Available online at www.sciencedirect.com
SCIENCE DIRECT®
Computer Networks 45 (2004) 537–562
COMPUTER NETWORKS
www.elsevier.com/locate/comnet

Probabilistic fault diagnosis in communication systems through incremental hypothesis updating ☆

M. Steinder [a,*], A.S. Sethi [b]

[a] IBM T.J. Watson Research Center, 19 Skyline Dr. Hawthorne, NY 10532, USA
[b] Computer and Information Sciences, University of Delaware, 102 Smith Hall, Newark, DE 19716, USA

Received 25 February 2003; received in revised form 2 November 2003; accepted 30 January 2004
Available online 7 April 2004

Responsible Editor: R. Stadler

Abstract

This paper presents a probabilistic event-driven fault localization technique, which uses a probabilistic symptom-fault map as a fault propagation model. The technique isolates the most probable set of faults through incremental updating of a 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 offers close-to-optimal accuracy and is resilient both to noise in the symptom data and to inaccuracies of the probabilistic fault propagation model.
© 2004 Elsevier B.V. All rights reserved.

Keywords: Fault localization; Probabilistic reasoning; Event correlation

1. Introduction

Fault diagnosis is a central aspect of network fault management. The core of fault diagnosis is fault localization [1–3]—a process of analyzing external symptoms of network disorder to isolate possibly unobservable faults responsible for the symptoms' occurrences. Until recently, fault localization concentrated mostly on diagnosing faults related to network connectivity in lower layers of the protocol stack (typically the physical and data-link layers), and its major goal was to isolate faults related to the availability of resources, such as broken cable, inactive interface, etc. Recent advances in the deployment of

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

* Corresponding author. Tel.: +1-914-7847014.
E-mail addresses: steinder@us.ibm.com (M. Steinder), sethi@cis.udel.edu (A.S. Sethi).

1389-1286/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.comnet.2004.01.007

enterprise services such as e-commerce, telecommuting, virtual private networks [4], application service provisioning [5], grid services [6], and Web services [7,8] require that fault localization also focus on diagnosing performance problems in multiple layers of the protocol stack including the application and service layers. Modern enterprise environments impose several challenges on the fault localization problem, which include modeling and reasoning about (1) the system state in various protocol layers, (2) interactions between protocol layers, (3) versatile types of failures, and (4) non-determinism within the system structure and its observed state.

Most fault management systems rely on an explicit fault propagation model (FPM) representing either causal relationships among events [3,9,10] or dependencies among communication system entities [2,11–13]. An event is an exceptional condition occurring in the operation of the hardware or software of the managed network. An event that may not be explained by any other event is considered a root cause or a fault. Most events are not observable to the management system. Events that are observable are called symptoms. A fault localization technique uses a given FPM to identify a set of faults that constitutes the best explanation of observed symptoms. In the management of modern communication systems, a fault localization technique should:

- Allow reasoning under uncertainty about the system model and its state [2,14–17], which is necessary to diagnose Byzantine problems, as their consequent observable failures are not guaranteed to occur or, when they occur, may not be severe enough to be detected by the management system. A non-deterministic model is also needed when causal relationships among system events cannot be learned with certainty, for example, if they change dynamically, or when information about these dependencies provided to the management system is not guaranteed to be accurate.
- Be able to isolate multiple simultaneous faults even if their symptoms overlap [2,15]. The single-fault assumption used by some fault localization techniques limits their scalability, since in large systems the probability that two or more faults exist at the same time may not be neglected.
- Be event-driven as opposed to window-based. Window-based techniques work with a set of symptoms observed over a specified period of time, which are analyzed together to propose their explanation. These techniques are inflexible, as they do not allow different time-windows in the analysis of different problems [18]. They may also be inaccurate by excluding some symptoms or including to many symptoms, when the time-window is set incorrectly. In contrast, event-driven techniques maintain a state which encodes partial fault-localization results computed based on previous symptoms' analysis. Symptoms are analyzed when they arrive independently of other symptoms. Results of their analysis are included in the fault-localization state. Thus, event-driven techniques are not prone to inaccuracies resulting from an incorrect time-window specification. In addition, they allow fault localization to be interleaved with testing, since, at any time, partial fault localization results may be used to find a set of tests with the highest information content given a current knowledge of the system state. (This problem is studied in [19,20].)
- Be resilient to lost and spurious symptoms [3,14,16], which may dramatically reduce fault localization accuracy if their presence is not taken into account by a fault localization algorithm.
- Have a high accuracy and a low-polynomial computational complexity.

This paper presents a probabilistic event-driven fault localization technique, which uses a probabilistic symptom-fault map [2,3,9] as an FPM. 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. It has to be mentioned that 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 and involve chains of unobservable events). In our previous work, we applied belief networks to fault localization based on such complex fault propagation models [21]. However, like many other fault

localization techniques proposed in the literature [2,3,9], this paper uses a bipartite FPM. The focus on this type of a model is justified by the following arguments:

- Performing fault localization with more complex representations is difficult. (In general, the problem is NP-hard [2].) To avoid this complexity, more detailed models are frequently reduced to bipartite ones through a sequence of graph reduction operations [3]. Constraining an FPM to a bipartite graph, allows us to develop a fault localization algorithm whose computational complexity is an order of magnitude lower than that of a more general algorithm proposed in [21].
- 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 [9].
- Some fault localization problems may be accurately represented by bipartite symptom-fault maps (e.g., the problem of end-to-end service failure diagnosis [21,22]), thereby necessitating fault localization algorithms suitable for bipartite FPMs.

The technique proposed in this paper is able to accurately isolate multiple simultaneous faults with overlapping sets of symptoms in the presence of observation noise. By using event-driven symptom processing, the technique is free from the limitations of window-based approaches. In addition to providing these features, the 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 symptoms without re-analyzing them. Thanks to this feature, the algorithm continuously provides the 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 [2,3]. The technique proposed here produces a set of alternative hypotheses rather than just a single explanation. These hypotheses are ranked according to the 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 the incorrect hypothesis with its alternative without repeating the entire fault localization process improves the robustness of the fault management system.

This paper is structured as follows. We first present basic concepts used in this paper and a combinatorial approach to fault localization, which is frequently used as an optimal technique for bipartite FPMs (Section 2). Then, we discuss the basic ideas behind the incremental approach (Section 3). These ideas are later refined to incorporate reasoning with positive and lost symptoms (Section 4) and to make the technique resilient against spurious symptoms (Section 5). The technique is evaluated using the problem of end-to-end service failure diagnosis as a case study (Section 6). We also discuss extensions to the algorithm that are necessary with other than noisy-OR canonical models of probability distribution (Section 7). Finally, we compare the incremental algorithm to other fault localization techniques proposed in the literature that are suitable for bipartite FPMs (Section 8).

2. Basic concepts

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 use $\mathscr{F}$ and $\mathscr{S}$ to denote the sets of all possible faults and symptoms, respectively. In a non-deterministic model, with every fault $f_i \in \mathscr{F}$ a probability of its independent failure is associated, which is denoted by $p(f_i)$. The edge between $f_i \in \mathscr{F}$ and $s_j \in \mathscr{S}$ indicates that f_i may cause s_j. The edge is weighted with the probability of the

causal implication, $p(s_j|f_i)$. Following previous modeling approaches [2,17] and their justification introduced in [23], we assume a noisy-OR model of probability distribution in which alternative causes of a symptom are combined using the logical operator *OR*. A subset of symptoms observed by a management application is denoted by $\mathscr{S}_O = \mathscr{S}_N \cup \mathscr{S}_P$, where $\mathscr{S}_N$ and $\mathscr{S}_P$ are the sets of all observed negative and positive symptoms, respectively. When positive symptoms are ignored, $\mathscr{S}_O = \mathscr{S}_N$ and the purpose of fault localization is to find $\mathscr{F}_d \subseteq \mathscr{F}$ that maximizes the probability that (1) all faults in $\mathscr{F}_d$ occur and (2) each symptom in $\mathscr{S}_O$ is explained by at least one fault from $\mathscr{F}_d$.

When a fault propagation model is represented by a bipartite probabilistic graph, exact fault localization may be performed with the combinatorial algorithm [24], which assumes a naive approach by evaluating all possible combinations of faults with regard to their ability to explain all observed symptoms. When two or more combinations of faults are able to explain all observed symptoms, the best combination is chosen. For a given combination of faults $\mathscr{F}_i$ and a set of observed symptoms $\mathscr{S}_O$, the measure of goodness $g(\mathscr{F}_i, \mathscr{S}_O)$ is defined as follows:

$$\begin{aligned} g(\mathscr{F}_i, \mathscr{S}_O) &= P\{\text{all faults in } \mathscr{F}_i \text{ occurred}\} \cdot P\{\text{each symptom in } \mathscr{S}_O \text{ is caused by at least one fault in } \mathscr{F}_i\} \\ &= \left(\prod_{f \in \mathscr{F}_i} p(f) \right) \prod_{s \in \mathscr{S}_O} \left(1 - \prod_{f \in \mathscr{F}_i} (1 - p(s|f)) \right). \end{aligned} \tag{1}$$

Note that in the calculation of $g(\mathscr{F}_i, \mathscr{S}_O)$ we assume that faults are independent. As a result, $P\{f_{i_1} \wedge f_{i_2} \wedge \ldots \wedge f_{i_k}\} = \prod_{f_{i_j} \in \mathscr{F}_i} p(f_{i_j})$, where $f_{i_1}, f_{i_2}, \ldots, f_{i_k} \in \mathscr{F}_i$. If this assumption is invalid, the calculation of $g(\mathscr{F}_i, \mathscr{S}_O)$ has to be modified by setting $P\{\text{all faults in } \mathscr{F}_i \text{ occurred}\} = \prod_{F_{i_j} \subseteq \mathscr{F}_i} P\{\text{all faults in } F_{i_j} \text{ occurred}\}$, where all $F_{i_j} \subseteq \mathscr{F}_i$ are disjoint sets of mutually dependent faults such that no dependencies among faults in different sets exist. For each such F_{i_j} the value of $P\{\text{all faults in } F_{i_j} \text{ occurred}\}$ must be explicitly given, or the FPM must contain probability-weighted dependency edges among faults in F_{i_j}. In the latter case, $P\{\text{all faults in } F_{i_j} \text{ occurred}\}$ may be obtained using techniques proposed in [2]. This approach can be also used to deal with dependent faults in the incremental technique proposed in this paper. For simplicity, however, we will present this algorithm assuming that faults are independent.

It is frequently assumed that the number of faults that occur simultaneously is small. This suggests that, in the combinatorial algorithm, we should start evaluating fault combinations from those that contain the fewest faults and terminate the search as soon as an explanation of all symptoms is known. This leads to the following combinatorial algorithm.

Algorithm 1 (*Combinatorial Algorithm*)
for $i = 1$ *until* $i < |\mathscr{F}|$ *do*
  for all i-fault combinations from $\mathscr{F}$, $\mathscr{F}_i$ *compute* $g(\mathscr{F}_i, \mathscr{S}_O)$
  if at least one $\mathscr{F}_i$ *is found such that* $g(\mathscr{F}_i, \mathscr{S}_O) > 0$
    return $\mathscr{F}_i$ *such that* $g(\mathscr{F}_i, \mathscr{S}_O)$ *is maximum*

It may be easily calculated that Algorithm 1 performs $\sum_{i=1}^{|\mathscr{F}|} \binom{|\mathscr{F}|}{i} \cdot i \cdot |\mathscr{S}_O| = \mathscr{O}(2^{|\mathscr{F}|})$ operations. However, when multiple concurrent faults are unlikely, the algorithm's practical complexity may be polynomial. In this paper, we use the combinatorial algorithm as a reference algorithm against which the incremental algorithm is compared.

3. Incremental hypothesis updating

In this section, a novel fault localization technique is introduced, called Incremental Hypothesis Updating (IHU), which creates a set of the most likely hypotheses explaining the set of observed symptoms. Rather than wait for a period of time before presenting a solution, the technique makes all these hypotheses

available on a continuous basis, and incrementally upgrades them with information learned from arriving symptoms. We first focus on the basic version of the incremental algorithm, which ignores positive, lost, and spurious symptoms.

The incremental algorithm creates a set of hypotheses, each of which is a subset of $\mathscr{F}$ that explains all symptoms in $\mathscr{S}_O$. We say that hypothesis $h_j \subseteq \mathscr{F}$ explains symptom $s_i \in \mathscr{S}_O$ if it contains at least one fault that explains s_i. The set of hypotheses does not include all subsets of $\mathscr{F}$ that explain symptoms in $\mathscr{S}_O$. Clearly, in the worst case, as many as $2^{|\mathscr{F}|}$ such subsets may exist. The incremental algorithm avoids this complexity by deliberately excluding most of these subsets based on the properties of the problem it tries to solve. To determine which subsets of $\mathscr{F}$ are included in the set of hypotheses, the incremental algorithm uses size-limiting heuristics, which are described in this section.

The hypotheses are ranked using a belief metric, b, which expresses the confidence associated with a given hypotheses relative to other hypotheses. The belief metric should not be interpreted as the conditional probability that all faults in a given hypotheses exist given symptoms in $\mathscr{S}_O$ have been observed. Such interpretation would be incorrect, as the set of hypotheses does not represent the universe of *all possible* explanations. The belief metric only encodes the relative importance of a given hypotheses in the space of all *considered* explanations. Therefore, a value of the belief metric could be any positive real number. Nevertheless, it is convenient to normalize belief metrics such that the sum of belief metrics associated with all considered hypotheses is equal to 1.

The algorithm proceeds in an event-driven and incremental fashion. The execution triggered by an observation of the ith symptom, s_i, creates a set of hypotheses, $\mathscr{H}_i$, each explaining symptoms s_1 through s_i. Set $\mathscr{H}_i$ is created by updating $\mathscr{H}_{i-1}$ with an explanation of symptom s_i. We define H_{s_i} as a set $\{f_k \in \mathscr{F}\}$ 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 [2], H_{s_i} is the domain of symptom s_i.

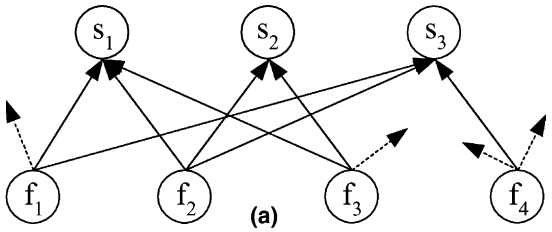
In the ith iteration of fault localization, the belief metric $b_i(h_j)$ is expressed using the probability that (1) all faults belonging to $h_j \in \mathscr{H}_i$ have occurred, and (2) h_j explains each observed symptom $s_k \in \mathscr{S}_{O,i} = \{s_1, \ldots, s_i\}$. Note, that $b_i(h_j) = \beta g(h_j, \mathscr{S}_{O,i})$ (Eq. (1)), where β is a normalization constant, and formally it is defined as follows:

$$b_i(h_j) = \beta\left(\prod_{f_k \in h_j} p(f_k)\right) \prod_{s_l \in \mathscr{S}_{O,i}} \left(1 - \prod_{f_k \in h_j} (1 - p(s_l \mid f_k))\right). \tag{2}$$

To incorporate the explanation of symptom s_i into a set of fault hypotheses, in the ith iteration of the algorithm, we analyze each $h_j \in \mathscr{H}_{i-1}$. If h_j is able to explain symptom s_i, we put h_j into $\mathscr{H}_i$. Otherwise, h_j has to be extended by adding to it a fault from H_{s_i}. In a greedy approach, a new hypothesis may be created for every fault from H_{s_i}. This approach would result in a very fast growth in the size of $\mathscr{H}_i$, and, consequently, would make the computational complexity of the algorithm unacceptable. Instead, we adopt the following heuristic. Fault $f_l \in H_{s_i}$ may be added to $h_j \in \mathscr{H}_{i-1}$ only if the size of h_j, $|h_j|$, is smaller than $\mu(f_l)$. Function $\mu(f_l)$ is defined as the minimal size of a hypothesis in $\mathscr{H}_{i-1}$ that contains f_l and explains symptom s_i. The usage of this heuristic is derived from the assumption, which is valid in most fault localization problems, that a probability of multiple simultaneous faults is smaller than a probability of any single fault. Therefore, of any two hypotheses containing f_l, the hypothesis that contains the fewest faults is the one most likely to constitute the optimal symptom explanation. Since it is not practical to keep all possible hypotheses, we remove those that are bigger in size.

We illustrate this heuristic with the following example.

Example 1. The fault model in Fig. 1(a) presents causal relationships between faults f_1, f_2, f_3, and f_4 and symptoms s_1, s_2, and s_3. Suppose the symptoms are observed in order s_1, s_3, and s_2. Initially, the only available hypothesis is $\emptyset$, which indicates that, given no symptom observations, we should conclude that no



$$s_1 : H_{s_1} = \{f_1, f_2, f_3\} \rightarrow \mathcal{H}_1 = \{\{f_1\}, \{f_2\}, \{f_3\}\}$$
$$s_3 : H_{s_3} = \{f_1, f_2, f_4\} \rightarrow \mathcal{H}_2 = \{\{f_1\}, \{f_2\}, \{f_3, f_4\}\}$$
$$s_2 : H_{s_2} = \{f_2, f_3\} \rightarrow \mathcal{H}_3 = \{\{f_1, f_3\}, \{f_2\}, \{f_3, f_4\}\}$$
(b)

Fig. 1. Example of incremental hypothesis updating: (a) example of causality graph and (b) sets of hypotheses created after observing a sequence of symptoms s_1, s_3 and s_2.

faults occurred. Then, symptom s_1 arrives, whose domain is $H_{s_1} = \{f_1, f_2, f_3\}$. As a result of extending $\emptyset$, we obtain $\mathscr{H}_1 = \{\{f_1\}, \{f_2\}, \{f_3\}\}$. The domain of symptom s_3 is $H_{s_3} = \{f_1, f_2, f_4\}$. Since f_1 and f_2 belong to hypotheses $\{f_1\}$ and $\{f_2\}$ in $\mathscr{H}_1$, respectively, hypotheses $\{f_1\}$ and $\{f_2\}$ explain s_3 and therefore they are placed in $\mathscr{H}_2$. Then, both $\mu(f_1)$ and $\mu(f_2)$ are set to 1. Hypothesis $\{f_3\}$ does not explain s_3; therefore, it has to be extended with faults in H_{s_3}. Out of faults in H_{s_3} we cannot use f_1 and f_2 since their $\mu(\cdot)$s $\leqslant |\{f_3\}| = 1$. The only extension possible is $\{f_3, f_4\}$. This way, we have created $\mathscr{H}_2 = \{\{f_1\}, \{f_2\}, \{f_3, f_4\}\}$. In the next iteration, after symptom s_2 has been observed, we are allowed to extend $\{f_1\} \in \mathscr{H}_2$ by adding fault f_3 since $\mu(f_3) = |\{f_3, f_4\}| = 2$ while $|\{f_1\}| = 1$, but we are not allowed to extend $\{f_1\}$ by adding fault f_2,because $\mu(f_2) = |\{f_1\}| = 1$. Thus the final set of hypotheses is $\mathscr{H}_3 = \{\{f_1, f_3\}, \{f_2\}, \{f_3, f_4\}\}$ (Fig. 1(b)).

An important problem to solve is the efficient computation of $b_i(h_j)$. We observe that $b_i(h_j)$ may be approximated iteratively based on $b_{i-1}(h_j)$ as follows:

1. If $h_j \in \mathscr{H}_{i-1}$ and h_j explains s_i,

$$b_i(h_j) = \beta b_{i-1}(h_j) \left(1 - \prod_{f_l \in h_j \cap H_{s_i}} (1 - p(s_i|f_l))\right). \tag{3}$$

2. Otherwise, if f_l explains s_i,

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

The incremental algorithm is defined by the following pseudo-code.

Algorithm 2 (*Incremental Hypothesis Updating*)
let $\mathscr{H}_0 = \{\emptyset\}$ *and* $b_0(\emptyset) = 1$
for every observed symptom s_i:
  let $\mathscr{H}_i = \emptyset$ *and for all* $f_l \in \mathscr{F}$ *let* $\mu(f_l) = |\mathscr{F}|$
  for all $h_j \in \mathscr{H}_{i-1}$ *do*
    for all $f_l \in h_j$ *such that* $f_l \in H_{s_i}$
      set $\mu(f_l) = \min(\mu(f_l), |h_j|)$
      add h_j *to* $\mathscr{H}_i$ *and calculate* $b_i(h_j)$
  for all $h_j \in \mathscr{H}_{i-1} \setminus \mathscr{H}_i$ *do*

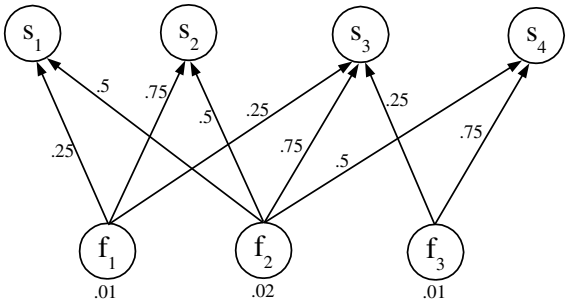


Fig. 2. Belief-network for Example 2.

for all $f_l \in \mathscr{F} \cap H_{s_i}$ *such that* $\mu(f_l) > |h_j|$ *do*
  add $h_j \cup \{f_l\}$ *to* $\mathscr{H}_i$ *and compute* $b_i(h_j \cup \{f_l\})$
choose $h_j \in \mathscr{H}_{|\mathscr{S}_o|}$ *such that* $b_{|\mathscr{S}_o|}(h_j)$ *is maximum*

To calculate the upper bound on the worst case computational complexity, we observe that the calculation of $b_i(h_j)$ is $\mathscr{O}(|h_j \cap H_{s_i}|) = \mathscr{O}(|H_{s_i}|) = \mathscr{O}(|\mathscr{F}|)$. The calculation of $b_i(h_j \cup \{f_l\})$ is $\mathscr{O}(1)$. The algorithm performs $|\mathscr{S}_O|$ iterations. In every iteration, we execute two *for* loops. The first loop requires $\mathscr{O}((\max_i(|\mathscr{H}_i|))|H_{s_i}|)$ steps. The second loop requires $\mathscr{O}(\max_i(|\mathscr{H}_i|)|H_{s_i}| \cdot 1)$ operations. Therefore, the complexity of the entire algorithm is $\mathscr{O}(|\mathscr{S}_O| \max_i(|\mathscr{H}_i|)|\mathscr{F}|)$. To get a precise bound we need to determine a bound for $\max_i(|\mathscr{H}_i|)$. It turns out that in rare cases the size of the hypothesis set may grow exponentially. To avoid this problem we set a limit on the number of hypotheses that may be created in each iteration; the least likely hypotheses are rejected when the limit is exceeded. The price we pay for this modification is that a hypothesis chosen by the algorithm may not be the one that maximizes the measure of goodness. If the limit on the size of the hypothesis set is $\mathscr{O}(|\mathscr{F}|)$, operations involved in controlling the size of $\mathscr{H}_i$ do not increase the theoretical bound on the complexity of the entire algorithm. To obtain experimental results presented in Section 6 the limit $2|\mathscr{F}|$ is used. Thus, the complexity of the entire algorithm is $\mathscr{O}(|\mathscr{S}_O||\mathscr{F}|^2)$.

Example 2. Consider the fault propagation model in Fig. 2. We will illustrate fault localization triggered by the observation of symptoms s_2 and s_4.

The initial set of hypotheses $\mathscr{H}_0$ is equal to $\{\emptyset\}$, and $b_0(\emptyset) = 1$. When symptom s_2 arrives, we create $\mathscr{H}_1 = \{\{f_1\}, \{f_2\}\}$ and calculate $b_1(\{f_1\}) = \beta_1 \cdot 0.01 \cdot 0.75 = \beta_1 \cdot 0.0075$ and $b_1(\{f_2\}) = \beta_1 \cdot 0.02 \cdot 0.5 = \beta_1 \cdot 0.01$. After normalization, $b_1(\{f_1\}) = 0.43$ and $b_1(\{f_2\}) = 0.57$.

The domain of the next observed symptom, s_4, is $H_{s_4} = \{f_2, f_3\}$. Since $\{f_2\}$ explains s_4 we set $\mu(f_2) = 1$ and place $\{f_2\}$ in $\mathscr{H}_2$. Then, we extend $\{f_1\}$ with f_3. We calculate $b_2(\{f_2\}) = \beta_2\beta_1 \cdot 0.01 \cdot 0.5 = \beta_2\beta_1 \cdot 0.005 = 0.99$ and $b_2(\{f_1, f_3\}) = \beta_2\beta_1 \cdot 0.0075 \cdot 0.01 \cdot 0.75 = \beta_2\beta_1 \cdot 0.00005625 = 0.01$. Since hypothesis $\{f_2\}$ is the best according to belief metric b_2, it is chosen as the final answer.

4. Analysis of positive symptoms

Algorithm IHU presented in Section 3 calculates a set of the most probable fault hypotheses based on the observed indications of system disorder. It does not take advantage of the fact that some possible indications of the disorder have not been observed. As many researchers point out [3,24], 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*. The inclusion of positive symptoms into the fault localization process may allow a more accurate fault hypothesis to be isolated [21,25].

To include positive symptoms in the analysis, the belief metric b_i^* associated with hypothesis $h_j \in \mathscr{H}_i$ needs to contain two components: a negative component b_i^{n} and a positive component b_i^{p}, where $b_i^*(h_j) = \beta b_i^{\mathrm{n}}(h_j) b_i^{\mathrm{p}}(h_j)$ and $b_i^{\mathrm{n}}(h_j) = b_i(h_j)$ of Eq. (2). The positive component is defined as the probability that faults in h_j have not generated any of the symptoms in $\mathscr{S} \setminus \mathscr{S}_{\mathrm{O},i}$. This probability is expressed through the following equation:

$$b_i^{\mathrm{p}}(h_j) = \prod_{s_l \in \mathscr{S} \setminus \mathscr{S}_{\mathrm{O},i}} \prod_{f_k \in h_j} (1 - p(s_l \mid f_k)). \tag{5}$$

Multiplier $b_i^{\mathrm{p}}(h_j)$ decreases the value of the belief metric associated with hypothesis h_j if many of the symptoms caused by faults in h_j have not been observed.

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.

4.1. Symptom observability ratio

Frequently, an indication of existing disorder may not be observed by the management system because its configuration excludes some system 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 $b_i^{\mathrm{p}}(h_j)$ 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 FPM. 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 a current configuration, the symptom is observable or not observable, respectively. We will denote by $\mathscr{S}_{\mathrm{O}} \subseteq \mathscr{S}$ 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 Eq. (5) is calculated over $s_l \in \mathscr{S}_{\mathrm{O}} \setminus \mathscr{S}_{\mathrm{O},i}$ rather than $s_l \in \mathscr{S} \setminus \mathscr{S}_{\mathrm{O},i}$.

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 $OR = |\mathscr{S}_{\mathrm{O}}|/|\mathscr{S}|$ [21]. 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.

4.2. 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 [26], 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, taking positive symptoms into account necessitates the analysis of lost symptoms as well.

Let us denote by $p_{\mathrm{loss}}(s_i)$ the probability that symptom $s_i \in \mathscr{S}$ is lost. The value of $p_{\mathrm{loss}}(s_i)$ 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 in the fault localization algorithm by modifying the definition of $b_i^{\mathrm{p}}(h_j)$ (Eq. (5)) as follows:

$$b_i^{\mathrm{p}}(h_j) = \prod_{s_l \in \mathscr{S}_{\mathrm{O}} \setminus \mathscr{S}_{\mathrm{O},i}} \left(p_{\mathrm{loss}}(s_l) + (1 - p_{\mathrm{loss}}(s_l)) \prod_{f_k \in h_j} (1 - p(s_l \,|\, f_k)) \right). \tag{6}$$

4.3. Incremental calculation of 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 $b_i^{\mathrm{n}}(\emptyset) = b_i^{\mathrm{p}}(\emptyset) = 1$. In the process of analyzing new symptoms, the value of belief metric $b_i^*(h_j)$ is calculated by multiplying $b_i^{\mathrm{n}}(h_j)$ and $b_i^{\mathrm{p}}(h_j)$, where $b_i^{\mathrm{n}}(h_j)$ is computed incrementally using Eqs. (3), (4). We obtain $b_i^{\mathrm{p}}(h_j)$ as follows:

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

$$b_i^{\mathrm{p}}(h_j) = \frac{b_{i-1}^{\mathrm{p}}(h_j)}{\prod_{f_k \in h_j} (p_{\mathrm{loss}}(s_l) + (1 - p_{\mathrm{loss}}(s_l))(1 - p(s_l \,|\, f_k)))}. \tag{7}$$

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

$$b_i^{\mathrm{p}}(h_j \cup \{f_l\}) = b_{i-1}^{\mathrm{p}}(h_j)\, b_i^{\mathrm{p}}(\{f_l\}). \tag{8}$$

Eq. (7) is derived from Eq. (6) by moving the second product in front of the parentheses. By doing this we make an assumption that a symptom may be caused by only one fault at a time. When the symptom is triggered by two or more faults in h_j simultaneously, we miscalculate $b_i^{\mathrm{p}}(h_j)$ by counting p_{loss} twice. In practice, this second case is less likely, and therefore the approximation is reasonable.

In Eq. (8), $b_i^{\mathrm{p}}(\{f_l\})$ 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 $b_i^{\mathrm{p}}(\{f_l\})$ are pre-computed when the model is initialized. After every symptom observation, $b_i^{\mathrm{p}}(\{f_l\})$ is incrementally updated using Eq. (7).

5. Dealing with spurious observations

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 [21]. As a result, frequently manual effort has to be unnecessarily undertaken to test and reject these false-positive propositions.

To deal with spurious symptoms Algorithm 2 has to be modified as follows. Let s_i be the ith observed symptom and let $p_{\mathrm{spurious}}(s_i)$ denote the probability that symptom s_i is spuriously generated. While deciding whether hypothesis $h_j \in \mathscr{H}_{i-1}$ should be placed in $\mathscr{H}_i$ 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 $\mathscr{H}_i$ 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 $\mathscr{H}_i$ without extension. Since the first and second cases occur with probability $1 - p_s(s_i)$ and $p_s(s_i)$, respectively, these values are used as multipliers embedded in the calculation of the corresponding values of the belief metric. Recall from Section 3 that the original algorithm does not allow adding $h_j \in \mathscr{H}_{i-1}$ to $\mathscr{H}_i$ 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., $b_i^{+p}(h_j) = b_i^p(h_j)$. The modified negative component, $b_i^{+n}(h_j)$ is calculated iteratively as follows:

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

$$b_i^{+n}(h_j) = b_{i-1}^{+n}(h_j)\left((1 - p_s(s_i))\left(1 - \prod_{f_l \in h_j \cap H_{s_i}} (1 - p(s_i|f_l))\right) + p_s(s_i)\right). \tag{9}$$

2. Otherwise

$$b_i^{+n}(h_j) = b_{i-1}^{+n}(h_j)p_s(s_i). \tag{10}$$

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

$$b_i^{+n}(h_j \cup \{f_l\}) = b_{i-1}^{+n}(h_j)p(f_l)p(s_i|f_l)(1 - p_s(s_i)). \tag{11}$$

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 2A (*IHU+*(*OR*,p_{loss},p_s))
let $\mathscr{H}_0 = \{\emptyset\}$, $b_0^{+n}(\emptyset) = b_0^{+p}(\emptyset) = 1$, $\alpha(\emptyset) = 0$
for every observed symptom s_i:
  let $\mathscr{H}_i = \emptyset$, *and for all* $f_l \in \mathscr{F}$ *let* $\mu(f_l) = |\mathscr{F}| + |\mathscr{S}_O|$
  for all $h_j \in \mathscr{H}_{i-1}$ *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* $\mathscr{H}_i$ *and calculate* $b_i^+(h_j)$
  for all $h_j \in \mathscr{H}_{i-1} \setminus \mathscr{H}_i$ *do*
    if $(p_s(s_i) > 0)$
      add h_j *to* $\mathscr{H}_i$, *calculate* $b_i^+(h_j)$, *and set* $\alpha(h_j) = \alpha(h_j) + 1$
    for all $f_l \in \mathscr{F} \cap H_{s_i}$ *such that* $\mu(f_l) > \alpha(h_j)$ *do*
      add $h_j \cup \{f_l\}$ *to* $\mathscr{H}_i$, *compute* $b_i^+(h_j \cup \{f_l\})$, *and set* $\alpha(h_j) = \alpha(h_j) + 1$
choose $h_j \in \mathscr{H}_{|\mathscr{S}_N|}$ *with maximum* $b_{|\mathscr{S}_N|}^+(h_j)$

While Algorithm 2A (IHU+) looks similar to Algorithm 2 (IHU) presented in Section 3, there are two significant differences between them. Recall that Algorithm 2 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 $h_j \in \mathscr{H}_i$ 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 2 limits the maximum size of the set of hypotheses to $k \in \mathcal{O}(|\mathscr{F}|)$ and removes the least probable hypotheses if this limit is exceeded. These two heuristics are modified in Algorithm 2A as described in the following sections.

5.1. Calculating hypothesis size

In Algorithm 2, function $\mu(f_l)$ is defined as the minimum size of $h_k \in \mathscr{H}_{i-1}$ that contains f_l and explains symptom s_i, where the size of h_k is $|h_k|$. In Algorithm 2A, the size of hypothesis h_k, $\alpha(h_k)$ is defined as the number of faults in h_k plus the number of symptoms observed so far that h_k considers spurious. This modification serves two purposes. It

1. Helps avoid duplicate hypotheses.
   Duplicate hypotheses introduce redundancy into the set of hypotheses, which may affect the accuracy of the technique. Since the maximum size of the set of hypotheses is limited, avoiding redundancy may allow us to keep a least likely hypothesis that may later turn out to be the optimal one, which would otherwise have to be removed. Although it is possible to remove duplicate hypotheses within the computational complexity bound of Algorithm 2A (duplicate hypotheses may be unified and their belief metrics may be added to one another), the necessity to do so renders the implementation of the algorithm more difficult. It is thus better to avoid creating duplicate hypotheses at all.
2. 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.

To explain the reasons behind this modification it is useful to consider the following example.

Example 3. Consider the FPM in Fig. 3. Assume that $p_s(s_i) = 0.001$ for $i = 1, 2, 3$. For the sake of simplicity, we also ignore positive symptoms. Consider a scenario, in which all three symptoms are observed in order s_2, s_3, and s_1. Let us first present how this fault scenario could be solved with our reference combinatorial Algorithm 1, extended to include spurious symptom probability in the calculation of function g as follows:

$$g(\mathscr{F}_i, \mathscr{S}_O) = \prod_{f_k \in \mathscr{F}_i} p(f_k) \prod_{s_l \in \mathscr{S}_O} \Pr\{s_l \text{ is spurious or caused by at least one } f \in \mathscr{F}_i\}$$

$$= \prod_{f_k \in \mathscr{F}_i} p(f_k) \prod_{s_l \in \mathscr{S}_O} \left(p_s(s_l) + (1 - p_s(s_l)) \left(1 - \prod_{f_k \in \mathscr{F}_i} (1 - p(s_l | f_k)) \right) \right).$$

The combinatorial algorithm enumerates all four possible combinations of faults from $\{f_1, f_2\}$, i.e., $\emptyset$, $\{f_1\}$, $\{f_2\}$, and $\{f_1, f_2\}$, as possible solutions to the scenario. Clearly, with a proper choice of how many and which symptoms to consider spurious, all four combinations may constitute a valid solution to the scenario. Thus, the best solution has to be chosen based on the value of the measure of goodness g. Using the modified definition of g one can show that combination $\{f_1, f_2\}$ is the optimal solution to the scenario.

Let us solve this scenario incrementally with Algorithm 2A using set cardinality rather than α as a hypothesis size (see left-hand side of Table 1). Initially, $\mathscr{H}_0 = \{\emptyset\}$, $b_0(\emptyset) = 1$, and $\mu(\emptyset) = 0$. The observation of symptom s_2 results in two extensions of hypothesis $\emptyset$, $\{f_1\}$ and $\{f_2\}$. Treating s_2 as spurious allows us to put hypothesis $\emptyset$ in $\mathscr{H}_1$. Only one hypothesis in $\mathscr{H}_1$, $\{f_2\}$, explains the next observed symptom, s_3. Other hypotheses in $\mathscr{H}_1$, $\{f_1\}$ and $\emptyset$, have to be extended with f_2 or their belief metric has to be modified to account for the possibility that s_3 is spurious. In the case of hypothesis $\{f_1\}$, only the second option is available, since $\mu(f_l) \leqslant |\{f_1\}|$. However, for hypothesis $\emptyset$, both options are available. When $\emptyset$ is extended with f_2, a duplicate hypothesis $\{f_2\}$ is created.

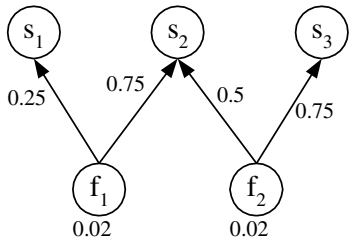


Fig. 3. FPM for Example 3.

Table 1
Solving scenario $\{s_2, s_3, s_1\}$ in Fig. 3 with Algorithm 2A using set cardinality and $\alpha(h_j)$ to calculate a hypothesis size

		Solution using $\lvert h_j\rvert$			Solution using $\alpha(h_j)$		
		h_j	$b_i(h_j)$	$\lvert h_j\rvert$	h_j	$b_i(h_j)$	$\alpha(h_j)$
	$\mathscr{H}_0$	$\emptyset$	1	0	$\emptyset$	1	0
s_2	$\mathscr{H}_1$	$\{f_1\}$	0.15×10^{-1}	1	$\{f_1\}$	0.15×10^{-1}	1
		$\{f_2\}$	0.1×10^{-1}	1	$\{f_2\}$	0.1×10^{-1}	1
		$\emptyset$	0.1×10^{-2}	0	$\emptyset$	0.1×10^{-2}	1
s_3	$\mathscr{H}_2$	$\{\mathbf{f_2}\}$	0.75×10^{-2}	**1**	$\{\mathbf{f_2}\}$	0.75×10^{-2}	**1**
		$\{f_2\}$	0.15×10^{-4}	1	$\{f_1\}$	0.15×10^{-4}	2
		$\{f_1\}$	0.15×10^{-4}	1	$\emptyset$	0.1×10^{-5}	2
		$\emptyset$	0.1×10^{-5}	0			
	Duplicates	$\{f_2\}$	0.77×10^{-2}	1			
	removed	$\{f_1\}$	0.15×10^{-4}	1			
		$\emptyset$	0.1×10^{-5}	0			
s_1	$\mathscr{H}_3$	$\{\mathbf{f_1}\}$	0.38×10^{-5}	**1**	$\{\mathbf{f_1}\}$	0.38×10^{-5}	**2**
		$\{f_1\}$	0.5×10^{-8}	1	$\{f_1, f_2\}$	0.38×10^{-4}	2
		$\{f_2\}$	0.75×10^{-5}	1	$\{f_2\}$	0.75×10^{-5}	2
		$\emptyset$	0.1×10^{-8}	0	$\emptyset$	0.1×10^{-8}	3
	Duplicates	$\{f_1\}$	0.38×10^{-5}	1			
	removed	$\{f_2\}$	0.75×10^{-5}	1			
		$\emptyset$	0.1×10^{-8}	0			

When symptom s_1 is analyzed, $\{f_2\}$ may not be extended with f_1 since $|\{f_2\}| = \mu(f_1) = |\{f_1\}| = 1$. As a result, the algorithm chooses $\{f_1\}$ as the best explanation of the observed symptoms. Recall that the optimal algorithm chose hypothesis $\{f_1, f_2\}$. Algorithm 2A was not even able to consider this hypothesis, because it was prevented from creating $\{f_1, f_2\}$ by the heuristic using the number of faults in a hypothesis as its size.

Let us now consider the process of analyzing symptoms s_2, s_3, and s_1 using Algorithm 2A with the modified definition of hypothesis size. This analysis is shown on the right-hand side of Table 1. The first difference in the created set of hypotheses is observed after analyzing symptom s_3; no duplicate hypotheses are created. When symptom s_1 is analyzed, hypothesis $\{f_2\}$, whose size $\alpha(\{f_2\}) = 1$, can be extended with f_1 since $\mu(f_1) = \alpha(\{f_1\}) = 2 > 1$. As a result, hypothesis $\{f_1, f_2\}$ is created, which turns out to be the best according to belief metric b_3. Thus, the modified algorithm gives the optimal answer.

5.2. Controlling hypotheses number

The second heuristic applied by Algorithm 2 limits the maximum size of the set of hypotheses to $k \in \mathcal{O}(|\mathscr{F}|)$. To add a new hypothesis to $\mathscr{H}_i$, when $|\mathscr{H}_i| = k$, a hypothesis h_l for which $b_i(h_l)$ is minimal must be first removed from $\mathscr{H}_i$. 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 an 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, it may usually be ignored. A hypothesis removal due to the big size of $\mathscr{H}_i$ 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 $\mathscr{H}_i$ 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 $b^{+\mathrm{p}}(h_j)$ is calculated as if the current set of observed symptoms was the final one.

Illustrating the problem of premature hypothesis removal is difficult as the problem becomes pronounced only in FPMs of considerable size. Nevertheless, we will consider the following rather trivial scenario.

Example 4. Consider the FPM in Fig. 4. Assume that $p_{\mathrm{s}}(s_i) = 0.001$ for $i = 1, 2, 3$. We assume that that all symptoms are observable and that the maximum size of the set of hypotheses is $|\mathscr{F}| = 2$. By performing a calculation similar to the one in Example 3, one can show that the optimal solution to scenario involving symptoms s_2 and s_3 is hypothesis $\{f_2\}$.

Let us solve this scenario incrementally with Algorithm 2A without modification to the second heuristic, i.e., the belief metric is used to choose a removed hypothesis. The process of solving the scenario is shown in Table 2. Hypotheses that are not removed are marked in bold typeface. Observe, that in the first iteration, hypothesis $\{f_2\}$ is removed, as its belief metric is the lowest, which is a consequence of the low value of the positive belief metric component. Although hypothesis $\{f_2\}$ is re-created in the second iteration, at this stage, its belief metric is lower that that of $\emptyset$, and therefore $\emptyset$ is chosen as the final answer. One can easily calculate that, had not hypothesis $\{f_2\}$ been removed in the first iteration, it would have become the best choice in the second iteration. Thus, hypothesis $\{f_2\}$ was removed prematurely.

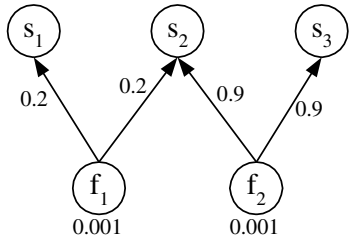


Fig. 4. FPM for Example 4.

Table 2
Solving scenario $\{s_2, s_3\}$ in Fig. 4 with Algorithm 2A with the original size-limiting heuristic

		h_j	$b_i^{+\mathrm{n}}(h_j)$	$b_i^{+\mathrm{p}}(h_j)$	$b_i^{+}(h_j)$
	$\mathscr{H}_0$	$\emptyset$	1	1	1
s_2	$\mathscr{H}_1$	$\{\mathbf{f_1}\}$	0.2×10^{-3}	0.8	0.16×10^{-3}
		$\{f_2\}$	0.9×10^{-3}	0.1	0.09×10^{-3}
		$\emptyset$	0.1×10^{-2}	1	0.1×10^{-2}
s_3	$\mathscr{H}_2$	$\{f_1\}$	0.2×10^{-6}	0.8	0.16×10^{-6}
		$\{f_1, f_2\}$	0.18×10^{-6}	0.8	0.14×10^{-6}
		$\{\mathbf{f_2}\}$	0.09×10^{-6}	1	0.09×10^{-6}
		$\emptyset$	0.1×10^{-5}	1	0.1×10^{-5}

Algorithm 2A avoids the problem of the premature hypothesis removal by using function rank_i rather than b_i^+ to choose a removed hypothesis. Function $\mathrm{rank}_i(h_j)$ is calculated by combining $b_i^{+\mathrm{p}}(h_j)$ and $b_i^{+\mathrm{n}}(h_j)$ while weighting the contribution of $b_i^{+\mathrm{p}}(h_j)$ according to the number of symptoms observed so far. Let $B_i^{+\mathrm{n}}(h_j)$ and $B_i^{+\mathrm{p}}(h_j)$ represent logarithmic-scale values of $b_i^{+\mathrm{n}}(h_j)$ and $b_i^{+\mathrm{p}}(h_j)$, respectively. The value of $\mathrm{rank}_i(h_j)$ is calculated using the following equation:

$$\mathrm{rank}_i(h_j) = B_i^{+\mathrm{n}}(h_j) + \beta(i)B_i^{+\mathrm{p}}(h_j). \tag{12}$$

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 symptoms 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:

$$\beta(i) = 1 - 2^{-\mathrm{SW}((i-1)/\mathrm{EEF})^2}. \tag{13}$$

In Eq. (13), 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., $\mathrm{EEF} = c|\mathscr{S}|OR/|\mathscr{F}|$. 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:

$$\mathrm{SW} = 1 - \frac{\sum_{s_i\in\mathscr{S}} p_{\mathrm{s}}(s_i)}{\sum_{s_i\in\mathscr{S}}\sum_{f_l\in\mathscr{F}} p(s_i|f_l) + \sum_{s_i\in\mathscr{S}} p_{\mathrm{s}}(s_i)}. \tag{14}$$

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. Clearly, other definitions of function $\beta(i)$ would be possible. For instance, we could incorporate a temporal aspect into function $\beta(i)$ by increasing its value with time. Such a definition could represent a property that, after a certain time since the fault localization is started, all relevant symptoms should have been observed.

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

Example 4 (*continued*). Consider again the scenario solved in Table 2. Observe that after the first symptom is observed, i.e., when $i = 1$, $\beta(i) = 0$. Consequently, $\mathrm{rank}_i(h_j) = B_i^{+\mathrm{n}}(h_j)$, which means that the impact of positive symptoms is ignored by the ranking scheme. When instead of the belief metric, function rank_i is used to single out a removed hypothesis, $\{f_1\}$ is eliminated rather than $\{f_2\}$. In the second iteration, $b_2^{+\mathrm{p}}(\{f_2\})$ becomes equal to 1, and $b_2^{+\mathrm{n}}(\{f_2\}) = 0.8 \cdot 10^{-4}$. Thus, $\{f_2\}$ becomes the best hypothesis.

6. Simulation study

In this section we evaluate the techniques presented in this paper using the problem of end-to-end service failure diagnosis as a case study. We first estimate the complexity of Algorithms 2 and 2A in the application to this problem. Then, we proceed to comparing the accuracy and efficiency of Algorithms 1 and 2. Next, we evaluate the impact of including and disregarding positive, lost, and spurious symptoms by comparing accuracies achievable with Algorithms 2 and 2A. Finally, we investigate the sensitivity of Algorithm 2 to inaccuracies of the FPM.

6.1. Application of algorithm IHU to end-to-end service failure diagnosis

The problem of end-to-end service-failure diagnosis 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 hop-to-hop services between intermediate nodes used to provide the end-to-end connectivity. In the problem of end-to-end service-failure diagnosis, a FPM is a bipartite causality graph with hop-to-hop and end-to-end service failures at the tails and at the heads of the edges, respectively. Since in an n-node network, there are at most n^2 end-to-end services and each of them is composed of at most n hop-to-hop services, the complexity of Algorithms 2 and 2A is $\mathcal{O}(n^3 \max_i(|\mathscr{H}_i|))$ (see Section 3). Limiting $\max_i(|\mathscr{H}_i|)$ to $\mathcal{O}(n)$ makes the computational complexity of the algorithms in the application to end-to-end service failure diagnosis in an n-node network be $\mathcal{O}(n^4)$.

6.2. Simulation model

The simulation study presented in this paper uses tree-shaped network topologies, which result, for example, from the usage of the Spanning Tree Protocol [27] 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 FPM is necessary.

We design the simulation described in this section according to the model we previously used to evaluate another fault localization algorithm based on belief propagation in belief networks [21]. We use *OR*, *LR*, and *SSR* to denote the observability ratio ($|\mathscr{S}_O|/|\mathscr{S}|$), 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 FPM 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 FPM. The independent failure probabilities and conditional probabilities are uniformly distributed in ranges $[0.001, 0.01]$ and $(0, 1)$, respectively, unless specified otherwise. We randomly choose $OR|\mathscr{S}|$ observable symptoms, and place them in the set of observable symptoms, $\mathscr{S}_O$. 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, $\mathscr{F}_c \subseteq \mathscr{F}$. Using $\mathscr{F}_c$ and the conditional probability distribution we randomly generate the set of observed negative symptoms $\mathscr{S}_N \subseteq \mathscr{S}_O$. When $SSR > 0$, we also randomly choose $SSR|\mathscr{S}_O|$ symptoms from $\mathscr{S}_O$, and add them to $\mathscr{S}_N$. When $LR > 0$, we remove $LR|\mathscr{S}_N|$ random symptoms from $\mathscr{S}_N$. We use Algorithms 1, 2, or 2A to produce the most probable explanation of $\mathscr{S}_N$, $\mathscr{F}_d$. We take into account only the most likely

hypothesis from the final set of hypotheses proposed by the fault localization algorithm. We compare $\mathscr{F}_d$ to $\mathscr{F}_c$, and calculate the detection rate, DR, and false positive rate, FPR, which are defined as follows:

$$\mathrm{DR} = \frac{|\mathscr{F}_c \cap \mathscr{F}_d|}{|\mathscr{F}_c|}, \qquad \mathrm{FPR} = \frac{|\mathscr{F}_d - \mathscr{F}_c|}{|\mathscr{F}_d|}.$$

6.3. Performance evaluation

The first simulation study was conducted to compare the performance and accuracy of fault localization performed with Algorithms 1 and 2. We intentionally ignore positive, lost, and spurious symptoms. Consequently, $LR = 0$ and $SSR = 0$. In this study, the link failure probabilities are uniformly distributed random values of the order of 10^{-6}, and the conditional probabilities on causal links are uniformly distributed random values in the range $[0.5, 1)$. Because of excessive simulation time we had to limit the tested network size range for Algorithm 1 to 20.

Fig. 5(a) and (b) present relationships between the detection rate (DR) and false positive rate (FPR), respectively, and network size. We observe that there is no statistically significant difference in the detection and false positive rates between the incremental and combinatorial algorithms. Both algorithms are very accurate, but Algorithm 2 may be used in networks of much bigger size than Algorithm 1. The accuracy of Algorithm 2 depends on the network size. This dependency is due to two competing factors that have opposite effects on the accuracy: (1) a system instrumentation level, which is lower for smaller networks, and (2) a frequency of multi-fault scenarios, which is higher for bigger networks. Nevertheless, the gradual drop (increase) of DR (FPR) observed in the case of Algorithm 2 suggests that this drop (increase) may be asymptotic.

Fig. 6(a)–(d) present a comparison of execution times for the combinatorial and incremental algorithms in the presence of 1–4 network faults. The incremental algorithm performs better than the combinatorial algorithm regardless of the number of faults and network size, and the difference between the algorithms increases sharply with the increasing number of faults in the system. The average execution time of Algorithm 2, which is of the order of several seconds, even for large networks and multi-fault scenarios, is very encouraging.

Table 3 summarizes the comparison of Algorithms 1, 2 and 2A.

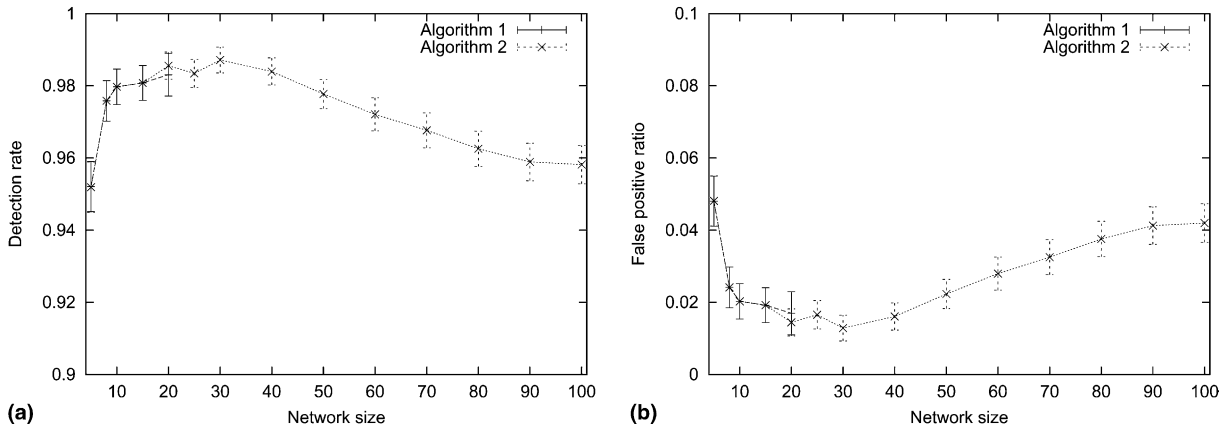


Fig. 5. Accuracy achievable with Algorithms 1 and 2: (a) detection rate and (b) false positive rate.

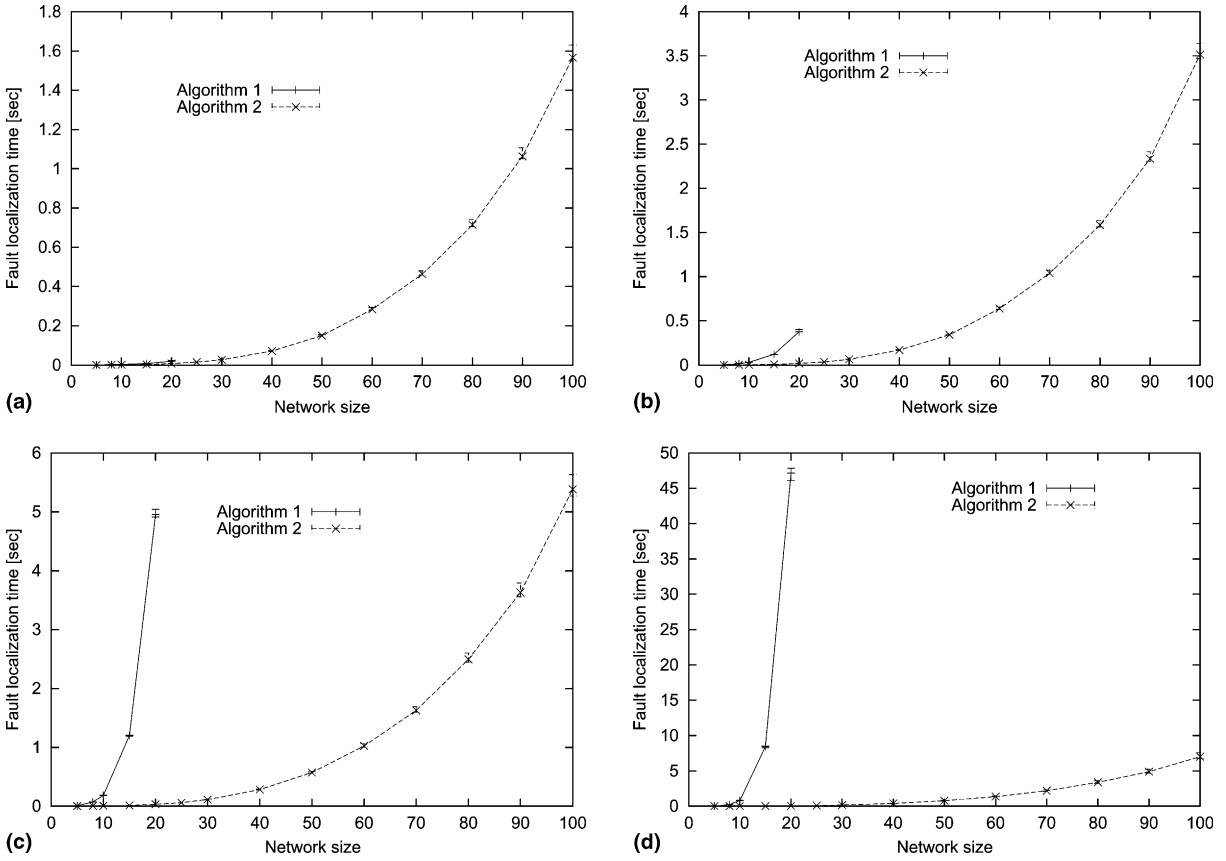


Fig. 6. Comparison of fault localization time for Algorithms 1 and 2 for different network sizes: (a) single-fault scenarios; (b) two-fault scenarios; (c) three-fault scenarios and (d) four-fault scenarios.

Table 3
Comparison of Algorithms 1, 2, and 2A

Algorithm	Combinatorial (Algorithm 1)	Incremental (Algorithms 2 and 2A)
Theoretical bound	$\exp(n)$	n^4
Detection rate[a] (%)	95–99	95–99
False positive rate[a] (%)	1–5	1–5
Max. network size with localization time <10 s[b]	15	100+
Multi-fault scenarios	Yes	Yes
Lost and spurious symptoms	Yes	Yes
Is algorithm event-driven?	No	Yes
Is algorithm incremental?	No	Yes

[a] Accuracy achievable disregarding positive symptoms in the absence of lost and spurious symptoms with system parameters described in this section.

[b] Average time spent to solve a scenario in the presence of up to four network faults.

6.4. Impact of positive symptoms

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

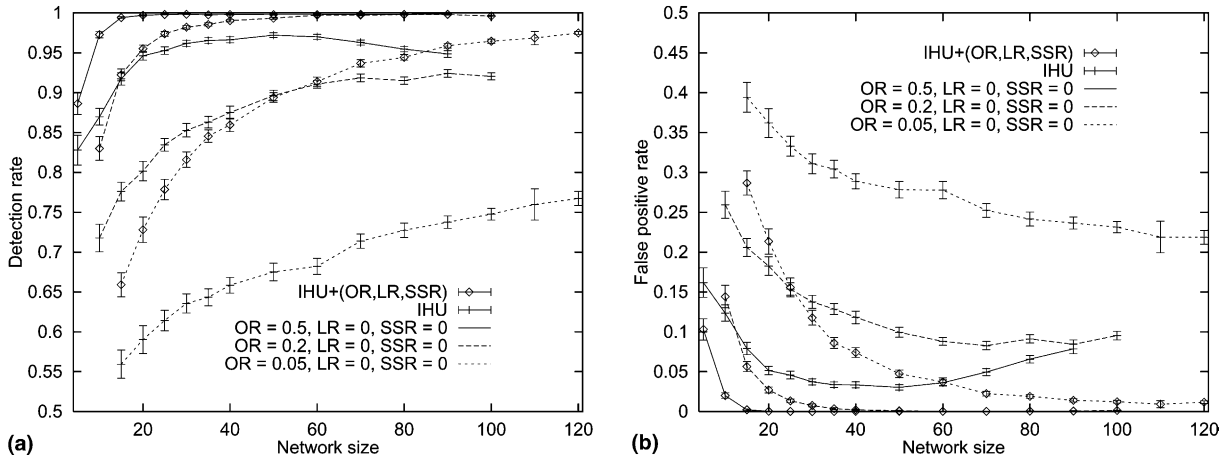


Fig. 7. Accuracy achievable with Algorithms 2 (IHU—disregarding positive symptoms) and 2A (IHU+—taking positive symptoms into account) for various observability ratios, *OR*: (a) detection rate and (b) false positive rate.

While setting *OR* to 0.5, 0.2, or 0.05, we compare DR and FPR achievable with Algorithm 2, which does not take positive symptoms into account, and Algorithm 2A, which includes positive symptoms in the analysis.

As presented in Fig. 7(a) and (b), including positive symptoms in the process of fault localization allows the DR to be significantly increased and the FPR to be significantly decreased. Overall, thanks to the positive information, the fault localization accuracy improves. We can also conclude that in poorly instrumented systems (either due to the small number of available symptoms or due to the small OR), positive symptoms may be effectively used to improve the accuracy of the fault localization process.

6.5. Impact of lost symptoms

To isolate the impact of symptom loss on the accuracy of fault localization, we set $SSR = 0$, and vary *LR* from 0.0 to 0.2. In the FPM, we use $p_{\text{loss}} = 0$, and $p_{\text{s}} = 0$. (The fault localization algorithm effectively ignores the symptom loss.) We apply Algorithm 2A to this model.

Symptom loss, when ignored by the fault localization process, does indeed decrease its accuracy: we observe the decrease of DR (Fig. 8(a)) and increase of FPR (Fig. 8(b)). The strength of the symptom-loss impact on the fault-localization 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 Algorithm 2A is resilient to symptom loss even when it relies on positive information to perform fault diagnosis.

To determine whether including an explicit representation of symptom loss into the FPM may improve the fault localization accuracy, we observe that the decreasing accuracy when symptoms are lost is due to two factors: (1) fewer symptoms are observed and therefore the system instrumentation level perceived by the fault management application is lower, and (2) some symptoms are incorrectly interpreted as positive. The relative contribution of these two factors determines the upper bound on the possible increase in the accuracy resulting from including a representation of a symptom loss in the FPM. Observe that the impact of only the second factor may be alleviated by including the representation of symptom loss in the model.

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) $OR = 0.05$, $LR = 0.0$, (2) $OR = 0.05$, $LR = 0.2$, and (3) $OR = 0.04$, $LR = 0.0$. The amount of information provided to the fault

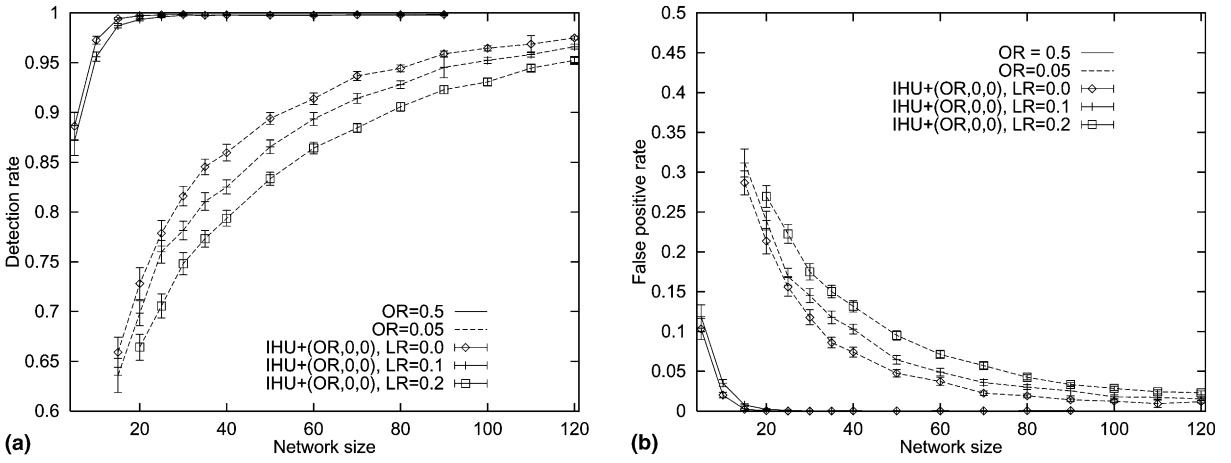


Fig. 8. The impact of symptom loss on the accuracy for various observability ratios, *OR* and symptom loss rates, *LR*: (a) detection rate and (b) false positive rate.

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 Fig. 9(a) and (b) the overall decrease of accuracy due to symptom loss is split evenly between the two factors. This lets us conclude that, should symptom loss be represented in the FPM, the resulting improvement in accuracy could not be greater than 2–2.5%. Indeed, our experiments with a FPM using $p_{\text{loss}}(s_i) = 0.2$ did not reveal any statistically provable improvement in accuracy. With higher values of *LR*, some small improvement in accuracy has been achieved.

This simulation study assumes that all symptoms are equally likely to be lost, while in reality $p_{\text{loss}}(s_i)$ is different for different symptoms. For example, when symptom loss is caused by a high packet loss rate in a network link, loss probabilities of symptoms which are transported to the management station using the malfunctioning link are higher. We expect that when symptom-loss probabilities are not equal, the benefit of including symptom loss into the FPM would be more noticeable.

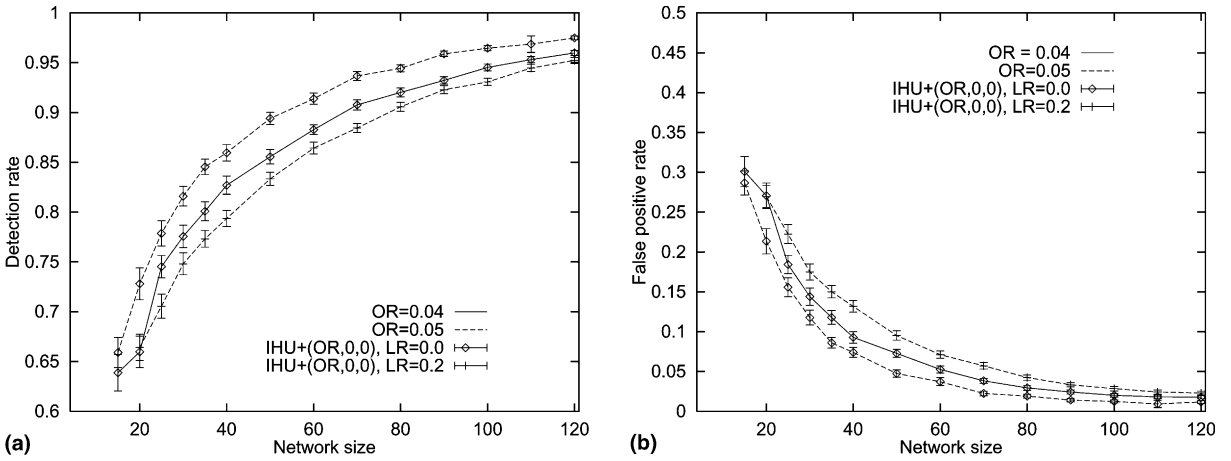


Fig. 9. The impact of factors (1) and (2) on accuracy achievable in system with $OR = 0.05$ and $OR = 0.2$: (a) detection rate and (b) false positive rate.

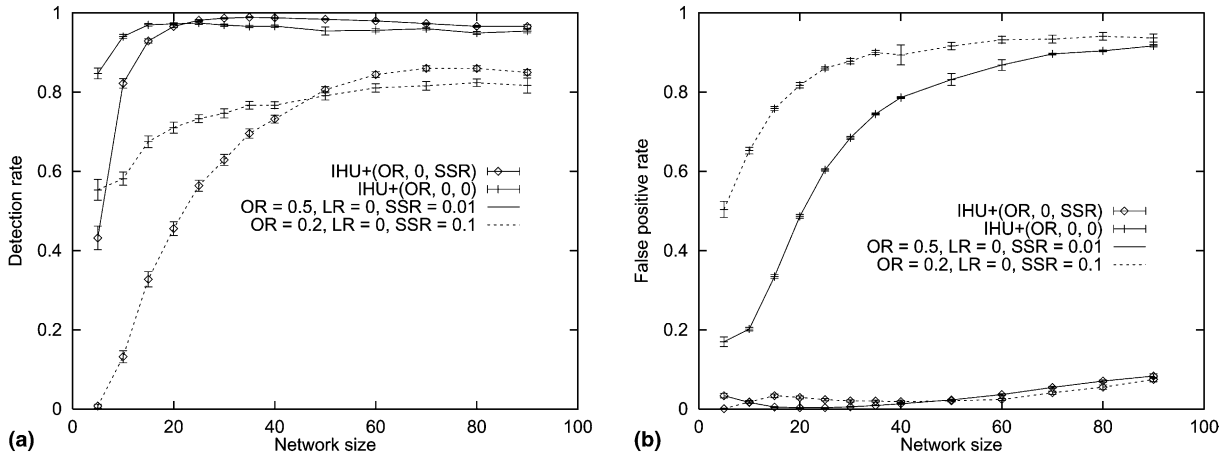


Fig. 10. The change of accuracy as a result of spurious symptoms analysis with Algorithm 2A: (a) detection rate and (b) false positive rate.

6.6. Impact of spurious symptoms

The impact of including spurious symptoms in the fault localization process is evaluated by applying Algorithm 2A to FPMs using $p_s(s_i) = 0$ and $p_s(s_i) = SSR$, respectively. We vary *OR* between 0.5 and 0.2, and use *SSR* of 0.01 and 0.1. As shown in Fig. 10(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. 10(a)) but FPR would be very high as well (Fig. 10(b)). When system instrumentation improves, so does the DR of Algorithm 2A with an accurate representation of spurious symptoms in the FPM. We conclude that including spurious symptoms in the FPM has a big impact on the accuracy of the fault localization algorithm. However, to take the full advantage of this capability, the system instrumentation level should be increased correspondingly to the rate with which spurious symptoms are generated.

Finally, we run a set of experiments to evaluate the impact of the problem of premature hypothesis removal. Fig. 11(a) and (b) compare the accuracy achievable with the incremental algorithm while disregarding spurious symptoms and while including spurious symptoms in the analysis using the unmodified and modified size-limiting heuristics. Note that when Algorithm 2A with the unmodified heuristic is used, the fault-localization accuracy with the incremental algorithm improves (i.e., FPR significantly decreases) compared to that of Algorithm 2. However, this big improvement is not consistently sustained as the network topology gets bigger: we observe a continuous decrease of DR and increase of FPR. The modified heuristic eliminates this behavior thereby improving the overall accuracy of the fault localization process.

6.7. Impact of probability estimation errors

So far in this paper, we assumed that the FPM contains probability distribution that accurately represents the modeled system. We did not discuss how these probabilities are obtained. Researchers frequently state that conditional probabilities may be assigned by an expert [2]. Since this process is error prone, it is likely that the probabilities assigned by the expert will differ from those describing the real

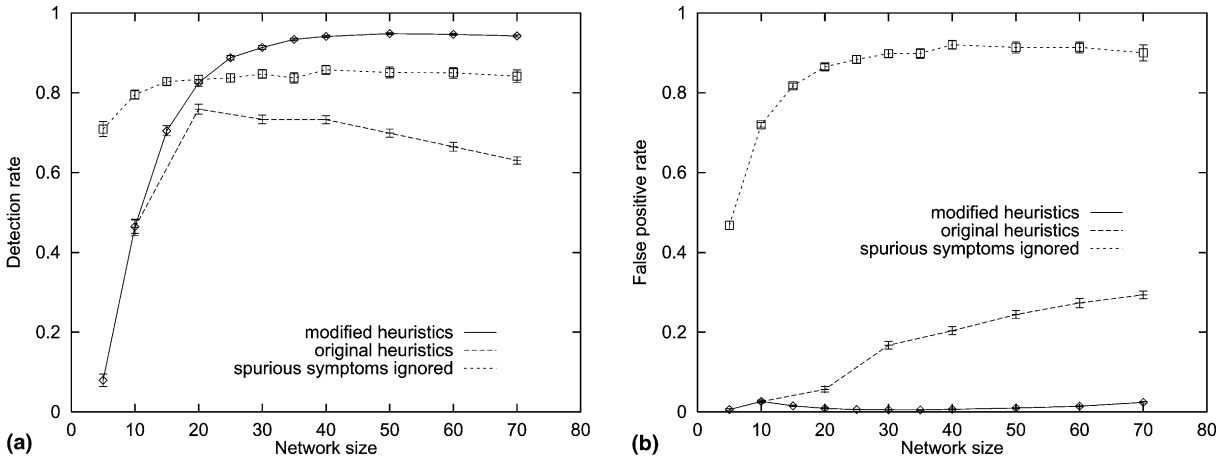


Fig. 11. The comparison of fault-localization accuracy with Algorithm 2 (spurious symptoms ignored), 2A using function b^+ (spurious symptoms handled using the original heuristic), and 2A using function rank (spurious symptoms handled using the modified heuristic), for $OR = 0.5$, $LR = 0$, and $SSR = 0.1$: (a) detection rate and (b) false positive rate.

system. In actuality, the expert assigns one of c discrete confidence levels rather than an exact probability. To represent the real-life probability p, the expert uses the ith confidence level, where $i = \lfloor pc \rfloor$. Thus, effectively real-system probability p is mapped into propagation-model weight $\lfloor pc \rfloor / c + 1/(2c)$. 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. 12(a) and (b) compares the DR and FPR of Algorithm 2 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.

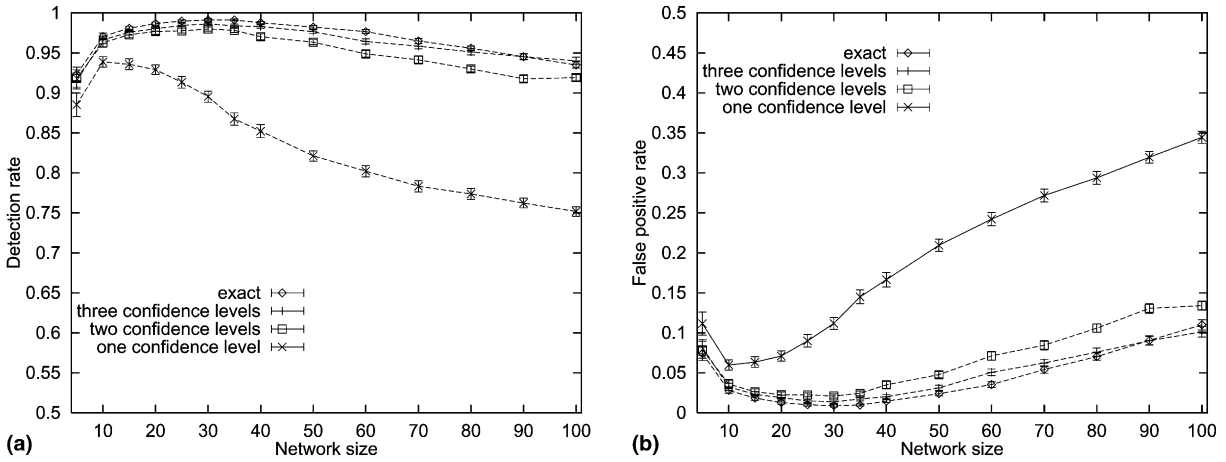


Fig. 12. Accuracy of Algorithm 2 for various granularities of confidence levels: (a) detection rate and (b) false positive rate.

7. Other canonical models

So far in this paper we assumed that the fault propagation model represents a noisy-OR model of probability distribution. However, for some fault localization problems this model may be inadequate. In this section, we present a general approach to incremental fault localization with other than noisy-OR canonical models.

7.1. AND model

Let us consider a popular high-availability scenario in which two alternative physical network connections are provided between two neighboring communication-system nodes. To model this situation using a belief network, we create vertex X to represent connectivity failure between the two nodes, and vertexes Y_1 and Y_2 to represent failures of the two physical connections, respectively, where X is caused by Y_1 and Y_2. When one of the physical connections fails, i.e., Y_1 or Y_2 occurs, the entire traffic between the two nodes is transferred to the second, still operating connection. Thus, the connectivity failure between the nodes may be observed only if both physical connections fail. Clearly, Y_1 and Y_2 do not independently contribute to X, and therefore this high-availability scenario may not be represented using the noisy-OR model. The relationship between X, and Y_1 and Y_2 should be modeled by combining X's predecessors' values using logical AND.

This section presents a general outline for the design of incremental hypothesis updating with noisy-AND models. Intuitively, in a bipartite FPM in which symptom s_i depends on faults in H_{s_i}, which are combined using operator AND, all faults in H_{s_i} have to simultaneously exist and influence s_i, for s_i to occur.

In the incremental algorithm for a noisy-AND model, hypothesis $h_j \in \mathscr{H}_i$ explains s_i if it contains all faults in H_{s_i}. Hypothesis h_j which does not explain s_i has to be extended with faults in $H_{s_i} \setminus h_j$. The belief metric b_i is calculated incrementally as follows:

1. If $h_j \in \mathscr{H}_{i-1}$ and h_j explains s_i,
$$b_i(h_j) = b_{i-1}(h_j) \prod_{f_l \in H_{s_i}} p(s_i \mid f_l). \tag{15}$$
2. Otherwise, if h_j is extended with $H'_{s_i} = H_{s_i} \setminus h_j$,
$$b_i(h_j \cup H_{s_i}) = b_{i-1}(h_j) \prod_{f_l \in H_{s_i}} p(f_l) \prod_{f_l \in H'_{s_i}} p(s_i \mid f_l). \tag{16}$$

7.2. NOT model

In the NOT model, a variable value is calculated as a logical negation of its single predecessor's value. In a bipartite fault propagation model, symptom s_i may not occur if its antecedent fault f_l occurs and influences s_i. Noisy-NOT relationship between a fault and a symptom is introduced into the calculation of b_i using the following equations, in which $p(\neg s_i \mid f_l)$ denotes the probability that symptom s_i does not occur given fault f_l occurred:

1. If $f_l \in h_j$,
$$b_i(h_j) = b_{i-1}(h_j)(1 - p(\neg s_i \mid f_l)) \tag{17}$$
2. Otherwise
$$b_i(h_j) = b_{i-1}(h_j). \tag{18}$$

7.3. A hybrid model

In real-life scenarios, a hybrid model is useful, in which a belief-network vertex may apply different logical operators to different subsets of its predecessors. In a hybrid model, symptom s_i is explained by an arbitrary logical combination of its predecessors, which may be represented as a logical clause (a disjunction of conjunctions of literals), D_{s_i}. Formally, $D_{s_i} = \{C_{s_i1}, \ldots, C_{s_in_i}\}$, where C_{s_ij}s are combined using operator $\vee$. Moreover, $C_{s_ij} = \{L_{s_ij1}, \ldots, L_{s_ijm_i}\}$, where L_{s_ijk}s are combined using operator $\wedge$. Finally, $L_{s_ijk} = f_s$ or $L_{s_ijk} = \neg f_s$, where $f_s \in \mathscr{F}$. We will also use symbol $C^+_{s_ij}$ to represent a set of all non-negative literals in C_{s_ij}, i.e., $C^+_{s_ij} = \{f_s \in \mathscr{F} \,|\, f_s \in C_{s_ij}\}$.

Given symptom s_i and hypothesis h_j, we define the following predicates:

$$h_j \text{ explains } s_i \equiv \exists_{C_{s_ij} \in D_{s_i}} C_{s_ij} \text{ is consistent with } h_j$$

$$C_{s_ij} \text{ is consistent with } h_j \equiv \forall_{L_{s_ijk} \in C_{s_ij}} L_{s_ijk} \text{ is consistent with } h_j$$

$$L_{s_ijk} \text{ is consistent with } h_j \equiv \begin{cases} f_s \in h_j \wedge p(s_i \,|\, f_s) > 0 & \text{if } L_{s_ijk} = f_s, \\ f_s \notin h_j \vee p(\neg s_i \,|\, f_s) < 1 & \text{if } L_{s_ijk} = \neg f_s, \end{cases}$$

Based on the definition of function $\mu(f_s)$ for $f_s \in \mathscr{F}$ introduced in Section 3, we also define function $\mu(\mathscr{F}_i)$, where $\mathscr{F}_i \subseteq \mathscr{F}$ such that

$$\mu(\mathscr{F}_i) = \min_{f_s \in \mathscr{F}_i} \mu(f_s).$$

In the incremental algorithm with a hybrid model, in the ith iteration, hypothesis $h_j \in \mathscr{H}_{i-1}$ is processed as follows:

1. If h_j explains s_i,
$$b_i(h_j) = b_{i-1}(h_j) P_{D_{s_i}}(s_i, h_j). \tag{19}$$
2. Otherwise, if $\alpha(h_j \cup H^*_{s_ik}) \leqslant \mu(H^*_{s_ik})$, where $H^*_{s_ik} = C^+_{s_ik} \setminus h_j$, $C_{s_ik} \in D_{s_i}$, and C_{s_ik} is consistent with $h_j \cup H^*_{s_ik}$, create hypothesis $h_j \cup H^*_{s_ik}$ and calculate b_i as follows:
$$b_i(h_j \cup H^*_{s_ik}) = b_{i-1}(h_j) \left(\prod_{f_l \in H^*_{s_ik}} p(f_l) \right) P_{C_{s_ik}}(s_i, h_j \cup H^*_{s_ik}). \tag{20}$$

Recall from Section 5 that $\alpha(h_j)$ denotes the size of hypothesis h_j. In the above algorithm, functions $P_{D_{s_i}}(s_i, h_j)$ and $P_{C_{s_ij}}(s_i, h_j)$ are defined as follows:

$$P_{D_{s_i}}(s_i, h_j) = 1 - \prod_{C_{s_ij} \in D_{s_i}} \Big(1 - P_{C_{s_ij}}(s_i, h_j)\Big),$$

$$P_{C_{s_ij}}(s_i, h_j) = \prod_{L_{s_ijk} \in C_{s_ij}} P_{L_{s_ijk}}(s_i, h_j),$$

$$P_{f_s}(s_i, h_j) = \begin{cases} p(s_i \,|\, f_s) & \text{if } f_s \in h_j, \\ 0 & \text{if } f_s \notin h_j, \end{cases}$$

$$P_{\neg f_s}(s_i, h_j) = \begin{cases} 1 - p(\neg s_i \,|\, f_s) & \text{if } f_s \in h_j, \\ 1 & \text{if } f_s \notin h_j. \end{cases}$$

8. Comparison with other fault localization techniques

Many fault localization techniques have been investigated in the literature, whose survey is presented in [28]. In the area of probabilistic fault diagnosis, several approaches have been proposed [2,9,14,17,29–31]. In this section, we briefly compare the incremental technique with other techniques that use a symptom-fault map as a fault propagation model [2,9,17,29].

So far, the most widely known fault localization technique using a symptom-fault map is the codebook technique [3,17], which is very efficient and robust against noise in alarm data. However, only deterministic-codebook algorithm has been presented and evaluated so far. The incremental algorithm is suitable as a probabilistic codebook-decoding algorithm. Katzela et al. [2] propose a fault localization algorithm using a symptom-fault map representing a simplified model of probability distribution, which assumes that all causal influences are certain. (Effectively, the FPM includes only prior failure probabilities.) Statistical methods have been applied to perform fault isolation using a non-deterministic symptom fault map [29]. Chao et al. [9] applies a symptom-fault map in a fault localization technique that isolates a LAN segment responsible for alarms observed in a multi-segment network.

The algorithm proposed in this paper focuses on event-driven and incremental diagnosis. To the best of our knowledge these are original features that have not been investigated before. The diagnosis performed with other techniques [2,9,17,29] is window-based. The incremental algorithm also focuses on the ability to deal with observation noise. This aspect has not been investigated by the techniques described in [2,9,29]. Unlike other approaches [9,2] the incremental technique does not assume any particular problem domain or probabilistic model and therefore it is more general. It is also resilient to lost and spurious symptoms, which is not the case with some other techniques [2,29].

IHU may be also compared to our previously investigated fault localization approach, which is based on belief updating in belief networks [21]. 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.

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, as discussed at the beginning of this section, the techniques proposed in the literature [2,3,9,21,29] that are suitable for bipartite models differ with respect to assumptions they are based on, capabilities, and problems they aim at addressing. 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.

9. 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 than 10 s.

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 [21], 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 [32]. [1]

References

[1] G. Jakobson, M.D. Weissman, Alarm correlation, IEEE Network 7 (6) (1993) 52–59.
[2] I. Katzela, M. Schwartz, Schemes for fault identification in communication networks, IEEE/ACM Transactions on Networking 3 (6) (1995) 733–764.
[3] S.A. Yemini, S. Kliger, E. Mozes, Y. Yemini, D. Ohsie, High speed and robust event correlation, IEEE Communications Magazine 34 (5) (1996) 82–90.
[4] C. Scott, P. Wolfe, M. Erwin, Virtual Private Networks, second ed., O'Reilly, Sebastopol, CA, 1999.
[5] R. Comerford, The new software paladins, IEEE Spectrum 37 (6) (2000) 56–61.
[6] I. Foster, C. Kesselman (Eds.), The Grid: Blueprint for a New Computing Infrastructure, Morgan Kaufmann, Los Altos, 1998.
[7] S. Graham, S. Simeonov, T. Boubez, D. Davis, G. Daniels, Y. Nakamura, R. Neyama, Building Web Services with "Java", SAMS Publishing, Indianapolis, IN, 2002.
[8] W.W.S.A.W. Group, Available from <http://www.w3.org/2002/ws/arch/>.
[9] C.S. Chao, D.L. Yang, A.C. Liu, An automated fault diagnosis system using hierarchical reasoning and alarm correlation, Journal of Network and Systems Management 9 (2) (2001) 183–202.
[10] M. Hasan, B. Sugla, R. Viswanathan, A conceptual framework for network management event correlation and filtering systems, in: M. Sloman, S. Mazumdar, E. Lupu (Eds.), Integrated Network Management VI, Chapman and Hall, London, 1999, pp. 233–246.
[11] R. Gopal, Layered model for supporting fault isolation and recovery, in: [33], pp. 729–742.
[12] S. Kätker, A modeling framework for integrated distributed systems fault management, in: C. Popien (Ed.), Proceedings of the IFIP/IEEE International Conference on Distributed Platforms, Dresden, Germany, 1996, pp. 187–198.
[13] S.H. Schwartz, D. Zager, Value-oriented network management, in: [33].
[14] R.H. Deng, A.A. Lazar, W. Wang, A probabilistic approach to fault diagnosis in linear lightwave networks, in: H.G. Hegering, Y. Yemini (Eds.), Integrated Network Management III, North-Holland, Amsterdam, 1993, pp. 697–708.
[15] A. Dupuy, J. Schwartz, Y. Yemini, G. Barzilai, A. Cahana, Network fault management: a user's view, in: B. Meandzija, J. Westcott (Eds.), Integrated Network Management I, North-Holland, Amsterdam, 1989, pp. 101–107.
[16] P. Hong, P. Sen, Incorporating non-deterministic reasoning in managing heterogeneous network faults, in: I. Krishnan, W. Zimmer (Eds.), Integrated Network Management II, North-Holland, Amsterdam, 1991, pp. 481–492.
[17] S. Kliger, S. Yemini, Y. Yemini, D. Ohsie, S. Stolfo, A coding approach to event correlation, in: A.S. Sethi, F. Faure-Vincent, Y. Raynaud (Eds.), Integrated Network Management IV, Chapman and Hall, London, 1995, pp. 266–277.
[18] K. Appleby, G. Goldszmidt, M. Steinder, Yemanja—a layered event correlation system for multi-domain computing utilities, Journal of Network and Systems Management 10 (2) (2002) 171–194.
[19] M. Brodie, I. Rish, S. Ma, Optimizing probe selection for fault localization, in: O. Festor, A. Pras (Eds.), Proceedings of the Twelfth International Workshop on Distributed Systems: Operations and Management, Nancy, France, 2001.
[20] M. Brodie, I. Rish, S. Ma, Intelligent probing: a cost-efficient approach to fault diagnosis in computer networks, IBM Systems Journal 41 (3) (2002) 372–385.
[21] M. Steinder, A.S. Sethi, Probabilistic fault localization in communication systems using belief networks, IEEE/ACM Transactions on Networking, in press.
[22] M. Steinder, A.S. Sethi, Non-deterministic diagnosis of end-to-end service failures in a multi-layer communication system, in: Proceedings of ICCCN, Scottsdale, AZ, 2001, pp. 374–379.

[1] 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 US Government.

[23] M. Steinder, A.S. Sethi, Non-deterministic fault localization in communication systems using belief networks, Technical Report 2003-03, CIS Department, University of Delaware, Available from <www.cis.udel.edu/~steinder/PAPERS/TR-2003-03.pdf> (September 2002).
[24] A.T. Bouloutas, S. Calo, A. Finkel, Alarm correlation and fault identification in communication networks, IEEE Transactions on Communications 42 (2/3/4) (1994) 523–533.
[25] M. Steinder, A.S. Sethi, Non-deterministic event-driven fault diagnosis through incremental hypothesis updating, in: G. Goldszmidt, J. Schoenwaelder (Eds.), Integrated Network Management VIII, Colorado Springs, CO, 2003, pp. 635–648.
[26] J.D. Case, K. McCloghrie, M.T. Rose, S. Waldbusser, Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2), IETF Network Working Group, 1996, RFC 1905.
[27] R. Perlman, Interconnections, Second Edition: Bridges, Routers, Switches, and Internetworking Protocols, Addison Wesley, Reading, MA, 1999.
[28] M. Steinder, Fault localization in communication networks—a survey, Technical Report 2001-01, CIS Department, University of Delaware, February 2001.
[29] M. Fecko, M. Steinder, Combinatorial designs in multiple faults localization for battlefield networks, in: IEEE Military Communications Conference (MILCOM), McLean, VA, 2001.
[30] D. Heckerman, M.P. Wellman, Bayesian networks, Communications of the ACM 38 (3) (1995) 27–30.
[31] C. Wang, M. Schwartz, Identification of faulty links in dynamic-routed networks, IEEE Journal on Selected Areas in Communications 11 (3) (1993) 1449–1460.
[32] M. Steinder, A.S. Sethi, Distributed fault localization in hierarchically routed networks, in: M. Feridun, P. Kropf, G. Babin (Eds.), 13th International Workshop on Distributed Systems: Operations and Management, Lecture Notes in Computer Science, vol. 2506, Springer, Montréal, Canada, 2002, pp. 195–207.
[33] J.W. Hong, R. Weihmayer (Eds.), Proceedings of the Network Operation and Management Symposium, Honolulu, Hawaii, 2000.

M. Steinder received her M.S. degree in Computer Science from AGH University of Science and Technology, Poland in 1994. She was a junior faculty member at AGH from 1994 to 1998. In 2003 she received a Ph.D. in Computer and Information Sciences from the University of Delaware. For her work on probabilistic fault localization techniques she was awarded Allan P. Colburn Prize for the Outstanding Doctoral Dissertation in Mathematical Sciences and Engineering from University of Delaware. In 2003, she joined Service Management Middleware Department in IBM T. J. Watson Research Center, as a Research Staff Member. She is currently working on dynamic resource management for WebSphere On-Demand Operating Environment, leading the effort in autonomic server and application provisioning. She also serves as a TPC member for IEEE INFOCOM 2004.

A.S. Sethi is a Professor in the Department of Computer and Information Sciences at the University of Delaware, Newark, Delaware, USA. He has an MS in Electrical Engineering and a PhD in Computer Science, both from the Indian Institute of Technology, Kanpur, India. He has served as the faculty at IIT Kanpur, was a visiting faculty at Washington State University, Pullman, WA, and Visiting Scientist at IBM Research Laboratories, Zurich, Switzerland, and at the US Army Research Laboratory, Aberdeen, MD. He is in the Editorial Advisory Board for the Journal of Network and Systems Management, an Editor for the electronic IEEE Transactions on Network and Service Management, and an Associate Editor for the Electronic Commerce Research Journal. He was co-Chair of the Program Committee for ISINM '95, and was General and Program Chair for DSOM '98; he has also been on the program committees of numerous conferences. His research interests include architectures and protocols for network management, fault management, quality-of-service and resource management, and management of wireless networks.