Bandwidth Monitoring

Unity Electrosystems, Coimbatore

PROJECT REPORT P-881

Submitted in partial fulfillment of the requirement for
the award of the degree

BACHELOR OF ENGINEERING - INFORMATION TECHNOLOGY
Bharathiar University, Coimbatore.

Submitted by:

Dileep.K	**(9927S0049)**
Nandakumar.M.B	**(9927S0060)**
Venkatesh.G	**(9927S0508)**

Guided by:

Miss P.Sudha B.E
Lecturer,
Kumaraguru College of Technology

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

KUMARAGURU COLLEGE OF TECHNOLOGY

COIMBATORE - 641006.

MARCH 2003

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

KUMARAGURU COLLEGE OF TECHNOLOGY

COIMBATORE - 641006.

BONAFIDE CERTIFICATE

This is to certify that the project report titled

"BANDWIDTH MONITORING"

is done by

Nandakumar.M.B	**(9927S0060)**
Venkatesh.G	**(9927S0047)**
Dileep.K	**(9927S0072)**

This report is submitted in partial fulfillment of the requirements for the award of the

Degree of Bachelor of Engineering in Information Technology

of the Bharathiar University.

.....
Head of the Department

.....
Internal Guide

Submitted for the University Examination held on 20-3-2003..

.....
Internal Examiner

.....
External Examiner

Unity Electro Systems

To whomsoever it may concern

This is to certify that the following students studying final year B.E Information Technology at Kumaraguru College Of Technology have completed their project "Bandwidth Monitoring" successfully in our concern.

Nandakumar.M.B	9927S0060
Dileep.K	9927S0049
Venkatesh.G	9927S0508

Yours Sincerely
For UNITY ELECTRO SYSTEMS Pvt Ltd

K.P.K.Swaminathan,
USP DIVISION

Acknowledgement

ACKNOWLEDGEMENT

First and foremost we praise and thank God, from the depth of our heart who has given us the unfailing source of strength and comfort and his blessing enabling us to complete the project work.

We are grateful to Dr. K.K.Padmanaban, Principal, Kumaraguru College of Technology for facilitating us to complete our project work.

We would like to express our heartful thanks to Dr. S.Thangasamy, Head of Department of Computer Science and Engineering and Mr.K.R.Baskaran, AP, Department of Information technology, Kumaraguru College of Technology for providing us guidance and inspiration and enabling us to complete the project work.

We express a special thanks to our guide Miss. P.Sudha, Lecturer, Department of Information technology for her excellent guidance and moral support enabling us to complete the project work.

I would particularly thank Mr. Bhaghath Singh & Mr. Sandeep of Unity Electrosystems, Peelamedu, Coimbatore for spending their valuable time and their timely assistance during my project work.

Never ever to be forgotten I thank Mr.Mahalingam and Mr.Swaminathan, Directors of Unity Electrosystems , Coimbatore who have given us all their valuable support and cooperation.

Last but not the least, my heartiest thanks to our beloved parents and friends for their kind cooperation and encouragement enabling us to complete the project work successfully.

Contents

CONTENTS

1. Introduction

1. INTRODUCTION

1.1. About The Company

Unity Electrosystems situated at Peelamedu in Coimbatore is a company which is providing internet facility to about thirty browsing centers in and around the city.The company has built a wireless network to connecting the main zones in the city.

From the tower there is a local area network connecting the nearby browsing centres.

The company also handles building up of virtual private networks and the subsequent maintenance of those.

Since the company's network comprises of wireless networking and local area networking together the available bandwidth has to be monitored continuously. Keeping in mind that customer satisfaction is the key to success of any concern we have developed this project, so that it facilitates the company to provide reliable and effective service to its' customers.

1.2. Current Status Of The Problem Taken Up

The Wireless ISP is mainly based on and working in behalf of the network and its' connections. Each and every call passes through this network, hence every second various calls pass through the network architecture. Since the network is an electronic setup some faults in the connections may occur.

Now about thirty clients are using the network under Unity Electro Systems. If any of the clients experiences a fault in the connection and in the same time the client doesn't know any further details about the fault , immediately the connection holder informs about the connection's disability to the mobile phone of the Managing Director of the company.

Like wise the fault of the network connection is notified by the MD of Unity Electrosystems, then he has to troubleshoot the faults according to the client's message.

- First and foremost is to check a particular connection.
- As a result of checking the fault is detected.
- According to the fault the recurring measures are being taken.
- After the network problem has been solved the circuit is connected to the whole network for its' general process.

1.3. Relevance And Importance Of The Package

In order to overcome the network authentication problem, a system which will tackle the problems on its' own was necessary. So we had proposed a system according to their needs.

We have proposed a system namely "BANDWITH MONITORING". The system is designed according to the needs of the UNITY ELECTRO SYSTEMS in order to solve the problem of bandwidth monitoring & authentication.

The proposed system is to automate the modules of:

- Sending SMS to the Managing Director's Mobile Phone when the connection is lost.
- After receiving the message the MD has further easy steps to solve the problem according to our proposed system.
- To identify where the fault has occurred the MD has to view the graphical representation of the network.
- In order to protect our system by unauthorized usage, an HTTP Authentication Scheme is used to restrict the access of the above utilities.

2. Proposed Approach

2. PROPOSED APPROACH

2.1. Hardware Specification

(Minimum Requirement)

Processor	:	Pentium I
Clock speed	:	533Mhz
Hard disk capacity	:	8 GB
Ram capacity	:	64 MB
Floppy drive	:	1.44 MB
CD-rom drive	:	10 X max
Monitor	:	15'' Digital color

A mobile phone and a data cable are also required.

2.1. Software Specification

Operating Systems	: Linux, Windows
Compilers	: GCC compiler, Turbo C compiler
Scripting Languages	: Shell scripting, php
Extra Softwares	:Gnokii (software modem)
Server	:Samba server

2.3. Introduction to Software

About Linux

Linux is a freely distributed implementation of a UNIX-like kernel, the low level core of an operating system. Linux is a 32-bit operating system that uses the smallest resources, without functionality. To be noted all the browsing centers under Unity Electrosystems have Linux installed in their LAN servers.

About Windows

Windows is also a 32-bit operating system which is very popular. Windows has a very good graphical user interface which facilitates even a novice user. Since graphics programming in Windows is very much feasible this operating system is made use of in our project.

Compiler specifications

GCC compiler under Linux is a very powerful compiler which is capable of compiling C codes effectively and it has a rich library source.

Turbo C compiler is also a powerful C compiler which can be used to compile graphics programs effectively.

About Gnokii

Gnokii is a software which acts as a modem and used to interface the mobile phone with computer which is connected to the serial port using a data cable. It is a freely available software which can be downloaded from the internet.

Samba server is a web server which has all types of facilities to provide authentication.

3. System Design

3. SYSTEM DESIGN

3.1. Modules Specification

- Automated SMS sending service.
- GUI design of the bandwidth.
- HTTP Authentication.

3.2. Explanation of the modules

SENDING SMS:

Unity Electrosystems is a wireless ISP that provides Internet service to many browsing centers. There are many connections running through this concern. Suppose if any problem like loss of connection occurs the client has to send a message to the Managing Director's mobile about the fault. For making the work easier for both the clients and the concern the SMS sending module is used. If any problem occurs like loss of connection, now by using this module the client need not send a message to the Managing Director's mobile manually, our SMS module will automatically send a SMS. This will be more helpful for both the concern and the client.

GUI DESIGN :

Once the message is received by the concern they have to rectify the problem. For this they will look the entire connection of the client for the purpose of detecting the fault. Evidently this is a very time consuming and tedious process. Using this module if any problem occurs, it will show the status of the bandwidth.

HTTP Authentication:

An HTTP Authentication scheme is used to restrict the access of an unauthorized person.

4. Implementation

4. IMPLEMENTATION

4.1. Introduction

Implementation is the stage where the theoretical design is turned into a working system. The most crucial stage is achieved a new successful system. It can be implemented only after through testing is done and if it is found to work according to specifications.

The system personnel check the feasibility of the system. It involves careful planning, investigation of the current system and its constraints on implementations design of methods to achieve the changeover and evaluation of changeover methods apart from planning.

The major tasks involved in implementation are

-Computer system testing

-Training the user personal

-Full system testing and making necessary changes

-File conversion

-Change over

-Maintenance

These are the main steps involved in system implementations.

4.2. Module Description

The three modules quoted above are implemented as follows

4.2.1. SMS Module

This module is written in C language in vi editor of Linux. The program is compiled using GCC compiler, a powerful compiler found in the Linux platform. The program mainly uses the function system( ).

This module involves the interfacing of the mobile phone with the computer. The mobile phone is connected to the computer through the serial port using a data cable.

Here no physical modem is used to connect the mobile phone with the computer. A software called Gnokii, which is a free software found in the internet. Gnokii is actually a software modem which interfaces the mobile phone with the computer.

4.2.2. GUI Module

This module is completely built in C++ language which is compiled in Turbo C++. This module makes use of the system function 'ping' to check whether a particular link is available or not.

This module is based on graphics programming in C++. In order to get good quality graphical output we have used the VGA (Video Graphics Array) driver in a mode called VGAMED.

As presented already this module will check the individual connections and the status of the link is presented in a graphical form. Provisions for adding and deleting IP addresses is also provided for easy updating of the network.

Authentication module

This module is actually some form of configuration of Samba server which is a web server. The above two modules are those which handles confidential aspects of the company. So the access to those modules has to be restricted to particular users.

We have used the basic authentication or HTTP authentication which involves two steps. They are as follows...

i. Creation of .htaccess file in the specified directory

ii. Creation of .htpasswd file

The .htaccess file consists of the list of authenticated users and the type of authentication used.

The .htpasswd file is automatically created by the usage of `htpasswd' command. The syntax of htpasswd command is

#htpasswd username filename

The password will be encrypted and stored by the above command.

The prompt asking for login and the checking is done using PHP language. PHP is an abbreviation of Hypertext Preprocessor.

5. Testing

5. SYSTEM TESTING

The system testing the process of checking if the developed system is working according to the original objectives and requirements. Initially the system should be tested experimentally with the test data so as to ensure that software works according to specifications and in the way expect it to work. When it is found to be working in the right manner then the actual data is fed and the whole system is tested to check its performances.

All the testing needs to be conducted in accordance to the test conditions specified earlier. This ensures that the test coverage meets the requirements. And that testing is done in a systematic manner. Testing can be done by desk checking comparing the program flowcharts with the original specifications and by running the programs using text data.

Test data should be manually compiled and the results produced by the system compared with the appropriate clerical figures on the current computer system. It is also important that the system correctly identifies errors and testing for these is especially critical.

DIFFERENT LEVELS OF TESTING:

-Unit testing

-Integration testing

-Validation testing

-System testing

5.1.Unit Testing:

Unit testing focuses on verification effort on the smallest unit of the design module.

The unit test considerations are

-Interface errors

-Integrity of local data structures

-Boundary conditions

-Independent paths

-Error handling paths

5.2. Integration Testing:

Integration testing is a systematic technique for constructing the program structure. Each testing is done in separately. The bottom up approach was applied.

5.3. Validation Testing:

Validation test is carried about to verify whether the software function is a manner that is expected by the customer, Alpha validity is done.

Thus system testing is done before the system is implemented.

6. *Conclusion*

6. CONCLUSION

6.1 CONCLUSION:

The complete design and development of the system is presented in this dissertation. The system has user friendly features. It is possible for any user to use this system.

The programming techniques used in the design of the system provide a scope for further expansion and implementation of any changes, which may occur in the future. The system has been tested by connecting with many systems and they provide satisfactory performance.

This system is developed with the specifications and abiding by the existing rules and regulations of the company.

Since the requirements of any organization and their standards are changing day to day the system has been designed in such a way that its scope and boundaries could be expanded in future with little modifications. As a further enhancement this system can be integrated with any other system

6.2. Limitations

- The SMS module requires a mobile phone to be connected to the computer always
- The continuous pinging of the network increase a small amount of traffic due to ICMP packets

6.3. Future Enhancements

- Automating the adding and deleting of nodes can be brought into existence.
- Building the GUI module to be compatible with Linux platform.

7. References

7.References

1. Yashwant Kanitkar, "Projects in 'C'", Tata McGraw Hill Publications, 2nd edition 1999
2. Yashwant Kanitkar, "Graphics under 'C'", Tata McGraw Hill Publications, 3rd edition 2002
3. Nance, "Network Programming in 'C'", Tata McGraw Hill Publications, 2nd edition 1999
4. Yahwant Kanitkar, "Let us C++", Tata McGraw Hill Publications, 2nd edition 2000
5. www.gnokii.org (14-2-03)
6. www.nixcraft.linux.com (20-2-03)
7. www.linux.com (4-2-03)

8. Appendix

8.1. Source code

Main Program

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
#include<process.h>
#include<string.h>
int Mainmenu();
int getaction(int sel);
void Mdisplay(int);
void Sdisplay(int ,int,int, char *str);
void add();
int Net();
void deleteip();
int Netdelete();
void view();
int Netview();
void box(int x1,int y1,int x2,int y2,char *message="Message",int bcolor=0,int tcolor=2);
void check();
void about();
void help();

char Mmenu[10][30]={" Ping "," list "," Help "};
int Ml[10][10]={ {5,4},{15,4},{27,4} };
int Nmain=3;
int Msel=-1;
int NSmenu[]={4,3,2};
char *Smenu1[10]={" Check "," View "," Ping NET"," Exit "};
char *Smenu2[10]={" Add "," Delete "," IP view "};
char *Smenu3[10]={" About Us "," Help "};
void cls()
{
                                        textbackground(0);
                                        clrscr();
                                        box(1,1,80,24,"NETWORK FAULT
DETECTION",0,15);
                                       for(int i=2;i<80;++i)
                                       {
                                               gotoxy(i,5);
                                               cprintf("%c",'Í');
                                       }
```

```
                                        window(2,4,79,4);
                                        textbackground(3);
                                        clrscr();
                                        window(1,1,80,25);
                                        Mdisplay(-1);
}
void main()
{
                                        int gd =VGA,gm=1;
                                        initgraph(&gd,&gm,"");
                                        closegraph();
                                        while(Mainmenu());
}
int Mainmenu()
{
                                         char ch;
//      int sel=0;
                                         while(1)
                                         {
                                        cls();
                                        _setcursortype(_NOCURSOR);
                                        Mdisplay(Msel);
                                        ch=getch();
                                        if(ch== 75)Msel--;
                                        if(ch== 77)Msel++;
                                        if(ch==13||ch==80)
                                        {
                                                int a=getaction(Msel);
                                                if(a==1)
                                                {
                                                        Mdisplay(-1);
                                                        break;
                                                }
                                        }

                                        if(Msel<0)Msel=0;
                                        if(Msel>=Nmain)Msel=Nmain-1;
                                         }

                                          return 0;
}
int getaction(int sel)
{
                                          char ch;
                                          int s=0;

                                        if(sel==0)
```

```
                while(1)
                {
                for(int i=0;i<NSmenu[sel];++i)
                 {
                        if(i==s)
                                Sdisplay(0,sel,i,Smenu1[i]);
                        else
                                Sdisplay(1,sel,i,Smenu1[i]);
                 }

                ch=getch();
                if(ch== 72)s--;
                if(ch== 80)s++;
                if(ch==77){Msel++; return 0;};
                if(ch==75){Msel--; return 0;};
                if(ch==27) return 0;
                if(ch==13)
                   {
                                cls();
                                if(s==0)
                                {
                                        check();
                                }
                                else if(s==1)
                                {

system("netping.exe");
                                }
                                else if(s==2)
                                {
                                        system("packet.exe");
                                }
                                else
                                   exit(0);
                          return 0;
                        }
                if(s<0)s=0;
                if(s>=NSmenu[sel])s=NSmenu[sel]-1;
                 }
}
else if(sel==1)
{
                 while(1)
                {
                for(int i=0;i<NSmenu[sel];++i)
                 {
                        if(i==s)
```

```
                else
                        Sdisplay(1,sel,i,Smenu2[i]);
        }

        ch=getch();
        if(ch== 72)s--;
        if(ch== 80)s++;
        if(ch==77){Msel++; return 0;};
        if(ch==75){Msel--; return 0;};
        if(ch==27) return 0;
        if(ch==13)
          {
                cls();
                if(s==0)
                {
                        add();
                }
                else if(s==1)
                {
                        deleteip();
                }
                else
                {
                        Netview();
                }
                return 0;
          }
        if(s<0)s=0;
        if(s>=NSmenu[sel])s=NSmenu[sel]-1;
        }


}
else
{
        while(1)
        {
        for(int i=0;i<NSmenu[sel];++i)
        {
                if(i==s)
                        Sdisplay(0,sel,i,Smenu3[i]);
                else
                        Sdisplay(1,sel,i,Smenu3[i]);
        }

        ch=getch();
        if(ch== 72)s--;
```

```
                    if(ch==77){Msel++; return 0;};
                    if(ch==75){Msel--; return 0;};
                    if(ch==27) return 0;
                    if(ch==13)
                      {
                                cls();
                                if(s==0)
                                {
                                        about();
                                }
                                else if(s==1)
                                {
                                        help();
                                }
                                return 0;
                        }
                    if(s<0)s=0;
                    if(s>=NSmenu[sel])s=NSmenu[sel]-1;
                     }
                }
}
void Mdisplay(int Msel)
{
                for(int i=0;i<Nmain;++i)
                {
                 if(Msel==i)
                 {
                        textbackground(1);
                        textcolor(15);
                        gotoxy(Ml[i][0],Ml[i][1]);
                        cprintf("%s",Mmenu[i]);
                 }
                 else
                 {
                        textbackground(3);
                        textcolor(4);
                        gotoxy(Ml[i][0],Ml[i][1]);
                        cprintf("%s",Mmenu[i]);
                 }
                }
}
void Sdisplay(int c,int sel,int pos,char *str)
{
                 if(c==0)
                 {
                        textbackground(6);
                        textcolor(15);
```

```
                cprintf("%s",str);
            }
            else
            {
                textbackground(3);
                textcolor(4);
                gotoxy(Ml[sel][0],Ml[sel][1]+1+pos);
                cprintf("%s",str);
            }
}

void add()
{
    char IP[30];
    box(15,7,65,22,"ADDING",1,15);
    for(int j=16;j<65;++j)
    {
        gotoxy(j,20);
        cprintf("%c",'Í');
    }
    window(18,21,40,21);
    textbackground(1);
    clrscr();
    window(1,1,80,25);
    gotoxy(18,21);
    cprintf("Enter the IP adress ");
    window(40,21,59,21);
    textbackground(0);
    clrscr();
    window(1,1,80,25);
    int l=Net();
    if(l==99)return;
    gotoxy(41,21);
    _setcursortype(_NORMALCURSOR);
    scanf("%s",IP);
    _setcursortype(_NOCURSOR);
    FILE *f;
    f=fopen("IP_VIEW.TXT","a");
    if(f==NULL)
    {
        printf("file Not open ..");
        exit(0);
    }
    fprintf(f,"\n%s %d",IP,l);
    fclose(f);
}
```

```
{
                                        int x1,x2,y1,y2,status;
                                        char IP_ad[30][30];
                                        int type;
                                        int count =0;
                                        int sub[30];
                                        FILE *f;
                                        f=fopen("IP_ADD.TXT","r");
                                        if(f==NULL)
                                        {
                                                printf("file Not open ..");
                                                exit(0);
                                        }

                                        while(fscanf(f,"%d %d %d %d %s %d %d
%d",&x1,&y1,&x2,&y2,IP_ad[count],&status,&type,&sub[count])==8)
                                        {
                                                 count++;
                                        }
                                        int pointer=0;
                                        int selp=0;
                                        char ch;
                                        while(1)
                                        {
                                        //      cls();
                                          window(25,11,55,17);
                                          textbackground(0);
                                          clrscr();
                                          window(1,1,79,25);
                                          for(int i=pointer;i<(pointer+4);++i)
                                          {
                                           if(pointer+selp==i)
                                           {
                                                textbackground(6);
                                                textcolor(15);
                                                gotoxy(35,12+i-pointer);
                                                cprintf("%s",IP_ad[i]);
                                           }
                                           else
                                           {
                                                textbackground(0);
                                                textcolor(4);
                                                gotoxy(35,12+i-pointer);
                                                cprintf("%s",IP_ad[i]);
                                           }
                                          }
```

```
                if(ch==27)return 99;
                if(ch==72)
                {
                        selp--;
                        if(selp<0)
                        {
                                selp=0;
                                pointer--;
                                if(pointer<0)pointer=0;
                        }
                }
                if(ch==80)
                {
                        selp++;
                        if(selp>3)
                        {
                                selp=3;
                                pointer++;
                                if(pointer>(count-
5))pointer=count-5;

                        }
                }
                if(ch==13)return sub[pointer+selp];
        }


}
void deleteip()
{
   while(1)
   {
                int l=Netdelete();
                if(l==99)return;
                char IP[30][30];
                int sub[30];
                int count=0;
                FILE *f;
                f=fopen("IP_VIEW.TXT","r");
                if(f==NULL)
                {
                        printf("file Not open ..");
                        exit(0);
                }
                while(fscanf(f,"%s
%d",IP[count],&sub[count])==2)
                {
                        if(sub[count]==l)
                                count++;
```

```
fclose(f);
int pointer=0;
int selp=0;
char ch;
int nodis=4;
if(count<4)nodis=count;
box(25,11,55,20,"SELECT THE IP TO BE
DELETED",0,15);

while(1)
{
//      cls();
window(26,14,54,19);
textbackground(0);
clrscr();
window(1,1,80,25);

 for(int i=pointer;i<(pointer+nodis);++i)
 {
  if(pointer+selp==i)
  {
        textbackground(6);
        textcolor(15);
        gotoxy(35,15+i-pointer);
        cprintf("%s",IP[i]);
  }
  else
  {
        textbackground(0);
        textcolor(4);
        gotoxy(35,15+i-pointer);
        cprintf("%s",IP[i]);
  }
 }

        ch=getch();
        if(ch==27)break;
        if(ch==72)
        {
                selp--;
                if(selp<0)
                {
                        selp=0;
                        pointer--;
                        if(pointer<0)pointer=0;
                }
        }
        if(ch==80)
```

```
                                        selp++;
                                        if(selp>3)
                                        {
                                                selp=3;
                                                pointer++;
                                                if(pointer>(count-
nodis))pointer=count-nodis;

                                        }
                                }
                                if(ch==13)
                                {
                                        FILE *f;
                                        f=fopen("IP_VIEW.TXT","w");
                                        if(f==NULL)
                                        {
                                                printf("file Not open ..");
                                                exit(0);
                                        }
                                        for(int i=0;i<count;++i)
                                        {

                                if(strcmp(IP[i],IP[pointer+selp])!=0)
                                                {
                                                        fprintf(f,"%s
%d\n",IP[i],sub[i]);

                                                }
                                        }
                                        fclose(f);
                                        break;
                                }
                        }

                        }

}
int Netdelete()
{
                        int x1,x2,y1,y2,status;
                        char IP_ad[30][30];
                        int type;
                        int count =0;
                        int sub[30];
                        FILE *f;
                        f=fopen("IP_ADD.TXT","r");
                        if(f==NULL)
                        {
                                printf("file Not open ..");
```

```
}

while(fscanf(f,"%d %d %d %d %s %d %d
%d",&x1,&y1,&x2,&y2,IP_ad[count],&status,&type,&sub[count])==8)
{
        count++;
}
int pointer=0;
int selp=0;
char ch;
box(15,7,65,22,"DELETING",1,15);
while(1)
{
//      cls();

window(25,11,55,20);
textbackground(0);
clrscr();
window(1,1,80,25);

 for(int i=pointer;i<(pointer+4);++i)
 {
  if(pointer+selp==i)
  {
        textbackground(6);
        textcolor(15);
        gotoxy(35,14+i-pointer);
        cprintf("%s",IP_ad[i]);
  }
  else
  {
        textbackground(0);
        textcolor(4);
        gotoxy(35,14+i-pointer);
        cprintf("%s",IP_ad[i]);
  }
 }

        ch=getch();
        if(ch==27)return 99;
        if(ch==72)
        {
                selp--;
                if(selp<0)
                {
                        selp=0;
                        pointer--;
```

```
                }
        }
        if(ch==80)
        {
                selp++;
                if(selp>3)
                {
                        selp=3;
                        pointer++;
                        if(pointer>(count-
5))pointer=count-5;

                }
        }
        if(ch==13)return sub[pointer+selp];

}
}
void view(int l)
{
        char IP[30][30];
        int sub[30];
        int count=0;
        FILE *f;
        f=fopen("IP_VIEW.TXT","r");
        if(f==NULL)
        {
                printf("file Not open ..");
                exit(0);
        }
        while(fscanf(f,"%s
%d",IP[count],&sub[count])==2)
        {
                if(sub[count]==l)
                        count++;
        }
        fclose(f);
        int pointer=0;
        int selp=0;
        char ch;
        int nodis=4;
        if(count<4)nodis=count;
//      while(1)
        {
        //      cls();

                window(44,12,63,20);
                textbackground(0);
```

```
                window(1,1,80,25);

 for(int i=pointer;i<(pointer+nodis);++i)
 {
  if(pointer+selp==i)
  {
        textbackground(1);
        textcolor(15);
        gotoxy(47,14+i-pointer);
        cprintf("%s",IP[i]);
 }
  else
  {
        textbackground(3);
        textcolor(4);
        gotoxy(47,14+i-pointer);
        cprintf("%s",IP[i]);
  }
 }

//                              ch=getch();
                                if(ch==72)
                                {
                                        selp--;
                                        if(selp<0)
                                        {
                                                selp=0;
                                                pointer--;
                                                if(pointer<0)pointer=0;
                                        }
                                }
                                if(ch==80)
                                {
                                        selp++;
                                        if(selp>3)
                                        {
                                                selp=3;
                                                pointer++;
                                                if(pointer>(count-
nodis))pointer=count-nodis;

                                        }
                                }

                        }

}
int Netview()
```

```
int x1,x2,y1,y2,status;
char IP_ad[30][30];
int type;
int count =0;
int sub[30];
FILE *f;
f=fopen("IP_ADD.TXT","r");
if(f==NULL)
{
        printf("file Not open ..");
        exit(0);
}

while(fscanf(f,"%d %d %d %d %s %d %d
%d",&x1,&y1,&x2,&y2,IP_ad[count],&status,&type,&sub[count])==8)
{
         count++;
}
int pointer=0;
int selp=0;
char ch;
box(15,7,65,22,"VIEW",1,15);
textcolor(15);
textbackground(1);
gotoxy(19,10);
cprintf("   TOWER");
gotoxy(45,10);
cprintf("   IP ADDRESS");
for(int i=10;i<22;++i)
{
        gotoxy(40,i);
        cprintf("%c",'°');
}

while(1)
{
//      cls();
 window(17,12,35,20);
 textbackground(0);
 clrscr();
 window(1,1,80,25);
 for(int i=pointer;i<(pointer+4);++i)
 {
  if(pointer+selp==i)
  {
        textbackground(1);
        textcolor(15);
```

```
                cprintf("%s",IP_ad[i]);
            }
            else
            {
                textbackground(3);
                textcolor(4);
                gotoxy(20,14+i-pointer);
                cprintf("%s",IP_ad[i]);
            }
        }

            ch=getch();
            if(ch==27)return 0;
            if(ch==72)
            {
                selp--;
                if(selp<0)
                {
                    selp=0;
                    pointer--;
                    if(pointer<0)pointer=0;
                }
            }
            if(ch==80)
            {
                selp++;
                if(selp>3)
                {
                    selp=3;
                    pointer++;
                    if(pointer>(count-
5))pointer=count-5;

                }
            }
             view(sub[pointer+selp]);
        }
}
void box(int x1,int y1,int x2,int y2,char *message,int bcolor,int tcolor)
{
        window(x1,y1,x2,y2);
        textbackground(bcolor);
        clrscr();
        window(1,1,80,25);
        textcolor(tcolor);
        for(int i=x1;i<x2;++i)
        {
            gotoxy(i,y1);
```

```
                gotoxy(i,y1+2);
                cprintf("%c",'Í');
                gotoxy(i,y2);
                cprintf("%c",'Í');
        }
        for(i=y1;i<y2;++i)
        {
                gotoxy(x1,i);
                cprintf("%c",'º');
                gotoxy(x2,i);
                cprintf("%c",'º');
        }
                gotoxy(x1,y1);
                cprintf("%c",'É');
                gotoxy(x1,y2);
                cprintf("%c",'È');
                gotoxy(x2,y1);
                cprintf("%c",'»');
                gotoxy(x2,y2);
                cprintf("%c",'¼');
                gotoxy(x1+(x2-x1)/2-
(strlen(message)/2),y1+1);

                cprintf("%s",message);

}

void check()
{
        box(10,6,70,23,"CHECKING THE IP",1,15);
        window(12,11,68,22);
        textbackground(0);
        clrscr();
        window(1,1,80,25);
        gotoxy(17,9);
        textbackground(1);
        textcolor(15);
        cprintf("%s","Enter the IP address");
        window(1,1,80,25);
        window(40,9,60,9);
        textbackground(0);
        clrscr();
        char IP[30];
        char ping[40]="ping ";
  _setcursortype(_NORMALCURSOR);
        scanf("%s",IP);
        _setcursortype(_NOCURSOR);
        strcat(ping,IP);
        strcat(ping,">>temp.txt");
```

```
                                        system(ping);
                                        window(12,11,68,22);
                                        FILE *f;
                                        f=fopen("temp.txt","r");
                                        if(f==NULL)
                                        {
                                                printf("File not open...");
                                                exit(0);
                                        }
                                        int slash=0;
                                        char ch=fgetc(f);
                                        while(ch!=EOF)
                                        {
                                                if(ch=='\n' && slash==0)
                                                {
                                                        cprintf("\n\r");
                                                        slash=1;
                                                }
                                                else
                                                {
                                                        cprintf("%c",ch);
                                                        slash=0;
                                                 }
                                                ch=fgetc(f);
                                        }
                                        fclose(f);
                                        window(1,1,80,25);
                                        getch();
}
void about()
{
                                        box(20,10,60,20,"ABOUT US",1,15);
                                        gotoxy(35,14);
                                        cprintf("%s","Developed by ");
                                        textcolor(6);
                                        gotoxy(35,16);
                                        cprintf("%s","M.B.Nandakumar");
                                        gotoxy(35,17);
                                        cprintf("%s","G.Venkatesh");
                                        gotoxy(35,18);
                                        cprintf("%s","K.Dileep");
                                        getch();
}
void help()
{
                                        box(10,10,75,20,"HELP",1,15);
                                        gotoxy(15,14);
```

```
                                        cprintf("%s","Select the following menu items to
examine the connections");
                                        textcolor(6);
                                        gotoxy(20,16);
                                        cprintf("%s","* `Check' for a particular IP");
                                        gotoxy(20,17);
                                        cprintf("%s","* `View' for all the towers");
                                        gotoxy(20,18);
                                        cprintf("%s","* `Netping' for all IP's under a
tower");
                                        gotoxy(20,19);
                                        cprintf("%s","* Use `add' ,`delete' for the same.");
                                        getch();
}
```

Program for pinging the towers

```
#include<stdio.h>
#include<conio.h>
#include<process.h>
#include<dos.h>
#include<graphics.h>
#include<string.h>
#include<stdlib.h>
int cnts=0;
class Tower
{
public:
                                        int Tx;
                                        int Ty;
```

```
                                        int Ny;
                                        char IP_address[20];
                                        int Status;
                                        int Type;
                                        Tower(int,int,int,int,char str[],int,int);
                                        void Tower_Icon();
                                        void Display();
                                        void Ping();
                                        int GetData(FILE*,char*);
};
void main()
{
                                        int gd=VGA,gm=VGAMED;
                                        FILE *f;
                                        f=fopen("IP_ADD.TXT","r");
                                        if(f==NULL)
                                        {
                                                printf("file Not open ..");
                                                exit(0);
                                        }
                                        int x1,x2,y1,y2,status;
                                        char IP_add[30];
                                        int type;
                                        Tower *T[30];
                                        int count =0;
                                        int sub;
                                        while(fscanf(f,"%d %d %d %d %s %d %d
%d",&x1,&y1,&x2,&y2,IP_add,&status,&type,&sub)==8)
                                        {
                                                T[count]=new
Tower(x1,y1,x2,y2,IP_add,status,type);
                                                count++;
                                        }
  while(1)
  {
                                        cnts=0;
                                        textbackground(0);
                                        clrscr();
                                        textcolor(15);
                                        printf("\n\n\n");
                                        cprintf("          Press ESC to Quit ");
                                        printf("\n");
                                        cprintf("          press any key to abort at this
level");
                                        printf("\n");
                                        for(int i=0;i<count;++i)
                                        {
```

```
                cnts++;
                if(kbhit())
                {
                 char ch=getch();
                 if(ch==27)
                 {
                        closegraph();
                        return;
                 }
                 else
                        break;
                }
        }

        initgraph(&gd,&gm,"egavga.bgi");
        rectangle(0,0,639,349);
        rectangle(10,7,629,343);
        setfillstyle(9,1);
        floodfill(5,5,15);
        settextstyle(4,0,7);
        setcolor(2);
        outtextxy(200,15,"NETWORK FAULT
DETECTION");

        settextstyle(4,0,0);
        for(i=0;i<cnts;++i)
        {
                T[i]->Display();
        }
        char ch=getch();
        if(ch==27)
        {
                closegraph();
                return;
        }
        closegraph();
  }
}

void Tower::Tower(int x1,int y1,int x2,int y2,char str[],int status,int type)
        {
                Tx=x1;
                Ty=y1;
                Nx=x2;
                Ny=y2;
                strcpy(IP_address,str);
                Status=status;
                Type=type;
```

```
void Tower::Display()
                                        {
                                                setcolor(7);
                                                Tower_Icon();
                                                rectangle(Nx-52,Ny+5,Nx+65,Ny+15);
                                                outtextxy(Nx-51,Ny+7,IP_address);

                                                if(Type)
                                                {
                                                        setcolor(2*(1-Status)+4*(1-
Status+1));
                                                        line(Tx,Ty+15,Nx,Ny-15);
                                                }

                                        }
void Tower::Tower_Icon()
                                        {
                                                if(Type==0||Type==1)
                                                {
                                                  int p[]={0,-5, -5,5, 5,5, 0,-5};
                                                  for(int i=0;i<8;i+=2)
                                                  {
                                                        p[i]+=Nx;
                                                        p[i+1]+=Ny-5;
                                                  }
                                                  setcolor(15);
                                                  drawpoly(4,p);
                                                  setfillstyle(9,2);
                                                  floodfill(p[0],p[1]+5,15);
                                                  ellipse(p[0],p[1]-5,0,360,10,3);
                                                  ellipse(p[0],p[1]-4,180,360,10,5);
                                                  setfillstyle(SOLID_FILL,4);
                                                  floodfill(p[0],p[1]-1,15);
                                                }
                                                else if(Type==2)
                                                {
                                                 line(Nx-30,Ny,Nx+35,Ny);
                                                 line(Nx,Ny-15,Nx,Ny);
                                                 setfillstyle(9,2);
                                                 bar3d(Nx-20,Ny-5,Nx+20,Ny+5,5,1);
                                                }
                                                else if(Type==3)
                                                {
                                                  setfillstyle(9,4);
                                                  bar3d(Nx-25,Ny-9,Nx+20,Ny,5,1);
                                                  setfillstyle(9,2);
```

```
                }
        }
void Tower::Ping()
{
        system("del temp.txt");
        char IP[40]="ping " ;
        strcat(IP,IP_address);
        strcat(IP," >> temp.txt");
        textcolor(6);
        cprintf("\n\r           Checking :: %s",IP_address);
        system(IP);

        FILE *f;
        f=fopen("temp.txt","r");
        if(f==NULL)
        {
                printf("File Not open for reading ..");
                exit(0);
        }
        char str[40];
        while(GetData(f,str)==1)
        {
                if(strcmp(str,"Lost")==0)
                {
                        if(GetData(f,str)==0)return;
                        if(GetData(f,str)==0)return;
                        Status=!!atoi(str);
                        if(Status==0)
                        {
                                textcolor(2);
                                gotoxy(40,7+cnts);
                                cprintf(" [ OK ]");
                        }
                        else
                        {
                                textcolor(4);
                                gotoxy(40,7+cnts);
                                cprintf(" [FAILED]");
                        }
                        break;
                }
        }
        fclose(f);
}

int Tower::GetData(FILE* f,char* str)
{
```

```
                {
                        return 1;
                }
                else
                {
                        return 0;
                }
}
```

Program for pinging the netzones

```
#include<stdio.h>
#include<conio.h>
#include<process.h>
#include<dos.h>
#include<graphics.h>
#include<string.h>
#include<stdlib.h>
void box(int x1,int y1,int x2,int y2,char *message,int bcolor,int tcolor);
void view(int l);
int Netview();

int x[]={70,550,120,500,170,450,220,400,270,90,34,342,323,233,232,344,4545};
int y[]={100,100,140,140,180,180,220,220,260,90,34,342,323,233,232,344,4545};
char IP[30][30];
int cnt=0;
int cnts=0;
char NETIP[30];
class Tower
{
```

```
                                        int Tx;
                                        int Ty;
                                        int Nx;
                                        int Ny;
                                        char IP_address[20];
                                        int Status;
                                        int Type;
                                        Tower(int,int,int,int,char str[],int,int);
                                        void Tower_Icon();
                                        void Display();
                                        void Ping();
                                        int GetData(FILE*,char*);
                                        void setdata(int x1,int y1,int x2,int y2,char str[],int
status,int type)
                                        {
                                                Tx=x1;
                                                Ty=y1;
                                                Nx=x2;
                                                Ny=y2;
                                                strcpy(IP_address,str);
                                                Status=status;
                                                Type=type;

                                        }
};
void main()
{
                                        Tower *T[30];
                                        int s=Netview();
                                        if(s==99)return;
                                        T[0]=new Tower(320,60,300,60,NETIP,0,0);
                                        for(int i=0;i<cnt;++i)
                                        {
                                                T[i+1]=new
Tower(300,60,x[i],y[i],IP[i],0,3);

                                        }
                                        int gd=VGA,gm=VGAMED;
  while(1)
  {
                                        cnts=0;
                                         textbackground(0);
                                         clrscr();
                                         textcolor(15);
                                         printf("\n\n\n");
                                         cprintf("               Press ESC to quit ");
                                         printf("\n");
                                         cprintf("               Press Any key to abort at this IP
```

```
                                        printf("\n");
                                        for(int i=0;i<=cnt;++i)
                                        {
                                                T[i]->Ping();
                                                cnts++;
                                                if(kbhit())
                                                {
                                                 char ch=getch();
                                                 if(ch==27)
                                                 {
                                                        closegraph();
                                                        return;
                                                 }
                                                 else
                                                 break;
                                                }
                                        }

                                        initgraph(&gd,&gm,"egavga.bgi");
                                        rectangle(0,0,639,349);
                                        rectangle(10,7,629,343);
                                        setfillstyle(9,1);
                                        floodfill(5,5,15);
                                        settextstyle(4,0,7);
                                        setcolor(2);
                                        outtextxy(200,15,"NETWORK FAULT
DETECTION");

                                        settextstyle(4,0,0);
                                        for(i=0;i<=cnts;++i)
                                        {
                                                T[i]->Display();
                                        }
                                        char ch=getch();
                                        if(ch==27)
                                        {
                                                closegraph();
                                                return;
                                        }
                                        closegraph();
  }
}

void Tower::Tower(int x1,int y1,int x2,int y2,char str[],int status,int type)
                                        {
                                                Tx=x1;
                                                Ty=y1;
                                                Nx=x2;
```

```
                strcpy(IP_address,str);
                Status=status;
                Type=type;
        }
void Tower::Display()
        {
                setcolor(7);
                Tower_Icon();
                rectangle(Nx-52,Ny+5,Nx+65,Ny+15);
                outtextxy(Nx-51,Ny+7,IP_address);

                if(Type)
                {
                        setcolor(2*(1-Status)+4*(1-
Status+1));
                        line(Tx,Ty+15,Nx,Ny-15);
                }

        }
void Tower::Tower_Icon()
        {
                if(Type==0||Type==1)
                {
                  int p[]={0,-5, -5,5, 5,5, 0,-5};
                  for(int i=0;i<8;i+=2)
                  {
                        p[i]+=Nx;
                        p[i+1]+=Ny-5;
                  }
                  setcolor(15);
                  drawpoly(4,p);
                  setfillstyle(9,2);
                  floodfill(p[0],p[1]+5,15);
                  ellipse(p[0],p[1]-5,0,360,10,3);
                  ellipse(p[0],p[1]-4,180,360,10,5);
                  setfillstyle(SOLID_FILL,4);
                  floodfill(p[0],p[1]-1,15);
                }
                else if(Type==2)
                {
                 line(Nx-30,Ny,Nx+35,Ny);
                 line(Nx,Ny-15,Nx,Ny);
                 setfillstyle(9,2);
                 bar3d(Nx-20,Ny-5,Nx+20,Ny+5,5,1);
                }
                else if(Type==3)
```

```
                setfillstyle(9,4);
                bar3d(Nx-25,Ny-9,Nx+20,Ny,5,1);
                setfillstyle(9,2);
                bar3d(Nx-10,Ny-20,Nx+10,Ny-10,5,1);
            }
        }

void Tower::Ping()
{
        system("del temp.txt");
        char IP[40]="ping " ;
        strcat(IP,IP_address);
        strcat(IP," >> temp.txt");
        textcolor(6);
        cprintf("\n\r                Checking

:=%s",IP_address);
//      cprintf("\n\r       Checking :=%s",IP);
        system(IP);

        FILE *f;
        f=fopen("temp.txt","r");
        if(f==NULL)
        {
                printf("File Not open for reading ..");
                exit(0);
        }
        char str[40];
        while(GetData(f,str)==1)
        {
                if(strcmp(str,"Lost")==0)
                {
                        if(GetData(f,str)==0)return;
                        if(GetData(f,str)==0)return;
                        Status=!!atoi(str);
                        if(Status==0)
                        {
                                textcolor(2);
                                gotoxy(50,7+cnts);
                                cprintf(" [  OK  ]");
                        }
                        else
                        {
                                textcolor(4);
                                gotoxy(50,7+cnts);
                                cprintf(" [FAILED]");
                        }
                        break;
                }
```

```
                                        fclose(f);
}

int Tower::GetData(FILE* f,char* str)
{
                                        if(fscanf(f,"%s",str)==1)
                                        {
                                                return 1;
                                        }
                                        else
                                        {
                                                return 0;
                                        }
}
void box(int x1,int y1,int x2,int y2,char *message,int bcolor,int tcolor)
{
                                        window(x1,y1,x2,y2);
                                        textbackground(bcolor);
                                        clrscr();
                                        window(1,1,80,25);
                                        textcolor(tcolor);
                                        for(int i=x1;i<x2;++i)
                                        {
                                                gotoxy(i,y1);
                                                cprintf("%c",'Í');
                                                gotoxy(i,y1+2);
                                                cprintf("%c",'Í');
                                                gotoxy(i,y2);
                                                cprintf("%c",'Í');
                                        }
                                        for(i=y1;i<y2;++i)
                                        {
                                                gotoxy(x1,i);
                                                cprintf("%c",'º');
                                                gotoxy(x2,i);
                                                cprintf("%c",'º');
                                        }
                                                gotoxy(x1,y1);
                                                cprintf("%c",'É');
                                                gotoxy(x1,y2);
                                                cprintf("%c",'È');
                                                gotoxy(x2,y1);
                                                cprintf("%c",'»');
                                                gotoxy(x2,y2);
                                                cprintf("%c",'¼');
                                                gotoxy(x1+(x2-x1)/2-
(strlen(message)/2),y1+1);
```

```
}
void view(int l)
{
                                                int sub[30];
                                                FILE *f;
                                                f=fopen("IP_VIEW.TXT","r");
                                                if(f==NULL)
                                                {
                                                        printf("file Not open ..");
                                                        exit(0);
                                                }
                                                cnt=0;
                                                while(fscanf(f,"%s
%d",IP[cnt],&sub[cnt])==2)
                                                {
                                                        if(sub[cnt]==l)
                                                                cnt++;
                                                }
                                                fclose(f);
                                                int pointer=0;
                                                int selp=0;
                                                char ch;
                                                int nodis=4;
                                                if(cnt<4)nodis=cnt;
//                                              while(1)
                                                {
                                                //      cls();

                                                        window(44,12,63,20);
                                                        textbackground(0);
                                                        clrscr();
                                                        window(1,1,80,25);

                                                 for(int i=pointer;i<(pointer+nodis);++i)
                                                 {
                                                  if(pointer+selp==i)
                                                  {
                                                        textbackground(1);
                                                        textcolor(15);
                                                        gotoxy(47,14+i-pointer);
                                                        cprintf("%s",IP[i]);
                                                 }
                                                  else
                                                  {
                                                        textbackground(3);
                                                        textcolor(4);
```

```
                                        cprintf("%s",IP[i]);
                                    }
                                }

//                                  ch=getch();
                                    if(ch==72)
                                    {
                                        selp--;
                                        if(selp<0)
                                        {
                                            selp=0;
                                            pointer--;
                                            if(pointer<0)pointer=0;
                                        }
                                    }
                                    if(ch==80)
                                    {
                                        selp++;
                                        if(selp>3)
                                        {
                                            selp=3;
                                            pointer++;
                                            if(pointer>(cnt-
nodis))pointer=cnt-nodis;
                                        }
                                    }

                                }

}
int Netview()
{
                                int x1,x2,y1,y2,status;
                                char IP_ad[30][30];
                                int type;
                                int count =0;
                                int sub[30];
                                FILE *f;
                                f=fopen("IP_ADD.TXT","r");
                                if(f==NULL)
                                {
                                    printf("file Not open ..");
                                    exit(0);
                                }

                                while(fscanf(f,"%d %d %d %d %s %d %d
%d",&x1,&y1,&x2,&y2,IP_ad[count],&status,&type,&sub[count])==8)
```

```
            count++;
}
int pointer=0;
int selp=0;
char ch;
box(15,7,65,22,"VIEW",1,15);
textcolor(15);
textbackground(1);
gotoxy(19,10);
cprintf("   TOWER");
gotoxy(45,10);
cprintf("   IP ADDRESS");

for(int i=10;i<22;++i)
{
        gotoxy(40,i);
        cprintf("%c",'0');
}

while(1)
{
//      cls();
 window(17,12,35,20);
 textbackground(0);
 clrscr();
 window(1,1,80,25);
 for(int i=pointer;i<(pointer+4);++i)
 {
  if(pointer+selp==i)
  {
        textbackground(1);
        textcolor(15);
        gotoxy(20,14+i-pointer);
        cprintf("%s",IP_ad[i]);
  }
  else
  {
        textbackground(3);
        textcolor(4);
        gotoxy(20,14+i-pointer);
        cprintf("%s",IP_ad[i]);
  }
 }

        ch=getch();
        if(ch==27)return 99;
        if(ch==72)
```

```
                selp--;
                if(selp<0)
                {
                        selp=0;
                        pointer--;
                        if(pointer<0)pointer=0;
                }
        }
        if(ch==80)
        {
                selp++;
                if(selp>3)
                {
                        selp=3;
                        pointer++;
                        if(pointer>(count-
5))pointer=count-5;

                }
        }
        view(sub[pointer+selp]);
        if(ch==13)
        {
                strcpy(NETIP,IP_ad[pointer+selp]);
         return 1;
        }

}
}
```

Program for sending SMS

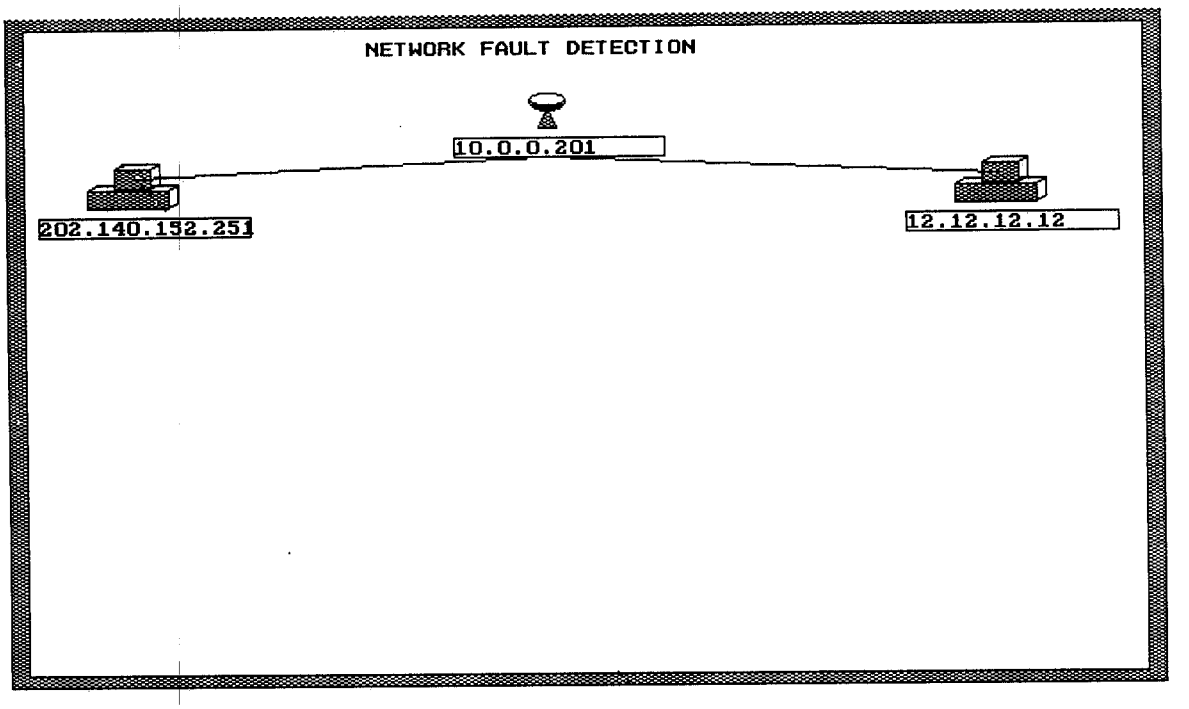
```
#include<stdio.h>
#include<string.h>
#include<process.h>
#include<stdlib.h>
int ping(char str[40]);
void main()
{
                                        int pkt;
                                        pkt=ping("ping 192.34.34.34 > temp.txt");
                                        printf("\nThe packet Received=%d",pkt);
                                        if(pkt<3)
                                        {
                                                system("gnokii –sendsms 9842288871
Connection to bridge lost");
                                                return;
                                        }

                                        pkt=ping("ping 192.34.34.34 > temp.txt");
                                        printf("\nThe packet Received=%d",pkt);
                                        if(pkt<3)
                                        {
                                                system("gnokii –sendsms 9842288871
Connection to DNS lost");
");
                                                return;
                                        }
                                        pkt=ping("ping 192.34.34.34 > temp.txt");
                                        printf("\nThe packet Received=%d",pkt);
                                        if(pkt<3)
                                        {
```

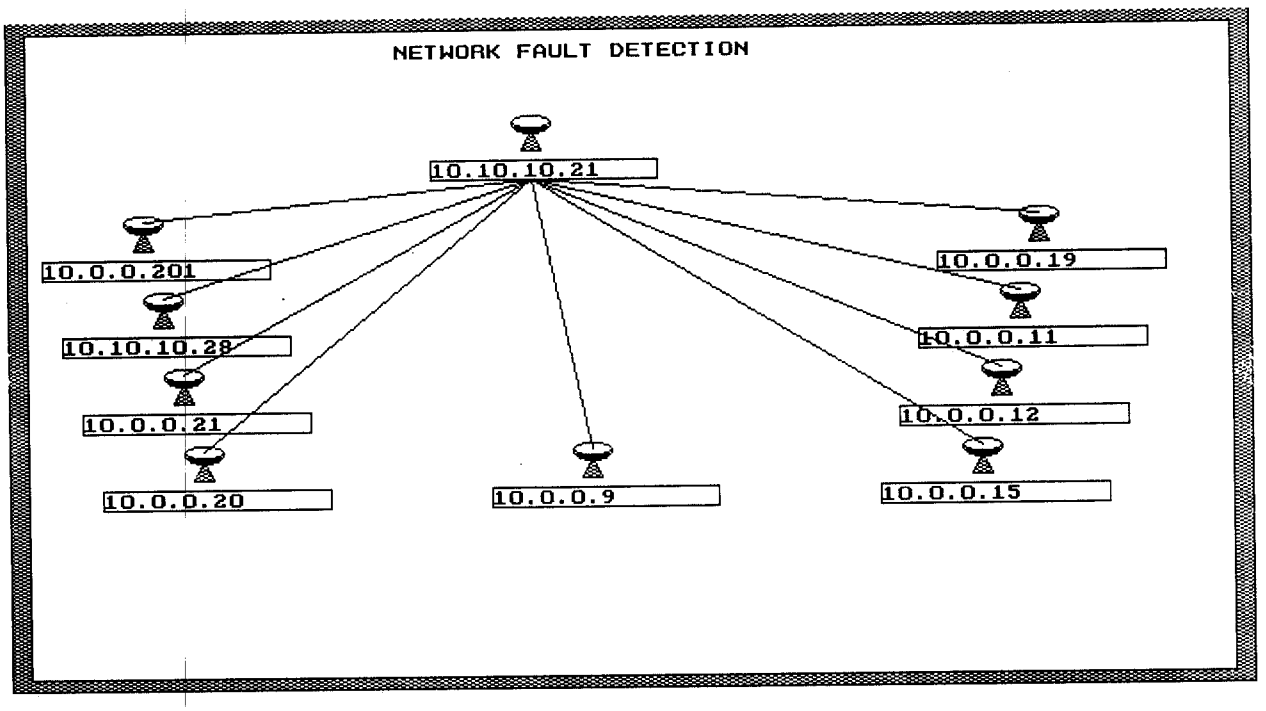
```
                        system ("gnokii –sendsms 9842288871
Connection to gateway lost");
");

                        return;
                }
                pkt=ping("ping 192.34.34.34 > temp.txt");
                printf("\nThe packet Received=%d",pkt);
                if(pkt<3)
                {
                        system("gnokii –sendsms 9842288871
Connection to net lost");
");

                        return;
                }

                getchar();

}
int ping(char str[40])
{
                char st[40];
                char Rev[40];
                int pkt=4;
                FILE *f;
                system(str);
                f=fopen("temp.txt","r");
                if(f==NULL)
                {
                        printf("file Not open");
                        exit(0);
                }
                while(fscanf(f,"%s",st)==1)
                {
                                if(strcmp(st,"received,")==0)
                                {
                                        pkt=atoi(Rev);
                                        break;
                                }
                                strcpy(Rev,st);
                }
                fclose(f);
                return pkt;

}
```

Sample Executions

Press ESC to quit
Press Any key to abort at this IP

```
Checking :=10.0.0.20              [FAILED]
Checking :=202.131.136.132        [FAILED]
Checking :=202.131.136.140        [FAILED]
Checking :=202.131.136.135        [FAILED]
```




NETWORK FAULT DETECTION
10.10.10.21
10.0.0.201
10.10.10.28
10.0.0.21
10.0.0.20
10.0.0.9
10.0.0.19
10.0.0.11
10.0.0.12
10.0.0.15