Robust Network Tomography: k-identifiability and Monitor Assignment

Wei Ren, and Wei Dong*
College of Computer Science, Zhejiang University
Email: renwei@emnets.org, dongw@zju.edu.cn

***Abstract*—Network tomography is an attractive approach for inferring internal network states at edge nodes. Recently, there is a growing interest in the basic understanding of the *topological* conditions that ensure identifiability in a general network. Unfortunately, existing works assume an *ideal* network model where all network elements are reliable. In this paper, we are aiming to propose topological conditions to ensure identifiability *in the presence of any* k *link failures* ($k \geq 0$), using measurement paths and cycles. We propose a novel concept called k-identifiability. We propose *sufficient* and *necessary* topological conditions to ensure k-identifiability. Based on the established theoretical foundations, we propose two efficient polynomial-time algorithms for addressing two closely related questions. (1) Given a network with specified monitors, which links are k-identifiable? (2) Given a network and κ monitors, where should these monitors be placed such that the number of k-identifiable links is maximized? Simulation results on real ISP network topologies show the effectiveness of our algorithms.**

I. Introduction

The past few years have seen a remarkable growth in the global network infrastructure. The Internet is becoming increasingly complex in terms of its traffic load and composition, as well as its commercial usage. The management of such large networks is challenging. Accurate and timely knowledge of the internal states of a network is essential for various network operations such as route selection, resource allocation, and performance diagnosis. Directly measuring the performance of individual network elements is, however, not always feasible due to the measurement overhead and the lack of support at the underlying network. *Network tomography* is an attractive approach for inferring internal states of a network based on *end-to-end* measurements at monitors located at the network edges. A commonly adopted approach is to establish a linear system that models the relationship between *known* path metrics and *unknown* link metrics, assuming that the link metrics are additive (e.g., delay or packet delivery ratio after applying the log() function). The link metrics could be identified by solving the linear equations.

From the perspective of linear algebra, the link metrics are uniquely identifiable if and only if the number of linearly independent measurement paths equals the number of links. In networks with certain topologies, the selection of such measurement paths is not always possible, i.e., some link metrics cannot be uniquely identified no matter which set of measurement paths are selected. For example, if two links (not necessarily adjacent) always appear together in the measurement paths, then we can at most identify the sum of their metrics but not the individual metrics. This motivates a basic understanding of the *topological* conditions that ensure identifiability so that efforts to select measurement paths (as most prior work does) can be saved if certain desired link metrics cannot be identified in the first place.

Recently, Ma *et al.* [1] have successfully solved the above problem by proposing necessary and sufficient topological conditions to ensure identifiability as well as efficient algorithms for minimizing the number of monitors. *Unfortunately*, they assume an ideal network model where all network elements are reliable. However, failure of network elements are common events in the Internet due to many practical reasons [2].

In this paper, we are aiming to propose sufficient and necessary topological conditions to ensure identifiability in the presence of any k link failures ($k \geq 0$), using measurement paths and cycles. We propose a novel concept called k-identifiability:

- A (non-failed) link is k-identifiable if it is still identifiable in the presence of any k link failures.
- A network is k-identifiable if all its links are k-identifiable.

It is worth noting that a k-identifiable link can survive any k link failures (and of course $k-1$ link failures). Such a guarantee is very useful for both network managers and network designers. For example, network managers can be instructed how to place a minimum number of monitors so that the network is k-identifiable, or more generally, how to place a given number of monitors so that the number of k-identifiable links is maximized. Network designers can be instructed how to construct a "good" network topology with a given number of monitors and network elements so that the number of k-identifiable links is maximized.

We address the above problems by answering two closely related questions:

- Given a network with specified monitors, which links are k-identifiable?
- Given a network and κ monitors, where should these monitors be placed such that the number of k-identifiable links is maximized?

Existing algorithms are not suitable for these questions since they have exponential search space by enumerating all possible failure scenarios. We gain insights from Gopalan *et al.*'s earlier results on monitor merging [3] and find a key observation that *the k-identifiability of a link is not affected by merging all the monitors into a single monitor*. Based on this observation, we carefully study the relationship between the k-

*This work is supported by the National Science Foundation of China under Grant No. 61472360, China Ministry of Education–China Mobile Joint Project under Grant No. MCM20150401, and Zhejiang Provincial Platform of IoT Technology under Grant No. 2013E60005. Wei Dong is the corresponding author.

identifiability of a link and the location of the merged monitor in different network topologies. These results form the basis of our efficient polynomial-time algorithms addressing the above questions.

We implement the algorithms and conduct extensive simulations based on real Internet topologies. Results show the effectiveness of our algorithms.

The contributions of this paper are summarized as follows:

- We propose a novel concept, k-identifiability, to cope with link failures for monitor assignment in network tomography.
- We propose sufficient and necessary topological conditions to ensure k-identifiability, using measurement paths and cycles.
- We propose efficient polynomial-time algorithms for addressing the above-mentioned two questions. Extensive simulation results on real network topologies show the effectiveness of the algorithms.

The rest of this paper is structured as follows. Section II discusses the related work. Section III introduces the backgrounds and formulates our problem. Section IV theoretically establishes the necessary and sufficient conditions for k-identifiability. Based on the theoretical foundations, Section V presents the algorithm for identifying k-identifiable links, and Section VI presents the algorithm for monitor assignment to achieve maximal k-identifiability. Section VII discusses how to implement controllable routing in practical networks. Section VIII shows the evaluation results. Finally, Section IX concludes this paper and gives future research directions.

II. Related Work

The term network tomography, coined by Vardi [4], refers to the inference of certain internal characteristics of networks based on end-to-end measurements.

For constant and additive link metrics, many approaches have been proposed from the perspective of linear algebra. Chen *et al.* [5] show that it is challenging to uniquely identify all the link metrics due to the presence of linearly dependent paths. Many approaches have been proposed to optimize the selection of measurement paths. Zheng *et al.* [6] address the problem of selecting the minimum number of probing paths that can uniquely determine all the solution for a set of target links. Chen *et al.* [5] propose an algebraic approach that selectively monitors a subset of linearly independent paths that can fully describe the complete set of paths, i.e., the loss rates and latency of the subset of paths can be used to estimate the loss rates and latency of all other paths. Gopalan *et al.* [7] develop an algorithm to determine the maximum number of linearly independent cycles and the corresponding identifiable links with a single monitor. Tati *et al.* [8] study the problem of selecting paths to improve the performance of network tomography applications in the presence of network element failures. Our current work focuses on setting up *topological* conditions for ensuring identifiability of additive link metrics in the presence of link failures.

There is a growing interest in relating link identifiability to network topology and monitor placement. If the network is directed, Xia *et al.* [9] prove that not all link metrics are identifiable unless every non-isolated node is a monitor. If the network is undirected, Gopalan *et al.* [3] derive topological conditions for network identifiability, assuming measurements of cycles. Ma *et al.* establish the necessary and sufficient conditions [1] for network identifiability and associate algorithms [10] for identifying link metrics, employing cycle-free measurements.

Since identifying all network links is not always possible, many algorithms are proposed to achieve maximal identifiability given a limited number of monitors, or, to employ minimal number of monitors for identifying only a subset of links. Ma *et al.* [11] propose efficient algorithms to achieve maximal identifiability by placing a given number of monitors, using cycle-free measurements. Kumar *et al.* [12] prove that the problem of placing the minimum number of monitors to ensure complete identifiability under uncontrollable routing is NP-hard. They design various approximate placement algorithms to yield suboptimal results. Gao *et al.* [13] study the problem of preferential link tomography where only a subset of interesting links need to be identified. They develop scalable algorithms to place a minimum number of monitors to identify the interesting links. All the above works assume an ideal network model where all links are reliable. To our knowledge, we are the first to study the *topological* conditions to ensure identifiability of additive link metrics in the presence of link failures.

There are also related works exploiting network tomography to identify failed links. Ahuja *et al.* [14] prove that the network must be (k+2)-edge-connected to identify up to k failed links by measuring cycles at a single monitor. Ma *et al.* [15] investigate the problem of uniquely localizing node failures with three types of measurement paths (i.e., arbitrarily controllable, controllable but cycle-free, and, uncontrollable), under the assumption that a path behaves normally if and only if it does not contain any failed nodes. Our current work focuses on optimizing the network measurement performance under link failures, instead of identifying failed links.

III. Problem Formulation

A. Models and Assumptions

We assume that the network topology is known and does not change during the measurement period. We consider the network as an undirected graph $\mathcal{G}(V(\mathcal{G}), L(\mathcal{G}))$, where $V(\mathcal{G})$ and $L(\mathcal{G})$ are the sets of nodes/vertices and links/edges. Without loss of generality, we assume that graph $\mathcal{G}$ is connected, since different connected components of the network can be monitored separately. We view the network $\mathcal{G}$ as a multigraph where: (1) there could be multiple links between two nodes; and (2) a link could loop at a node. Even though a real-life network need not be modeled as a multigraph, viewing it so significantly simplifies the understanding of the proofs and notations employed [7]. We use $l \in L(\mathcal{G})$ to denote a link in the graph. The two edge nodes (i.e., endpoints) of a link l is denoted as $u(l)$ and $v(l)$. We assume that the links are symmetric, i.e., link metrics are the same in both directions. A subset of vertices in $V(\mathcal{G})$ can be assigned as monitors, which can initiate/collect end-to-end measurements on paths or cycles.

We make the same assumption as in [3] about the measurement paths or cycles. Measurement paths start and end at distinct nodes while measurement cycles start and end at the same monitor. We allow the use of nonsimple cycles and paths. In this paper, nonsimple cycles (paths) are those where a node may appear more than once, however a link will not [3]. Monitors can control their measurement paths or cycles. In

Table 1: Notations used in this paper

Symbol	Description
$\mathcal{G}$	The graph representation of a network.
$\mathcal{G}'$	The transformed graph of $\mathcal{G}$ by merging all monitors.
$\mathcal{G}_{f_i}$	The graph under a failure scenario f_i.
$V(\mathcal{G})$, $L(\mathcal{G})$	Set of vertices/links in graph $\mathcal{G}$.
$\mathcal{B}, \mathcal{T}, \mathcal{C}^k$	2-edge-connected-component, 3-edge-connected-component, or k-edge-connected component in $\mathcal{G}$.
$\mathcal{B}_m, \mathcal{T}_m, \mathcal{C}^k_m$	2-edge-connected-component, 3-edge-connected-component, or k-edge-connected component in which the monitor is located.
Type 1 component	3-edge-connected component $\mathcal{T}$: $\mathcal{T} \neq \mathcal{T}_m$ and $\mathcal{T}$ remains 3-edge-connected after the removal of two paths connecting $\mathcal{T}$ and the monitor m.
Type 2 component	3-edge-connected component $\mathcal{T}$: $\mathcal{T} \neq \mathcal{T}_m$ and $\mathcal{T}$ is not 3-edge-connected after the removal of two paths connecting $\mathcal{T}$ and the monitor m.
Type 3 component	3-edge-connected component in $\mathcal{T}'$ where $\mathcal{T}'$ is a Type 2 component $\mathcal{T}$ without the two paths connecting $\mathcal{T}$ and the monitor m.

Section VII, we discuss how controllable measurement paths or cycles can be implemented in existing IP networks.

Let $n = |L|$ be the number of links in $\mathcal{G}$, $\{l_i\}_{i=1}^n$ be the set of links (and also link metrics[1]), $\mathbf{l} = (l_1, ..., l_n)^T$ be the column vector of all link metrics (which are *unknown*), $\{p_i\}_{i=1}^\gamma$ be the set of γ measurement paths or cycles (and also path metrics which are *observed*[2]), and $\mathbf{p} = \{p_1, ..., p_\gamma\}$ be the column vector of all path metrics. Given path measurements, we have the following linear system:

$$\mathbf{Rl} = \mathbf{p}, \tag{1}$$

where $\mathbf{R} = (R_{ij})$ is a $\gamma \times n$ *measurement matrix*, with $R_{ij} \in \{0, 1\}$ indicating whether link l_j is in the measurement path or cycle p_i.

We say a link l_i is identifiable (or 0-identifiable) for a given monitor placement in $\mathcal{G}$ if there exists a measurement matrix with which the link metric l_i can be uniquely determined. The network $\mathcal{G}$ is completely identifiable if all links in $\mathcal{G}$ are identifiable.

When link failures occur, a link may not be identifiable since the original measurement paths containing the failed links may not exist. We consider failure scenarios in which any k links fail. There can be $\binom{n}{k}$ failure scenarios. For a particular failure scenario f_i, we can represent the network as a graph $\mathcal{G}_{f_i}$ which is transformed from $\mathcal{G}$ by removing the failed links.

We say a link l_i is k-identifiable for a given monitor placement in $\mathcal{G}$ if it is still identifiable for the same monitor placement in all failure scenarios $\mathcal{G}_{f_i}$ in which l_i itself does not fail. The network $\mathcal{G}$ is k-identifiable if all links in $\mathcal{G}$ are k-identifiable. It is worth noting that a (k+1)-identifiable link must be k-identifiable. Hence, a link is more robust with a larger value of k.

Table 1 summarizes the graph-theoretical notations used in this paper.

B. Objective

The objective of this paper are two-fold:

- to determine the k-identifiable links in $\mathcal{G}$ for a given monitor placement,
- to find an optimal placement of a given number of monitors to maximize the number of k-identifiable links.

[1]It should be clear from the context whether we use l_i to represent a link or its link metric.

[2]It should be clear from the context whether we use p_i to represent a path or its path metric.

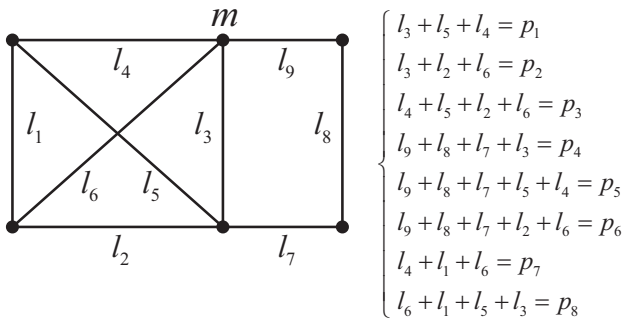


Fig. 1: An illustrative example. m is the monitor. The equations in the right of the figure correspond to eight possible measurement cycles conducted at m. In this example, l_3 is 1-identifiable and l_4 is 0-identifiable.

C. Illustrative Example

Fig. 1 shows a network with six nodes and nine links (l_1 to l_9). We conduct measurements with a single monitor m using only cycles that start and end at m. The equations listed in the right of Fig. 1 correspond to eight possible measurements with cycles. For illustrative purpose, we focus on two links l_3 and l_4.

For l_4, it can be uniquely determined as $(p_1 + p_7 - p_8)/2$. Hence, it is identifiable or 0-identifiable. l_4 can no longer be uniquely determined when l_5 fails. Hence, l_4 is 0-identifiable but not 1-identifiable.

For l_3, it can be uniquely determined as $(p_1 + p_2 - p_3)/2$. Hence, it is identifiable or 0-identifiable. We show that l_3 is 1-identifiable since it can still be uniquely determined when any of the other links fails.

- When one link in $\{l_1, l_7, l_8, l_9\}$ fails, l_3 is still identifiable since the failure does not affect the measurement cycles p_1, p_2, p_3.
- When one link in $\{l_4, l_5\}$ fails, l_3 can still be identified by measurement cycles p_2, p_4, p_6 (which do not contain the failed link), i.e., $l_3 = (p_2 + p_4 - p_6)/2$.
- Similarly, when one link in $\{l_2, l_6\}$ fails, l_3 can be uniquely determined as $(p_1 + p_4 - p_5)/2$.

Note that l_3 is not 2-identifiable since it cannot be uniquely determined when links l_4 and l_9 simultaneously fail.

IV. Sufficient and Necessary Conditions for k-identifiability

A. Preliminaries

We first introduce necessary concepts and previous results, which are used in this paper.

- A graph is *connected* when there exists at least one path from any vertex to any other vertex in the graph.
- Generally, a graph is *k-edge-connected* when the graph is still connected whenever fewer than k links are removed, or equivalently, a graph is *k-edge-connected* when any two vertices in the graph have at least k disjoint paths.

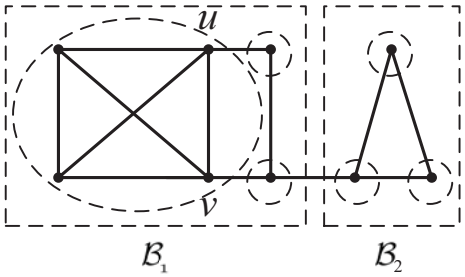


Fig. 2: An example graph with two 2-edge-connected components $\mathcal{B}_1$ and $\mathcal{B}_2$. A dotted rectangle denotes a 2-edge-connected component and a dotted circle denotes a 3-edge-connected component.

- For two vertices $u, v \in V(\mathcal{G})$, they are k-edge-connected, denoted as $u \sim_{\mathcal{G}} v$, if and only if $u = v$ or there exists k edge-disjoint paths between u and v.
- According to [16], $\sim_{\mathcal{G}}$ is an equivalence relation in $V(\mathcal{G})$. For $v \in V(\mathcal{G})$, the equivalence class of v with respect to $\sim_{\mathcal{G}}$ is a k-edge-connected component of $\mathcal{G}$, denoted as C^k, which is still connected whenever fewer than k links in $\mathcal{G}$ are removed.
- A link is *inside* a component when its two edge nodes belong to the same component. A link is *outside* a component when there is at least one edge node does not belong to the component.

Fig. 2 shows an example with two 2-edge-connected components $\mathcal{B}_1$ and $\mathcal{B}_2$. $\mathcal{B}_1$ is further partitioned into three 3-edge-connected components. We should note that each single node of $\mathcal{B}_2$ is a 3-edge-connected component too. It is also worth noting that $\{u, v\}$ is a 4-edge-connected component and there exists 4 disjoint paths between u and v, of which three paths are outside the component $\{u, v\}$.

Gopalan *et al.* [3] give a sufficient condition for 0-identifiability with a single monitor.

Theorem IV.1 ([3])**.** *If $\mathcal{G}$ is a 3-edge-connected graph, all of its link metrics are identifiable by assigning one monitor.*

With multiple monitors, Gopalan *et al.* also introduce a merge operation, i.e., merging all the monitors into a single monitor. If the original graph $\mathcal{G}$ with multiple monitors is not 3-edge-connected but the resultant graph after merging $\mathcal{G}'$ is 3-edge-connected, it is suffice to identify all links.

We formally define the merge operation on multiple monitors as follows.

Definition IV.1 (Merge operation)**.** *A merge operation on a set of monitors $\mathcal{M}$ in a graph $\mathcal{G}$ transforms the original graph $\mathcal{G}$ into a resultant graph $\mathcal{G}'$, with $V(\mathcal{G}') = \{V(\mathcal{G}) - \mathcal{M} + \{m\}\}$ where m is the merged monitor, and there is a one-to-one mapping between edges in $\mathcal{G}$ and edges in $\mathcal{G}'$: $l \in \mathcal{G} \mapsto l' \in \mathcal{G}'$ with two edge nodes of l' being:*

$$(u', v') = \begin{cases} (u, v) \text{ if } u, v \notin \mathcal{M} \\ (u, m) \text{ if } u \in \mathcal{M} \text{ and } v \notin \mathcal{M} \\ (m, m) \text{ if } u, v \in \mathcal{M} \end{cases} \quad (2)$$

where u and v are two edge nodes of l. Fig. 3 illustrates the merge operation by an example. The original graph is given in Fig. 3(a) and the graph after merge is given in Fig. 3(b).

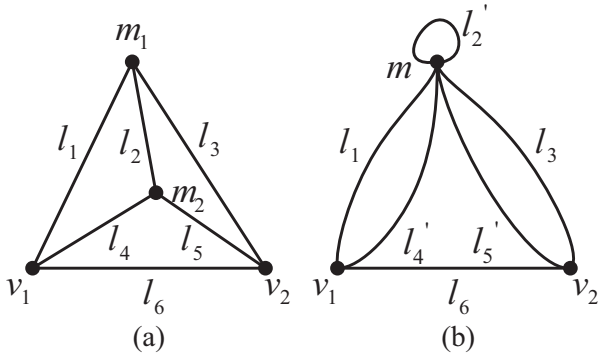


Fig. 3: An example which illustrates the merge operation: (a) the original graph $\mathcal{G}$, (b) the graph after merge $\mathcal{G}'$.

We see that all links incident to m_1 and m_2 are transformed to links incident to the merged monitor m. The merged graph $\mathcal{G}'$ is also a multigraph which may contain self-loops or multi-edges.

B. Main Results

Theorem IV.2. *For a general network $\mathcal{G}$ with specified monitors, we obtain $\mathcal{G}'$ by merging all the monitors into a single monitor. Such a merge operation preserves the k-identifiability of a link, i.e., for a particular link $l \in \mathcal{G}$, it is k-identifiable if and only if its corresponding link $l' \in \mathcal{G}'$ is k-identifiable.*

Theorem IV.3 (0-identifiability)**.** *In a general graph $\mathcal{G}'$ with only one monitor, a link is 0-identifiable if and only if it resides in the $\mathcal{T}_m$ or Type 1, Type 3 components[3] of the graph.*

Theorem IV.4 (k-identifiability, $k > 0$)**.** *In a general graph $\mathcal{G}'$ with only one monitor, a link is k-identifiable if and only if it resides in the (k+3)-edge connected components of the (k+2)-edge connected component in which the monitor is located.*

The above theorems give direct guidance on how to determine the k-identifiable links in a graph $\mathcal{G}$. First, we merge the monitors into a single monitor and obtain the merged graph $\mathcal{G}'$. As indicated by Theorem IV.2, this operation preserves the k-identifiability of each link. Second, we divide graph $\mathcal{G}'$ into corresponding components and use Theorem IV.3 and Theorem IV.4 to determine the k-identifiable links.

C. Proof of Theorem IV.2

Note that there is a one-to-one mapping from edges in $\mathcal{G}$ to edges in $\mathcal{G}'$. For each link, if it is identifiable by selecting a set of measurement paths or cycles in $\mathcal{G}$, the corresponding link in $\mathcal{G}'$ is also identifiable by selecting the same set of mapped cycles and vice versa. For k-identifiability, the result also holds since for each failure graph $\mathcal{G}_f$, there also exists a merged graph $\mathcal{G}'_f$. A link in $\mathcal{G}_f$ is identifiable if and only if the corresponding link in $\mathcal{G}'_f$ is also identifiable.

D. Proof of Theorem IV.3

For 0-identifiability, we first divide the merged graph $\mathcal{G}'$ into 2-edge-connected components. We further divide $\mathcal{B}_m$ (i.e., the 2-edge-connected component in which the monitor is located) into 3-edge-connected components. We use $\mathcal{T}_m$ to denote the 3-edge-connected component in which the monitor

[3]See Table 1 for the definitions of $\mathcal{T}_m$ and Type 1, Type 3 components.

is located. The above operations classify each link in $\mathcal{G}'$ into three categories:

- **Category 1**: links outside $\mathcal{B}_m$.
- **Category 2**: links between the 3-edge-connected components of $\mathcal{B}_m$.
- **Category 3**: links inside 3-edge-connected components of $\mathcal{B}_m$.

Theorem IV.5. *Links in **Category 1** are unidentifiable.*

Proof: See [17]. ■

Theorem IV.6. *Links in **Category 2** are unidentifiable.*

Proof: See [17]. ■

Next, we would like to show the identifiability of **Category 3** links. We further divide **Category 3** links into several sub-categories. The first sub-category, **Category 3a**, includes links inside $\mathcal{T}_m$. The remaining sub-categories include links inside $\mathcal{T}$: $\mathcal{T}$ is a 3-edge-connected component of $\mathcal{B}_m$ and it does not contain the monitor node.

Before introducing the sub-categories of links, it is necessary to introduce different types of $\mathcal{T}$. There are only two disjoint paths that connect $\mathcal{T}$ with the monitor by viewing $\mathcal{T}$ as a virtual node (e.g., p_x and p_y in Fig. 4).

- Type 1: $\mathcal{T}$ remains 3-edge-connected after the removal of the two paths
- Type 2: $\mathcal{T}$ is not 3-edge-connected after the removal of the two paths.

Two nodes in Type 2 component are *at least* 2-edge-connected after the removal of the two paths, since the concatenation of the two paths can *at most* provide one disjoint path between two vertices in Type 2 component. Two nodes would remain 3-edge-connected if the removal of the two paths has no impact on the connectivity of the two nodes. We further introduce the Type 3 component to denote the 3-edge-connected component in the Type 2 component after the removal of the two paths.

Based on the different types of component, we have the following four sub-categories for **Category 3** links:

- **Category 3a**: links inside $\mathcal{T}_m$.
- **Category 3b**: links inside Type 1 components.
- **Category 3c**: links inside Type 3 components in Type 2 components.
- **Category 3d**: links inside Type 2 components but outside Type 3 components.

We first show the identifiability of **Category 3a** links by extending Theorem IV.1 by considering 3-edge-connected component instead of 3-edge-connected graph.

Theorem IV.7. *If $\mathcal{T}$ is a 3-edge-connected component, all links inside $\mathcal{T}$ are identifiable by assigning one monitor in $\mathcal{T}$. In other words, **Category 3a** links are identifiable.*

Proof: See [17]. ■

We proceed to investigate the identifiability of the remaining links, i.e., links in $\mathcal{T}$ where $m \notin \mathcal{T}$, and both m and $\mathcal{T}$ belong to $\mathcal{B}_m$. We can find two disjoint paths starting from m to any vertex in $\mathcal{T}$, say v_0. We denote the first vertex in $V(\mathcal{T})$ on the first path as v_1, and the first vertex in $V(\mathcal{T})$ on the second path as v_2. We use p_x to denote the subpath connecting m and v_1, and we use p_y to denote the subpath connecting m and v_2.

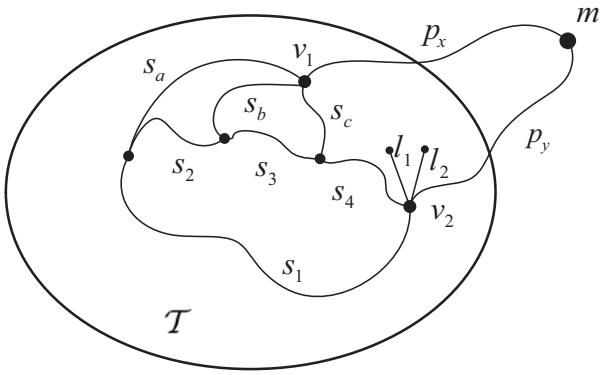


Fig. 4: A 3-edge-connect component $\mathcal{T}$ in the graph. The monitor m is outside $\mathcal{T}$.

Fig. 4 shows the network topology for this case. In order to show the identifiability of **Category 3b** links, we first prove the following lemma.

Lemma IV.8. *For the network topology shown in Fig. 4, the sum $(p_x + c + p_y)$[4] can be uniquely determined for any cycle c that starts and ends at v_1 (or v_2).*

Proof: See [17]. ■

Theorem IV.9. *Links in **Category 3b** are identifiable.*

Proof: See [17]. ■

Theorem IV.10. *Links in **Category 3c** are identifiable, while links in **Category 3d** are unidentifiable.*

Proof: See [17]. ■

E. Proof of Theorem IV.4

Let $\mathcal{C}_m^{k+2}$ be the (k+2)-edge-connected component where the monitor m is located, and $\mathcal{C}_m^{k+3}$ be the (k+3)-edge-connected component where the monitor m is located.

Sufficient part. We consider a link l which is incident to u and v. If l is in a (k+3)-edge-connected component of $\mathcal{C}_m^{k+2}$, we will prove that in an arbitrary failure graph $\mathcal{G}_f$ by removing k links in the original graph, l is either in $\mathcal{T}_m$ or in Type 1, Type 3 component of $\mathcal{G}_f$. Thus, according to Theorem IV.3, l is identifiable in $\mathcal{G}_f$ and k-identifiable in the original graph. There are two cases for l.

(1) l resides in $\mathcal{C}_m^{k+3}$. According to the definition of k-edge-connected component, after removing any k links in the original graph, u, v and m belong to the same 3-edge-connected component in $\mathcal{G}_f$, indicating that $l \in \mathcal{T}_m$ of $\mathcal{G}_f$. According to Theorem IV.3, l is identifiable in $\mathcal{G}_f$ and is k-identifiable in $\mathcal{G}$.

(2) l resides in other (k+3)-edge-connected component $\mathcal{C}^{k+3}$ of $\mathcal{C}_m^{k+2}$. There are (k+3) disjoint paths between u and v, and (k+2) disjoint paths which connect $\mathcal{C}^{k+3}$ and m. We use S_i to denote a set of (k+2) disjoint paths whose removal can disconnect $\mathcal{C}^{k+3}$ and m. Note that different i's denote different sets. For a particular i, there are at most $\lfloor \frac{k+2}{2} \rfloor = \lfloor \frac{k}{2} \rfloor + 1$ disjoint paths between u and v containing links in S_i (see Fig. 5 for an example). This is because if a path connecting u and v includes a link in S_i, it must include another link in

[4]We use "+" to denote the concatenation of two paths, or the sum of the two path metrics.

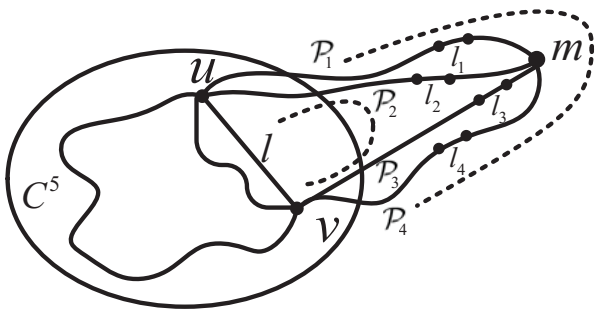


Fig. 5: An example for the proof of Theorem IV.4. k=2 and $S_i = \{l_1, l_2, l_3, l_4\}$. There are two u-to-v paths (e.g., p_1+p_4 and p_2+p_3) containing links in S_i. In addition, there are three u-to-v paths containing no links in S_i.

S_i. Otherwise, the existence of another link outside S_i implies that the removal of S_i will not disconnect $\mathcal{C}^{k+3}$ and m. There are two cases for the removal of k links.

(2a) $\nexists i$: k links $\in S_i$. In this case, there exist at least 3 disjoint paths connecting $\mathcal{C}^{k+3}$ and m in $\mathcal{G}_f$. Therefore, $\mathcal{C}^{k+3}$ and m belong to the same 3-edge-connected component in $\mathcal{G}_f$, indicating that $l \in \mathcal{T}_m$ of $\mathcal{G}_f$. According to Theorem IV.3, l is identifiable in $\mathcal{G}_f$ and is k-identifiable in $\mathcal{G}$.

(2b) $\exists i$: k links $\in S_i$. For u and v, there are at least $(k+3)-(\lfloor \frac{k}{2} \rfloor + 1) = \lceil \frac{k}{2} \rceil + 2$ disjoint paths containing no links in S_i. The number of these paths is at least 3 when $k > 0$. Since these paths do not contain the k links to be removed from $\mathcal{G}$, the number of these paths between u and v in $\mathcal{G}_f$ is at least 3. There are at least two disjoint paths connecting $\mathcal{C}^{k+3}$ and m in $\mathcal{G}_f$, forming at least one u-to-v path which contains links in S_i and remains in $\mathcal{G}_f$. Hence in $\mathcal{G}_f$, u and v are 3-edge-connected after the removal of two paths connecting $\mathcal{C}^{k+3}$ and m. According to the definition of Type 1/2/3 component, u and v are in the same Type 1 or Type 3 component of $\mathcal{G}_f$. According to Theorem IV.3, l is identifiable in $\mathcal{G}_f$ and is k-identifiable in $\mathcal{G}$.

Necessary part. If l is not in a (k+3)-edge-connected component of $\mathcal{C}_m^{k+2}$, there are two cases.

(1) l is outside $\mathcal{C}_m^{k+2}$. Suppose $u \notin V(\mathcal{C}_m^{k+2})$. There must exist a set $\mathcal{I}$ of $k+1$ links which can disconnect u and m. Otherwise, u and m will reside in the same (k+2)-edge-connected component. We consider the failure scenario $\mathcal{G}_f$ in which k links in $\mathcal{I}$ are failed. In graph $\mathcal{G}_f$, u is not in $\mathcal{B}_m$. Hence, according to Theorem IV.5, l is not identifiable in the failure graph G_f. According to the definition, l is not k-identifiable in the original graph $\mathcal{G}$.

(2) l is in $\mathcal{C}_m^{k+2}$ but not in a (k+3)-edge-connected component of $\mathcal{C}_m^{k+2}$. There must be a set $\mathcal{I}$ of $k+2$ links which can disconnect u and v. Otherwise, u and v will reside in the same (k+3)-edge-connected component. We consider the failure scenario $\mathcal{G}_f$ in which k links in $\mathcal{I}$ fail. In graph $\mathcal{G}_f$, u and v do not reside in the same 3-edge-connected component. Hence, according to Theorem IV.6, l is not identifiable in the failure graph $\mathcal{G}_f$. According to the definition, l is not k-identifiable in the original graph $\mathcal{G}$.

V. Identification of k-identifiable Links

The theoretical results in Section IV give us direct guidance on how to devise an efficient algorithm for identifying all k-identifiable links in a general graph $\mathcal{G}$. We first merge the specified monitors into a single monitor and obtain a new graph $\mathcal{G}'$. We then divide the graph $\mathcal{G}'$ into (k+2)-edge-connected components [14], and in the (k+2)-edge-connected component where the monitor is located ($\mathcal{C}_m^{k+2}$), we further divide it into Type 1 and Type 3 components if $k = 0$, or, (k+3)-edge-connected components if $k > 0$. According to Theorem IV.3 and Theorem IV.4, the links in Type 1 and Type 3 components of $\mathcal{C}_m^{k+2}$ are k-identifiable when $k = 0$; the links in (k+3)-edge-connected components of $\mathcal{C}_m^{k+2}$ are k-identifiable when $k > 0$. The remaining links are *not* k-identifiable.

Algorithm 1 Identification of k-identifiable links (IDK)

Input: Graph $\mathcal{G}$, monitor set $\mathcal{M} = \{v_1, v_2, v_3, ..., v_\kappa\}$, parameter k.
Output: Set of k-identifiable links $\mathcal{I}$.

```
if |M| > 1 then
    merge monitors in M to a single monitor m, and the new graph after merge is denoted as G'.
for each l'_i: l'_i is a self loop at m in G' do
    I ← I ∪ {l_i}, where l_i is l'_i's corresponding link in G
    remove l'_i from G'
divide G' into (k+2)-edge-connected components: C_1^{k+2}, C_2^{k+2}, ..., C_β^{k+2}.
if k = 0 then
    choose B_m where the monitor is located, and further divide it into T_m and Type 1, Type 3 components: C_1, C_2, ..., C_δ.
    for i: 1 ≤ i ≤ δ do
        for each l'_j ∈ C_i do
            I ← I ∪ l_j where l_j is l'_j's corresponding link in G.
else
    choose C_i^{k+2} where m ∈ C_i^{k+2}, and divide it into (k+3)-edge-connected components: C_1^{k+3}, C_2^{k+3}, ..., C_θ^{k+3}.
    for i: 1 ≤ i ≤ θ do
        for each l'_j ∈ C_i^{k+3} do
            I ← I ∪ l_j where l_j is l'_j's corresponding link in G.
return I
```

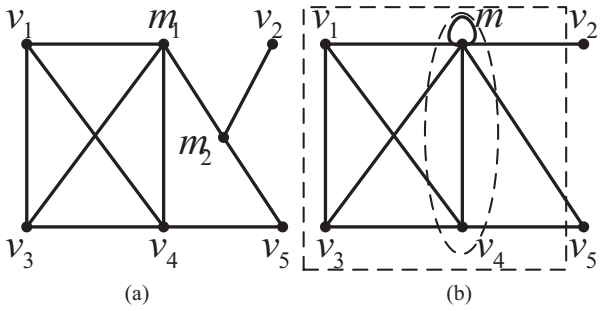


Fig. 6: An example network topology to illustrate how Algorithm 1 works. (a) the original graph with two monitors m_1 and m_2. (b) the merged graph with a single monitor m.

Algorithm 1 shows the pseudocode of our algorithm. The inputs of the algorithm are the graph $\mathcal{G}$, the monitor set $\mathcal{M}$ and the parameter k. The output of the algorithm is the link set $\mathcal{I}$ in which links are k-identifiable.

To illustrate how Algorithm 1 works, we use an example network shown in Fig. 6 to explain how we identify 1-identifiable links (i.e., $k = 1$). Fig. 6(a) shows the original network topology with two monitors m_1 and m_2.

- Lines 1-2: The algorithm judges whether the number of monitors is larger than one. If it is, the algorithm performs a merge operation (see Definition IV.1) and the new graph $\mathcal{G}'$ is obtained. For the original graph $\mathcal{G}$ shown in Fig. 6(a), the merged graph is shown in Fig. 6(b).
- Lines 3-5: The algorithm adds self-loop edges located at the merged monitor m into the the set of k-identifiable links. These links are obviously k-identifiable since these links are directly observable. For the example shown in Fig. 6(b), the algorithm adds m_1m_2 (corresponds to mm) to the set $\mathcal{I}$.
- Line 6: The algorithm divides $\mathcal{G}'$ into (k+2)-edge-connected components. For the example shown in Fig. 6(b), the algorithm divides it into 3-edge-connected components: $V(\mathcal{C}_1^3) = \{v_2\}$, $V(\mathcal{C}_2^3) = \{v_5\}$, $V(\mathcal{C}_3^3) = \{v_1, v_3, v_4, m\}$.
- Lines 7-11: The algorithm judges whether k equals 0. If it is, the algorithm divides (k+2)-edge-connected component where the monitor is located into $\mathcal{T}_m$ and Type 1, Type 3 components. Then, the algorithm adds links inside these components into the set of k-identifiable links.
- Lines 12-13: If k does not equal 0, the algorithm further divides the (k+2)-edge-connected component where the monitor is located into (k+3)-edge-connected components. For the example shown in Fig. 6(b), the algorithm further divides $\mathcal{C}_3^3$ where the monitor is located into 4-edge-connected components: $V(\mathcal{C}_1^4) = \{v_1\}$, $V(\mathcal{C}_2^4) = \{v_3\}$, $V(\mathcal{C}_3^4) = \{v_4, m\}$.
- Lines 14-16: According to Theorem IV.3, the algorithm adds links in $\mathcal{C}_1^{k+3}, \cdots, \mathcal{C}_\delta^{k+3}$ into the set of k-identifiable links. For the example shown in Fig. 6(b), there is only one 1-identifiable link m_1v_4 for the original graph since there is only one link mv_4 in components $\mathcal{C}_1^4$, $\mathcal{C}_2^4$, or $\mathcal{C}_3^4$.

The complexity of the algorithm mainly depends on the complexity of dividing the graph into different kinds of components. There are algorithms for dividing a graph into 2-edge-connected components or 3-edge-connected components with complexity of $O(|L|)$ [16]. There are algorithms for dividing a graph into k-edge-connected components with complexity of $O(k|V|^4)$ [14]. Therefore, Algorithm 1 has a time complexity of $O(|L|)$ when $k = 0$ and $O(k|V|^4)$ when $k > 0$.

VI. Monitor Placement for Maximal k-identifiability

The algorithm shown in the previous section allows us to determine the set of k-identifiable links in an arbitrary network with a given placement of monitors. A more practical question is: given a number of κ monitors, how to place these monitors so that the number of k-identifiable links can be maximized?

This problem is, unfortunately, NP-complete for general network topologies. Therefore, we devise an efficient greedy algorithm for this problem. The key idea of our algorithm is to incrementally place monitors in (k+3)-edge-connected components of the graph since a single monitor is sufficient to identify all links in this component in the presence of k link failures. We choose monitors so that the number of k-identifiable links is maximized in each step.

Algorithm 2 shows the pseudocode of the greedy algorithm.

Algorithm 2 Monitor placement for maximal k-identifiability (MPK)

Input: Graph $\mathcal{G}$, number of monitors κ, parameter k
Output: Monitor set $\mathcal{M}$.
1: $\mathcal{M} \leftarrow \phi$
2: Initialize the candidate set S so that it contains one randomly chosen node from each (k+3)-edge-connected component.
3: $n \leftarrow 0$
4: **for** each v: $v \in S$ **do**
5: $\quad \mathcal{M}' \leftarrow \{v\}$
6: $\quad S' \leftarrow S \setminus v$
7: $\quad$ **for** j: $1 \leq j \leq \kappa - 1$ **do**
8: $\quad\quad v'_{\max} \leftarrow \arg\max_{v' \in S'}(\text{IDK}(\mathcal{G}, \mathcal{M}' \cup \{v'\}, k))$
9: $\quad\quad \mathcal{M}' \leftarrow \mathcal{M}' \cup v'_{\max}$
10: $\quad\quad \mathcal{S}' = \mathcal{S}' \setminus v'_{\max}$
11: $\quad$ **if** $|\text{IDK}(\mathcal{G}, \mathcal{M}', k)| > n$ **then**
12: $\quad\quad \mathcal{M} \leftarrow \mathcal{M}'$
13: $\quad\quad n \leftarrow |\text{IDK}(\mathcal{G}, \mathcal{M}', k)|$
14: return $\mathcal{M}$

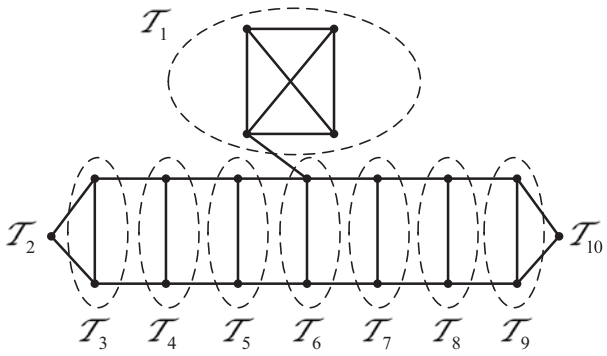


Fig. 7: An example network topology to illustrate how Algorithm 2 works.

The algorithm selects κ monitors out of the candidate set S which comprises of one randomly chosen node from each (k+3)-edge-connected component. Note that the algorithm enumerates all possible placement of the first monitor (the first loop) so that we can find a better solution when the placement of the first monitor in multiple different (k+3)-edge-connected components yields the same number of k-identifiable links. When the first monitor is selected, the algorithm incrementally selects the remaining $\kappa - 1$ monitors so that the number of k-identifiable links is maximized in each iteration of the second loop. The algorithm employs Algorithm 1 (the IDK() procedure) developed in the previous section to determine the number of k-identifiable links for a given network topology and given placements of monitors.

We use the example shown in Fig. 7 to illustrate how Algorithm 2 works. The network topology contains two 2-edge-connected components and ten 3-edge-connected components. (1) If we want to maximize the number of identifiable (i.e., 0-identifiable) links with only one monitor, Algorithm 2 will try to place the monitor in each of the ten 3-edge-connected components. The algorithm can find the optimal solution with the monitor placed in $\mathcal{T}_1$ since this placement yields the maximum number of identifiable links (i.e., 6 links) among all ten possible placements in each 3-edge-connected component. (2) If we want to maximize the number of identifiable links with two monitors, Algorithm 2 will enumerate ten possible

placement of the first monitor. If the first monitor is placed in $\mathcal{T}_1$, the maximum number of identifiable links is 19 with the second monitor placed in $\mathcal{T}_2$. If the first monitor is placed in $\mathcal{T}_2$, the maximum number of identifiable links is 23 with the second monitor is placed in $\mathcal{T}_{10}$. We can see that Algorithm 2 can also find the optimal solution in this case. Note that the enumeration of the first monitor in different components is necessary in this case. Otherwise, we will end up with the first monitor placed in $\mathcal{T}_1$ and the number of identifiable links is only 19.

We can see that Algorithm 2 works correctly for the example shown in Fig. 7. We find that it can find the optimal solution in certain topologies. For general network topologies, we experimentally find that it achieves near-optimal performance.

Algorithm 2 executes the outer loop $|S|$ times. In each iteration, it executes statement 8 $(\kappa - 1)$ times. Statement 8 invokes Algorithm 1 $|S| - j$ times. Hence, the complexity of Algorithm 2 is $O(\kappa|S|^2|L|)$ for $k = 0$ and $O(\kappa|S|^2k|V|^4)$ for $k > 0$.

VII. Discussion

Performance monitoring for the Internet is an important topic of research. The key to our work is *controllable routing*. Many approaches such as source routing, MPLS [18], ATM, can enable controllable routing. We now discuss how to perform network tomography in real networks using controllable measurement paths and cycles.

Recently, Hu *et al.* introduces XPath [19], a simple, practical and readily-deployable way to implement explicit path control, using existing commodity switches. At its core, XPath explicitly identifies an end-to-end path with a path ID and pre-installs all the desired path IDs between any source-destination pairs into IP LPM tables using a two-step compression algorithm. When a XPath node sends a packet to a destination, it first queries the XPath controller for the path ID. Every node on the routing path forwards the packet according to its routing table which maintains a one-to-one mapping between path ID and the output port. Controllable measurement paths can be directly supported by XPath.

For IP networks, in order to enable a monitor to send packets along a cycle, we may employ IP-in-IP tunneling [3]. We can reuse the method developed in [3] to setup undirected spanning trees for establish the measurement paths and cycles. The monitor m may send a packet on the first tree to node n and node n forwards that packet back to node m on the second tree. For these spanning trees, every node maintains a routing table entry for every other node. The monitor m may create a packet that is destined to itself to be routed over the second tree, which is then encapsulated in another header that is destined to node n to be routed on the first tree.

VIII. Evaluation

In this section, we evaluate our algorithms on real ISP network topologies. We first introduce the evaluation methodology. Then we present the evaluation results.

A. Methodology

We collect real network topologies for the evaluation. We use the ISP topologies collected by the Rocketfuel project [20], which represents physical connections between backbone/gateway routers of several ISPs around the world. We select three ISPs, i.e., AS1755, AS3257, and AS3967.

Table 2: Parameters of different network topologies

Topology	$\|L\|$	$\|V\|$	$N_{\mathcal{B}}$	$N_{\mathcal{T}}$	$\|\mathcal{B}_{\max}\|$	$\|\mathcal{T}_{\max}\|$	L/V
AS1755	381	172	27	60	145	113	2.21
AS3257	405	248	8	170	240	79	1.63
AS3967	443	215	44	100	166	115	2.06

Table 2 shows the detailed information of the topologies, where $N_{\mathcal{B}}, N_{\mathcal{T}}$ denote the number of two- and there edge-connected components respectively, and $|\mathcal{B}_{\max}|, |\mathcal{T}_{\max}|$ denote the number of nodes in the maximum two- and there edge-connected components respectively.

We use the ratio of k-identifiable links (i.e., the number of k-identifiable links divided by the number of links in the network) to compare different monitor placement algorithms in real network topologies with a given number of k. We compare three algorithms:

- Random Monitor Placement (RMP). The algorithm randomly selects κ nodes as monitors and examines the number of k-identifiable links using Algorithm 1. The algorithm repeats the above process w times and returns the solution with the maximum number of k-identifiable links.
- Enhanced RMP (ERMP). The algorithm randomly selects κ nodes as monitors from a candidate set. The candidate set comprises of one randomly chosen node from each (k+3)-edge-connected component. The algorithm also repeats the above process several times and returns the solution with the maximum number of k-identifiable links.
- MPK. This is our proposed algorithm. In the evaluation, we remove the enumeration for the first monitor in order to speed up its execution.

For a fair comparison, we select w such that all three algorithms have the same execution overhead.

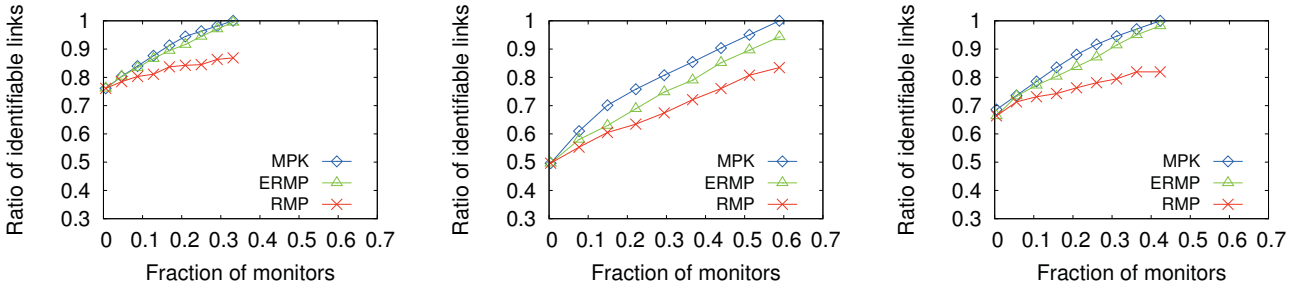
B. Results for ISP topologies

Fig. 8 shows the ratios of identifiable links for three different ISP topologies with increasing number of monitors. It is obvious that the ratio of identifiable links increases with the increase of the number of monitors. Our algorithm achieves the best performance in all three topologies compared with RMP and ERMP. Although complete identification of the networks requires a substantial fraction of monitors, many links can be identified with a few monitors, e.g., MPK can identify about 60% links with 10% monitors. It is also worth noting that different topologies require different number of monitors to achieve the same ratio of identifiable links. For example, AS1755 requires 35% monitors for identifying all links while AS3257 requires more than 50% monitors. This is because AS1755 has a larger node degree.

Fig. 9 shows the ratios of 1-identifiable links for the ISP topologies with increasing number of monitors. We see that the required number of monitors for achieving 1-identifiability is larger than that for achieving 0-identifiability. Our algorithm still achieves the best performance in all three topologies when $k = 1$.

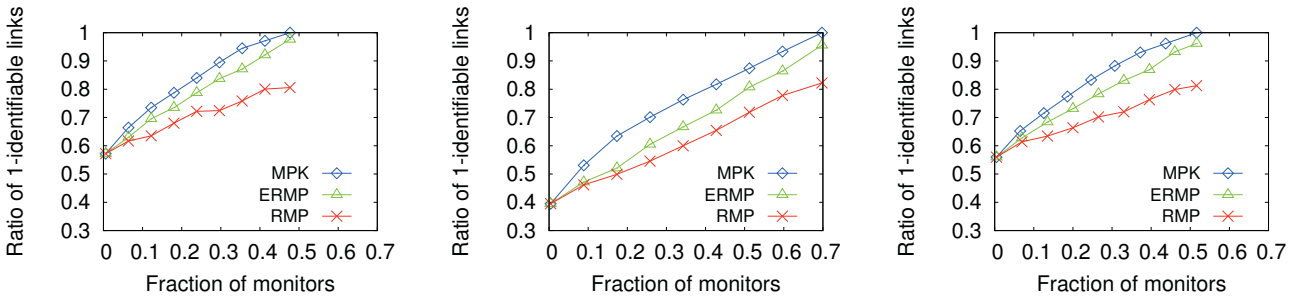
IX. Conclusion

In this paper, we propose topological conditions to ensure identifiability in the presence of any k link failures, using measurement paths and cycles. We propose a novel concept called k-identifiability to cope with link failure during network



(a) AS1755 (b) AS3257 (c) AS3967

Fig. 8: Ratios of identifiable links in three ISP topologies ($k = 0$)



(a) AS1755 (b) AS3257 (c) AS3967

Fig. 9: Ratios of 1-identifiable links in three ISP topologies ($k = 1$).

measurement. We propose sufficient and necessary topological conditions to ensure k-identifiability. Based on the established theoretical foundations, we propose two efficient polynomial-time algorithms for addressing two closely related questions. (1) Given a network with specified monitors, which links are k-identifiable? (2) Given a network and κ monitors, where should these monitors be placed such that the number of k-identifiable links is maximized? Simulation results on real ISP network topologies show the effectiveness of our algorithm.

There are multiple directions to explore. First, we would like to develop algorithms to find the measurement paths for network tomography. Second, we would like to extend our algorithm by considering more practical network settings.

REFERENCES

[1] L. Ma, T. He, K. K. Leung, A. Swami, and D. Towsley, "Identifiability of link metrics based on end-to-end path measurements," in *Proc. of ACM IMC*, 2013.

[2] D. Turner, K. Levchenko, A. C. Snoeren, and S. Savage, "California fault lines: Understanding the causes and impact of network failures," in *Proc. of ACM SIGCOMM*, 2010.

[3] A. Gopalan and S. Ramasubramanian, "On identifying additive link metrics using linearly independent cycles and paths," *IEEE/ACM Trans. on Networking*, vol. 20, no. 3, pp. 906–916, 2012.

[4] Y. Vardi, "Network tomography: Estimating source-destination traffic intensities from link data," *Journal of the American Statistical Association*, vol. 91, no. 433, pp. 365–377, 1996.

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

[6] Q. Zheng and G. Cao, "Minimizing probing cost and achieving identifiability in probe-based network link monitoring," *IEEE Transactions on Computers*, vol. 62, no. 3, pp. 510–523, 2013.

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

[8] S. Tati, S. Silvestri, T. He, and T. L. Porta, "Robust network tomography in the presence of failures," in *Proc. of IEEE ICDCS*, 2014.

[9] Y. Xia and D. Tse, "Inference of link delay in communication networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 12, pp. 2235–2248, 2006.

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

[11] L. Ma, T. He, K. K. Leung, A. Swami, and D. Towsley, "Monitor placement for maximal identifiability in network tomography," in *Proc. of IEEE INFOCOM*, 2014.

[12] R. Kumar and J. Kaur, "Practical beacon placement for link monitoring using network tomography," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 12, pp. 2196–2209, 2006.

[13] Y. Gao, W. Wu, W. Dong, C. Chen, X.-Y. Li, and J. Bu, "Preferential link tomography: Monitor assignment for inferring interesting link metrics," in *Proc. of IEEE ICNP*, 2014.

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

[15] L. Ma, T. He, A. Swami, D. Towsley, K. K. Leung, and J. Lowe, "Node failure localization via network tomography," in *Proc. of ACM IMC*, 2014.

[16] Y. H. Tsin, "A simple 3-edge-connected component algorithm," *Theory of Computing Systems*, vol. 40, no. 2, pp. 125–142, 2007.

[17] W. Ren and W. Dong, "Robust Network Tomography: k-identifiability and Monitor Assignment," Zhejiang University, Tech. Rep., June 2015. [Online]. Available: http://www.emnets.org/pub/tech-kid.pdf

[18] E. Rosen, A. Viswanathan, R. Callon *et al.*, "Multiprotocol label switching architecture," *RFC 3031*, 2001.

[19] S. Hu, K. C. H. W. W. Bai, C. Lan, H. W. H. Zhao, and C. Guo, "Explicit path control in commodity data centers: Design and applications," in *Proc. of USENIX NSDI*, 2015.

[20] N. Spring, R. Mahajan, D. Wetherall, and T. Anderson, "Measuring ISP topologies with Rocketfuel," *IEEE/ACM Transactions on Networking*, vol. 12, no. 1, pp. 2–16, 2004.