Active Probing Strategies for Problem Diagnosis in Distributed Systems

Mark Brodie, Irina Rish, Sheng Ma, Natalia Odintsova
IBM T.J. Watson Research Center, 19 Skyline Drive, Hawthorne, NY 10532
mbmdie, rish, shengma@us.ibm.com, nodintsova@hotmail.com

Abstract

We address the task of problem determination in a distributed system using *probes,* or test transactions, which gather information about system components. Effective probing requires minimizing the cost of probing while maximizing the diagnostic accuracy of the probe set. We show that pre-planning an optimal probe set is NP-hard and present polynomial-time approximation algorithms that perform well. We then implement an active probing strategy which selects probes dynamically and show that it yields a significant reduction in probe set size in both simulation and a real system environment.

Active Probing System

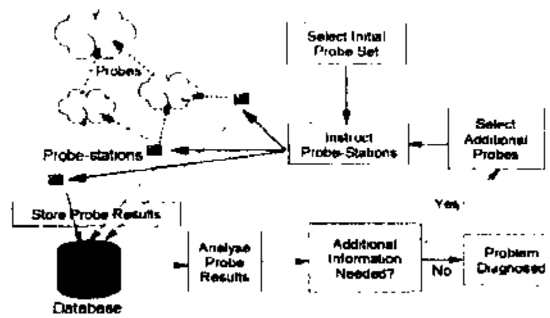


Figure 1: Active Probing System.

1 Introduction

Accurate diagnosis in a complex, multi-component system by making inferences based on the results of various tests and measurements is a common practical problem. Developing cost-effective techniques for diagnosis in such systems requires that high accuracy be achieved with a small number of tests. In this work we present a generic approach to this problem and apply it specifically to the area of distributed systems management.

The key component of our approach is an "active" measurement approach, called *probing.* A probe is a test transaction whose outcome depends on some of the system's components. Diagnosis is performed by appropriately selecting the probes and analyzing the results. For distributed systems, a probe is a program that executes on a particular machine (called a probe station) by sending a command or transaction to a server or network element and measuring the response. The *ping* program is probably the most popular probing tool that can be used to detect network availability. Other probing tools, such as IBM's EPP technology, provide more sophisticated, application-level probes. For example, probes can be sent in the form of test e-mail messages, web-access requests, and so on.

Previous work studied the problem of probe-selection [Brodie *et al,* 2001], [Ozmutlu *et al,* 2002] and efficient diagnosis from probe outcomes using approximate inference in Bayesian networks [Rish *et al,* 2002], but the NP-hardness of probe-selection and the idea of active probing were not considered there.

2 Pre-Pianned Probing

In pre-planned probing, given an initial set of probes, we want to compute the smallest subset of probes such that each system state will produce a different set of probe outcomes, allowing the state to be uniquely determined.

Proposition 1 *Probe-set selection is NP-hard.*

Proof: The proof is via a reduction from 3-Dimensional Matching (defined in [Garey and Johnson, 1979]). Details can be found in [Brodie *et al,* 2003].

We implemented two approximation algorithms for pre-planned probing - greedy (quadratic) search and subtractive (linear) search. Greedy search starts with the empty set and adds at each step the "best" of the remaining probes - the probe which maximizes the information gained about the system state. Computational complexity is quadratic in the size of the initial probe set. Subtractive search starts with the complete set of available probes, considers each one in turn, and discards it if it is not needed. Computational complexity is linear in the size of the initial probe set. Neither algorithm is optimal in general - empirical results are provided below.

Active Probing

Input: A set of available probes $\mathbf{P}$ and a prior distribution over system states $Pr(\mathbf{X})$.
Output: A set $\mathbf{P}_a$ of probes and their outcomes, posterior distribution $Belief(\mathbf{X})$ and its support $\mathbf{S}$.
Initialize: $Belief(\mathbf{X}) = Pr(\mathbf{X})$, $\mathbf{P}_a = \emptyset$, $\mathbf{S}$ = support of $Pr(\mathbf{X})$.
do
1. select current most-informative probe: $Y = \arg\max_{Y \in \mathbf{P} \setminus \mathbf{P}_a} I(\mathbf{X}; Y|\mathbf{P}_a)$
2. execute Y; it returns $Y = y$ (0 or 1)
3. update $\mathbf{P}_a = \mathbf{P}_a \cup \{Y = y\}$
4. update $Belief(\mathbf{X}) = Pr(\mathbf{X}|\mathbf{P}_a)$

while $\exists Y \in P$ such that $I(\mathbf{X}; Y|\mathbf{P}_a) > 0$
Return $\mathbf{P}_a$, $Belief(\mathbf{X})$, S =support of $Belief(\mathbf{X})$.

Figure 2: Algorithm for Diagnosis using Active Probing.

3 Active Probing

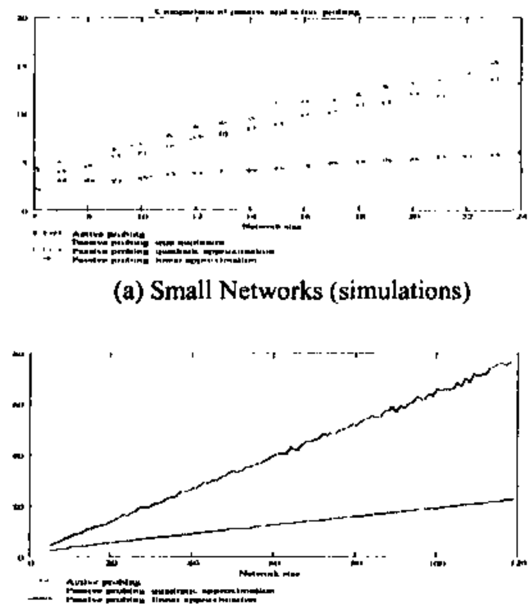
Active probing, allows the selection of later probes to depend on the results of earlier probes. An active probing system is outlined in Figure 1. Probe-stations issue probes which traverse different parts of the network. The results of the probes are analyzed to infer what problems might be occurring. If additional information is needed in order to locate the problem, the "most-informative" probe, which provides the largest information gain about the true state of the system, is computed and sent. When additional probe results are received further inferences are made, and the process repeats until the fault is localized. The algorithm is described in Figure 2.

Active probing allows fewer probes to be used than if the entire probe set has to be pre-planned, though more complex inferential machinery is needed to support it.

4 Results

Simulation results, using randomly generated networks, are shown in Figure 3a (small networks) and Figure 3b (large networks, where finding the true minimum is impractical). The approximation algorithms for finding the smallest probe set perform well and are close to the true minimum set found by exhaustive search. It is also clear that active probing considerably reduces the size of the probe set when compared with pre-planned, or "passive", probing.

In Figure 4, we report the results on a real network. This network, supporting e-business applications, includes many servers and routers, and its performance and availability depend on a number of software components. A set of 29 probes was manually selected by an expert. Exhaustive search shows that the minimum number of probes is actually only 24 probes, a saving of 17%. Greedy search returned 24 probes, while subtractive search found 25 probes. For active probing, Figure 3c shows the number of probes needed for localization for each of the different single faults. The number of probes needed never exceeds 16 probes and averaged only 7.5 probes, a large improvement over the 24 probes used by pre-planned probing.



(a) Small Networks (simulations)

(b) Large Networks (simulations)

Figure 3: Active versus pre-planned probing results for randomly generated networks.

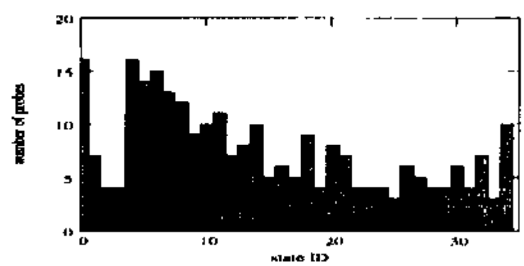


Figure 4: Active probing averages 7.5 probes in a real application where pre-planned probing requires 24 probes.

References

[Brodie te al., 2001] M. Brodie, I. Rish, and S. Ma. Optimizing probe selection for fault localization. In *Distributed Systems Operation and Management,* 2001.

[Brodie *et al,* 2003] M Brodie, I. Rish, Sheng. Ma, and N. Odintsova. Strategies for Problem Determination Using Probing. Technical report, IBM T.J. Watson Research Center, 2003.

[Garey and Johnson, 1979] M.R. Garey and D.S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-completeness.* W.H. Freeman and Co., San Francisco, 1979.

[Ozmutlu et al., 2002] H.C. Ozmutlu, N. Gautam, and R. Barton. Zone recovery methodology for probe-subset selection in end-to-end network monitoring. In *Network Operations and Management Symposium,* pages 451-464, 2002.

[Rish *et al.,* 2002] I. Rish, M. Brodie, and S. Ma. Accuracy vs. Efficiency Trade-offs in Probabilistic Diagnosis. In *Proceedings of the The Eighteenth National Conference on Artificial Intelligence (AAAI-2002), Edmonton, Alberta, Canada,* 2002.