

Link Identifiability in Communication Networks with Two Monitors

Liang Ma[†], Ting He[‡], Kin K. Leung[†], Ananthram Swami[§], and Don Towsley^{*}

[†]Imperial College, London, UK. Email: {l.ma10, kin.leung}@imperial.ac.uk

[‡]IBM T. J. Watson Research Center, Yorktown, NY, USA. Email: the@us.ibm.com

[§]Army Research Laboratory, Adelphi, MD, USA. Email: ananthram.swami.civ@mail.mil

^{*}University of Massachusetts, Amherst, MA, USA. Email: towsley@cs.umass.edu

Abstract—We investigate the problem of identifying individual link performance metrics in a communication network by measuring end-to-end metrics of selected paths between monitors, under the assumption that link metrics are additive and constant during the measurement, and measurement paths cannot contain cycles. In a previous work, we developed an algorithm that places the minimum number of monitors to identify all link metrics. However, even the minimum number can be large in some practical networks (e.g., 60% of all the nodes), suggesting high monitor deployment cost. In this paper, we study the dual problem where given a fixed number of monitors, we want to place them to maximize the number of identifiable link metrics, with concrete results for the case of two monitors. The significance of the two-monitor case is that all the tomographic computation can be performed at the destination monitor without shipping measurements to a central node, thus enabling endhost-based network monitoring. We develop an efficient algorithm to determine all identifiable links in an arbitrary network with a given placement of two monitors, based on which we propose an optimal two-monitor placement algorithm to maximize the number of identifiable links. Our evaluation on real ISP topologies shows that although a large number of monitors is needed to identify all link metrics, we can usually identify a substantial portion (up to 97%) of the links using a single pair of optimally placed monitors.

I. INTRODUCTION

Accurate and timely knowledge of the internal state of a network (e.g., delays on individual links) is essential for various network operations such as route selection, resource allocation, and fault diagnosis. Directly measuring the performance of individual network elements (e.g., nodes/links) is, however, not always feasible due to the traffic overhead of the measurement process and the lack of support at internal network elements for making such measurements [1]. These limitations motivate the need for *external* approaches, where we infer the states of internal network elements by measuring the performance between a subset of nodes with monitoring capabilities, hereafter referred to as *monitors*.

Compared with hop-by-hop measurements, end-to-end measurement between monitors has the advantage requiring the minimal cooperation from internal nodes. However, such measurements impose a challenge on extracting network state as each end-to-end measurement usually contains information about multiple network elements along the measurement path. To this end, *network tomography* [2] provides a methodology for inferring network internal characteristics through

externally-available end-to-end measurements. Besides eliminating the need for special cooperation from internal nodes, network tomography can also utilize passive measurements from data packets to obtain path-level information [3], thus reducing traffic overhead.

In this paper, we consider the particular network tomography problem of inferring *additive* link metrics from end-to-end path measurements, where each measurement equals the sum of link metrics along the corresponding path. For instance, delays are additive, and multiplicative metrics (e.g., packet delivery ratio) can be expressed in an additive form using the $\log(\cdot)$ function. For additive metrics, we can formulate the problem as a system of linear equations, where the unknown variables are the link metrics, and the known constants are the end-to-end path measurements. The goal of network tomography is essentially to solve this system of linear equations.

Past experience shows that it is frequently impossible to uniquely identify all link metrics from path measurements [4], [5], [6]. In the language of linear algebra, this is because the linear system associated with the measurement paths is noninvertible, i.e., the number of *linearly independent* paths is smaller than the number of links. To ensure invertibility, we developed an algorithm in [7] to select the minimum number of nodes as monitors such that all link metrics are identifiable, where we restrict measurements to cycle-free paths to conform with the requirement of routing protocols. However, to fully identify a network, even the minimum required number of monitors can be large, e.g., up to 64% of the nodes in real ISP networks [7]. In cases where the network operator does not have sufficient budget to deploy the minimum set of monitors required for complete identification, it is still possible to identify a subset of the link metrics through a partial deployment. Therefore, it is essential to determine the fraction of link metrics identifiable for a given monitor placement and the optimal placement of a given number of monitors so as to maximize the fraction.

In this paper, we take a first step toward addressing the above problem by solving two closely related subproblems: (a) Given a network with two monitors, which links are identifiable from end-to-end measurements along cycle-free paths between the monitors? (b) Given an arbitrary network, what is the optimal placement of two monitors such that the number of identifiable links is maximized? We focus on the case of two monitors because this is the minimum possible number of monitors for cycle-free measurement, and more importantly, it represents the paradigm of endhost-based tomography. In a network with symmetric link metrics (equal in both directions), all measurements are collected by the same node (destination

Research was partially sponsored by the U.S. Army Research Laboratory and the U.K. Ministry of Defence and was accomplished under Agreement Number W911NF-06-3-0001.

monitor) and hence ready for processing by this node to yield a (partial) network state without further assistance from other nodes. We answer these two questions by developing efficient polynomial-time algorithms with guaranteed correctness. Throughout this paper, we assume that the link metrics are additive and constant, where the “constant” assumption means that either the link metrics change slowly relative to the measurement process, or they are statistical characteristics (e.g., mean, variance) that stay constant over time.

A. Related Work

When link metrics are constant, [5] shows that it is challenging to solve the inverse problem due to the presence of linearly dependent paths. If all link metrics are *binary* (normal/failed), [8] proves that the network must be $(k + 2)$ -edge-connected to identify up to k failed links using one monitor measuring cycle. For arbitrarily valued link metrics, few positive results are known. If the network is directed (links have different metrics in different directions), [9] proves that not all link metrics are identifiable unless every non-isolated node is a monitor. If the network is undirected (links have equal metrics in both directions), [10] establishes a linear time algorithm to determine the maximum number of linearly independent cycles and the corresponding identifiable links in an arbitrary network employing only one monitor. Since routing along cycles is typically prohibited in real networks, it remains open as to how monitor placement affects the number of identifiable links if only cycle-free paths can be measured. In previous work [7], we established the first topological conditions for characterizing all the links that can be identified using two monitors employing cycle-free measurements. In this work, we extend the result by quantifying the fraction of identifiable links for arbitrary network topologies.

B. Summary of Contributions

Our contributions are two-fold:

1) We develop an algorithm to determine all identifiable links for a given placement of two monitors by decomposing the network into triconnected components and applying previously established identifiability conditions to each component. The total complexity is only $O(|V| + |L|)$ ($|V|$: the number of nodes, $|L|$: the number of links);

2) We propose an optimal two-monitor placement algorithm based on the above algorithm to maximize the number of identifiable links. Our evaluation on real ISP topologies shows that with proper placement, we can identify up to 97% of the links using only two monitors, suggesting the possibility of tomography-based monitoring system with an extremely small footprint (a pair of endhosts).

The rest of the paper is organized as follows. Section II formulates the problem. Section III summarizes the theoretical conditions for complete identifiability, and Section IV presents algorithms for quantifying partial identifiability, both using two monitors. Evaluations are given in Section V. Finally, Section VI concludes the paper.

II. PROBLEM FORMULATION

A. Models and Assumptions

We assume that the network topology is known and model it as an undirected graph¹ $\mathcal{G} = (V, L)$, where V and L are the

TABLE I. NOTATIONS IN GRAPH THEORY

Symbol	Meaning
$V(\mathcal{G}), L(\mathcal{G})$	set of nodes/links in graph \mathcal{G}
$ \mathcal{G} $	degree of \mathcal{G} : $ \mathcal{G} = V(\mathcal{G}) $ (number of nodes)
\mathcal{H}	interior graph (see Definition 1)
\mathcal{P}	simple path
m_i	$m_i \in V(\mathcal{G})$ is the i -th ($i = \{1, 2\}$) monitor in \mathcal{G}

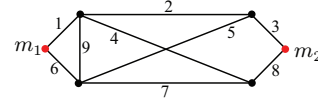


Fig. 1. Sample network with two monitors: m_1 and m_2 .

sets of nodes and links, respectively. Without loss of generality, we assume \mathcal{G} is connected, as different connected components have to be monitored separately. Denote the link incident to nodes i and j by ij ; links ij and ji are assumed to have the same metric. A subset of nodes in V are monitors, which can initiate/collect measurements. In this paper, we focus on the case where exactly two nodes are monitors, denoted by m_1 and m_2 , which is the minimum number of monitors feasible under the measurement constraint specified below. We assume that each link in \mathcal{G} has two distinct end-points (i.e., no self-loop), and there is at most one link connecting a pair of nodes. Table I summarizes the graph-theoretical notations used in this paper.

Let $n := |L|$ be the number of links in \mathcal{G} , $\{l_i\}_{i=1}^n$ the link set in \mathcal{G} , $\mathbf{w} = (w_{l_1}, \dots, w_{l_n})^T$ the column vector of all link metrics (which are unknown), and $\mathbf{c} = (c_{\mathcal{P}_1}, \dots, c_{\mathcal{P}_\gamma})^T$ the column vector of path measurements (which are observed), where γ is the number of measurement paths and $c_{\mathcal{P}_i}$ is the sum of link metrics along path \mathcal{P}_i . We assume that monitors can control the routing of measurement packets (i.e., source routing) as long as the path starts and ends at distinct monitors and does not contain repeated nodes. In the language of graph theory, we limit measurements to *simple paths* (in contrast, a *non-simple path* may contain repeated nodes). Given all the path measurements, we have a linear system:

$$\mathbf{R}\mathbf{w} = \mathbf{c}, \quad (1)$$

where $\mathbf{R} = (R_{ij})$ is a $\gamma \times n$ *measurement matrix*, with each entry $R_{ij} \in \{0, 1\}$ indicating whether link j is present on path i .

We say a link is *identifiable* if the associated link metric can be uniquely determined from path measurements; we say network \mathcal{G} is *fully identifiable* if all link metrics in \mathcal{G} are identifiable. By linear algebra, \mathcal{G} is fully identifiable if and only if \mathbf{R} in (1) has full column rank, i.e., $\text{rank}(\mathbf{R}) = n$. In other words, to uniquely determine \mathbf{w} , there must be n linearly independent simple paths between monitors. However, as illustrated in Section II-C, even if $\text{rank}(\mathbf{R}) < n$, it is still possible to identify some of the link metrics in \mathbf{w} , i.e., \mathcal{G} can be *partially identifiable*. The number of identifiable links therefore provides a *quantitative* measure of network identifiability.

B. Objective

The first objective of this paper is to determine, for an arbitrary placement of two monitors in \mathcal{G} , the set of links that are identifiable. The second objective is to compute the optimal monitor placement such that the number of identifiable links is maximized. This maximum number of identifiable links thus quantifies the identifiability of \mathcal{G} .

C. Illustrative Example

Fig. 1 displays a sample network with two monitors (m_1 , m_2) and nine links (link 1–9). To determine the metrics of

¹In this paper, the terms *network* and *graph* are used interchangeably.

these links, eight $m_1 \rightarrow m_2$ simple paths are constructed to form the measurement matrix \mathbf{R} :

$$\begin{aligned} \mathcal{P}_1 &: 1\ 2\ 3 \\ \mathcal{P}_2 &: 6\ 7\ 8 \\ \mathcal{P}_3 &: 6\ 5\ 3 \\ \mathcal{P}_4 &: 1\ 2\ 5\ 7\ 8 \\ \mathcal{P}_5 &: 1\ 4\ 8 \\ \mathcal{P}_6 &: 6\ 7\ 4\ 2\ 3 \\ \mathcal{P}_7 &: 1\ 9\ 7\ 8 \\ \mathcal{P}_8 &: 6\ 9\ 2\ 3 \end{aligned} \Rightarrow \mathbf{R} = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}, \quad (2)$$

where $R_{ij} = 1$ if and only if link j is on path i . Measuring the performance along these paths provides a set of linear equations $\mathbf{R}\mathbf{w} = \mathbf{c}$, where \mathbf{c} is the vector of path metrics measured at m_2 . In Fig. 1, it can be verified that measuring any other path² will not provide further information about the link metrics since they are linearly dependent with the paths in (2). In this example, \mathbf{R} is *non-invertible*, and thus the metrics in \mathbf{w} cannot be fully identified. Nevertheless, measurements of the paths in (2) provide sufficient information to identify the metrics of five links: link 2, 4, 5, 7 and 9. Specially, $w_2 = (c_4 - c_3 + c_8 - c_7)/2$, $w_4 = (c_5 - c_1 + c_6 - c_2)/2$, $w_5 = (c_3 - c_1 + c_4 - c_2)/2$, $w_7 = (c_6 - c_5 + c_7 - c_8)/2$ and $w_9 = (c_7 - c_1 + c_8 - c_2)/2$. Moreover, it can be verified that if we change the locations of these two monitors, then the number of identifiable links will be less than 5; therefore, the monitor placement in Fig. 1 maximizes the number of identifiable links.

III. CONDITIONS FOR COMPLETE IDENTIFIABILITY USING TWO MONITORS

In previous work [7], we established the fundamental topological conditions under which the network is (fully) identifiable using two monitors. Our results were based on a partition of the network as defined below.

Definition 1.

1) The *interior graph* \mathcal{H} of \mathcal{G} is the subgraph obtained by removing the monitors (m_1 and m_2) and their incident links. Links in \mathcal{H} are called *interior links* (e.g., links 2, 4, 5, 7 and 9 in Fig. 1); the rest of the links are called *exterior links* (e.g., links 1, 3, 6 and 8 in Fig. 1).

Based on this partition, we established the following result when only two monitors are used.

Theorem III.1. [7] None of the exterior links is identifiable.

Theorem III.2. [7] When the interior graph \mathcal{H} of \mathcal{G} is connected, the necessary and sufficient conditions for identifying all link metrics in \mathcal{H} are

- ① The graph remaining after deleting any interior link in \mathcal{G} is 2-edge-connected;
- ② The augmented graph after adding link $m_1 m_2$ (if it does not exist) to \mathcal{G} is 3-vertex-connected.

For graphs that do not satisfy conditions ①–②, Theorem III.2 only states that not all interior links can be identified; it remains to be determined which subset of interior links can be identified. Nevertheless, we will show that Theorems III.1 and III.2 provide a foundation on which to address this problem.

²Note that a path such as 674953 cannot be used for path measurement because it contains a cycle.

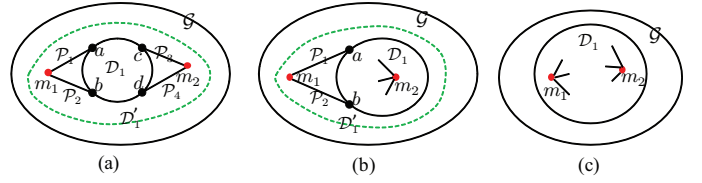


Fig. 2. Three cases in identifying links of a 3-vertex-connected subgraph \mathcal{D}_1 .

IV. CHARACTERIZATION OF PARTIAL IDENTIFIABILITY USING TWO MONITORS

In general, we may only be able to identify a subset of the (interior) links using two monitors. To characterize such partial identifiability, we develop an algorithm to determine the subset of identifiable links for a given two-monitor placement. We then propose an optimal two-monitor placement algorithm to maximize the number of identifiable links.

A. Set of Identifiable Links for a Given Two-Monitor Placement

1) *Algorithm Overview:* Our algorithm for determining the set of identifiable links originates from a simple observation rooted in Theorems III.1 and III.2: if the interior graph \mathcal{H} is 3-vertex-connected and connects to each monitor via at least two exterior links, then \mathcal{G} satisfies the conditions in Theorem III.2, and hence the identifiability of all links can be easily determined (all interior links are identifiable, all exterior links are unidentifiable). Given an arbitrary network \mathcal{G} , if we decompose it into subgraphs such that each subgraph is 3-vertex-connected, then the above observation can be applied to determine link identifiability in each subgraph. Specifically, as illustrated in Fig. 2, let \mathcal{D}_1 denote a subgraph of \mathcal{G} that is 3-vertex-connected. Our observation applies to three cases:

- (1) $\{m_1, m_2\} \notin \mathcal{D}_1$ and \exists internally vertex disjoint³ paths \mathcal{P}_1 and \mathcal{P}_2 connecting m_1 to different nodes in \mathcal{D}_1 , and paths \mathcal{P}_3 and \mathcal{P}_4 connecting m_2 to different nodes in \mathcal{D}_1 (Fig. 2-a): Abstracting each path as a single link, we can apply our observation to the graph \mathcal{D}'_1 formed by these four links together with \mathcal{D}_1 and the two monitors, which implies that all links in \mathcal{D}_1 are identifiable.
- (2) $m_1 \notin \mathcal{D}_1$ but $m_2 \in \mathcal{D}_1$ (or vice versa) and \exists internally vertex disjoint paths \mathcal{P}_1 and \mathcal{P}_2 connecting m_1 to different nodes in \mathcal{D}_1 (Fig. 2-b): Our observation applies to the graph \mathcal{D}'_1 consisting of \mathcal{D}_1 , m_1 , \mathcal{P}_1 and \mathcal{P}_2 (modeled as single links), implying that all links in \mathcal{D}_1 except for those incident to m_2 are identifiable.
- (3) $\{m_1, m_2\} \in \mathcal{D}_1$ (Fig. 2-c): Our observation applies to \mathcal{D}_1 , implying that all its links except for those incident to m_1 or m_2 are identifiable. Note that due to the 3-vertex-connectivity of \mathcal{D}_1 , a monitor in \mathcal{D}_1 must connect to other nodes in \mathcal{D}_1 via at least three links.

These three cases are not exclusive; we will show, however, that they represent the majority of the cases (see explanations of Algorithm 2). Our strategy is to first decompose the graph into subgraphs falling into one of the above cases (plus some boundary cases), and then determine the set of identifiable links in each subgraph accordingly.

2) *Algorithm Design:* To formally present the proposed algorithm, we introduce the following definition.

Definition 2. A k -connected component of \mathcal{G} is a maximal subgraph of \mathcal{G} that is either (i) k -vertex-connected, or (ii) a

³That is, the paths are disjoint except at end-points.

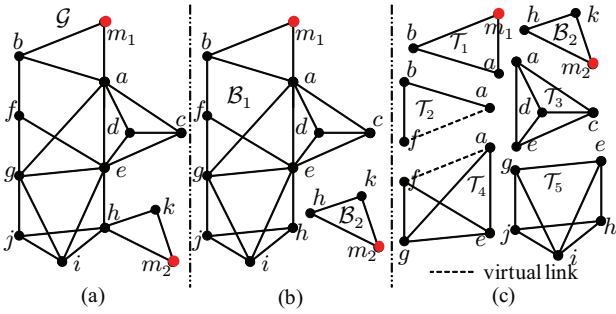


Fig. 3. Partitioning of a sample network. (a) Sample network \mathcal{G} , (b) Biconnected components of \mathcal{G} , (c) Triconnected components of \mathcal{G} .

Algorithm 1: Classification of Triconnected Components

```

input : Connected graph  $\mathcal{G}$ 
output: Triconnected components and their categories
1 partition  $\mathcal{G}$  into biconnected components  $\mathcal{B}_1, \mathcal{B}_2, \dots$  [11] and
  identify monitoring agents for each biconnected component;
2 foreach biconnected component  $\mathcal{B}_i$  with 2 monitoring agents
  do
3   partition  $\mathcal{B}_i$  into triconnected components  $\mathcal{T}_1, \mathcal{T}_2, \dots$  [12];
4   foreach triconnected component  $\mathcal{T}_j$  of  $\mathcal{B}_i$  with  $|\mathcal{T}_j| \geq 3$ 
     do
5     determine the types of all 2-vertex cuts of  $\mathcal{T}_j$ ;
6     if  $|\mathcal{T}_j| = 3$  then
7       record all neighboring triconnected components
         connected to  $\mathcal{T}_j$  through 2-vertex cuts;
8     end
9     determine the category of  $\mathcal{T}_j$ ;
10  end
11 end

```

complete graph with up to k vertices. The case of $k = 2$ is also called a *biconnected component*, and $k = 3$ a *triconnected component*.

By this definition, a biconnected component can be either a 2-vertex-connected subgraph or a single link (the case of an isolated vertex is excluded by assuming \mathcal{G} to be connected). Similarly, a triconnected component can be 3-vertex-connected, a triangle, or a single link. Intuitively, a biconnected component is a subgraph connected to the rest of the graph by cut-vertices⁴, and a triconnected component (within a biconnected component) is a subgraph connected to the rest by 2-vertex cuts⁵. For instance, Fig. 3-b shows the biconnected components of Fig. 3-a, separated by cut-vertex h . Fig. 3-c shows the triconnected components, separated by the above cut-vertex and the 2-vertex cuts $\{a, b\}$, $\{a, f\}$, $\{a, e\}$, and $\{e, g\}$. Not all subgraphs separated by 2-vertex cuts form triconnected components according to Definition 2 (e.g., \mathcal{T}_2 in Fig. 3-c). To fix this issue, we process the graph to add *virtual links* as follows: for each 2-vertex cut whose vertices are not neighbors (e.g., $\{a, f\}$), connect these vertices by a virtual link; repeat this on the resulting graph until no such cut exists. It can be verified that all subgraphs separated by cut-vertices or 2-vertex cuts in the processed graph are triconnected components.

There exist fast algorithms to partition an arbitrary graph \mathcal{G} into biconnected components [11] and then into triconnected components [12]. Note that the output of [12] is the set of triconnected components of the processed \mathcal{G} . Based on these algorithms, we propose an algorithm, *Determining Identifiable*

⁴A *cut-vertex* is a vertex whose removal will disconnect the graph.

⁵A *2-vertex cut* is a set of two vertices $\{v_1, v_2\}$ such that removing v_1 or v_2 alone does not disconnect \mathcal{G} , but removing both disconnects \mathcal{G} .

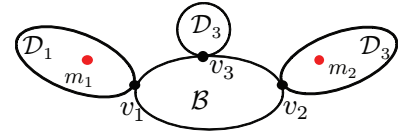


Fig. 4. Monitoring agents: v_1 and v_2 are monitoring agents wrt \mathcal{B} , but v_3 is not.

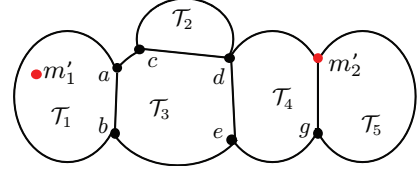


Fig. 5. Triconnected components $\mathcal{T}_1, \dots, \mathcal{T}_5$ within the same biconnected component (m'_1, m'_2 : monitoring agents).

Links under Two Monitors (DIL-2M) (see Algorithm 2), to determine all identifiable links in \mathcal{G} for a given two-monitor placement. Before presenting this algorithm, we first introduce an auxiliary algorithm (Algorithm 1), which builds upon the graph partition algorithms [11], [12] to identify triconnected components and classify them into various categories based on their connectivity to the monitors. The classification is based on the following two concepts.

Definition 3. A *monitoring agent* wrt a subgraph \mathcal{G}' is either a monitor in \mathcal{G}' or a cut-vertex that separates \mathcal{G}' from a monitor. A *Type- k vertex cut* (denoted by *Type- k -VC*) wrt a triconnected component \mathcal{T} is a 2-vertex cut $\{v_1, v_2\}$ in \mathcal{T} that separates \mathcal{T} from k monitoring agents (including monitoring agents in $\{v_1, v_2\}$) in its parent biconnected component (i.e., the biconnected component containing \mathcal{T}).

We illustrate the concept of a monitoring agent in Fig. 4. Consider the biconnected component \mathcal{B} illustrated in Fig. 4, which is separated from subgraphs \mathcal{D}_j by cut-vertices v_j ($j = 1, 2, 3$). We say that v_1 and v_2 are monitoring agents wrt \mathcal{B} since \mathcal{D}_1 and \mathcal{D}_2 both contain monitors; v_3 is not a monitoring agent wrt \mathcal{B} as \mathcal{D}_3 does not contain any monitors (but it is a monitoring agent wrt \mathcal{D}_3). The significance of monitoring agents is that they essentially serve as “monitors” for identifying links in the corresponding biconnected component.

Lemma IV.1. Let \mathcal{B} be a biconnected component with monitoring agents m'_1 and m'_2 . The set of identifiable links in $L(\mathcal{B})$ does not depend on whether m'_1 or m'_2 are monitors or not, except for link $m'_1 m'_2$ (if it exists). Link $m'_1 m'_2$ is identifiable if and only if m'_1 and m'_2 are both monitors.

Proof: see [13]. ■

Remark: Since each monitoring agent is associated with at least one monitor, the number of monitoring agents wrt a biconnected component cannot exceed the total number of monitors (i.e., two). Clearly, for any biconnected component with fewer than two monitoring agents, none of its links is identifiable since all monitor-to-monitor paths going into this component must leave through the same node, creating a cycle. Therefore, it suffices to only consider biconnected components with two monitoring agents.

We illustrate the concept of a Type- k vertex cut in Fig. 5, which contains a biconnected component consisting of five triconnected components. Each triconnected component \mathcal{T}_i in Fig. 5 is separated from the rest by a set of 2-vertex cuts, each cut separating \mathcal{T}_i from $k \in \{0, 1, 2\}$ monitoring agents. The value of k determines the type of this cut. Specifically, in Fig. 5, we have: (i) Type-1-VC $\{a, b\}$ wrt \mathcal{T}_1 , (ii) Type-2-VC $\{c, d\}$ wrt \mathcal{T}_2 , (iii) Type-1-VC $\{a, b\}$, Type-1-VC $\{d, e\}$, and

Algorithm 2: Determining Identifiable Links under Two Monitors (DIL-2M)

```

input : Connected graph  $\mathcal{G}$ 
output: All identifiable links in  $\mathcal{G}$ 
1 partition  $\mathcal{G}$  into triconnected components and determine the
  category of each triconnected component by Algorithm 1;
2 if  $\exists$  direct link  $l_{m_1 m_2}$  connecting  $m_1$  and  $m_2$  then
3   |  $l_{m_1 m_2}$  is identifiable;
4 end
5 foreach triconnected component  $\mathcal{T}_i$  within a biconnected
  component with 2 monitoring agents do
6   | if  $\mathcal{T}_i$  is in Category 1 then
7     | all links in  $\mathcal{T}_i$  except for the ones incident to the
      | monitoring agent are identifiable;
8   | else if  $\mathcal{T}_i$  is in Category 2 then
9     | all links in  $\mathcal{T}_i$  except for the ones incident to the
      | monitoring agents or the vertices in the Type-2-VC
      | are identifiable;
10  | else if  $\mathcal{T}_i$  is in Category 3 then
11    | all links in  $\mathcal{T}_i$  are identifiable;
12  | else
13    |  $\mathcal{T}_i$  must be of Category 4 (identified by Algorithm 3);
14  | end
15 end

```

Type-0-VC $\{c, d\}$ wrt \mathcal{T}_3 , (iv) Type-1-VC $\{d, e\}$ and Type-1-VC $\{m'_2, g\}$ wrt \mathcal{T}_4 , and (v) Type-2-VC $\{m'_2, g\}$ wrt \mathcal{T}_5 .

Based on these two concepts, Algorithm 1 categorizes each triconnected component of \mathcal{G} . It first decomposes \mathcal{G} into biconnected components using the algorithm in [11] and identifies the monitoring agents for each biconnected component (line 1). Each biconnected component of interest (with two monitoring agents) is further decomposed into triconnected components using the algorithm in [12] (line 3). Since Lemma IV.1 already establishes the necessary and sufficient condition for identifying a degenerate triconnected component consisting of a single link (both end-points must be monitors), we only need to consider triconnected components with three or more nodes (lines 4-10). For each such triconnected component, we determine the types of all its 2-vertex cuts (wrt itself) by counting the number of monitoring agents on the other side of each 2-vertex cut (within the parent biconnected component). For triconnected components with exactly three nodes (i.e., triangles), additional information is recorded (line 7), the purpose of which will be explained later in Algorithm 3. Based on the monitoring agents and the types of 2-vertex cuts, Algorithm 1 classifies each triconnected component into one of the following four categories (line 9):

- (i) **Category 1**: containing only one monitoring agent and at least one Type-1-VC, e.g., \mathcal{T}_1 and \mathcal{T}_4 in Fig. 5;
- (ii) **Category 2**: containing one Type-2-VC, e.g., \mathcal{T}_2 and \mathcal{T}_5 in Fig. 5, or containing two monitoring agents;
- (iii) **Category 3**: a 3-vertex-connected component containing two Type-1-VCs and no monitoring agent, e.g., \mathcal{T}_3 in Fig. 5;
- (iv) **Category 4**: a triangle component containing two Type-1-VCs and no monitoring agent.

We have proved [13] that each triconnected component with three or more nodes falls into one of these four categories.

Now we formally present DIL-2M (Algorithm 2). Algorithm 2 determines identifiable links one triconnected component at a time based on the category of the triconnected component (lines 6-14). These steps are best understood in comparison with the illustration in Fig. 2. Viewing the monitoring agents as real monitors (by Lemma IV.1), Category 1, 2, and 3 are

Algorithm 3: Determining Identifiable Links in Triangles

```

input : Triangle component  $\mathcal{T}$  of Category 4 with two
  Type-1-VCs  $\{v_1, v_2\}$  and  $\{v_1, v_3\}$ , and a set  $S_1$  ( $S_2$ )
  of its immediately neighboring triconnected
  components connected to  $\mathcal{T}$  via  $\{v_1, v_2\}$  ( $\{v_1, v_3\}$ )
output: Identifiable links in  $\mathcal{T}$ 
1  $v_1 v_2$  ( $v_1 v_3$ ) is identifiable if it is a real link;
2 if (link  $v_1 v_2$  is real OR  $|S_1| \geq 2$  OR one component in  $S_1$  is
  3-vertex-connected) AND (link  $v_1 v_3$  is real OR  $|S_2| \geq 2$  OR
  one component in  $S_2$  is 3-vertex-connected) then
3   |  $v_2 v_3$  is identifiable if it is a real link;
4 end

```

equivalent to the cases in Fig. 2-b, Fig. 2-c, and Fig. 2-a, respectively. This is because the 2-vertex-connectivity of the parent biconnected component ensures the existence of pairwise internally vertex disjoint paths $\{\mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3, \mathcal{P}_4\}$ as shown in Fig. 2-a-b; see [13] for detailed proofs. Moreover, Theorem III.1 (together with the observation that no link is identifiable in a biconnected component with fewer than two monitoring agents) guarantees that links not classified as identifiable by DIL-2M are unidentifiable, i.e., it determines *all* the identifiable links. The complete proof of the correctness of Algorithm 2 is detailed in [13].

Remark: Note that the two Type-1-VCs in Category 3 can have one common node, which does not alter the identifiability of links in a triconnected component of this category. For Category 2, the direct link (if any) connecting the 2 monitoring agents or the 2 vertices in the Type-2-VC cannot be immediately determined in the current triconnected component; nevertheless, it can be determined in a neighboring triconnected component, e.g., in Fig. 5, the identifiability of link cd cannot be determined in \mathcal{T}_2 , but can be determined in \mathcal{T}_3 (as identifiable).

Category 4 is a special case requiring separate treatment. Consider the case that \mathcal{D}_1 in Fig. 2-a is a triangle, and a and c are the same node. We cannot directly apply Theorem III.2 to \mathcal{D}'_1 as \mathcal{D}_1 may contain virtual links (see the explanation of Definition 2). To handle this case, we introduce another auxiliary algorithm, Algorithm 3. The key question that Algorithm 3 answers is: when virtual links are used in identifying a link, are there real paths in neighboring components to replace these virtual links? Let the two Type-1-VCs in triangle \mathcal{T} be $\{v_1, v_2\}$ and $\{v_1, v_3\}$. To identify $v_2 v_3$, the path replacement of $v_1 v_2$ or $v_1 v_3$ (if they are virtual) may involve monitoring agents, which implies that Condition ② in Theorem III.2 is not satisfied (see the proof in [13]). However, any path replacement of $v_2 v_3$ in neighboring components of \mathcal{T} does not involve monitoring agents. Thus, this freedom of path selections benefits the identification of $\{v_1, v_2\}$ and $\{v_1, v_3\}$. In fact, $v_1 v_2$ and $v_1 v_3$ (if any) are always identifiable (line 1). To identify $v_2 v_3$, line 2 of Algorithm 3 specifies the conditions to ensure the existence of path replacements that do not contain any monitoring agents. Note that for Categories 1–3, even if the associated triconnected component contains virtual links, the conclusions in lines 7, 9, and 11 still hold (see the proof in [13]).

3) *Complexity Analysis:* In Algorithm 1, partitioning \mathcal{G} into biconnected components (line 1) and then triconnected components (line 3) takes $O(V(\mathcal{G}) + L(\mathcal{G}))$ time [11], [12]. The classification of triconnected components (lines 4-10) can be performed during the partitioning with a constant-factor increase in the complexity. Therefore, Algorithm 1 (also line 1 of Algorithm 2) has $O(|V(\mathcal{G})| + |L(\mathcal{G})|)$ time complexity. In Algorithm 2, lines 2-4 take $O(1)$ time. The complexity of

TABLE II. ISP NETWORKS

ISP	n	$ V $	κ	n_{\max}	n_{\max}/n	$n_{\max}\kappa/(2n)$
Abovenet (US)	294	182	117	126	0.43	25.1
EBONE (Europe)	381	172	55	286	0.75	20.7
Tiscali (Europe)	404	240	138	200	0.49	34.2
Exodus (US)	434	201	85	295	0.68	28.9
Telstra (Australia)	758	318	164	480	0.63	51.9
AT&T (US)	2078	631	208	1708	0.82	85.5
Sprintlink (US)	2268	604	163	2037	0.90	73.2
Verio (US)	2821	960	408	2210	0.78	159.7
Level3 (US)	5298	624	94	5146	0.97	45.6

lines 5-15 is $O(|L(\mathcal{T}_i)|)$ for each triconnected component \mathcal{T}_i under consideration (Algorithm 3 takes $O(1)$ time). Therefore, the overall complexity of Algorithm 2 is $O(|V(\mathcal{G})| + |L(\mathcal{G})|)$.

4) *Example:* The sample network in Fig. 3-a has 2 bi-connected components \mathcal{B}_1 and \mathcal{B}_2 (Fig. 3-b) with monitoring agents $\{m_1, h\}$ and $\{h, m_2\}$, respectively. There are 5 triconnected components in \mathcal{B}_1 (Fig. 3-c), where $\{\mathcal{T}_1, \mathcal{T}_5\}$, $\{\mathcal{T}_3\}$, $\{\mathcal{T}_4\}$ and $\{\mathcal{T}_2\}$ belong to Categories 1, 2, 3 and 4, respectively. Moreover, \mathcal{B}_2 itself is a Category 2 triconnected component. Therefore, the identifiable links are $\{ab, bf, cd, ae, ag, fe, fg, eg, ei, gi, gj, ij\}$.

B. Optimal Two-Monitor Placement

Algorithm 2 allows us to determine the number of identifiable links in an arbitrary network with a given placement of two monitors in $O(|V(\mathcal{G})| + |L(\mathcal{G})|)$ time. This enables us to maximize the number of identifiable links using a simple algorithm, referred to as *Optimal Monitor Placement (OMP)*, that works as follows. OMP enumerates all possible placements of m_1 and m_2 in \mathcal{G} and computes the number of identifiable links for each placement so as to determine the optimal placement with the maximum number of identifiable links. Since there are $\binom{|V(\mathcal{G})|}{2}$ possible two-monitor placements, the complexity of OMP is $O(|V(\mathcal{G})|^3 + |V(\mathcal{G})|^2|L(\mathcal{G})|)$.

V. EVALUATIONS ON ISP TOPOLOGIES

We evaluate OMP on the *Internet Service Provider (ISP)* topologies collected by the Rocketfuel project [14], which represent physical connections between backbone/gateway routers of several ISPs⁶ around the globe. Given a network topology, OMP is applied to compute the maximum number of identifiable links (n_{\max}) that can be achieved by two monitors as well as the corresponding monitor placement⁷. For comparison, we also calculate the minimum number of monitors (denoted by κ) required to identify all link metrics in the same network via the MMP algorithm proposed in [7]. The focus of the evaluation is on the *identification ratio*, i.e., the fraction of links that can be identified (n_{\max}/n).

Simulation results are presented in Table II, where $|V|$ is the number of nodes and we sort the networks according to the number of links (i.e., n). We notice that to identify all link metrics, a large fraction of nodes must serve as monitors, ranging from 30% (EBONE, AT&T, Sprintlink) to more than

60% (Abovenet). This is because ISP networks contain a large number of gateway routers to connect to customers or other ISPs, which appear as dangling nodes that have to be selected as monitors; see MMP in [7] for detailed explanations. In contrast, OMP can identify a significant portion of the links using only two monitors, achieving an identification ratio greater than 0.4 for all the networks and as large as 0.97 for Level3. This is because most ISP networks contain at least one densely-connected subnetwork (likely the backbone network) which tends to be 3-vertex-connected, thus allowing a properly placed monitor pair to identify all links in this subnetwork analogous to the case of Fig. 2 (where \mathcal{D}_1 is large enough to cover most links in \mathcal{G}). Moreover, OMP also exhibits superior performance in terms of *cost efficiency* measured by the number of identifiable links per monitor, i.e., n/κ for MMP and $n_{\max}/2$ for OMP. We see from their ratio $n_{\max}\kappa/(2n)$ that OMP improves the cost efficiency by more than 20 times for all ISPs, and as much as 159 times for Verio (even with underestimated n_{\max} for the last four networks). Over all, OMP has better performance for large, densely-connected networks.

VI. CONCLUSION

We studied the network tomography problem of identifying additive link metrics from end-to-end measurements using two monitors. We quantitatively characterized network identifiability by developing efficient algorithms to determine and then maximize the number of identifiable links in an arbitrary network using two monitors. Simulations driven by real networks show that although a large number of monitors are needed to identify all the links, a majority of the links can be identified using only two monitors, suggesting the efficacy of endhost-based tomographic solutions.

REFERENCES

- [1] F. Lo Presti, N. Duffield, J. Horowitz, and D. Towsley, "Multicast-based inference of network-internal delay distributions," *IEEE/ACM Transactions on Networking*, vol. 10, no. 6, pp. 761–775, Dec. 2002.
- [2] Y. Vardi, "Estimating source-destination traffic intensities from link data," *Journal of the American Statistical Assoc.*, pp. 365–377, 1996.
- [3] E. Lawrence and G. Michailidis, "Network tomography: A review and recent developments," *Frontiers in Statistics*, vol. 54, 2006.
- [4] W. Xu, E. Mallada, and A. Tang, "Compressive sensing over graphs," in *IEEE INFOCOM*, 2011.
- [5] Y. Chen, D. Bindel, and R. H. Katz, "An algebraic approach to practical and scalable overlay network monitoring," in *ACM SIGCOMM*, 2004.
- [6] A. Chen, J. Cao, and T. Bu, "Network Tomography: Identifiability and Fourier domain estimation," in *IEEE INFOCOM*, 2007.
- [7] L. Ma, T. He, K. K. Leung, A. Swami, and D. Towsley, "Identifiability of link metrics based on end-to-end path measurements," in *ACM IMC*, 2013.
- [8] S. Ahuja, S. Ramasubramanian, and M. Krunz, "SRLG failure localization in all-optical networks using monitoring cycles and paths," in *IEEE INFOCOM*, 2008.
- [9] Y. Xia and D. Tse, "Inference of link delay in communication networks," *IEEE Journal of Selected Areas in Communications*, 2006.
- [10] A. Gopalan and S. Ramasubramanian, "On the maximum number of linearly independent cycles and paths in a network," Technical Report, University of Arizona, August 2012.
- [11] R. Tarjan, "Depth-first search and linear graph algorithms," *SIAM Journal on Computing*, vol. 1, pp. 146–160, 1972.
- [12] J. E. Hopcroft and R. E. Tarjan, "Dividing a graph into triconnected components," *SIAM Journal on Computing*, vol. 2, pp. 135–158, 1973.
- [13] L. Ma, T. He, K. K. Leung, A. Swami, and D. Towsley, "Link identifiability with two monitors: Proof of selected theorems," Technical Report, Imperial College, London, UK., Feb. 2013. [Online]. Available: <http://www.commsp.ee.ic.ac.uk/%7elm110/pdf/MaTRFeb13.pdf>
- [14] "Rocketfuel: An ISP topology mapping engine," University of Washington, 2002. [Online]. Available: <http://www.cs.washington.edu/research/networking/rocketfuel/>

⁶ISP networks have undergone changes since Rocketfuel project, one notable change being the presence of parallel links between the same node pairs. Although we assume simple graphs, it is easy to see that if one link is identifiable, then all parallel links between the same node pair are identifiable as long as monitors can control which link the probe traverses.

⁷To bound running time, only 2000 randomly chosen monitor placements are tested for the four largest networks (AT&T, Sprintlink, Verio, and Level3), and we select the placement with the largest number of identifiable links.