Node Failure Localization via Network Tomography *

Liang Ma
IBM T. J. Watson Research
Yorktown Heights, NY, USA
liang.ma@ieee.org

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

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

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

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

Jessica Lowe
DSTL
Salisbury, UK
jjlowe@dstl.gov.uk

Abstract

We investigate the problem of localizing node failures in a communication network from end-to-end path measurements, under the assumption that a path behaves normally if and only if it does not contain any failed nodes. To uniquely localize node failures, the measurement paths must show different symptoms under different failure events, i.e., for any two distinct sets of failed nodes, there must be a measurement path traversing one and only one of them. This condition is, however, impractical to test for large networks. Our first contribution is a characterization of this condition in terms of easily verifiable conditions on the network topology with given monitor placements under three families of probing mechanisms, which differ in whether measurement paths are (i) arbitrarily controllable, (ii) controllable but cycle-free, or (iii) uncontrollable (i.e., determined by the default routing protocol). Our second contribution is a characterization of the maximum identifiability of node failures, measured by the maximum number of simultaneous failures that can always be uniquely localized. Specifically, we bound the maximal identifiability from both the upper and the lower bounds which differ by at most one, and show that these bounds can be evaluated in polynomial time. Finally, we quantify the impact of the probing mechanism on the capability of node failure localization under different probing mechanisms on both random and real network topologies. We observe that despite a higher implementation cost, probing along controllable paths can significantly improve a network's capability to localize simultaneous node failures.

*Research was sponsored by the U.S. Army Research Laboratory and the U.K. Ministry of Defence and was accomplished under Agreement Number W911NF-06-3-0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, the U.K. Ministry of Defence or the U.K. Government. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

IMC'14, November 5–7, 2014, Vancouver, BC, Canada.

http://dx.doi.org/10.1145/2663716.2663723.

Categories and Subject Descriptors

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

Keywords

Network Tomography; Node Failure Localization; Identifiability Condition; Maximum Identifiability

1. INTRODUCTION

Effective monitoring of network performance is essential for network operators in building a reliable communication network that is robust against service disruptions. In order to achieve this goal, the monitoring infrastructure must be able to detect network misbehaviors (e.g., unusually high loss/latency, unreachability) and localize the sources of the anomaly (e.g., malfunction of certain routers) in an accurate and timely manner. Knowledge of where problematic network elements reside in the network is particularly useful for fast service recovery, e.g., the network operator can migrate affected services and/or reroute traffic. However, localizing network elements that cause a service disruption can be challenging. The straightforward approach of directly monitoring the health of individual elements is not always feasible due to traffic overhead, access control, or lack of protocol support at internal nodes. Moreover, built-in monitoring agents running on network elements cannot detect problems caused by misconfigured/unanticipated interactions between network layers, where end-to-end communication is disrupted but individual network elements along the path remain functional (a.k.a. *silent failures*) [1]. These limitations call for a different approach that can diagnose the health of network elements from the health of end-to-end communications perceived between measurement points.

This different approach is generally known as *network tomography* [2], where a canonical application is inferring internal network characteristics by measuring end-to-end performance from a subset of nodes with monitoring capabilities, referred to as *monitors*. Unlike direct measurement, network tomography only relies on end-to-end performance (e.g., path connectivity) experienced by data packets, thus addressing issues such as overhead, lack of protocol support, and silent failures. In cases where the network characteristic of interest is binary (e.g., *normal* or *failed*), the problem is known as *Boolean network tomography* [3].

In this paper, we study an application of Boolean network tomography to localize node failures from measurements of path states. Assuming that a measurement path is normal if and only if all nodes on this path behave normally, we formulate the problem as a system of Boolean equations, where the unknown variables are the binary node states, and the known constants are the observed states of measurement paths. The goal of Boolean network tomography is essentially to solve this system of Boolean equations.

Because the observations are coarse-grained (path normal/failed), it is usually impossible to uniquely identify node states from path measurements. For example, if two nodes always appear together in measurement paths, then upon observing failures of all these paths, we can at most deduce that one of these nodes (or both) has failed but cannot determine which one. Observing that there are often multiple explanations for given path failures, existing work mostly focuses on finding the most probable explanation that involves the minimum set of failed nodes. There is, however, no guarantee that nodes in this minimum set have failed or that nodes outside the set have not. Generally, to distinguish between two possible failure sets, there must exist a measurement path that traverses one and only one of these two sets. There is, however, a lack of understanding of what this requires in terms of observable network settings such as topology, monitor placement, and measurement routing.

In this paper, we consider two closely related problems: In a network with any given monitor placement, (1) if the number of simultaneous node failures is bounded by k, then under what conditions can one uniquely localize failed nodes from path measurements? (2) what is the maximum number of simultaneous node failures (i.e., the largest value of k) that can be uniquely localized in this network? We study both problems in the context of the following families of probing mechanisms: (i) *Controllable Arbitrary-path Probing (CAP)*, where measurement paths are arbitrarily controllable, (ii) *Controllable Simple-path Probing (CSP)*, where measurement paths are controllable but cycle-free, and (iii) *Uncontrollable Probing (UP)*, where measurement paths are determined by the default routing protocol. These probing mechanisms assume different levels of control over the routing of probing packets and are feasible in different network scenarios (see Section 2.3); answers to the above two problems under these probing mechanisms thus provide insights on how the level of control bestowed on the monitoring system affects its capability in failure localization.

In the sequel, we assume that node failures are persistent, i.e., a failed node remains failed throughout the measurement process and leads to failures of all paths traversing it.

1.1 Related Work

Based on the number of simultaneously failed elements, existing work can be broadly classified into single failure localization and multiple failure localization. Single failure localization assumes that multiple simultaneous failures happen with negligible probability. Under this assumption, [4,5] propose efficient algorithms for monitor placement such that any single failure can be detected and localized. To improve the resolution in characterizing failures, range tomography in [6] not only localizes the failure, but also estimates its severity (e.g., congestion level). These works, however, ignore the fact that multiple failures occur more frequently than one may imagine [7]. In this paper, we consider the general case of localizing multiple failures.

Multiple failure localization often faces inherent uncertainty in the number of failures. Most existing works address this uncertainty by attempting to find the minimum set of network elements whose failures can explain the observed path states. Under the assumption that failures are low-probability events, this approach generates the most probable failure set among all possibilities. Using this approach, [8,9] propose solutions for networks with tree topologies, which are later extended to general topologies by [1]. Similarly, [11] proposes to localize link failures by minimizing false positives; however, it cannot guarantee unique failure localization. In a Bayesian formulation, [12] proposes a two-staged solution which first estimates the failure (loss rate above threshold) probabilities of different links and then infers the most likely failure set for subsequent measurements. Augmenting path measurements with (partially) available control plane information (e.g., routing messages), [13,14] propose a greedy heuristic for troubleshooting network unreachability in multi-AS (Autonomous System) networks that has better accuracy than benchmarks using only path measurements.

Little is known when we insist on *uniquely* localizing network failures. Given a set of monitors known to uniquely localize failures on paths between themselves, [15] develops an algorithm to remove redundant monitors such that all failures remain identifiable. If the number of failed links is upper bounded by k and the monitors can probe arbitrary cycles or paths containing cycles, [16] proves that the network must be $(k+2)$-edge-connected to identify any failures up to k links using one monitor, which is then used to derive requirements on monitor placement for general topologies. However, the condition remains unknown if the failures are associated with nodes instead of links, or constraints (e.g., cycle-free) are imposed on measurement paths by the routing protocols (see discussions in Section 4.1 for why the results of [16] do not apply to our problem). In this paper, we investigate the fundamental relationships between node failure identifiability and explicit network settings such as topology, placement of monitors, and probing mechanism, with focus on developing efficient algorithms to characterize the capability of failure localization under given settings.

1.2 Summary of Contributions

We study, for the first time, the fundamental capability of a network with arbitrarily given monitor placements to uniquely localize node failures from binary end-to-end measurements between monitors. Our contributions are fivefold:

1) We propose a novel measure, referred to as *maximum identifiability*, to characterize a network's capability in failure localization as the maximum number of simultaneous node failures it can uniquely localize.

2) We establish abstract necessary/sufficient conditions for uniquely localizing a bounded number of failures, which are applicable to all probing mechanisms.

3) We translate the abstract conditions into more concrete conditions in terms of network topology, placement of monitors, and measurement paths under three different probing mechanisms (CAP, CSP, and UP), which can be tested in polynomial time.

4) We show that a special relationship between the above necessary/sufficient conditions leads to tight upper and lower bounds on the maximum identifiability that narrows its value to at most two consecutive integers. The bounds are polynomial-

Table 1: Graph-related Notations

Symbol	Meaning
V, L	set of nodes/links
M, N	set of monitors/non-monitors ($M \cup N = V$, $\mu := \|M\|$, $\sigma := \|N\|$)
$\mathcal{N}(M)$	set of non-monitors that are neighbors of at least one monitor in M
$\mathcal{L}(V, W)$	$\mathcal{L}(V, W) = \{\text{link } vw : \forall v \in V, w \in W, v \neq w\}$
$\mathcal{G} - L'$	delete links: $\mathcal{G} - L' = (V, L \setminus L')$, where "\" is setminus
$\mathcal{G} + L'$	add links: $\mathcal{G} + L' = (V, L \cup L')$, where the endpoints of links in L' must be in V
$\mathcal{G} - V'$	delete nodes: $\mathcal{G} - V' = (V \setminus V', L \setminus L(V'))$, where $L(V')$ is the set of links incident to nodes in V'
$\mathcal{G} + V'$	add nodes: $\mathcal{G} + V' = (V \cup V', L)$

time computable under CAP and CSP; while they are NP-hard to compute under UP, we give a greedy heuristic to compute a pair of relaxed bounds that frequently coincide with the original bounds in practice.

5) We extensively compare the maximum identifiability under different probing mechanisms on random and real topologies. Our comparison shows that although controllable probing, especially CAP, is more difficult to implement, it significantly improves the capability of failure localization in terms of maximum identifiability.

Note that the proposed model captures network state at a small time scale (time for conducting probing) and all above results are valid as long as node failures are persistent during probing. Moreover, we have limited our observations to binary states (normal/failed) of measurement paths. It is possible in some networks to obtain extra information from probes, e.g., rerouted paths after a default path fails, in which case our solution provides lower bounds on the maximum identifiability. Furthermore, we do not make any assumption on the distribution or correlation of node failures across the network. In some application scenarios (e.g., datacenter networks), failures of some nodes may be correlated (e.g., all routers sharing the same power/chiller). We leave the characterization of maximum identifiability in the presence of such additional information to future work.

The rest of the paper is organized as follows. Section 2 formulates the problem. Section 3 presents abstract conditions for identifying node failures, followed by concrete, verifiable conditions for specific families of probing mechanisms in Section 4. Based on the derived conditions, Section 5 presents bounds on the maximum identifiability that can be efficiently evaluated. The bounds are evaluated on various synthetic/real topologies in Section 6 to study the impact of the probing mechanism on the capability of node failure localization. Finally, Section 7 concludes the paper.

2. PROBLEM FORMULATION

2.1 Models and Assumptions

We assume that the network topology is known and can be modeled as an undirected graph[1] $\mathcal{G} = (V, L)$, where V and L are the sets of nodes and links. In $\mathcal{G}$, the number of neighbors of node v is called the *degree* of v. Note that graph $\mathcal{G}$ can represent a logical topology where each node in $\mathcal{G}$ corresponds to a physical subnetwork. Without loss of generality, we assume $\mathcal{G}$ is connected, as different connected components have to be monitored separately.

A subset of nodes M ($M \subseteq V$) are *monitors* that can initiate and collect measurements. The rest of the nodes, denoted by $N := V \setminus M$, are *non-monitors*. Let $\mu := |M|$ and $\sigma := |N|$ denote the numbers of monitors and non-monitors. We assume that monitors do not fail during the measurement process, as failed monitors can be directly detected and filtered out within the monitoring system. Non-monitors, on the other hand, may fail, and a failure event may involve simultaneous failures of multiple non-monitors. Depending on the adopted probing mechanism, monitors can measure and determine the states of nodes by sending probes along certain paths. Let P denote the set of all *possible measurement paths* under a given probing mechanism; for given $\mathcal{G}$ and M, different probing mechanisms can lead to different sets of measurement paths, which will be specified later. We use *node state* (*path state*) to refer to the state, failed or normal, of nodes (paths), where a path fails if and only if at least one node on the path fails. To avoid trivial cases, we assume that each non-monitor is traversed by at least one measurement path, as otherwise the non-monitor is unobservable to the monitoring system and has to be excluded in failure localization. Table 1 summarizes graph-related notations used in this paper (following the convention of [10]).

Let $\mathbf{w} = (W_1, \ldots, W_\sigma)^T$ be the binary column vector of the states of all non-monitors and $\mathbf{c} = (C_1, \ldots, C_\gamma)^T$ the binary column vector of the states of all measurement paths. For both node and path states, 0 represents "normal" and 1 represents "failed". We can relate the path states to the node states through the following Boolean linear system:

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

where $\mathbf{R} = (R_{ij})$ is a $\gamma \times \sigma$ *measurement matrix*, with each entry $R_{ij} \in \{0, 1\}$ denoting whether non-monitor v_j is present on path $\mathcal{P}_i$ (1: yes, 0: no), and "$\odot$" is the Boolean matrix product, i.e., $C_i = \vee_{j=1}^{\sigma}(R_{ij} \wedge W_j)$. The goal of Boolean network tomography is to invert this Boolean linear system to solve for $\mathbf{w}$ given $\mathbf{R}$ and $\mathbf{c}$. Intuitively, node failures are identifiable if and only if (1) has a unique solution.

2.2 Definitions

Let a *failure set* F be a set of non-monitors ($F \subseteq N$) that may fail simultaneously. The challenge in failure localization is that the solution to (1) is usually not unique, i.e., there are multiple possible failure sets leading to the observed path states. To reduce ambiguity, we limit the solution space to a predetermined collection Ψ of likely failure sets and only seek to ensure uniqueness within this collection. Let P_F denote the set of all measurement paths affected by a failure set F (i.e., traversing at least one node in F). We now formally define the notion of identifiability in node failure localization.

DEFINITION 1. *Given a network $\mathcal{G}$, a set of measurement paths P, and a collection Ψ of likely failure sets in $\mathcal{G}$:*

1. *Two failure sets F_1 and F_2 in Ψ can be* distinguished *from each other if and only if $P_{F_1} \neq P_{F_2}$, i.e., $\exists$ a path that traverses one and only one of F_1 and F_2.*
2. *In Ψ, failure set F is* identifiable *if and only if F can be distinguished from every other failure set in Ψ.*
3. *Ψ is* identifiable *if and only if every failure set in Ψ is identifiable.*

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

It is clear from the definition that whether a failure set is identifiable or not depends on the collection of potential failure sets it is compared against. Furthermore, Ψ being identifiable means that we can always uniquely localize node failures as long as the set of failed nodes falls into Ψ. Since a failure set may contain more than one node, we define the following notions to characterize network capability in localizing simultaneous node failures.

DEFINITION 2. *Given a network $\mathcal{G}$ and a set of measurement paths P in $\mathcal{G}$:*

1. *We say $\mathcal{G}$ is* k-identifiable *$(0 \leq k \leq \sigma)$ if the collection Ψ of all subsets of N with cardinality bounded by k is identifiable, i.e., any failure of up to k nodes can be uniquely localized.*
2. *The* maximum identifiability *of $\mathcal{G}$, denoted by $\Omega(\mathcal{G})$, is the maximum value of k such that $\mathcal{G}$ is k-identifiable.*

The maximum identifiability of a network characterizes its capability to localize failures in the worst case. That is, no matter where the failures occur, as long as the number of failed nodes is bounded by Ω, we can uniquely localize the failures from observed path states. Note that it is possible to uniquely localize a larger number of failures when they occur at a particular set of nodes, but localization cannot be guaranteed if the failures occur elsewhere. Both k-identifiability and maximum identifiability are defined with respect to a given P, which will be clear from the context.

2.3 Classification of Probing Mechanisms

Given the topology $\mathcal{G}$ and the monitor locations M, the probing mechanism plays a crucial role in failure localization by determining the set of measurement paths P. Depending on the flexibility of probing and the cost of deployment, we consider three families of probing mechanisms:

1. *Controllable Arbitrary-path Probing (CAP): P* includes any path/cycle, allowing repeated nodes/links, as long as each path/cycle starts and ends at monitors.
2. *Controllable Simple-path Probing (CSP): P* includes any *simple* path between distinct monitors, not including repeated nodes.
3. *Uncontrollable Probing (UP): P* is the set of paths between monitors determined by the routing protocol used by the network, not controllable by the monitors.

In particular, although CAP allows probes to traverse each node/link an arbitrary number of times, it suffices for probes to traverse each link at most once in either direction for the sake of localizing node failures.

These probing mechanisms clearly provide decreasing flexibility to the monitors and therefore decreasing capability to localize failures. Further, they also offer decreasing deployment cost. At the IP layer, CAP is feasible only if *(strict) source routing* (an IP option) [17] is enabled at all non-monitors, which allows them to modify the source and the destination addresses in packet headers hop by hop to probe a path prescribed by the monitor initiating the measurement probe[2]. If implemented at the application layer (e.g., to localize failures in overlay networks), CAP requires equivalent "source routing" to be supported by the application. Similarly, CSP is feasible under source routing (or equivalent capability at the application layer). It is also feasible under an emerging networking paradigm called software-defined networking (SDN) [18], where monitors can instruct the SDN controller to set up arbitrary cycle-free paths for the probing traffic. Note that the cycle-free constraint is crucial in SDN, as data forwarding is performed in a distributed manner by switches according to forwarding tables configured by the controller during route setup, which will encounter forwarding loops if the path has cycles. In contrast, UP only requires basic data forwarding and is generally feasible.

[2]The probe can follow the reverse path to return to the original monitor, thus effectively probing any path with *at least* one end at a monitor.

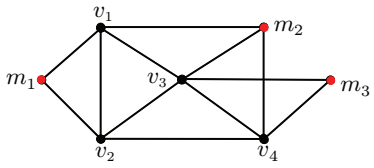


Figure 1: Sample network with three monitors: m_1, m_2, and m_3.

In this paper, we quantify how the flexibility of probing schemes affects the network's capability in failure localization. Although concrete results are only provided for the above families of probing mechanisms, our framework and our abstract identifiability conditions (see Section 3) can also be used to evaluate the failure localization capability of other probing mechanisms.

2.4 Objective

Given a network topology $\mathcal{G}$, a set of monitors M, and a probing mechanism (CAP, CSP, or UP), we seek to answer the following closely-related questions: (i) Given a bound k on the number of simultaneous failures, can we uniquely localize up to k failed nodes from observed path states? (ii) What is the maximum number of simultaneous failures we can localize? Clearly, answers to these questions require algorithms that can efficiently test for k-identifiability and determine the maximum identifiability.

2.5 Illustrative Example

Consider the sample network in Fig. 1 with three monitors (m_1–m_3) and four non-monitors (v_1–v_4). Clearly, the monitors' capability to identify failures of the non-monitors depends on the probing mechanism, i.e., which paths are measurable between the monitors. In this example, we will examine this capability and how it can be improved by relaxing constraints on measurement paths. Under UP, suppose that the default routing protocol only allows the monitors to probe the following paths: $\mathcal{P}_1 = m_1v_2v_1m_2$, $\mathcal{P}_2 = m_1v_2v_4m_3$, and $\mathcal{P}_3 = m_2v_3m_3$, which form a measurement matrix $\mathbf{R}^{\text{UP}}$:

$$\begin{array}{l} \mathcal{P}_1 = m_1v_2v_1m_2 \\ \mathcal{P}_2 = m_1v_2v_4m_3 \\ \mathcal{P}_3 = m_2v_3m_3 \end{array} \Rightarrow \mathbf{R}^{\text{UP}} = \begin{array}{c} \begin{array}{cccc} W_1 & W_2 & W_3 & W_4 \end{array} \\ \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \end{array}, \quad (2)$$

where $R^{\text{UP}}_{ij} = 1$ if and only if node v_j is on path $\mathcal{P}_i$. Then we have $\mathbf{R}^{\text{UP}} \odot \mathbf{w} = \mathbf{c}$, where $\mathbf{c}$ is the binary vector of path states observed at the destination monitors. Based on Definition 1, we can verify that any single node failure is identifiable, as for every two non-monitors, there is a measurement path traversing one and only one of them. However, these three paths cannot identify simultaneous failures of two nodes. This is because if node v_2 fails, then we cannot determine if v_1 (or v_4) fails or not. Identifiability can be improved if more measurement paths are allowed. For example, under CSP,

besides the three paths in (2), we can probe three additional paths: $\mathcal{P}_4 = m_1v_2v_3m_2$, $\mathcal{P}_5 = m_1v_1m_2$, and $\mathcal{P}_6 = m_2v_4m_3$, yielding an expanded measurement matrix in (3):

$$\begin{array}{l}\mathcal{P}_1 = m_1v_2v_1m_2\\ \mathcal{P}_2 = m_1v_2v_4m_3\\ \mathcal{P}_3 = m_2v_3m_3\\ \mathcal{P}_4 = m_1v_2v_3m_2\\ \mathcal{P}_5 = m_1v_1m_2\\ \mathcal{P}_6 = m_2v_4m_3\end{array} \Rrightarrow \mathbf{R}^{\text{CSP}} = \begin{array}{c} \begin{array}{cccc} W_1 & W_2 & W_3 & W_4 \end{array} \\ \left(\begin{array}{cccc} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ \hline 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{array}\right) \end{array} \begin{array}{l} \left.\begin{array}{l}\\ \\ \\ \end{array}\right\} \mathbf{R}^{\text{UP}} \\ \\ \\ \\ \end{array} \tag{3}$$

Using the six paths in (3), we can identify up to three failed nodes, a notable improvement over UP. However, if v_1, v_3, and v_4 all fail, then there is no measurement path under CSP that can be used to determine the state of v_2. Nevertheless, if CAP is supported, then we can send probes along a cycle $\mathcal{P}_7 = m_1v_2m_1$. In conjunction with the paths in (3), this yields the measurement matrix in (4):

$$\begin{array}{l}\mathcal{P}_5 = m_1v_1m_2\\ \mathcal{P}_7 = m_1v_2m_1\\ \mathcal{P}_3 = m_2v_3m_3\\ \mathcal{P}_6 = m_2v_4m_3\end{array} \Rrightarrow \mathbf{R}^{\text{CAP}} = \begin{array}{c} \begin{array}{cccc} W_1 & W_2 & W_3 & W_4 \end{array} \\ \left(\begin{array}{cccc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array}\right) \end{array} \tag{4}$$

Since the paths in (4) can independently determine the state of each non-monitor, CAP achieves full identifiability for the network in Fig. 1.

This example shows that in addition to the network topology and the monitor placement, the probing mechanism also significantly affects a network's capability to localize failures. In the rest of the paper, we will study this relationship both theoretically and algorithmically.

3. ABSTRACT IDENTIFIABILITY CONDITIONS

The definitions of identifiability in Definitions 1 and 2 are based on the enumeration of all possible failure scenarios and does not directly allow efficient testing and characterization of identifiability. To address this issue, we need more explicit identifiability conditions that can support efficient algorithm design. In this section, we will establish abstract sufficient/necessary conditions for k-identifiability under an arbitrary probing mechanism, which will later be developed into more concrete conditions for specific families of probing mechanisms.

Our sufficient condition is inspired by a result known in a related field called *combinatorial group testing* [19]. In short, group testing aims to find abnormal elements in a given set by running tests on subsets of elements, each test indicating whether any element in the subset is abnormal. This is analogous to our problem where abnormal elements are failed nodes and tests are conducted by probing measurement paths. A subtle but critical difference is that in our problem, the subsets of elements that can be tested together are limited by the set of measurement paths P, which are in turn limited by the topology, probing mechanism, and placement of monitors.

Most existing solutions for (nonadaptive) group testing aim at constructing a *disjunct testing matrix*. Specifically, a testing matrix R is a binary matrix, where $R_{i,j} = 1$ if and only if element j is included in the i-th test. Then R is called *k-disjunct* if the Boolean sum of any k columns does not "contain" any other column[3] [20]. In our problem, the existence of a disjunct testing matrix translates into a sufficient identifiability condition as follows.

LEMMA 3 (ABSTRACT SUFFICIENT CONDITION). *Any set of up to k failed nodes is identifiable if for any non-monitor v and failure set F with $|F| \leq k$ ($v \notin F$), there is a measurement path going through v but no node in F.*

PROOF. Consider two distinct failure sets F and F', each containing no more than k nodes. There exists a node v in only one of these sets; suppose $v \in F' \setminus F$. By the condition in the lemma, $\exists$ a path p traversing v but not F, thus distinguishing F from F'. □

Our necessary condition is based on the simple observation that to identify k failures, we must be able to identify the remaining $k - s$ ($1 \leq s \leq k-1$) failures after identifying and removing s of the failed nodes from the network, which leads to the following necessary condition.

LEMMA 4 (ABSTRACT NECESSARY CONDITION). *Any set of up to k failed nodes is identifiable only if for any set V' of non-monitors with $|V'| < k$, any set of up to $k - |V'|$ node failures is identifiable in $\mathcal{G} - V'$.*

PROOF. Suppose that $\exists$ two non-empty sets V' and V'' of non-monitors, with $V' \cap V'' = \emptyset$ and $|V'| + |V''| = k$, such that V'' is not identifiable in $\mathcal{G} - V'$. Then the union $F = V' \cup V''$ must be unidentifiable in $\mathcal{G}$, as even if we have identified failures in V', we still cannot identify the rest of the failures. □

Although neither of the above conditions directly lead to efficient testing algorithms, the significance of these conditions is that they are valid without relying on the probing mechanism being used. Moreover, we will show in the next section that these conditions provide theoretical foundations for efficient testing under specific families of probing mechanisms (CAP, CSP, and UP). Efficient testing for other families of probing mechanisms can also be explored using these abstract conditions.

4. VERIFIABLE IDENTIFIABILITY CONDITIONS

In this section, we develop the abstract conditions in Section 3 into concrete conditions suitable for efficient testing for the three families of probing mechanisms in Section 2.3.

4.1 Conditions under CAP

Under CAP, we can essentially "ping" any node from a monitor along any path. In the face of failures, this implies that a monitor's ability to determine the state of a node depends on its connectivity to the monitors after removing nodes that are known/hypothesized to have failed. This observation allows us to translate the abstract conditions in Section 3 into more concrete identifiability conditions as follows.

THEOREM 5 (k-IDENTIFIABILITY UNDER CAP). *Network $\mathcal{G}$ is k-identifiable under CAP:*

a) if for any set V' of up to k non-monitors, each connected component in $\mathcal{G} - V'$ contains a monitor;

[3]That is, for any subset of k column indices S and any other column index $j \notin S$, there exists a row index i such that $R_{i,j} = 1$ and $R_{i,j'} = 0$ for all $j' \in S$.

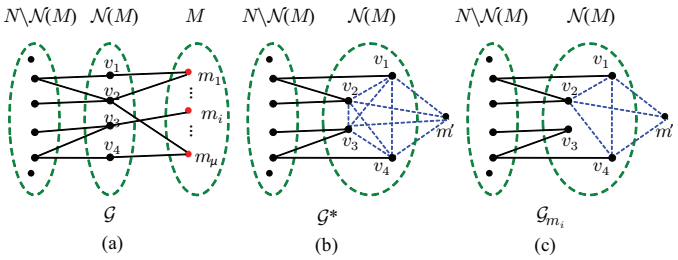


Figure 2: Auxiliary graphs: (a) Original graph $\mathcal{G}$; (b) $\mathcal{G}^*$ of $\mathcal{G}$; (c) $\mathcal{G}_{m_i}$ of $\mathcal{G}$ w.r.t. monitor m_i.

b) only if for any set V' of up to $k-1$ non-monitors, each connected component in $\mathcal{G}-V'$ contains a monitor.

PROOF. Suppose condition (a) holds, and consider a candidate failure set V' and a non-monitor v ($v \notin V'$). Since the connected component in $\mathcal{G}-V'$ that contains v has a monitor, there must exist a path connecting v to a monitor that does not traverse any node in V'. Following this path from the monitor to v and then back to the monitor then gives a path measurable under CAP that satisfies Lemma 3. Thus, condition (a) is sufficient.

Suppose condition (b) does not hold, i.e., there exists a non-monitor v that is disconnected from all monitors in $\mathcal{G}-V'$ for a set V' of up to $k-1$ non-monitors ($v \notin V'$). Then if nodes in V' fail, no remaining measurement path can probe v, and thus it is impossible to determine whether v has failed or not. This violates the condition in Lemma 4, and thus condition (b) is necessary. □

Simple as they look, these conditions still cannot be tested efficiently because they enumerate over a combinatorial number of sets V'. Fortunately, we are able to reduce them into explicit conditions on the vertex-connectivity of a related topology, which can then be tested in polynomial time. We use the following notions from graph theory.

DEFINITION 6. *[10] Graph $\mathcal{G}$ of $|V|$ vertices is said to be* k-vertex-connected *if $k \leq |V|-1$ and deleting any subset of up to $k-1$ vertices does not disconnect $\mathcal{G}$. The* vertex-connectivity *of $\mathcal{G}$, denoted by $\delta(\mathcal{G})$, is the maximum k such that $\mathcal{G}$ is k-vertex-connected.*

In our problem, the key observation is that requiring each connected component in $\mathcal{G}-V'$ to contain a monitor is equivalent to requiring each connected component in $\mathcal{G}-M-V'$ (i.e., after removing all monitors) to contain a neighbor of a monitor. Thus, if we add *virtual links* between these neighbors, the resulting graph $\mathcal{G}-M-V'+\mathcal{L}(\mathcal{N}(M), \mathcal{N}(M))$ should be connected. However, this does not mean the conditions are equivalent, because if $\mathcal{G}-M-V'$ is already connected, $\mathcal{G}-M-V'+\mathcal{L}(\mathcal{N}(M), \mathcal{N}(M))$ will certainly be connected but $\mathcal{G}-M-V'$ may not contain any neighbors of monitors. This special case can be avoided by introducing a *virtual monitor* m' connected to all neighbors of monitors via virtual links, resulting in an *auxiliary graph* $\mathcal{G}^* := \mathcal{G}-M+\{m'\}+\mathcal{L}\big(\mathcal{N}(M), \mathcal{N}(M)\big)+\mathcal{L}\big(\{m'\}, \mathcal{N}(M)\big)$ as illustrated in Fig. 2 (b). We will show that requiring at least one monitor per connected component in $\mathcal{G}-V'$ is equivalent to requiring $\mathcal{G}^*-V'$ to be connected.

The beauty of this new condition is that it reduces the tests over all possible V' to a single test of the vertex-connectivity of $\mathcal{G}^*$, as stated below.

LEMMA 7. *Each connected component in $\mathcal{G}-V'$ contains a monitor for any set V' of up to s ($s \leq \sigma-1$) non-monitors if and only if $\mathcal{G}^*$ is $(s+1)$-vertex-connected.*

PROOF. We first show the equivalence between the first condition and the connectivity of $\mathcal{G}^*-V'$. If the first condition holds, then each connected component in $\mathcal{G}-M-V'$ contains a neighbor of a monitor. Since these neighbors are connected with each other and also with m' in $\mathcal{G}^*-V'$, $\mathcal{G}^*-V'$ is connected. If the first condition is violated, i.e., there exists a connected component in $\mathcal{G}-M-V'$ without any neighbor of any monitor, then this component must be disconnected from m', and hence $\mathcal{G}^*-V'$ must be disconnected.

We then show that requiring $\mathcal{G}^*-V'$ to be connected for any V' of up to s non-monitors is equivalent to requiring it to be connected for any V' of up to s nodes in $\mathcal{G}^*$, including m', i.e., requiring $\mathcal{G}^*$ to be $(s+1)$-vertex-connected. It suffices to show that $\mathcal{G}^*-V'$ being connected for any V' of up to s non-monitors implies the connectivity of $\mathcal{G}^*-\{m'\}-V''$ for any V'' of up to $s-1$ non-monitors. Fixing a V'' of up to $s-1$ non-monitors, we assert that each connected component of $\mathcal{G}^*-\{m'\}-V''$ must contain a neighbor of a monitor, as otherwise $\mathcal{G}^*-V''$ will be disconnected. Since all these neighbors are connected via virtual links, $\mathcal{G}^*-\{m'\}-V''$ must be connected. □

Lemma 7 allows us to rewrite the identifiability conditions in Theorem 5 in terms of the vertex-connectivity of $\mathcal{G}^*$.

COROLLARY 8. *Network $\mathcal{G}$ is k-identifiable under CAP:*
a) if $\mathcal{G}^*$ is $(k+1)$-vertex-connected ($k \leq \sigma-1$);
b) only if $\mathcal{G}^*$ is k-vertex-connected ($k \leq \sigma$).

A special case not covered by this corollary is the case of $k=\sigma$ (the total number of non-monitors), i.e., if we wish to know whether the failure of *any* subset of non-monitors is identifiable. We address this case separately in the following proposition.

PROPOSITION 9. *Network $\mathcal{G}$ is σ-identifiable under CAP if and only if each non-monitor is the neighbor of a monitor.*

PROOF. If each non-monitor has a monitor as a neighbor, then their states can be determined independently through 1-hop probing, and hence any failure set is identifiable. On the other hand, if there exists a non-monitor v that is only reachable by monitors via other non-monitors, then the state of v cannot be determined in the case that all the other non-monitors fail, and hence $\mathcal{G}$ is not σ-identifiable. □

Discussion: A previous study [16] has provided necessary and sufficient conditions for a related problem of *link failure localization*, under the assumption that probes can traverse paths/cycles with possibly repeated nodes but no repeated links. Although the problem is analogous to node failure localization, the results of [16] do not apply to our problem[4].

Testing algorithm: A key advantage of the newly derived conditions over the abstract conditions in Section 3 is that they can be tested efficiently. Given a value of k, we

[4]Solving node failure localization using the results of [16] requires a topology transformation that maps each node to a link while maintaining adjacency between nodes and feasibility of measurement paths. To our knowledge, no such transformation exists whose output satisfies the assumptions of [16] (undirected graph, measurement paths not containing repeated links).

can evaluate the vertex-connectivity of $\mathcal{G}^*$, $\delta(\mathcal{G}^*)$, by the algorithm for determining network vertex connectivity in [21] in $O(\sigma^{3.75})$ time and compare the result with $k+1$ or k to test the conditions in Corollary 8.

4.2 Conditions under CSP

Under CSP, we restrict measurement paths P be the set of *simple paths* between monitors, i.e., paths starting/ending at distinct monitors and containing no cycles. As in the case of CAP, our goal here is again to translate the abstract conditions in Section 3 into concrete sufficient/necessary conditions that can be efficiently verified. We first give the following result analogous to Theorem 5.

THEOREM 10 (k-IDENTIFIABILITY UNDER CSP). *Network $\mathcal{G}$ is k-identifiable under CSP:*

a) if for any node set V', $|V'| \leq k+1$, containing at most one monitor, each connected component in $\mathcal{G}-V'$ contains a monitor;

b) only if for any node set V', $|V'| \leq k$, containing at most one monitor, each connected component in $\mathcal{G}-V'$ contains a monitor.

PROOF. Suppose condition (a) holds, and consider a candidate failure set F, $|F| \leq k$ and a non-monitor $v \notin F$. We argue that v must have two simple *vertex disjoint* paths to monitors in $\mathcal{G}-F$, and thus concatenating these paths provides a monitor-monitor simple path that traverses v but not F, satisfying the abstract sufficient condition in Lemma 3. Indeed, if such paths do not exist, i.e., $\exists$ a (monitor or non-monitor) node w ($w \neq v$) that resides on all paths from v to monitors in $\mathcal{G}-F$, then v will be disconnected from all monitors in $\mathcal{G}-F-\{w\}$, i.e., the connected component containing v in $\mathcal{G}-V'$, where $V'=F\cup\{w\}$, has no monitor, contradicting condition (a).

Suppose condition (b) does not hold, i.e., there exists a non-monitor v, a (monitor or non-monitor) node w, and a set of up to $k-1$ non-monitors F ($v \neq w$ and $v, w \notin F$) such that the connected component containing v in $\mathcal{G}-V'$, $V'=F\cup\{w\}$, contains no monitor. Then any path from v to monitors in $\mathcal{G}-F$ must traverse w, which means no monitor-monitor simple path in $\mathcal{G}-F$ will traverse v (as any monitor-monitor path traversing v must form a cycle at w). This violates the necessary condition in Lemma 4 because if node v fails, the failure cannot be identified in $\mathcal{G}-F$. □

As expected, due to the restriction to simple paths, the identifiability conditions in Theorem 10 are stronger than those in Theorem 5. As with Theorem 5, the conditions in Theorem 10 do not directly lead to efficient tests, and we again seek equivalent conditions in terms of topological properties. Each condition in the form of Theorem 10 (a–b) covers two cases: (i) V' only contains non-monitors; (ii) V' contains a monitor and $|V'|-1$ non-monitors. The first case has been converted to a vertex-connectivity condition on an auxiliary topology $\mathcal{G}^*$ by Lemma 7; we now establish a similar condition for the second case using similar arguments.

Fix a set $V'=F\cup\{m\}$, where m is a monitor in M and F a set of non-monitors. Again, the key observation is that each connected component in $\mathcal{G}-V'$ containing a monitor is equivalent to each connected component in $\mathcal{G}-M-F$ containing a neighbor of a monitor other than m (i.e., a node in $\mathcal{N}(M\setminus\{m\})$). To capture this, we introduce another *auxiliary graph* $\mathcal{G}_m := \mathcal{G}-M+\{m'\}+\mathcal{L}\big(\mathcal{N}(M\setminus\{m\}), \mathcal{N}(M\setminus\{m\})\big)+\mathcal{L}\big(\{m'\}, \mathcal{N}(M\setminus\{m\})\big)$ with respect to (w.r.t.) monitor m as illustrated in Fig. 2 (c), where m' is again a virtual monitor. We will show that the last condition is equivalent to requiring $\mathcal{G}_m-F$ to be connected, and thus the following holds.

LEMMA 11. *The following two conditions are equivalent:*

(1) Each connected component in $\mathcal{G}-V'$ contains a monitor for any set V' consisting of monitor m ($m\in M$) and up to s ($s\leq\sigma-1$) non-monitors;

(2) $\mathcal{G}_m$ is $(s+1)$-vertex-connected.

PROOF. The proof is similar to that of Lemma 7. If the first condition holds, then each connected component in $\mathcal{G}-M-F$ for $F:=V'\setminus\{m\}$ contains a node in $\mathcal{N}(M\setminus\{m\})$, and thus $\mathcal{G}_m-F$ is connected. If the first condition is violated, then there is a connected component in $\mathcal{G}-M-F$ that does not contain any node in $\mathcal{N}(M\setminus\{m\})$. This component must be disconnected from m' in $\mathcal{G}_m-F$, and thus $\mathcal{G}_m-F$ must be disconnected. Hence, the first condition is equivalent to $\mathcal{G}_m-F$ being connected for any set F of up to s non-monitors. Moreover, $\mathcal{G}_m-F$ being connected for any set F of up to s non-monitors implies that $\mathcal{G}_m-\{m'\}-F'$ (m' is the virtual monitor in $\mathcal{G}_m$) is connected for any F' of up to $s-1$ non-monitors, because otherwise $\mathcal{G}_m-F'$ will be disconnected. Therefore, the first condition is equivalent to $\mathcal{G}_m-F$ being connected for any set F of up to s nodes in $\mathcal{G}_m$, i.e., the first and second conditions in Lemma 11 are equivalent. □

Based on Lemmas 7 and 11, we can rewrite Theorem 10 as follows.

COROLLARY 12. *Network $\mathcal{G}$ is k-identifiable under CSP:*

a) if $\mathcal{G}^*$ is $(k+2)$-vertex-connected, and $\mathcal{G}_m$ is $(k+1)$-vertex-connected for each monitor $m\in M$ ($k\leq\sigma-2$);

b) only if $\mathcal{G}^*$ is $(k+1)$-vertex-connected, and $\mathcal{G}_m$ is k-vertex-connected for each monitor $m\in M$ ($k\leq\sigma-1$).

Special cases left out by this corollary are the cases of $k=\sigma$ and $k=\sigma-1$, which are addressed separately as follows.

PROPOSITION 13. *Network $\mathcal{G}$ is σ-identifiable under CSP if and only if each non-monitor has at least two monitors as neighbors.*

PROOF. If each non-monitor has at least two monitors as neighbors, then their states can be determined independently by cycle-free 2-hop probing between monitors, and thus the network is σ-identifiable. On the other hand, suppose $\exists$ a non-monitor v with zero or only one monitor neighbor. Then $\nexists$ simple paths going through v without traversing another non-monitor, and hence the state of v cannot be determined if all the other non-monitors fail. □

PROPOSITION 14. *Network $\mathcal{G}$ is $(\sigma-1)$-identifiable under CSP if and only if all but one non-monitor, denoted by v, have at least two monitors as neighbors, and v either has (i) two or more monitors as neighbors, or (ii) one monitor and all the other non-monitors (i.e., $N\setminus\{v\}$) as neighbors.*

PROOF. *a) Necessity:* Suppose that $\mathcal{G}$ is $(\sigma-1)$-identifiable under CSP. If it is also σ-identifiable, then each non-monitor must have at least two monitors as neighbors according to Proposition 13. Otherwise, we have $\Omega(\mathcal{G})=\sigma-1$. In this case, $\exists$ at least one non-monitor, denoted by v, with at most one monitor neighbor. Let $\mathcal{N}(v)$ denote all neighbors of v

including monitors. Suppose that v has λ neighbors (i.e., $|\mathcal{N}(v)| = \lambda$). Then there are two cases: (i) $\mathcal{N}(v)$ contains a monitor, denoted by $\widetilde{m}$; (ii) all nodes in $\mathcal{N}(v)$ are non-monitors. In case (i), the sets $F_1 = \mathcal{N}(v) \setminus \{\widetilde{m}\}$ and $F_2 = F_1 \cup \{v\}$ are not distinguishable because $\nexists$ monitor-to-monitor simple paths traversing v but not nodes in F_1. In case (ii), the sets $F_1 = \mathcal{N}(v) \setminus \{w\}$ (where w is an arbitrary node in $\mathcal{N}(v)$) and $F_2 = F_1 \cup \{v\}$ are not distinguishable as all monitor-to-monitor simple paths traversing v must go through at least one node in F_1. Based on (i–ii), we conclude that $\Omega(\mathcal{G}) \leq \lambda - 1$, where λ is the degree of any non-monitor with at most one monitor neighbor. For $\Omega(\mathcal{G}) = \sigma - 1$, we must have $\lambda \geq \sigma$, which can only be satisfied if all such non-monitors have one monitor and all the other non-monitors as neighbors. Moreover, if there are two such non-monitors v and u, then the sets $F \cup \{v\}$ and $F \cup \{u\}$, where $F = N \setminus \{v, u\}$, are not distinguishable as all monitor-to-monitor simple paths traversing v must go through F or u and vice versa. Therefore, such non-monitor must be unique.

b) Sufficiency: If each non-monitor has at least two monitors as neighbors, then $\mathcal{G}$ is σ-identifiable (hence also $(\sigma-1)$-identifiable) according to Proposition 13. If all but one non-monitor v have at least two monitors as neighbors, and v has one monitor $\widetilde{m}$ and all the other non-monitors (i.e., $N \setminus \{v\}$) as neighbors, then for any two failure sets F_1 and F_2 with $|F_i| \leq \sigma - 1$ ($i = 1, 2$), there are two cases: (i) F_1 and F_2 differ on a non-monitor other than v; (ii) F_1 and F_2 only differ on v. In case (i), since the states of all non-monitors other than v can be independently determined, F_1 and F_2 are distinguishable. In case (ii), suppose that $F_1 = F \cup \{v\}$ and $F_2 = F$ for $F \subseteq N \setminus \{v\}$. Since $|F_1| \leq \sigma - 1$, $|F| \leq \sigma - 2$ and $\exists$ a non-monitor $w \in (N \setminus \{v\}) \setminus F$. We know that v is a neighbor of w (as v is a neighbor of all the other non-monitors) and w is a neighbor of a monitor m other than $\widetilde{m}$ (as it has at least two monitor neighbors). Thus, $\widetilde{m}vwm$ is a monitor-to-monitor simple path traversing v but not F, whose measurement can distinguish F_1 and F_2. Therefore, $\mathcal{G}$ is $(\sigma - 1)$-identifiable under CSP. □

Testing algorithm: Similar to the case of CAP, we can use the algorithm in [21] to compute the vertex-connectivities of the auxiliary graphs $\mathcal{G}^*$ and $\mathcal{G}_m$ ($\forall m \in M$), and then compare the results with $k+2$ and $k+1$ (or $k+1$ and k) to test the conditions in Corollary 12 for any given k. The overall complexity of the test is $O(\mu\sigma^{3.75})$.

4.3 Conditions under UP

Under UP, monitors have no control over the paths between monitors, and the set of measurement paths P is limited to the paths between monitors predetermined by the network's native routing protocol. In contrast to the previous cases (CAP, CSP), identifiability under UP can no longer be characterized in terms of topological properties. We can, nevertheless, establish conditions more explicit than the abstract conditions in Section 3. The idea is to examine how many non-monitors need to be removed to disconnect all measurement paths traversing a given non-monitor v. If the number is sufficiently large (greater than k), then we can still infer the state of v from some measurement path when a set of other non-monitors fail; if the number is too small (smaller than or equal to $k-1$), then we will not be able to determine the state of v as the failures of all paths traversing v can already be explained by the failures of other non-monitors. This intuition leads to the following results.

In the sequel, $P_v \subseteq P$ denotes the set of measurement paths traversing a non-monitor v, and $\mathcal{S}_v := \{P_w : w \in N, w \neq v\}$ denotes the collection of path sets traversing non-monitors in $N \setminus \{v\}$. We use $\text{MSC}(v)$ to denote the *minimum set cover* of P_v by $\mathcal{S}_v$, i.e., $\text{MSC}(v) := |V'|$ for the minimum set $V' \subseteq N \setminus \{v\}$ such that $P_v \subseteq \bigcup_{w \in V'} P_w$. Note that covering is only feasible if v is not on any 2-hop measurement path (i.e., monitor-v-monitor), in which case we know $P_v \subseteq \bigcup_{w \in N, w \neq v} P_w$ and thus $\text{MSC}(v) \leq \sigma - 1$. If v is on a 2-hop path, then we define $\text{MSC}(v) := \sigma$.

THEOREM 15 (k-IDENTIFIABILITY UNDER UP). *Network $\mathcal{G}$ is k-identifiable under UP with measurement paths P:*

a) if $MSC(v) > k$ for any non-monitor v;

b) only if $MSC(v) > k-1$ for any non-monitor v.

PROOF. Suppose condition (a) holds. Then for any candidate failure set F with $|F| \leq k$ and any other non-monitor v ($v \notin F$), there must be a path in P_v that is not in $\bigcup_{w \in F} P_w$, i.e., traversing v but not F, which satisfies the abstract sufficient condition in Lemma 3.

Suppose condition (b) does not hold, i.e., there exists a non-monitor v and a set of non-monitors V' with $|V'| \leq k-1$ and $v \notin V'$, such that $P_v \subseteq \bigcup_{w \in V'} P_w$. Then given failures of all nodes in V', the state of v has no impact on observed path states and is thus unidentifiable, violating the abstract necessary condition in Lemma 4. □

Testing algorithm: The conditions in Theorem 15 provide an explicit way of testing the k-identifiability under UP, using tests of the form $\text{MSC}(v) > s$. Unfortunately, evaluating such a test, known as the decision problem of the *set covering problem*, is known to be NP-complete. Nevertheless, we can use approximation algorithms to compute bounds on $\text{MSC}(v)$. The best-known algorithm with approximation guarantee is the *greedy algorithm*, which iteratively selects the set in $\mathcal{S}_v$ that contains the largest number of uncovered paths in P_v until all the paths in P_v are covered (assuming that v is not on any 2-hop path).

Let $\text{GSC}(v)$ denote the number of sets selected by the greedy algorithm. This immediately provides an upper bound: $\text{MSC}(v) \leq \text{GSC}(v)$. Moreover, since the greedy algorithm has an approximation ratio of $\log(|P_v|) + 1$ [22], we can also bound $\text{MSC}(v)$ from below: $\text{MSC}(v) \geq \text{GSC}(v)/(\log(|P_v|) + 1)$. Applying these bounds to Theorem 15 yields a pair of relaxed conditions:

- $\mathcal{G}$ is k-identifiable under UP if $k < \lceil \min_{v \in N} \frac{\text{GSC}(v)}{\log(|P_v|)+1} \rceil$;
- $\mathcal{G}$ is *not* k-identifiable under UP if $k > \min_{v \in N} \text{GSC}(v)$.

These conditions can be tested by running the greedy algorithm for all non-monitors, each taking time $O(|P_v|^2\sigma) = O(|P|^2\sigma)$, and the overall test has a complexity of $O(|P|^2\sigma^2)$ (or $O(\mu^4\sigma^2)$ if there is a measurement path between each pair of monitors). However, we point out that it is unlikely that one can obtain stronger conditions based on Theorem 15 that are polynomial-time verifiable, as the greedy algorithm is known to give the best approximation for $\text{MSC}(v)$.

4.4 Special Case: 1-identifiability

In practice, the most common failure event consists of the failure of a single node. Thus, a question of particular interest is whether $\mathcal{G}$ is 1-identifiable under a given placement of monitors and a given probing mechanism. Although our

previous results (Corollaries 8 and 12, Theorem 15) provide an answer to the above question if the sufficient condition is satisfied or the necessary condition is violated for $k = 1$, the answer remains unknown if $\mathcal{G}$ satisfies the necessary condition but violates the sufficient condition. To address this case, we develop explicit methods to test the 1-identifiability.

4.4.1 Conditions for 1-identifiability

We start with a generic necessary and sufficient condition that applies to all probing mechanisms. Recall that P_v denotes the set of measurement paths traversing a non-monitor v. By Definition 1, we have the following claim:

CLAIM 16. *$\mathcal{G}$ is 1-identifiable if and only if:*
(1) $P_v \neq \emptyset$ for any $v \in N$, and
(2) $P_v \neq P_w$ for any $v, w \in N$ and $v \neq w$.

In Claim 16, the first condition guarantees that any non-empty failure set is distinguishable from the empty set (i.e., no failure), and the second condition guarantees that the observed path states can uniquely localize the failed node. An efficient test of these conditions, however, requires different strategies under different probing mechanisms.

4.4.2 Test under CAP

Under CAP, we know from Corollary 8 that for $\mathcal{G}$ to be 1-identifiable, the auxiliary graph $\mathcal{G}^*$ must be connected. Below, we will show that this condition is also sufficient.

LEMMA 17. *Network $\mathcal{G}$ is 1-identifiable under CAP if and only if $\mathcal{G}^*$ is connected, i.e., $\mathcal{G}$ has at least one monitor.*

PROOF. It suffices to show that $\mathcal{G}^*$ being connected is sufficient for the 1-identifiability of $\mathcal{G}$, which we prove by construction. First, it is easy to see that $\mathcal{G}^*$ is connected if and only if $\mathcal{G}$ has at least one monitor, denoted by m (recall that $\mathcal{G}$ is assumed to be connected). We build a spanning tree of $\mathcal{G}$ rooted at m, and sequentially probe each non-monitor in a breadth-first or depth-first order until (i) finding a first failure or (ii) completing probing without any failure. As each non-monitor is only probed after all its predecessors in the tree have been probed, we know in case (i) that the last probed non-monitor has failed, and in case (ii) that no node has failed. Therefore, $\mathcal{G}$ is 1-identifiable. □

Testing for 1-identifiability under CAP is therefore reduced to determining if the network has a monitor.

4.4.3 Test under CSP

Under CSP, we derive conditions that are equivalent to those in Section 4.4.1 but easier to test.

Condition (1) in Claim 16 requires that every non-monitor reside on a monitor-monitor simple path. While an exhaustive search for such a path will incur exponential complexity, we can test for its existence efficiently using the following observation. The idea is to construct an *extended graph* $\mathcal{G}' := \mathcal{G} + \{m'\} + \mathcal{L}(\{m'\}, M)$, i.e., by adding a virtual monitor m' and connecting it to all the monitors; see an illustration in Fig. 3. We claim that a non-monitor v is on a monitor-monitor simple path if and only if the size of the minimum vertex cut between v and m' in $\mathcal{G}'$ is at least two.

Indeed, by Menger's Theorem [10], the size of the *minimum vertex cut between v and m'* (minimum v-m' vertex cut) equals the maximum number of *vertex-independent*[5] simple paths between them. Therefore, a minimum v-m' vertex cut of size at least two implies the existence of two vertex-independent simple paths between v and m', illustrated as paths vm_2m' and vm_im' in Fig. 3. Truncating these two paths (vm_2m' and vm_im') at the first monitors (along the way from v to m') and concatenating the remaining two path segments gives a monitor-to-monitor simple path traversing v, i.e., m_2vm_i in Fig. 3. On the other hand, if $\exists$ a monitor-to-monitor simple path traversing v, then it can be split into two simple paths connecting v to two distinct monitors, which implies a minimum v-m' cut of at least two vertices in the extended graph $\mathcal{G}'$ as each of these two distinct monitors connects to m' by a virtual link.

Condition (2) in Claim 16 is violated if and only if there exist two non-monitors $v \neq w$ such that all monitor-to-monitor simple paths traversing v must traverse w (i.e., $P_v \subseteq P_w$) and vice versa. Since $P_v \subseteq P_w$ means that there is no monitor-to-monitor simple path traversing v in $\mathcal{G} - \{w\}$, by the above argument, we see that $P_v \subseteq P_w$ if and only if the size of the minimum vertex cut between v and m' in a new graph $\mathcal{G}'_w := \mathcal{G} - \{w\} + \{m'\} + \mathcal{L}(\{m'\}, M)$ is smaller than two. Therefore, condition (2) in Claim 16 is satisfied if and only if for every two distinct non-monitors v and w, either the minimum v-m' cut in $\mathcal{G}'_w$ or the minimum w-m' cut in $\mathcal{G}'_v$ has a size of at least two.

In summary, the necessary and sufficient condition for 1-identifiability under CSP is ($C_{v,w}$ denotes the minimum vertex cut between v and w):

i) $|C_{v,m'}| \geq 2$ in $\mathcal{G}'$ for each $v \in N$, and
ii) $|C_{v,m'}| \geq 2$ in $\mathcal{G}'_w$ or $|C_{w,m'}| \geq 2$ in $\mathcal{G}'_v$ for all $v, w \in N$ and $v \neq w$.

Since for a graph $\mathcal{G}$ of $|V|$ nodes and $|L|$ links, $|C_{v,w}| \geq 2$ can be tested in $O(|V| + |L|)$ time[6], the overall test takes $O(\sigma^2(|V| + |L|)) = O(\sigma^2(\mu + \sigma)^2)$ time.

4.4.4 Test under UP

Under UP, the total number of measurement paths $|P|$ is reduced to $O(\mu^2)$ (from exponentially many as in the case of CAP/CSP) since the measurable routes are predetermined. This reduction makes it feasible to directly test the generic conditions (1–2) in Claim 16 by testing condition (1) for each non-monitor and condition (2) for each pair of non-monitors. The overall complexity of this test is $O(\sigma^2\mu^2)$, dominated by the testing of condition (2) in Claim 16.

5. CHARACTERIZATION OF MAXIMUM IDENTIFIABILITY

Although it is challenging to determine the exact value of the maximum identifiability $\Omega(\mathcal{G})$ without a (polynomial-time verifiable) necessary *and* sufficient condition for testing k-identifiability (it remains open as to whether it is NP-hard to determine the value of $\Omega(\mathcal{G})$), we will show that the conditions derived in Section 4 have a nice structure that allows us to provide tight upper and lower bounds on $\Omega(\mathcal{G})$.

5.1 Maximum Identifiability under CAP

Observing that both the sufficient and the necessary conditions in Corollary 8 are imposed on the same property, i.e., vertex-connectivity of the auxiliary graph $\mathcal{G}^*$, we obtain a

[5]Two paths are vertex-independent if they have no common vertex except for the endpoints.

[6]For example, we can compute the biconnected component decomposition [23] and test if v and w belong to the same biconnected component.

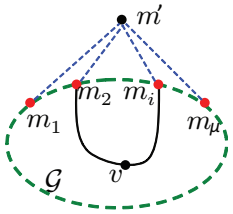


Figure 3: Extended graph $\mathcal{G}'$.

tight characterization of the maximum identifiability under CAP as follows. Here $\delta(\mathcal{G})$ is the vertex connectivity of $\mathcal{G}$ defined in Definition 6.

THEOREM 18 (MAXIMUM IDENTIFIABILITY UNDER CAP). *If $\delta(\mathcal{G}^*) \leq \sigma - 1$, the maximum identifiability of $\mathcal{G}$ under CAP, $\Omega^{CAP}(\mathcal{G})$, is bounded by $\delta(\mathcal{G}^*) - 1 \leq \Omega^{CAP}(\mathcal{G}) \leq \delta(\mathcal{G}^*)$.*

PROOF. Given $\delta(\mathcal{G}^*)$, we know that $\mathcal{G}^*$ is $\delta(\mathcal{G}^*)$-vertex-connected but not $(\delta(\mathcal{G}^*)+1)$-vertex-connected. By Corollary 8, this means that $\mathcal{G}$ is $(\delta(\mathcal{G}^*)-1)$-identifiable but not $(\delta(\mathcal{G}^*)+1)$-identifiable, which yields the above bounds on the maximum identifiability. Note that applying Corollary 8 requires $\delta(\mathcal{G}^*) \leq \sigma - 1$. □

Remark: In the special case of $\delta(\mathcal{G}^*) = \sigma$ (note that $\delta(\mathcal{G}^*) \leq \sigma$ by definition), $\mathcal{G}^*$ must be a clique, which means that all non-monitors must be neighbors of monitors. By Proposition 9, this implies that $\Omega^{\text{CAP}}(\mathcal{G}) = \sigma$.

Evaluation algorithm: Using the algorithm for determining network vertex connectivity in [21], we can compute $\delta(\mathcal{G}^*)$ and evaluate $\Omega^{\text{CAP}}(\mathcal{G})$ by the bounds in Theorem 18 in $O(\sigma^{3.75})$ time. The special case of $\Omega^{\text{CAP}}(\mathcal{G}) = \sigma$ can be checked separately in $O(\sigma)$ time using the condition in Proposition 9.

5.2 Maximum Identifiability under CSP

As in the case of CAP, we can leverage the analogy between the sufficient and the necessary conditions in Corollary 12 to bound the maximum identifiability under CSP from both sides. Specifically, let $\delta_{\min} := \min_{m:m\in M} \delta(\mathcal{G}_m)$ be the minimum vertex-connectivity for auxiliary graphs $\mathcal{G}_m$. Then the maximum identifiability is bounded as follows.

THEOREM 19 (MAXIMUM IDENTIFIABILITY UNDER CSP). *If $\min(\delta_{min}, \delta(\mathcal{G}^*)-1) \leq \sigma - 2$, the maximum identifiability of $\mathcal{G}$ under CSP, $\Omega^{CSP}(\mathcal{G})$, is bounded by $\min(\delta_{min}-1, \delta(\mathcal{G}^*)-2) \leq \Omega^{CSP}(\mathcal{G}) \leq \min(\delta_{min}, \delta(\mathcal{G}^*)-1)$.*

PROOF. By definition of vertex-connectivity, $\mathcal{G}^*$ is $\delta(\mathcal{G}^*)$-vertex-connected, and $\mathcal{G}_m$ is $\delta_{\min}$-vertex-connected for each monitor $m \in M$. This satisfies the condition in Corollary 12 (a) for $k = \min(\delta_{\min}-1, \delta(\mathcal{G}^*)-2)$, and thus $\Omega^{\text{CSP}}(\mathcal{G}) \geq \min(\delta_{\min}-1, \delta(\mathcal{G}^*)-2)$. Meanwhile, since $\mathcal{G}^*$ is not $(\delta(\mathcal{G}^*)+1)$-vertex-connected, and $\mathcal{G}_m$ is not $(\delta_{\min}+1)$-vertex-connected for some $m \in M$, the condition in Corollary 12 (b) is violated for $k = \min(\delta_{\min}+1, \delta(\mathcal{G}^*))$ (which requires $\min(\delta_{\min}+1, \delta(\mathcal{G}^*)) \leq \sigma - 1$). Thus, $\Omega^{\text{CSP}}(\mathcal{G}) \leq \min(\delta_{\min}, \delta(\mathcal{G}^*)-1)$. □

Remark: Because the set of links in $\mathcal{G}_m$ is a subset of those in $\mathcal{G}^*$ while the nodes are the same, we always have $\delta_{\min} \leq \delta(\mathcal{G}^*)$. Therefore, the above bounds simplify to:

- $\delta_{\min} - 2 \leq \Omega^{\text{CSP}}(\mathcal{G}) \leq \delta_{\min} - 1$ if $\delta_{\min} = \delta(\mathcal{G}^*)$;
- $\delta_{\min} - 1 \leq \Omega^{\text{CSP}}(\mathcal{G}) \leq \delta_{\min}$ if $\delta_{\min} < \delta(\mathcal{G}^*)$.

In particular, if $\delta(\mathcal{G}^*) = 1$ (i.e., there is a cut-vertex in $\mathcal{G}^*$), then $\Omega^{\text{CSP}}(\mathcal{G}) = 0$, i.e., even single-node failures cannot always be localized.

The only cases when $\min(\delta_{\min}, \delta(\mathcal{G}^*)-1) \leq \sigma - 2$ is violated are: (i) $\delta_{\min} = \delta(\mathcal{G}^*) = \sigma$, or (ii) $\delta_{\min} = \sigma - 1$ and $\delta(\mathcal{G}^*) = \sigma$. In case (i), $\mathcal{G}_m$ is a clique for all $m \in M$, i.e., each non-monitor still has a monitor as a neighbor after removing m; by Proposition 13, this implies that $\Omega^{\text{CSP}}(\mathcal{G}) = \sigma$. In case (ii), Corollary 12 (a) can still be applied to show that $\Omega^{\text{CSP}}(\mathcal{G}) \geq \sigma - 2$, and one can verify that the condition in Proposition 13 is violated, which implies that $\Omega^{\text{CSP}}(\mathcal{G}) \leq \sigma - 1$. In fact, we can leverage Proposition 14 to uniquely determine $\Omega^{\text{CSP}}(\mathcal{G})$ in this case. If condition (ii) in Proposition 14 is satisfied, then $\Omega^{\text{CSP}}(\mathcal{G}) = \sigma - 1$; otherwise, $\Omega^{\text{CSP}}(\mathcal{G}) = \sigma - 2$.

Evaluation algorithm: Evaluating $\Omega^{\text{CSP}}(\mathcal{G})$ by Theorem 19 involves computing the vertex-connectivities of the auxiliary graphs $\mathcal{G}^*$ and $\mathcal{G}_m$ $(\forall m \in M)$ using the algorithm for determining network vertex connectivity in [21], which altogether takes $O(\mu\sigma^{3.75})$ time.

5.3 Maximum Identifiability under UP

Let $\Delta := \min_{v\in N} \text{MSC}(v)$ be the minimum set cover over all non-monitors. The conditions in Theorem 15 imply the following bounds on the maximum identifiability under UP.

THEOREM 20 (MAXIMUM IDENTIFIABILITY UNDER UP). *The maximum identifiability of $\mathcal{G}$ under UP, $\Omega^{UP}(\mathcal{G})$, with measurement paths P is bounded by $\Delta - 1 \leq \Omega^{UP}(\mathcal{G}) \leq \Delta$.*

PROOF. Since $\text{MSC}(v) > \Delta - 1$ for all $v \in N$, $\mathcal{G}$ is $(\Delta-1)$-identifiable by Theorem 15 (a). Meanwhile, since there exists a node $v \in N$ with $\text{MSC}(v) = \Delta$, $\mathcal{G}$ is not $(\Delta+1)$-identifiable by Theorem 15 (b). Together, they imply the bounds on $\Omega^{\text{UP}}(\mathcal{G})$. □

Remark: Recall that $\Delta \leq \sigma$ by definition. In the special case of $\Delta = \sigma$, we know that all non-monitors are on 2-hop measurement paths, whose states can be determined independently. Thus, $\Omega^{\text{UP}}(\mathcal{G}) = \sigma$ in this case.

Evaluation algorithm: The original bounds in Theorem 20 are hard to evaluate due to the NP-hardness of computing $\text{MSC}(\cdot)$. As in Section 4.3, we resort to the greedy algorithm, which implies the following relaxed bounds:

$$\left\lceil \min_{v\in N} \frac{\text{GSC}(v)}{\log(|P_v|)+1} \right\rceil - 1 \leq \Omega^{\text{UP}}(\mathcal{G}) \leq \min_{v\in N} \text{GSC}(v). \quad (5)$$

Evaluating these bounds involves invoking the greedy algorithm for each non-monitor, with an overall complexity of $O(|P|^2\sigma^2)$ (or $O(\mu^4\sigma^2)$ if all monitors can probe each other).

6. IMPACT OF PROBING MECHANISM

Given the above results, we are now ready to quantify the impact of the probing mechanism on node failure localization. We aim to quantify this impact by evaluating, using our bounds on the maximum identifiability, the number of simultaneous failures we can uniquely localize in a given network with a given monitor placement under each of the three probing mechanisms (CAP, CSP, UP). In this study, we assume (hop count-based) shortest path routing as the default routing protocol under UP, i.e., the measurement paths under UP are the shortest paths between monitors, with ties broken arbitrarily.

6.1 Topologies for Evaluation

We evaluate the proposed metrics on both synthetic and real network topologies detailed as follows.

6.1.1 Synthetic 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 (RPL) graphs. We randomly generate graph realizations of each model[7], with each realization containing 20 nodes (i.e., $|V| = 20$). The generated graphs are then used to evaluate the impact of probing mechanisms. We now describe the models and present the corresponding results separately.

Erdös-Rényi (ER) graph: The ER graph [24] is 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 (note that the number of nodes is a predetermined parameter).

Random Geometric (RG) graph: The RG graph [25] 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 the node communication range. The resulting topology contains well-connected sub-graphs in densely populated areas and poorly-connected sub-graphs in sparsely populated areas.

Barabási-Albert (BA) graphs: The BA model [26] 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 the new node to existing 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., citation networks, and social networks.

Random Power Law (RPL) graphs: The BA model introduces an artifact that all node degrees are lower bounded by $n_{\min}$. Alternatively, the RPL graph [27] 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 RPL 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 RPL graph are connected by a link with probability $p_{ij} = d_i d_j / \sum_{k=1}^{|V|} d_k$.

Remark: Our motivation for performing evaluations on random topologies is that they allow comprehensive evaluation without artifacts of specific network deployments, which are common in real topologies. Moreover, the selected graph models can provide insights on how the topological property affects node failure localization.

6.1.2 Real Topologies

For real topologies, we use the *Autonomous System* (AS) topologies collected by the Rocketfuel [28] and the CAIDA [29] projects, which represents IP-level connections between backbone/gateway routers of several ASes from major *Internet Service Providers (ISPs)* around the globe.

[7]All realizations are guaranteed to be connected, as we discard disconnected realizations in the generation process.

Algorithm 1: Enhanced Random Monitor Placement (ERMP)

input : Network topology $\mathcal{G}$, all possible measurement paths Q under UP, number of monitors μ
output: Set of monitors M

1 $M \leftarrow$ {all degree-1 nodes} $\cup$ {one in every two neighboring degree-2 nodes};
2 **if** $M = \emptyset$ **then**
3 | $M \leftarrow$ {endpoints of the longest path in Q};
4 **end**
5 $U \leftarrow V \setminus (\bigcup_{m,m' \in M} V_{mm'})$; // uncovered nodes
6 **while** $U \neq \emptyset$ **do**
7 | $m = \arg\max_{w \in V \setminus M} |U \cap \mathcal{V}(w, M)|$;
8 | $U \leftarrow U \setminus \mathcal{V}(m, M)$;
9 | $M \leftarrow M \cup \{m\}$;
10 **end**
11 **if** $|M| < \mu$ **then**
12 | $M \leftarrow M \cup \{\mu - |M|$ nodes randomly selected from $V \setminus M\}$
13 **end**

6.2 Placement of Monitors

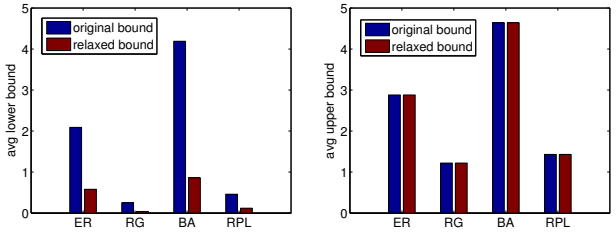
Since the maximum identifiability Ω depends on the given placement of monitors, we want to randomize this given monitor placement for a comprehensive evaluation. A purely random placement, however, is likely to lead to trivial Ω, since $\Omega = 0$ whenever there is a non-monitor not traversed by any measurement path. Specifically, Ω^{CSP} and Ω^{UP} will be zero if any degree-1 node or any two neighboring degree-2 nodes are non-monitors; moreover, Ω^{UP} will also be zero if not all non-monitors lie on shortest paths between monitors. To avoid these trivial cases, we adopt an *Enhanced Random Monitor Placement (ERMP)* strategy, which consists of the following two steps.

Step (i): place monitors to avoid the obvious cases of zero-maximum identifiability mentioned above;

Step (ii): place additional monitors, if available, randomly.

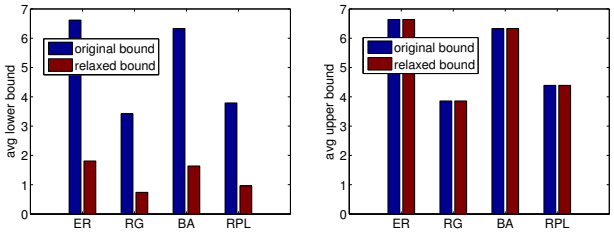
See Algorithm 1 for the pseudo code of ERMP. It is assumed that the total number of monitors is sufficient for step (i) above. Note that ERMP is only used for evaluating Ω, while our bounds established in Section 5 are valid under arbitrary monitor placements, i.e., the expression of our bounds does not depend on the choice of monitor placement algorithm or the fraction of monitors $(\mu/|V|)$. The optimization of monitor placement to achieve a desired level of identifiability will be pursued separately in a future work.

Specifically, it suffices to consider the most restrictive probing mechanism UP. Given the set of all *potentially* measurement paths Q under UP (here it is the set of all-pair shortest paths), let V_{vw} denote the set of nodes covered by the path between nodes v and w (inclusive). Given a set of existing monitors $M \subseteq V$ and a candidate monitor $w \in V \setminus M$, define $\mathcal{V}(w, M) := \bigcup_{m \in M} V_{wm}$ as the set of nodes covered by the paths between w and the existing monitors. We perform step (i) above by a greedy heuristic. We first jump-start the monitor placement with an initial set of monitors required to achieve a non-zero value for Ω^{CSP} (line 1); if this initial set is empty, we select the two monitors covering the maximum number of nodes (line 3). We then enlarge this set by selecting a new monitor in each iteration whose paths to the existing monitors cover the maximum number of *uncovered* nodes (line 7), until all nodes are covered by at least one



(a) Average lower bound (b) Average upper bound

Figure 4: Original and relaxed bounds on the maximum identifiability Ω^{UP} under UP for sparsely-connected random topologies ($|V| = 20$, $\mu = 10$, $\mathbb{E}[|L|] = 51$, 100 graph instances per model).



(a) Average lower bound (b) Average upper bound

Figure 5: Original and relaxed bounds on the maximum identifiability Ω^{UP} under UP for densely-connected random topologies ($|V| = 20$, $\mu = 10$, $\mathbb{E}[|L|] = 99$, 100 graph instances per model).

measurement path under UP (line 10). Finally, extra monitors, if any, are placed randomly among the remaining nodes.

6.3 Impact on Identifiability

6.3.1 Tightness of Bounds

To measure the impact of probing on the maximum identifiability Ω, we need tight bounds on Ω under all probing mechanisms. Although we have achieved this theoretically by deriving upper and lower bounds that differ by at most one (Theorems 18, 19, 20), only the bounds under CAP and CSP can be evaluated efficiently, and the bounds under UP have to be relaxed by a logarithmic factor to be computable in polynomial time (see (5)). The first question is therefore how tight the relaxed bounds are.

To this end, we compare the original bounds (Theorem 20) and the relaxed bounds (5) on a variety of topologies synthetically generated from the models in Section 6.1.1 in two scenarios, i.e., sparsely-connected and densely-connected topologies. To make the models comparable in each scenario, we have tuned each model to generate graphs with the same number of nodes and (average) number of links. We evaluate both bounds on multiple graph instances per model, each with a fixed number of monitors placed by ERMP, and present the average lower/upper bounds in Fig. 4 and Fig. 5. As expected, in both scenarios, the relaxed lower bounds are quite loose due to the logarithmic factor, but the relaxed upper bounds coincide with the original bounds for all graph instances. This indicates that although the relaxed upper bound $\min_{v \in N} \text{GSC}(v)$ can be a logarithmic-factor larger than the original upper bound Δ in the worst case, this worst case rarely occurs, and we can approximate Δ by $\min_{v \in N} \text{GSC}(v)$ to apply Theorem 20. This provides a tight characterization of Ω^{UP} for large networks, where computing the original bounds is infeasible.

6.3.2 Comparison of Probing Mechanisms

We are now ready to compare[8] Ω^{CAP}, Ω^{CSP}, and Ω^{UP}.

Comparison Using Random Topologies: Similar to Section 6.3.1, Ω^{CAP}, Ω^{CSP}, and Ω^{UP} are compared on both sparsely-connected and densely-connected topologies generated from the four random graph models. Under each scenario, we generate multiple graph instances from each of the four models and sequentially place monitors in each instance using ERMP such that the set of monitors grows strictly monotonically as the number of monitors increases.

[8]In the case of $0 \leq \Omega \leq 1$, we use the tests in Section 4.4 to uniquely determine the value of Ω.

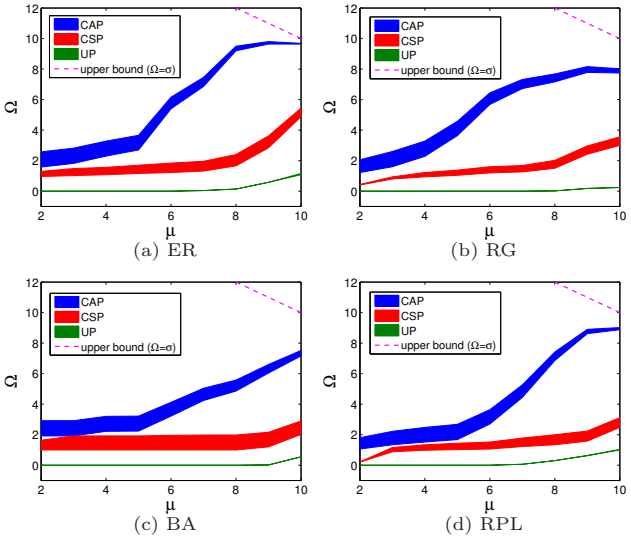
We then evaluate our bounds on the maximum identifiability Ω^{CAP}, Ω^{CSP}, and Ω^{UP} for each graph instance under each monitor placement.

The average results in sparsely-connected networks are shown in Fig. 6. The results show large differences in the maximum identifiabilities of the different probing mechanisms: while UP can barely localize a single node failure even if half of the nodes are monitors, CAP can provide unique localization even if up to 90% of the non-monitors simultaneously fail. We also observe that the maximum identifiability is larger for ER graphs and smaller for BA graphs. Intuitively, this is because while nodes in ER graphs have uniform connectivity, those in BA graphs have highly variable connectivity, which creates poorly connected subgraphs whose node failures are more difficult to localize. Note that Ω eventually decreases as the number of monitors μ increases, as the maximum identifiability is always upper bounded by the total number of non-monitors ($\sigma = |V| - \mu$); we have verified that the *normalized maximum identifiability* Ω/σ increases monotonically with μ.

The average results for densely-connected networks are reported in Fig. 7. In comparison to Fig. 6, both CAP and CSP achieve greater maximum identifiability in densely-connected networks; in particular, CAP uniquely localizes arbitrary node failures, i.e., $\Omega^{\text{CAP}} = \sigma$, for all the graph models when $\mu/|V| \geq 45\%$. Meanwhile, as Fig. 7 shows, increasing the number of links (and hence the density of the graphs) hardly affects the maximum identifiability under UP. Intuitively, this is because densely-connected graphs yield more measurement paths between each pair of monitors under controllable probing mechanisms (CAP and CSP), thus enabling them to identify more simultaneous failures. In contrast, there is only one measurement path (the shortest path) between each pair of monitors under UP, independent of the number of links in the network. Therefore, UP exhibits similar maximum identifiability for both sparsely-connected and densely-connected topologies.

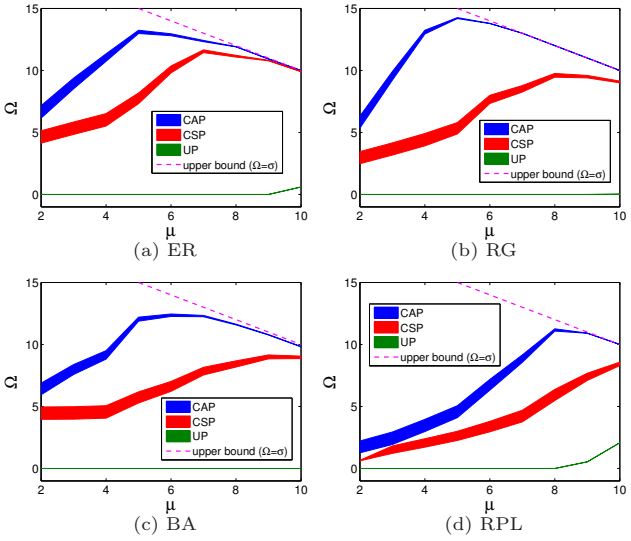
Comparison Using AS Topologies: For AS topologies, we first compute the minimum number of monitors required by step (i) of ERMP, denoted by μ_c, and then vary the fraction of monitors $\mu/|V|$ such that $\mu \geq \mu_c$ for all the topologies. To facilitate comparison, we use the same range of values for $\mu/|V|$ for each dataset. For each topology, we independently select 20 sets of monitors using ERMP (only $\mu - \mu_c$ monitors in each set are randomly placed), under which Ω^{CAP}, Ω^{CSP}, and Ω^{UP} are evaluated.

Fig. 8 shows bounds on the maximum identifiability averaged over different monitor placements for the Rocketfuel AS topologies. In the selected Rocketfuel topologies (Fig. 8),



(a) ER (b) RG (c) BA (d) RPL

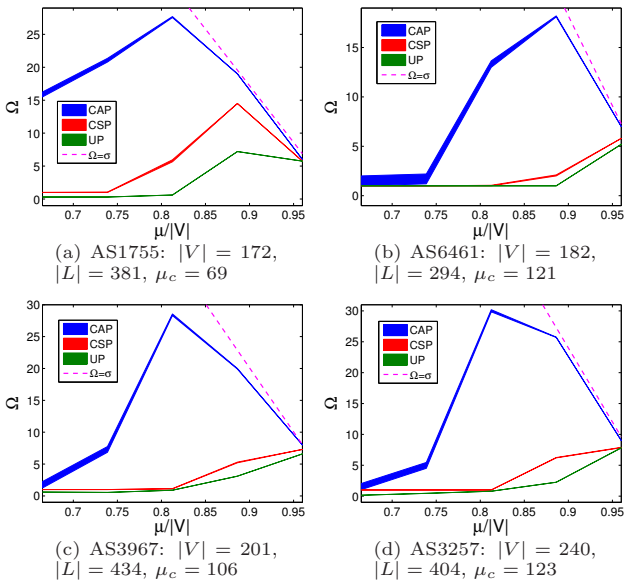
Figure 6: Bounds on the maximum identifiability Ω under CAP, CSP, and UP for sparsely-connected random topologies ($|V| = 20$, $\mu = 2, \ldots, 10$, $\mathbb{E}[|L|] = 51$, 50 graph instances per model).



(a) ER (b) RG (c) BA (d) RPL

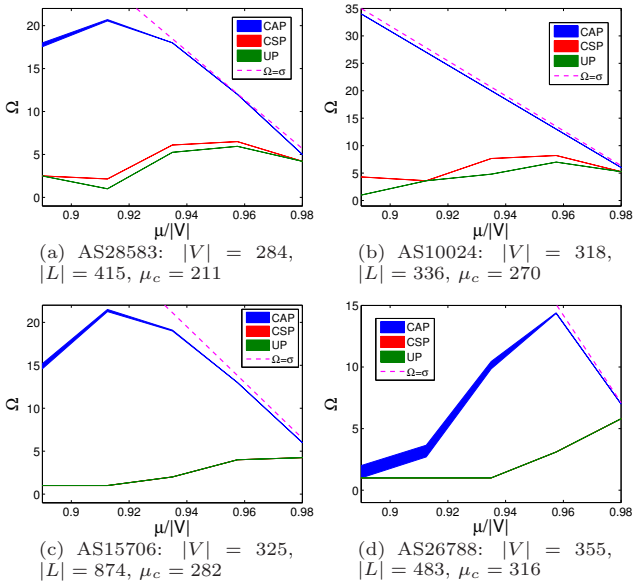
Figure 7: Bounds on the maximum identifiability Ω under CAP, CSP, and UP for densely-connected random topologies ($|V| = 20$, $\mu = 2, \ldots, 10$, $\mathbb{E}[|L|] = 99$, 50 graph instances per model).

AS6461 has the maximum ratio $\mu_c/|V|$ of approximately 0.665. Thus, we start the simulation from $\mu/|V| = 0.665$. Similar to the case of random topologies, the results show clear differences between the maximum identifiability under different probing mechanisms, especially between Ω^{CAP} and the other two. For most of the networks, UP and CSP only guarantee unique localization of single-node failures, while CAP can handle multi-node failures for all the networks. We also observe a much larger gap between Ω^{CAP} and Ω^{CSP} than the gap between Ω^{CSP} and Ω^{UP}, as the AS topologies are relatively sparse, leaving monitors little flexibility in selecting probing paths under cycle-free constraints (i.e., CSP). Across the networks, we observe that the ordering of the normalized maximum identifiability Ω/σ is roughly consistent with the minimum monitor fraction of monitors



(a) AS1755: $|V| = 172$, $|L| = 381$, $\mu_c = 69$ (b) AS6461: $|V| = 182$, $|L| = 294$, $\mu_c = 121$ (c) AS3967: $|V| = 201$, $|L| = 434$, $\mu_c = 106$ (d) AS3257: $|V| = 240$, $|L| = 404$, $\mu_c = 123$

Figure 8: Bounds on maximum Identifiability for Rocketfuel AS Topologies.



(a) AS28583: $|V| = 284$, $|L| = 415$, $\mu_c = 211$ (b) AS10024: $|V| = 318$, $|L| = 336$, $\mu_c = 270$ (c) AS15706: $|V| = 325$, $|L| = 874$, $\mu_c = 282$ (d) AS26788: $|V| = 355$, $|L| = 483$, $\mu_c = 316$

Figure 9: Bounds on maximum Identifiability for CAIDA AS Topologies.

$\mu_c/|V|$ required by ERMP: the larger $\mu_c/|V|$, the smaller Ω/σ.

Because ISP topologies have evolved since the Rocketfuel project, we repeat the above evaluation on a recent dataset obtained by the CAIDA project; see results in Fig. 9. Compared with the Rocketfuel ASes, we observe that the CAIDA ASes require more monitors (computed by step (i) of ERMP), e.g., the largest fraction $\mu_c/|V|$ is 0.89 for AS26788 while the smallest is 0.74 for AS28583. Hence, we start from $\mu/|V| = 0.89$ in Fig. 9. Similar to Fig. 8, we again observe a huge gap between Ω^{CAP} and the other two, and Ω/σ is inversely related to $\mu_c/|V|$. In addition, Fig. 9 also shows that under the minimum monitor placement, even the most flexible probing mechanism, CAP, can only localize failures of a couple of nodes. Our results suggest that without control

of probing paths (i.e., UP), randomly placed monitors are unlikely to guarantee unique failure localization; therefore, optimized monitor placement is needed to handle simultaneous failures of multiple nodes. Moreover, these results also imply that in the absence of deploying monitors at the vast majority of nodes, implementing controllable probing is an effective way to guarantee unique failure localization.

7. CONCLUSION

We have studied the fundamental capability of a network to localize failed nodes from the health condition of end-to-end paths between monitors. We proposed a novel measure, called the maximum identifiability, to quantify this capability as the maximum number of simultaneous failures that can be uniquely localized. We studied this measure in detail for three representative families of probing mechanisms that offer different tradeoffs between the controllability of probes and the cost of implementation. For each family of probing mechanisms, we established necessary/sufficient conditions for unique failure localization based on the network topology, the placement of monitors, the constraints on measurement paths, and the maximum number of simultaneous failures. We further showed that these conditions lead to tight upper/lower bounds on the maximum identifiability that differ by at most one. We showed that both the conditions and the bounds can be evaluated efficiently using polynomial-time algorithms. Our evaluations on random and real network topologies reveal that although incurring a higher implementation cost, giving the monitors more control over the routing of probes can significantly improve their capability to localize simultaneous failures.

8. REFERENCES

[1] R. R. Kompella, J. Yates, A. G. Greenberg, and A. C. Snoeren, "Detection and localization of network black holes," in *IEEE INFOCOM*, 2007.

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

[3] D. Ghita, C. Karakus, K. Argyraki, and P. Thiran, "Shifting network tomography toward a practical goal," in *ACM CoNEXT*, 2011.

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

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

[6] S. Zarifzadeh, M. Gowdagere, and C. Dovrolis, "Range tomography: Combining the practicality of boolean tomography with the resolution of analog tomography," in *ACM IMC*, 2012.

[7] A. Markopoulou, G. Iannaccone, S. Bhattacharyya, C.-N. Chuah, and C. Diot, "Characterization of failures in an IP backbone," in *IEEE INFOCOM*, 2004.

[8] N. Duffield, "Simple network performance tomography," in *ACM IMC*, 2003.

[9] ——, "Network tomography of binary network performance characteristics," *IEEE Transactions on Information Theory*, vol. 52, pp. 5373–5388, 2006.

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

[11] H. Zeng, P. Kazemian, G. Varghese, and N. McKeown, "Automatic test packet generation," in *ACM CoNEXT*, 2012.

[12] H. Nguyen and P. Thiran, "The boolean solution to the congested IP link location problem: Theory and practice," in *IEEE INFOCOM*, 2007.

[13] A. Dhamdhere, R. Teixeira, C. Dovrolis, and C. Diot, "Netdiagnoser: Troubleshooting network unreachabilities using end-to-end probes and routing data," in *ACM CoNEXT*, 2007.

[14] Y. Huang, N. Feamster, and R. Teixeira, "Practical issues with using network tomography for fault diagnosis," *ACM SIGCOMM Computer Communication Review*, vol. 38, pp. 53–58, 2008.

[15] H. X. Nguyen and P. Thiran, "Active measurement for multiple link failures diagnosis in IP networks," in *Passive and Active Measurement*, 2004.

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

[17] [Online]. Available: http://www.ietf.org/rfc/rfc0791.txt

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

[19] R. Dorfman, "The detection of defective members of large populations," *The Annals of Mathematical Statistics*, vol. 14, 1943.

[20] H.-G. Yeh, "d-Disjunct matrices: Bounds and Lovasz local lemma," *Discrete Math*, vol. 253, pp. 97–107, 2002.

[21] H. Gabow, "Using expander graphs to find vertex connectivity," *Journal of the ACM*, vol. 53, no. 5, pp. 800–844, September 2006.

[22] V. Chvatal, "A greedy heuristic for the set-covering problem," *Mathematics of Operations Research*, vol. 4, pp. 233–235, 1979.

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

[24] 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.

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

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

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

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

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