Bound Inference and Reinforcement Learning-based Path Construction in Bandwidth Tomography

Cuiying Feng*, Jianwei An*, Kui Wu*, and Jianping Wang†
*Department of Computer Science, University of Victoria, Victoria, BC, Canada
† Department of Computer Science, City University of Hong Kong

***Abstract*—Inferring the bandwidth of internal links from the bandwidth of end-to-end paths, so-termed bandwidth tomography, is a long-standing open problem in the network tomography literature. The difficulty is due to the fact that no existing mathematical tool is directly applicable to solve the inverse problem with a set of min-equations. We systematically tackle this challenge by designing a polynomial-time algorithm that returns the exact bandwidth value for all identifiable links and the tightest error bound for unidentifiable links for a given set of measurement paths. When measurement paths are not given in advance, we prove the hardness of building measurement paths that can be used for deriving the global tightest error bounds for unidentifiable links. Accordingly, we develop a reinforcement learning (RL) approach for measurement path construction, that utilizes the special knowledge in bandwidth tomography and integrates both offline training and online prediction. Evaluation results with real-world ISP as well as simulated networks demonstrate that compared to other path construction methods, *Random* and *Diversity Preferred*, our RL-based path construction method can build measurement paths that result in much smaller average error bound of link bandwidth.**

***Index Terms*—Bandwidth Tomography, Measurement Path Construction**

I. Introduction

To guarantee the quality of service (QoS) and the smooth operation of networks, any Internet service provider (ISP) needs to closely monitor the network performance, including delay, available bandwidth, network congestion, link losses, and so on. In a large physical network, directly measuring the performance of each link is infeasible due to the high measurement overhead. In the new era of network virtualization, ISPs can utilize network slices to dynamically form different virtual networks for dedicated network applications. Network virtualization, however, does not necessarily lead to easy performance monitoring. In contrast, due to the dynamic changes of network configuration, it becomes more critical yet more challenging to directly measure the performance of virtual links or virtual services. A well-known strategy is to infer the performance/status of physical/virtual links via end-to-end measurements. This method was first termed as network tomography [1] in 1996 and has attracted substantial research since then [2]–[6].

So far, most work on network tomography focuses on additive metrics, e.g., delay, where the value of an end-to-end path is the total value of all the links on the path. An important performance metric, available bandwidth (or bandwidth for short), has been largely avoided in the network tomography literature. It is well known that many research papers have investigated the method to estimate the end-to-end bandwidth [7], and accordingly measurement tools have been developed. For instance, *pathchar*, *clink*, *pchar*, and *bfind* use ICMP "TTL exceeded" messages for bandwidth estimation. Nevertheless, all the above research mainly targets at estimating the end-to-end bandwidth[1]. A network tomography approach to estimating the bandwidth of every *individual link* in the whole network via a small number of monitoring nodes, termed as *bandwidth tomography*, is surprisingly missing.

Boolean-based network tomography that identifies link failure or link congestion is loosely relevant but starkly different to bandwidth tomography. The goal of Boolean-based network tomography [10] is to infer whether or not a link/node is congested or failed based on the congestion/failure status of end-to-end paths. In other words, the input and output of Boolean-based network tomography are both binary. In contrast, bandwidth tomography is to infer the bandwidth *value* of individual links from end-to-end path bandwidth.

A piece of work in the network calculus domain [11], [12] has touched a similar problem as bandwidth tomography: inferring the service curves of links based on the service curve of end-to-end path, i.e., service curve decomposition. Service curve denotes the *cumulative* service amount *over time* offered to a given traffic flow. Nevertheless, only line topology and tree topology were studied. The limited results in [11], the fact that no significant progress has been found since 2008, and our own investigation, all suggest that service curve decomposition is a hard-to-achieve goal in general networks.

The bandwidth tomography research is missing in the literature for two main reasons. First, unlike additive metrics, bandwidth is a metric that uses the minimum, i.e. the bandwidth of a path equals the minimal bandwidth over all the links along the path. No existing mathematical tool can be applied directly to solve the inverse problem with a set of min-equations (refer to Section II for details). The only analytical tool that might help is max-plus [13] and min-plus [14] algebras. Our deep investigation, however, concludes that these algebras cannot be used to solve our problem since the min-equations do not

[1]Some tools such as *Tailgating* [8] and *pathneck* [9] use probe packets to estimate the bandwidth of a link along the measurement path. Nevertheless, these tools are inaccurate on paths longer than a few hops [8] or require intermediate routers to support ICMP [9]. Actually, assuming the capability of estimating the metric (e.g., bandwidth/delay) of links along a path just from the end-to-end measurement of this path would invalidate most, if not all, work in network tomography.

satisfy the algebraic properties of max-plus [13] and min-plus [14]. Second, the minimum operation results in high information loss, since we only know that the bandwidth of a path is not higher than the bandwidth of the constituting links. As an analogy to lossy compression, the minimum operation on link bandwidth values is similar to a quantizer that replaces all the values with the same (smallest) value. Without the help of other side information, it is theoretically impossible to recover the original values after this quantization step.

We, for the first time, formally formulate and systematically study two core problems in bandwidth tomography:

1) Given a network, a set of end-to-end measurement paths, and the measured end-to-end bandwidth, is a link in the network identifiable[2]? If not, what are the lower and upper bounds of its bandwidth?
2) How to construct measurement paths so that any link is (a) either identifiable or (b) the error bound (i.e., the gap between the upper and lower bounds) is the tightest if its identifiability is impossible.

Our solutions to the above two problems lay a solid foundation for future research in this important area. The contributions of the paper include:

- For the first problem, we develop a polynomial-time algorithm that returns the exact bandwidth value for all identifiable links and the tightest error bounds for unidentifiable links. We also present the necessary and sufficient condition for a link to be identifiable.
- For the second problem, we prove that in the worst case we must list all possible measurement paths (MPs) in order to derive the global tightest error bounds[3]. In other words, there is no polynomial-time algorithm to derive the global tightest error bounds unless P=NP, since listing all measurement paths is $\#P$-complete [15]. Note that $\#P$-complete is at least as difficult as NP-complete [15].
- We then design a reinforcement learning (RL) based path construction method, called Guided Sequential Path Construction (*GSPC*). Quite different from traditional reinforcement learning methods, *GSPC* utilizes the special knowledge from our analysis and integrates both offline training over simulated networks and online prediction over the target network. This RL structure can effectively handle the difficulties of applying RL in the special application context of bandwidth tomography.
- We perform extensive evaluation of *GSPC* over real-world ISP topology as well as simulated networks. Compared with two baseline path construction methods, *Random* and Diversity Preferred (*DP*), *GSPC* improves *Random* and *DP* in terms of average error bound by 238% and 193%, respectively. *GSPC* also returns near-optimal results in small-scale simulated networks where listing all measurement paths for deriving the ground-truth global optimum is possible.

[2] A link is identifiable if its bandwidth value can be uniquely determined.

[3] A link error bound is called globally tightest if it is the smallest among all possible error bounds derived for the link with different sets of MPs.

Finally, the algorithms developed in this paper have significant practical meaning. Using the popular network measurement tools such as *pathchar*, *clink*, and *pchar* as the basic building block for measuring the end-to-end bandwidth, our algorithms (1) guide the construction of measurement paths and (2) return the network-wide, link-level bandwidth results.

II. System Model

A network is modelled as a graph $\mathcal{G} = (V, L)$ that consists of $|V|$ vertices and $|L|$ links. With a set of monitors deployed in the network, we can use existing methods, such as *pathload* [16], to measure the bandwidth of a measurement path (defined below). We are interested in the bandwidth of all links in the network. To facilitate bound analysis, we assume that the maximum bandwidth over all links is b_{max}. This value can be set based on the physical specification of the network. Note that with minor changes, the analytical results of this paper are applicable for the scenario where different links have different maximum bandwidth values.

Following the convention in network performance tomography [3], [17], we introduce basic assumptions and notations as follows:

- $\mathcal{G}$: A connected and undirected graph. Each link has distinct end nodes (i.e., no self loop), and no two links in $\mathcal{G}$ connect to the same pair of nodes.
- *Measurement path (MP)*: A non-loop path that only contains two monitors at its end nodes. For test purpose, probing packets along an MP could be routed via source routing. This assumption has been used in most existing work [3], [5].
- The network under consideration is assumed to be "static", implying that either the bandwidth changes slowly relative to the measurement process or it represents statistical characteristics (e.g., mean) that stay constant over time. This assumption has been broadly adopted in most network tomography work [3], [5], [17].

Fundamentally different from performance tomography with additive metrics [3], [18], bandwidth tomography uses min-operation, i.e., the bandwidth of an MP is the minimum bandwidth of all links along the MP. In other words, traditional linear algebra is not applicable for bandwidth tomography.

We use the example in Fig. 1 (a) to illustrate the concept. The network has three monitors marked in red, and there are three MPs among the monitors. Different MPs may lead to different results of end-to-end bandwidth. We use $x_{p,q}$, which is unknown, to denote the bandwidth on link $l_{p,q}(p, q = 1, 2, 3, 4, 5)$. If there is no link between node v_p and v_q, $x_{p,q} = 0$. b_i denotes the end-to-end bandwidth on MP $P_i(i = 1, 2, 3)$. Like most network tomography work [3], [5], we assume an undirected graph, i.e., $x_{p,q} = x_{q,p}$.

Due to the property of bandwidth, we have the following system of min-equations, short-termed as min-system in the rest of the paper:

$$\begin{cases} x_{1,5} \wedge x_{5,2} \wedge x_{2,3} = b_1 \\ x_{1,5} \wedge x_{5,2} \wedge x_{2,4} = b_2 \\ x_{3,2} \wedge x_{2,4} = b_3 \end{cases} \tag{1}$$

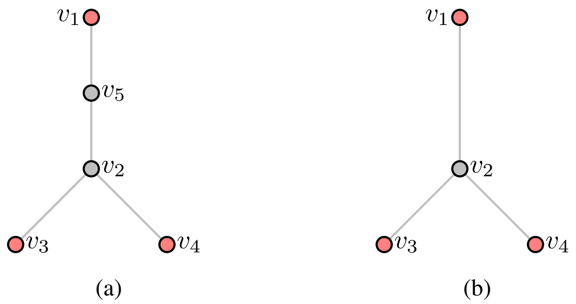


Fig. 1: Example topology (a) and its simplified topology (b).

where $\wedge$ means the min operation.

To simplify analysis, we can remove some non-monitor node of degree 2 (e.g., node v_5 in Fig. 1 (a)), and establish a virtual link between this node's two neighbors (e.g., nodes v_1 and v_2) if there is no link between them[4]. This is because we have no way to distinguish the bandwidth value $x_{1,5}$ and the bandwidth value $x_{5,2}$ based on end-to-end measurements. In terms of bandwidth analysis, we can only infer the bandwidth on the path segment $v_1 \to v_5 \to v_2$. As such, we treat this path segment as a virtual link $l_{1,2}$. The simplified network after the above pre-processing is shown in Fig. 1 (b). In addition, we should ignore any non-monitor node of degree 1 since there is no way to build an MP passing through this node. Therefore, in the rest of the paper, the target network $\mathcal{G}$ is the simplified network after the above pre-processing.

After the above pre-processing, the min-system for Fig. 1 (b) is

$$\begin{cases} x_{1,2} \wedge x_{2,3} = b_1 \\ x_{1,2} \wedge x_{2,4} = b_2 \\ x_{2,3} \wedge x_{2,4} = b_3 \end{cases} \tag{2}$$

With a slight abuse of notation $\wedge$, let's denote the above linear system into *equivalent* matrix form $\mathbf{R} \wedge \mathbf{x} = \mathbf{b}$, where

$$R = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix} \tag{3}$$

$$\mathbf{x} = \begin{pmatrix} x_{1,2} & x_{2,3} & x_{2,4} \end{pmatrix}^\intercal \tag{4}$$

$$\mathbf{b} = \begin{pmatrix} b_1 & b_2 & b_3 \end{pmatrix}^\intercal \tag{5}$$

The first problem in bandwidth tomography is: **given $\mathbf{R}$ and $\mathbf{b}$, can we infer the values of $\mathbf{x}$?** We answer this question in the next section.

III. Bound Analysis and Algorithm for Obtaining the Tightest Bound

Different from the linear system with additive metrics [3], [5], the min-system cannot take advantage of existing results in linear algebra. Nevertheless, each min-equation can help to bound the bandwidth of related links. For instance, (2) can be formulated as:

$$\begin{cases} b_1 \le x_{1,2} \le b_{max}, \; b_1 \le x_{2,3} \le b_{max} \\ b_2 \le x_{1,2} \le b_{max}, \; b_2 \le x_{2,4} \le b_{max} \\ b_3 \le x_{2,3} \le b_{max}, \; b_3 \le x_{2,4} \le b_{max} \end{cases} \tag{6}$$

Hence, we can obtain the lower and upper bounds of the bandwidth for each link from the min-system. Define the *error bound* as the gap between the upper bound and the lower bound. Our goal is to shrink the above naïve error bound as much as possible. That is, if a link is identifiable, the error bound of its bandwidth should be 0; otherwise, its error bound should be the tightest. Note that the tightest error bound in this section is *conditional* in the sense that it is the best bound that we can derive from the given min-system.

To solve the problem, we have the following observation: a min-equation of k variables $\wedge_{i=1}^{k} x_i = b$ is equivalent to the following two conditions:

$$\begin{cases} (1)\ b \le x_i \le b_{max}, i = 1, \dots, k \\ (2)\ \text{at least one of } x_i \text{ is equal to } b. \end{cases} \tag{7}$$

Based on this observation, we can prove the following lemma:

Lemma 1. *Assume that a min-system has n variables and m min-equations, $\mathbf{R} \wedge \mathbf{x} = \mathbf{b}$, where $\mathbf{R}$ is an $m \times n$ Boolean matrix, $\mathbf{x} = \begin{pmatrix} x_1 & \dots & x_n \end{pmatrix}^\intercal$, and $\mathbf{b} = \begin{pmatrix} b_1 & \dots & b_m \end{pmatrix}^\intercal$. Assume that the number of distinct $b_i (i = 1, \dots, m)$ is d. W.L.O.G., assume that $b_i \le b_j (i \le j)$ and the d distinct values are $b'_1, \dots, b'_d$, respectively. We have:*

1) *The links corresponding to $\mathbf{x}$ can be divided into d disjoint nonempty sets, denoted by $S_{b'_k} (k = 1, \dots, d)$.*
2) *Furthermore, if $|S_{b'_k}| = 1$, then the link in $S_{b'_k}$ is identifiable (i.e., its error bound is zero), otherwise, b'_k is the greatest lower bound for the links in $S_{b'_k}$.*

Proof. Base on the first condition in (7), we can build d intervals $[b'_1, b_{max}]$, $[b'_2, b_{max}]$, ..., $[b'_d, b_{max}]$.

To prove (1), we only need to show that based on the min-system $\mathbf{R} \wedge \mathbf{x} = \mathbf{b}$ we can find a way to *uniquely* assign any $x_i \in \mathbf{x}$ to one of the d intervals. The method to assign x_i into an interval is as follows: If x_i appears in l min-equations, whose values are ordered in non-decreasing order and are $b'_{j_1}, \dots, b'_{j_l}$, respectively, then we put x_i into the interval $[b'_{j_l}, b_{max}]$. Clearly, such an assignment is unique for x_i. Since based on the second condition of (7), each interval must include at least one measurement value and hence (1) is proved. Since any measurement value in $[b'_i, b_{max}]$ must also fall in $[b'_j, b_{max}]$ $(i > j)$, the above assignment implies the greatest lower bound of x_i that we can obtain from the min-system.

Based on the second condition in (7), every interval must include at least one measurement value (i.e., correspondingly one link). If an interval $[b'_k, b_{max}]$ only includes one measurement value, the link corresponding to this value is hence identified. If the interval includes multiple measurement

[4]Note that if a link already exists between this node's two neighbors, we cannot remove this node and add the virtual link because otherwise the new graph will have two links connecting the same pair of nodes.

Algorithm 1: Calculate the Tightest Bounds (CTB)

input : a min-system, $\mathbf{R} \wedge \mathbf{x} = \mathbf{b}$, where $\mathbf{R} = [r_{ij}]_{m \times n}$ is a Boolean matrix, $\mathbf{x} = \begin{pmatrix} x_1 & \dots & x_n \end{pmatrix}^\intercal$, and $\mathbf{b} = \begin{pmatrix} b_1 & \dots & b_m \end{pmatrix}^\intercal$

output: the tightest error bound for every link

begin
  Order the m min-equations in the *non-increasing* order of their values;
  /* Variable elimination (like Gaussian elimination) */
  for $i = 1$ *to* $m - 1$ **do**
    for $j = i + 1$ *to* m **do**
      if $b_i == b_j$ **then**
        continue;
      end
      $r_{jk} = \max\{r_{jk} - r_{ik}, 0\}, k = 1, \dots, n$;
    end
  end
  /* Assigning interval and determine bounds */
  for $i = 1$ *to* m **do**
    for $j = 1$ *to* n **do**
      if $r_{ij} == 1$ **then**
        Assign x_j into interval $[b_i, b_{max}]$, i.e., x_j's error bound is $b_{max} - b_i$;
      end
    end
  end
  for $i = 1$ *to* m **do**
    if *there is only one* $r_{ij} == 1 (j = 1, \dots, n)$ **then**
      $x_j = b_i$, i.e., x_j's error bound reduces to 0;
    end
  end
end

values, however, we cannot identify all the corresponding links. In this case, b'_k is the greatest lower bound for these links. Hence, (2) is proved. □

The proof of Lemma 1 is constructive (w.r.t the link-interval assignment), based on which we can design an Algorithm, called CTB, to find the tightest error bound for every link in the given min-system. The pseudo-code is shown in Algorithm 1. Algorithm 1 is correct due to Lemma 1 and the fact that if a link is not identifiable, then the min-system offers no information to reduce its upper bound. The worst-case complexity of Algorithm 1 is $O(mn + mlogm)$, since the first step takes $O(m \log m)$ and the rest of code is mainly two loops over m and n.

Lemma 1 and CTB are important to understand the rest analysis in the paper. CTB works in a way similar to Gaussian elimination. We give an example to illustrate the operations in CTB so that readers can understand the intuition of the lemmas and theorems in the rest of the paper, *whose proofs are omitted due to space limit.*

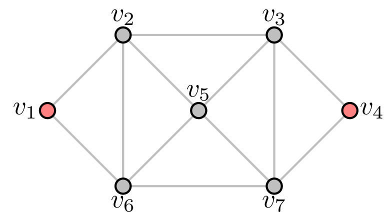


Fig. 2: An example network to illustrate CTB.

Example 1. ***(Example of CTB)*** *An example network topology is shown in Fig. 2, where two monitors are marked in red. Assume that we have built four MPs and their corresponding min-system is shown below:*

$$\begin{cases} P_1 : x_{1,2} \wedge x_{2,3} \wedge x_{3,4} = 2 \\ P_2 : x_{1,6} \wedge x_{6,7} \wedge x_{7,4} = 1 \\ P_3 : x_{1,6} \wedge x_{6,5} \wedge x_{5,7} \wedge x_{7,4} = 2 \\ P_4 : x_{1,2} \wedge x_{2,6} \wedge x_{6,7} \wedge x_{7,4} = 1 \end{cases} \quad (8)$$

This min-system only has two distinct end-to-end bandwidth values and the maximum is 2. This means that the greatest lower bound for covered links is at most 2. Therefore, in order to find out the links with the greatest lower bound 2, we sort the min-equations in the non-increasing order based on their values.

$$R = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix} \quad (9)$$

$$\mathbf{x} = \begin{pmatrix} x_{1,2} & x_{2,3} & x_{3,4} & x_{1,6} & x_{6,5} & x_{5,7} & x_{7,4} & x_{6,7} & x_{2,6} \end{pmatrix}^\intercal \quad (10)$$

$$\mathbf{b} = \begin{pmatrix} 2 & 2 & 1 & 1 \end{pmatrix}^\intercal \quad (11)$$

Now, we consider the paths with the maximum end-to-end bandwidth (i.e., P_1 and P_3). Obviously, the links on these paths cannot be identified. Nevertheless, we can conclude that all links covered by the two paths must have a bandwidth no smaller than 2 and hence these links cannot appear in other paths (i.e., P_2 and P_4) that have smaller end-to-end bandwidth. We thus can set the lower bandwidth bound of these links to 2, and in the meantime we do not need to consider these links anymore. After removing these links, we have the following new min-system.

$$R_{new} = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \quad (12)$$

$$\mathbf{x}_{new} = \begin{pmatrix} x_{6,7} & x_{2,6} \end{pmatrix}^\intercal \quad (13)$$

$$\mathbf{b}_{new} = \begin{pmatrix} 1 & 1 \end{pmatrix}^\intercal \quad (14)$$

The above process is equivalent to the variable elimination part in CTB. In this new min-system, all the "paths" have

TABLE I: Dividing links into disjoint sets

	Covered links
S_2	$l_{1,2}, l_{2,3}, l_{3,4}, l_{1,6}, l_{6,5}, l_{5,7}, l_{7,4}$
S_1	$l^*_{6,7}, l_{2,6}$

Note: * means identifiable.

the same end-to-end bandwidth 1. *Because all the links with higher lower bound on P_2 (i.e., $l_{1,6}$ and $l_{7,4}$) have been removed, it's clear that $l_{6,7}$ is identifiable and its value $x_{6,7}$ is* 1. *This is the only link that can be identified by the original min-system.*

The last part of CTB (Line 11-Line 22) is to find out all the identifiable links in the min-system and divide other unidentifiable links into the disjoint sets according to their greatest lower bound. In this example, all the covered links can be divided into two disjoint nonempty sets S_1 and S_2. The final result is shown in Table I.

Based on CTB, we can prove the necessary and sufficient condition for a link to be identifiable. The sufficient condition is straightforward. The necessary condition can be easily proved with contradiction.

Lemma 2. *Assume that a min-system has n variables and m min-equations, $\mathbf{R} \wedge \mathbf{x} = \mathbf{b}$, where $\mathbf{R}$ is an $m \times n$ Boolean matrix, $\mathbf{x} = \begin{pmatrix} x_1 & \dots & x_n \end{pmatrix}^\intercal$, and $\mathbf{b} = \begin{pmatrix} b_1 & \dots & b_m \end{pmatrix}^\intercal$. A link l is identifiable in this min-system iff there is a path P containing link l with end-to-end bandwidth b_P, such that $\forall l' \in P \setminus l$, the greatest lower bandwidth bound of l' derived from CTB is $b_{l'}$ where $b_{l'} > b_P$.*

IV. On the Hardness of Path Construction for the (Global) Tightest Bound

In the previous section, the tightest error bound of every link is conditional on a given min-system. The tightest error bound is called the *global* tightest error bound if it is the smallest among all possible error bounds derived for the link with different sets of MPs. Obviously, the error bound derived with all possible MPs is the global tightest error bound. Nevertheless, the total number of possible MPs may be huge, and it is well known that listing MPs between two monitors is $\#P$-complete [15]. We hence need to answer two questions: (1) Is it possible to find the error bound identical to the global tightest error bound without listing all possible MPs? (2) Can we design a method to reduce the number of MPs to achieve the error bound close to the *global* tightest error bound? In the rest of the paper, the tightest error bound by default means the global tightest error bound unless stated otherwise.

First of all, we *only need to study the bandwidth tomography with two monitors*. This is because if there are multiple monitors, we can introduce two virtual monitors such that a virtual monitor only has virtual links of bandwidth b_{max} to connect each (physical) monitors. Then the multi-monitor bandwidth tomography problem is reduced to bandwidth tomography problem with the two virtual monitors. The concept of virtual monitors was also used in [19] to simplify theoretical analysis.

In this section, we show the negative answer to the first question. For this, we only need to construct a scenario and prove in this scenario that we must list all possible MPs between the two monitors in order to find the tightest error bound for every link. The proof needs two preliminary results: Lemma 3 and Theorem 1.

Lemma 3. *For a given network $\mathcal{G}$ and two monitors, assume that $\mathcal{P}_m$ is a set of MPs that covers all links in $\mathcal{G}$. With CTB, we can obtain d nonempty sets $\{S_{b'_j}\}, j = 1, 2, \dots, d$ and corresponding identifiable links from $\mathcal{P}_m$. When a new MP P_{m+1} is added, we denote its end-to-end bandwidth as b_{m+1} and the set of MPs as $\mathcal{P}_{m+1}$. Let $b'_{\min}, b'_{\max}$ denote the minimum and maximum $b'_j (j = 1, 2, \dots, d)$ whose $S_{b'_j}$ contains at least one link on P_{m+1}, respectively. Denote $S_{update} = \{l | l \in P_{m+1}, l \text{ is not identifiable within } \mathcal{P}_m\}$, i.e., a set of links whose tightest error bounds may be updated.*

If $|S_{update}| > 0$, we need to update the d nonempty sets based on the value of b_{m+1}:

- ***Case a:*** *If $b_{m+1} \notin \{b'_j, j = 1, 2, \dots, d\}$:*
  - ☐ ***Case a1:*** *If $b_{m+1} > b'_{\max}$, we move all the links in S_{update} from their original sets to a new nonempty set $S_{b_{m+1}}$.*
  - ☐ ***Case a2:*** *If $b_{m+1} < b'_{\max}$, let b'_{a2} denote the minimum $b'_j (j = 1, 2, ..., d)$ which is strictly larger than b_{m+1}. Denote $S_{remained} = \bigcup_{b'_j \geq b'_{a2}} S_{b'_j}$. If $|S_{update} \setminus S_{remained}| > 0$, we move all the links in $S_{update} \setminus S_{remained}$ from their original sets to a new nonempty set $S_{b_{m+1}}$.*
- ***Case b:*** *If $b_{m+1} \in \{b'_j, j = 1, 2, \dots, d\}$:*
  - ☐ ***Case b1:*** *If $|S_{update} \cap S_{b'_{\min}}| = 1$, the link in $S_{update} \cap S_{b'_{\min}}$ is identifiable.*
  - ☐ ***Case b2:*** *If $|S_{update} \cap S_{b'_{\min}}| > 1$, we denote $S_{remained} = \bigcup_{b'_j \geq b_{m+1}} S_{b'_j}$. If $|S_{update} \setminus S_{remained}| > 0$, we move all the links in $S_{update} \setminus S_{remained}$ from their original sets to $S_{b_{m+1}}$.*

After the update, if any updated set only has one link, the link in it is identifiable.

The updated nonempty sets and the new identifiable links are the same as those obtained from $\mathcal{P}_{m+1}$ with CTB.

Lemma 3 indicates that we can perform sequential update on the (conditional) tightest error bounds when a new MP is constructed. It is easy to see that Lemma 3 is essentially a different way to perform "variable elimination" (Line 3-Line 10 of CTB) on $\mathcal{P}_{m+1}$.

Theorem 1. *For a given network $\mathcal{G}$ and two monitors, assume that $\mathcal{P}_m$ is a set of MPs. With CTB, we can obtain a group of d nonempty sets $\{S_{b'_j}\}, j = 1, 2, \dots, d$ and the identifiable links from $\mathcal{P}_m$.*

$\mathcal{P}_m$ can obtain the greatest lower bound of each link in $\mathcal{G}$ if and only if it satisfies the following three conditions:

1) *$\mathcal{P}_m$ covers all links in $\mathcal{G}$;*
2) *For any other d' nonempty sets derived from a different set of paths $\mathcal{P}'_{m'}$ by CTB, $d \geq d'$;*

3) *Any new MP P_{m+1} will not cause the movement of link(s) between two distinct nonempty sets by Lemma 3.*

Adding one more condition, we can find all identifiable links:

4) *The number of identifiable links by $\mathcal{P}_m$ is maximum.*

Theorem 1 gives us criteria to determine whether a set of paths $\mathcal{P}_m$ can obtain the tightest error bound of each link in the network $\mathcal{G}$. Nevertheless, in the context of bandwidth tomography, the bandwidth of each link is unknown before hand. As such, the criteria only serves as a guideline. It does not warrant a polynomial-time solution to the first question raised in this section. With construction (using the concept of graph cut in particular), we can show that there is a class of special cases in which we have to probe all possible paths to obtain the tightest error bounds.

Theorem 2. *In the worst case, it is impossible to derive the error bound identical to the global tightest error bound without listing all possible MPs.*

Theorem 2 means that finding global tightest error bound for links is $\#P$-complete [15]. To tackle the challenge, we adopt a reinforcement learning approach that utilizes the special knowledge in bandwidth tomography for effective learning.

V. A Reinforcement Learning Approach for Path Construction

In this section, we investigate how to build MPs step by step, based on which we can derive error bounds close to the global tightest error bounds. Constructing MPs sequentially in our context is similar to playing a chess game: once an MP is built and its end-to-end bandwidth is probed, we cannot regret if the MP is not able to help reduce the error bounds, since the cost involved in the measurement has occurred. Therefore, we borrow the similar idea in developing a chess game and use the special knowledge in bandwidth tomography as well as off-policy[5] reinforcement learning for constructing MPs.

Note that in Section IV, to ease theoretical analysis, we showed that we only need to analyze the case of two (virtual) monitors. This section does not need the concept of virtual monitors and all monitors are referred as physical monitors.

A. Special Knowledge in Bandwidth Tomography for Action Design

Based on Lemma 3 and Theorem 1, it is easy to have the following proposition:

Proposition 1. *For a set of MPs $\mathcal{P}$, the sufficient and necessary conditions for it to give the global tightest error bounds are:*

i) *For every link in the network $\mathcal{G}$, it must be contained in at least one MP in $\mathcal{P}$, i.e., $\mathcal{P}$ covers the whole network.*
ii) *For any link l_I that is identifiable, l_I must be identified by the min-based system formed by $\mathcal{P}$.*
iii) *For any link l_U that is unidentifiable, if there exists an MP P such that 1) l_U is contained in it and 2) every other link on it has bandwidth no smaller than that of l_U, P must be included in $\mathcal{P}$.*

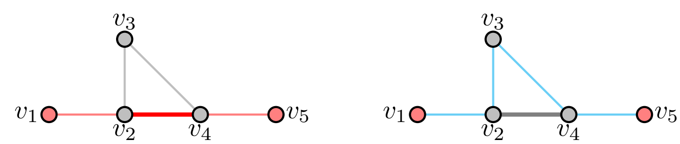
[5]Off-policy means that learning is from data "off" the target policy, i.e., the policy being learned about [20].

Proposition 1 gives insights on how to take proper actions in a reinforcement learning-based MP construction method.

Analysis for identifiable links: Based on Proposition 1, we can see that in the effort of obtaining the global tightest error bounds, if a link l is identifiable, we wish the set of MPs can identify it. We first give a sufficient condition for identifying identifiable links in the following lemma.

Lemma 4. *For an identifiable link l_I, a sufficient condition to identify it is that there are at least one pair of MPs (P_1, P_2) satisfying the following conditions:*

i) *(I-structure): for every link l in P_1 except l_I, $l \in P_2$ (as illustrated in Fig. 3).*
ii) *The measurement value of P_1 is smaller than that of P_2.*



(a) (b)

Fig. 3: Illustration of *I-structure*: P_1 and P_2 are marked in red and cyan, respectively; every link of P_1 except $l_{2,4}$ is also a link of P_2.

Note that knowing the sufficient condition for identifiability is not equivalent to having a polynomial-solution to determine whether a given link is identifiable or not. For a given link l, we do not know in advance which MPs will satisfy conditions in Lemma 4, since the measurement value of an MP can only be revealed after a trail (sending probes along the MP). However, we are able to select MPs that satisfy the topological requirement (i), i.e., when we select a new MP P_{new} into the measurement path set $\mathcal{P}$, we could try to select a P_{new} that forms an *I-Structure* with an MP in $\mathcal{P}$. This knowledge helps to (gradually) build the appropriate $\mathcal{P}$. More details on the action designed based on the above analysis are described in Section V-B.

Analysis for unidentifiable links: For an unidentifiable link l_U, even if we include all possible MPs in $\mathcal{G}$, l_U still cannot be identified, and the possible tightest error bound of l_U is given by the following lemma.

Lemma 5. *For an unidentifiable link l_U, the possible tightest bound is $[b_{l_U}, b_{max}]$, where b_{l_U} is the bandwidth value of l_U and b_{max} is the maximum bandwidth over all links in the network.*

Based on Lemma 5, to achieve the best error bound for an unidentifiable link l_U, we need to find an MP P that satisfies 1) P includes l_U, and 2) every other link on P has bandwidth no smaller than that of l_U. In other words, we wish the bandwidth of every other link along the path to be as large as possible, so that b_{l_U} may become the smallest and thus the measurement value of P. In this way, l_U can achieve its best error bound. Based on this analysis, we design the corresponding actions in Section V-B.

B. Guided Sequential Path Construction (GSPC)

In reinforcement learning (RL), an *agent* takes *actions* and interacts with the *environment*, which gives the agent feedback for the action in form of *reward* [20]. The goal of the agent is to maximize the cumulative total reward in the long run. An RL problem is usually cast in the framework of Markov Decision Process (MDP) [20], where the agent can make action decisions round-by-round based on the current state of the environment. In the context of constructing MPs for bandwidth tomography, the core RL elements are:

- **Environment**: The environment consists of the network topology $\mathcal{G}$ and its (hidden) bandwidth value b_i on each link l_i.
- **Agent**: It is a controller that decides which MP to traverse at each state.
- **Action**: An action a_t of the agent means constructing an MP and measuring the bandwidth of the MP.
- **Policy**: It is a mapping from the agent's perceived states of the environment to the actions to be taken when in those states.
- **State**: A state is defined as $s_t = \mathcal{P}_t$, where $\mathcal{P}_t$ denotes the set of constructed MPs up to round t.
- **Reward**: The reward for taking action a_t at state s_t is the *negative* total error bound computed with CTB over $\mathcal{P}_{t+1}$.

We adopt a model-free off-policy RL approach, a variant of Q-learning, and use Guided Sequential Path Construction (*GSPC*), to guide the agent to make better decisions. Off-policy means that the agent is trained with offline simulated networks at *each step*. The simulated networks have the *same* topology of the target network, i.e., the network for which we need to construct MPs. As time rolls out, we can either identify or derive the bounds of links in the target network with existing MPs built so far. For each identified link in the target network, we set the same bandwidth value for the corresponding link in each simulated network; for each link where we only know the bounds in the target network, we set a random number uniformly distributed within the lower and upper bounds for the corresponding link in each simulated network. The rationale of this offline training is that if the policy is learned from many simulated networks of the same topology, the policy should (statistically) work well for the target network as well. After this offline training, the agent updates the policy, i.e., action-value function $Q(s_t, a_t)$, iteratively to quantify the predicated quality of taking action a_t at state s_t. The process repeats until the designated number of MPs is reached.

One special difficulty is to control the dimension of the action space, because the general term of action "generating a new path" would result in exponential number of actions. Fortunately, the special knowledge introduced in the previous section can help us design the appropriate action space.

Action Space Design*:* The actions are divided into 2 phases. *Phase* 1: before the network $\mathcal{G}$ is covered by the selected MPs (i.e., not every link of $\mathcal{G}$ is included in at least one MP):

- *Action-Random-b* (AR_b): randomly select an MP between 2 random monitors, and guarantee that at least 1 uncovered link is contained in this MP.
- *Action-I-b* (AI_b): utilize the smallest MP P_{smp}, i.e., MP with the smallest measurement value in $\mathcal{P}_t$, and select a new MP P_{new} which contains at least 1 uncovered link and forms an ***I-Structure*** with P_{smp}.
- *Action-U-b* (AU_b): utilize the biggest MP P_{bmp}, i.e., MP with the largest measurement value in $\mathcal{P}_t$, and take a random inner vertex of P_{bmp} v_r such that $P_{bmp} = M_1 \to v_r \to M_2$, $P_{new} = M_1 \to v_r \to M_3$, where M_3 is a third monitor, and the segment $v_r \to M_3$ is node-disjoint with P_{bmp}. In addition, P_{new} contains at least 1 uncovered link.

Phase 2*:* after $\mathcal{G}$ is covered by selected MPs:

- *Action-Random-a* (AR_a): randomly select an MP between 2 random monitors.
- *Action-I-a* (AI_a): utilize the smallest MP P_{smp}, and select a new MP P_{new} that forms an ***I-Structure*** with P_{smp}.
- *Action-U-a* (AU_a): utilize the biggest MP P_{bmp}, and take a random inner vertex of P_{bmp} v_r such that $P_{bmp} = M_1 \to v_r \to M_2$, $P_{new} = M_1 \to v_r \to M_3$, where M_3 is a third monitor, and the segment $v_r \to M_3$ is node-disjoint with P_{bmp}.

Remark 1. *The actions utilize the knowledge of bandwidth tomography. In particular, the actions of AI_b and AI_a target at covering identifiable links (based on Lemma 4), and the actions of AU_b and AU_a target at covering unidentifiable links (based on Lemma 5). The random actions (AR_b and AR_a) give the agent a chance of exploring other possibilities besides guided searches. In the first phase, each action needs to cover at least a new link as the basic requirement. This guarantees that MPs will eventually cover $\mathcal{G}$, instead of hovering over covered links for a long time. Once all links are covered, we do not need to consider this requirement in the second phase.*

The pseudo code of the offline training and *GSPC* is shown in Algorithm 2 and Algorithm 3, respectively.

VI. Performance Evaluation

Since there is no existing work to address the path construction problem for bandwidth tomography, we compare *GSPC* with two naïve methods:

- *Random*: Randomly generate an MP at each round.
- *Diversity Preferred (DP)*: Before the graph is covered, select an MP that consists of at least one uncovered links; after the graph is covered, use *Random* for generating new MPs.

We use the metric total error bound (TEB), defined as the sum of error bounds of all links in the network, to show

Algorithm 2: OfflineTrainer

input : current state in target network s_t, training rounds N, target network $\mathcal{G}$
output: action a_t suggested by offline trainer

get each covered link's bound with Algorithm CTB;
determine phase 1 or phase 2, and according to the phase initialize reward $R_i = 0$ ($i = 1, 2, 3$) for the actions $a_i (i = 1, 2, 3)$, respectively, in the action space;
for *round* $j \leftarrow 1$ **to** N **do**
  initialize a simulated network $\mathcal{SN}_j$ that has the same topology as $\mathcal{G}$ but has no bandwidth value assigned to each link;
  foreach *link* l *in* $\mathcal{SN}_j$ **do**
    if l *in* s_t **then** select a random number r between l's bound interval (calculated with CTB) $[l_{lower\_bound}, l_{upper\_bound}]$;
    /* l_{lower_bound} and l_{upper_bound} for an identified link are the same */
    else choose a random number $r \in [0, b_{max}]$;
    assign r as l's bandwidth value in $\mathcal{SN}_j$;
  end
  foreach $a_i (i = 1, 2, 3)$ **do**
    build a new MP P_{new} by performing action a_i on simulated network $\mathcal{SN}_j$;
    $s'_t = s_t + P_{new}$; /* add the new MP to the set s_t */
    get the total error bound teb under s'_t with CTB;
    $R_i = R_i + teb$;
  end
end
return a_i =*argmax*$_i$(-R_i);

the advantages of *GSPC* over the above two methods. We evaluate their performance with real-world ISP networks (Section VI-A). While we do not exclude the possibility that other better path construction methods might be found in the future, our evaluation results in small-scale simulated networks, where the ground-truth global optimal solutions can be numerically calculated (Section VI-B), suggest that the room for further improving *GSPC* may be marginal.

A. Experiment on Real-World ISP Networks

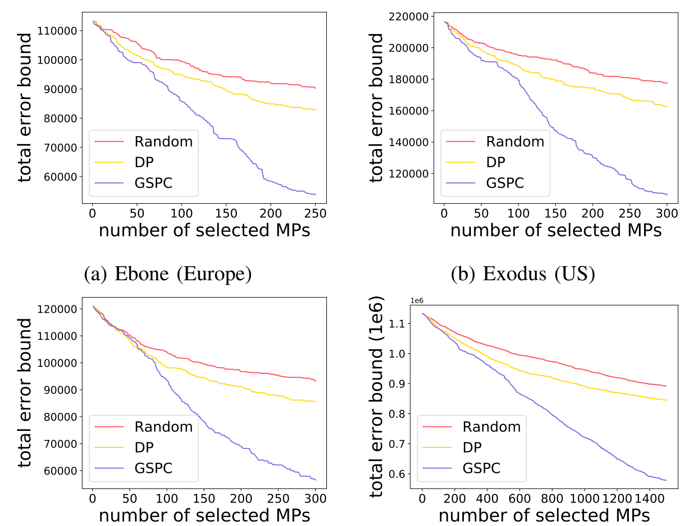
We select four real-world autonomous system (AS) networks collected by the Rocketfuel project [21]. The networks have different sizes, whose parameters are listed in Table II. On each network, the ground-truth bandwidth of each link is set to a random number in $[2, 300]$ for Ebone and Tiscali, and a random number in $[2, 500]$ for Exoduc and Sprintlink.

In order to make sure that the graph can be covered by simple MPs, every dangling points (i.e., node with degree 1) has to be a monitor and each bi-connected components must have at least 2 monitors inside. Besides the above two necessary premises for covering the whole network, a small random number of monitors are deployed based on the size of each network. For each of the ISP network, we conduct the experiment for 3 different methods until designated number of MPs is exhausted (250 for Ebone, 300 for Exodus and Tiscali, 1500 for Sprintlink). These numbers of MPs are chosen based on the observation that the numbers are high enough to cover all the links in the network with both *DP* and *GSPC*.

Algorithm 3: Guided Sequential Path Construction (*GSPC*)

input : $\mathcal{G}$, total number of MPs T, offline training rounds N
output: T measurement paths

Initialize state $s = Null$;
for *episode* $t \leftarrow 1$ **to** T **do**
  observe state s;
  a_t = OfflineTrainer(s, N);
  get an MP P by performing action a_t on the target network $\mathcal{G}$;
  $s = s + P$; /* add the new MP to set s */
end
return s;



(a) Ebone (Europe) (b) Exodus (US)

(c) Tiscali (Europe) (d) Sprintlink (US)

Fig. 4: Performance of random, diversity preferred and GSPC.

We perform multiple runs, which all show similar performance trends. To save space, we show the performance result of one sample run in Fig. 4. We can see that with the growth in number of MPs, the advantage of *GSPC* becomes significant. Compared to *Random* and *DP*, the TEB of *GSPC* is decreased with the fastest speed, regardless of the topology. *DP* outper-

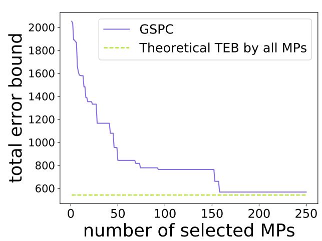
TABLE II: Network parameters

ISP name	$\lvert L\rvert$	$\lvert V\rvert$	Average node degree
Ebone (AS1755)	381	172	4.43
Exodus (AS3967)	434	201	4.31
Tiscali (AS3257)	404	240	3.36
Sprintlink (AS1239)	2268	604	7.51

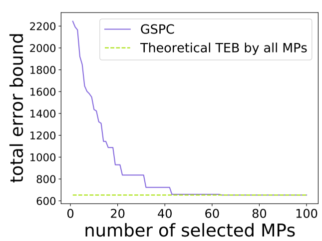
forms *Random* except in the beginning phase. Computing with the MP set at the first round and at the last round, the average TEB reductions of *Random*, *DP* and *GSPC* are 82827, 102077, 197639, respectively. *GSPC* bears a 238% improvement over *Random* and 193% improvement over *DP*.

B. Performance of GSPC v.s. Theoretical Smallest TEB (TS-TEB)

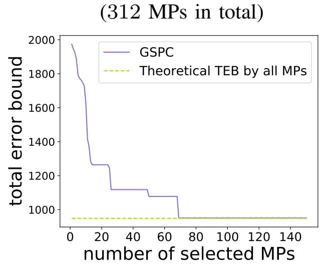
From the previous experiment results, *GSPC* has significantly better performance than the two baseline methods. Nevertheless, due to the large network size, we cannot afford finding the theoretical smallest TEB (TS-TEB), i.e., TEB calculated with the set of all possible MPs, and thus we still do not know how close the performance of *GSPC* is to TS-TEB. For this study, we adopt 4 randomly-generated small networks where we can practically list all MPs. For each network, we assign each link's bandwidth value a random value in $[2, 100]$ and randomly select 2 nodes as monitors. We generate all simple paths between the monitors, based on which the TS-TEB is computed. We then conduct *GSPC* and compare its performance with TS-TEB. We can see that in two networks (Figs. 5 (b) and (c)) *GSPC* achieves TS-TEB with a much smaller number of MPs (the total number of all possible MPs listed in the captions). In addition, the gap between the TEB with *GSPC* and TS-TEB is very small in the other two networks (Figs. 5 (a) and (d)).



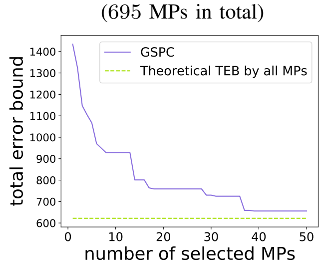
(a) Random network I
(312 MPs in total)



(b) Random network II
(695 MPs in total)



(c) Random network III
(252 MPs in total)



(d) Random network IV
(85 MPs in total)

Fig. 5: Compare the performance of *GSPC* with theoretical smallest TEB (TS-TEB).

VII. Related Work

Since Vardi introduced the concept of network tomography in 1996, extensive research has been done in this domain [2]–[5], [17], [22]–[28]. Most work on network tomography focuses on additive metrics. Regarding bandwidth tomography which involves a min-system, very few can be found in the literature due to the difficulties introduced in Section I.

Extensive research can be found on end-to-end bandwidth estimation [7], and many measurement tools have been developed for this purpose, e.g., *pathchar*, *clink*, *pchar*, and *bfind*. This category of research mainly focuses on *one* end-to-end bandwidth measurement. They serve as the basic building block (w.r.t. end-to-end path bandwidth) for bandwidth tomography but have a large gap towards bandwidth tomography.

Boolean-based network tomography [10] that identifies link failure or link congestion is loosely relevant. Boolean-based tomography, however, is quite different from bandwidth tomography, since bandwidth tomography is to infer the bandwidth *value* rather than the binary status of individual links.

The only work similar to bandwidth tomography is *service curve decomposition* in the deterministic network calculus domain [11], [12]. Service curve, in a nutshell, denotes the *cumulative* service amount provided to a given traffic flow. Due to the cumulative nature in its definition, time plays a critical role, making the problem of service curve decomposition orthogonal to the bandwidth tomography problem studied herein. So far, only line topology and tree topology were studied for service curve decomposition.

VIII. Conclusion

Solving the open problem of bandwidth tomography, this paper has made two novel contributions in the field of network tomography. *First*, it presented a systematic theoretical investigation on the inverse problem of min-system. Unlike additive metrics [1], [18], [19], bandwidth uses the min operation, which only records the minimum bandwidth along a path and thus results in a great loss of link-level bandwidth information. No existing mathematical tool has been found effective to recover the link-level bandwidth from the measurements of a given set of end-to-end paths. We, for the first time, designed a variable elimination algorithm, that calculates the tightest bounds for bandwidth of links in polynomial time by listing all possible intervals and assigning variables into their corresponding intervals. *Second*, to tackle the hardness of constructing measurement paths for deriving the global tightest error bounds, we proposed an off-policy RL-based framework that utilizes the special knowledge in bandwidth tomography and integrates offline training and online prediction for building measurement paths in the target network. Evaluation results over real-world ISP as well as simulated networks demonstrate the superior performance of this method over other baseline path construction methods. As our future research, we plan to use this RL-based framework to solve the measurement path construction problem in other network tomography contexts.

Acknowledgement

This work was partially supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) under Discovery Grant RGPIN-2018-03896, and Hong Kong Research Grant Council under GRF 11204917.

REFERENCES

[1] Y. Vardi, "Network Tomography: Estimating Source-Destination Traffic Intensities from Link Data," *Journal of the American Statistical Association*, vol. 91, no. 433, pp. 365–377, 1996.

[2] E. Lawrence, G. Michailidis, V. Nair, and B. Xi, "Network tomography: A review and recent developments," *Ann Arbor*, vol. 1001, no. 48, pp. 109–1107, 2006.

[3] L. Ma, T. He, K. K. Leung, A. Swami, and D. Towsley, "Inferring link metrics from end-to-end path measurements: Identifiability and monitor placement," *IEEE/ACM Transactions on Networking (TON)*, vol. 22, no. 4, pp. 1351–1368, 2014.

[4] W. Dong, Y. Gao, W. Wu, J. Bu, C. Chen, and X. Y. Li, "Optimal Monitor Assignment for Preferential Link Tomography in Communication Networks," *IEEE/ACM Transactions on Networking*, vol. 25, no. 1, pp. 210–223, 2017.

[5] R. Yang, C. Feng, L. Wang, W. Wu, K. Wu, J. Wang, and Y. Xu, "On the optimal monitor placement for inferring additive metrics of interested paths," in *IEEE INFOCOM*, Honolulu, HI, April 2018.

[6] 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," *IEEE/ACM Transactions on Networking*, 2020.

[7] S. S. Chaudhari and R. C. Biradar, "Survey of bandwidth estimation techniques in communication networks," *wireless personal communications*, vol. 83, no. 2, pp. 1425–1476, 2015.

[8] K. Lai and M. Baker, "Measuring link bandwidths using a deterministic model of packet delay," in *Proceedings of the conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, 2000, pp. 283–294.

[9] N. Hu, L. Li, Z. M. Mao, P. Steenkiste, and J. Wang, "Locating internet bottlenecks: Algorithms, measurements, and implications," *ACM SIGCOMM Computer Communication Review*, vol. 34, no. 4, pp. 41–54, 2004.

[10] N. Bartolini, T. He, V. Arrigoni, A. Massini, F. Trombetti, and H. Khamfroush, "On fundamental bounds on failure identifiability by boolean network tomography," *IEEE/ACM Transactions on Networking*, vol. 28, no. 2, pp. 588–601, 2020.

[11] A. Rizk, "Network tomography using min-plus system theory," *Master's Thesis, TU Darmstadt*, 2008.

[12] A. Rizk and M. Fidler, "On the identifiability of link service curves from end-host measurements," in *International Conference on Network Control and Optimization*. Springer, 2008, pp. 53–61.

[13] F. Baccelli, G. Cohen, G. Olsder, and J. Quadrat, "Synchronization and linearity - an algebra for discrete event systems," *The Journal of the Operational Research Society*, vol. 45, 01 1994.

[14] J.-Y. Le Boudec and P. Thiran, *Network calculus: a theory of deterministic queuing systems for the internet*. Springer Science & Business Media, 2001, vol. 2050.

[15] L. G. Valiant, "The complexity of enumeration and reliability problems," *SIAM Journal on Computing*, vol. 8, no. 3, pp. 410–421, 1979.

[16] M. Jain and C. Dovrolis, "Pathload: A measurement tool for end-to-end available bandwidth," in *In Proceedings of Passive and Active Measurements (PAM) Workshop*. Citeseer, 2002.

[17] L. Ma, T. He, K. K. Leung, D. Towsley, and A. Swami, "Efficient Identification of Additive Link Metrics via Network Tomography," in *2013 IEEE 33rd International Conference on Distributed Computing Systems*, 2013, pp. 581–590.

[18] C. Feng, L. Wang, K. Wu, and J. Wang, "Bound-based network tomography with additive metrics," in *IEEE INFOCOM*, Paris, France, April 2019.

[19] L. Ma, T. He, K. K. Leung, A.Swami and D.Towsley, "Identifiability of Link Metrics Based on End-to-end Path Measurements," in *Proceedings of the 2013 Conference on Internet Measurement Conference*. New York, NY, USA: ACM, 2013, pp. 391–404.

[20] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[21] N. Spring, R. Mahajan, D. Wetherall, and H. Hagerstrom, "Rocketfuel: An ISP topology mapping engine," 2002. [Online]. Available: http://research.cs.washington.edu/networking/rocketfuel/

[22] Y. Xia and D. Tse, "Inference of Link Delay in Communication Networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 12, pp. 2235–2248, 2006.

[23] Y. Bejerano and R. Rastogi, "Robust Monitoring of Link Delays and Faults in IP Networks," *IEEE/ACM Transactions on Networking*, vol. 14, no. 5, pp. 1092–1103, 2006.

[24] R. Kumar and J. Kaur, "Practical Beacon Placement for Link Monitoring Using Network Tomography," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 12, pp. 2196–2209, 2006.

[25] A. Gopalan and S. Ramasubramanian, "On Identifying Additive Link Metrics Using Linearly Independent Cycles and Paths," *IEEE/ACM Transactions on Networking*, vol. 20, no. 3, pp. 906–916, 2012.

[26] Gopalan, Abishek and S.Ramasubramanian, "On the maximum number of linearly independent cycles and paths in a network," *IEEE/ACM Transactions on Networking*, vol. 22, no. 5, pp. 1373–1388, 2014.

[27] Y. Gao, W. Dong, W. Wu, C. Chen, X. Y. Li, and J. Bu, "Scalpel: Scalable Preferential Link Tomography Based on Graph Trimming," *IEEE/ACM Transactions on Networking*, vol. 24, no. 3, pp. 1392–1403, 2016.

[28] L. Ma, T. He, K. K. Leung, A. Swami, and D. Towsley, "Monitor placement for maximal identifiability in network tomography," in *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, 2014, pp. 1447–1455.