Node Failure Localization: Theorem Proof

Liang Ma†, Ting He†, Ananthram Swami§, Don Towsley*, and Kin K. Leung‡
†IBM T. J. Watson Research Center, Yorktown, NY, USA. Email: {maliang, the}@us.ibm.com
§Army Research Laboratory, Adelphi, MD, USA. Email: ananthram.swami.civ@mail.mil
*University of Massachusetts, Amherst, MA, USA. Email: towsley@cs.umass.edu
‡Imperial College, London, UK. Email: kin.leung@imperial.ac.uk

I. INTRODUCTION

Selected theorem proof in [1] are presented in detail in this report. We first list the theorems in Section II and then give the corresponding proofs in Section III. See the original paper [1] for terms and definitions. Table I summarizes all graph-theoretical notions used in this report (following the convention in [2]).

TABLE I

| Symbol | Meaning |
|---|---|
| $V$, $L$ | set of nodes/links |
| $M$, $N$ | set of monitors/non-monitors ($M \cup N = V$) |
| $G - L'$ | delete links: $G - L' = (V, L \setminus L')$, where \$\setminus\$ is setminus |
| $G + L'$ | add links: $G + L' = (V, L \cup L')$, where the end-points of links in $L'$ must be in $V$ |
| $G + G'$ | combine two graphs: $G + G' = (V(G) \cup V(G'), L(G) \cup L(G'))$, where $V(G)$ is the set of nodes and $L(G)$ is the set of links in $G$ |

II. THEOREMS

Lemma II.1. Algorithm 3 places the minimum number of monitors to ensure the network 1-identifiability in any given connected graph $G'$, where each biconnected component in $G'$ (i) has $\beta$ ($\beta = \{0, 1, 2\}$) neighboring biconnected components, (ii) has $2 - \beta$ non-cut-vertex nodes connecting to external monitors (can be outside $G'$), and (iii) is a PLC.

Theorem II.2. OMP-CSP ensures that any single-node failure in a given network is uniquely identifiable under CSP using the minimum number of monitors.

III. PROOFS

A. Proof of Lemma II.1

For Algorithm 3 the input network is not necessarily a connected graph. Lemma II.1, however, only considers the case that the input network satisfying the three conditions (in Lemma II.1) is connected. Then it suffices to show that Algorithm 3 places the minimum number of monitors to ensure that any two non-monitors in $G'$ (satisfying the three conditions in Lemma II.1) are distinguishable. Such connected input network can be represented as a tandem network, shown in Fig. 1. As illustrated in Fig. 1 suppose there are $z$ biconnected components in $G'$, where $B_3$ and $B_4$ have external monitor connections via $v_0$ and $v_2$ to $m_1$ and $m_2$ outside $G'$. (1) We first prove that any non-cut-vertex (excluding the nodes connecting to external monitors, e.g., $v_0$ and $v_2$ in Fig. 1, denoted by $w$, in $G'$ is 1-identifiable if $\exists$ another monitor, denoted by $m_3$, in $G'$ ($m_3$ can be anywhere in $G'$).

\[\text{Algorithm 3:} \text{ Monitors-in-Polygon-less-Network}(G', S)\]

\begin{verbatim}
input : Network topology $G$, node set $S$
output: Sub-set of nodes in $G$ as monitors
1 if $|L| = 0$ then
2 return;
3 end
4 foreach connected component $G_i$ in $G$ do
5 if $G_i$ contains only one biconnected component then
6 randomly choose a node in $G_i$ as a monitor;
7 else
8 in $G_i$, label one biconnected component with 0 or 1 cut-vertex as $B_1$, one neighboring biconnected component of $B_1$ as $B_2$ (if any), and one neighboring biconnected component of $B_2$ other than $B_1$ as $B_3$ (if any);
9 if $B_2$ is a bond then
10 choose the common node between $B_1$ and $B_2$ as a monitor;
11 $G_i' \leftarrow G_i \cup (B_1 + B_2)$;
12 else $//B_2$ is not a bond
13 randomly choose node $v (v \notin S)$ in $B_2$ as a monitor;
14 $G_i' \leftarrow G_i \cup (B_1 + B_2 + B_3)$ (if $B_3$ exists);
15 end
16 end
17 end

Fig. 1. Necessary monitor placement in tandem networks.
\end{verbatim}
The case that $m_1 \neq m_2$. Suppose $w$ is in biconnected component $B_i$. Then according to Theorem 15, $w$ is distinguishable from any other node (including cut-vertices or nodes connecting to external monitors) in $B_i$. Moreover, for a node outside $B_i$, say $u$, it is impossible that measurement path (from $m_1$ to $m_2$ in Fig. 1) must go through $u$ and $w$ at the same time, since $w$ must be a cut-vertex in Fig. 1 otherwise, contradicting the assumption. Therefore, $w$ is also distinguishable from nodes outside $B_i$. Thus, $w$ is 1-identifiable when $m_1 \neq m_2$ even without $m_3$.

The case that $m_1 = m_2$. To ensure each node in $G'$ is 1-identifiable, at least one extra monitor (besides $m_1$ and $m_2$) that can generate simple measurement paths traversing $G'$ is required. Now we have monitor $m_3$ in $G'$. In this case, $G'$ can be further decomposed into subgraphs. Then the argument in (1.1) applies to each subgraph by using $m_3$.

In sum, $w$ is 1-identifiable in $G'$, i.e., any non-cut-vertex (excluding the nodes connecting to external monitors) in $G'$ is 1-identifiable if $\exists$ a monitor in $G'$.

(2) Based on the argument in (1), we know that additional monitor placement (besides $m_1$ and $m_2$) in Fig. 1 in $G'$ is only for distinguishing the cut-vertices and nodes connecting to external monitors. Now we consider how to place the minimum number (at least one monitor, thus the argument in (1.1) still holds) of monitors to distinguish all these nodes in $G'$. In Algorithm 1 line 2 assigns a sequence number to each biconnected component. This is feasible as $G'$ is a tandem network. Suppose $|B_i| \geq 3$ (i.e., $B_i$ is not bond). Then, as Fig. 1 illustrates, to ensure the 1-identifiability of $v_0$, there are three possible locations for necessary monitor placement: (i) $v_0$ connects to another external monitor; (ii) place a monitor in $B_1$; or (iii) select a monitor from $V(B_2) \setminus \{v_2\}$. Similarly, there are also three possible locations for monitor placement such that $v_1$ and $v_2$ are 1-identifiable. To ensure the 1-identifiability of $v_0$, $v_1$, and $v_2$, we notice that there exists a common location, i.e., a node in $V(B_2) \setminus \{v_1, v_2\}$, where placing one monitor can guarantee that $v_0$, $v_1$, and $v_2$ are all 1-identifiable. Moreover, no other places can guarantee that $v_0$, $v_1$, and $v_2$ are 1-identifiable at the same time. Thus, line 13 selects a monitor, denoted $m_3$, from $V(B_2) \setminus \{v_1, v_2\}$. Note that the selection of $m_3$ only guarantees that $v_0$, $v_1$, and $v_2$ are 1-identifiable, i.e., the identifiability of all other nodes remain the same. Hence, $m_3$ does not affect the necessity of previously deployed monitors. However, if $B_2$ is a bond and $B_1$ is not a bond, then to ensure that $v_1$ is 1-identifiable, there are only two possible locations for monitor placement: (i) $v_0$ connects to another external monitor or (ii) place a monitor in $B_1$. Meanwhile, to ensure that $v_1$ is 1-identifiable, the possible monitor locations are also reduced to two: (i) place a monitor in $B_1$ or (ii) place a monitor in $B_2$. In such case, the common location to ensure the 1-identifiability of both $v_0$ and $v_1$ is in $B_1$. Thus, line 13 deploys a monitor at the common node between $B_1$ and $B_2$. As aforementioned, for this newly selected monitor, it is only for identifying a specific set of nodes. All previously deployed monitors remain necessary. After this placement, lines 11 and 14 remove the processed biconnected components. Then the remaining graph is processed by Algorithm 5 recursively, as further monitor placement is independent of the monitors that are already deployed. Finally, one trivial case we have not discussed is that $G_i$ contains only one biconnected component, where a randomly chosen monitor (line 6) can ensure the network 1-identifiability. Therefore, for the input network satisfying the conditions in Lemma 11, Algorithm 1 can place the minimum number of monitors for achieving the network 1-identifiability.

B. Proof of Theorem II.2

First, we consider the case that the input connected network is not 2-connected. In this case, there exist at least one cut-vertex and two biconnected components, and auxiliary algorithm Algorithm A is not invoked. For such input network, we discuss it as follows.

1) We first place necessary monitors in each biconnected component. If biconnected component $B_i$ has only one cut-
vertex, denoted by \( v_c \), then at least one non-cut-vertex in \( B_i \) should be a monitor.

(1.a) If \( B_i \) is a PLC, then we can randomly select a non-cut-vertex, denoted by \( m_i \), as a monitor (line 6) according to Theorem 15 and Corollary 16 [1], such that any node \( w \) with \( w \in V(B_i) \setminus \{v_c\} \) is distinguishable from any node in \( V(B_1) \). Moreover, \( w \) is also distinguishable from nodes outside \( B_i \) due to the existence of path from \( m_i \) to \( v_c \) without traversing \( w \). Therefore, all nodes in \( V(B_i) \setminus \{v_c\} \) are 1-identifiable when \( B_i \) is a PLC and has only one cut-vertex.

(1.b) If \( B_i \) is not a PLC, i.e., \( B_i \) contains at least one polygon, then the required monitor in \( B_i \) cannot be randomly placed. For this case, we need to first find all nodes in \( B_i \) that are guaranteed to be 1-identifiable. For these 1-identifiable nodes, there are three cases:

1. A PLC with 3 or more agents. As shown in Fig. 2 PLC \( \Lambda_1 \) has three agents and \( B_i \) must have a monitor (say \( m_{B_i} \)) that is not \( v_c \). Moreover, outside \( B_i \), \( v_c \) must connect to a monitor, e.g., a monitor in \( \Lambda_4 \) (denoted by \( m_{\Lambda_4} \)). Within \( \Lambda_1 \), for any two nodes \( u_1 \) and \( u_2 \) (\( u_1 \neq u_2 \neq v_c \)), we can find a path traversing only \( u_1 \) but not \( u_2 \) (and vice versa) using \( m_{B_i} \) and \( m_{\Lambda_4} \) as \( \Lambda_1 \) is a PLC. Therefore, each node in \( V(\Lambda_1) \setminus \{v_c\} \) is 1-identifiable. Then following the similar argument in (1) of Section III-A we know each node in \( V(A_2 + A_3 + A_4 + A_5) \setminus \{v_j\} \) is 1-identifiable, where \( A_2, A_3, A_4, \) and \( A_5 \) are the neighboring PLCs of \( \Lambda_1 \).

2. A PLC with 4 or more neighboring PLCs. This case is illustrated in Fig. 3 and Fig. 4. Following the similar argument in (1) of Section III-A we know that each node in \( V(A_1 + A_2) \setminus \{v_1, v_2, v_c\} \) is 1-identifiable, where \( A_1 \) and \( A_2 \) have the common node \( v_c \). However, \( v_1 \) and \( v_2 \) may or may not be distinguishable depending on the topology of \( B_i^* \).

Lines [10][11] consider the above three cases to remove all 1-identifiable nodes. Note that in lines [10][11] we get sets \( A, C, \) and \( E \), the common non-cut-vertices among these three sets may not be marked as 1-identifiable in any of the above three cases. Nevertheless, we can prove these common non-cut-vertices are also 1-identifiable as follows: Let \( U_c \) be the set containing all such common non-cut-vertices, and \( U_r = A \cup C \cup E \setminus U_c \). Now consider a random node \( z \) with \( z \in U_c \) and another node \( x \). There are 3 cases: (i) If \( x \in U_r \), then we know \( x \) is distinguishable from \( z \) based on previous results; (ii) if \( x \in U_c \), then \( x \) is a path traversing \( x \) without going through \( z \) using nodes in sets \( A, C, \) and \( E \); (iii) if \( x \not\in U_r \cup U_c \), then as (ii) shows that \( z \) and \( x \) are also distinguishable because of the existence of paths bypassing \( z \) using nodes in \( U_r \cup U_c \).

In Fig. 2 and Fig. 3, \( v_c \) is also temporarily removed; however, it still exists in neighboring biconnected components and the 1-identifiability of \( v_c \) will be considered later in lines [16][20].
Therefore, the common non-cut-vertex $z$ is 1-identifiable. Using this union set $A \cup C \cup E$, the remaining graph obtained by line 12 is a collection of tandem networks. Within this collection of tandem networks, consider two non-monitors $w_1$ and $w_2$ in two different connected tandem networks. We can show that $w_1$ and $w_2$ are distinguishable. This is because each connected tandem network must have additional necessary monitors. Using these additional monitors and also the removed components, we can find paths traversing only $w_1$ or $w_2$. Moreover, for these additional monitors, each has at least two internally vertex disjoint paths to any cut-vertex in the parent biconnected component. Thus, each non-monitor in one of these connected tandem network is distinguishable from any non-monitor outside its parent biconnected component. Therefore, it suffices to only consider how to enable the 1-identifiability in each of these connected tandem networks. This goal can be achieved by Algorithm 3; the correctness of which is shown in Lemma [1,1]

(2) In processing each biconnected component, we only place the necessary monitors. These necessary monitor placements are proved to be able to ensure that all non-cut-vertex nodes in biconnected components are 1-identifiable. Next, we can consider the 1-identifiability of cut-vertices. Note that for a biconnected component, if no necessary monitors are placed so far, then this means this biconnected component is either a PLC or contains a sufficient number of cut-vertices so that no additional monitors are required. For these biconnected components without monitors, it is still possible to find 1-identifiable cut-vertices in the following two cases:

1. A biconnected component with 2 cut-vertices and 3 or more neighboring biconnected components, such as $B_1$ in Fig. 5. In this case, we know that $v_2$ must have two monitor connections, one through $B_2$ and the other through $B_3$. Meanwhile, $v_1$ also has a monitor connection through $B_4$. Using these monitor connections, each node in $V(B_1) \setminus \{v_1\}$ is 1-identifiable. Note that unlike our previous discussion in (1) that connecting point may not be 1-identifiable. In Fig. 5 cut-vertex $v_2$ is guaranteed to be 1-identifiable, because it has more than two internally vertex disjoint monitor connections.

2. A biconnected component with 3 or more cut-vertices, such as $B_1$ in Fig. 6. In this case, $v_i$ (i = \{2, 3, 4\}) must have a monitor connection through $B_i$. Using these monitor connections, all nodes in $B_1$ (including the cut-vertices) are 1-identifiable. Similarly, nodes in $V(B_2 + B_3 + B_4) \setminus \{v_5, v_6, v_7\}$ are also 1-identifiable, where $B_2$, $B_3$, and $B_4$ are neighboring biconnected components of $B_1$.

Lines [17][18] consider all above cases to determine the 1-identifiable nodes. Removing the 1-identifiable nodes by line 19 we get a collection of tandem networks without containing any monitors. Following our previous arguments about monitor placement in a collection of tandem networks (arguments after (3) in (1)), we further deploy monitors optimally by lines [19][20]

In this way, we use the minimum number of monitors to ensure that any cut-vertex is 1-identifiable in a given non-2-connected network.

Finally, we consider the case that the input connected network is 2-connected. In this case, there are no cut-vertices. However, it is still possible to determine some 1-identifiable nodes. Specifically, Case-(2) in (1) can be applied to 2-connected network $G$ with 2 or more polygons. This particular case is handled by line [4][2] in Algorithm A. However, if no 1-identifiable nodes can be found, then it implies that 2-connected $G$ itself is a PLC or contains one and only one polygon. If the given 2-connect network is a PLC, then randomly selecting two monitors (line 2 of Algorithm A) can ensure network 1-identifiability according to Theorem 15 [1]. While for a 2-connected network with one and only one polygon, our strategy is to deploy the first monitor, remove the 1-identifiable nodes using our previous methods in Algorithm 2 and then apply Algorithm 3 to optimally deploy monitors in the remaining graph. For a 2-connected network with only one polygon, there are two cases:

1. All PLCs in $G$ are non-bonds. For this case, there is no difference in selecting which PLC to deploy the first monitor because all PLCs have the same structure with at least one non-agent node. This case is captured by lines [9][14] in Algorithm A.

2. There is at least one bond in $G$. In this case, we randomly select a bond PLC, denoted by $\Lambda'$ with two end-points $v_1$ and $v_2$ and two neighboring PLCs $\Lambda_1 (v_1 \in \Lambda_1)$ and...
Algorithm A: Monitors-in-Biconnected-Network

input: 2-connected network $G$ and its PLCs
output: Set of monitors that achieves 1-identifiability in $G$ under CSP

1. if $G$ is a PLC then
   2. randomly select two nodes as monitors; return;
end

4. if $\exists$ non-empty set $A$ containing all PLCs with 4 or more neighboring PLCs within $G$ then
   5. find set $C$ containing all neighboring PLCs of each PLC in set $A$ within $G$;
   6. $G' \leftarrow G \ominus (A \cup C)$;
   7. Monitors-in-Polygon-less-Network($G'$, $S_a$), where $S_a$ denotes the set of agents in $G$;
//In the following cases, $G$ must contain only one polygon;

9. else if all PLCs in $G$ are non-bonds then
   10. randomly select a non-agent node in a PLC, denoted by $\Lambda$, as a monitor;
   11. $G' \leftarrow G \ominus (\{\Lambda\} \cup E)$, where $E$ is the set containing all neighboring PLCs of $\Lambda$;
   12. Monitors-in-Polygon-less-Network($G'$, $S_a$);
   13. else // $\exists$ at least one bond PLC
      14. randomly select a bond PLC $\Lambda'$ with two end-points $v_1$ and $v_2$ and two neighboring PLCs $\Lambda_1$ ($v_1 \in \Lambda_1$) and $\Lambda_2$ ($v_2 \in \Lambda_2$);
      15. if $\Lambda_1$ ($\Lambda_2$) is a bond then
         16. $w_1 \leftarrow v_1$ ($w_2 \leftarrow v_2$);
      17. else
         18. $w_1$ ($w_2$) $\leftarrow$ a (random) non-agent node in $\Lambda_1$ ($\Lambda_2$);
      19. end
      20. foreach $i = 1, 2$ do
         21. select $w_i$ as a monitor;
         22. $G'_i \leftarrow G \ominus \Gamma_{w_i}$, where $\Gamma_{w_i}$ is the set involving (i) PLCs that contain $w_i$, and (ii) neighboring PLCs of the PLCs in (i) if $w_i$ is not an agent;
      23. Monitors-in-Polygon-less-Network($G'_i$, $S_a$);
      end
   24. in above two monitor placements, select the one with the minimum number of monitors as the final output;
end

$\Lambda_2$ ($v_2 \in \Lambda_2$), shown in Fig. 7. To distinguish $v_1$ and $v_2$ in Fig. 7 we need a monitor in $\Lambda_1$ or $\Lambda_2$ or both. Depending on if $\Lambda_1$ or $\Lambda_2$ is a bond, we have to select $v_1$ or $v_2$ as a monitor. To get such monitor candidates, we use lines 15–19 to get $w_1$ and $w_2$ for possibly placing the first monitor. Unfortunately, we have no knowledge on which one (selecting $w_1$ or $w_2$ as a monitor) can generate the optimal solution. Therefore, we test them both, and select the one with the minimum number of monitors as the final output; see lines 20–25 of Algorithm A.

In all, the above discussion on 2-connected input network is complete to cover all cases of 2-connected networks.

Consequently, OMP-CSP (Algorithm 2) can guarantee network 1-identifiability using the minimum number of monitors for any given network topology.

References

[1] L. Ma, T. He, A. Swami, D. Towsley, and K. K. Leung, “On optimal monitor placement for localizing node failures via network tomography,” in IFIP WG 7.3 Performance, 2015.

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