PROBABILISTIC EVENT-DRIVEN FAULT DIAGNOSIS THROUGH INCREMENTAL HYPOTHESIS UPDATING

M. Steinder and A. S. Sethi
Computer and Information Sciences Department
University of Delaware, Newark, DE
{*steinder,sethi*}*@cis.udel.edu*

Abstract: A probabilistic event-driven fault localization technique is presented, which uses a symptom-fault map as a fault propagation model. The technique isolates the most probable set of faults through incremental updating of the symptom explanation hypothesis. At any time, it provides a set of alternative hypotheses, each of which is a complete explanation of the set of symptoms observed thus far. The hypotheses are ranked according to a measure of their goodness. The technique allows multiple simultaneous independent faults to be identified and incorporates both negative and positive symptoms in the analysis. As shown in a simulation study, the technique is resilient both to noise in the symptom data and to the inaccuracies of the probabilistic fault propagation model. [1]

1. Introduction

This paper presents a non-deterministic event-driven fault localization [9, 10, 17] technique, which uses a probabilistic symptom-fault map as a fault propagation model. While investigating fault localization techniques suitable for bipartite fault propagation models, this paper states the following objectives:

- Usage of probabilistic reasoning, which is necessary to diagnose Byzantine problems or when relationships among system events may not be determined with certainty, e.g., due to their dynamic nature [5, 6, 8, 10, 11].
- Ability to isolate multiple simultaneous faults even if their symptoms overlap [6, 10], which improves the technique's applicability to large systems.
- Event-driven diagnosis, which avoids the inflexibility of window-based tools [1], is not prone to inaccuracies resulting from an incorrect time-window specification, and allows fault localization to be interleaved with testing.
- Resilience to lost and spurious symptoms [5, 8, 17], which may dramatically reduce its accuracy if not taken into account by a fault localization algorithm.
- High accuracy and low-polynomial computational complexity.

In addition to providing the above features, the fault localization technique proposed in this paper is incremental, i.e., the interpretation of an observed symptom is incorporated in a solution resulting from the interpretation of the previously observed

[1] Prepared through collaborative participation in the Communications and Networks Consortium sponsored by the U. S. Army Research Laboratory under the Collaborative Technology Alliance Program, Cooperative Agreement DAAD19-01-2-0011. The U. S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation thereon.

symptoms without re-analyzing them. Thanks to this feature, the algorithm continuously provides a system administrator with information about which faults are likely to exist in the system given symptoms observed thus far. In non-incremental techniques, such information is available on a periodic basis only [10, 17]. The technique proposed here produces a set of alternative hypotheses rather than just a single explanation. These hypotheses are ranked according to their measure of goodness. As a result, the system administrator obtains a better understanding of the system state. This feature also facilitates exchanging the hypotheses order as dictated by hypothesis ranking schemes that are not easy to express through a goodness function, e.g., those taking into account fault gravity, testing difficulty, or urgency of repair. Since an occasional inaccuracy of the most likely hypothesis may not be avoided, the ability to replace an incorrect hypothesis with its alternative without repeating the entire fault localization process improves the robustness of the fault management system.

While relationships between faults and symptoms in real-life systems are usually more complex than may be represented by a bipartite graph (in particular, they are frequently indirect), many fault localization techniques proposed in the literature [4, 10, 16, 17] use bipartite fault propagation models. The focus on this type of a model is justified by the following arguments: (1) Performing fault localization with more complex representations is difficult. (In general, the problem is NP-hard [10].) To avoid this complexity, more detailed models are frequently reduced to bipartite ones through a sequence of graph reduction operations [17]. (2) Building more complex models requires a profound knowledge of the underlying system, while symptom-fault maps may be obtained through external observation. In many real-life problems, only bipartite symptom-fault models are feasible [4]. (3) Some fault localization sub-problems may be accurately represented by bipartite symptom-fault maps [16], thereby necessitating fault localization algorithms suitable for bipartite fault propagation models. The distinguishing features of the approach presented in this paper, when compared to previous fault localization techniques suitable to use with a symptom-fault map as a fault propagation model, are as follows: the technique is more general by not assuming any particular problem domain or probabilistic model [4, 10, 17], resilient to lost and spurious symptoms [7, 10], event-driven [4, 7, 10, 17], incremental [4, 7, 10, 17, 15, 16], and more efficient [15, 16].

The paper is structured as follows. Section 2 describes the concept of probabilistic incremental hypothesis updating, which was originally introduced in [13]. In Section 3, incremental hypothesis updating is extended to include positive and lost symptoms in the analysis. Section 4 presents the methodology of dealing with spurious symptoms and discusses the necessary modifications to the original algorithm. In Section 5, the experimental study of the technique is described.

2. Incremental hypothesis updating

A symptom-fault map is a bipartite directed graph that, for every fault, encodes direct causal relationships between the fault and a set of symptoms observed when the fault occurs. We will use $\mathcal{F}$ and $\mathcal{S}$ to denote the sets of all possible faults and symptoms, respectively. In a non-deterministic model, with every fault $f_i \in \mathcal{F}$ a probability of its independent failure is associated, which is denoted by $p(f_i)$. The edge between $f_i \in \mathcal{F}$ and $s_j \in \mathcal{S}$ indicates that f_i may cause s_j. The edge is weighted with the probability of the causal implication, $p(s_j|f_i)$. A subset of symptoms observed by the

management application is denoted by $\mathcal{S}_O$. The purpose of fault localization is to find $\mathcal{F}_d \subseteq \mathcal{F}$ that maximizes the probability that (1) all faults in $\mathcal{F}_d$ occur and (2) each symptom in $\mathcal{S}_O$ is explained by at least one fault from $\mathcal{F}_d$.

The technique we present in this section, which is called *incremental hypothesis updating* [13] (IHU), creates a set of most likely hypotheses, which may all be presented to the system administrator. Rather than waiting for a specific period of time before presenting a solution, the technique makes all these hypotheses available on a continuous basis, and constantly upgrades them with the information learned from the arriving symptoms. This allows the administrator to initiate recovery actions sooner, and it allows additional testing procedures to be performed. Each hypothesis is a subset of $\mathcal{F}$ that explains all symptoms in $\mathcal{S}_O$. We say that hypothesis $h_j \subseteq \mathcal{F}$ explains symptom $s_i \in \mathcal{S}_O$ if it contains at least one fault that explains s_i. The hypotheses are ranked using a belief metric, b. The algorithm proceeds in an event-driven and incremental fashion. The execution triggered by the i^{th} symptom, s_i, creates a set of hypotheses, $\mathcal{H}_i$, each explaining symptoms s_1 through s_i. Set $\mathcal{H}_i$ is created by updating $\mathcal{H}_{i-1}$ with an explanation of symptom s_i. We define H_{s_i} as a set $\{f_k \in \mathcal{F}\}$ such that f_k may cause s_i, i.e., the fault propagation model contains a directed edge from f_k to s_i. Using the notation from [10], H_{s_i} is the domain of symptom s_i.

After the i^{th} symptom is processed, belief metric b_i represents the probability that (1) all faults belonging to h_j have occurred, and (2) h_j explains every observed symptom $s_k \in \mathcal{S}_{O,i} = \{s_1, \ldots, s_i\}$. Formally, $b_i(h_j)$ is defined as follows:

$$b_i(h_j) = \Big(\prod_{f_k \in h_j} p(f_k) \Big) \prod_{s_l \in \mathcal{S}_{O,i}} \Big(1 - \prod_{f_k \in h_j} (1 - p(s_l|f_k) \Big) \tag{1}$$

To incorporate an explanation of symptom s_i into the set of fault hypotheses, in the i^{th} iteration of the algorithm, we analyze each $h_j \in \mathcal{H}_{i-1}$. If h_j is able to explain symptom s_i, we put h_j into $\mathcal{H}_i$. Otherwise, h_j has to be extended by adding to it a fault from H_{s_i}. To avoid a very fast growth in the size of $\mathcal{H}_i$, the following heuristic is used. Fault $f_l \in H_{s_i}$ may be added to $h_j \in \mathcal{H}_{i-1}$ only if the size of h_j, $|h_j|$, is smaller than $\mu(f_l)$, the minimum size of a hypothesis in $\mathcal{H}_{i-1}$ that contains f_l and explains s_i. The usage of this heuristic is derived from the fact that the probability of multiple simultaneous faults is small. Therefore, of any two hypotheses containing f_l, the hypothesis that contains the fewest faults is the most likely to constitute the optimal symptom explanation. Thus, since it is not efficient to keep all possible hypotheses, we remove those that are bigger in size. While updating the set of hypothesis, $b_i(h_j)$ is approximated iteratively based on $b_{i-1}(h_j)$ using the following equations:

– If $h_j \in \mathcal{H}_{i-1}$ and h_j explains s_i

$$b_i(h_j) = b_{i-1}(h_j) \Big(1 - \prod_{f_l \in h_j \cap H_{s_i}} (1 - p(s_i|f_l)) \Big) \tag{2}$$

– Otherwise, if f_l explains s_i

$$b_i(h_j \cup \{f_l\}) = b_{i-1}(h_j)\, p(f_l)\, p(s_i|f_l) \tag{4}$$

The upper bound on the worst case computational complexity of the resultant algorithm is $\mathcal{O}(|\mathcal{S}_O|k|\mathcal{F}|)$, where k is the maximum size of the set of hypotheses and k is $\mathcal{O}(|\mathcal{F}|)$ (in our study, $k = 2|\mathcal{F}|$). When $|\mathcal{H}_i| = k$, a new hypothesis may be added to $\mathcal{H}_i$ only after a hypothesis with the smallest $b_i()$ is removed.

3. The analysis of positive symptoms

The original version of IHU [13] formulates explanations of observed system disorder while not taking advantage of the fact that some possible indications of the disorder have not been observed. As many researchers point out [2, 17], the fact that many of its possible symptoms have not been observed should decrease our confidence in the fault's occurrence. In the realm of fault localization, an observation of network disorder is called a *negative symptom*. Both an opposite observation and the lack of any observation are considered *positive symptoms*. As it was shown in the study on fault localization with belief networks [15], the inclusion of positive symptoms into the fault localization process may significantly increase its accuracy.

To include the analysis if positive symptoms in IHU, the belief metric b_i^* associated with hypothesis $h_j \in \mathcal{H}_i$ needs to contain two components: a negative component b_i^n and a positive component b_i^p, where $b_i^*(h_j) = b_i^n(h_j)\, b_i^p(h_j)$ and $b_i^n(h_j) = b_i(h_j)$ of Equation (1). The positive component is defined as the probability that faults in h_j have not generated any of the symptoms in $\mathcal{S} - \mathcal{S}_{o,i}$. It decreases the value of the belief metric associated with hypothesis h_j if many of the symptoms that can occur as a result of faults in h_j have not been observed. The positive component of $b_i^*(h_j)$ is expressed through the following equation.

$$b_i^p(h_j) = \prod_{s_l \in \mathcal{S} - \mathcal{S}_{O,i}} \prod_{f_k \in h_j} (1 - p(s_l|f_k)) \tag{4}$$

When investigating a fault localization technique that takes advantage of positive symptoms, two properties of the managed system have to be taken into account: symptom observability ratio and symptom loss rate, which lead to refinements in the calculation of b_i^p presented in the following sections.

3.1 Symptom observability ratio

Frequently, an indication of an existing disorder may not be observed by the management system because the system configuration configuration excludes some conditions from being monitored, or filters out some of the symptoms before they reach the management application. If this fact is not taken into account, the reduction of $b_i^*(h_j)$ caused by the positive multiplier $b_i^p(h_j)$ may be excessive. Symptoms which may not be observed as a result of the management system configuration may be dealt with by not including them in the fault propagation model. An alternative solution, which preserves the model despite the management system configuration changes, associates a flag 1 or 0 with every symptom in the model to indicate that, in the current configuration, the symptom is observable or not observable, respectively. We will denote by $\mathcal{S}^o \subseteq \mathcal{S}$ the set of all symptoms which are observable in a current management system configuration. When symptom observability status is taken into account, the second product in Equation (4) is calculated over $s_l \in \mathcal{S}^o - \mathcal{S}_{O,i}$ rather than $s_l \in \mathcal{S} - \mathcal{S}_{O,i}$.

The ratio of the number of all observable symptoms to the number of all possible symptoms is called an observability ratio, and is denoted by $OR = |\mathcal{S}^o|/|\mathcal{S}|$ [15]. The observability ratio is an important parameter of the fault management system, which informs us of the extensiveness of the system instrumentation. It may be expected that a higher instrumentation level allows fault localization to be more accurate, but causes it to be less efficient as it requires the processing of more symptoms.

3.2 Symptom loss

In a real-life system, a symptom that has been triggered by faults in h_j may be lost before it reaches the management application as a result of using an unreliable communication mechanism to transfer alarms from their origin to the management node, as is the case with the SNMP protocol [3], or too liberal threshold values which prevent an existing problem from being reported. When a fault localization algorithm relies on positive information, a high rate of lost symptoms, if ignored by the algorithm, can reduce its accuracy. Thus, in the management system in which symptom delivery is not guaranteed, including positive symptoms into account necessitates the analysis of lost symptoms as well.

Let us denote by $p_{\text{loss}}(s_i)$ the probability that symptom $s_i \in \mathcal{S}$ is lost. The value of $p_{\text{loss}}(s_i)$ may be derived from a packet loss rate in the communication system, or from the confidence measure associated with the system baselining tool used to calculate the monitored threshold values. Symptom loss is included into the fault localization algorithm by modifying the definition of $b_i^p(h_j)$ (Equation (4)) as follows.

$$b_i^p(h_j) = \prod_{s_l \in \mathcal{S}^o - \mathcal{S}_{O,i}} \Big(p_{\text{loss}}(s_l) + (1 - p_{\text{loss}}(s_l)) \prod_{f_k \in h_j} (1 - p(s_l|f_k)) \Big) \tag{5}$$

3.3 Incremental calculation of $\boldsymbol{b^p}$

IHU based on both positive and negative symptoms proceeds as follows. Initially, all observable alarms are considered positive symptoms. The only valid hypothesis is $\emptyset$, and $b_i^n(\emptyset) = b_i^p(\emptyset) = 1$. In the process of analyzing new symptoms, the value of belief metric $b_i^*(h_j)$ is calculated by multiplying $b_i^n(h_j)$ and $b_i^p(h_j)$, where $b_i^n(h_j)$ is computed incrementally using Equations (2)-(3). We obtain $b_i^p(h_j)$ as follows.

– If $h_j \in \mathcal{H}_{i-1}$ explains symptom s_i, then $b_i^p(h_j)$ may be approximated using the following formula.

$$b_i^p(h_j) = \frac{b_{i-1}^p(h_j)}{\prod_{f_l \in h_j} (p_{\text{loss}}(s_l) + (1 - p_{\text{loss}}(s_l))(1 - p(s_l|f_k)))} \tag{6}$$

– Otherwise, let $f_l \in H_{s_i}$ be a fault used to extend h_j. The value of $b_i^p(h_j \cup \{f_l\})$ is calculated as follows.

$$b_i^p(h_j \cup \{f_l\}) = b_{i-1}^p(h_j)\, b_i^p(\{f_l\}) \tag{7}$$

In Equation (7), $b_i^p(\{f_l\})$ denotes the positive component of a belief metric associated with a singleton hypothesis $\{f_l\}$ calculated given all symptoms observed thus far. The values of $b_i^p(\{f_l\})$ are pre-computed when the model is initialized. After every symptom observation, $b_i^p(\{f_l\})$ is incrementally updated using Equation (6).

4. Dealing with spurious symptoms

In real-life communication systems, an observation of a network state is frequently disturbed by the presence of spurious symptoms, which are caused by intermittent network faults or by overly restrictive threshold values. Spurious symptoms, if not taken into account by the fault localization process, may significantly deteriorate its accuracy. When a fault localization algorithm does not recognize that some symptoms may be spurious (as such they do not require an explanation), it strives to find the

explanation of all the observed symptoms, thereby creating hypotheses which contain many non-existent faults [15]. In this section, we introduce an extended version of IHU, IHU+, which incorporates spurious symptoms in the analysis.

To deal with spurious symptoms IHU has to be modified as follows. Let s_i be the i^{th} observed symptom and let $p_{\text{s}}(s_i)$ denote the probability that symptom s_i is spuriously generated. While deciding whether hypothesis $h_j \in \mathcal{H}_{i-1}$ should be placed in $\mathcal{H}_i$ without modification or extended, the algorithm has to consider two possibilities: (1) that the symptom is valid and (2) that the symptom is spurious. When hypothesis h_j explains s_i, then regardless of these two possible interpretations of symptom s_i, hypothesis h_j can be added to $\mathcal{H}_i$ and the two choices are incorporated in the calculation of the belief metric for h_j. When hypothesis h_j does not explain s_i, then treating s_i as valid necessitates extending h_j, and treating s_i as spurious allows us to put h_j in $\mathcal{H}_i$ without extension. Since the first and second cases occur with probability $1 - p_{\text{s}}(s_i)$ and $p_{\text{s}}(s_i)$, these values are used as multipliers embedded in the calculation of the corresponding values of the belief metric. Recall from Section 2, that the original algorithm does not allow adding $h_j \in \mathcal{H}_{i-1}$ to $\mathcal{H}_i$ unless it explains or is extended to explain symptom s_i.

The inclusion of spurious symptoms into the analysis only affects the calculation of the negative component, $b_i^{+n}(h_j)$, of the belief metric, $b_i^{+}(h_j)$, while the positive component remains the same, i.e., $b_i^{+p}(h_j) = b_i^{p}(h_j)$ (Eqns. (6)-(7)). The modified negative component, $b_i^{+n}(h_j)$, is calculated iteratively as follows.

– If $h_j \in \mathcal{H}_{i-1}$ explains symptom s_i, then

$$b_i^{+n}(h_j) = b_{i-1}^{+n}(h_j)\Big(p_{\text{s}}(s_i) + (1 - p_{\text{s}}(s_i))\,(1 - \prod_{f_l \in h_j \cap H_{s_i}} (1 - p(s_i|f_l))\Big) \quad (8)$$

– Otherwise

$$b_i^{+n}(h_j) = b_{i-1}^{+n}(h_j)\, p_{\text{s}}(s_i) \quad (9)$$

In addition, for every fault $f_l \in H_{s_i}$ used to extend h_j

$$b_i^{+n}(h_j \cup \{f_l\}) = b_{i-1}^{+n}(h_j)\, p(f_l)\, p(s_i|f_l)\, (1 - p_{\text{s}}(s_i)) \quad (10)$$

Besides modifying the definition of the belief metric, the inclusion of the spurious symptoms' analysis in the fault localization process necessitates two additional changes in the original IHU. Recall from Section 2 that IHU takes advantage of two heuristics that allow us to limit the size of the set of hypotheses. The first heuristic forbids adding fault f_l to hypothesis $h_j \in \mathcal{H}_i$ if the size of the resultant hypothesis $h_j \cup \{f_l\}$ would be greater than $\mu(f_l)$. The second heuristic applied by Algorithm IHU limits the maximum size of the set of hypotheses to $k \in \mathcal{O}(|\mathcal{F}|)$ and removes the least probable hypotheses if this limit is exceeded. These two heuristics are modified in IHU+ as described in the following sections.

4.1 Calculating hypothesis size

In IHU, function $\mu(f_l)$ is defined as the minimum size of $h_k \in \mathcal{H}_{i-1}$ that contains f_l and explains symptom s_i, where the size of h_k is $|h_k|$. In IHU+, the size of h_j, $\alpha(h_j)$ is defined as the number of faults in h_j plus the number of symptoms observed so far that h_j considers spurious. This modification serves two purposes. It:

– Helps avoid the creation of duplicate hypotheses.
  Duplicate hypotheses introduce redundancy into the set of hypotheses, which may affect the accuracy of the technique. They increase the size of the set of hypotheses thereby making hypothesis removal due to the excessive set size more frequent. Thus, they increase the probability of removing a (currently) least likely hypothesis that may later turn out to be optimal. Although it is possible to unify duplicate hypotheses within the computational complexity bound of IHU+, the necessity to do so renders the implementation of the algorithm more difficult.
– Prevents hypotheses that contain fewer faults while not explaining many symptoms from being given unwarranted preference.
  When small hypotheses are unfairly favored over bigger hypotheses, it is difficult for the algorithm to extend a small hypothesis so that it provides an explanation to a bigger number of symptoms. As a result, the algorithm is likely not to provide an explanation to many observed symptoms.

4.2 Controlling hypotheses number

The second heuristic applied by Algorithm IHU limits the maximum size of the set of hypotheses to $k \in \mathcal{O}(|\mathcal{F}|)$. To add a new hypothesis to $\mathcal{H}_i$, when $|\mathcal{H}_i| = k$, a hypothesis h_l for which $b_i(h_l)$ is minimal must be first removed from $\mathcal{H}_i$. It is possible that symptoms to be received in the next iterations would increase the belief associated with h_l so that h_l would become the most probable hypothesis. If such h_l is removed at an earlier stage of the fault localization process, the algorithm will not propose the optimal solution. The phenomenon of removing a hypothesis that would become optimal at a later stage of fault localization, if it was kept in the set of hypotheses, will be referred to as the problem of *premature hypothesis removal*.

Although the problem of premature hypothesis removal exists regardless of including positive, lost, and spurious symptoms into the analysis, in most cases it may be ignored. A hypothesis removal due to the big size of $\mathcal{H}_i$ is a rare event, and it usually happens after many symptoms have been observed and analyzed. At this stage, the algorithm is already converging to the final solution, thus the removed hypothesis is not likely to become optimal in the future. However, when spurious symptoms are included in the analysis, the size of $\mathcal{H}_i$ grows much faster, and therefore the probability of prematurely removing an optimal hypothesis is high. The early removal of an optimal hypothesis is caused by the positive component of the belief metric, whose value may be very small if at this stage of fault localization, only a few symptoms related to the optimal hypothesis have been observed. The crux of the problem is that $b^{+p}(h_j)$ is calculated as if the current set of observed symptoms was the final one.

IHU+ avoids the problem of the premature hypothesis removal by using function rank_i rather than b^+ to choose a hypothesis that has to be removed. Similar to the belief metric, function rank_i is composed of positive and negative components b^{+p} and b^{+n}, but the contribution of b_i^{+p} is weighted according to the number of symptoms observed so far. In the following definition of $\mathrm{rank}_i(h_j)$, $B_i^{+n}(h_j)$ and $B_i^{+p}(h_j)$ represent logarithmic-scale values of $b_i^{+n}(h_j)$ and $b_i^{+p}(h_j)$, respectively.

$$\mathrm{rank}_i(h_j) = B_i^{+n}(h_j) + \beta(i) B_i^{+p}(h_j) \tag{11}$$

Function $\beta(i)$ represents the contribution of the positive belief-metric component. In general, function $\beta(i)$ should assume a very small value when the number of symp-

toms observed so far, i, is small, and increase asymptotically to 1 as the value of i increases. In this study, we define $\beta(i)$ as follows.

$$\beta(i) = 1 - 2^{-SW(\frac{i-1}{EEF})^2} \tag{12}$$

In Equation (12), the expected evidence factor, EEF, and the average symptom weight, SW, are model-dependent. The expected evidence factor determines how quickly the value of $\beta(i)$ should converge to 1 in the absence of spurious symptoms. It is proportional to the average number of symptoms which may be observed per fault, i.e., $\mathrm{EEF} = c\frac{|\mathcal{S}|\mathrm{OR}}{|\mathcal{F}|}$. In this study, we use $c = 4$. The average symptom weight accounts for the fact that some symptoms may be spurious, and, as such, should not increase the value of $\beta(i)$. This value should be equal to 1 when no spurious symptoms occur, and decrease as the spurious symptom probability increases. We define SW using the following formula.

$$\mathrm{SW} = 1 - \frac{\sum_{s_i \in \mathcal{S}} p_\mathrm{s}(s_i)}{\sum_{s_i \in \mathcal{S}} \sum_{f_l \in \mathcal{F}} p(s_i|f_l) + \sum_{s_i \in \mathcal{S}} p_\mathrm{s}(s_i)} \tag{13}$$

The values of EEF and SW are pre-computed at the model initialization phase, and remain constant during the process of fault localization, as long as the fault propagation model is not changed. Other definitions of function β are possible. For instance, we could incorporate a temporal aspect into function β by increasing its value with time. Such a definition could represent a property that, after a certain time since the fault localization process is started, all relevant symptoms should have been observed.

4.3 IHU+ algorithm

We are now ready to define an extended version of the incremental algorithm, IHU+, which incorporates positive, lost, and spurious symptoms in the analysis and is parametrized by observability ratio OR, symptom-loss probability function p_loss, and spurious-symptom probability function p_s.

Algorithm: Incremental hypothesis updating – IHU+(OR,$\mathbf{p_{loss}}$,$\mathbf{p_s}$)

let $\mathcal{H}_0 = \{\emptyset\}$, $b_0^{+n}(\emptyset) = b_0^{+p}(\emptyset) = 1$, and $\alpha(\emptyset) = 0$
for every observed symptom s_i*:*
  let $\mathcal{H}_i = \emptyset$*, and for all* $f_l \in \mathcal{F}$ *let* $\mu(f_l) = |\mathcal{F}| + |\mathcal{S}_O|$
  for all $h_j \in \mathcal{H}_{i-1}$ *do*
    for all $f_l \in h_j$ *such that* $f_l \in H_{s_i}$
      set $\mu(f_l) = \min(\mu(f_l), \alpha(h_j))$
      add h_j *to* $\mathcal{H}_i$ *and calculate* $b_i^+(h_j)$
  for all $h_j \in \mathcal{H}_{i-1} \setminus \mathcal{H}_i$ *do*
    if $p_\mathrm{s}(s_i) > 0$
      add h_j *to* $\mathcal{H}_i$*, calculate* $b_i^+(h_j)$*, and set* $\alpha(h_j) = \alpha(h_j) + 1$
    for all $f_l \in \mathcal{F} \cap H_{s_i}$ *such that* $\mu(f_l) > \alpha(h_j)$ *do*
      add $h_j \cup \{f_l\}$ *to* $\mathcal{H}_i$*, compute* $b_i^+(h_j \cup \{f_l\})$*, and set* $\alpha(h_j) = \alpha(h_j) + 1$
choose $h_j \in \mathcal{H}_{|\mathcal{S}_N|}$ *such that* $b^+_{|\mathcal{S}_N|}(h_j)$ *is maximum*

Observe that the worst-case computational complexity of the algorithm that takes positive, lost, and spurious symptoms into account is still $\mathcal{O}(|\mathcal{S}_O||\mathcal{F}|^2)$.

5. Simulation study

In this section, we describe a simulation study performed to evaluate the techniques presented in Sections 2, 3, and 4. As a real-life application domain we chose end-to-end service failure diagnosis [16], which deals with isolating faults responsible for a malfunctioning of end-to-end connectivity between systems. The first step toward diagnosing these problems is to isolate the responsible host-to-host services, where a host is an intermediate node used to provide the end-to-end connectivity. In the problem of end-to-end service-failure diagnosis, a fault propagation model is a bipartite causality graph with host-to-host and end-to-end service failures at the tails and at the heads of the edges, respectively.

The simulation study presented in this paper uses tree-shaped network topologies, which result, for example, from the usage of the Spanning Tree Protocol [12] as the data-link layer routing protocol. The usage of tree-shaped topologies greatly simplifies their random generation, while it does not affect the validity of the results presented in this section. We focus on diagnosing Byzantine types of problems, for which the usage of a non-deterministic fault propagation model is necessary.

We design the simulation described in this section according to the model we previously used to evaluate another fault localization algorithm [15], which is based on belief propagation in belief networks. We use OR, LR , and SSR to denote the observability ratio ($|\mathcal{S}_O|/|\mathcal{S}|$), ratio of the number of generated alarms that were lost to the number of all generated alarms (i.e., alarm loss rate), and probability that an alarm is generated in a spurious manner (i.e., spurious symptom rate), respectively. We aim at creating a homogeneous set of test scenarios to establish the upper limit on the accuracy of the proposed techniques and its relationship to the parameters of the simulation model. Consequently, we assume that the fault propagation model used in the study accurately approximates the relationships that exist in the real system.

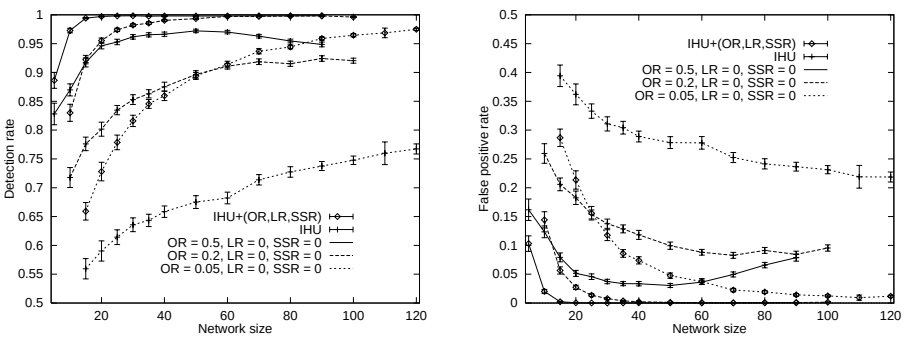
Given the simulation model with parameters OR, LR, and SSR for a given network topology of size n, where n represents the number of intermediate network nodes, we design 100 simulation cases. We build a random tree-shaped topology, and generate the probability distribution in the fault localization model. The independent failure probabilities and conditional probabilities are uniformly distributed in ranges $[0.001, 0.01]$ and $(0, 1)$, respectively. We randomly choose $\mathrm{OR}|\mathcal{S}|$ observable symptoms, and place them in the set of observable symptoms, $\mathcal{S}^o$. In a simulation case, we create a number of simulation scenarios (typically 100-200) as follows. We randomly generate a set of faults that exist in the system, $\mathcal{F}_c \subseteq \mathcal{F}$. Using $\mathcal{F}_c$ and the conditional probability distribution we randomly generate the set of observed negative symptoms $\mathcal{S}_O \subseteq \mathcal{S}^o$. When $\mathrm{SSR} > 0$, we also randomly choose $\mathrm{SSR}\,|\mathcal{S}^o|$ symptoms from $\mathcal{S}^o$, and add them to $\mathcal{S}_O$. When $\mathrm{LR} > 0$, we remove $\mathrm{LR}\,|\mathcal{S}_O|$ random symptoms from $\mathcal{S}_O$. Then, we run algorithms IHU, IHU+, or both to produce the most probable explanation of $\mathcal{S}_O$, $\mathcal{F}_d$, i.e., the hypothesis with the highest value of belief metric in the final set of hypotheses proposed by the algorithm. We compare $\mathcal{F}_d$ to $\mathcal{F}_c$, and calculate the detection rate $\mathrm{DR} = \frac{|\mathcal{F}_c \cap \mathcal{F}_d|}{|\mathcal{F}_c|}$ and false positive rate $\mathrm{FPR} = \frac{|\mathcal{F}_d - \mathcal{F}_c|}{|\mathcal{F}_d|}$.

5.1 The impact of including positive symptoms

To evaluate the impact of including positive symptoms into fault localization, we set $\mathrm{LR} = 0$, and $\mathrm{SSR} = 0$ in the simulation model. Correspondingly, we use $p_{\mathrm{loss}}(s_i) = 0$

and $p_{\mathrm{s}}(s_i) = 0$ in the fault propagation model. While setting OR to 0.5, 0.2, or 0.05, we compare DR and FPR achievable with IHU, which does not take positive symptoms into account, and IHU+, which includes positive symptoms in the analysis.

As presented in Figs. 1(a) and 1(b), including positive symptoms in the process of fault localization allows DR to be significantly increased and FPR to be significantly decreased. Overall, thanks to the positive information, the fault localization accuracy improves. The smaller OR, the bigger the improvement. Parameter OR determines the system instrumentation level defined as the average number of symptoms that may be observed per fault. (Note that the average number of symptoms in the system is a squared function of n, thus the system instrumentation level naturally improves when n increases.) It may be concluded that, in poorly instrumented systems, positive symptoms may be effectively used to improve the accuracy of the fault localization process without worsening its performance.



(a) Detection rate (b) False positive rate

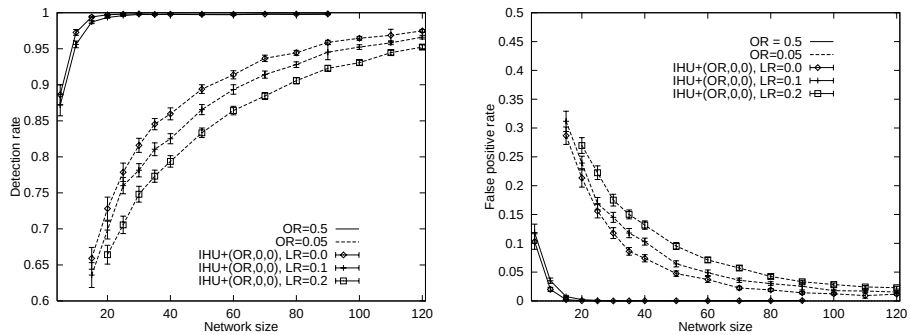
Figure 1. Accuracy achievable with algorithms IHU and IHU+ for various ORs.

5.2 The impact of ignoring symptom loss on the accuracy of fault localization

To isolate the impact of symptom loss on the accuracy of fault localization, we set $\mathrm{SSR} = 0$, and vary LR from 0.0 to 0.2. In the fault propagation model, we use $p_{\mathrm{loss}} = 0$, and $p_{\mathrm{s}} = 0$. (The fault localization algorithm effectively ignores the symptom loss.) We apply algorithm IHU+ to this model.

Symptom loss, when ignored by the fault localization process, does indeed decrease its accuracy: we observe a decrease of DR (Fig. 2(a)) and an increase of FPR (Fig. 2(b)). The strength of the symptom-loss impact on the accuracy is related to the value of LR and the system instrumentation level. Nonetheless, the decrease of accuracy caused by symptom loss is small (within 5% for both DR and FPR), which allows us to conclude that IHU+ is resilient to symptom loss even when it relies on positive information to perform fault diagnosis and does not include the explicit representation of lost symptoms in its model. To determine whether including this representation may improve the fault localization accuracy, we observe that decreasing accuracy when symptoms may be lost is due to two factors: (1) fewer symptoms are observed and therefore the system instrumentation level perceived by the fault management application decreases, and (2) some symptoms are incorrectly interpreted as positive ones. The relative contribution of these two factors determines the upper bound on the possible increase in the accuracy resulting from including symptom loss

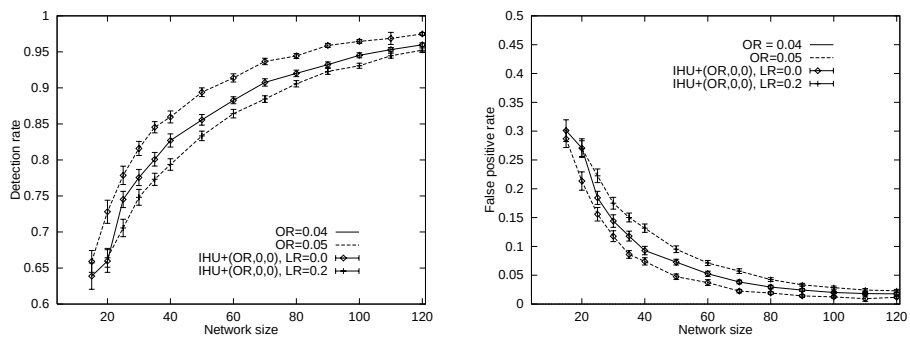
in the fault propagation model. Observe that the impact of only the second factor may be alleviated by including the representation of symptom loss in the model.



(a) Detection rate (b) False positive rate

Figure 2. The impact of symptom loss on the accuracy for various ORs and LRs.

To estimate the relative impact of factors (1) and (2), we perform another experiment. We execute the simulation study using the following parameters of the simulation model: (1) $\mathrm{OR} = 0.05$, $\mathrm{LR} = 0.0$, (2) $\mathrm{OR} = 0.05$, $\mathrm{LR} = 0.2$, and (3) $\mathrm{OR} = 0.04$, $\mathrm{LR} = 0.0$. The amount of information provided to the fault localization algorithm in the second and third cases is the same, because 0.05(1-0.2)=0.04. Thus the difference between the accuracies observed in the first and second cases represents the impact of factor (1). The difference between the accuracies observed in the second and third cases represents the impact of factor (2). As shown in Figs 3(a) and 3(b) the overall decrease of accuracy due to symptom loss is split evenly between the two factors. This lets us conclude that, were symptom loss represented in the fault propagation model, the resulting improvement in accuracy could not be greater than 2-2.5%. Indeed, our experiments with a fault propagation model using $p_{\mathrm{loss}}(s_i) = 0.2$ did not reveal any statistically provable improvement in accuracy. With higher values of LR, some small improvement in accuracy has been observed.



(a) Detection rate (b) False positive rate

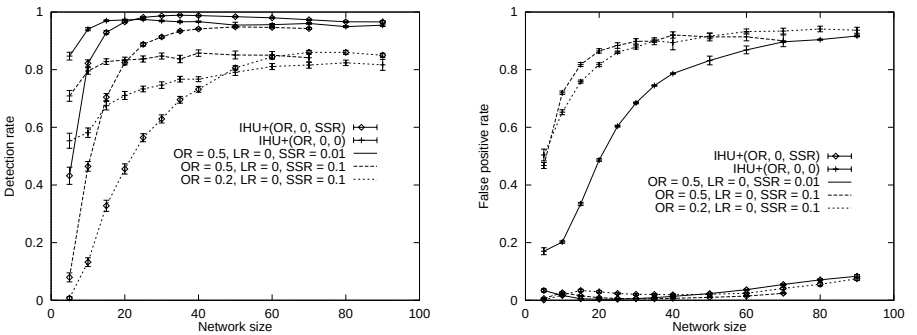
Figure 3. The impact of factors (1) and (2) in system with $\mathrm{OR} = 0.05$ and $\mathrm{OR} = 0.2$.

This simulation study assumes that all symptoms are equally likely to be lost, while in reality $p_{\mathrm{loss}}(s_i)$ is different for different symptoms, e.g., when symptom are trans-

mitted in-band. We expect that when the symptom-loss probabilities are not equal, the benefit of including symptom loss in the analysis would be more evident.

5.3 The impact of analyzing spurious symptoms

The impact of including spurious symptoms in the fault localization process is evaluated by applying IHU+ to fault propagation models using $p_{\mathrm{s}}(s_i) = 0$ and $p_{\mathrm{s}}(s_i) = \mathrm{SSR}$, respectively. We vary OR between 0.5 and 0.2, and use SSR of 0.01 and 0.1. As shown in Fig. 4(a), the inclusion of spurious symptoms in the fault localization process in small networks decreases DR. This is explained by the fact that in poorly instrumented networks only a few symptoms are available to the fault localization process. When the possibility of spurious symptoms is taken into account, and the amount of available evidence is small, the algorithm concludes that there is no sufficient evidential support for the existence of faults, and considers all the observed symptoms spurious. Otherwise, DR would be higher (Fig. 4(a)) but FPR would be very high as well (Fig. 4(b)). When system instrumentation improves, so does the DR of IHU+ with an accurate representation of spurious symptoms in the fault propagation model. Overall, we conclude that including spurious symptoms in the fault propagation model has a big impact on the accuracy of the fault localization algorithm. However, to take full advantage of this capability, the system instrumentation level should be increased correspondingly to the rate with which spurious symptoms are generated.



(a) Detection rate (b) False positive rate

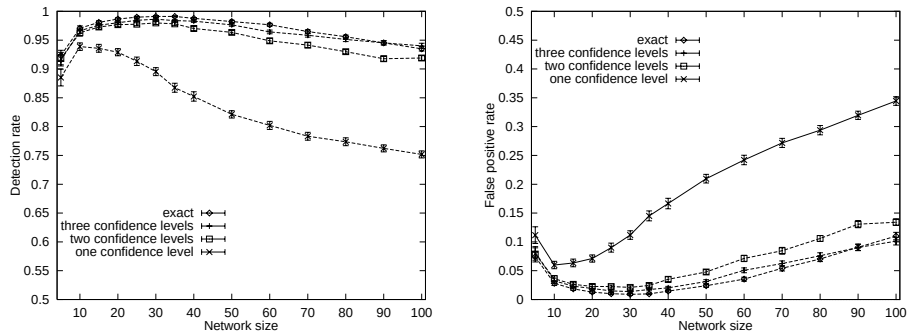
Figure 4. The change of accuracy as a result of spurious symptoms analysis.

5.4 The impact of conditional probability estimation errors

In the final set of experiments we evaluate the impact of conditional probability estimation errors on the fault localization accuracy. We consider a scenario in which instead of the accurate conditional probability values, a small number of confidence levels, c, are being used. To represent the real-life probability p, the model uses the i^{th} confidence level, where $i = \lfloor pc \rfloor$. Thus, effectively, real-system probability p is mapped into propagation-model weight $\frac{\lfloor pc \rfloor}{c} + \frac{1}{2c}$. The creation of the probability model by a human is feasible, if high fault-localization accuracy may be achieved even when only a small number of confidence levels is used.

Fig. 5(a) and 5(b) compare the DR and FPR of Algorithm IHU having exact knowledge of the probability distribution with the DR and FPR achieved using one,

two, and three confidence levels for various observability ratios. The figures prove an important property of the algorithm presented in this paper: it allows the expert to use a small set of meaningful qualitative probability assignments such as *unlikely*, *possible*, and *likely*, rather than exact probabilities, while preserving very high accuracy.



(a) Detection rate (b) False positive rate

Figure 5. Accuracy for various granularities of confidence levels.

6. Conclusion

The technique proposed in this paper isolates the most probable set of faults through incremental updating of the symptom explanation hypothesis. It uses a probabilistic model, which makes the technique applicable to systems with a high degree of non-determinism. While assuming the pre-existence of such a model, the technique is robust against the model's imperfection. As shown in the simulation study, the technique offers high accuracy, even in the presence of observation noise. It also has low polynomial complexity. When applied to the problem of end-to-end service failure diagnosis, our implementation of the technique solves multi-fault scenarios in networks composed of more than 100 routers or bridges within less that 10 seconds.

Since fault localization is not a new problem and many fault localization techniques have already been proposed, it is important to consider comparing these techniques with respect to their accuracy and performance. Unfortunately, the techniques proposed in the literature [4, 7, 10, 15–17] that are suitable for bipartite models differ with respect to assumptions they are based on, capabilities, and problems they aim at addressing. Some of the properties that distinguish IHU from the previously published approaches are introduced in Section 1. The different assumptions and capabilities render the techniques difficult to compare in quantitative terms as they make any such comparison inherently unfair. A set of objective criteria that allow the comparison to be performed have yet to be identified, which is an interesting future research problem.

Nevertheless, IHU may be compared to our previously investigated fault localization approach, which is based on belief updating in belief networks [15, 16]. The belief-network approach is more flexible as it does not constrain the shape of a fault propagation model to a bipartite one, but it is not incremental and its computational complexity, even in bipartite models, is higher. Thus, while the belief-network approach offers similar accuracy and resilience to model imperfections and observation noise as IHU, its scalability is significantly lower.

Some of the observations made in the simulation study presented in this paper, e.g., the dependence of the benefit resulting from positive symptoms analysis on the system instrumentation level or necessity to increase system instrumentation level in systems with high spurious symptoms rates, are rather natural and could have been anticipated. Our study allows these observations to be quantified. Since similar results have also been obtained in the analogous study on the belief-network approach [15], we believe these results apply to the fault localization problem in general.

Future work will include designing a distributed version of the algorithm, which explores the domain semantics of management systems. In the application to end-to-end service failure diagnosis, the distributed technique will follow the initial ideas presented in [14]. The algorithm presented in this paper assumes that alternative causes of the same event should be combined using logical OR. It will be extended to allow other models such as AND or NOT models. [2]

References

[1] K. Appleby, G. Goldszmidt, and M. Steinder. Yemanja–a layered event correlation system for multi-domain computing utilities. *Journal of Network and Systems Management*, 10(2):171–194, 2002.

[2] A. T. Bouloutas, S. Calo, and A. Finkel. Alarm correlation and fault identification in communication networks. *IEEE Transactions on Communications*, 42(2/3/4):523–533, 1994.

[3] J. D. Case, K. McCloghrie, M. T. Rose, and S. Waldbusser. *Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2)*. IETF Network Working Group, 1996. RFC 1905.

[4] C. S. Chao, D. L. Yang, and A. C. Liu. An automated fault diagnosis system using hierarchical reasoning and alarm correlation. *Journal of Network and Systems Management*, 9(2):183–202, 2001.

[5] R. H. Deng, A. A. Lazar, and W. Wang. A probabilistic approach to fault diagnosis in linear lightwave networks. In *Integrated Network Management III*, pp. 697–708. Apr. 1993.

[6] A. Dupuy, J. Schwartz, Y. Yemini, G. Barzilai, and A. Cahana. Network fault management: A user's view. In *Integrated Network Management I*, pp. 101–107. May 1989.

[7] M. Fecko and M. Steinder. Combinatorial designs in multiple faults localization for battlefield networks. In *IEEE Military Commun. Conf. (MILCOM)*, McLean, VA, 2001.

[8] P. Hong and P. Sen. Incorporating non-deterministic reasoning in managing heterogeneous network faults. In *Integrated Network Management II*, pp. 481–492. Apr. 1991.

[9] G. Jakobson and M. D. Weissman. Alarm correlation. *IEEE Network*, 7(6):52–59, Nov. 1993.

[10] I. Katzela and M. Schwartz. Schemes for fault identification in communication networks. *IEEE Transactions on Networking*, 3(6):733–764, 1995.

[11] S. Kliger, S. Yemini, Y. Yemini, D. Ohsie, and S. Stolfo. A coding approach to event correlation. In *Integrated Network Management IV*, pp. 266–277. May 1995.

[12] R. Perlman. *Interconnections, Second Edition: Bridges, Routers, Switches, and Internetworking Protocols*. Addison Wesley, 1999.

[13] M. Steinder and A. S. Sethi. Non-deterministic diagnosis of end-to-end service failures in a multi-layer communication system. In *Proc. of ICCCN*, pp. 374–379, Scottsdale, AZ, 2001.

[14] M. Steinder and A. S. Sethi. Distributed fault localization in hierarchically routed networks. In *Int'l Wksp on Distributed Systems: Operations and Management*, pp. 195–207, Montreal, QC, Oct. 2002.

[15] M. Steinder and A. S. Sethi. Increasing robustness of fault localization through analysis of lost, spurious, and positive symptoms. In *Proc. of IEEE INFOCOM*, New York, NY, 2002.

[16] M. Steinder and A.S. Sethi. End-to-end service failure diagnosis using belief networks. In *Proc. Network Operation and Management Symposium*, pp. 375–390, Florence, Italy, Apr. 2002.

[17] S. A. Yemini, S. Kliger, E. Mozes, Y. Yemini, and D. Ohsie. High speed and robust event correlation. *IEEE Communications Magazine*, 34(5):82–90, 1996.

[2] The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied of the Army Research Lab or the U.S. Government.