PROJECT PROPOSAL

Project: Sniff 'N' Stop

Submitted to:

Dr. Faisal Bashir

Submitted by:

NC Veena Safdar(Group Leader)
NC Rabbiya Iftikhar
NC Nadeem Sarwar Abbasi
NC Bilal Haider Naqvi

Name of Project:

Sniff 'N' Stop

Name of Supervisor:

Dr. Faisal Bashir
Co-Supervisor: Bilal Rauf

Name of group leader:

NC Veena Safdar

Group members:

NC Veena Safdar
NC Rabbiya Iftikhar
NC Nadeem Sarwar Abbasi
NC Bilal Haider Naqvi

Domain of the project:

Networking

Stakeholders:	Project Team Members Project Supervisor CS Department, MCS
Purpose/Aim of Project:	Aim of the project is to develop a Distributed Application that will allow a network administrator to monitor the network traffic and also provide remote access to network systems. It will be a network management tool (**network analyzer**) as well a system to **control activities** of network users when needed. It is similar to a distributed **backdoor application** that will **monitor the user's desktop activities** and can **report** all the activities to the system administrator. In case of any illegal activity, the system administrator will be able to take adequate action remotely. This system would be helpful in keeping an eye on the users to prevent any unauthorized activity.
Scope of Project:	Scope of the project is any organization that has its network and needs 'monitoring of network traffic' , 'remote access to all network users' , 'management of network' and 'control over the activities' of its users like university networks(labs) , banks, offices.
Need of Project:	With the advent of computers and the internet in organization and institutes there is a need to closely monitor user's desktop activities. It is not possible to keep a close eye on each and every user. So there is a need of a software that can **manage the network and control the**

	activities in addition to “just observing the network”. Previously the tools available to us were “only observing the network” and not managing them, so we need a tool to: • Analyze the network, • Monitor network traffic • Provide remote access to all network users • Manage it, monitor it, and • Control the activities without interrupting the user in his/her activities (hidden software).
Project Perspective:	Chief deliverable would be a distributed application that can monitor and manage the network, with remote access to network users and can also control activities of each and every user when needed.
Deliverable A:	“Monitoring” module will reside on the user’s system. - It will observe ‘passive monitoring’ meaning it will not modify anything rather will just monitor the activities happening on the user’s system and will just do a live streaming of snapshots of the user desktop. -**Traffic Monitoring:** It will monitor all the incoming/outgoing traffic to check for any unauthorized activities on any IP address.
Deliverable B:	“Device Controller” will deal with the following: - **USB port enabling/disabling:** The USB port(s) can be disabled/enabled by the administrator from a central location. -**System Shutdown:** The user’s system(s) can be safely shutdown anytime by the administrator. -**Keyboard/mouse lock:** The keyboard/mouse (s) can be locked anytime to put a curb on illegal activities.
Deliverable C:	“Network Analyzer” does the following functions: - **Protocol filter:** It will check all the protocol traffic and can filter any protocol if desired. -**IP Tracker (OPTIONAL) :** It will deal with the routing between source and destination and vice versa and will also calculate the total hops between the two. -**Format Converter (OPTIONAL) :** The packets which are filtered are in Hexadecimal format and this service will change it into textual form.
Operating system:	The application will be capable of running on any version of MS Windows Operating System.
Tools & technologies:	Development Tools: - C # - MS Visual Studio -

Nature of Application:	The application will be a distributed application.

Table of Contents

CHAPTER 1

INTRODUCTION

1.1 Introduction:

Sniff n Block is a Distributed Application that allows a network administrator to monitor the network traffic and also provides remote access to network systems through a graphical user interface (GUI). This application is basically prepared to assist the network administrator to monitor the traffic and also to assist in conduction of quizzes in labs. The system follows a client server architecture i.e. two tier architecture and runs simultaneously at two ends i.e. client and server. The overall system follows the push model i.e. a system where server initiates a session; whereas the application at the client end would be hidden from the user. At the server end the user would get the list of all the nodes connected to it and via GUI he/she can select the various options made available by the system to assist better administration. The various functions available are:

- Port blocking
- Screen shot viewing
- Remote shutdown
- Packet sniffing
- Packet filtering

1.2 Purpose

Aim of the project is to develop a Distributed Application that allows a network administrator to monitor the network traffic and also provide remote access to network systems. It is a network management tool (network analyzer) as well a system to control activities of network users when needed. It is similar to a distributed backdoor application that monitors the user’s desktop activities and can report all the activities to the system administrator. In case of any illegal activity, the system administrator will be able to take adequate action remotely. This system would be helpful in keeping an eye on the users to prevent any unauthorized activity.

1.3 Scope

Sniff 'n' Block is a system that can be implemented on any network. Its purpose is to effectively monitor user activities over the network and the desktop. So it provides the network administrator the authority to control user hardware and network. That is because Sniff 'n' Block can block hardware such as keyboard, mouse and USB ports remotely. It can key an eye on the user activities by regularly reporting his desktop activities to the network administrator.

The application can be used in any organization that has its network and needs ‘monitoring of network traffic’ , ‘remote access to all network users’ , ‘management of network’ and ‘control over the activities’ of its users like university networks(labs) , banks, offices.

Sniff 'n' Block besides monitoring also provides security by not allowing the users to stop the application from monitoring and reporting him to the network administrator. These functionalities are analyzed in the later chapters.

1.4 Sniff n Block (Product Perspective)

The system is a client server form of a Distributed application which is hidden on client side and provides backdoor information to the network administrator. The overall system is composed of three main modules and nine sub modules (eight of which have been implemented). A pictorial representation has been shown in Figure 1.4-1 that shows the overall system hierarchy and all the modules.

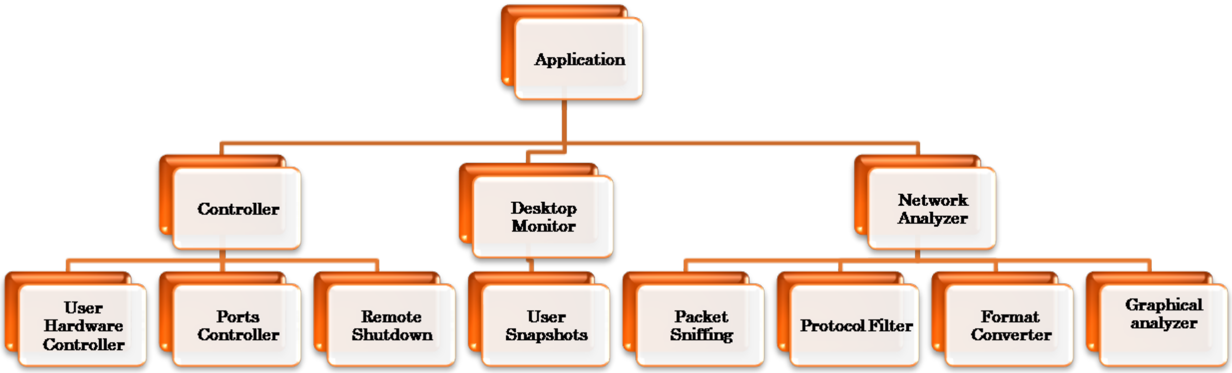


Figure 1.4-1

Sniff 'n' Block consists of a server and several clients. The server is controlled by the system administrator. Each client is controlled by the network administrator. The system administrator has an interface with a number of options to take action on the client. The administrator can block hardware such as keyboard or mouse of any client. He has the option of remotely shutting down any or all the computers on the network without closing his own sessions.

He can block USB ports of clients if he wants to. There is also the ability of viewing client's desktops remotely. System administrator can sniff client's network traffic and can filter specific protocols and then sniff packets if he wants to.

We can see three levels in the Figure 1 shown above. At the top most level is the application that itself consists of three modules that are:

1. Controller
2. Network monitor
3. Network analyzer

At level 3 the modules at level 2 have been decomposed to sub modules which are covered in detail below:

1. **Controller:**

   a. User Hardware Controller
   b. Ports Controller
   c. Remote shutdown

a. User Hardware Controller:

The keyboard/mouse (s) can be locked anytime to put a curb on illegal activities.

b. Port Controller:

The USB port(s) can be disabled/enabled by the administrator from a central location.

c. Remote Shutdown:

The user's system(s) can be safely shutdown anytime by the administrator.

2. Network monitor

a. User snapshots

a. User snapshots

It will observe 'passive monitoring' meaning it will not modify anything rather will just monitor the activities happening on the user's system and will just do a live streaming of snapshots of the user desktop.

3. Network analyzer

a. Protocol filter
b. Format converter
c. Packet sniffing
d. Graphical analyzer

a. Protocol filter

It will check all the protocol traffic and can filter any protocol if desired.

b. Format converter

The packets which are filtered are in Hexadecimal format and this service will change it into textual form.

c. Packet sniffing

It will monitor all the incoming/outgoing traffic to check for any unauthorized activities on any IP address.

d. Graphical analyzer

It graphs the captured protocols according to frequency of usage.

1.5 Intended Audience and Reading Suggestions

This requirement document contains all the information about Sniff ‘n’ Block, overview, scope, system architecture, system design, main classes and use cases, functions, features and special technologies. It describes in detail that entire Sniff ‘n’ Block needs to work properly and with safety.

The rest of the document is divided into chapters for better understanding.

Chapter 2: literature review

Chapter 3: SRS (Software Requirement Specification)

Chapter 4: Software Design

Chapter 5: Implementation

Chapter 6: Results and analysis

Chapter 7: Testing

Chapter 8: Graphical User Interface

Chapter 9: Conclusion and future work

This document is intended for

Developers: in order to be sure they are developing the right project that fulfills requirements provided in this document.

Testers: in order to have an exact list of the features and functions that have to respond according to requirements and provided diagrams.

Users: in order to get familiar with the idea of the project and suggest other features that would make it even more functional.

Documentation writers: to know what features and in what way they have to explain. What security technologies are required, how the system will response in each user's action etc.

Advanced end users, end users/desktop and system administrators: in order to know exactly what they have to expect from the system, right inputs and outputs and response in error situations.

CHAPTER 2

LITERATURE REVIEW

2.1 Distributed Application:

2.1.1 Definition:

Software that executes on two or more computers in a network. In a client-server environment, distributed applications have two parts: (1) the 'front end' that requires minimal computer resources and runs on the client computer(s), and (2) the 'back end' that requires large amounts of data crunching power and/or specialized hardware, and runs on a suitably equipped server computer.

2.1.2 Properties:

The following defining properties are commonly used:

1. There are several autonomous computational entities, each of which has its own local memory.
2. The entities communicate with each other by message passing.

A distributed system may have a common goal, such as solving a large computational problem. Alternatively, each computer may have its own user with individual needs, and the purpose of the distributed system is to coordinate the use of shared resources or provide communication services to the users. Other typical properties of distributed systems include the following:

1. The system has to tolerate failures in individual computers.
2. The structure of the system (network topology, network latency, number of computers) is not known in advance, the system may consist of different kinds of computers and network links, and the system may change during the execution of a distributed program.
3. Each computer has only a limited, incomplete view of the system. Each computer may know only one part of the input.

2.1.3 What happens in a distributed application

Distributed application programs have multiple parts that are on different virtual machines. The different virtual machines can be on the same or different systems. Distributed application programs, in cooperation with CRR, use protected conversations to ensure distributed transaction resource integrity.

Figure 2.1 illustrates user's distributed application program that consists of three parts. Each part is distributed among three virtual machines. Part #1 communicates with part #2 by means of a protected conversation (PC #1 in the figure). Part #2 communicates with part #3 by means of a protected conversation (PC #2 in the figure). Also, there are two protected resources on both VM1 and VM2.

The sequences of events in this example are:

1. Distributed application program (part #1) updates (by means of the resource adapter) both protected resources on VM1.
2. Distributed application program (part #1) starts a protected conversation (shown as PC #1) to distributed application program (part #2) on VM1.
3. Distributed application program (part #2) updates (by means of the resource adapter) both protected resources on VM1.
4. Distributed application program (part #2) starts a protected conversation (shown as PC #2) to distributed application program (part #3) on VM2.
5. Distributed application program (part #3) updates (by means of the resource adapter) both protected resources on VM2.
6. Distributed application program (part #1) issues a commit.
7. The SPM in distributed application program's (part #1) virtual machine starts sync point processing by preparing the two protected resources on VM1 and tells distributed application program (part #2) to issue a commit.
8. Distributed application program (part #2) issues a commit.
9. The SPM in distributed application program's (part #2) virtual machine starts sync point processing by preparing the two protected resources on VM1 and tells distributed application program (part #3) to issue a commit.
10. Distributed application program (part #3) issues a commit.
11. The SPM in distributed application program's (part #3) virtual machine starts sync point processing by preparing the two protected resources on VM2.
12. CRR commits all the prepared work at all the protected resources and reports this result to all three parts of the distributed application program.

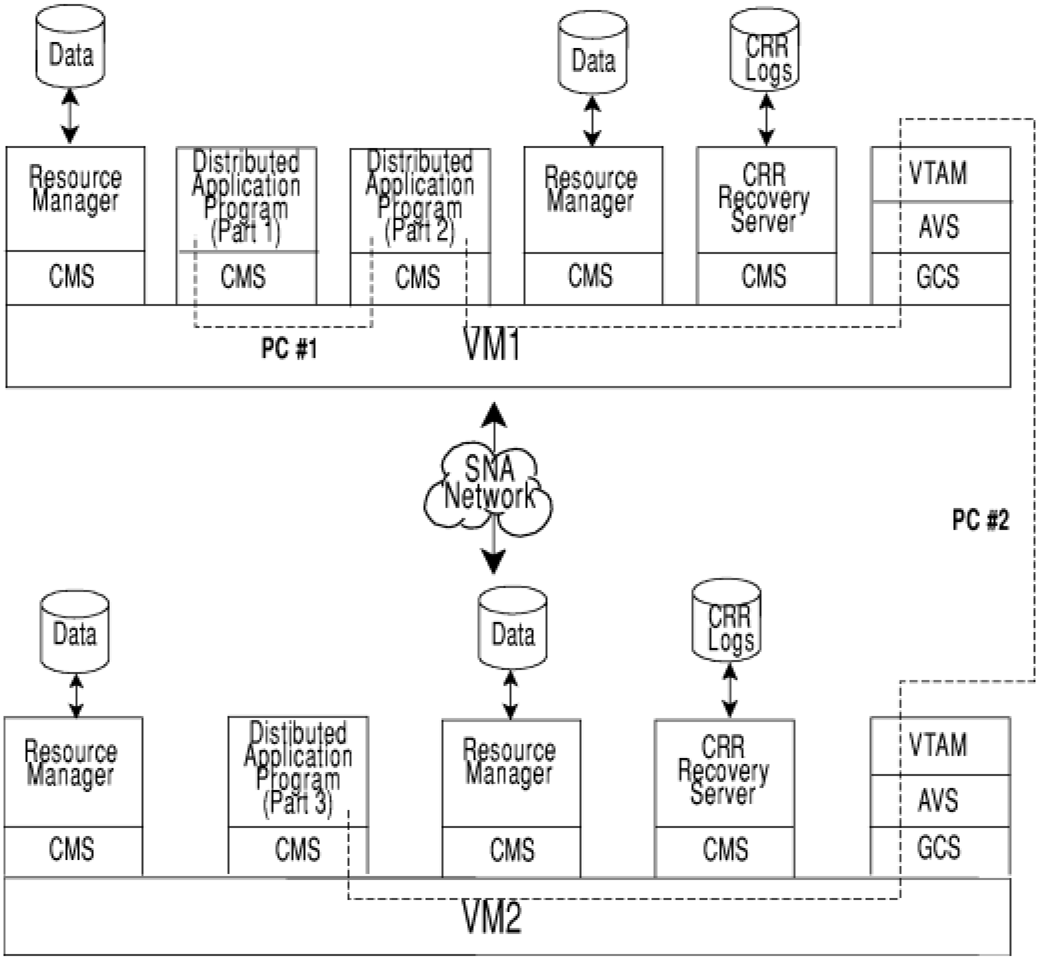


Figure 2.1-1

2.1.4 Architectures:

Various hardware and software architectures are used for distributed computing. At a lower level, it is necessary to interconnect multiple CPUs with some sort of network, regardless of whether that network is printed onto a circuit board or made up of loosely-coupled devices and cables. At a higher level, it is necessary to interconnect processes running on those CPUs with some sort of communication system.

Distributed programming typically falls into one of several basic architectures or categories:

- client–server
- 3-tier architecture
- *n*-tier architecture
- distributed objects
- loose coupling, or tight coupling.

Client–server:

Smart client code contacts the server for data then formats and displays it to the user. Input at the client is committed back to the server when it represents a permanent change.

3-tier architecture:

Three tier systems move the client intelligence to a middle tier so that stateless clients can be used. This simplifies application deployment. Most web applications are 3-Tier.

***n*-tier architecture:**

n-tier refers typically to web applications which further forward their requests to other enterprise services. This type of application is the one most responsible for the success of application servers.

Tightly coupled (clustered):

Typically refers to a cluster of machines that closely work together, running a shared process in parallel. The task is subdivided in parts that are made individually by each one and then put back together to make the final result.

Peer-to-peer:

An architecture where there is no special machine or machines that provide a service or manage the network resources. Instead all responsibilities are uniformly divided among all machines, known as peers. Peers can serve both as clients and servers.

Space based:

This refers to an infrastructure that creates the illusion (virtualization) of one single address-space. Data are transparently replicated according to application needs. Decoupling in time, space and reference is achieved.

Another basic aspect of distributed computing architecture is the method of communicating and coordinating work among concurrent processes. Through various message passing protocols, processes may communicate directly with one another, typically in a master/slave relationship. Alternatively, a "database-centric" architecture can enable distributed computing to be done without any form of direct inter-process communication, by utilizing a shared database.

2.2WIRE SHARK:

2.2.1 INTRODUCTION:

It is a network packet analyzer, otherwise known as a ‘packet sniffer’. It can capture and decode packets of information from a network. A network packet analyzer will try to capture network packets and tries to display that packet data as detailed as possible. You could think of a network packet analyzer as a measuring device used to examine what's going on inside a network cable. Network administrators use it to *troubleshoot network problems*. Wire Shark can capture live network traffic or read data from a file. Network analyzers such as

Wire Shark are invaluable tools for network administrators to diagnose and troubleshoot problems with but it can also be used by intruders to obtain unauthorized information.

Wire shark is packet-centric. Data can be captured "from the wire" from a live network connection or read from a file that records the already-captured packets. Captured network data can be browsed through a GUI or the command line version of the utility.

The Wire shark strength comes from:

- Its easiness to install.
- The simplicity of use of its GUI interface.
- The very high number of functionality available

2.2.2 WHAT DOES WIRESHARK DO?

It has a graphical front-end. Wire shark can be used to capture and analyze network packets and discover an array of information like:

- Troubleshooting network issues
- Log network traffic etc.

It can also be used for more nefarious purposes by intruders or attackers like capturing usernames and passwords, capturing sensitive information.

2.2.3 SUPPORTED PROTOCOLS:

Wire shark supports almost 700 protocols more than most even know exist. Because it is open source, new drivers that let Wire shark decode and translate different protocols, are created regularly as users have a need for them. For that reason, the list of supported protocols grows on a regular basis.

2.2.4 LIMITATIONS:

There are some things that Wire shark does not provide namely Wire shark isn't an intrusion detection system. It will not warn you when someone does strange things on your network that he/she isn't allowed to do.

2.2.5 WIRESHARK AND OUR SYSTEM:

So in our proposed system, we will not only capture and analyze packets but will also build on it so that in case of any suspicious activity it will provide a warning message.

2.3 TROJAN:

2.3.1 Introduction:

A **Trojan horse** (sometimes shortened to ***Trojan***), is non-self-replicating malware that appears to perform a desirable function for the user but instead facilitates unauthorized access to the user's computer system.

Trojan Horses are a type of computer program that appear to perform a certain function, but actually perform other, often malicious, actions. They differ from traditional viruses because they do not have the means to automatically replicate. Trojan horses can be classified based on how they breach and damage systems. They are primarily used for unauthorized remote access, mass-mailing spam, data destruction, file transfer, toll-line dialing, and denial-of-service attacks.

It is a malware that presents itself as a useful program but actually does extensive damage by allowing unauthorized access to your computer.

Today, Trojans are notorious for implementing backdoor programs that allow attackers unauthorized access to victims' systems. Unlike traditional computer viruses and other types of malware, they are designed with the intention of exploiting infected systems, rather than replicating themselves or disabling a system completely. However, in order for an attacker to gain access to a system, the victim must run an executable server file. These are usually spread through email attachments, peer-to-peer file sharing systems, and Internet downloads. These remote-access servers arrive unbeknownst to the end-user, disguised as important, useful, or desirable files or documents. When executed, these server programs open backdoors that allow attackers to remotely access and control the infected system.

2.3.2 Working of Trojan:

Trojans spread over the Internet through a number of ways, such as through emails, chat programs, and the download of files that may actually contain beneficial material but also include concealed Trojans. Once these files are opened or executed, the malicious program is installed on your computer. And once installed, the program will run automatically every time your computer is turned on.

Also, many Trojans also incorporate a worm that accesses your email addresses and sends them a message with the Trojan attachment. Malicious hackers, also called crackers, can create a network of zombie computers through this worm. This network of zombie computers, also called botnets, can then be used to spread even more Trojans throughout the network. They are called zombies because their users rarely know the computers are infected.

As soon as an infected computer is powered up, Trojan sends it IP Address to the attacker. This allows the attacker to communicate with the infected computer and access its files or even erase them.

Trojans can log every keystroke you type (even when you're offline) and have your e-mail program send the information to the person who planted the Trojan without your knowing it. Trojans can get all your passwords, credit card numbers and other information stored on your computer - or even things that you type into the computer and don't save. They can be used to read, delete or change all your files, turn your screen upside down, abruptly disconnect you from the Internet, or direct your browser to only certain web sites and other nuisances. It gets worse - Trojans can be used to spy on you through your chat and instant message programs, web cam or microphone, and even destroy your hardware.

They can damage your reputation as well as your hardware and data. Trojans can be used to get into your address book and send very convincing looking e-mails saying whatever someone else likes from you to your employer, bank manager, clients, whomever, and they can make you seem to say really awful things to people in on-line chats or conferences.

2.3.3 Classification of Trojan:

Trojan horses are broken down in classification based on how they breach systems and the damage they cause. The seven main types of Trojan horses are described below:

- **Remote Access Trojans**

  Abbreviated as RATs, a Remote Access Trojan is one of seven major types of Trojan designed to provide the attacker with complete control of the victim's system. Attackers usually hide these Trojan in games and other small programs that unsuspecting users then execute on their PCs.

- **Data Sending Trojans**

  A type of a Trojan that is designed to provide the attacker with sensitive data such as passwords, credit card information, log files, e-mail address or IM contacts lists. These Trojans can look for specific pre-defined data (e.g., just credit card information or passwords), or they could install a key logger and send all recorded keystrokes back to the attacker.

- **Destructive Trojans**

  A type of Trojan horse designed to destroy and delete files, and is more like a virus than any other Trojan. It can often go undetected by antivirus software.

- **Proxy Trojans**

  A type Trojan horse that makes the proxy not to work properly by destroying cache logs and making to take longer time to respond.

- **FTP Trojans**

  A type of Trojan designed to open port 21 (the port for FTP transfer) and lets the attacker connect to your computer using File Transfer Protocol (FTP).

- **Security software disabler Trojans**

  A type of Trojan designed stop or kill security programs such as an antivirus program or firewall without the user knowing. This Trojan type is normally combined with another type of Trojan as a payload.

- **Denial-of-service attack (DoS) Trojans**

  Short for ***d**enial-**o**f-**s**ervice attack,* a type of attack on a network that is designed to bring the network to its knees by flooding it with useless traffic. Many DoS attacks, such as the *Ping of Death* and *Teardrop* attacks, exploit limitations in the TCP/IP protocols. For all known DoS attacks, there are software fixes that system administrators can install to limit the damage caused by the attacks. The piece of code to initiate the DoS attack is usually in the form of DoS attack Trojans.

2.3.4 Types of Trojan:

Trojan Horse viruses do not refer to a single type. It actually refers to a class of malware that can inflict damage on a computer or network of computers. Technically they are not viruses, but their activities lead them to being classified as such.

- **Torpig :** Torpig is a dangerous Trojan Horse that can destroy valuable data on your PC.
- **Rustock.C:** Rustock.C is a Trojan that has affected thousands of computers worldwide.

- **Mocmex:** The Mocmex Trojan can cause serious system damage or loss of vital information.
- **MacSweeper** : MacSweeper is a potentially dangerous program that can inflict damage on your Mac system.
- **Koobface** : The Koobface virus can infect social networking sites, leading to data loss.
- **Conficker :** Conficker is one of the most widespread computer worms in the world today.
- **Bohmini.A :** A system with the Bohmini.A Trojan can become unstable and lead to serious problems.

2.3.5 Trojan and our System:

Trojans can be far more malicious than viruses and you should care - they're programs that let someone else remotely administer your computer without your knowing about it. Our aim is to design a legitimate system that does this, that can be used by systems administrators to administer networks. Unlike Trojans, our system will be secure and will not be having any destructive effect.

Our system is designed to analyze the network traffic of the users, monitor their desktops and all ongoing activities without their knowledge (hidden on user side). It can also control their hardware, ports remotely and can also shutdown the systems remotely, but unlike Trojan our system is not wasting the target system’s memory, not harming it.

2.4 Introduction to client server architecture:

The peer to peer system has worked well locally and even across the Internet as the file sharing systems have shown. But there were several issues that have not been handled well, including security, scalability, and privileged access. Those issues forced the development of a client-server system.

As you can see, peer-to-peer is still used quite a bit, but only for file sharing. Client-Server networking is a different story. Business enterprises have other needs, especially when it comes to managing large data files. Their processing and distribution operations are large. Also security with authentication, bandwidth control, and load balancing of operations from server to server cannot be done in a peer-to-peer environment.

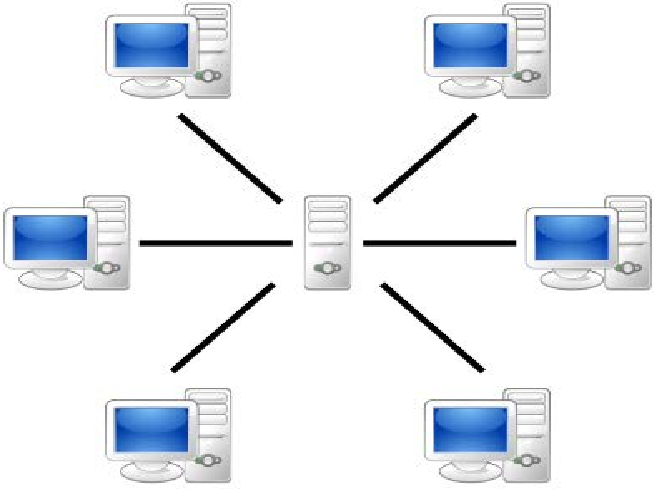


Figure 2.4-1

Image Courtesy: Wikipedia

In a client-server environment there is a central point of management and there is also the push-down operation from server to client where data is pushed down to the client where the client performs operations and the results are sent back up to the server. This minimizes the use of the server, and it allows other clients to perform their individual tasks without taxing the central server, which will act as a repository of information. Here the management of operations is performed at the central site level, while the actual operations are performed at the client level. Here are some of the other elements of client-server administration.

Security administration

An important feature in a client-server network is security administration. Here a user must authenticate her credentials. Users cannot access the network unless they are legitimate users. This security management separates a peer-to-peer network from client-server, because authentication at P2P is not necessary.

Microsoft and Domains

On a Microsoft client-server network domains are created. These are administered with a special program called Active Directory. It controls the users, the computers, the passwords, and provides other security features.

Other Client-Server Networks

Microsoft is not the only company that provides client-server computing. Unix, Linux, Novell, and Sun Microsystems all produce servers for a client-server environment.

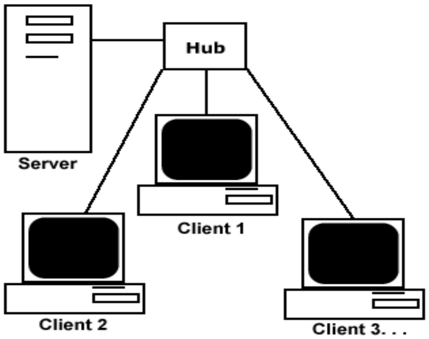


Figure 2.4-2

2.4.1 Advantages and Disadvantages of Client-Server Networks

Here are some of the advantages of client-server networking.

Start with the fact that it is centralized. Central control of your resources is provided, and computers, users, and passwords are under the control of the central system. Another feature is the ability to grow or have the flexibility of scalability features; this is important. Networks tend to grow by adding more users or more computers. The network servers have to be flexible. New software can be added to improve operations. But also, adding additional hardware components provide the capability to make the network load balanced where processing can be redirected to several clients. Network servers see network clients or the ability to interoperate. They work together. Finally they are accessible. Server do not work in isolation and can be accessed remotely.

Now here are the disadvantages.

For starters they are expensive. Special software is required and a dedicated server must be isolated as the central controller. Part of the cost is the expertise required for maintenance. Expertise is required to run client-server networks. As they grow larger they will require more staff. Finally, when the central server goes down the rest of the network will go down. So there is a dependence on the central server that is not present with peer to peer.

2.4.2 Summary

Unlike peer-to-peer (P2P) networks which are stand alone computers connected with cross-over cables or to a hub with straight-through cables, client-server networks take very different administration elements to work well. Staff, software, and a special server are part of the components to make them work.

Client-server networks, however, scale well, provide authentication features for security, and they are flexible in the use of software and operations that they provide. They allow the network to operate more efficiently.

CHAPTER 3

Software Requirement Specification (SRS)

3.1 Introduction

3.1.1 Purpose

This *section* document includes software re*quirements for Sniff 'n' Block.* Sniff 'n Block is a two-tier client server based network application. The system gives provides solution to monitoring network users and managing network. Its purpose is to monitor all of the user’s desktop activities and data. The system can be implemented anywhere where there is a network of computers that requires users to be monitored. It provides several functionalities like hardware controlling, using monitoring and network control.

3.1.2 Document Conventions

- When writing this *section* it was inherited that all requirements have the same priority.
- First there is presented an overall view about Sniff ‘n’ Block and then all features and functions are analyzed in detail.

3.1.3 Project Scope

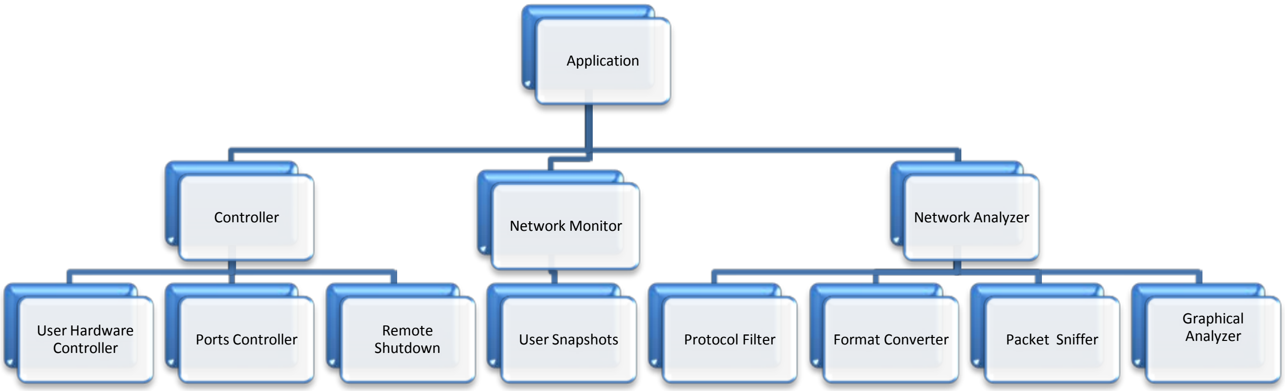
Sniff 'n' Block is a system that can be implemented on any network. Its purpose is to effectively monitor user activities over the network and the desktop. So it provides the network administrator the authority to control user hardware and network. That is because Sniff 'n' Block can block hardware such as keyboard, mouse and USB ports remotely. It can key an eye on the user activities by regularly reporting his desktop activities to the network administrator.

Sniff 'n' Block besides monitoring also provides security by not allowing the users to stop the application from monitoring and reporting him to the network administrator. These functionalities are analyzed in the following pages.

3.2 Overall Description

3.2.1 Product Perspective

Sniff ‘n’ Block consists of a server and several clients. The server is controlled by the system administrator. Each client is controlled by the network administrator. The system administrator has an interface with a number of options to take action on the client. The administrator can block hardware such as keyboard or mouse of any client. He has the option of remotely shutting down any or all the computers on the network without closing his own sessions. He can block USB ports of clients if he wants to. There is also the ability of viewing client's desktops remotely. System administrator can sniff client's network traffic and can filter specific protocols and then sniff packets if he wants to.



3.2.2 Product Features

Sniff n' Block provides the user with the following functions:

- **Hardware Control – Keyboard, Mouse**

  The network administrator can block/unblock client's keyboard and mouse remotely from the server. The administrator chooses to the client he wants to block from the list. Then he chooses what to block/unblock.

- **Port Blocking**

  Sniff 'n' Block allows the administrator to block the client USB ports remotely instead of doing it manually for each system.

- **System Shutdown -- Remotely**

  The clients system can be shutdown remotely by the administrator when needed.

- **Snapshots**

  In Sniff 'n' Block there's the option of viewing users desktop activities without have to go to him. By taking the regular snapshots of the user's desktop and sending them over the network, the administrator can know what's actually happening at the other end.

- **Packet Sniffing**

  The network administrator can select any client and know about his network traffic but checking the packets at his network interface.

- **Protocol Filtering**

  The administrator has the authority to select any protocol from any client's network and check the packets.

- **Security**

  Sniff ‘n’ Block provides security by remaining hidden and inaccessible from clients and not allowing them to stop it. So the application is ready to take action on administrators request anytime.

- Graphic analyzer:

  It takes the protocols from captured file and graphs those according to frequency of their usage.

3.3 User Classes and Characteristics

- **End users/Desktop:** users with no particular knowledge on computer programming.
- **System administrators:** administrators working on computers that support a lot of accounts and personal data for other users. Using Sniff ‘n’ Block the administrator control all the client systems.

3.4 Operating Environment

Sniff ‘n’ Block should run on Operating Systems: WINE, 32-bit MS Windows (98), 32-bit MS Windows (NT/2000/XP), All 32-bit MS Windows (95/98/NT/2000/XP),Win2K, WinXP, Microsoft Windows Server 2003. The user interfaces used are: NET, Win32 (MS Windows)

3.5 Design and Implementation Constraints

Timing requirements in Sniff ‘n’ Block:

When a password is copied for any reason, (e.g. copy to an application, account, and website) it remains in the memory for only 10 seconds. After 10 seconds pass there is nothing to paste and you have to recopy again. That provides security in a case a password is copied and not pasted anywhere so no one can find it out by pasting later.

Language Requirements in Sniff 'n' Block:

Not in all translations translated help files and tutorials are available.

Specific Technologies used in Sniff 'n' Block:

- In order to keep the user's data fully protected, 2 very secure algorithms are used:

Cipher	**Block Size**	**Key Size**
Advanced Encryption Standard (AES / Rijndael)	128 bits	256 bits
Twofish	128 bits	256 bits

In both algorithms every time the user saves a database, a random 128-bit initialization vector is generated.

- For the creation of the 256-bit key the Cipher uses, the Secure Hash Algorithm SHA-256 is used.
- All the bytes needed for the Initialization Vector, the master key salt, etc are generated via pseudo-random sources: current tick count, performance counter, system date/time, mouse cursor position, memory status, active window focus handles, window message stack, process heap status, process startup information and several system information structures.
- When the Sniff 'n' Block is active, all passwords are stored encrypted in process memory so in order for them to be completely safe the ARC4 encryption algorithm is used, using a random 12 bytes long key.

3.6 System Features

3.6.1 Controller:

3.6.1.2 Description:

It consists of three sub modules each having its own specialized purpose.

- User Hardware Controller
- Ports Controller
- Remote shutdown

Using **User Hardware Controller** the keyboard/mouse (s) can be locked anytime to put a curb on illegal activities. The USB port(s) can be disabled/ enabled by the administrator from a central location using P**ort Controller**. The user's system(s) can be safely shutdown anytime by the administrator using R**emote Shutdown functionality.**

3.6.1.2 Stimulus/Response Sequences:

Data Flow

3.6.1.2.1 Basic Data Flow

1. User selects the controller option of the other options available.
2. User gets a list of functions made available by this module.
3. User selects the option.

3.6.1.2.2 Alternative Data Flows

3.6.1.2.2.1 Alternative Data Flow 1

3. User selects block keyboard for particular system

4. The keyboard gets blocked

3.6.1.2.2.2 Alternative Data Flow 2

3. User selects block mouse for particular system

4. The mouse gets blocked

3.6.1.2.2.3 Alternative Data Flow 3

3. User selects block ports for particular system

4. The USB ports gets blocked

3.6.1.2.2.4 Alternative Data Flow 4

3. User selects remote shutdown for particular system

4. The system gets shutdown.

3.6.1.3 Functional Requirements:

REQ-1: the system should support remote shut down

REQ-2: the system should support remote hardware control such as mouse, keyboard and port enable/disable

3.6.2 Network monitor

3.6.2.1 Description:

It consists of two sub modules each having its own specialized purpose.

- User snapshots

Using **User snapshots** sub module user will just monitor the activities happening on the user's system and will just do a live streaming of snapshots of the user desktop.

3.6.2.2 Stimulus/Response Sequences:

Data Flow

3.6.2.2.1 Basic Data Flow

1. User selects the network monitor option of the other options available.

2. User gets a list of functions made available by this module.

3. User selects the option.

3.6.2.2.2 Alternative Data Flows

3.6.2.2.2.1 Alternative Data Flow 1

3. User selects snapshot viewing for particular system

4. The snapshots after each fixed interval can be viewed

3.6.2.3 Functional Requirements

REQ-1: The client system allows network administrator to monitor desktop activities.

REQ-2: supports client network monitoring activities.

3.6.3 Network analyzer

3.6.3.1 Description:

It consists of three sub modules each having its own specialized purpose.

- Protocol filter
- Format converter
- Packet sniffing
- Graphical analyzer

Using Protocol filter the user can check all the protocol traffic and can filter any protocol if desired. IP tracker (optional) deals with the routing between source and destination and vice versa and will also calculate the total hops between the two. Format converter changes the contents of packet into textual form. Using **Packet sniffing** user will monitor all the incoming/outgoing traffic to check for any unauthorized activities on any IP address.

3.6.3.2 Stimulus/Response Sequences:

Data Flow

3.6.3.2.1 Basic Data Flow

1. User selects the network analyzer option of the other options available.
2. User gets a list of functions made available by this module.
3. User selects the option.

3.6.3.2.2 Alternative Data Flows

3.6.3.2.2.1 Alternative Data Flow 1

3. User selects Protocol filter for particular system

4. The user will get all the protocol traffic and can filter any protocol if desired.

3.6.3.2.2.2 Alternative Data Flow 2

3. User selects format convertor option for particular system

4. User gets the translated version of hexadecimal packets into textual format.

3.6.3.2.2.3 Alternative Data Flow 3

3. User selects packet sniffing option for particular system

4. User gets all the incoming/outgoing traffic of the system.

3.6.3.2.2.4 Alternative Data Flow 4

3. User selects the graphing option

4. User gets the graphic representation of protocols captured

3.6.3.3 Functional Requirements

REQ-1: It prevents client from disabling the monitoring service.

3.7 External Interface Requirements

3.7.1 User Interfaces:

The application will navigate the user through a number of user friendly interfaces. The user at the GUI end should be a computer literate as to understand the different states the application will be at any given time due to the user’s commands like error messages, connection establishment messages. However the error messages should be clear enough that a new user will understand without any doubt what the error message was about.

The application will use a graphical user interface (GUI) to support its operation. It provides an interface to get all the client computers connected to the application. There will also be an interface provided to display the desktop activity information coming in from other computers. It will also provide an interface to show the information coming in from the clients’ network cards. More interfaces can be developed and integrated if required by the user/system.

3.7.2 Hardware Interfaces

From NIC the program captures packets by first identifying the network adapters on the computer. Then it gives the option of choosing the network adapter. After the adapter is decided it captures buffer parameters and then chooses the capture mode. Finally it captures the packets for the specified amount of time.

3.7.3 Software Interfaces

Packet X has been used as an interface to capture packets; since the library used to capture packets is Win PCAP to make it compatible with the developing language Packet X has been used.

3.8 Other Non-functional Requirements

3.8.1 Performance Requirements

The system must be available during lab quizzes. It must be available during the lab timings. The system must be able to block any client's hardware on request and monitor any clients desktop activities upon request from the administrator. The system must be able to filter protocols on request. It should be able to shutdown any client on request. The system must be able to provide user's network activities information. The system must not consume too much network bandwidth. The system user interface must be user friendly. The system must be able to quickly recover from network failures.

3.8.2 Safety Requirements

The system should be in compliance with the network policy of the department. It should be hidden on client's side and the options available to administrator should not harm any other processes on the server side.

3.8.3 Security Requirements

To ensure security the administrator has to enter password before making use of this application.

3.8.4 Software Quality Attributes

The system should be available whenever required as well as maintainable, adaptable and flexible. It should be robust to failures and should be having attributes of usability.

Appendix A: Glossary

Class Diagram: A UML Static structure diagram that describes the structure of a system by showing the system's classes, their attributes and the relationships between the classes.

Data Flow Diagram (DFD): A graphical description of the flow of data through an information system.

Use Case: A description of a system's behavior as it originates to a request that originates from outside that system.

Client/Server: Client/server describes the relationship between two computer programs in which one program, the client, makes a service request from another program, the server, which fulfills the request.

Network Card (Network Interface Controller): A **network interface card**, **network adapter**, **network interface controller** (**NIC**), or **LAN adapter** is a computer hardware component designed to allow computers to communicate over a computer network.

Pushdown Model: A client server interaction in which server initiates the connection.

Winpcap (**P**acket **Cap**ture) consists of an application programming interface (API) for capturing network traffic.

Packet X: An adapter to make C# compatible with Win Pcap

Device Controller: It will deal with the following:

i)USB port Enable/disable

ii) System Shutdown

iii) Keyboard/mouse lock

Monitor: It will observe passive monitoring and will monitor the activities on the user's computer through 'packet sniffer' and does streaming of the snapshots of the user desktop.

Network Analyzer: It will constitute the 'protocol filter' which will check the protocol traffic and filter any protocol if required.

Keyboard hook: Hook is a mechanism by which a function can intercept events before they reach an application

Mouse Hook: Hook is a mechanism by which a function can intercept events before they reach an application

Win Pcap: This is the packet Capturing and Network monitoring library for Windows

Packet X Wrapper: It is a wrapper for the Win Pcap packet capture library. Wrapper translates one interface for a class into a compatible interface.

USB port Enable/ Disable: It will deal with the enabling or disabling of USB ports.

System Shutdown: This will deal with two scenarios, shutdown immediately without prompting the user and shutdown after a specific period of time.

Keyboard/mouse lock: It stops the mouse and the keyboard from receiving messages from the OS and hence 'locks' the devices.

Packet Sniffer: It monitors the traffic on the user's computer.

CHAPTER 4

Software Design

4.1 Introduction

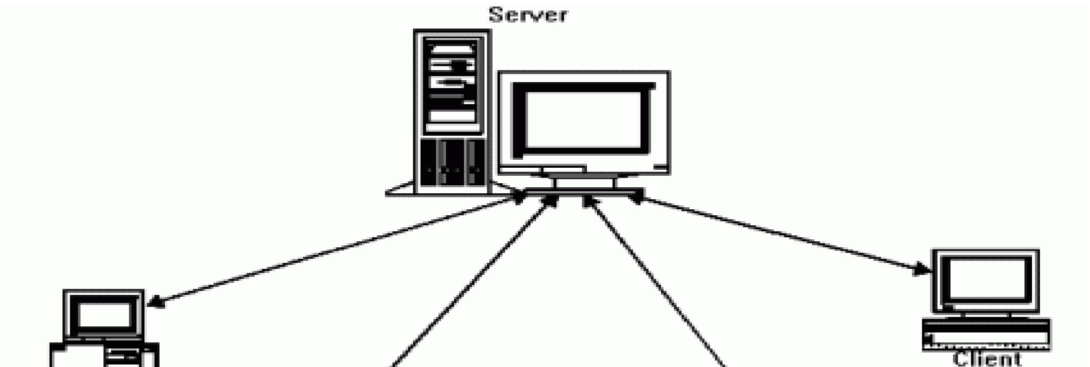
This section provides a software design description for the software tool 'SNIFF N STOP'. This portion of the document shows the detailed internal design of our proposed system and also describes the requested behavior of the system. This documentation describes what is needed by the system user as well as the requested properties of inputs and outputs. This chapter focuses on what various outside agents (e.g. people using the program, or other computers) might 'observe' when interacting with the system. This describes what has to be implemented and what the program aims to achieve. While testing is performed, the behavior of the program is compared against the expected behavior as defined in the functional specification.

The audience of this section is the users and developers of the 'SNIFF N STOP' tool.

4.2 System Architecture

4.2.1 CLIENT SERVER ARCHITECTURE:

The basic definition of a client is that it requests for services and servers provide services. It is up to the server to decide how to get the job done. A client may become a server and a server may become a client. The Client/Server paradigm has been used by programs executing on the same computer, but typically an application is written that executes on one computer and request information from an application running on a separate computer. The network provides the mechanism used to interconnect programs that are distributed across different location. The application is modeled as a set of services that are provided by the servers and a set of clients that use these services. We will use two-tier architecture



4.2.2 System Diagram:

Sniff ‘n’ Block is divided into three main modules as seen in the figure. The three modules namely:

1. Controller
2. Network Monitor
3. Network Analyzer

Provide the functionality to monitor the client’s network and desktop activities combined. The system administrator monitors and controls the activities of client’s through these modules. We can see three levels in the diagram given.

At level 3 the modules at level 2 have been decomposed to sub modules which are covered in detail below:

4.2.2.1 Controller:

The first module, the Controller, is assigned the responsibility to control client’s hardware upon request from the user. The Controller modules is subdivided into three more components, which are

a. User Hardware Controller
b. Ports Controller
c. Remote Shutdown

a. User Hardware Controller:

The keyboard/mouse (s) can be locked anytime to put a curb on illegal activities.

b. Port Controller:

The USB port(s) can be disabled/ enabled by the administrator from a central location.

c. **Remote Shutdown:**

The user's system(s) can be safely shutdown anytime by the administrator.

4.2.2.2 Network monitor

As the name suggests, the Network Monitor will monitor user's network activities. It will also monitor the user's desktop activities and will report them to the administrator. It contains to sub modules

a. User snapshots

a. **User snapshots**

It will observe 'passive monitoring' meaning it will not modify anything rather will just monitor the activities happening on the user's system and will just do a live streaming of snapshots of the user desktop.

4.2.2.3 Network analyzer

This module consists of:

a. Protocol filter
b. Format converter
c. Packet sniffing
d. Graphical analyzer

a. **Protocol filter**

Protocol filter will filter different protocols running on the client and will be able to see the information required from the required protocol.

b. Format converter

The packets which are filtered are in Hexadecimal format and this service will change it into textual form.

c. **Packet sniffing**

It will monitor all the incoming/outgoing traffic to check for any unauthorized activities on any IP address.

d. Graphical analyzer

It graphs the captured protocols according to frequency of usage.

4.3 Data Dictionary

Device Controller	It will deal with the following: i) USB port Enable/disable ii) System Shutdown iii) Keyboard/mouse lock
Monitor	It will observe passive monitoring and will monitor the activities on the user’s computer through ‘packet sniffer’ and does streaming of the snapshots of the user desktop.
Network Analyzer	It will constitute the ‘protocol filter’ which will check the protocol traffic and filter any protocol if required.

Keyboard hook	Hook is a mechanism by which a function can intercept events before they reach an application
Mouse Hook	Hook is a mechanism by which a function can intercept events before they reach an application
Win Pcap	This is the packet Capturing and Network monitoring library for Windows
PacketX (Wrapper)	It is a wrapper for the Win Pcap packet capture library. Wrapper translates one interface for a class into a compatible interface.
USB port Enable/ Disable	It will deal with the enabling or disabling of USB ports.
System Shutdown	This will deal with two scenarios, shutdown immediately without prompting the user and shutdown after a specific period of time.
Keyboard/mouse lock	It stops the mouse and the keyboard from receiving messages from the OS and hence ‘locks’ the devices.
Packet Sniffer	It monitors the traffic on the user’s computer.

4.4 Data Design

4.4.1 Class Diagram

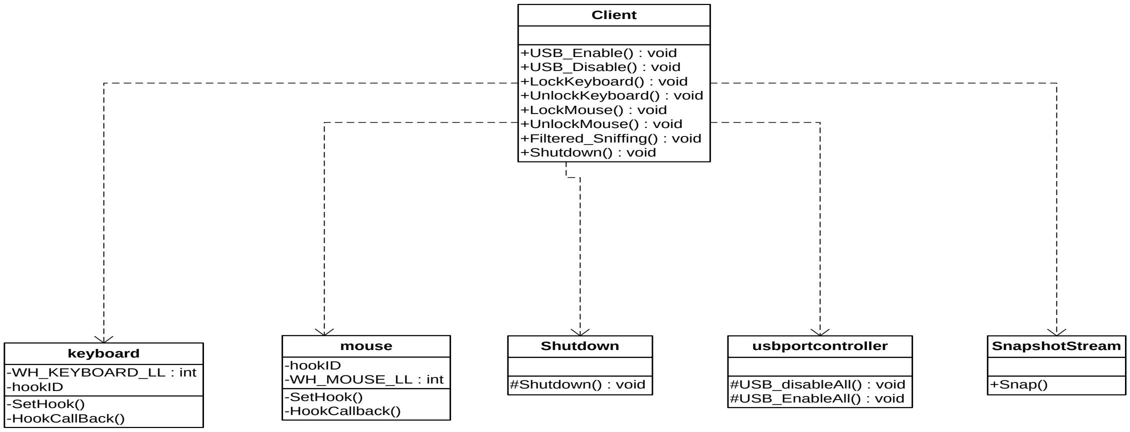


Figure 4.4.1-1

Figure 4.4.1-1 is class diagram of client with functionalities implemented in separate classes and the relationship between classes is weak aggregation. There are no class diagram for server because there are no classes.

4.4.2 Data Flow:

This figure 4.4.2-1 shows the process of capturing of data packets from the Network card of a particular computer system.

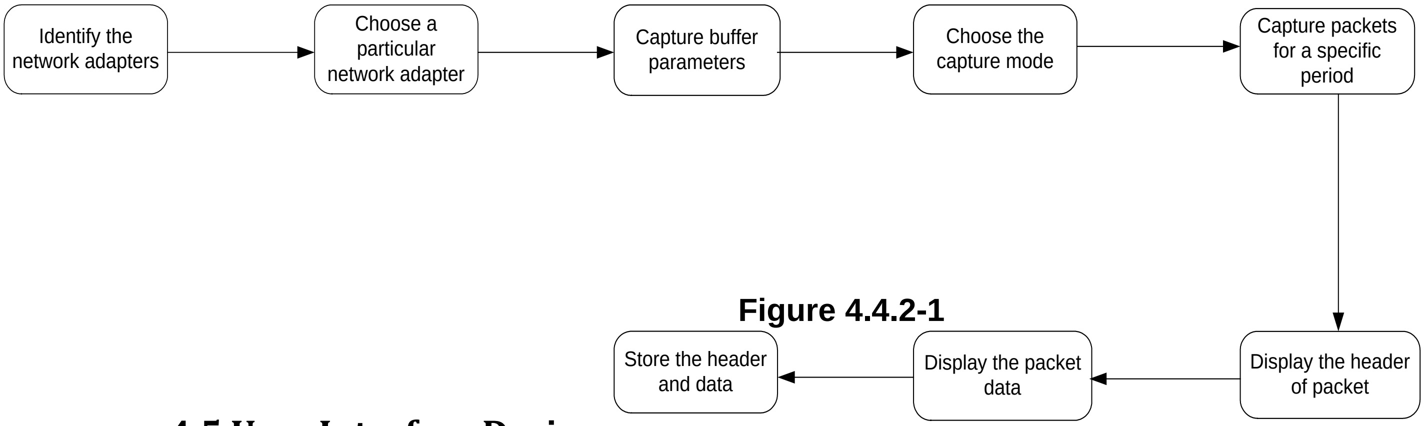

Identify the network adapters
Choose a particular network adapter
Capture buffer parameters
Choose the capture mode
Capture packets for a specific period
Display the header of packet
Display the packet data
Store the header and data

Figure 4.4.2-1

4.5 User Interface Design

4.5.1 User Interface Design Overview

Many technological innovations rely upon User Interface Design to elevate their technical complexity to a usable product. Technology alone may not win user acceptance and subsequent marketability. The User Experience, or how the user experiences the end product, is the key to acceptance. And that is where User Interface Design enters the design process. So besides technology engineers have to focus on interface, because products usability can be judged by the extent to which its interface is user friendly.

User interface for Sniff n Block allows the user to access four major features of the system that are:

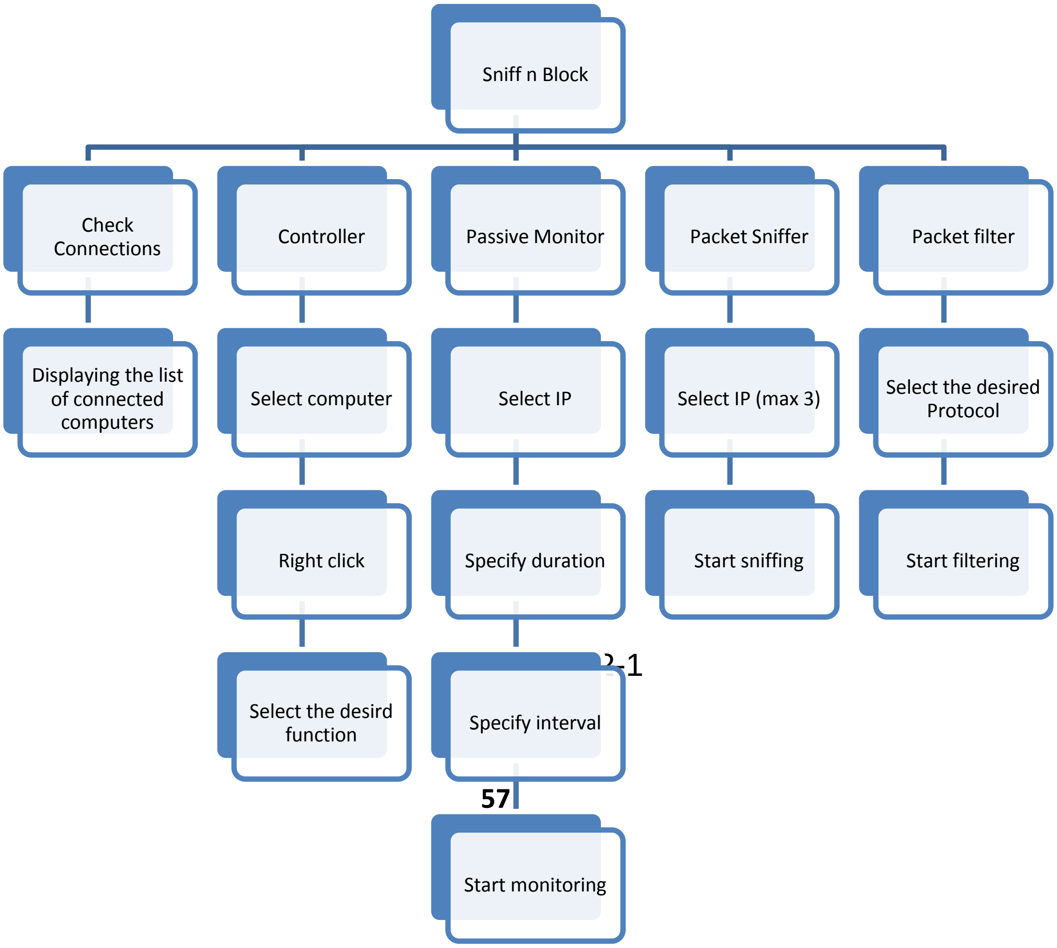
- Controller
- Passive monitor
- Packet sniffer
- Packet filter

And in addition a ‘View List’ feature that will show the connectivity of computers with it, after that user can decide which feature he wants to use. After selecting the controller feature user have to select the IP from displayed list then right click the selected IP and select the desired function to perform like blocking hardware, USB port or system shutdown. In passive monitor feature, after selecting IP user have to specify the duration and interval and then start. In the sniffer feature he has to select IP, here he can select maximum of 3 IPs , but not more than this because of some efficiency issues, and then start sniffing.

For the filtering feature, a pre-condition is that some packets must have been sniffed only then user can filter them. So filtering option is also given in the screen displaying the sniffed packets. User can go back to the main menu using back button.

4.5.2 User Interface Navigation Hierarchy

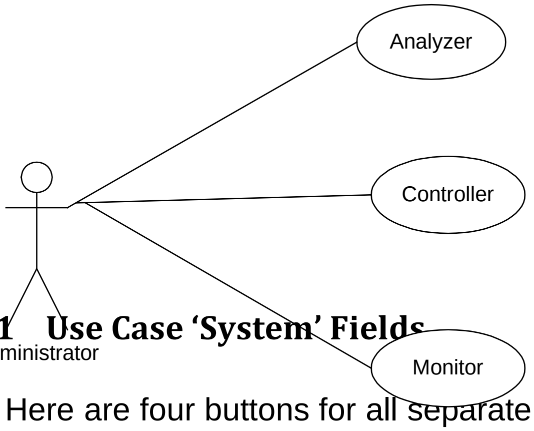
Here is the navigation hierarchy that illustrates that how user will move through the user interface of Sniff n Block (figure 4.5.2-1).



2-1

4.6 User Function Categories (or Use Cases)

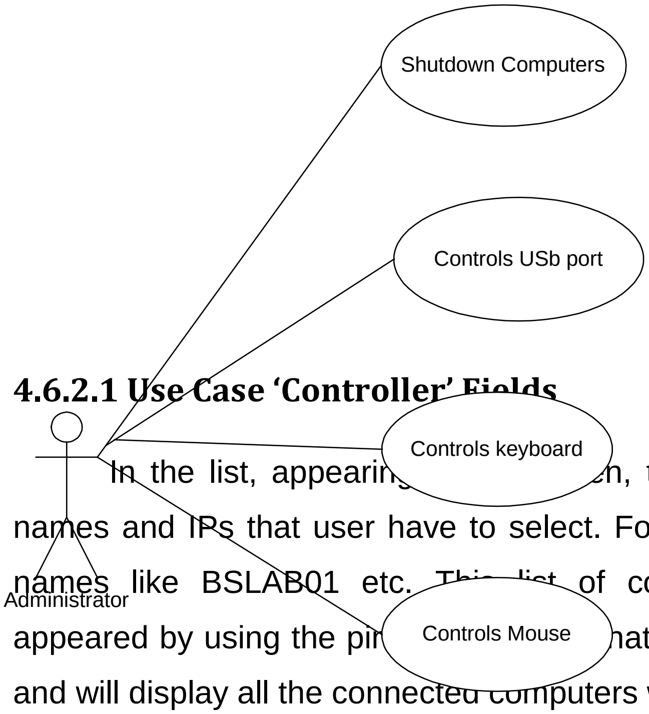
4.6.1 Use case 'System'



4.6.1.1 Use Case 'System' Fields

Here are four buttons for all separate features and one for checking connections, that will display the list of connected computers is the empty field shown in the screen.

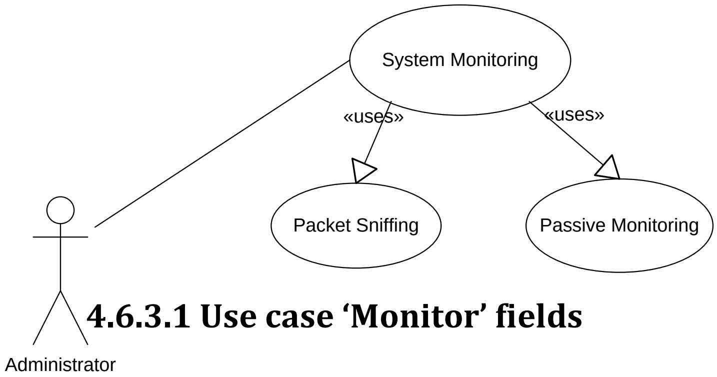
4.6.2 Use case 'Controller'



4.6.2.1 Use Case 'Controller' Fields

In the list, appearin en, there will be list of computer names and IPs that user have to select. For example, 111.222.33.0 and names like BSLAB01 etc. This list of connected computers will be appeared by using the pi hat will check for the connection and will display all the connected computers with their IPs.

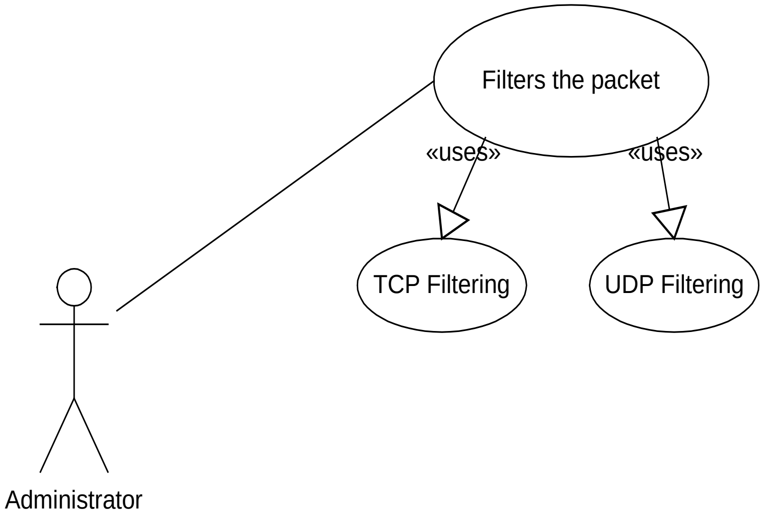
4.6.3 Use case 'Monitor'



4.6.3.1 Use case 'Monitor' fields

Here are two screens for the monitoring module. Fields here are the list of computers and IPs, and in passive monitor screen three more fields for, selected IP, duration (entered by user), and interval (specified by user).

4.6.4 Use case 'Analyzer'



4.6.4.1 Use case 'Analyzer' Fields

Here are two data fields one is to be filled by user to specify the desired protocol, and in the next tabular field it will show the filtered packets.

4.7 Other Interfaces

4.7.1 Network Interface Card

From NIC the program captures packets by first identifying the network adapters on the computer. Then it gives the option of choosing the network adapter. After the adapter is decided it captures buffer parameters and chooses the capture mode. Finally it captures the packets for the specified amount of time.

4.8 References

Document No.	Document Title	Date	Author
1.	Applying UML and Patterns(An introduction to Object-Oriented Analysis and Design)		Craig Larman
2.	Software Engineering: A practitioner's approach		Roger S. Pressman
3.	Software Engineering: Theory and Practice		Shari Lawrence P fleeger

4.9 Glossary

Class Diagram: A UML Static structure diagram that describes the structure of a system by showing the system's classes, their attributes and the relationships between the classes.

Data Flow Diagram (DFD): A graphical description of the flow of data through an information system.

Use Case: A description of a system's behavior as it originates to a request that originates from outside that system.

Client/Server: Client/server describes the relationship between two computer programs in which one program, the client, makes a service request from another program, the server, which fulfills the request.

Network Card (Network Interface Controller): A **network interface card**, **network adapter**, **network interface controller** (**NIC**), or **LAN adapter** is a computer hardware component designed to allow computers to communicate over a computer network.

CHAPTER 5

SYSTEM IMPLEMENTATION

5.1 Introduction:

This section of the document contains the implementation details of the system. It describes the functionality of the system and also an explanation of the system from the implementation point of view. The basic idea is to make the reader familiar with the implementation details of the system so that he can have some idea about the actual working of the system.

5.1.1 Overview:

Our application is composed of a server that receives connection requests from clients. The server is built with an asynchronous TCP/IP socket. As soon as the server application starts working, it starts listening for any incoming client requests. It maintains a separate thread to continuously listen for any client requests coming from any IP address which is part of the network. As soon as a connection request is received, the server initiates a new connection between itself and the requesting machine on which the client application is running. The server side application then issues commands to the client(s). The client side application contains the implementation for 'controller', 'monitor' and 'network analyzer'. The 'Controller' module can remotely shutdown a client(s), enable/disable mouse and keyboard, and enable/disable the USB ports. 'Monitor' transmits the information regarding user desktop activity on the client side to the server side. 'Network Analyzer' takes the information gained from the clients' network card which can be converted into textual form and also filtered according to a specified network protocol. All this can be used by a system administrator to monitor and control the users in a network.

5.2 Controller Module

5.2.1 Overview:

The first module ‘Controller’ is assigned the responsibility to control client’s hardware upon request from the user. It is further subdivided into three components: Hardware Controller, USB port Controller and Remote shutdown.

5.2.2 Explanation:

After the connection has been established, the user at the server end can issue a command to enable/disable the keyboard or mouse, enable /disable the USB ports of the client computer or to remotely shutdown the client computer. The entire implementation takes place on the client side and once the server has issued a command, there will be no output to be transmitted to the server side. The logic has entirely been kept on the client side.

WMI protocol has been used to shutdown the local or remote computer in a LAN. Windows Management Instrumentation (WMI) provides access to information about objects in a managed environment. Using System.Management Namespace provides access to a rich set of management information about the system, devices, and applications instrumented to the Windows Management Instrumentation (WMI) infrastructure. A computer can be shutdown using System.Management classes, but the required security privileges are needed.

Global mouse and keyboard hooks have been used to enable/disable mouse and keyboard respectively. In order to set a low level mouse and keyboard hook, P/Invoke Windows API is used. SetWindowsHookEx specifies the type of hook procedure to be installed. WH_KEYBOARD_LL or WH_MOUSE_LL is used here. When we want to stop mouse or keyboard hook, we can use the UnhookWindowsHookEx function to remove the hook procedure.

For enable/disable of the USB ports, registry values have been modified. This is useful for mass storage devices, not for mouse or keyboard connected to the USB bus.

5.3 Monitor:

5.3.1 Overview

It will monitor the user's desktop activities and will report them to the administrator. It is composed of the user desktop activity monitor

5.3.2 Explanation:

As soon as connection has been established between the server application and the client, the user can issue a command to operate the user desktop activity monitor. The command will go to the client side and the client application will take a image of the desktop in jpg format, convert it into bytes and transfer it to the server side. Using System.Drawing Namespace is required to take a screenshot of the entire screen of the client desktop. The server will take the bytes, convert it back into image form and display it in a picture box so that they can be displayed at the rate they were received. The server will receive the images after a certain interval of time and this activity will continue until told to stop.

5.4 Network Analyzer:

5.4.1 Overview:

It is comprised of packet sniffer and protocol filter on the client side and format converter on the server side. The packet data extracted from the network card is in hex decimal form which is not readable for a common user so it needs to be converted to textual form to make it legible. The protocol filter deals with separating the network information according to protocols.

5.4.2 Explanation:

In packet sniffing, after receiving the command from the server, data packets are captured from the network card of a particular client(s) and then that information containing the header of the packet and the packet data is sent on to the server.

The server takes a network protocol as input from the user, for which the packets are to be extracted. It captures the packets of the required protocol and saves them to a file. The files are sent to the server and the result is displayed on the server side. Winpcap library is used in the protocol filter.

WIN PCAP:

In the field of computer network administration, **pcap** (**p**acket **cap**ture) consists of an application programming interface (API) for capturing network traffic.

Monitoring software may use WinPcap to capture packets travelling over a network and, in newer versions, to transmit packets on a network at the link layer, as well as to get a list of network interfaces for possible use with WinPcap.

WinPcap also supports saving captured packets to a file, and reading files containing saved packets; applications can be written, using WinPcap, to be able to capture network traffic and analyze it, or to read a saved capture and analyze it, using the same analysis code. A capture file saved in the format that WinPcap use can be read by applications that understand that format.

WinPcap provides the packet-capture and filtering engines of many open source and commercial network tools, including protocol analyzers (packet sniffers), network monitors, network intrusion detection systems, traffic-generators and network-testers. Due to in compatibility of winPcap with C# Packet X has been used as a wrapper. Packet X is a small COM class library that makes it easy to use WinPcap packet capture functionality from almost any modern programming language supporting ActiveX technology.

PacketX uses WinPcap Packet Driver API implemented by packet.dll and BPF filtering support from pcap.dll. This means that you can use PacketX to capture, send (and optionally filter) packets and collect network statistics. However PacketX cannot be used to block network traffic to build a firewall.

In case of format converter, it takes the sniffed packets, which are stored in a file, as input and separates TCP/IP and UDP packets. It extracts information like IPversion, Packet Length, Packet Identifier, TTLProtocol, IP check sum, Source and destination IP, MAC, port nos. etc.

CHAPTER 6

Results and Analysis

6.1 Introduction:

In this section of the document, we analyze our application and then compare it with applications having similar attributes. This doesnot include testing techniques rather it just gives the analysis.

6.2 Analysis:

In comparison to various other network analysis tools, Sniff n' Block provide additional features of remotely controlling client's hardware, USB ports and remote shutdown of complete network computers, individual computer or particular group of computers, and remote desktop monitoring.

Sniff n' Block is a network analyzer tool as well as remotely monitors and controls the network computers. Network analyzer and Remote desktop monitor are usually separate applications but in our application we have combined the features of both. There are many tools available for sniffing the network traffic like Wireshark, Tcpdump, Ntop, NGrep all of them are just sniffing tools and don't provide the remote controlling and monitoring features that Sniff n’ Block does provide.

In Sniff n’ Block desktop activity of users can be shown on the server side. It gets images of desktop activity of client(s) after a certain interval of time. It shows which computer(s)/client(s) are connected at one time with the server along with their IP addresses. It can store groups of clients in file on the server side to be accessed later if wanted and also there is provision for selecting multiple clients or a single client at a time for sending commands. It can group together users belonging to a certain network (when we have users belonging to more than one network) in a list through GUI. Whenever a client disconnects it is intimated to the server user.

6.2.1 Comparison with Wire Shark

In comparison to **Wireshark**, which allows to examine data from a live network, Sniff n’ Block not only provide the analysis of network traffic and filtering of protocol but also remotely controls and monitors the network computers. Sniff n’ Block is hidden on client side and act as a backdoor application as opposed to Wireshark. Wire Shark is visible to user and is initiated by user to start sniffing whereas Sniff n’ Block is initiated by administrator on server side that can sniff, control and monitor the network users without telling them or interrupting their routine work. Wire shark supports a lot of protocols but it does not decode information unlike Sniff n’ block which decodes the information from hex to textual format.*Sniff n’ Block* has a user friendly, clear and easy interface. In addition tool tips are present to support the user at the server side whereas Wire Shark may confuse the average user who is not familiar with programming.

6.2.2 Comparison with Tcpdump

Tcpdump is a sniffer used for network monitoring. It doesn't have a user friendly interface to help the users that are not programmers or don't have sufficient technical knowledge whereas Sniff n' Block is easy to use and understand. There is no provision of remote controlling and monitoring the network computers in this tool that is being provided by Sniff n' block.

CHAPTER 7

Testing And Validation

7.1 Introduction

Software testing is one of the most crucial phases of software development life-cycle. This can be termed as an element of a broader topic that is referred to as ‘Verification and Validation’ (V&V). Verification refers to the set of activities that ensure that software correctly implements a specific function. Validation refers to the different set of activities that ensures that the software that has been built is traceable to customer requirements. We performed white box testing on our system.

7.2 Validation and Verification

Validation and verification is intended to be a systematic and technical evaluation of the system and its processes.

Sniff n Block was tested for validation by giving it different intervals as inputs and getting the captured results as desired outputs. In verification testing it was assured that software meets all functional, behavioral, and performance requirements.

7.3 System Testing

In software testing phase overall system is tested as a whole. System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. The purpose of integration testing is to detect any inconsistencies between the software units that are integrated together. System testing is a more limiting type of testing; it seeks to detect defects within the system as a whole.

7.4 White Box Testing

White box testing (a.k.a. clear box testing, glass box testing, transparent box testing, or structural testing) is a method of testing software that tests internal structures or workings of an application as opposed to its functionality (black box testing). An internal perspective of the system, as well as programming skills, are required and used to design test cases. The tester chooses inputs to exercise paths through the code and determine the appropriate outputs. It is analogous to testing nodes in a circuit, e.g. in-circuit testing (ICT).

While white box testing can be applied at the unit, integration and system levels of the software testing process, it is usually done at the unit level. It can test paths within a unit, paths between units during integration, and between subsystems during a system level test. Though this method of test design can uncover many errors or problems, it might not detect unimplemented parts of the specification or missing requirements.

In white box testing we have used code **Statement Coverage.** In this type of testing the code is executed in such a manner that every statement of the application is executed at least once. It helps in assuring that all the statements execute without any side effect.

All the test data provided for a single and multiple nodes yielded the desired result.

CHAPTER 8

Graphical User Interface (GUI)

8.1 Walk through of the application

Sniff ‘n’ Block is a very simple system, which is designed with the concern to be user friendly, and ease of use. The application consists of two parts but user can interact only on the server side as application is disguised on the client side. Effort has been made to keep the application simple and user friendly thus providing the ease of use.

When the application starts the following window appears:

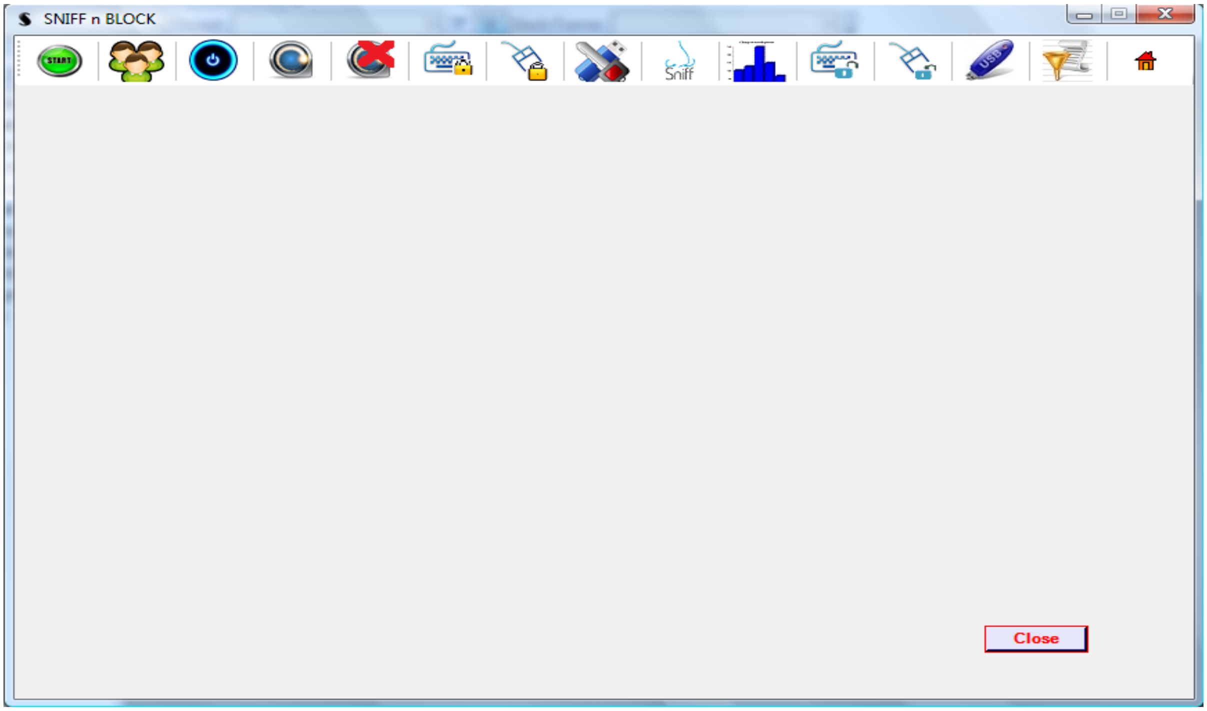


Figure 8.1-1

On the top different icons can be seen each of which corresponds to different function and also a close button at the right bottom to close the application.

1. To start user has to press the start button as shown in the figure 8.1-1 given below

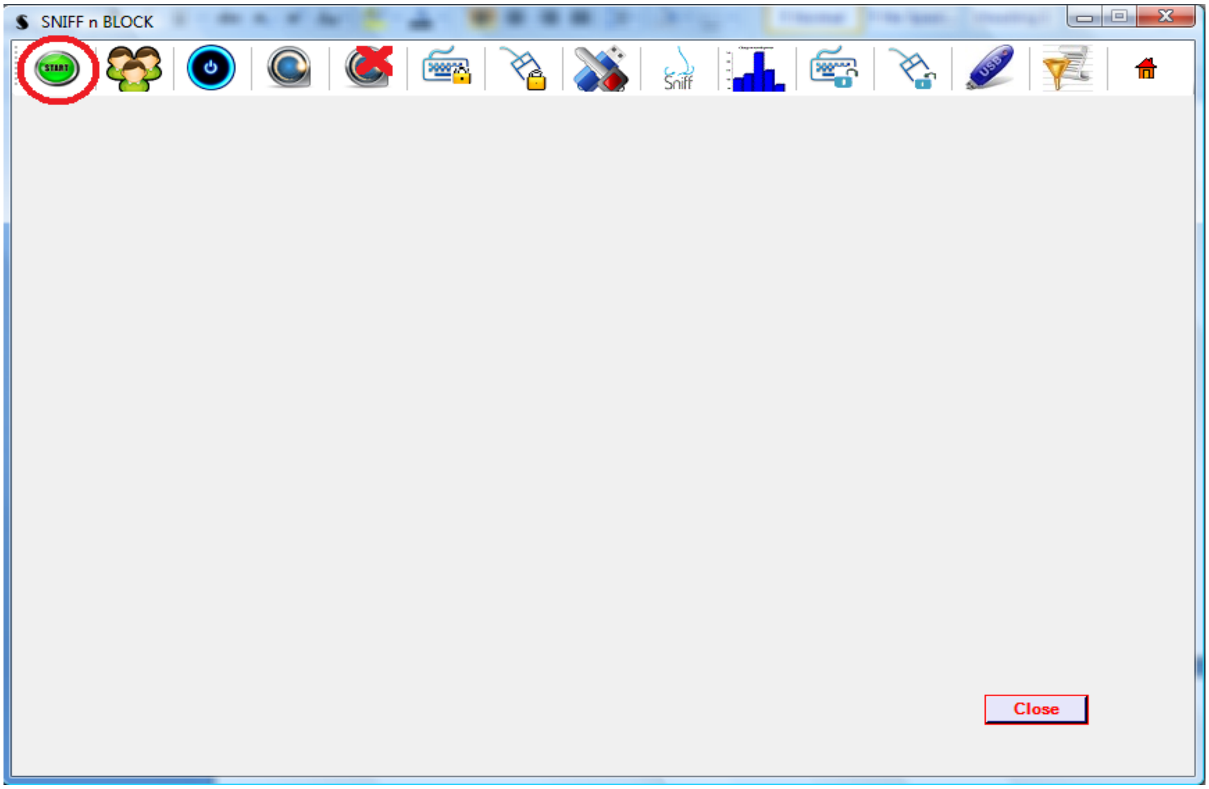
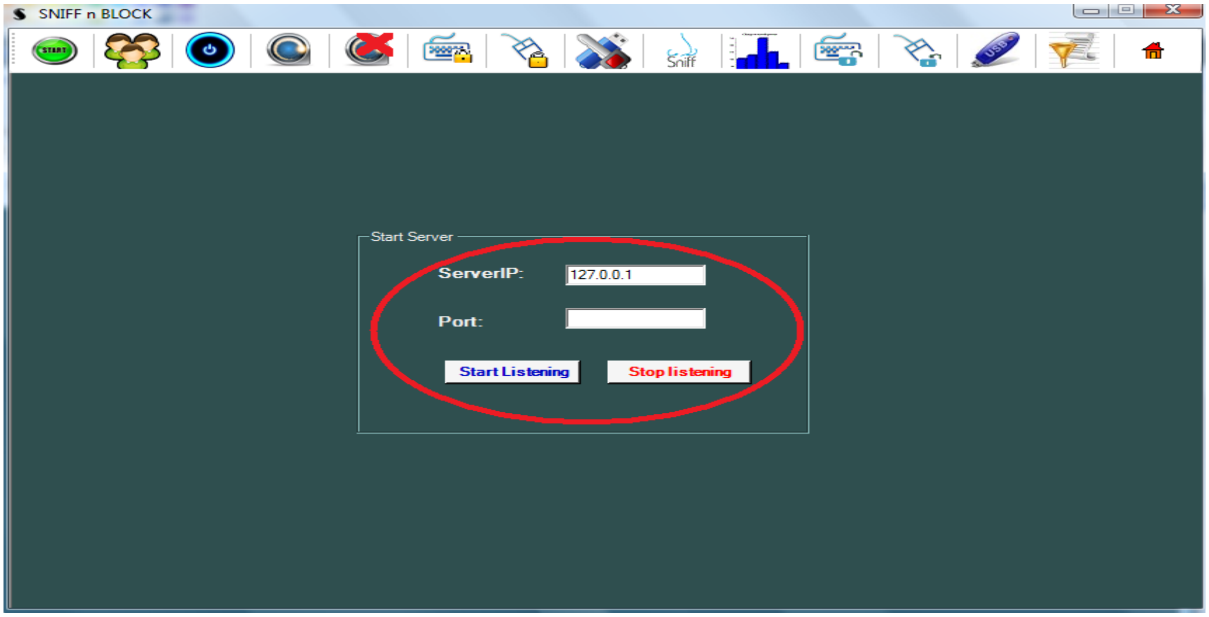


Figure 8.1-2

2. After clicking start the following window appears



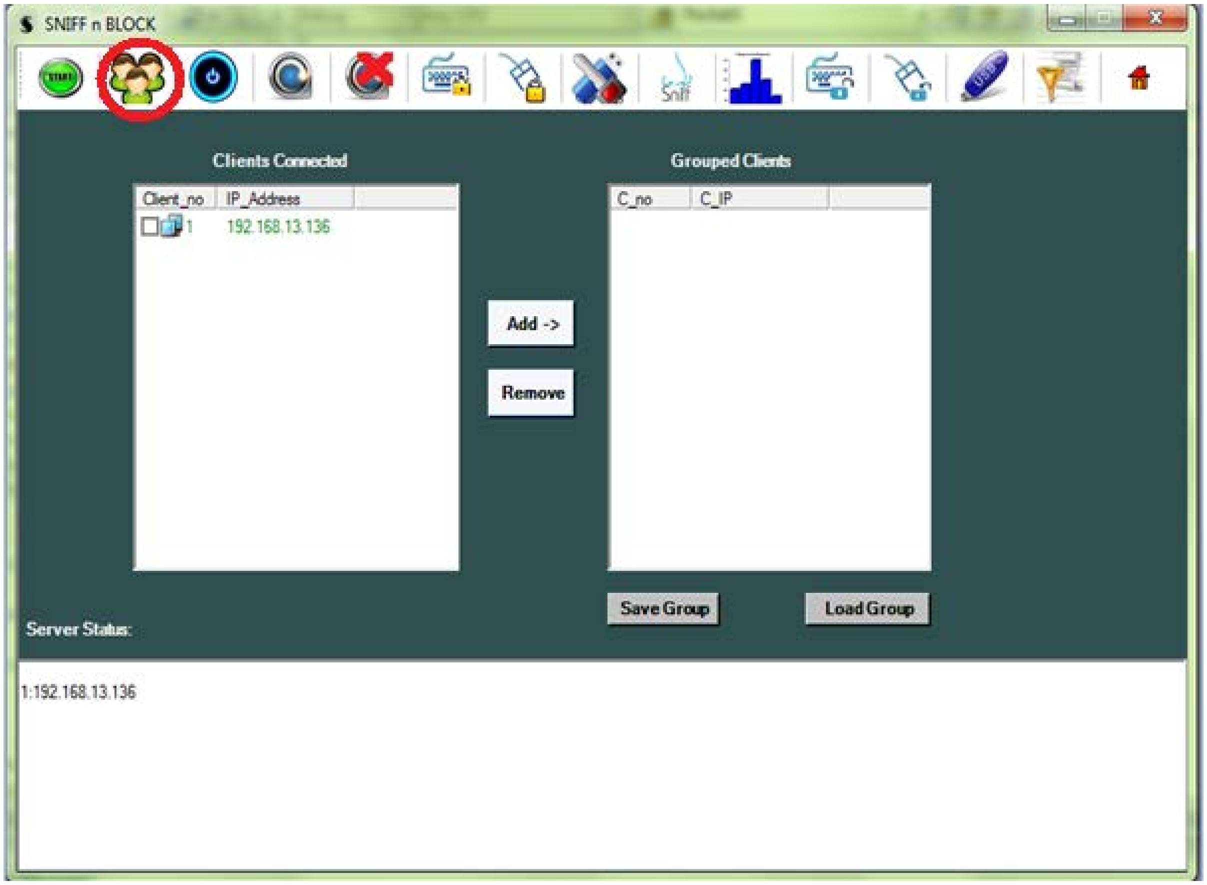
ServerIP: 127.0.0.1

Port:

Figure 8.1-3

The user has to enter the port number to specify the port to listen and press the start listening button to start listening other stop listening can be used when the user does not want to be available to client any more.

3. By clicking the check clients button as indicated the user would get a list of nodes currently connected to the server.



Client_no	IP_Address
□ 1	192.168.13.136

C_no	C_IP

```
1:192.168.13.136
```

Figure 8.1-4

Each client in the view list appears and can be selected with a check box and then after checking each client it can be added to list of grouped clients using the add button. Also there are save button and load button to save and load the saved group respectively. Before selecting any of next features user has to first select clients/groups to perform the function.

Remove button is used to remove the client from grouped clients list.

4. By checking a single client or a group when user clicks ⏻ the selected client/ group is shutdown remotely.

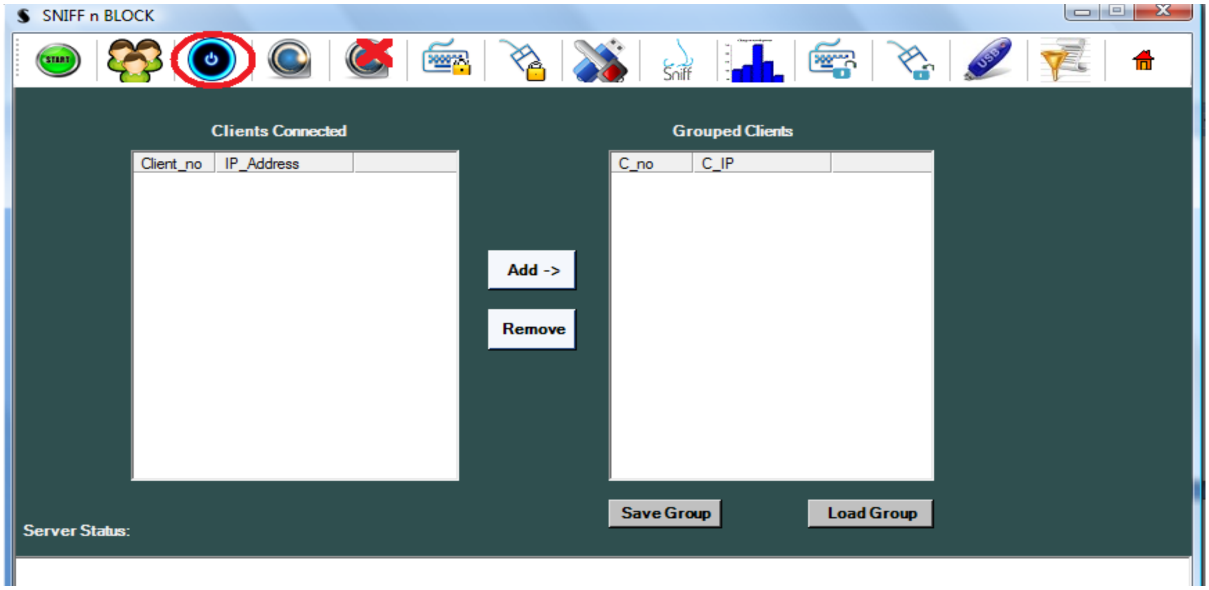
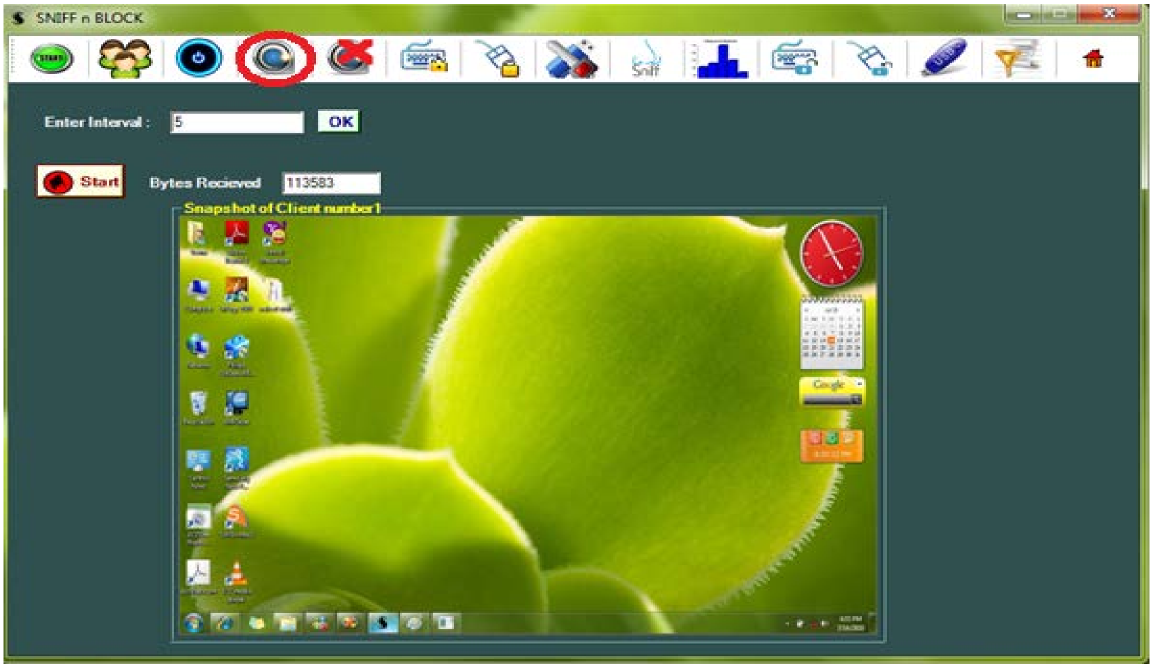


Figure 8.1-5

5. By clicking the ◉ (client screen shot) button user has to specify the intervals after which he wants the screen shots of remote desktop to be generated. Also there is a button to start the capture of selected client/group after specified interval.



Enter Interval : 5

Bytes Recieved 113583

Snapshot of Client number1

Figure 8.1-6

6. Monitoring can be stopped by clicking on .

7. For the selected client/group the keyboard can be blocked by clicking and after the keyboard gets locked a confirmation message box pops up.

8. For the selected client/group the mouse can be blocked by clicking and after the mouse gets locked a confirmation message box pops up.

9. To block the USB ports of the selected nodes the user has to click and after the operation has been performed the the message box pops up.

When the user wants to monitor the clients network card traffic he has to click the Sniff prompted to enter the interval to sniff and also there are buttons to start and show sniffed data.

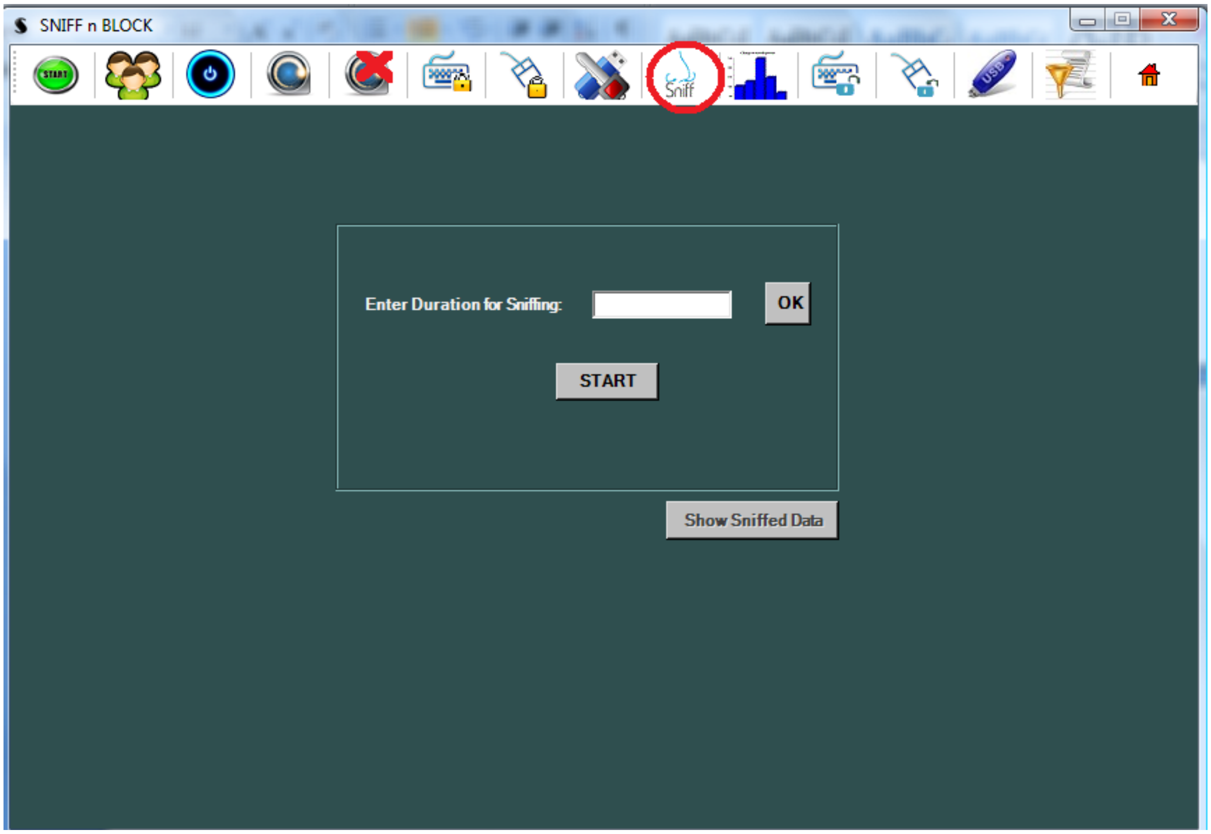


Figure 8.1-7

10. To retrieve the protocols according to frequency in which they were used the user has to click 📊 and gets the resultant graph as shown in figure 8.1-8.

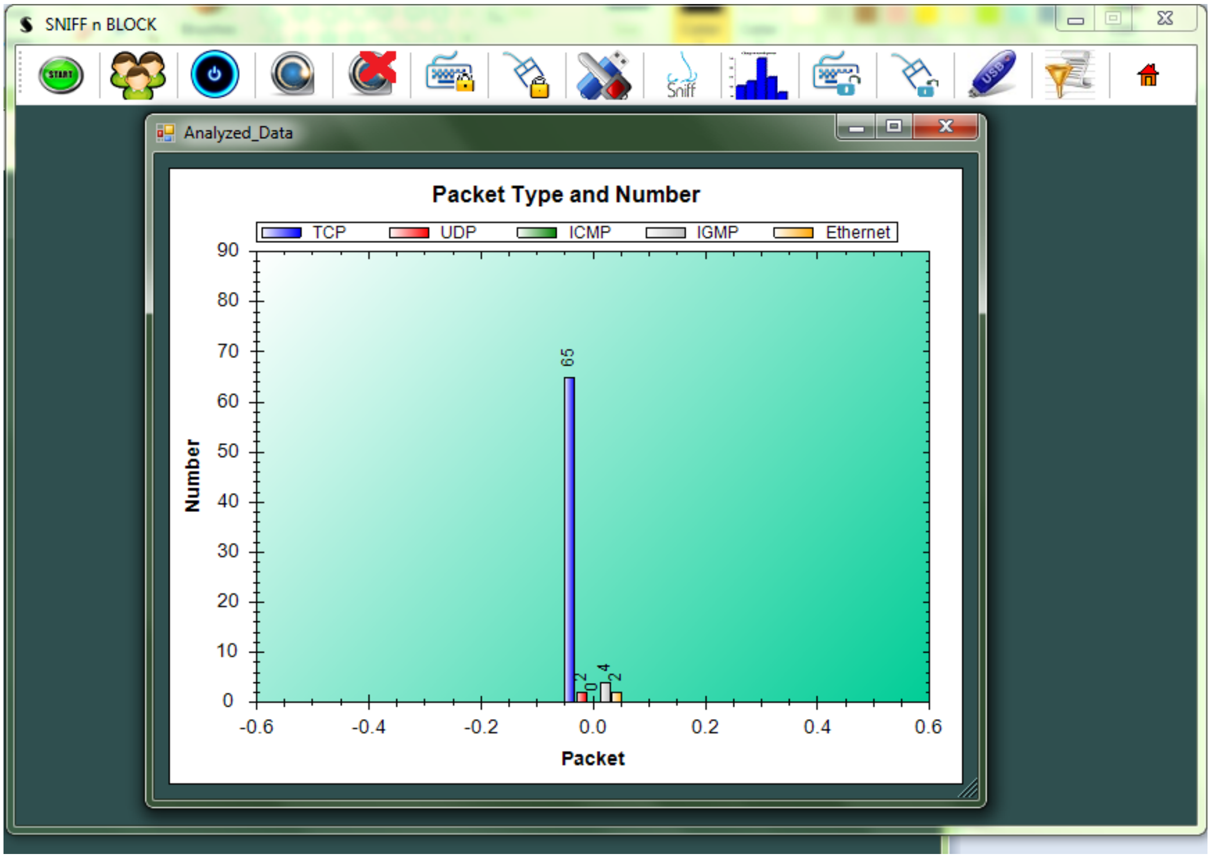
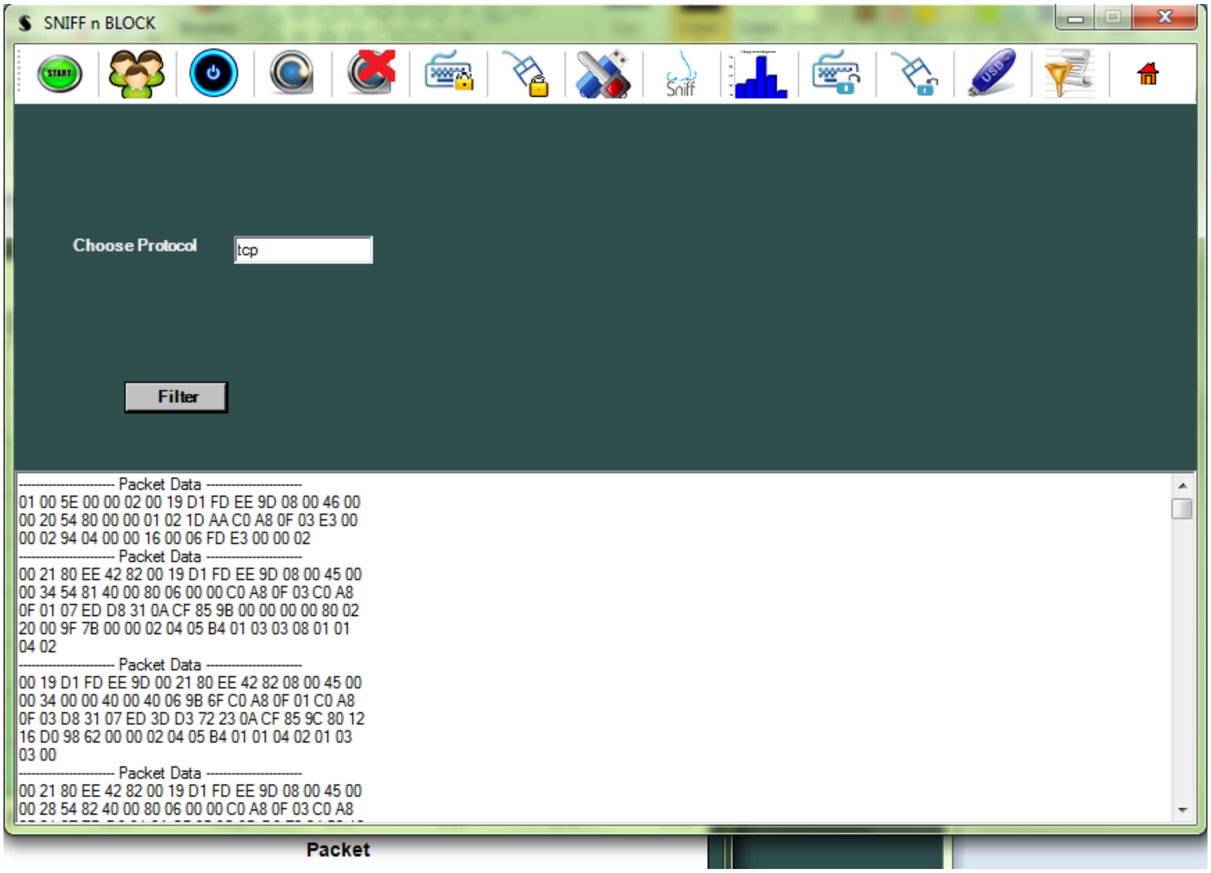


Figure 8.1-8

11. The locked keyboard, mouse and disabled USB ports can be unlocked/ unblocked by clicking respectively available in the top menu. After the action has been performed a confirmatory pop up message box appears.

12. To sniff some selected protocol on selected machine client has to click after which a window appears asking user to enter the protocol. After entering it the client the client gets list of packets with required protocol.



Choose Protocol tcp

Filter

```
--------------------- Packet Data ---------------------
01 00 5E 00 00 02 00 19 D1 FD EE 9D 08 00 46 00
00 20 54 80 00 00 01 02 1D AA C0 A8 0F 03 E3 00
00 02 94 04 00 00 16 00 06 FD E3 00 00 02
--------------------- Packet Data ---------------------
00 21 80 EE 42 82 00 19 D1 FD EE 9D 08 00 45 00
00 34 54 81 40 00 80 06 00 00 C0 A8 0F 03 C0 A8
0F 01 07 ED D8 31 0A CF 85 9B 00 00 00 00 80 02
20 00 9F 7B 00 00 02 04 05 B4 01 03 03 08 01 01
04 02
--------------------- Packet Data ---------------------
00 19 D1 FD EE 9D 00 21 80 EE 42 82 08 00 45 00
00 34 00 00 40 00 40 06 9B 6F C0 A8 0F 01 C0 A8
0F 03 D8 31 07 ED 3D D3 72 23 0A CF 85 9C 80 12
16 D0 98 62 00 00 02 04 05 B4 01 01 04 02 01 03
03 00
--------------------- Packet Data ---------------------
00 21 80 EE 42 82 00 19 D1 FD EE 9D 08 00 45 00
00 28 54 82 40 00 80 06 00 00 C0 A8 0F 03 C0 A8
```

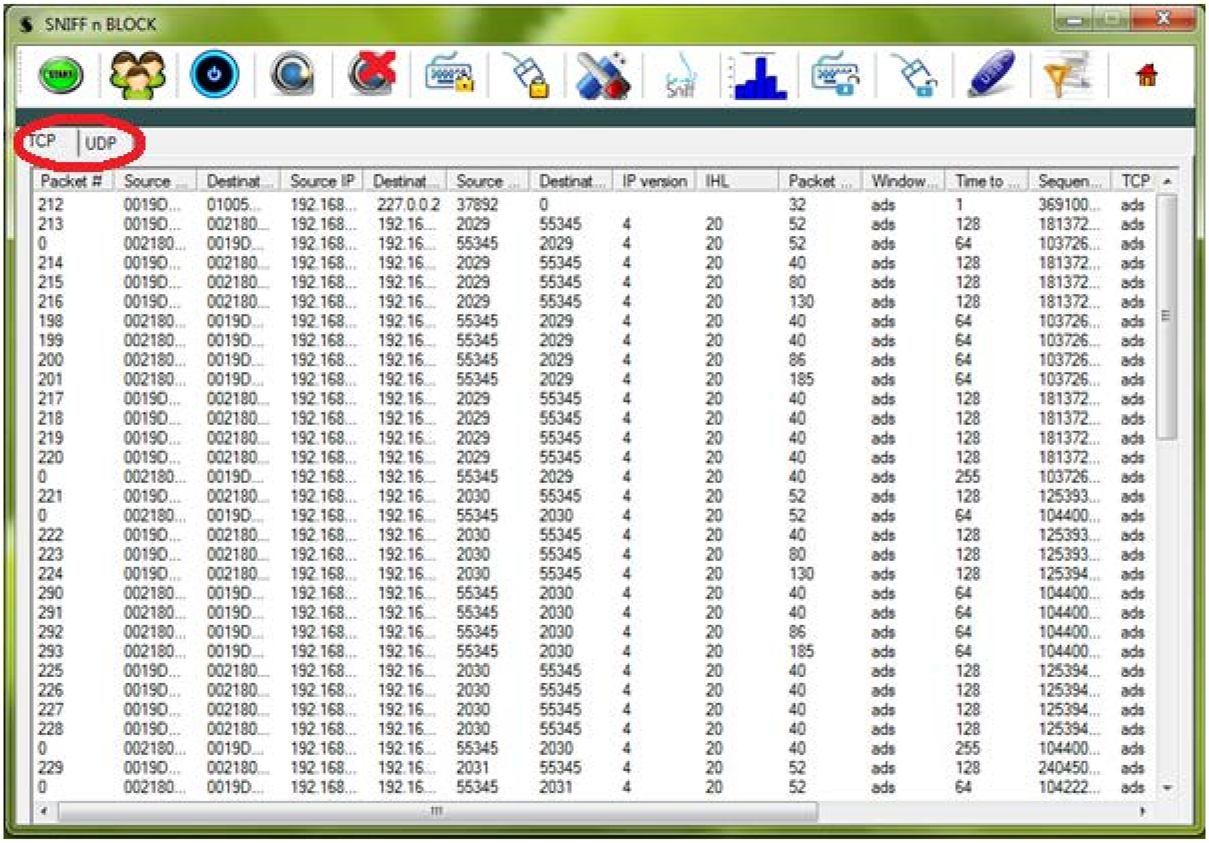
Figure 8.1-9

Figure 8.1-9 shows the result.

13. To go back to home user can click 🏠 after which we go to where you started.

Packet sniffer results window:

When user clicks the show sniffed data button the window in figure u.10 appears and to check the TCP and UDP data captured user gets the required data after selecting the tabs available.



Packet #	Source ...	Destinat...	Source IP	Destinat...	Source ...	Destinat...	IP version	IHL	Packet ...	Window...	Time to ...	Sequen...	TCP
212	0019D...	01005...	192.168...	227.0.0.2	37892	0			32	ads	1	369100...	ads
213	0019D...	002180...	192.168...	192.16...	2029	55345	4	20	52	ads	128	181372...	ads
0	002180...	0019D...	192.168...	192.16...	55345	2029	4	20	52	ads	64	103726...	ads
214	0019D...	002180...	192.168...	192.16...	2029	55345	4	20	40	ads	128	181372...	ads
215	0019D...	002180...	192.168...	192.16...	2029	55345	4	20	80	ads	128	181372...	ads
216	0019D...	002180...	192.168...	192.16...	2029	55345	4	20	130	ads	128	181372...	ads
198	002180...	0019D...	192.168...	192.16...	55345	2029	4	20	40	ads	64	103726...	ads
199	002180...	0019D...	192.168...	192.16...	55345	2029	4	20	40	ads	64	103726...	ads
200	002180...	0019D...	192.168...	192.16...	55345	2029	4	20	86	ads	64	103726...	ads
201	002180...	0019D...	192.168...	192.16...	55345	2029	4	20	185	ads	64	103726...	ads
217	0019D...	002180...	192.168...	192.16...	2029	55345	4	20	40	ads	128	181372...	ads
218	0019D...	002180...	192.168...	192.16...	2029	55345	4	20	40	ads	128	181372...	ads
219	0019D...	002180...	192.168...	192.16...	2029	55345	4	20	40	ads	128	181372...	ads
220	0019D...	002180...	192.168...	192.16...	2029	55345	4	20	40	ads	128	181372...	ads
0	002180...	0019D...	192.168...	192.16...	55345	2029	4	20	40	ads	255	103726...	ads
221	0019D...	002180...	192.168...	192.16...	2030	55345	4	20	52	ads	128	125393...	ads
0	002180...	0019D...	192.168...	192.16...	55345	2030	4	20	52	ads	64	104400...	ads
222	0019D...	002180...	192.168...	192.16...	2030	55345	4	20	40	ads	128	125393...	ads
223	0019D...	002180...	192.168...	192.16...	2030	55345	4	20	80	ads	128	125393...	ads
224	0019D...	002180...	192.168...	192.16...	2030	55345	4	20	130	ads	128	125394...	ads
290	002180...	0019D...	192.168...	192.16...	55345	2030	4	20	40	ads	64	104400...	ads
291	002180...	0019D...	192.168...	192.16...	55345	2030	4	20	40	ads	64	104400...	ads
292	002180...	0019D...	192.168...	192.16...	55345	2030	4	20	86	ads	64	104400...	ads
293	002180...	0019D...	192.168...	192.16...	55345	2030	4	20	185	ads	64	104400...	ads
225	0019D...	002180...	192.168...	192.16...	2030	55345	4	20	40	ads	128	125394...	ads
226	0019D...	002180...	192.168...	192.16...	2030	55345	4	20	40	ads	128	125394...	ads
227	0019D...	002180...	192.168...	192.16...	2030	55345	4	20	40	ads	128	125394...	ads
228	0019D...	002180...	192.168...	192.16...	2030	55345	4	20	40	ads	128	125394...	ads
0	002180...	0019D...	192.168...	192.16...	55345	2030	4	20	40	ads	255	104400...	ads
229	0019D...	002180...	192.168...	192.16...	2031	55345	4	20	52	ads	128	240450...	ads
0	002180...	0019D...	192.168...	192.16...	55345	2031	4	20	52	ads	64	104222...	ads

Figure 8.1-10

CHAPTER 9

Conclusion and Future Work

9.1 Conclusion:

In this global environment where usage of internet and information sharing is increasing day by day, it is imperative for organizations, having a local network, to monitor their users. So sniff n' block application has been designed to solve this problem by providing the controlling and monitoring features to better control and monitor the nodes connected in any environment. It flexibility allows it to be used in any workplace or institutional environment that has its local network and requires control and monitoring.

9.2 Future Work:

The main aim of the project was to develop a control and monitor mechanism for local networks combining hardware control with desktop and network activity monitoring of client(s). The main goal has been achieved. Following are the improvements which can be made to our application in future:

1. Further work can be done to make this application cater to much bigger networks so as to cater to a larger number of clients.
2. The application could be changed to support as many network protocols as possible.
3. The payload packet information could be converted to ASCII.
4. Blocked IP and MAC addresses could be automatically scanned through passive scanning.
5. Client application can be modified into a fire wall type activity.
6. On the server side, information from client side can be used to construct a network map.
7. Also we can enhance the application with database connectivity to automatic unblocking/ stop monitoring/stop analyzing when a user logs off.

BIBLIOGRAPHY

1. “Wireshark User guide” [Online]. Available: http://www.wireshark.org/docs/wsug_html/#Preface [Accessed: 27th Dec, 2009]
2. “Wireshark Forensics “[Online]. Available: http://www.forensicswiki.org/wiki/Wireshark [Accessed: 27th Dec, 2009]
3. “Wireshark Tutorial—Introduction” [Online]. Available http://openmaniak.com/wireshark.php [Accessed: 27th Dec, 2009]
4. “Profile of Wireshark Network Protocol Analyzer”[Online]. Available: http://netsecurity.about.com/od/securitytoolprofiles/p/wireshark.htm [Accessed: 27th Dec, 2009]
5. “Wireshark :How to sniff network traffic” [Online]. Available: http://searchsecurity.techtarget.com/tip/0,289483,sid14_gci1334483,00.html [Accessed: 27th Dec, 2009]
6. “N-Tier Client Server Architecture” [Online]. Available: http://www.exforsys.com/tutorials/client-server/n-tier-client-server-architecture.html [Accessed: 15th Oct, 2009]
7. “Client Server Architecture: The Importance of Flexibility in a Changing World” [Online]. Available: http://www.exforsys.com/tutorials/client-server/client-server-architecture.html [Accessed: 16th Oct, 2009]
8. “The Evolution of Client/Server Computing “[Online]. Available: http://cis.cuyamaca.net/draney/214/web_server/client.htm [Accessed: 16th Oct, 2009]
9. “Two (2) tier Client/Server “[Online]. Available: http://www.zhtech.com/clientserver.htm [Accessed: 18th Oct, 2009]
10. “Trojan Horse” [Online]. Available: http://www.macworld.com/article/60823/2007/10/trojanhorse.html [Accessed: 18th Oct, 2009]
11. [Online]. Available: http://www.infoniac.com/hi-tech/the-history-and-description-of-trojan-horse-virus.html [Accessed: 19th Oct, 2009]
12. [Online]. Available: http://www.webtechgeek.com/computer-trojan-horse-virus-p1.htm [Accessed: 19th Oct, 2009]

13. [Online]. Available: http://www.topbits.com/trojan-horse-virus.html [Accessed: 19th Oct, 2009]
14. “Use Cases” [Online]. Available: http://en.wikipedia.org/wiki/Use_cases [Accessed: 4th Jan, 2010]
15. [Online]. Available: http://www.soi.wide.ad.jp/class/20010020/slides/05/28.html [Accessed: 4th Jan, 2010]
16. [Online]. Available: http://www.gatherspace.com/static/use_case_example.html#1 [Accessed: 4th Jan, 2010]
17. [Online]. Available: http://www.bredemeyer.com/Workshops/Descriptions/ArchitectureReqtsForBusinessAnalysts.htm [Accessed: 10th Jan, 2010]
18. Craig Larman, Applying UML and Patterns (An Introduction to Object-Oriented Analysis and Design), 3rd ed. Prentice Hall, 2004.
19. Roger S. Pressman, Software Engineering: A practitioner’s approach, 6th ed. McGraw-Hill, 2004.
20. Shari Lawrence P. fleeger, Software Engineering: Theory and Practice, 3rd ed. Prentice Hall, 2005.

APPENDIX A

User Manual