

Monitor Placement for Maximal Identifiability in Network Tomography

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 placing a given number of monitors in a communication network to identify the maximum number of link metrics from end-to-end measurements between monitors, assuming that link metrics are additive, and measurement paths cannot contain cycles. Motivated by our previous result that complete identification of all link metrics can require a large number of monitors, we focus on partial identification using a limited number of monitors. The basis to our solution is an efficient algorithm for determining all identifiable links for a given monitor placement. Based on this algorithm, we develop a polynomial-time greedy algorithm to incrementally place monitors such that each newly placed monitor maximizes the number of additional identifiable links. We prove that the proposed algorithm is optimal for 2-vertex-connected networks, and demonstrate that it is near-optimal for several real ISP topologies that are not 2-vertex-connected. Our solution provides a quantifiable tradeoff between level of identifiability and available monitor resources.

I. INTRODUCTION

Accurate and timely knowledge of internal network states (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 a 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 pairs of nodes with monitoring capabilities, hereafter referred to as *monitors*.

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

also utilize passive measurements from data traffic to obtain path-level information [3], thus reducing traffic overhead.

In this paper, we consider the particular network tomographic 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 ratios) 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 determine (i.e., *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, restricting measurements to cycle-free paths to conform with the requirement of routing protocols. However, to identify all link metrics, even the minimum required number of monitors can be large, e.g., up to 64% of all nodes in ISP networks [7]. In cases where the network operator does not have sufficient budget to deploy all the monitors required for complete identification, an open question is: which subset of link metrics can be identified through a partial deployment of monitors, and where should the monitors be placed to identify the largest subset of link metrics?

In this paper, we address the above problems by answering two closely related questions: (a) Given a network with specified monitors, which links are identifiable from end-to-end measurements along cycle-free paths between monitors? (b) Given a network and κ monitors, where should these monitors be placed in order to identify the maximum number of links? While one may answer these questions by enumerating all possible measurement paths and monitor placements, the complexity of such approaches will explode exponentially as the network size grows. We take a different approach by drawing insights from previously established identifiability conditions that relate link identifiability to network topology and monitor placements. These insights then form the basis of our efficient polynomial-time algorithms addressing the above questions.

Throughout this paper, we assume that the link metrics are additive and constant, where the “constant metric” assumption means that either the link metrics change slowly relative to the measurement process, or they are statistical characteristics

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

(e.g., mean, variance) that stay constant over time.

A. Related Work

It is generally challenging to determine network identifiability [5]. If the network is directed, [8] proves that not all link metrics are identifiable unless every non-isolated node is a monitor. If the network is undirected, [9] derived topological conditions for network identifiability, assuming measurements of cycles. Since routing along cycles is typically prohibited in communication networks, we established the first necessary and sufficient conditions [7] for network identifiability and associated algorithms [10] for identifying link metrics, employing only cycle-free measurements. Observing that a network can be partially identifiable even if not completely identifiable, [11] developed an algorithm to determine the maximum number of linearly independent cycles and the corresponding identifiable links with a single monitor. For cycle-free measurements, we proposed efficient algorithms [12] to determine all identifiable links in a network with two monitors.

When it comes to placing a limited number of monitors to maximize network identifiability, only a few positive results are known. In [13], [14], the problem of placing the minimum number of monitors to ensure complete identifiability under uncontrollable routing is proved to be NP-hard, and the NP-hardness persists even if some nodes have control over their local routing policy [15]. In contrast, we showed in [7] that under controllable cycle-free routing, this problem is solvable in linear time. However, in cases where we cannot afford even the minimum number of monitors required for complete identification, it remains open how a limited number of monitors affects network identifiability. We address this problem by quantifying the number of identifiable links for a given placement of monitors, and then presenting an efficient monitor placement algorithm to maximize identifiability.

B. Summary of Contributions

Our contributions in this paper are three-fold:

1) We develop an algorithm (called DAIL) to determine¹ all identifiable links for a given placement of monitors by decomposing the network into triconnected components and applying the identifiability conditions we established in [7] to each component. DAIL has a complexity of $O(|V| + |L|)$ ($|V|$: the number of nodes, $|L|$: the number of links).

2) To maximize the number of identifiable links using κ monitors ($\kappa \geq 2$), we first notice that only a subset of nodes can be involved in an optimal placement, based on which we develop a preprocessing algorithm to select² a set of candidate monitors \mathcal{S} , guaranteed to contain an optimal κ -monitor placement. The size of \mathcal{S} is proportional to the number of triconnected components and usually much smaller than $|V|$. We then develop a polynomial-time algorithm that incrementally selects κ monitors from \mathcal{S} such that each selected monitor maximizes the additional number of identifiable links.

3) The proposed monitor placement algorithm is optimal for 2-vertex-connected networks. For non-2-vertex-connected networks, we show via numerical evaluations on ISP topologies that our algorithm closely approximates the optimal number of identifiable links for all considered values of κ , while incurring a much lower complexity than an exhaustive search.

The rest of the paper is organized as follows. Section II formulates the problem. Section III summarizes the theoretical

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	i -th monitor in \mathcal{G}
m'_i	i -th agent in a biconnected component of \mathcal{G} (see Definition 3)
μ_i	i -th vantage in a triconnected component of \mathcal{G} (see Definition 5)
N_B, N_T	number of biconnected/triconnected components in \mathcal{G}
κ_{MMP}	minimum number of monitors for complete network identification computed by MMP [7]
$N(\mathcal{O})$	number of identifiable links for monitor placement \mathcal{O} ($\mathcal{O} \subseteq V(\mathcal{G})$)

conditions for complete network identifiability. Section IV and Section V present algorithms for quantifying and maximizing partial identifiability using κ monitors. Section VI presents evaluation results. Finally, Section VII concludes the paper.

II. PROBLEM FORMULATION

A. Models and Assumptions

We assume that the network topology is known and model it as an undirected graph³ $\mathcal{G} = (V, L)$, where V and L are the 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. We denote the link incident to nodes u and v by uv , and assume that links uv and vu have the same metric. A subset of nodes in V are monitors, which can initiate/collect measurements. 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} , and $\mathbf{w} = (w_1, \dots, w_n)^T$ the column vector of all link metrics (which are *unknown*). Given a set of measurement paths $\{\mathcal{P}_i\}_{i=1}^\gamma$, let $\mathbf{c} = (c_1, \dots, c_\gamma)^T$ be the column vector of path measurements (which are *observed*), where c_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 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 $R_{ij} \in \{0, 1\}$ indicating whether link l_j is present on path \mathcal{P}_i .

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

B. Objective

The objective of this paper is two-fold: (i) to determine the set of links in \mathcal{G} that are identifiable for a given monitor placement, and (ii) to find an optimal placement of a given number of monitors to maximize the number of identifiable links.

C. Illustrative Example

Fig. 1 displays a sample network with two monitors $\{m_1, m_2\}$, and nine links (links l_1 – l_9). To identify these links,

¹The issue of measurement path construction for inferring metrics of all identifiable links determined by DAIL remains open and is left to future work.

²The phrase “select a (candidate) monitor” is used as abbreviation for “select a (candidate) node for placing a monitor” throughout this paper.

³In this paper, the terms *network* and *graph* are used interchangeably.

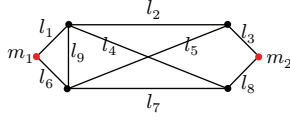


Fig. 1. Sample network with two monitors m_1 and m_2 .

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

$$\begin{aligned} \mathcal{P}_1 &: l_1 l_2 l_3 \\ \mathcal{P}_2 &: l_6 l_7 l_8 \\ \mathcal{P}_3 &: l_6 l_5 l_3 \\ \mathcal{P}_4 &: l_1 l_2 l_5 l_7 l_8 \\ \mathcal{P}_5 &: l_1 l_4 l_8 \\ \mathcal{P}_6 &: l_6 l_7 l_4 l_2 l_3 \\ \mathcal{P}_7 &: l_1 l_9 l_7 l_8 \\ \mathcal{P}_8 &: l_6 l_9 l_2 l_3 \end{aligned} \Rightarrow \mathbf{R} = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}. \quad (2)$$

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 adding any other simple path between m_1 and m_2 does not provide further information about the link metrics since it is linearly dependent with the paths in (2). In this example, \mathbf{R} is *non-invertible*, and thus the metrics in \mathbf{w} cannot be completely identified. Nevertheless, measurements of the paths in (2) provide sufficient information to identify the metrics of five links: l_2, l_4, l_5, l_7 , and l_9 . Specifically, $w_2 = (c_4 - c_3 + c_8 - c_7)/2$, $w_4 = (c_5 - c_1 + c_6 - c_2)/2$, $w_5 = (c_3 - c_1 + c_4 - c_2)/2$, $w_7 = (c_6 - c_5 + c_7 - c_8)/2$, 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 decrease. Therefore, the monitor placement in Fig. 1 is optimal in that it maximizes the number of identifiable links among all 2-monitor placements.

III. FUNDAMENTALS OF NETWORK IDENTIFIABILITY

We start by reviewing our previous results on complete identifiability, i.e., all link metrics are identifiable.

A. Conditions for Complete Identifiability

In previous work [7], we established the following fundamental conditions on network topology and placement of monitors for complete identifiability.

1) Identifiability With Two Monitors

Employing two monitors, the results are based on a network partition 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 l_2, l_4, l_5, l_7 , and l_9 in Fig. 1); the remaining links are called *exterior links* (e.g., links l_1, l_3, l_6 , and l_8 in Fig. 1).

Based on this partition, we established Theorems III.1 and III.2 when only two monitors are used.

Theorem III.1. [7] If only two monitors m_1 and m_2 are used, then none of the exterior links, except for link $m_1 m_2$ (if it exists), is identifiable.

Theorem III.2. [7] If only two monitors m_1 and m_2 are used, the interior graph \mathcal{H} (with $|L(\mathcal{H})| > 1$) of \mathcal{G} is connected and link $m_1 m_2$ does not exist, then the necessary and sufficient conditions for identifying *all* link metrics in \mathcal{H} are:

- ① The remaining graph after deleting any interior link in \mathcal{G} is 2-edge-connected;

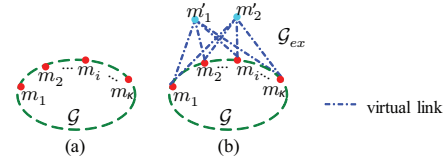


Fig. 2. (a) \mathcal{G} with κ ($\kappa \geq 3$) monitors; (b) \mathcal{G}_{ex} with two virtual monitors.

- ② The augmented graph after adding link $m_1 m_2$ to \mathcal{G} is 3-vertex-connected.

2) Identifiability With Three or More Monitors

For a network with three or more monitors, the identifiability condition is expressed in terms of an extended graph \mathcal{G}_{ex} , constructed as follows, as illustrated in Fig. 2: given a network \mathcal{G} with κ monitors, \mathcal{G}_{ex} is obtained by adding two *virtual monitors* m'_1 and m'_2 , and 2κ *virtual links* between each pair of virtual-actual monitors.

Theorem III.3. [7] Using κ ($\kappa \geq 3$) monitors, \mathcal{G} is completely identifiable *if and only if* the associated extended graph \mathcal{G}_{ex} is 3-vertex-connected.

B. Minimum Monitor Placement for Complete Identifiability

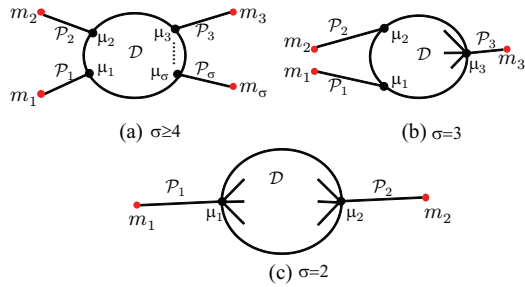
The above results have direct application in algorithm design. One application is monitor placement, where the objective is to identify all link metrics with the minimum number of monitors. The algorithm, *Minimum Monitor Placement (MMP)*, decomposes \mathcal{G} into subgraphs with certain properties (triconnected components) and sequentially places monitors in each subgraph to satisfy the condition in Theorem III.3; see [7] for details. MMP is provably optimal in that it minimizes the number of monitors to achieve complete identifiability. It is also computationally efficient with $O(|V| + |L|)$ complexity.

C. Limitation of MMP

Although MMP produces the minimum number (κ_{MMP}) of monitors required for complete identifiability, this minimum number can be large for some practical networks (see Table II in Section VI). Given a limited resource of κ monitors ($\kappa < \kappa_{\text{MMP}}$), the above result implies that it is impossible to identify all link metrics in \mathcal{G} . Nevertheless, for a given κ -monitor placement, subgraphs of \mathcal{G} may satisfy the conditions in Theorem III.2 or III.3 and thus contain identifiable links. Therefore, it is important to determine how to place the κ monitors to identify the maximum number of links. We will show that Theorems III.1–III.3 provide a theoretical foundation for addressing this problem. In the sequel, we assume $\kappa < \kappa_{\text{MMP}}$ in order to focus on the unresolved case of partially identifiable networks.

IV. QUANTITATIVE CHARACTERIZATION OF NETWORK IDENTIFIABILITY

In general, we may only be able to identify a subset of links in \mathcal{G} under an arbitrary monitor placement. This observation motivates the need for a finer-grained characterization of network identifiability in terms of the identifiability of individual links. However, there is no efficient method to determine which links are identifiable. While one can find out identifiable links by explicitly writing down the measurement matrix \mathbf{R} based on *all* possible measurement paths and then finding the null space of \mathbf{R} (links whose corresponding entries are always zero in the basis of the null space are identifiable [5]), such an approach will incur exponential complexity due to the exponentially many measurement paths. In this section, we propose an alternative algorithm that leverages the identifiability conditions in Section III to find *all* the identifiable links in \mathcal{G} (for a given monitor placement) in *linear* time.


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

A. Algorithm Overview

Our algorithm is based on two basic observations rooted in Theorems III.1–III.3 in Section III. For a 3-vertex-connected subgraph \mathcal{D} of \mathcal{G} :

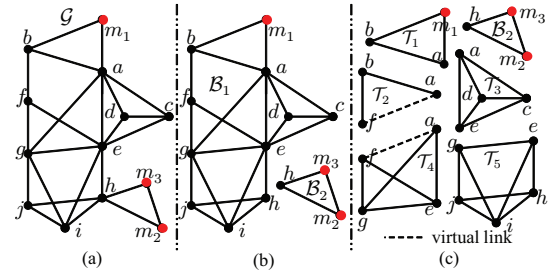
Observation 1: if \mathcal{D} contains or is connected via disjoint external (i.e., outside \mathcal{D}) links to three or more monitors, then \mathcal{D} is completely identifiable since it can be verified that the supergraph of \mathcal{D} including the monitors and their links to \mathcal{D} satisfies the condition in Theorem III.3;

Observation 2: if \mathcal{D} is connected to two external monitors, each via at least two external links, then all the links in \mathcal{D} are identifiable since it can be verified that the supergraph of \mathcal{D} including the monitors and their links to \mathcal{D} satisfies the conditions in Theorem III.2; moreover, all the external links connected to the two monitors are unidentifiable by Theorem III.1.

Given an arbitrary network \mathcal{G} , if we decompose it into 3-vertex-connected subgraphs, then the above observation can be applied to determine link identifiability in each subgraph. In general, \mathcal{D} can be connected to monitors via multi-hop paths; for the sake of identifying links in \mathcal{D} , however, we will show that it suffices to only consider *internally vertex disjoint*⁴ paths, which are then modeled as logical links. Specifically, as illustrated in Fig. 3, \mathcal{D} is connected to each monitor m_i from node μ_i via an internally vertex disjoint path \mathcal{P}_i . Note that m_i 's may contain duplicates to model multiple (internally vertex disjoint) connections to the same monitor, while μ_i 's are all distinct. Intuitively, each μ_i provides a *vantage* (see Definition 5 for the formal definition) for monitoring the internal state of \mathcal{D} . We classify the possibilities in identifying \mathcal{D} (assumed to be 3-vertex-connected) into three cases based on the number of vantages σ :

- (1) $\sigma \geq 4$ (Fig. 3(a)): Abstracting each \mathcal{P}_i as a single link, we see from Observation 1 that all links in \mathcal{D} are identifiable if at least three of m_1, \dots, m_σ are distinct; even if there are only two distinct monitors among m_1, \dots, m_σ , as long as each monitor is connected to at least two μ_i 's, we can still conclude the same from Observation 2.
- (2) $\sigma = 3$ (Fig. 3(b)): In this case, if m_1, m_2 and m_3 are distinct, then we can apply Observation 1 to conclude that all links in \mathcal{D} are identifiable; if two of m_1, m_2, m_3 are the same (e.g., $m_1 = m_2$), then Observation 2 applies, viewing $\mathcal{P}_1, \mathcal{P}_2$, and each concatenation of \mathcal{P}_3 and an adjacent link of μ_3 within \mathcal{D} as single links, implying that all links in \mathcal{D} except for those incident to μ_3 are identifiable (note that μ_3 has at least three links in \mathcal{D} since \mathcal{D} is 3-vertex-connected).
- (3) $\sigma = 2$ (Fig. 3(c)): If m_1 and m_2 are distinct, then we can apply Observation 2, viewing each concatenation of \mathcal{P}_1 (\mathcal{P}_2) and an adjacent link of μ_1 (μ_2) as a single link, to

⁴That is, the paths are disjoint except at end-points.


 Fig. 4. Decomposition of a sample network: (a) sample network \mathcal{G} ; (b) biconnected components of \mathcal{G} ; (c) triconnected components of \mathcal{G} .

conclude that all links in \mathcal{D} except for those incident to μ_1 or μ_2 are identifiable.

We will show that these three cases represent all scenarios of interest in determining link identifiability of a 3-vertex-connected subgraph (e.g., the scenario of \mathcal{D} connected to only one monitor is not of interest since none of its links can be probed via cycle-free paths); see Lemma IV.2 and Section IV-B3. Our strategy for determining identifiable links in a general graph is to first decompose the graph into subgraphs falling into one of the above cases, and then determine the set of identifiable links in each subgraph.

B. Algorithm Details

1) *Decomposition into Triconnected Components:* The first step of our solution is to decompose the graph into suitably connected subgraphs. To this end, 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 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 as in Section II-A). Similarly, a triconnected component can be a 3-vertex-connected subgraph, a triangle, or a single link. Intuitively, a biconnected component is a subgraph connected to the rest of the graph by cut-vertices⁵, and a triconnected component (which always resides in a biconnected component) is a subgraph connected to the rest of the biconnected component by 2-vertex cuts⁶. For instance, Fig. 4(b) shows the biconnected components of Fig. 4(a), separated by cut-vertex h . Fig. 4(c) shows the triconnected components, separated by cut-vertex h and 2-vertex cuts $\{a, b\}$, $\{a, f\}$, $\{a, e\}$, and $\{e, g\}$. However, not all subgraphs separated by 2-vertex cuts form triconnected components according to Definition 2 (e.g., \mathcal{T}_2 in Fig. 4(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 2-vertex 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 decompose an arbitrary graph \mathcal{G} into biconnected components [16] and then triconnected

⁵A *cut-vertex* is a vertex whose removal will disconnect the graph.

⁶A *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} .

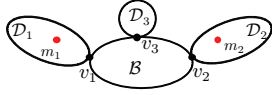


Fig. 5. Agent: v_1 and v_2 are agents w.r.t. \mathcal{B} , but v_3 is not (m_1, m_2 : monitors).

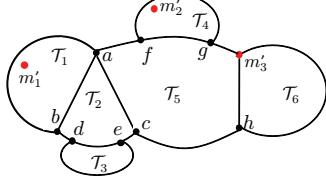


Fig. 6. Type- k vertex cut: $\mathcal{T}_1, \dots, \mathcal{T}_6$ are triconnected components within the same biconnected component, for which m'_1, m'_2, m'_3 are agents.

components [17]. Note that the output of [17] is the set of triconnected components of the processed graph⁷.

2) *Classification of Triconnected Components*: After the decomposition, the next step is to determine to which case of Fig. 3 each triconnected component belongs. To this end, we introduce an auxiliary algorithm (Algorithm 1), which builds upon the graph decomposition algorithms [16], [17] to classify the triconnected components according to their connections to monitors. Intuitively, nodes on the boundary of a triconnected component, i.e., cut-vertices and nodes in 2-vertex cuts, play crucial roles in determining its connections to monitors. The classification is therefore based on a few notions about these boundary nodes as introduced in Definitions 3–5 below.

The first notion, *agent*, relaxes the notion of monitor to also include nodes that effectively act as monitors.

Definition 3. An *agent* w.r.t. a subgraph \mathcal{G}' is either a monitor in \mathcal{G}' or a cut-vertex that separates \mathcal{G}' from at least one monitor.

We illustrate the concept of an agent in Fig. 5. Consider a subgraph \mathcal{B} illustrated in Fig. 5, 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 agents w.r.t. \mathcal{B} since \mathcal{D}_1 and \mathcal{D}_2 both contain monitors; v_3 is not an agent w.r.t. \mathcal{B} as \mathcal{D}_3 does not contain any monitor (but it is an agent w.r.t. \mathcal{D}_3). The significance of agents is that if \mathcal{B} is a biconnected component, they essentially serve as “monitors” for identifying links in \mathcal{B} as stated below.

Lemma IV.1. Let \mathcal{B} be a biconnected component with agents m'_1, \dots, m'_κ . The identifiability of links in \mathcal{B} does not depend on whether m'_1, \dots, m'_κ are monitors or not, except for link $m'_1 m'_2$ (if it exists) when $\kappa = 2$.

Proof: see [18]. ■

Remark: Since each agent is associated with at least one monitor, the number of agents w.r.t. a biconnected component cannot exceed the total number of monitors. Clearly, for any biconnected component with fewer than two agents, none of its links is identifiable since any monitor-to-monitor path going into this component via a node must leave through the same node, creating a cycle. Therefore, it suffices to only consider biconnected components with two or more agents.

The second notion, *type- k vertex cut*, classifies 2-vertex cuts based on the number of (effective) monitors behind a cut.

Definition 4. A *type- k vertex cut* (denoted by Type- k -VC) w.r.t. a triconnected component \mathcal{T} is a 2-vertex cut $\{v_1, v_2\}$ in \mathcal{T} that separates \mathcal{T} from k agents (excluding agents in $\{v_1, v_2\}$) in its parent biconnected component (i.e., the biconnected component containing \mathcal{T}).

⁷The algorithm in [17] does not require preprocessing of adding virtual links. Virtual links are added as needed during graph partitioning.

Algorithm 1: Classification of Triconnected Components

```

input : Connected graph  $\mathcal{G}$  with given monitor placement
output: Triconnected components and their categories
1 partition  $\mathcal{G}$  into biconnected components  $\mathcal{B}_1, \mathcal{B}_2, \dots$  [16] and
   identify agents for each biconnected component;
2 foreach biconnected component  $\mathcal{B}_i$  with 2 or more agents do
3   partition  $\mathcal{B}_i$  into triconnected components  $\mathcal{T}_1, \mathcal{T}_2, \dots$  [17];
4   foreach triconnected component  $\mathcal{T}_j$  of  $\mathcal{B}_i$  do
5     identify vantages of  $\mathcal{T}_j$  to determine its category;
6     if  $|\mathcal{T}_j| = 3$  then
7       record all neighboring triconnected components;
8     end
9   end
10 end

```

We illustrate the concept of a type- k vertex cut in Fig. 6, which contains a biconnected component consisting of six triconnected components and three agents. Each triconnected component \mathcal{T}_i in Fig. 6 is separated from the rest by a set of 2-vertex cuts, each cut separating \mathcal{T}_i from $k \in \{0, 1, 2, 3\}$ agents. The value of k determines the type of this cut. Specifically, in Fig. 6, we have: (i) Type-2-VC $\{a, b\}$ w.r.t. \mathcal{T}_1 , (ii) Type-1-VC $\{a, b\}$, Type-2-VC $\{a, c\}$ and Type-0-VC $\{d, e\}$ w.r.t. \mathcal{T}_2 , (iii) Type-3-VC $\{d, e\}$ w.r.t. \mathcal{T}_3 , (iv) Type-2-VC $\{f, g\}$ w.r.t. \mathcal{T}_4 , (v) Type-1-VC $\{f, g\}$, Type-0-VC $\{m'_3, h\}$ and Type-1-VC $\{a, c\}$ w.r.t. \mathcal{T}_5 , and (vi) Type-2-VC $\{m'_3, h\}$ w.r.t. \mathcal{T}_6 .

The third notion, *vantage*, further relaxes the notion of agent to also include nodes in 2-vertex cuts with connections to monitors.

Definition 5. A *vantage* (denoted by μ) w.r.t. to a triconnected component \mathcal{T} is either an agent w.r.t. \mathcal{T} or a node in a Type- k -VC w.r.t. \mathcal{T} with $k \geq 1$.

Vantages w.r.t. a triconnected component \mathcal{T} are the nodes in \mathcal{T} that are connected to monitors, or equivalently agents, through *external paths* (i.e., paths not involving links in \mathcal{T} , including the degenerate path if the vantage is also an agent). For example, if the subgraph \mathcal{D} in Fig. 3 is a triconnected component, then all the nodes labeled μ_i ($i = 1, 2, \dots$) are vantages w.r.t. \mathcal{D} . We use Fig. 6 to illustrate the concept of vantages. According to Definition 5, the vantages for different triconnected components are (i) a, b, m'_1 for \mathcal{T}_1 ; (ii) a, b, c for \mathcal{T}_2 ; (iii) d, e for \mathcal{T}_3 ; (iv) f, g, m'_2 for \mathcal{T}_4 ; (v) a, c, f, g, m'_3 for \mathcal{T}_5 ; and (vi) h, m'_3 for \mathcal{T}_6 .

Based on the above definitions, Algorithm 1 classifies all triconnected components of \mathcal{G} according to the numbers of vantages that they contain. It first decomposes \mathcal{G} into biconnected components using the algorithm in [16] and identifies the agents in each biconnected component (line 1). It then further decomposes each biconnected component of interest (i.e., with two or more agents) into triconnected components using the algorithm in [17] (line 3). For each triconnected component, it identifies all the vantages (line 5) by first determining the types of its 2-vertex cuts (w.r.t. itself) according to Definition 4, and then applying Definition 5. Based on the number of vantages, Algorithm 1 classifies the triconnected component into one of the following three categories (line 5):

- (i) **Category 1**: containing 4 or more vantages (e.g., \mathcal{T}_5 in Fig. 6);
- (ii) **Category 2**: containing 3 vantages (e.g., $\mathcal{T}_1, \mathcal{T}_2$ and \mathcal{T}_4 in Fig. 6);
- (iii) **Category 3**: containing 2 vantages (e.g., \mathcal{T}_3 and \mathcal{T}_6 in Fig. 6).

If the triconnected component has only three nodes (i.e., triangles), additional information is recorded (line 7), the purpose of

Algorithm 2: Determination of All Identifiable Links (DAIL)

input : Connected graph \mathcal{G} with a given monitor placement
output: All identifiable links in \mathcal{G}

- 1 decompose \mathcal{G} into triconnected components⁸ and determine the category of each triconnected component by Algorithm 1;
- 2 **foreach** triconnected component \mathcal{T}_i within a biconnected component with 2 or more agents **do**
- 3 **if** \mathcal{T}_i is of Category 1 **then**
- 4 all links in \mathcal{T}_i are identifiable;
- 5 **else if** \mathcal{T}_i is of Category 2 and 3-vertex-connected **then**
- 6 **if** only one of the three vantages in \mathcal{T}_i is an agent, the other two form a Type-1-VC, and there is no other Type- k -VC with $k \geq 1$ in \mathcal{T}_i **then**
- 7 all links in \mathcal{T}_i except for the ones incident to the agent are identifiable;
- 8 **else**
- 9 all links in \mathcal{T}_i are identifiable;
- 10 **end**
- 11 **else if** \mathcal{T}_i is of Category 2 and \mathcal{T}_i is a triangle **then**
- 12 the identification of \mathcal{T}_i is determined by Algorithm A in [18];
- 13 **else** // \mathcal{T}_i must be of Category 3
- 14 all links in \mathcal{T}_i except for the ones incident to the two vantages (μ_1 and μ_2) are identifiable; if \exists link $\mu_1\mu_2$, then the identifiability of $\mu_1\mu_2$ is determined by Algorithm B in [18];
- 15 **end**
- 16 **end**

which will be explained later (see Remark in Section IV-B3).

As explained later in Section IV-B3, these categories are in 1-1 correspondence with the three cases introduced in Section IV-A and illustrated in Fig. 3. The following lemma guarantees that the above classification is complete for all the non-trivial cases (recall that biconnected components with fewer than two agents cannot have any identifiable link as discussed after Lemma IV.1).

Lemma IV.2. Any triconnected component within a biconnected component with two or more agents belongs to either one of **Category 1, 2, or 3**.

Proof: See [18]. ■

3) *Determination of All Identifiable Links:* Based on the classification of triconnected components, we propose Algorithm 2, *Determination of All Identifiable Links (DAIL)*, to determine all the identifiable links in \mathcal{G} under a given monitor placement⁹. DAIL sequentially determines the identifiable links in each triconnected component of interest (i.e., in a biconnected component with 2 or more agents) based on its category (lines 3-15). These steps are best understood by referring to the illustration in Fig. 3, where each m_i represents an agent.

Category 1 (processed by line 4) is equivalent to the case in Fig. 3(a). In particular, Category 1 allows different vantages to connect to the same agent (i.e., m_1, \dots, m_σ in Fig. 3(a) may not be distinct), implying that all links in a Category 1 triconnected component are identifiable as long as the number of vantages is at least four. This is because the 2-vertex-

⁸According to Theorem III.10 in [18], all triconnected components that are 3-vertex-connected (i.e., containing 4 or more nodes) or consist of a single-link are unique over all triconnected component decompositions, but the triangle components may not be unique. Nevertheless, Theorem III.11 in [18] guarantees that the set of identifiable links determined by Algorithm 2 does not depend on the specific decomposition used.

⁹Although we focus on the case of $\kappa < \kappa_{\text{MMP}}$, DAIL is also applicable to networks with $\kappa \geq \kappa_{\text{MMP}}$ monitors.

connectivity of the parent biconnected component ensures the existence (see Lemma III.5 in [18]) of pairwise internally vertex disjoint paths $\{\mathcal{P}_1, \dots, \mathcal{P}_\sigma\}$ as shown in Fig. 3(a). Accordingly, abstracting all these paths as single links, the graph in Fig. 3(a) always satisfies Theorem III.3. Similarly, Category 2 (processed by lines 6–10 and 12) and Category 3 (processed by line 14) are equivalent to the cases in Fig. 3(b) and Fig. 3(c), respectively. In particular, for a 3-vertex-connected component \mathcal{T} of Category 2, we have shown in the explanation of Fig. 3(b) (Section IV-A) that the connection between \mathcal{T} and (external) agents is crucial in determining the identifiability of \mathcal{T} , i.e., depending on the number of distinct agents, \mathcal{T} can be partially or completely identifiable. In this regard, lines 7 and 9 in Algorithm 2 are used to capture these factors affecting the identifiability of \mathcal{T} of Category 2.

Remark: Link identifiability in a triconnected component cannot be directly determined from our previous observations in some special cases as follows. First, we have assumed that all the links can be employed by measurement paths, whereas the virtual links added during triconnected component decomposition (see explanations after Definition 2) cannot be used for measurements. Moreover, we have indicated in Lemma IV.1 that the identifiability of the direct agent-to-agent link in a biconnected component (or the link between two vantages in a triconnected component of Category 3) cannot be determined based on the internal structure of the component. For the first issue, we have proved in [18] that link identifiability in most triconnected components remains the same in the presence of virtual links. The only exceptions are Categories 2 triangle components, which are handled separately in line 12 (by calling Algorithm A in [18]). To determine link identifiability of a (Categories 2) triangle component \mathcal{T} , Algorithm A requires its neighboring triconnected components (see [18] for explanations); therefore, line 7 in Algorithm 1 is needed to provide this information. For the second issue, we separately determine the identifiability of direct links in triconnected components of Category 3 using Algorithm B in [18].

For Algorithm 2, Theorem III.1 (together with the observation that no link is identifiable in a biconnected component with fewer than two agents) guarantees that links not classified as identifiable by DAIL are unidentifiable, i.e., DAIL determines *all* the identifiable links. Thus, we have Theorem IV.3.

Theorem IV.3. Algorithm 2 (DAIL) can determine all identifiable links in \mathcal{G} for a given monitor placement.

Proof: See [18]. ■

C. Complexity Analysis

In Algorithm 1, partitioning \mathcal{G} into biconnected components (line 1) and then triconnected components (line 3) takes $O(|V| + |L|)$ time [16], [17]. The classification of triconnected components (lines 4-9 of Algorithm 1) can be performed during the partitioning with a constant-factor increase in the complexity. Thus, Algorithm 1, line 1 of Algorithm 2 (DAIL), has $O(|V| + |L|)$ time complexity. In DAIL, the complexity of lines 3-15 is $O(|L(\mathcal{T}_i)|)$ for each triconnected component \mathcal{T}_i (Algorithms A and B in [18] each takes $O(1)$ time). Therefore, the overall complexity of DAIL is $O(|V| + |L|)$.

D. Example

The sample network in Fig. 4(a) has 2 biconnected components \mathcal{B}_1 and \mathcal{B}_2 (Fig. 4(b)) with agents $\{m_1, h\}$ and $\{h, m_2, m_3\}$, respectively. There are 5 triconnected components in \mathcal{B}_1 (Fig. 4(c)), where $\{\mathcal{T}_4\}$, $\{\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_5\}$, and $\{\mathcal{T}_3\}$ belong to Categories 1, 2, and 3, respectively. Moreover, \mathcal{B}_2 itself

is a Category 2 triconnected component. Therefore, the identifiable links are $\{ab, bf, cd, ae, ag, fe, fg, eg, ei, gi, gj, ij\}$ in \mathcal{B}_1 , and $\{hm_2, hm_3, m_2m_3\}$ in \mathcal{B}_2 .

V. MAXIMAL IDENTIFIABILITY MONITOR PLACEMENT

DAIL (Algorithm 2) allows us to determine the number of identifiable links in an arbitrary network with a given placement of monitors in $O(|V| + |L|)$ time. A natural followup question is: Given the constraint of deploying only κ monitors with $\kappa < \kappa_{\text{MMP}}$, i.e., the network cannot be completely identified, what is the monitor placement that maximizes the number of identifiable links? While theoretically the problem can be solved by running DAIL on all possible placements of κ monitors and selecting the one with the maximum number of identifiable links, such a solution cannot scale to large networks due to the large number of possible placements (which is at least $\Omega(|V|^\kappa)$ for fixed κ , and can be exponential in $|V|$ for κ proportional to $|V|$). In this section, we develop an efficient solution to this problem by first identifying candidates that can be part of an optimal monitor placement, and then developing an efficient algorithm to incrementally select monitors from the candidates. The significance of this algorithm is that it achieves optimality for a certain family of network topologies.

A. Candidate Monitor Selection

The first step of candidate selection is performed by Algorithm 3, based on the following argument. For a 3-vertex-connected network \mathcal{G} , Theorems III.1 and III.2 imply that, for a given 2-monitor placement, all links in \mathcal{G} except for the links incident to the monitors are identifiable (note that the direct link connecting the two monitors is always identifiable). Therefore, the optimal 2-monitor placement in \mathcal{G} must be two nodes v_1 and v_2 with the minimum total number of adjacent links, excluding the direct link (as in lines 3 and 5 of Algorithm 3). For the same \mathcal{G} , if three monitors can be deployed, then \mathcal{G} is completely identifiable according to Theorem III.3, and no extra monitor is needed (although any placement of 3 monitors will achieve complete identifiability, the specific choice in line 9 is based on Theorem III.12 in [18]). These observations inspire us to deploy monitors only in a subset \mathcal{S} of nodes, known as *candidates*. The value of \mathcal{S} is that it helps to reduce the set of nodes considered in monitor placement without losing optimality.

Algorithm 3: Candidate Monitor Selection

input : Triconnected components of \mathcal{G} (computed by [17])
output: Set of candidates \mathcal{S}

```

1 foreach triconnected component  $\mathcal{T}$  of  $\mathcal{G}$  do
2    $\mathcal{A}^{\mathcal{T}}(v)$ : set of adjacent (non-virtual)
   links of node  $v$  in  $\mathcal{T}$ ;
3   Under  $|\mathcal{A}^{\mathcal{T}}(v_1)| = \min_{v \in V(\mathcal{T})} |\mathcal{A}^{\mathcal{T}}(v)|$ , select  $v_1$  and  $v_2$ 
   with the minimum  $|\mathcal{A}^{\mathcal{T}}(v_1) \cup \mathcal{A}^{\mathcal{T}}(v_2)|$ ;
4   if the selection in line 3 is not unique then
5     Among alternatives generated by line 3, select
      $\{v_1, v_2\}$  with the minimum
      $|\mathcal{A}^{\mathcal{T}}(v_1) \cup \mathcal{A}^{\mathcal{T}}(v_2) \setminus \{v_1v_2\}|$  (break ties arbitrarily);
6   end
7   add  $\{v_1, v_2\}$  to  $\mathcal{S}$ ;
8   if  $|\mathcal{T}| \geq 3$  then
9     find  $v_3$  with the minimum  $|\mathcal{A}^{\mathcal{T}}(v)|$  over
      $v \in V(\mathcal{T}) \setminus \{v_1, v_2\}$ ;
10    add  $v_3$  to  $\mathcal{S}$ ;
11  end
12 end

```

Algorithm 4: Greedy Maximal identifiability Monitor Placement (GMMP)

input : Connected graph \mathcal{G} , its candidate set \mathcal{S}
output: A κ -monitor placement \mathcal{O}_κ

```

1 if  $\kappa = 2$  then
2   enumerate all pairs of nodes in  $\mathcal{S}$ , return
    $\mathcal{O}_2 = \arg \max_{v_1, v_2 \in \mathcal{S}} N(\{v_1, v_2\})$ ;
3 else if  $\kappa \geq 3$  then
4   enumerate all triples of nodes in  $\mathcal{S}$ , return
    $\mathcal{O}_3 = \arg \max_{v_1, v_2, v_3 \in \mathcal{S}} N(\{v_1, v_2, v_3\})$ ;
5   foreach  $i, i = \{4, \dots, \kappa\}$  do
6     find  $v_m = \arg \max_v N(\mathcal{O}_{i-1} \cup \{v\})$  over
      $v \in \mathcal{S} \setminus \mathcal{O}_{i-1}$ ;
7     set  $\mathcal{O}_i = \mathcal{O}_{i-1} \cup \{v_m\}$ ;
8   end
9 end
10 return  $\mathcal{O}_\kappa$ ;

```

Theorem V.1. The candidate set \mathcal{S} selected by Algorithm 3 always contains an optimal monitor placement as a subset.

Proof: see [18]. ■

Let $N_{\mathcal{T}}$ denote the total number of triconnected components¹⁰ in \mathcal{G} . Since each triconnected component contains at most three candidates, the total number of candidates in \mathcal{S} ($|\mathcal{S}|$ can be much smaller than $|V|$) is upper bounded by $3N_{\mathcal{T}}$.

B. Monitor Placement Algorithm

Given an algorithm to calculate the number of identifiable links (using Algorithm 2) and preprocessing to identify candidate monitors (using Algorithm 3), we are ready to develop a monitor placement algorithm to maximize (partial) identifiability. While the optimal solution is combinatorial in nature and therefore likely to be computationally expensive, we propose a simple low-complexity algorithm that we show, both theoretically and empirically, to have satisfactory performance.

We propose an incremental greedy algorithm, *Greedy Maximal identifiability Monitor Placement (GMMP)*, that builds upon a suitably-constructed initial placement to select additional monitors, one at a time, so that each selection maximizes the gain in identifiability; see Algorithm 4, where $N(\mathcal{O})$ denotes the number of identifiable links under the monitor placement \mathcal{O} (as defined in Table I). The bulk of computation is in constructing this initial placement. GMMP constructs an optimal κ_0 -monitor placement for a small value of κ_0 ($\kappa_0 = 2$ for $\kappa = 2$, or 3 for $\kappa \geq 3$) by exhaustively enumerating all selections of κ_0 monitors from the candidate set \mathcal{S} (lines 2, 4). If more monitors are available ($\kappa > 3$), it then iteratively selects additional monitors so that each newly selected monitor maximizes the number of additional identifiable links (lines 5-8).

The reason for handling the cases of $\kappa = 2$ and $\kappa \geq 3$ separately is that the optimal 3-monitor placement \mathcal{O}_3 is not necessarily a superset of the optimal 2-monitor placement \mathcal{O}_2 . For example, Fig. 7 is a graph consisting of a large triconnected component \mathcal{T} and three dangling nodes connected to \mathcal{T} via three cut-vertices. Then \mathcal{O}_2 can be two nodes $\{v_1, v_2\}$ in \mathcal{T} with the minimum number of neighbors within \mathcal{T} , while \mathcal{O}_3 are the dangling nodes $\{v_a, v_b, v_c\}$ (which can be shown to identify all the links in this graph by Theorem III.3).

Example: Fig. 8 illustrates which nodes are selected as monitors by GMMP in a sample network for $\kappa = 3, 4$, and 5.

¹⁰Although the decomposition into triconnected components may not be unique, the total number of triconnected components $N_{\mathcal{T}}$ is always the same according to Theorem III.10 in [18].

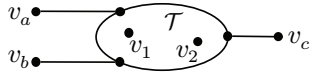


Fig. 7. Example: $\mathcal{O}_2 = \{v_1, v_2\}$ is not a subset of $\mathcal{O}_3 = \{v_a, v_b, v_c\}$.

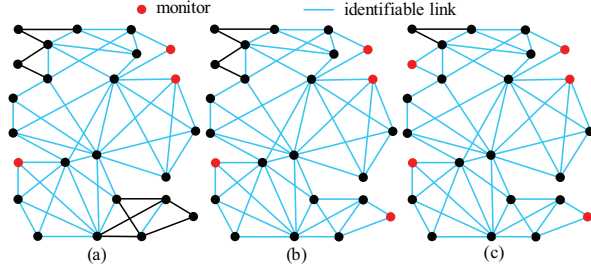


Fig. 8. Monitor placement by GMMP in a sample network: (a) $\kappa = 3$; (b) $\kappa = 4$; (c) $\kappa = 5$, where $\mathcal{O}_3 \subset \mathcal{O}_4 \subset \mathcal{O}_5$.

Complexity analysis: The complexity of GMMP depends on the number of monitors κ . If $\kappa = 2$, \mathcal{O}_2 is directly obtained by line 2. Since the complexity for computing the number of identifiable links (using Algorithm 2) is $O(|V| + |L|)$, and there are $\binom{|S|}{2}$ possible two-monitor placements, the complexity of line 2 is $O(|S|^2(|V| + |L|))$. If $\kappa \geq 3$, the computation involves constructing \mathcal{O}_3 by line 4 and incrementally expanding it into \mathcal{O}_κ . Line 4 incurs $O(|S|^3(|V| + |L|))$ complexity for examining identifiability of all the $\binom{|S|}{3}$ monitor placements. In line 6, the $\arg \max$ operation involves $|S| - i + 1$ calls of Algorithm 2 to compute v_m , resulting in a complexity of $O(\kappa|S|(|V| + |L|))$ for lines 5-8. Therefore, the overall complexity for the case of $\kappa \geq 3$ is $O((|S|^3 + \kappa|S|)(|V| + |L|))$, which is the same as $O(N_{\mathcal{T}}^3(|V| + |L|))$, as both $|S|$ and κ ($\kappa < \kappa_{\text{MMP}} \leq |S|$) are in $O(N_{\mathcal{T}})$ (see Algorithm MMP in [7] for the reason of $\kappa_{\text{MMP}} \leq |S|$). Note that usually $N_{\mathcal{T}} \ll |V|$.

C. Optimality for 2-vertex-connected Networks

Besides being an efficient heuristic, GMMP also has theoretical significance in that it is optimal for a family of network topologies that are 2-vertex-connected. Consider a graph \mathcal{G} that is 2-vertex-connected, i.e., containing no cut-vertex. Let \mathcal{M} be a minimum set of monitors given by MMP¹¹ [7] for complete identification of \mathcal{G} . Let \mathcal{O}_κ^* denote an optimal κ -monitor placement in \mathcal{G} (\mathcal{O}_κ^* may not be unique). We have the following theorem ($N(\mathcal{O})$ is defined in Table I).

Theorem V.2. If \mathcal{G} is 2-vertex-connected and $3 \leq \kappa < |\mathcal{M}|$, then (i) $\exists \mathcal{O}_\kappa^*$ such that $\mathcal{O}_\kappa^* \subseteq \mathcal{M}$, and (ii) for any given \mathcal{O}_κ^* with $\mathcal{O}_\kappa^* \subseteq \mathcal{M}$, $\exists \mathcal{O}_{\kappa+1}^*$ such that $\mathcal{O}_{\kappa+1}^* = \mathcal{O}_\kappa^* \cup \{v_m\}$, where $v_m = \arg \max_v N(\mathcal{O}_\kappa^* \cup \{v\})$ over $v \in \mathcal{M} \setminus \mathcal{O}_\kappa^*$.

Proof: see [18]. ■

Theorem V.2 states that for 2-vertex-connected networks, it is always possible to find an optimal κ -monitor placement ($\kappa \geq 3$) within \mathcal{M} (which is a stronger statement than finding an optimal placement within \mathcal{S} as \mathcal{M} is a subset of \mathcal{S}); moreover, we can always obtain an optimal $(\kappa + 1)$ -monitor placement by expanding an optimal κ -monitor placement with the monitor that maximizes the increase in the number of identifiable links. Applied to GMMP, this result implies that the monitor placement computed by GMMP is optimal as long as \mathcal{G} is 2-vertex-connected (e.g., the monitor placement for the sample 2-vertex-connected network in Fig. 8 is optimal, and

¹¹This set may not be unique, but the number of nodes in \mathcal{M} , i.e., $|\mathcal{M}|$, is unique, and our argument in Theorem V.2 holds for any such \mathcal{M} .

TABLE II. ISP NETWORKS

ISP	$ L $	$ V $	$ S $	N_B	$N_{\mathcal{T}}$	κ_{MMP}	$\kappa_{\text{MMP}}/ V $
Abovenet (US)	294	182	148	87	129	117	0.64
EBONE (Europe)	381	172	81	28	65	55	0.32
Exodus (US)	434	201	108	38	92	85	0.42

$\mathcal{O}_3^* \subset \mathcal{O}_4^* \subset \mathcal{O}_5^*$). We point out that the 2-vertex-connectivity is naturally satisfied by any network that has no single point of failure (no failure of one node will disconnect the network). Nevertheless, we will show in Section VI that GMMP is near-optimal even if the network is not 2-vertex-connected.

VI. EVALUATION

We evaluate GMMP on the *Internet Service Provider (ISP)* topologies collected by the Rocketfuel project [19], which represent physical connections between backbone/gateway routers of several ISPs around the globe. We select three ISPs of similar sizes to evaluate the impact of different topologies; see Table II for the details, where N_B denotes the number of biconnected components. We apply GMMP to each ISP, with a focus on evaluating the *identification ratio* (ratio of number of identifiable links over total number of links) to compare networks of different sizes. We also evaluate algorithm efficiency, measured by its running time.

As a benchmark for GMMP, we use the following algorithm, referred to as Random Monitor Placement (RMP): Given a total number of monitors κ and a node set Φ , RMP randomly selects κ nodes out of Φ as monitors and examines the corresponding number of identifiable links by DAIL (Algorithm 2). This process is repeated δ times, and the selection yielding the largest number of identifiable links is returned as the final placement by RMP. The basic version for RMP, denoted by RMP- V , is to select nodes from the entire network, i.e., $\Phi = V$. A more advanced version of RMP, denoted by RMP- \mathcal{S} , limits selection to the candidate set \mathcal{S} , i.e., $\Phi = \mathcal{S}$, based on the observation in Theorem V.1. We employ both versions of RMP to evaluate the performance impact of using a refined candidate set. As a randomized alternative to exhaustive enumeration, RMP trades off complexity against performance via tuning parameter δ . In this evaluation, we set δ to $\kappa|S|$ for both versions of RMP, which implies a complexity of $O(\kappa|S|(|V| + |L|))$. We notice that GMMP in its current form (Algorithm 4) will incur a higher complexity due to the enumeration of all 3-monitor placements (line 4); in fact, even a complete enumeration of 2-monitor placements (line 2) will incur complexity of $O(|S|^2(|V| + |L|))$, which is already higher than the complexity of RMP for small κ . To ensure fair comparison with RMP, we simplify GMMP by only examining $\kappa|S|$ randomly selected 2-monitor placements and using the one with the maximum number of identifiable links to approximate \mathcal{O}_2 ; $\mathcal{O}_3, \dots, \mathcal{O}_\kappa$ are then incrementally constructed based on \mathcal{O}_2 by repeating lines 6-7 of Algorithm 4. In the following, all results labeled “GMMP” are for this simplified GMMP. We perform the simulations using Matlab R2010a on a computer with Intel(R) Core i7-2720QM CPU @ 2.2GHz, 16.0 GB memory, and 64-bit Win7 OS.

We report the identification ratio in Fig. 9, where the optimal values are obtained by an exhaustive search over all κ -monitor placements. We make the following observations:

(1) Although complete identification of the ISP networks requires a substantial fraction of nodes to be monitors (0.32 to 0.64 as shown in Table II), many links can be identified with only a few monitors, e.g., both GMMP and RMP- \mathcal{S} identify more than half of the links in all the ISPs by placing monitors at only 5% of the nodes. This is because most ISP networks

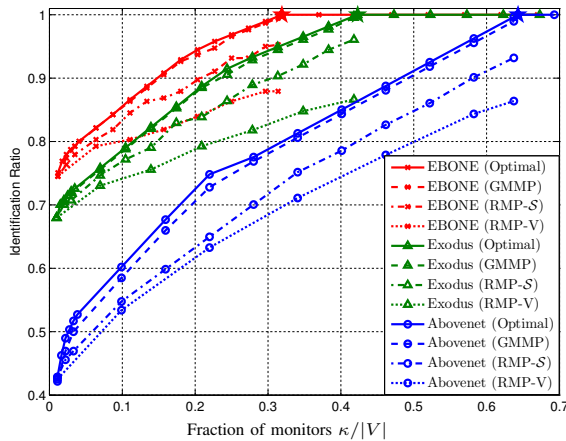


Fig. 9. Identification ratio in ISP networks; \star : $(\kappa_{\text{MMP}}/|V|, 1)$, i.e., transition point for complete identification¹².

contain at least one large subnetwork (likely the backbone) that tends to be 3-vertex-connected, which can be completely identified using a small number of properly placed monitors.

(2) GMMP can be suboptimal for non-2-vertex-connected networks (none of the ISPs is 2-vertex-connected, as verified by $N_B > 1$ in Table II). Nevertheless, it closely approximates the optimal identification ratio for all the networks over all considered values of κ . This observation suggests that, besides being provably optimal for 2-vertex-connected networks, GMMP is also near-optimal for general networks (note that this is by the simplified GMMP, and the identification ratio of the original GMMP can be higher). In comparison, RMP-V shows a notably lower identification ratio, e.g., to reach identification ratio 0.85 in EBONE, 16 monitors are required by GMMP, while 39 are required by RMP-V. Notice that RMP-S achieves intermediate performance, e.g., requiring 21 monitors in EBONE to achieve identification ratio 0.85. For the same set of evaluations, we have also simulated algorithm running time which shows that GMMP is faster than RMP-V and RMP-S; see Section V in [18] for details.

These results not only confirm the performance improvement of GMMP over the benchmarks, but also reveal the factors contributing to this improvement: refined set of candidate monitors (shown by the improvement of RMP-S over RMP-V) and strategic selection of monitors from the candidate set (shown by the improvement of GMMP over RMP-S). Note that all the algorithms, including the exhaustive search, depend on the proposed algorithm DAIL to evaluate the number of identifiable links under each candidate placement, which otherwise cannot be determined efficiently (see discussions at the beginning of Section IV).

(3) Among the three ISPs, we see that EBONE has the highest identification ratio while Abovenet has the lowest. Intuitively, this is because EBONE is more densely connected (as shown by a smaller N_B in Table II), which gives more flexibility in constructing measurement paths between monitors, and thus requires fewer monitors for identifying its links; in contrast, Abovenet is poorly connected with a large number of cut-vertices (as shown by a larger N_B), which limits the set of measurable paths between each pair of monitors and thus requires more monitors. This observation suggests that networks with better connectivity are more amenable to

network tomography under limited monitor resources.

Discussions: Although we have used the *number* of identifiable links as the optimization criterion, all the evaluated algorithms are applicable to a more general criterion given as an arbitrary mapping from the set of identifiable links to a utility value, as DAIL can actually determine the *set* of identifiable links. We leave performance analysis and evaluation in this generalized setting to future work.

VII. CONCLUSION

We studied the problem of maximal identifiability monitor placement in network tomography, for which we developed an efficient greedy algorithm that incrementally places monitors to maximize the number of identifiable links, based on an algorithm that determines all identifiable links in linear time. We showed that the proposed algorithm is both provably optimal for networks without cut-vertices and empirically near-optimal for several real network topologies. Compared with existing monitor placement solutions that focus on complete identification, the proposed algorithm provides a quantifiable tradeoff between the level of identifiability and the number of monitors.

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, H. Song, 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] Y. Xia and D. Tse, "Inference of link delay in communication networks," *IEEE JSAC*, vol. 24, no. 12, pp. 2235–2248, 2006.
- [9] 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, 2012.
- [10] L. Ma, T. He, K. Leung, D. Towsley, and A. Swami, "Efficient identification of additive link metrics via network tomography," in *ICDCS*, 2013.
- [11] A. Gopalan and S. Ramasubramanian, "On the maximum number of linearly independent cycles and paths in a network," Technical Report, August 2012. [Online]. Available: <http://www2.engr.arizona.edu/%7Esrini/papers/lirank.pdf>
- [12] L. Ma, T. He, K. K. Leung, A. Swami, and D. Towsley, "Link identifiability in communication networks with two monitors," in *IEEE GLOBECOM*, 2013.
- [13] Y. Bejerano and R. Rastogi, "Robust monitoring of link delays and faults in IP networks," in *IEEE INFOCOM*, 2005.
- [14] R. Kumar and J. Kaur, "Practical beacon placement for link monitoring using network tomography," *IEEE JSAC*, vol. 24, no. 12, pp. 2196–2209, 2006.
- [15] J. D. Horton and A. López-Ortiz, "On the number of distributed measurement points for network tomography," in *ACM IMC*, 2003.
- [16] R. Tarjan, "Depth-first search and linear graph algorithms," *SIAM Journal on Computing*, vol. 1, pp. 146–160, 1972.
- [17] J. E. Hopcroft and R. E. Tarjan, "Dividing a graph into triconnected components," *SIAM Journal on Computing*, vol. 2, pp. 135–158, 1973.
- [18] L. Ma, T. He, K. K. Leung, A. Swami, and D. Towsley, "Partial network identifiability: Theorem proof and evaluation," Technical Report, Imperial College, London, UK., Jul. 2013. [Online]. Available: <http://www.commsp.ee.ic.ac.uk/%7Eelm110/pdf/MaTRJul13.pdf>
- [19] "Rocketfuel: An ISP topology mapping engine," University of Washington, 2002. [Online]. Available: <http://www.cs.washington.edu/research/networking/rocketfuel/>

¹²We only simulate $\kappa/|V| \in (0, \kappa_{\text{MMP}}/|V|)$, as the identification ratio of 1 can always be achieved by MMP in [7] for $\kappa/|V| \geq \kappa_{\text{MMP}}/|V|$.