PROJECT PHASE-I REPORT

# NETWORK MAPPER

**Guided by**
*Mr. Rajesh Dhakad*

**Co-guided by**
*Ms. Neha Mehra*

**Submitted by**
Sarvagya Soni
Shivam Joshi
Sourabh Shrivastava
Shubham Kumawat
Yashil Jijhotiya

**Computer Engineering Department**
**Shri G.S. Institute of Technology and Science**
**Indore (M.P.)**
**2015-16**

# Shri G.S. Institute of Technology and Science Indore (M.P.)

## RECOMMENDATION

This is to certify that the project report entitled **“NETWORK MAPPER”** submitted by Sarvagya Soni, Shivam Joshi, Sourabh Shrivastava, Shubham Kumawat, Yashil Jijhotiya, students of final year B.E. (Computer Engineering) in the year 2015-2016 of this Institute, is a satisfactory account of their Project Phase-I work based on syllabus and the same is recommended for the Phase-I examination.

Mr. Rajesh Dhakad
**Guide**
Computer Engg. Department
S.G.S.I.T.S.
Indore

Prof. D.A. Mehta
**Head**
Computer Engg. Department
S.G.S.I.T.S.
Indore

Dean Academic
S.G.S.I.T.S.
Indore

**Shri G.S. Institute of Technology and Science Indore (M.P.)**

## CERTIFICATE

This is to certify that the  project report entitled " NETWORK MAPPER" submitted by , students of final Sarvagya Soni, Shivam Joshi, Sourabh Shrivastava, Shubham Kumawat, Yashil Jijhotiya year B.E. (Computer Engineering) in the year 2015-2016  of this Institute, is a satisfactory account of their Project Phase-I work based on syllabus.

**Internal Examiner** **External Examiner**

# Contents

# Chapter 1

# Introduction

## 1.1 Preamble

The aim of this project report is to describe the network mapping tool, for surveillance and monitoring of network infrastructure using active probing technique.

## 1.2 Need of the project

Network mapper is a dynamic discovery and mapping tool that helps network administrators/ Data center admins to visualize their complete network infrastructure with a live network map.

There's no facility available for the network admin for getting a notification when a link goes down. This network mapper will generate an automatic notification when a link goes down.

There are other tools which are present to perform network mapping. But they are generally based on SNMP(Simple Network Management Protocol). The disadvantages associated with SNMP based approach are :

- It is not compatible with all the devices.
- It has a long header size.
- It has security issues in SNMP v1 and v2 i.e there's no encryption of data.
- In SNMP v3 cryptographic calculations is present but it can increase CPU load approx. By 20 %.

## 1.3 Problem statement

In a large network setup it is hard to visualize the network topology. The network admin faces several problems, such as broken link. The network administrator is not aware of these problems instantly.

Even when he detects that something is going wrong, it becomes an uphill task to trace those broken links on the network. Also, he doesn't know the uptime and downtime of that link, and no records are stored.

There is a requirement for a tool to dynamically generate and monitor network graph and to notify the admin when a broken link is detected with its location and downtime.

## 1.4 Objective

- To create a network mapping tool, for surveillance and monitoring of network infrastructure using active probing technique.
- To generate generate notification message to network admin when a link goes down.

## 1.5 Solution approach

The network mapper provides a tool to visually monitor the entire LAN network. Whenever a device is added in our network(manually), then it will be shown automatically. A message will be generated whenever a link is down. An entry of every device will be maintained in a log file for network administrator.

We are using active probing technique because it is compatible with all the devices, no such security is needed.

Active probing

This technique relies on traceroute-like probing on the IP address space. These probes report back IP forwarding paths to the destination address. By combining these paths one can infer router level topology for a given POP. Active probing is advantageous in that the paths returned by probes constitute the actual forwarding path that data takes through networks. It is also more likely to find peering links between ISP's.

# Chapter 2

# Background

## 2.1 Related tools

**Icinga**

Icinga is an offshoot of Nagios that is currently being rebuilt anew. It offers a thorough monitoring and alerting framework that's designed to be as open and extensible as Nagios is, but with several different Web UI options. Icinga 1 is closely related to Nagios, while Icinga 2 is the rewrite. Both versions are currently supported, and Nagios users can migrate to Icinga 1 very easily.

**NeDi**

NeDi may not be as well known as some of the others, but it's a great solution for tracking devices across a network. It continuously walks through a network infrastructure and catalogs devices, keeping track of everything it discovers. It can provide the current location of any device, as well as a history.

NeDi can be used to locate stolen or lost devices by alerting you if they reappear on the network. It can even display all known and discovered connections on a map, showing how every network interconnect is laid out, down to the physical port level.

**Zabbix**

Zabbix monitors servers and networks with an extensive array of tools. There are Zabbix agents for most operating systems, or you can use passive or external checks, including SNMP to monitor hosts and network devices. You'll also find extensive alerting and notification facilities, and a highly customizable Web UI that can be adapted to a variety of heads-up displays. In addition, Zabbix has specific tools that monitor Web application stacks and virtualization hypervisors.

Zabbix can also produce logical interconnection diagrams detailing how certain monitored objects are interconnected. These maps are customizable, and maps can be created for groups of monitored devices and hosts.

## 2.2 Literature review

Problem determination is one of the most important tasks in managing networking systems. Probing (both at the transaction and network levels) has been widely used for assessing compliance with Service Level Agreements and locating problems in networking systems. However a probing scheme which uses a fixed setof regularly scheduled probes can be expensive in terms of the number of synthetictransactions needed, especially for the task of problem determination. This paperintroduces an active probing scheme to reduce the number of probes needed. Our key idea is to divide the problem determination task into two steps. We first usea relatively small number of fixed, regularly scheduled, probes for detecting thata problem has occurred. In the second phase, once occurrence of a problem isdetected, additional probes are issued on-the-fly to acquire additional informationuntil the problem is localized. We develop algorithms for selecting an optimal setof probes for problem detection and choosing which probes to send next based onwhat is currently known. We demonstrate through both analysis and simulationthat the active probing scheme can greatly reduce the number of probes and thetime needed for localizing the problem when compared with a non-active probingscheme.

1 Introduction

The rapid growth in size and complexity of distributed systems makes performancemanagement tasks such as problem determination – detecting system problems and isolatingtheir root causes – an increasingly important but also extremely difficult task. Forexample, in IP network management, we would like to quickly identify which router orlink has a problem when a failure or performance degradation occurs in the network. Inthe e-Commerce context, our objective could be to trace the root-cause of unsuccessfulor slow user transactions (e.g. purchase requests sent through a web server) in order toidentify whether it is a network problem, a web or back-end database serverproblem,etc. Another example is real-time monitoring, diagnosis and prediction of the“health”of a large cluster system containing hundreds or thousands of workstations performingdistributed computations (e.g., Linux clusters or GRID-computing systems).Two general approaches are commonly used for problem determination. The firstis event correlation ([13, 7, 12]), in which every managed device is instrumented to1emit an alarm when its status changes. By correlating the received alarms a centralizedmanager is able to identify the problem. However, this approach usually requiresheavy instrumentation, since each device needs to have the ability to send out the appropriatealarms.

Also it may be difficult to ensure that alarms are sent out, e.g. by adevice that is down. To avoid these problems, which arise from using a fixed, “passive”data-gathering approach, a more active probing technology has been developed, whichallows one to test network and system components in order to provide more accurateand cost efficientproblemdetermination.Aprobe is a test transaction whose outcome depends on some of the system’s components;accurate diagnosis can be achieved by appropriately selecting the probes andanalyzing the probe outcomes. Previous work has focussed on selecting probes in anoff-line, pre-planned fashion. Prior information about network and system dependencies,which problems need to be detected most urgently (perhaps because they are moreimportant or more likely to occur), and other forms of prior knowledge are used to constructa set of probes that is small (thereby reducing probing costs such as data storagerequirements and network load) yet provides extensive coverage so that problems canbe diagnosed. These probes are scheduled to run periodically to provide informationabout what problems may be occurring. A typical example is IBM’s EPP technology

Using pre-planned probe sets suffers from considerable limitations. Because theprobe set is computed off-line, it needs to be able to diagnose all possible problemswhich might occur. However in practice constructing such a probe set can be quitedifficult, because one must envisage all problems one would like to be able to diagnoseand construct probes for them. A pre-planned probe set may also be very wasteful,because many problems that might occur do not in fact ever happen. Probes to detectsuch problems enlarge the probe set unnecessarily, but knowing which probes can safely omitted can usually only be determined on-line by monitoring which problemsin fact occur.

Another disadvantage of pre-planned probe sets is that because the probes run periodicallyat regularly scheduled intervals, there may be a considerable delay in obtaining information when a problem occurs. It is clearly desirable to detect the occurrence ofa problem as quickly as possible. Furthermore, once the occurrence of a problem hasbeen detected, additional information may be needed to diagnose the problem precisely.This information may not be obtainable from the results of the pre-planned probes - additionalprobes may need to be sent to obtain it. These probes should be appropriatelyselected “on-demand”, based on the results of the previous probes.Our works develops a methodology called active probing that addresses these limitations.This involves probing in an interactive mode, where probe results are analyzedto determine the most likely diagnosis, and then additional probes are selected and sentin order to gain further information. This process may repeat - once additional probe resultsare obtained, the diagnosis is refined, and, if necessary, more probes are selected,and so on, until the problem is completely determined. The idea of this approach is to“ask the right questions at the right time”.

Active probing selects and sends probes as needed in response to problems thatactually occur. It therefore avoids both the difficulty of constructing probes for allpossible problems as well as the waste inherent in using probes for problems that infact never occur. Furthermore, because probes are selected on-line to obtain furtherinformation about particular problems that have occurred, they need not circulate regularlythroughout the entire network; instead they can be targeted quickly and directlyto the points of interest. Thus fewer probes are needed than in a pre-planned approach,allowing for a considerable reduction in probing costs.

Implementing active probing requires developing solutions for the following issues:

1. A small probe set must be pre-selected, so that when a problem occurs we candetect that something has gone wrong.

2. The probe results must be integrated and analyzed, to determine the most-likelynetwork state.

3. The “most-informative” probes to send next must be selected, based on the analysis of previous probe results.

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

In this paper we describe efficient solutions for each of these issues and integratethem into a practical system for active probing. We also analyze the costs and benefitsof active probing and show experimentally that active probing substantially reducesthe number of probes needed to diagnose problems when compared with pre-plannedprobing. Our preliminary results suggest that active probing may be a powerful andeffective technique for problem determination.

## 3.1.2 Existing discovery systems :

Initial topology discovery systems, such as Mercator, based on hop-limited probing methodssent probes out from a single location. Probing in such a way is likely to result in a tree-liketopology and will tend to miss out ”cross-links”, i.e. traversal branches. Subsequent worktherefore adopted multiple source approaches, in which multiple probing sources are used.Quite like looking glasses, a large number of public traceroute servers have been made available and can be found at [73]. Initially their use was mainly intended for debugging purposes butthey can also be used by topology discovery tools, although their web interfaces sometimes lackconvenience.

An example of such a platform is PlanetLab [83] which has currently 694 machineshosted on 335 different sites which provides a network testbed for many active research projectsof all kinds, covering file sharing, content distribution networks, QoS overlays, anomaly detectionmechanisms, and network measurement tools. An implementation making use of PlanetLab is scriptroute  aimed at providing traceroute servers’ accessibility but with the flexibility ofrunning a variety of measurement tools.Other approaches have deployed their own measurement infrastructure, like the well-knownSkitter project of CAIDA which has between 20 and 30 available monitors world-wideproviding traffic measurements services for researchers. The Active Measurement Project(AMP) of the National Laboratory for Applied Network Research (NLANR), althoughrecently terminated in July 2006, took a slightly different approach by letting 150 monitors,mainly deployed in the United States, probe each other in a full mesh in order to obtaindense coverage of the underlying network. Another project, Rocketfuel , concentrates on discovering the internal topologies of ISPs. A more recent trend has been to move towards a more distributed system as demonstrated bya project like DIMES from the Tel-Aviv University. DIMES relies on the distribution of small portable agents to the community, in a similar spirit to ones such as SETI@home, whichperform typical traceroute and ping probes around the globe.

Our praposed approach :

Active probing is an active network monitoring technique that has potential for developing effective solutions for fault localization. In this paper we use active probing to present an approach to develop tools for performing fault localization. We discuss various design issues involved and propose architecture for building such a tool. We describe an algorithm for probe set selection for problem detection and present simulation results to show its effectiveness.

# Chapter 3
# Requirement Analysis

## 3.1 Functional requirements

In a large network setup it is hard to visualize the network topology. The network admin faces several problems, such as broken link. The network administrator is not aware of these problems instantly.

The network admin should know about anamolies present in his network instantly and should be able to solve them in quick succession of time.Also the admin should know the the status of a link whether it is up or down but without a proper tool it is very tedious job for the network admin.

What he requires is a tool which can generate a graph for the given network so that he can visualize his network eaisly and can find the anamolies present in it instantly.

Fuctional requirements -

1. **Visually moniter entire LAN network.**

Every device in a network will be shown as an icon, the live link connecting two nodes in a green connector and the dead link in red connector. For every type of device we have a defined icon, such as for routers, desktops and servers.

2. **Dynamic identification.**

If a device is added in our network(manually), then it will be shown automatically. The system will refresh itself after defined time period.

3. **Automatic message generation**

Whenever any link is down, (which will be shown in red on map) an automatic message/SMS will be generated to the network administrator.

**4. Maintain log**

The system will maintain a log of links, indicating the uptime and down time of the network device. It will also show the current status of link.

**Additional functionalities (for CCTV monitoring)**

Identify if the CCTV camera is blocked or not using image processing in matlab.

## 3.2 Diagrams

**1) Use case Diagram**

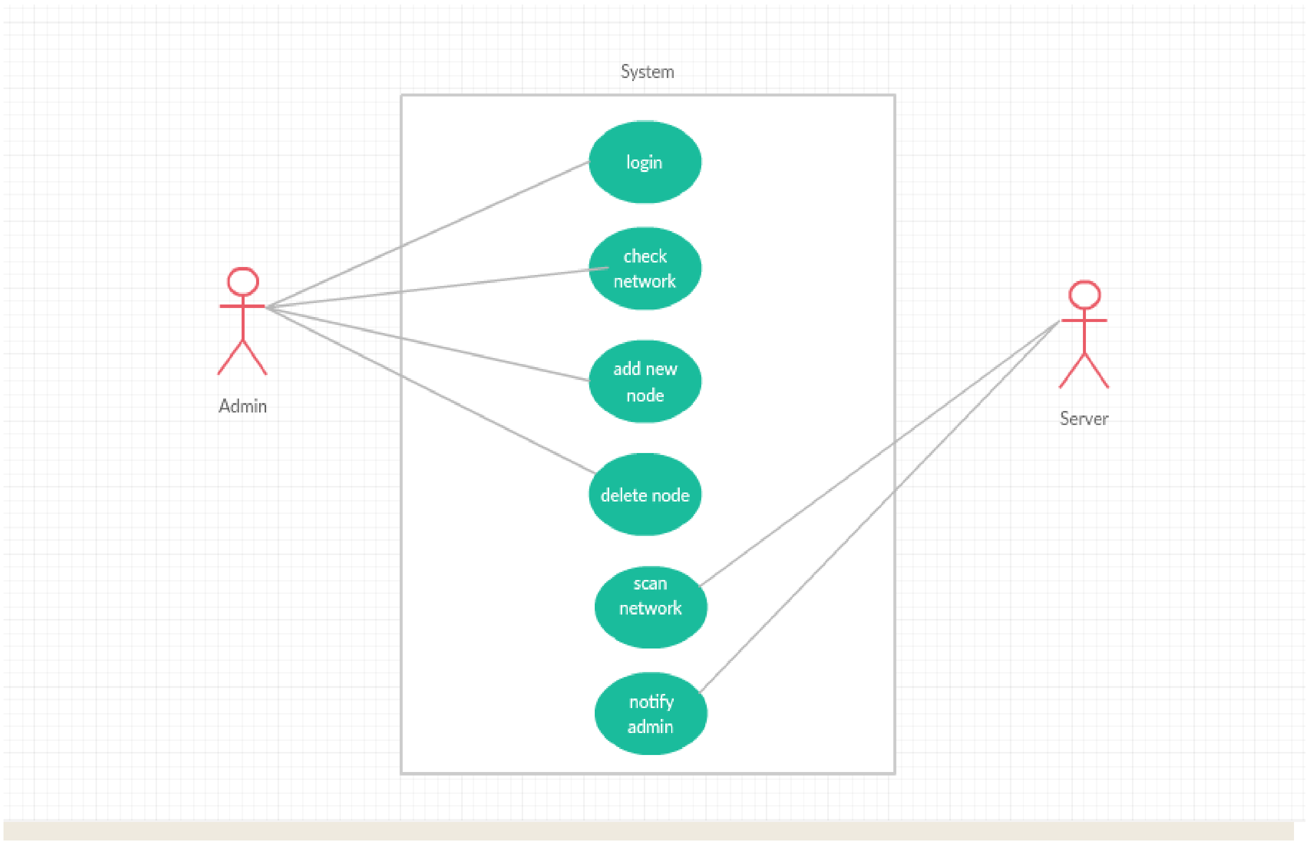

Fig 4.1

## 2) Activity Diagrams

1. Login

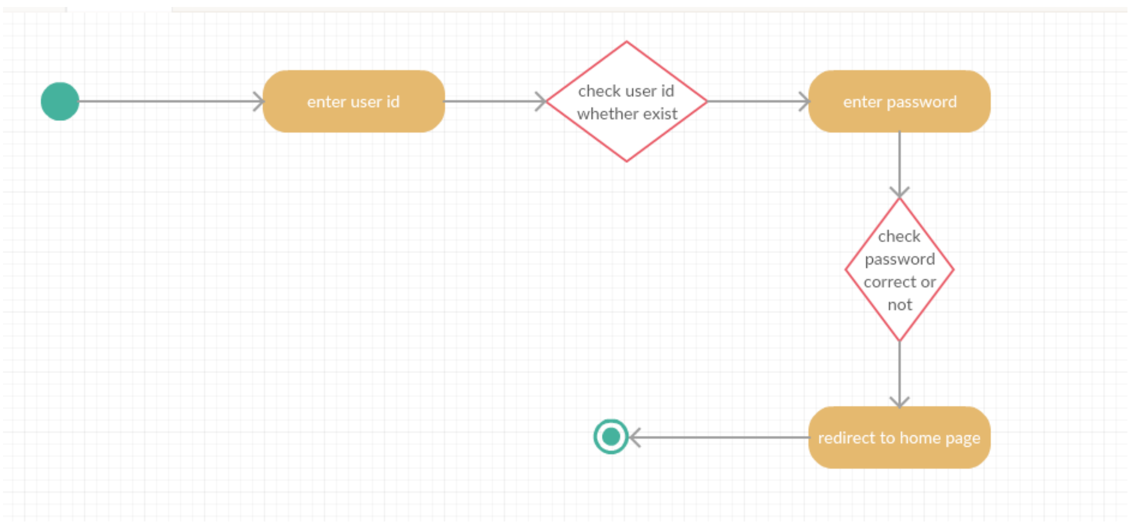

Fig 4.2

2. Check network

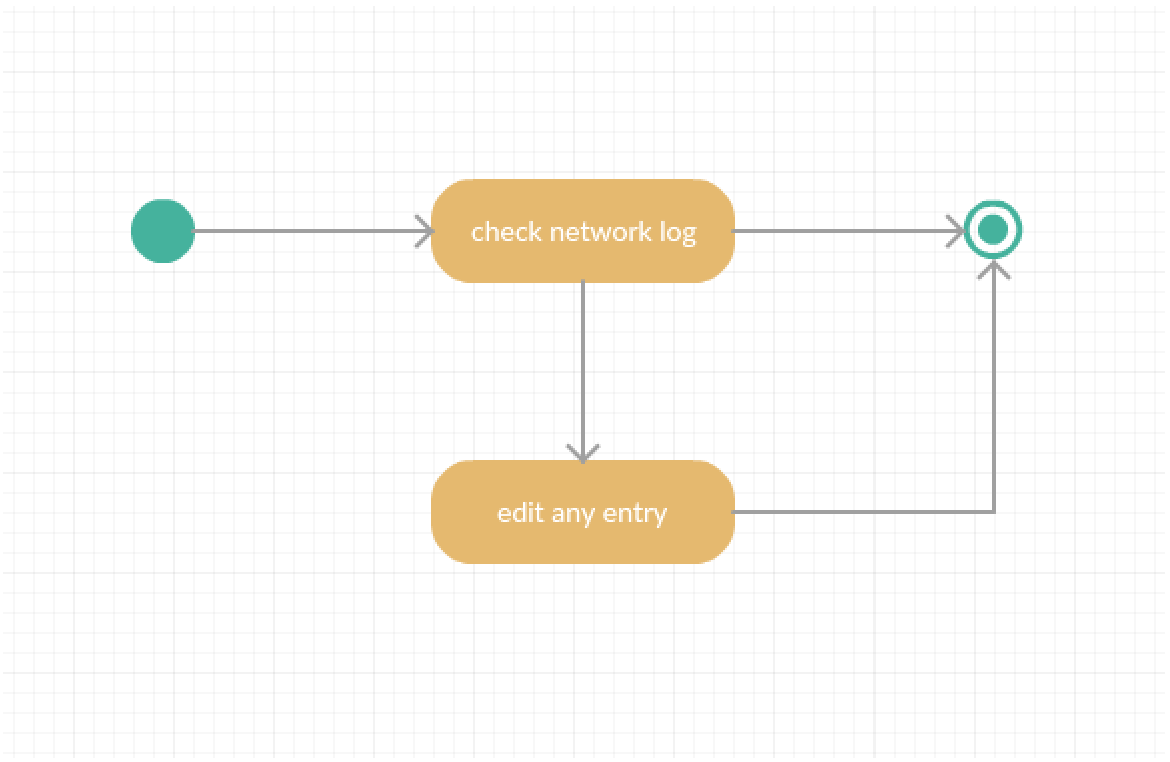

Fig 4.3

3. Add new node

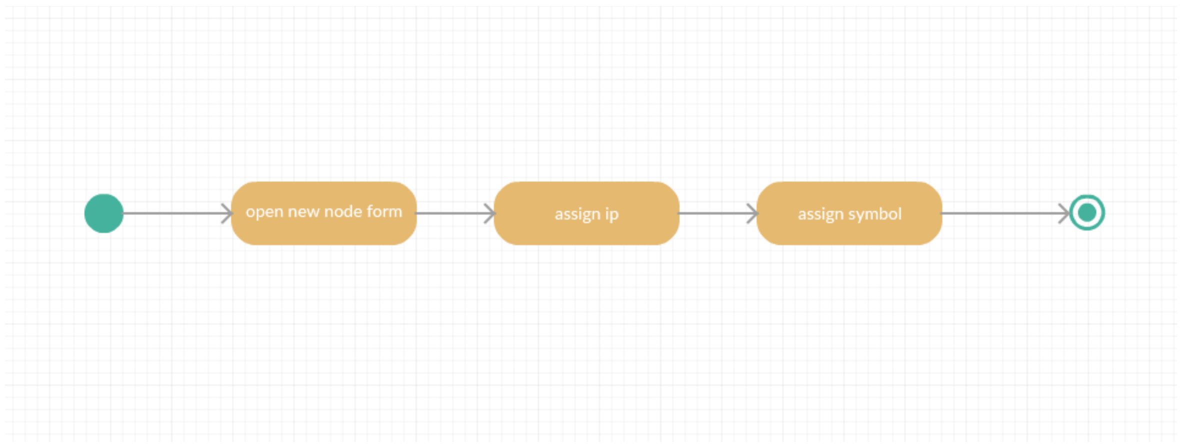

Fig 4.4

4. Delete node

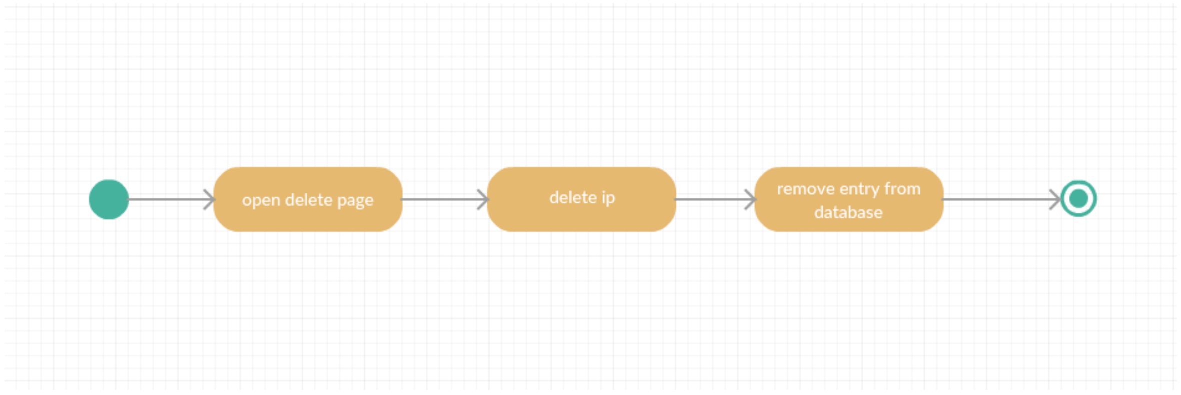

Fig 4.5

5. Scan network

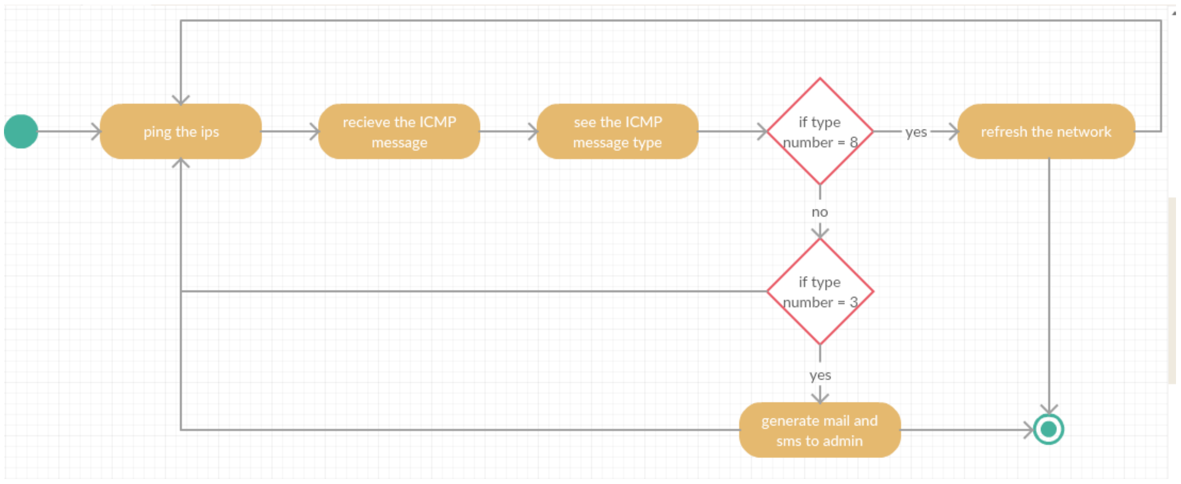

Fig 4.6

6. Notify admin

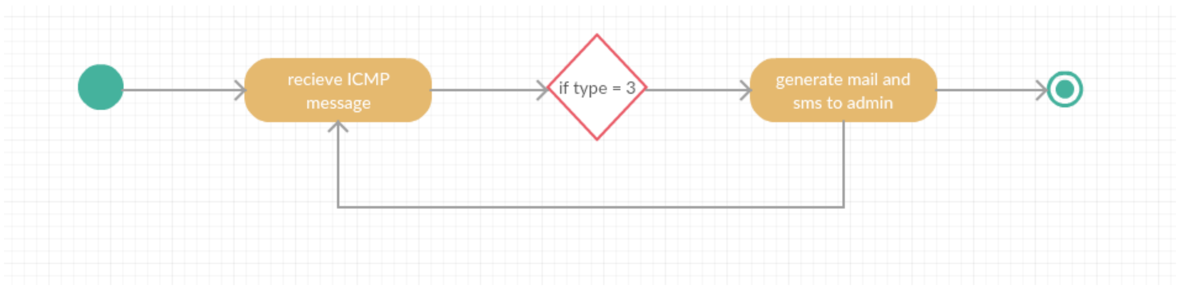

Fig 4.7

### 3.3 Feasibility analysis

- Technical feasibility :

Technical feasibility access the current resources and technology, which are required to accomplish user requirements in the software within the allocation time and for this, the software development team ascertains whether the current resources and technology can be upgraded or added in the software to accomplish specified user requirements.

Technical feasibility of the product has been studied under following heads :

Hardware availability :- The application needs a server, camera and computer system(PC).
Software availability :- Brackets, My-SQL.

- Economic Feasibility :

This system requires server, camera for surveillance and computer system.
(i) Hardware cost :- As such no separate hardware is required. We need a server and cameras.
(ii) Software cost :- Software needed for development of this system are open source software( Front hand : Html ,Java script and CSS ; Back hand : C++ and SQL,PL-SQL ) hence, there is no cost associated with them.

- Operational Feasibility :

Operational feasibility is an measure of how well a proposed system solves the problem. The existing systems have a problem that they are based on SNMP protocol which is not quite secured and also their header size is too large. So the proposed system removes these problem as:-

Here we are using Active Probing to solve the issues involved in previous systems. This proposal is secured such that It does not require any data fetching from the tables. It has nothing to do with Header size problem because it uses ping command, so it will not create any data congestion.

In current system, there is no provision for detect to notify the Admin about the down links. But in our proposed system, if any link is down so it will notify the admin about down links.

- Schedule Feasibility :

A project will fail if it takes too long to be completed before it is useful. Typically this means estimating how long the system will take to develop and if it can be completed in a given time period. Schedule feasibility is a measure of how reasonable the project timetable is.

The system is scheduled accordingly to the needs of the project. It is being divided into two phase I and II.

(i) Phase I - This phase needs completeion till design phase as per the predefined schedule, Phase- I is completed as per the expected schedule.

(ii) Phase II - This system has independent front end and back end modules. All identified modules have been clearly defined, divided to team memebers and the construction has been completed successfully.