Multicast-based Weight Inference in General Network Topologies

Yilei Lin*, Ting He*, Shiqiang Wang†, Kevin Chan‡, and Stephen Pasteris§
*Pennsylvania State University, University Park, PA 16802, USA. Email: {yjl5282,tzh58}@psu.edu
†IBM T. J. Watson Research Center, Yorktown Heights, NY 10598, USA. Email: wangshiq@us.ibm.com
‡US Army Research Laboratory, Adelphi, MD 20783, USA. Email: kevin.s.chan.civ@mail.mil
§University College London, London WC1E 6EA, UK. Email: s.pasteris@cs.ucl.ac.uk

***Abstract*—Network topology plays an important role in many network operations. However, it is very difficult to obtain the topology of public networks due to the lack of internal cooperation. Network tomography provides a powerful solution that can infer the network routing topology from end-to-end measurements. Existing solutions all assume that routes from a single source form a tree. However, with the rapid deployment of Software Defined Networking (SDN) and Network Function Virtualization (NFV), the routing paths in modern networks are becoming more complex. To address this problem, we propose a novel inference problem, called the weight inference problem, which infers the finest-granularity information from end-to-end measurements on general routing paths in general topologies. Our measurements are based on emulated multicast probes with a controllable "width". We show that the problem has a unique solution when the multicast width is unconstrained; otherwise, we show that the problem can be treated as a sparse approximation problem, which allows us to apply variations of the pursuit algorithms. Simulations based on real network topologies show that our solution significantly outperforms a state-of-the-art network tomography algorithm, and increasing the width of multicast substantially improves the inference accuracy.**

I. Introduction

Topology information is at the foundation of many network operations such as path selection, service placement, overlay construction, and load balancing. Meanwhile, for public networks such as the Internet, it is very hard to obtain the global topology information as such information is distributed across multiple service providers. While there have been several experimental projects to map the Internet, e.g., Skitter [1], Archipelago [2], and Rocketfuel [3], these projects heavily rely on measurement primitives (e.g., `traceroute`) and data feeds (e.g., BGP tables, DNS records), which require cooperation of the target network. However, due to security concerns, an increasing fraction of service providers start to block `traceroute` [4], [5] or even return false measurements [6].

Alternatively, it is known that end-to-end performance measurements can reveal topology information. Techniques known as *network tomography* have been developed to *infer* the network routing topology from end-to-end measurements such as delays and losses, e.g., [7], [8] and followups. However, all the existing solutions are designed for traditional communication networks, where probes from each source follow an (unknown) routing tree.

The tree assumption causes existing network tomography solutions to severely underestimate the complexity of modern communication networks, where technologies like Software Defined Networking (SDN) [9] and Network Function Virtualization (NFV) [10] can generate complex non-tree routing topologies. For example, the generalized forwarding rules in SDN allow probes with the same source and destination to follow different paths, and the requirement of service chains in NFV can cause certain probes to deviate from their default routing paths. This triggers a research question: *Can we still infer useful topology information from end-to-end measurements in networks with general (possibly non-tree) routing topologies?*

In this work, we take a first step towards answering this question by inferring the network's internal performance at the "finest granularity" (see Section II-C), which provides valuable information about the routing topology.

A. Related Work

Network (topology) tomography was initially studied based on multicast probing [7], [8], where correlation among probes is used to infer the multicast tree. Unicast-based solutions were also developed, using stripes of back-to-back unicast probes [11], [12] or "sandwiches" of small and large probes [13]. It was shown in [14] that one can use stripes of unicast probes to emulate multicast.

Only a few works considered non-tree topologies. Solutions in [15], [16] still constructed trees, except that the accuracy was analyzed with respect to a non-tree ground truth. Solutions in [17], [18], [19], [20] merged 2-by-2 topologies (i.e., *quartets*) depicting the connections between two sources and two destinations, and a similar idea was used in [21] by merging 1-by-3 topologies. *However, all these solutions assumed that routes from/to each node form a tree.*

B. Summary of Contributions

To our knowledge, we are the first to investigate network tomography for general topologies under general routing. Our main contributions are:

1) We characterize the finest-granularity information, called

This research was sponsored by the U.S. Army Research Laboratory and the U.K. Ministry of Defence under Agreement Number W911NF-16-3-0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, the U.K. Ministry of Defence or the U.K. Government. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

category weights (see Section II-C), that can be uniquely determined from end-to-end measurements based on multicast.
2) We show that (i) if a multicast can involve all the paths (i.e., broadcast), then the solution of category weights is unique; (ii) otherwise, the solution is not unique, but always has a sparse approximation.
3) Our simulations show that (i) our solution based on sparse approximation can significantly outperform a state-of-the-art network tomography algorithm, and (ii) increasing the "width" of multicast significantly improves the inference accuracy.

Roadmap. Section II formulates our problem. Section III presents our multicast-based solution and analyzes its properties. Section IV evaluates our solution against a state-of-the-art algorithm. Finally, Section V concludes the paper.

II. Problem Formulation

A. Network Model

We model the network (routing) topology as an edge-weighted directed graph $\mathcal{G} = (V, E)$. Each vertex $v \in V$ represents a probing source, a probing destination, or a branching/joining point between multiple measurement paths. Each edge $e \in E$ represents a connection between two vertices, which may map to a sequence of links. Given an edge e, let u_e be its weight which can represent various performance metrics as defined below. We use "edge" to refer to a logical link (i.e., a connection), and "link" to refer to a physical link.

In this work, we focus on *loss-based weight*. Specifically, let α_e be the probability for a probe to successfully traverse edge e. Then its weight is defined as $u_e := -\log \alpha_e$ [11]. This weight has the properties that (i) it is non-negative, and (ii) the sum weight over edges on a path or a set of simultaneously probed paths can be measured, as explained in Section II-B. This definition can be extended to other performance metrics, e.g., by replacing α_e by the no-queueing probability or the congestion-free probability.

B. Observation Model

We measure the network from a given probing source s, which can send probes on n different paths, e.g., using different combinations of the header fields[1]. Let $\{p_1, \ldots, p_n\}$ denote the entire set of measurement paths. To model generalized forwarding (in SDN) and service chain traversal (in NFV), we allow the paths to be non-simple, i.e., a path can traverse the same edge/vertex multiple times [22].

It is well-known [7], [8] that multicast probing is particularly useful for topology inference. However, IP multicast is not widely deployed, and SDN offers no support of multicast out of the box [23]. Nevertheless, existing studies [11], [12] have shown that stripes of k unicast probes sent back to back on k paths can emulate multicast on these paths (solutions therein are limited to $k = 2$). In this work, we use this trick to emulate multicast, where $k \in \{1, \ldots, n\} =: [n]$ is a design parameter. In the sequel, we call such an emulated multicast a "k-cast", and the parameter k the "width" of the multicast. A "probe" refers to a k-cast probe, emulated by k back-to-back unicast probes. For simplicity, we assume that all the unicast probes in a k-cast probe experience the same performance at shared edges. In practice, this is usually a good approximation for small k, as the total duration of transmitting k unicast probes is typically much smaller than the duration of network congestion events [14]. We leave evaluation of the approximation error to future work.

The power of k-cast is that it allows us to measure the joint success probability on up to k paths. Let X_C ($C \subseteq [n]$, $0 < |C| \leq k$) be the indicator that all the unicast probes sent back to back on paths $\{p_i : i \in C\}$ successfully reach their destinations. Under the assumption that different edges exhibit independent losses, we have

$$\phi_C := -\log(\Pr\{X_C = 1\}) = -\log(\prod_{e \in \bigcup_{i \in C} p_i} \alpha_e) = \sum_{e \in \bigcup_{i \in C} p_i} u_e. \quad (1)$$

We define ϕ_C as the *cast weight* of a $|C|$-cast on paths $\{p_i : i \in C\}$. As we can estimate $\Pr\{X_C = 1\}$ by the fraction of joint successes on paths $\{p_i : i \in C\}$ among all the k-cast probes covering these paths, we can estimate ϕ_C consistently, i.e., the estimated value converges to the true value as the number of probes goes to infinity. Let $\mathcal{C} := \{C \subseteq [n] : 0 < |C| \leq k\}$ be the subsets of paths for which the cast weights can be measured.

C. Weight Inference Problem

We are interested in inferring the edge weights from the measured cast weights. However, instead of specifying the weights of individual edges, the measurements can only specify the weights at the level of *(edge) categories*. Each category is a subset of edges, all traversed by the same set of measurement paths, and we use the set of path indices to index the category. That is, for $A \subseteq [n]$ and $A \neq \emptyset$, category Γ_A is defined as $\{e \in E : e \in p_i \text{ iff } i \in A\}$. By this definition, we have a total of $2^n - 1$ categories, which form a partition of E. Let $\mathcal{A} := 2^{[n]} \setminus \{\emptyset\}$ (where $2^{[n]}$ denotes the power set of $[n]$). Let w_A denote the sum of the weights of the edges in category Γ_A, referred to as *category weight*.

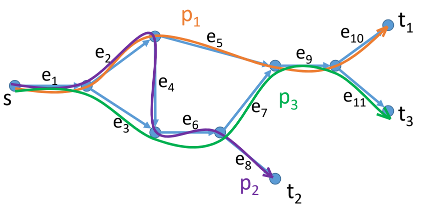
Definition 1. *The* weight inference problem *aims at inferring the category weights* $(w_A)_{A \in \mathcal{A}}$ *from the measured cast weights* $(\phi_C)_{C \in \mathcal{C}}$.

The reason for targeting at category weights is that (i) they represent the *finest granularity of information* that can be inferred about edge weights, as the end-to-end performances are invariant to variations in individual edge weights as long as the category weights are fixed (as indicated by (2)), and (ii) their relationship to the measurements is known even if the topology is unknown. Specifically, by definition, we have

$$\sum_{A \in \mathcal{A} : A \cap C \neq \emptyset} w_A = \phi_C, \quad \forall C \in \mathcal{C}. \quad (2)$$

Moreover, category weights provide valuable information about the topology. If $w_A > 0$, the paths in $\{p_i : i \in A\}$ must share at least one edge. We can thus infer which paths

[1] While it is possible for different headers to result in the same path, we can easily detect this by sending back-to-back unicast probes with each of the combinations in the header and measuring similarity in their performances.



(a) ground truth topology

e_1:	$u_{e_1} = 0.0175$	{1,2,3}
e_2:	$u_{e_2} = 0.0071$	{1,2}
e_3:	$u_{e_3} = 0.0094$	{3}
e_4:	$u_{e_4} = 0.0421$	{2}
e_5:	$u_{e_5} = 0.0193$	{1}
e_6:	$u_{e_6} = 0.0363$	{2,3}
e_7:	$u_{e_7} = 0.0478$	{3}
e_8:	$u_{e_8} = 0.0394$	{2}
e_9:	$u_{e_9} = 0.0066$	{1,3}
e_{10}:	$u_{e_{10}} = 0.0247$	{1}
e_{11}:	$u_{e_{11}} = 0.0222$	{3}

(b) edge weight and edge category

w_1=0.0440
w_2=0.0815
w_3=0.0794
$w_{1,2}$=0.0071
$w_{1,3}$=0.0066
$w_{2,3}$=0.0363
$w_{1,2,3}$=0.0175

(c) category weight

$\phi_1 = w_1 + w_{1,2} + w_{1,3} + w_{1,2,3} = 0.0752$
$\phi_2 = w_2 + w_{1,2} + w_{2,3} + w_{1,2,3} = 0.1424$
$\phi_3 = w_3 + w_{1,3} + w_{2,3} + w_{1,2,3} = 0.1398$
$\phi_{1,2} = w_1 + w_2 + w_{1,2} + w_{1,3} + w_{2,3} + w_{1,2,3} = 0.1930$
$\phi_{1,3} = w_1 + w_3 + w_{1,2} + w_{1,3} + w_{2,3} + w_{1,2,3} = 0.1909$
$\phi_{2,3} = w_2 + w_3 + w_{1,2} + w_{1,3} + w_{2,3} + w_{1,2,3} = 0.2284$
$\phi_{1,2,3} = w_1 + w_2 + w_3 + w_{1,2} + w_{1,3} + w_{2,3} + w_{1,2,3} = 0.2724$

(d) cast weight ($k = 3$)

Fig. 1. An example network and the corresponding weight inference problem (inferring (c) from (d)).

share edges, which is very useful in constructing overlay paths (e.g., in CDN) and planning backup paths, where disjoint paths are desired to maximize throughput or resilience. There are also algorithms to construct topologies that can explain all the measurements based on category weights [24].

Example: Fig. 1 gives an illustrative example. The ground truth topology shown in Fig. 1 (a) has 10 vertices and 11 edges. Suppose that source s can measure 3 paths in this network, marked as $p_1, \ldots, p_3$. The weight and category of each edge are listed in Fig. 1 (b). For example, $u_{e_1} = 0.0175$ means that packets have $e^{-0.0175} = 98.27\%$ chance to successfully traverse edge e_1 without being lost. Since p_1, p_2, and p_3 all traverse edge e_1, the category for edge e_1 is $\{1, 2, 3\}$. The category weights we want to infer are shown in Fig. 1 (c). For instance, $w_1 = u_{e_5} + u_{e_{10}} = 0.0440$, which is the sum weight of all the edges in category $\{1\}$. Fig. 1 (d) lists all the cast weights that can be measured by 3-cast (ignoring measurement error), and their relationship to the category weights according to (2). A few observations are in order:

First, the linear system in Fig. 1 (d) has a full rank, and hence we can uniquely determine the category weights from 3-cast. *Second*, no tree-based solution can reconstruct these category weights, as at most one of $w_{1,2}$, $w_{1,3}$, and $w_{2,3}$ can be non-zero in any rooted tree with three leaves. *Moreover*, if we send bi-cast (i.e., $k = 2$), we can only measure ϕ_1, ϕ_2, ϕ_3, $\phi_{1,2}$, $\phi_{1,3}$, and $\phi_{2,3}$. The linear system will not have a unique solution, and could give a feasible solution like $(w_1, w_2, w_3, w_{1,2}, w_{1,3}, w_{2,3}, w_{1,2,3})$ $= (0.0506, 0.0881, 0.0860, 0.0005, 0, 0.0297, 0.0241)$, which differs from the ground truth.

III. Weight Inference Algorithms

Written in vector form, the weight inference problem aims at solving the linear system

$$\boldsymbol{D} \cdot \boldsymbol{w} = \boldsymbol{\phi} \tag{3}$$

under the constraint

$$\boldsymbol{w} \geq \boldsymbol{0}, \tag{4}$$

where $\boldsymbol{w} := (w_A)_{A \in \mathcal{A}}$, $\boldsymbol{\phi} := (\phi_C)_{C \in \mathcal{C}}$, and $\boldsymbol{D} := (\mathbb{1}_{A \cap C \neq \emptyset})_{C \in \mathcal{C}, A \in \mathcal{A}}$ ($\mathbb{1}$.: indicator function).

When $k = n$, the linear system has a unique solution.

Theorem III.1. *If $k = n$, then (3) has a unique solution.*

Proof. We will show that each w_A ($A \in \mathcal{A}$) is uniquely determined by $\boldsymbol{\phi}$ by an induction on $|A|$.

For $|A| = 1$, i.e., $A = \{i\}$ ($i \in [n]$), (3) implies that

$$w_{\{i\}} = \phi_{[n]} - \phi_{[n]\setminus\{i\}}. \tag{5}$$

For $|A| = m > 1$, suppose that $w_{A'}$ is uniquely determined by $\boldsymbol{\phi}$ for all $A' \in \mathcal{A}$ with $|A'| \leq m - 1$. By (3), we have

$$\sum_{A' \in \mathcal{A}: A' \subseteq A} w_{A'} = \phi_{[n]} - \phi_{[n]\setminus A}. \tag{6}$$

By induction, terms on the left-hand side of (6) except for w_A are all uniquely determined by $\boldsymbol{\phi}$. Hence, w_A is also uniquely determined by $\boldsymbol{\phi}$. This completes the proof. □

When $k < n$, the linear system (3) is underdetermined, and hence there is no unique solution. To resolve the ambiguity, we adopt the principle of *Occam's razor*, i.e., preferring the "simplest" solution over alternatives. In our context, the simplest solution is the one with the fewest non-zero category weights. As the number of non-zero-weight categories gives a lower bound on the size of the inferred topology [24], this objective helps to find the simplest topology that can explain the measurements, and is consistent with the objectives in [13], [21]. Then the weight inference problem becomes[2]

$$\min \ \|\boldsymbol{w}\|_0 \tag{7a}$$

$$\text{s.t. } (3), (4). \tag{7b}$$

Problem (7) is an instance of the *sparse approximation problem with noiseless observations* [25]. Here we assume that sufficiently many probes have been sent to accurately measure the cast weights. Otherwise, we can easily incorporate measurement errors by relaxing (3) into an error bound.

Since such problems are generally NP-hard [26], approximate solutions have been proposed. A popular solution is *basis pursuit (BP)*, which aims to minimize the ℓ-1 norm, i.e., $\min \|\boldsymbol{w}\|_1$. As the category weights are non-negative, this objective becomes minimizing the sum weight over all the categories, i.e., $\min \sum_{A \in \mathcal{A}} w_A$, which is a linear program (LP). We can thus apply existing LP solvers. In particular, we find that the *simplex method* provides a guaranteed sparsity.

Theorem III.2. *BP based on the simplex method provides a solution to (7) with no more than $\sum_{i=1}^{k} \binom{n}{i}$ non-zero entries, i.e., the solution is $O(n^k)$-sparse if $k = O(1)$.*

Proof. Consider the polytope that forms the feasible region of the problem, defined by (3), (4). Each vertex of the polytope must satisfy $2^n - 1$ constraints with equality. Since there are only $\sum_{i=1}^{k} \binom{n}{i}$ constraints in (3), at least $2^n - 1 - \sum_{i=1}^{k} \binom{n}{i}$

[2] Here $\|\cdot\|_q$ ($q \geq 0$) denotes the ℓ-q norm.

Algorithm 1 Non-negative Matching Pursuit

```
1: Initialization: $\boldsymbol{w} = \boldsymbol{0}$ and $\boldsymbol{r} = \boldsymbol{\phi}$
2: while $\max \boldsymbol{D}^T \boldsymbol{r} > 0$ do
3:     $l \leftarrow \arg\max \boldsymbol{D}^T \boldsymbol{r}$
4:     $w_l \leftarrow w_l + \boldsymbol{D}_l^T \boldsymbol{r}$
5:     $\boldsymbol{r} \leftarrow \boldsymbol{\phi} - \boldsymbol{D}\boldsymbol{w}$
6: end while
```

of these equations are in the form of $w_A = 0$. Hence, each vertex has at most $\sum_{i=1}^{k} \binom{n}{i}$ non-zero entries. The conclusion follows from the fact that the solution found by the simplex method always lies on a vertex of the polytope. □

The main challenge in applying this solution is the complexity. Since the linear system (3) has $2^n - 1$ variables, the worst-case complexity of the simplex method is exponential in $2^n - 1$, which is *super-exponential* in the number of measurement paths n.

To reduce the complexity, we borrow from the greedy heuristics for sparse approximation problems, known as *matching pursuit (MP)* and *orthogonal matching pursuit (OMP)*, both iteratively finding non-zero entries one at a time. By default, however, these algorithms can produce negative solutions, as the original sparse approximation problem does not have the nonnegativity constraint (4). Only a couple of works have discussed how to extend the pursuit algorithms for non-negative sparse approximation problems [27], [28]. Below we summarize them in the context of our problem.

Non-negative matching pursuit (nMP): Just like the original MP, nMP is a greedy algorithm that iteratively updates one variable at a time, as in Algorithm 1. Here $\boldsymbol{r}$ denotes the residual $\boldsymbol{\phi} - \boldsymbol{D}\boldsymbol{w}$, and $\boldsymbol{D}_l$ denotes the l-th column in $\boldsymbol{D}$ (called an *atom*). The difference from MP is that instead of selecting the most correlated atom, i.e., $l \leftarrow \arg\max |\boldsymbol{D}^T \boldsymbol{r}|$ ($|\cdot|$ takes absolute values for each element), nMP selects the most *positively* correlated atom as in line 3 of Algorithm 1.

Non-negative orthogonal matching pursuit (nOMP): MP has several drawbacks, e.g., the solution may not provide the best approximation using the selected atoms, and the same atom may be selected repeatedly, which slows down the convergence. OMP is designed to eliminate these issues at the cost of more complex computation, by computing an orthogonal projection onto the selected atoms using least square programming. Similarly, as shown in Algorithm 2, nOMP updates the solution using non-negative least square programming $\arg\min_{\boldsymbol{x} \geq \boldsymbol{0}} \|\boldsymbol{\phi} - \boldsymbol{D}_L \boldsymbol{x}\|_2$ (line 5 of Algorithm 2), where L denotes the set of indices of the selected atoms, $\boldsymbol{D}_L$ the sub-matrix of $\boldsymbol{D}$ formed by these atoms, and $\boldsymbol{w}_L$ the sub-vector of $\boldsymbol{w}$ with indices in L.

Remark: We note that since our problem (7) has $O(2^n)$ variables, even greedy algorithms like nMP and nOMP have a complexity that is exponential in n, the number of measurement paths. This is likely to be the inherent complexity of our problem, as in the language of sparse approximation, our problem has an exponentially large "dictionary".

Algorithm 2 Non-negative Orthogonal Matching Pursuit

```
1: Initialization: $L = \emptyset$, $\boldsymbol{w} = \boldsymbol{0}$ and $\boldsymbol{r} = \boldsymbol{\phi}$
2: while $\max \boldsymbol{D}^T \boldsymbol{r} > 0$ do
3:     $l \leftarrow \arg\max \boldsymbol{D}^T \boldsymbol{r}$
4:     $L \leftarrow L \cup \{l\}$
5:     $\boldsymbol{w}_L \leftarrow \arg\min_{\boldsymbol{x} \geq \boldsymbol{0}} \|\boldsymbol{\phi} - \boldsymbol{D}_L \boldsymbol{x}\|_2$
6:     $\boldsymbol{r} \leftarrow \boldsymbol{\phi} - \boldsymbol{D}_L \boldsymbol{w}_L$
7: end while
```

IV. Performance Evaluation

We evaluate the performance of the proposed algorithms (BP based on the simplex method, nMP, and nOMP) against a state-of-the-art topology inference algorithm in terms of their performance for the weight inference problem.

Benchmark: As we are the first to study the weight inference problem, we use a state-of-the-art topology inference algorithm as the benchmark, and deduce the category weights from the inferred topology and paths. The algorithm is called *Rooted Neighbor-Joining (RNJ)* [11], which infers a tree topology using bi-cast probing, and is guaranteed to be accurate when the ground truth topology is a canonical tree.

Simulation setting: Our simulation is conducted on Internet Service Provider (ISP) topologies from the Rocketfuel project [3], which are router-level topologies collected from diverse ISPs. In our experiment, we choose topology AS6461, which represents the ISP Abovenet in US with 182 vertices and 294 edges. The weight of each edge is uniformly distributed in $[0.005, 0.05]$, which means that the success rate of each edge ranges from 95.12% to 99.50%.

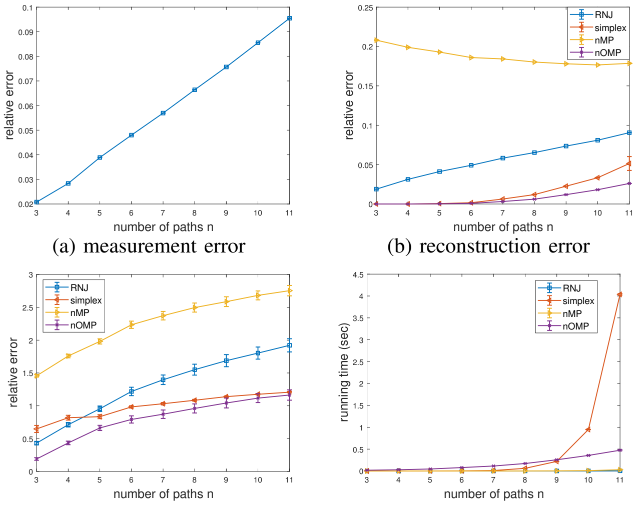
In our simulation, we randomly choose destinations from nodes with degree ≤ 2, and the source from nodes with degree ≥ 6. To create sufficiently complex paths, we randomly choose a sequence of 8 to 10 nodes with degree ≥ 6 (excluding the source node) as waypoints a path must traverse (in the specified order), which can represent locations of the required network functions. We use Dijkstra's algorithm to route between consecutive waypoints, which generates piecewise shortest paths with an average length of 30 hops. All our results are averaged over 300 Monte Carlo runs.

All algorithms are implemented in Matlab. The non-negative least square program in nOMP (line 5 of Algorithm 2) is solved by CPLEX.

Performance measure: We measure the performance of inferring $\boldsymbol{\rho}$ by $\hat{\boldsymbol{\rho}}$ by the relative error, defined as $\|\hat{\boldsymbol{\rho}} - \boldsymbol{\rho}\|_2 / \|\boldsymbol{\rho}\|_2$. If $\boldsymbol{\rho}$ is the vector of true cast weights and $\hat{\boldsymbol{\rho}}$ is the estimated value, this is the *measurement error*[3]. If $\boldsymbol{\rho}$ is the vector of estimated cast weights and $\hat{\boldsymbol{\rho}}$ is the value computed from the inferred category weights by (2), this is the *reconstruction error*. If $\boldsymbol{\rho}$ is the vector of true category weights and $\hat{\boldsymbol{\rho}}$ is the inferred value, this is the *inference error*.

Bi-cast accuracy: We first set $k = 2$ (bi-cast) to compare our algorithms with RNJ. We fix the number of bi-cast probes to 10000 and vary the number of paths n. The probes are

[3] Actually, the measurement error is the error in estimating cast weights from raw measurements. We refer to it as the "measurement error" to differentiate from the "inference error" in estimating the category weights.



(a) measurement error (b) reconstruction error

(c) inference error (d) running time

Fig. 2. Relative error as the number of paths n varies ($k = 2$, 10000 k-cast probes).

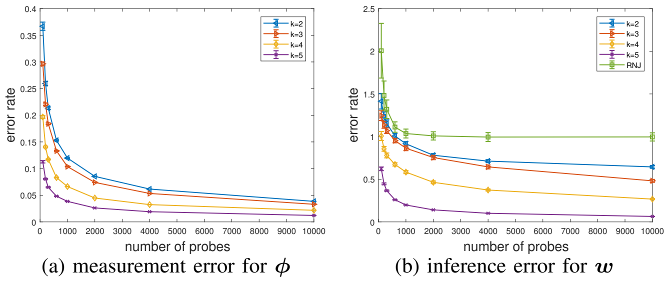
evenly distributed among the $\binom{n}{2}$ path pairs. To solve (7), we apply BP (simplex), nMP, and nOMP, respectively.

Fig. 2 (a) shows that the error of estimating the true cast weights $\boldsymbol{\phi}$ from raw measurements (i.e., loss indicators) increases with the increase of n, due to the decrease in the number of probes per pair of paths. Note that the measurement error is independent of the weight inference algorithm.

After we infer the category weights $\hat{\boldsymbol{w}}$ from the measured cast weights $\hat{\boldsymbol{\phi}}$, we compare the cast weights $\widetilde{\boldsymbol{\phi}}$ reconstructed from $\hat{\boldsymbol{w}}$ to the input $\hat{\boldsymbol{\phi}}$ in Fig. 2 (b). We find that: (i) our proposed nOMP approach performs the best in reconstruction, followed by the simplex method, RNJ, and nMP; (ii) none of these methods reconstruct the measured cast weights perfectly, because measurement error prevents the linear system from having a feasible solution. The poor performance of nMP and RNJ is because nMP has known limitations (see discussions about Algorithm 2), and RNJ is limited to tree topologies, resulting in missing categories.

Fig. 2 (c) shows the error in inferring the category weights $\boldsymbol{w}$. Our proposed nOMP approach has the highest accuracy among all the evaluated algorithms, followed closely by the simplex method. nMP still performs poorly. RNJ achieves a higher accuracy, especially when n is very small (e.g., 3 or 4). This is because in this case the paths are likely to form a tree, and RNJ can infer the category weights in trees correctly. As expected, the inference error increases with the number of paths for all the algorithms due to the increasing error in the input (Fig. 2 (a)).

Fig. 2 (d) shows that the simplex method differs significantly from nOMP in running time. As n grows, the running time of the simplex method grows much faster than the other algorithms. This makes nOMP more attractive as it can achieve similar accuracy with a much shorter running time[4].



(a) measurement error for $\boldsymbol{\phi}$ (b) inference error for $\boldsymbol{w}$

Fig. 3. Relative error as k and the number of k-cast probes vary ($n = 5$).

K-cast accuracy: We then evaluate the impact of multicast width k and the rate of convergence with respect to the number of probes. In this simulation, we fix $n = 5$ and vary k from 2 to 5. We send 10000 k-cast probes for each value of k, evenly distributed among all subsets containing k paths. At 100, 200, 300, 600, 1000, 2000, 4000 and 10000 probes, we estimate the cast weights $\boldsymbol{\phi}$, and use the estimates to infer the category weights $\boldsymbol{w}$ by nOMP method. For $k = 2$, we also compare our results with the category weights inferred by RNJ.

Fig. 3 (a) shows the error in estimating the cast weights $\boldsymbol{\phi}$ (i.e., measurement error). Given the number of k-cast probes, the measurement error decreases as we increase k. For example, the error is below 5% if we send 1000 5-cast probes, but it is above 10% if we send 1000 2-cast or 3-cast probes. This is because the absolute number of packets sent on each path increases (linearly) with k.

Fig. 3 (b) shows the error of inferring category weights $\boldsymbol{w}$. These curves show the same trend as the curves in Fig. 3 (a), but with much larger gaps. In particular, the error remains large when $k < n$, even with a very large number of probes, since in these cases the weight inference problem does not have a unique solution. While the proposed algorithms can find a feasible solution, it may not be the closest approximation to the ground truth. Nevertheless, increasing k still helps to reduce the error due to having more constraints. When $k = n$, we have a unique solution, which equals the ground truth if there is no measurement error. With measurement error, this plot shows that n-cast still significantly outperforms the other k-casts ($k < n$), e.g., 5-cast reduces the error of 4-cast by 80% at 10000 probes. These plots show the importance of (emulating) broadcast in inferring general topologies.

Breakdown of error: From the previous simulation, we have seen that increasing k leads to a more accurate estimation of $\boldsymbol{\phi}$. On one hand, when k increases, we send more packets in one k-cast probe, which means that we actually collect more information from each path, resulting in the increase of the accuracy. On the other hand, when k increases, the dimension of $\boldsymbol{\phi}$ increases. For instance, when $k = 2$ and $n = 5$, the dimension of $\boldsymbol{\phi}$ is 15. However, if $k = 5$ and $n = 5$, the dimension of $\boldsymbol{\phi}$ becomes 31. The increase of the dimension implies more unknown values to estimate. To understand the exact impact of k, we analyze the measurement error for each type of cast weights. To this end, we define a vector $\boldsymbol{\psi}_i := (\phi_A)_{|A|=i}$ to represent the weights of i-casts. For $k \geq i$, each k-cast probe also provides an i-cast measurement for each subset of i of the k probed paths, and hence we can

[4] For a fair comparison with the Matlab-based implementation of the other algorithms, we used a Matlab-based implementation of the simplex method available at `https://www.12000.org/my_notes/simplex/index.htm`. Although faster implementation exists, the simplex method will eventually be the slowest due to its super-exponential complexity.

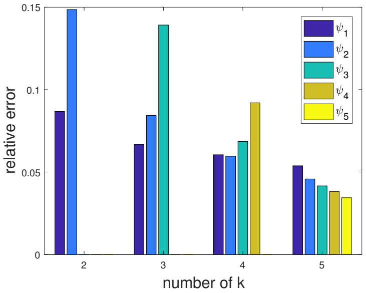


Fig. 4. Breakdown of measurement error ($n = 5$, 1000 k-cast probes).

TABLE I
Number of Probes for Each Element in ψ_i

	ψ_1	ψ_2	ψ_3	ψ_4	ψ_5
2-cast	400	100	0	0	0
3-cast	600	300	100	0	0
4-cast	800	600	400	200	0
5-cast	1000	1000	1000	1000	1000

estimate ψ_i from k-cast probes for any $k \geq i$. We break down the measurement error into the errors in estimating each ψ_i ($i \leq k$), as shown in Fig. 4. The variation in these errors is mainly caused by the different numbers of probes used to estimate each element in ψ_i. To be precise, Table I gives the number of (effective) i-cast probes for estimating each element in ψ_i.

We see that as k increases, the error in estimating each ψ_i decreases, accompanied by an increase in the number of i-cast probes. Our results indicate that the error for estimating each ψ_i strongly depends on the number of probes.

V. Conclusion

Motivated by the complex forwarding behaviors in SDN and NFV, we revisit the problem of inferring routing topologies from end-to-end measurements, under a new assumption that the underlying routing paths may not follow routing trees. As a first step towards solving this problem, we formulate the weight inference problem to infer the finest-granularity information about edge weights from multicast probes. Modeling the problem as a linear system, we show that the problem has a unique solution when using unconstrained multicast (i.e., broadcast), and a sparse approximation when using constrained multicast. Our empirical evaluations show that applying sparse approximation algorithms to our problem can yield much higher reconstruction and inference accuracy than a state-of-the-art network tomography algorithm. We also find that increasing the width of multicast can significantly improve the inference accuracy. Our result is in sharp contrast to the existing result on inferring tree topologies, where bi-cast probing suffices for reconstructing the ground truth [11].

References

[1] K. Claffy and S. McCreary, "Caida skitter project." [Online]. Available: https://www.caida.org/tools/measurement/skitter/

[2] CAIDA, "Archipelago measurement infrastructure," 2015. [Online]. Available: http://www.caida.org/projects/ark/

[3] N. Spring, R. Mahajan, D. Wetherall, and T. Anderson, "Measuring isp topologies with rocketfuel," *IEEE/ACM Transactions on Networking*, vol. 12, no. 1, pp. 2–16, February 2004.

[4] B. Yao, R. Viswanathan, F. Chang, and D. Waddington, "Topology inference in the presence of anonymous routers," in *INFOCOM*. IEEE, 2003.

[5] M. H. Gunes and K. Sarac, "Resolving IP aliases in building traceroute-based internet maps," *IEEE/ACM Transactions on Networking*, vol. 17, no. 6, pp. 1738–1751, December 2009.

[6] S. T. Trassare, R. Beverly, and D. Alderson, "A technique for network topology deception," in *MILCOM*. IEEE, 2013.

[7] R. Caceres, N. G. Duffield, J. Horowitz, F. L. Presti, and D. Towsley, "Loss-based inference of multicast network topology," in *IEEE CDC*, 1999.

[8] S. Ratnasamy and S. McCanne, "Inference of multicast routing trees and bottleneck bandwidths using end-to-end measurements," in *IEEE INFOCOM*, 1999.

[9] "Software-defined networking: the new norm for networks," Open Networking Foundation White Paper, April 2012.

[10] "Network Functions Virtualisation — Introductory White Paper," White Paper, ETSI, 2012. [Online]. Available: https://portal.etsi.org/nfv/nfv_white_paper.pdf

[11] J. Ni, H. Xie, S. Tatikonda, and Y. R. Yang, "Efficient and dynamic routing topology inference from end-to-end measurements," *IEEE/ACM Transactions on Networking*, vol. 18, no. 1, pp. 123–135, February 2010.

[12] J. Ni and S. Tatikonda, "Network tomography based on additive metrics," *IEEE Transactions on Information Theory*, vol. 57, no. 12, pp. 7798–7809, December 2011.

[13] M. Coates, R. Castro, M. Gadhiok, R. King, Y. Tsang, and R. Nowak, "Maximum likelihood network topology identification from edge-based unicast measurements," in *ACM SIGMETRICS*, June 2002.

[14] N. Duffield, F. L. Presti, V. Paxson, and D. Towsley, "Network loss tomography using striped unicast probes," *IEEE/ACM Transactions on Networking*, vol. 14, no. 4, pp. 697–710, August 2006.

[15] A. Krishnamurthy and A. Singh, "Robust multi-source network tomography using selective probes," in *IEEE INFOCOM*, March 2012.

[16] V. Ramasubramanian, D. Malkhi, F. Kuhn, M. Balakrishnan, A. Gupta, and A. Akella, "On the treeness of internet latency and bandwidth," in *ACM SIGMETRICS*, June 2009.

[17] M. Rabbat, R. Nowak, and M. Coates, "Multiple source, multiple destination network tomography," in *IEEE INFOCOM*, 2004.

[18] M. Rabbat, M. Coates, and R. Nowak, "Multiple source Internet tomography," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 12, pp. 2221–2234, December 2006.

[19] A. Anandkumar, A. Hassidim, and J. Kelner, "Topology discovery of sparse random graphs with few participants," in *ACM SIGMETRICS*, June 2011.

[20] P. Sattari, C. Fragouli, and A. Markopoulou, "Active topology inference using network coding," *Physical Communication*, vol. 6, pp. 142–163, March 2013.

[21] A. Sabnis, R. K. Sitaraman, and D. Towsley, "OCCAM: An optimization based approach to network inference," in *The Workshop on MAthematical performance Modeling and Analysis (MAMA)*, June 2018.

[22] T. Kuo, B. Liou, K. C. Lin, and M. Tsai, "Deploying chains of virtual network functions: On the relation between link and server usage," in *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, April 2016, pp. 1–9.

[23] S. Islam, N. Muslim, and J. Atwood, "A survey on multicasting in software-defined networking," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 1, pp. 355–387, 2018.

[24] Y. Lin, T. He, S. Wang, K. Chan, and S. Pasteris, "Looking glass of NFV: Inferring the structure and state of NFV network from external observations," in *IEEE INFOCOM*, April 2019.

[25] D. Donoho, "For most large underdetermined systems of linear equations the minimal l1-norm solution is also the sparsest solution," *Communications on Pure and Applied Mathematics*, vol. 56, no. 6, pp. 797–829, 2006.

[26] B. Natarajan, "Sparse approximate solutions to linear systems," *SIAM Journal on Computing*, vol. 24, no. 2, pp. 227–234, April 1995.

[27] A. Bruckstein, M. Elad, and M. Zibulevsky, "Sparse non-negative solution of a linear system of equations is unique," in *IEEE ISCCSP*, March 2008.

[28] M. Yaghoobi, D. Wu, and M. Davies, "Fast non-negative orthogonal matching pursuit," *IEEE Signal Processing Letters*, vol. 22, no. 9, pp. 1229–1233, September 2015.