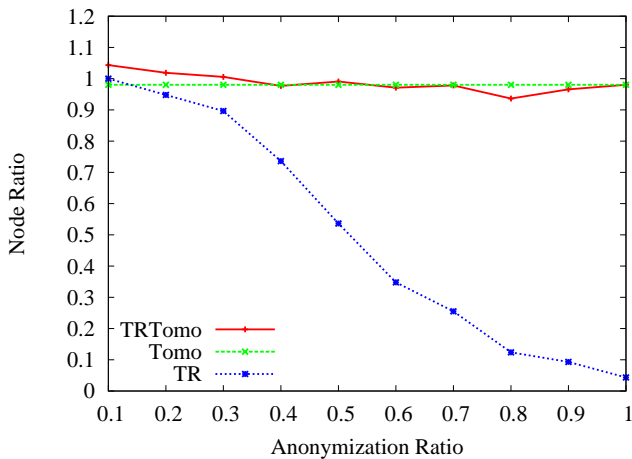


Fig. 7. International-experiment: node ratio of inferred topology.

small but decreases to 0 with increasing anonymization ratio. In contrast, the TRTomo scheme has a node ratio close to 1 in all experiments regardless of anonymization ratios, although it may introduce a few more internal nodes in the inferred tree topology. The node ratio of the Tomo scheme is determined by the TRTomo scheme at anonymization ratio 1.

VII. CONCLUSIONS

In this paper, we proposed a general framework for designing topology inference algorithms based on additive metrics. Our framework allows the integration of both end-to-end packet probing measurements and traceroute type measurements to achieve best accuracy. Based on the framework we designed several computationally efficient topology inference algorithms. In particular, we proposed a novel sequential topology inference algorithm to address the probing scalability problem and handle dynamic node joining and leaving. We demonstrated the effectiveness of the proposed topology inference algorithms via rigorous analysis and Internet experiments. In the future we will study how to utilize the inferred

information and enrich the inference framework for efficient and effective network monitoring and application design.

ACKNOWLEDGMENTS

We would like to thank the anonymous reviewers for their helpful comments and suggestions.

REFERENCES

- [1] PlanetLab, <http://www.planet-lab.org>.
- [2] D. G. Andersen, H. Balakrishnan, M. F. Kaashoek, R. Morris, "Resilient Overlay Networks," *Proc. SOSP 2001*, Oct. 2001.
- [3] D. Antonova, A. Krishnamurthy, Z. Ma, R. Sundaram, "Managing a Portfolio of Overlay Paths," *Proc. NOSSDAV 2004*, Kinsale, Ireland, June 2004.
- [4] D. Barry and J. A. Hartigan, "Asynchronous Distance Between Homogeneous DNA Sequences," *Biometrics*, vol. 43, pp. 261-276, June 1987.
- [5] J.-C. Bolot, "End-to-End Packet Delay and Loss Behavior in the Internet," *Proc. SIGCOMM 93*, Sept. 1993.
- [6] P. Buneman, "The Recovery of Trees from Measures of Dissimilarity," *Mathematics in the Archaeological and Historical Sciences*, Edinburgh University Press, pp. 387-395, 1971.
- [7] R. Caceres, N. G. Duffield, J. Horowitz, D. Towsley, "Multicast-Based Inference of Network-Internal Loss Characteristics," *IEEE Transactions on Information Theory*, vol. 45, no. 7, pp. 2462-2480, Nov. 1999.
- [8] R. Castro, M. Coates, G. Liang, R. Nowak, B. Yu, "Network Tomography: Recent Developments," *Statistical Science*, vol. 19, no. 3, pp. 499-517, 2004.
- [9] J. T. Chang, "Full Reconstruction of Markov Models on Evolutionary Trees: Identifiability and Consistency," *Mathematical Biosciences*, vol. 137, pp. 51-73, 1996.
- [10] M. Coates and R. Nowak, "Network Loss Inference using Unicast End-to-End Measurement," *Proc. ITC Conference on IP Traffic, Modelling and Management*, Monterey, CA, Sept. 2000.
- [11] M. Coates, A. O. Hero III, R. Nowak, B. Yu, "Internet Tomography," *IEEE Signal Processing Magazine*, vol. 19, no. 3, pp. 47-65, May 2002.
- [12] N. G. Duffield, J. Horowitz, F. Lo Presti, "Adaptive Multicast Topology Inference," *Proc. IEEE INFOCOM 2001*, Anchorage, Alaska, Apr. 2001.
- [13] N. G. Duffield, J. Horowitz, F. Lo Presti, D. Towsley, "Multicast Topology Inference From Measured End-to-End Loss," *IEEE Transactions on Information Theory*, vol. 48, no. 1, pp. 26-45, Jan. 2002.
- [14] N. G. Duffield, F. Lo Presti, V. Paxson, D. Towsley, "Network Loss Tomography Using Striped Unicast Probes," *IEEE/ACM Transactions on Networking*, vol. 14, no. 4, pp. 697-710, Aug. 2006.
- [15] J. Hartigan, *Clustering Algorithms*, John Wiley & Sons, 1975.
- [16] J. Ni and S. Tatikonda, "A Markov Random Field Approach to Multicast-Based Network Inference Problems," *Proc. IEEE ISIT 2006*, Seattle, July 2006.
- [17] J. Ni and S. Tatikonda, "Explicit Link Parameter Estimators Based on End-to-End Measurements," *Proc. Allerton Conference on Communication, Control, and Computing*, Sept. 2007.
- [18] J. Ni, H. Xie, S. Tatikonda, Y. R. Yang, "Network Routing Topology Inference From End-to-End Measurements," *Technical Report*, Yale University, 2007.
- [19] F. L. Presti, N. G. Duffield, J. Horowitz, D. Towsley, "Multicast-Based Inference of Network-Internal Delay Distributions," *IEEE/ACM Transactions on Networking*, vol. 10, no. 6, pp. 761-775, Dec. 2002.
- [20] S. Ratnasamy and S. McCanne, "Inference of Multicast Routing Trees and Bottleneck Bandwidths using End-to-end Measurements," *Proc. IEEE INFOCOM 1999*, Mar. 1999.
- [21] N. Saitou and M. Nei, "The Neighbor-Joining Method: A New Method for Reconstruction of Phylogenetic Trees," *Molecular Biology and Evolution*, vol. 4, no. 4, pp. 406-425, 1987.
- [22] Y. Tsang, M. Coates, R. Nowak, "Network Delay Tomography," *IEEE Transactions on Signal Processing*, vol. 51, no. 8, pp. 2125-36, Aug. 2003.
- [23] M. Yajnik, S. Moon, J. Kurose, D. Towsley, "Measurement and Modelling of the Temporal Dependence in Packet Loss," *Proc. IEEE INFOCOM 1999*, Mar. 1999.
- [24] B. Yao, R. Viswanathan, F. Chang, D. Waddington, "Topology Inference in the Presence of Anonymous Routers," *Proc. IEEE INFOCOM 2003*, pp. 353-363, Apr. 2003.