Identifiability of Link Metrics Based on End-to-end Path Measurements *

Liang Ma
Imperial College
London, UK
l.ma10@imperial.ac.uk

Ting He
IBM T. J. Watson Research
Yorktown, NY, USA
the@us.ibm.com

Kin K. Leung
Imperial College
London, UK
kin.leung@imperial.ac.uk

Ananthram Swami
Army Research Laboratory
Adelphi, MD, USA
a.swami@ieee.org

Don Towsley
University of Massachusetts
Amherst, MA, USA
towsley@cs.umass.edu

Abstract

We investigate the problem of identifying individual link metrics in a communication network from end-to-end path measurements, under the assumption that link metrics are additive and constant. To uniquely identify the link metrics, the number of linearly independent measurement paths must equal the number of links. Our contribution is to characterize this condition in terms of the network topology and the number/placement of monitors, under the constraint that measurement paths must be cycle-free. Our main results are: (i) it is generally impossible to identify all the link metrics by using two monitors; (ii) nevertheless, metrics of all the interior links not incident to any monitor are identifiable by two monitors if the topology satisfies a set of necessary and sufficient connectivity conditions; (iii) these conditions naturally extend to a necessary and sufficient condition for identifying all the link metrics using three or more monitors. We show that these conditions not only allow efficient identifiability tests, but also enable an efficient algorithm to place the minimum number of monitors in order to identify all link metrics. Our evaluations on both random and real topologies show that the proposed algorithm achieves identifiability using a much smaller number of monitors than a baseline solution.

Categories and Subject Descriptors

C.2.3 [**Computer-communication Networks**]: Network Operations—*Network monitoring*; G.2.2 [**Discrete Mathematics**]: Graph Theory—*Network problems*

General Terms

Algorithms, Measurement, Performance, Theory

Keywords

Network Tomography; Linear Algebraic Model; Identifiability Condition; Monitor Placement

1. 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 along selected paths from a subset of nodes with monitoring capabilities, hereafter referred to as *monitors*.

Depending on the granularity of observations, external approaches can be classified as *hop-by-hop* approaches or *end-to-end* approaches. The former rely on special diagnostic tools such as *traceroute*, *pathchar* [2], and *Network Characterization Service (NCS)* [3] to reveal fine-grained performance metrics of individual links by sending active probes. Traceroute reports delay for each hop on the probed path by gradually increasing the time-to-live (TTL) field of probing packets. Its refinement, pathchar, returns hop-by-hop capacities, delays, and loss rates. A later advancement, NCS, also returns available capacities on each link. While providing fine-grained information, the above tools require that Internet Control Message Protocol (ICMP) be supported at each internal node. Even then, they suffer inaccuracies caused by asymmetry in routes and different priorities of ICMP and data packets. Moreover, these tools can generate a large number of probing packets, causing extra load

and, potentially, congestion. In risk-sensitive applications, security policies may even block hop-by-hop measurements.

Alternatively, the end-to-end approach provides a solution that does not rely on the cooperation of internal network elements or the equal treatment of control/data packets. It relies on end-to-end performance metrics (e.g., end-to-end delays) experienced by data packets to solve for the corresponding hop-by-hop metrics using *network tomography.* Network tomography [4] refers to the methodology of inferring internal network characteristics through end-to-end measurements. Without requiring special cooperation from internal nodes, network tomography can utilize measurements from data packets to obtain path-level information [5], thus reducing the need for active probes.

In many cases, link metrics are *additive*, i.e., the combined metric over multiple links is the sum of individual link metrics. For instance, delays are additive, while a multiplicative metric (e.g., packet delivery ratio) can be expressed in an additive form by using the $\log(\cdot)$ function. For additive metrics, we can model the problem as that of solving a system of linear equations, where the unknown variables are the link metrics, and the known constants are the end-to-end path measurements, each equal to the sum of the corresponding link metrics along a path. Thus, network tomography essentially solves this linear system of equations.

Existing work on network tomography emphasizes extracting as much information about link metrics as possible from available measurements. However, past experience shows that it is frequently impossible to uniquely identify all link metrics from path measurements [6–8]. For example, if two links (not necessarily adjacent) always appear together in measurement paths, then we can at most identify their sum metric but not the individual metrics. Generally, many measurement paths are *linearly dependent* in that some paths are linear combinations of the rest, and hence their measurements do not provide new information. From the perspective of linear algebra, link metrics are uniquely identifiable if and only if the number of *linearly independent* measurement paths equals the number of links. There is, however, a lack of basic understanding of the topological conditions that ensure identifiability, even in the simplified scenario of constant link metrics.

In this paper, we consider two closely related fundamental problems: (i) Under what conditions can one uniquely identify all link metrics from end-to-end measurements; (ii) Given an arbitrary network topology, how can one place monitors to satisfy the above conditions using the minimum number of monitors? We study both problems in the context of *controllable, cycle-free* measurements, i.e., monitors can direct measurement packets to selected paths as long as they do not contain cycles. Such routing is generally supported for networks under common administration (e.g., single-ISP networks), or overlay networks formed by overlay nodes and physical paths between these nodes (modeled as overlay links), where network tomography can be applied to accurately monitor performance experienced by data traffic while reducing measurement overhead. Moreover, these assumptions capture capabilities of a new generation of networks performing *Software-Defined Networking (SDN)* [9], where monitors, in cooperation with SDN controller, can dictate paths of measurement packets in the route setup phase, while the cycle-free constraint precludes formation of endless cycles in the data forwarding phase. We assume that all link metrics are additive and constant. Our "constant" link metric refers to one that either changes slowly relative to the measurement process, or that is a statistical characteristic (e.g., mean, variance) of the link that stays constant over time[1].

Although the answer to the first question is straightforward in linear algebra (the number of linearly independent paths equals the number of links), a useful answer should be expressed in terms of externally verifiable network properties such as the network topology and the number/placement of monitors. To this end, we establish necessary and sufficient conditions on network topology and monitor placement for uniquely identifying link metrics from controllable, cycle-free measurements between the monitors. We further develop an efficient algorithm that leverages the established conditions to place monitors in a given network such that all link metrics can be uniquely identified by using the minimum number of monitors.

1.1 Further Discussions on Related Work

Based on the model of link metrics, existing work can be broadly classified as statistical and algebraic approaches. Statistical approaches model link metrics as random variables with (partially) unknown probability distributions, and apply various parametric/nonparametric techniques to estimate the link metric distributions from realizations of path metrics [1,10,11]. Algebraic approaches consider link metrics as unknown constants, and use linear algebraic techniques to compute link metrics from cumulative path metrics [6,7].

With link metrics modeled as random variables, multicast, if supported, can be exploited as a measurement method with broad coverage and low overhead [12,13]. Sub-trees and unicast are employed in [5,14] as alternatives, due to the inflexibility of multicasting to all receivers. Employing multicast, [5,15] derive the necessary and sufficient conditions on the multicast tree for identifying all link metric distributions. If most links do not exhibit severe losses or delays, [10] proposes algorithms to identify the worst performing links. A novel approach proposed in [8] employs the Fourier transform of the observable path metric distributions to estimate the unobservable link metric distributions. All the above methods implicitly assume the links to be identifiable, and the multicast-based methods require multiple monitors to participate in the measurement process. In contrast, we assume unicast measurements and focus on establishing topological conditions for identifying all link metrics by using the minimum number of monitors.

For constant link metrics, [7] shows that it is challenging to solve the inverse problem due to the presence of linearly dependent paths. When all but k link metrics are *zero*, compressive sensing techniques are used to identify the k non-zero link metrics [16,17]. If all link metrics are *binary* (normal/failed), [18] proves that the network must be $(k+2)$-edge-connected to identify up to k failed links by using one monitor measuring cycles. For arbitrary valued link metrics, few positive results are known. If the network is directed (links have different metrics in different directions), [11] proves that not all link metrics are identifiable unless every non-isolated node is a monitor. Even if every

---

[1]In this case, end-to-end measurements are also statistical characteristics, e.g., path mean/variance. In the case of variance, we also need the independence between link qualities to make the metric additive.

node is a monitor, unique link identification is still impossible if measurement paths are constrained to cycles [6]. If the network is undirected (links have equal metrics in both directions), [19] derives the first necessary and sufficient conditions on the network topology for identifying all link metrics, given that monitors can measure cycles or paths possibly containing cycles. A similar study in [20] characterizes the minimum number of measurements needed to identify a broader set of link metrics (including both additive and nonadditive metrics), under the stronger assumption that measurement paths can contain repeated links. Since routing along cycles is typically prohibited in real networks, it remains open as to what the conditions become if only cycle-free paths can be measured. In this regard, we investigate the fundamental relationships between link identifiability, network topology, and the number/placement of monitors. Once identifiability is confirmed, a follow-up problem is to construct linearly independent paths between monitors to perform measurements, for which we have developed an efficient path construction algorithm in [21].

Little is known when it comes to placing monitors to ensure network identifiability. In [22,23], the problem of placing the minimum number of monitors to identify all link metrics under *uncontrollable* routing is proved to be NP-hard, and the NP-hardness persists even if a subset of nodes have control over their local routing policy [24]. In contrast, we show in this paper that under *controllable* cycle-free routing, this problem can be solved optimally in linear time.

1.2 Summary of Contributions

We study here for the first time the fundamental topological conditions for identifying additive link metrics by using end-to-end measurements on cycle-free paths. Our contributions are four-fold:

1) We prove that it is generally impossible to identify all link metrics by using only two monitors, irrespective of the network topology and the placement of monitors.

2) We establish necessary and sufficient conditions for identifying the metrics of all *interior links* (links not incident to any monitor) by using two monitors: the network graph is (i) 2-edge-connected after removing any interior link and (ii) 3-vertex-connected after adding a direct link between the two monitors. These conditions are shown to be verifiable in $O(|L|(|V| + |L|))$ time, where $|V|$ is the number of nodes and $|L|$ is the number of links.

3) We transform the above result into a necessary and sufficient condition for identifying all link metrics using κ ($\kappa \geq 3$) monitors by embedding the network graph in an extended graph, with two virtual monitors connected to all the real monitors. This condition can be verified in $O(|V|+|L|)$ time.

4) We propose an algorithm to place monitors in an arbitrary network in $O(|V| + |L|)$ time, which guarantees the identifiability of all link metrics by using the minimum number of monitors. Our evaluations on both randomly-generated and real topologies show that the proposed algorithm requires a substantially smaller number of monitors than a baseline solution.

We note that our goal is to characterize identifiable scenarios for additive link metrics, and the identification of non-additive link metrics (e.g., bit error rates) is beyond the scope of this paper.

The rest of the paper is organized as follows. Section 2 formulates the problem. Section 3 summarizes our main results. Sections 4–5 present identifiability conditions for the case of two monitors, and Section 6 addresses the case of three or more monitors. Algorithms for testing network identifiability and deploying monitors are presented in Section 7. Finally, Section 8 concludes the paper.

Table 1: Notations in Graph Theory

Symbol	Meaning
$V(\mathcal{G})$, $L(\mathcal{G})$	set of nodes/links in graph $\mathcal{G}$
$|\mathcal{G}|$	degree of graph $\mathcal{G}$: $|\mathcal{G}| = |V(\mathcal{G})|$ (number of nodes)
$||\mathcal{G}||$	order of graph $\mathcal{G}$: $||\mathcal{G}|| = |L(\mathcal{G})|$ (number of links)
$\mathcal{H}$	interior graph (see Definition 1)
$L(v)$	set of links incident to node v
$\mathcal{G} - l$	delete a link: $\mathcal{G} - l = (V(\mathcal{G}), L(\mathcal{G}) \setminus \{l\})$, where $l \in L(\mathcal{G})$ and "$\setminus$" is setminus
$\mathcal{G} + l$	add a link: $\mathcal{G} + l = (V(\mathcal{G}), L(\mathcal{G}) \cup \{l\})$, where the end-points of link l are in $V(\mathcal{G})$
$\mathcal{G} - v$	delete a node: $\mathcal{G} - v = (V(\mathcal{G}) \setminus \{v\}, L(\mathcal{G}) \setminus L(v))$, where $v \in V(\mathcal{G})$
$\mathcal{G}_s + v$	add a node: $\mathcal{G}_s + v = (V(\mathcal{G}_s) \cup \{v\}, L(\mathcal{G}_s) \cup L_v)$, where $\mathcal{G}_s$ is a sub-graph of $\mathcal{G}$, $v \in V(\mathcal{G}) \setminus V(\mathcal{G}_s)$, and L_v is the set of all links between v and nodes in $V(\mathcal{G}_s)$
$\mathcal{G} \setminus \mathcal{G}'$	From $\mathcal{G}$, delete all common nodes with $\mathcal{G}'$ and their incident links
$\mathcal{G} \cap \mathcal{G}'$	intersection of graphs: $\mathcal{G} \cap \mathcal{G}' = (V(\mathcal{G}) \cap V(\mathcal{G}'), L(\mathcal{G}) \cap L(\mathcal{G}'))$
$\mathcal{G} \cup \mathcal{G}'$	union of graphs: $\mathcal{G} \cup \mathcal{G}' = (V(\mathcal{G}) \cup V(\mathcal{G}'), L(\mathcal{G}) \cup L(\mathcal{G}'))$
$\mathcal{P}(v_0, v_k)$	simple path connecting nodes v_0 and v_k, defined as a special graph with $V(\mathcal{P}) = \{v_0, \ldots, v_k\}$ and $L(\mathcal{P}) = \{v_0 v_1, v_1 v_2, \ldots, v_{k-1} v_k\}$
$\mathcal{C}$	cycle: if $(v_0, \ldots, v_k)$ ($k \geq 2$) is a sequence of nodes on a simple path $\mathcal{P}$, then $\mathcal{C} = \mathcal{P} + v_k v_0$ is a *cycle*
$\mathcal{F}$	a non-separating cycle (see Definition 4)
m_i	$m_i \in V(\mathcal{G})$ is the i-th monitor in $\mathcal{G}$
W_l, $W_{\mathcal{P}}$	metric on link l and sum metric on path $\mathcal{P}$

2. PROBLEM FORMULATION

2.1 Models and Assumptions

We assume that the network topology is known and model it as an undirected graph[2] $\mathcal{G} = (V, L)$, where V and L are the sets of nodes and links, respectively. Note that graph $\mathcal{G}$ can represent a logical topology where each link in $\mathcal{G}$ corresponds to a combination of physical links/nodes, e.g., in an overlay network. 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 u and v by uv; links uv and vu are assumed to have the same metric. Certain nodes in V are monitors and can initiate/collect measurements. 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. Last, no link metrics in $\mathcal{G}$ are known prior to taking measurements.
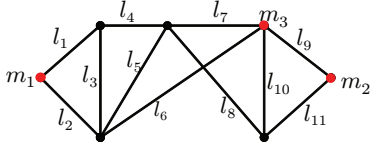
---

[2]We use the terms *network* and *graph* interchangeably.

Figure 1: Sample network with three monitors: m_1, m_2, and m_3.

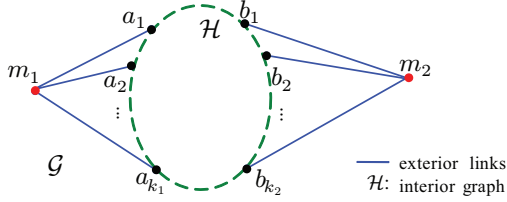**Figure 2: Reorganizing graph $\mathcal{G}$ into exterior links and interior graph ($\{a_i\}_{i=1}^{k_1}$ and $\{b_j\}_{j=1}^{k_2}$ may have overlap).**

Table 1 summarizes all graph theory notations used in this paper (following the convention of [25]).

Let $n := |L|$ denote the number of links in $\mathcal{G}$, $\{l_i\}_{i=1}^n$ the set of links in $\mathcal{G}$, $\mathbf{w} = (W_{l_1}, \ldots, W_{l_n})^T$ the column vector of all link metrics, and $\mathbf{c} = (W_{\mathcal{P}_1}, \ldots, W_{\mathcal{P}_\gamma})^T$ the column vector of all available path measurements, where γ is the number of measurement paths and $W_{\mathcal{P}_i}$ is the sum of link metrics along measurement path $\mathcal{P}_i$. We assume that monitors can control the routing of measurement packets 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). The path measurements are expressed in terms of the unknown link metrics through the following linear system:

$$\mathbf{R}\mathbf{w} = \mathbf{c}, \tag{1}$$

where $\mathbf{R} = (R_{ij})$ is a $\gamma \times n$ *measurement matrix*, with each entry $R_{ij} \in \{0, 1\}$ denoting whether link l_j is present on path $\mathcal{P}_i$. The network tomography problem is to invert this linear system to solve for $\mathbf{w}$ given $\mathbf{R}$ and $\mathbf{c}$.

A link is *identifiable* if the associated link metric can be uniquely determined from path measurements; network $\mathcal{G}$ is identifiable if all links in $\mathcal{G}$ are identifiable. Otherwise, the link or the network is said to be *unidentifiable*. Given the above linear system, $\mathcal{G}$ is identifiable if and only if $\mathbf{R}$ in (1) has full column rank, i.e., rank$(\mathbf{R}) = n$. In other words, to uniquely determine $\mathbf{w}$, there must be n linearly independent simple paths between monitors.

2.2 Objective

Given a network topology $\mathcal{G}$, the first objective of this paper is to derive necessary and sufficient conditions for identifying all link metrics in $\mathcal{G}$ (or certain subgraphs of $\mathcal{G}$) under a given placement of κ ($\kappa \geq 2$) monitors by solving the linear system (1). Our second objective is to compute a placement of the minimum number of monitors in $\mathcal{G}$ that enables the identification of all link metrics.

2.3 Illustrative Example

Fig. 1 displays a sample network with three monitors (m_1–m_3) and eleven links (links l_1–$l_1 1$). To identify all eleven link metrics, eleven end-to-end paths (one $m_1 \rightarrow m_2$ path, seven

$m_1 \rightarrow m_3$ paths and three $m_3 \rightarrow m_2$ paths) are constructed to form the measurement matrix $\mathbf{R}$:

$$
\begin{array}{l}
m_1 \rightarrow m_2 : \\
\quad l_1\ l_4\ l_8\ l_{11} \\
m_1 \rightarrow m_3 : \\
\quad l_1\ l_4\ l_7 \\
\quad l_2\ l_6 \\
\quad l_2\ l_3\ l_4\ l_7 \\
\quad l_2\ l_5\ l_8\ l_{10} \\
\quad l_2\ l_5\ l_7 \\
\quad l_1\ l_3\ l_6 \\
\quad l_1\ l_4\ l_5\ l_6 \\
m_3 \rightarrow m_2 : \\
\quad l_9 \\
\quad l_{10}\ l_{11} \\
\quad l_6\ l_5\ l_8\ l_{11}
\end{array}
\Rightarrow \mathbf{R} =
\begin{pmatrix}
1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\
1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1
\end{pmatrix},
$$

where $R_{ij} = 1$ if and only if link l_j is on path $\mathcal{P}_i$. Then we have $\mathbf{R}\mathbf{w} = \mathbf{c}$, where $\mathbf{c}$ is the vector of end-to-end measurements taken at the destination monitors. In this example, $\mathbf{R}$ is invertible, and thus $\mathbf{w}$ can be uniquely identified, i.e., $\mathbf{w} = \mathbf{R}^{-1}\mathbf{c}$. In Fig. 1, other simple paths can be measured as well, although they do not provide further information since the measurement matrix already reaches full rank. However, if we remove a monitor, say m_3, then it can be verified that the remaining paths can no longer form an invertible measurement matrix. Note that a path such as $l_2 l_5 l_4 l_3 l_6 l_9$ cannot be measured because it contains a cycle.

3. MAIN RESULTS

Our main contributions are a set of necessary and sufficient conditions for network identification that are explicitly expressed in terms of network topology and the number/placement of monitors (proofs of theorems can be found in Sections 4–6). To begin, we first establish a negative result that no matter where we place the monitors, we cannot identify all link metrics using only two monitors.

THEOREM 3.1. *For any given network topology $\mathcal{G}$ with $n \geq 2$ (n is the number of links), $\mathcal{G}$ is unidentifiable with two monitors, irrespective of their placement.*

Second, we examine the two-monitor case in more detail and discover that the unidentifiability issue only applies to a small subset of links, and that the majority of links can be identified under certain conditions. Specifically, given two monitors m_1 and m_2, we can reorganize $\mathcal{G}$ into two parts[3] as illustrated in Fig. 2.

DEFINITION 1.
1) *The* interior graph $\mathcal{H}$ *of $\mathcal{G}$ is the sub-graph obtained by removing the monitors and their incident links, i.e., $\mathcal{H} := (V \setminus M, L \setminus L_M)$ for $M = \{m_1, m_2\}$ and $L_M = L(m_1) \cup L(m_2)$.*
2) *We refer to links incident to monitors, i.e., $L(m_1) \cup L(m_2)$, as* exterior links, *and the remaining links as* interior links.

We show that the exterior links can never be identified using two monitors (see Corollary 4.1), but the interior links can be identified under the following conditions.

---

[3]An area with a dashed border denotes a sub-graph (nodes/links on the dashed border are also part of the sub-graph, e.g., $\{a_i\}_{i=1}^{k_1}$ and $\{b_j\}_{j=1}^{k_2}$ in Fig. 2 are part of $\mathcal{H}$), and a solid line denotes a link/path/cycle.
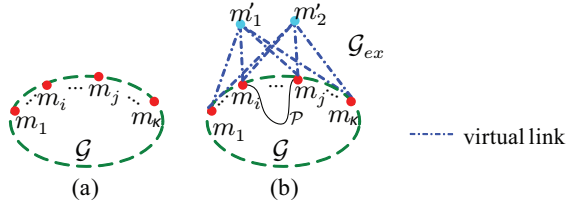
Figure 3: (a) $\mathcal{G}$ with κ ($\kappa \geq 3$) monitors; (b) $\mathcal{G}_{ex}$ with two virtual monitors.

THEOREM 3.2. *Assume that the interior graph $\mathcal{H}$ (with $||\mathcal{H}|| \geq 1$) of $\mathcal{G}$ under a given placement of two monitors (m_1 and m_2) is connected and direct link $m_1 m_2$ (incident to m_1 and m_2) does not exist in $L(\mathcal{G})$. The necessary and sufficient conditions for identifying all link metrics in $\mathcal{H}$ are:*

① *$\mathcal{G} - l$ is 2-edge-connected for every interior link l in $\mathcal{H}$;*

② *$\mathcal{G} + m_1 m_2$ is 3-vertex-connected.*

Third, we show that the above conditions can be naturally extended to a necessary and sufficient condition for identifying *all* link metrics in $\mathcal{G}$ using three or more monitors. This condition is based on an *extended graph* $\mathcal{G}_{ex}$ obtained from $\mathcal{G}$ as follows. As illustrated in Fig. 3, given a graph $\mathcal{G}$ with κ monitors, its extended graph $\mathcal{G}_{ex}$ is obtained by adding two *virtual monitors* m'_1 and m'_2, and 2κ *virtual links* between each pair of virtual-actual monitors. The identifiability of $\mathcal{G}$ is characterized by a simple condition on $\mathcal{G}_{ex}$ as follows.

THEOREM 3.3. *Assume that κ ($\kappa \geq 3$) monitors are used to measure simple paths. The necessary and sufficient condition on the network topology $\mathcal{G}$ for identifying all link metrics in $\mathcal{G}$ is that the associated extended graph $\mathcal{G}_{ex}$ be 3-vertex-connected.*

Finally, we develop efficient algorithms that can: (i) test whether a given placement of monitors can identify all link metrics, and (ii) compute a placement of the minimum number of monitors needed to identify all link metrics (see Section 7). Both algorithms run in linear time w.r.t. network size (i.e., $O(|V(\mathcal{G})| + |L(\mathcal{G})|)$) and hence are suitable for large networks.

4. UNIDENTIFIABILITY WITH TWO MONITORS

At least *two* monitors are required to identify link metrics through monitoring simple paths. In this section, we investigate if two monitors suffice to identify all link metrics in the network. Suppose that two distinct nodes are selected to serve as monitors. Each measurement starts at one monitor and terminates at the other via a controllable simple path. The termination node then reports the end-to-end metric, which becomes an entry in the measurement vector **c**. From the perspective of graph theory, such a network can be represented as $\mathcal{G} = (\{m_1, m_2, v_0, \ldots, v_k\}, L)$, where m_1 and m_2 are the monitors, $\{v_0, \ldots, v_k\}$ are the non-monitors, and $|L| = n$. Let $m_1 m_2$ be a direct link between m_1 and m_2 (if it exists). Since $m_1 m_2$ can be easily identified through a one-hop measurement, we assume without loss of generality that $m_1 m_2 \notin L(\mathcal{G})$ (i.e., there is no direct link) in Sections 4 and 5, where only two monitors are used.

4.1 Proof of Theorem 3.1

Any $\mathcal{G}$ with $||\mathcal{G}|| \geq 2$ can be reorganized as in[4] Fig. 2. Let $N(v)$ denote the set of neighboring nodes of node v. We define $A := \{a_1, a_2, \ldots, a_{k_1}\} = N(m_1)$ and $B := \{b_1, b_2, \ldots, b_{k_2}\} = N(m_2)$ to be the sets of neighbors of m_1 and m_2, respectively, where $k_1 := |A|$, $k_2 := |B|$ and A, B can overlap ($m_1, m_2 \notin A \cup B$).

Assuming that $\mathcal{H}$ is connected and all link metrics in $\mathcal{H}$ are known, we can reduce any equation associated with a simple path $\mathcal{P}$ between m_1 and m_2 to the form (see Table 1 for notations):

$$W_{m_1 a_i} + W_{b_j m_2} = \phi_{ij} \tag{2}$$

for some $a_i \in A$ and $b_j \in B$. This is obtained by rewriting the original equation $W_{m_1 a_i} + W_{\mathcal{P}'_{ij}} + W_{b_j m_2} = W_{\mathcal{P}_{ij}}$ ($\mathcal{P}'_{ij}$ is the segment of $\mathcal{P}_{ij}$ in $\mathcal{H}$) to place the unknowns on the left-hand side, and setting $\phi_{ij} := W_{\mathcal{P}_{ij}} - W_{\mathcal{P}'_{ij}}$. Thus, we obtain $k_1 \times k_2$ equations from all of the simple paths between m_1 and m_2, each corresponding to the sum of the metrics of one link incident to m_1 and one link incident to m_2. The corresponding reduced measurement matrix is (each column corresponding to an unknown link metric):

$$
\mathbf{R} = \begin{pmatrix} & & \\ & & \end{pmatrix}, \tag{3}
$$

where the blank entries are zero. We apply the following linear transformations to $\mathbf{R}$. For each $q = 1, \ldots, k_1 - 1$ and $i = 2, \ldots, k_2$, replace $\text{row}(qk_2 + i)$ by $\text{row}(qk_2 + i) - \text{row}(i) - \text{row}(qk_2 + 1) + \text{row}(1)$; it can be verified that the result is a row of zeros. Ignoring rows of zeros, $\mathbf{R}$ transforms into

$$
\mathbf{R}' = \begin{pmatrix} & & \\ & & \end{pmatrix}, \tag{4}
$$

where the rows are linearly independent, and the number of rows equals $k_1 + k_2 - 1$. This transformation preserves the rank of $\mathbf{R}$ because $\mathbf{R}$ can be reconstructed from $\mathbf{R}'$ as follows: the first k_2 rows in $\mathbf{R}$ and $\mathbf{R}'$ are identical; $\text{row}(qk_2 + 1)$ in $\mathbf{R}$ equals $\text{row}(k_2 + q)$ in $\mathbf{R}'$ ($q = 1, \ldots, k_1 - 1$); $\text{row}(qk_2 + i)$

---

[4]If certain links in $\mathcal{G}$ cannot be included in any possible paths constructed from m_1 to m_2 in Fig. 2, then these links are unidentifiable, resulting in a disconnected or one-edge-connected interior graph $\mathcal{H}$.

in $\mathbf{R}$ equals $\mathrm{row}(i) + \mathrm{row}(k_2 + q) - \mathrm{row}(1)$ in $\mathbf{R}'$ ($q = 1, \ldots,$ $k_1 - 1$, $i = 2, \ldots, k_2$). Hence, the rank of $\mathbf{R}$ equals $k_1 + k_2 - 1$. When $\mathcal{H}$ is not connected, some rows in $\mathbf{R}$ may not exist because there is no simple path connecting the corresponding nodes in A and B, and the rank of $\mathbf{R}$ may be even smaller. Since there are $k_1 + k_2$ unknown variables $(W_{m_1 a_i})_{i=1}^{k_1}$ and $(W_{b_j m_2})_{j=1}^{k_2}$, they cannot be uniquely determined even if all link metrics in $\mathcal{H}$ are already known. Therefore, $\mathcal{G}$ with $||\mathcal{G}|| \geq 2$ is unidentifiable by using two monitors. □

In fact, we can show that none of the exterior links is identifiable because metric of any one exterior link will uniquely determine metrics of the rest; see proof in [26].

COROLLARY 4.1. *None of the exterior links (except $m_1 m_2$) can be identified with two monitors.*

4.2 Discussions on Paths with Cycles

At the end of Section III-B in [19], the authors raise the question whether or not monitoring non-simple paths (i.e., paths that may contain cycles) between two monitors suffices to identify all link metrics in the network[5]. According to Corollary 4.1, the exterior links cannot be identified even if all the interior link metrics are known; allowing cycles in the interior graph $\mathcal{H}$ provides no additional information regarding the exterior links. Consequently, the answer to that question in [19] is that monitoring (simple or non-simple) paths between two monitors is *not* sufficient to identify all link metrics.

5. IDENTIFIABILITY OF INTERIOR LINKS WITH TWO MONITORS

Network administrators are more interested in using end-to-end measurements to infer the qualities of links that are at least *one-hop away*. Therefore, in this section, we only focus on the interior graph $\mathcal{H}$ and derive necessary and sufficient conditions on the network topology $\mathcal{G}$ for identifying all links in $\mathcal{H}$ using two monitors (m_1 and m_2), under the earlier assumption that no link metrics (including those of links incident to m_1 and m_2) are known ahead of time.

Before going into details, we first point out that it is sufficient to solve the case in which $\mathcal{H}$ is a connected graph. This is because if $\mathcal{H}$ consists of K_H ($K_H \geq 2$) connected components $\mathcal{H}_i$ ($i = 1, \ldots, K_H$), we can decompose the entire graph $\mathcal{G}$ into sub-graphs $\mathcal{G}_i := \mathcal{H}_i + m_1 + m_2$, with $\mathcal{G} = \cup_{i=1}^{K_H} \mathcal{G}_i$ (see the definition of graph union in Table 1). Since none of the $m_1 \to m_2$ simple paths in $\mathcal{G}_i$ can traverse $\mathcal{G}_j$ ($i \neq j$), the identification of links within different $\mathcal{G}_i$'s is mutually independent. Therefore, in the rest of this section, we assume $\mathcal{H}$ to be connected with $||\mathcal{H}|| \geq 1$. Our result can be applied to each $\mathcal{G}_i$ separately when $\mathcal{H}$ is disconnected.

5.1 Proof of Theorem 3.2: Necessary Part

Suppose all links in $\mathcal{H}$ are identifiable. We prove the necessity of Conditions ①–② in Theorem 3.2 by contradiction.

a) Let $l_1 \in L(\mathcal{H})$ be an arbitrary interior link. If $\mathcal{G} - l_1$ is disconnected, then l_1 is a bridge[6] in $\mathcal{G}$ (shown in Fig. 4(a)). If $\mathcal{G}_1$ and $\mathcal{G}_2$ each contains a monitor, then l_1 is unidentifiable by Lemma A.1 (Appendix). If m_1 and m_2 are both in $\mathcal{G}_1$

---

[5]We restrict the paths from containing repeated monitors to exclude measurement of cycles, as is already solved in [19].
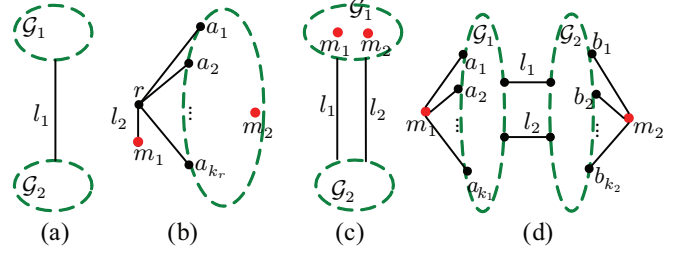[6]A link whose removal will disconnect the graph is a bridge [25].

Figure 4: Illustration of Condition ①, where $\{l_1, l_2\}$ is an edge cut in (c) and (d).

(or $\mathcal{G}_2$), then l_1 cannot be included in any measurement path (otherwise, l_1 will be used more than once), and is thus unidentifiable. Both cases contradict the assumption that all interior links are identifiable.

b) Suppose there is a bridge l_2 in $\mathcal{G} - l_1$. If l_2 is an exterior link, as shown in Fig. 4(b), then by Lemma A.1 (Appendix), its adjacent interior links $ra_i \in L(\mathcal{H})$ are unidentifiable, contradicting the assumption that all interior links are identifiable. Thus, l_2 must be an interior link. Since by *a)*, an interior link cannot be a bridge in $\mathcal{G}$, $\{l_1, l_2\}$ must be an edge cut as shown in Fig. 4(c) and 4(d). If both m_1 and m_2 are in $\mathcal{G}_1$ as in Fig. 4(c), then all $m_1 \to m_2$ paths traversing l_1 must traverse l_2 as well. Thus we can at most identify $W_{l_1} + W_{l_2}$, but not W_{l_1} and W_{l_2} individually. If m_1 is in $\mathcal{G}_1$ and m_2 is in $\mathcal{G}_2$ as in Fig. 4(d), then all $m_1 \to m_2$ paths must traverse either l_1 or l_2, but not both. Assuming that $\mathcal{G}_1$ and $\mathcal{G}_2$ in Fig. 4(d) are connected and all link metrics in them are known, then the resulting measurement matrix $\mathbf{R}_1$ is similar to (3), except that each row in $\mathbf{R}_1$ has a new entry associated with W_{l_1} or W_{l_2}:

$$\mathbf{R}_1 = \left(\begin{array}{c|c|c} \overset{\text{Exterior}}{\underset{\text{Links}}{}} & W_{l_1} & W_{l_2} \\ \mathbf{R} & \mathbf{e}_1 & \\ \mathbf{R} & & \mathbf{e}_1 \end{array} \right) .$$

Here blank entries correspond to zeroes, $\mathbf{R}$ is given by (3), and $\mathbf{e}_1$ is a ($k_1 \times k_2$)-element column vector of all ones. Let $\mathbf{R}_2$ and $\mathbf{R}_3$ denote the first and last $k_1 \times k_2$ rows in $\mathbf{R}_1$, respectively. Then applying the same linear transformations as applied to (3) to both $\mathbf{R}_2$ and $\mathbf{R}_3$, we can transform $\mathbf{R}_1$ into $\mathbf{R}_1'$ (ignoring rows of zeros):

$$\mathbf{R}_1' = \left(\begin{array}{c|c|c} \overset{\text{Exterior}}{\underset{\text{Links}}{}} & W_{l_1} & W_{l_2} \\ \mathbf{R}' & \mathbf{e}_2 & \\ \mathbf{R}' & & \mathbf{e}_2 \end{array} \right) ,$$

where $\mathbf{R}'$ is given by (4) and $\mathbf{e}_2$ is a ($k_1 + k_2 - 1$)-element column vector of all ones. In $\mathbf{R}_1'$, for each $q = k_1 + k_2, \ldots,$ $2k_1 + 2k_2 - 2$, replace $\mathrm{row}(q)$ with $\mathrm{row}(q)$-$\mathrm{row}(q - k_1 - k_2 + 1)$. Ignoring duplicate rows, $\mathbf{R}_1'$ transforms into

$$\mathbf{R}_1'' = \left(\begin{array}{c|c|c} \overset{\text{Exterior}}{\underset{\text{Links}}{}} & W_{l_1} & W_{l_2} \\ \mathbf{R}' & \begin{matrix} \mathbf{e}_2 \\ -1 \end{matrix} & \begin{matrix} \\ 1 \end{matrix} \end{array} \right) ,$$

where all rows are linearly independent. Since any subset of k ($k \geq 1$) equations in $\mathbf{R}_1''$ contains more than k unknown variables, none of these variables can be identified. When $\mathcal{G}_1$ and $\mathcal{G}_2$ are not connected, the rank of $\mathbf{R}_1''$ can be even smaller. Thus, W_{l_1} and W_{l_2} are unidentifiable, contradicting the assumption that all the interior links are identifiable.
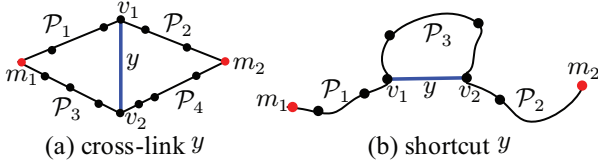
(a) cross-link y (b) shortcut y

Figure 5: Two types of identifiable links in $\mathcal{H}$.

Based on *a)* and *b)*, we see that $\mathcal{G} - l_1$ must be 2-edge-connected for any $l_1 \in L(\mathcal{H})$ (i.e., Condition ① holds).

c) We can also prove Condition ② by contradiction; see Proposition A.2 (Appendix). □

5.2 Proof of Theorem 3.2: Sufficient Part

Given Conditions ①–②, we need to show that all links in $\mathcal{H}$ are identifiable. We first introduce two types of identifiable links. The argument then consists of showing that every interior link belongs to one of these two types.

5.2.1 Cross-link, Shortcut, and Non-separating Cycle

We begin by introducing a few notions essential to our proof.

The first notion is a cross-link that connects nodes on two simple paths between the monitors.

DEFINITION 2. *As illustrated in Fig. 5(a), link y is a cross-link if $\exists$ four $m_1 \to m_2$ paths $\mathcal{P}_A$, $\mathcal{P}_B$, $\mathcal{P}_C$, and $\mathcal{P}_D$ formed from simple paths $\mathcal{P}_1, \ldots, \mathcal{P}_4$ by:*

$$\begin{cases} \mathcal{P}_A = \mathcal{P}_1 \cup \mathcal{P}_2 \\ \mathcal{P}_B = \mathcal{P}_3 \cup \mathcal{P}_4 \end{cases}, \quad \begin{cases} \mathcal{P}_C = \mathcal{P}_1 \cup y \cup \mathcal{P}_4 \\ \mathcal{P}_D = \mathcal{P}_3 \cup y \cup \mathcal{P}_2 \end{cases}, \quad (5)$$

such that

$$\begin{cases} |\mathcal{P}_1 \cap \mathcal{P}_2| = 1 \\ |\mathcal{P}_3 \cap \mathcal{P}_4| = 1 \end{cases}, \begin{cases} |\mathcal{P}_2 \cap \mathcal{P}_3| = 0 \\ |\mathcal{P}_1 \cap \mathcal{P}_4| = 0 \end{cases}. \quad (6)$$

See Table 1 for definitions of graph union/intersection and $|\cdot|$; note that paths are also graphs. The constraints in (6) are used to ensure that $\mathcal{P}_A$–$\mathcal{P}_D$ are *simple* paths, e.g., $|\mathcal{P}_1 \cap \mathcal{P}_2| = 1$ ($\mathcal{P}_1$ and $\mathcal{P}_2$ have no common node other than v_1) ensures that no cycles exist in $\mathcal{P}_A$. However, this does not require $\mathcal{P}_1$–$\mathcal{P}_4$ to be node disjoint, e.g., $\mathcal{P}_1$ and $\mathcal{P}_3$ can have common nodes. A cross-link y can then be identified by

$$W_y = \frac{1}{2}(W_{\mathcal{P}_C} + W_{\mathcal{P}_D} - W_{\mathcal{P}_A} - W_{\mathcal{P}_B}). \quad (7)$$

The second notion is a shortcut that connects the endpoints of a simple path whose metric is known.

DEFINITION 3. *As illustrated in Fig. 5(b), link y is a shortcut if $\exists$ a simple path $\mathcal{P}_3$ whose metric has been identified such that the following $m_1 \to m_2$ simple paths can be formed:*

$$\mathcal{P}_A = \mathcal{P}_1 \cup y \cup \mathcal{P}_2, \quad \mathcal{P}_B = \mathcal{P}_1 \cup \mathcal{P}_3 \cup \mathcal{P}_2, \quad (8)$$

satisfying $|\mathcal{P}_1 \cap \mathcal{P}_3| = 1, |\mathcal{P}_2 \cap \mathcal{P}_3| = 1$, *and* $|\mathcal{P}_1 \cap \mathcal{P}_2| = 0$.

Again, the constraints are used to guarantee that $\mathcal{P}_A$ and $\mathcal{P}_B$ are simple paths. A shortcut y can be identified by

$$W_y = W_{\mathcal{P}_A} - W_{\mathcal{P}_B} + W_{\mathcal{P}_3}. \quad (9)$$

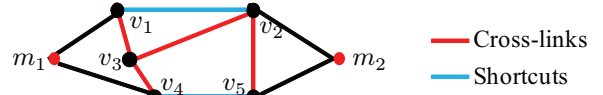The third notion is a special kind of cycle defined as follows.

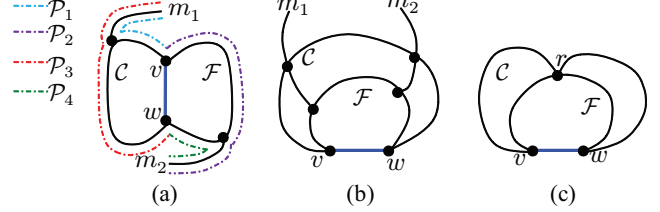**Figure 6: Sample network with identifiable interior graph.**

(a)   (b)   (c)

Figure 7: Possible cases of interior link vw: (a) Case A, (b) Case B-1, (c) Case B-2.

DEFINITION 4. *A non-separating cycle in $\mathcal{G}$, denoted by $\mathcal{F}$, is an induced sub-graph[7] such that: (i) $\mathcal{F}$ is a cycle (see definition in Table 1), and (ii) $\mathcal{F}$ does not separate any node from monitors, i.e., each connected component in $\mathcal{G} \setminus \mathcal{F}$ contains at least one monitor.*

For example, there are four non-separating cycles in Fig. 6: $v_1 v_2 v_3 v_1$, $v_4 v_3 v_2 v_5 v_4$, $m_1 v_1 v_3 v_4 m_1$, and $v_5 v_2 m_2 v_5$. Cycle $v_4 v_3 v_1 v_2 v_5 v_4$ is not a non-separating cycle as it is not induced (due to link $v_2 v_3$), neither is $v_4 m_1 v_1 v_2 v_5 v_4$ as it separates v_3 from monitors.

5.2.2 The Proof

The key to the sufficiency proof is to show that each interior link can be categorized as either a cross-link or a shortcut when the network satisfies Conditions ① and ②. The proof proceeds in three steps.

The *first step* is to show that under Conditions ① and ②, every interior link vw satisfies one of the three cases as shown in Fig. 7. Specifically, we have the following lemma.

LEMMA 5.1. *If graph $\mathcal{G}$ satisfies Conditions ① and ②, then for any interior link vw, there exists a non-separating cycle $\mathcal{F}$ with $vw \in L(\mathcal{F})$, a cycle $\mathcal{C}$ with $vw \in L(\mathcal{C})$, a simple path $\mathcal{P}_1$ connecting one monitor with a node on $\mathcal{F} - v - w$, and a simple path $\mathcal{P}_2$ connecting the other monitor with a node on $\mathcal{C} - v - w$ such that*

(a) *$\mathcal{F}$ and $\mathcal{C}$ have at most one common node other than v, w (i.e., $|V(\mathcal{F}) \cap V(\mathcal{C})| \leq 3$);*

(b) *$\mathcal{P}_1$ and $\mathcal{P}_2$ are disjoint, neither going through v nor w (i.e., $\mathcal{P}_1 \cap \mathcal{P}_2 = \emptyset$, $v, w \notin V(\mathcal{P}_1)$ and $v, w \notin V(\mathcal{P}_2)$);*

(c) *$|V(\mathcal{P}_1) \cap V(\mathcal{F})| = 1, |V(\mathcal{P}_2) \cap V(\mathcal{C})| = 1$.*

PROOF. See [26]. □

Lemma 5.1 states that there must exist two cycles sharing link vw, a non-separating cycle $\mathcal{F}$ and a (not necessarily non-separating) cycle $\mathcal{C}$ that satisfy conditions (a)–(c) in the lemma. These conditions imply three possible cases:

---

[7]An *induced sub-graph* $\mathcal{G}'$ of $\mathcal{G}$ is a sub-graph such that for any pair of vertices v and w in $\mathcal{G}'$, vw is an edge in $\mathcal{G}'$ if and only if vw is an edge in $\mathcal{G}$.

- *Case A (Fig. 7(a)):* $\mathcal{C}$ is also a non-separating cycle, $\mathcal{F}$ and $\mathcal{C}$ have no common node other than v and w, and $\mathcal{F}/\mathcal{C}$ each connects to a different monitor by a disjoint simple path;

- *Case B-1 (Fig. 7(b)):* any path from any node in $\mathcal{F} - v - w$ to monitors must have a common node with $\mathcal{C} - v - w$;

- *Case B-2 (Fig. 7(c)):* $\mathcal{F}$ and $\mathcal{C}$ must have at least one common node (denoted by r) other than v and w.

These cases are guaranteed to be complete by Lemma 5.1. We will show later that Case A links are cross-links and Case B (including B-1 and B-2) links are shortcuts.

The *second step* is to recognize that in Case A (Fig. 7(a)), we can construct four measurement paths $\mathcal{P}_A$–$\mathcal{P}_D$ by (5), using $\mathcal{P}_1$–$\mathcal{P}_4$ marked in the figure. Thus, a Case A link vw is a cross-link and thus can be identified as in (7). The challenge is that it is impossible to construct measurement paths as in (5) for the other two cases (Fig. 7(b) and 7(c)), since the conditions in (6) cannot be satisfied. This motivates us to explore the possibility of identifying W_{vw} in these two cases by (9), i.e., to investigate whether vw in Fig. 7(b) and 7(c) can be characterized as a shortcut.

The *third step* is to show that any link of Case B-1 or B-2 is indeed a shortcut. We prove this by showing that for any Case B link, we can find a detour path connecting its end-points such that all the links in this detour path are cross-links, and thus can be identified by (7). To this end, we show the following lemma.

LEMMA 5.2. *Let $m_1^*, m_2^* \in \{m_1, m_2\}$ with $m_1^* \neq m_2^*$. If $\mathcal{G}$ satisfies Conditions ① and ②, then*

(a) for any non-separating cycle in $\mathcal{G}$, there is at most one Case B link in this non-separating cycle;

(b) for any Case B link vw in the interior graph of $\mathcal{G}$, there exists a non-separating cycle $\mathcal{F}_{vw}$ with $vw \in L(\mathcal{F}_{vw})$ and $m_1, m_2 \notin V(\mathcal{F}_{vw})$. For this non-separating cycle $\mathcal{F}_{vw}$, there exist disjoint simple paths $\mathcal{P}(m_1^*, v)$ and $\mathcal{P}(m_2^*, w)$, each intersecting with $\mathcal{F}_{vw}$ only at the end-point, i.e., $V\big(\mathcal{P}(m_1^*, v) \cap \mathcal{F}_{vw}\big) = \{v\}$ and $V\big(\mathcal{P}(m_2^*, w) \cap \mathcal{F}_{vw}\big) = \{w\}$.

PROOF. See [26]. □

Lemma 5.2-(b) implies that a Case B link y must reside on a non-separating cycle $\mathcal{F}$ within the interior graph. Meanwhile, Lemma 5.2-(a) implies that all the other links on cycle $\mathcal{F}$ are cross-links, as there is at most one Case B link on each non-separating cycle. Furthermore, we show in Lemma 5.2-(b) that there exist disjoint simple paths $\mathcal{P}_1$ and $\mathcal{P}_2$ connecting the end-points of link y to different monitors, each sharing only one common node with $\mathcal{F}$ at the end-points of link y. Together, these allow us to construct two measurement paths $\mathcal{P}_A$ and $\mathcal{P}_B$ as in (8), where $\mathcal{P}_3 = \mathcal{F} - y$ consists of only cross-links whose metrics are known from (7). Thus, a Case B link y is a shortcut and can be identified by (9).

Therefore, under Conditions ① and ②, every interior link is identifiable. □

As an example, Fig. 6 displays a network satisfying Conditions ① and ②, where each interior link can be characterized as either a cross-link or a shortcut.

6. IDENTIFIABILITY WITH THREE OR MORE MONITORS

Since two monitors are not sufficient to identify all link metrics in $\mathcal{G}$, we explore in this section the case where the network contains three or more monitors.

6.1 Conversion into Two-Monitor Problem

Section 5 suggests that it is easier to identify links that are one-hop away from the monitors. This observation motivates us to construct an extended graph $\mathcal{G}_{ex}$ of $\mathcal{G}$ (see Fig. 3), so that all links of interest (actual links in $\mathcal{G}$) are at least one-hop away from virtual monitors m_1' and m_2'. This construction immediately converts the problem of identifying $\mathcal{G}$ using κ monitors to a problem of identifying the interior graph of $\mathcal{G}_{ex}$ using two monitors (again we have no prior knowledge of link metrics in $\mathcal{G}$ or $\mathcal{G}_{ex}$). Therefore, we can apply Theorem 3.2 to obtain the following result.

LEMMA 6.1. *Employing κ ($\kappa \geq 3$) monitors to measure simple paths, the necessary and sufficient condition on the network topology $\mathcal{G}$ for identifying all link metrics in $\mathcal{G}$ is that the associated extended graph $\mathcal{G}_{ex}$ has an identifiable interior graph, i.e., $\mathcal{G}_{ex}$ satisfies Conditions ① and ② in Theorem 3.2.*

PROOF. Since $\mathcal{G}$ is the interior graph of $\mathcal{G}_{ex}$, it suffices to show that the information attainable by the real monitors $m_1, \ldots, m_\kappa$ is the same as the information attainable by the virtual monitors m_1' and m_2', if the virtual monitors can make end-to-end measurements along simple paths in $\mathcal{G}_{ex}$.

First, we show that any measurement between the real monitors can be obtained from measurements between m_1' and m_2'. To this end, consider a path $m_i \mathcal{P} m_j$ ($i, j \in \{1, \ldots, \kappa\}$, $i \neq j$) in $\mathcal{G}$, as shown in Fig. 3(b). Four simple paths between m_1' and m_2' can be constructed:

$$\begin{cases} \mathcal{P}_A = m_1' m_i m_2', \\ \mathcal{P}_B = m_1' m_j m_2', \\ \mathcal{P}_C = m_1' m_i \mathcal{P} m_j m_2', \\ \mathcal{P}_D = m_1' m_j \mathcal{P} m_i m_2'. \end{cases} \quad (10)$$

Viewing $m_i \mathcal{P} m_j$ as a "cross-link", we can compute $W_{m_i \mathcal{P} m_j}$ from the measurements on these four paths via (7) (replacing W_y by $W_{m_i \mathcal{P} m_j}$).

Moreover, we show that measurements between m_1' and m_2' in $\mathcal{G}_{ex}$ do not provide extra information for identifying links in $\mathcal{G}$ compared with measurements attainable by the real monitors. This is proved by observing that for any $m_1' \to m_2'$ simple path $m_1' m_i \mathcal{P} m_j m_2'$ ($i, j \in \{1, \ldots, \kappa\}$, $i \neq j$) containing at least one link in $\mathcal{G}$, the information relevant for identifying links in $\mathcal{G}$ can be obtained by measuring its $m_i \to m_j$ sub-path $m_i \mathcal{P} m_j$, which must also be a simple path. □

6.2 Sketch of Proof for Theorem 3.3

The special structure of $\mathcal{G}_{ex}$ allows us to consolidate the two Conditions ① and ② into a single condition as stated in Theorem 3.3, based on the following arguments.

From the structure of $\mathcal{G}_{ex}$ (see Fig. 3), we can prove in Propositions A.3 and A.4 (Appendix) that $\mathcal{G}_{ex}$ satisfies Conditions ① and ② in Theorem 3.2 if and only if $\mathcal{G}_{ex}$ is both 3-edge-connected and 3-vertex-connected. According to Proposition 1.4.2 in [25], a 3-vertex-connected graph is also 3-edge-connected. Thus, the necessary and sufficient conditions in

Lemma 6.1 can be simplified to a single condition that $\mathcal{G}_{ex}$ be 3-vertex-connected. $\square$

7. TESTING AND ENSURING IDENTIFIA-BILITY

The conditions we have derived have broader impact than mere theoretical interest. A major benefit of characterizing network identifiability in terms of network topology is that we can leverage existing graph-processing algorithms to efficiently test for or ensure the identifiability of a given network. In this section, we present efficient algorithms that can (i) test for the identifiability of a given network with a given monitor placement, and (ii) place the minimum number of monitors in a given network to identify all its link metrics.

7.1 Efficient Identifiability Test

The first question we want to answer is: Given a network topology $\mathcal{G}$ and a placement of $\kappa \geq 2$ monitors, how do we efficiently determine if $\mathcal{G}$ is identifiable or not? If $\kappa = 2$, then we know from Theorem 3.1 that it is impossible to identify the entire $\mathcal{G}$. Nevertheless, we can test whether the interior graph is identifiable using Conditions ① and ② in Theorem 3.2, which transform into multiple tests of edge/vertex connectivity. The problem of determining whether a given graph is k-edge/vertex-connected has been well studied. Specifically, fast algorithms have been proposed to test if a graph is: (i) 2-edge-connected [27], or (ii) 3-vertex-connected [28], both in time $O(|V|+|L|)$ ($|V|$: number of nodes; $|L|$: number of links). Using these algorithms, we can test for the identifiability of the *interior graph* of $\mathcal{G}$ as follows:

1. For each interior link l, apply the 2-edge-connectivity test in [27] to $\mathcal{G} - l$. $\mathcal{G}$ is unidentifiable if the test fails;

2. Apply the 3-vertex-connectivity test in [28] to $\mathcal{G} + m_1 m_2$. $\mathcal{G}$ is unidentifiable if the test fails.

The interior graph of $\mathcal{G}$ is identifiable if all the tests succeed. The overall complexity is $O(|L(\mathcal{G})|(|V(\mathcal{G})| + |L(\mathcal{G})|))$.

Similarly, if $\kappa \geq 3$, then we can test for the identifiability of the *entire* $\mathcal{G}$ using the condition in Theorem 3.3:

1. Construct the extended graph $\mathcal{G}_{ex}$ as in Fig. 3;

2. Apply the 3-vertex-connectivity test in [28] to $\mathcal{G}_{ex}$. $\mathcal{G}$ is identifiable if the test succeeds, and unidentifiable otherwise.

The complexity of this algorithm is $O(|V(\mathcal{G}_{ex})| + |L(\mathcal{G}_{ex})|)$, which is the same as $O(|V(\mathcal{G})| + |L(\mathcal{G})|)$.

7.2 Optimal Monitor Placement

The next question we want to answer is: Given an arbitrary topology $\mathcal{G}$, what is the minimum number of monitors needed and where should they be placed to identify all link metrics in $\mathcal{G}$? While one can enumerate all possible placements for $\kappa = 3, 4, \ldots$ monitors and test for identifiability until an identifiable placement is found, what we really want is an efficient algorithm to achieve the same. This requires a deeper understanding of the structure of an identifiable graph.

We illustrate our idea by an example in Fig. 8(a). We first consider the minimum deployment: Using fewer monitors
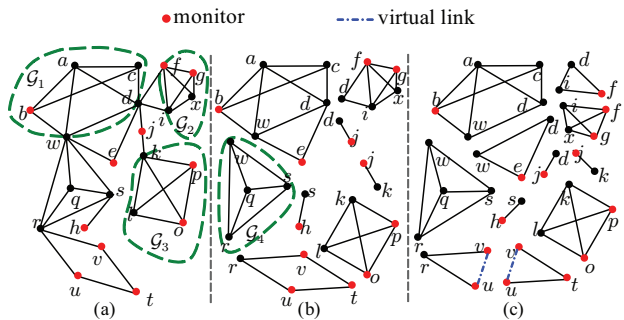

Figure 8: (a) Original graph; (b) biconnected components; (c) triconnected components.

certainly renders the network unidentifiable. This yields the following rules:

(i) *Dangling* node (e.g., h) must be a monitor, as otherwise its adjacent link cannot be measured using simple paths;

(ii) A node on a *tandem* of links (e.g., j) must be a monitor, as otherwise we can only identify the sum of its adjacent link metrics (W_{dj} and W_{jk}) and not the individual metrics;

(iii) For a sub-graph with *two cut-vertices*[8] (e.g., $\mathcal{G}_1$) or a *2-vertex cut*[9] (e.g., $\mathcal{G}_2$), at least one node other than those cuts must be a monitor, as otherwise even if all links outside this sub-graph have been identified, the vertices in the cuts (w and d for $\mathcal{G}_1$, f and i for $\mathcal{G}_2$) are effectively the two "monitors" for this sub-graph, and Theorem 3.1 states that this sub-graph cannot be identified;

(iv) Similarly, for a sub-graph with *one cut-vertex* (e.g., $\mathcal{G}_3$), at least two nodes other than the cut-vertex must be monitors.

Our strategy is to use the above four rules to deploy the necessary monitors. If we can prove that these necessary monitors are also sufficient to identify all links, then our deployment is optimal. To formally present the algorithm, we introduce the following definitions.

DEFINITION 5. *A k-connected component of $\mathcal{G}$ is a maximal sub-graph 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.

Intuitively, a biconnected component is a sub-graph connected to the rest of the graph by cut-vertices, and a triconnected component (within a biconnected component) is a sub-graph connected to the rest by 2-vertex cuts. For instance, Fig. 8(b) shows the biconnected components of Fig. 8(a), separated by cut-vertices d, j, k, w, s, and r. Fig. 8(c) shows the triconnected components, separated by the above cut-vertices and 2-vertex cuts $\{w, d\}$, $\{f, i\}$, and

---

[8]A *cut-vertex* is a vertex whose removal will disconnect the graph.

[9]A *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 $\mathcal{G}$.

Algorithm 1: Minimum Monitor Placement (MMP)

---

input : Connected graph $\mathcal{G}$
output: A subset of nodes in $\mathcal{G}$ as monitors

1 choose all the nodes with degree less than 3 as monitors;
2 partition $\mathcal{G}$ into biconnected components $\mathcal{B}_1, \mathcal{B}_2, \ldots$;
3 foreach *biconnected component $\mathcal{B}_i$ with $|\mathcal{B}_i| \geq 3$* **do**
4   partition $\mathcal{B}_i$ into triconnected components $\mathcal{T}_1, \mathcal{T}_2, \ldots$;
5   **foreach** *triconnected component $\mathcal{T}_j$ of $\mathcal{B}_i$ with $|\mathcal{T}_j| \geq 3$* **do**
6   **if** $0 < s_{\mathcal{T}_j} < 3$ *and* $s_{\mathcal{T}_j} + M_{\mathcal{T}_j} < 3$ **then**
7   randomly choose $3 - s_{\mathcal{T}_j} - M_{\mathcal{T}_j}$ nodes in $\mathcal{T}_j$ that are neither separation vertices nor monitors as monitors;
8   **end**
9   **end**
10   **if** $0 < c_{\mathcal{B}_i} < 3$ *and* $c_{\mathcal{B}_i} + M_{\mathcal{B}_i} < 3$ **then**
11   randomly choose $3 - c_{\mathcal{B}_i} - M_{\mathcal{B}_i}$ nodes in $\mathcal{B}_i$ that are neither cut-vertices nor monitors as monitors;
12   **end**
13 end
14 if *the total number of monitors $\kappa < 3$* **then**
15   randomly choose $3 - \kappa$ non-monitor nodes as monitors;
16 end

$\{u, v\}$. To identify the triconnected components, we process the graph by adding *virtual links* as follows: if $\exists$ a *minimal 2-vertex cut*[10] whose vertices are not neighbors (e.g., $\{u, v\}$), connect them by a virtual link; repeat this on the resulting graph until no such cut exists. In the sequel, nodes that are cut-vertices or part of 2-vertex cuts are called *separation vertices* (e.g., w, d, f, i, j, k, s, r, u, and v).

Monitor Placement Algorithm: There exist fast algorithms to partition an arbitrary graph $\mathcal{G}$ into biconnected components [29] and then into triconnected components [28]. Note that the output of [28] is the set of triconnected components of $\mathcal{G}$ augmented with the virtual links[11]. Based on these algorithms, we propose a master algorithm, *Minimum Monitor Placement (MMP)*, to place the minimum number of monitors needed to identify $\mathcal{G}$. As shown in Algorithm 1, MMP first applies rules (i) and (ii) to select all the dangling vertices and vertices on tandems as monitors (line 1), and then applies rules (iii) and (iv) to select additional monitors in each[12] triconnected/biconnected component. For a component $\mathcal{D}$, let $s_{\mathcal{D}}$ denote the number of separation vertices, $c_{\mathcal{D}}$ the number of cut-vertices, and $M_{\mathcal{D}}$ the number of (already selected) monitors in $\mathcal{D}$. MMP goes through each triconnected and then biconnected component that contains three or more nodes to ensure that: (i) each triconnected component has at least three nodes that are either separation vertices or monitors (lines 6–8), and (ii) each biconnected component has at least three nodes that are either cut-vertices or monitors (lines 10–12). Finally, it selects additional monitors as needed to ensure that the total number of monitors is at least three (lines 14–16).

Optimality: It is easy to see from rules (i)–(iv) that MMP only deploys monitors when needed, and thus no algorithm can achieve identifiability with fewer monitors; on the other hand, we show that the monitor placement by MMP is also sufficient, i.e., all link metrics can be iden-

---

[10]That is, neither of the vertices are cut-vertices.
[11]The algorithm in [28] does not require a separate step to add virtual links; it adds virtual links as needed in the process of graph partitioning.
[12]MMP does not depend on the order of biconnected/triconnected components being considered for monitor selection.

---

tified from end-to-end measurements between the selected monitors. Thus, MMP is optimal as it places the *minimum* number of monitors to identify all link metrics in $\mathcal{G}$, as stated in the following theorem.

THEOREM 7.1. *For an arbitrary connected network $\mathcal{G}$, Algorithm 1 (MMP) generates the* optimal *monitor placement in the sense that: (1) all link metrics in $\mathcal{G}$ are identifiable under this placement, and (2) no placement can identify all link metrics in $\mathcal{G}$ with a smaller number of monitors.*

PROOF. See [26]. □

Complexity: In Algorithm 1, lines 1 and 14–16 take $O(|V(\mathcal{G})|)$ time. Splitting $\mathcal{G}$ into biconnected (line 2) and then triconnected components (line 4) takes $O(V(\mathcal{G}) + L(\mathcal{G}))$ time [28, 29]. Selecting monitors takes $O(1)$ time per component, and the counters ($s_{\mathcal{T}_j}$, $M_{\mathcal{T}_j}$, $c_{\mathcal{B}_i}$, and $M_{\mathcal{B}_i}$) can be computed during the splitting/selecting process. Therefore, the entire algorithm has $O(|V(\mathcal{G})| + |L(\mathcal{G})|)$ time complexity.

Example: Given the graph in Fig. 8(a) as input (totally 22 nodes), MMP selects 11 monitors, where nodes $\{h, u, v, t, j, e\}$ are selected by line 1, $\{b, g, o, p\}$ by lines 6–8, and f by lines 10–12. It can be verified that the resulting graph satisfies the identifiability condition in Theorem 3.3.

7.3 Evaluation of MMP

We evaluate MMP through a set of simulations on both randomly-generated and real network topologies. We use the following algorithm, referred to as *Random Monitor Placement (RMP)*, as a benchmark for comparison[13]: Given network $\mathcal{G}$, randomly select κ ($\kappa = 2, \cdots, |V|$) nodes as monitors and test the identifiability of the resulting network using the algorithm in Section 7.1. Generally, RMP cannot guarantee network identifiability for arbitrary $\mathcal{G}$ and κ. Therefore, we measure its performance by the fraction of random placements achieving network identifiability over multiple Monte Carlo runs.

7.3.1 Random Topologies

We first consider synthetic topologies generated according to four widely used random graph models: Erdös-Rényi (ER) graphs, Random Geometric (RG) graphs, Barabási-Albert (BA) graphs, and Random Power Law (PL) graphs. We randomly generate 100 graph realizations of each model[14], with each realization containing 150 nodes (i.e., $|V| = 150$). The generated graphs are then input to the monitor placement algorithms. We now explain the models and the corresponding results separately.

Erdös-Rényi (ER) graph: The ER graph is a simple random graph generated by independently connecting each pair of nodes by a link with a fixed probability p. The result is a purely random topology where all graphs with an equal number of links are equally likely to be selected. It is known [30] that $p_0 = \log |V|/|V|$ is a sharp threshold for the graph to be connected with high probability, which implies a minimum value of $p = 0.0334$ for $|V| = 150$.

---

[13]To our knowledge, MMP is the first monitor placement algorithm for identifying additive link metrics by measuring controllable, cycle-free paths. Thus, we use random placement to represent the average performance of an arbitrary monitor placement for comparison.
[14]All realizations are guaranteed to be connected, as we discard disconnected realizations in the generation process.

Random Geometric (RG) graph: The RG graph is frequently used to model the topology of wireless ad hoc networks. It generates a random graph by first randomly distributing nodes in a unit square, and then connecting each pair of nodes by a link if their distance is no larger than a threshold d_c, which denotes node communication range. The resulting topology contains well-connected sub-graphs in densely populated areas and poorly-connected sub-graphs in sparsely populated areas. It is known that $d_c \geq \sqrt{\log|V|/(\pi|V|)}$ ensures a connected graph with high probability [31], which gives a minimum range of $d_c = 0.1031$ for $|V| = 150$.

Barabási-Albert (BA) graphs: The BA model [32] provides a random power-law graph generated by the following preferential attachment mechanism. We begin with a small connected graph $\mathcal{G}_0 := (\{v_1, v_2, v_3, v_4\}, \{v_1v_2, v_1v_3, v_1v_4\})$ and add nodes sequentially. For each new node v, we connect v to $n_{\min}$ existing nodes, where $n_{\min}$ specifies (a lower bound on) the minimum node degree, such that the probability of connecting to node w is proportional to the degree of w. If the number of existing nodes is smaller than $n_{\min}$, then v connects to all the existing nodes. The BA graph has been used to model many naturally occurring networks, e.g., Internet, citation networks, and social networks.

Random Power Law (PL) graphs: The BA model introduces an artifact that all node degrees are lower bounded by $n_{\min}$. Alternatively, the PL graph [33] provides another way of generating power-law graphs by directly specifying a sequence of expected node degrees $(d_1, ..., d_{|V|})$ according to the power law, i.e., $d_i = i^\alpha$ ($\alpha > 0$). The generation of a PL graph is similar to that of an ER graph, except that instead of connecting each pair of nodes with the same probability, nodes i and j in a PL graph are connected by a link with probability $p_{ij} = d_i d_j / \sum_{k=1}^{|V|} d_k$.

We conduct simulations in two scenarios, densely-connected graphs (Fig. 9 with parameter configurations: $p = 0.039$ for ER, $d_c = 0.11943$ for RG, $n_{\min} = 3$ for BA and $\alpha = 0.42$ for PL) and sparsely-connected graphs (Fig. 10 with parameter configurations: $p = 0.0253$ for ER, $n_{\min} = 2$ for BA and $\alpha = 0.32$ for PL), to evaluate how graph density affects the performance of MMP and RMP. Since the number of links n and the minimum number of monitors κ_{MMP} (computed by MMP) vary across graph realizations, we present the average values denoted by $\overline{n}$ and $\overline{\kappa}_{\text{MMP}}$ for each graph model, shown in the captions of Fig. 9 and 10. In each scenario, we have tuned parameters of each model to generate roughly the same average number of links. As RMP is a randomized algorithm, we repeat it for 2000 Monte Carlo runs to obtain the average performance for each graph realization. We then average the results over the 100 graph realizations to obtain the final results[15], shown in Fig. 9 and 10.

For densely-connected graphs (Fig. 9), the probability that RMP is able to identify all the links increases with the number of monitors κ. However, fewer than 20% of the ER, RG, and PL graphs are identifiable when $\kappa \leq 120$ (recall $|V| = 150$), whereas a careful monitor deployment in the same graphs by MMP ensures identifiability with significantly fewer monitors ($\kappa < 25$). Therefore, in most cases,

---

[15] For each graph realization, MMP achieves identifiability with probability one for $\kappa \geq \kappa_{\text{MMP}}$ and zero for $\kappa < \kappa_{\text{MMP}}$. Therefore, the overall probability for MMP to achieve identifiability using κ monitors is computed as the fraction of graph realizations with $\kappa_{\text{MMP}} \leq \kappa$.
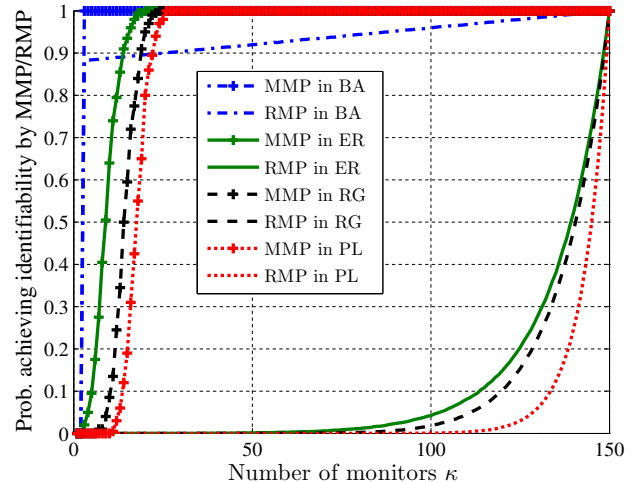

Figure 9: Comparison between RMP and MMP: densely-connected graphs ($\overline{n} = 441, \overline{\kappa}_{\text{MMP}} = 3$ for BA, $\overline{n} = 437, \overline{\kappa}_{\text{MMP}} = 9.36$ for ER, $\overline{n} = 437, \overline{\kappa}_{\text{MMP}} = 19.42$ for PL and $\overline{n} = 451, \overline{\kappa}_{\text{MMP}} = 14.52$ for RG).

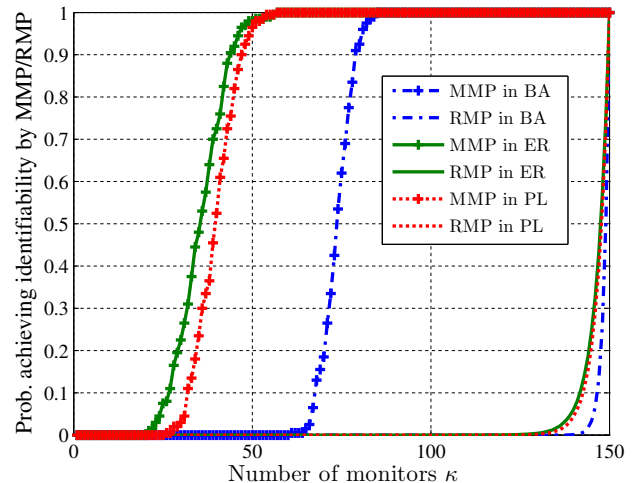**Figure 10: Comparison between RMP and MMP: sparsely-connected graphs ($\overline{n} = 295, \overline{\kappa}_{\text{MMP}} = 73.51$ for BA, $\overline{n} = 293, \overline{\kappa}_{\text{MMP}} = 36.76$ for ER and $\overline{n} = 297, \overline{\kappa}_{\text{MMP}} = 40.24$ for PL).**

the proposed algorithm (MMP) substantially outperforms a randomized scheme (RMP) in the required number of monitors. One exception is the BA graphs, where RMP achieves similar performance as MMP. This can be explained as follows: When $n_{\min} = 3$, further simulations show that 87.8% of the generated BA graphs are 3-vertex-connected as each node (except the initial nodes v_2, v_3, and v_4) has at least three neighbors; according to MMP, this implies that an arbitrary placement of three monitors will achieve identifiability, and there is no need for sophisticated placement algorithms. Meanwhile, a comparison of the BA and the PL models implies that the exceptional performance of RMP in the BA model is due to the requirement that all node degrees be lower bounded by three rather than the power law distribution of node degrees, since it does not achieve the same superior performance on PL graphs. For ER, RG, and PL
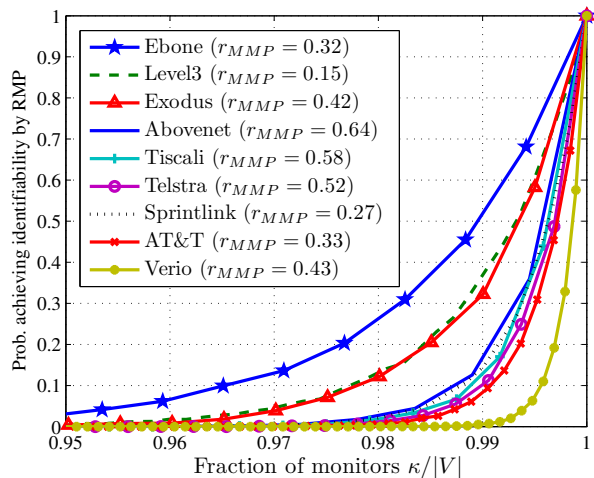
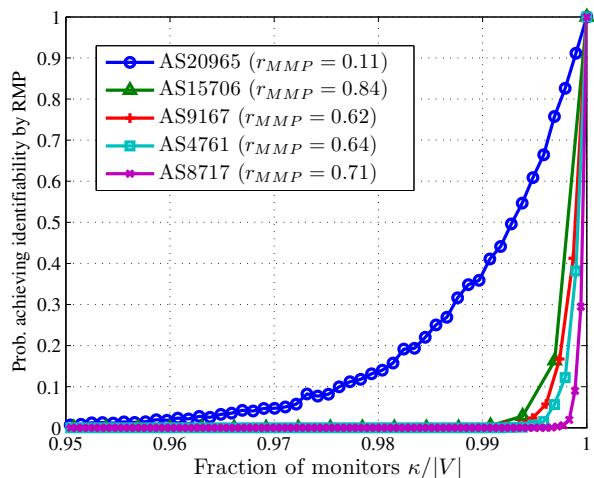**Figure 11: Comparison between RMP and MMP: ISP topologies in Rocketfuel** ($15 \cdot |V|$ **Monte Carlo**

Figure 12: Comparison between RMP and MMP: AS topologies in CAIDA (3000 **Monte Carlo runs,** $r_{\mathrm{MMP}} := \kappa_{\mathrm{MMP}}/|V|$**).**

graphs, it is easy to form sparsely-connected subgraphs, thus requiring more monitors to guarantee identifiability. Among the four models, we observe that RMP exhibits worse performance in the model with a larger $\overline{\kappa}_{\mathrm{MMP}}$. This is because large $\overline{\kappa}_{\mathrm{MMP}}$ implies poor connectivity within certain subgraphs, where a considerable number of nodes have to be selected as monitors to achieve identifiability. Thus, RMP will fail to achieve identifiability if it does not select sufficient nodes in any of these subgraphs.

We perform similar simulations for sparsely-connected graphs by adjusting the parameters[16], shown in Fig. 10. In comparison with Fig. 9, both MMP and RMP perform worse in sparsely-connected graphs, requiring a larger number of

---

[16]To generate roughly the same number of links as in the other models, d_c for the RG graph becomes much smaller than $\sqrt{\log|V|/(\pi|V|)}$, making the probability of generating a connected RG graph very small. Thus, the RG model is omitted from Fig. 10.

Table 2: Parameters of AS Topologies in Rocketfuel

AS	ISP Name	$|L|$	$|V|$	κ_{MMP}	r_{MMP}
6461	Abovenet (US)	294	182	117	0.64
1755	Ebone (Europe)	381	172	55	0.32
3257	Tiscali (Europe)	404	240	138	0.58
3967	Exodus (US)	434	201	85	0.42
1221	Telstra (Australia)	758	318	164	0.52
7018	AT&T (US)	2078	631	208	0.33
1239	Sprintlink (US)	2268	604	163	0.27
2914	Verio (US)	2821	960	408	0.43
3356	Level3 (US)	5298	624	94	0.15

Table 3: Parameters of AS Topologies in CAIDA

AS	$|L|$	$|V|$	κ_{MMP}	r_{MMP}
15706	874	325	276	0.84
9167	1590	769	483	0.62
8717	3755	1778	1266	0.71
4761	3760	969	624	0.64
20965	8283	968	110	0.11

monitors to achieve the same probability of identifiability. This is because sparser graphs impose more constraints on candidate measurement paths between a pair of monitors, and thus more monitors are needed to identify all links. As we compare the performance of the same graph model in Fig. 9 and 10, we observe that all models exhibit smooth transitions when we vary the average number of links $\overline{n}$ except for the BA model, which has extremely good performance for $n_{\min} = 3$ but the worst performance (except for RG) for $n_{\min} = 2$. This can again be explained by examining the 3-vertex-connectivity of the generated graph realizations. We have verified that as we reduce $n_{\min}$ from 3 to 2, the probability for BA graphs to be 3-vertex-connected suddenly drops from 87.8% to 0%, making it unlikely for an arbitrary placement to achieve identifiability. In fact, when $n_{\min} = 2$, on average 49.2% of the nodes in BA graphs have degrees less than 3, which have to be selected as monitors by rules (i) and (ii) in MMP. Therefore, RMP will fail to achieve identifiability even if it misses one of these nodes, resulting in its poor performance.

7.3.2 Autonomous System Topologies

We now test MMP and RMP on real network topologies. We use the *Autonomous System* (AS) topologies from both the Rocketfuel [34] and the CAIDA [35] projects, which represent IP-level connections between backbone/gateway routers of several ASes from major *Internet Service Providers (ISPs)* around the globe. The parameters of selected networks obtained from these two projects are listed in Tables 2 and 3, where we sort the networks according to their numbers of links, and $r_{\mathrm{MMP}} := \kappa_{\mathrm{MMP}}/|V|$ denotes the minimum fraction of monitors computed by MMP in a network with $|V|$ nodes.

1) AS topologies from Rocketfuel: As shown in Table 2, each AS in Rocketfuel corresponds to an ISP. To identify the entire network, we observe that most ISPs need a significant fraction of nodes to be monitors, ranging from around 30% (Ebone, AT&T, Sprintlink) to more than 60%

(Abovenet). This is because ISP networks contain a large number of gateway[17] routers to connect to customer networks or other ISPs, which appear as dangling nodes that have to be selected as monitors (see rule (i) in Section 7.2). We repeat RMP for $15 \cdot |V|$ Monte Carlo runs for each ISP to evaluate its average performance, measured by the fraction of Monte Carlo runs achieving identifiability, as shown in Fig. 11. To facilitate comparison, we mark the fraction of monitors r_{MMP} needed by MMP in the legends of the same plot; note that MMP guarantees identifiability for each network as long as $\kappa/|V| \geq r_{\mathrm{MMP}}$. Here we normalize the number of monitors by the total number of nodes for the x-axis to compare networks of various sizes. Moreover, we only plot the results of RMP for $\kappa/|V| \geq 95\%$ since RMP fails to achieve identifiability in almost all the Monte Carlo runs for all the ISPs when $\kappa/|V| < 95\%$.

As in the case of synthetic graphs, we again observe a significant improvement of MMP over RMP. Specifically, RMP has at most 50% probability of identifying all the links even if 99% of the nodes are monitors, whereas MMP guarantees identifiability using at most 64% of nodes as monitors. The poor performance of RMP is due to the heterogeneous connectivity within ISP networks, which contain poorly-connected subnetworks that need a large fraction of monitors, a requirement unlikely to be fulfilled by random placement. We also observe that the relative performance of MMP and RMP varies for different networks, e.g., Level3 and Exodus experience similar performance under RMP, whereas their minimum fractions of monitors computed by MMP differ significantly (0.15 for Level3 and 0.42 for Exodus). Intuitively, this is because RMP performance is determined by the relative number of valid placements, i.e., the fraction of all the $\binom{|V|}{\kappa}$ candidate placements that achieve identifiability, whereas MMP performance is determined by the smallest κ for which this fraction is non-zero.

2) AS topologies from CAIDA: Because ISP topologies have evolved since the Rocketfuel project, we repeat the above evaluation on a recent data set obtained by the CAIDA project; see results in Table 3 and Fig. 12. Compared with the ASes in Rocketfuel data set, we notice that ASes with similar average node degrees (i.e., $2|L|/|V|$) in CAIDA data set require a larger fraction of monitors for complete identification, e.g., $r_{\mathrm{MMP}} = 0.71$ for AS8717 with average node degree 4.2 in Table 3, whereas $r_{\mathrm{MMP}} = 0.32$ for AS1755 (Ebone) with average node degree 4.4 in Table 2. This is because ASes in CAIDA data set tend to be more skewed in connectivity, with more densely-connected cores and a larger number of dangling nodes (likely the gateways for peer/customer connections). As all dangling nodes have to be selected as monitors, such skewed connectivity leads to a higher ratio of monitors although the average node degree is not necessarily lower. Meanwhile, comparison with RMP (3000 Monte Carlo runs), presented in Fig. 12, shows that the improvement of MMP over RMP remains significant. As in Fig. 11, RMP again exhibits poor performance for all the ASes considered in Fig. 12, which has less than 35% probability of identifying all links even if the fraction of monitors $\kappa/|V|$ is as large as 0.99. In particular, for almost all the

networks (except for AS20965) in Fig. 12, RMP fails to identify all the links in more than 60% of the Monte Carlo runs even if *all but one* node are monitors. In contrast, carefully selected monitors by MMP can guarantee complete identification while substantially reducing the required number of monitors.

8. CONCLUSION

In this paper, we study the fundamental conditions on network topology and placement of monitors for identifying additive link metrics using end-to-end measurements along simple paths between monitors. We show that with two monitors, it is impossible to identify all the link metrics, but is possible to identify the metrics of interior links that are at least one hop away from the monitors, for which we derive the necessary and sufficient conditions in terms of edge/vertex connectivity of the network topology. We further study the case of three or more monitors and derive the corresponding necessary and sufficient conditions for identifying all the link metrics. We show that these conditions are not only useful for testing network identifiability under a given monitor placement, but also enable an efficient monitor placement algorithm that guarantees identifiability using the minimum number of monitors. Our evaluations on both random and real network topologies verify that the proposed algorithm significantly outperforms a baseline solution.

9. 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] A. B. Downey, "Using pathchar to estimate internet link characteristics," in *IEEE SIGCOMM*, 1999.

[3] G. Jin, G. Yang, B. Crowley, and D. Agarwal, "Network characterization service (ncs)," in *IEEE HPDC*, 2001.

[4] M. Coates, A. O. Hero, R. Nowak, and B. Yu, "Internet tomography," *IEEE Signal Processing Magzine*, vol. 19, pp. 47–65, 2002.

[5] E. Lawrence and G. Michailidis, "Network tomography: A review and recent developments," *Frontiers in Statistics*, vol. 54, 2006.

[6] O. Gurewitz and M. Sidi, "Estimating one-way delays from cyclic-path delay measurements," in *IEEE INFOCOM*, 2001.

[7] Y. Chen, D. Bindel, and R. H. Katz, "An algebraic approach to practical and scalable overlay network monitoring," in *ACM SIGCOMM*, 2004.

[8] A. Chen, J. Cao, and T. Bu, "Network Tomography: Identifiability and Fourier domain estimation," in *IEEE INFOCOM*, 2007.

[9] "Open networking foundation." [Online]. Available: http://www.opennetworkingfoundation.org

[10] N. Duffield and F. Lo Presti, "Multicast inference of packet delay variance at interior network links," in *IEEE INFOCOM*, 2000.

[11] Y. Xia and D. Tse, "Inference of link delay in communication networks," *IEEE Journal of Selected Areas in Communications*, 2006.

---

[17]In real networks, monitor selection may be constrained to a subset of nodes, e.g., gateways. Under such constraint, the issue of achievable number of identifiable links (i.e., partial network identifiability) is left for future work.

[12] A. Adams, T. Bu, T. Friedman, J. Horowitz, D. Towsley, R. Caceres, N. Duffield, F. Presti, and V. Paxson, "The use of end-to-end multicast measurements for characterizing internal network behavior," *IEEE Communications Magazine*, vol. 38, no. 5, pp. 152–159, May 2000.

[13] R. Castro, M. Coates, G. Liang, R. Nowak, and B. Yu, "Network tomography: recent developments," *Statistical Science*, 2004.

[14] M.-F. Shih and A. Hero, "Unicast inference of network link delay distributions from edge measurements," in *IEEE ICASSP*, 2001.

[15] T. Bu, N. Duffield, and F. Lo Presti, "Network tomography on general topologies," in *ACM SIGMETRICS*, 2002.

[16] M. Firooz and S. Roy, "Network tomography via compressed sensing," in *IEEE GLOBECOM*, 2010.

[17] W. Xu, E. Mallada, and A. Tang, "Compressive sensing over graphs," in *IEEE INFOCOM*, 2011.

[18] S. Ahuja, S. Ramasubramanian, and M. Krunz, "SRLG failure localization in all-optical networks using monitoring cycles and paths," in *IEEE INFOCOM*, 2008.

[19] A. Gopalan and S. Ramasubramanian, "On identifying additive link metrics using linearly independent cycles and paths," *IEEE/ACM Transactions on Networking*, vol. PP, no. 99, 2011.

[20] N. Alon, Y. Emek, M. Feldman, and M. Tennenholtz, "Economical graph discovery," in *Symposium on Innovations in Computer Science*, 2011.

[21] L. Ma, T. He, K. Leung, D. Towsley, and A. Swami, "Efficient identification of additive link metrics via network tomography," in *ICDCS*, 2013.

[22] Y. Bejerano and R. Rastogi, "Robust monitoring of link delays and faults in IP networks," in *IEEE INFOCOM*, 2003.

[23] R. Kumar and J. Kaur, "Practical beacon placement for link monitoring using network tomography," *IEEE JSAC*, vol. 24, no. 12, pp. 2196–2209, 2006.

[24] J. D. Horton and A. Lopez-Ortiz, "On the number of distributed measurement points for network tomography," in *ACM/USENIX IMC*, 2003.

[25] R. Diestel, *Graph theory*. Springer-Verlag Heidelberg, New York, 2005.

[26] L. Ma, T. He, K. K. Leung, A. Swami, and D. Towsley, "Identification of additive link metrics: Proof of selected theorems," Technical Report, Imperial College, London, UK., July 2012. [Online]. Available: http://www.commsp.ee.ic.ac.uk/%7elm110/pdf/MaTechreportJul12.pdf

[27] R. E. Tarjan, "A note on finding the bridges of a graph," *Information Processing Letters*, pp. 160–161, 1974.

[28] J. E. Hopcroft and R. E. Tarjan, "Dividing a graph into triconnected components," *SIAM Journal on Computing*, vol. 2, pp. 135–158, 1973.

[29] R. Tarjan, "Depth-first search and linear graph algorithms," *SIAM Journal on Computing*, vol. 1, pp. 146–160, 1972.

[30] P. Erdös and A. Rényi, "On the evolution of random graphs," *Publications of the Mathematical Institute of the Hungarian Academy of Sciences*, vol. 5, pp. 17–61, 1960.

[31] P. Gupta and P. Kumar, "Critical power for asymptotic connectivity in wireless networks," *Stochastic Analysis, Control, Optimization and Applications*, pp. 547–566, 1999.

[32] R. Albert and A.-L. Barabási, "Statistical mechanics of complex networks," *Reviews of Modern Physics*, vol. 74, pp. 47–97, Jan. 2002.

[33] F. Chung and L. Lu, *Complex Graphs and Networks*. American Mathematical Society, 2006.

[34] "Rocketfuel: An ISP topology mapping engine," University of Washington, 2002. [Online]. Available: http://www.cs.washington.edu/research/networking/rocketfuel/

[35] "Macroscopic Internet Topology Data Kit (ITDK)," The Cooperative Association for Internet Data Analysis (CAIDA), Apr. 2013. [Online]. Available: http://www.caida.org/data/active/internet-topology-data-kit/

APPENDIX

A. LEMMAS AND PROPOSITIONS

LEMMA A.1. *Suppose two monitors are deployed in $\mathcal{G}$ to measure simple paths. If link l is a bridge in $\mathcal{G}$ with one monitor on each side, as illustrated in Fig. 13, then neither l nor its adjacent links are identifiable.*
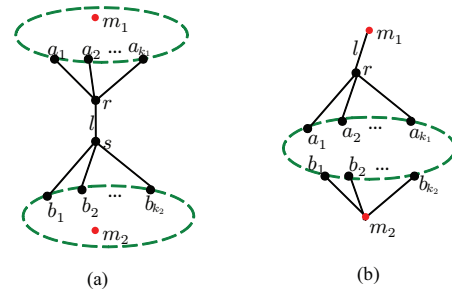
PROOF. See [26]. □

Figure 13: Two cases of bridge link l: (a) interior bridge, (b) exterior bridge.

PROPOSITION A.2. *Using two monitors measuring simple paths, if all link metrics in the interior graph of $\mathcal{G}$ are identifiable, then $\mathcal{G} + m_1 m_2$ is 3-vertex-connected.*

PROOF. See [26]. □

PROPOSITION A.3. *Given a graph G employing κ ($\kappa \geq 3$) monitors, the extended graph $\mathcal{G}_{ex}$ of $\mathcal{G}$ satisfies Conditions ① (i.e., $\mathcal{G}_{ex} - l$ is 2-edge-connected for each link l in $\mathcal{G}$) if and only if $\mathcal{G}_{ex}$ is 3-edge-connected.*

PROOF. See [26]. □

PROPOSITION A.4. *Given a graph G employing κ ($\kappa \geq 3$) monitors, the extended graph $\mathcal{G}_{ex}$ of $\mathcal{G}$ satisfies Conditions ② (i.e., $\mathcal{G}_{ex} + m'_1 m'_2$ is 3-vertex-connected) if and only if $\mathcal{G}_{ex}$ is 3-vertex-connected.*

PROOF. See [26]. □