Localization of Single Link-Level Network Anomalies: Problem Formulation and Heuristic

Emna Salhi, Samer Lahoud, Bernard Cousin

Localization of Single Link-Level Network Anomalies: Problem Formulation and Heuristic

Emna Salhi
University of Rennes 1
IRISA, France
emna.salhi@irisa.fr

Samer Lahoud
University of Rennes 1
IRISA, France
samer.lahoud@irisa.fr

Bernard Cousin
University of Rennes 1
IRISA, France
bernard.cousin@irisa.fr

***Abstract*—Achieving accurate, cost-efficient, and fast anomaly localization is a highly desired feature in computer networks. A necessary and sufficient condition on the set of paths that need to be monitored upon detecting a single link-level anomaly in order to localize its source unambiguously have been established. However, this paper demonstrates that this condition is sufficient but not necessary. A necessary and sufficient condition that reduces the localization overhead, cost and delay significantly, as compared to the existing condition, is established. Furthermore, an ILP algorithm that selects monitoring paths and monitor locations which satisfy the established condition jointly, thereby enabling a trade-off between the number and locations of monitoring devices and the quality of monitoring paths, is devised. The problem is shown to be $\mathcal{NP}$-hard through a polynomial-time reduction from the $\mathcal{NP}$-hard facility location problem, and therefore, a scalable near-optimal heuristic is proposed. The effectiveness and the correctness of the proposed anomaly localization scheme are verified through theoretical analysis and extensive simulations.**

***Index Terms*—Network monitoring, network diagnosis, anomaly localization, anomaly detection, link-level anomalies.**

I. Introduction

Anomaly localization aims at identifying unambiguously the link that causes an anomalous behavior of the network (*e.g.* excessive delay, high packet loss rate, etc.). Recent research works argued that continuous anomaly localization results in high overhead on the underlying network, and hence, is likely to interfere with the network services . Subsequently, most recent monitoring schemes proceed in two phases (*e.g.* [1] [2] [3] [12] [13] [14] [15]). The first phase, the detection phase, uses as few network resources as possible to only detect anomalies. A necessary and sufficient condition to detect all link-level anomalies is to cover all links of the network. Upon detecting an anomaly, the detection phase returns a set of suspect links. The localization phase is triggered then. It aims at reducing the set of suspect links to the anomalous link(s). Clearly, this reactive anomaly localization approach reduces significantly the monitoring overhead compared to the continuous anomaly localization approach. However it presents a serious challenge: *the localization must be as fast as possible, in order to enable a fast recovery of the network*.

Agrawal et al. [1] proposed an accurate link-level anomaly localization scheme that can localize all potential single link-level anomalies in a given network. The key idea is to deploy resources that enable the monitoring of a set of paths that distinguishes all links of the network pairwise. Two links are said to be distinguished from each other if we are able to decide which one is anomalous when an anomaly occurs on one of them. Whenever an anomaly is detected, this set of paths is monitored in order to pinpoint the anomalous link. This technique is suboptimal in that it considers all the network links as suspect, ignoring the information provided by the detection process, which generates unnecessary overhead and delays the localization. More recently, Barford et al. [2] proposed another scheme that selects paths that are to be monitored during the localization phase. Although this technique minimizes the localization overhead, because the monitored paths distinguish only between the suspect links pairwise, it suffers from two imperfections. The first is the non-negligible time of computing the set of paths that are to be monitored upon detecting an anomaly, which increases the localization delay (*i.e.*, time elapsed between the moment when an anomaly is detected and the moment when the anomalous link is pinpointed). The second is that there is no guarantee to localize all potential anomalies, because the deployed monitors ensure only the coverage of links [1]. In this paper, we demonstrate that 1) not all links of the network need to be distinguishable pairwise for localizing any potential anomaly, 2) all potential anomaly scenarios can be derived offline from any detection solution that covers all the network links. Thus, we compute full and cost-efficient localization solutions, *i.e.*, monitors that are to be activated and paths that are to be monitored upon detecting an anomaly, for all potential anomalies offline. Subsequently, we achieve an important gain in the localization delay and overhead.

Multiple works propose to compute the set of paths that are to be monitored dynamically upon detecting an anomaly (*e.g.*, [4]-[10]). Practically, this means that one probe that maximizes the information gain given the previous probe observations is selected and sent in the network at a time. Such an approach is practical for highly dynamic environments. However, it is not practical for networks where anomalies are rare events,

This work was supported by Alcatel-Lucent Bell Labs, France, under the grant n. 09CT310-01

[1]The monitors used for anomaly detection are deployed such that all the network links are covered by at least one monitoring paths. They can not necessarily localize all potential anomalies

especially, because it yields excessive delays.

Furthermore, most existing works consider only one criterion for monitoring path selection that is the minimization of the number of monitored paths, and only one criterion for monitor location selection that is the minimization of the number of deployed monitoring devices (*e.g.*, [2] [1]). However, these criteria do not reflect the localization cost properly. Indeed, to reduce the localization delay and overhead, monitoring of links that do not provide extra localization information during the localization phase must be avoided. Moreover, monitor locations must be selected carefully towards minimizing the delay of communications between the Network Operation Center (NOC) and the deployed monitors. A novel anomaly localization cost model that considers the infrastructure cost, the localization overhead and the localization delay is, therefore, proposed. Besides, our anomaly localization scheme selects monitor locations and monitoring paths jointly, thereby enabling a trade-off between the number and locations of deployed monitoring devices and the quality of selected monitoring paths. We formulate the problem as an ILP, and we show that it is $\mathcal{NP}$-hard through a polynomial-time reduction from the facility location problem.

Prior works on anomaly localization propose greedy approaches for computing localization solutions (*e.g.*, [1], [2], [11], [12]). In order to ensure the scalability, the number of candidate monitoring paths should reduced to a small subset of the network paths. Unfortunately, none of these works described how candidate monitoring paths are selected, however, the choice of candidate paths has a great impact on the quality of the localization solution. In this work we propose a heuristic that implements our anomaly localization scheme. We devise an efficient algorithm for candidate path computation that makes the heuristic scalable and near-optimal at a time. The key idea is to use a mathematically proven properties that enable us to find the best candidate monitoring paths between two given monitor locations by exploring a very small proportion of the network paths.

We verify the effectiveness of our anomaly localization scheme through extensive simulations and by comparing it with an hybrid anomaly localization scheme that combines the strengths of the scheme proposed in [1] and the scheme proposed in [2].

II. Network Model and Problem Statement

We model the network as an undirected graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ comprising a set of nodes $\mathcal{N}$ connected by a set of undirected links [2] in $\mathcal{E}$. Let $\mathcal{P}$ be the set of all non-looping paths of the network. Unless otherwise mentioned, without loss of generality, we assume that all paths in $\mathcal{P}$ are candidate to be monitored and all the network nodes are candidate to support monitoring devices. We use the term monitoring paths to designate paths that are monitored during the detection phase, also referred to as detection paths, or during the localization phase, also referred to as localization paths. We consider that a network path is a set of links, instead of a sequence of links, and therefore, we apply set operations (*e.g.*, $\cap, \cup$) on paths. We denote the anomaly detection solution by $(\mathcal{D}_m, \mathcal{D}_p)$. $\mathcal{D}_m$ is the set of monitor locations where to deploy monitoring devices. $\mathcal{D}_p$ is a set of monitoring paths between the selected monitor locations that covers all the network links, $\cup_{p \in \mathcal{D}_p} p = \mathcal{E}$. Furthermore, We note that we use the vertical bar notation to denote both set cardinality and absolute value.

We consider separable anomalies (*e.g.*, connectivity, high-low loss model, delay spike model, etc) that satisfy the following property: *a path experiences an anomaly if and only if at least one of its constituent links is anomalous* [16]. According to this property all links that are traversed by at least one detection path not exhibiting an anomaly are not anomalous, and all paths crossing an anomalous links exhibit the same anomaly. The remaining links constitute the set of suspect links. Anomaly localization aims at reducing the set of suspect links, inferred upon detecting an anomaly from the detection information, to the anomalous link. This requires monitoring additional paths that can distinguish between suspect links pairwise. Two links are said to be distinguishable from each other if we are able to decide which one is anomalous when an anomaly occurs on one of them.

The objective of this work is to come up with a localization scheme that enables the localization of all potential link-level anomalies accurately; while minimizing the cost of acquiring and deploying monitoring devices, the localization overhead and the localization delay. Our localization scheme infers all potential anomaly scenarios from any detection solution that covers all links of the network. This has two major benefits. The first is that we do not need to monitor a set of paths that can distinguish between every single pair of the network links whenever an anomaly is detected. The second is that we pre-compute full localization solutions for all anomaly scenarios offline, thereby accelerating the localization process. The inputs into our localization problem are an instance of the graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ and a set of detection paths $\mathcal{D}_p$ that covers all links in $\mathcal{E}$, and the outputs are a set of monitor locations whose monitors are to be activated and a set of paths that are to be monitored for each potential anomaly. The localization solution must achieve a good trade-off between the monitor deployment cost, the localization overhead and the localization delay. To this end, a novel cost model that measures these three metrics is proposed. Also, our localization scheme selects monitor locations and localization paths jointly; as opposed to existing schemes that apply a two-step selection procedure, therefore omitting the trade-off between the number and locations of monitors and the quality of localization paths.

III. Not all link pairs need to be distinguishable for localizing any single link-level anomaly

In this section, we first establish a necessary and sufficient condition to distinguish between two links. Then, we prove that not all link pairs need to be distinguishable for localizing any potential single link-level anomaly accurately. This excludes an already established condition claiming that it is

[2]This work can be easily applied for directed links. Each directed link is duplicated into

necessary to monitor a set of paths that can distinguish between all links of the network pairwise whenever an anomaly is detected [1].

Theorem 1: The necessary and sufficient condition for two links e_1 and e_2 to be distinguishable from each other is the existence of a monitoring path that crosses either e_1 or e_2, but not both.

Proof: We first demonstrate the sufficiency condition. Assume that either e_1 or e_2 is anomalous. Let p be a path that crosses e_1 (interchangeably e_2) but not e_2 (interchangeably e_1). If p exhibits an anomaly, then the anomalous link must be covered by p. We conclude that e_1 is the anomalous link. If, p does not exhibit an anomaly, then all its constituent links are not anomalous. It follows that the anomalous link is e_2. Thus, p is sufficient to distinguish between e_1 and e_2.

The necessary condition can be proved as follows. Assume that there does not exist any path that crosses only one of the two links. Then, the monitoring path set can be divided into two types of paths: paths that cross both e_1 and e_2, and paths that neither cross e_1 nor e_2. An anomaly on a given link affects all the monitoring paths that cross that link. Therefore, the latter type of paths is not affected by the anomalies that occur on any the two links, whereas the former type of paths is affected by the anomalies that occur on any of the two links. Thus, the set of monitoring paths that are affected by an anomaly on e_1 is exactly the same set of paths that is affected by an anomaly on e_2. This means that e_1 and e_2 cannot be distinguished from each other. ■

Existing localization schemes (*e.g.*, [1]) claim that all links of the network must be distinguished pairwise in order to localize any potential anomalies. According to Theorem 1, this means that $\forall e_1, e_2 \in \mathcal{E}$ there exists a localization path that crosses either e_2 or e_2, but not both. However, we will demonstrate that this is a sufficient but not necessary condition, and we show how to infer the minimal set of pair of links that are to be distinguished from a given detection solution that covers all the network links.

Consider a network link $e \in \mathcal{E}$. We denote by D_{e_+} and D_{e_-} the set of detection paths that cross e and the set of detection paths that do not cross e, respectively. The set of suspect links associated to an anomaly on a link e is the set of all links that cannot be distinguished from e using only the detection information.

Theorem 2: The set of suspect links associated to an anomaly on a given link $e \in \mathcal{E}$ equals $\cap_{p \in D_{e_+}} p - \cup_{p \in D_{e_-}} p$.

Proof: We prove this theorem by construction. The set of detection paths can be divided into two sets:

- D_{e_+}: paths that cross link e.
- D_{e_-}: paths that do not cross link e.

An anomaly on link e affects only paths that cross this link. Subsequently, paths in D_{e_-} do not exhibit an anomaly. It follows that all the links that are traversed by paths in D_{e_-} are not suspect. Now, let L be the set of links that are traversed by paths in D_{e_+} and that are not traversed by paths in D_{e_-}, $L = \cup_{p \in D_{e_+}} p$ - $\cup_{p \in D_{e_-}} p$. L can be divided into two subsets of links:

- L_1: links that do not belong to $\cap_{p \in D_{e_+}} p - \cup_{p \in D_{e_-}} p$
- L_2: links that belong to $\cap_{p \in D_{e_+}} p - \cup_{p \in D_{e_-}} p$

We prove by contradiction that all links in L_1 are not suspect. Assume to the contrary that a link $l \in L_1$ is suspect. This means that there does not exist any path in D_{e_+} that distinguishes between l and e. It follows that for each $p \in D_{e_+}$, p crosses e and l. Thus $l \in \cap_{p \in D_{e_+}} p - \cup_{p \in D_{e_-}} p$, leading to a contradiction.

Likewise, we prove by contradiction that all links in L_2 are suspect. Assume to the contrary that a link $l \in L_2$ is not suspect, then, there exists at least one path $p \in D_{e_+}$ such that p distinguishes between e and l. Since all paths in D_{e_+} cross e, then p does not cross l. It follows that $l \notin \cap_{p \in D_{e_+}} p - \cup_{p \in D_{e_-}} p$, leading to a contradiction. ■

Corollary 1: A sufficient and necessary condition for localizing any potential link-level anomaly is to distinguish each link $e \in \mathcal{E}$ from links that belong to $\cap_{p \in D_{e_+}} p - \{\cup_{p \in D_{e_-}} p \cup \{e\}\}$.

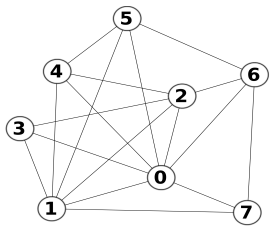
Let $\mathcal{S}(e)$ denotes the set of suspect links associated to anomalies on link e, $\mathcal{S}(e) = \cap_{p \in D_{e_+}} p - \{\cup_{p \in D_{e_-}} p \cup \{e\}\}$.

Corollary 2: $e_1 \in \mathcal{S}(e_2) \Leftrightarrow \mathcal{S}(e_1) = \mathcal{S}(e_2)$, $\forall e_1, e_2 \in \mathcal{E}$

Corollary 3: $\mathcal{S}(e_1) \neq \mathcal{S}(e_2) \Leftrightarrow \mathcal{S}(e_1) \cap \mathcal{S}(e_2) = \emptyset$

The properties presented in the above corollaries are demonstrated in Appendix A.

IV. Derivation of potential anomaly scenarios



(a)

Monitor locations	nodes 0, 1 and 7
Detection Paths	$\langle(0,7)\rangle$
	$\langle(0,1)\rangle$
	$\langle(0,4),(4,1)\rangle$
	$\langle(0,2),(2,3),(3,1),(1,7)\rangle$
	$\langle(0,6),(6,5),(5,4),(4,2),(2,1)\rangle$
	$\langle(1,5),(5,0),(0,3),(3,2),(2,6),(6,7)\rangle$

(b)

Fig. 1: Illustrative network topology, (a), and an associated detection solution, (b).

Theorem 2 states that the set of suspect links returned at the end of the detection phase whenever an anomaly on link e occurs is $\cap_{p \in D_{e_+}} p - \cup_{p \in D_{e_-}} p$. Therefore, instead of computing monitors that are to be activated and paths that are to be monitored during the localization phase whenever an anomaly is detected, we propose to perform these computations for all potential anomalies only once offline. Having a set of detection paths that cover all links of the network, we infer the set of suspect links for all potential anomalies as described in Theorem 2. Then, a single anomaly scenario is

TABLE I: Sets of suspect links for all potential anomalies

Anomalous link	Set of suspect links
$(0,1)$	{(0, 1)}
$(0,2)$	{(0, 2), (1, 3), (1, 7)}
$(1,2)$	{(0, 6), (5, 6), (4, 5), (2, 4), (1, 2)}
$(0,3)$	{(1, 5), (0, 5), (0, 3), (2, 6), (6, 7)}
$(1,3)$	{(0, 2), (1, 3), (1, 7)}
$(2,3)$	{(2, 3)}
$(0,4)$	{(0, 4), (1, 4)}
$(1,4)$	{(0, 4), (1, 4)}
$(2,4)$	{(0, 6), (5, 6), (4, 5), (2, 4), (1, 2)}
$(0,5)$	{(1, 5), (0, 5), (0, 3), (2, 6), (6, 7)}
$(1,5)$	{(1, 5), (0, 5), (0, 3), (2, 6), (6, 7)}
$(4,5)$	{(0, 6), (5, 6), (4, 5), (2, 4), (1, 2)}
$(0,6)$	{(0, 6), (5, 6), (4, 5), (2, 4), (1, 2)}
$(2,6)$	{(1, 5), (0, 5), (0, 3), (2, 6), (6, 7)}
$(5,6)$	{(0, 6), (5, 6), (4, 5), (2, 4), (1, 2)}
$(0,7)$	{(0, 7)}
$(1,7)$	{(0, 2), (1, 3), (1, 7)}
$(6,7)$	{(1, 5), (0, 5), (0, 3), (2, 6), (6, 7)}

TABLE II: Anomaly scenarios

Anomaly scenario	Set of suspect links
a_1	S_{a_1} = {(0, 2), (1, 3), (1, 7)}
a_2	S_{a_2} = {(0, 6), (5, 6), (4, 5), (2, 4), (1, 2)}
a_3	S_{a_3} = {(1, 5), (0, 5), (0, 3), (2, 6), (6, 7)}
a_4	S_{a_1} = {(0, 4), (1, 4)}

created for all links that have the same set of suspect links, *i.e.*, an anomaly scenario is created for each distinct set of suspect links. Let us denote by $\mathcal{A}$ the set of all anomaly scenarios, and let $\mathcal{S}_a$ denotes the set of suspect links associated to the anomaly scenario $a \in \mathcal{A}$. Let $dS = \{\mathcal{S}_a, \forall a \in \mathcal{A}\}$. dS have the following properties.

Corollary 4: $\cup_{e\in\mathcal{E}}\mathcal{S}(e) = \cup_{\mathcal{S}(i)\in dS}\mathcal{S}(i) = \mathcal{E}$

Corollary 5: $\sum_{\mathcal{S}(i)\in dS} \mid \mathcal{S}(i) \mid = \mid \mathcal{E} \mid$

Clearly, an upper bound of the number of anomaly scenarios, whatever the topology of network and whatever the detection solution, is the number of the network links. It is easy to show that when this bound is reached, the set of suspect links for an anomaly on link e, $\forall e \in \mathcal{E}$, is reduced to the link e. In such case, the localization of all potential anomalies is immediate from the detection information. According to Corollary 2, we need to deploy monitors that enable the monitoring of a set of paths distinguishing links of each anomaly scenario pairwise in order to ensure the localization of all potential anomalies.

To illustrate, consider the sample network topology depicted in Figure 1(a). An associated anomaly detection solution that covers all links of the network is depicted in Figure 1(b). We use Theorem 2 to compute the set of suspect links for all potential anomalies. The result is depicted in Table I. The sets of suspect links associated to link $(2,3)$ and link $(0,7)$ are unitary. When an anomaly occurs on one of these two links, there is no need to trigger the localization phase because the anomalous link is immediately pinpointed by intersecting the detection paths that exhibit the anomaly. Furthermore, four non-unitary anomaly scenarios (a_1, a_2, a_3, a_4) are created for this topology (see table II). These are the four distinct non-unitary sets of suspect links.

Let $AllPairs$ denotes the number of all the network link pairs. Clearly, $AllPairs = (\mid \mathcal{E} \mid (\mid \mathcal{E} \mid -1))/2$. Let $dPairs$ denotes the number of pair of links that need be distinguishable for localizing any potential link-level anomaly.

Corollary 6: $dPairs = AllPairs - \sum_{\mathcal{S}(i),\mathcal{S}(j)\in dS:i<j} \mid \mathcal{S}(i) \mid\mid \mathcal{S}(j) \mid$

Corollary 6 confirms that we do not need to distinguish between all the network link pairs unless the number of detection paths equals 1, which is very unlikely.

The proofs of Corollary 4, Corollary 5 and Corollary 6 are described in Appendix A.

V. Anomaly localization cost

Consider a set of candidate monitor locations, $\mathcal{M}$, a set of network paths that are candidate to be monitored, $\mathcal{P}$, and a set of anomaly scenarios $\mathcal{A}$. The anomaly localization cost includes two costs:

- *Monitor cost*: it includes the effective cost of acquiring hardware and software monitoring devices and the cost of their maintenance. In addition, it includes the cost of communications between the monitors and the NOC. For instance, the cost of communications between a monitor and the NOC can be expressed as a function of the number of routing hops that separates them. Let us denote by C_n the cost of deploying a monitor on node n. Let Y_n be a binary variable that indicates whether node n is selected to hold a monitoring device. The monitor cost can be expressed as follows:
$$\sum_{n\in M} C_n Y_n \tag{1}$$
- *Probe cost*: it expresses the overhead of monitoring flows on the underlying network. Measurements of links that do not provide localization information should be avoided in order to minimize the monitoring overhead. Clearly, measuring links that do not belong to the set of suspect links of an anomaly scenario does not provide any extra localization information. Furthermore, measurement of links that belong to the set of suspect links might be useless. Revisit Figure 1 and table I to illustrate. Consider an anomaly on link $(6,7)$. The associated set of suspect links is $\mathcal{S}_{a_3} = \{(1,5),(0,5),(0,3),(2,6),(6,7)\}$. Consider now the set of localization paths $\{p_1{:}\langle(1,5)(5,6)(2,6)\rangle$; $p_2{:}\langle(1,5)(0,5)(0,2)\rangle$; $p_3{:}\langle(1,7)(6,7)(2,6)\rangle\}$ that distinguishes between all the links of $\mathcal{S}_{a_3}$ pairwise. Path p_1 divides $\mathcal{S}_{a_3}$ into two subsets: $\mathcal{S}^1_{a_3}\{(1,5),(2,6)\}$ and $\mathcal{S}^2_{a_3}\{(0,5),(0,3),(6,7)\}$. p_1 distinguishes each link of $\mathcal{S}^1_{a_3}$ from each link of $\mathcal{S}^2_{a_3}$. Link $(5,6)$ that is traversed by p_1 does not belong to $\mathcal{S}_{a_3}$, and therefore, it does not provide any localization information. Path p_2 divides $\mathcal{S}^1_{a_3}$ into two subsets: $\mathcal{S}^{11}_{a_3}\{(1,5)\}$ and $\mathcal{S}^{12}_{a_3}\{(2,6)\}$, and divides $\mathcal{S}^2_{a_3}$ into two subsets: $\mathcal{S}^{21}_{a_3}\{(0,5),(6,7)\}$ and $\mathcal{S}^{22}_{a_3}\{(0,3)\}$. Finally, p_3 distinguishes between $(0,5)$ and $(6,7)$. However, it crosses $(2,6)$ that is already distinguished from all the other suspect links. Thus, measuring $(2,6)$ by p_3 does not provide extra localization information, although it belongs to $\mathcal{S}_{a_3}$.

Let us denote by C_e the cost of measuring link e. C_e should be proportional to the load of link e, in order to avoid multiple measurements of the most overloaded links of the network.

Consider an anomaly scenario $a \in \mathcal{A}$. Let us denote by $\mathcal{S}_a$ the set of suspect links associated to the anomaly scenario a. Let X_{pa} be a binary variable that specifies whether path p is part of the localization solution of a. Let δ_{pe} be a binary input parameter that indicates whether path p crosses link e. The probe cost of the localization solution of a reads as follows:

$$\sum_{e \in \mathcal{E}, p \in \mathcal{P}'} C_e \delta_{pe} X_{pa} \tag{2}$$

VI. ILP Formulation

The objective of the ILP is to find a localization solution for each anomaly scenario in $\mathcal{A}$ such that the anomaly localization cost is minimized. Let δ_{pn} be a binary parameter that indicates whether node n is an end-node of path p. For simplicity of notation, we define the following sets:

- $\delta_{\mathcal{PE}} = \{\delta_{pe};\ p \in \mathcal{P}, e \in \mathcal{E}\}$
- $\delta_{\mathcal{PM}} = \{\delta_{pn};\ p \in \mathcal{P}, n \in \mathcal{M}\}$
- $C_{\mathcal{M}} = \{C_n;\ n \in \mathcal{M}\}$
- $C_{\mathcal{E}} = \{C_e;\ e \in \mathcal{E}\}$

Let α be the weight associated to the monitor cost, and let β be the weight associated to the probe cost. $\alpha, \beta \in \mathbb{R}$. The input into the ILP is an instance of the graph $G = (\mathcal{E}, \mathcal{M}, \mathcal{P}, \mathcal{A}, \delta_{\mathcal{PE}}, \delta_{\mathcal{PM}}, C_{\mathcal{E}}, C_{\mathcal{M}}, \alpha, \beta)$. The objective function minimizes the sum of the monitor cost and the probe cost. It reads as follows:

$$\alpha \sum_{n \in M} C_n Y_n + \beta \sum_{a \in \mathcal{A}, e \in \mathcal{E}, p \in \mathcal{P}} C_e \delta_{pe} X_{pa} \tag{3}$$

The ILP is subject to two constraints. The first constraint ensures that either end node of each selected monitoring paths is selected as monitor location. It reads as follows:

$$Y_n \geq \delta_{pn} X_{pa};\ \forall n \in \mathcal{M}, \forall p \in \mathcal{P}, \forall a \in \mathcal{A} \tag{4}$$

The second constraint ensures that the suspect links associated to each anomaly scenario are distinguishable pairwise. To this end, according to Theorem 2, the constraint ensures that for each anomaly scenario a and for each pair of suspect links $(e_1, e_2) : e_1, e_2 \in S_a$ there exists at least one monitoring path that crosses either e_1 or e_2, but not both. This constraint reads as follows:

$$\textstyle\sum_{p \in \mathcal{P}} (\delta_{pe_1} + \delta_{pe_2} - 2\delta_{pe_1}\delta_{pe_2}) X_{pa} > 0;\quad \forall a \in \mathcal{A}; \forall e_1, e_2 \in \mathcal{S}_a \tag{5}$$

We show that the above inequality is sufficient to distinguish between all the link pairs of each anomaly scenario using the argument of the following theorem.

Theorem 3: Let P_1 be the subset of paths of $\mathcal{P}$ that cross either e_1 or e_2, but not both. $\sum_{p \in \mathcal{P}} (\delta_{pe_1} + \delta_{pe_2} - 2\delta_{pe_1}\delta_{pe_2}) = \mid P_1 \mid$.

Proof: Refer to Appendix B. ∎

Corollary 7: ***If*** $\sum_{p \in \mathcal{P}} (\delta_{pe_1} + \delta_{pe_2} - 2\delta_{pe_1}\delta_{pe_2}) X_{pa} > 0$, ***then*** *there exists at least one path in $\mathcal{P}$ that crosses either e_1 or e_2 but not both,* ***then*** *there exists at least one path in $\mathcal{P}$ that distinguishes between e_1 and e_2.*

VII. The Anomaly Localization Problem is $\mathcal{NP}$-Hard

Theorem 4: The anomaly localization problem presented in the previous section is $\mathcal{NP}$-Hard.

Proof: Our formulation of the anomaly localization problem can be reduced from the $\mathcal{NP}$-Hard facility location problem.

Facility location problem [17]: consider a set of potential facility locations $\mathcal{F}$, and a set of clients $\mathcal{D}$. Opening a facility at location i incurs a non-negative cost that is equal to f_i. The cost of servicing client $j \in \mathcal{D}$ by a facility installed at location $i \in \mathcal{F}$ is d_{ij}. The problem is to find an assignment of each client to exactly one facility such that the sum of the facility opening costs and the service costs is minimized.

We denote by f the set of facility opening costs, $f = \{f_i,\ i \in \mathcal{F}\}$, and by d the set of service costs, $d = \{d_{ij};\ i \in \mathcal{F}, j \in \mathcal{D}\}$. Given an instance $\mathcal{I} = (\mathcal{D}, \mathcal{F}, f, d)$ of the facility location problem, we produce an instance $\mathcal{R}(\mathcal{I}) = (\mathcal{E}, \mathcal{M}, \mathcal{P}, \mathcal{A}, \delta_{\mathcal{PE}}, \delta_{\mathcal{PM}}, C_{\mathcal{E}}, C_{\mathcal{M}}, \alpha, \beta)$ of the localization problem as follows. For each client $j \in \mathcal{D}$, we create:

- Three nodes labeled by n_{j1}, n_{j2}, and n_{j3}.
- One link connecting n_{j1} to n_{j2}, labeled by e_{j1}.
- One link connecting n_{j2} to n_{j3}, labeled by e_{j2}.
- An anomaly scenario a_j such that $S_{a_j} = \{e_{j1}, e_{j2}\}$.

For each facility location $i \in \mathcal{F}$, we create two nodes labeled by m_{i1} and m_{i2}. For each $i \in \mathcal{F}$ and for each $j \in \mathcal{D}$, we create one link connecting m_{i1} to n_{j1}, labeled by e^1_{ij}, and one link connecting m_{i2} to n_{j2}, labeled by e^2_{ij}. We obtain a graph $\mathcal{G} = (\mathcal{E}, \mathcal{N})$, where $\mathcal{N} = \{n_{ik}; i \in \mathcal{D}, k \in [1;3]\} \cup \{m_{jk}; i \in \mathcal{F}, k \in [1;2]\}$, and $\mathcal{E} = \{e_{jk}; j \in \mathcal{D}, k \in [1;3]\} \cup \{e^k_{ij}; i \in \mathcal{F}, j \in \mathcal{D}, k \in [1;2]\}$. An example of a graph constructed out of a facility location instance with four facility locations and four clients is shown in Figure 2.

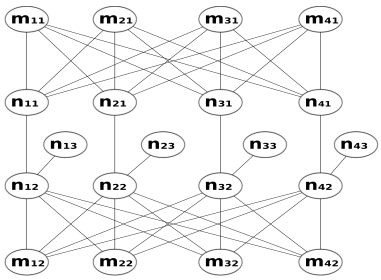


Fig. 2: Example of a graph constructed out of a facility location instance with four facility locations and four clients

The candidate monitor location set is $\mathcal{M} = \{m_{jk}; i \in \mathcal{F}, k \in [1;2]\}$. The set of anomaly scenarios is $\mathcal{A} = \{a_j; j \in \mathcal{D}\}$. The set of candidate localization paths is $\mathcal{P} = \{p_{ij}; i \in \mathcal{F}, j \in \mathcal{D}\}$, where p_{ij} is the non-looping path between m_{i1} and m_{i2} that crosses the links e^1_{ij}, e_{j1} and e^2_{ij}. The monitor deployment costs are defined as follows: $C_{m_{i1}} = C_{m_{i2}} = f_i/2$. The link measurement costs are defined as follows: $C_{e_{i1}} = C_{e_{i2}} = 0$, $C_{e^1_{ij}} = C_{e^2_{ij}} = d_{ij}/2$. The remaining input parameters can be inferred easily from $\mathcal{G}$, $\mathcal{M}$, $\mathcal{A}$ and $\mathcal{P}$ as follows:

- $\delta_{a_j e_{j'k}} = \begin{cases} 1 & \text{if } j = j' \\ 0 & \text{otherwise} \end{cases}; \quad \forall j, j' \in \mathcal{D}, k \in [1;2]$
- $\delta_{a_j e^k_{ij}} = 0; \quad \forall i \in \mathcal{F}, j \in \mathcal{D}, k \in [1;2]$
- $\delta_{p_{ij} m_{i'k}} = \begin{cases} 1 & \text{if } i = i' \\ 0 & \text{otherwise} \end{cases}; \quad \forall i, i' \in \mathcal{F}, k \in [1;2]$
- $\delta_{p_{ij} e_{j1}} = \delta_{p_{ij} e^1_{ij}} = \delta_{p_{ij} e^2_{ij}} = 1; \quad \forall i \in \mathcal{F}, j \in \mathcal{D}$
- $\delta_{p_{ij} e_{j2}} = 0; \quad \forall i \in \mathcal{F}, j \in \mathcal{D}$
- $\alpha = \beta = 1$

It can be easily shown that the time complexity of the above reduction is $\mathcal{O}(\mid \mathcal{F} \mid \times \mid \mathcal{D} \mid)$, and therefore, it can be carried out in polynomial-time. In the sequel, we show that there is an optimal solution to the Instance $\mathcal{I}$ of the facility location problem if and only of there is an optimal solution to the instance $\mathcal{R}(\mathcal{I})$ of our anomaly localization problem.

Let us start by demonstrating that if there is an optimal solution to the facility location instance, then there is a feasible solution to the anomaly localization instance. Let the facility location solution assign each client j to a facility installed at location i. Consider the anomaly localization solution that selects for each anomaly scenario a_j the path p_{ij} and the monitor locations m_{i1} and m_{i2}. Then, let us fix an anomaly scenario a_j. By construction, path p_{ij} crosses three links that are e_{j1} and e^1_{ij} and e^2_{ij}. It follows, according to Theorem 1, that p_{ij} distinguishes between e_{j1} and e_{j2}. Constraint (4) states that if p_{ij} is selected to be monitored, then, its end nodes must be selected to hold monitoring devices. Thus, the solution that selects for each anomaly scenario a_j the path p_{ij} to be monitored, and its end nodes, m_{i1} and m_{i2}, as monitor locations is a feasible solution to the anomaly localization instance.

Conversely, we demonstrate that if there is an optimal solution to the anomaly localization instance, then there is a feasible solution to the facility location instance. An optimal solution to the facility location problem selects exactly one path for each anomaly scenario. This is because, by construction, for each anomaly scenario $a_i \in \mathcal{A} \mid \mathcal{S}_{a_i} \mid = 2$. Thus, monitoring one path that crosses exactly one of the two links is sufficient to distinguish between them. Let the optimal anomaly localization solution selects for each anomaly scenario a_j the path p_{ij}, and naturally, the monitor locations m_{i1} and m_{i2}. Trivially, the solution that assigns to each client $j \in \mathcal{D}$ the facility installed at location i is a feasible solution to the facility location instance.

We now prove that the constructed anomaly localization solution has the same cost as its corresponding optimal facility location solution (the proof holds in the converse case). Let W_i be a binary variable that indicates whether a facility is installed at location i, and let Z_{ij} be a binary variable that indicates whether client j is serviced by a facility installed at location i. Using the arguments that $Z_{ij} = X_{p_{ij} a_j}$ and $W_i = Y_{m_{i1}} = Y_{m_{i2}}$ [3], we show that the cost of the localization solution, denoted by $Cost(S_{\mathcal{R}(\mathcal{I})})$, is equal to the cost of its corresponding facility location solution, denoted by $Cost(S_{\mathcal{I}})$, as follows:

$$\begin{aligned} Cost(S_{\mathcal{R}(\mathcal{I})}) &= \alpha \sum_{m_{ik} \in \mathcal{M}} C_{m_{ik}} Y_{m_{ik}} + \beta \sum_{a_j \in \mathcal{A}, e \in \mathcal{E}, p_{ij} \in \mathcal{P}} C_e X_{p_{ij} a_j} \\ &= \sum_{m_{ik} \in \mathcal{M}} C_{m_{ik}} Y_{m_{ik}} + \sum_{a_j \in \mathcal{A}, p_{ij} \in \mathcal{P}} (C_{e^1_{ij}} + C_{e^2_{ij}}) X_{p_{ij} a_j} \\ &= \sum_{m_{i1} \in \mathcal{M}} f_i Y_{m_{i1}} + \sum_{a_j \in \mathcal{A}, p_{ij} \in \mathcal{P}} d_{ij} X_{p_{ij} a_j} \\ &= \sum_{i \in \mathcal{F}} f_i W_i + \sum_{j \in \mathcal{D}, i \in \mathcal{F}} d_{ij} Z_{ij} \\ &= Cost(S_{\mathcal{I}}) \end{aligned}$$

Now, we show that the solution to the anomaly localization instance, denoted by $S_{R(I)}$, that is constructed out of an optimal solution to the facility location instance, denoted by S^*_I, is optimal. Assume to the contrary that $S_{R(I)}$ is not optimal. Let $S^{'*}_{R(I)}$ be an optimal solution to the anomaly localization instance, and let $S^{'}_I$ be the facility location solution constructed out of $S^{'*}_{R(I)}$. We have $Cost(S^*_I) = Cost(S_{R(I)}) < Cost(S^{'*}_{R(I)}) = Cost(S^{'}_I)$, leading to a contradiction. Using the same arguments, we can show that the solution to the facility location instance constructed out of an optimal solution to the anomaly localization instance is optimal. ■

VIII. Heuristic solution

In this section, we provide a monitor location and path selection algorithm for localizing single link-level anomalies. The inputs of the algorithm are a network graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, a set of anomaly scenarios $\mathcal{A}$, a set of candidate monitor locations $\mathcal{M}$, the costs of deploying monitoring devices on the network nodes $C_{\mathcal{M}} = \{C_n;\ n \in \mathcal{M}\}$, and the costs of monitoring the network links $C_{\mathcal{E}} = \{C_e;\ e \in \mathcal{E}\}$. The outputs are a set of monitor locations, $\mathcal{SM}_a$, and a set of monitoring paths, $\mathcal{SP}_a$, that can distinguish between all links of $\mathcal{S}_a$ pairwise, for each $a \in \mathcal{A}$.

Similarly to the ILP, the heuristic solution aims at minimizing the infrastructure cost, the communication cost and the probe cost jointly. To this end, we use a nested greedy approach that selects monitor locations jointly with monitoring paths. Algorithm 1 describes the pseudo-code. ProbeCost$(p, C_{\mathcal{E}})$ is a function that returns the probe cost incurred by monitoring path p. This cost is computed as described in section V. m_s stores the best current candidate monitor location. $\mathcal{SM}$ stores the monitor locations selected at the previous iterations. Pcost$_{min}$ stores the current lowest probe cost, and lc_{max} stores the current largest localization capacity, *i.e.*, the number of link pairs that can be distinguished by monitors in $\mathcal{SM} \cup \{m_s\}$. CP stores paths selected by the current best solution. In the sequel, we define the criteria of monitor location selection and monitoring path selection.

A detailed description of how monitor locations and monitoring paths are selected, and how candidate localization paths are computed is provided in the following subsections.

A. Selection of monitor locations

The algorithm starts by selecting one candidate monitor location randomly. Alternatively, the candidate monitor location with the smallest monitor cost (sum of the infrastructure cost and the communication cost) can be selected. However, we advocate random selection for two reasons. The first is

[3] Recall that X_{pa} is a binary variable that indicates whether path p is part of the localization solution of the anomaly scenario a, and Y_n is a binary variable that indicates whether node n is selected as a monitor location

Algorithm 1: Monitor location and path selection algorithm for single anomaly localization

1 nbPairs = $\sum_{a\in\mathcal{A}}\sum_{k=1}^{|\mathcal{S}_a|-1} k$; Pcost$_{min}$ $\leftarrow$ INT_MAX; lc_{max} $\leftarrow 0$; $\mathcal{CP}\leftarrow\emptyset$;
2 $\mathcal{SM}\leftarrow\{$`selectRandomElement`$(\mathcal{M})\}$; Remove the selected monitor location from $\mathcal{M}$;
3 **while** $(\mathcal{M}\neq\emptyset)$ **do**
4 Reset $m_s\leftarrow Null$;
5 **foreach** $(m\in\mathcal{M})$ **do**
6 **if** $((\ lc_{max}=nbPairs$ and $\beta Pcost_{min}\ \leq\ \alpha C_m))$ **then** Jump to line 5;
7 Reset $lc\leftarrow 0$; Reset Pcost $\leftarrow 0$;
8 **for** $(a\in\mathcal{A})$ **do**
9 Clear $\mathcal{P}_a$; Clear $\mathcal{M}_a$;
10 $j\leftarrow 1$; $s_{(0)}\leftarrow 1$; $\mathcal{S}_a^{(0)1}\leftarrow\mathcal{S}_a$;
11 **while** $(s_{(j)}>0)$ **do**
12 $\mathcal{S}_a^{(j)}\leftarrow\{\mathcal{S}_a^{(j)1},...\ \mathcal{S}_a^{(j)k},\mathcal{S}_a^{(j)k+1}\ ...,\mathcal{S}_a^{(j)s_{(j)}}\}$;
13 $p_{a(j)}\leftarrow$ CandidatePathSelection$(m,\mathcal{SM},\mathcal{G},\mathcal{S}_a^{(j)},\mathcal{CP})$;
 lc += $\sum_{1\leq k\leq s_{(j)}}$ `lc`$(p_{a(j)},\mathcal{S}_a^{(j)k})$;
14 Pcost += $ProbeCost(p_{a(j)},C_\mathcal{E})$;
15 $l\leftarrow 1$;
16 **for** $(1\leq k\leq s_{(j)})$ **do**
17 **if** $(|\ p_{a(j)}\cap\mathcal{S}_a^{(j)k}\ |>1)$ **then**
 $\mathcal{S}_a^{(j+1)l}\leftarrow p_{a(j)}\cap\mathcal{S}_a^{(j)k}$; $l\leftarrow l+1$;
18 **if** $(|\ \mathcal{S}_a^{(j)k}-\{p_{a(j)}\cap\mathcal{S}_a^{(j)k}\}\ |>1)$ **then**
 $\mathcal{S}_a^{(j+1)l+1}=\mathcal{S}_a^{(j)k}-\{p_{a(j)}\cap\mathcal{S}_a^{(j)k}\}$;
 $l\leftarrow l+1$;
19 **end**
20 $s_{(j)}\leftarrow l-1$;
21 **if** $(lc_{max}=nbPairs$ and $(\alpha C_m+\beta($*PCost* $+\sum_{l=1}^{s_{(j+1)}}$ *ThMinPCost*$(\mathcal{S}_a^{(j+1)l})$ $+$ $\sum_{a'\in\mathcal{A},a'>a}$*ThMinPCost*$(\mathcal{S}_{a'}))\ \geq$ $\alpha C_{m_s}+\beta Pcost_{min}))$ **then**
22 /*Stop the exploration of the current candidate monitor location*/
23 Jump to line 5;
24 **end**
25 Add the end nodes of $p_{a(j)}$ to $\mathcal{M}_a$;
26 Add $p_{a(j)}$ to $\mathcal{P}_a$;
27 $j\leftarrow j+1$;
28 **end**
29 **end**
30 **if** $(lc>lc_{max}$ or $(lc=lc_{max}$ and $\alpha C_m+\beta Pcost<\alpha C_{m_s}+\beta Pcost_{min}))$ **then**
31 $m_s\leftarrow m$; $lc_{max}\leftarrow lc$;
32 Pcost$_{min}\leftarrow PCost$; $\mathcal{SP}_a\leftarrow\mathcal{P}_a$; $\mathcal{SM}_a\leftarrow\mathcal{M}_a$;
33 **end**
34
35 **end**
36 **if** $(m_s=Null)$ **then return** $(\{\mathcal{SP}_a,\mathcal{SM}_a\};\ \ \forall a\in\mathcal{A})$
37 Update $\mathcal{CP}\leftarrow\bigcup_{a\in\mathcal{A}}\mathcal{SP}_a$;
38 Add m_s to $\mathcal{SM}$; Remove m_s from $\mathcal{M}$;
39 **end**
40 **return** $(\{\mathcal{SP}_a,\mathcal{SM}_a;\ \ \forall a\in\mathcal{A}\})$;

that the monitor location with the smallest monitor cost does not necessarily incur the smallest probe cost. The second is that selecting the starting point randomly enlarges the space of explored solutions over multiple runs of the algorithm. Monitor locations are, then, added to the solution greedily until all link pairs of all the anomaly scenarios are distinguished. At each greedy iteration (lines 5-36), all the remaining candidate monitor locations are explored. Let us fix a candidate monitor location m. A set of monitoring paths whose end nodes are in $\mathcal{SM}\cup\{m\}$ is selected greedily (lines 7-35). The path selection procedure is described in details in section VIII-B. The candidate monitor location whose associated monitoring paths can distinguish between the largest number of link pairs over all the anomaly scenarios is selected (line 31). In case of a tie, a monitor location that incurs the smallest localization cost ($\alpha\times$ monitor cost + $\beta\times$ probe cost, where the probe cost is the summation of the probe costs of the associated monitoring paths) is selected.

When a solution that distinguishes between all the link pairs of all the anomaly scenarios is found, the algorithm continues the exploration of the remaining candidate monitor locations, if any, towards reducing the probe cost. However, a filter is applied on these locations before exploring them (line 6). Only candidate locations whose monitor cost is smaller than the probe cost of the current best solution are explored. Clearly, the localization cost of any solution that selects a monitor location not satisfying this filter would be larger than the localization cost of the current best solution. The algorithm ends when the set of candidate monitor locations gets empty, *i.e.*, all candidate monitor locations have been selected, or when remaining candidate monitor locations can neither improve the localization capacity nor the probe cost of the current best solution.

B. Selection of localization paths

Given a candidate monitor location m and a set of already selected monitor locations $\mathcal{SM}$, the procedure of selecting an associated set of monitoring paths, (lines 7-35), is as follows. Let us fix an anomaly scenario a. A set of monitoring paths that maximizes the number of distinguished pair of links of $\mathcal{S}_a$ while minimizing the probe cost is selected greedily as follows. First, one path that can distinguish between the largest number of link pairs of $\mathcal{S}_a$ is selected. We refer to the number of pair of links of a set of suspect links $\mathcal{S}_a$ that can be distinguished by a path p as the localization capacity of p with respect to $\mathcal{S}_a$, denoted by $lc(p,\mathcal{S}_a)$. It can be easily shown that:

$$lc(p,\mathcal{S}_a)=|\ p\cap\mathcal{S}_a\ |\ (|\ \mathcal{S}_a\ |-|\ p\cap\mathcal{S}_a\ |) \tag{6}$$

In case of a tie, a path that minimizes the probe cost is selected. The algorithm used for computing the candidate monitoring path is described in section VIII-C. Let $p_{a(1)}$ be the selected path. Two subsets of suspect links are generated: $\mathcal{S}_a^{(1)1}=\mathcal{S}_a\cap p_{a(1)}$ and $\mathcal{S}_a^{(1)2}=\mathcal{S}_a-\{\mathcal{S}_a\cap p_{a(1)}\}$. According to Theorem 1, $p_{a(1)}$ distinguishes between every pair of links (e_1,e_2) such that $e_1\in\mathcal{S}_a^{(1)1}$ and $e_2\in\mathcal{S}_a^{(1)2}$. At the next step, we need to distinguish between the links of $\mathcal{S}_a^{(1)1}$ pairwise and between the links of $\mathcal{S}_a^{(1)2}$ pairwise. Hence, a path that maximizes $lc(p,\mathcal{S}_a^{(1)1})+lc(p,\mathcal{S}_a^{(1)1})$ is selected. Ties are broken by selecting a path that minimizes the probe cost.

Let $p_{a(j)}$ be the monitoring path selected at step (j). Let $s_{(j-1)}$ be the number of non-unitary subsets of suspect links generated at step $(j-1)$. $p_{a(j)}$ is selected such that $\sum_{1\leq k\leq s_{(j-1)}} lc(p,\ \mathcal{S}_a^{(j-1)k})$ is maximized. In case of a tie, a path that minimizes the probe cost is selected. For each $\mathcal{S}_a^{(j-1)k}$, $1 \leq k \leq s_{(j-1)}$, two subsets of suspect links are generated: $\mathcal{S}_a^{(j-1)k} \cap p_{a(j)}$ and $\mathcal{S}_a^{(j-1)k} - \{\mathcal{S}_a^{(j-1)k} \cap p_{a(j)}\}$. Each link of the former subset is distinguished from each link of the latter subset. Only non-unitary subsets, whose links need to be distinguished from each other, are considered at the next step. This greedy process is re-iterated until all the generated subsets of suspect links are unitary or until no candidate localization path can distinguish between the pair of links of non-unitary subsets. For each selected path $p_{a(j)}$, the localization capacity of m is incremented by $lc(p_{a(j)}, \mathcal{S}_a^{(j)})$ (line 13), and its probe cost is incremented by $probeCost(p_{a(j)}, C_\mathcal{E})$ (line 14). The above procedure is applied on the all the anomaly scenarios in $\mathcal{A}$. Then, the localization capacity and the probe cost of m are evaluated (line 31). If the localization capacity of m is greater than lc_{max}, or if the localization capacity of m equals lc_{max} and its probe cost is less than Pcost_{min}; then lc_{max} is set equal to the localization capacity of m, Pcost_{min} is set equal to the probe cost of m and m_s is set equal to m.

Furthermore, using the argument of the following theorem, we can compute a lower bound of the probe cost of the explored monitor location at any step in the path selection procedure.

Theorem 5: The theoretical minimal probe cost relative to a set of suspect links $\mathcal{S}$ denoted by $ThMinPcost(\mathcal{S})$ reads as follows:

$$ThMinPcost(S) = \sum_{e\in\mathcal{S}} C_e - \max_{e\in\mathcal{S}} C_e \tag{7}$$

Proof: Refer to Appendix C. ∎

The lower bound of the probe cost of a candidate monitor location after path $p_{a(j)}$ is added to the set of its associated monitoring paths reads as follows:

$$Pcost\ +\ \sum_{k=1}^{s_{(j)}} ThMinPcost(\mathcal{S}_a^{(j)k})\ +\ \sum_{a'\in\mathcal{A},a'>a} ThMinPcost(\mathcal{S}_a^{'}) \tag{8}$$

where $Pcost$ is the summation of the probe costs of the already selected paths.

When the algorithm finds a solution that can distinguish between all link pairs of all the anomaly scenarios, it continues exploring the remaining candidate monitor locations that satisfy the monitor cost filter (line 6) towards reducing the probe cost. Using (8), we propose an optimization of the exploration process of these candidate monitor locations. The idea is to update the lower bound of the probe cost of the explored monitor location whenever a monitoring path is selected, and to infer a lower bound of the localization cost (line 21). The exploration of the considered candidate monitor location is abandoned if, at any step of the path selection procedure, the calculated lower bound of the localization cost dominates the localization cost of the current best solution.

Procedure 2: candidatePathSelection($m, \mathcal{SM}, \mathcal{G}, \mathcal{S}_a^{(j)}, \mathcal{CP}$)

1 $p_c \leftarrow$ newPath();
2 $li_{min} \leftarrow \sum_{\mathcal{S}_a^{(j)k}\in\mathcal{S}_a^{(j)}} \mid \mathcal{S}_a^{(j)k} \mid /2-1$; $\text{Pcost}_{min} \leftarrow \sum_{e\in\mathcal{E}} C_e$;
3 **foreach** $q \in \mathcal{CP}$ **do**
4 $\quad li = \sum_{\mathcal{S}_a^{(j)k}\in\mathcal{S}_a^{(j)}} \mid \mathcal{S}_a^{(j)k} \mid /2- \mid \mathcal{S}_a^{(j)k} \cap q \mid$;
5 $\quad$ **if** ($li < li_{min}$ or ($li = li_{min}$ and $probeCost(p_c, C_\mathcal{E}) < Pc_{min}$)) **then**
6 $\quad\quad li_{min} = li$; $\text{Pc}_{min} = probeCost(p_c, C_\mathcal{E})$; $p_s = q$;
7 $\quad$ **end**
8 **end**
9 add-node-to-path(m, p_c);
10 **depthFirst** (m, p_c){
11 $\quad$ **foreach** ($n \in children(m, \mathcal{G})$ and $(m, n) \notin p_c$) **do**
12 $\quad\quad$ add-node-to-path(n, p_c);
13 $\quad\quad li(p_c, \mathcal{S}_a^{(j)}) = \sum_{\mathcal{S}_a^{(j)k}\in\mathcal{S}_a^{(j)}} \mid\mid \mathcal{S}_a^{(j)k} \mid /2- \mid \mathcal{S}_a^{(j)k} \cap p_c \mid\mid$;
14 $\quad\quad$ **if** ($n \in \mathcal{SM}$) **then**
15 $\quad\quad\quad$ **if** ($li(p_c, \mathcal{S}_a^{(j)}) < li_{min}$ or ($li(p_c, \mathcal{S}_a^{(j)}) = li_{min}$ and $probeCost(p_c, C_\mathcal{E}) <= Pc_{min}$)) **then**
16 $\quad\quad\quad\quad p_s \leftarrow p_c$; $li_{min} \leftarrow li(p_c, \mathcal{S}_a^{(j)})$;
$\quad\quad\quad\quad \text{Pc}_{min} = probeCost(p_c, C_\mathcal{E})$;
17 $\quad\quad\quad\quad$ **if** ($li_{min} = 0$ *and* $Pc_{min} = 0$) **then**
18 $\quad\quad\quad\quad\quad$ /*Stop the algorithm*/ Jump to line 31;
19 $\quad\quad\quad\quad$ **end**
20 $\quad\quad\quad$ **end**
21 $\quad\quad$ **else**
22 $\quad\quad\quad$ **if**
$\quad\quad\quad$ (($li_{min} = 0$ and ($probeCost(p_c, C_\mathcal{E}) + li(p_c, \mathcal{S}_a^{(j)}) - li_{min} >= Pc_{min}$ or $\exists\ \mathcal{S}_a^{(j)k} \in \mathcal{S}_a^{(j)}$ *such that*
23 $\quad\quad\quad$ $\mid \mathcal{S}_a^{(j)k} \cap p_c \mid > \mid \mathcal{S}_a^{(j)k} \mid /2)$) or ($li(p_c, \mathcal{S}_a^{(j)}) > li_{min}$ and $\forall\ \mathcal{S}_a^{(j)k} \in \mathcal{S}_a^{(j)}\ \mid \mathcal{S}_a^{(j)k} \cap p_c \mid > \mid \mathcal{S}_a^{(j)k} \mid /2$)) **then**
24 $\quad\quad\quad\quad$ do not explore the descendants of n;
25 $\quad\quad\quad$ **else**
26 $\quad\quad\quad\quad$ Recursively call **depthFirst** (n, p_c);
27 $\quad\quad\quad$ **end**
28 $\quad\quad$ **end**
29 $\quad$ **end**
30 }
31 return p_s;

C. Candidate path selection algorithm

This section describes the procedure *candidatePathSelection* called by Algorithm 1 at line 13. The inputs into this procedure are the network graph, the currently explored monitor location m, the subsets of suspect links generated at the current step of the path selection procedure $\mathcal{S}_a^{(j)} = \{\mathcal{S}_a^{(j)1}, ...\ \mathcal{S}_a^{(j)k}, \mathcal{S}_a^{(j)k+1}\ ..., \mathcal{S}_a^{(j)s_{(j)}}\}$, the set of the already selected monitor locations $\mathcal{SM}$, and the set of monitoring paths selected by the current best solution $\mathcal{CP}$. The output is one monitoring path, whose end nodes are in $\mathcal{SM} \cup \{m\}$, that maximizes the localization capacity while minimizing the probe cost.

The main difficulty of this procedure is the computation of the set of candidate paths. Generally, the smaller the set of candidate paths is, the worst the quality of the heuristic is. This is because good paths might be missed when reducing the number of candidate paths. However, this reduction is imper-

ative to ensure the scalability of the heuristic. The procedure *candidatePathSelection* implements an algorithm for candidate localization path computation. The algorithm considers all the network paths whose end nodes belong to $\{m\} \times \mathcal{SM}$ as candidate to be monitored. However, computing this set of paths is computationally expensive, because it requires exploring all the network graph. Moreover, since Algorithm 1 explores all remaining candidate monitor locations at each iteration, the graph would be explored multiple times; which makes the heuristic non-scalable and non-practical for dense networks. An alternative solution is to compute and store all paths traveling between all candidate monitor locations offline, thereby reducing the number of times the network graph is explored to one. Clearly, this solution is impractical due to memory issues. We conclude, based on the above discussion, that our candidate path computation algorithm must minimize the number of paths that are to be explored, while guaranteeing that good candidate paths are not missed. To this end, we make use of the argument of the following theorem:

Theorem 6:
let $lc(\mathcal{S}_a^{(j)}, p) = \sum_{\mathcal{S}_a^{(j)k} \in \mathcal{S}_a^{(j)}} lc(\mathcal{S}_a^{(j)k}, p)$ *be the localization capacity of* p *with respect to* $\mathcal{S}^{(j)}$. *We have,*

$$\max_{p \in \mathcal{P}} \; lc(\mathcal{S}_a^{(j)}, p) = \min_{p \in \mathcal{P}} \sum_{\mathcal{S}_a^{(j)k} \in \mathcal{S}_a^{(j)}} || \; \mathcal{S}_a^{(j)k} \mid /2 - \mid \mathcal{S}_a^{(j)k} \cap p \; || \quad (9)$$

Proof: Refer to Appendix D. ∎

We refer to $\sum_{\mathcal{S}_a^{(j)k} \in \mathcal{S}_a^{(j)}} || \; \mathcal{S}_a^{(j)k} \mid /2 - \mid \mathcal{S}_a^{(j)k} \cap p \; ||$ as the localization indicator of path p with respect to $\mathcal{S}_a^{(j)}$, and we denote it by $li(p, \mathcal{S}_a^{(j)})$. According to Theorem 6, the smaller $li(p, \mathcal{S}_a^{(j)})$ is, the higher the localization capacity of p with respect to $\mathcal{S}_a^{(j)}$ is. The localization indicator is used along with the probe cost to avoid exploring all the network graph, while guaranteeing that good candidate paths are not missed. Procedure 2 provides an overview of the pseudo-code. p_s stores the current best candidate path, li_{min} stores the localization indicator of p_s, and Pc_{min} stores its probe cost. p_s, li_{min} and Pc_{min} are initialized to $Null$, $\sum_{\mathcal{S}_a^{(j)k} \in \mathcal{S}_a^{(j)}} \mid \mathcal{S}_a^{(j)k} \mid /2 - 1$ and $\sum_{e \in \mathcal{E}} C_e$, respectively. Note that the least upper bound of the localization indicator is $\sum_{\mathcal{S}_a^{(j)k} \in \mathcal{S}_a^{(j)}} \mid \mathcal{S}_a^{(j)k} \mid /2$, which corresponds to a path that does not provide any localization information (*i.e.*, $\forall \mathcal{S}_a^{(j)k} \in \mathcal{S}_a^{(j)}, \mathcal{S}_a^{(j)k} \cap p = \emptyset$ or $\exists \mathcal{S}_a^{(j)k} \in \mathcal{S}_a^{(j)}$ such that $p = \mathcal{S}_a^{(j)k}$)[4]; whereas the least upper bound of the probe cost is $\sum_{e \in \mathcal{E}} C_e$, which corresponds to a path that crosses all the network nodes and does not provide any localization information. However, if $\mathcal{CP}$ is not empty, then p_s is set equal to the best path in $\mathcal{CP}$, *i.e.*, the path that maximizes the localization capacity (in case of a tie, a path that minimizes the probe cost); and li_{min} and Pc_{min} are initialized to the localization capacity and the probe cost of that path, respectively. The rational behind considering paths in $\mathcal{CP}$ is to avoid re-exploring all candidate paths traveling between the already selected monitors.

[4]By construction, $\bigcap_k \mathcal{S}_a^{(j)k} = \emptyset$

The network graph is, then, explored in depth-first order starting from the candidate monitor location m. It is worth noting that we believe that a breadth-first search can find candidate paths faster. However, the depth-first search approach requires much less memory.

We now introduce the optimizations made to avoid exploring all the network graph, which speeds up the search and ensures the scalability of the algorithm. Let n be the currently explored node and p_c the current path to that node. p_s , li_{min} and Pc_{min} are set equal to p_c, $li(p_c, \mathcal{S}_a^{(j)})$, and $probeCost(p_c, C_{\mathcal{E}})$, respectively, if the following condition is true:

$$n \in \mathcal{SM} \text{ and} (li(p_c, \mathcal{S}_a^{(j)}) < li_{min} \text{ or } (li(p_c, \mathcal{S}_a^{(j)}) = li_{min} \text{ and } probeCost(p_c, C_{\mathcal{E}}) < \text{Pc}_{min})) \quad (10)$$

The above condition implies that the path selection criterion is the minimization of the localization indicator, which is equivalent to the maximization to the localization capacity, and that ties are broken by minimizing the probe cost. Moreover, it ensures that the end nodes of the selected path are in $\mathcal{SM} \cup \{m\}$.

Now, the most important feature of the algorithm is that it is able, using Theorem (6), to decide whether all paths having a given prefix are not good. A good path is a path that dominates the current best path, *i.e.*, a path that satisfies Condition (10). In fact, all paths having as prefix the current path p_c are undoubtedly inefficient if one of the following conditions is true:

$$li_{min} = 0 \text{ and } \exists \; \mathcal{S}_a^{(j)k} \in \mathcal{S}_a^{(j)} \text{ such that } \mid \mathcal{S}_a^{(j)k} \cap p_c \mid > \mid \mathcal{S}_a^{(j)k} \mid /2 \quad (11)$$

$$li_{min} = 0 \text{ and } \forall \mathcal{S}_a^{(j)k} \in \mathcal{S}_a^{(j)} \mid \mathcal{S}_a^{(j)k} \cap p_c \mid \leq \mid \mathcal{S}_a^{(j)k} \mid /2 \text{ and } probeCost(p_c, C_{\mathcal{E}}) + \min_{e \in \mathcal{E}} C_e li(p_c, \mathcal{S}_a^{(j)}) \geq \text{Pc}_{min} \quad (12)$$

$$li(p_c, \mathcal{S}_a^{(j)}) > li_{min} \text{ and } \forall \mathcal{S}_a^{(j)k} \in \mathcal{S}_a^{(j)} \mid \mathcal{S}_a^{(j)k} \cap p_c \mid \geq \mid \mathcal{S}_a^{(j)k} \mid /2 \quad (13)$$

Whenever a node that is not in $\mathcal{SM}$ is explored, the current path to that node is examined. If it satisfies one of the above conditions, then the descendant nodes of the current node will not be explored, *i.e.*, all paths having as prefix the current path will be discarded without exploring their suffixes. This achieves great savings in terms of the number of explored paths and in terms of time. The accuracy of conditions (11), (12) and (13) is demonstrated in Appendix E.

IX. Performance Evaluation

Extensive simulations are conducted on network topologies built using the BRITE generator [18] (Waxman model [19]: $\alpha = \beta = 0.4$, random node placement [5]). We use Cplex11.2 [20] to solve ILPs and we implement our algorithms using C_{++}. All the numerical results presented in this section are the mean over 30 simulations on random simulations. Our experiments indicate that the results are almost the same for larger number of simulations. Table III depicts a summary of the topologies considered. Our localization scheme takes as input

[5]These parameters are not to be confused with the monitor cost weight (α) and the probe cost weight (β) introduced in Section VI. Their values equal the values used by Waxman to generate network topologies [19].

any detection solution that covers all links of the network. For small topologies, *i.e.*, TOP(8, 18), optimal detection solutions are computed using the detection scheme proposed in [13]; whereas the anomaly detection heuristic proposed in [14] is used to compute detection solutions for larger topologies. Note that the anomaly detection problem is $\mathcal{NP}$-Hard, therefore, optimal detection solutions could not be computed for large topologies.

TABLE III: Summary of the topologies considered in the evaluation

Topology	Nb. of nodes	Nb. of links
TOP(8, 18)	8	18
TOP(10, 31)	10	31
TOP(12, 41)	12	41
TOP(15, 59)	15	59
TOP(20, 80)	20	80

The evaluations are performed on a PC equipped with a 2,992.47 MHz Intel(R) Core(TM)2 Duo processor and 3.9 GB of RAM. We assume that every nodes of the network is candidate to support a monitoring device and all paths of the networks are candidate to be monitored. We set $C_n = C_e = 1, \forall n \in \mathcal{N}$ and $\forall e \in \mathcal{E}$.

A. Comparing our Anomaly Localization Scheme with Existing Schemes

We compare our anomaly localization scheme with an hybrid anomaly localization scheme that combines the strengths of the schemes proposed in [1] and [2]. As proposed in [2], a set of paths that distinguishes only between the suspect links is monitored during the localization phase. However, to guarantee that all potential anomalies can be localized uniquely, a set of monitors that can distinguish between all pairs of the network links is deployed [1]. Such a scheme can be formulated as two ILPs. The first ILP computes a minimal subset of monitor locations that enables the localization of all potential anomalies. This ILP is run only once offline. It reads as follows:

Minimize $\sum_{n\in\mathcal{M}} Y_n$
Subject to:
$\sum_{p\in\mathcal{P}}(\delta_{pe_1} + \delta_{pe_2} - 2\delta_{pe_1}\delta_{pe_2})Z_p > 0; \forall e_1, e_2 \in \mathcal{E}; \forall p \in \mathcal{P}$
$\delta_{pn} Y_n \geq Z_p; \quad \forall p \in \mathcal{P}, \forall n \in \mathcal{N}$

The second ILP is run whenever an anomaly is detected. The input is the set of monitor locations selected by the first ILP, $\mathcal{M}^{'}$, and a set of suspect links $\mathcal{S}$. The output is a minimal set of monitoring paths that can distinguish between the suspect links pairwise. This ILP reads as follows:

Minimize $\sum_{p\in P} Z_p$
Subject to:
$\sum_{p\in\mathcal{P}}(\delta_{pe_1} + \delta_{pe_2} - 2\delta_{pe_1}\delta_{pe_2})Z_p > 0; \forall e_1, e_2 \in \mathcal{S}; \forall p \in \mathcal{P}$
$Z_p \leq \delta_{pn} Y_n; \quad \forall p \in \mathcal{P}, \forall n \in \mathcal{M}^{'}$

We refer to this hybrid anomaly localization scheme as HLS.

Only small topologies for which the ILPs can deliver solutions in tractable time are considered. We set the weight associated to the probe cost $\beta = 1$, and we vary the weight associated to the monitor cost, $\alpha \in [1, 2, 4]$ and $\alpha \geq 6$.

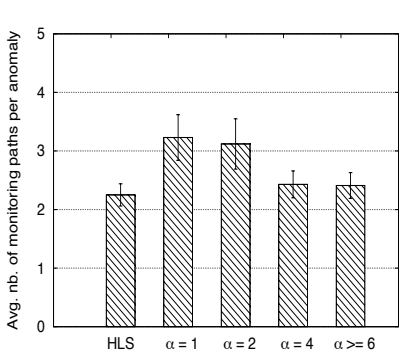


Fig. 3: Average number of monitoring paths per anomaly for TOP(8, 18). The first histogram to the left presents results for solutions computed using the hybrid localization scheme (HLS), and the other histograms present results for the solutions computed using our anomaly localization ILP with different values of α ($\beta = 1$).

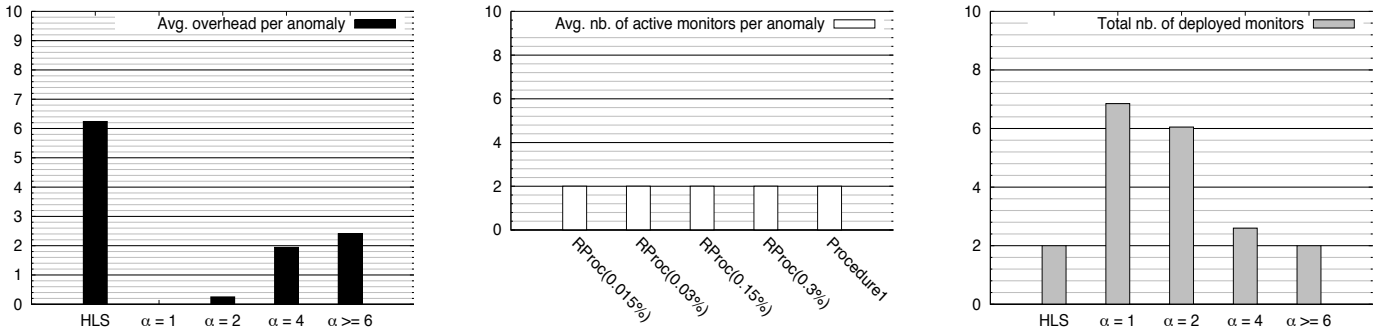
We define three metrics for the comparison. The first metric is the time of computing the localization solution, *i.e.*, monitors that are to be activated and paths that are to be monitored when an anomaly is detected. This metric reflects the speed of the localization scheme. The better is to avoid online computations, *i.e.*, computations done upon detecting an anomaly, in order to shorten the localization delay.

TABLE IV: Average ILP computation time for TOP(8, 18)

	Hybrid scheme	Our scheme
Offline Computation Time	$64.16\ s$	$6.67\ s$
Online Computation Time	$25.7\ 10^{-3}\ s$	$0\ s$

Table IV depicts the online computation time and the offline computation time for the hybrid localization scheme and for our localization scheme. Intuitively, as shown in the table, the online computation time is zero for our localization scheme. This is because we compute full localization solutions for all potential anomalies offline. In contradiction, the hybrid scheme leaves the selection of monitoring paths upon detecting an anomaly, thereby achieving a non-negligible online computation time. This time can be relatively high for large topologies where the number of candidate monitoring paths is large. For the offline computation time, the table shows that our scheme is about 10 times faster than the hybrid scheme, although, it computes full localization solutions for all potential anomalies. We explain this result by the fact that, unlike the hybrid scheme, our scheme does not distinguish between every pair of the network links.

The second metric is the localization cost. Figure 4 plots the total number of deployed monitors (Figure 4c), the average number of monitors activated per anomaly (Figure 4b), and the average overhead (4a), *i.e.*, the number of links monitored that provide no localization information, per anomaly for the hybrid localization scheme and for our localization scheme with different values of α. Three conclusions can be drawn from the numerical results. The first is that there is an interplay between the monitor location cost and the probe cost. The different results for the different values of α illustrate this conclusion. Indeed, the larger the value of α is, the fewer the number of monitors is and the larger the localization



(a) Average overhead per anomaly (b) Average nb. of monitors activated per anomaly (c) Total nb. of deployed monitors

Fig. 4: Localization costs for TOP(8, 18)

overhead is. For instance, for $\alpha = 1$, we have localization solutions with zero overhead and 7 monitors, *i.e.*, 7 of the 8 nodes of the network hold monitoring devices. The second is that the existing localization scheme that deploys monitors offline and selects monitoring paths online does not take into consideration this interplay, and therefore, delivers sub-optimal localization solutions. Using the same number of monitors, for $\alpha \geq 6$, our localization scheme can localize any potential anomaly with about 65% less overhead than the existing localization scheme.

The third metric is the number of monitoring paths. Recall that this is the path selection criterion for the existing localization scheme. We do not consider this criterion in our localization scheme for two reasons. The first is that, upon detecting an anomaly, the set of paths that distinguish between the suspect links are monitored simultaneously. Therefore, the minimization of the number of monitoring paths does not reduce the localization delay. The second reason is that this metric is tightly correlated to the number of monitors and the localization overhead. Indeed, if we relax the constraint on the localization overhead, this would allow long monitoring paths that cross a large number of links. Therefore, the number of monitoring paths that can distinguish between the suspect links would decrease. Similarly, if we relax the constraint on the number of monitors, we would deploy more monitors in the network, thus, the monitoring paths would get shorter. Therefore, the number of monitoring paths that can distinguish between the suspect links would increase. Figure 3 validates these claims. Hereby, we can observe that the larger α is, the more monitoring paths we have. Not surprisingly, for $\alpha \geq 6$, our localization scheme monitors only 8% more paths than the hybrid localization scheme, while deploying the same number of monitors and incurring 65% less overhead.

B. Evaluating the Scalability and Quality of the Heuristic

In this section, we evaluate the performance of our anomaly localization heuristic. We set $\alpha >> \beta$. For each network topology, we run the heuristic n times, where n is the number of the network nodes. The first monitor location that is selected randomly must be different for each run. Then, we consider the solution with the smallest localization cost. For TOP(8, 18), we compare the results obtained using the heuristic with the results obtained using our anomaly localization ILP ($\alpha \geq 6$), and the results obtained using the hybrid localization scheme. Furthermore, we evaluate the evolution of resource consumption and computation time with respect to the network size to evaluate the performance of the heuristic on larger topologies.

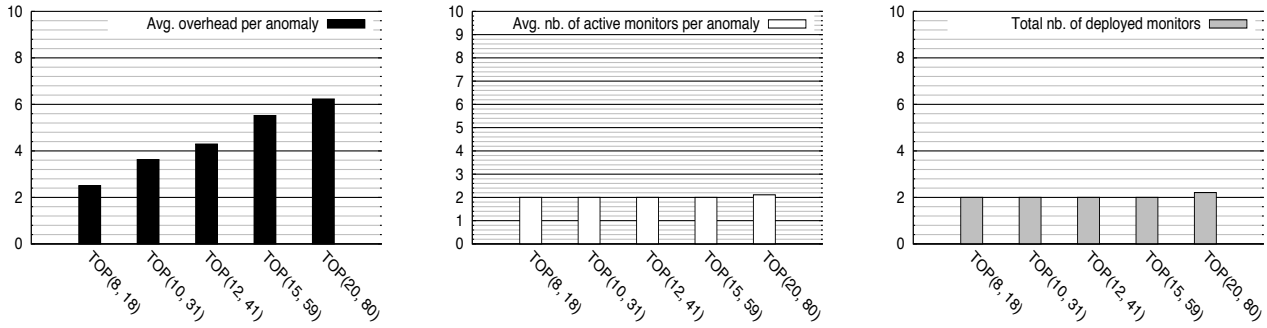
Table V depicts the heuristic computation time (this is the time of the n runs of the heuristic) and the average percentage of the network paths explored in one execution of Procedure 2 for all the topologies considered. For TOP(8, 18) the heuristic computation time is about 29.10^3 times faster than our ILP, and about 27.10^4 times faster than the hybrid localization scheme. Recall that all computations are done offline. For TOP(10, 31), TOP(12, 41) and TOP(15, 59) the heuristic computation time is in the order of few seconds (< 25 s), while it was infeasible to obtain the ILP results for these topologies in tractable time. For TOP(20, 80), whose number of paths is in the order of hundreds of billions, it was impossible to run the ILPs due to memory insufficiency. However, the heuristic succeeded to compute solutions in less than one hour for these topologies. This confirms the efficiency of our candidate path computation algorithm that minimizes the number of the networks paths that are to be explored. For instance, we found that only 0.007% of the network paths are explored in one execution of Procedure 2 for TOP(20, 80).

TABLE V: Heuristic computation time (all computations are done offline) and percentage of paths explored in one execution of Procedure 2

Topology	Heuristic computation time	% of paths explored in one execution of Procedure 2
TOP(8, 18)	0.00023 s	1.22%
TOP(10, 31)	0.08 s	0.21%
TOP(12, 41)	0.78 s	0.07%
TOP(15, 59)	24.11 s	0.02%
TOP(20, 80)	3525.52 s	0.007%

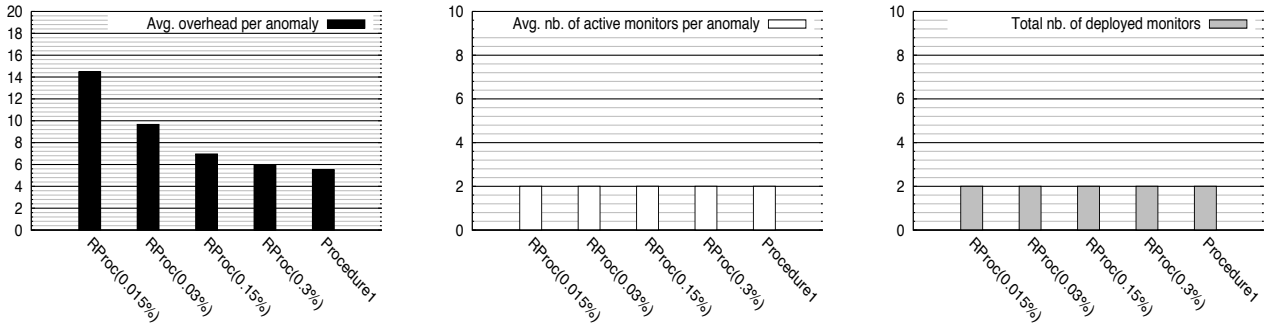
We now investigate the quality of the solutions delivered by the heuristic. Figure 5 plots the total number of monitors deployed (5c), the average number of monitors activated per anomaly (5b), and the average overhead per anomaly for the topologies considered in the evaluation5a.

First, we notice that two monitors are sufficient to localize all potential anomalies for all topologies, except TOP(20, 80)



(a) Average overhead per anomaly (b) Average nb. of monitors activated per anomaly (c) Total nb. of deployed monitors

Fig. 5: Localization cost of the heuristic solutions, $\alpha >> \beta$



(a) Average overhead per anomaly (b) Avgerage nb. of monitors activated per anomaly (c) Total nb. of deployed monitors

Fig. 6: Impact of the number and the quality of candidate monitoring paths on the quality of the localization solution. RProc means random procedure (numerical results for TOP(15, 59))

for which the average number of monitors deployed and the average number of monitors activated per anomaly are slightly larger than two. This is expected, since we set $\alpha >> \beta$, which means that the heuristic minimizes in priority the number of monitors that are to be deployed. A comparison of Figure 5 with Figure 4 shows that, for TOP(8, 18), the solutions computed using our ILP ($\alpha \geq 6$) is very close to the solutions computed using the heuristic: the heuristic solution gnenerates about 9% more overhead, however, the two solutions deploy the same number of monitors and activate, in average, the same number of monitors when an anomaly occurs. This confirms that the candidate path computation algorithm that avoids exploring all paths of the network does not miss good paths. Moreover, the overhead of the heuristic solutions for TOP(10, 31), TOP(12, 41) and TOP(12, 59) is smaller than the overhead of the hybrid localization scheme solutions for TOP(8, 18). It is worth to recall that the hybrid localization solutions for TOP(8, 18) are exact solutions. This confirms that i) the heuristic succeeds to minimize the localization costs, *i.e.*, the monitor cost and the probe cost, jointly; ii) the heuristic outperforms the hybrid localization scheme, since the former can localize anomalies in large topologies using less resources that those used by the latter to localize anomalies in smaller topologies.

We finally evaluate the impact of the number and the quality of candidate monitoring paths on the quality of the localization solution. To this end, we compare the localization solutions obtained using the proposed heuristic, *i.e.*, Algorithm 1 and Procedure 2, to the localization solutions obtained using Algorithm 1 and a procedure that computes candidate paths randomly (instead of Procedure 2). In the latter case, we variate the number of paths explored per one execution of the random candidate path computation procedure (0.015%, 0.03%, 0.15%, 0.3%). We report the results for TOP(15, 59) when $\alpha >> \beta$ in Figure 6 (The results are essentially the same for the other topologies). Not surprisingly, Figure 6 shows that, when candidate paths are explored randomly, the larger the number of paths explored is the smaller the localization overhead is. Furthermore, it shows that the proposed heuristic achieves smaller overhead than the random approach, thought it explores more than 15 times less paths as shown in Table V. This validates our claim on the correlation between the number and quality of monitoring paths and the quality of the localization solution.

X. Discussion

The anomaly localization solution must be updated whenever the detection solution changes. However, the detection solution changes in rare cases where a persistent anomaly makes a network link unavailable for a long period of time, or where the network topology is modified voluntary (*e.g.*, add and/or removal of network equipments).

Usually, in the first case, the detection solution is updated partially. Only the detection paths that are affected by the anomaly are re-computed. The anomaly scenarios are updated accordingly, and the localization solution is re-computed,

partially, for the affected anomaly scenarios. The evaluation results show that, for instance, the average computation time of the localization solution for one anomaly scenario using the heuristic is in the order of 5 minutes for TOP(20, 80). Knowing that anomalies are rare events, we assert that it is rather unlikely that anomalies occur before the localization solution is updated. However, in case an anomaly occurs before the localization solution is updated, the localization process could be executed for the current solution, though, not all anomalies could be localized accurately. The best solution for such situation is to provide backup detection and localization solutions. However, this issue is out of the scope of this paper.

Furthermore, voluntary network changes are usually planned in advance. Thus, detection and localization updates could be computed offline before voluntary network changes are made.

XI. Conclusion

In this paper, we addressed the problem of localizing single link-level anomalies. Two findings were presented and demonstrated: 1) Not all pairs of the network links need to be distinguishable for localizing all potential link-level anomalies, 2) All potential anomaly scenarios can be derived offline from any detection solution that covers all the network links. These findings were exploited to develop an anomaly localization scheme that computes full localization solutions offline. In order to achieve a good trade-off between the number and locations of monitoring devices and the quality of monitoring paths, monitor locations and monitoring paths are selected jointly. A novel anomaly localization cost model is proposed, and the localization scheme is formulated as an ILP. However, it is demonstrated that the problem is $\mathcal{NP}$-hard. Therefore, an efficient heuristic is proposed. The proposed scheme is compared with an hybrid anomaly localization scheme that combines the strengths of two existing schemes through extensive simulations. Results demonstrate the superiority of the proposed anomaly localization scheme, and the efficiency of the heuristic solution. Our ongoing work is on extending the proposed scheme to localize multiple link-level anomalies.

Appendix A

This section presents the proofs of corollaries 2, 3, 4, 5 and 6.

Corollary 2. $e_1 \in \mathcal{S}(e_2) \Leftrightarrow \mathcal{S}(e_1) = \mathcal{S}(e_2)$, $\forall e_1, e_2 \in \mathcal{E}$

Proof: $e_1 \in \mathcal{S}(e_2) \Leftrightarrow$ (according to Theorem 1) there does not exist any path that crosses either e_1 or e_2, but not both $\Leftrightarrow$ for each $p \in \mathcal{P}$, p crosses both e_2 and e_1, or p neither crosses e_1 nor $e_2 \Leftrightarrow D_{e_1+} = D_{e_2+}$ and $D_{e_1-} = D_{e_2-} \Leftrightarrow$ (according to Theorem 2) $\mathcal{S}(e_1) = \mathcal{S}(e_2)$ ■

Corollary 3. $\mathcal{S}(e_1) \neq \mathcal{S}(e_2) \Leftrightarrow \mathcal{S}(e_1) \cap \mathcal{S}(e_2) = \emptyset$

Proof: We prove the direct implication by contradiction. Assume to the contrary that $\mathcal{S}(e_1) \neq \mathcal{S}(e_2)$ and $\mathcal{S}(e_1) \cap \mathcal{S}(e_2) \neq \emptyset$. Let $e_3 \in \mathcal{S}(e_1) \cap \mathcal{S}(e_2)$. According Corollary 2, $\mathcal{S}(e_3) = \mathcal{S}(e_1)$ and $\mathcal{S}(e_3) = \mathcal{S}(e_2)$. thus, $S(e_1) = \mathcal{S}(e_2)$, leading to a contradiction. The indirect implication is trivially true. ■

Corollary 4. $\cup_{e\in\mathcal{E}}\mathcal{S}(e) = \cup_{\mathcal{S}(i)\in dS}\mathcal{S}(i) = \mathcal{E}$

Proof: According to Theorem 2, $e \in \mathcal{S}(e), \forall e \in \mathcal{E}$. Thus, $\cup_{e\in\mathcal{E}}\mathcal{S}(e) = \mathcal{E}$. Obviously, $\cup_{e\in\mathcal{E}}\mathcal{S}(e) = \cup_{S(i)\in dS}S(i)$. ■

Corollary 5. $\sum_{\mathcal{S}(i)\in dS} \mid \mathcal{S}(i) \mid = \mid \mathcal{E} \mid$

Proof: According to Corollary 4, $\mid \cup_{\mathcal{S}(i)\in dS}\mathcal{S}(i) \mid = \mid \mathcal{E} \mid$, and according to Corollary 2, $\cap_{\mathcal{S}(i)\in dS}\mathcal{S}(i) = \emptyset$. Thus, $\sum_{\mathcal{S}(i)\in dS} \mid \mathcal{S}(i) \mid = \mid \mathcal{E} \mid$. ■

Corollary 6. $dPairs = AllPairs - \sum_{\mathcal{S}(i),\mathcal{S}(j)\in dS:i<j} \mid \mathcal{S}(i) \mid\mid \mathcal{S}(j) \mid$

Proof: According to Corollary 1, only links that belong to same set of suspect links need to be distinguishable pairwise. Therefore, the set of link pairs that are to be distinguished can be expressed as $\{\{(e_i, e_j); e_i, e_j \in \mathcal{E}\} - \{(e_i, e_j); \mathcal{S}(e_i) \neq \mathcal{S}(e_j)\}\}$. We conclude that $dPairs = AllPairs - \sum_{\mathcal{S}(i),\mathcal{S}(j)\in dS:i<j} \mid \mathcal{S}(i) \mid * \mid \mathcal{S}(j) \mid$. Clearly, the number of pair of links that need to be distinguishable equals the number of all link pairs of the network if and only if the number of distinct sets of suspect links equals 1, *i.e.* the number of detection paths equals 1. ■

Appendix B

This section presents the proof of Theorem 3.

Proof: Paths in $\mathcal{P}'$ can be divided into three subsets of paths.

- P_1: paths that cross either e_1 or e_2, but not both.
- P_2: paths that cross both e_1 and e_2.
- P_3: paths that neither cross e_1 nor e_2.

On the one hand, we have
$\forall p \in P_2, \quad \delta_{pe_1} = 0$ and $\delta_{pe_2} = 0$.
Thus, $\forall p \in P_2, \ (\delta_{pe_1} + \delta_{pe_2} - 2\delta_{pe_1}\delta_{pe_2}) = 0$.
Contributing to $\sum_{p\in P_2}(\delta_{pe_1} + \delta_{pe_2} - 2\delta_{pe_1}\delta_{pe_2}) > 0$.
On the other hand, we have $\forall p \in P_3, \quad \delta_{pe_1} = 1$ and $\delta_{pe_2} = 1$.
Thus, $\forall p \in P_3, \quad (\delta_{pe_1} + \delta_{pe_2} - 2\delta_{pe_1}\delta_{pe_2}) = 0$.
Contributing to $\sum_{p\in P_3}(\delta_{pe_1} + \delta_{pe_2} - 2\delta_{pe_1}\delta_{pe_2}) = 0$.
Subsequently, $\sum_{p\in\mathcal{P}'}(\delta_{pe_1} + \delta_{pe_2} - 2\delta_{pe_1}\delta_{pe_2}) = \sum_{p\in P_1}(\delta_{pe_1} + \delta_{pe_2} - 2\delta_{pe_1}\delta_{pe_2})$.
Now, we have $\forall p \in P_1 \quad \delta_{pe_1} + \delta_{pe_2} = 1$ and $\delta_{pe_1}\delta_{pe_2} = 0$.
Thus, $\delta_{pe_1} + \delta_{pe_2} - 2\delta_{pe_1}\delta_{pe_2} = 1$.
Therefore, $\sum_{p\in P_1}(\delta_{pe_1} + \delta_{pe_2} - 2\delta_{pe_1}\delta_{pe_2}) = \mid P_1 \mid$.
We conclude that $\sum_{p\in\mathcal{P}'}(\delta_{pe_1} + \delta_{pe_2} - 2\delta_{pe_1}\delta_{pe_2}) = \mid P_1 \mid$. ■

Appendix C

This section presents the proof of Theorem 5.

Proof: Let $\mathcal{P}'$ be a set of paths that can distinguish between all links of $\mathcal{S}$. According to Theorem 1, for each $e_1, e_2 \in \mathcal{S}$ $\exists p \in \mathcal{P}'$ such that p crosses either e_1 or e_2, but not both. Thus, at most one link of $\mathcal{S}$ is not traversed by paths in $\mathcal{P}'$. We conclude that any localization solution must imperatively monitor $\mid \mathcal{S} \mid -1$ links of $\mathcal{S}$ in order to distinguish between all links. It follows that the localization solution that incurs the minimal probe cost is a solution that monitors exactly $\mid \mathcal{S} \mid -1$ links of $\mathcal{S}$ whose have the lowest measurement costs. Thus, $ThMinPcost(\mathcal{S}) = \sum_{e\in\mathcal{S}} C_e - \max_{e\in\mathcal{S}} C_e$. Note that such a solution is feasible only if each link of the

$|\mathcal{S}|-1$ links is monitored separately, which requires to have monitors deployed on the end nodes of each of these links. ■

APPENDIX D

This section presents the proof of Theorem 6.

Proof: We have $\max_{p\in\mathcal{P}} lc(\mathcal{S}_a^{(j)},p) = \sum_{\mathcal{S}_a^{(j)k}\in\mathcal{S}_a^{(j)}} \max_{p\in\mathcal{P}} lc(\mathcal{S}_a^{(j)k},p)$, where $lc(\mathcal{S}_a^{(j)k},p) = | p\cap\mathcal{S}_a^{(j)k} | * (| \mathcal{S}_a^{(j)k} | - | p\cap\mathcal{S}_a^{(j)k} |)$. Consider the variations of $lc(\mathcal{S}_a^{(j)k},p)$ with respect to the values of $| p\cap\mathcal{S}_a^{(j)k} |$. It can be easily shown that:

- $lc(\mathcal{S}_a^{(j)k},p)$ is increasing for $| p\cap\mathcal{S}_a^{(j)k} | < | \mathcal{S}_a^{(j)k} | /2$, and decreasing for $| p\cap\mathcal{S}_a^{(j)k} | > | \mathcal{S}_a^{(j)k} | /2$
- $\forall\ p_1, p_2 \in \mathcal{P}$, if $|| \mathcal{S}_a^{(j)k} | /2 - | p_1\cap\mathcal{S}_a^{(j)k} || = || \mathcal{S}_a^{(j)k} | /2 - | p_2\cap\mathcal{S}_a^{(j)k} ||$, then, $lc(\mathcal{S}_a^{(j)k},p_1) = lc(\mathcal{S}_a^{(j)k},p_2)$
- The global maximum of $lc(\mathcal{S}_a^{(j)k},p)$ is achieved at $| p\cap\mathcal{S}_a^{(j)k} | = | \mathcal{S}_a^{(j)k} | /2$

It follows that $\max_{p\in\mathcal{P}} lc(\mathcal{S}_a^{(j)k},p) = \min_{p\in\mathcal{P}} || \mathcal{S}_a^{(j)k} | /2 - | p_2\cap\mathcal{S}_a^{(j)k} ||$. Subsequently, $\max_{p\in\mathcal{P}} lc(\mathcal{S}_a^{(j)},p) = \min_{p\in\mathcal{P}} \sum_{\mathcal{S}_a^{(j)k}\in\mathcal{S}_a^{(j)}} || \mathcal{S}_a^{(j)k} | /2 - | \mathcal{S}_a^{(j)k}\cap p ||$ ■

APPENDIX E

This appendix demonstrates the correctness of conditions (11), (12) and (13).

Let q be a path, and let p_c be a prefix of q. We have:

(i) $\forall\mathcal{S}_a^{(j)k}, | \mathcal{S}_a^{(j)k}\cap q | \geq | \mathcal{S}_a^{(j)k}\cap p_c |$

(ii) $\exists\mathcal{S}_a^{(j)k}\in\mathcal{S}_a^{(j)}$ such that $| \mathcal{S}_a^{(j)k}\cap p_c | > | \mathcal{S}_a^{(j)k} | /2 \Rightarrow li(q,\mathcal{S}_a^{(j)}) > 0$

Proof: It is clear that $li(q,\mathcal{S}_a^{(j)}) = 0 \iff \forall\mathcal{S}_a^{(j)k}\in\mathcal{S}_a^{(j)} absval(| \mathcal{S}_a^{(j)k} | /2 - | q\cap\mathcal{S}_a^{(j)k} |) = 0 \iff \forall\mathcal{S}_a^{(j)k}\in\mathcal{S}_a^{(j)} | q\cap\mathcal{S}_a^{(j)k} | = | \mathcal{S}_a^{(j)k} | /2$. However, according to (i), $\forall\mathcal{S}_a^{(j)k} | \mathcal{S}_a^{(j)k}\cap p_c | > | \mathcal{S}_a^{(j)k} | /2 \Rightarrow | \mathcal{S}_a^{(j)k}\cap q | > | \mathcal{S}_a^{(j)k} | /2$. Therefore, **(ii)** is true. ■

(iii) $\forall\mathcal{S}_a^{(j)k} | \mathcal{S}_a^{(j)k}\cap p_c | > | \mathcal{S}_a^{(j)k} | /2 \Rightarrow li(q,\mathcal{S}_a^{(j)}) = li(p_c,\mathcal{S}_a^{(j)}) + \sum_{\mathcal{S}_a^{(j)k}\in\mathcal{S}_a^{(j)}} | \mathcal{S}_a^{(j)k}\cap p_c | - | \mathcal{S}_a^{(j)k}\cap q |$

Proof: $\forall\mathcal{S}_a^{(j)k} | \mathcal{S}_a^{(j)k}\cap p_c | > | \mathcal{S}_a^{(j)k} | /2 \Rightarrow \forall\mathcal{S}_a^{(j)k} | \mathcal{S}_a^{(j)k}\cap q | > | \mathcal{S}_a^{(j)k} | /2 \Rightarrow li(q,\mathcal{S}_a^{(j)}) = \sum_{\mathcal{S}_a^{(j)k}\in\mathcal{S}_a^{(j)}} | \mathcal{S}_a^{(j)k} | /2 - | \mathcal{S}_a^{(j)k}\cap q | = li(p_c,\mathcal{S}_a^{(j)}) - \sum_{\mathcal{S}_a^{(j)k}\in\mathcal{S}_a^{(j)}} | \mathcal{S}_a^{(j)k}\cap q | + \sum_{\mathcal{S}_a^{(j)k}\in\mathcal{S}_a^{(j)}} | \mathcal{S}_a^{(j)k}\cap p_c |$. ■

(iv) $\forall\mathcal{S}_a^{(j)k} | \mathcal{S}_a^{(j)k}\cap p_c | > | \mathcal{S}_a^{(j)k} | /2 \Rightarrow ProbeCost(q) \leq probeCost(p_c) + min_{e\in\mathcal{E}} C_e * li(q,\mathcal{S}_a^{(j)}) - li(p_c,\mathcal{S}_a^{(j)})$

Proof:

We have $ProbeCost(q) = \sum_{e\in p} C_e = \sum e\in p_c C_e + \sum_{e\in e\in q\setminus p_c} C_e = probeCost(p_c) + \sum_{e\in q\setminus p_c} C_e \leq probeCost(p_c) + min_{e\in\mathcal{E}} C_e * | q | - | p_c |$ By construction, $\bigcap_k \mathcal{S}_a^{(j)k} = \emptyset$. Therefore, $\forall p\in\mathcal{P} | p | \leq \sum_{\mathcal{S}_a^{(j)k}\in\mathcal{S}_a^{(j)}} | \mathcal{S}_a^{(j)k}\cap p |$. Hence, $ProbeCost(q) \leq probeCost(p_c) + min_{e\in\mathcal{E}} C_e * \sum_{\mathcal{S}_a^{(j)k}\in\mathcal{S}_a^{(j)}} | \mathcal{S}_a^{(j)k}\cap q | - | \mathcal{S}_a^{(j)k}\cap p_c |$. Further, $\forall\mathcal{S}_a^{(j)k} | \mathcal{S}_a^{(j)k}\cap p_c | > | \mathcal{S}_a^{(j)k} | /2$, thus, according to **(iii)**, $ProbeCost(q) \leq probeCost(p_c) + min_{e\in\mathcal{E}} C_e * li(q,\mathcal{S}_a^{(j)}) - li(p_c,\mathcal{S}_a^{(j)})$ ■

(v) $\forall\mathcal{S}_a^{(j)k}\in\mathcal{S}_a^{(j)} | \mathcal{S}_a^{(j)k}\cap p_c | \geq | \mathcal{S}_a^{(j)k} | /2 \Rightarrow li(q,\mathcal{S}_a^{(j)}) \geq li(p_c,\mathcal{S}_a^{(j)})$

Proof: According to **(i)**, $\sum_{\mathcal{S}_a^{(j)k}\in\mathcal{S}_a^{(j)}} | \mathcal{S}_a^{(j)k}\cap q | \geq \sum_{\mathcal{S}_a^{(j)k}\in\mathcal{S}_a^{(j)}} | \mathcal{S}_a^{(j)k}\cap p_c |$. Therefore, according to **(iii)**, **(v)** is true. ■

The correctness proof of Conditions (11), (12) and (13) is based on (ii), (iv) and (v). According to (10), when $minli = 0$, a good path must have a zero localization indicator and a probe cost that subordinates $minPc$. Therefore, according to (ii), all paths having a prefix that satisfies (11) are not good; and according to (iv), all paths having a prefix that satisfies (12) are not good. Further, according to (v), regardless the value of $minli$, the localization indicator of any path having a prefix that satisfies (13) dominates $minli$. Subsequently, according to (10), any path having a prefix that satisfies (13) is not good.

REFERENCES

[1] AGRAWAL, S., NAIDU, K. V. M., AND RASTOGI, R. Diagnosing link-level anomalies using passive probes. In *Proceedings of INFOCOM* 2007.

[2] BARFORD, P., DUFFIELD, N. G., RON, A., AND SOMMERS, J. Network performance anomaly detection and localization. In *Proceedings of INFOCOM* 2009.

[3] NAIDU, K. V.M., AND PANIGRAHI, D., AND RASTOGI, R. Detecting anomalies using end-to-end path measurements. In *INFOCOM* (2008).

[4] RISH, I., AND BRODIE, M. , MA, S., AND ODINTSOVA, N., AND BEYGELZIMER, A., AND GRABARNIK, G., AND HERNANDEZ, K. Adaptive diagnosis in distributed systems *IEEE Transactions on Neural Networks* Vol. 16 NO. 4 pp. 1088-1109 2005.

[5] RISH, I., AND BRODIE, M., AND ODINTSOVA, N., AND MA, S., AND GRABARNIK, G. Real-time problem determination in distributed systems using active probing In *Proceedings of the Network Operations and Management Symposium* 2004.

[6] CHENG, L., AND QIU, X., AND MENG, L., AND QIAO, Y., AND BOUTABA, R. Efficient active probing for fault diagnosis in large scale and noisy networks In *Proceedings of IEEE INFOCOM* 2010.

[7] BRODIE, M., AND RISH, I., AND MA, S., AND ODINTSOVA, N. Active probing strategies for problem diagnosis in distributed systems In *Proceedings of the International Joint Conferences on Artificial Intelligence* 2003.

[8] MAITREYA, N., AND SETHI, A. S. Probabilistic fault diagnosis using adaptive probing In *Proceedings of the Distributed systems: operations and management 18th IFIP/IEEE international conference on Managing virtualization of networks and services* 2007.

[9] ODINTSOVA, N., AND RISH, I., AND MA, S. Multi-fault Diagnosis in Dynamic Systems In *Proceedings of Integrated Management* 2005.

[10] BRODIE, M., AND RISH, I., AND MA, S. Optimizing Probe Selection for Fault Localization In *Proceedings of Distributed Systems Operation and Management* 2001.

[11] BEJERANO, Y., AND RASTOGI, R. Robust monitoring of link delays and faults in IP networks. *IEEE/ACM Transsaction on Networking*, Vol. 14 NO. 5 pp. 1092-1103 2006.

[12] ZHAO, Y., AND ZHU, Z., AND CHEN, Y., AND PEI, D., AND WANG, J. Towards efficient large-scale vpn monitoring and diagnosis under operational constraints. In *Proceedings of INFOCOM* 2009.

[13] SALHI, E., AND LAHOUD, S., AND COUSIN, B. Joint optimization of monitor location and network anomaly detection. In *Proceedings of IEEE LCN*. 2010.

[14] SALHI, E., AND LAHOUD, S., AND COUSIN, B. Heuristics for joint optimization of monitor location and network anomaly detection. In *Proceedings of IEEE ICC*. 2011.

[15] SALHI, E., AND LAHOUD, S., AND COUSIN, B. Localization of single network link-level anomalies. In *Proceedings of IEEE ICCCN*. 2012.

[16] DUFFIELD, N. Simple network performance tomography. In *Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement*. 2003.

[17] CHUDAK, F. AND CHMYOS, D. Improved Approximation Algorithms for the Uncapacitated Facility Location Problem. *ACM SIAM Journal on Computing* Vol. 33 NO. 1 pp. 1-25 2004.

[18] BRITE, 2012. http://www.cs.bu.edu/brite/.

[19] WAXMAN, B. Routing of Multipoint Connections *IEEE Journal on Selected Areas in Communications* vol. 6.9, pp. 1617 - 1622 , 1988.

[20] CPLEX, 2012. http://www.ilog.com/products/cplex/.