Network Monitoring System

Shirin Mazaheri

Submitted to the
Institute of Graduate Studies and Research
in partial fulfillment of the requirements for the Degree of

Master of Science
in
Computer Engineering

Eastern Mediterranean University
February 2015
Gazimağusa, North Cyprus

Approval of the Institute of Graduate Studies and Research

Prof. Dr. Serhan Çiftçioğlu
Acting Director

I certify that this thesis satisfies the requirements as a thesis for the degree of Master of Science in Computer Engineering.

Prof. Dr. Işık Aybay
Chair, Department of Computer Engineering

We certify that we have read this thesis and that in our opinion it is fully adequate in scope and quality as a thesis for the degree of Master of Science in Computer Engineering.

Asst. Prof. Dr. Gürcü Öz
Supervisor

Examining Committee

1. Assoc. Prof. Dr. Muhammed Salamah

2. Asst. Prof. Dr. Gürcü Öz

3. Asst. Prof. Dr. Önsen Toygar

ABSTRACT

Network monitoring system plays a significant role in the network security and management. Network monitoring refers to the observation on the events, happening through the network with the aim of providing a secure and persistent network. However, many small and medium sized companies and organizations prefer to escape from this fact. The most important reason for such companies is that they do not have professional network administrators to be able to use the available network monitoring system in the market. This avoidance would cost them a lot of funds or even failure in capital, in case of network failure. The proposed network monitoring in this thesis, will solve this problem because of its ease of use as well as providing all the required functionalities for monitoring a network. So, the advantage of using the proposed system, would be ability of being used by even novice users who have just basic understanding of using computer applications. According to this fact, this application could be used for small and medium sized organizations, in case of not having professional network administrators.

Keywords: Network monitoring, Remote Access, Network Security, Network Management.

ÖZ

Ağ izleme sistemi ağ güvenliği ve yönetiminde önemli bir rol oynamaktadır. Ağ izleme sistemi, güvenli ve kalıcı bir ağ sağlama amacıyla, ağ üzerinde olan, olaylar üzerinden gözlem ifade eder. Ancak, birçok küçük ve orta ölçekli şirketler ve kuruluşlar bu gerçekten kaçmayı tercih eder. Bu tür şirketlerin en önemli kaçma nedeni ise piyasada ağ izleme sistemini kullanabilen profesyonel ağ yöneticilerinin olmamasıdır. Bu kaçınma, ağ arızası durumunda, sermaye ve iş kaybına neden olacaktır. Bu tezde önerilen ağ izleme sisteminin kullanım kolaylığı sağlamasının yanı sıra, ağ izleme için gerekli tüm işlevleri de sağlayarak bu sorunu çözecektir. Önerilen sistemi kullanmanın avantajı, bilgisayar uygulamalarını kullanmak için sadece temel bir anlayışa sahip acemi kullanıcılar tarafından da kullanılabilir olması olacaktır. Bu gerçeğe göre, bu uygulama, küçük ve orta ölçekli kuruluşlar tarafından, profesyonel ağ yöneticileri olmaması durumunda bile kullanılabilir bir uygulamadır.

Anahtar Kelimeler: Ağ izleme, Uzaktan Erişim, Ağ Güvenliği, Ağ Yönetimi.

To my family. A special feeling of gratitude to my loving parents, whose words of encouragement and push for tenacity ring in my ears.

ACKNOWLEDGMENT

I would like to express my special gratitude and thanks to my supervisor Asst. Prof. Dr. Gurcu Oz, you have been a marvelous mentor for me. I would like to thank you for encouraging my research and for allowing me to grow as a research scientist. I would also like to thank my committee members, Assoc. Prof. Dr. Muhammed Salamah and Asst. Prof. Dr. Önsen Toygar for serving as my committee members even at hardship. I also want to thank you for letting my defense be an enjoyable moment, and for your brilliant comments and suggestions, thanks to you.

A special thanks to my family. Words cannot express how grateful I am to my mother, my father for all of the sacrifices that you have made on my behalf. Your prayer for me was what sustained me thus far. I would also like to thank my brother, all of my friends who supported me in this period of time, and incented me to strive towards my goal. At the end I would like express appreciation to my boyfriend, Mohammad, who was always my support in the moments when there was no one to answer my queries and helped me a lot in this thesis.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS/ABBREVIATIONS

API	Application Programming Interface
DNS	Domain Name System
FTP	File Transferring Protocol
GUI	Graphical User Interface
P2P	Peer to Peer
RRs	Resource Records
TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol
UDP	User Datagram Protocol
UML	Unified Modeling Language
VoIP	Voice over Internet Protocol

Chapter 1

INTRODUCTION

1.1 Overview

First computers, which have been invented many years ago, had a room-sized frame with a stand-alone processing unit that were able to do some simple mathematical calculations [1]. Through these many years, which has passed from the invention of the first computers, computers have been improved. Nowadays, connectivity is the biggest challenging issues for this technology. In the recent years, each simple workplace is containing many interconnected computers, printers, scanners, servers and so on. Also, even personal computers are connected to other computers, smart home electronic systems and such examples from a private place. The mentioned connectivity is provided by networks. By creating networks, communication, transferring and even connectivity have become easier than before.

Through the years and by improving technology, networks have been changed from connection between a few computers to connecting number of computers in many networks, which is describing the term of Internet. This complex network would be demanding for having a rich management. So network management has been a demand, from the formation of Internet.

The term of Network Management is related to monitoring and controlling the network as well as each connected device in that network. Network monitoring will be

responsible for checking the connectivity of the devices in the network, checking and detecting malicious activities in the network and so many other tasks [2] with the goal of providing a healthy network with high performance.

In a small sized network such as home network, network management might not be considerable issue. However, for large organizations having a smooth and healthy network with high performance, can be considered as a priority. In case of not having good network management, organizations can lose even a big amount of their profit and it can cause bankruptcy for that organization. For some instance, network of banks, airlines, libraries and so many other organizations can be considered that in case of having problem in their network, they would not be able to provide the promised services. So each organization must keep its network up and healthy to be able to provide any services it promised for [3] [17].

Thus, nowadays in organizations having smooth and healthy network is the top most priority. For achieving this goal, organizations need to provide good management on their network.

1.2 Thesis Contribution

The proposed Network Monitoring System will have client and server architecture. The tasks of server side, would be controlling and monitoring entire network to find and fix all failures and disconnections of the workstations connected in that network. Since in an organization, all the workstations connected to the network have same resources and profit, so providing a smooth and secure network is necessary [17]. This goal would be achieved by providing network monitoring system which will be able to control the security of entire network and be aware of any malicious activities in the network. Besides the security and organizations' concerns, each activity which has been done by the workstations connected to the network, would be monitored in order to proving awareness about any violence or illegal activities which might affect other organizations. According to provided explanations, network monitoring and management is required for each organization in order to have smooth and secure network [18]. There are many systems which provide this ability for administrator of organizations. Most of such systems are designed to be used by expert and professional users which might not be existed in some organizations.

The proposed application in this thesis, has been designed for novice users who have the basic knowledge related to computer applications. The aim of this thesis would be providing network monitoring system for the organizations which do not have professional personnel in this field as well as students and people who are interested in learning and practice, so they would be able to use the proposed application for monitoring the desired network. The proposed network monitoring system has many functionalities that one of them is related to testing the connectivity of the workstations connected to the network as well as testing the network's performance. This goal

would be achieved by using ping method. In this application, the server will send pings, which are included of packets, to workstations. The sent packets to the workstations will be sent back. Server, by calculating the timing of sending and receiving process, will be able to analyze the performance of network and also can be noticed about disconnections in the network. Another functionality of the proposed system would be related to its enhancement. The proposed network monitoring system, will provide the ability of monitoring and controlling the workstations. In this application, having remote access to the workstations has been assigned to the server. So the server would be able for having access to workstations' desktop for some purposes such as shutting down or restarting them.

As it has been described previously, the major focus of the proposed system, besides the monitoring tasks, is ease of use to be useful for all users even without professional knowledge of networking. This application would be used for monitoring and checking the clients' devices, connected to network and the hardware characteristics such as connection of cable or adapter type. Besides that, there are some other features that this system provides for the users in order to monitor the entire network, which will be described completely in implementation chapter. So this system would be a good choice for small and medium sized organizations and enterprises in order to manage and monitor their entire network. Also, being easy to use would provide an opportunity for practicing and learning monitoring of the network.

The primary purpose of implementing this project is to provide a simple to use network monitoring system which contains all necessary functionalities. The network monitoring system implemented for this thesis, has the capability of being used by students and novice users. So, this application can be used for educational and training

purpose. Besides the mentioned purposes, the main focus of proposed network monitoring system is network security and management. Monitoring and sniffing packets will allow the admin of user to control the security of entire network. So, in a nutshell, the main purpose of implementing the proposed network monitoring system is to provide an easy to use tool. The main usage would be in small and medium sized organizations, as well as for students and with the purpose of training.

1.3 Thesis Outline

This documentation contains reports on all processes related to development of the proposed system. The structure of this documentation would be as follow: In Chapter 2, an overview of networking in organizations and also a literature review on network monitoring will be given. Chapter 3 will be presenting the architecture of the implemented system. In Chapter 4, the implemented system will be discussed, along with prepared screenshots for each section. In the last chapter, a conclusion on the work, which has been done for this thesis, will be given.

Chapter 2

LITERATURE REVIEW

2.1 Introduction

Prior to implementation of the proposed system, a sufficient research on the published literatures related to this topic, had been done. In this part, the summary of the mentioned research and investigation would be discussed. All the information and knowledge gained from this investigation and research, had been thoroughly helpful, in order for the proposed system to be implemented. The main idea has been inspired from the researches which had been previously done.

Two main areas, useful for the implementation of the proposed system, have been investigated in this part, which are Network Monitoring and Remote Access. The purpose of this research will be defined and concepts, which have been helpful in understanding technologies used in development of proposed application, have been collected.

Finding the faults and disconnections in the network of any organizations, which might be affecting the performance of network, would be performed by having a proper supervision and management on the network [4]. The network must be managed to capture the entire network's performance and maintain any possible disconnections. In case of any occurred failure in the network, an urgent monitoring and maintaining will be required.

In an enterprise, for an instance, having an urgent monitoring is significant as in such these cases even a short delay can led the enterprise to failure in achieving its goals. Thus, in an enterprise or a large organization, the main focus of the network monitoring must be on monitoring the internal and external behavior and activities of the workstations connected to the network. External activities is related to the activities taking place in the network as well as connection of workstations to network. On the other hand, internal behaviors can be mentioned as the activities of each workstation and all the interactions between them. So in the internal behavior analysis, administrator would be able to monitor all the vents on operating system of the workstations, which are connected to the network. However, as the external behavior of the network determines the performance of the network, the main focus in most of the network monitoring systems is on the external behaviors.

2.2 Network Monitoring

In this part, some previously proposed models for network monitoring will be discussed. Network monitoring system has three essential roles, which are performing smooth and healthy network, providing report on network status and giving report on event reports. Morris Sloman, have proposed a new model for network monitoring which modifies the event management in previously proposed models [15] [16]. The main focus of this proposed new model was on the events and status of the connected systems to the network and the goal was creating monitoring report to be sent to the administrator of the network. In this method, Ping had been used for either automatically produced report or by administrator's request [16].

In the mentioned model, event and status reports have been produced in order for the administrator to be able to analyze the network performance as well as interactions between workstations [16]. Figure 2.1, presents the design of the proposed system.

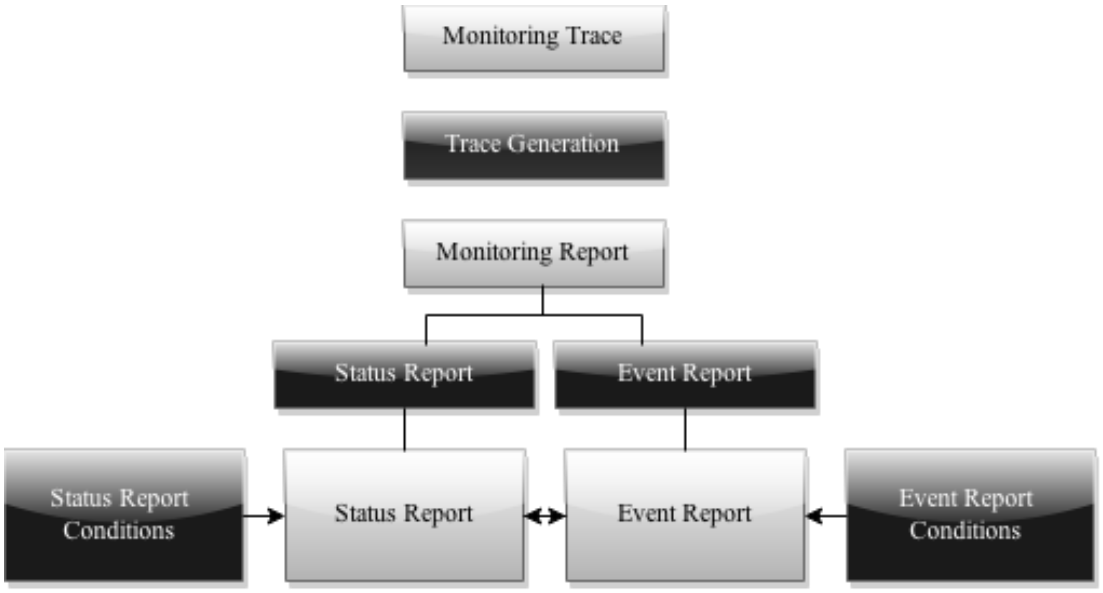


Figure 2.1: Network Management Model [16]

The other model, which has been analyzed, is the DiMAPI model [5]. DiMAPI model proposed a network monitoring Application Programming Interface (API), which added some modifications on the traditional monitoring models. This proposed model has been inspired from traditional models, using ping method. This model assumes that a smooth network must be monitored from numerous points. DiMAPI did some modifications on the previously proposed method, called as MAPI. MAPI model, which had been proposed before DiMAPI, relies on a single point monitoring system [5].

DiMAPI, unlike MAPI, focuses on monitoring packets via ping by performing network monitoring on various point of the network. DiMAPI suggests that using ping method can be helpful for many purposes such as calculating bandwidth, monitoring event, checking network status and so on.

In [5], authors presented the main features of DiMAPI which are monitoring traffic in the network and Peer to Peer (P2P) interactions monitoring. P2P, as a network

terminology, refers to client server architecture, in which, there must be a server and several workstations connected to that server in order to have commutations and interactions. For monitoring the traffic in the network, ping method had been used in this proposed model. Ping, by sending packets to different workstations connected to the network and also by receiving back, provides valuable information about the network traffic.

Furthermore, in the Network Monitoring Systems the significant point is providing preset functionalities for packets, in order for them to be able in delivering the specific information. In the proposed model in this study, the workstations also must be equipped with the client version of the network monitoring system.

However, monitoring an active and complex network, such as network in a manufacturing enterprise, can be difficult in case of being done manually. Regarding this fact, instead of using manual monitoring for an active network, intelligent monitoring model had been presented. In the intelligent monitoring model, in a preset time interval, packets will be sent by server and sent back by workstation, in order to have fixed connection between workstations and server. According to tests, have been done on an active network monitoring, DiMAPI model is an intelligent network monitoring system which assists in real-time monitoring as well as providing more secure network [6].

2.3 Remote Access

In a network monitoring system, remote access will give authorized access to network administrator for performing required actions on the systems connected to the network such as switches, routers and workstations.

For providing access remotely, information of Transmission Control Protocol (TCP), will be translated by the proxy server, for the router's console port and will receive it back. As regards this process, all connected devices to the mentioned proxy server would be ready for remote access, performing by serve's user. The admin of the server will decide about the required activity on each connected device. The Figure 2.2 displays this process.

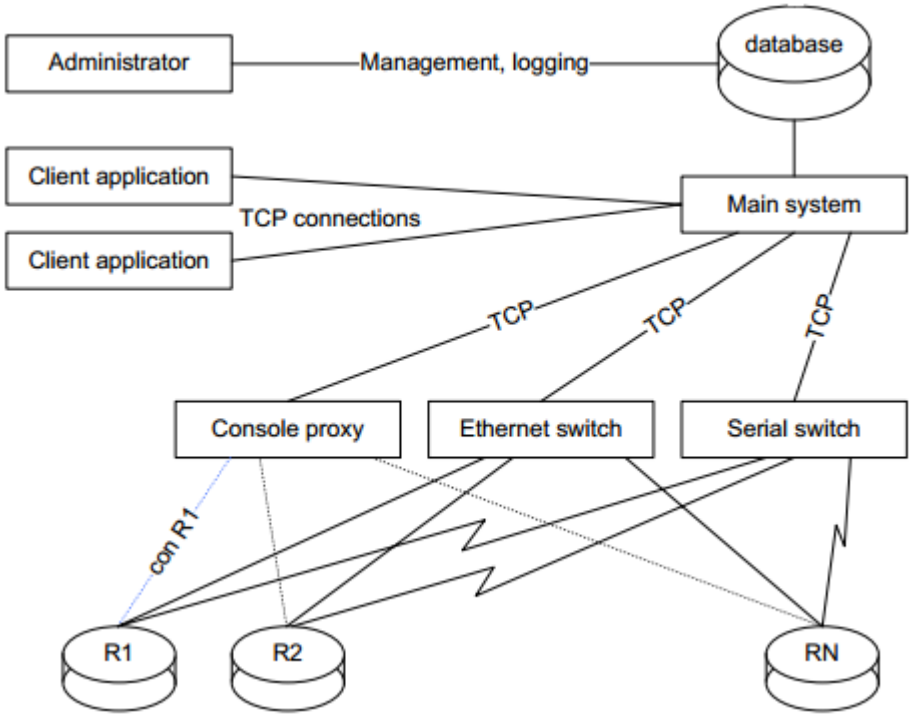


Figure 2.2: Process of Remote Access Systems [7]

The network administrator establishes a connection between two routers which are the connected one to network monitoring system's interface and the router which is connected to workstations. As it has been mentioned previously, each workstation must be equipped with the client version of the system. By establishing the connection between server and workstations, all the necessary activities can be done between them [7].

The IP address of each connected device will be stored in the server for providing the ability of choosing specific client or workstation by server, in order to perform any required actions on the desired device.

2.4 Similar Works

In this part of this chapter, the similar works to the proposed network monitoring application will be discussed. For each mentioned software, some explanations along with its pros and probably cons will be given in order to providing some familiarities with such these applications.

2.4.1 Wireshark

One of the most famous tools for monitoring and analyzing network is Wireshark [8]. Wireshark provides many functionalities such as packet monitoring and analyzing, Voice over Internet Protocol (VoIP) analyzing, traffic monitoring and so forth. This application captures the packets in the network and allows to analyze and save the captured packets [8]. This tool has many advantages as well as some disadvantages. For some instances, being free of charge, being open source and ability of running on all operating systems can be considered as some of its strengths. Besides being an admirable packet analyzer, its interface is difficult to use. In addition, Wireshark requires full understanding of Transmission Control Protocol/Internet Protocol (TCP/IP). The mentioned challenging issues, can be considered as some weaknesses of this tool.

2.4.2 Spiceworks

Another famous tool for network monitoring is Spiceworks. This tool allows the users to monitor events, happening in the network [9]. Another functionality of this tool is analyzing the network's bandwidth and performance. This software, as an in-built server, allows the administrator to control configurations of the network. In addition,

Spiceworks allows the administrator to view the information of each connected device as well as having access to their accounts and their data. Besides, administrator would be able to assist the workstations by any network administration requests [9]. Although Spiceworks is a powerful and easy to use network monitoring system, it cannot be run on Linux based operating systems. Another weakness point of this software is that Spiceworks does not give its users the ability of having control on the network and the user is only able to see what is happening on the monitored network [9].

2.4.3 OpManager

Another powerful and famous network monitoring system for enterprises, is OpManager [11] [12]. The entire network has been shown as a map (also incorporated with Google maps), listing the connected devices. So the user would be able to see each workstation on the map.

Visualization techniques are among the best features of OpManager. By using visualization techniques, administrator would be able to manage the network. This software, also provides the ability of creating a customized network map for its users. So the users would be able to customize the network map based on their own business and location of the workstations. Besides the basic features provided by this software, it also helps in device recognition and configuration, which is similar to Spiceworks. OpManager, also provides MySQL management tool which can be useful for network administrator to be able to check all stored data [11]. OpManager, same as other offered applications in the market, has some weaknesses beside its strengths. This application needs to be configured manually, which is time-consuming and also its user interface is not easy to use. The mentioned point are some facts about OpManager software which can be considered as some of its weaknesses [12].

2.4.4 Tcpdump

Tcpdump is another famous network monitoring tool. Usually, the packets which are not addressed to the network, are subject to be ignored in the network interface [8] [10]. This software captures all the packets in the network, regardless of the addressing by placing the interface on promiscuous mode. So by using this tool for monitoring network, administrator would be able to examine the packets and also based on the type of information stored in the packets' header, administrator will be able to analyze the network traffic [8]. Tcpdump is a powerful tool which is able to being run on UNIX based platforms and its structure is command line. In practice, this software is similar to Wireshark and both of them are packet capturing programs which capture packets of any protocols and display the captured packets to administrator [10].

Table 2.1: Comparison of Provided System With Other Systems

Features	**Wireshark**	**TCPdump**	**OpManager**	**SolarWinds**	**MOSH**
Reading Packets	✓	✓	✓	✓	✓
Capturing Packets	✓	✓	✓	✓	✓
Filters for Displaying Data	✓	X	X	X	X
Detecting VoIP calls	✓	X	✓	X	X
GUI Based	✓	X	✓	✓	✓
Command Line Based	X	✓	X	X	X
Hardware Monitoring	X	✓	✓	✓	✓
Average Response Time	✓	✓	✓	X	✓
Network Trace	✓	✓	✓	✓	✓
Chatting System	X	X	X	X	✓
Client-Server Architecture	X	X	✓	✓	✓
Packet Loss	✓	✓	✓	X	X
Saving Log	✓	✓	✓	✓	✓
Statistical Report	✓	X	X	✓	✓

The comparison between four existing tools in the market and the proposed network monitoring system, called as MOSH, is shown in Table 2.1. The comparison is based

on the functionalities of each tool [8] [9] [10] [11] [12]. The functionalities of the proposed system in this thesis, would be similar to existing network monitoring systems. Additional functionality of the proposed system, in compare with similar systems, is the chatting system. Chatting system is extra feature, which has been provided for easiness. Besides that, MOSH network monitoring system would be useful for practice purpose as well as being useful for novice users, as it has easy to use user interface.

Chapter 3

SYSTEM OVERVIEW

3.1 Unified Modeling Language

For giving better understanding about the implemented network monitoring system, Unified Modeling Language (UML) has been provided. UML, is a standard graphical symbolization for explaining design of the implemented software or system. Generally, UML is included of five diagrams and specifications, in order to offering some familiarities with the proposed system for each functionality provided in the application [13] [14].

3.1.1 Use Case Diagram

The first step in the UML is Use Case diagram, which can be seen in Figure 3.1. Use case diagram is a graphical design of the main functionalities of proposed system. The provided use case in this thesis, Figure 3.1, shows accessibility of two actors, which are client and the server. The following diagram shows the core functionality of the implemented application [14].

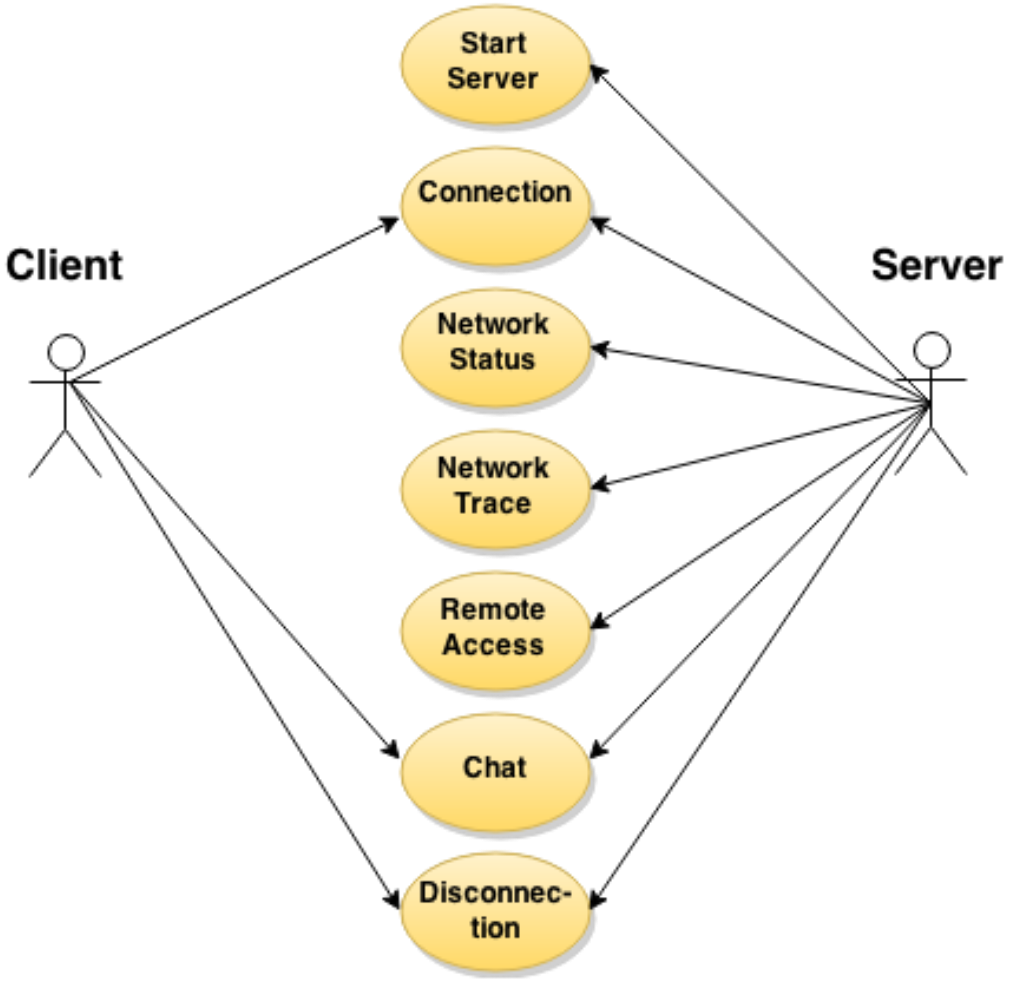


Figure 3.1: Use case diagram

Description about Use Case

In this part, in the following tables, some detailed information regarding presented functionalities in use case diagram, shown in Figures 3.1, would be given. The following Tables 3.1-3.7, would be containing mentioned descriptions.

Table 3.1: Definition of Starting Server

Starting Server	
Actors	Access is assigned to Server
Description	Admin of the server is able to start it
Regulations	Admin has permission for running the server
Prerequisite	--
Result	Admin of the server must start the system to let client be connected

Table 3.1 is describing the process of starting the server side application. As it can be understood from the mentioned table, the access is only assigned to the admin of server. The first step of the network monitoring process would be starting the server by administrator. Also the permission is cleared in the mentioned table, which the admin is allowed to run the server. In this step, there is no pre-request, as it is the first step of this application. So, by the end of this step, the server will be run and the client would be able to be connected to it.

Table 3.2: Definition of Connecting to the System

Connection	
Actors	Access is assigned to Server and Client
Description	Server as well as client are able to connect to the system
Regulations	Admin of the server establishes the connection, then client will be able to connect to the server
Prerequisite	Server sets the port number, so client by using port number and server's IP address can be connected to the server
Result	The connection between client and server will be established

The next step is related to the process of connection, as it is shown in Table 3.2. The mentioned table is a definition of this process. Connection can be done by both server's user and client's user. So, both sides of this application, which are client and server, will be able to connect to the system. The procedure is establishing the connection by server and then client will be able to connect to the server. This step has a significant pre-request, which is using port number and IP address of the server by client. Thus, Server must set the port number, and client by using that port number and server's IP address can be connected to the server. The final result of this step would be establishing the connection between client and server. This should be noticed that in the implemented system for this thesis, the port number is pre-set, which makes the system simpler to use.

Table 3.3: Definition of Network Status

Network Status	
Actors	Access is assigned to admin of the Server and user of Client
Description	Information of the network devises of server side will be seen by admin of the Server and information about client side will be seen by Client
Regulations	--
Prerequisite	Selecting the desired network adapter
Result	Admin of the server and user of client will be able to check the network status

The section of network status, can be observed in Table 3.3. This table is describing the mentioned section. Network status section is related to network devices' information. This feature has been provided for both admin of server and user of Client side. In this section, the server side would be able to check the networking devices' of server side and client would be able to check the networking devices' of the client side. This feature, as the name suggests, provides the ability of checking and controlling the networking devices and in case of having any problem, the system would detect it and provides the solution, for the purpose of troubleshooting. Client by receiving the information about any fault or disconnection in the network, which would be provided in this section, can either use the troubleshooting solution to fix it or gives the information about the problem to the admin of server for getting more help. The process of transferring the information can be done via chatting system.

Table 3.4: Definition of Network Trace

Network Trace	
Actors	Access is assigned to admin of the Server
Description	Tracing the IP addresses of desired destination
Regulations	Inserting preferred IP address for tracing
Prerequisite	The chosen IP address must be existed
Result	Admin of the server will be able to trace the desired IP address

The section of network trace is described in Table 3.4. Access to this section is assigned to the server side of this application. By inserting the host name or IP address of desired destination, admin of server will be able to check the connection between server and destination. This would be possible by checking the connection to the devices such as routers and switches, which are in the path between server and destination.

Table 3.5: Definition of Remote Access

Remote Access	
Actors	Access is assigned to admin of the Server
Description	Admin of the server will be able to have remote access to the clients connected to the server
Regulations	Admin of the server will select the desired client from the dropdown list and then chooses the preferred action
Prerequisite	--
Result	Remote access will be assigned to server for the connected client

Description of the section of remote access, can be observed in Table 3.5. This section has been provided for server side, so the access is assigned to the admin of server. After establishing the connection between the server and client, admin of the server will be able to have remote access to the clients connected to the server. This feature allows the admin to perform some required actions, in case of need, on the client's device. Remote access's actions are shutdown, restart, abort and logoff the client's device. The procedure is that admin of the server will select the desired client from the dropdown list and then chooses the preferred action.

Table 3.6: Definition of Chatting Feature

Chatting	
Actors	Access is assigned to admin of the Server and Client
Description	Both server and client, will be able to start chatting
Regulations	--
Prerequisite	For sending message to client, only the server must chose the receiver
Result	Provided ability of having conversation between server and client

The chatting feature has been assigned to both client and server side of this application. So both of them will be able to start chatting. For sending message to client, only the server must chose the receiver. So, result would be the ability of chatting for client and server side in case of being connected to each other. This process is defined in Table 3.6.

Table 3.7: Definition of Disconnection

Disconnection	
Actors	Access is assigned to admin of the Client and Server
Description	Both client and server will be able to disconnect their systems
Regulations	--
Prerequisite	--
Result	The connection will be disconnected

Table 3.7 is showing the description about disconnection process. The ability of disconnecting the system is assigned to both client and server side. So, both client and server will be able to disconnect their systems.

3.1.2 Class Diagram

Class diagram is another step of the UML technique. Class diagrams, generally, are used for object-oriented coding and they suggest how the system will be designed. This step of UML technique is included of class names, relationships between classes, properties and attributes used during the implementation of the proposed system [14]. Figure 3.2, displays the class diagram of the implemented network monitoring system, which has been generated in Visual Studio.

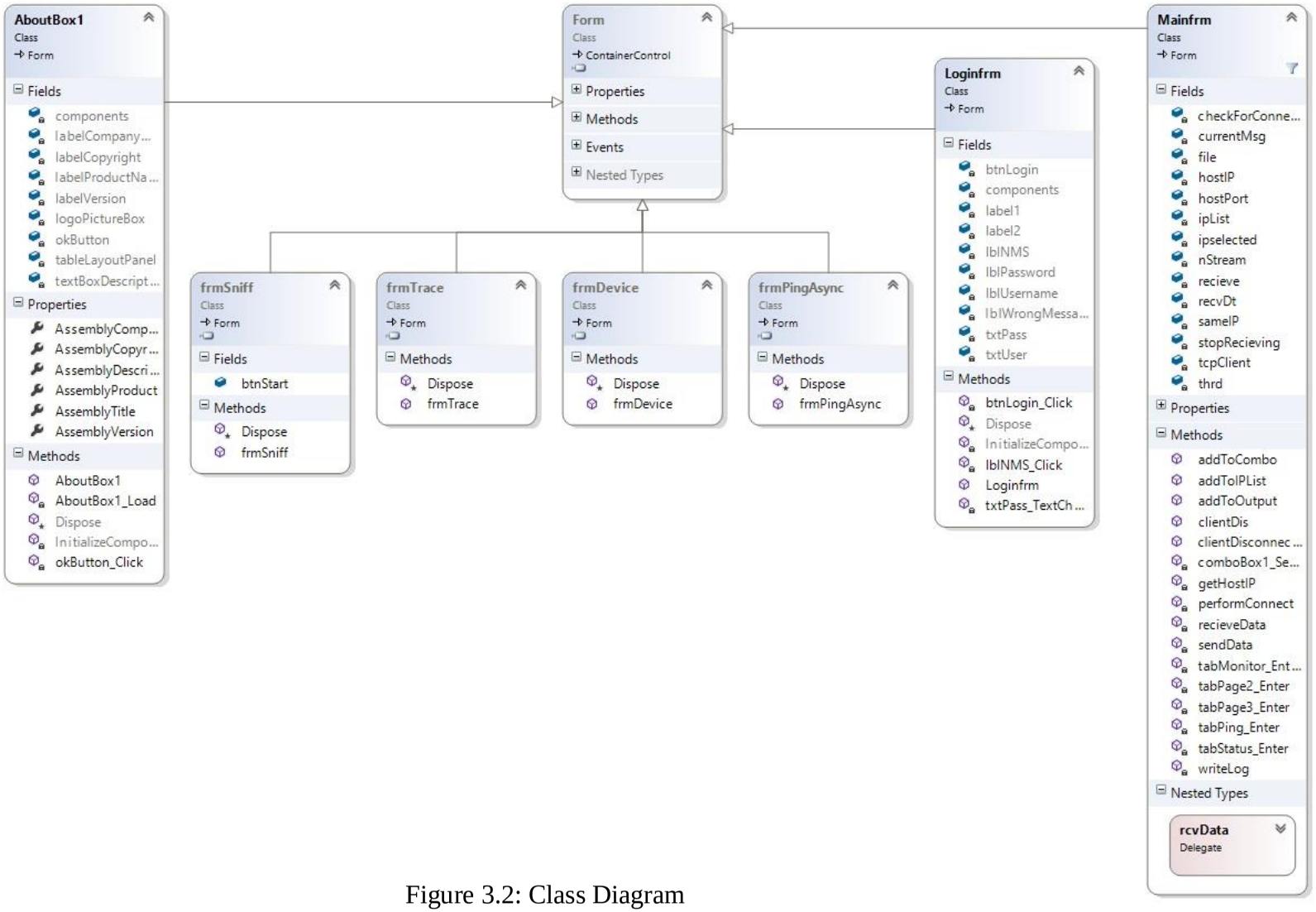


Figure 3.2: Class Diagram

3.1.3 Sequence Diagram

Sequence diagram is another method for UML technique. This method represents the sequence of activities in this application. In this diagram the arrows are presenting methods and the rectangles show the classes. Generally the required number of sequence diagrams must be identified based on the nature of application [14]. For the application implemented in this thesis, for the connection process, that client and server are both involved, a sequence diagram has been created, which can be observed in Figure 3.3.

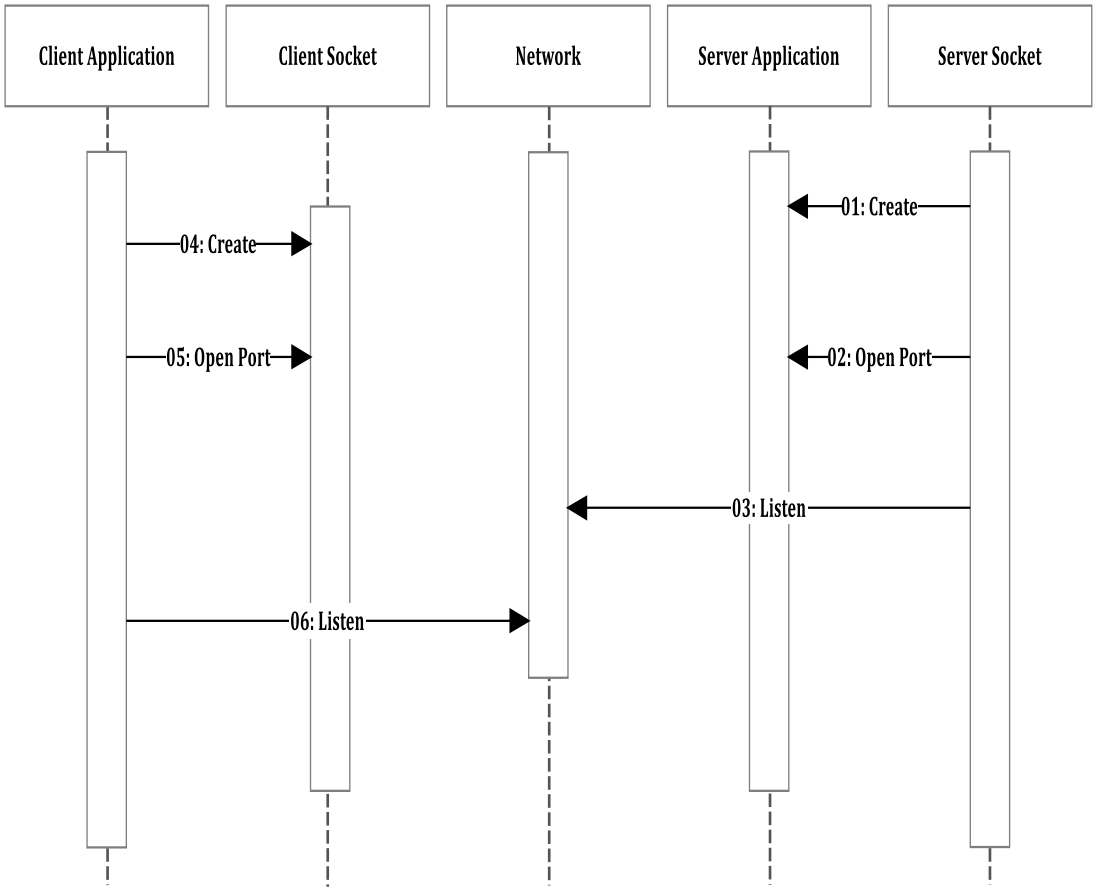


Figure 3.3: Sequence Diagram, showing Connection Process

3.1.4 Activity Diagram

Another diagram in UML technique is activity diagram, which presents the design of complex operations, by dividing the whole process into smaller activities for giving better understanding about the functionality of implemented application. As it can be observed in Figure 3.4, activity diagram for the implemented application is presenting the design of activities, which take place in remote access process.

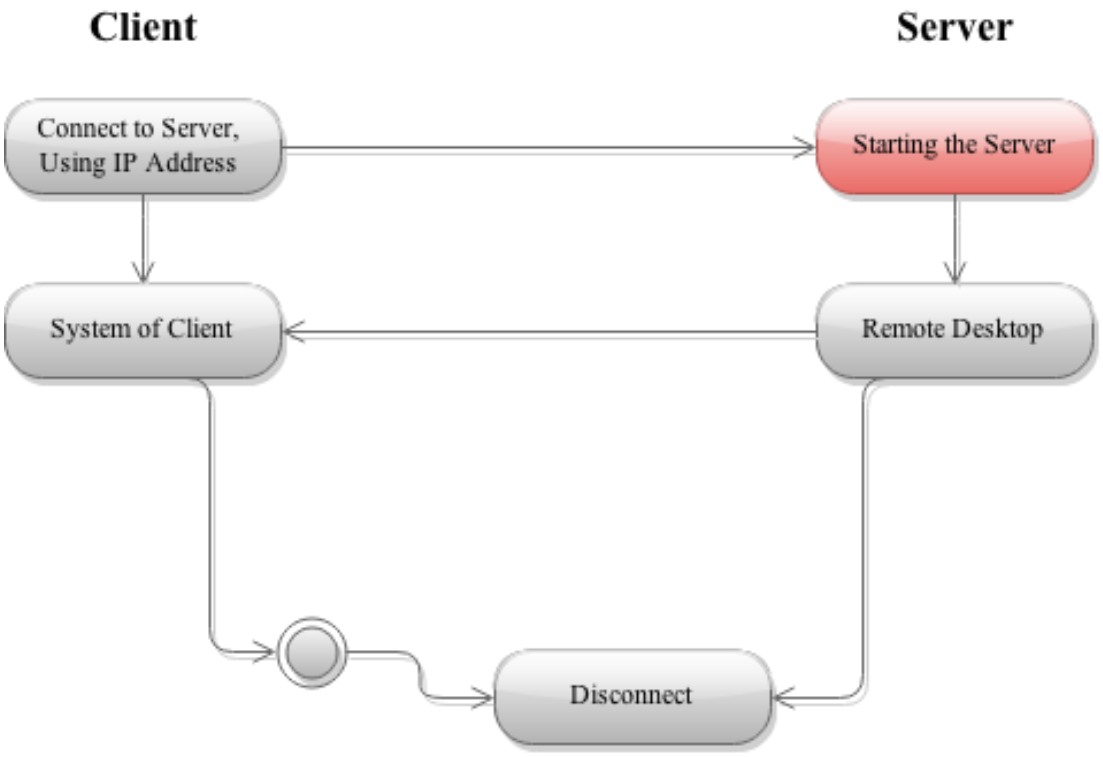


Figure 3.4: Activity Diagram, Showing Remote Access

Chapter 4

IMPLEMENTATION AND RESULT

In this chapter, the main focus will be on the implementation of proposed network monitoring system. This system has been done, using C# programming language in Visual Studio environment. The Microsoft .Net framework is a good environment for network programmers. The reason of choosing C#, is that this programming language lets the programmers to design network applications by using the features of windows networking. Thus, regarding using .Net framework and C# programming language, the implemented application would be able to be run on windows Operating System.

In the following parts, detailed information about each part of the application would be given, along with the related screenshot with the intent of providing thorough comprehension about scope of the project. Also, for each core section of the implemented application, a summary of significant part of coding will be observed in Appendix section. Additionally, the structure of this application is included of two parts, which are server side and client ide. Thus, the specification of each part would be explained separately.

As it has been mentioned before, this application is based on client-server architecture. The core system is constructed based on having this application on both, client and server devices. Since the server is connected to the network, clients are allowed to be connect to the server by using appropriate port number and server's IP address. The

structure of client-server architecture, which is used for implementing this application is showing in Figure 4.1.

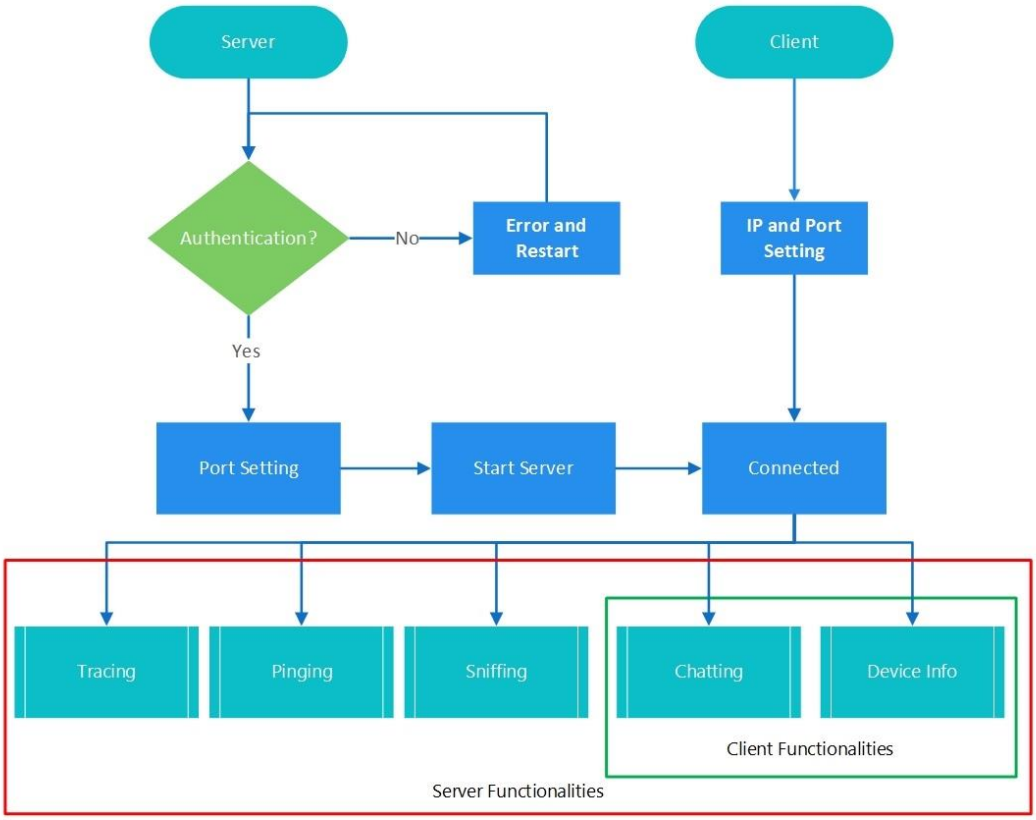


Figure 4.1: Client-Server Architecture

4.1 Server Side

In the server side, as it has been completely explained in previous part of this study, all the functionalities of the implemented system can be observed, as will be described in following parts.

4.1.2 Login Section

According to the provided screenshot for this section, shown in Figure 4.1, a login page has been designed. Administrator would be able to sign into the server side of the system, by using pre-set username and password. The username and password had been set in the application and are stored in database.

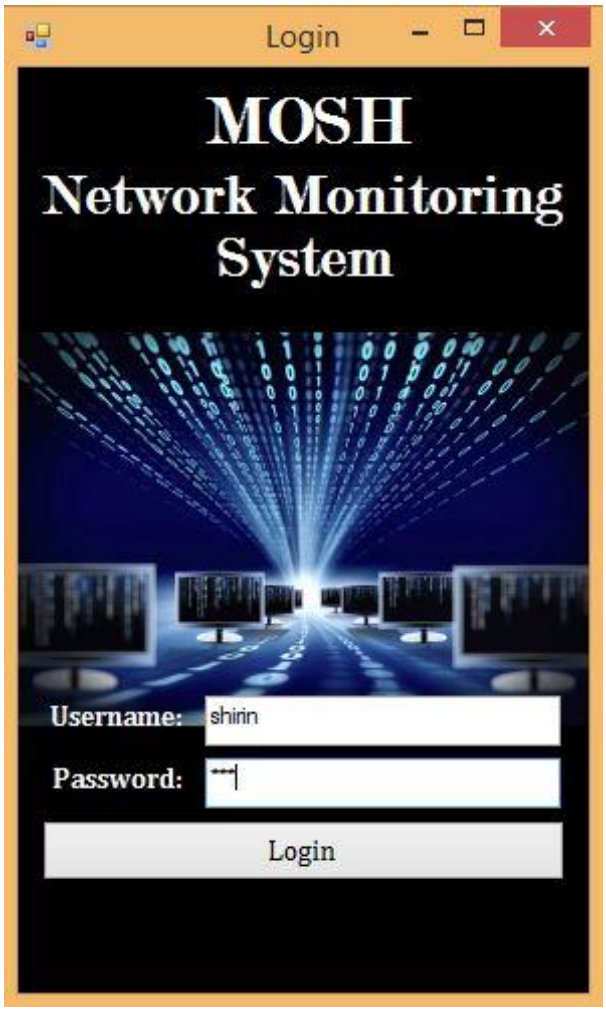


Figure 4.2: Login Page

4.1.3 Remote Control Section

This section is the first section, provided for administrator of the server. In this part, as it can be observed in Figure 4.2, remote control section has been divided into 4 main parts, which are setting port number, remote control, show log and finally chatting environment. The first task for the server must be establishing connection for allowing the clients, to be connected to the server. Although a pre-set port number has been set for the ease of use, but in case of need admin will be able to change the port number. In this section, after starting the connection between client and server, some options are provided to be used by server administrator. In the next step, the server's user must

choose the desired client's IP address from the dropdown box, provided in this window. By selecting the preferred client, which is the connected device to the server, the server will be able to send to that precise client a message and also server will have access to the client's desktop for performing required activities. By pressing the show log button in this window, admin will see the saved logs in a predefined log form.

As it can be observed in Figure 4.2, by connecting each client to the server, a message will be appeared in the chatting part of this system which informs the server's user about the new client's IP address. Additionally, Server's user by choosing the desired client's IP address, will be able for some pre-defined operations. For an instance, in remote controller part, after choosing the preferred IP address, server can shut down the client's device by pressing remote command button.

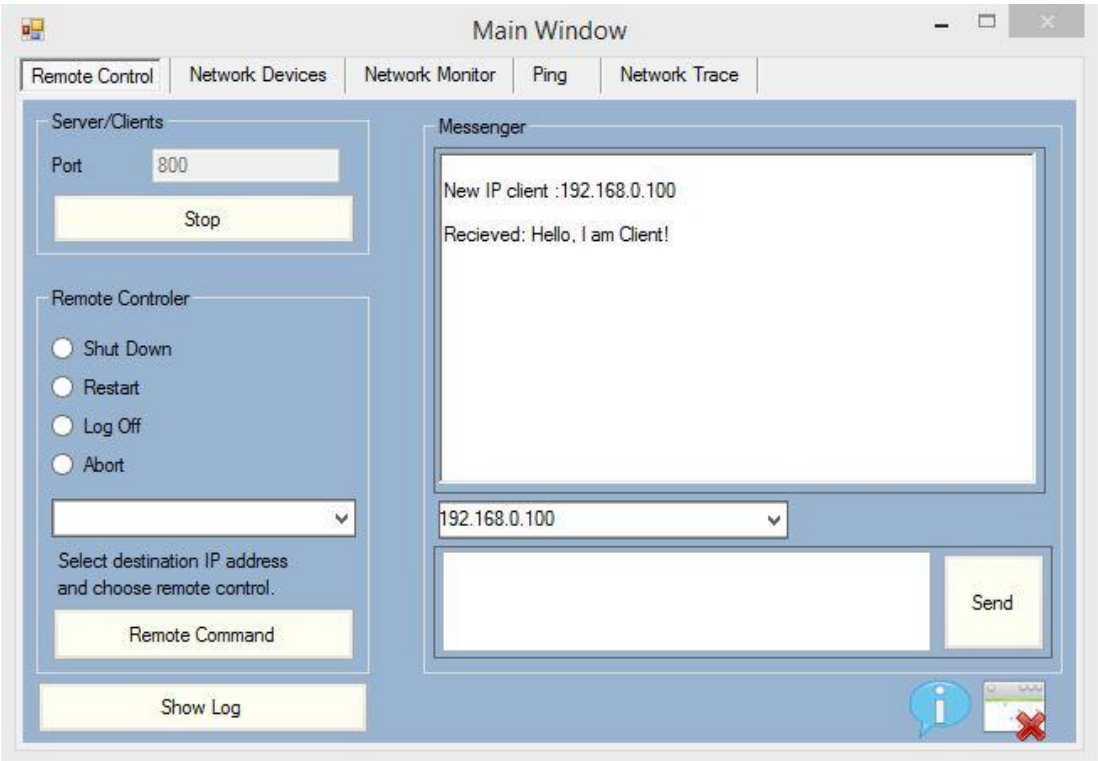


Figure 4.3: Remote Control

4.1.4 Network Devices Section

The following coding sample, shown in Figure 4.4, describes one of the important parts of this section. First the pre-defined integer, which is named as m_Index, must be empty and for achieving this goal, the value is set to be equal to -1, to be initialized from 0. For presenting the information of network devices, the host must be alive. To check the availability of host in the network, the specified StartMonitor method has been used. If the desired host is alive, then user interface must be invoked, using MethodInvoker, and then will be threaded to the pre-set place in system's interface.

```
private void frmMain_Load(object sender, EventArgs e)
{
    try
    {
        m_Index = -1;
        NetworkManager.Instance.StartMonitor();
        m_IsAlive = true;
        UIInvoke = new MethodInvoker(UpdateUI);
        Initial();
        _thread = new Thread(new ThreadStart(UpdateStatus));
        _thread.Start();
    }
    catch (Exception)
    {
        MessageBox.Show("Network Adapter Incompatibily");
    }
}
```

Figure 4.4: Sample part of Network Monitoring

This section, as it can be seen in Figure 4.5, is divided into two parts. The first part is related to information on network devices, which can be selected from the pre-set dropdown menu. The second part is system summary, which will be displaying the information about selected network device. So, by selecting each network device,

provided in the dropdown menu, automatically the related information will be displayed in the provided place.

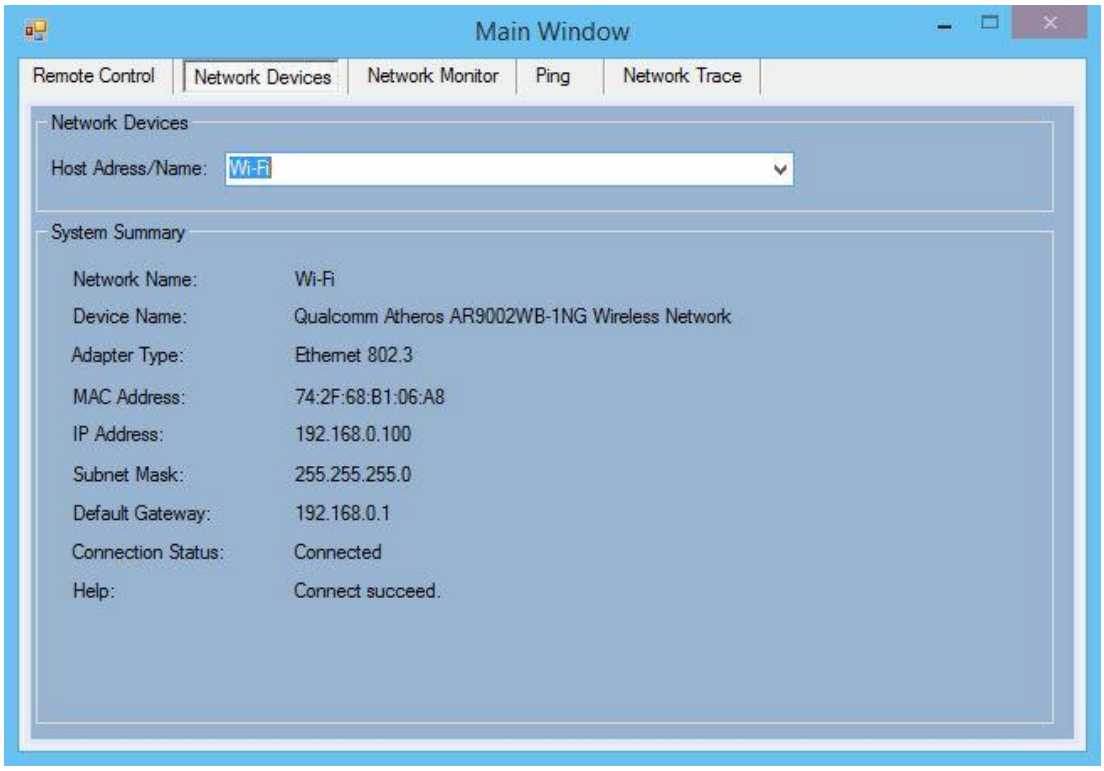


Figure 4.5: Network Device

In the Figure 4.5, Wi-Fi connection has been selected and as it can be observed, all the necessary information have been given. For an instances, information about the brand of processor of wireless card, which is used in the server's device. The explained example is shown in device name row. Also, as another example, connection status is shown. In this example, connection is successfully established and according to this fact, there would be no need for help. In case of having problem with establishing the connection, the section of help would give some information for troubleshooting the problem.

4.1.5 Network Monitor Section

In this section, server's user will be able to choose the desired IP address from dropdown menu. By pressing Start button, all the information about the packets, either sent or received, in the selected IP address would be displayed in the provided place below. Also the gathered information can be cleared by using the button next to Start button, shown in Figure 4.6.

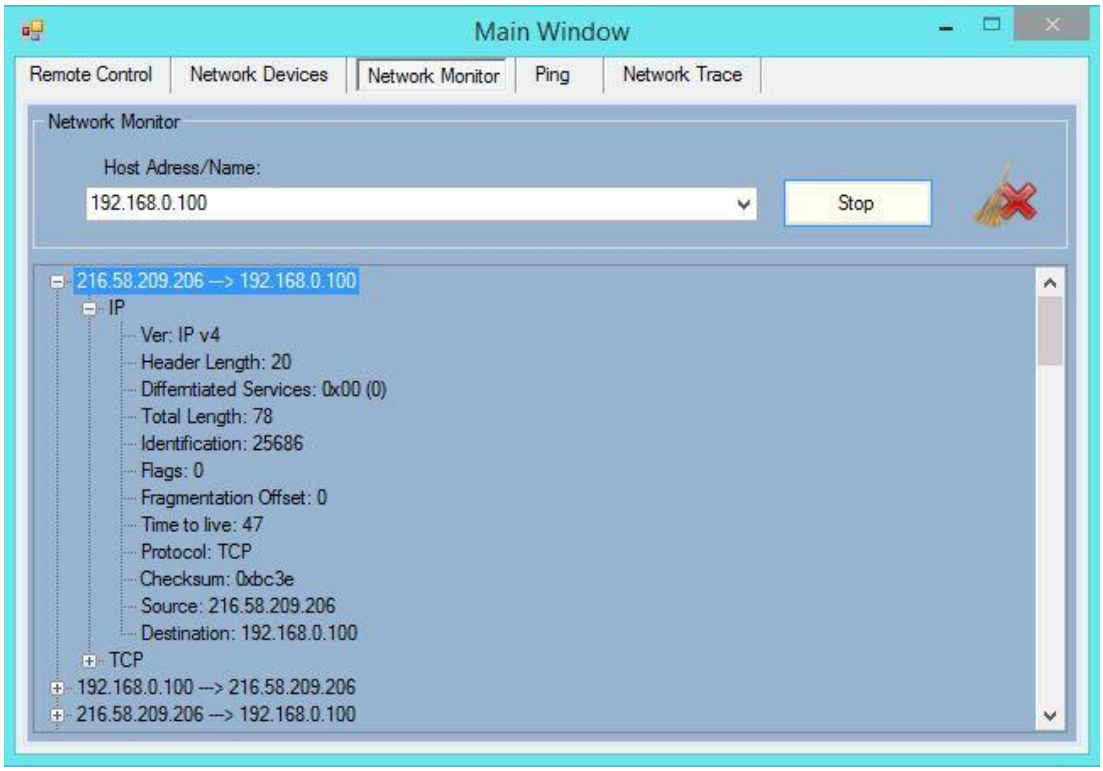


Figure 4.6: Network Monitor (Showing IP)

In the provided example, shown in Figure 4.6, one of the packets is opened to show the details in one of its sections. In this example, detailed information is under section of IP, which is presenting IP version, length of packet's header, protocol, source and destination IP addresses and some other information. The provided information, as it has been described above, helps the server's user to be aware of the activities which has been done through the network by selected client.

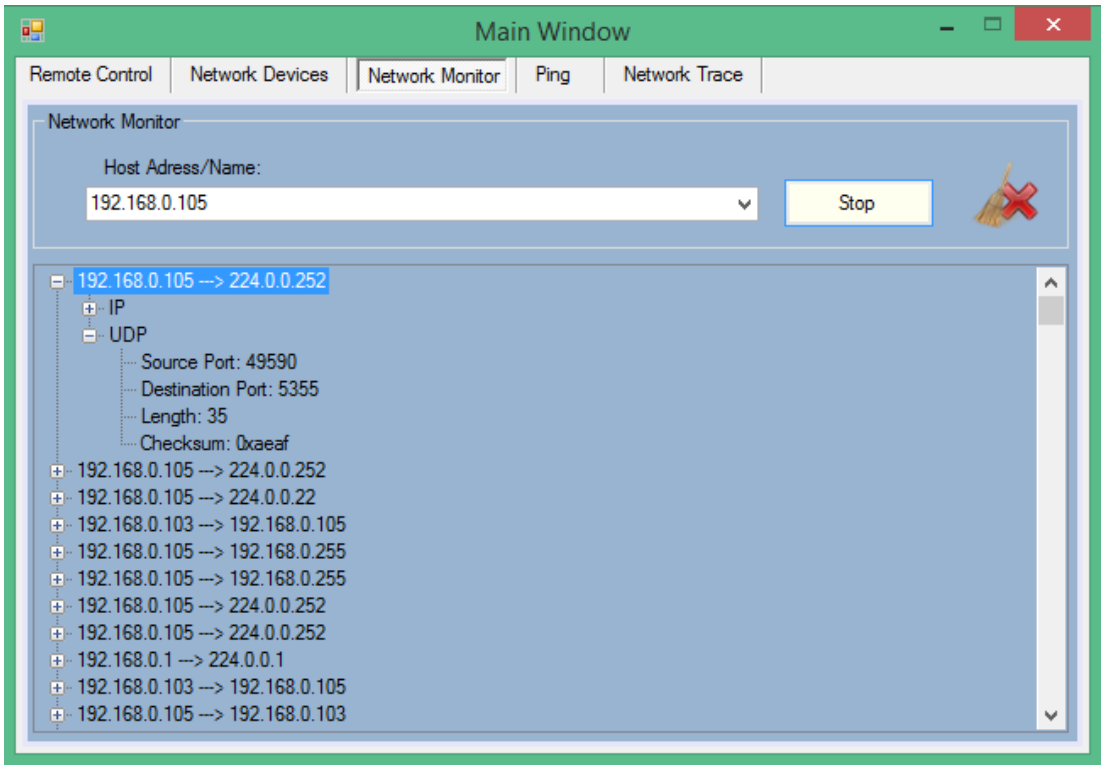


Figure 4.7: Network Monitor (Showing UDP)

User Datagram Protocol (UDP) is a communications' protocol that offers a limited amount of service when messages are exchanged between computers in a network that uses the Internet Protocol (IP). Figure 4.7 shows an example for presenting the section of UDP. In this example, detailed information under section of UDP is given. As it can be observed in the Figure 4.7, information provided under this section are used port in source and destination, length of message and checksum.

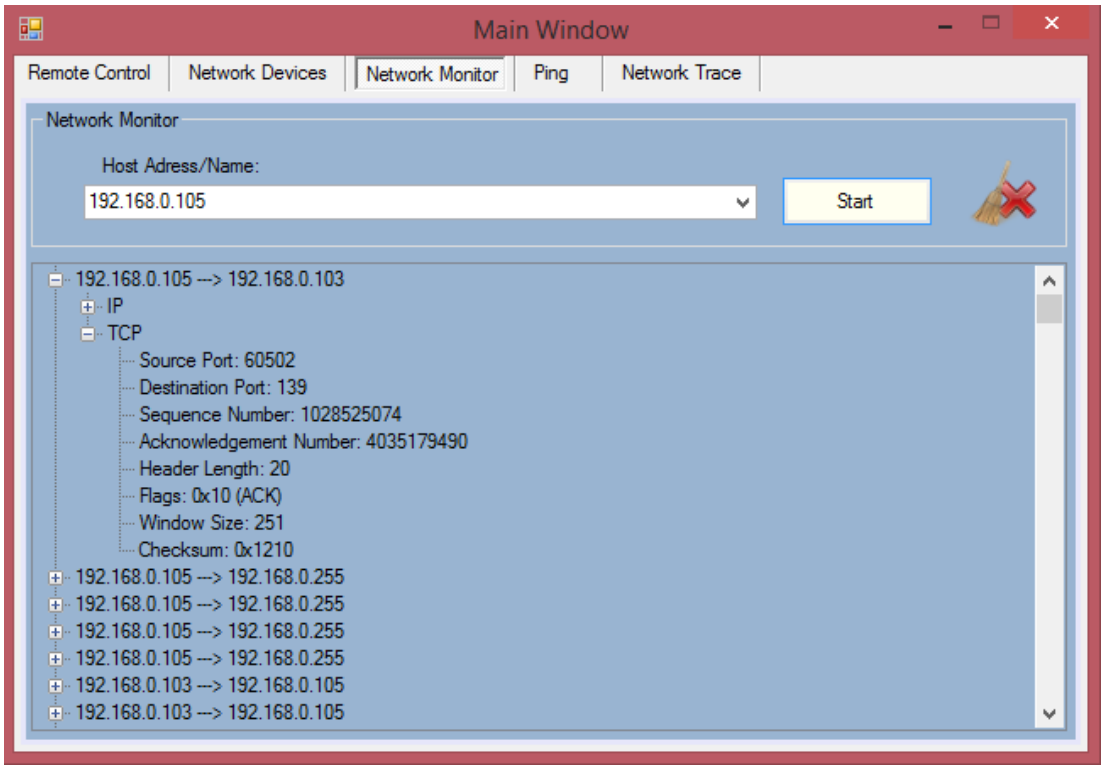


Figure 4.8: Network Monitoring (Showing TCP)

Figure 4.8 shows an example for presenting the section of TCP. In this example, detailed information about header of each packet transferred in the network, under section of TCP is given. As it can be observed in the Figure 4.8, information provided under this section are port number of source and destination, sequence number of packet, Acknowledgement number, length of header, flags and so on.

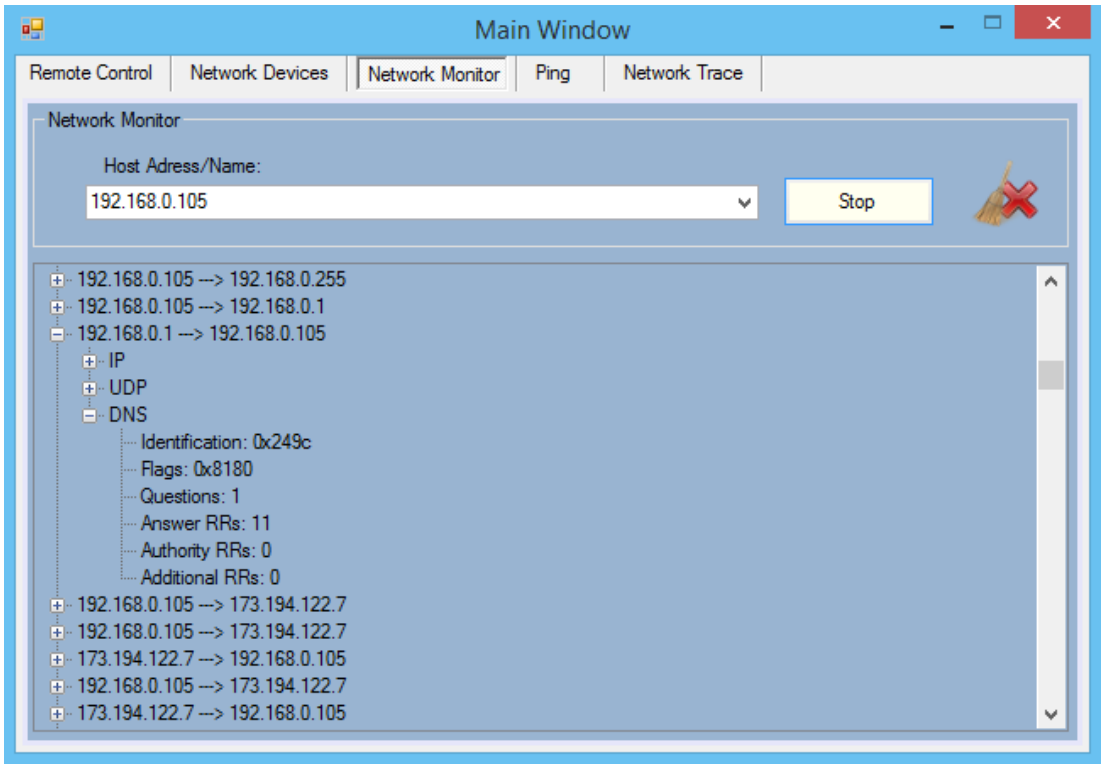


Figure 4.9: Network Monitoring (Showing DNS)

Figure 4.9 shows an example for presenting the section of Domain Name System (DNS). In this example, detailed information about header of packets under section of DNS is shown. As it can be observed in the Figure 4.9, information provided under this section are identification number, flags, number of questions, number of answers Resource Records (RRs), number of authority RRs and number of additional RRs.

4.1.6 Ping Section

A sample part of coding for ping method can be observed in Figure 4.10. This part of coding is divided into two conditions related to the pre-set place for inserting host name or IP address. The first condition is related to being null, which means nothing has been written in text area. In this case, nothing will be appeared in the provided place for gathered information by ping method. Otherwise, the program will start to apply the ping method to the address for 4 times by BeingPingHost method, based on the AsyncCallback value. The result of this part will be displayed in the provided place, which is list item box. The second condition is related to saving the result in the log sheet. If the text area was not null, then the result will be stored in the pre-defined log sheet.

```
private void cmdPing_Click(object sender, System.EventArgs e)
{
    if (txtHostname.Text == null || txtHostname.Text.Length == 0)
        return;

    cancel = false;
    lstResponses.Items.Clear();

    result = netMon.BeginPingHost(new AsyncCallback(EndPing), txtHostname.Text, 4);

    if (result != null)
    {
        cmdCancel.Visible = true;
        writeLog("Ping \t" + txtHostname.Text + "\t- " + System.DateTime.Now.ToString());
    }
}
```

Figure 4.10: Sample part of Ping Method

In the ping section, server's user inserts the desired host's IP address or name in the provided text box and by pressing the Start button, all the information would be displayed in the provided place below. The given information will be about the host

and shows to the server that the selected host is up or down. Figure 4.10 is presenting the section of ping.

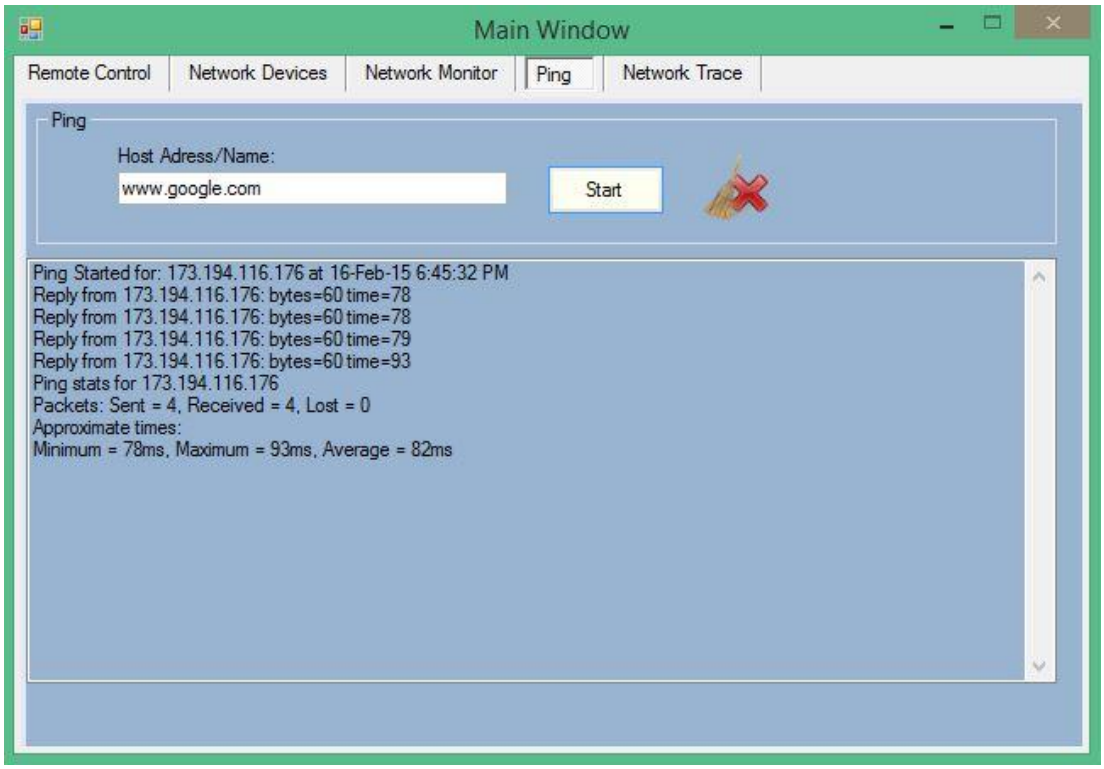


Figure 4.11: Ping Section

The provided example, Shown in Figure 4.11, presents the ping method for Google's website by using its name and all the sending and receiving process of packets for establishing connection between server's device and host, have been shown. Ping performs by sending request packets to the destination host and waits for response. In this process it calculates the time from sending to responding and records any packet loss. The results in this example, are given as information about the minimum, maximum, and the mean time of this process, as well as information about packet loss, packet sent and packet received.

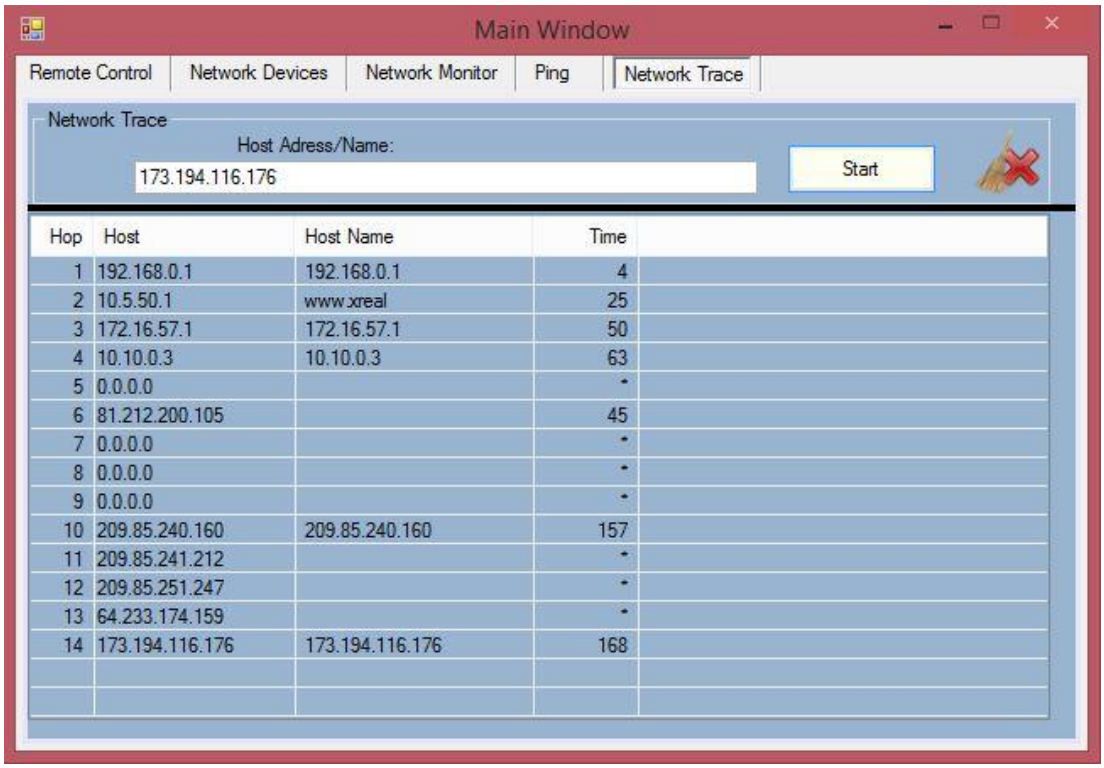
4.1.7 Network Trace Section

The sample code of this part is shown in Figure 4.12. As it can be observed from the sample of code, the IP address or host name, which has been inserted in provided text area, will be achieved. Then the provided item list for displaying the information, will be cleared to be ready for new trace information. In order to trace the network, the route between host and server will be investigated by using tracert.Trace method. The displayed information of this process, will be stored in log sheet.

```
private void startTrace_Click(object sender, EventArgs e)
{
    try
    {
        tracert.HostNameOrAddress = destination.Text;
        routeList.Items.Clear();
        tracert.Trace();
        writeLog("Trace \t" + destination.Text + "\t- " + System.DateTime.Now.ToString());
    }
    catch (SocketException ex)
    {
        MessageBox.Show(ex.Message, "Tracert Demo");
    }
}
```

Figure 4.12: Sample part of Network Trace

As it is shown in Figure 4.12, in this section, the administrator of the server will be able to insert the desired host's IP address in the provided area. By pressing the start button, all the information about the path between server's device and the selected host would be displayed in the provided environment.



Hop	Host	Host Name	Time
1	192.168.0.1	192.168.0.1	4
2	10.5.50.1	www.xreal	25
3	172.16.57.1	172.16.57.1	50
4	10.10.0.3	10.10.0.3	63
5	0.0.0.0		*
6	81.212.200.105		45
7	0.0.0.0		*
8	0.0.0.0		*
9	0.0.0.0		*
10	209.85.240.160	209.85.240.160	157
11	209.85.241.212		*
12	209.85.251.247		*
13	64.233.174.159		*
14	173.194.116.176	173.194.116.176	168

Figure 4.13: Network Trace

Provided example, shown in Figure 4.13, shows the path between server's device and the desired IP address, which in this example is IP address of www.google.com. The information presents the routers and access points, which are in the middle of the path between server's device and Google website. This should be noticed that international routers' IP addresses will not be identified for the security reasons. One of the provided information are IP address of the networking devices in the middle of the path between server and destination. The other information is host name that in case of having any name, the name would be appeared. Time, as the other information, is measured in millisecond. Time is related to the calculated time of receiving packet to the each networking device in the route.

In the section of server side, this should be considered that server's device must have static IP address. In case of having dynamic IP address, in the network monitoring

section, client’s IP address would not be appeared in the dropdown box. The reason is that this system is implemented on the third layer of TCP/IP, which is network layer.

4.2 Client Side

In the client side of this application, as it is presented in Figure 4.14, some options have been provided for client to be connected to the server. By inserting the IP address of the server and also setting the port number, which must be the same as the set port number in server side, client would be able to be connected to the server's system. After this step, client will be able to have conversation with the server. Also, client has the ability of controlling the networking devices information and in case of having any disconnection or fault in the system, some solutions would be provided for troubleshooting, shown in Figure 4.15.

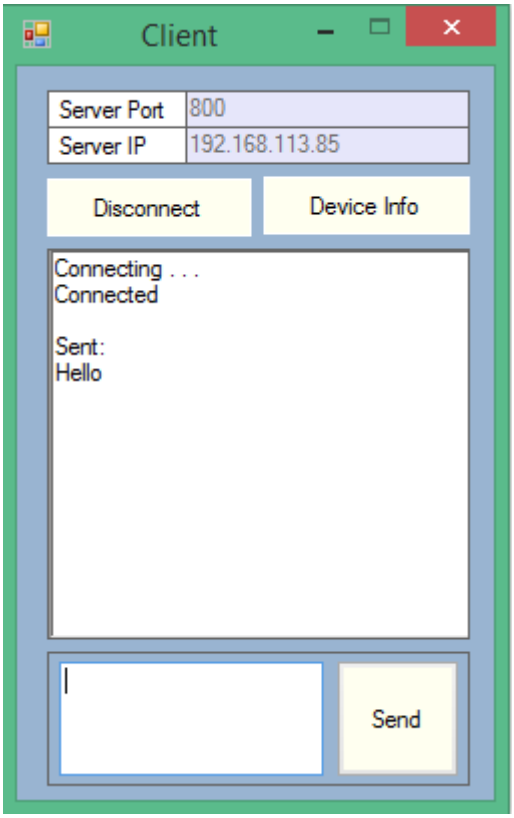


Figure 4.14: Client Side

The provided example, shown in Figure 4.14, shows the sent messages from client to server. The example, shown in Figure 4.15, shows the device information for client side.

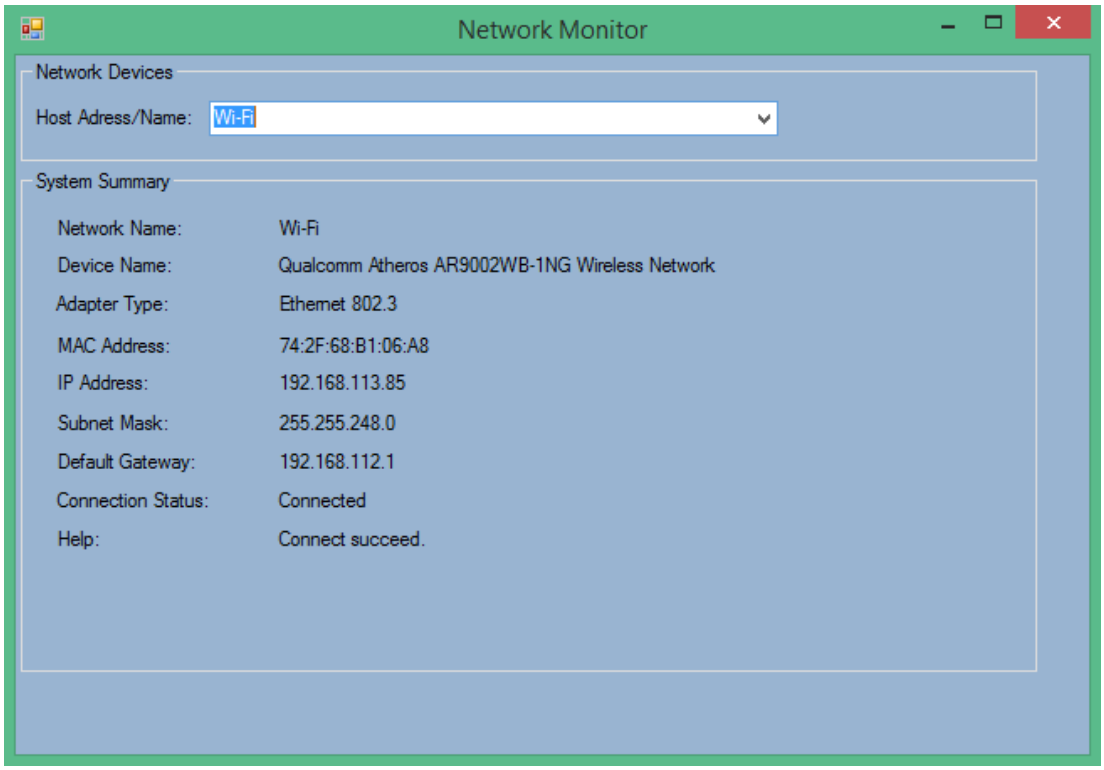


Figure 4.15: Device Information for Client Side

CHAPTER 5

CONCLUSION

Through this study, concept of network monitoring, remote access and other concepts related to this work, had been reflected. Besides that, the similar works which previously had been done in this field, were investigated and comparison between the implemented application and the existing tools had been given. In addition, the process, procedures and conditions of the implemented application, step-by-step, have been thoroughly explained. According to the scope of this thesis, a simplified network monitoring system has been implemented. The researches, which have been done prior to choosing this topic, show that all existing network monitoring tools in the market have complicated user interface if they do not work based on command line structure. Regarding this fact, the existing tools cannot be used by beginners and they are designed for professional users with having strong understanding of network. Regarding the researches on related works, the attempts were to provide a simple but functional network monitoring system to be useful for professional users as well as novice users. The primary purpose of implementing this project was to provide a simple to use network monitoring system which contains most of the necessary functionalities. Besides the mentioned purposes, the main focus of proposed network monitoring system is network security and management. Monitoring and sniffing packets will allow the admin of user to control the security of entire network. The network monitoring system implemented for this thesis, has the capability of being

used by students and novice users. So, this application can be used for educational and training purpose.

As the future work, some features can be added to this application, such as providing statistical report on the result of monitoring and transferring data by using File Transferring Protocol (FTP). Additionally, this application, which has been designed to be run on Windows operating system, might be written with another programming language for being used on other operating systems.

REFERENCES

[1] Englander, I., (2013), The Architecture Of Computer Hardware, Systems Software & Networking, Book, Fourth Edition.

[2] http://www.techopedia.com/definition/20974/network-management

[3] Richard, T. & Watson, (2007), Information Systems, University of Georgia.

[4] Sloman, M. & Jonathan, D., (1994) Policy Conflict Analysis in Journal of Organizational Computing, Vol. 4, No. 1, pp. 1-22.

[5] Trimintzios, P., Polychronakis, M., Papadogiannakis, A., Foukarakis, M., Markatos, E. P. & Oslebo, A., (2006), DiMAPI: An application programming interface for distributed network monitoring, Conference on Network Operations and Management Symposium, IEEE, pp. 382-393.

[6] Fang, W., Zhijin, Z. & Xueyi, Y., (2008), A New Dynamic Network Monitoring Based on IA, International Symposium on Computer Science and Computational Technology, IEEE, Vol. 2, pp. 637 - 640.

[7] Michalski, M., (2009), A Software and Hardware System for a Fully Functional Remote Access to Laboratory Networks, Fifth International Conference on Networking and Services, IEEE, pp. 561 – 565.

[8] Suri, S. & Batra, V., (2010), Comparative Study of Network Monitoring Tools, International Journal of Innovative Technology and Exploring Engineering (IJITEE), Vol. 1, No. 3, pp. 63-65.

[9] http://www.itqlick.com/spiceworks/feedback

[10] Stephen, P., Olejniczak & Kirby, B., (2007), Asterisk for Dummies, chapter 10.

[11] http://www.service-desk.co/pdf/opmanagerproduct-overview.pdf

[12] http://www.networkmanagementsoftware.com/network-management-software-smackdown

[13] Rosenberg, D. & Scott, K. (1999), Use Case Driven Object Modeling with UML: A Practical Approach, Molecular Informatics, Massachusetts, Addison-Wesley.

[14] http://agile.csc.ncsu.edu/SEMaterials/UMLOverview.pdf.

[15] Feldkuhn, L. & and Erickson, J., (1989). Event management as a common functional area of open systems management, Proceedings of the First IFIP Symposium on Integrated Network Management, pp. 365-376.

[16] Sloman, M. (ed.), (1994), Networks and Distributed Systems Management, Addison Wesley Longman Publishing Co., Inc., Boston, MA, USA.

[17] Easley, D. & Jon Kleinberg, (2010), Networks, Crowds, and Markets: Reasoning About a Highly Connected World.

[18] Kaspersky Lab., (2013), Global Corporate IT Security Risks.

APPENDIX

APPENDIX

1. Coding Part for Remote Command in Client Side

```
private void execute_command(String Comande)
        {
            if (Comande.Contains("Disconnect"))
            {
                cmd_con_Click(cmd_con, EventArgs.Empty);
            }

            else if (Comande.Contains("shutdown"))
            {
                System.Diagnostics.Process.Start("shutdown", "-s");
                currentNotice = "Shutdown initiating . . .";
                this.Invoke(new addNotification(addNotice));
                NetworkStream ns = tcpclnt.GetStream();
                String notify = "++--Shutdown Completed . . .";
                if (ns.CanWrite)
                {
                    byte[] bf = new ASCIIEncoding().GetBytes(notify);
                    ns.Write(bf, 0, bf.Length);
                    ns.Flush();
                }
            }

            else if (Comande.Contains("restart"))
            {
                System.Diagnostics.Process.Start("restart", "-r");
                currentNotice = "Restart initiating . . .";
                this.Invoke(new addNotification(addNotice));
                NetworkStream ns = tcpclnt.GetStream();
                String notify = "++--Restart Completed . . .";
                if (ns.CanWrite)
                {
                    byte[] bf = new ASCIIEncoding().GetBytes(notify);
                    ns.Write(bf, 0, bf.Length);
                    ns.Flush();
                }
            }
            else if (Comande.Contains("logoff"))
            {
                System.Diagnostics.Process.Start("logoff", "-l");
                currentNotice = "Logoff initiating . . .";
                this.Invoke(new addNotification(addNotice));
                NetworkStream ns = tcpclnt.GetStream();
                String notify = "++--Logoff Completed . . .";
                if (ns.CanWrite)
                {
                    byte[] bf = new ASCIIEncoding().GetBytes(notify);
                    ns.Write(bf, 0, bf.Length);
                    ns.Flush();
                }
            }
            else if (Comande.Contains("abort"))
            {
```

```
                System.Diagnostics.Process.Start("abort", "-a");
                currentNotice = "Abort initiating . . .";
                this.Invoke(new addNotification(addNotice));

                NetworkStream ns = tcpclnt.GetStream();
                String notify = "++--Abort Completed . . .";
                if (ns.CanWrite)
                {
                    byte[] bf = new ASCIIEncoding().GetBytes(notify);
                    ns.Write(bf, 0, bf.Length);
                    ns.Flush();
                }
            }
            tcpclnt.Close();
            Thread.Sleep(7000);
            Application.Restart();
        }
```

2. Coding Part for Remote Command in Server Side

```
private void cmd_comand_Click(object sender, EventArgs e)
        {
            try
            {
                if (radioButton1.Checked)
                {
                    currentMsg = "*+*-shutdown";
                    sendData(currentMsg, comboBoxIP2.SelectedItem.ToString());
                    currentMsg = "\nShutdown initiating to " +
comboBoxIP2.SelectedItem.ToString() + " Ip";
                    writeLog(currentMsg + "\t- " +
System.DateTime.Now.ToString());
                    this.Invoke(new rcvData(addNotification));
                    recvDt = comboBoxIP2.SelectedItem.ToString();
                    clientDis();

                }
                else if (radioButton2.Checked)
                {
                    currentMsg = "*+*-restart";
                    sendData(currentMsg, comboBoxIP2.SelectedItem.ToString());
                    currentMsg = "\nRestart initiating to " +
comboBoxIP2.SelectedItem.ToString() + " Ip";
                    writeLog(currentMsg + "\t- " +
System.DateTime.Now.ToString());
                    this.Invoke(new rcvData(addNotification));
                    recvDt = comboBoxIP2.SelectedItem.ToString();
                    clientDis();

                }
                else if (radioButton3.Checked)
                {
                    currentMsg = "*+*-logoff";
                    sendData(currentMsg, comboBoxIP2.SelectedItem.ToString());
                    currentMsg = "\nLogoff to " +
comboBoxIP2.SelectedItem.ToString() + " Ip";
                    writeLog(currentMsg + "\t- " +
System.DateTime.Now.ToString());
```

```
                    this.Invoke(new rcvData(addNotification));
                    recvDt = comboBoxIP2.SelectedItem.ToString();
                    clientDis();
                }
                else if (radioButton4.Checked)
                {
                    currentMsg = "*+*-abort";
                    sendData(currentMsg, comboBoxIP2.SelectedItem.ToString());
                    currentMsg = "\nShutdown / Restart aborted to " +
comboBoxIP2.SelectedItem.ToString() + " Ip";
                    writeLog(currentMsg + "\t- " +
System.DateTime.Now.ToString());
                    this.Invoke(new rcvData(addNotification));
                    recvDt = comboBoxIP2.SelectedItem.ToString();
                    clientDis();
                }
            }

            catch (NullReferenceException exp)
            {
                Console.WriteLine(exp.Message);
            }

            Thread.Sleep(2000);
        }
```

2.1 Network Status

```
void Update()
        {
            ManagementObjectSearcher searcher = new
ManagementObjectSearcher("SELECT * FROM Win32_NetworkAdapter WHERE
NetConnectionID IS NOT NULL");
            foreach (ManagementObject mo in searcher.Get())
            {
                try
                {
                    if
(m_Informations.ContainsKey(ParseProperty(mo["NetConnectionID"])))
                    {
                        NetConnectionStatus status =
(NetConnectionStatus)Convert.ToInt32(mo["NetConnectionStatus"]);
                        NetworkInfo info =
m_Informations[ParseProperty(mo["NetConnectionID"])];
                        info.DeviceName = ParseProperty(mo["Description"]);
                        info.AdapterType = ParseProperty(mo["AdapterType"]);
                        info.MacAddress = ParseProperty(mo["MACAddress"]);
                        info.ConnectionID =
ParseProperty(mo["NetConnectionID"]);
                        info.Status = status;
                        if (info.Status != NetConnectionStatus.Connected)
                        {
                            info.IP = "0.0.0.0";
                            info.Mask = "0.0.0.0";
                            info.DefaultGateway = "0.0.0.0";
                        }
                        else
                        {
```

```
                    SetIP(info);
                }
            }
        }
        catch(Exception ex)
        {
            Debug.WriteLine("[Update]:" + ex.Message);
        }
    }
}
```