Looking Glass of NFV: Inferring the Structure and State of NFV Network from External Observations

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

***Abstract*—The rapid development of network function virtualization (NFV) enables a communication network to provide in-network services using virtual network functions (VNFs) deployed on general IT hardware. While existing studies on NFV focused on how to provision VNFs from the provider's perspective, little is known about how to validate the provisioned resources from the user's perspective. In this work, we take a first step towards this problem by developing an inference framework designed to "look into" the NFV network. Our framework infers the structure and state of the overlay formed by VNF instances, ingress/egress points of measurement flows, and critical points on their paths (branching/joining points). Our solution only uses external observations such as the required service chains and the end-to-end performance measurements. Besides the novel application scenario, our work also fundamentally advances the state of the art on topology discovery by considering (i) general topologies with general measurement paths, and (ii) information of service chains. Evaluations based on real network topologies show that the proposed solution significantly improves the accuracy over existing solutions, and service chaining information is critical in revealing the structure of the underlying topology.**

I. Introduction

Modern communication networks have outgrown empty bit pipes. Increasingly, network providers use network appliances (a.k.a. middleboxes) to provide in-network services, e.g., Network Address Translators (NATs), firewalls, Intrusion Detection Systems (IDSs), Intrusion Prevention Systems (IPSs), Deep Packet Inspectors (DPIs), web proxies, and WAN optimizers [1]. While traditionally deployed as physical middleboxes implemented by special-purpose hardware, next-generation network appliances are increasingly deployed as software middleboxes, referred to as *Virtual Network Functions (VNFs)*, running on general-purpose servers. This technology, known as Network Function Virtualization (NFV) [2], is empowering network providers to partner with cloud providers and software vendors to provide innovative value-adding services within the communication network [3].

On one hand, NFV opens up a whole new solution space for configuring the network. Encapsulated as virtual machine (VM) instances, VNFs can be scaled up/down, replicated, and/or migrated to suit the current demands. Moreover, multiple VNFs can be organized into a chain (a.k.a. *service chain*) to serve flows with multiple processing needs. Solutions have been developed to exploit the enlarged solution space from the provider's perspective, by optimizing the placement of VNFs [4], the routing among VNFs [5], or a combination of these actions [6], [7].

On the other hand, the presence of (virtual or physical) network appliances significantly complicates network management. Due to the widespread deployment of network appliances, the network administrator needs to manage not only routers and switches, but also a variety of network appliances, leading to high operational expenses and administrative headaches [1]. The problem remains even with the virtualization of network appliances, as the network administrator still needs to manage VNFs based on software that is often developed by independent vendors [3]. Furthermore, as NFV becomes widely adopted by network providers, there will be needs for a client (network administrator) to validate the service received from its network provider, or for a network provider to validate the service received from its peers, just as in today's Internet. It is therefore highly desirable to have a method that can "look into" the NFV network without directly measuring individual routers or VNF instances.

In this work, we take a first step towards addressing this problem by jointly inferring the internal structure and state of an NFV network using external observations. We consider two types of observations: (i) parameters of flow demands (e.g., ingress/egress points and service chains) and (ii) end-to-end performance measurements (e.g., delays and losses). While these observations do not directly specify the physical network topology, we argue that they can provide useful information about the *VNF overlay*, such as: the deployment of VNF instances, the chaining of these instances for each flow, and the performance of each VNF instance in processing the traffic.

We model the above information by a directed, vertex-labeled, and edge-weighted graph, referred to as the *VNF topology*, where the graph topology represents the interconnections between VNF instances, the vertex labels represent the (logical) VNF placement, and the edge weights represent the VNF performances[1]. We refer to the problem of inferring

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.

[1]More precisely, the weight of an edge $e = (s(e), t(e))$ represents the overall performance for data transfer on e and data processing at $t(e)$, the physical meaning of which will be explained in Section II-C.

the VNF topology as the *VNF topology discovery problem*.

A. Related Work

NFV resource management: From an application perspective, our work is related to network management in NFV. Existing works on this topic have addressed VNF placement [4], admission control and path selection [5], and joint optimization of multiple control knobs [6], [7]. Specifically, [6] jointly optimizes VNF placement, routing, and admission control under hard capacity constraints, and [7] jointly optimizes VNF placement, routing, and resource allocation under soft capacity constraints. However, all the above are from the provider's perspective. *To our knowledge, we are the first to investigate the monitoring of NFV network structure and state from an outsider's perspective.*

Network topology discovery: Technically, our work is more related to topology discovery based on end-to-end measurements. In communication networks, the problem was initially studied based on multicast probing [8], where correlation among losses observed at multicast receivers is used to infer the multicast tree. Over the years, the technique was extended to exploit a variety of multicast measurements, including losses [9], delays [10], and a combination of these [11]. Meanwhile, due to limited support of multicast, unicast-based solutions were developed, using stripes of back-to-back unicast probes [12] or "sandwiches" of small and large probes [13]. Most of these algorithms are inspired by *phylogenetic tree algorithms*, which aim at constructing a tree based on the measured distances between leaf nodes [14].

Only a few works considered ground truth topologies that are not trees, all based on measurements from multiple sources. Solutions in [15], [16] still constructed tree topologies, except that the accuracy was analyzed with respect to a ground truth that may not be a tree. Solutions in [17], [18], [19], [20] constructed directed acyclic graphs (DAGs) by merging 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. Assuming measurements of 1-by-2 and 2-by-1 topologies, [22] presented a necessary and sufficient condition for the underlying topology to be identifiable and an algorithm to do so. However, all the above solutions assumed that there is a single route for every source-destination pair, and the routes from/to each node form a tree. In NFV networks, the requirement of VNF traversals can cause flows to deviate from the default routes, and hence the topology traversed by probes from a source (or to a destination) may not be a tree. *To our knowledge, we are the first to investigate topology discovery based on end-to-end measurements for arbitrary topologies under arbitrary routing.*

B. Summary of Contributions

The main contributions of this work are:

1) We are the first to consider external observation-based topology discovery in NFV networks.

2) We show that the approach of tree approximation, as is used by existing solutions, is insufficient for NFV networks, and we propose a two-step solution, which gives a near-smallest logical topology that is equivalent to the ground truth.

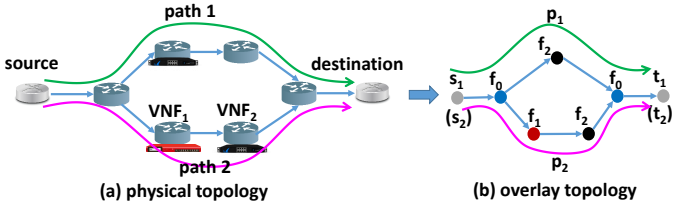


Fig. 1. Topologies of the physical substrate and the VNF overlay.

3) We extend our solution to incorporate service chains, by reformulating our problem as a novel string augmentation problem that can be solved as integer linear programs (ILPs).

4) Via simulations based on real topologies, we verify that our solution significantly outperforms a state-of-the-art solution in both fitting the measurements and approximating the ground truth, and service information plays a critical role.

Although motivated by NFV, our solution is equally applicable to networks with traditional network appliances.

Roadmap. Section II formalizes our problem. Section III addresses a simplified version of our problem in a classical setting, and Section IV addresses the full version that incorporates service information. Section V evaluates the proposed solution against benchmarks. Section VI concludes the paper.

II. Problem Formulation

A. Network Model

We model the VNF overlay, illustrated in Fig. 1, by a directed, vertex-labeled, and edge-weighted graph $\mathcal{G} = (V, E, L, W)$, referred to as the *VNF topology*. The vertex set V denotes the set of VNF instances and critical points on measurement paths (sources, destinations, and branching/joining points), and the edge set E denotes the connections between these points. The label set L denotes the VNF placement, where $l_v \in L$ denotes the type of VNF at vertex $v \in V$. Let $\mathcal{F} = \{f_1, f_2, \ldots\}$ be the set of all types of VNFs supported by the network. As measurement paths may branch/join at a point that does not run any VNF (e.g., a pure router/switch), we introduce a dummy VNF $f_0 \notin \mathcal{F}$ to label such vertices. Lastly, the weight set W represents the network performance as a multi-set of edge weights, where $w_e \in W$ for edge $e = (s(e), t(e)) \in E$ models the overall performance in transferring a packet from $s(e)$ to $t(e)$ *and* processing the packet at $t(e)$. In this work, we consider a family of performance metrics that can be modeled as additive edge weights as detailed in Section II-C. We assume that the sources/destinations of measurement paths do not run VNFs and are observable; the rest of $\mathcal{G}$ is not observable.

B. Flow Model

We measure the network by monitoring a set of flows $\mathcal{D} = \{d_i\}_{i=1}^{n}$, each demanding a source (or ingress point) s_i, a destination (or egress point) t_i, and a service chain $\mathbf{c}_i = (c_{i,j})_{j=1}^{n_i}$, where $c_{i,j} \in \mathcal{F}$ is the type of VNF required at step j of processing flow d_i. As the flow demands are provided by the users (or their proxy), they are assumed to be observable to the inference engine. After a flow d_i is admitted by the network, it is mapped onto a path p_i that goes from s_i to t_i and traverses the service chain $\mathbf{c}_i$ in between. The internal portion

of p_i (i.e., excluding s_i and t_i) is not observable. Note that due to the VNF traversal requirements, a flow may follow a *non-simple path* which may traverse a vertex/edge multiple times.

C. Performance Model

We consider a family of edge weights with two properties: (i) the weights are nonnegative and additive, i.e., the path weight equals the sum weight of the traversed edges, and (ii) the weights can be reliably inferred (by an unbiased estimator) from end-to-end measurements for each path and the shared portion of each pair of paths. Let ρ_i denote the sum weight for path p_i, referred to as the *path length*, and ρ_{ij} denote the sum weight for the shared portion of paths p_i and p_j, referred to as the *shared path length*.

It is known [12] that several important performance metrics satisfy these requirements, listed below for completeness. In the following, we use a "probe" to refer to the smallest unit of measurement, e.g., one packet. As in [23], [12], we assume that probes are sent in pairs on a pair of paths at a time, so that probes in the same pair experience the same performance at shared edges. Moreover, an edge performs independently for different probe pairs, and different edges perform independently. The definitions below can be modified to account for imperfect correlation at shared edges [12].

1) Loss-based Weight: If we measure the end-to-end losses, then the edge weight can be defined as $w_e := -\log \alpha_e$, where α_e is the success rate of edge e (i.e., the probability for a probe to successfully traverse edge e and get processed by the VNF at vertex $t(e)$). Let X_p be the success indicator for path p. Then we have

$$\rho_i = \sum_{e \in p_i} -\log \alpha_e = -\log \Pr\{X_{p_i} = 1\}, \tag{1}$$

$$\rho_{ij} = \sum_{e \in p_i \cap p_j} -\log \alpha_e = -\log \left(\frac{\Pr\{X_{p_i} = 1\} \Pr\{X_{p_j} = 1\}}{\Pr\{X_{p_i} = X_{p_j} = 1\}} \right). \tag{2}$$

Thus, we can calculate the path lengths and the shared path lengths by estimating the success probability of each path and the joint success probability for each pair of paths from the end-to-end losses. It is known that the unbiased estimators of these probabilities are simply their empirical values.

2) Utilization-based Weight: If we measure the end-to-end delays, then the edge weight can be defined as $w_e := -\log \beta_e$, where β_e is the no-queueing probability of edge e (i.e., the probability that a probe incurs no queueing delay in traversing edge e and getting processed at vertex $t(e)$). Let Y_p be the no-queueing indicator for path p. Then we have

$$\rho_i = \sum_{e \in p_i} -\log \beta_e = -\log \Pr\{Y_{p_i} = 1\}, \tag{3}$$

$$\rho_{ij} = \sum_{e \in p_i \cap p_j} -\log \beta_e = -\log \left(\frac{\Pr\{Y_{p_i} = 1\} \Pr\{Y_{p_j} = 1\}}{\Pr\{Y_{p_i} = Y_{p_j} = 1\}} \right). \tag{4}$$

Similar to loss-based weights, we can calculate the path lengths and the shared path lengths by estimating the no-queueing probabilities of each path and each pair of paths from end-to-end queueing indicators. In practice, this can be achieved by comparing each delay measurement with a threshold representing the "maximum end-to-end delay" on that path without queueing (estimated from delays measured when the network is lightly loaded), and counting the fraction of measurements below the threshold.

D. VNF Topology Discovery Problem

Given observations from a set of flows $\{d_i\}_{i \in [n]}$ ($[n] := \{1, \ldots, n\}$), including the sources, the destinations, the service chains, and the corresponding path lengths $\{\rho_i\}_{i \in [n]}$ and shared path lengths $\{\rho_{ij}\}_{i,j \in [n]}$, we want to infer the underlying VNF topology and the paths of these flows.

Topology Selection Criteria: The solution to the VNF topology discovery problem will not be unique, e.g., dummy VNFs can be added without changing the service chains, and the sum weight of two edges traversed by the same set of paths can be split arbitrarily between them without affecting path lengths or shared path lengths. This is an inherent limitation of topology discovery problems [24], [19]. To resolve the ambiguity, additional criteria are needed. Theoretically, the optimal solution should maximize the likelihood of the given measurements [13], [25]. In practice, however, simpler criteria are often used to avoid requiring statistical knowledge of the measurements (i.e., the likelihood function). In this work, we adopt a set of such nonparametric criteria.

Generally, given a set of feasible topologies, each consistent with all the observations, we want to select the topology that is:

1) a *minimum weight representation* that minimizes the total edge weight, or
2) a *minimum size representation* that minimizes the number of edges, or
3) a *minimum order representation* that minimizes the number of vertices.

Intuitively, (1) represents the "best-performing" topology in terms of the total weight, and (2–3) represent the "simplest" topology in terms of the number of edges or vertices. Any topology discovery algorithm can only reconstruct the ground truth up to its minimum weight/size/order representation.

Remark: Objective (1) is consistent with the minimum spanning/Steiner tree model commonly used in phylogenetic inference [26], where the goal is to use a minimum weight tree to convey the relationships between species. Objective (2) is consistent with the penalized likelihood criterion in [13] (where the penalty is the number of edges) and the notion of "simplest topology" in [21]. Objective (3) is the same as the "minimal representation" criterion used in [19], where the goal is to reconstruct the measured distances between participating nodes using a minimum number of hidden nodes. We have renamed these criteria in the convention of graph theory.

III. Solutions based on Path Length Information

We begin with a simplified version of the problem, where only path lengths and shared path lengths are used as in classical topology discovery problems [12], [22]. Accordingly, the goal is reduced to inferring a directed, edge-weighted graph $\mathcal{G} = (V, E, W)$, such that the flows can be mapped to paths in this graph that match the given path lengths and shared path lengths. Although this problem has been studied outside the context of NFV, existing solutions assumed that the

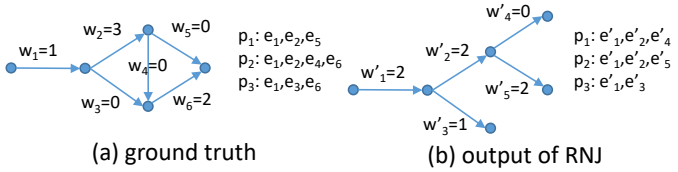


Fig. 2. Counterexample for tree approximation.

underlying topology is either a tree or a union of single-source trees (see Section I-A), neither valid in the context of NFV.

A. Deficiency of Tree Approximation

We use an example to illustrate that it is not always possible to match the given path lengths and shared path lengths by constructing a tree. Consider the ground truth in Fig. 2 (a) with a single source, where w_i denotes the weight of edge e_i. Ignoring measurement errors, we will observe the following: $\rho_1 = 4$, $\rho_2 = 6$, $\rho_3 = 3$, $\rho_{12} = 4$, $\rho_{13} = 1$, and $\rho_{23} = 3$.

Existing solutions will attempt to use an edge-weighted rooted tree to reconstruct these lengths. In particular, the *Rooted Neighbor-Joining (RNJ)* algorithm [12] guarantees correct reconstruction if the ground truth topology is a canonical tree and there is no measurement error. In this case, it returns a topology in Fig. 2 (b), which does not resemble the ground truth. Furthermore, the inferred topology does not match the measurements either, as $\rho'_{13} = 2 \neq \rho_{13}$ and $\rho'_{23} = 2 \neq \rho_{23}$. This is not just a limitation of RNJ: any tree topology will require at least two shared path lengths to be equal, which is inconsistent with the input. *This is a fundamental limitation of tree approximation,* indicating the need of a new topology discovery algorithm that can construct non-tree topologies.

B. Solution for General Topologies

We propose a solution for discovering a general topology based on path lengths and shared path lengths. Our solution consists of two steps: (1) weight inference, and (2) topology construction, where step (1) aims at inferring edge weights at the finest granularity, and step (2) aims at constructing a graph based on the inferred weights to route the flows.

1) Weight Inference: We start by trying to infer edge weights based on the given path length information. Despite the unknown topology, we show that it is still possible to deduce weights at a finer granularity than paths/shared paths.

Problem Definition: We partition the edges in the ground truth topology into $2^n - 1$ *categories* (n: number of flows), where each category A ($A \subseteq [n]$, $A \neq \emptyset$) contains all the edges traversed *only by* the paths in $\{p_i : i \in A\}$. For example, for $n = 3$, we have 7 categories, and category $\{1, 2\}$ contains all the edges traversed by p_1 and p_2 but not p_3. Let w_A denote the sum weight for category-A edges, and $\mathcal{A} := 2^{[n]} \setminus \emptyset$ denote all the categories.

The *weight inference problem* aims at determining $(w_A)_{A\in\mathcal{A}}$ from the given path lengths and shared path lengths. Note that the lengths only specify edge weights up to their sum per category, as one can split each w_A arbitrarily among edges in category A without affecting any path length or shared path length. In this sense, the weight inference problem aims at inferring the edge weights at the finest granularity.

By definition, category-A edges are traversed by a path p_i if and only if $i \in A$. Similarly, category-A edges are shared by paths p_i and p_j if and only if $\{i, j\} \subseteq A$. Therefore, we can formulate the problem as solving the linear equations:

$$\sum_{A:i\in A} w_A = \rho_i, \qquad \forall i \in [n], \tag{5a}$$

$$\sum_{A:\{i,j\}\subseteq A} w_A = \rho_{ij}, \qquad \forall i, j \in [n], \tag{5b}$$

subject to (s.t.) the constraint that $w_A \geq 0$ ($\forall A \in \mathcal{A}$) due to the nonnegativity of edge weights (see Section II-C).

Challenges: There are several practical challenges in solving (5). First, there are exponentially many variables, suggesting that solving this linear system will incur *exponential complexity*. Moreover, there is only a quadratic number of equations, and thus we generally have an under-constrained linear system that *does not have a unique solution*. Furthermore, in practice we can only estimate the values of ρ_i's and ρ_{ij}'s from raw measurements, and the estimation errors can cause the linear system to be *infeasible*.

Results: For the first challenge, we first note that for each input, there is a solution where majority of the categories have zero weight.

Lemma III.1. For each topology, there exists a feasible solution to the weight inference problem that is $(n + \binom{n}{2})$-sparse, i.e., containing at most $n+\binom{n}{2}$ non-zero variables (i.e., per-category weights). Moreover, there exists a solution with the minimum total weight that is $(n + \binom{n}{2})$-sparse.

Proof. We note that the entire set of feasible solutions given by (5) and $w_A \geq 0$ ($\forall A \in \mathcal{A}$) is a bounded nonempty polytope in $\mathbb{R}^{2^n-1}$ space. Every vertex of this polytope, which is a feasible solution, is given by a subset of $2^n - 1$ constraints, where the inequality constraint $w_A \geq 0$ is satisfied with equality. As at least $2^n - 1 - n - \binom{n}{2}$ of these constraints are of the form $w_A = 0$, at most $n + \binom{n}{2}$ variables can be non-zero, i.e., feasible solutions corresponding to vertices of the polytope are $(n+\binom{n}{2})$-sparse. The second claim follows from the fact that if we further minimize $\sum_{A\in\mathcal{A}} w_A$ over the polytope, optimality can always be achieved at a vertex, which gives a minimum weight solution that is $(n + \binom{n}{2})$-sparse. □

Meanwhile, we have shown that no category can be ignored, i.e., with a weight always set to zero.

Lemma III.2. For each $A \in \mathcal{A}$, there exists a ground truth topology for which w_A must be positive.

Proof. We prove the claim by contradiction. Suppose that there exists a weight inference algorithm π that always sets $w_A \equiv 0$ for all inputs. Consider a ground truth topology where only one edge in category A has a non-zero weight of 1; other edge weights are zero. Thus, $\rho_i = 1$ if $i \in A$, and $\rho_i = 0$ otherwise; $\rho_{ij} = 1$ if $\{i, j\} \subseteq A$, and $\rho_{ij} = 0$ otherwise. Let $\mathcal{A}'$ be the set of categories assigned non-zero weights by π. We argue that $\bigcup_{A'\in\mathcal{A}'} A'$ must equal A. Otherwise, we must have either (i) $i \in \bigcup_{A'\in\mathcal{A}'} A' \setminus A$, for which $\sum_{A':i\in A'} w_{A'} > 0$ but $\rho_i = 0$, or (ii) $i \in A \setminus (\bigcup_{A'\in\mathcal{A}'} A')$, for which $\sum_{A':i\in A'} w_{A'} = 0$ but $\rho_i = 1$. If $|A| = 1$, then $\mathcal{A}' = \{A\}$, i.e., π assigns a non-zero

weight to category A, contradicting our assumption. If $|A| > 1$, we argue that for any $\{i,j\} \subseteq A$, $\nexists A' \in \mathcal{A}'$ that contains i but not j, because otherwise we must have $\rho_i > \rho_{ij}$. It implies that $\mathcal{A}' = \{A\}$, again contradicting our assumption. □

Due to Lemma III.2, any solution to the weight inference problem has to deal with exponentially many variables. It remains open whether given an input, one can find, in polynomial time, a polynomial number of categories such that it suffices to only give positive weights to these categories.

To address the second and the third challenges, we first relax the requirements from perfect reconstruction as in (5) to best-effort reconstruction, formulated as a constrained optimization:

$$\min \sum_{i\in[n]} |\sum_{A:i\in A} w_A - \rho_i| + \sum_{i,j\in[n]} |\sum_{A:\{i,j\}\subseteq A} w_A - \rho_{ij}| \quad (6a)$$

$$\text{s.t. } w_A \geq 0, \qquad \forall A \in \mathcal{A}. \quad (6b)$$

This is a convex optimization that can be solved by convex optimization solvers (with input size exponential in n). We note that the ℓ-1 norm in (6a) can be replaced by other norms. The optimal value of (6), denoted by ϵ^*, gives the minimum reconstruction error we have to tolerate due to measurement errors.

We then revisit the problem to include the intention of minimizing the total weight:

$$\min \sum_{A\in\mathcal{A}} w_A \quad (7a)$$

$$\text{s.t. } \sum_{i\in[n]} |\sum_{A:i\in A} w_A - \rho_i| + \sum_{i,j\in[n]} |\sum_{A:\{i,j\}\subseteq A} w_A - \rho_{ij}| \leq \epsilon, \quad (7b)$$

$$w_A \geq 0, \qquad \forall A \in \mathcal{A}. \quad (7c)$$

This optimization tries to minimize the total weight (7a) subject to the constraints of approximately satisfying the measurements (7b) and ensuring nonnegativity. The parameter ϵ is used to trade off the reconstruction error and the total weight of the inferred topology. At the minimum, it should account for measurement errors, i.e., $\epsilon \geq \epsilon^*$. As in (6), other norms can be used instead of the ℓ-1 norm in (7b). Problem (7) is again a convex optimization (with input size exponential in n).

Remark: In cases that the distribution $g(\cdot)$ of measurement errors is known, we can incorporate this information by performing the *maximum likelihood estimation (MLE)* of the per-category weights. This is a constrained optimization similar to (6), with the objective (6a) replaced by $\max g((\sum_{A:i\in A} w_A - \rho_i)_{i\in[n]}, (\sum_{A:\{i,j\}\subseteq A} w_A - \rho_{ij})_{i,j\in[n]})$. Similarly, we can compute a minimum weight representation by solving a variation of (7), with (7b) replaced by a constraint of the form $g(\cdot) \geq \delta$, where δ is no greater than the maximum likelihood.

2) Topology Construction: Given the per-category weights $(w_A)_{A\in\mathcal{A}}$, there are many topologies satisfying these weights, and our objective is to find the "simplest" topology in the sense of minimum size/order representation. If a topology contains at least one edge in category A, we say that category A is *represented* in this topology. Although the optimal solutions for these representations need not be the same, we are able to develop an algorithm that is both near-optimal for minimum order representation and asymptotically near-optimal for minimum size representation.

Algorithm: Our idea is to embed edges representing the categories with non-zero weights into the minimum clique that has sufficiently many edges. This idea is based on the observations that we must construct at least one edge for each category with non-zero weight, and the directed clique (i.e., complete directed graph) is the smallest directed graph that can embed a given number of edges. This is the initial idea behind our topology construction algorithm, referred to as *Clique Embedding (CE)*, shown in Algorithm 1.

Algorithm 1: Clique Embedding (CE)

input : Number of measurement flows n and per-category weights $(w_A)_{A\in\mathcal{A}}$
output: Inferred topology $\mathcal{G}$ and flow paths $\{p_i\}_{i=1}^n$
1 find the minimum directed clique $\mathcal{C}$ with at least $|\{A \in \mathcal{A} : w_A > 0\}|$ edges;
2 **foreach** $A \in \mathcal{A}$ *such that* $w_A > 0$ **do**
3 randomly select an unselected edge in $\mathcal{C}$, and assign it category A and weight w_A;
4 create a new vertex r;
5 **foreach** $i = 1, \ldots, n$ **do**
6 find continuous edge sequences $\{p_{i,j}\}_{j=1}^{m_i}$ that are formed by edges assigned to categories $\{A \in \mathcal{A} : i \in A\}$;
7 **foreach** *edge sequence* $p_{i,j}$ **do**
8 create an edge from r to the beginning of $p_{i,j}$ and an edge from the end of $p_{i,j}$ to r, both of zero weight;
9 p_i is the concatenation of the cycles formed by going from r to $p_{i,j}$ and back to r for $j = 1, \ldots, m_i$;
10 $\mathcal{G}$ consists of all the selected edges in $\mathcal{C}$, vertex r, and all the edges between r and $\mathcal{C}$;

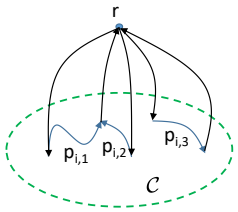


Fig. 3. Illustration of Clique Embedding.

However, the embedded edges may not form valid paths, i.e., for a given $i \in [n]$, the embedded edges in categories $\{A \in \mathcal{A} : i \in A\}$ may not form a sequence of pairwise adjacent edges. To generate valid paths, we construct a special vertex r (line 4), which is connected to/from each continuous sequence of embedded edges that need to be traversed by p_i (lines 7–8). Thus, we can "stitch together" the edge sequences via r to form a valid path (line 9). Fig. 3 illustrates the idea: if the embedding generates three continuous edge sequences $p_{i,1}$, $p_{i,2}$, and $p_{i,3}$ for some $i \in [n]$, then the constructed path p_i goes from r to $p_{i,1}$ and back to r, then to $p_{i,2}$ and back to r, and finally to $p_{i,3}$ and back to r.

Given the set E_i of embedded edges that need to be traversed by p_i (i.e., in categories $\{A \in \mathcal{A} : i \in A\}$), we can find the continuous edge sequences (line 6) as follows:
(i) initialize each edge sequence $p_{i,j}$ as a one-hop sequence containing a randomly selected edge in E_i that has not been covered by the existing edge sequences;
(ii) iteratively extend $p_{i,j}$ by adding one edge at a time to either endpoint from the uncovered edges in E_i, until no more extension can be made;
(iii) if there are still uncovered edges in E_i, repeat (i–ii).

Performance: Among all the feasible topologies, Algorithm 1 gives a near-optimal representation of the ground truth

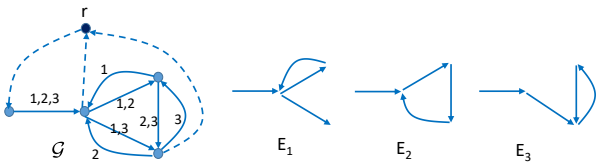


Fig. 4. A possible outcome of Algorithm 1 for $n = 3$. Solid line: embedded edges; dashed line: edges to/from r; E_i: embedded edges that need to be traversed by p_i; edge label: the category.

topology in the following sense.

Theorem III.3. The topology $\mathcal{G}$ given by Algorithm 1 is

(a) near-optimal in minimizing the order, in that $\mathcal{G}$ has at most one more vertex than the minimum order representation, and

(b) asymptotically near-optimal in minimizing the size, in that for any $\epsilon > 0$, the number of edges in $\mathcal{G}$ is no more than $(1 + \epsilon)$ times the number of edges in the minimum size representation for all sufficiently large $|\{A \in \mathcal{A} : w_A > 0\}|$.

Proof. Let $k_e := |\{A \in \mathcal{A} : w_A > 0\}|$ and $h(k_e) := \min\{m : m(m-1) \geq k_e\}$ be the number of vertices in the minimum clique with at least k_e edges.

First, the minimum order representation needs at least one edge in each positive-weight category, and hence its number of vertices is at least $h(k_e)$. The topology given by Algorithm 1 contains $h(k_e) + 1$ vertices. Hence, claim (a) holds.

Moreover, Algorithm 1 constructs at most $k_e + 2h(k_e)$ edges, as there are at most $2h(k_e)$ edges between r and vertices in the clique. The minimum size representation has at least k_e edges. The approximation ratio is thus upper-bounded by $1 + 2h(k_e)/k_e$. As $h(k_e) = O(\sqrt{k_e})$, for every $\epsilon > 0$, $\exists k_0$ such that $2h(k_e)/k_e \leq \epsilon$ for all $k_e \geq k_0$, proving claim (b). □

Example: Consider the input of $n = 3$ and $w_A > 0$ for all $A \in \mathcal{A}$. Fig. 4 illustrates a possible outcome of Algorithm 1, together with the set of embedded edges that need to be traversed by each path. In this case, there is actually no need to add vertex r, i.e., $\mathcal{G} - r$ is still a feasible solution, as the embedded edges for each i already form a valid path.

IV. Solutions Based on Path Length and Service Information

We now revisit the problem when information about the services required by each flow is also used for inference, including the source s_i, the destination t_i, and the service chain $\mathbf{c}_i$ ($i \in [n]$).

While the service information distinguishes our problem from all the existing topology discovery problems, we can still reuse some of the previous solutions. Specifically, as the service information does not inform us about the edge weights, we can still divide the problem into two subproblems: (1) *weight inference*, and (2) *VNF topology construction*. Subproblem (1) has the same input and output as in Section III-B1, and hence results therein apply. Subproblem (2) takes both the inferred per-category weights and the service information as input, and outputs a directed, vertex-labeled and edge-weighted graph $\mathcal{G} = (V, E, L, W)$ that represents the VNF overlay topology. The focus here is subproblem (2).

We note that the graph formed by the union of service chains may not be a feasible solution, e.g., two flows with disjoint service chains may share a subpath and thus have a positive shared path length. The challenge in VNF topology construction is to preserve the service chains while constructing at least one edge in each positive-weight category.

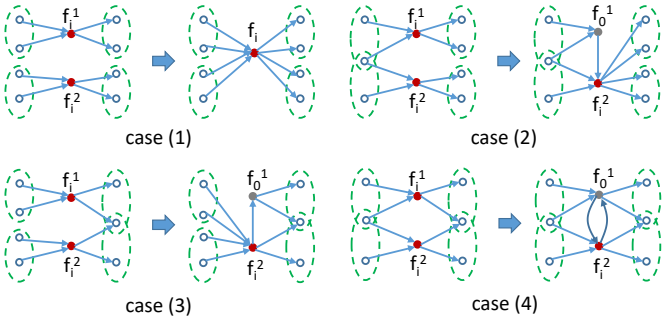


Fig. 5. Merge operation (f_i^1, f_i^2: instances of the same VNF; f_0^1: dummy).

A. Existence of Single-copy Representation

While the ground truth topology may contain multiple instances of the same VNF, we show that it is always possible to construct an equivalent topology that contains at most one instance per VNF, referred to as a *single-copy representation*.

Theorem IV.1. For each VNF topology $\mathcal{G}$, there exists an equivalent single-copy representation $\widetilde{\mathcal{G}}$, i.e., each $f_i \in \mathcal{F}$ is assigned to at most one vertex in $\widetilde{\mathcal{G}}$.

Proof. We prove by construction. Consider an arbitrary VNF topology $\mathcal{G}$. Let $\mathcal{N}^-(v)$ and $\mathcal{N}^+(v)$ denote the incoming/outgoing neighbors of vertex v, i.e., vertices with edges to/from v. For every two vertices labeled by the same (non-dummy) VNF f_i, denoted by f_i^1 and f_i^2, we have four cases: (1) $\mathcal{N}^-(f_i^1) \cap \mathcal{N}^-(f_i^2) = \emptyset$ and $\mathcal{N}^+(f_i^1) \cap \mathcal{N}^+(f_i^2) = \emptyset$, (2) $\mathcal{N}^-(f_i^1) \cap \mathcal{N}^-(f_i^2) \neq \emptyset$ and $\mathcal{N}^+(f_i^1) \cap \mathcal{N}^+(f_i^2) = \emptyset$, (3) $\mathcal{N}^-(f_i^1) \cap \mathcal{N}^-(f_i^2) = \emptyset$ and $\mathcal{N}^+(f_i^1) \cap \mathcal{N}^+(f_i^2) \neq \emptyset$, and (4) $\mathcal{N}^-(f_i^1) \cap \mathcal{N}^-(f_i^2) \neq \emptyset$ and $\mathcal{N}^+(f_i^1) \cap \mathcal{N}^+(f_i^2) \neq \emptyset$. We "merge" f_i^1 and f_i^2 as in Fig. 5: in case (1), we directly merge them; in case (2), we replace f_i^1 by a dummy denoted by f_0^1, which is connected to f_i^2, and rewire outgoing edges of f_i^1 to start from f_i^2; in case (3), we replace f_i^1 by a dummy f_0^1, connected from f_i^2, and rewire incoming edges of f_i^1 to end at f_i^2; in case (4), we replace f_i^1 by a dummy f_0^1, connected to/from f_i^2. Each path traversing f_i^1 will traverse (f_0^1, f_i^2) in case (2), (f_i^2, f_0^1) in case (3), and (f_0^1, f_i^2, f_0^1) in case (4). Each merge operation reduces the number of duplicate VNF instances by one, while preserving the service chains and the represented categories. Repeatedly applying this operation will then give a single-copy representation that is equivalent to $\mathcal{G}$. □

Moreover, the simplest single-copy representation is nearly as simple as the overall simplest representation.

Corollary IV.2. a) The minimum order single-copy representation has as few vertices as the minimum order representation.

b) The minimum size single-copy representation has at most $2R$ more edges than the minimum size representation $\mathcal{G}^*$, where R is the number of duplicate VNF instances in $\mathcal{G}^*$.

Proof. As the merge operation defined in the proof of Theorem IV.1 reduces the number of duplicates by one, while

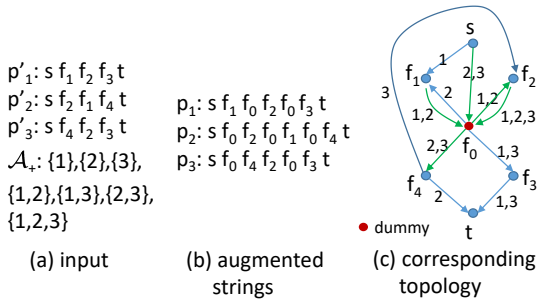


Fig. 6. Example of string augmentation (edge label denotes category).

creating no extra vertex and at most 2 extra edges, applying it to the minimum order/size representation yields the result. □

B. Construction of Single-copy Representation

Although we can extend CE to construct a feasible single-copy representation by incorporating service chains, this solution may introduce more vertices/edges than necessary. In the sequel, we present an optimization-based approach to construct the minimum order/size single-copy representation.

1) String Augmentation Problem (SAP): Let $\mathcal{A}_+$ be the set of categories with positive weights, and $p'_i := s_i \oplus \mathbf{c}_i \oplus t_i$ ($\oplus$: concatenation) be the service chain of flow d_i plus the endpoints. Viewing each path as a string of vertices, we can interpret the problem of constructing a single-copy representation as a *string augmentation problem (SAP)*: augment strings $(p'_i)_{i\in[n]}$ by inserting dummy letters $f_0^1, f_0^2, \ldots$ (each can be inserted multiple times) such that every $A \in \mathcal{A}_+$ is represented, i.e., $\exists$ a pair of letters (f_1, f_2) which appear consecutively in string i ($i \in [n]$) if and only if $i \in A$. Fig. 6 gives an example of the input/output of SAP. The augmented strings provide a VNF topology, where each letter corresponds to a vertex and each string to a path.

The minimum order objective transforms into minimizing the number of distinct dummy letters, and the minimum size objective transforms into minimizing the number of distinct pairs of consecutive letters.

2) Integer Linear Programming (ILP) Formulation: We can formulate SAP as ILPs with a polynomial number of variables and constraints. Our formulation assumes that *the service chains are cycle-free* (i.e., no duplicate letters in p'_i). Let $m_{\max}$ be an upper bound on the number of dummy letters and $l_{\max}$ an upper bound on the length of each string. Let $\mathcal{B} := \{s_i, t_i\}_{i\in[n]} \cup \mathcal{F} \cup \{f_0^k\}_{k\in[m_{\max}]}$ denote the set of all the letters. Based on the extension of CE, we know that $m_{\max} = O(n)$ and $l_{\max} = O(|\mathcal{F}| + n^2)$ suffice.

Variables: We use variable $x_{i,j}^f \in \{0,1\}$ to denote if the j-th letter in string i is f. Moreover, we use variable $\delta_k \in \{0,1\}$ to indicate if dummy f_0^k is used in any string, and $\delta_{f_1,f_2} \in \{0,1\}$ to indicate if (f_1, f_2) is used (consecutively) in any string.

Constraints: The first constraint is that there is at most one letter in each position of each string:

$$\sum_{f\in\mathcal{B}} x_{i,j}^f \le 1, \qquad \forall i \in [n],\ j \in [l_{\max}], \tag{8}$$

and the first (or last) letter must correspond to the source (or destination) of the flow:

$$x_{i,1}^{s_i} = 1, \quad x_{i,l_{\max}}^{t_i} = 1, \qquad \forall i \in [n]. \tag{9}$$

We allow $\sum_{f\in\mathcal{B}} x_{i,j}^f = 0$ in (8) to denote that there might be no letter in a position (and hence the augmented string can be shorter than $l_{\max}$). The second constraint is that service chains must be preserved:

$$\sum_{1\le j_1<j_2\le l_{\max}} x_{i,j_1}^{f_1} \cdot x_{i,j_2}^{f_2} = 1, \qquad \forall i \in [n],\ (f_1, f_2) \in p'_i, \tag{10}$$

$$\sum_{j\in[l_{\max}]} x_{i,j}^f = \mathbf{1}(f \in p'_i), \qquad \forall i \in [n],\ f \in \{s_i, t_i\}_{i\in[n]} \cup \mathcal{F}, \tag{11}$$

which includes preserving the set of non-dummy letters (11) and the order of them (10). Here $\mathbf{1}(\cdot)$ is the indicator function. The third constraint is that each positive-weight category must be represented:

$$\sum_{f_1,f_2\in\mathcal{B}} \prod_{i\in A} \mathbf{1}\Big(\sum_{j=1}^{l_{\max}-1} x_{i,j}^{f_1} x_{i,j+1}^{f_2} > 0\Big) \cdot \prod_{i\notin A} \mathbf{1}\Big(\sum_{j=1}^{l_{\max}-1} x_{i,j}^{f_1} x_{i,j+1}^{f_2} = 0\Big) > 0, \quad \forall A \in \mathcal{A}_+, \tag{12}$$

where $\sum_{j=1}^{l_{\max}-1} x_{i,j}^{f_1} x_{i,j+1}^{f_2}$ is the number of times (f_1, f_2) appears consecutively in string i. Additionally, for minimum order representation, we need

$$x_{i,j}^{f_0^k} \le \delta_k, \qquad \forall k \in [m_{\max}], i \in [n], j \in [l_{\max}], \tag{13}$$

and for minimum size representation, we need

$$\mathbf{1}\Big(\sum_{j=1}^{l_{\max}-1} x_{i,j}^{f_1} x_{i,j+1}^{f_2} > 0\Big) \le \delta_{f_1,f_2}, \qquad \forall f_1, f_2 \in \mathcal{B} \text{ and } i \in [n]. \tag{14}$$

Objective: For minimum order representation, the objective is to minimize $\sum_{k=1}^{m_{\max}} \delta_k$ s.t. constraints (8–13). For minimum size representation, the objective is to minimize $\sum_{f_1,f_2\in\mathcal{B}} \delta_{f1,f2}$ s.t. constraints (8–12) and (14).

Linearization: Constraints (10,12,14) are non-linear. To linearize them, we introduce the following dependent variables, all in $\{0,1\}$. Variable $\gamma_{f_1,f_2,j_1,j_2}^i$ s.t.

$$\gamma_{f_1,f_2,j_1,j_2}^i \le x_{i,j_1}^{f_1}, \tag{15a}$$

$$\gamma_{f_1,f_2,j_1,j_2}^i \le x_{i,j_2}^{f_2}, \tag{15b}$$

$$\gamma_{f_1,f_2,j_1,j_2}^i \ge x_{i,j_1}^{f_1} + x_{i,j_2}^{f_2} - 1 \tag{15c}$$

replaces $x_{i,j_1}^{f_1} \cdot x_{i,j_2}^{f_2}$. Variable ζ_{f_1,f_2}^i s.t.

$$\sum_{j=1}^{l_{\max}-1} \gamma_{f_1,f_2,j,j+1}^i \le l_{\max} \zeta_{f_1,f_2}^i, \tag{16a}$$

$$\sum_{j=1}^{l_{\max}-1} \gamma_{f_1,f_2,j,j+1}^i \ge \zeta_{f_1,f_2}^i \tag{16b}$$

replaces $\mathbf{1}(\sum_{j=1}^{l_{\max}-1} x_{i,j}^{f_1} x_{i,j+1}^{f_2} > 0)$. Variable ξ_{f_1,f_2}^A s.t.

$$\xi_{f_1,f_2}^A \le \zeta_{f_1,f_2}^i, \qquad \forall i \in A, \tag{17a}$$

$$\xi_{f_1,f_2}^A \le 1 - \zeta_{f_1,f_2}^i, \qquad \forall i \notin A, \tag{17b}$$

$$\xi_{f_1,f_2}^A \ge \sum_{i\in A} \zeta_{f_1,f_2}^i + \sum_{i\notin A} (1 - \zeta_{f_1,f_2}^i) - n + 1 \tag{17c}$$

replaces $\prod_{i\in A} \mathbf{1}(\sum_{j=1}^{l_{\max}-1} x_{i,j}^{f_1} x_{i,j+1}^{f_2} > 0) \cdot \prod_{i\notin A} \mathbf{1}(\sum_{j=1}^{l_{\max}-1} x_{i,j}^{f_1} x_{i,j+1}^{f_2} = 0)$.

Using these variables, we can rewrite (10,12,14) as

$$\sum_{1\le j_1<j_2\le l_{\max}} \gamma_{f_1,f_2,j_1,j_2}^i = 1, \qquad \forall i \in [n],\ (f_1, f_2) \in p'_i, \tag{18}$$

$$\sum_{f_1,f_2\in\mathcal{B}} \xi_{f_1,f_2}^A > 0, \qquad \forall A \in \mathcal{A}_+, \tag{19}$$

$$\zeta_{f_1,f_2}^i \le \delta_{f_1,f_2}, \qquad \forall f_1, f_2 \in \mathcal{B} \text{ and } i \in [n], \tag{20}$$

TABLE I
PARAMETERS OF AS TOPOLOGIES

AS	ISP	#nodes	#links
1755	Ebone (Europe)	172	381
6461	Abovenet (US)	182	294
3967	Exodus (US)	201	434

which converts the problem of constructing the minimum order/size single-copy representation into ILPs.

Complexity: The number of variables in these ILPs is $O(|\mathcal{B}|^2(nl_{\max}^2 + |\mathcal{A}_+|))$, and the number of constraints is $O(n|\mathcal{B}|^2(l_{\max}^2 + |\mathcal{A}_+|))$. As $|\mathcal{B}| = O(n + |\mathcal{F}|)$, $l_{\max} = O(|\mathcal{F}| + n^2)$, and $|\mathcal{A}_+| = O(n^2)$, both numbers are in $O(n(n + |\mathcal{F}|)^2(n^2 + |\mathcal{F}|)^2)$. This implies that the heuristic of LP relaxation with rounding will have a polynomial complexity. We conjecture that SAP is generally NP-hard.

V. PERFORMANCE EVALUATION

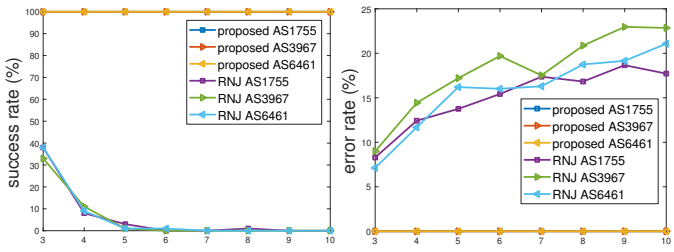
Setting: We evaluate the proposed solutions on Rocketfuel *Autonomous System (AS)* topologies [27], which represent IP-level connections between routers of several *Internet service providers (ISPs)*. Parameters of the considered topologies are given in Table I. We randomly assign each link a (symmetric) weight in $[0.02, 0.1]$, and treat these topologies as substrates.

We generate VNF overlays by randomly selecting S of the high-degree nodes (degree $\geq$ 9) as servers, and randomly placing $|\mathcal{F}|$ types of VNF instances at these servers (one per server), while ensuring at least one instance per type. We then randomly select $n+1$ of the low-degree nodes (degree ≤ 2) as endpoints, where one is designated as the source and the rest as destinations. Each service chain is a random permutation of $|\mathbf{c}_i|$ different VNFs, where $|\mathbf{c}_i|$ is uniformly distributed in $\{1, \ldots, |\mathcal{F}|\}$. The path of each flow is a concatenation of the shortest (hop count) paths from the source to the nearest instance of the first VNF, then to the nearest instance of the second VNF, etc. All results are averaged over at least 10 Monte Carlo runs. We use "node/link" to refer to elements in the substrate, and "vertex/edge" to refer to those in the overlay.

Benchmark: We use *Rooted Neighbor-Joining (RNJ)* [12] as the benchmark. RNJ represents the state of the art, as all existing solutions for single-source probing assume tree topology, and RNJ guarantees correct reconstruction in this case.

Reconstruction Accuracy: First, we compare the accuracy of the proposed solution (by solving (7) with $\epsilon = 0$) against RNJ in reconstructing the given path and shared path lengths, where we assume accurate measurements for both algorithms. Fig. 7 (a) shows the success rate, defined as the fraction of time that all the lengths are reconstructed correctly. Fig. 7 (b) shows the normalized reconstruction error, defined as $||\hat{\boldsymbol{\rho}} - \boldsymbol{\rho}||_1/||\boldsymbol{\rho}||_1$, where $\boldsymbol{\rho}$ is the vector of given path/shared path lengths, and $\hat{\boldsymbol{\rho}}$ the reconstructed values. RNJ fails to match the measurements when the ground truth topology is no longer a tree, while our solution is always accurate. This highlights the need to consider general graphs in VNF topology discovery.

Inference Accuracy: Next, we compare RNJ, CE (Algorithm 1), and SAP (Section IV-B2) in the accuracy of the inferred topology. We extend CE to connect vertex r from/to each source/destination with zero-weight edges. We solve SAP



(a) reconstruction success rate (b) reconstruction error

Fig. 7. Accuracy of reconstructing path/shared path lengths ($|\mathcal{F}| = 5$, $S = 5$).

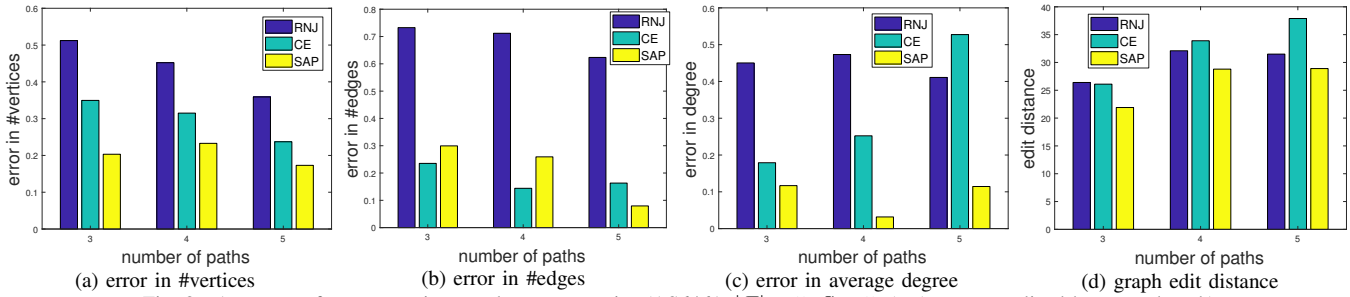
with the minimum order objective by the CPLEX Optimizer. To measure the accuracy, we evaluate the normalized error in several graph properties, including #vertices, #edges, and average vertex degree (including in/out-degree). Moreover, we evaluate the *graph edit distance*, defined as the minimum number of graph edits (vertex/edge insertion, deletion, substitution) to make the inferred topology identical to the ground truth, up to a permutation of internal vertices.

Fig. 8 shows the result when varying the number of paths n. RNJ incurs substantial error, as it significantly underestimates the complexity of the ground truth topology by only constructing trees. Meanwhile, CE tends to give overly dense topologies as it aims at embedding the positive-weight categories into the smallest graph. By jointly considering path length information and service information, SAP achieves the best accuracy. We have verified the comparison when varying the number of servers S, which also shows that while SAP only constructs single-copy representations, its accuracy is not sensitive to the replication of VNFs. We note that although RNJ and SAP have similar edit distances, they differ significantly in the structure of the inferred topology (tree vs. general graph). As shown in Fig. 9, only SAP resembles the structure of the ground truth, where the source and the destinations are connected via a densely-connected core that hosts the VNFs.

Impact of Measurement Error: Finally, we repeat the comparison by packet-level simulations. We send N pairs of probes on each pair of paths to estimate the path/shared path lengths as described in Section II-C1 (discarding negative values), where an edge with weight w_e drops packets with probability $1 - e^{-w_e}$. Fig. 10 (a) shows the normalized error in estimating the path/shared path lengths, and Fig. 10 (b) shows the topology inference accuracy measured by graph edit distance. We see that SAP starts to outperform the other algorithms once the estimation error goes below 30%.

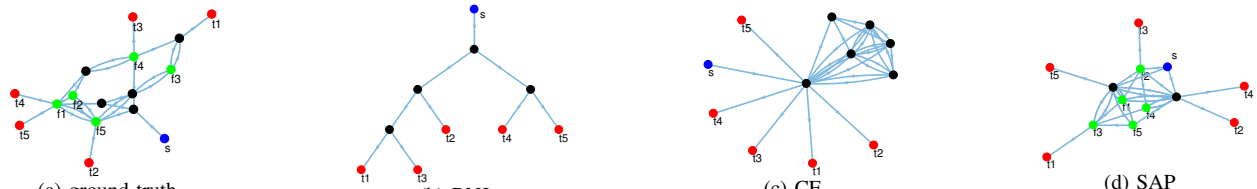
VI. CONCLUSION

We consider, for the first time, inferring the structure and state of NFV networks based on the service chains and the end-to-end performances of measurement flows. We show that existing tree-based algorithms cannot guarantee a feasible solution that is consistent with all the observations, which motivates us to propose a novel two-step solution designed to construct the simplest logical topology that is equivalent to the ground truth with respect to the given observations. Extensive evaluations show that the proposed solution significantly improves the accuracy over a state-of-the-art algorithm.



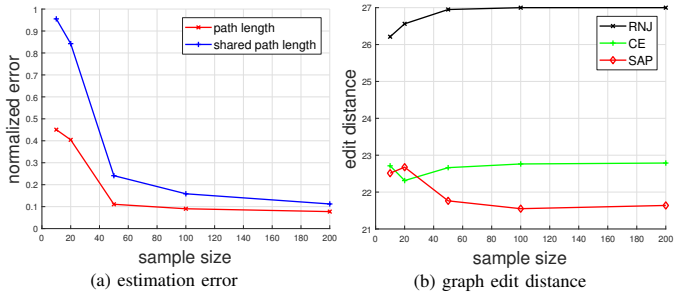
(a) error in #vertices (b) error in #edges (c) error in average degree (d) graph edit distance

Fig. 8. Accuracy of reconstructing topology as n varies (AS6461, $|\mathcal{F}| = 5$, $S = 5$, (a-c) are normalized by ground truth).



(a) ground truth (b) RNJ (c) CE (d) SAP

Fig. 9. Example of inferred topologies and ground truth ($|\mathcal{F}| = 5$, $S = 5$, $n = 5$). ●: source; ●: destination; ●: VNF; ●: dummy.



(a) estimation error (b) graph edit distance

Fig. 10. Results of packet-level simulation (AS6461, $|\mathcal{F}| = 5$, $S = 5$, $n = 3$).

REFERENCES

[1] J. Sherry and S. Ratnasamy, "A survey of enterprise middlebox deployments," EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2012-24, Feb 2012. [Online]. Available: http://www2.eecs.berkeley.edu/Pubs/TechRpts/2012/EECS-2012-24.html

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

[3] "AT&T vision alignment challenge technology survey," AT&T Domain 2.0 Vision White Paper, November 2013. [Online]. Available: https://www.att.com/Common/about_us/pdf/AT&TDomain2.0VisionWhitePaper.pdf

[4] R. Cohen, L. Lewin-Eytan, J. Naor, and D. Raz, "Near optimal placement of virtual network functions," in *IEEE INFOCOM*, April 2015.

[5] T. Loukovszki and S. Schmid, "Online admission control and embedding of service chains," in *ACM SIROCCO*, July 2015.

[6] T.-W. Kuo, B.-H. Liou, K. C.-J. Lin, and M.-J. Tsai, "Deploying chains of virtual network functions: On the relation between link and server usage," in *IEEE INFOCOM*, April 2016.

[7] H. Feng, J. Llorca, A. M. Tulino, D. Raz, and A. F. Molisch, "Approximation algorithms for the nfv service distribution problem," in *IEEE INFOCOM*, April 2017.

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

[9] N. Duffield, J. Horowitz, F. L. Presti, and D. Towsley, "Multicast topology inference from measured end-to-end loss," *IEEE Transactions on Information Theory*, vol. 48, no. 1, pp. 26–45, January 2002.

[10] N. G. Duffield and F. L. Presti, "Network tomography from measured end-to-end delay covariance," *IEEE/ACM Transactions on Networking*, vol. 12, no. 6, pp. 978–992, December 2004.

[11] N. G. Duffield, J. Horowitz, and F. L. Presti, "Adaptive multicast topology inference," in *IEEE INFOCOM*, 2001.

[12] 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.

[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] R. Durbin, S. R. Eddy, A. Krogh, and G. Mitchison, *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, 1999.

[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] G. Berkolaiko, N. Duffield, M. Ettehad, and K. Manousakis, "Graph Reconstruction from Path Correlation Data," *ArXiv e-prints*, 2018.

[23] 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.

[24] J. Pearl, *Probabilistic Reasoning in Intelligent Systems—Networks of Plausible Inference*. Morgan Kaufmann, 1988.

[25] R. Castro, M. Coates, and R. Nowak, "Likelihood based hierarchical clustering," *IEEE Transactions on Signal Processing*, vol. 52, no. 8, pp. 2308–2321, August 2004.

[26] A. S. Teixeira, P. T. Monteiro, J. A. Carrico, M. Ramirez, and A. P. Fancisco, "Not seeing the forest for the trees: Size of the minimum spanning trees (MSTs) forest and branch significance in MST-based phylogenetic analysis," *PLoS ONE*, vol. 10, no. 3, March 2015.

[27] N. Spring, R. Mahajan, and D. Wetherall, "Measuring ISP topologies with Rocketfuel," in *ACM SIGCOMM*, August 2002.