Bound-based Network Tomography with Additive Metrics

Cuiying Feng*, Luning Wang†, Kui Wu*, Jianping Wang†
*Department of Computer Science, University of Victoria, Canada
†Department of Computer Science, City University of Hong Kong, Hong Kong
miriam10@uvic.ca, luninwang4-c@my.cityu.edu.hk, wkui@uvic.ca, jianwang@cityu.edu.hk

***Abstract*—Network performance tomography infers performance metrics on internal network links with end-to-end measurements. Existing results in this domain are mainly Boolean-based, i.e., they check whether or not a link is identifiable, and return the exact value on identifiable links. If a link is not identifiable, Boolean-based solution gives no performance result for the link. In this paper, we extend Boolean-based network tomography to bound-based network tomography where the lower and upper bounds are derived for unidentifiable links. We develop an efficient algorithm to obtain the tightest total error bound, and present a solution that can significantly reduce the total number of measurement paths required for deriving the tightest total error bound. Furthermore, we propose a method to deploy a new monitor over existing ones such that the total error bound could be maximally reduced. Compared to the random monitor deployment and the monitor deployment that maximizes the total number of identifiable links, our monitor deployment method can lead to up to 15 and 2.4 times more reduction on total error bound, respectively.**

I. Introduction

Network performance monitoring has been a critical yet tedious task for any Internet service provider (ISP) to guarantee the smooth operation of its network. Due to the large size of a network, it is generally prohibitive to monitor all links. Recent technical development on network slices allows ISPs to dynamically form different virtual networks, each for a dedicated network application. Monitoring and validating the performance of a virtual network are basic requirements for the providers of virtual networks. Due to the dynamic changes of network configuration, it is extremely challenging to directly measure the performance of virtual links. A well-known strategy is to infer the performance of physical/virtual links via end-to-end measurements. This solution is termed as network performance tomography [1]–[4].

Existing solutions for network performance tomography are mostly *Boolean based*: they determine whether or not a given set of links/paths are identifiable[1] (i.e., Boolean), and if yes, the inferred values on the link/paths of interest are returned. If a link/path is not identifiable, no useful information about the performance on that link/path is provided in Boolean-based network tomography.

Given a set of measurements, a link/path may not be identifiable. However, the performance upper bound and lower bound on that link/path can be derived, which is also important to the ISPs. After all, what an ISP cares is whether the SLA with the customers is violated or not, i.e., whether the performance metric is within the performance bound specified in the SLA. We thus shift our focus towards performance bounds: what are the tightest upper and lower bounds of the performance metric on the links/paths of interest? We call this shift of focus as *bound-based* network performance tomography.

Clearly, *bound-based* network tomography is more general and extends *Boolean-based* network tomography, because in *bound-based* network tomography the upper and lower bounds are the same for identifiable links/paths. Whenever a link/path is not identifiable, bound-based network tomography tells the upper and lower values on the link/path. In practice, bound-based network tomography offers more information to ISPs to diagnose performance problems and manage its network resources with less measurement overhead.

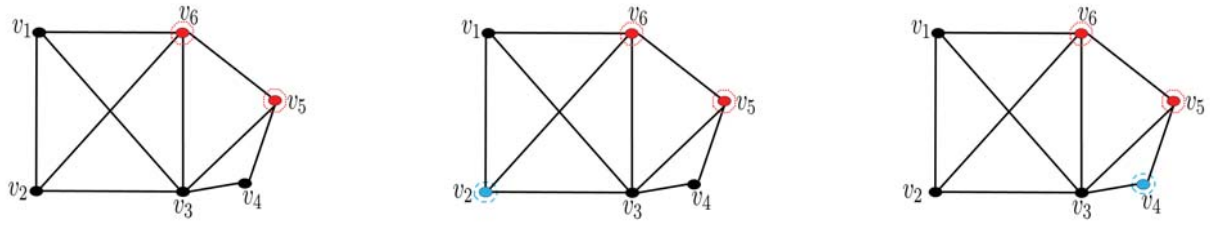
We study bound-based network tomography with *additive* (e.g., delay) metrics for links, where the metric of an end-to-end path is the sum of metric of all the links on the path. Bound-based network tomography, while extending Boolean-based network tomography only slightly in concept, drastically changes the landscape of solution space as illustrated in the following example.

Motivating Example: Fig. 1 shows an example network with 6 nodes and 10 links. Assume that the link metrics are $\{x_{1,2} = 7, x_{2,3} = 3, x_{3,4} = 10, x_{4,5} = 13, x_{5,6} = 8, x_{1,6} = 1, x_{1,3} = 5, x_{2,6} = 4, x_{3,6} = 6, x_{3,5} = 3\}$, which are unknown in advance and should be inferred through end-to-end measurements with monitors. Assume that initially two monitors are placed at nodes v_5 and v_6 (in red color), as shown in Fig 1 (a).

With the Boolean-based network tomography introduced in [2], only links $l_{5,6}$ and $l_{1,2}$ are identifiable. The metrics on other links remain unknown. In contrast, we can derive the bounds on unidentifiable links, using the bound-based network tomography disclosed later in Section III, as $\mathcal{B}(x_{3,5}) = [0, 7]$, $\mathcal{B}(x_{3,4}) = [0, 27]$, $\mathcal{B}(x_{2,6}) = [3, 10]$, $\mathcal{B}(x_{3,6}) = [2, 9]$, $\mathcal{B}(x_{1,3}) = [2, 9]$, $\mathcal{B}(x_{4,5}) = [0, 27]$, $\mathcal{B}(x_{2,3}) = [0, 7]$, $\mathcal{B}(x_{1,6}) = [0, 7]$. If we denote the *error bound* of a link metric as the inferred upper bound minus the inferred lower bound, the *total error bound* of all links is 96.

After we obtain the tightest lower and upper bounds on unidentifiable links, a natural question is: how to further tighten the total error bound by deploying an extra monitor?

[1]Identifiable means the value on a link/path can be uniquely determined.



(a) (b) (c)

Fig. 1: An example: (a) initial monitor deployment, (b) new monitor deployment with Boolean-based network tomography, (c) new monitor deployment with bound-based network tomography.

Boolean-based network tomography cannot answer this question directly. One may wonder if the monitor placement solution to minimize the total number of unidentifiable links [2] would also lead to the tightest total error bound. This solution is shown in Fig 1 (b) where the new monitor is placed at v_2 (in blue color). The total number of unidentifiable links is reduced to 2, which are $l_{3,4}$ and $l_{4,5}$ with new bounds $\mathcal{B}(x_{3,4}) = [0, 23]$ and $\mathcal{B}(x_{4,5}) = [0, 23]$, respectively. The total error bound is reduced to 46.

With the bound-based network tomography which will be presented in this paper, we should deploy the new monitor at v_4, as shown in Fig 1 (c), which reduces the total error bound to 16 even if the total number of unidentifiable links is 4 (i.e., $\mathcal{B}(x_{1,6}) = [0, 4]$, $\mathcal{B}(x_{2,3}) = [0, 4]$, $\mathcal{B}(x_{1,3}) = [2, 6]$, and $\mathcal{B}(x_{2,6}) = [3, 7]$). Clearly, bound-based network tomography finds a better solution w.r.t. reducing the total error bound.

Motivated by the above example, we aim to answer the following two questions: (1) given a set of deployed monitors, *how to infer the tightest lower and upper bounds of unidentifiable links?* (2) given a set of deployed monitors, *how to deploy extra monitors to minimize the total error bound?*

To answer the first question, we first study how to derive the tightest upper and lower bounds for unidentifiable links given a set of deployed monitors and end-to-end measurements. We then study how to use the minimum number of end-to-end measurements among a set of deployed monitors to achieve the tightest total error bound.

The main challenge of the second problem is that the reduced total error bound remains unknown until we have actually deployed the new monitor and collected new measurement results. Thus, the second problem is theoretically unsolvable, because we are not allowed to use the "trial-and-error" method and thus have no way to compare the total error bound reduction if deploying the new monitor at different places. To avoid this situation, we replace each bound interval obtained by existing monitor deployment with the mean value of the lower and upper bounds. The practical meaning of this processing is to estimate end-to-end delays that we would expect with the new monitor and existing monitors, before physically deploying the new monitor. With this pre-processing, we are able to identify the optimal location to place a new monitor to minimize the total error bound.

The contributions of the paper are as follows:

- Given a set of deployed monitors and end-to-end measurements with the monitors, we present a solution to find the tightest total error bound for unidentifiable links.
- Given a set of deployed monitors, the number of possible measurement paths could be huge. We propose a method that minimizes the total number of measurement paths, with which the tightest total error bound can be derived[2].
- Given a set of deployed monitors and aforementioned preprocessing, we develop a monitor deployment method so that a newly-added monitor can maximally reduce the total error bound.
- We thoroughly evaluate our monitor deployment method by comparing it with two benchmarks: random deployment and the deployment that maximizes the total number of identifiable links. The results show that our monitor deployment method can significantly reduce the total error bound in bound-based network topography.

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 that in [5], to derive the delay on identifiable links. In this paper, we are interested in the delay bounds on those unidentifiable links.

Following the convention in network performance tomography [2], [5], we introduce basic assumptions and notations:

- $\mathcal{G}$: A connected and undirected graph. Each link has distinct end nodes (i.e., no self loop), and no 2 links in $\mathcal{G}$ connect to the same pair of nodes.
- *Measurement path (MP)*: A non-loop path that only contains 2 monitors at its end nodes.
- *Bi-connected component (BC)*: A maximal subgraph of $\mathcal{G}$ that is either (i) a single link, or (ii) 2-vertex-connected.
- *Tri-connected component (TC)*: A maximal subgraph of $\mathcal{G}$ that is either (i) a circle, or (ii) 3-vertex-connected.
- *Vantage w.r.t. a TC $\mathcal{T}$*: A node that is either (i) a monitor in $\mathcal{T}$, or (ii) a cut node that separates $\mathcal{T}$ from at least one monitor.

[2]Note that this does not necessarily mean we know the exact value of tightest total error bound. The exact value of tightest total error bound is unknown until measurement results are collected following the suggested measurement paths.

- *Exterior links*: the links incident to only 1 vantage in a TC $\mathcal{T}$.
- *2-bridge-cut*: a pair of links that satisfies i) the removal of the pair of links increases the number of connected components in $\mathcal{G}$, ii) if only one link is removed, the number of connected components in $\mathcal{G}$ remains unchanged.
- *2-vertex-cut*: a pair of nodes whose removal disconnects $\mathcal{G}$, but removing only one would not disconnect $\mathcal{G}$.

End-to-end delays along MPs form a linear system. We use the example in Fig. 2 to illustrate the concept. The network has 2 monitors in red color. First, the end-to-end delays along all MPs between the two monitors form the following linear system, where $x_{i,j}$ denotes the delay on link $l_{i,j}$ and w_l denotes the end-to-end delay on MP l.

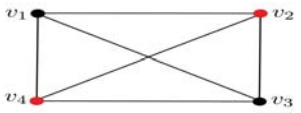


Fig. 2: An example network.

$$\begin{cases} x_{1,2}+x_{1,4}=w_1 \\ x_{2,4}=w_2 \\ x_{2,3}+x_{3,4}=w_3 \\ x_{1,2}+x_{1,3}+x_{3,4}=w_4 \\ x_{2,3}+x_{1,3}+x_{1,4}=w_5 \end{cases} \tag{1}$$

The above linear system could be written into $R'\mathbf{x}=\mathbf{w}'$, where

$$R'=\begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 \end{pmatrix} \tag{2}$$

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

$$\mathbf{w}'=\begin{pmatrix} w_1 & w_2 & w_3 & w_4 & w_5 \end{pmatrix}^\intercal \tag{4}$$

From linear system (1), we can derive that $x_{2,4}=w_2$ and $x_{1,3}=(w_4+w_5-w_3-w_1)/2$. Since we can derive the values of $x_{2,4}$ and $x_{1,3}$, we could move the known metric values to the right-hand-side in (1) to form a new linear system:

$$\begin{cases} x_{1,2}+x_{1,4}=w_1 \\ 0=w_2-x_{2,4} \\ x_{2,3}+x_{3,4}=w_3 \\ x_{1,2}+x_{3,4}=w_4-x_{1,3} \\ x_{2,3}+x_{1,4}=w_5-x_{1,3} \end{cases} \tag{5}$$

Since $x_{2,4}$ only appears in the second equation, it cannot be used to determine the bounds for other unidentifiable links. As such, the second equation $0=w_2-x_{2,4}$ is out of our interest and can be safely removed. The resulting linear system is the final mathematical model that is useful for our purpose.

$$\begin{cases} x_{1,2}+x_{1,4}=w_1 \\ x_{2,3}+x_{3,4}=w_3 \\ x_{1,2}+x_{3,4}=w_4-x_{1,3} \\ x_{2,3}+x_{1,4}=w_5-x_{1,3} \end{cases} \tag{6}$$

which could be simplified as $R\mathbf{x}=\mathbf{w}$ where

$$R=\begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 \end{pmatrix} \tag{7}$$

$$\mathbf{x}=\begin{pmatrix} x_{1,2} & x_{1,3} & x_{1,4} & x_{2,3} & x_{2,4} & x_{3,4} \end{pmatrix}^\intercal \tag{8}$$

$$\mathbf{w}=\begin{pmatrix} w_1 & w_3 & w_4' & w_5' \end{pmatrix}^\intercal \tag{9}$$

where $w_4'=w_4-x_{1,3}$ and $w_5'=w_5-x_{1,3}$.

Remark 1. *In the rest of this paper, the linear system model (LSM), $R\mathbf{x}=\mathbf{w}$, by default refers to the final linear system after the above preliminary processing. We stress that the LSM is non-invertable, because links in the LSM are unidentifiable (otherwise there is no need for bound-based tomography.).*

III. Bound Analysis of Unidentifiable Links

A. Solution Space of a Non-invertable Linear System

Since *LSM* $R\mathbf{x}=\mathbf{w}$ is non-invertable, the rank of R (i.e., the number of linearly independent measurement paths) is less than the number of links in $\mathcal{G}$. In linear algebra [6], the solutions of non-invertable *LSM* form a solution space, which is constructed by free variables. In other words, the solution $\mathbf{x}$ is presented in the form that every other pivot variable (i.e., non-free variable) is the linear combination of free variables.

B. Utilization of Free Variables

To derive the tightest total bound of undetermined variables in the *LSM*, we first introduce the following property of free variables [6]:

Remark 2. *For a non-invertable linear system $R\mathbf{x}=\mathbf{w}$ having n variables, there are different combinations of n' variables that can serve as free variables, where $n'=n-rank(R)$.*

As an example, the solution space of linear system (6) can be represented as

$$\begin{cases} x_{1,2}=w_1-w_5'+x_{2,3} \\ x_{1,4}=w_5'-x_{2,3} \\ x_{3,4}=w_3-x_{2,3} \end{cases} \tag{10}$$

if we select $x_{2,3}$ as the free variable, or

$$\begin{cases} x_{1,4}=w_1-x_{1,2} \\ x_{2,3}=w_5'-w_1+x_{1,2} \\ x_{3,4}=w_1+w_3-w_5'-x_{1,2} \end{cases} \tag{11}$$

if we select $x_{1,2}$ as the free variable.

While the same solution space can be represented in different forms, we need to determine which representation leads to the tightest total error bound. This question is answered in the next section.

C. Steps to Obtain the Tightest Total Error Bound

Let $\mathcal{TEB} = \sum \mathcal{B}(x_i)$ denote the total error bound of a *LSM*, where $\mathcal{B}(x_i)$ is the length of bound interval for x_i. We first define the concept of *natural bound interval*:

Definition 1. ***Natural bound interval (NBI):*** *In an LSM, the natural bound interval of a variable x_i is the interval $[0, b_i]$, where b_i is the (end-to-end) measurement value corresponding to an equation that contains x_i. If there are multiple equations containing the variable x_i, the* ***tightest natural bound interval (TNB)*** *of x_i is the interval $[0, b_{min}]$, where b_{min} is the smallest value among all the (end-to-end) measurements corresponding to those equations.*

1. Obtain the *TNB* of every variable in the *LSM*.
2. Sort the variables in an ascendant order by the upper bound in *TNB*, denoted as $\mathcal{O} = \{x_1^s, x_2^s, \cdots, x_n^s\}$. Find each of the variable combinations, denoted as F_c, satisfying i) the variables in F_c together can serve as free variables, ii) F_c has the tightest total $|TNB|$ ($|TNB|$ is the length of *TNB*). Put all F_c's into a candidate free variable set $\mathcal{CFV}$.
3. For each F_c in $\mathcal{CFV}$, the final bound interval of each free variable in F_c is its *TNB*. Since each pivot variable x_p is a linear combination of the free variables in F_c, we can obtain a bound interval for x_p, denoted as $B_c(x_p)$, by plugging free variables' *TNB*s into the linear combination. The intersection of $B_c(x_p)$ and $TNB(x_p)$ is the final bound interval of x_p.
4. Each F_c in $\mathcal{CFV}$ gives a total error bound $\mathcal{TEB}$. Return the smallest $\mathcal{TEB}$ over all F_c's in $\mathcal{CFV}$.

Remark 3. *Note that in Step 2, not every combination of $n - rank(R)$ variables can be used as free variables. Use the following linear system as an example:*

$$R_1\mathbf{x}_1 = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} \tag{12}$$

could be solved as

$$\mathbf{x} = \begin{pmatrix} w_1 \\ w_2 \\ 0 \\ 0 \end{pmatrix} + x_3 \begin{pmatrix} -1 \\ 0 \\ 1 \\ 0 \end{pmatrix} + x_4 \begin{pmatrix} -1 \\ -1 \\ 0 \\ 1 \end{pmatrix}, \tag{13}$$

where x_3 and x_4 correspond to the last 2 columns in R_1 and they are free variables. However, if we re-organize the above linear system as

$$R_2\mathbf{x}_2 = \begin{pmatrix} 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_3 \\ x_2 \\ x_4 \end{pmatrix} = \begin{pmatrix} w_1 \\ w_2 \end{pmatrix}, \tag{14}$$

then x_2 and x_4 together cannot serve as free variables, because x_1 and x_3 do not form an identity sub-matrix in the reduced row echelon form of R_2. In other words, x_1 and x_3 cannot be the pivot variables together because their corresponding columns in R_2 are linearly dependent.

Remark 4. *The worst-case time complexity for the above search algorithm is $O(\binom{n}{rank(R)})$. Since there are cases that any combination of $n-rank(R)$ variables can be free variables and we need to check all possible combinations of free variables to know the tightest total $|\text{TNB}|$,* ***no other search algorithm can have a smaller worst-case time complexity****. As a fast approximation, after we sort the variables in the second step, we can select variables, following that order, that together serve as free variables, and then use the first free variable combination in Step 3 and omit Step 4.*

D. Theoretical Guarantee

The above method assures that $\mathcal{TEB}$ is minimum. The proof needs the following lemma.

Lemma 1. *For a full measurement matrix R (including all possible MPs), its reduced row-echelon form is a matrix whose element can only be -1, 0, or 1.*

Theorem 1. *The total error bound $\mathcal{TEB}$ obtained in Section III-C is the minimum $\mathcal{TEB}$ that could be derived from the LSM.*

Proof. Due to space limit, we only provide the main idea of the proof.

We first prove the case where any $n - r$ variables ($r = \text{rank}(R)$) can serve as free variables. Assume that the free variable combination providing tightest $\mathcal{TEB}$ is $\{x_{r+1}, x_{r+2}, \ldots, x_n\}$, then the solution form should be arranged as

$$\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_r \\ x_{r+1} \\ x_{r+2} \\ x_{r+3} \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_r \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} + \begin{pmatrix} c_{r+1,1} \\ c_{r+1,2} \\ \vdots \\ c_{r+1,r} \\ 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} x_{r+1} + \begin{pmatrix} c_{r+2,1} \\ c_{r+2,2} \\ \vdots \\ c_{r+2,r} \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} x_{r+2} + \cdots + \begin{pmatrix} c_{n,1} \\ c_{n,2} \\ \vdots \\ c_{n,r} \\ 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix} x_n \tag{15}$$

where the first r variables are pivot variables. Moreover, $c_{i,j} \in \{-1, 0, 1\}$ $(i = r, r+1, \cdots, n,\ j = 1, 2, \cdots, n)$ in (15) according to Lemma 1.

We will show that any other solution form, whose free variables' total $|TNB|$ is strictly less than the total $|TNB|$ of $\{x_{r+1}, x_{r+2}, \ldots, x_n\}$, would lead to a larger total error bound. To change (15) into another solution form, at least one pivot variable should exchange with one free variable. Without loss of generality, we let pivot variable x_r exchange with free variable x_{r+1} which satisfies condition $|TNB(x_{r+1})| < |TNB(x_r)|$ (otherwise the total $|TNB|$ of new free variables is equal to the total $|TNB|$ of new free variables) and show that the resulting total error bound becomes larger.

Exchanging pivot variable x_r and free variable x_{r+1} could be done in the rth equation in (15)

$$x_r = d_r + c_{r+1,r}x_{r+1} + c_{r+2,r}x_{r+1} + \cdots + c_{n,r}x_n$$

where every $c_{i,r}$ $(i = r+1, r+2, \cdots, n)$ is -1, 0 or 1 by Lemma 1, and under our particular assumption (i.e, any $n-r$

variables can be used as free variables), $c_{r+1,r}$ cannot be 0 (otherwise we cannot exchange x_{r+1} with x_r). Afterwards, we could arrive

$$x_{r+1} = \begin{cases} d_r - x_r + \cdots + c_{n,r}x_n, & \text{when } c_{r+1,r} = -1 \\ -d_r + x_r - \cdots - c_{n,r}x_n & \text{when } c_{r+1,r} = 1 \end{cases} \tag{16}$$

Furthermore, Equation (16) can be written into

$$x_{r+1} = \begin{cases} -x_r + C & \text{when } c_{r+1,r} = -1 \\ x_r - C & \text{when } c_{r+1,r} = 1 \end{cases} \tag{17}$$

where $C = d_r + c_{r+2,r}x_{r+2} + \cdots + c_{n,r}x_n$.

By analyzing bound interval changes of x_{r+1} and x_r and their impact on $\mathcal{TEB}$ after such an exchange, we can conclude that the new form leads to $\mathcal{TEB}$ no smaller than that before the exchange. The lengthy analysis is omitted to save space.

For the case where not any $n - r$ variables can serve as the free variables, we can use the same analysis but with a restriction that only eligible free variable combinations are considered.

Lastly, for the situation where there are multiple free variable combinations with the tightest total $|TNB|$, the last step guarantees we will obtain the minimum $\mathcal{TEB}$.

□

IV. Minimizing the Number of Measurement Paths

Given a set of deployed monitors, we can form an *LSM* by listing all possible MPs and use the method in the previous section to obtain the tightest $\mathcal{TEB}$. Nevertheless, the total number of possible MPs is huge; and it is well known that listing MPs between two monitors is $\#P$-complete [7]. Naturally, we need to answer the following question: given a set of deployed monitors, what are the least MPs with which we can derive the tightest total error bound? We stress again that the exact value of tightest total error bound remains unknown until measurement results are collected following the suggested measurement paths.

We first classify the identifiability of links, using the same technique of decomposing the network into tri-connected components (TCs) [2], [4]. We then obtain the smallest set of MPs using the TC tree.

Given a network $\mathcal{G} =< V, L >$ with a pre-determined deployment of k ($k > 2$) monitors, the link identifiability problem could be converted to a 2-monitor problem by introducing two virtual monitors, each connecting to existing monitors with virtual links [2]. Following the same graph construction in [4], we can obtain the extended graph $\mathcal{G}_{new}$ with only 2 virtual monitors m'_1 and m'_2, in which $\mathcal{G}$ is the *interior graph* of $\mathcal{G}_{new}$. Next, we decompose $\mathcal{G}_{new}$ into TCs, which are recorded in a TC tree [8]. According to [4], the resulted TCs have the following topological properties:

1) The two virtual monitors, m'_1 and m'_2, are included in the same TC, denoted by $\mathcal{T}_0$;
2) Each TC $\mathcal{T}$ includes only two vantages. The vantages of $\mathcal{T}_0$ are m'_1 and m'_2, and each of the other TCs includes only one 2-vertex-cut that separates itself from m'_1 and m'_2.
3) All the TCs can be arranged in a *tree* structure, if we treat $\mathcal{T}_0$ as the *root*, the other TCs as *nodes*, and the 2-vertex-cut between the TCs as the *edge*. An example is shown in Fig. 3.

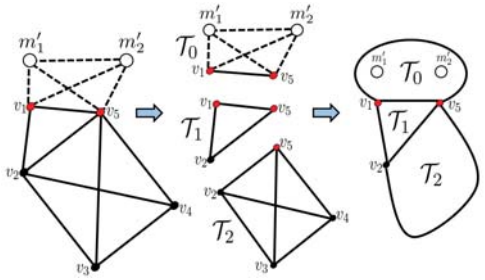


Fig. 3: An example of graph decomposition. Note that v_1 and v_5 are the vantages w.r.t. $\mathcal{T}_1$, and v_2 and v_5 are the vantages w.r.t. $\mathcal{T}_2$.

Given a TC $\mathcal{T}$ with two vantages μ_1 and μ_2, denote $\widetilde{\mathcal{T}}$ as the subgraph of $\mathcal{T}$ obtained by removing the direct link between μ_1 and μ_2 (Note that if there is no real link between μ_1 and μ_2, then a virtual link l_{μ_1,μ_2} was generated during the graph partition process). The TC tree has the following features to help us determine the identifiability of links [4]:

1) Every real link in $\mathcal{T}_0$ is identifiable ($k \geq 3$). Only 3 types of links are unidentifiable in other TCs as follows.
2) Exterior links of a 3-vertex-connected TC $\mathcal{T}_{3vc}$, i.e., links that are incident to only 1 vantage of the $\mathcal{T}_{3vc}$, are unidentifiable.
3) For a 3-vertex-connected TC $\mathcal{T}_{3vc}$, links of a 2-bridge-cut in $\widetilde{\mathcal{T}}_{3vc}$ are unidentifiable.
4) All the links in $\widetilde{\mathcal{T}}_{circle}$, where $\mathcal{T}_{circle}$ is a circle TC, are unidentifiable.

Remark 5. *The identifiability of the direct link between the two vantages can be determined in the parent TC which they belong to. As shown in Fig. 3, v_2 and v_5 are the vantages w.r.t. $\mathcal{T}_2$. The identifiability of link between them can be determined in $\mathcal{T}_1$, following the above procedure.*

From now on, we divide all the TCs into 2 types:

1) *Normal TC*: 3-vertex-connected TC $\mathcal{T}_{3vc}$ without 2-bridge-cut link in $\widetilde{\mathcal{T}}_{3vc}$.
2) *Special TC*: 3-vertex-connected TC $\mathcal{T}_{3vc}$ with 2-bridge-cut link in $\widetilde{\mathcal{T}}_{3vc}$ and circle TC $\mathcal{T}_{circle}$.

Because of the special topological features of these unidentifiable links and the fact that the delay on identifiable links' can be uniquely determined [5], we can construct the smallest set of MPs to obtain the tightest $\mathcal{TEB}$ by the following results.

Theorem 2. *After G_{new} is partitioned into TCs recorded in a TC tree, for a particular non-root TC $\mathcal{T}$ with a parent TC $\mathcal{T}_p$:*

1) When $\mathcal{T}_p$ is normal TC and doesn't share vantage with $\mathcal{T}$:

- *If $\mathcal{T}$ is a 3-vertex-connected TC, which has 2 vantages μ_1 (with n_{μ_1} exterior links incident to it) , μ_2 (with n_{μ_2} exterior link incident to it), and n_p pairs of*

Algorithm 1 Minimal Measurement Path Construction

Input: a network graph $\mathcal{G}$ and deployed monitors
Output: the necessary and sufficient measurement paths to obtain the tightest $\mathcal{TEB}$

1: obtain $\mathcal{G}_{new}$ by adding two virtual monitors and corresponding virtual links [2];
2: partition $\mathcal{G}_{new}$ into tri-connected components, which are recorded in a TC tree [8];
3: create an empty set $\mathcal{MP}$;
4: put all the leaf TCs in a set $\mathcal{L} = \{\mathcal{T}_l^1, \mathcal{T}_l^2, \cdots, \mathcal{T}_l^n\}$, also put their corresponding tandem trees in set $\mathcal{ST} = \{\mathcal{ST}_l^1, \mathcal{ST}_l^2, \cdots, \mathcal{ST}_l^n\}$;
5: **for** each $\mathcal{ST}_l^i$ **do**
6: mark all the TCs from the leaf TC to the first non-root TC with index $1, 2, 3, \cdots, n_i$;
7: **for** $i = 1$ to n_i **do**
8: mark the 2 vantages of $\mathcal{T}_i$ as $\{\mu_1, \mu_2\}$;
9: call $FindPathsInsideTC(\mathcal{T}_i)$ and record the returned paths in $\mathcal{P}_i$.
10: **for** any monitor pair (m_1, m_2) **do**
11: identify a one-to-one, monitor-vantage mapping, denoted as (m_1, u_1) and (m_2, u_2) without loss of generality;
12: call $FindPathsFromMonitorToTC(\mathcal{T}_i)$ and record the returned paths from m_1 to u_1 in $\mathcal{P}_{m_1,\mu_1}$ and paths from m_2 to u_2 in $\mathcal{P}_{m_2,\mu_2}$;
13: construct an MP by concatenating three paths from the sets, $\mathcal{P}_{m_1,\mu_1}$, $\mathcal{P}_i$, and $\mathcal{P}_{m_2,\mu_2}$, respectively; construct all MPs by finding all possible path concatenations;
14: put all the constructed MPs into $\mathcal{MP}$;
15: **return** $\mathcal{MP}$;

non-exterior 2-b-c link (since some 2-b-c links could also be exterior link), then the minimal number of MPs needed to derive the tightest $\mathcal{TEB}$ for links in $\mathcal{T}$ is $2^{n_p} \times n_{\mu_1} \times n_{\mu_2}$.

- *If $\mathcal{T}$ is a circle TC, then the minimal number of MPs needed to derive the tightest $\mathcal{TEB}$ in $\mathcal{T}$ is 1.*

2) All the other cases, i.e., i) $\mathcal{T}_p$ is normal TC but share vantage with $\mathcal{T}$ or ii) $\mathcal{T}_p$ is special TC, need special treatment[3].

Finally, the minimal total number of MPs is then obtained by summing over the number of MPs assoicated with each non-root TC.

Theorem 2 only tells the minimum number of MPs. The steps of constructing these MPs is given in Algorithm 1. Ignoring the complexity in partitioning the network into TCs, the worst-case time complexity of Algorithm 1 is $O(\alpha N_{\mathcal{T}}^2)$, where α is the maximum TC size among all the TCs and $N_{\mathcal{T}}$ is the total number of TCs.

[3]While we omit the detail due to space limit, the main idea is to look up the ancestor TCs of $\mathcal{T}_p$, one by one towards the root, until reaching the first normal TC.

Remark 6. *In Line 4 of Algorithm 1, we call the path from the root to the leaf node as a **tandem tree**, with each node on the tree representing a TC in the network. In Line 10, all monitors are in the root TC, i.e., $\mathcal{T}_0$. In Line 13, two paths can be concatenated only if they share a common end node. Algorithm 1 needs to call two functions $FindPathsFromMonitorToTC$ and $FindPathsInsideTC$, whose details are omitted due to page limit. Simply put, $FindPathsFromMonitorToTC$ is to find a path from the monitor to a vantage point of TC $\mathcal{T}_i$, which travels all TCs along the TC tree and uses one of the vantage point when passing an intermediate TC. $FindPathsInsideTC$ is to find a path between the two vantage points of $\mathcal{T}_i$, which uses links adjacent to the vantage points inside $\mathcal{T}_i$. An example is shown in Fig. 4.*

Remark 7. *Due to the topology features of the TC tree, the constructed set of MPs is necessary and sufficient because (1) some TC would not be covered if any MP in the set is remove (i.e., necessary), and (2) any other MP not in the set is either an MP in the set plus some identifiable links or the concatenation of several MPs already in the set (i.e., sufficient).*

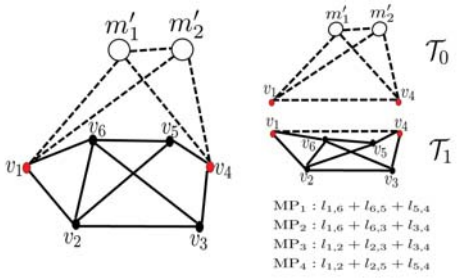


Fig. 4: An example: there are 12 possible MPs between the 2 monitors (in red color), whereas by Theorem 2 and Algorithm 1, only $2 \times 2 = 4$ MPs shown above are needed.

V. Deploying New Monitors to Reduce the Total Error Bound

So far we have the tightest $\mathcal{TEB}$ under the existing monitor deployment. A follow-up question is: if we are allowed to put an extra monitor, where should we put it so that $\mathcal{TEB}$ is maximally reduced by the new monitor? As discussed in Section I, this question is theoretically unsolvable. Hence, we use the mean value of the lower and upper bounds on each link to estimate the delay of a path between an existing monitor and a candidate node, should the new monitor be deployed at the candidate node. *The solution and theorems presented in this section are based on this assumption.*

A. Total Error Bound Reduction Analysis

After partitioning G_{new} into a TC tree, *the place of the new monitor, m_{new}, could be determined at the TC level.* This is based on two observations.

First, for a particular TC $\mathcal{T}$, placing m_{new} at different *interior nodes* of $\mathcal{T}$ (i.e., nodes in $\mathcal{T}$ except vantage node) will result in the same link identifiability [2], [4], because i)

if $\mathcal{T}$ is a 3-vertex-connected TC, then no matter where we put m_{new} inside $\mathcal{T}$, it will lead to the same new TC tree structure, based on which the link identifiability is determined, and ii) if $\mathcal{T}$ is a circle TC, although putting m_{new} at different interior nodes of $\mathcal{T}$ will lead to different new TC tree structures, for any 2 different new TC trees $\mathcal{GT}_1$ and $\mathcal{GT}_2$, the identifiable links in $\mathcal{GT}_1$ and $\mathcal{GT}_2$ are the same.

Second, same link identifiability (with the existing and new monitors) leads to same $\mathcal{TEB}$, because i) the same set of *additional* identifiable links (due to the new monitor) creates the same reduction on $\mathcal{TEB}$, ii) the total reduction on the error bounds for the remaining unidentifiable links is only affected by the additional identifiable links.

Therefore, we only need to determine the right TC to place m_{new}. We address this problem by analyzing how the new monitor would transform the original TC tree as follows.

Adding one more monitor to $\mathcal{G}_{new}$ will transform the original TC tree into a new TC tree that is restructured by the virtual links between m_{new} and $\{m'_1, m'_2\}$. Use Fig. 5a as an example, there are 6 non-root TCs in the original TC tree, which all are the candidate TC for deploying m_{new}.

Remark 8. *$\mathcal{T}_0$ is ruled out from the candidate set because placing m_{new} into $\mathcal{T}_0$ does not change the original TC tree structure and thus will not make any diffrence on $\mathcal{TEB}$.*

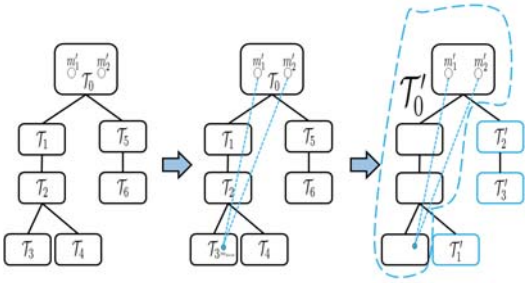
We first consider the situation where the non-root TCs are all 3-vertex-connected TCs. For instance, if m_{new} is placed into $\mathcal{T}_3$, as shown in Fig 5a, the edges (*2-vertex-cut*s) between $\mathcal{T}_0$ and $\mathcal{T}_1$ ($e_{0,1}$), between $\mathcal{T}_1$ and $\mathcal{T}_2$ ($e_{1,2}$), and between $\mathcal{T}_2$ and $\mathcal{T}_3$ ($e_{2,3}$) will all disappear in the new TC tree because with the virtual links l_{m_{new},m'_1}, l_{m_{new},m'_2}, the *2-vertex-cut*s $e_{0,1}$, $e_{1,2}$, $e_{2,3}$ are no longer *2-vertex-cut*s in the new TC tree. Hence, $\mathcal{T}_0$, $\mathcal{T}_1$, $\mathcal{T}_2$ and $\mathcal{T}_3$ merge into $\mathcal{T}'_0$ in the new TC tree. But if m_{new} is placed into $\mathcal{T}_2$, as shown in Fig 5b, only $\mathcal{T}_0$, $\mathcal{T}_1$, and $\mathcal{T}_2$ are merged into $\mathcal{T}'_0$ in the new TC tree.

The other possible situation is that some non-root TCs are circle TCs. These circle TCs should appear along some tandem sub-tree. After m_{new} is placed into the leaf node of this tandem tree, such tandem sub-tree will not merge into $\mathcal{T}'_0$ in the new TC tree, because there may exist *2-b-c* links in a circle TC, which prevent this tandem tree to be 3-vertex-connected component in the new TC tree. For example, in Fig 6, there is a circle TC $\mathcal{T}_1$ in the leftmost original TC tree. After placing m_{new} in the leaf TC $\mathcal{T}_2$, say at node v_6, this tandem TC tree will not merge into $\mathcal{T}'_0$.

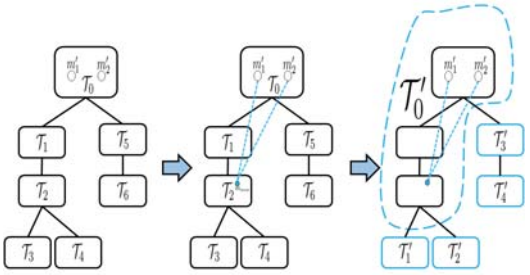
So far, we have disclosed the TC tree transform should a new monitor be added into a non-leaf TC. We next present the following theorem, with which an algorithm can be designed for optimal deployment of the new monitor.

Theorem 3. *For a tandem TC tree with non-root nodes $\{\mathcal{T}_1, \mathcal{T}_2, \cdots, \mathcal{T}_n\}$ ($n \geq 1$) in tandem (i.e., $\mathcal{T}_n$ is the leaf TC on the TC tree), placing m_{new} into the interior node of $\mathcal{T}_n$ will maximally reduce the $\mathcal{TEB}$.*

While we omit the proof due to space limit, Theorem 3 is easy to understand, because putting the new monitor at the leaf TC will merge more TCs into $\mathcal{T}'_0$, and links in $\mathcal{T}'_0$ are identifiable [4].



(a) Place m_{new} into $\mathcal{T}_3$



(b) Place m_{new} into $\mathcal{T}_2$

Fig. 5: Examples for monitor placement.

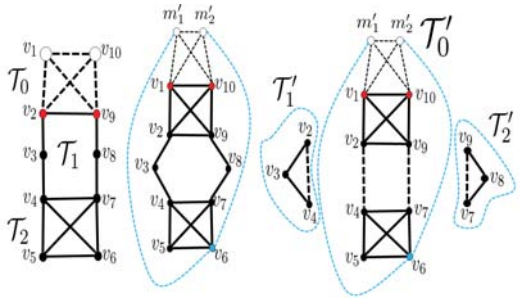


Fig. 6: Monitor placement when non-root $\mathcal{T}_1$ is circle TC.

We now analyze a general TC tree. A random TC tree could be decomposed into several tandem sub-trees, each from the root $\mathcal{T}_0$ all the way down to a leaf node. For instance, in Fig. 5a, this general TC tree has 3 tandem sub-trees $\mathcal{ST}_1 = \{\mathcal{T}_0, \mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3\}$, $\mathcal{ST}_2 = \{\mathcal{T}_0, \mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_4\}$, $\mathcal{ST}_3 = \{\mathcal{T}_0, \mathcal{T}_5, \mathcal{T}_6\}$. Because of Theorem 3, leaf nodes $\mathcal{T}_3$, $\mathcal{T}_4$ and $\mathcal{T}_6$ are the candidate TCs for placing m_{new}. Topologically speaking, $\mathcal{ST}_3$ is *independent* since it shares no TC with other tandem sub-tree, $\mathcal{ST}_1$ (or $\mathcal{ST}_2$) is *dependent* (with other tandem sub-trees), because it shares TCs with $\mathcal{ST}_2$ ($\mathcal{ST}_1$).

The two types of tandem sub-trees require different ways of calculating the reduced total error bound: placing m_{new} into $\mathcal{T}_6$ could only affect the unidentifiable links in $\mathcal{ST}_3$, i.e., $\mathcal{T}_6$'s corresponding tandem sub-tree. Placing m_{new} into $\mathcal{T}_3$ (or $\mathcal{T}_4$), however, will not only affect its corresponding tandem sub-tree, but also other tandem sub-trees that share TC with it. For example, if m_{new} is put in $\mathcal{T}_3$, the $\mathcal{TEB}$ of unidentifiable links in $\mathcal{T}_4$ could also be reduced if there are links turn identifiable

in the communal TCs of $\mathcal{ST}_1$ and $\mathcal{ST}_2$, i.e., $\mathcal{T}_1$ and $\mathcal{T}_2$.

Therefore, $\{\mathcal{ST}_1, \mathcal{ST}_2\}$ and $\{\mathcal{ST}_3\}$ should be classified into different types of tandem sub-trees, which need different ways of calculating the reduced total error bound:

1) *Dependent tandem sub-tree*: it shares a TC or multiple TCs with other tandem sub-tree(s).
2) *Independent tandem sub-tree*: it does not share TC with any other tandem sub-tree.

In Algorithm 2, $\mathcal{ST}$ in different categories will be treated differently. The algorithm calls $Get\Delta\mathcal{TEB}ForDependentTree$ and $Get\Delta\mathcal{TEB}ForIndependentTree$ to calculate the $\mathcal{TEB}$ reduction for dependent tandem sub-tree and that for independent tandem sub-tree, respectively. The details of the two functions are omitted to save space, but the main idea is to identify all TCs that will be impacted by the new monitor.

B. Maximally Reducing the Total Error Bound (MREB)

Based on the above analysis, for a general TC tree with several leaf TC nodes as $\{\mathcal{T}_{l_1}, \mathcal{T}_{l_2}, \cdots, \mathcal{T}_{l_n}\}$, each leaf node corresponding to a tandem sub-tree, placing m_{new} in $\mathcal{T}_{l_i}$ makes a series of unidentifiable links in its tandem sub-tree become identifiable, based on which the reduction on the minimum $\mathcal{TEB}$ can be calculated. The best place for deploying the new monitor is in a leaf TC that maximally reduces the mininum $\mathcal{TEB}$.

Ignoring the complexity of partitioning the network into TCs, the worst-case time complexity of Algorithm 2 is $O(\beta N_{\mathcal{T}})$, where $N_{\mathcal{T}}$ is the total number of TCs and β is the complexity of calculating $\mathcal{TEB}$ as discussed in Remark 4.

Algorithm 2 MREB: Maximally Reducing $\mathcal{TEB}$

Input: network graph $\mathcal{G}$ and initial monitors deployment
Output: the location of m_{new} that maximally reduces $\mathcal{TEB}$

1: obtain $\mathcal{G}_{new}$ and partition $\mathcal{G}_{new}$ into TCs recorded in a TC tree [8]
2: put all the leaf node TCs in a set $\mathcal{L} = \{\mathcal{T}_l^1, \mathcal{T}_l^2, \cdots, \mathcal{T}_l^n\}$; also put their corresponding tandem trees in set $\mathcal{ST} = \{\mathcal{ST}_l^1, \mathcal{ST}_l^2, \cdots, \mathcal{ST}_l^n\}$
3: **for** each $\mathcal{ST}_l^i \in \mathcal{ST}$ **do**
4: **if** $\mathcal{ST}_l^i$ is *Independent* **then**
5: $\Delta\mathcal{TEB}^i = Get\Delta\mathcal{TEB}ForIndependentTree$
6: **else**
7: $\Delta\mathcal{TEB}^i = Get\Delta\mathcal{TEB}ForDependentTree$
8: $\Delta\mathcal{TEB}^j = \max(\Delta\mathcal{TEB}^1, \Delta\mathcal{TEB}^2, \cdots, \Delta\mathcal{TEB}^n)$
9: **return** a random interior node of $\mathcal{T}_l^j$

VI. Performance Evaluation

We test the performance of MREB with real-world ISP networks, using several representative autonomous system (AS) Internet topologies collected by the Rocketfuel [9] project. Among the four AS networks, two are from Europe, one from Australia, and one from US. Their size is different to represent different scales of AS networks. Their parameters are shown in Table I, where $|L|, |V|$, and $N_{\mathcal{T}}$ denote the number of links, the number of nodes, and the number of TCs, respectively.

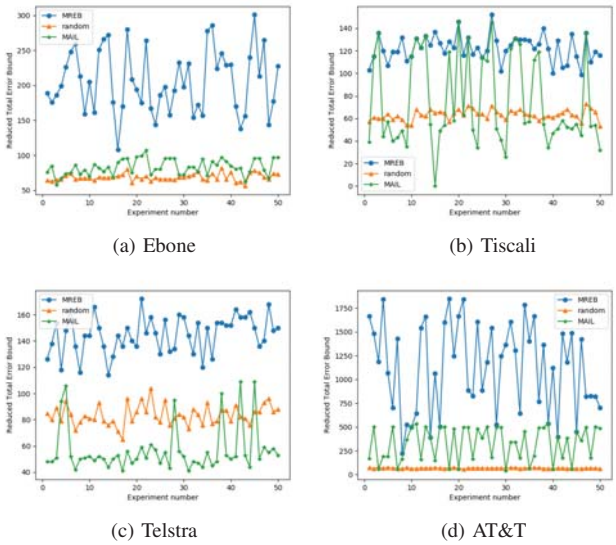
We compare MREB with the following two monitor deployment strategies:

- **Random**: Deploy the new monitor at a randomly-selected node, excluding the nodes that already have a monitor.
- **MAIL**: Deploy the new monitor at a node that maximizes the number of *additional* identifiable links.

TABLE I: Parameters of AS Network Topology

ISP Name	$\|L\|$	$\|V\|$	$N_{\mathcal{T}}$
Ebone (Europe)	381	172	37
Tiscali (Europe)	404	240	52
Telstra (Australia)	758	318	50
AT&T (US)	2078	631	154

For each AS network, we perform 50 rounds of test. In each round, we set the link delay as a random number in the range from 1 second to 30 seconds. We partition the network into bi-connected components (BCs), and in each BC, we randomly deploy a monitor. This way of deploying initial monitors is to spread the monitors across the whole network and avoid "bad" deployment that puts all initial monitors in one BC. In addition, spreading initial monitors in each BC instead of each TC will lead to a good number of unindentifiable links, because otherwise there is no strong need of bound-based performance tomography if most links, if not all, are identifiable already. We then calculate $\mathcal{TEB}$ with the deployed monitors. After that, we run different algorithms, i.e., MREB, Random, and MAIL, to deployed a new monitor, and calculate the $\mathcal{TEB}$ again using all deployed monitors. By comparing the value change of $\mathcal{TEB}$ before and after the new monitor deployment, we obtain $\mathcal{TEB}$ reduction due to the new monitor.



(a) Ebone (b) Tiscali

(c) Telstra (d) AT&T

Fig. 7: Performance of MREB, random, and MAIL.

From Fig. 7, we can see that MREB consistently leads to the highest $\mathcal{TEB}$ reduction in different ISP networks. Interestingly, the performance of MAIL is sensitive to network topology. Comparing Fig. 7 (a) and (c), we can see that MAIL may work better or worse than Random, depending on the underlying network topology. Random deployment has a relatively stable

performance, but may suffer terribly if the network is big (e.g., in the AT&T network).

From Table II, in the AS networks MREB can improve Random and MAIL w.r.t. average $\mathcal{TEB}$ reduction by up to 1587% and 245%, respectively. Even if checked with the smallest improvement, MREB can still improve Random by 72%, and MAIL by 63%.

TABLE II: Average $\mathcal{TEB}$ Reduction over 50 Tests

Topology \ Method	MREB Imp. over (Random, MAIL)	Random	MAIL
Ebone (Europe)	206 (198%, 148%)	69	83
Tiscali (Europe)	122 (93%, 63%)	63	75
Telstra (Australia)	145 (72%, 154%)	84	57
AT&T (US)	1130 (1587%, 245%)	67	328
Simulated small graph	343 (50%, 0%)	228	343
	278 (58%, 0%)	176	278

Remark 9. *Results in the first 5 rows in Table II are obtained by omitting the last step in the calculation of $\mathcal{TEB}$ (Remark 4). If using the last step in the calculation of $\mathcal{TEB}$, we cannot obtain results for the four ISP networks due to high time complexity and can only get the results (the last row) for simulated small graphs (i.e., a graph with 15 nodes and an average node degree of about 3.5). In small networks, nearly all links are identifiable and thus the benefit of MREB is small.*

VII. Related Work

The concept of network tomography was first introduced in [1]. Since then, network tomography has been extensively studied, with the goal of inferring internal network performance based on end-to-end measurements [2]–[5], [10]–[17].

Most existing work targeted at the identifiability problem. The objects in study are either all links [2], or preferential links [3], [16], or preferential paths [4]. Associated with the identifiability problem are three highly-correlated problems: (1) whether or not the objects are identifiable with existing monitors? (2) what is the minimum number of monitors that are required to identify the objects? and (3) how to deploy the monitors to identify the objects? Some papers [2], [15] studied other relevant problems: when the number of monitors is given or the monitor placement is known, what is the maximum number of links that can be identified.

Regarding measurement paths between monitors, existing work can be classified into two categories. One is that probing packets are routed according to the routing table in the network [12], [13], so called uncontrollable measurement. The other is that probing packets follow specified path using source routing [2], [3], [14], [16], [17], so called controllable measurement. This paper belongs to the second category.

VIII. Conclusion

We extended Boolean-based network tomography to bound-based network tomography, in which not only existing results on link identifiability in Boolean-based network tomography hold as before, but also performance bounds on unidentifiable links are derived. While conceptually simple, bound-based network tomography poses much harder technical challenges and renders existing Boolean-based solutions suboptimal or even infeasible. We tackled several core technical challenges in bound-based network tomography, including how to derive the tightest performance bounds, how to build minimal set of measurement paths for deriving the tightest performance bounds, and how to deploy extra monitors to tighten performance bounds over existing ones.

Acknowledgement

The work was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) under Discovery Grant RGPIN-2018-03896, Hong Kong Research Grant Council under GRF 11204917, NSFC-Guangdong Joint Fund under project U1501254, Science Technology and Innovation Committee of Shenzhen Municipality Under project JCYJ20170818095109386.

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] 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.

[3] 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.

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

[5] 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.

[6] W. Cheney and D. R. Kincaid, *Linear Algebra: Theory and Applications*, 1st ed. Jones and Bartlett Publishers, Inc., 2008.

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

[8] G. Di Battista and R. Tamassia, "On-line planarity testing," *SIAM Journal on Computing*, vol. 25, no. 5, pp. 956–997, 1996.

[9] 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/

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.