Multidomain Diagnosis of End-to-End Service Failures in Hierarchically Routed Networks

Malgorzata Steinder, *Member*, *IEEE*, and Adarshpal S. Sethi, *Member*, *IEEE*

Abstract—Probabilistic inference was shown effective in the nondeterministic diagnosis of end-to-end service failures when applied in a centralized management system where the manager possesses a global knowledge of the system structure and state. Since many networks are organized into multiple administrative domains that may be unable to share configuration and state information, these centralized techniques are not applicable to them. This paper proposes a fault localization technique suitable for multidomain networks with hierarchical routing. The proposed technique divides the computational effort and system knowledge among multiple, hierarchically organized managers. Each manager performs fault localization in the domain it manages and requires only the knowledge of its own domain. We show through simulation that the proposed approach not only improves the feasibility of fault localization in multidomain networks, but also increases the effectiveness of probabilistic diagnosis and makes it realizable in networks of considerable size.

Index Terms—Distributed fault localization, probabilistic inference.

1 Introduction

End-to-end connectivity in a given protocol layer is frequently provided through a sequence of intermediate nodes such as bridges in the data-link layer or routers in the network layer. Communication problems between a pair of these nodes may cause disruption on one or more end-to-end paths provided using the failing node-to-node link. For example, a malfunctioning interface of a router may cause bit errors or high packet loss rate to be observed on one or more end-to-end paths that are routed over the affected interface. These end-to-end problems propagate to higher system layers causing various application-level events, e.g., aborted transactions, session timeouts, abnormal delays, etc. Therefore, it is important that node-to-node problems, both availability and performance-related ones, be identified quickly and accurately. Unfortunately, node-to-node failures can often not be detected directly by monitoring node-to-node connectivity. This is due to the fact that certain failure conditions cannot be monitored on a node-to-node basis either because there is no appropriate monitoring mechanism or because of the associated overhead. Moreover, an end-to-end service user frequently does not have the administrative authority allowing her to monitor node-to-node connectivity. In these situations, node-to-node problems have to be identified by correlating indications of end-to-end disorder.

This paper adopts a service-oriented view of the network [7], in which end-to-end or node-to-node connectivity between two nodes in a given protocol layer is considered a service provided by this layer to higher layers. End-to-end service between nodes a and b is implemented using (i.e., depends on) a set of node-to-node services between neighboring nodes on a path from a to b.

Diagnosis of end-to-end network service failures [20] is a subtask of fault localization [8], [12], [21], [22] that isolates node-to-node services responsible for availability or performance problems experienced by end-to-end services. In our previous work [20], [19], we investigated an application of probabilistic reasoning to end-to-end service failure diagnosis and proposed algorithms, which were shown effective in the diagnosis of end-to-end service failures in networks composed of tens of nodes. In addition, they proved to be resilient against lost and spurious symptoms and to be insensitive to the inaccuracies of the probabilistic FPM [20], [19].

Today's networks are frequently composed of multiple domains, each with a different organization and management policy. The algorithms proposed in [19], [20] are difficult to apply to such big multidomain topologies. They exhibit shortcomings typical of any centralized management scheme, which include infeasibility, inefficiency, inflexibility, insecurity, and unreliability.

This paper introduces a multidomain fault-localization technique, which increases the admissible network size by an order of magnitude by taking advantage of the domain semantics of communication systems. The proposed technique divides the computational effort and system knowledge involved in end-to-end service-failure diagnosis among multiple hierarchically organized managers. Each manager is responsible for fault localization within the network domain it governs and reports to a higher-level manager that oversees and coordinates the fault-localization process of multiple domains. With this organization, the technique is suitable for the distributed diagnosis of end-to-end service failures in hierarchically routed networks. The paper is an extended version of [18].

Distributed fault localization has been recognized as an important objective of fault management systems [4], [11], [22], but few such distributed techniques have actually been

- *M. Steinder is with the IBM T.J. Watson Research Center, 19 Skyline Dr., Hawthorne, NY 10532. E-mail: steinder@us.ibm.com.*
- *A.S. Sethi is with the Department of Computer and Information Sciences, University of Delaware, Newark, DE 19716. E-mail: sethi@cis.udel.edu.*

Manuscript received 31 May 2005; revised 23 Nov. 2005; accepted 25 Jan. 2006; published online 25 Jan. 2007.
Recommended for acceptance by Y. Pan.
For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number TPDS-0269-0505.

1045-9219/07/$25.00 © 2007 IEEE Published by the IEEE Computer Society

proposed. A theoretical foundation for the design of such systems has been laid by Bouloutas et al. [4] and Katzela et al. [10], who investigate different schemes of noncentralized fault localization: decentralized and distributed schemes. The technique proposed in this paper has properties of both these schemes. Similar to the decentralized scheme [10], we envision a hierarchy of managers with a central manager making the final fault determination. Unlike the decentralized scheme, however, higher-level managers not only arbitrate among solutions proposed by lower-level managers, but also participate in the actual fault determination by proposing their own hypotheses composed of network faults that cannot be identified by the lower-level managers.

Our goal in this paper is to find a solution to end-to-end service failure diagnosis in a multidomain network while addressing the problems of failure propagation among domains and the lack of global knowledge. We recognize that various probabilistic reasoning mechanisms may be used by managers. Therefore, we aim at presenting a generic technique of decomposing the problem of end-to-end service failure diagnosis into multiple smaller subproblems. This decomposition complies with the domain semantics of the communication systems. The resulting technique may then be specialized for a variety of such probabilistic reasoning mechanisms. We also aim at showing two specializations of the generic technique tailored toward the iterative belief updating [20] and incremental hypothesis updating [19] as probabilistic reasoning mechanisms.

This paper is structured as follows: In Section 2, the motivation behind multidomain fault localization is discussed. In Section 3, an outline of a multidomain fault localization technique for hierarchically routed networks is proposed. A distributed fault propagation model is proposed in Section 4 and a multidomain fault localization algorithm is presented in Section 5. In Section 6, a multidomain algorithm based on event-driven belief updating [20] is introduced. In Section 7, a multidomain algorithm derived from incremental hypothesis updating [19] is proposed. Section 8 presents results of the simulation study conducted to verify the effectiveness of the proposed multidomain techniques.

2 Probabilistic Diagnosis of End-to-End Service Failures

When connectivity between nodes a and b in a given network layer is achieved through a sequence of intermediate nodes, we say that the service of end-to-end communication between hosts a and b provided by this layer to higher layers is implemented in terms of multiple services of node-to-node communication between subsequent hops on the path from node a to node b. A failure of a node-to-node service, such as excessive delay, high packet loss rate, erroneous packet transmission, or total loss of connectivity, propagates to an end-to-end service implemented using the failing node-to-node service. How a specific failure of a node-to-node service affects a dependent end-to-end service is decided by the communication protocol used in the given layer. For example, when the protocol implements an error detection mechanism, erroneous output produced by a node-to-node service results in data loss in a dependent end-to-end service. When the protocol does not implement an error detection mechanism, erroneous output produced by a node-to-node service does not affect the data loss rate of a dependent end-to-end service. Instead, erroneous output will be observed.

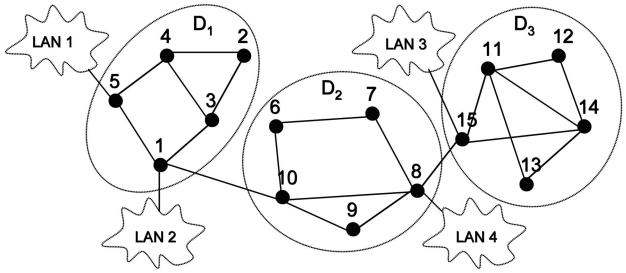
The problem of end-to-end service failure diagnosis is to identify the set of node-to-node service failures that are the most probable causes of the observed end-to-end disorder based on the information on causal relationships between node-to-node and end-to-end service failures provided in the form of a fault propagation model (FPM). The FPM for end-to-end service failure diagnosis is a bipartite causality graph in which parentless nodes (called *link* nodes) represent node-to-node service failures (faults) and childless nodes (called *path* nodes) represent end-to-end service failures (symptoms). Each node-to-node or end-to-end service may have multiple link or path nodes that correspond to different types of failures. Since causal relationships between node-to-node and end-to-end service failures are difficult to determine due to their dynamic and unpredictable nature, the FPM is a probabilistic one, in which each *link* node is labeled with the probability of the corresponding fault's independent occurrence and causal edges between *link* nodes and *path* nodes are weighted with the probability of the causal implication between corresponding faults and symptoms.

Fig. 1c shows a partial FPM for a sample multidomain network in Fig. 1a which is based on routing information in Fig. 1b. For brevity and clarity, only a subset of all possible link and path failures is shown. Paths are denoted by $r\langle a, b\rangle$, where a and b are network routers that are closest to end hosts. Links are denoted by $l\langle a, b\rangle$.

In this paper, we denote by $\mathcal{S}$ and $\mathcal{F}$ the set of all possible end-to-end service failures (symptoms) and the set of all possible node-to-node service failures (faults), respectively. The set of all observed symptoms is denoted by $\mathcal{S}_O$. In the process of fault localization, each observed symptom is mapped into the corresponding *path* node of the FPM. End-to-end service diagnosis correlates all observed symptoms to isolate one or more responsible faults, i.e., *link* failures.

Distributed fault localization has been recognized as an important objective of fault management systems [14, 91, 173]. Its goal is to divide the system knowledge and computational effort among multiple managers. Distributed fault localization alleviates the problems associated with centralized systems, but it is significantly more difficult to achieve due to the following reasons:

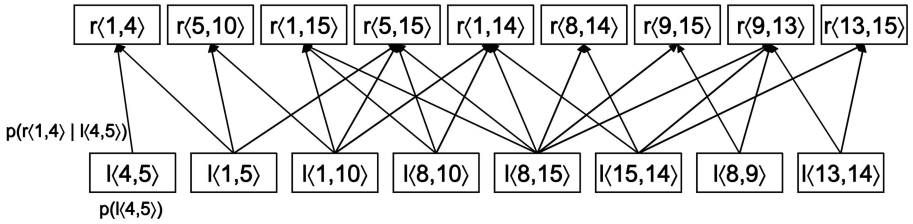
- Failure propagation among domains: Symptoms of a fault which occurred in one domain may be observed in other domains. In fact, it is possible that a fault is not at all detected in the domain in which it occurred.
- A lack of global information about the system structure and state: A symptom diagnosis is complicated because not all of its possible causes are visible in a domain in which the symptom was observed. For example, when diagnosing high-latency of communication to a particular destination, the manager of the origin does not have a complete knowledge of the state of all network links that are used by paths from the origin to the destination.



(a)

Path	Visited nodes
$r\langle 1,4\rangle$	1, 5, 4
$r\langle 5,10\rangle$	5, 1, 10
$r\langle 1,15\rangle$	1, 10, 8, 15
$r\langle 5,15\rangle$	5, 1, 10, 8, 15
$r\langle 1,14\rangle$	1, 10, 8, 15, 14
$r\langle 8,14\rangle$	8, 15, 14
$r\langle 9,15\rangle$	9, 8, 15
$r\langle 9,13\rangle$	9, 8, 15, 14, 13
$r\langle 13,15\rangle$	13, 14, 15

(b)



(c)

Fig. 1. The construction of an FPM for an example network. The FPM models only one failure type per path or link. It assumes that all routes are bidirectional. Necessarily, only subsets of possible links and paths are shown. (a) Network. (b) Routes. (c) Fault propagation model.

For example, a failing interface of router 10 in Fig. 1a may cause bit errors to be introduced into packets forwarded from this router to router 8. This failure may remain unobserved in domain D_2 if there are no end users of the failing link in this domain. Instead, increased packet loss or error rates will be observed, for example, by users in LAN 1, which is connected to the network via router 5 in domain D_1 when they try to establish sessions to hosts in LAN 3 in domain D_3.

These issues are resolved by allowing managers to cooperate to reach the solution. The technique introduced in this paper defines the scope of each manager's knowledge to cover only the domain it manages. It also proposes a hierarchical algorithm that allows the managers to reach a consensus diagnosis of symptoms observed by each of them. The managers do not need to reveal to each other the complete information of the topology and state of domains they govern nor do they need to give each other access to their devices. The only data they communicate involves certain aggregate probability values.

As an example, let us consider a failure of end-to-end path between hosts located in LAN 1 and LAN 3. This failure will be first reported to the manager of domain $\mathcal{D}_1$, which is unable to diagnose the problem since the path is not completely contained within $\mathcal{D}_1$. In the hierarchical solution, $\mathcal{D}_1$ delegates the diagnosis of the failed path to a higher level manager. The higher level manager recognizes that the communication problem between domains $\mathcal{D}_1$ and $\mathcal{D}_3$ may be caused by failures located in $\mathcal{D}_1$, $\mathcal{D}_2$, or $\mathcal{D}_3$ or by a failure of links between these domains. The higher level manager splits the end-to-end path failure into segments contained in domains $\mathcal{D}_1$, $\mathcal{D}_2$, or $\mathcal{D}_3$ and reports these segments to the managers of the respective domains as possible symptoms. The lower-level managers correlate these external symptoms with their internally observed symptoms. At the end, they report the probability that the end-to-end path failure is in fact caused in their domain. This allows the higher-level manager to pinpoint the responsible domain or interdomain link. If the failure is caused inside a lower-level domain, the manager of the domain is responsible for identifying the fault precisely.

The technique proposed in this paper is not a complete fault localization algorithm as it relies on the prior existence of some centralized fault localization algorithm capable of:

- analyzing each symptom $s_i \in \mathcal{S}$ in an event driven manner and
- at any time in the process of fault localization, providing the probability that a fault exists or does not exist in the system based on evidence (symptoms) observed thus far.

Algorithms proposed in [20], [19] are examples of algorithms that meet these criteria. In the following sections, centralized algorithm functions that deliver functionality listed above are named $inference(s_i)$, $Prob(f_j)$, and $Prob(\neg f_j)$, respectively. The proposed technique shows how these centralized algorithms may be used in a hierarchical fashion to allow multiple managers to reach a consensus explanation of the set of observed symptoms.

3 Multidomain Approach to End-to-End Service Failure Diagnosis

In this section, we introduce a multidomain approach to the probabilistic diagnosis of end-to-end service failures in hierarchically organized networks.

TABLE 1
Notation Used in the Paper

$l\langle n_k, n_l\rangle$	A directed link from node n_k to node n_l, where n_k and n_l are node identifiers that are unique network-wide, e.g., IP addresses
$r\langle n_{p_1}, n_{p_m}\rangle$	A directed, possibly multi-hop path from node n_{p_1} to node n_{p_m} consisting of links $l\langle n_{p_1}, n_{p_2}\rangle$, ..., $l\langle n_{p_{m-1}}, n_{p_m}\rangle$.
$R\langle i, j\rangle$	The set of all paths that begin in domain $\mathcal{D}_i$ and end in domain $\mathcal{D}_j$, i.e., $R\langle i,j\rangle = \{r\langle n_k, n_l\rangle \mid n_k \in \mathcal{D}_i$ and $n_l \in \mathcal{D}_j\}$, where i and j are unique domain identifiers, e.g., IP subnet masks.
$d(n_k)$	A function mapping a node identifier into an identifier of a domain to which the node belongs. In IP networks, function $d(n_k)$ is implemented using an IP address mask.

We introduce the notation presented in Table 1.

Although the technique proposed in this paper may be applied in networks with multiple levels of the hierarchy, for simplicity, we focus on a two-level architecture. Consequently, we use $\mathcal{N}$ and $\mathcal{D}_i$ to denote the entire network and its subdomain, respectively. At the root of the management hierarchy, we position a network manager NM, which oversees and coordinates the operation of domain managers DM_i.

Furthermore, the technique presented in this paper allows multiple types of failures to be associated with each path and link. However, for clarity, we will assume that only one failure type may be associated with a path or a link. Consequently, we use symbol $r\langle n_{p_1}, n_{p_m}\rangle$ to represent both a path and its observable failure (symptom). Correspondingly, we use symbol $l\langle n_k, n_l\rangle$ to represent both a link and its failure (fault). Similarly, we use symbol $R\langle i, j\rangle$ to denote both a set of end-to-end paths and a symptom associated with it, where symptom $R\langle i, j\rangle$ occurs if at least one symptom $r\langle n_{p_1}, n_{p_m}\rangle \in R\langle i, j\rangle$ occurs.

For an end-to-end path $r\langle n_{p_1}, n_{p_m}\rangle$ consisting of links $l\langle n_{p_1}, n_{p_2}\rangle, \ldots, l\langle n_{p_{m-1}}, n_{p_m}\rangle$, we define the following concepts:

Definition 1. *Path $r\langle n_{p_1}, n_{p_m}\rangle$ traverses $\mathcal{D}_i$ if and only if there exists $n_{p_j} \in \mathcal{D}_i$. Path $r\langle n_{p_1}, n_{p_m}\rangle$ is an* intradomain path *in $\mathcal{D}_i$ if each n_{p_j} belongs to $\mathcal{D}_i$. Path $r\langle n_{p_1}, n_{p_m}\rangle$ that traverses $\mathcal{D}_i$ but is not an intradomain path in $\mathcal{D}_i$ is an* interdomain path *with respect to $\mathcal{D}_i$.*

Definition 2. *Let path $r\langle n_{p_1}, n_{p_m}\rangle$ be an interdomain path with respect to $\mathcal{D}_l$. Let $\mathcal{D}_i$ and $\mathcal{D}_j$ be domains such that $n_{p_1} \in \mathcal{D}_i$ and $n_{p_m} \in \mathcal{D}_j$. Node $n_{p_k} \in \mathcal{D}_l$ such that $1 < k \leq m$ and $n_{p_{k-1}} \notin \mathcal{D}_l$ is an ingress gateway from $\mathcal{D}_i$ to $\mathcal{D}_j$ in $\mathcal{D}_l$ and is denoted by $I^l_{i,j}$. Similarly, node $n_{p_n} \in \mathcal{D}_l$ such that $1 \leq n < m$ and $n_{p_{n+1}} \notin \mathcal{D}_l$ is an egress gateway from $\mathcal{D}_i$ to $\mathcal{D}_j$ in $\mathcal{D}_l$ and is denoted by $E^l_{i,j}$ (Fig. 2). When routes are bidirectional, for any i,j, and l, $I^l_{i,j} E^l_{j,i} = G^l_{i,j}$ and $I^l_{j,i} = E^l_{i,j} = G^l_{j,i}$.*

Definition 3. *Let path $r\langle n_{p_1}, n_{p_m}\rangle$ such that $n_{p_1} \in \mathcal{D}_i$ and $n_{p_m} \in \mathcal{D}_j$ be an interdomain path with respect to $\mathcal{D}_l$. Path $r\langle I^l_{i,j}, E^l_{i,j}\rangle$ is called an* intra-$\mathcal{D}_l$ *segment of path $r\langle n_{p_1}, n_{p_m}\rangle$ (Fig. 2).*

In the presented technique, we rely on the following properties of the hierarchical networks:

1. Management domains are either disjoint or all-inclusive, i.e., for any $\mathcal{D}_i$ and $\mathcal{D}_j$, either $\mathcal{D}_i \cap \mathcal{D}_j = \emptyset$ or $\mathcal{D}_i = \mathcal{D}_j$.
2. No path enters the same domain more than once, i.e., for any path $r\langle n_{p_1}, n_{p_m}\rangle$ consisting of links $l\langle n_{p_1}, n_{p_m}\rangle, \ldots, l\langle n_{p_{m-1}}, n_{p_m}\rangle$, if $d(n_{p_i}) \neq d(n_{p_{i+1}})$, then for all j, such that $m > j > i+1$, $d(n_{p_j}) \neq d(n_{p_i})$.

We also make a simplifying assumption that all end-to-end paths that begin in domain $\mathcal{D}_i$ end in domain $\mathcal{D}_j$, and those that traverse domain $\mathcal{D}_l$ enter $\mathcal{D}_l$ through the same node, i.e., if $r\langle n_{p_1}, n_{p_m}\rangle$ and $r\langle n_{q_1}, n_{q_n}\rangle$ are two paths that traverse links $l\langle n_{p_1}, n_{p_m}\rangle, \ldots, l\langle n_{p_{m-1}}, n_{p_m}\rangle$ and $l\langle n_{q_1}, n_{q_2}\rangle, \ldots, l\langle n_{q_{n-1}}, n_{q_n}\rangle$, respectively, such that $n_{p_1}, n_{q_1} \in \mathcal{D}_i$, $n_{p_m}, n_{q_n} \in \mathcal{D}_j$, and both paths traverse domain $\mathcal{D}_l$, then for n_{p_t} and n_{q_s} such that $n_{p_{t-1}}, n_{q_{s-1}} \notin \mathcal{D}_l$ and $n_{p_t}, n_{q_s} \in \mathcal{D}_l$, $n_{p_t} = n_{q_s} = I^l_{i,j}$. In addition, $r\langle n_{p_1}, n_{p_m}\rangle$ and $r\langle n_{q_1}, n_{q_n}\rangle$ leave $\mathcal{D}_i$ through the same node, i.e., for n_{p_t} and n_{q_s} such that $n_{p_t}, n_{q_s} \in \mathcal{D}_l$ and $n_{p_{t+1}}, n_{q_{s+1}} \notin \mathcal{D}_l$, $n_{p_t} = n_{q_s} = E^l_{i,j}$. This assumption restricts the applicability of our solution to networks without multipath routing. It is not difficult to extend the proposed solution such that multipath routing is allowed. However, for the sake of clarity, we shall not attempt this extension in this paper.

The proposed solution requires that each DM has the minimum knowledge necessary for fault diagnosis, i.e., it is able to obtain topology and routing information only in the domain it directly manages. Thus, DM_i is aware of link $l\langle n_k, n_l\rangle$ if and only if both n_k and n_l belong to $\mathcal{D}_i$, whereas NM is aware of link $l\langle n_k, n_l\rangle$ if and only if $l\langle n_k, n_l\rangle$ is a link between $\mathcal{D}_i$ and $\mathcal{D}_j$ and nodes n_k and n_l are egress and ingress gateways in $\mathcal{D}_i$ and $\mathcal{D}_j$, respectively. Consequently, NM is able to transform any path $r\langle n_{p_1}, n_{p_m}\rangle$ that traverses domains $\mathcal{D}_{l_1}, \ldots, \mathcal{D}_{l_k}$ into a sequence of intradomain path segments and links

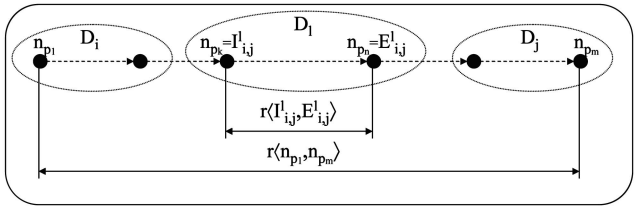


Fig. 2. Definition of a path segment, and ingress and egress gateways.

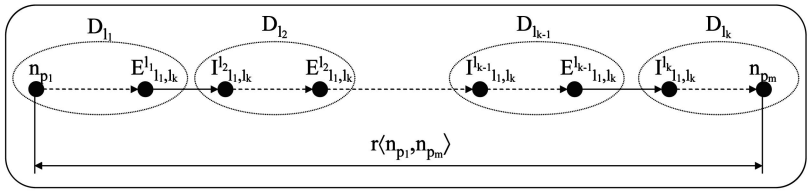


Fig. 3. Transformation of an end-to-end path into a sequence of interdomain links and intradomain path segments.

$r\langle n_{p_1}, E^{l_1}_{l_1,l_k}\rangle$, $l\langle E^{l_1}_{l_1,l_k}, I^{l_2}_{l_1,l_k}\rangle$, $r\langle I^{l_2}_{l_1,l_k}, E^{l_2}_{l_1,l_k}\rangle, \ldots, l\langle E^{l_{k-1}}_{l_1,l_k}, I^{l_k}_{l_1,l_k}\rangle$, and $r\langle I^{l_k}_{l_1,l_k}, n_{p_m}\rangle$ (Fig. 3). Moreover, DM_i is able to obtain a complete route for each end-to-end path $r\langle n_k, n_l\rangle$ such that $d(n_k) = d(n_l) = i$, but it cannot obtain the topology and routing information for any parts of the network located outside of $\mathcal{D}_i$. Consequently, DM_i is unable to determine either a complete route or a path-segment sequence for any path that is interdomain with respect to $\mathcal{D}_i$.

4 Distributed Fault Propagation Model

In the multidomain solution, the fault propagation model (FPM) of the entire network is distributed among DMs. Each manager maintains a part of the distributed FPM that represents the manager's knowledge of the system structure. An FPM built by DM_i is a bipartite causality graph with end-to-end and node-to-node service failures at the heads and at the tails of the edges, respectively, similar to the model described in Section 2. However, in the multidomain approach, the FPM of DM_i includes failures of only these end-to-end paths and node-to-node links that are entirely located in $\mathcal{D}_i$. Similarly, the FPM of NM includes failures of links that join different domains of $\mathcal{N}$. Failures of links that are completely contained in domains of $\mathcal{N}$ but which may affect end-to-end paths that span multiple domains are not explicitly included in the FPM of NM but are represented in it by proxy fault nodes, called P-nodes. Similarly, failures located outside $\mathcal{D}_i$ that may result in an observation of a symptom corresponding to an end-to-end path located in $\mathcal{D}_i$ are represented in the FPM of DM_i by proxy fault nodes, called $\tilde{P}$-nodes. Thus, the FPM built by DM_i has the same structure as in the centralized approach, but its scope is smaller and the interpretation of some of the nodes is different.

4.1 Fault Propagation Model of NM

Let us consider path $r\langle n_{p_1}, n_{p_m}\rangle$ that traverses domains $\mathcal{D}_{l_1}, \ldots, \mathcal{D}_{l_k}$. Recall that NM transforms this path into a sequence of intradomain path segments and links $r\langle n_{p_1}, E^{l_1}_{l_1,l_k}\rangle$, $l\langle E^{l_1}_{l_1,l_k}, I^{l_2}_{l_1,l_k}\rangle$, $r\langle I^{l_2}_{l_1,l_k}, E^{l_2}_{l_1,l_k}\rangle, \ldots, l\langle E^{l_{k-1}}_{l_1,l_k}, I^{l_k}_{l_1,l_k}\rangle$, and $r\langle I^{l_k}_{l_1,l_k}, n_{p_m}\rangle$ (Fig. 3). In its FPM, NM has to represent the fact that a failure of end-to-end path $r\langle n_{p_1}, n_{p_m}\rangle$ may be caused by failures of one or more of these links and path segments. This can be achieved by creating a symptom node representing path $r\langle n_{p_1}, n_{p_m}\rangle$ and fault nodes representing failures of its corresponding links and intradomain path segments. However, we can observe that, with the exception of $r\langle n_{p_1}, E^{l_1}_{l_1,l_k}\rangle$ and $r\langle I^{l_k}_{l_1,l_k}, n_{p_m}\rangle$, all paths that begin in $\mathcal{D}_{l_1}$ and end in $\mathcal{D}_{l_k}$ are transformed into the same sequence of intradomain path segments and links. (This follows from the hierarchical routing assumption.) Therefore, we can simplify the FPM by creating a single symptom node labeled $R\langle l_1, l_k\rangle$ (Fig. 4) that represents all paths that begin in $\mathcal{D}_{l_1}$ and end in $\mathcal{D}_{l_k}$. For any such path $r\langle n_{p_1}, n_{p_m}\rangle$, we say that node $R\langle l_1, l_k\rangle$ represents symptom $r\langle n_{p_1}, n_{p_m}\rangle$ in the FPM of NM. Symptom $R\langle l_1, l_k\rangle$ occurs when a failure of at least one path that begins in $\mathcal{D}_{l_1}$ and ends in $\mathcal{D}_{l_k}$, e.g., $r\langle n_{p_1}, n_{p_m}\rangle$, occurs.

The failure of $r\langle n_{p_1}, n_{p_m}\rangle$ may be caused by a failure of one or more interdomain links or links located in domains traversed by $r\langle n_{p_1}, n_{p_m}\rangle$. Yet, NM can only identify interdomain links and intradomain path segments of $r\langle n_{p_1}, n_{p_m}\rangle$. Hence, failures located in domains traversed by $r\langle n_{p_1}, n_{p_m}\rangle$ are represented by NM in its FPM using proxy nodes, which are created for each domain as follows:

- For every ingress gateway node in $\mathcal{D}_i$, $I^i_{l,i}$, we create node $P\langle I^i_{l,i}, *\rangle$ that represents the set of intra-$\mathcal{D}_i$ paths that begin in node $I^i_{l,i}$. We write that $P\langle I^i_{l,i}, *\rangle = \{r\langle I^i_{l,i}, n_r\rangle | n_r \in \mathcal{D}_i\}$.
- For every egress gateway node in $\mathcal{D}_i$, $E^i_{i,k}$, we create node $P\langle *, E^i_{i,k}\rangle$ that represents the set of intra-$\mathcal{D}_i$ paths that end in node $E^i_{i,k}$. We write that $P\langle *, E^i_{i,k}\rangle = \{r\langle n_r, E^i_{i,k}\rangle | n_r \in \mathcal{D}_i\}$.
- Moreover, for each pair of gateway nodes $I^i_{l,k}$ and $E^i_{l,k}$, we create node $P\langle I^i_{l,k}, E^i_{l,k}\rangle$ that represents the intra-$\mathcal{D}_i$ path segment $r\langle I^i_{l,k}, E^i_{l,k}\rangle$, i.e., $P\langle I^i_{l,k}, E^i_{l,k}\rangle = \{r\langle I^i_{l,k}, E^i_{l,k}\rangle\}$.

Besides, the FPM of NM includes fault nodes corresponding to interdomain links. In the FPM of NM, symptom

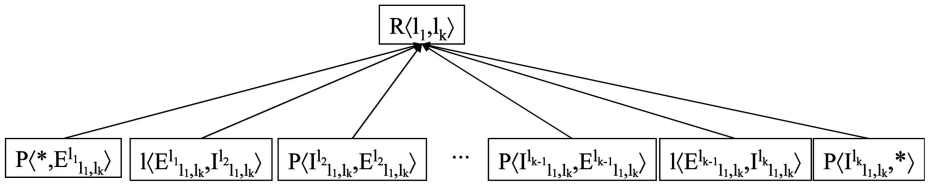


Fig. 4. Construction of the FPM for NM including P-nodes representing domains.

node $r\langle l_1, l_k\rangle$ is connected to nodes $P\langle *, E^{l_1}_{l_1,l_k}\rangle$, $l\langle E^{l_1}_{l_1,l_k}, I^{l_2}_{l_1,l_k}\rangle$, $P\langle I^{l_2}_{l_1,l_k}, E^{l_2}_{l_1,l_k}\rangle, \ldots, l\langle E^{l_{k-1}}_{l_1,l_k}, I^{l_k}_{l_1,l_k}\rangle$, and $P\langle I^{l_k}_{l_1,l_k}, *\rangle$ (Fig. 4). Overall, the FPM of NM contains multiple such symptom nodes for all pairs of domains in $\mathcal{N}$. These symptom nodes are connected to overlapping sets of fault and P-nodes. Thus, the FPM of NM is a connected bipartite graph.

The final step in the creation of the FPM for NM is assigning prior failure probabilities to P-nodes and conditional probabilities to causal edges between P-nodes and symptom nodes. Conditional probabilities between P-nodes and symptom nodes do not have any intuitive interpretation. The approach chosen in this paper assigns all conditional probabilities between P-nodes and symptom nodes to 1. The strength with which faults located in subdomains influence failures of paths that span multiple domains is modeled using prior failure probabilities assigned to P-nodes. P-nodes do not have real-life correspondents, either, since they are synthetic elements. As a result, their prior failure probabilities cannot be either assigned by an expert or learned through system observation, as it is the case with ordinary fault nodes. In addition, P-nodes represent failures located in other domains and their prior failure probabilities change during the process of fault localization. Thus, prior failure probabilities associated with P-nodes $P\langle I^i_{l,i}, *\rangle$, $P\langle *, E^i_{i,k}\rangle$, and $P\langle I^i_{l,k}, E^i_{l,k}\rangle$ in the FPM of NM must be calculated by the multidomain technique based on the state of the fault localization process in $\mathcal{D}_i$. Since this state is not accessible to NM, the probabilities have to be calculated by DM_i. The process in which it is done is discussed in Section 5.

4.2 Fault Propagation Model of a DM

As stated at the beginning of Section 4, the FPM built by DM_i includes all intra-$\mathcal{D}_i$ paths and links, i.e., all the information contained in the centralized model of $\mathcal{D}_i$. Such a model is sufficient for the diagnosis of symptoms observed in $\mathcal{D}_i$, but it is not sufficient for the diagnosis of symptoms DM_i receives from NM. In the hierarchical fault management solution presented in this paper, diagnosis of a path failure is delegated up and down the management hierarchy until managers of all domains traversed by the path are notified. In particular, NM may delegate to DM_i a part of a task involved in the diagnosis of path $r\langle n_{p_1}, n_{p_m}\rangle$ that traverses $\mathcal{D}_i$. In this case, DM_i will be notified about a failure of its intradomain path that constitutes the intra-$\mathcal{D}_i$ path segment of $r\langle n_{p_1}, n_{p_m}\rangle$. Observe that this notification does not mean that the intra-$\mathcal{D}_i$ path has necessarily failed. It only indicates a possibility of this segment's failure, since $r\langle n_{p_1}, n_{p_m}\rangle$ could have been caused by its path-segment or link that is not located in domain $\mathcal{D}_i$. Thus, symptoms received by DM from NM are typically associated with a high degree of uncertainty, i.e., they are likely to be spurious. In our previous work [20], we showed that spurious symptoms may be modeled by introducing extra fault nodes.

Let $r\langle n_r, n_t\rangle$ be an intra-$\mathcal{D}_i$ symptom received by DM_i from NM in the process of diagnosing a failure of interdomain path $r\langle n_{p_1}, n_{p_m}\rangle$. To model the possibility that $r\langle n_r, n_t\rangle$ is spurious in the FPM of DM_i, we create a proxy fault node, called $\tilde{P}$-node, that represents all possible causes

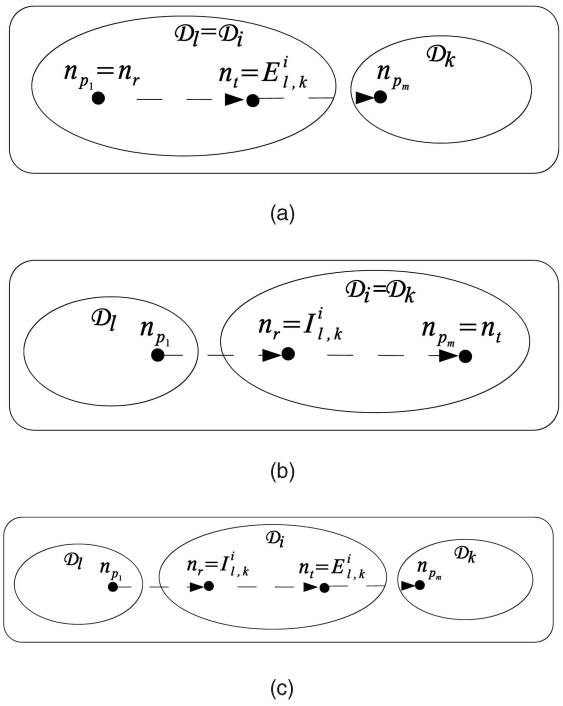


Fig. 5. Definition of proxy nodes in the FPM of DM_i. (a) $i = l$. (b) $i = k$. (c) $i \neq k$ and $i \neq l$.

of $r\langle n_r, n_t\rangle$ that are not located in $\mathcal{D}_i$. Observe that, since path $r\langle n_r, n_t\rangle$ constitutes a segment of an interdomain path, at least one of nodes n_r, n_t is a gateway node in $\mathcal{D}_i$. Let l and k be identifiers of domains that contain nodes n_{p_1} and n_{p_m}, respectively. Consider three possible cases:

- $i = l$, and in consequence, $n_t = E^i_{l,k} = E^i_{i,k}$ (Fig. 5a). We create node $\tilde{P}\langle *, E^i_{i,k}\rangle$ and connect it to $r\langle n_r, n_t\rangle$.
- $i = k$, and in consequence, $n_r = I^i_{l,k} = I^i_{l,i}$ (Fig. 5b). We create node $\tilde{P}\langle I^i_{l,i}, *\rangle$ and connect it to $r\langle n_r, n_t\rangle$.
- $i \neq l$ and $i \neq k$, and in consequence, $n_r = I^i_{l,k}$ and $n_t = E^i_{l,k}$ (Fig. 5c). We create node $\tilde{P}\langle I^i_{l,k}, E^i_{l,k}\rangle$ and connect it to $r\langle n_r, n_t\rangle$.

Observe that each $\tilde{P}$-node $\tilde{P}\langle *, E^i_{i,k}\rangle$, $\tilde{P}\langle I^i_{l,i}, *\rangle$, or $\tilde{P}\langle I^i_{l,k}, E^i_{l,k}\rangle$ in the FPM of DM_i corresponds to P-node $P\langle *, E^i_{i,k}\rangle$, $P\langle I^i_{l,i}, *\rangle$, or $P\langle I^i_{l,k}, E^i_{l,k}\rangle$, respectively, in the FPM of NM. Similar to what we did in the case of NM, conditional probabilities on edges between $\tilde{P}$-nodes and symptom nodes in the FPM of DM_i are set to 1, while prior failure probabilities assigned to $\tilde{P}$-nodes in the FPM of DM_i are calculated by NM and sent to DM_i together with reported symptoms. In fact, in the FPM of DM_i, $\tilde{P}$-nodes can be created dynamically when corresponding symptoms are reported by the NM.

Illustrative examples of FPMs built by the NM and DMs as well as an example of the algorithm usage are given in [17].

5 Multidomain Fault Localization Algorithm

In this section, we present an outline of a multidomain fault localization algorithm, which may be refined to be used

with various inferencing mechanisms. In the pseudocode in Section 5.2, parts of the algorithm that need to be specialized for different probabilistic reasoning mechanisms are underlined.

The multidomain fault localization algorithm proceeds in three phases performed by every DM and NM: 1) model initialization, 2) symptom analysis, and 3) fault selection. In the initialization phase (see Algorithm MDA), the model is reset by assigning prior failure probabilities to proxy nodes. In our implementation, these probabilities are set to 0 in the FPM of NM, while in the FPM of DM, no $\tilde{P}$-nodes exist at the beginning and, therefore, no assignment is needed.

Symptom-analysis and fault-selection phases progress by traversing the hierarchy of managers in a bottom-up or top-down manner, where *bottom-up* and *top-down* indicate the direction of the information flow. Processing performed by DM or NM is triggered by a symptom arrival or by a message received from NM or DM, respectively. For the clarity of presentation, in the pseudocode of Algorithm MDA, we use function calls to indicate the exchange of information between managers. In the distributed implementation, the function calls should be implemented by asynchronous message exchange rather than RPC-style invocations.

5.1 Symptom Analysis Phase

The symptom analysis phase is executed for every received alarm that indicates a failure of an end-to-end path between two nodes. This alarm can be received either by NM or DM. DM can start the symptom analysis only if the entire failed path belongs to its domain. If DM is not able to diagnose the symptom, it forwards it to NM, which initiates the symptom diagnosis (function *analyze_internal*).

5.1.1 Symptom Processing by NM

Suppose NM diagnoses symptom $r\langle n_{p_1}, n_{p_m}\rangle$ reported to it from one of the DMs. NM maps $r\langle n_{p_1}, n_{p_m}\rangle$ into $R\langle l_1, l_k\rangle$ in its FPM, such that $r\langle n_{p_1}, n_{p_m}\rangle \in R\langle l_1, l_k\rangle$ (or $n_{p_1} \in \mathcal{D}_{l_1}$ and $n_{p_m} \in \mathcal{D}_{l_k}$). Then, it splits the failed path into its intradomain path segments and interdomain links, i.e., into a sequence $r\langle n_{p_1}, E^{l_1}_{l_1,l_k}\rangle$, $l\langle E^{l_1}_{l_1,l_k}, I^{l_2}_{l_1,l_k}\rangle$, $r\langle I^{l_2}_{l_1,l_k}, E^{l_2}_{l_1,l_k}\rangle, \ldots, l\langle E^{l_{k-1}}_{l_1,l_k}, I^{l_k}_{l_1,l_k}\rangle$, and $r\langle I^{l_k}_{l_1,l_k}, n_{p_m}\rangle$. Possible failures of one or more links within segments $r\langle n_{p_1}, E^{l_1}_{l_1,l_k}\rangle$, $r\langle I^{l_2}_{l_1,l_k}, E^{l_2}_{l_1,l_k}\rangle, \ldots, r\langle I^{l_k}_{l_1,l_k}, n_{p_m}\rangle$ are then reported as symptoms $s_1, s_2, \ldots, s_k$ to $\mathrm{DM}_{l_1}, \mathrm{DM}_{l_2}, \ldots, \mathrm{DM}_{l_k}$, respectively. Since high uncertainty is associated with these symptoms, the NM needs to calculate the probability with which the symptoms should be considered spurious by DMs. Note that, in the FPM of DM_{l_j}, all causes of symptom s_j reported by NM that are not located in $\mathcal{D}_{l_j}$ are represented by a $\tilde{P}$-node that is attached to node s_j. Let us label this node $\tilde{P}(s_j)$. Thus, the NM needs to calculate the prior probability $p(\tilde{P}(s_j))$ that should be associated with $\tilde{P}(s_j)$ in the FPM of DM_i. Suppose that s_j corresponds to $r\langle n_r, n_t\rangle$. Then, the probability that s_j is spurious is obtained as follows (recall that underlined font represents functions that are different depending on the probabilistic inference mechanism used as a basis of the algorithm):

$$p_s(r\langle n_t n_r\rangle) = \prod_{P \epsilon \mathbb{IP}(r\langle n_t, n_r\rangle)} \underline{Prob\{\neg P\}},$$

$$\mathbb{IP}(r\langle n_t, n_r\rangle) = \begin{cases} \{P\langle n_r, n_t\rangle, & P\langle *, n_t, \rangle, P\langle n_r, *\rangle\} \\ & \text{if } n_r \text{ and } n_t \text{ are ingress and} \\ & \text{egress gateways} \\ \{P\langle *, n_t\rangle\} & \text{if } n_r \text{ is an ingress gateway} \\ \{P\langle n_r, *\rangle\} & \text{if } n_t \text{ is an egress gateway.} \end{cases} \quad (1)$$

After calculating $p_s(s_j)$, NM delegates the diagnosis of s_j to DM_{l_j}, for $j = 1 \ldots k$, by invoking *analyze_external*. To limit duplicate delegations of the same symptom to DM_{l_j}, NM marks nodes as either `UNOBSERVED` or `OBSERVED_INTERNAL`. While analyzing $r\langle n_{p_1}, n_{p_m}\rangle$, when $r\langle l_1, l_k\rangle$ is marked `OBSERVED_INTERNAL`, the NM does not delegate symptoms to DM_{l_j}s for $j = 2 \ldots k-1$. It does, however, delegate the analysis to DM_{l_1} and DM_{l_k}, since paths represented by $r\langle l_1, l_k\rangle$ differ in their segments located in $\mathcal{D}_{l_1}$ and $\mathcal{D}_{l_k}$.

When DMs complete the analysis of symptoms that have been delegated to them by NM, they return the values of corresponding $p(P^j_{l_1,l_k})$s, where $P^j_{l_1,l_k}$ is the P-node representing $\mathcal{D}_j$ that is connected to $r\langle l_1, l_k\rangle$ in the FPM of NM. Then, the NM updates its FPM and incorporates the changed values of $p(P^j_{l_1,l_k})$s in its state of fault localization. Finally, NM analyzes $r\langle l_1, l_k\rangle$ using the inference mechanism of NM (function *inference*).

5.1.2 Symptom Processing by DM

DM_i may start the processing of symptom $r\langle n_{p_1}, n_{p_m}\rangle$ as a result of two events: 1) it may observe a failure of path $r\langle n_{p_1}, n_{p_m}\rangle$, whose nodes all belong to $\mathcal{D}_i$, or 2) the symptom may be delegated to DM_i by NM. In the former case, $r\langle n_{p_1}, n_{p_m}\rangle$ is an internal symptom; in the latter case, it is called an external symptom. Internal symptoms are considered more significant, since they cannot be explained by faults located outside DM_i. However, in the absence of internal symptoms, the external ones help the DM_i make correct diagnoses. To distinguish between different observations of the same symptom, DM_i marks symptom nodes as either `UNOBSERVED`, `OBSERVED_INTERNAL`, or `OBSERVED_EXTERNAL` when they are not processed, processed as a result of internal observation, or processed as a result of a delegation by NM, respectively.

Internal symptoms are processed by function *analyze_internal*. First, the association between the observed symptom and its $\tilde{P}$-node (if one exists) is removed, as the symptom can no longer be explained by external causes. Then, a probabilistic inference mechanism chosen for this DM is used to analyze the symptom.

The processing of external symptoms is done by function *analyze_external*. Assume that $r\langle n_{p_1}, n_{p_m}\rangle$ has been delegated to DM_i as a result of a failure of an end-to-end path between domains $\mathcal{D}_l$ and $\mathcal{D}_k$. DM_i also receives two parameters from NM: $P^i_{l,k}$ and p_s. Recall that $P^i_{l,k}$ is a description of a P-node that represents $\mathcal{D}_i$ and is connected to node $r\langle l, k\rangle$ in the FPM of NM, and p_s is the probability with which $r\langle n_{p_1}, n_{p_m}\rangle$ should be considered spurious by DM_i. Recall also that, for each P-node that represents DM_i, the FPM of NM, there is a corresponding $\tilde{P}$-node in the FPM of DM_i. Let us denote by

$\tilde{P}^i_{l,k}$ the $\tilde{P}$-node in the FPM of DM_i that corresponds to P-node $P^i_{l,k}$ in the FPM of NM. At the beginning of function *analyze_external*, $\tilde{P}^i_{l,k}$ must be connected to $r\langle n_{p_1}, n_{p_m}\rangle$ if this has not been done before, and it must be labeled with the prior failure probability of p_s.

If the symptom has been previously analyzed, DM_i takes no further action and returns the stored value of $p(P^i_{l,k})$. Otherwise, a probabilistic reasoning mechanism is used to analyze $r\langle n_{p_1}, n_{p_m}\rangle$ and $r\langle n_{p_1}, n_{p_m}\rangle$ is marked as `OBSERVED_EXTERNAL`. Finally, $p(P^i_{l,k})$ is calculated as follows:

$$p(P^i_{l,k}) = \begin{cases} 0 & \mathcal{S}^i_{l,k} = \emptyset \\ \prod_{s_i \in \mathcal{S}^i_{l,k}} bel(s_i) & \text{otherwise,} \end{cases} \tag{2}$$

$$\mathcal{S}^i_{l,k} = \{r\langle n_r, n_t\rangle | r\langle n_r, n_t\rangle \text{ is connected to } \tilde{P}^i_{l,k} \text{ and } r\langle n_r, n_t\rangle \text{ is not } \texttt{UNOBSERVED}\}, \tag{3}$$

$$bel(s_i) = \begin{cases} 1 \text{ if } s_i \text{ is } \texttt{OBSERVED\_INTERNAL} \\ 1 - \prod_{f_j \in \mathcal{F}} (1 - p(s_i|f_j)\underline{Prob\{f_j\}}) \\ \text{otherwise.} \end{cases} \tag{4}$$

Intuitively, function $p(P^i_{l,k})$ expresses the probability that all failures of intra-$\mathcal{D}_i$ path segments represented by $P^i_{l,k}$ in the FPM of NM that have been reported by NM can be explained by DM_i. Clearly, if a failure reported by NM has also been observed by DM_i as an internal symptom, then the probability that it was caused in $\mathcal{D}_i$ is 1. Otherwise, DM_i needs to calculate the probability that a failure reported by NM but not observed as an internal symptom was caused by one or more faults in $\mathcal{D}_i$. Fault probabilities used in this case are obtained based on symptom diagnosis performed by DM_i up to this point. When no symptoms have been reported to DM_i, function $p(P^i_{l,k})$ is 0.

5.2 Fault Selection Phase

Fault selection phase is a cooperative process initiated by NM, which first obtains from DMs the prior failure probabilities associated with proxy nodes in its FPM, and then calculates spurious symptom probabilities that are assigned to proxy nodes in the FPMs of DMs.

Afterward, DMs and NM can proceed independently of one another to update their fault localization states and choose the most likely hypothesis.

Algorithm MDA: Multidomain algorithm
Initialization:
NM: FOR *every* $P^i_{l,k}$ DO $p(P^i_{l,k}) = 0$ DONE
Symptom analysis phase:
DM_i: FOR *every observed symptom* $r\langle n_{p_1,n_{pm}}\rangle$ DO
IF $d(n_{p_1}) = i$ AND $d(n_{p_m}) = i$ THEN
analyze_internal$(r\langle n_{p_1}, n_{p_m}\rangle)$
ELSE NM$\rightarrow$ *analyze_internal*$(r\langle n_{p_1}, n_{p_m}\rangle)$
DONE

NM: FOR *every observed symptom* $r\langle n_{p_1,n_{pm}}\rangle$ DO
analyze_internal$(r\langle n_{p_1}, n_{p_m}\rangle)$ DONE

DM_i: FUNCTION *analyze_internal*$(r\langle n_{p_1}, n_{p_m}\rangle)$
IF $r\langle n_{p_1}, n_{p_m}\rangle$ *is not marked* `OBSERVED_INTERNAL` THEN
remove association between $r\langle n_{p_1}, n_{p_m}\rangle$ *and any* $\tilde{P}$*-nodes*
mark $r\langle n_{p_1}, n_{p_m}\rangle$ *as* `OBSERVED_INTERNAL`
inference$(r\langle n_{p_1}, n_{p_m}\rangle)$
END

NM: FUNCTION *analyze_internal*$(r\langle n_{p_1}, n_{p_m}\rangle)$
map $r\langle n_{p_1}, n_{p_m}\rangle$ *to* $R\langle l_1, l_k\rangle$ *such that* $r\langle n_{p_1}, n_{p_m}\rangle \in R\langle l_1, l_k\rangle$
transform $r\langle n_{p_1}, n_{p_m}\rangle$ *into* $r\langle n_{p_1}, E^{l_1}_{l_1,l_k}\rangle$, $l\langle E^{l_1}_{l_1,l_k}, I^{l_2}_{l_1,l_k}\rangle$, $r\langle I^{l_2}_{l_1,l_k}, E^{l_2}_{l_1,l_k}\rangle, \ldots, l\langle E^{l_{k-1}}_{l_1,l_k}, I^{l_k}_{l_1,l_k}\rangle$, $r\langle I^{l_k}_{l_1,l_k}, n_{p_m}\rangle$
determine proxy nodes connected to $r\langle l_1, l_k\rangle$:
$P^{l_1}_{l_1,l_k} = P\langle *, E^{l_1}_{l_1,l_k}\rangle, P^{l_2}_{l_1,l_k} = P\langle I^{l_2}_{l_1,l_k}, E^{l_2}_{l_1,l_k}\rangle, \ldots, P^{l_k}_{l_1,l_k} = P\langle I^{l_k}_{l_1,l_k}, *\rangle$
set $s_1 = r\langle n_{p_1}, E^{l_1}_{l_1,l_k}\rangle$, $s_2 = r\langle I^{l_2}_{l_1,l_k}, E^{l_2}_{l_1,l_k}\rangle, \ldots, s_k = r\langle I^{l_k}_{l_1,l_k}, n_{p_m}\rangle$
FOR $1 \le j \le k$ DO
IF $r\langle l_1, l_k\rangle$ *is marked* `UNOBSERVED` OR $j = 1$ OR $j = k$ THEN
$p(P^{l_j}_{l_1,l_k}) = \mathrm{DM}_{l_j} \rightarrow$ *analyze_external*$(s_j, P^{l_j}_{l_1,l_k}, p_s(s_j))$
DONE
IF $r\langle l_1, l_k\rangle$ *is not marked* `OBSERVED_INTERNAL` THEN
update model
mark $r\langle l_1, l_k\rangle$ *as* `OBSERVED_INTERNAL`
inference$(r\langle l_1, l_k\rangle)$
END

DM_i: FUNCTION *analyze_external*$(s_r, P^i_{l,k}, p_s)$
set $p(\tilde{P}^i_{l,k}) = p_s$
IF s_r *is not marked* `UNOBSERVED` THEN *return* $p(P^i_{l,k})$
ELSE *update model*, *run* *inference*(s_r), AND RETURN $p(P^i_{l,k})$
END
Fault selection phase:
NM: FOR *every* $P^i_{l,k}$ DO *obtain* $p(P^i_{l,k})$ *from* DM_i AND *update model*
FOR *every* $P^i_{l,k}$ DO *send* $Prob\{\neg P^i_{l,k}\}$ *to* DM_i
FOR *every* DM_i DO *obtain the most likely set of faults from* DM_i
obtain the most likely set of faults in NM

5.3 Signaling Overhead

The signaling overhead of MDA results from the exchange of symptoms and probabilities between NM and DMs and is therefore related to the number of probability values produced in every phase of the algorithm. In the entire algorithm, the messaging overhead is $\mathcal{O}(\max(|\mathcal{S}_O|, |\mathcal{N}|^3))$.

6 Multidomain Fault Localization Using Belief Networks

In this section, we introduce a refinement of MDA resulting from adopting belief updating [20] as an inferencing mechanism. In our previous work [20], we have shown that the FPM for end-to-end service failure diagnosis can be interpreted as a belief network [16], in which every node represents a binary valued random variable. Symptom observation is represented by assigning 1 to the corresponding belief network node and constitutes a part of *evidence*. In this context, the fault localization problem is to find the most probable assignment of *link* nodes given the observed

evidence. In [20], we have proposed a centralized algorithm (labeled BUA) that uses iterative belief updating as an inferencing mechanism.

Iterative belief updating, proposed by Pearl [16] for singly-connected belief networks, utilizes a message-passing schema in which the belief network nodes exchange λ and π messages that encode various conditional probabilities [16]. Belief network node X receives messages λ and π from its children and parents, respectively. Based on these messages, it calculates new messages λ and π that it sends to its parents and children, respectively. Moreover, node X calculates function $bel : \{0,1\} \rightarrow [0,1]$, where $bel(x)(x \in \{0,1\})$ represents the probability that $X = x$ given the observed evidence. The belief-updating algorithm in polytrees starts from an evidence node and propagates the changed belief along the graph edges by computing λ and π messages. In the application to belief networks with undirected loops, several such propagations are performed to enforce the algorithm's convergence.

In Algorithm BUA, which is based on iterative belief updating, one traversal of the entire belief network is performed for every observed symptom. The computational complexity of the algorithm is bounded by $\mathcal{O}(|\mathcal{S}_O|^2|\mathcal{F}|)$ [20]. In particular, in the application to the problem of end-to-end service failure diagnosis, it is bounded by $\mathcal{O}(n^5)$, where n denotes the number of intermediate network nodes, such as routers or bridges. Instead of iterative belief updating, one could consider a more advanced inferencing mechanism such as the forward/backward and Viterbi algorithms [13].

In this section, we refine MDA for use with the inferencing mechanism of BUA. We label the resultant algorithm MD-BUA. The refinement consists in defining functions $inference(s_i)$, $Prob(f_j)$, and $Prob(\neg f_j)$.

Using the probabilistic reasoning mechanism of BUA, probability $Prob\{\neg P^i_{l,k}\}$ used in (1) may be expressed using λ messages received by node $P^i_{l,k}$ from its children nodes in the belief network that constitutes the FPM of NM. Let $\lambda_{P^i_{l,k}}(x)$ indicate a product of messages λ received by $P^i_{l,k}$ from its children. Using this information, we can derive $Prob\{\neg P^i_{l,k}\}$ as follows:

$$Prob\{\neg P^i_{l,k}\} = \frac{\lambda_{P^i_{l,k}}(0)}{\lambda_{P^i_{l,k}}(1)}.$$

Probability $Prob\{f_i\}$ from (4) is calculated as follows: Let $\lambda_{f_i}(x)$ indicate a product of messages λ sent to node f_i by its children and α be a normalizing constant. We calculate $Prob\{f_i\}$ as follows:

$$Prob\{f_i\} = \alpha\lambda_{f_i}(1)p(f_i).$$

Function $inference(s_i)$ is the symptom analysis procedure of BUA [20].

We can express the computational complexity of fault localization performed by NM as $\mathcal{O}(|\mathcal{N}|^5))$. The computational complexity of fault localization performed by DM_i is $\mathcal{O}(|\mathcal{D}_i|^5)$.

7 Multidomain Incremental Hypothesis Updating

In this section, we introduce a refinement of MDA, MD-IHUA resulting from adopting belief updating [19] as an inferencing mechanism.

Incremental hypothesis updating [19] (IHU) creates a set of most likely hypotheses and makes all of them available to a system administrator on a continuous basis. 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. After the ith symptom analysis, the hypotheses are ranked using belief metric b_i. The algorithm proceeds in an event-driven and incremental fashion. The execution triggered by the ith 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. After the ith 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 $s_k \in \mathcal{S}_{O,i} = \{s_1, \ldots, s_i\}$. The upper bound on the worst case computational complexity of the resultant algorithm is $\mathcal{O}(|\mathcal{S}_O||\mathcal{F}|^2)$ [19]. In the application to end-to-end service failure diagnosis in an n-node network, the worst case computational complexity of IHU is $\mathcal{O}(n^4)$.

Compared to BUA, IHU is equally accurate, but it proved much more efficient, allowing the isolation of up to four simultaneous faults in a 100-node network in less than 10 seconds. Using a 10-second fault-localization time as an admissibility criterion, the admissible network size for BUA, in a similar scenario, is 35. In addition, while both algorithms analyze symptoms in an event-driven manner, IHU is also incremental at any time, offering a complete solution to the symptoms observed thus far.

In the context of the incremental technique, $Prob\{\neg P^i_{l,k}\}$ from (1) calculates the conditional probability that faults represented by $P^i_{l,k}$ in the FPM of NM did not occur, given the observed evidence. This probability may be expressed as follows:

$$Prob\{\neg P^i_{l,k}\} = 1 - \sum_{h \in \mathcal{H}_j | P^i_{l,k} \in h} b_j(h).$$

$Prob\{f_k\}$ is calculated by summing the belief metric associated with hypotheses that contain f_k.

$$Prob\{f_k\} = \sum_{h \in \mathcal{H}_j | f_k \in h} b_j(h).$$

Function $inference(s_i)$ is a symptom analysis procedure of IHU [19].

The computational costs incurred by NM in the symptom analysis phase and fault selection phase are $\mathcal{O}(\min(|\mathcal{S}_O||\mathcal{N}|^2, |\mathcal{N}|^4))$ and $\mathcal{O}(|\mathcal{N}|^3)$, respectively. Therefore, we can express the computational complexity of fault localization performed by NM as $\mathcal{O}(|\mathcal{N}|^4)$. The computational costs of the symptom analysis phase and fault selection phase of DM_i are $\mathcal{O}(\min(|\mathcal{S}_O||\mathcal{D}_i|^2, |\mathcal{D}_i|^4))$ and $\mathcal{O}(|\mathcal{D}_i|^3)$. Thus, the computational complexity of fault localization performed by DM_i is $\mathcal{O}(|\mathcal{D}_i|^4)$.

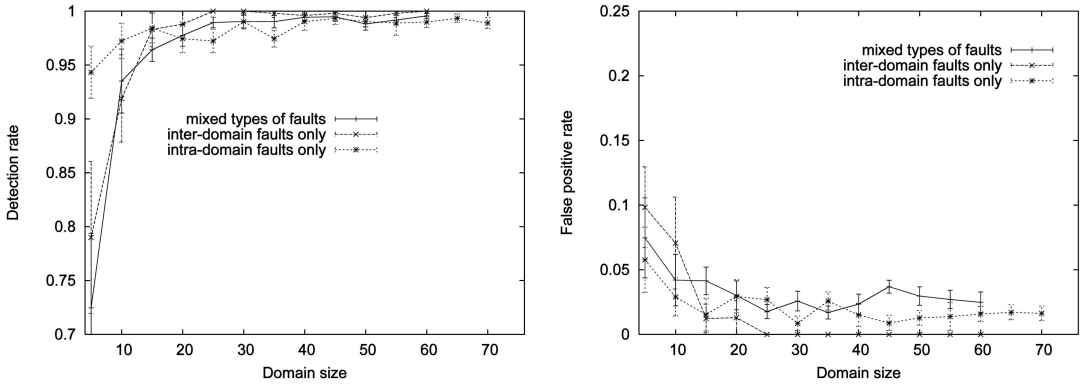


Fig. 6. Accuracy of MD-BUA in a 10-domain network. (a) Detection rate. (b) False positive rate.

8 Simulation Study

In this section, we evaluate the performance of MD-BUA and MD-IHUA through simulation. Our purpose is to assess the accuracy of both algorithms in a multidomain communication network. The study uses sets of fault localization scenarios in which faults and symptoms are randomly generated based on the conditional probability distribution that describes nondeterministic causal relationships between faults and symptoms in a real-life system. In the distribution, for every fault f and symptom s, we set $p(s|f) = 0$ if the end-to-end service whose failure is represented by s is not provided using the node-to-node service whose failure is represented by f. Otherwise, $0 < p(s|f) \leq 1$.

In this section, we first describe the design of the simulation experiments and then present and explain the results of the study.

8.1 Simulation Design

The simulation study presented in this section uses network topologies similar to those of the Internet. The generation of random graphs resembling the topology of real-life networks has been a widely studied research area [1], [2], [3], [6], [9], [15]. Out of several topology generators available, we choose one based on the Barabasi-Albert power-law model [3] because its implementation is available in the public domain and because topologies built based on this model have been shown to be reasonably representative of the Internet topology [5]. We use an implementation of the Barabasi-Albert model provided by the BRITE generator [14], which is capable of generating hierarchical network topologies: AS-level and router-level ones.

The simulation model of the study created two-level hierarchical topologies using N and n to denote the number of domains and the number of routers in every domain, respectively. To investigate the impact of network topology, we use $N = 10$ and $N = 50$, and we vary n from 5 to 75. Typically, we choose a maximum domain size such that the fault localization time of a single scenario does not exceed 10 s. Our experiments assume that the observation of the system state is accurate. Clearly, the accuracy of fault localization may be diminished by lost and spurious symptoms. In [20], [19], we have shown techniques that allow us to deal with such noise in symptom data. The same techniques apply to the multidomain solution.

Using the topology generator, we create a random network composed of N domains and n nodes in each domain. We determine routes between any source and destination using the shortest-path policy for intradomain routes. We choose interdomain routes such that the number of visited domains is minimized. Then, we generate prior failure probabilities for interdomain and intradomain links, which are uniformly distributed over the range [0.0001, 0.001]. For each intradomain link l and path p, we randomly choose the probability that p fails if l fails from set {0.25, 0.5, 0.75}. For each interdomain path p, we assume that, if any path segment or link involved in p fails, then p fails as well. Consequently, in the FPM of NM, the conditional probabilities are all equal to 1. Furthermore, we randomly generate a subset of symptoms observable in every domain to include 50 paths. The observability ratio for interdomain paths is 2; the observability ratio [20], [19] is a measure of the system instrumentation degree. By using it, we recognize that only some failure conditions are monitored by the management systems. As a result, a manager can see only a fraction of failures that exist in the system it manages.

Test scenarios are generated using the same conditional probability distribution that is used by the managers in their FPMs. This technique of generating scenarios assumes that the fault propagation model accurately represents relationships among faults and symptoms. However, from our previous studies, we know that the fault localization techniques considered in this paper are accurate even if the FPM they are executed on is approximate.

We distinguish three types of experiments: those involving only intradomain link failures, those involving only interdomain link failures, and those involving both types of failures. In every study, two performance metrics are calculated: detection rate, DR, defined as a percentage of faults occurring in the network which are isolated by the technique, and false positive rate, FPR, defined as a percentage of faults reported by the technique that are not occurring in the network [20], [19].

8.2 Experimental Results

In Figs. 6a and 6b, we show the accuracy of MD-BUA applied to fault localization in a 10-domain network in which each domain is composed of up to 70 nodes. Thus, the entire network consists of up to 700 nodes. Figs. 7a and

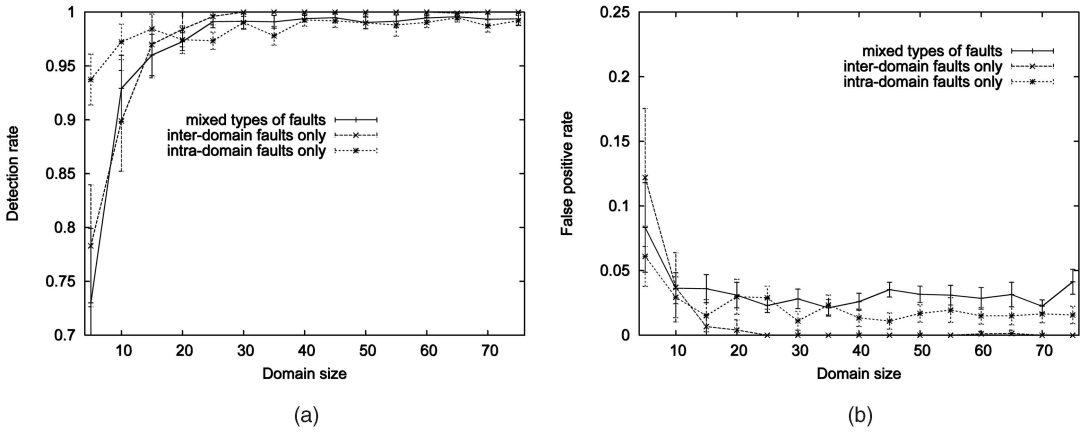


Fig. 7. Accuracy of MD-IHUA in a 10-domain network. (a) Detection rate. (b) False positive rate.

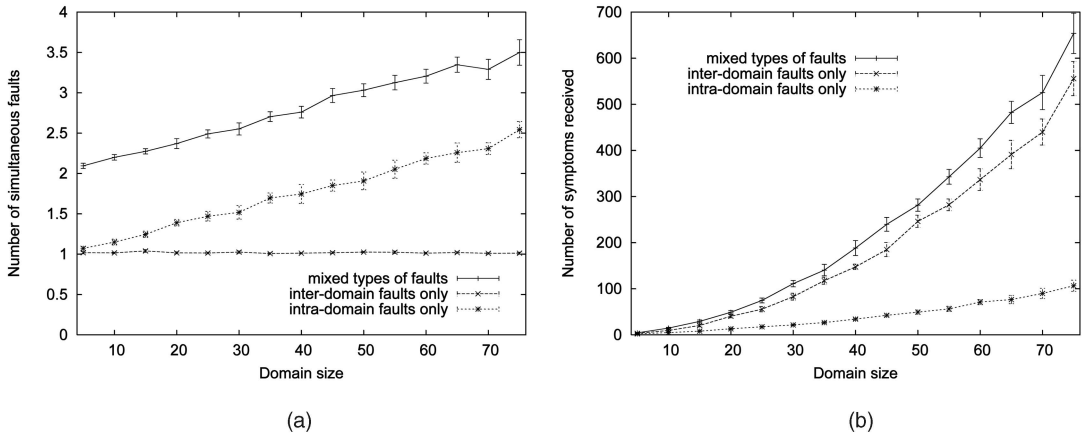


Fig. 8. Average number of faults and symptoms generated in experiment scenarios for a 10-domain network. (a) Average number of simultaneous faults. (b) Average number of observed symptoms.

7b present the results of the same experiment executed using MD-IHUA.

The figures compare the accuracy achievable in scenarios involving only interdomain, only intradomain, and both types of faults. Clearly, the mixed-failure scenarios are the most difficult to diagnose since they always involve at least two concurrent faults and the interpretation of their symptoms, which may overlap, leads to ambiguity. This difficulty results in a lower fault-localization accuracy of mixed-fault scenarios compared to that of other types of scenarios, which is conspicuous in networks of small size. Scenarios involving only interdomain symptoms are the easiest to solve, as the number of suspect faults is usually small compared to the amount of evidence available even with the very small observability ratio we have chosen. In addition, in our two-level setup, NM does not receive any ambiguous information (from a higher-level manager). Henceforth, it knows that all symptoms have to be explained in its domain. Intradomain scenarios are similar to mixed scenarios because both interdomain and intradomain symptoms may be generated as a result of intradomain faults. Thus, in intradomain scenarios, domain managers have to deal with the same level of ambiguity as is the case with mixed-fault scenarios.

To understand the difference among these three types of experiments, it is useful to compare the numbers of simultaneous faults and symptoms generated in each experiment, which are presented in Figs. 8a and 8b. These figures show that, in interdomain scenarios, the number of faults existing in the network is small (in most experiments, only one fault was present) and does not change as the domain size increases, while the number of symptoms observed grows fast with the growing domain size. When the number of observed symptoms is big and the number of faults to isolate is small, fault localization may be performed with very high accuracy. Naturally, a big number of symptoms to diagnose increases the fault localization time. In intradomain and mixed-fault scenarios, increasing the domain size also increases the frequency of multifault scenarios.

Figs. 9a and 9b compare the fault localization times of MD-BUA and MD-IHUA. The fault localization time is defined as the time needed to analyze all symptoms received in the considered fault localization scenario and to propose the most probable hypothesis. It is measured under the assumption that each symptom is available to the fault localization process as soon as the analysis of the previous symptom has completed. Thus, this measurement ignores the impact of symptom latencies. As expected, MD-IHUA offers a much better performance than MD-BUA, which is due to its lower computational complexity. The difference in performance among mixed, intradomain, and

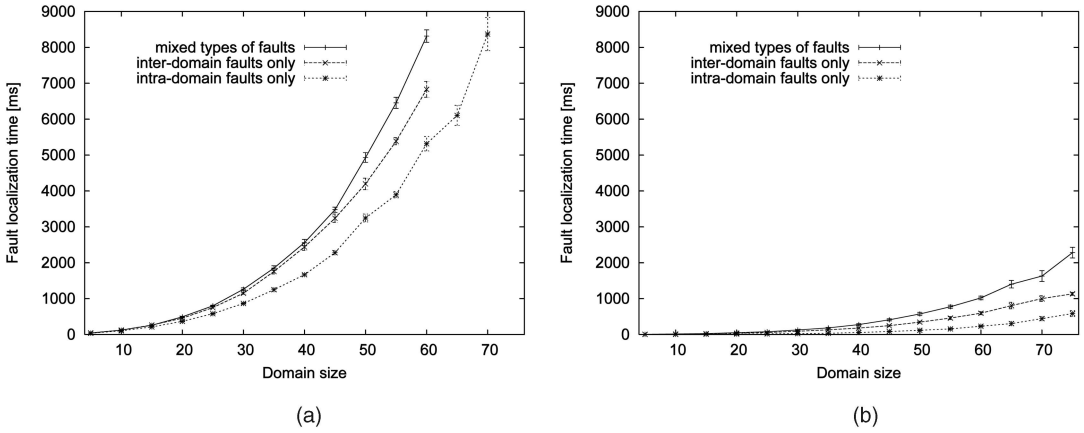


Fig. 9. Fault localization time in a network composed of 10 domains. (a) MD-BUA. (b) MD-IHUA.

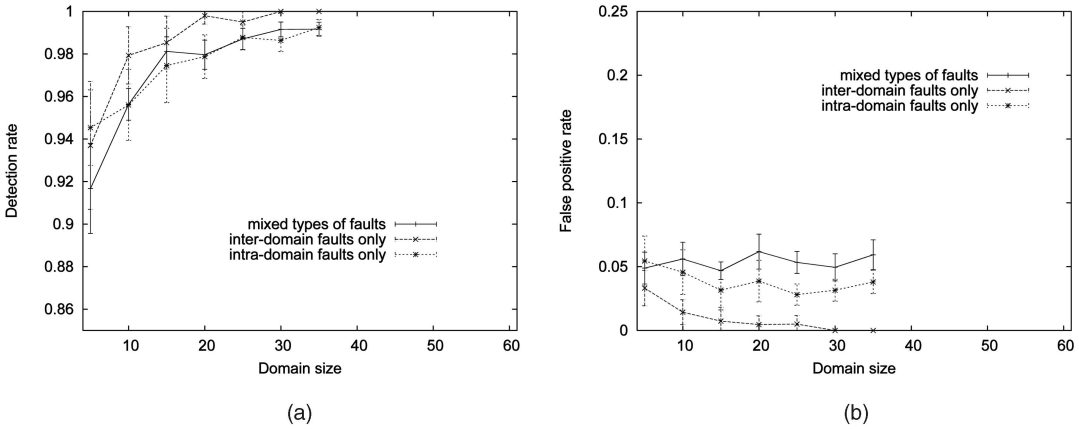


Fig. 10. Accuracy of MD-BUA in a 50-domain network. (a) Detection rate. (b) False positive rate.

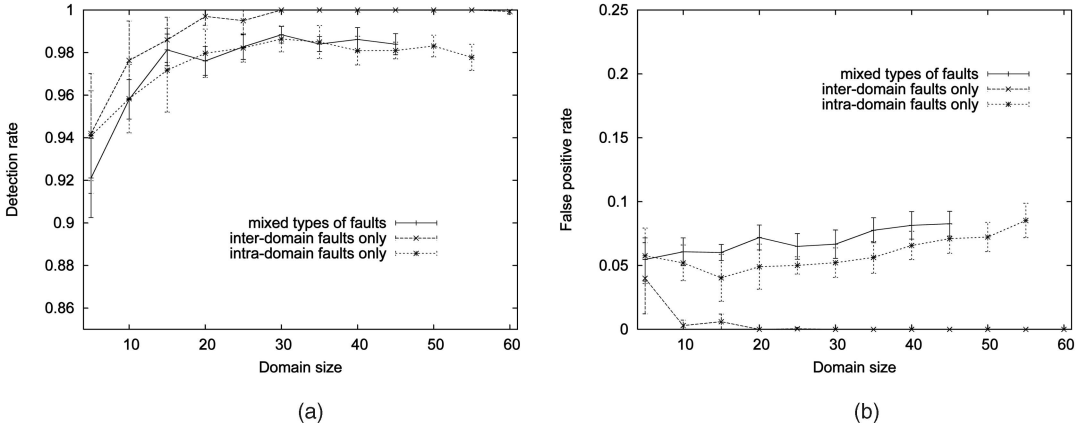


Fig. 11. Accuracy of MD-IHUA in a 50-domain network. (a) Detection rate. (b) False positive rate.

interdomain-fault scenarios results from their different complexities expressed by the number of simultaneous faults and the number of received symptoms.

We repeat the same set of experiments using networks composed of 50 domains. The results of the study are presented in Figs. 10a and 10b for MD-BUA and in Figs. 11a and 11b for MD-IHUA, respectively. Note that, in the case of MD-IHUA, we now work with networks composed of as many as 3,000 nodes. For completeness, we also include Figs. 12a and 12b, which show the average numbers of faults and symptoms generated in the considered fault scenarios. Figs. 13a and 13b compare the fault localization times of MD-BUA and MD-IHUA. The study performed on a 50-domain network confirms the results discussed previously. However, note that, in a bigger network, the complexity of scenarios is much higher: In a 50-domain network, our fault localization techniques are required to accurately diagnose scenarios that involve more than six simultaneous faults (Fig. 12a) and more than 2,500 symptoms (Fig. 12b).

In Figs. 14a, 14b, and 14c, we present the comparison of detection rate, false positive rate, and fault localization time for centralized and multidomain versions of BUA and IHU. Due to the excessive computation time of centralized

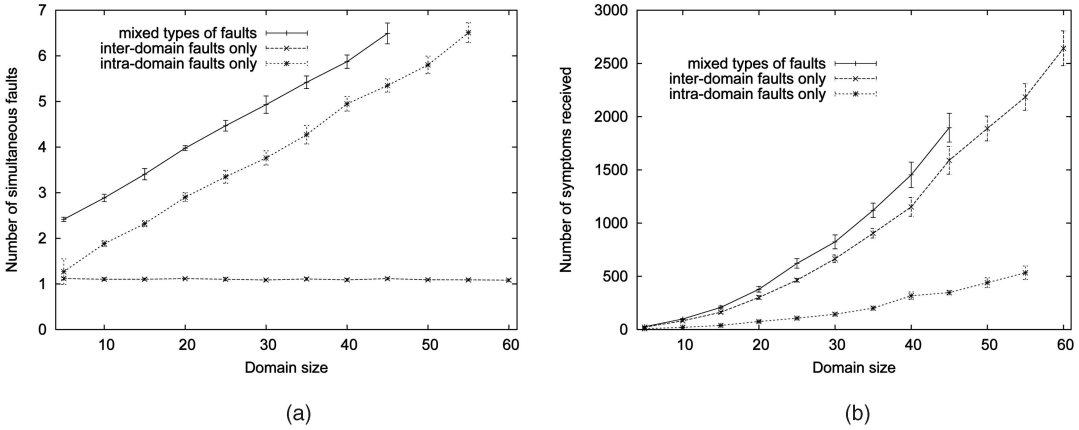


Fig. 12. Average number of faults and symptoms generated in experiment scenarios for a 50-domain network. (a) Average number of simultaneous faults. (b) Average number of observed symptoms.

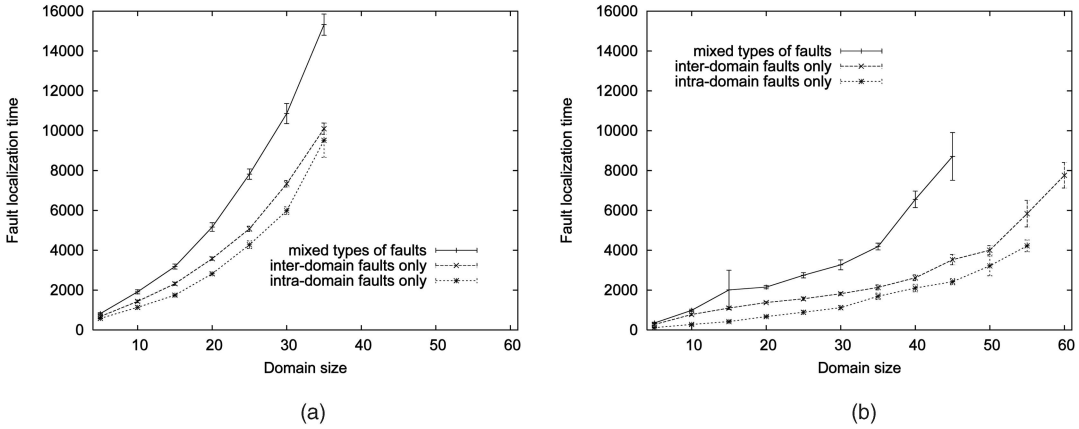


Fig. 13. Fault localization time in a network composed of 50 domains. (a) MD-BUA. (b) MD-IHUA.

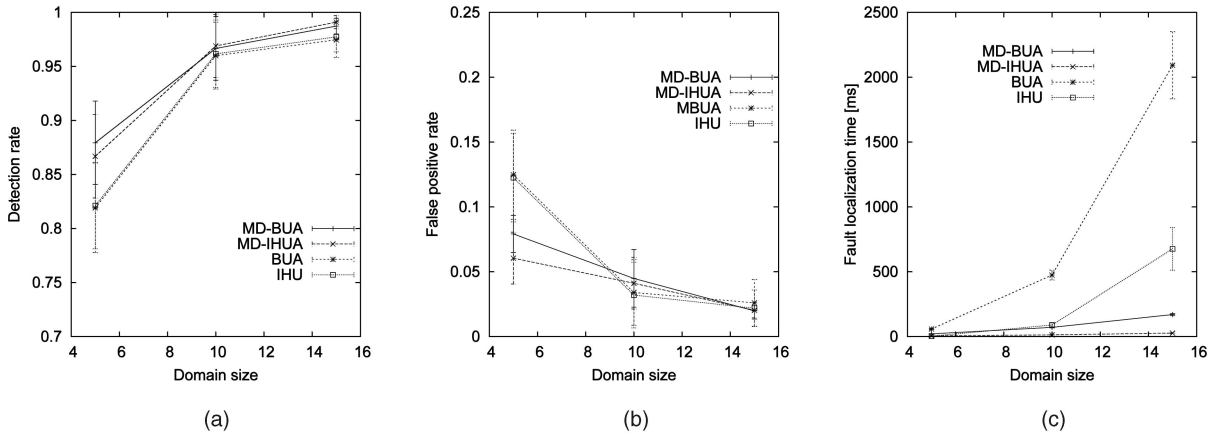


Fig. 14. Comparison of centralized and multidomain fault localization. (a) Detection rate. (b) False positive rate. (c) Fault localization time.

algorithms, we had to significantly limit the scope of the experiments, which were executed in a five-domain network in which the domain size varied between 5 and 15. The observability ratios of intradomain and interdomain symptoms were 0.5 and 0.1, respectively. The figures show that distributed fault localization performed according to the framework defined by MDA may be as accurate as centralized fault localization, while offering much better scalability. In fact, in smaller networks, multidomain fault localization may be even more accurate than the centralized one because it takes advantage of the hierarchical composition of network paths. Multidomain fault localization proves much more efficient than the centralized one, decreasing the fault localization time by an order of magnitude.

9 Conclusion

This paper introduces a multidomain fault localization approach to end-to-end service failure diagnosis in hierarchically routed networks. This approach divides the

computational effort and system knowledge involved in end-to-end service-failure diagnosis among multiple, hierarchically organized managers. Each manager is responsible for fault localization within the network domain it governs and reports to a higher-level manager that oversees and coordinates the fault-localization process of multiple domains. The paper identifies two main difficulties of fault management in multidomain networks: failure propagation among domains and a lack of global information about the system structure and state. To address these challenges, the paper first proposes an algorithmic framework for the design of probabilistic hierarchical multidomain fault localization techniques. It then introduces two refinements that expand on the centralized algorithms introduced in our previous work: iterative belief updating [20] and incremental hypothesis updating [19]. The multidomain approach is shown to provide high accuracy while increasing the admissible network size by an order of magnitude.

ACKNOWLEDGMENTS

This paper was 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 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. The US Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation thereon.

REFERENCES

[1] W. Aiello, F. Chung, and L. Lu, "A Random Graph Model for Massive Graphs," *Proc. 32nd Ann. ACM Symp. Theory of Computing,* pp. 171-180, May 2000.
[2] R. Albert and A. Barabasi, "Topology of Evolving Networks: Local Events and Universality," *Physical Rev. Letters,* pp. 5137-5234, 2000.
[3] A. Barabasi and R. Albert, "Emergence of Scaling in Random Networks," *Science,* pp. 509-512, Oct. 1999.
[4] A.T. Bouloutas, S.B. Calo, A. Finkel, and I. Katzela, "Distributed Fault Identification in Telecommunication Networks," *J. Network and Systems Management,* vol. 3, no. 3, 1995.
[5] T. Bu and D. Towsley, "On Distinguishing between Internet Power Law Topology Generators," *Proc. IEEE INFOCOM,* June 2002.
[6] K. Calvert, M. Doar, and E. Zegura, "Modeling Internet Topology," *IEEE Trans. Comm.,* pp. 160-163, Dec. 1997.
[7] P. Hasselmeyer, "An Infrastructure for the Management of Dynamic Service Networks," *IEEE Comm. Magazine,* vol. 41, no. 4, pp. 120-126, 2003.
[8] G. Jakobson and M.D. Weissman, "Alarm Correlation," *IEEE Network,* vol. 7, no. 6, pp. 52-59, Nov. 1993.
[9] C. Jin, Q. Chen, and S. Jamin, "Inet: Internet Topology Generator," Technical Report CSE-TR443-00, Dept. of Electrical Eng. and Computer Science, Univ. of Michigan, 2000.
[10] I. Katzela, A.T. Bouloutas, and S. Calo, "Comparison of Distributed Fault Identification Schemes in Communication Networks," Technical Report RC 19630 (87058), T.J. Watson Research Center, IBM Corp., Sept. 1993.
[11] I. Katzela, A.T. Bouloutas, and S.B. Calo, "Centralized vs Distributed Fault Localization," *Integrated Network Management IV,* A.S. Sethi, F. Faure-Vincent, and Y. Raynaud, eds., pp. 250-263, Chapman and Hall, May 1995.
[12] I. Katzela and M. Schwartz, "Schemes for Fault Identification in Communication Networks," *IEEE/ACM Trans. Networking,* vol. 3, no. 6, pp. 733-764, 1995.
[13] F.R. Kschischang, B.J. Frey, and H.-A. Loeliger, "Factor Graphs and the Sum-Product Algorithm," *IEEE/ACM Trans. Information Theory,* vol. 47, no. 2, pp. 498-519, Feb. 2001.
[14] A. Medina, A. Lakhina, I. Matta, and J. Byers, "BRITE: Universal Topology Generation from a User's Perspective," Technical Report BUCS-TR-2001-003, Computer Science Dept., Boston Univ., Apr. 2001.
[15] A. Medina, I. Matta, and J. Byers, "On the Origin of Power Laws in Internet Topologies," *ACM Computer Comm. Rev.,* vol. 30, no. 2, pp. 18-28, Apr. 2000.
[16] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann Publishers, 1988.
[17] M. Steinder and A.S. Sethi, "Multi-Domain Diagnosis of End-to-End Service Failures in Hierarchically Routed Networks," Technical Report 2003-10, Dept. of Computer and Information Sciences, Univ. of Delaware, 2003, www.cis.udel.edu/~steinder/PAPERS/TR-2003-10.pdf.
[18] M. Steinder and A.S. Sethi, "Multi-Domain Diagnosis of End-to-End Service Failures in Hierarchically Routed Networks," *IFIP Networking,* May 2004.
[19] M. Steinder and A.S. Sethi, "Probabilistic Fault Diagnosis in Communication Systems through Incremental Hypothesis Updating," *Computer Networks,* vol. 45, no. 4, pp. 537-562, July 2004.
[20] M. Steinder and A.S. Sethi, "Probabilistic Fault Localization in Communication Systems Using Belief Networks," *IEEE/ACM Trans. Networking,* vol. 12, no. 5, pp. 809-822, 2004.
[21] M. Steinder and A.S. Sethi, "A Survey of Fault Localization Techniques in Computer Networks," *Science of Computer Programming,* special issue on network and system administration, vol. 53, pp. 165-194, 2004.
[22] S.A. Yemini, S. Kliger, E. Mozes, Y. Yemini, and D. Ohsie, "High Speed and Robust Event Correlation," *IEEE Comm. Magazine,* vol. 34, no. 5, pp. 82-90, 1996.

Malgorzata Steinder (M'99) received the MS degree in computer science from the AGH University of Science and Technology, Poland, and the PhD degree in computer and information sciences from the University of Delaware. For her work on probabilistic fault localization techniques, she was awarded the Allan P. Colburn Prize for the Outstanding Doctoral Dissertation in Mathematical Sciences and Engineering from the University of Delaware. She served as a junior faculty member at the AGH University of Science and Technology. Currently, she is a research staff member at the IBM T.J. Watson Research Center. She is working on dynamic resource management for distributed middleware. She also served as a TPC member for IEEE INFOCOM '05 and '06 and IM '07. She is a member of the IEEE.

Adarshpal S. Sethi (M'85) received the MS degree in electrical engineering and the PhD degree in computer science, both from the Indian Institute of Technology (IIT), Kanpur, India. He has served on the faculty at IIT Kanpur, was a visiting faculty member at the Washington State University, Pullman, and was a visiting scientist at IBM Research Laboratories, Zurich, Switzerland, and the US Army Research Laboratory, Aberdeen, Maryland. He is currently a professor in the Department of Computer & Information Sciences at the University of Delaware, Newark. Dr. Sethi is on the editorial boards of the *Electronic IEEE Transactions on Network and Service Management*, the *Journal of Network and Systems Management*, the *International Journal of Network Management*, and the *Electronic Commerce Research Journal*. He was cochair of the program committee for ISINM '95 and was a general and program chair for DSOM '98; he has also been on the program committees of numerous other conferences. Dr. Sethi's research interests include architectures and protocols for network management, fault management, quality-of-service and resource management, and management of wireless networks. He is a member of the IEEE and the ACM.