RC22369 (W0203-038) March 12, 2002
Computer Science

IBM Research Report

Efficient fault diagnosis using probing

Irina Rish, Mark A. Brodie, Sheng Ma
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598

Research Division
Almaden - Austin - Beijing - Delhi - Haifa - India - T. J. Watson - Tokyo - Zurich

Efficient fault diagnosis using probing

Irina Rish, Mark Brodie, Sheng Ma
IBM T.J. Watson Research Center, 30 Saw Mill River Road, Hawthorne, NY 10532
mbrodie,rish,shengma@us.ibm.com

Abstract

In this paper, we address the problem of efficient diagnosis in real-time systems capable of on-line information gathering, such as sending ”probes” (i.e., test transactions, such as ”traceroute” or ”ping”) in order to identify network faults and evaluate performance of distributed computer systems. We use a Bayesian network to model probabilistic relations between the problems (faults, performance degradation) and symptoms (probe outcomes). Due to intractability of exact probabilistic inference in large systems, we investigated approximation techniques, such as a local-inference scheme called mini-buckets(Dechter & Rish 1997). Our empirical study demonstrates advantages of local approximations for large diagnostic problems: the approximation is very efficient and ”degrades gracefully” with noise; also, the approximation error gets smaller on networks with higher confidence (probability) of the exact diagnosis. Since the accuracy of diagnosis depends on how much information the probes can provide about the system states, the second part of our work is focused on the probe selection task. Small probe sets are desirable in order to minimize the costs imposed by probing, such as additional network load and data management requirements. Our results show that, although finding the optimal collection of probes is expensive for large networks, efficient approximation algorithms can be used to find a nearly-optimal set.

Keywords: uncertainty management, Bayesian networks, approximate inference, adaptive optimization.

Introduction

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/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 email 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-connectivity, database- or web-access). A probe returns a set of measurements, such as response times, status code (OK/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. However, applying this technology to fault diagnosis and problem determination, as well as developing optimal strategies for adaptive, on-line probe scheduling, is still an open research area.

Fault diagnosis in real-life scenarios often involves handling noise and uncertainty. For example, a probe can fail even though all the nodes it goes through are OK (e.g., due to packet loss). 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 approach the problem of handling uncertainty by using the graphical framework of Bayesian networks (Pearl 1988) 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[1] and the bottom-layer nodes represent probe results. Since the exact inference in Bayesian networks is generally hard (NP-hard) (Cooper 1990), 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

[1] If the problems are not marginally independent, appropriate edges must be added between them.

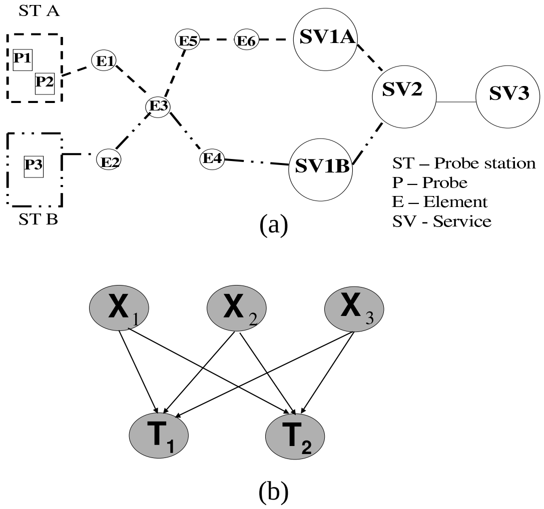


Figure 1: (a) An example of probing environment; (b) a two-layer Bayesian network structure for a set $\mathbf{X} = (X_1, X_2, X_3)$ of network elements and a set of probes $\mathbf{T} = (T_1, T_2)$.

in large networks.

Another important issue involved in the diagnosis is the selection of a most-informative probe set. 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 which station each probe should originate from. 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 yields a quadratic-time algorithm which finds near-optimal probe sets. We also implement a linear-time algorithm which can be used to find small probe sets very quickly; a reduction of almost 50% in the probe set size is achieved.

Problem formulation

We now 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 $\mathbf{X} = (X_1, ..., X_n)$ of *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). We also make an assumption that source routing is supported, i.e. we can specify the probe path in advance. A vector $\mathbf{T} = (T_1, ..., T_m)$ of *observed* boolean variables denoting the outcomes (0 - failure, 1 - OK) of m probes. Lower-case letters, such as x_i and t_j, denote the values of the corresponding variables, i.e. $\mathbf{x} = (x_1, ..., x_n)$ denotes a particular assignment of node states, and $\mathbf{t} = (t_1, .., t_m)$ 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 marginally independent. These assumptions yield a causal structure depicted by a two-layer Bayesian network, such as one in Figure 1b. The joint probability $P(\mathbf{x}, \mathbf{t})$ for such network can be then written as follows:

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

where $P(t_j|\mathbf{pa}(t_j))$ is the *conditional probability distribution (CPD)* of node T_i given the set of its *parents* $\mathbf{Pa_i}$, i.e. the nodes pointing to T_i in the directed graph, and $P(x_i)$ is the prior probability that $X_i = x_i$.

We now specify the quantitative part of those network, i.e. the CPDs $P(t_j|\mathbf{pa}(t_j)$. 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 long probe path (i.e. large parent set). It seems reasonable to assume that each element on the probe's path affects the probe's outcome independently, so that there is no need to specify the probability of T_i for all possible value combinations of $X_{i_1}, ..., X_{i_k}$ (the assumption known as *causal independence* (Heckerman & Breese 1995)). For example, in the absence of uncertainty, a probe fails if and only if at least one node on its path fails, i.e. $T_i = X_{i_1} \wedge ... \wedge X_{i_k}$, where $\wedge$ denotes logical AND, and $X_{i_1}, ..., X_{i_k}$ 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 OK (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 is failed, e.g. due to routing change. Such uncertainties yield a *noisy-AND* model which implies that several causes (e.g., node failures) contribute independently to a common effect (probe failure)and is formally defined as follows:

$$P(t = 1|x_1, \ldots, x_k) = (1 - l) \prod_{x_i = 0}^{n} q_i, \text{ and} \qquad (2)$$

$$P(t = 1|x_1 = 1, ..., x_k = 1) = 1 - l, \qquad (3)$$

where l is the *leak probability* which accounts for the cases of probe failing even when all the nodes on its path are OK, and the *link probabilities*, q_i, account for the second kind of "noise" in the noisy-AND relationship, namely, for cases when probe succeeds with a small probability q_i even if node X_i on its path fails[2].

[2] Note that this noisy-AND definition is equivalent to the *noisy-*

In a noise-free environment (i.e., when probe outcomes are logical-AND functions of the nodes they go through), the bipartite Bayesian network described above can be reduce to a simpler representation by *dependency matrix* $D(i,j)$, where $D(i,j) = 1$ if probe T_i passes through node X_j, and $D(i,j) = 0$ otherwise. Thus, D is an m-by-n matrix, where m is the number of probes. (This representation is motivated by the "coding" approach to event correlation suggested by (Kliger *et al.* 1997).)

Once a Bayesian network is specified, the diagnosis task can be formulated as finding the *maximum probable explanation (MPE)*, i.e. a most-likely assignment to all X_i nodes given the probe outcomes, i.e. $\mathbf{x}^* = \arg\max_{\mathbf{x}} P(\mathbf{x}|\mathbf{t})$. Since $P(\mathbf{x}|\mathbf{t}) = \frac{P(\mathbf{x},\mathbf{t})}{P(\mathbf{t})}$, where $P(\mathbf{t})$ does not depend on $\mathbf{x}$, we get $\mathbf{x}^* = \arg\max_{\mathbf{x}} P(\mathbf{x},\mathbf{t})$.

When there is no noise in noisy-AND (i.e. leak and link probabilities are zero), the CPDs become deterministic, i.e. each probe $T_i = t_i$ imposes a constraint $t_i = x_{i_1} \wedge ... \wedge x_{i_k}$ on the values of its parent nodes $X_{i_1}, ..., X_{i_k}$. Now, finding an MPE can be viewed as a constrained optimization problem of finding $\mathbf{x}^* = \arg\max_{x_1,...,x_n} \prod_{j=1}^{n} P(x_j)$ subject to those constraints. In a particular case of uniform priors $P(x_j)$, diagnosis is reduced to solving a constraint satisfaction problem. Clearly, the quality of diagnosis depends on the set of probes: 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.

Accuracy of diagnosis

In this section, we investigate the accuracy of the MPE diagnosis as a function of the number of probes and the noise parameters. A *lower* bound on the diagnosis error allows to reject the parameter regions that can never yield an asymptotically error-free diagnosis.

The *MPE error*, Err_{MPE}, is the probability that diagnosis $\mathbf{x}^*$ differs from the true state $\mathbf{x}$ (by at least one value x_i). Given an assignment $\mathbf{T} = \mathbf{t}$, $\mathbf{X} = \mathbf{x}$, and diagnosis $\mathbf{X}^* = \mathbf{x}^*$, we get $P(\mathbf{x} \neq \mathbf{x}*|\mathbf{t}) = I_{\mathbf{x} \neq \mathbf{x}*}$ where I_s is the *indicator function*, $I_s = 1$ if $s = true$ and $I_s = 0$ otherwise. Then the MPE error can be written as

$$Err_{MPE} = P(\mathbf{X} \neq \mathbf{X}^*|\mathbf{T}) = \sum_{\mathbf{x},\mathbf{t}} P(\mathbf{x},\mathbf{t}) I_{\mathbf{x} \neq \mathbf{x}^*|\mathbf{t}} =$$

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

where $\mathbf{x}^*$ is an MPE assignment.

Lemma 1 *Given Bayesian network defining a joint distribution* $P(\mathbf{x},\mathbf{t})$ *as specified by the equation 1, where all* $P(t_j|\mathbf{pa}(\mathbf{t_j}))$ *are noisy-AND CPDs having same link probability* q*, leak probability* l*, and the number of parents* r*, the MPE diagnosis error is bounded from below as follows:*

$$Err_{MPE} \geq L_M = 1 - (1-p)^n((1-l)(1-q^r)+1)^m. \qquad (5)$$

Proof. See (Rish *et al.* 2001) for details. ■

Note that in the absence of noise (l=0 and q=0) we get $Err_{MPE} \geq L_M = 1 - (1-p)^n 2^m$, 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 smaller number of probes. Namely, solving $\underline{L_M} \leq 0$ for m yields the necessary condition for zero lower bound, $m \geq -n \frac{log(1-p)}{log(1+(1-l)(1-q^r))}$, plotted in Figure 2a as a function of p. Generally, solving $L_M \leq 0$ for m provides a way of specifying the minimum necessary number of probes that yield zero lower bound for a specified values of other parameters[3].

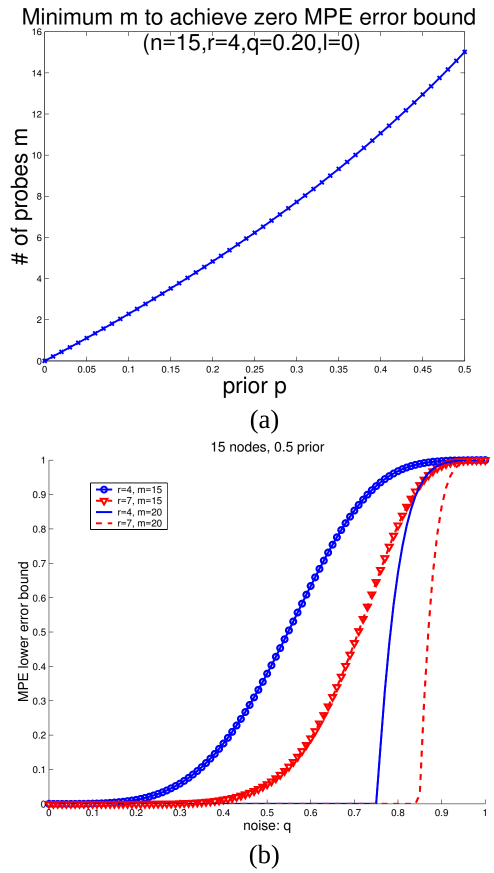


Figure 2: (a) Minimum number of probes m to guarantee zero error bound, versus fault prior p: low prior yields lower than $n = 15$ number of probes. (b) lower bound on MPE error versus link probability q ("noise"): the longer the probes (higher r), and the more of them (higher m), the lower the MPE error bound for fixed noise q; also, the sharper the transition from 0 to 100% error.

OR definition in (Henrion *et al.* 1996) if we replace every value by its logical negation (all 0's will be replaced by 1's 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 0 that affects an outcome of a probe T_i according to its link probability $(1 - l_i)$.

[3] Clearly, finding a set of probes that may actually *achieve* the bound, if such set of probes exists, is a much harder task.

Also, from the expression 5 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 increasing number of nodes n, fault probability p, leak probability l, and link probability q, but decreases with increasing number of probes m and probe route length r, which agrees with ones intuition that having more nodes on probe's path, 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 2b: note a quite sharp transition form 0 to 100%-error with increasing noise; it sharpness increases with increasing m and r.

Computational complexity and approximations

Let us first consider the complexity of diagnosis in the absence of noise. Finding the most-likely diagnosis is reduced to constraint satisfaction in the following 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 the uniform priors $P(x_i)$ which also yield the 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 $x_{i_1} \wedge ... \wedge x_{i_k} = 1$ which implies $x_i = 1$ for any node X_i on its path; the rest of the nodes are only included in constraints of the form $x_{i_1} \wedge ... \wedge x_{i_k} = 0$, or equivalently,$\neg x_{i_1} \wedge ... \wedge \neg x_{i_k} = 1$ imposed by failed probes. Thus, a $O(n)$-time algorithm assigns 1 to every node appearing on the path of a successful probe, and 0 to the rest of 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. Thus, finding MPE diagnosis is $O(n)$ time when it is equivalent to a constraint satisfaction in the absence of noise, as in cases of either uniform priors, or in case of unique diagnosis. In general, however, even in the absence of noise finding MPE is an NP-hard constrained optimization problem, with worst-case complexity $O(exp(n))$.

Similarly, in the presence of noise, finding MPE solution in a Bayesian network yields the complexity $O(exp(w^*))$ where w^* is the induced width of the network (Dechter & Pearl 1987), i.e. the size of 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 parents of a probe node, and $w^* = n$ in the worst case[4].

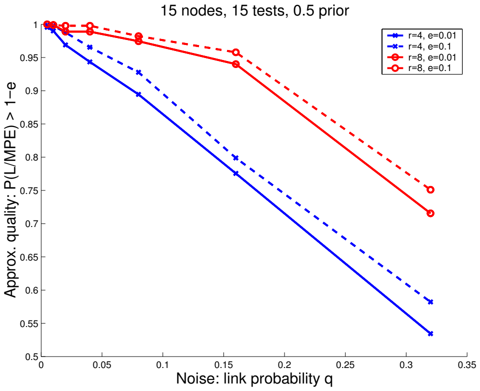


Figure 3: (a) ”Graceful degradation” of MPE approximation quality with noise, where the approximation quality is measured as $P(L/MPE) > 1 - e$ for $e = 0.01$ and $e = 0.1$: the quality is higher when the noise is smaller, and when the probe path is longer ($r = 8$ vs. $r = 4$).

Thus, we focused on approximating MPE, and studied empirically algorithm *approx-mpe(i)* (with $i = 1$, to be precise), which belongs to the family of *mini-bucket* approximations for general constrained optimization, and particularly, for finding MPE (Dechter & Rish 1997; Rish 1999). Generally, the mini-bucket algorithms *approx-mpe(i)* perform a limited level of variable-elimination, similar to enforcing directional i-consistency, and then greedily compute a suboptimal variable assignment in linear time[5]. The preprocessing allows to find an upper bound U on $M = \max_{\mathbf{x}} P(\mathbf{x}, \mathbf{t})$, where $\mathbf{t}$ is the evidence (clearly, $MPE = M/P(\mathbf{t})$), while the probability $L = P(\mathbf{x}', \mathbf{e})$ of their suboptimal solution provides an lower bound on M. Generally, L increases with the level of preprocessing controlled by i, thus allowing a flexible accuracy vs. efficiency trade-off. The algorithm returns the suboptimal solution $\mathbf{x}'$ and the upper and lower bounds, U and L, on M; ratio U/L is a measure of the approximation error.

(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 the 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). Note that in the absence of noise (i.e., for deterministic probe outcomes) *approx-mpe(1)* is equivalent to enforcing arc-consistency in a constraint network, or performing unit propagation on a propositional theory (Rish 1999), thus, in deterministic case, its solution coincides with the optimal one. Adding noise in a form of link probability q caused graceful degradation of

[4] Algorithm *Quickscore*(Heckerman 1989), 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.

[5] A closely related example is local belief propagation (Pearl 1988), a linear-time approximation which became a surprisingly effective state-of-the art technique in error-correcting coding (Frey & MacKay 1998).

the approximation quality, as shown in Figure 3. The figure summarizes the results for 50 randomly generated networks with $n = 15$ unobserved nodes (having uniform fault priors $p = P(x_i = 0) = 0.5$), $n = 15$ direct probes, one for each node, and $n = 15$ noisy-AND probes, each with $r = 4$ randomly selected parents among the unobserved nodes, zero leak $l = 0$ probability. The link probability (noise level) q varied from 0.01 to 0.64, taking 15 different values; the results are shown for all noise levels together. For each network, 100 instances of evidence (probe outcomes) were generated by Monte-Carlo simulation of $\mathbf{x}$ and $\mathbf{t}$ according to their conditional distributions.

Also, as demonstrated in Figures 4a and b, there is a clear positive correlation between MPE value and approximation quality measured both as L/MPE (Figure 4a) and U/L (Figure 4a); 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. An initial theoretical explanation of such phenomenon can be found in (Rish, Brodie, & Ma 2002a).

Probe set construction

In this section, we address the other important problem involved in fault diagnosis, namely, constructing the optimal probe set. Generally, the accuracy of diagnosis increases with increasing the number of probes (see the analytical results on the diagnostic error); however, in order to decrease the cost of probing (e.g., additional network load), we would like to achieve the maximum diagnostic accuracy with the minimum number of probes.

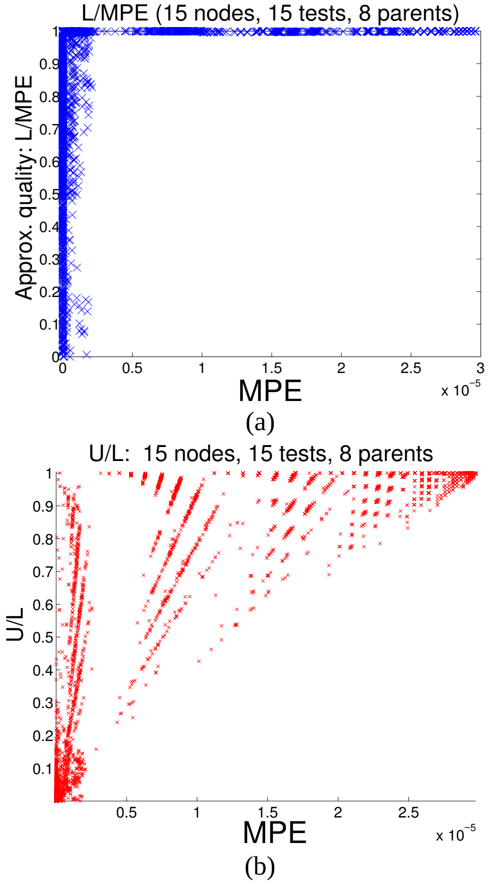
Assumptions: in this section, we only consider *noise-free* diagnosis, i.e. deterministic (logical-AND) dependence of each probe outcome on the node states, and will therefore use the dependency matrix representation of the corresponding Bayesian network. Another simplifying assumption is that the probability of simultaneous faults of different nodes is very small and can be ignored; thus, no more than one node can fail at a time. (In other words, we impose a constraint on the prior distribution $P(\mathbf{X})$, assuming that $P(\mathbf{x}) = 0$ if $\sum_{i=1}^{n} x_i > 1$.) However, this assumption does not restrict the generality of our approach and is dictated only by computational reasons since the complexity of our algorithms is linear in the number of possible fault combinations, which is $O(2^n)$ in the unrestricted case.

We will further use the example in Figure 5 as an illustration of our approach. Here we assume that one probe is sent along the path $N_1 \to N_2 \to N_5$ while another is sent along the path $N_1 \to N_3 \to N_6$. The resulting dependency matrix is shown to the right of the network (probes are indexed by their start and end nodes).

Problem Statement

Our objective is to find a minimal subset of probes (given set of all possible probes) that can always diagnose a single failed node.

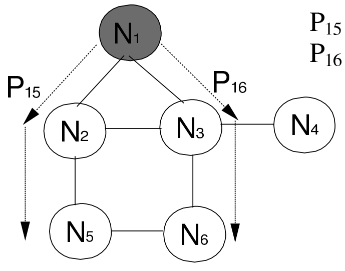
We can formulate probe selection as a **constrained optimization** problem, as follows. Let P denote the initial probe



(a)

(b)

Figure 4: Approximation quality of algorithm approx-mpe(1) tends to be higher for higher MPE, i.e. for more likely diagnosis: a) L/MPE vs. MPE and b) U/L vs. MPE. Note: a) 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; b) lower bound L is often more accurate than the upper bound U (U/L is far from 1 when L/MPE is near 1).



	N_1	N_2	N_3	N_4	N_5	N_6
P_{15}	1	1	0	0	1	0
P_{16}	1	0	1	0	0	1

Figure 5: An example network and dependency matrix.

set P, and let D denote the dependency matrix. 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 j'th sub-column 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:
$h(P') = \sum_1^n c_j$, $c_j = 1$ if C_j is distinct from C_1, ..., C_{j-1} (otherwise $c_j = 0$).

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

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; other considerations, such as gaining access to the machines, may be more important for choosing probe stations.) A probe can be sent to any node from any probe station. Thus the candidate set of probes could conceivably contain a probe for every possible route between every probe station and every node. In practice it cannot be guaranteed that a probe follows a particular path through the network, and thus routing strategies restrict the set of available probes; for example a probe may follow the least cost path through the network. In the example we will assume that the probe follows the shortest path from probe-station to target. This creates a candidate set of probes of size $m = O(n)$; 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.

Diagnostic ability of a set of probes

The dependency matrix decomposes the network into a disjoint collection of nodes, where each group consists of the nodes N_j whose columns C_j are identical; i.e. each group contains those nodes whose failure cannot be distinguished from one another by the given set of probes. A naive approach to computing this decomposition would compare each column with every other column. We can do better by proceeding row-by-row and computing the decomposition incrementally. The key is that adding a row (i.e. 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.

For example, let us consider the probe set defined for the network having 6 nodes denoted $N1, ..., N6$ (the column NF denotes the additional "no failure" situation)

	N1	N2	N3	N4	N5	N6	NF
P15	1	1	0	0	1	0	0
P16	1	0	1	0	0	1	0

It is easy to see that the corresponding decomposition is (using N_7 as the extra "node" representing "no failure")

S={{1}, {2,5}, {3,6}, {4,7}}

Suppose we add the probe $P_{42} : N_4 \to N_3 \to N_2$, giving the following dependency matrix:

	N1	N2	N3	N4	N5	N6	NF
P15	1	1	0	0	1	0	0
P16	1	0	1	0	0	1	0
P42	0	1	1	1	0	0	0

To compute the new decomposition we traverse the current decomposition and split each subset into further subsets depending on whether or not the new probe passes through each node. For example, since P_{42} passes through N_2 but not N_5, the subset $\{2, 5\}$ in DP gets split into $\{2\}$ and $\{5\}$. The new decomposition is

S={{1}, {2}, {3}, {4}, {5}, {6}, {7}}

Note that since each column is unique, any single node failure among the 6 nodes can be uniquely diagnosed by this set of three probes.

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

We define the **diagnostic ability** $H(P)$ of a set of probes P to be the expected minimal number of further probes needed to uniquely diagnose all nodes. The decomposition S induced by P is a disjoint collection of k groups; each group contains nodes whose failures cannot be distinguished from one another. Let n_i be the number of nodes in group g_i. Note that 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 given by

$$H(P) \quad = \quad \sum_{i=1}^{k} \frac{n_i}{n} log(n_i) \qquad (6)$$

Alternatively, if $N = \{1, ..., n\}$ is the variable denoting the node, and $G = \{1, ..., k\}$ is the variable denoting which group contains the node, then the diagnostic ability can be regarded as the conditional entropy $H(N|G) = \sum_{i=1}^{k} p(G = g_i) H(N|G = g_i)$ (by definition in (Cover & Thomas 1991)). Simple algebraic manipulations (see (Rish, Brodie, & Ma 2002b) for the derivation) yield:

$$H(N|G) = \sum_{i=1}^{k} \frac{n_i}{n} log(n_i).$$

Note that lower values for $H(P)$ correspond to better probe sets. For example, $H(S_1) = H(\{\{1, 2\}, \{3, 4\}\}) = \frac{1}{2} log 2 + \frac{1}{2} log 2 = log 2 = 1$, while $H(S_2) = H(\{\{1\}, \{2, 3, 4\}\}) = \frac{1}{4} log 1 + \frac{3}{4} log 3 = \frac{3}{4} log 3 = 1.19$.

This formula for $H(S)$ 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.

Finding the Minimal Set of Probes

We now investigate the question of finding the minimal set of probes that has the same diagnostic ability as the initial set. Clearly, the minimal set of probes may not be unique, although the minimal number of probes is.

We now examine algorithms for finding the minimal probe set. Since exhaustive search is easily seen to require exponential-time and 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 in Section .

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*; i.e. 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; this gives a computational complexity of $O(m)$, which is $O(n)$ if $m = O(n)$, as it was mentioned above.

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 in Section . The additive algorithm starts with the empty set and repeatedly adds the probe which gives the decomposition of highest diagnostic ability. This algorithm also finds a non-optimal probe subset with the same diagnostic ability as the original set.

The running time of this algorithm is quadratic in m, 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. This gives a computational complexity of $O(n^2)$ if $m = O(n)$.

Experiments

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

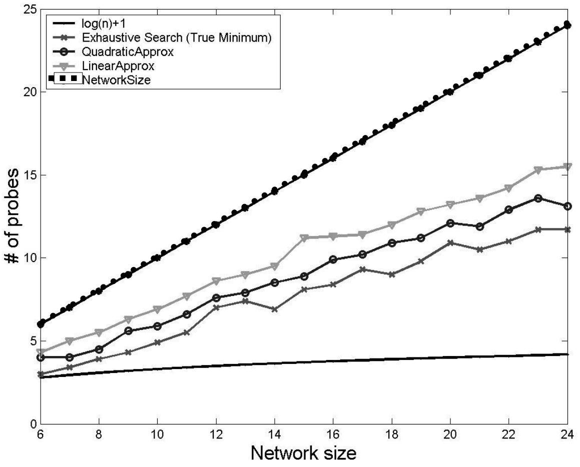


Figure 6: Three Algorithms for Computing Probe Sets.

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 in the previous sections are then executed. This process is repeated ten times for each network size and the results averaged.

Figure 6 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 short, passing through few nodes. This reduces the 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. Experiments on larger networks having up to 150 nodes for which exhaustive search is not feasible (Figure 7), showed that 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. The savings are almost 50% over the naive approach of sending a probe to every node.

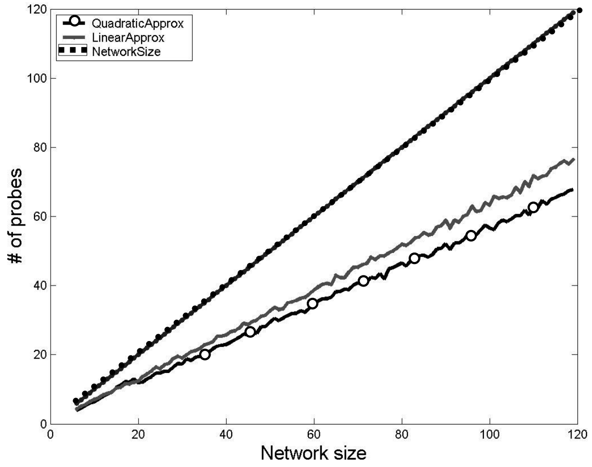


Figure 7: Approximation algorithms on large networks.

Related Work

The formulation of problem diagnosis using a matrix approach, where ”problem events” are ”decoded” from ”symptom events”, was first proposed by (Kliger *et al.* 1997). In our framework, the result of a probe constitutes a ”symptom event”, while a node failure is a ”problem event”. The major difference between the two approaches is that we use an active probing approach versus a ”passive” analysis of symptom events. Another important difference is that (Kliger *et al.* 1997) lacks a detailed discussion of efficient algorithms.

Approaches using probabilistic graphical models to find the most likely explanation of a collection of alarms have been suggested by (Gruschke 1998; I.Katzela & M.Schwartz 1995; Hood & Ji 1997). 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, it lacks 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

In this paper, we address both theoretically and empirically the problem of the most-likely diagnosis given the observations (MPE diagnosis), studying as an example the fault diagnosis in computer networks using probing technology. The key efficiency issues include minimizing both the number of tests and the computational complexity of diagnosis while maximizing its accuracy. Herein, we derive a bound on the diagnosis accuracy and analyze it with respect to the problem parameters such as noise level and the number of tests, suggesting feasible regions when an asymptotic (with problem size) error-free diagnosis can be achieved. Since the exact diagnosis is often intractable, we also provide an empirical study of some efficient approximate diagnosis algorithms and a theoretical analysis of their behavior with increasing noise. Our studies demonstrate a ”graceful degradation” of the approximation accuracy with increasing noise and suggest the applicability of such approximations to nearly-deterministic diagnosis problems that are often encountered in practical applications.

Since the accuracy of diagnosis depends on how much information the probes can provide about the system states, the second part of our work is focused on the probe selection task. Small probe sets are desirable in order to minimize the costs imposed by probing, such as additional network load and data management requirements. Our results show that, although finding the optimal collection of probes is expensive for large networks, efficient approximation algorithms can be used to find a nearly-optimal set. Extending our probe selection approach to noisy environments is a direction for future work.

References

Cooper, G. 1990. The computational complexity of probabilistic inference using Bayesian belief networks. *Artificial Intelligence* 42(2–3):393–405.

Cover, T., and Thomas, J. 1991. *Elements of information theory*. New York:John Wiley & Sons.

Dechter, R., and Pearl, J. 1987. Network-based heuristics for constraint satisfaction problems. *Artificial Intelligence* 34:1–38.

Dechter, R., and Rish, I. 1997. A scheme for approximating probabilistic inference. In *Proc. Thirteenth Conf. on Uncertainty in Artificial Intelligence (UAI97)*.

Frey, B., and MacKay, D. 1998. A revolution: Belief propagation in graphs with cycles. *Advances in Neural Information Processing Systems* 10.

Gruschke, B. 1998. Integrated Event Management: Event Correlation Using Dependency Graphs. In *DSOM*.

Heckerman, D., and Breese, J. 1995. Causal independence for probability assessment and inference using Bayesian networks. Technical Report MSR-TR-94-08, Microsoft Research.

Heckerman, D. 1989. A tractable inference algorithm for diagnosing multiple diseases. In *Proc. Fifth Conf. on Uncertainty in Artificial Intelligence*, 174–181.

Henrion, M.; Pradhan, M.; Favero, B. D.; Huang, K.; Provan, G.; and O'Rorke, P. 1996. Why is diagnosis using belief networks insensitive to imprecision in probabilities? In *Proc. Twelfth Conf. on Uncertainty in Artificial Intelligence*.

Hood, C., and Ji, C. 1997. Proactive network fault detection. In *Proceedings of INFOCOM*.

I.Katzela, and M.Schwartz. 1995. Fault identification schemes in communication networks. In *IEEE/ACM Transactions on Networking*.

Kliger, S.; Yemini, S.; Yemini, Y.; Ohsie, D.; and Stolfo, S. 1997. A coding approach to event correlation. In *Intelligent Network Management (IM)*.

Pearl, J. 1988. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann.

Rish, I.; Brodie, M.; Wang, H.; and Ma, S. 2001. Efficient fault diagnosis using local inference. Technical Report RC22229, IBM T.J. Watson Research Center.

Rish, I.; Brodie, M.; and Ma, S. 2002a. Accuracy versus efficiency in probabilistic diagnosis. Technical report, IBM T.J. Watson Research Center.

Rish, I.; Brodie, M.; and Ma, S. 2002b. Intelligent probing: a Cost-Efficient Approach to Fault Diagnosis in Computer Networks. *Submitted to IBM Systems Journal*.

Rish, I. 1999. *Efficient reasoning in graphical models*. PhD thesis.