SOCIETY FOR SCIENCE AND EDUCATION UNITED KINGDOM
TNC Transactions on Networks and Communications
VOLUME 3, ISSUE 4
ISSN: 2054 - 7420

Survey of Probe Set and Probe Station Selection Algorithms for Fault Detection and Localization in Computer Networks

[1]Balaji Patil and [2]Vinay Kumar Pathak
[1]UTU, Dehradun. Maharashtra Institute of Technology, Pune, India;
[2]Computer Sci. & Engg. Dept. HBTI, Kanpur, India;
balaji.patil@mitpune.edu.in; vinay@vpathak.in

ABSTRACT

Probing has evolved as a promising approach for fault diagnosis in a network management. It is based on the principle of actively sending out probes in the network to infer the health of network components. Probes are test transactions whose success or failure depends on the health of the probed network components. 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. Small probe sets is desirable in order to minimize the costs imposed by probing, such as additional network load and data management requirements.

In this paper we have presented an overview of various probe set selection algorithms for network fault detection and localization. We have evaluated these algorithms on a sample network for better understanding.

Keywords- Active probing, fault detection, fault localization, probe set selection algorithms.

1 Introduction

Today, high-speed communication network play an increasingly important role and lead to a demand for a higher quality network management level. It becomes important to provide an efficient solution to monitor the network for availability and performance.

Some key challenges on fault management are as follows [13]: (1) Incomplete and inaccurate information of real network (2) Real-time fault detection and localization to achieve system's automatic repair (3) Generation of minimal traffic by management station to reduce pressure on the network (4) Existence of multiple faults in the network.

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 task usually include two phases: fault detection and fault diagnosis. Fault detection is to discover if there is at least one faulty component in the system. Fault diagnosis is aim to find all the faulty components by sending additional probes to the region of interests. Fault Detection is carried out by two ways i.e. Active Monitoring and Passive Monitoring.

DOI: 10.14738/tnc.34.1023
Publication Date: 30th August 2015
URL: http://dx.doi.org/10.14738/tnc.34.1023

1.1 Active Monitoring

Active monitoring deploys probing methods to gather health status and performance statistics of network entities in the managed system [1].

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 placement and probe set selection.

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. 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.

1.2 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 sets to detect those failures. Also, sending this large number of probes at a periodic interval generates large amount of management traffic. The 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 quickly as they occur, this delay might not be acceptable. Moreover, this delay will potentially delay in next step of fault localization.

1.3 Adaptive Probing

Active probing 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 to be obtained, which probes to send next decision is important, until finally the problem is completely determined. It greatly reduces management traffic and provides more accurate and timely diagnosis.

The outline of the paper is as follows. Section II provides the basic framework and notation. Section III covers different probe set selection algorithms for fault detection and localization. Section IV covers different probe station selection algorithms for fault detection and localization. Section V gives tabular comparison of algorithms. Section VI refers to proposed work and section VII covers the conclusion.

2 Notations

This section focuses on understanding basic concepts involved in fault management.

2.1 Faults

Event, defined as an exceptional condition occurring in the operation of hardware or software of a managed network, is a central concept pertaining to fault diagnosis. Faults (also referred to as

problems or root causes) constitute a class of network events that can cause other events but are not themselves caused by other events. In a network fault may be a particular node or link (or both) is down.

Assuming there are finite set O of objects in a network. The fault can be any subset of $f \subseteq O$. A single element in f represents a network failure and similarly an empty set represents no failures in the network.

2.2 Dependency Matrix

A dependency matrix captures the relationships between system states and probes [10].

Consider a sample network as shown in Figure 1 where N_2 and N_9 are probe stations.

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

$$D_{P,N}(i; j) = 1 \text{ if } N_j \cap N(p_i) \neq \phi$$

$$= 0 \text{ otherwise:}$$

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

Table 1 is a dependency matrix for a sample network shown in Figure 1. The sample network has four available probes as shown in Figure 1.

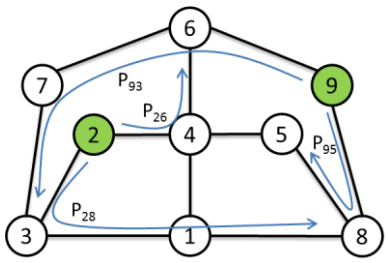

Figure 1: Sample network showing few available probes

Table 1: Dependency matrix for Sample network shown in Figure 1.

Nodes→ Probes↓	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_{23}	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

2.3 Fault Detection

Fault detection is to discover if there is at least one faulty component in the system. Which means the deployment of measurements must cover all nodes in the target network. Once any network anomalies are detected then the fault diagnosis process gets started. Fault diagnosis aims to find all the faulty components by sending additional probes to the region of failure.

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 $D_{P',F}$ [10].

By monitoring the probes we will be able to know about fault in the network as soon as the probe gets failed; but fault cannot be exactly localized with this much information.

2.4 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 [10]. 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 $\forall f_i, f_j \in F, n_{ij} \geq 1$ [10].

Referring to sample network in Figure 1 and Table 1, fault detection requires finding the smallest number of rows such that every column has at least one '1'. In above sample network 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:

Nodes→ / Probes↓	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. Minimal set is the least number of probes which uniquely localize the fault in the network.

Nodes→ / Probes↓	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.

3 Survey of Fault Detection and Localization Algorithms

In this section, we discuss various algorithms for fault detection and localization using preplanned and active probing. We have considered the sample network in Figure 1 for elaboration of different algorithms of probe set reduction.

3.1 Greedy Search Algorithm

3.1.1 Approach

There are various techniques researchers have suggested that are based on greedy algorithm [10]. The simplest one tries to identify a probe that covers maximum number of nodes.

3.1.2 Implementation

Consider sample network in Figure 1, node N2 and N9 are probe stations. Below are the probes that originate from these two probe stations.

Table 2: Greedy Search - nodes covered through available probes.

Available Probes	Nodes it Cover	Available Probes	Nodes it Cover
P_{21}	$\{N_1,N_2,N_3\}$	P_{91}	$\{N_1,N_8,N_9\}$
P_{23}	$\{N_2,N_3\}$	P_{92}	$\{N_3,N_4,N_9\}$
P_{24}	$\{N_2,N_4\}$	P_{93}	$\{N_3,N_6,N_7,N_9\}$
P_{25}	$\{N_2,N_4,N_5\}$	P_{94}	$\{N_4,N_6,N_9\}$
P_{26}	$\{N_2,N_4,N_6\}$	P_{95}	$\{N_5,N_8,N_9\}$
P_{27}	$\{N_2,N_3,N_7\}$	P_{96}	$\{N_6,N_9\}$
P_{28}	$\{N_1,N_2,N_3,N_8\}$	P_{97}	$\{N_6,N_7,N_9\}$
P_{29}	$\{N_2,N_4,N_6,N_9\}$	P_{98}	$\{N_8,N_9\}$

Above information can also be reframed such that each node is represented by a set of probes passing through it

Table 3: Greedy Search - probes covering each node in the network

Node	Probes Covering this node	Node	Probes Covering this node
N_1	$\{P_{21},P_{28},P_{91}\}$	N_6	$\{P_{26},P_{29},P_{92},P_{93},P_{94},P_{96},P_{97}\}$
N_3	$\{P_{23},P_{27},P_{28},P_{93}\}$	N_7	$\{P_{27},P_{93},P_{97}\}$
N_4	$\{P_{24},P_{25},P_{26},P_{29},P_{92},P_{94}\}$	N_8	$\{P_{28},P_{91},P_{95},P_{98}\}$
N_5	$\{P_{25},P_{95}\}$		

Since N_2 and N_9 are probe stations, we can define Non-probed nodes as

$$NPN = \{N_1,N_3,N_4,N_5,N_6,N_7,N_8\}$$

As per greedy search algorithm, first step is to identify a node with minimum cardinality, node (N_5) which is covered by least number of probes (P_{25}, P_{95}). Out of these probes, select a probe that has got maximum cardinality (P_{25}) – one which can probe maximum number of other nodes. After each step the algorithm updates Node list and Probe set.

Table 4: Greedy Search algorithm - iteration wise progress on sample network

Step	Minimum Cardinality Node	Probes covering this node	Selected probe with max cardinality	NPN
I	N_5	$\{P_{25},P_{95}\}$	P_{25}	$\{N_1,N_3,N_6,N_7,N_8\}$
II	N_1	$\{P_{21},P_{28},P_{91}\}$	P_{28}	$\{N_6,N_7\}$
III	N_7	P_{93}	P_{93}	$\{\}$

Thus probe set $\{P_{25}, P_{28}, P_{93}\}$ can detect any failure in the sample network shown in Figure 1.

3.2 Additive Search Algorithm

3.2.1 Approach

In this form of greedy search each probe is evaluated in terms of their localization quality [5]. Localization quality of a set of probes is defined as amount of information provided by a probe set for faults in a network.

3.2.2 Implementation

The localization decomposition $S_{P,F}$ is a collection of groups $\{G_1,...,G_k\}$, 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

$$Q(P,F) = \sum_{i=1}^{k} \frac{ni}{n} \; logni$$

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

Input: Dependency matrix $D_{P,F}$, with rows $p_1,p_2,...,p_r$

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

Algorithm:

$P' = \phi =$ empty set

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

$p^* = argmin_{p \in p \setminus p'} Q(p' \cup \{p\}, F)$

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

Output P'

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

Every iteration of this algorithm will select a probe with minimum *Q(P,F)* and calculate decomposing induced by this probe. This is continued until selected probe finally results into decomposition into singleton set.

Following Table 5 shows minimum probe set, its corresponding *Q(P,F)* and decomposition induced by each probe f_n denotes failure in Node N_n.

Table 5: *Q(P,F)* value and decomposition induced by each probe

Probe	*Q(P,F)*	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	$\{f_1\},\{f_2\},\{f_3\},\{f_4\},\{f_5\},\{f_6\},\{f_7\},\{f_8\},\{f_9\}$

Thus probe set $\{P_{28}, P_{26}, P_{93}, P_{95}\}$ can localize any fault in the sample network shown in Figure 1.

3.3 Subtractive Search Algorithm

3.3.1 Approach

Subtractive search begins with the assumption that all probes are available for fault detection [5]. Gradually probes are dropped from this available set such that the localization quality is always maintained - same as obtained from the original set.

3.3.2 Implementation

Consider sample network as in Figure 1. As per Subtractive Search algorithm, initially all probes are considered to be Selected Probes (SP).

Probes are removed from SP until there is at least one probe available to discover any particular node i.e. $N_i \neq \{\Phi\}$

The probes are being considered for removal in every iteration still the removal of that probe doesn't affect the localization quality; it will get removed from the SP.

Table 6: Subtractive search algorithm - iteration wise progress on sample network

Probe	Localization quality		Probe	Localization quality	
P_{21}	*Node*	*Probes it Cover*	***P_{91}***	***Node***	***Probes it Cover***
	N_1	$\{P_{28},P_{91}\}$		***N_1***	***$\{P_{91}\}$***
	N_2	$\{P_{23},P_{24},P_{25},P_{26},P_{27},P_{28},P_{29}\}$		***N_2***	***$\{P_{29}\}$***
	N_3	$\{P_{23},P_{27},P_{28},,P_{92},P_{93}\}$		N_3	$\{P_{93}\}$
	N_4	$\{P_{24},P_{25},P_{26},P_{29},P_{92},P_{94}\}$		N_4	$\{P_{29},P_{94}\}$
	N_5	$\{P_{25},P_{95}\}$		N_5	$\{P_{95}\}$
	N_6	$\{P_{26},P_{29},P_{92},P_{93},P_{94},P_{96},P_{97}\}$		N_6	$\{P_{29},P_{93},P_{94},P_{96},P_{97}\}$
	N_7	$\{P_{27},P_{93},P_{97}\}$		N_7	$\{P_{93},P_{97}\}$
	N_8	$\{P_{28},P_{91},P_{95},P_{98}\}$		N_8	$\{P_{91},P_{95},P_{98}\}$
P_{23}	*Node*	*Probes it Cover*	P_{93}	*Node*	*Probes it Cover*
	N_1	$\{P_{28},P_{91}\}$		N_1	$\{P_{91}\}$
	N_2	$\{P_{24},P_{25},P_{26},P_{27},P_{28},P_{29}\}$		N_2	$\{P_{29}\}$
	N_3	$\{P_{27},P_{28},,P_{92},P_{93}\}$		N_3	$\{P_{93}\}$
	N_4	$\{P_{24},P_{25},P_{26},P_{29},P_{92},P_{94}\}$		N_4	$\{P_{29},P_{94}\}$
	N_5	$\{P_{25},P_{95}\}$		N_5	$\{P_{95}\}$
	N_6	$\{P_{26},P_{29},P_{92},P_{93},P_{94},P_{96},P_{97}\}$		N_6	$\{P_{29},P_{93},P_{94},P_{96},P_{97}\}$
	N_7	$\{P_{27},P_{93},P_{97}\}$		N_7	$\{P_{93},P_{97}\}$
	N_8	$\{P_{28},P_{91},P_{95},P_{98}\}$		N_8	$\{P_{91},P_{95},P_{98}\}$
P_{24}	*Node*	*Probes it Cover*	P_{94}	*Node*	*Probes it Cover*
	N_1	$\{P_{28},P_{91}\}$		N_1	$\{P_{91}\}$
	N_2	$\{P_{25},P_{26},P_{27},P_{28},P_{29}\}$		N_2	$\{P_{29}\}$
	N_3	$\{P_{27},P_{28},,P_{92},P_{93}\}$		N_3	$\{P_{93}\}$
	N_4	$\{P_{25},P_{26},P_{29},P_{92},P_{94}\}$		N_4	$\{P_{29}\}$
	N_5	$\{P_{25},P_{95}\}$		N_5	$\{P_{95}\}$
	N_6	$\{P_{26},P_{29},P_{92},P_{93},P_{94},P_{96},P_{97}\}$		N_6	$\{P_{29},P_{93},P_{96},P_{97}\}$
	N_7	$\{P_{27},P_{93},P_{97}\}$		N_7	$\{P_{93},P_{97}\}$
	N_8	$\{P_{28},P_{91},P_{95},P_{98}\}$		N_8	$\{P_{91},P_{95},P_{98}\}$
P_{25}	*Node*	*Probes it Cover*	***P_{95}***	***Node***	***Probes it Cover***
	N_1	$\{P_{28},P_{91}\}$		***N_1***	***$\{P_{91}\}$***
	N_2	$\{P_{26},P_{27},P_{28},P_{29}\}$		***N_2***	***$\{P_{29}\}$***
	N_3	$\{P_{27},P_{28},P_{93}\}$		***N_3***	***$\{P_{93}\}$***
	N_4	$\{P_{26},P_{29},P_{94}\}$		***N_4***	***$\{P_{29}\}$***
	N_5	$\{P_{95}\}$		***N_5***	***$\{P_{95}\}$***

	N_6	$\{P_{26},P_{29},P_{93},P_{94},P_{96},P_{97}\}$		$\mathbf{N_6}$	$\mathbf{\{P_{29},P_{93}, P_{96},P_{97}\}}$
	N_7	$\{P_{27},P_{93},P_{97}\}$		$\mathbf{N_7}$	$\mathbf{\{P_{93},P_{97}\}}$
	N_8	$\{P_{28},P_{91},P_{95},P_{98}\}$		$\mathbf{N_8}$	$\mathbf{\{ P_{91},P_{95},P_{98}\}}$
P_{26}	*Node*	*Probes it Cover*	P_{96}	*Node*	*Probes it Cover*
	N_1	$\{P_{28},P_{91}\}$		N_1	$\{P_{91}\}$
	N_2	$\{P_{27},P_{28},P_{29}\}$			$\{P_{29}\}$
	N_3	$\{P_{27},P_{28},P_{93}\}$		N_3	$\{ P_{93}\}$
	N_4	$\{P_{29},P_{94}\}$		N_4	$\{P_{29}\}$
	N_5	$\{P_{95} \}$		N_5	$\{P_{95} \}$
	N_6	$\{P_{29},P_{93},P_{94},P_{96},P_{97}\}$		N_6	$\{P_{29},P_{93}, P_{97}\}$
	N_7	$\{P_{27},P_{93},P_{97}\}$		N_7	$\{P_{93},P_{97}\}$
	N_8	$\{P_{28},P_{91},P_{95},P_{98}\}$		N_8	$\{ P_{91},P_{95},P_{98}\}$
P_{27}	*Node*	*Probes it Cover*	P_{97}	*Node*	*Probes it Cover*
	N_1	$\{P_{28},P_{91}\}$		N_1	$\{P_{91}\}$
	N_2	$\{ P_{28},P_{29}\}$		N_2	$\{P_{29}\}$
	N_3	$\{P_{28},P_{93}\}$		N_3	$\{ P_{93}\}$
	N_4	$\{P_{29},P_{94}\}$		N_4	$\{P_{29}\}$
	N_5	$\{P_{95} \}$		N_5	$\{P_{95} \}$
	N_6	$\{P_{29},P_{93},P_{94},P_{96},P_{97}\}$		N_6	$\{P_{29},P_{93} \}$
	N_7	$\{P_{93},P_{97}\}$		N_7	$\{P_{93} \}$
	N_8	$\{P_{28},P_{91},P_{95},P_{98}\}$		N_8	$\{P_{91},P_{95},P_{98}\}$
P_{28}	*Node*	*Probes it Cover*	P_{98}	*Node*	*Probes it Cover*
	N_1	$\{P_{91}\}$		N_1	$\{P_{91}\}$
	N_2	$\{P_{29}\}$		N_2	$\{P_{29}\}$
	N_3	$\{ P_{93}\}$		N_3	$\{ P_{93}\}$
	N_4	$\{P_{29},P_{94}\}$		N_4	$\{P_{29}\}$
	N_5	$\{P_{95} \}$		N_5	$\{P_{95} \}$
	N_6	$\{P_{29},P_{93},P_{94},P_{96},P_{97}\}$		N_6	$\{P_{29},P_{93} \}$
	N_7	$\{P_{93},P_{97}\}$		N_7	$\{P_{93} \}$
	N_8	$\{ P_{91},P_{95},P_{98}\}$		N_8	$\{ P_{91},P_{95} \}$
$\mathbf{P_{29}}$	***Node***	***Probes it Cover***			
	$\mathbf{N_1}$	$\mathbf{\{P_{91}\}}$			
	$\mathbf{N_2}$	$\mathbf{\{P_{29}\}}$			
	$\mathbf{N_3}$	$\mathbf{\{ P_{93}\}}$			
	$\mathbf{N_4}$	$\mathbf{\{P_{29},P_{94}\}}$			
	$\mathbf{N_5}$	$\mathbf{\{P_{95} \}}$			
	$\mathbf{N_6}$	$\mathbf{\{P_{29},P_{93},P_{94},P_{96},P_{97}\}}$			
	$\mathbf{N_7}$	$\mathbf{\{P_{93},P_{97}\}}$			
	$\mathbf{N_8}$	$\mathbf{\{ P_{91},P_{95},P_{98}\}}$			

Finally, four probes that are covering all nodes in the network are selected probes

$$SP = \{P_{29}, P_{91}, P_{93}, P_{95}\}$$

3.4 Constraint Satisfaction Problem (CSP) Algorithm

3.4.1 Approach

In constraint satisfaction problem approach authors have defined some constraints to find out the healthy and suspected set [13]. These constraints map the nodes to healthy set and suspected set.

3.4.2 Implementation

Consider the dependency matrix D, obtained for the sample network in Figure 1 is represented by Table 1.

Assuming a single node failure may occur in the managed network, let Rpi refer to response of the probe *pi* where *i = 1 … K*. *K* is the number of probes in the fault detection probe set. Rpi may take value only from the set *{0, 1}*. If the probe *pi* fails, then Rpi = 0; otherwise Rpi = *1*.

If a single probe fails (Rpi = 0) and other probes succeed (Rpi =*1*), then the initial suspect and healthy sets are constructed as follows-

$$F = \{n \mid n \in \big(pi - (pi \cap pj)\big)\}$$

$$G = \{n \mid n \in \big((N - pi) \cup (pi \cap pj)\big)\}$$

Where *i, j =1...k, i ≠ j.*

Using greedy search scheme, first three probes are selected that can completely cover all nodes in the managed network. Should one or both of these probes report the occurrence of a failure, the fault localization function will be invoked promptly. The information carried by the detection probes will be fully exploited as follows:

Nodes→ / Probes↓	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

The following set of 3 probes suffices available probes and nodes it covers:

Available Probes	Nodes it Cover
P_{25}	$\{N_2,N_4,N_5\}$
P_{28}	$\{N_1,N_2,N_3,N_8\}$
P_{93}	$\{N_3,N_6,N_7,N9\}$

- If P_{25} has failed and P_{28} and P_{93} has succeeded

Suspect set F = $P_{25} - (P_{25} \cap (P_{28}+P_{93}))$ = $\{N_4, N_5\}$; Healthy set G = P_{45} = $\{N_1, N_2, N_3, N_6, N_7, N_8\}$

- If P_{25} and P_{28} had failed and P_{93} has succeeded

Suspect set F = $P_{25} \cap P_{28} - P_{93}$ = $\{N_2\}$; Healthy set G = $\{N_1$, N3, N4, N5, N6, N7, N8, N9}

3.5 Binary Search Algorithm

3.5.1 Approach

As name suggests, binary search algorithm uses binary search to localize faulty nodes in the network [4]. If any probe used for fault detection fails, then all nodes in the path of failed probe are treated as suspected nodes. In order to identify the failed node from these suspected nodes, additional probes are sent recursively until the search narrows down to failed node.

3.5.2 Implementation

Assuming optimal probe set $\{P_{25}, P_{28}, P_{93}\}$ used for fault detection is derived using greedy algorithm mentioned in section 3.1 for the sample network shown in Figure 1 and its dependency matrix in Table 1.

Say P_{28} has failed making Suspected Node set

SN= {N_3, N_1, N_8) - excluding probe station N_2

Starting from probe station, each node on the failure path are numbered sequentially,

N_2 (1) → N_3 (2) → N_1 (3) → N_8(4)

And binary search is applied to identify mid (Target) node in the path such that additional probe can be sent to this node

*TargetNode = (StartNodePosition+EndNodePosition)/*2

TargetNode = (1 + 4)/2 = round (2.5) = 3

Hence, additional probe P_{21} is selected to check if node identified by position 3 (N_1) is faulty.

If P_{21} passes, then clearly failure is with node N_8 and if P_{21} fails, search is repeated with EndNodePosition set to TargetNode.

Next *TargetNode* = (1+3)/2 = 2

Hence, additional probe P_{23} is selected to check if node identified by position 2 (N_3) is faulty.

If P_{23} passed, then clearly failure is at node N_1 else failure is at node N_3. This situation is represented by the Figure 2.

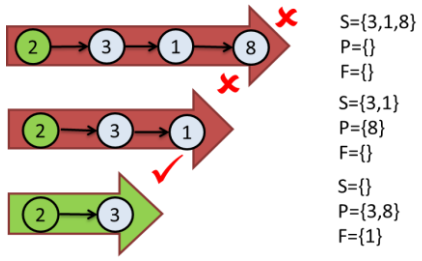

Figure 2: Probes sent in a binary search fashion on a failed probe path 2→8 to identify failed node on the path

3.6 Max Search Algorithm

3.6.1 Approach

In Max search algorithm implementation, probes are iteratively selected from available probe set such that selected probes cover maximum number of uncovered nodes till all the suspected nodes are covered [4]. This is represented in Figure 3.

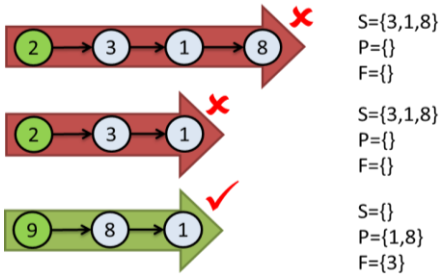

Figure 3: Probes sent in a Max search fashion on a failed probe path 2→8 to identify failed node on the path

3.6.2 Implementation

Again assuming probe P_{28} has failed resulting into Suspected Node set as

SN ={ N_3,N_1,N_8 } - excluding probe station N_2

From dependency matrix shown in Table 1, probes that cover maximum nodes from SN are P_{21} = {N_2, N_3, N_1} and P_{91} = {N_9, N_8, N_1}

These additional probes are sent to localize the failure node on the path.

Thus, if P_{21} pass and P_{91} fails, Faulty Node (F)

$F = P_{91} – (P_{91} \cap P_{21}) = \{N_8\}$ – excluding probe station N_9

If P_{21} fails and P_{91} pass, Faulty node

$F=P_{21} – (P_{21} \cap P_{91}) = \{N_3\}$

If both P_{21} and P_{91} fails,

$F=P_{21} \cap P_{91} = \{N_1\}$ is faulty

3.7 Min Search Algorithm

3.7.1 Approach

The Min search algorithm works on the concept of selecting a probe for each Suspected Node (SN) set such that the selected probe goes through minimum number of other nodes in the suspected node set [4].

3.7.2 Implementation

Again assuming probe P_{28} has failed resulting into suspected node set as

SN = {N_3, N_1, N_8} - excluding probe station N_2

From dependency matrix Table 1, probes that cover minimum nodes from SN are P_{12}= *{N_1, N2}*, P_{13}=*{N_1,N_3}* and P_{43}=*{N_3,N_4 }*

These additional probes are sent to localize the failure node on the path.

Thus, if any of the above probes fails, it can uniquely identify failure in nodes N_2 and N_3. Whereas success of all three probes will uniquely identify failure in node N_5. This situation is shown in Figure 4.

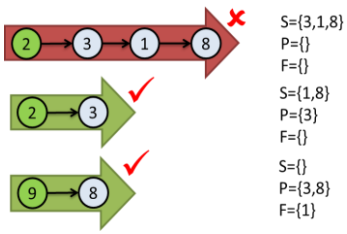

Figure 4: Probes sent in a Min search fashion on a failed probe path 2→8 to identify failed node on the path

4 Survey of Probe Station Selection Algorithms

In this section we have presented some of the recent approaches proposed for probe station selection/placement. Location and responsibilities assigned to probe stations must be decided while building an active probing solution. These decisions are based on nature of routes, nature of targeted

failures, availability of dependency information etc. [11]. Below we discuss various such factors that contribute to the overall decision making of probe station selection:

- **Nature of targeted failures**: Probe station selection depends on the nature of faults to be diagnosed viz. a node failure or an edge failure. A single probe station might not be sufficient to detect all of node and edge failures. For instance, consider the network shown in Figure 5. Consider node 1 to be a probe station; it can detect any single node failure in this network. However, it can detect failure of only those links that are used in reaching other nodes in the network, i.e., the links shown in red.

- **Maximum numbers of failures:** 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 6 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.

- **Probe station failure:** The assumption of fault tolerant probe station may not be practical and hence probe station selection problem becomes even more challenging. In case of probe station failure, probe stations are selected such that there exists k independent paths to each of the probe station's as well.

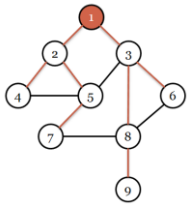

Figure 5: Link failures not being covered by Probe station 1[3]

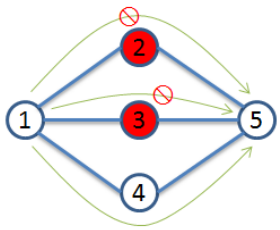

Figure 6: k Independent paths allow detection of k node failures

- **Topological constraints:** Another important criterion involved in probe station selection is the topological constraint. The node with less connectivity needs special treatment. Special topology structures like chains and rings also demand specific probe station placement requirements. One approach to simplify this problem could be to devise a solution by reducing the network into smaller sub-networks connected by such specific network structures like rings, chains, leaves, etc.

4.1 Shadow Node Reduction Algorithm (SNR)

4.1.1 Approach

In SNR a node is selected as probe station having maximum connectivity and minimum shadow node set.

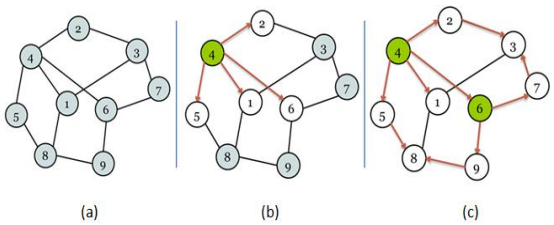

Figure 7: Probe station selection to detect any two node failures

4.1.2 Implementation

Figure 7 shows an example of how the probe station selection algorithm selects probe stations to detect any two node failures in the network. Figure 7(a) shows a network topology with nine nodes

considering all nodes as uncovered nodes. Figure 7(b) shows node 4, being the node with largest degree, as the selected probe station removing neighboring nodes 2, 6, 1, and 5 from the uncovered node set. Figure 7(c) shows node 6 as the next selected probe station, which removes neighboring nodes 9 and 7 from the uncovered node set. Nodes 3 and 8 are not neighbors of any probe station, but they have two independent probe paths from probe station 4 and 6 as shown in the Figure 7(c). Thus both nodes 3 and 8 are also removed from the uncovered node set. Thus the probe station placement at nodes 4 and 6 can detect any two node failures in the network.

4.2 Min. Hitting Set Reduction Algorithm

4.2.1 Approach

This approach uses algorithm for probe station selection using a reduction of the probe station selection problem to the Minimum Hitting Set problem. This approach reduces no. of probe stations to an optimum value [14].

4.2.2 Implementation

Referring to Figure 8: Consider a network of five nodes (1, 2, 3, 4, 5). Form all possible pairs of size two viz. {1, 2}, {1, 3} …. {4, 5} (Section - C). Now we draw a mapping between a 'Node' and a pair, Section C', if the pair can provide two independent paths to that particular node. For e.g. if node 1 gets two independent paths from pairs {1, 2} and {1, 3}, draw the mappings between node 1 and the two pairs. Identify and draw such mappings for each node in the network. Rename each pair using some naming convention, viz. rename {1, 2} as 'a'(Section –S). Now for each node in the network, write the pairs that can give two independent paths for that node e.g. for node 5, the set will be {a, d, i, j}. Each such set makes for an element in S.

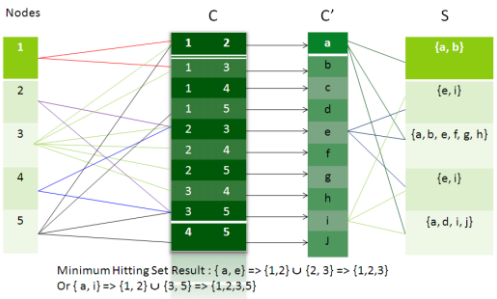

Figure 8: Minimum hitting set reduction algorithm.

Now if you look at the right half of the Figure 8, you see the familiar problem of k-sized Minimum Hitting Set Problem, which can be solved by using any approximation algorithms for the MHSP. It also addresses issues involved while selecting probe stations such as link failures and probe station failures.

5 Tabular Comparison of Algorithms

In this section we have summarized most of the recent conventional algorithms for Probe set selection and Probe station selection algorithms. These algorithms play a vital role in development of optimal solution for fault detection and localization in networks. The tabular representation in Table 7 is focused on algorithms used, limitations and future directions for the research in the field of fault localization.

Table 7: Comparative study of various probe set selection and probe station selection algorithms

SN	Algorithm	Scope	Complexity	Mechanism	Seed Idea	Advantages	Limitations
1	**Greedy Algorithm**	Fault detection	Quadratic $O(r^2)$ r- no of nodes in network	Identify a node that covers maximum number of nodes. i.e. Maximum co-cordiality	Maximize the information gain Probes set generated- (P25, P28, P93)		May not necessarily result in minimal set but as compared to Additive search gives better results.
2	**Additive Search**	Fault Localization	Quadratic O(r) r- no of nodes in network	Localization quality of a probe set is defined by the probe set provided for fault in network	-Select a probe with minimum QPF Probes set generated- (P28, P26, P93, P95)		May not necessarily result in minimal set.
3	**Exhaustive Search**	Fault detection	Exponential K^n worst case complexity K- no of distinct probes passing through a nodes	A node can be diagnosed if there is at least one probe passing through it. Probes can be added incrementally In all feasible combinations until the minimum set is reached.			Impractical for large networks.
4	**Subtractive Search**	Fault detection	Linear O(N) N- no of nodes in network	Initially all probes are available for fault detection gradually probes are dropped from the available set such that the localization quality is always maintained.	Select probes keeping localization quality always maintained. Probes set generated- (P29, P91, P93, P95)		May not necessarily result in minimal set. Effectiveness in finding minimal set depends on the order in which probes are explored
5	**Max Search**	Fault detection & localization	$O(N^2)$ N- no of nodes in network	Probes are iteratively selected from available probe set such that selected probe covers maximum number of uncovered nodes till all the suspected nodes are covered.	Selects probes that covers maximum no of suspected nodes.		- As compared to min and Binary search, it takes more time (round trips) to localize a fault.

SN	Algorithm	Scope	Complexity	Mechanism	Seed Idea	Advantages	Limitations
6	**Min Search**	Fault localization	$O(N^2)$ N- no of nodes in network	Selecting a probe for each suspected node set such that the selected probe goes through minimum no. of other nodes in the suspected node set.		Min search diagnoses all suspected nodes in parallel. Smaller probe size	-
7	**Binary Search**	Fault localization	$O(N^2)$ N- no of nodes in network	Probes are sent in binary search fashion. On a failed probe path, probe is first sent from the probe station half way on the probe path. If this probe fails, further diagnosis is done on the first half of the probe path. On the other hand, if the probe succeeds, then the later half of the probe path is diagnosed in similar fashion.	Probes are sent in binary search fashion.		No of probes and iterations required to localize failure is more as compared to min search.
8	**CSP-based model**	Fault detection & localization	-	This approach is based on Constraint Satisfaction Problem (CSP) techniques to find an appropriate and optimal collection of available probes for the fault identification.	To find an appropriate & optimal collection of available probes for the purpose of fault identification Probes set generated- (P25, P28, P93)	It performs better than Greedy and subtractive	Proper Constraints should set to get optimal solution Very few people had applied.
9	**Active probing**	Fault localization	$< O(N^2)$	In active probing selection of later probes depends on the results of earlier probes. The results of the probes are analyzed to infer what problems might be occurring identifying those and useful probes are to be send next are determined and send.	In active probing the selection of later probes depends on the results of earlier probes	Fewer probes can be used than the pre-planned probing.	Requires a more complicated technology to determine which probes to send

Probe Station Selection Algorithms

SN	Algorithm	Scope	Complexity	Mechanism	Seed Idea	Advantage	Limitations
1	**SNR Algorithm**	Probe station selection	$O(N^{MAXPSSETSIZE})$ N-no of node MAXPSSETSIZE-max no of probe stations	In the first iteration probe stations are added such that each non probe station node that is either a neighbor of a probe station or has two independent probe paths. The nodes are added in this fashion by incrementally increasing the overall diagnostic capability, till *k* faults in the network can be localized.	Select a highest node degree as first probe station.	Computationally feasible to implement.	The algorithms in this paper ensure the availability of k independent probe paths but do not aim to optimize probe traffic or the localization time.

SN	Algorithm	Scope	Complexity	Mechanism	Seed Idea	Advantages	Limitations
2	**Greedy Approximation Algorithm**	Probe station selection	O (nk) -n no of nodes -k no of failures	For Minimum Hitting Set problem Greedy approximation algorithm is used. Inapproximability results show that the greedy algorithm is essentially the best-possible polynomial time approximation algorithm for Minimum Hitting Set problem under plausible complexity assumptions.	A novel reduction of the Minimum Probe Station Selection problem to the Minimum Hitting Set problem.	Computationally feasible to implement.	The algorithms in this paper ensure the availability of k independent probe paths but do not aim to optimize probe traffic or the localization time.
3	**Exhaustive Search**	Probe station selection	Exponential	Probe station placement should be such that Probe stations are able to send enough probes to all the nodes to localize the fault.	Is a combinatorial approach		Computationally too expensive to be deployed practically for large networks.

6 Way Ahead

In this section we would like to describe the motivation for survey of these algorithms. Distributed networks are complex in nature and day by day its utilization and dynamism is increasingly changing which makes these networks more prone to failures. To cope up with these issues network fault management is important. After surveying the existing research in this field it gave us pointers for future research.

Most of the researchers have worked in isolation on the problem related either to probe station or to probe set selection algorithms only. We intend to integrate both these methods and develop a method which outperforms the results achieved through individualist approach. Experimentation will be done using OMNeT++ simulator as it supports the required functionality and is one of the most popular open source software with a very good GUI.

7 Conclusion

In this paper we have evaluated popular strategies used by researchers for solving the problem of minimizing the probe set for fault detection and localization.

The survey indicates that the greedy search approach for probe set selection is better and optimized than the other available active probing algorithms for fault detection and localization. Use of pre-planned probing is not an efficient approach for monitoring distributed networks due to the overhead of increase in management traffic. Adoptive probing is a hybrid approach which will be the best option for network fault management as it overcomes all the drawback of both active and passive monitoring.

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. Shwartz, 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] Mark Brodie, Irina Rish, Sheng Ma, Alina Beygelzimer, Natalia Odintsova, Strategies for Problem Determination using Probing, In IEEE INFOCOM.

[11] M. Natu and A. S. Sethi. Probe station placement for robust monitoring of networks. Submitted to Journal of Network and Systems Management.

[12] M. Steinder and A. S. Sethi. A survey of fault localization techniques in computer networks.Science of Computer Programming, Special Edition, 53(2): 165-194, Nov., 2004.

[13] Abduljalil A. Mohamed and Otman Basir. A New Probing Scheme for Fault Detection and Identification, in IEEE 2009.

[14] Deepak Jeswani, Nakul Korde, Dinesh Patil, Maitreya Natu and John Augustine. Probe Station Selection Algorithms for Fault Management in Computer Networks, in IEEE 2010.

[15] Yongjin Liu, Yanan Wang and Fangping Li. Fault Management of Computer Networks based on Probe Station Selection Algorithms, in International Coriference on Educational and Network Technology (ICENT 2010)

[16] Likun Yu, Xuesong Qiu,Yan Qiao, Xingyu Chen,Yanguang Liu. Optimizing Probe Selection Algorithms For Fault Localization , in IEEE 2010.

[17] Maitreya Natu · Adarshpal S. Sethi · Errol L. Lloyd. Efficient probe selection algorithms for fault diagnosis , Springer Science+Business Media, LLC 2008.

[18] Likun Yu, Lu Cheng, Yan Qiao,Yiguo Yuan,Xingyu Chen. An Efficient Active Probing Approach Based On The Combination Of Online And Offline Strategies, , in IEEE 2010.

[19] Francois Baccelli, Sridhar Machiraju, Darryl Veitch, *Fellow, IEEE*, and Jean Bolot. Probing for Loss: The Case Against Probe Trains,in IEEE Communications Letters, Vol. 15, No. 5, May 2011

[20] Mark Brodie , Irina Rish , Sheng Ma , Natalia Odintsova, Active probing strategies for problem diagnosis in distributed systems, IJCAI, 2003.

