

April 2012

EFFICIENT PROBE STATION PLACEMENT AND PROBE SET SELECTION FOR FAULT LOCALIZATION

SHAKTI KINGER

Computer Science Dept., Maharashtra Institute of Technology, Pune 411038, INDIA.,
shakti.kinger@mitpune.edu.in

B.M. PATIL

Computer Science Dept., Maharashtra Institute of Technology, Pune 411038, INDIA.,
balaji.patil@mitpune.edu.in

Follow this and additional works at: <https://www.interscience.in/ijcns>



Part of the [Computer Engineering Commons](#), and the [Systems and Communications Commons](#)

Recommended Citation

KINGER, SHAKTI and PATIL, B.M. (2012) "EFFICIENT PROBE STATION PLACEMENT AND PROBE SET SELECTION FOR FAULT LOCALIZATION," *International Journal of Communication Networks and Security*: Vol. 1 : Iss. 4 , Article 11.

Available at: <https://www.interscience.in/ijcns/vol1/iss4/11>

This Article is brought to you for free and open access by Interscience Research Network. It has been accepted for inclusion in International Journal of Communication Networks and Security by an authorized editor of Interscience Research Network. For more information, please contact sritampatnaik@gmail.com.

EFFICIENT PROBE STATION PLACEMENT AND PROBE SET SELECTION FOR FAULT LOCALIZATION

SHAKTI KINGER¹ & B.M. PATIL²

^{1,2}Computer Science Dept., Maharashtra Institute of Technology, Pune 411038, INDIA.
E-mail : shakti.kinger@mitpune.edu.in & balaji.patil@mitpune.edu.in

Abstract – Network fault management has been a focus of research activity with more emphasis on fault localization – zero down exact source of a failure from set of observed failures. Fault diagnosis is a central aspect of network fault management. Since faults are unavoidable in communication systems, their quick detection and isolation is essential for the robustness, reliability, and accessibility of a system.

Probing technique for fault localization involves placement of probe stations (Probe stations are specially instrumented nodes from where probes can be sent to monitor the network) which affects the diagnosis capability of the probes sent by the probe stations. Probe station locations affect probing efficiency, monitoring capability, and deployment cost. We present probe station selection algorithms and aim to minimize the number of probe stations and make the monitoring robust against failures in a deterministic as well as a non-deterministic environment. We then implement algorithms that exploit interactions between probe paths to find a small collection of probes that can be used to locate faults. Small probe sets are desirable in order to minimize the costs imposed by probing, such as additional network load and data management requirements. We discuss a novel integrated approach of probe station and probe set selection for fault localization. A better placing of probe stations would produce fewer probes and probe set maintaining same diagnostic power. We provide experimental evaluation of the proposed algorithms through simulation results.

Keywords — Adaptive probing, Probe station selection, Fault diagnosis, Network monitoring, Probabilistic dependency model.

I. INTRODUCTION

With increasing complexity in computer networks, effective network management has become even more crucial and challenging. The network management aims at ensuring networks are monitored and kept running as smoothly as possible. Network monitoring generates huge information that needs to be processed and diagnosed to detect/localize the failure. This information is generated by either monitoring tools [1,2,3,4,5] or by network entities themselves (in the form of alarms) [6,7,8,9]. Fault Management system broadly deploys two types of monitoring (1) Active Monitoring – actively send probes to gather performance data (2) Passive Monitoring – rely on network devices to send alarms, as shown in fig-1. Both approaches have their own advantages and bear their own limitations. Combined, they are used to effectively solve network management problem.

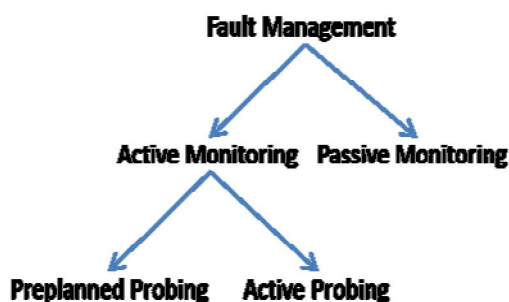


Figure 1 Fault management technologies

A. Active Monitoring

Active monitoring deploys probing methods to gather health status and performance statistics of network entities in the managed system. The main component of probing-based techniques is a sample measurement called *probe*. A probe is basically a dedicated program (such as ping or traceroute) or an application entity (such as email or web access). These probes are installed, sent and their results analyzed from network nodes called as *probing station*. A probe is periodically sent to examine a subset of network nodes in the managed system. Once a probe is sent to the network it either successfully returns to its probing station, signifying that all the network nodes in its path are in working order, or it fails to return to its probing station, indicating that one node or more in its path are in a failure state

Probing based techniques have various advantages over passive monitoring techniques, such as (1) less instrumentation (2) capability to compute end-to-end performance (3) quicker localization, etc. Developing probing based monitoring solution involves solving two major problems, namely probe station selection and probe set selection. The probe station selection problem addresses the problem of selecting minimum subset of nodes in the managed network where probe stations should be placed such that the required diagnosis capability can be achieved through probes. Probe station selection is followed by task to select optimal probes such that any failure in network can be detected and localized.

Different criteria's are imposed on probe set

selection for fault detection and fault localization [4]. Probe set for fault detection is selected such that all elements in the managed network are probed. On the other hand, fault localization requires minimal probe set that can uniquely diagnose the suspected network element failure. Probes for failure detection are sent periodically and thus the management traffic produced should be low enough that it does not affect the performance of other applications. Moreover the time constraints on probe set selection for failure detection are less stringent than that for fault localization. Fault localization is done only when some problem is encountered. Thus probes for fault localization should be selected such that the fault localization can be done in minimum amount of time and at the same time the network in the identified problem areas should not be overwhelmed with the management traffic.

B. Preplanned Probing

Preplanned probing involves offline selection of probes those are periodically sent out in the network [2]. The results are then analyzed to infer the network state. This approach requires probe set selection such that every failure in the network can be uniquely localized. It is practically difficult envisaging all possible failures that might occur and come up with probe set to detect those failures. Also, sending this large number of probes at a periodic interval generates large amount of management traffic. Moreover, large part of this network traffic can be waste as many problems that are envisaged may not ever happen. Another disadvantage of this approach is that because probes are sent at periodically at scheduled intervals, there might be considerable delay in obtaining information when problem occurs. As it is desirable to detect and localize failures immediately, this delay might not be acceptable. Moreover, this delay will potentially delay in next step of fault localization.

C. Active Probing

It initially selects probes for fault detection [2]. The probe stations send these probes and observe the network. Additional probes are sent out to obtain further information about the problem, and this process may repeat - as more data is obtained, decisions are made as to which probes to send next, until finally the problem is completely determined. It greatly reduces management traffic and provides more accurate and timely diagnosis.

Active probing implementation involves developing solutions for the following issues:

- An initial minimum probe set must be pre-selected for any problem detection in network.
- The network state is determined by analyzing probe results.
- The probes to send next must be selected such that it should be “most-informative”, based on the analysis of previous probe results.

- This process must be repeated until the problem diagnosis task is complete.

II. APPROACH FOR PROBE STATION SELECTION

In this section, we present an algorithm that incrementally selects nodes which provide a suitable location to instantiate a probe station. The algorithm is based on the concept that to diagnose k failures in a network, the probe stations should be placed such that each node can be probed through k independent (node disjoint) paths.

A. Assumptions

Our algorithm for probe station selection is based on the assumption that there only exist node failures in a network. However, this approach can be extended to monitor link failures as well. We assume that network has a static single path routing model and there are no loops in the routing model.

We place a limit on the maximum number of node failures that can be diagnosed. In a connected network consisting of k failures, a set of probe stations can localize any k non-probe-station node failures if and only if there exists k independent probe paths to each non-probe-station nodes.

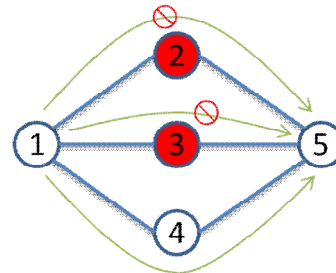


Figure 2 k Independent paths allow detection of k node failures

Figure 2 shows 3 independent (node disjoint) paths to node 5 from probe station 1. Even if there are failures in two paths, node 5 can still be probed.

We also assume that probe stations are not required to be fault tolerant. However, with our approach probe stations are selected such that there exists k independent paths to each of probe station as well.

B. Probe Station Algorithm

We model the network by an undirected graph $G(V, E)$, where the graph nodes, V , represent the network nodes (routers, end hosts) and the edges, E , represent the communication links connecting the nodes. We use $P_{u,v}$ to denote the path traversed by a probe from a source node u to a destination node v .

Probe Station Selection: find the set $Q \subseteq V$ of least cardinality such that every node $u \in \{V - Q\}$ has k independent paths from the nodes in Q .

Initially the selected probe station set is empty and all nodes belong to the uncovered node set. Selecting highest degree node as first probe station can remove large number of nodes from uncovered node set. However, from its spanning tree it is observed that such a probe station results into large number of shorter probes. This results into larger probe set size that is required to localize a failure. Therefore, we don't select highest degree node as first probe station; instead we select one of its neighbor that has got least number of neighbors having node number less than the max degree node.

When only one probe station has been selected, all nodes that are not neighbors of the selected probe station belong to the set of uncovered nodes. All the nodes that do not belong to the selected probe station set are candidates for the next probe station selection. For each candidate probe station, the algorithm determines how the uncovered node set would change if the candidate was selected as a probe station. This uncovered node set will consist of

- i) nodes that are not neighbors of selected probe stations, and
- ii) nodes that do not have k unique paths from the selected probe stations.

Of all the candidate probe station nodes, the node that produces the smallest set of uncovered nodes is selected as the next probe station node. The algorithm iteratively adds a new node to the probe station set till the desired capacity of diagnosing k faults is achieved. The algorithm terminates when no uncovered nodes are present or the probe station set size reaches the maximum limit.

Algorithm: Probeset Reduction

input: MAXFAULTS

output: Probe station set

1. Define: N =no.of nodes in the network
 UN = Uncovered nodes set
 PS = probe station set
 V = set of nodes in the network
2. initialize $PS \leftarrow \text{NULL}$, $UN \leftarrow V$
3. $u = \text{SelectFirstProbeStation}()$
4. add node u to PS and remove u from UN .
5. remove neighbors of u from UN
6. foreach node $c \notin PS$, compute uncovered node set $S(c)$ such that there are k independent paths from these probe stations to remaining uncovered and non-neighbor nodes
7. select node c with smallest $|S(c)|$ as next probe station
8. Add c to PS and set $UN \leftarrow S(c)$
9. repeat step 6 thru 8 until $|UN| = 0$

Procedure SelectFirstProbeStation()

1. Define: NN = Neighbor nodes
2. Identify the node x with highest degree
3. Identify neighbors NN of node x having minimum degree
4. For each node $n \notin NN$, compute set (S) of neighbor nodes of n having node number less than x

5. Select node n having minimum $|S|$ as the first probe station.

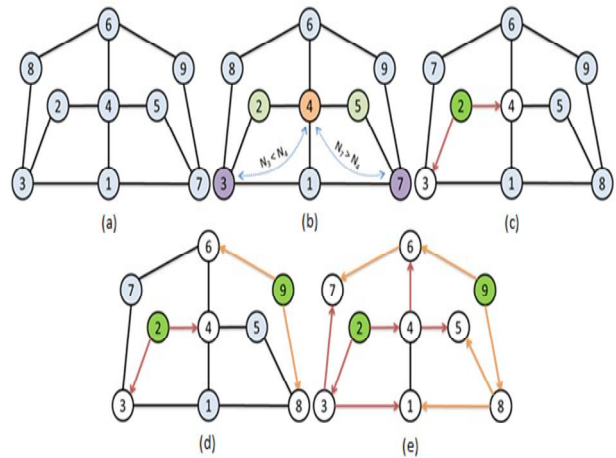


Figure 3 : Probe station selection

Figure 3 presents an example of how the probe station selection algorithm selects probe stations to detect any two node failures in the network. Figure 3(a) shows a network topology with nine nodes considering all nodes as uncovered nodes. Figure 3(b) shows nodes 2 & 5 (minimum degree nodes) as neighbors of node 4 which has largest degree in the network. Both 2 and 5 have one neighbor node, but only node 2 has got neighbor node 2 which is less than 4. Hence node 2 will be selected as first probe station removing neighboring nodes 3 and 4 from the uncovered node set, as shown in Figure 3(c). Figure 3(d) shows node 9 as the next selected probe station, which removes neighboring nodes 6 and 8 from the uncovered node set. Nodes 1, 5 and 7 are not neighbors of any probe station, but they have two independent probe paths from probe station 2 and 9 as shown in the Figure 3(e). Thus nodes 1, 5 and 7 are also removed from the uncovered node set. Thus the probe station placement at nodes 2 and 9 can detect any two node failures in the network.

III. PROBE SET SELECTION

In this section, we propose an algorithm for selecting minimum set of probe set for fault localization. As discussed earlier there are different criteria to be taken into consideration for fault detection and localization. Before getting into those details, it's important to understand the concept of dependency matrix.

A. Notation

We have a set of nodes (components) $N = \{N_1, \dots, N_n\}$, each of which can be either "up", functioning correctly, or "down", not functioning correctly. In a distributed system, the nodes may be physical entities such as routers, servers, and links, or logical entities such as software components, database tables, etc. The state of the system is

denoted by a vector $X = (X_1, \dots, X_n)$ of Boolean variables, where X_i represents the state of node (component) N_i . Lower-case letters denote the values of the corresponding variables, e.g. $x = (x_1, \dots, x_n)$ denotes a particular assignment of node values. In general, there are 2^n different system states; however, in practice it can often be assumed that only k faults can occur simultaneously - indeed the case $k = 1$ is often sufficient.

A probe is a method of obtaining information about the system components. The set of components tested by a probe p (i.e. the components p depends on) is denoted $N(p) \subseteq \{N_1, \dots, N_n\}$. A probe either succeeds or fails: if it succeeds, then every component it tests is up; it fails if any of the components it tests are down.

A **dependency matrix** captures the relationships between system states and probes.

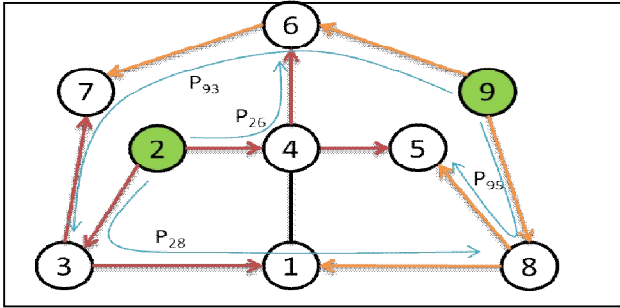


Figure 4: Sample network

	N_1	N_2	N_3	N_4	N_5	N_6	N_7	N_8	N_9
p_{21}	1	1	1	0	0	0	0	0	0
p_{13}	0	1	1	0	0	0	0	0	0
p_{24}	0	1	0	1	0	0	0	0	0
p_{25}	0	1	0	1	1	0	0	0	0
p_{26}	0	1	0	1	0	1	0	0	0
p_{27}	0	1	1	0	0	0	1	0	0
p_{28}	1	1	1	0	0	0	0	1	0
p_{29}	0	1	0	1	0	1	0	0	1
p_{91}	1	0	0	0	0	0	0	1	1
p_{92}	0	1	0	1	0	1	0	0	1
p_{93}	0	0	1	0	0	1	1	0	1
p_{94}	0	0	0	1	0	1	0	0	1
p_{95}	0	0	0	0	1	0	0	1	1
p_{96}	0	0	0	0	0	1	0	0	1
p_{97}	0	0	0	0	0	1	1	0	1
p_{98}	0	0	0	0	0	0	0	1	1

Table 1: Dependency matrix for sample network

N_2 and N_9 are probe stations.

Given any set of nodes $N = \{N_1, N_2, \dots, N_n\}$ and probes $P = \{p_1, p_2, \dots, p_r\}$, the dependency matrix $D_{P,N}$ is given by:

$D_{P,N}(i; j) = 1$ if $N_j \cap N(p_i) \neq \emptyset = 0$ otherwise:

$D_{P,N}$ is an r -by- n matrix, where each row represents a probe and each column represents a node.

Table-1 shows an example dependency matrix for a simple network - the network and two of the probes are shown in Figure-1.

B. Fault Detection

The task of fault detection is to find the smallest subset P' of the probe set P such that, if any (non-empty) $f \in F$ occurs, there is some probe $p \in P'$ that

is affected by f . This can be formulated in terms of the dependency matrix:

Detection: Given $D_{P,F}$, find P^* that minimizes $|P'|$, where $P' \subseteq P$ such that there is at least one 1 in every column of $D_{P',F}$.

By monitoring the probes we will know, as soon as a probe fails to return, that there is a problem somewhere in the network, but we may not know exactly what the problem is.

C. Fault Localization

Fault localization requires finding the smallest probe set such that every fault has a unique probe signal, since in that case exactly which fault has occurred can be determined from the probe results. Since the probe signal of fault f_j is the column c_j of $D_{P,F}$, each fault has a unique probe signal if and only if each column in $D_{P,F}$ is **unique**; i.e. differs from every other column. Since two columns c_i, c_j differ if and only if there is some entry where one of them has the value 1 while the other has the value 0 (i.e. there is some probe which is affected by one of the faults but not the other), fault localization can be expressed using the number of non-zero elements, denoted by n_{ij} , in $c_i \oplus c_j$, where \oplus denotes exclusive-OR:

Localization: Given $D_{P,F}$, find P^* which minimizes $|P'|$, where $P' \subseteq P$ satisfies $\forall f_i, f_j \in F, n_{ij} \geq 1$.

Referring to same network in Figure 4, fault detection requires finding the smallest number of rows such that every column has at least one 1. In this example, this means the smallest set of probes which pass through every node, so that, no matter which node fails, there is a probe that will detect it. The following set of 3 probes suffices:

	N_1	N_2	N_3	N_4	N_5	N_6	N_7	N_8	N_9
P_{25}	0	1	0	1	1	0	0	0	0
P_{28}	1	1	1	0	0	0	0	1	0
P_{93}	0	0	1	0	0	1	1	0	1

Since no single probe passes through all the nodes, this is clearly a smallest subset for fault detection. However this set fails for the task of fault localization because, for example, failures in nodes N_4 and N_5 cannot be distinguished from each other and failures in nodes N_6 and N_7 cannot be distinguished from each other - they generate the same signal, since their columns are identical. However the following set of 4 probes is a minimal set for fault localization:

	N_1	N_2	N_3	N_4	N_5	N_6	N_7	N_8	N_9
P_{28}	1	1	1	0	0	0	0	1	0
P_{26}	0	1	0	1	0	1	0	0	0
P_{93}	0	0	1	0	0	1	1	0	1
P_{95}	0	0	0	0	1	0	0	1	1

Since all 9 columns are unique, the results of these 4 probes allow us to determine exactly which node has failed. For example, if p_{26} and p_{93} both fail, then we infer that node N_6 has failed.

D. Probe set selection algorithm

After the deployment of probe stations, appropriate probes need to be selected such that the required diagnosis capability can be obtained. As probes involve sending additional network traffic, it is important to minimize the number of probes to perform fault diagnosis. We use a form greedy search algorithm where each probe is evaluated in terms of their localization quality. Localization quality of a set of probes is defined as amount of information provided by a probe set for faults in a network.

The localization decomposition $S_{P,F}$ is a collection of groups $\{G_1, \dots, G_k\}$, where each group G_i contains the faults $f_i \in F$, that cannot be distinguished from one another by P . Then localization quality of P is defined as the conditional entropy $H(F/G)$, where F is random variable denoting fault and G the random variable denoting which group of $S_{P,F}$ contains the fault.

$$Q(P, F) = H(F/G)$$

If the faults are independent and equally likely, then

$$Q(P, F) = \sum_{i=1}^k \frac{n_i}{n} \log n_i$$

Where n_i is the number of faults in group G_i of $S_{P,F}$ and $n=|F|$.

Algorithm: Greedy search

input: Dependency matrix $D_{P,F}$, with rows p_1, p_2, \dots, p_r

output: Probe set P' (possibly non-minimal size)

$P' = \phi =$ empty set

While $S_{P',F} \neq S_{P,F}$

$p^* = \operatorname{argmin}_{p \in P \setminus P'} Q(P' \cup \{p\}, F)$

$P' \leftarrow P' \cup \{p^*\}$

Output P'

As an example, consider the dependency matrix shown in Table 1 corresponding sample network shown in Figure 4.

Greedy search algorithm will select a probe with minimum QPF and calculate decomposing induced by this probe.

Following table shows minimum probe set, its corresponding QPF and decomposition induced by each probe - f_n denotes failure in Node N_n .

Probe	QPF	Decomposition
P_{28}	2.17	$\{f_1, f_2, f_3, f_8\}, \{f_4, f_5, f_6, f_7, f_9\}$
P_{26}	1.27	$\{f_1, f_3, f_8\}, \{f_2\}, \{f_4, f_6\}, \{f_5, f_7, f_9\}$
P_{93}	0.44	$\{f_1, f_8\}, \{f_2\}, \{f_3\}, \{f_4\}, \{f_5\}, \{f_6\}, \{f_7, f_9\}$
P_{95}	0	$\{f_1\}, \{f_2\}, \{f_3\}, \{f_4\}, \{f_5\}, \{f_6\}, \{f_7\}, \{f_8\}, \{f_9\}$

Table 2: QPF value and decomposition induced by each probe

IV. EXPERIMENTAL EVALUATION

In this section, we present the experimental evaluation of the proposed algorithm. We apply algorithms to select minimal set of probe station followed by minimal set of probe set for fault localization.

A. Experiment setup

We are using OMNET++ as simulation tool to simulate network, test our algorithms and capture results. We produce different scale networks using OMNET++ random network generator. Given a network topology the simulation proceeds with

- Selecting probe stations using algorithm explained in section II-B
- It next generates dependency matrix for the network
- Using algorithm explained in section III-D it selects probe set

B. Simulation Results

We have studied results of our algorithm with different size of networks and compared it with results obtained from random probe selection algorithm. We conducted experiments with network size varying between 10 and 50 nodes. Figure 6 shows that the proposed algorithm, ProbesetReduction, provides better results as compared to random algorithm as network size increases.

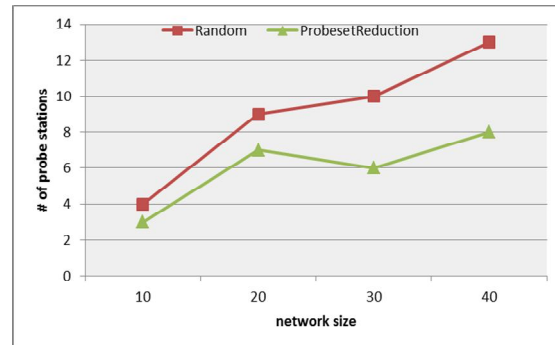


Figure 5: Number of probe stations with different network sizes

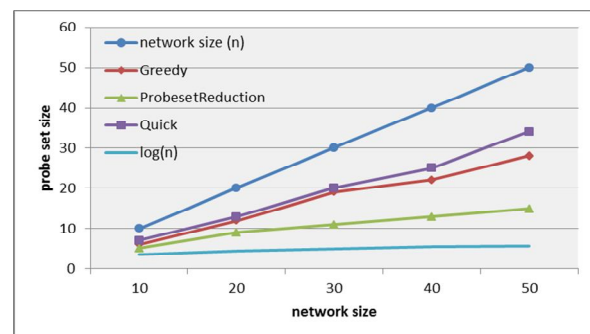


Figure 6: Number of probe stations and probes

The faults we are interested in diagnosing are any single node being down or no failure anywhere in the network. We assume that each node has the same prior probability of failure, and that there is no noise

in the probe results. Note that in this case n probes are sufficient, because one can always use just one probe-station and probe every single node. Thus we expect that the minimal number of probes should lie between $\log n$ and n . To test the algorithm on networks of different sizes, we ran the Greedy, Quick search and ProbesetReduction algorithm on networks with varying sizes having the average node degree 3. The comparison of these algorithms is shown in Figures 7. Figure 7 shows that the probe sets computed by the ProbesetReduction are smaller than those computed by the Greedy algorithm and Quick algorithm. The results of experiments with integrated probe station and probe set selection algorithm reveals that probe station selection plays a pivotal role in identifying minimal set of probes. A better placing of probe stations producing fewer probes close to $\log n$ than to n .

V. CONCLUSION

In this paper, we address the problem of diagnosis in distributed systems using test transactions, or probes. Probes offer an approach to diagnosis that is more active than traditional “passive” techniques like event correlation. Our main objective is developing a cost-efficient probing strategy; we want a small probe set which at the same time provides wide coverage for locating or detecting problems anywhere in the network.

We first presented algorithms to select suitable locations to deploy the probe stations which will generate long probes and will return minimum probe set for fault detection and localization. We presented the algorithm assuming the availability of complete and accurate information about the underlying network. Analysis and experiments show that better placing of probe stations can greatly reduce the probe set for fault localization.

Directions for future work include developing algorithm for probe station selection based on nodes and links covered and can produce probe set for detecting node and link failure.

REFERENCES

- [1] M. Brodie, I. Rish, S. Ma, Optimizing probe selection for fault localization, In the 12th International Workshop on Distributed Systems Operations Management, 2001.
- [2] M. Brodie, I. Rish, S. Ma, G. Grabarnik, N. Odintsova, Active probing, Technical Report IBM, 2002.
- [3] M. Natu, A. S. Sethi, Active probing approach for fault localization in computer network”, In E2EMON’06, Vancouver, Canada, 2006.
- [4] M. Natu, A. S. Sethi, Efficient probing techniques for fault diagnosis, Second International Conference on Internet Monitoring and Protection, IEEE, 2007.
- [5] M. Brodie, I. Rish, S. Ma, N. Odintsova, A. Beygelzimer, G. Grabarnik, K. Hernandez, Adaptive diagnosis in distributed systems, Technical Report IBM, 2002.
- [6] S. A. Yemini, S. Kliger, E. Mozes, Y. Yemini, D. Ohsie, High speed and robust event correlation, IEEE communications Magazine 34 (5) (1996) 82-90.
- [7] R. Gardner, D. Harle, Alarm correlation and network fault resolution using Kohonen Self-Organizing map, Globecom 97 proceedings, pp. 1398-1402, 1997.
- [8] A. T. Bouloutas, G. W. Hart, M. Schwartz, Fault identification using a FSM model with unreliable partially observed data sequences, IEEE Transactions on Communications, 41(7):pp. 1074-1083, 1993.
- [9] C. Wang, M. Schwartz, Identification of faulty links in dynamic-routed networks, IEEE Journal on Selected Areas in Communications, 11 (3) 1449-1460, 1993.
- [10] M. Stemm, R. Kaltz, and S. Seshan. A network measurement architecture for adaptive applications. In IEEE INFOCOM, Mar 2000.
- [11] M. Natu and A. S. Sethi. Probe station placement for robust monitoring of networks. Submitted to Journal of Network and Systems Management.

