Intelligent probing: A cost-effective approach to fault diagnosis in computer networks

by M. Brodie
I. Rish
S. Ma

We consider the use of probing technology for cost-effective fault diagnosis in computer networks. Probes are test transactions that can be actively selected and sent through the network. This work addresses the probing problem using methods from artificial intelligence. We call the resulting approach intelligent probing. The probes are selected by reasoning about the interactions between the probe paths. Although finding the optimal probe set is prohibitively expensive for large networks, we implement algorithms that find near-optimal probe sets in linear time. In the diagnosis phase, we use a Bayesian network approach and use a local-inference approximation scheme that avoids the intractability of exact inference for large networks. Our results show that the quality of this approximate inference "degrades gracefully" under increasing uncertainty and increases as the quality of the probe set increases.

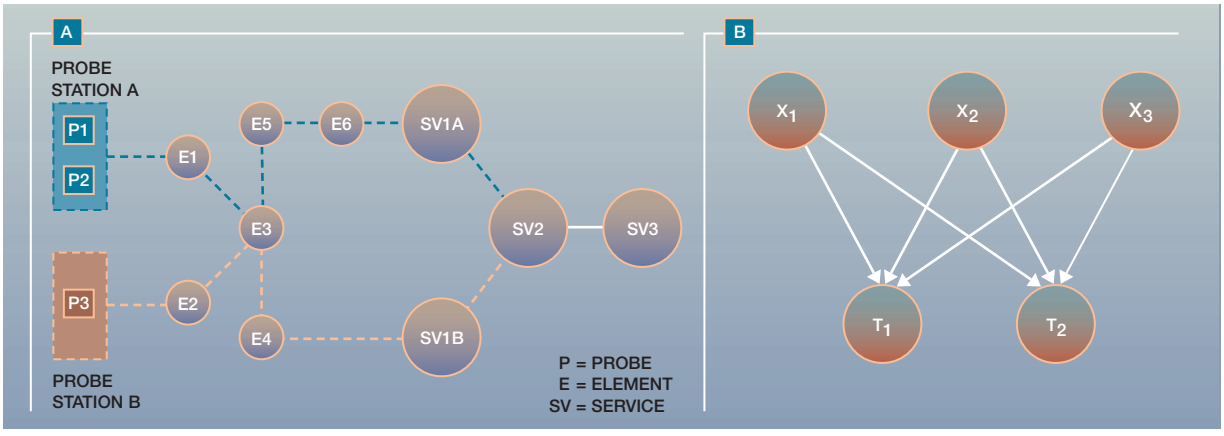
As distributed systems and networks continue to grow in size and complexity, tasks such as fault localization and problem diagnosis become significantly more challenging. As a result, tools are needed that can assist in performing these management tasks by both responding quickly and accurately to the ever-increasing volume of system measurements, such as alarms and other events, and also actively selecting informative tests to minimize the cost of diagnosis while maximizing its accuracy.

In this paper, we address the problem of diagnosis in distributed computer systems by using test transactions, or *probes*. A distributed system can be represented as a "dependency graph," where nodes can be either hardware elements (e.g., workstations, servers, routers) or software components or services, and links can represent both physical and logical connections between the elements (see Figure 1A). Probes offer the opportunity to develop an approach to diagnosis that is more active than traditional "passive" event correlation and similar techniques. A probe is a command or transaction (e.g., *ping* or *traceroute* command, an e-mail message, or a Web-page access request), sent from a particular machine called a *probing station* to a server or a network element in order to test a particular service (e.g., IP [Internet Protocol]-connectivity, database access, or Web access). A probe returns a set of measurements, such as response times, status code (OK or not OK), and so on. Probing technology is widely used to measure the quality of network performance, often motivated by the requirements of service-level agreements. Examples of probing technology include the IBM T. J. Watson Research Center EPP technology[1] and the Keynote measurement product.[2]

The use of probing technology for cost-effective diagnosis requires addressing two issues: a planning phase in which the probes are selected, followed by a diagnosis phase in which problem determination

0018-8670/02/$5.00 © 2002 IBM

Figure 1 (A) An example of a probing environment; (B) a two-layer Bayesian network structure



is performed using the results of the probes. The planning phase requires selecting a small but effective subset of all the possible probes. The diagnosis phase requires making inferences about the state of the network from the probe results.

To use probes, probing stations must first be selected at one or more locations in the network. Then the probes must be configured; it must be decided which network elements to target and from which station each probe should originate. Using probes imposes a cost, both because of the additional network load that their use entails and also because the probe results must be collected, stored, and analyzed. Cost-effective diagnosis requires a small probe set, yet the probe set must also provide wide coverage in order to locate problems anywhere in the network.

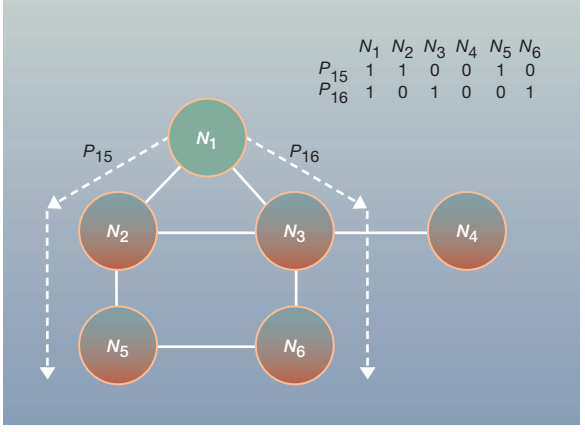
By reasoning about the interactions among the probe paths, an information-theoretic estimate of which probes are valuable can be constructed. This estimate yields a quadratic-time algorithm that finds near-optimal probe sets. We also implement a linear-time algorithm that can be used to find small probe sets very quickly; a reduction of almost 50 percent in the probe set size is achieved.

Once the probes have been selected and sent, fault diagnosis is performed by analyzing the probe outcomes. In real-life scenarios this analysis must be done in an environment of noise and uncertainty. For example, a probe can fail (e.g., because of packet loss) even though all the nodes it goes through are operational. Conversely, there is a chance that a probe succeeds even if a node on its path has failed (e.g., dynamic routing may result in the probe following a different path). Thus the task is to determine the most likely configuration of the states of the network elements.

We use the graphical framework of Bayesian networks[3] that provides both a compact factorized representation for multivariate probabilistic distributions as well as a convenient tool for probabilistic inference. An example of a simple Bayesian network for problem diagnosis is shown in Figure 1B: a bipartite (two-layer) graph where the top-layer nodes represent marginally independent faults or other problems[4] as a set $\mathbf{X} = (X_1, X_2, X_3)$ of network elements and the bottom-layer nodes represent probe results, $\mathbf{T} = (T_1, T_2)$. Since the exact inference in Bayesian networks is generally hard (NP-hard, Nondeterministic Polynomial-time hard—a set or property of certain problems in computational complexity theory),[5] we investigate the applicability of approximation techniques and present experimental results that suggest that a local-inference approach performs well and provides a cost-effective method for fault diagnosis in large networks.

The probing problem is typical of many practical applications in which a set of tests must first be selected and then a diagnosis is made from the test outcomes. Other examples arise in medical diagnosis, the identification of defective parts, code construction for noisy channel transmission, and so on. In general, the planning phase requires selecting a small but effective subset of all the possible tests. The diag-

Figure 2 An example network and dependency matrix



nosis phase requires making inferences from the test results in an environment of noise and uncertainty.

In the next section of this paper we formulate the probe selection problem as a constrained optimization problem—find the minimal subset of probes that has the ability to diagnose the problems of interest. Algorithms for solving this problem are presented, and their experimental results are shown. In the subsequent section we formulate the problem of probabilistic fault diagnosis using a noisy-AND Bayesian network framework, derive a lower bound on the error of diagnosis, and examine a local-inference approximation scheme for performing diagnosis. This approach yields better approximations for higher quality probe sets and "degrades gracefully" with increasing noise. Related work is discussed in the fourth section, and the last section presents some overall conclusions and directions for future work.

Probe set construction

We now formulate the probe selection problem.

Notation and approach. We first describe our notation and explain our approach.

Suppose the network has n nodes. Each probe is represented as a binary string of length n, where a 1 in position j denotes that the probe passes through node N_j. This defines a dependency matrix $D(i,j)$, where $D(i,j) = 1$ if probe P_i passes through node N_j, $D(i,j) = 0$ otherwise. D is an r-by-n matrix, where r is the number of probes. (This formulation is motivated by the "coding" approach to event correlation suggested by Kliger et al.[6])

For example, consider the network in Figure 2. Suppose one probe is sent along the path $N_1 \rightarrow N_2 \rightarrow N_5$ while another is sent along the path $N_1 \rightarrow N_3 \rightarrow N_6$. The resulting dependency matrix is shown to the right of the network (probes are indexed by their start and end nodes).

Each probe that is sent out either returns successfully or fails to do so. In a noise-free environment, if a probe is successful, then every node and link along its path must be up; conversely, if a node or link is down, then any probe passing through that node or link fails to return. Thus r probes result in a "signal" of a binary string of length r, each digit denoting whether or not that probe returned successfully (we do not consider exploiting the actual value of the return time if the probe is successful).

For example, in Figure 2 if only N_2 is down, then probe P_{15} (subscript denotes the origin and destination nodes) fails, but P_{16} succeeds. Similarly, if only N_5 is down, then P_{15} fails, but P_{16} succeeds. Thus, these two failures result in the same signal, because their columns in the dependency matrix are identical.

If N_1 is down, both probes will fail, and no other single node failure causes both probes to fail. Thus a failure in N_1 can be uniquely identified by these two probes, as shown by the fact that the column of N_1 in the dependency matrix is unique.

In general, any problem whose column in the dependency matrix is unique generates a unique signal and as a result can be unambiguously diagnosed. (However, a problem whose column is all zeroes cannot be detected even if its column is unique, so to avoid this technicality we add a column of all zeroes to the dependency matrix to represent the case of no failure occurring.) The goal is to find the smallest probe subset that can uniquely diagnose a failure in any node. (To extend this approach to multiple simultaneous failures see Reference 7.)

It is important to note that the network model is quite general. For example, layering can be accommodated if a Web server depends on Transmission Control Protocol/Internet Protocol (TCP/IP) running, which depends on the box being up; this can be modeled as a node for the box with a link to TCP/IP from the box and a further link from TCP/IP to the Web server. Thus, nodes may represent applications and link de-

pendencies between those applications. Similarly, a node may represent a subnetwork of many nodes whose interconnections are unknown. In this case, probing will determine that the problem lies somewhere in that subnetwork, at which point some form of local system management (perhaps including local probing) may be used to pinpoint the problem.

Problem statement. We can formulate probe selection as a constrained optimization problem, as follows. The initial probe set P and the dependency matrix D are given. Let P' be any subset of P. Define, for $j = 1$ to n, $C_j = \{D_{ij}\}$, $P_i \in P'$; C_j is the jth subcolumn of D, with the extracted rows corresponding to the probes in P'. Then the number of diagnosable problems is given by counting the number of unique columns: $h(P') = \Sigma_1^n c_j$, $c_j = 1$ if C_j is distinct from $C_1, \ldots, C_{j-1}$ (otherwise $c_j = 0$).

The probe selection problem is to find the smallest probe subset that can diagnose all the problems, that is, min $|P'|$ such that $h(P') = n$.

Determining the initial probe set. The set of candidate probes can be provided from whatever sources are available; for example, a human expert may specify which probes are possible. However, it may also be useful to compute the available probes from the network structure and the location of the probe stations.

We begin by selecting from the n nodes a subset of k nodes as the probe stations. (In this work we do not address the question of how to select the probe stations, since they usually cannot be chosen to optimize the probing strategy.) A probe can be sent to any node from any probe station. In the example we will assume that the probe follows the shortest path from probe station to target. This assumption creates a candidate set of probes of size $r = O(n)$; note that this set is sufficient to diagnose any single node being down because one can simply use one probe station and send a probe to every node.

Determining the diagnostic ability of a set of probes. In general, the count $h(P)$ of the number of unique problems detectable by a probe subset P is not a good measure of the diagnostic ability of P (unless $h(P) = n$). For example, suppose probe set P_1 induces the decomposition $S_1 = \{\{1, 2\}, \{3, 4\}\}$, and probe set P_2 induces the decomposition $S_2 = \{\{1\}, \{2, 3, 4\}\}$. Although P_2 can uniquely diagnose one node and P_1 cannot, it is possible to add just a single probe to P_1 and diagnose all the nodes, whereas at least two additional probes must be added to P_2 before all nodes can be diagnosed. Therefore, S_1 is a "better" decomposition than S_2.

We define the diagnostic ability $H(P)$ of a set of probes P to be the conditional entropy (see Reference 8) $H(N|G)$, where $N = \{1, \ldots, n\}$ denotes the node, and $G = \{1, \ldots, k\}$ denotes which group contains the node in the decomposition induced by P; each group contains nodes whose failures cannot be distinguished from one another. Let n_i be the number of nodes in group g_i. Then:

$$
\begin{aligned}
H(P) &= H(N|G) \\
&= \sum_{i=1}^{k} p(G = g_i) H(N|G = g_i) \\
&= \sum_{i=1}^{k} \frac{n_i}{n} \left[- \sum_{j=1}^{n} p(N = j|G = g_i) \right. \\
&\qquad \left. log(P(N = j|G = g_i)) \right] \\
&= \sum_{i=1}^{k} \frac{n_i}{n} \left[- \sum_{j=1}^{n_i} \frac{1}{n_i} log\left(\frac{1}{n_i}\right) \right] = \sum_{i=1}^{k} \frac{n_i}{n} log(n_i)
\end{aligned}
$$

For any node in g_i, at least $\log(n_i)$ additional probes are needed to uniquely diagnose that node. Since a random node lies in g_i with probability n_i/n, the diagnostic ability $H(P)$ is simply the expected minimal number of further probes needed to uniquely diagnose all nodes. Note that lower values for $H(P)$ correspond to better probe sets.

For example,

$$
\begin{aligned}
H(P_1) &= H(\{\{1, 2\}, \{3, 4\}\}) \\
&= \frac{1}{2} log2 + \frac{1}{2} log2 = log2 = 1
\end{aligned}
$$

whereas

$$
\begin{aligned}
H(P_2) &= H(\{\{1\}, \{2, 3, 4\}\}) \\
&= \frac{1}{4} log1 + \frac{3}{4} log3 = \frac{3}{4} log3 = 1.19
\end{aligned}
$$

This formula for $H(P)$ is valid if failures are equally likely in any node. If this is not the case, prior knowledge about the likelihood of different types of failures can be incorporated into the measure of diagnostic ability. The decomposition induced by a probe set can be efficiently computed row-by-row using the fact that adding a probe always results in a more extensive decomposition, because nodes in distinct groups remain distinguishable. An additional probe can only have the effect of distinguishing previously indistinguishable nodes.

Finding the minimal set of probes. In general, one probe station and n probes can locate any single failed node because a probe can be sent to every node. However, in many situations far fewer probes may suffice. Because r probes generate 2^r possible signals (one of which corresponds to the case that there is no failure), in the ideal situation only $\log(n) + 1$ probes are needed to locate a single failure in any of n nodes. However, this condition is only achievable if all the necessary links exist in the network and it is possible to guarantee that a probe follows a specified path. In the case of shortest-path routing with an arbitrary network structure, the minimal number of probes may lie anywhere between $\log(n) + 1$ and n; the exact value depends on the network structure and the location of the probe stations.

We now examine algorithms for finding the minimal probe set. Since the probe selection problem is NP-hard[9] and an exhaustive search is therefore impractical for large networks, two approximation algorithms are considered: one ("subtractive search") requiring linear time and the other ("greedy search") requiring quadratic time. An experimental comparison of the algorithms is presented shortly.

Subtractive search. Subtractive search starts with the initial set of r probes, considers each probe in turn, and *discards it if it is not needed*, that is, if the diagnostic ability remains the same even if it is dropped from the probe set. This process terminates in a subset with the same diagnostic ability as the original set but which may not necessarily be of minimal size. The running time is linear in the size of the original probe set, because each probe is considered only once.

The order of the initial probe set is quite important for the performance of this algorithm. If the probes are ordered by probe station, the algorithm will remove all the probes until the last n (all of which are from the last probe station), since these suffice to diagnose any node. This reduces the opportunity of exploiting probes from different probe stations. The size of the probe set can be reduced by randomly ordering the initial probe set, or ordering it by target node.

Greedy search. Another approach is a greedy search algorithm where at each step we add the probe that results in the "most informative" decomposition, using the measure of diagnostic ability defined previously. The additive algorithm starts with the empty set and repeatedly adds the probe that gives the decomposition of highest diagnostic ability. This algorithm also finds a nonoptimal probe subset with the same diagnostic ability as the original set. The running time of this algorithm is quadratic in r, the size of the original probe set, because at each step the diagnostic ability achieved by adding each of the remaining probes must be computed.

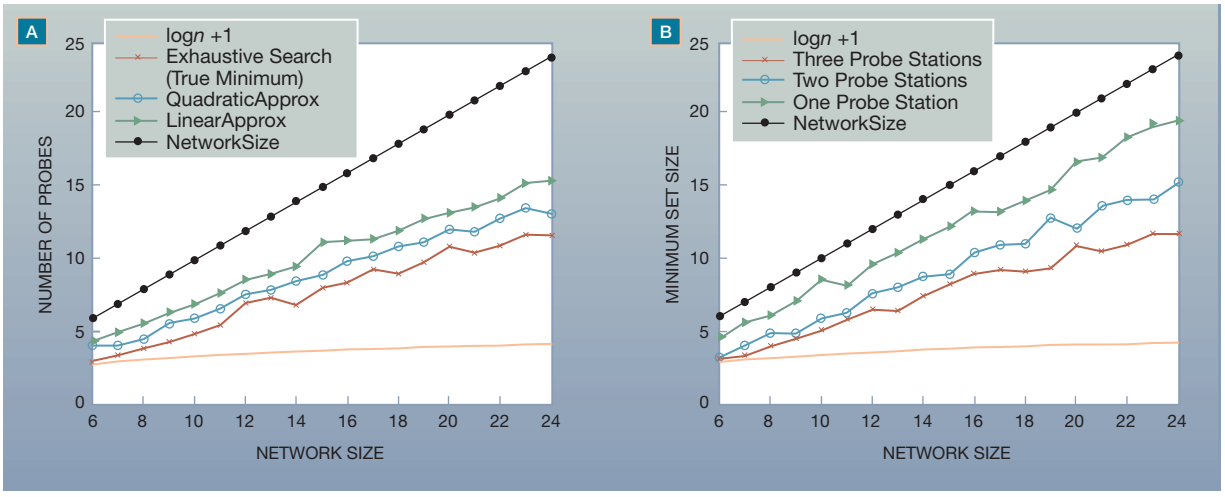
Experiments. This subsection investigates experimentally both the general behavior of the minimum set size and how the two approximation algorithms compare with exhaustive searching in computing the probe set. The main result is that the approximation algorithms find a probe set that is very close to the true minimum set size and can be effectively used on large networks where exhaustive searching is impractical.

For each network size n, we generate a network with n nodes by randomly connecting each node to four other nodes. Each link is then given a randomly generated weight to reflect network load. The probe stations are selected randomly. One probe is generated from each probe station to every node using shortest-path routing. The three algorithms described previously are then executed. This process is repeated ten times for each network size and the results averaged.

Figure 3 shows the case of three probe stations. The size of the probe set found by all the algorithms lies between $\log(n) + 1$ and n, as expected. The minimal size is always larger than the theoretical lower bound of $\log(n) + 1$, for two reasons:

- The networks are not very dense; since each node is linked to four other nodes, the number of edges increases only linearly with network size. Thus many probe paths are simply not possible.
- Since the probes follow the least-cost path from probe station to node, the probe paths tend to be

Figure 3 Algorithms for computing probe sets: (A) True minimum and two approximation algorithms on small networks; (B) the approximation algorithms on large networks



short, passing through few nodes, resulting in a reduction of opportunities for exploiting interactions between probe paths.

The results also show that the approximation algorithms perform well. The size of the probe set is much closer to the true minimum than to the upper bound. Figure 3 also illustrates the performance of these algorithms on larger networks for which exhaustive searching is not feasible. The quadratic-time algorithm slightly outperforms the linear-time algorithm, but its computational cost is higher. An alternative approach is to run the linear-time algorithm many times with different initial orderings and take the best result.

Probabilistic diagnosis

In this section we assume that a set of probes has already been constructed and focus on using it for fault diagnosis. Until now, we have assumed that the probe signal is received correctly; no data in the network are lost or spuriously altered. If network errors are possible, we require that the distance between probe signals (the codebook "radius" in the terminology of Kliger et al.[6]) is larger than a single bit, thereby providing robustness to noise and lost packets. Dynamic network routing is another source of uncertainty, since the path probes through the network may not be known accurately. Other changes to the network may occur; for example, nodes and links are continually being added, removed, and reconfigured. For these reasons the dependency matrix may need to be regularly updated. Another approach is to include the uncertainties in the model itself. We will next show how the dependency matrix can be naturally extended to a Bayesian network that encodes probabilistic dependencies between the possible faults in the network (causes) and the probe outcomes (symptoms).

A noisy-AND Bayesian network for fault diagnosis. As before, we consider a simplified model of a computer network where each node (router, server, or workstation) can be in one of two states, 0 (fault) or 1 (no fault). To avoid confusion with the previous section, we change notation slightly: the states of the network elements are denoted by a vector $\mathbf{X} = (X_1, \ldots, X_n)$ of n *unobserved* Boolean variables. Each probe, or test, T_j, originates at a particular node (probing workstation) and goes to some destination node (server or router). A vector $\mathbf{T} = (T_1, \ldots, T_m)$ of *observed* Boolean variables denotes the outcomes (0 = failure, 1 = OK) of m probes. Lowercase letters, such as x_i and t_j, denote the values of the corresponding variables, i.e., $\mathbf{x} = (x_1, \ldots, x_n)$ denotes a particular assignment of node states, and $\mathbf{t} = (t_1, \ldots, t_m)$ denotes a particular outcome of m probes.

We assume that the probe outcome is affected by all nodes on its path, and that node failures are mar-

ginally independent. These assumptions yield a causal structure depicted by a two-layer Bayesian network, such as the one in Figure 1B. Let $\mathbf{pa}(\mathbf{T}_1)$ denote the set of *parents* of T_i, that is, the nodes pointing to T_i in the directed graph (the nodes on the probe path). The joint probability $P(\mathbf{x}, \mathbf{t})$ for such a network can then be written as follows:

$$P(\mathbf{x}, \mathbf{t}) = \prod_{i=1}^{n} P(x_i) \prod_{j=1}^{m} P(t_j|\mathbf{pa}(t_j)) \quad (1)$$

where $P(t_j|\mathbf{pa}(t_j))$ is the *conditional probability distribution* (CPD) of node T_i given the set of its parents, and $P(x_i)$ is the prior probability that $X_i = x_i$ before any probes have been sent.

In general, a CPD defined on binary variables is represented as a k-dimensional table where $k = |Pa(t_j)|$. Thus, just the specification complexity is $O(2^k)$ which is very inefficient, if not intractable, in large networks with a long probe path (i.e., large parent set). It seems reasonable to assume that each element on the path of the probe affects the outcome of the probe independently (the assumption known as *causal independence*[10]). For example, in the absence of uncertainty, a probe fails if and only if at least one node on its path fails, i.e., $T_i = X_{i_1} \wedge \ldots \wedge X_{i_k}$, where $\wedge$ denotes logical AND, and $X_{i_1}, \ldots, X_{i_k}$ are all the nodes probe T_i goes through. Therefore, once it is known that some $X_{i_j} = 0$, the probe fails independently of the values of other components. In practice, however, this relationship may be disturbed by "noise." For example, a probe can fail even though all nodes it goes through are fully operational (e.g., if network performance degradation leads to high response times interpreted as a failure). Vice versa, there is a chance the probe succeeds even if a node on its path has failed, e.g., because of a routing change. Such uncertainties yield a noisy-AND model that implies that several causes (e.g., node failures) contribute independently to a common effect (probe failure) and is formally defined as follows:

$$P(t = 1|x_1, \ldots, x_n) = (1 - l) \prod_{x_i=0}^{n} q_i, \text{ and}$$

$$P(t = 0|x_1 = 1, \ldots, x_n = 1) = 1 - l \quad (2)$$

where l is the leak probability that accounts for the cases of a probe failing even when all the nodes on its path are operational, and the link probabilities, q_i, account for the second kind of "noise" in the noisy-AND relationship, namely, for cases when a probe succeeds with a small probability q_i even if node X_i on its path fails.[11,12]

Once a Bayesian network is specified, the diagnosis task can be formulated as finding the maximum probable explanation (MPE), that is, a most likely assignment to all nodes X_i given the probe outcomes,

$$\mathbf{x}^* = \arg\max_{\mathbf{x}} P(\mathbf{x}|\mathbf{t}) = \arg\max_{\mathbf{x}} P(\mathbf{x}, \mathbf{t})$$

$$= \max_{\mathbf{x}} \prod_{j=1}^{n} P(x_j) \prod_{i=1}^{m} P(t_i|\mathbf{pa}(t_i)) \quad (3)$$

An alternative approach is to look for the most likely value x_i^* of each node X_i separately, namely, to construct a diagnosis $\mathbf{x}' = (x'_1, \ldots, x'_n)$, where $x'_i = \arg\max_{x_i} P(x_i|\mathbf{t})$, $i = 1, \ldots, n$. Computing $\mathbf{x}'$ is sometimes easier than finding the MPE, but generally, $x^* \neq \mathbf{x}'$.

When there is no noise in noisy-AND (i.e., leak and link probabilities are zero), the CPDs become deterministic, that is, each probe outcome $T_i = t_i$ imposes a constraint $t_i = x_{i_1} \wedge \ldots \wedge x_{i_k}$ on the values of its parent nodes $X_{i_1}, \ldots, X_{i_k}$. Now, finding the MPE can be viewed as a constrained optimization problem of finding $\mathbf{x}^* = \arg\max_{x_1,\ldots,x_n} \Pi_{j=1}^{n} P(x_j)$ subject to those constraints. Clearly, the quality of diagnosis depends on the set of probes: in general, the only way to guarantee the correct diagnosis is to have a constraint set with a unique solution. This guarantee can only be achieved for $m \geq n$, since 2^m probe outcomes must "code" uniquely for 2^n node state assignments. Earlier we explored the single fault case and showed how considerable savings could be achieved in this case.

Accuracy of diagnosis. In this subsection, we derive a lower bound on the MPE diagnosis error. The MPE *error*, or loss, L_M, is the probability that the MPE diagnosis X^* differs from the true state X (by at least one bit). Given particular values $\mathbf{T} = \mathbf{t}$, $\mathbf{X} = \mathbf{x}$, and diagnosis $\mathbf{X}^* = \mathbf{x}^*$, we have $P(\mathbf{x} \neq \mathbf{x}^*|\mathbf{t}) = I_{\mathbf{x}\neq\mathbf{x}^*|\mathbf{t}}$ where I_s is the *indicator function*, $I_s = 1$ if $s = true$ and $I_s = 0$ otherwise. Then the MPE error is

$$L_M = P(\mathbf{X} \neq \mathbf{X}^*|\mathbf{T}) = E_{\mathbf{x},\mathbf{t}} I_{\mathbf{x}\neq\mathbf{x}^*|\mathbf{t}}$$

$$= E_{\mathbf{t}}(1 - P(\mathbf{x}^*|\mathbf{t})) = 1 - \sum_{\mathbf{t}} P(\mathbf{x}^*, \mathbf{t}) \quad (4)$$

where $\mathbf{x}^*$ is an MPE assignment, and E_z denotes expectation over z. Similarly, we can define the *bit error*, or bit loss, $L_b = P(X_i \neq X_i^*|\mathbf{T})$. (Clearly, the MPE error is generally higher than the bit error.)

In the following, we assume that the priors—the prior probability of node failure before sending any probes—are the same for all nodes, that is, $P(X_i = 0) = p$ for all $i = 1, \ldots, n$; without loss of generality, we also assume $p \leq 0.5$. Then

$$P(\mathbf{x}^*, \mathbf{t}) = \max_{\mathbf{x}} \prod_{j=1}^{n} P(x_j) \prod_{i=1}^{m} P(t_i|\mathbf{pa}(t_i))$$

$$\leq (1-p)^n \prod_{i=1}^{m} \max_{\mathbf{x}} P(t_i|\mathbf{pa}(t_i)) \quad (5)$$

The noisy-AND definition (Expression 2) and the fact that $0 \leq q_i \leq 1$ yield $\max_{\mathbf{x}} P(t_i = 1|\mathbf{pa}(t_i)) = 1 - l_i$, and $\max_{\mathbf{x}} P(t_i = 0|\mathbf{pa}(t_i)) = 1 - \min_{\mathbf{x}} P(t_i = 1|\mathbf{pa}(t_i)) = 1 - (1 - l_i) \prod_{x_j \in \mathbf{pa}(t_i)}^{n} q_j$. Substitution of these two expressions in Expression 5 yields

$$P(\mathbf{x}^*, \mathbf{t}) \leq (1-p)^n \prod_{t_i=1} (1 - l_i) \prod_{t_i=0} \cdot \left[1 - (1 - l_i) \prod_{x_j \in \mathbf{pa}(t_i)}^{n} q_j \right] \quad (6)$$

In order to further simplify the derivation, we assume equal leak probabilities, $l_i = l$ for $i = 1, \ldots, m$, equal link probabilities $q_j = q$ for $j = 1, \ldots, n$, and equal parent set size (or probe route length) $|pa(t_i)| = r$. Using the notation $P_k(\mathbf{x}^*, \mathbf{t})$ instead of $P(\mathbf{x}^*, \mathbf{t})$ where k is the number of $t_i = 1$ in $\mathbf{t}$, we can write Expression 6 as $P_k(\mathbf{x}^*, \mathbf{t}) \leq (1-p)^n \alpha^k (1 - \alpha q^r)^{m-k} = (1-p)^n \alpha^k \beta^{m-k}$, where $\alpha = 1 - l$ and $\beta = 1 - \alpha q^r$. Since there are $\binom{m}{k}$ vectors $\mathbf{t}$ having exactly k positive components t_i, we obtain

$$\sum_{\mathbf{t}} P(\mathbf{x}^*, \mathbf{t}) \leq (1-p)^n \sum_{k=0}^{m} \binom{m}{k} \alpha^k \beta^{m-k}$$

$$= (1-p)^n (\alpha + \beta)^m \quad (7)$$

Since $\alpha + \beta = (1 - l)(1 - q^r) + 1$, we finally obtain a lower bound on MPE error $\underline{L_M}$:

$$L_M = 1 - \sum_{\mathbf{t}} P(\mathbf{x}^*, \mathbf{t}) \geq 1 - (1-p)^n$$

$$((1 - l)(1 - q^r) + 1)^m = \underline{L_M} \quad (8)$$

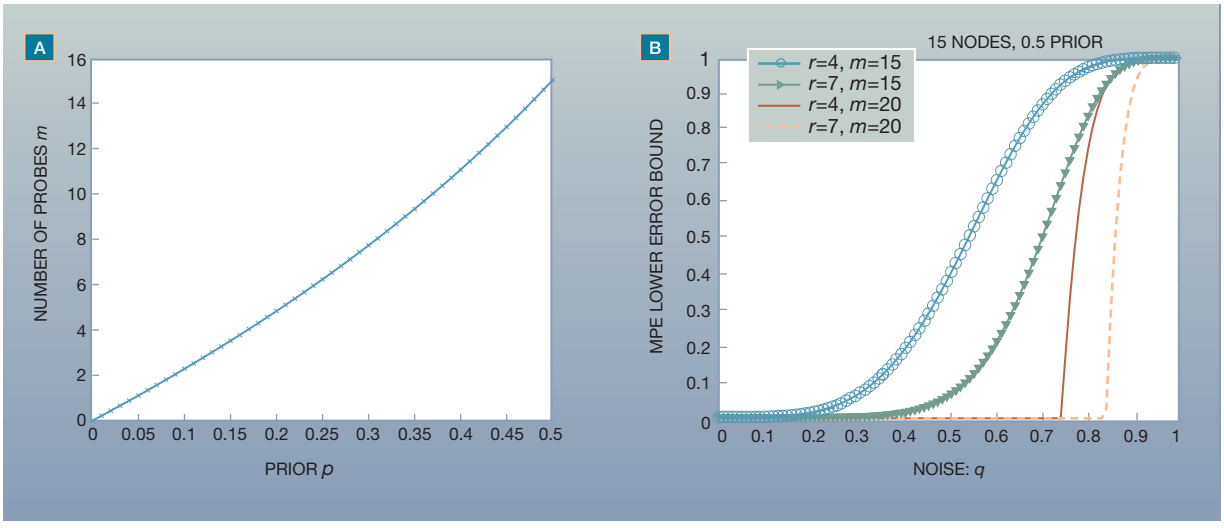
Note that in the absence of noise ($l = 0$ and $q = 0$), we obtain $L_M \geq 1 - (1-p)^n 2^m$. Thus, for uniform fault priors, $p = 0.5$, an error-free MPE diagnosis is only possible if $n = m$, as we noted before; however, for smaller p, zero error can be achieved with a smaller number of probes. Namely, solving $\underline{L_M} = 0$ for m yields the necessary condition for zero lower bound, $m \geq -n[\log(1 - p)/\log(1 + (1 - l)(1 - q^r))]$, plotted in Figure 4A as a function of p. Generally, solving $\underline{L_M} = 0$ for m provides a way of specifying the minimum necessary number of probes that yield zero lower bound for specified values of other parameters.[13]

Also, from Expression 8 we can see that the lower bound on the MPE diagnosis error is a monotone function of each parameter, n, m, p, l, q, or r, given that other parameters are fixed. Namely, the error (bound) increases with an increasing number of nodes n, fault probability p, leak probability l, and link probability q, but decreases with an increasing number of probes m and probe route length r, which agrees with one's intuition that having more nodes on the path of a probe, as well as a larger number of probes, provides more information about the true node states. For example, the sensitivity of the error bound to noise is illustrated in Figure 4B: note a relatively sharp transition from zero to 100 percent error with increasing noise; sharpness increases with increasing m and r.

Computational complexity of diagnosis and MPE approximations. Let us first consider the complexity of diagnosis in the absence of noise. Finding the most likely diagnosis reduces to constraint satisfaction in two cases. The first case is when the probe constraints allow exactly one solution (an assignment $\mathbf{x}$ simultaneously satisfying all constraints). The second case corresponds to uniform priors $P(x_i)$, which yield uniform posterior probability $P(\mathbf{x}|\mathbf{t})$. Therefore, any assignment $\mathbf{x}$ consistent with probe constraints is an MPE solution. Although constraint satisfaction is generally NP-hard, the particular problem induced by probing constraints can be solved in $O(n)$ time as follows.

Each successful probe yields a constraint $x_{i_1} \wedge \ldots \wedge x_{i_k} = 1$ that implies $x_i = 1$ for any node X_i on its path; the rest of the nodes are only included in con-

Figure 4 (A) Minimum number of probes to guarantee zero error bound versus fault prior p; (B) lower bound on MPE error versus link probability q ("noise")



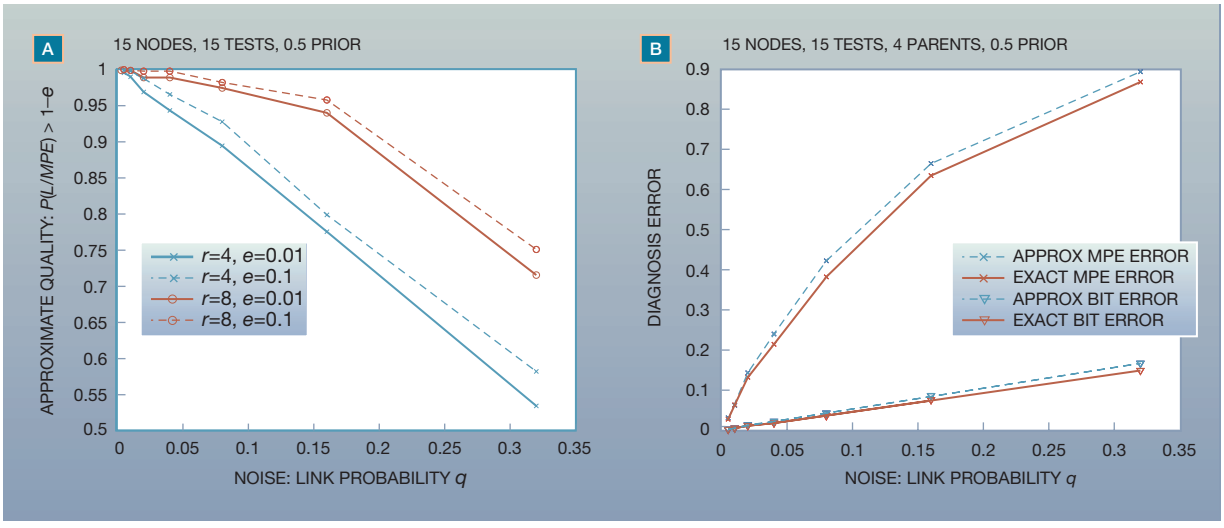
straints of the form $x_{i_1} \wedge \ldots \wedge x_{i_k} = 0$, or equivalently, $\neg x_{i_1} \vee \ldots \vee \neg x_{i_k} = 1$ imposed by failed probes. Thus, an $O(n)$-time algorithm assigns 1 to every node appearing on the path of a successful probe, and 0 to the rest of the nodes. This is equivalent to unit propagation in *Horn theories*, which are propositional theories defined as a conjunction of clauses, or disjuncts, where each disjunct includes no more than one positive literal. It is easy to see that probe constraints yield a Horn theory and thus can be solved by unit propagation in linear time. Therefore, finding the MPE diagnosis takes $O(n)$ time when it is equivalent to constraint satisfaction in the absence of noise, as in cases of uniform priors or unique diagnosis. In general, however, even in the absence of noise, finding the MPE is an NP-hard constrained optimization problem with worst-case complexity $O(exp(n))$.

Similarly, in the presence of noise, finding the MPE solution in a Bayesian network has complexity $O(exp(w^*))$, where w^* is the induced width of the network,[14] that is, the size of the largest clique created by an exact inference algorithm, such as variable elimination. It is easy to show that $w^* \geq k$, where k is the maximum number of parents of a probe node, and $w^* = n$ in the worst case.[15,16]

Thus, we focused on approximating MPE and studied empirically the algorithm $approx - mpe(i)$ (with $i = 1$, to be precise), which belongs to a family of the mini-bucket approximations for general constrained optimization, and particularly, for finding MPE.[17–19] The idea of the mini-bucket approximation[20,21] is to compute an upper bound on MPE $= \max_{\mathbf{x}} \Pi_i P(x_i|\mathbf{pa}_i) \leq \Pi_i \max_{x_i,\mathbf{pa}_i} P(x_i|\mathbf{pa}_i) = U$ and a lower bound L as a probability of an assignment $\mathbf{x}$ computed in a particular way, similarly to finding a solution to a constraint satisfaction problem after partial constraint propagation. Indeed, in the deterministic case, the $approx - mpe(1)$ scheme is equivalent to arc-consistency in a constraint network, or unit propagation in propositional satisfiability[19] (increasing the parameter i corresponds to a more "coarse" partitioning of $P(x_i|\mathbf{pa}_i)$ into subproducts before maximization, e.g., $i = n$ yields the exact MPE computation).

We tested $approx - mpe(1)$ on networks constructed in a way that guarantees the unique diagnosis in the absence of noise. (Particularly, besides m probes each having r randomly selected parents, we also generated n additional probes each having exactly one parent node, so that all X_i nodes are tested directly.) Since $approx - mpe(1)$ is equivalent to unit propagation in the absence of noise, its diagnosis coincides with the MPE. Adding noise in the form of link probability q caused "graceful degradation" of the approximation quality, as shown in Figure 5A, which plots the fraction of cases when the ratio L/MPE

Figure 5 (A) "Graceful degradation" of MPE approximation quality with noise; (B) MPE diagnosis error and bit diagnosis error for both exact and approximate diagnosis



was within the interval $[1 - e, 1]$ for small values of e, that is, where the approximation quality is measured as $P(L/MPE) > 1 - e$ for $e = 0.01$ and $e = 0.1$. The quality is higher when the noise is smaller and when the probe path is longer ($r = 8$ vs $r = 4$). This resulted in a diagnosis error very close to the error obtained by exact diagnosis, both for MPE error and bit error (Figure 5B).

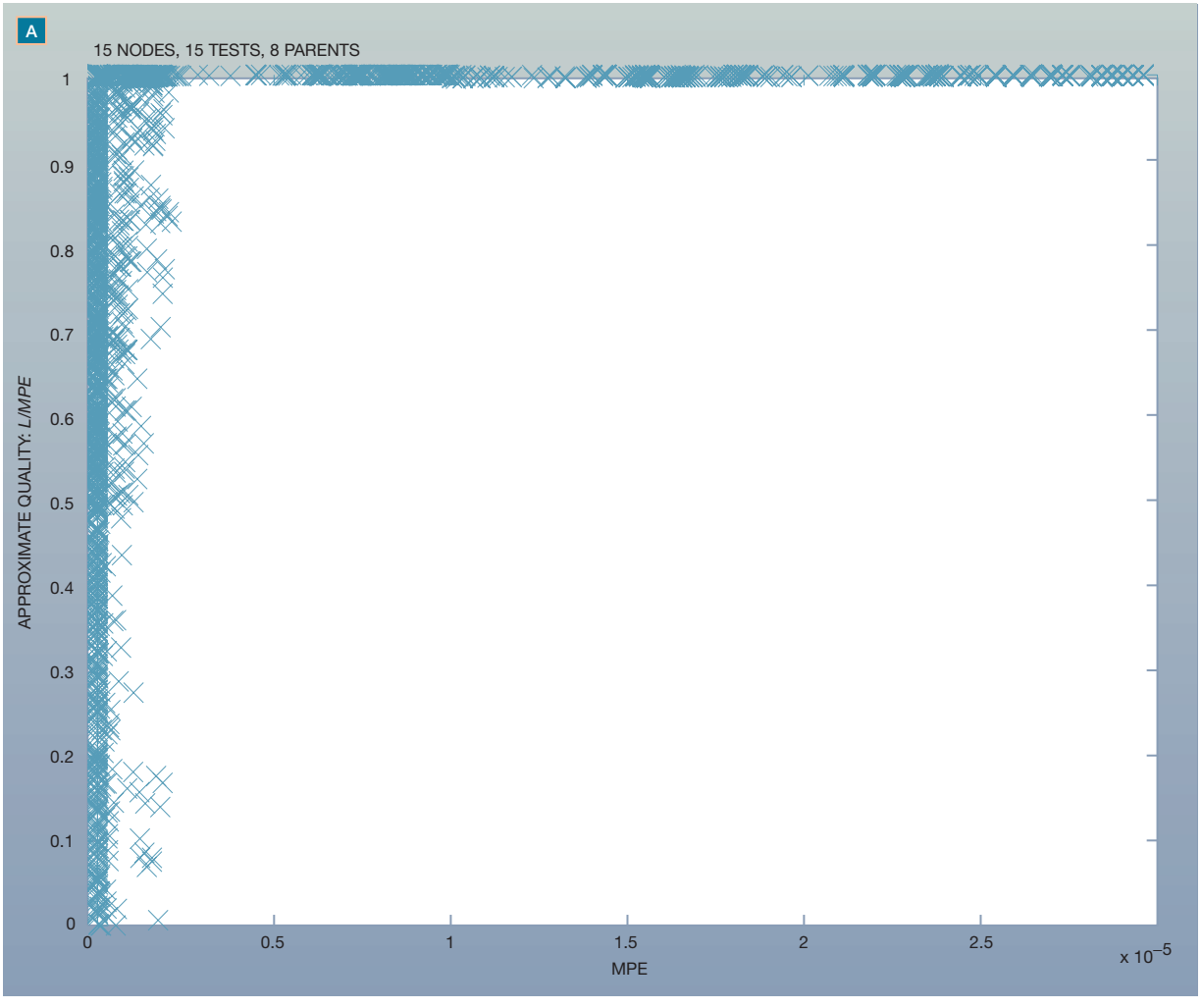
Also, as demonstrated in Figures 6A and B, there is a clear positive correlation between MPE value and approximation quality measured both as L/MPE (Figure 6A) and U/L (Figure 6A). There is also an interesting threshold phenomenon: the approximation quality suddenly increases to practically perfect ($L/MPE = 1$) once the MPE reaches a certain threshold value determined by the network parameters m, n, and r. The results are summarized for 30 "signals" per network, 30 random networks with $n = 15$ nodes, $n + n$ probes, $r = 8$ parents per probe, leak $l = 0$ and varying link q ("noise") from 0.005 to 0.64. Figure 6A shows a sharp transition in approximation quality for $MPE \approx 2e - 6$; similar results observed for other networks, where the "transition point" is determined by parameters n, m and p. Figure 6B shows that the lower bound L is often more accurate than the upper bound U (U/L is far from 1 when L/MPE is near 1).

Related work

The problem of fault diagnosis in a system of interconnected components dates back to the papers in References 22 and 23. Since that time a large body of literature has developed.[24] In contrast with that work, in our case it is not possible for every node in the network to be used to test other nodes—only a small number of nodes can be used as probe stations to generate the tests. As a result of this limitation, the probing problem becomes a "constrained-coding" problem, as explained above.

The formulation of problem diagnosis as a "decoding" problem, where "problem events" are decoded from "symptom events," was first proposed by Kliger et al.[6] In our framework, the result of a probe constitutes a "symptom event," whereas a node failure is a "problem event." However, beyond this conceptual similarity the two approaches are quite different. The major difference is that we use an active probing approach versus a "passive" analysis of symptom events: namely Kliger et al.[6] select codebooks (a combination of symptoms encoding particular problems) from a specified set of symptoms, and we actively construct those symptoms (probes), a much more flexible approach. Another important difference is that Kliger et al.[6] lack a detailed dis-

Figure 6 Approximation quality of algorithm *approx - mpe*(1) tends to be higher for higher MPE: (A) *L/MPE* versus MPE

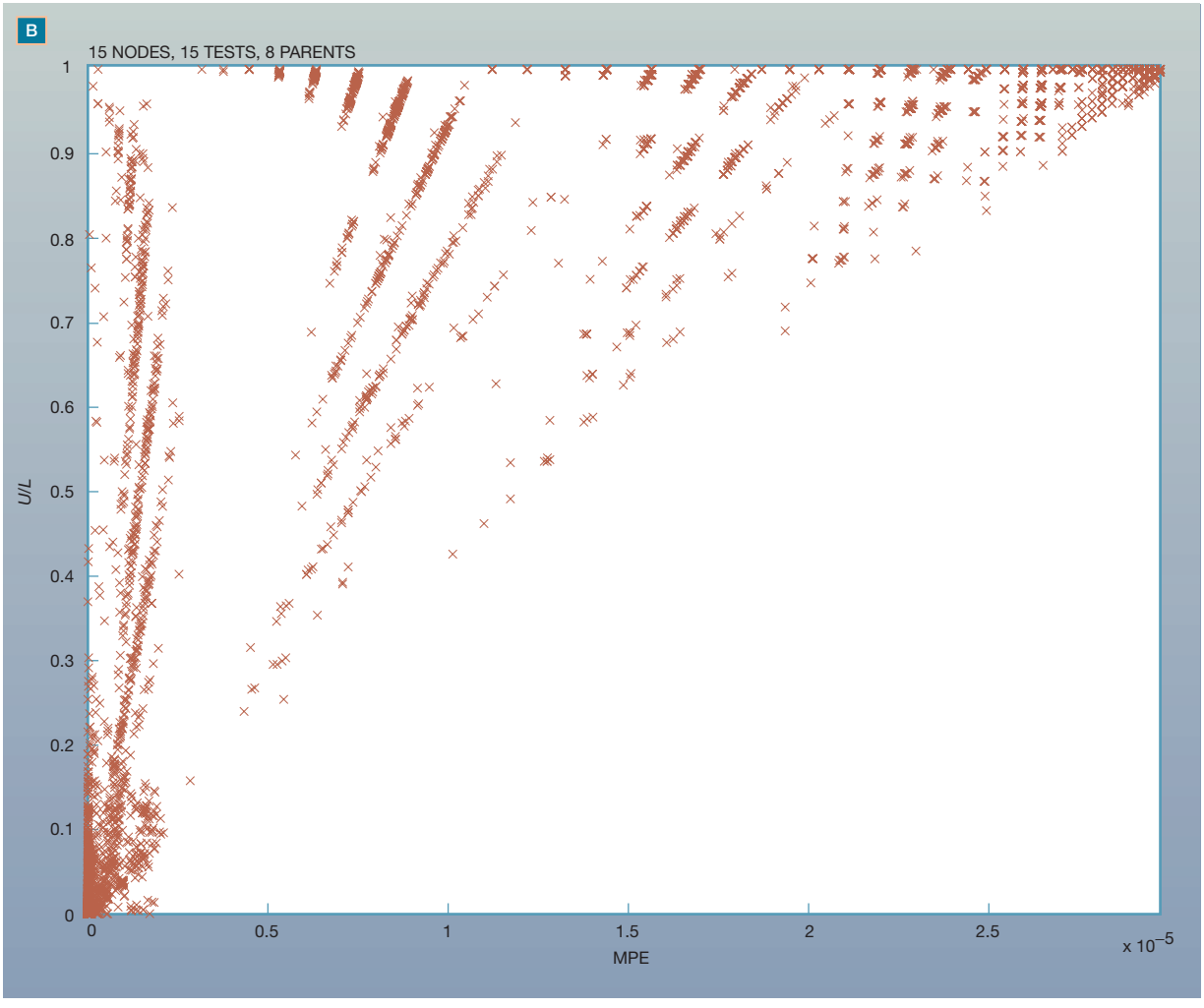


cussion of efficient algorithms for constructing optimal codebooks; they mention only a greedy pruning algorithm. For more detail on event correlation see also Leinwand and Fang-Conroy[25] and Gruschke.[26]

Other approaches to fault diagnosis in communication networks and distributed computer systems have been presented during the past decade, including Bayesian networks[27] and other probabilistic dependency models.[28] Another approach is statistical learning to detect deviations from the normal behavior of the network.[29]

In Huard and Lazar,[27] a decision-theoretic approach using Bayesian networks is presented. The goal is to find the minimum-cost diagnosis of problems occurring in a network. Dependencies between a problem and its possible causes and symptoms are represented using Bayesian networks, which are manually constructed for each problem, and probabilities are assigned using expert knowledge. The goal is to minimize the total cost of tests needed to diagnose a fault; a single fault at a time is assumed. This approach may become intractable in large networks because of the NP-hardness of inference in Bayesian networks; also, considering more than one

Figure 6 Continued: (B) *U*/*L* versus MPE



fault at a time leads to an exponential increase in complexity. Therefore, approximation methods as proposed in this paper will be needed in practical applications that involve a large number of dependent components.

The approach in Katzela and Schwartz[28] uses a graph model in which the prior and conditional probabilities of node failure are given and the objective is to find the most likely explanation of a collection of alarms. It is shown that the problem is NP-hard, and a polynomial-time approximation algorithm is given; the performance of this algorithm can be improved by assuming that the probabilities of node failure are independent of one another.

However, to the best of our knowledge, none of those previous works includes an active approach to probe set selection, which allows us to control the quality of diagnosis. Also, they lack a systematic study of diagnosis with a focus on using probes, which would include theoretical bounds on the diagnostic error, asymptotic behavior of diagnosis quality, and a systematic study of the quality of approximate solutions, as presented herein.

Conclusions

Using probing technology for the purposes of fault diagnosis in distributed computer systems requires that the number of probes be kept small, in order

to control network load and data storage costs. In this paper we have proposed a framework in which this can be done by exploiting interactions among the paths traversed by the probes. However, finding the smallest number of probes that can diagnose a particular set of problems is computationally expensive for large networks. We have shown that approximation algorithms can be used to find small probe sets that are very close to the optimal size and still suffice for problem diagnosis. These approximation algorithms enable system managers to select their own trade-off between the computational cost and probe set size needed for effective fault localization. Probing provides a flexible approach to fault localization because of the control that can be exercised in the process of probe selection.

Next, we extended the deterministic framework to a probabilistic one in order to handle uncertainties. We use a Bayesian network approach and investigate the accuracy versus efficiency trade-off when using approximate diagnostic techniques instead of exact ones, since the latter are often intractable for large networks. An empirical study of a local-inference approximation scheme demonstrates the promise of using such approximations for network diagnosis: the approximation quality "degrades gracefully" with increasing uncertainty ("noise level") and increases with the increasing quality of the probe set, which is measured by the information gain it provides about the unknown variables.

Finally, we are planning to pursue several directions for future work:

- Efficient search algorithms for finding a probe set able to diagnose an arbitrary combination of faults, in other words, an algorithm for constructing a set of constraints (probes) over the set of variables (node states) so that the number of possible solutions (diagnoses) to the resulting constraint-satisfaction problem are minimized (can also be viewed as constructing a good "code")
- Probe selection in the presence of uncertainty
- Adaptive probing, that is, adjusting the probe set dynamically in response to the state of the network
- Handling temporal information, such as changes in the state of the network (thus, changing probe results), and nonstationarities in the network statistics. The latter would require tuning the model, that is, on-line learning, which can also be done actively by using probe selection to improve learning (e.g., only update the part of the model that is currently relevant).

Cited references and notes

1. A. Frenkiel and H. Lee, "EPP: A Framework for Measuring the End-to-End Performance of Distributed Applications," *Proceedings of Performance Engineering 'Best Practices' Conference*, IBM Academy of Technology (1999).
2. *Using Keynote Measurements to Evaluate Content Delivery Networks*, KEYNOTE, 2855 Campus Drive, San Mateo, CA 94403 (2000); available at www.keynote.com/services/html/product lib.html.
3. J. Pearl, *Probabilistic Reasoning in Intelligent Systems*, Morgan Kaufmann Publishers, San Francisco, CA (1988).
4. If the problems are not marginally independent, appropriate edges must be added between them.
5. G. F. Cooper, "The Computational Complexity of Probabilistic Inference Using Bayesian Belief Networks," *Artificial Intelligence* **42**, Nos. 2–3, 393–405 (1990).
6. S. Kliger, S. Yemini, Y. Yemini, D. Ohsie, and S. Stolfo, "A Coding Approach to Event Correlation," *Proceedings of the Fourth International Symposium on Intelligent Network Management* (1995), pp. 266–277.
7. M. Brodie, I. Rish, and S. Ma, "Optimizing Probe Selection for Fault Localization," *Twelfth International Workshop on Distributed Systems Operation and Management* (October 15–17, 2001).
8. T. M. Cover and J. A. Thomas, *Elements of Information Theory*, John Wiley & Sons, Inc., New York (1991).
9. A paper on the NP-hardness result is to be submitted to the NIPS (Neural Information Processing Systems) 2002 Conference to be held December 10–12, 2002.
10. D. Heckerman and J. Breese, *Causal Independence for Probability Assessment and Inference Using Bayesian Networks*, Technical Report MSR-TR-94-08, Microsoft Research, Redmond, WA (1995).
11. Note that this noisy-AND definition is equivalent to the noisy-OR definition in Henrion et al.[12] if we replace every value by its logical negation (all zeroes will be replaced by ones and vice versa). We also note that instead of considering the leak probability separately, we may assume there is an additional "leak node" always set to zero that affects an outcome of a probe T_i according to its link probability $(1 - l_i)$.
12. M. Henrion, M. Pradhan, B. Del Favero, K. Huang, G. Provan, and P. O'Rorke, "Why Is Diagnosis Using Belief Networks Insensitive to Imprecision in Probabilities?" *Proceedings of the Twelfth Conference on Uncertainty in Artificial Intelligence* (1996), pp. 307–314.
13. Clearly, finding a set of probes that may actually achieve the bound, if such a set of probes exists, is a much harder task.
14. R. Dechter and J. Pearl, "Network-Based Heuristics for Constraint Satisfaction Problems," *Artificial Intelligence* **34**, 1–38 (1987).
15. Algorithm *Quickscore*,[16] specifically derived for noisy-OR networks, has the complexity $O(2^p)$, where p is the number of "positive findings" (failed probes in our case). However, the algorithm is tailored to belief updating and cannot be used for finding MPE.
16. D. Heckerman, "A Tractable Inference Algorithm for Diagnosing Multiple Diseases," *Proceedings of the Fifth Conference on Uncertainty in Artificial Intelligence* (1989), pp. 174–181.
17. R. Dechter and I. Rish, "A Scheme for Approximating Probabilistic Inference," *Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence* (1997), pp. 132–141.

18. K. Kask, I. Rish, and R. Dechter, "Empirical Evaluation of Approximation Algorithms for Probabilistic Decoding," *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence* (1998), pp. 455–463.
19. I. Rish, *Efficient Reasoning in Graphical Models*, Ph.D. thesis, University of California, Irvine, Irvine, CA (1999).
20. A closely related example is local belief propagation,[3] a linear-time approximation that became a surprisingly effective state-of-the-art technique in error-correcting coding.[21]
21. B. J. Frey and D. J. C. MacKay, "A Revolution: Belief Propagation in Graphs with Cycles," *Advances in Neural Information Processing Systems*, Vol. 10, M. I. Jordan, M. J. Kearns, and S. A. Solla, Editors, MIT Press, Cambridge, MA (1998).
22. F. P. Preparata, G. Metze, and R. T. Chien, "On the Connection Assignment Problem of Diagnosable Systems," *IEEE Transactions on Electronic Computers* **16**, No. 6, 848–854 (December 1967).
23. J. D. Russell and C. R. Kime, "System Fault Diagnosis: Closure and Diagnosability with Repair," *IEEE Transactions on Computers* **C-24**, No. 11, 1078–1089 (November 1975).
24. A. T. Dahbura, "System Level Diagnosis," *Concurrent Computation: Algorithms, Architectures, Technologies*, Plenum Publishers, New York (1988), pp. 411–434.
25. A. Leinwand and K. Fang-Conroy, *Network Management: A Practical Perspective*, 2nd Edition, Addison-Wesley Publishing Co., Reading, MA (1995).
26. B. Gruschke, "Integrated Event Management: Event Correlation Using Dependency Graphs," *Proceedings of the 9th IFIP/IEEE International Workshop on Distributed Systems Operations and Management* (1998).
27. J. F. Huard and A. A. Lazar, *Fault Isolation Based on Decision-Theoretic Troubleshooting*, Technical Report 442-96-08, Center for Telecommunications Research, Columbia University, New York (1996).
28. I. Katzela and M. Schwartz, "Fault Identification Schemes in Communication Networks," Vol. 3, *IEEE/ACM Transactions on Networking* (1995).
29. C.-S. Hood and C. Ji, "Proactive Network Fault Detection," *Proceedings of IEEE INFOCOM*, Kobe, Japan (April 7–12, 1997), pp. 1147–1155.

Accepted for publication April 25, 2002.

Mark Brodie *IBM Research Division, Thomas J. Watson Research Center, P.O. Box 704, Yorktown Heights, New York 10598 (electronic mail: mbrodie@us.ibm.com).* Dr. Brodie is a research staff member in the Machine Learning for Systems group at the T. J. Watson Research Center and an adjunct professor at Columbia University. He did his undergraduate work at the University of the Witwatersrand in South Africa. After coming to the United States, he received his Ph.D. degree in computer science in 2000 from the University of Illinois at Urbana-Champaign, working with Gerald DeJong on explanation-based learning, and has been at IBM since then. His research interests include machine learning, data mining, and intrusion detection.

Irina Rish *IBM Research Division, Thomas J. Watson Research Center, P.O. Box 704, Yorktown Heights, New York 10598 (electronic mail: rish@us.ibm.com).* Dr. Rish is a research staff member at the T. J. Watson Research Center, working in the Machine Learning for Systems group. She received her M.S. degree in applied mathematics in 1992 at the Moscow Gubkin Institute, Russia, and her Ph.D. degree in computer science in 1999 at the University of California, Irvine, working with Rina Dechter on efficient reasoning techniques for constraint networks and Bayesian networks. Her primary interests are probabilistic inference and learning using Bayesian networks and other statistical machine-learning techniques. She is also working on approximation algorithms for efficient inference in large networks. She currently leads the Intelligent Probing project, which aims at developing tools for cost-efficient, real-time diagnosis and prediction in distributed computer systems using probing technology.

Sheng Ma *IBM Research Division, Thomas J. Watson Research Center, P.O. Box 704, Yorktown Heights, New York 10598 (electronic mail: shengma@us.ibm.com).* Dr. Ma received his B.S. degree in electrical engineering from Tsinghua University, China, in 1992. He received M.S. and Ph.D. (with honors) degrees in electrical engineering from Rensselaer Polytechnic Institute in 1995 and 1998, respectively. He joined the T. J. Watson Research Center as a research staff member in 1998, where he is now manager of the Machine Learning for Systems Department. His current research interests include network and computer system management, machine learning, data mining, and network traffic modeling and control.