B. TECH. PROJECT REPORT

On

Congestion Aware Probing for Fault Detection in Networks

BY

Shah Vinit Haresh

DISCIPLINE OF COMPUTER SCIENCE AND ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY INDORE

December 2018

Congestion Aware Probing for Fault Detection in Networks

A PROJECT REPORT

Submitted in partial fulfillment of the requirements for the award of the degrees

of
BACHELOR OF TECHNOLOGY
in

COMPUTER SCIENCE AND ENGINEERING

Submitted by:
Shah Vinit Haresh

Guided by:
Dr. Neminath Hubballi

INDIAN INSTITUTE OF TECHNOLOGY INDORE
DECEMBER 2018

B. TECH PROJECT REPORT

Congestion Aware Probing for Fault Detection in Networks

Prepared By:

Shah Vinit Haresh

Under the guidance of

Dr. Neminath Hubballi

Discipline of Computer Science and Engineering

Indian Institute of Technology, Indore

December, 2018

Candidate's Declaration

I, Vinit Shah (ID : 150001029), declare that this project titled, 'Congestion Aware Probe Detection in Networks' and the work presented in it are my own. I confirm that:

This work was done in partial fulfillment for the award of the degree of Bachelor of Technology in 'Computer Science and Engineering' completed under the supervision of Dr. Neminath Hubballi, Assistant Professor, Discipline of Computer Science and Engineering, IIT Indore. Where I have consulted the published work of others, this is always clearly attributed. Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this project is entirely my own work.

Signed :

Date :

Certificate by BTP Guide

It is certified that the statement made by my student is correct to the best of my knowledge.

Dr. Neminath Hubballi
Assistant Professor
Discipline of Computer Science and Engineering
IIT Indore

Abstract

Fault localization is a very common problem faced in network fault management. It is the process of inferring the exact failure in a network from the set of observed symptoms. Since faults are unavoidable, their quick and accurate detection and diagnosis is required. Probing is a method which is commonly used to tackle this problem. Active probing is an active network monitoring technique in which the problem of selecting probes is done online i.e. adapting to the changes in traffic dynamics of nodes and links. This helps to reduce the probe induced traffic with the network traffic and thus efficiently solves the problem. We then propose a sub-optimal solution to the probe selection problem by reducing the problem to a binary linear programming optimization problem. We subsequently simulate a series of experiments by varying the graph structure, average degree of the nodes, number of probe stations, maximum possible probe length and recorded the observations. The trade-off between the number of probe stations and the number of probes selected is also verified.

Acknowledgements

I wish to thank my supervisor, Dr. Neminath Hubballi for his kind support and valuable guidance. It is his help and support, due to which the project design and report came to life.

The opportunity to work on a project under his guidance was a great chance for learning and professional development. In spite of being busy with his duties, he always spared time to listen to my problems, clear my doubts and guide me at every step of my project. Having said that, I would like to express my heartfelt gratitude to Anuja Tayal and Nikhil Tripathi who have also guided throughout the project.

Last but not the least, I would like to thank my family and friends for their constant support.

Shah Vinit Haresh
B.Tech. IV Year
Discipline of Computer Science and Engineering
IIT Indore

Contents

Chapter 1

Introduction

Probing is the most crucial step in assessing the network for any vulnerabilities. Probing is used for the fault identification and localization in networks. One of the most important features of any probing-based scheme is that is an active approach. A set of probes can be sent at periodic intervals. If any failure is detected, the outcome of these probes can be used for further analysis of the faults. If the probe set is kept huge, this adds to the burden of the network management system as well as extra storage and bandwidth. Thus, it is highly desirable that the size of such probe set is minimized.

1.1. Motivation

With the ever growing size and complexity of distributed systems and networks, fault localization have become more frequent and problem diagnosis have become more challenging. Failures are unavoidable in large communication networks. The demand for the Quality of Services (QoS) has been increasing ever since, fault localization has become an important network management task. The timely identification and rectification is important for the reliable operation of the networks. Monitoring is one such technique which is widely used for fault detection in networks. [1] Monitoring is further classified as Passive Monitoring and Active Monitoring. Probing is one such method of active monitoring. A set of nodes of the network are being set up as probe stations and then probes are sent throughout the network for identifying the potential problems. The optimal probe selection problem is proved to be NP-Complete. [2] However with the growing need of fault detection techniques, the problem has to be tackled in a better way with good heuristics. This project is an attempt to do the same.

1.2. Related Works

There have been prior works [3], [4], [5] which try to figure out the solution of detecting faults and localizing them in networks efficiently. These works mainly fall into two broad categories - passive and active monitoring. Recent works in advances fault localization techniques [1] are mostly based on active monitoring. Proper description of various types of fault detection techniques can be found in [6]. This project is an attempt to extend the same by using different heuristics.

1.3. Objective

The main objective of the project is to solve the probe selection problem in a reasonable time using better heuristics so that is can be applied on bigger networks. It involves design and development of the algorithm, testing on various example graphs, sampling of the algorithm on real network graphs and analyzing the performance of the algorithm. The variation of the number of probes selected is then studied on various types of graphs.

1.4. Fault Detection Techniques

Faults are network events that are the root cause of problems that can occur in the network. A failure refers to an inability of a device or service to function correctly. Failure of a single component might disrupt the functioning of the whole network. Faults and failures can occur anywhere in the network i.e. in the software or hardware devices (e.g. links, routers). There are symptoms generally for the failures such as alarms in managed networks, using monitoring mechanisms, or sometimes even by human observation.

In order to guarantee a up time and the quality of services, tools are needed which can help in performing these tasks quickly and correctly while minimizing the cost of diagnosis and maximizing the accuracy. A fault diagnosis solution for current day computer networks should ideally have the small deployment cost, quick localization, high accuracy, diagnosis with low management traffic overhead and ability of diagnosis of cases with multiple failures. [4] Monitoring is one such technique used for identifying the faults in the network. Monitoring involves setting up of monitoring devices at various network components. Monitoring is of two types : *Passive Monitoring* and *Active Monitoring*.

Passive Monitoring does not produce extra traffic. Passive measurements are mainly used to measure metrics pertaining to a certain network element. Any failure condition in the network could generate multiple alarms by monitoring agents. The alarms are then used as symptoms by the Network Management System (NMS) to analyze the exact failure condition in the network. Active monitoring on the other side involves sending probe signals onto the network to sample its behavior. These probes are sent from specially instrumented probe stations periodically at regular time intervals. These probes can vary from small pings to complex test transactions. The use of probes helps the NMS to respond more quickly and accurately to the large number of network events, as opposed to the traditional passive event correlation approach.

1.5. Probing

Active Monitoring is used to obtain end-to-end statistics such as latency, loss and route availability. These statistics are then used to deduce the health of the network components. Network parameters and conditions can also be extracted from these probes. It also provides flexibility in the design of probes with particular properties to get corresponding readings. However, there is also a disadvantage of probing. Probes may modify route conditions and increase the congestion on some routes and disturb the traffic behaviour for which monitoring devices are set up. To minimize these effects, probe signals of low bandwidth are used.

Diagnosing a network requires probe stations to be set up on various nodes of the network. The probes are then sent from these probe stations to all the nodes and links throughout the network. The configuration of both the probing stations and probes impose a cost to network management. The probing stations add to the cost because the probing code must be installed, operated, and maintained. On the other hand probes impose an additional cost because their use entails additional network traffic overhead and also due to the collection, storage and analysis of probe results. The probe stations selected should be such that they are able to probe the entire network. Thus there exists a trade-off between the number of probe stations and number of probes to be sent. More number of probe stations implies lesser number of probes and vice-versa. Thus our aim is to find the optimal balance point between the number of probe stations and the number of probes.

There are two types of probing techniques :

1. Pre-planned probing : This involves the offline selection of the probes and periodically sending the probes into the network. This is then followed by a passive data

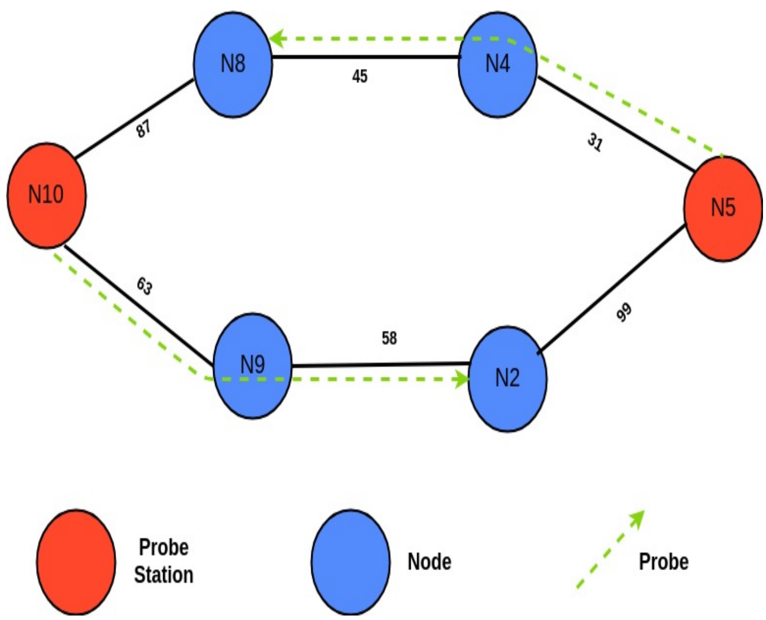

Figure 1.1: Example of probes and probe stations

mining approach to get the knowledge of the current state of the network. Another significant drawback in this approach is the involved difficulty in envisaging all possible problems and generating a probe set for it. This approach is mainly useful because it does not cause a overhead of selecting the probes.

2. Active probing : It adapts the probing strategy to the observed network state. Instead of sending probes for locating all potential problems in the network, it sends a small number of probes initially and then adapts the probe set to the observed network state. This approach significantly reduces management traffic and provide more accurate and timely diagnosis than pre-planned probing. Our main aim is to be able to obtain accurate, reliable estimates using only a small number of probes and using probe signals of low bandwidth. This however has the overhead of selection of probes for each new state of the network.

Chapter 2

Proposed Method for Probe Selection

The network can be considered as a weighted graph $G = (V, E)$ where V denotes the vertices and E denotes the edges. The vertices represent the nodes of the network. The edges represent a link between the nodes and the weight on the edge represent the amount of traffic i.e. I/O statistics of the link. The path followed by the probe originating from a probe station is known as the probe-path. For our problem, we have a set of identified probe stations (PS). Since the probe selection problem is NP-complete (Set Cover problem can be reduced to the problem) [4], we need to have some heuristics. Therefore, we have put a bound on the maximum probe-length to a constant L instead of the original unbounded problem. After adding this constraint, the solution to this new problem will be a sub-optimal solution of the original probe-selection problem. Thus our problem formally is :

Given a weighted graph $G = (V, E)$ with a set of probe stations $PS \subseteq V$, find an optimal set of probes such that probe-induced traffic add to the less congested nodes while satisfying the constraints that every node failure can be discovered by at-least one probe and any probe is of maximum L length.

2.1. Candidate Probe Selection

The candidate probe (CP) set is generated by running Dijkstra algorithm from all probe stations. This set is then used to define the probe-matrix A_{n*cp} where $cp = |CP|$ and $n = |V|$. $A_{i,j}$ indicates whether j^{th} candidate probe passes through the i^{th} node (boolean value).

Consider the sample network network graph in Figure 2.1. Let us restrict the maximum probe length to 2 hops. The possible candidate probes are

1. CP_1 = ({10,8,1}, 225) , CP_2 = ({10,9,2}, 149), CP_3 = ({10,8,3}, 257), CP_4 = ({10,5,4}, 132), CP_5 = ({10,5,6}, 144), CP_6 = ({10,5}, 45), CP_7 = ({10,9,7}, 156),

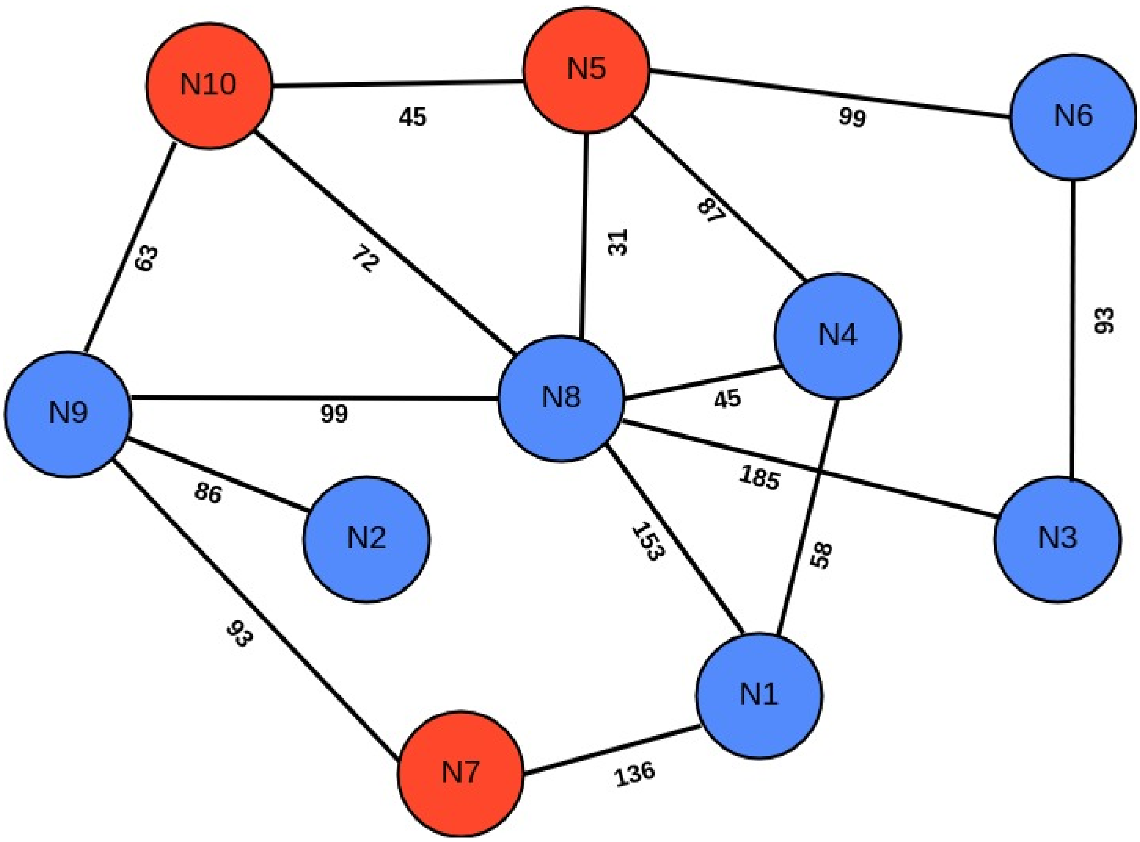

Figure 2.1: Sample Network Graph

CP_8 = ({10,8}, 72), CP_9 = ({10,9}, 63)

2. CP_{10} = ({5,4,1}, 145) , CP_{11} = ({5,6,3}, 192), CP_{12} = ({5,4}, 87), CP_{13} = ({5,6}, 99), CP_{14} = ({5,8}, 31), CP_{15} = ({5,8,9}, 130), CP_{16} = ({5, 10}, 45)

3. CP_{17} = ({7,1}, 136) , CP_{18} = ({7,9,2}, 179), CP_{19} = ({7,1,4}, 194), CP_{20} = ({7,1,8}, 289), CP_{21} = ({7,9}, 93), CP_{22} = ({7,9,10}, 156)

2.2. Final Probe Selection

After getting the candidate probe set, the main problem is to find the final probe $P \subseteq CP$ set in the network. This problem can now be modelled as a linear optimisation problem in the following manner :

1. Each probe has its own cost. The cost of selecting the probe is the total sum of weights of all the links which this probe has to travel. Thus, we now define the

Probe Weight Vector (PWV_{cp*1}) as the vector containing the cost of all candidate probes where PWV_i represents the cost of the i^{th} probe.

2. Now, each candidate probe will either be present in the final probe set or not. Thus the final outcome of each candidate probe can be viewed as a boolean variable where 1 indicates the probe is selected and 0 if it is not. We have thus defined a vector X_{cp*1} which is binary in nature where X_i represents whether the i^{th} probe is as a part of final probe set.

3. The constraint still has to be put the each node is detected by atleast one probe. This can be easily imposed if the following inequality is imposed :

$$A_{n*cp} * X_{cp*1} \geq 1$$

4. Finally the objective of the problem is to minimise the total cost of probes selected which can be represented by :

$$PWV^T_{cp*1} * X_{cp*1}$$

Thus, we have changed our probe selection problem to the following binary integer linear programming problem :

$$\text{min. } PWV^T_{cp*1} * X_{cp*1} \text{ subject to}$$
$$A_{n*cp} * X_{cp*1} \geq 1$$
$$\text{where } X_j \in 0, 1 \forall j \in CP$$

Chapter 3

Experimental Simulations and Results

3.1. Small-sized Computer Networks

The above example network was simulated in NS-3. The NS-3 file takes the adjacency matrix of the undirected unweighted graph as its input. CBR flows were then sent between various nodes which were randomly picked. This leads to the generation of various pcap trace files. The number of packets (I/O packets) were then extracted from these pcap files which act as the edge weights. This adjacency matrix of the weighted graph is then fed to the R script file for selecting the optimal probe set.

For the example network depicted in the graph from Figure 2.1, the probes selected in the optimal probe set are :

$CP_2 = (\{10,9,2\}, 149)$, $CP_{14} = (\{5,8\}, 31)$, $CP_{11} = (\{5,6,3\}, 192)$, $CP_{19} = (\{7,1,4\}, 194)$.

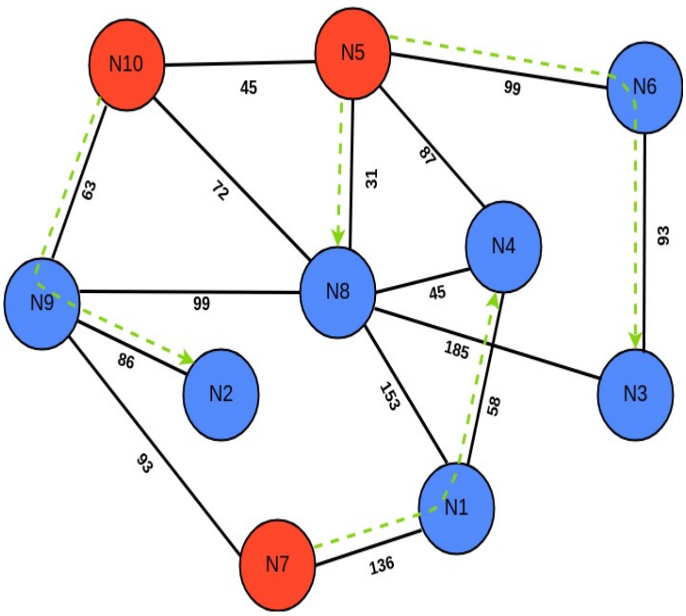

Figure 3.1: Probes Selected

3.1.1. Variation of Probes Selected with Average Degree

An incremental analysis of the above experiment was performed with different graph structures having a fixed number of nodes. In our experiments, we took graphs of 20 nodes. We kept the maximum probe length as 5 and the number of probe stations as 3. We then incrementally started adding edges to the graph starting from 23 edges to 52 edges, taking 10 sample graphs each time. The experiment was left to run for about 6 hours out of which most of the time was consumed due to the packet transfers between the nodes during simulation in NS-3. The probe set selection by solving the binary optimization problem is done in the R file which usually took around 2-3 seconds for selecting the probe set for a single graph.

The observations got are plotted in Figure 3.2. Correlation of 85% was observed suggesting a near linear increase in the number of probes selected with the increase in the average degree.

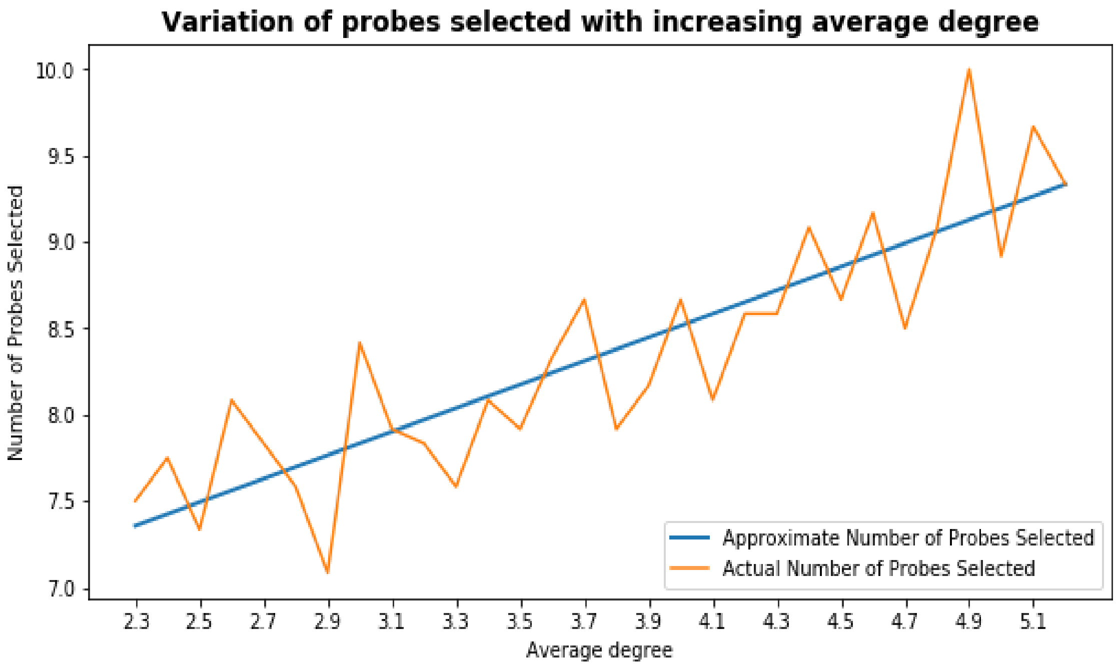

Figure 3.2: Probes Selected

3.1.2. Variation of Probes Selected with Graph Size

The experiment was then simulated with various graphs having a fixed average degree, number of probe stations, maximum probe length. In our simulations, the average degree, number of probe stations and maximum probe length were taken as 8, 4, and 7 respectively.

$$\text{Average degree} = \frac{2 * \text{Number of edges}}{\text{Number of nodes}} \ldots\ldots [7]$$

The experiment was run on small graphs with number of nodes ranging from 10 to 30. The experiment was kept to run for about 5 hours collecting various readings for the same parameters and the number of probes selected were then plotted as a function of the number of nodes. Correlation of 99.22% was found, clearly suggesting a linear increase of the number of probes selected with the number of nodes.

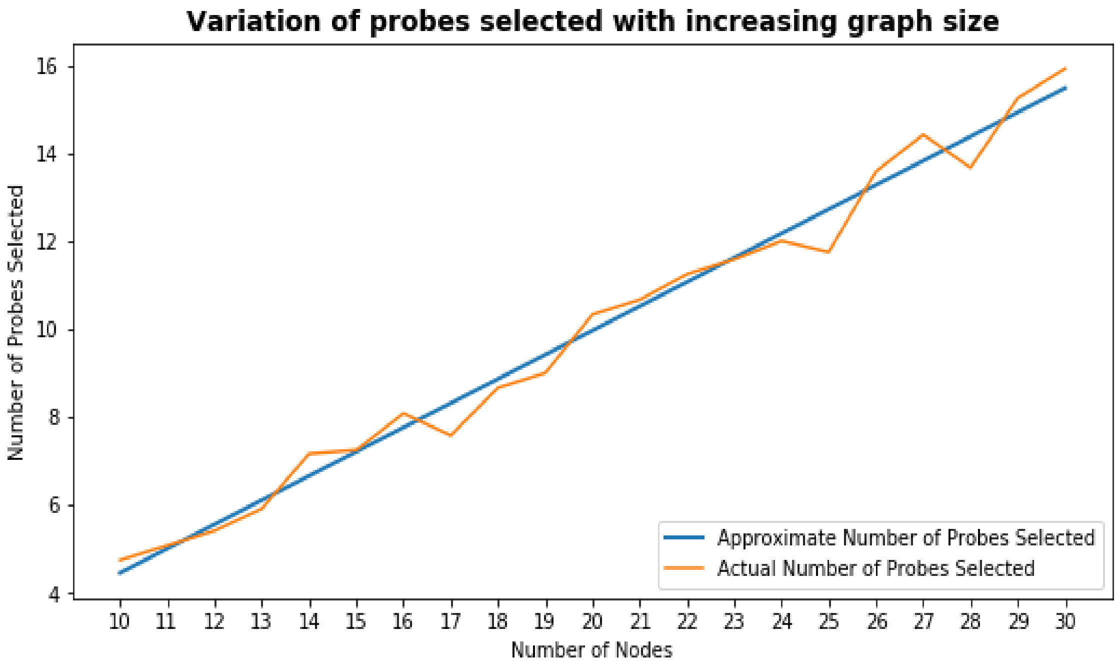

Figure 3.3: Probes Selected

3.2. Large-sized Computer Networks

Simulating large networks in NS-3 is very difficult. Owing to the large size of the graph networks, the edge weights for the graphs which were initially chosen by recording packets transferred on each link, were then chosen randomly from a fixed range. The experiment was then simulated with various real computer networks. One such example graph was the Rocketfuel engine. [8] It is a mapping of 10 ISPs in Europe, Australia and the United States with a database of over 50 thousand IP addresses representing 45 thousand routers in 537 POPs connected by 80 thousand links. These ISPs having graph sizes ranging from approximately 50 nodes to 350 nodes. This formed the adjacency matrix for the experiments. The overall process of probe selection is done quite fast and usually took 3-4 seconds in all our experiments.

3.2.1. Tradeoff between Number of Probes Selected and Probe Stations

Both probe stations and probes have their own cost of installation. Probe stations have a cost because the probing code for the probes must be installed as well as the probe station has its own maintenance cost. On the other hand, probes have an cost mostly because they add to the traffic of the network as well as they have to be collected, stored and analysed. If there is only probe station in the network, the number of probes sent throughout can be N where N is the number of nodes in the graph. If probe stations are set up on all nodes of the graph, the number of probes sent will be 0. This suggests, that the number of probes selected decreases with the increase in number of probe stations.

The above claim was experimentally verified using the Rocketfuel graphs. We took one with 315 nodes and 972 edges. The probe stations were made to gradually increase from 3 to 10. The observations obtained are plotted in the figure 3.4.

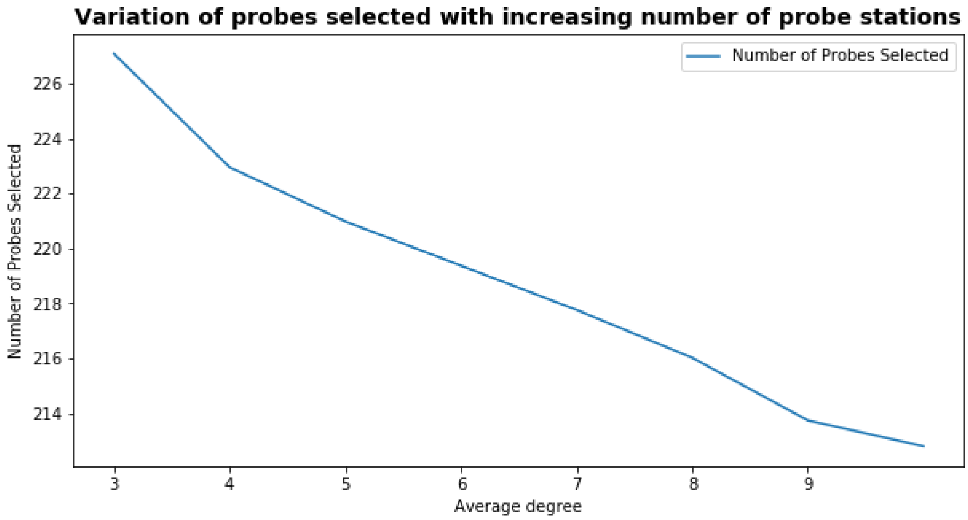

Figure 3.4: Probes Selected

3.2.2. Variation of Probes Selected with Average Degree

Different graphs having 500 nodes and an average degree ranging from 4 to 5.4 were generated randomly. The maximum probe length chosen 10 and 3 probe stations were used. 10 samples of each parameter set were used so that the average could be used as a good estimate for the corresponding parameters. This time a correlation percentage of 91% was observed.

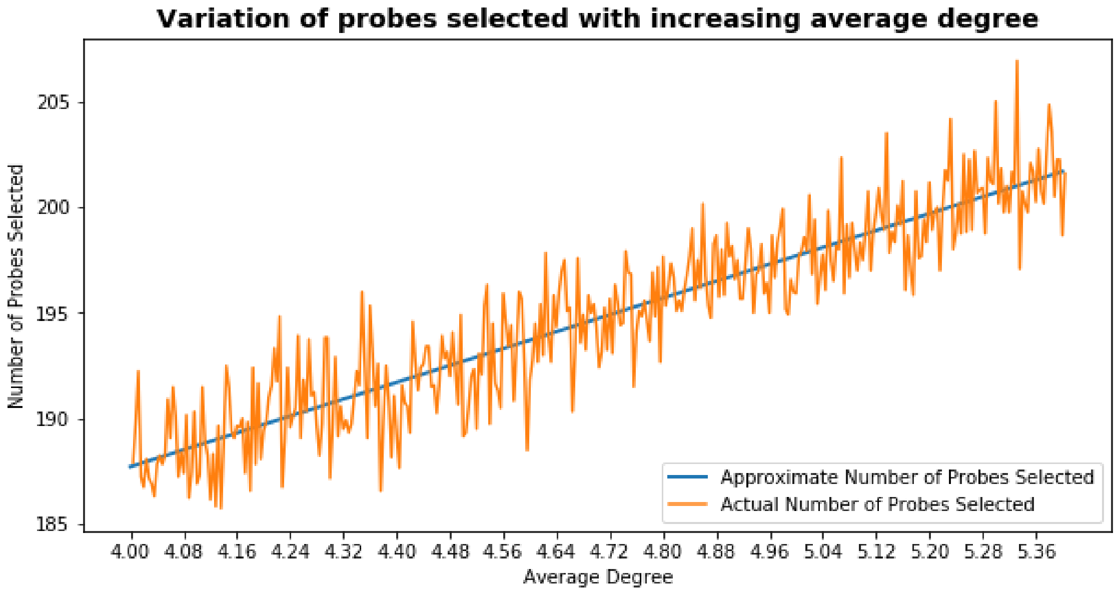

Figure 3.5: Probes Selected

Similar experiment was performed with different set of parameters. For this experiment, graph size chosen was 200 nodes and the number of probe stations was increased to 4. The maximum probe length was again kept at 10 and the average degree was gradually increased from 5 to 6. This time, 50 samples of each parameter set were used so as to get a better estimate. Correlation of 95% was obtained this time.

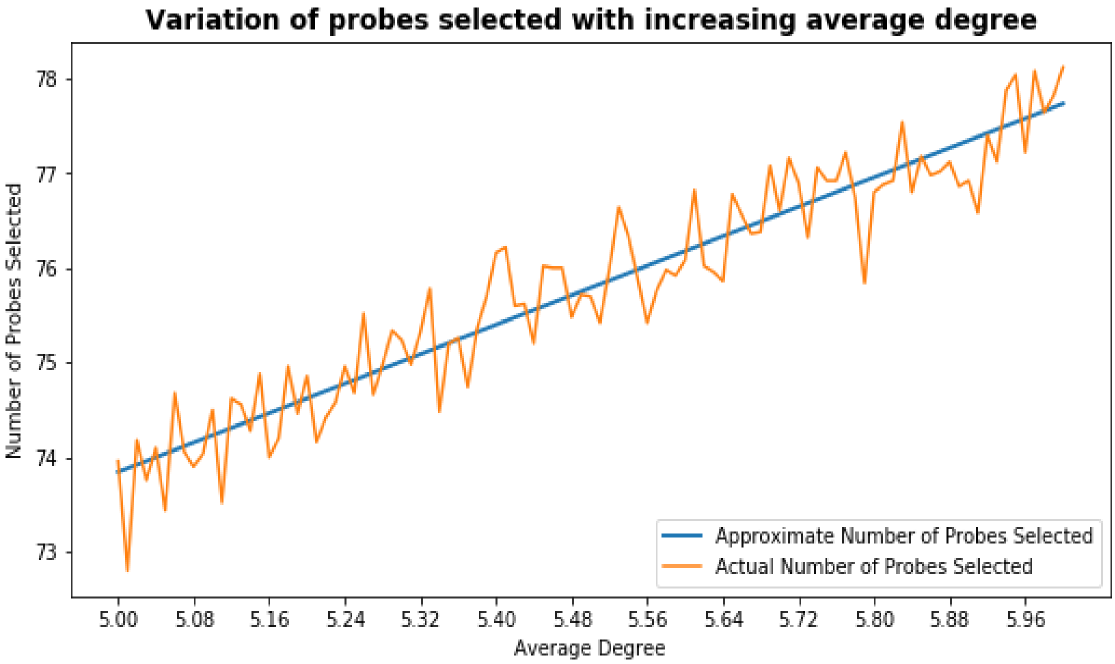

Figure 3.6: Probes Selected

The increase in number of probes selected with the average degree seems counter-intuitive since increase in average degree implies increase in the number of edges (keeping number of nodes constant) and increase in number of edges suggests that some existing probes might extend to reach the nodes traversing these edges thereby keeping the probes selected constant. However, with the increase in number of edges, the shortest distance of nodes from the probe stations gets decreased and since our algorithm picks the optimal probe set greedily instead of extending a previous probe, it is better to assign a new node to cover these new nodes, thereby increasing the number of probes selected.

3.2.3. Variation of Probes Selected with Graph Size

For these set of experiments, the maximum probe length, number of probe stations and the average degree of nodes were kept constant to study the behaviour of the probes selected with the number of nodes.

In the first set of experiments, the average degree of the nodes was kept at 6, maximum probe length was kept at 10 and the number of probe stations was kept at 3. The algorithm was run 20 times for a fixed parameter set. The number of nodes was gradually increased from 400 to 450 and the observations recorded are depicted in Figure 3.7. A correlation of 98.45% was observed.

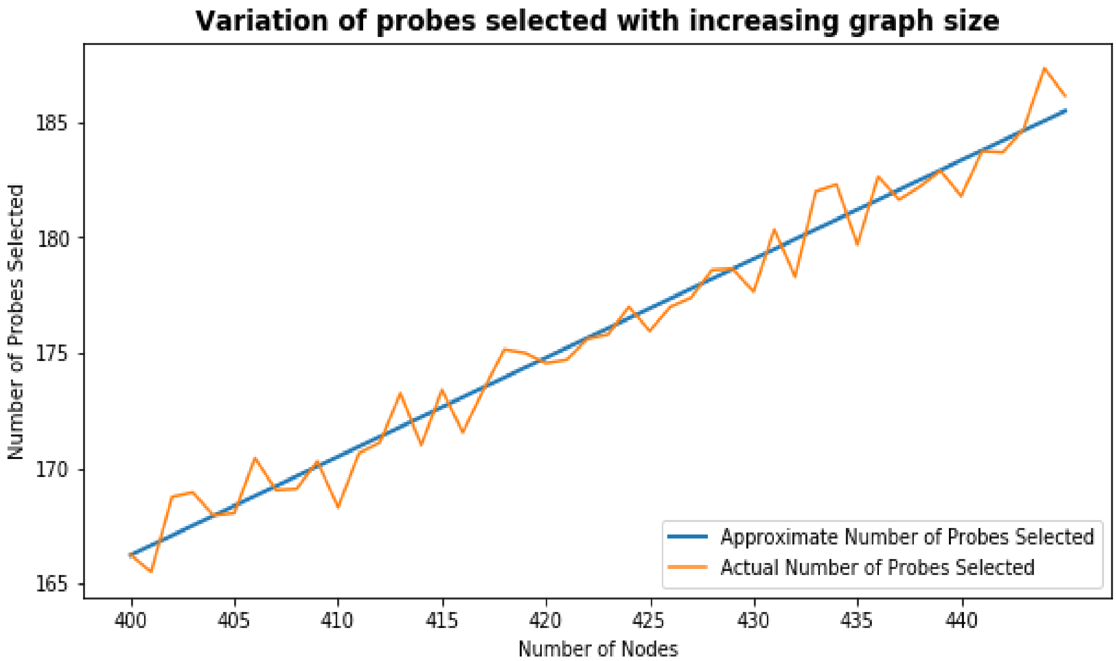

Figure 3.7: Probes Selected

In the second set of experiments, the average degree was again kept at 6, the maximum probe length was again kept at 10 and the number of probe stations was 4. This time each parameter set was run for 50 times and the corresponding number of probes selected was recorded. A correlation of 99.34% was observed.

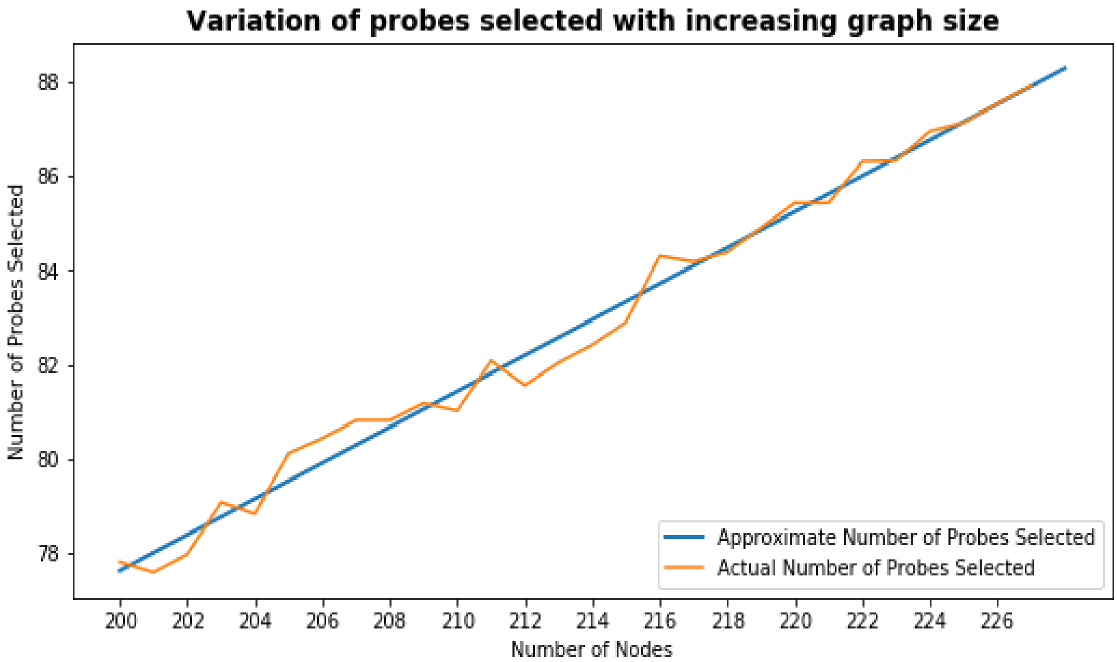

Figure 3.8: Probes Selected

From the above sets of experiments, it is clear that the the number of probes selected in the optimal probe set linearly increases with the increasing graph size keeping other parameters constant.

Chapter 4

Conclusions and Future Works

4.1. Conclusions

In this project, we proposed an approach for adaptive probing for detection of faults in computer networks. Our method adapts to the current network state i.e. which routers and links are working and have less traffic. The current I/O statistics of the network are fed as the parameters to our network graph. We then proposed a greedy algorithm for the candidate probe set selection by limiting the maximum probe length to a fixed constant L. This was then formulated as a binary linear optimization problem for the probe set selection.

Simulation experiments were then performed using NS-3 and R to show that the probes indeed adapt to the traffic dynamics. The proposed algorithm was then tested on various real computer network topologies to check the scalability of the algorithm. Each additional probe has its own cost hence the number of probes selected is an important factor to be considered while choosing the probe stations. The variation of the number of probes selected was then studied with respect to different parameters. The trade-off between the number of probe stations and the number of probes selected was also experimentally verified.

4.2. Future Works

Future works include estimating theoretical bounds on the number of probes selected depending on the structre of graphs, probe stations, maximum probe length. Sometimes any probe station, node or link between routers can go down. In such cases, it is always essential to have a backup route so that the network can still be monitored. If any probe passes passes through this link or node, the probe signal might not be sent. Thus, these cases need to be studied properly so that fault detection is ensured.

Bibliography

[1] Ayush Dusia and Adarshpal S. Sethi. "Recent Advances in Fault Localization in Computer Networks". In: *Communication Surveys and Tutorials* 18 (2016), pp. 3030–3051.

[2] Maitreya Natu, Adarshpal S. Sethi, and Errol L. Lloyd. "Efficient Probe Selection Algorithms for Fault Diagnosis". In: *Telecommunication Systems* 37 (2008).

[3] Anuja Tayal et al. "Congestion-Aware Probe Selection for Fault Detection in Networks". In: *Communication Systems AND Networks (COMSNETS), 2018 10th International Conference* (2018), pp. 407–409.

[4] Deepak Jeswani, Maitreya Natu, and R. K. Ghosh. "Adaptive Monitoring: Application of Probing to Adapt Passive Monitoring". In: *Journal of Network and Systems Management* 23.4 (2015), pp. 950–977.

[5] Deepak Jeswani et al. "Probe Station Selection Algorithms for Fault Management in Computer Networks". In: *Communication Systems AND Networks (COMSNETS), 2010 2nd International Conference*. IEEE, 2010, pp. 1–9.

[6] Malgorzata Steinder and Adarshpal S. Sethi. "A Survey of Fault Localization Techniques in Computer Networks." In: *Science of Computer Programming* 53 (2004), pp. 165–194.

[7] Maximiliano Pinto Damas and Lilian Markenzon Nair Maria Maia de Abreu. "New Concepts and Results on the Average Degree of a Graph". In: *Applicable Analysis and Discrete Mathematics* 1 (2006), pp. 284–292.

[8] Neil Spring, Ratul Mahajan, and David Wetherall. "Measuring ISP Topologies with Rocketfuel". In: *SIGCOMM Comput. Commun. Rev.* 32.4 (2002), pp. 133–145.