KWAME NKRUMAH UNIVERSITY OF SCIENCE AND TECHNOLOGY,

KUMASI

SCHOOL OF GRADUATE STUDIES

MONITORING NETWORK USERS' USING PACKET ANALYSIS

By

Kwabena Kyeremateng (B.SC. Computer Science)

A Thesis submitted to the department of Computer Science, in partial fulfillment of the requirement for the degree of

MASTER OF SCIENCE

In

INFORMATION TECHNOLOGY

College of Science

June, 2012

DECLARATION

I hereby declare that the work submitted in this research is the results of my own designed work and investigation, except where otherwise stated. It has not already been accepted for any degree, and is not being currently submitted for any degree. However, all aspects of this study have been discussed with and approved by my supervisor.

Kwabena Kyerematenng Signature:…… Date:.....

(PG5093410/20137943)

Certified by:

Dr Michael Asante Signature:..... Date: .....

Dr Benjamin Hayfron Acquah Signature:…… Date:.....

Certify by:

Dr Benjamin Hayfron Acquah Signature:…… Date:……

Head of Department

DEDICATION

To God be the glory, great things he has done. To my Mother Mary Kwarteng, my cousin Mr Patrick Opong and my Wife Marian Nkum, I say a big thank you for their continual support and encouragement. And to the Lecturers at the department I say thank you for your support.

May the almighty God richly bless you all.

ACKNOWLEGEMENT

I wish to thank the Almighty God for whose grace I have been able to complete this work.

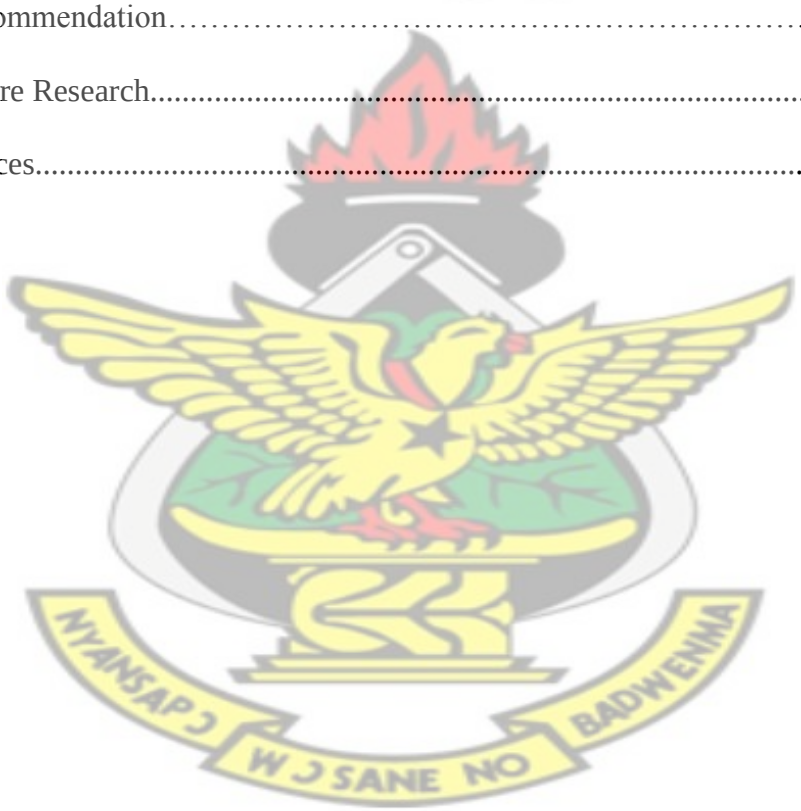
This work will not have being possible without the help, patience and guidance of my supervisor Dr. Michael Asante.

Many million thanks to the Lecturers at the department for their constant reminder and support.

Finally, I am indebted to all those who in various ways helped to make this research work a success.

TABLE OF CONTENTS

Title **Page**



ABSTRACT

This study was conducted to come out with a tool that aid in monitoring network users that traverses outside their domains during peak working hours in a network environment using packet analysis. The main aim of this study was to design non-router based application to monitor users' in a LAN or WAN.

As an enhancement to the previous ones, the tool combined passive and active monitoring and analyzes the packets communication based on source and destination IP address. Find out the effects of this traversal and what these network traversal brings characterized by time, packet size, delay, hop count and bandwidth consumption. In this work, communication of analyzed results to network users and their support was made possible as an internal helpdesk was included in the design as well as network interface card verifier was also included as an enhancement. The captured packet from live network under surveillance obtained using the applications were presented in a tabular and graphical form.

The results of the table and graph were then used as the basis for the analysis. The results showed that network users can be visually monitored using IP address and they can be made to give account of network usage outside their domain; users' activities outside their normal work schedule on the internet leads to additional traffic. This was seen in the graph as packet size, hop count, TCP window size and transmission delay affected network performance.

LIST OF FIGURES

FIGURE **PAGE**

LIST OF TABLES

TABLE **PAGE**

List of Acronyms

LAN	Local Area Network
WAN	Wide Area Network
JDK	Java Development Kit
WINCAP	Windows Capture
JPCAP	Java Capture
IP	Internet Protocol
TCP	Transmission Control Protocol
UDP	User Data Protocol
OSI	Open System Interconnect
MTU	Maximum Transmission Unit
URL	Universal Resource Locator
ICMP	Internet Control Message Protocol
IANA	Internet Assigned Number Authority
NIC	Network Interface Card
NDIS	Network Driver Interface Specification
IEE	Institute of Electrical and Electronic Engineering
API	Application Programmable Interface
JNI	Java Native Interface
SIGAR	System Information Gatherer and Reporter
JRE	Java Run Environment
UML	Unified Modeling Language
ARP	Address Resolution Protocol
MSS	Maximum Segment Size
MAC	Media Access Control

CHAPTER ONE

INTRODUCTION

1.1. **Background**

The increasing importance of networks has given rise to a high demand for monitoring the activities of network users. In this modern day, computers are no longer treated as stand-alone machines. Instead, they communicate to share resources and data through computer networks. However, in the process of communication, they carry packets from source to destination. Network packets are units of data moving in these computer networks, and they carry all the important information from its source to its final destination (Papadopouli, 2009). Alongside the packet payload (the data) which contains lots of valuable information, the packet headers themselves also contain important information about the network infrastructure, network topologies, and may also indicate some kind of general behaviour of the network traffic. For example, the header information can be used to discover the congestion sources in the network traffic. According to Yang (2010), these pieces of information are thus important to the network monitoring tool or system and network administrators from both the analytical point of view and the network problem-solving point of view.

In addition, they are most useful starting points in investigating possible intrusions by attackers who wants to compromise the computer systems and those who want to track both local and remote users who misuse and waste organizational resource in a fully functional computer networks (Ding, 2010).

The need for computer networks for business operations in the area of insurance, banking, airline, educational institutions etc. continue to become bigger and more heterogeneous.

To limit usage or operating cost as well as loss of revenue due to unaccounted users' activities causing much delay and congestion, research on monitoring user's activities is required (Asante et al, 2010).

More importantly, as business needs continue to grow it is important that administrators are aware of and have control of packets that traverses their networks. However, monitoring user's activities in a network using packets analysis is essential, in the sense that, in order to effectively troubleshoot and resolve issues that occur, sources of unwanted traffic must be identified, analyzed and reported.

Furthermore, network administrators are constantly striving to maintain smooth operation of their networks. If a network was to be down even for a small period of time, productivity within a company would decline, and in the case of public service departments, the ability to provide essential services would be compromised. In order to be proactive, administrators need to monitor packets to determine possible source of problem like initiators of congestion and delay (Akindeinde, 2009).

A network poor performance, such as bad link or node may occur as results of several reasons causing service performance bottleneck and delay of varying range from microseconds to weeks. Most causes of the poor performance are cable cuts, poor termination, hardware malfunctions, software errors, unnecessary downloads, and interference both from local and remote links, nodes and human errors such as configuration mismatch etc . However, most of these causes of poor performance indicators are outside the control of network or Internet providers(Asante et al, 2010) and therefore the need for non-router (passive and active) based monitoring technique to monitor the effects of transmission delay, bandwidth consumption and congestion emanating from users activities from both local and remote source. In the analysis, delay, bandwidth consumption, congestion are applied on the following packet

parameters; source and destination IP address, packet length in byte, hop count, window size are explained using graphs. Thus, these performance metrics depends on source of traffic within the network from both local and remote (Lucas, 2010).

Packet in transit is normally affected by sum of delays, which are

total propagation,total transmission ,end-to-end and total processing, but the study looks at the transmission delay for the analysis. In this analysis, much emphasis is on the effect of number of packet captured, hop count, window size, source and destination IP address, packet length and packet type on bandwidth consumption, and delay,.

Monitoring packet movement in LANs and WANs is the key element to maintaining a productive environment in the networking world. As more services like email, messaging on Internet become available to more users, the performances of networks suffer. Network administrators through constant monitoring must recognize and be able to rectify problems before they become noticeable to end users.

Various tools are available to monitor the network on a local and remote basis. A comprehensive understanding of these tools is critical to effective network management. So, the first step to solve this problem is by monitoring the traffic with suitable and cost effective system. Most of the works use router based monitoring or non-router based monitoring (passive and active) technique. Even though router based (agent) is good, yet very expensive. There have been extensive works on non-router based monitoring with concentration on packet capture, still very little has been done in Bandwidth, delay, helpdesk and congestion analysis components.

The proposed work used non-router based monitoring technique and internal help desk for communication. This work designs and develops an application, used to monitor users' activities in a network, and this is based on captured packets and

analysis. Also, the application so developed captures packets transmitted between hosts on LAN/WAN/Internet, monitor network interfaces both hardware and software status as whether they are up and running, analyze packets based on source and destination IP address, communicate observed results of analysis back to users through live communicator and help desk.

1.2. Problem statement

One of the major problems in network research in terms of monitoring is the difficulty of using non-router base approach in monitoring user's activities on the internet using network packets analysis and reporting individual users' technical problem.

You can only locate hosts content by accessing the node that manages that content. It is done by specifying the desired content attribute or names. For any system, the monitoring techniques depend on the needs of the application in question.

At the moment, users in LAN environment where packet monitoring are being used depends on either person to person for solutions to LAN problems or phone calls to administrators for solution on IT related problems. Most of the time administrators might not know the problems whenever they are called for technical assistance. The phone calls and the search for experts waste business precious time. In addition, enterprise monitoring applications are normally deployed on strategic points across the networks with traffic only in mind, but none is deployed at the end node, hosts and visually monitor users' movement. This leaves some hosts invisible to the monitoring applications. Thus, they might bring additional packet which is not needed at the time of use by their organisation leading to the introduction of additional traffic to the network that is not given much attention by the monitoring element.

In networks, traffic is the common problem that always occurs to the end users. As the network expands, traffic becomes heavy, thus possibilities for the networks to congest are very high. The rapidly growing number of users and application and especially the growth in recreational traffic, has led to application slowdowns and user complaints, and more importantly activities of both LAN/WAN users in organizations. The problem is **who is responsible** for causing this additional traffic during peak hours of work in organization and how can packet analysis in passive monitoring be used to visually monitor users movement, delay, bandwidth consumption and congestion resulting from users' activities. We are looking at an application where clients could also report on their own problems by just a click of a mouse in a LAN and its activities be monitored. At the moment, network monitoring problems in LAN can only be detected by professionals who sometimes cost organizations in terms of time to attend to LAN monitoring problems which are often trivial. This helps to reduce the time between failures and the time of restoration.

1.3. Research objective

Thus, the research seeks to address the following specific objectives:

1. To design a tool which is able to visually monitor packet between hosts' or users' movement on the network/Internet for management decision?
2. Verify network interface card and driver status without introducing additional traffic
3. To communicate monitored results to clients and resolve their problems through internal help desk.

4. Analyze the effects of users' activities characterized by bandwidth consumption, delay, and congestion on the packets parameters within the network/internet.

1.4. Research Question

In respect of the objectives outlined, the study seeks to answer the following research questions.

1. How can packet parameters be used to visually monitor network hosts' or users' movement in the network and on the Internet based for management decision?
2. How can network driver status and card status be verified without introducing more traffic?
3. How can the monitored results be communicated to clients and their network problems communicated back to administrators?
4. What are the effects of users' activities on bandwidth consumption, delay, and congestion on packet parameters?

1.5. Research Methodology

In order to accomplish the objectives of this research, a number of methods were used.

Standard waterfall model are used for the design of the application. In addition, live packets were captured from the WAN. However, some tools were needed to make the design a success, which are: Netbeans, Mysql, Java, Wincap, Primefaces, Jpcap.

1.7. Justification of the research

The contribution is using IP address numbers to monitor the activities of users based on packet movement.

Network computing are growing with the rate at which business grows in the world, computer network engineers are responsible for the design and installation. For security and resource monitoring reasons, network needs to be monitored against source of threats and traffic that are not needed.

The work makes it possible to detect host communicating with each other, internet and in that way administrators can easily track user's packets movement if they are not in line with the organizational objectives. In addition, the effects of delay and congestion are analyzed using graph.

1.8. Scope of the study

The study looked at the following: application, which consist of the design of the various component involved. In addition, the application is expected to capture data for analysis.

The main Stakeholders involved in the study are; IT administrators, IT technicians, Local Area Network and internet users as well as network engineers.

1.9. Organization of the Study

The study is organized into 5 chapters. The contents are arranged such that each previous chapter provides a brief idea about the next chapter. The First, chapter introduces the background principles of network monitoring using packet analysis and our research objectives as well as contributions.

In Chapter 2, the study review literature and fundamental concepts related to the work and issues surrounding it. The reasons why we choose the methodologies for the work are discussed, as well as related work. Chapter 3 covers the methodology of how the proposed work was designed from the requirements analysis. In addition, in-depth logical including UML modelling was made. Chapter 4 deals with packet analysis

which comes from the data obtained from the captured packets through developed application.

Finally, chapter 4 is the primary content of chapter 5. This chapter serves as the conclusion of the study and recommendation. Conclusions were made based on the findings in the analysis and designed application.

Chapter 5 serves as the conclusion of the study and recommendation. The study revisits the research contributions and objectives with regard to methods in Chapter 3 and its results in Chapter 4. Finally, a discussion and suggestion for future work with regards to the work is presented.

CHAPTER TWO

LITERATURE REVIEW

2.1. Overview

This chapter is the beginning of the framework that is being used as reference or guideline to the research. In this chapter, review of all information related to the study to get better understanding of the work is done. Besides, it reviews the concept of TCP/IP, UDP, packet analysis, and network monitoring as well as monitoring techniques.

2.2. Approaches to Solve Similar problems

a. Passive Distributed Network Analysis Using Remote Packet Capture in Java. The objective was to avoid the use of proxy server to monitor conversation that users of LAN are having with outside world using passive monitoring (Judge, 2005). The difference with this work is that, this work includes active monitoring utilities but this does not.

b. **Network Traffic Monitoring Analysis on Quality of Service**

The background and objective was to make analysis from monitoring network traffic by application or protocol: HTTP, FTP, Telnet, and SMTP, TCP. The findings in this work were the best way to improve the network performance is by implementing the Quality of Service Solution. This research work share some of the protocols common but the emphasis on this work is performance and accountability for network use (Mashita, 2003)

c. **Developing a web-based Packet Monitoring Tool**

The background and objective was to develop a web-based program to monitor network traffic over a local host and a local network based on the flow of the User Datagram Protocol. (Mohammed, 2006). The difference with this work is that, this

work is able to combine both passive and active. In addition, this work is able to monitor host in Wide Area Network.

d. **Developing TCP/IP and UDP Traffic Monitoring Tool.**

The main objective was to develop a tool to monitor the TCP/IP and UDP traffic.

The implementation was done by using basic language (Visual Basic 6.0) program.

Methodology used was an analysis, design, implementation and testing. The differences with this work are the development of this work is with Java program, adaptor capture and the main objective, which is users' activity with respect to delay, bandwidth and congestion (Rafiq, 2005)

e. **Developing a Packet Capturing Program**

This work was to develop a program to capture packet from the network interface card, and analyze some protocol such as ICMP, TCP and UDP. The differences with our work are that her packet-capturing program written in C language based on Linux Operating System platform and the fact that she does not consider graph. (Hamzah, 2001).

All the above approaches did not include active monitoring, this work combine both passive and active monitoring, graphs and bandwidth analysis.

2.3. TCP/IP Protocols

The Internet protocols consist of a suite of communication protocols, of which the two best known are the Transmission Control Protocol (TCP) and the Internet Protocol (IP). The Internet protocol suite not only includes lower-layer protocols (such as TCP and IP), but it also specifies common applications such as electronic mail, terminal emulation, and file transfer (Miller, 2010). They can be used to communicate across any set of interconnected networks and are equally well suited

for LAN and WAN communications. Thus, the work uses the TCP/IP as the base for the protocol. Figure 2.1. shows the TCP/IP and OSI protocol model, showing both upper and lower layer protocols.

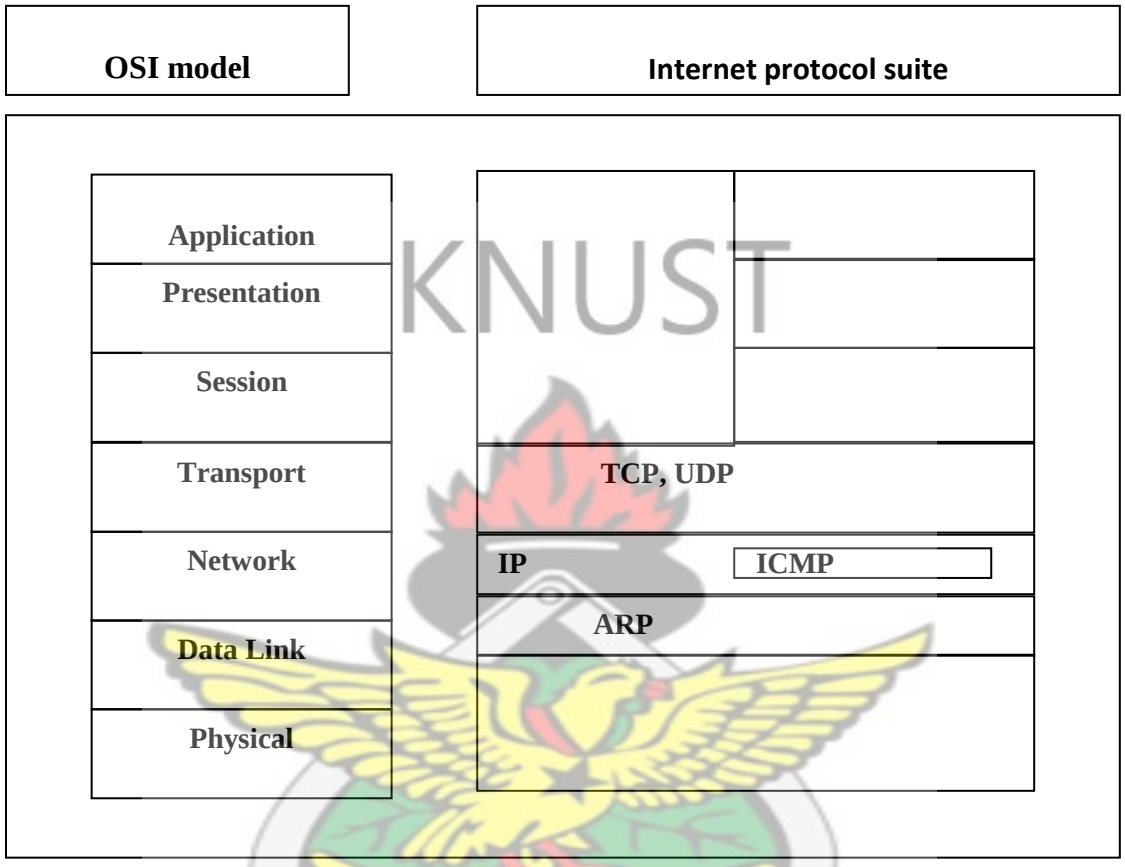


Figure 2.1 Internet protocols and range of OSI model layers.

(Adopted from www.tcpguide.com)

The Internet Protocol (IP) is a network-layer (layer 3) protocol that contains addressing information and some control information that enables packets to be routed. IP is the primary network- layer protocol in the Internet protocol suite. Along with Transmission Control Protocol (TCP), IP represents the heart of the Internet protocol (Fall, 2011).

It provides connectionless and best effort delivery of datagram through an inter-network; and providing fragmentation and reassembly of datagram to support data links with different maximum-transmission unit sizes (Harpence, 2011).

2.3.1. **Packet**

A packet as a unit of data is routed between an origin and a destination on the Internet or any other network. The entire file downloads, Web page retrievals, email, all these Internet communications always occur in the form of packets. When any file is sent from one place to another, the Transmission Control Protocol (TCP) layer of TCP/IP divides the file into "chunks" of an efficient size for routing. Each of these packets is separately numbered and includes the Internet address of the destination (Sanders, 2007). Basically as series of digital numbers, it conveys the following:

- the source IP address and port numbers;
- the destination IP address and port numbers;
- hop count information;

Depending on the protocol(s) they need to support, packets are constructed in a standard **packet format**. It formats include a header, the body containing the message data (*payload*), and a *trailer* (Mitchell, 2012).

The packets carry the data in the protocols that the Internet uses: Transmission Control Protocol/Internet Protocol (TCP/IP). Each packet contains part of the body of your message. A typical packet contains perhaps 1,000 or 1,500 bytes (Lucas, 2010).

Most packets are split into three parts:

a. Header - The header contains instructions about the data carried by the packet.

Some of these instructions may include:

- Protocol defines what type of packet is being transmitted: e-mail, Web page, streaming video

- Destination address (where the packet is going)
- Originating address, where the packet came from (Odom et al, 2007).

Each packet header contains the proper protocols, the originating address, the destination, and the packet number (1, 2, 3 or 4). Routers in the network look at the destination address in the header and compare it to their lookup table to find out where to send the packet (Todd, 2007). For the purposes of this work, the adoption of this nature of protocol was used.

2.3.2. IP Packet Format

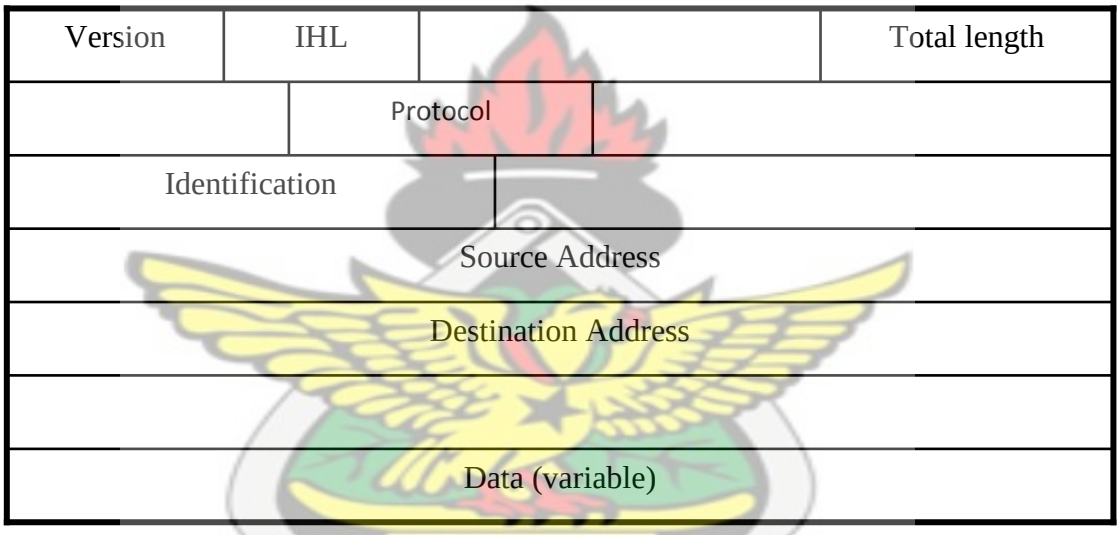


Figure 2.2 Fields of an IP packet (Harpence. 2011).

(Source: adopted from Commer, 2006)

The information below describes the IP packet fields illustrated in Figure.2.2 above (Todd, 2007). IP packet parameters that were used for the work are described as follows.

IP Header Length (**IHL)** — indicates the datagram header length in 32-bit words.

Total Length **—** specifies the length, in bytes, of the entire IP packet, including the data and header.

Identification — contains an integer that identifies the current datagram. This field is used to help put together datagram fragments (Lucas, 2010).

- **S**ource Address — specifies the sending node.
- Destination Address — specifies the receiving node (Odom et al., 2006)

Even though the IP packet format has quite number of parameters, the source and destination address, packet length, identification field, packet type, window size were used in this work.

2.3.3. **IP Addressing**

As with any other network-layer protocol, the IP addressing scheme is integral to the process of routing IP datagram through an inter-network. Each IP address has specific components and follows a basic format. These IP addresses can be subdivided and used to create addresses for sub-networks (Rooney, 2011).

Based on this addressing scheme, users within organizations are monitored by the kind of domain or network they belong to, thereby tracking their source of generating traffic (Buck, 2001).

2.3.4. **Internet Routing**

The movement of packet from source to destination experiences some elements of setbacks called delay (myipaddressinfo.com, 2006). For the purposes of this work, only transmission delay will be looked at: time it takes to transmit a packet. Furthermore, since packets crosses router, the number of router which is called hop count also affects the packet movement as a result of the delay (computer.howstuffworks.com, 2012)

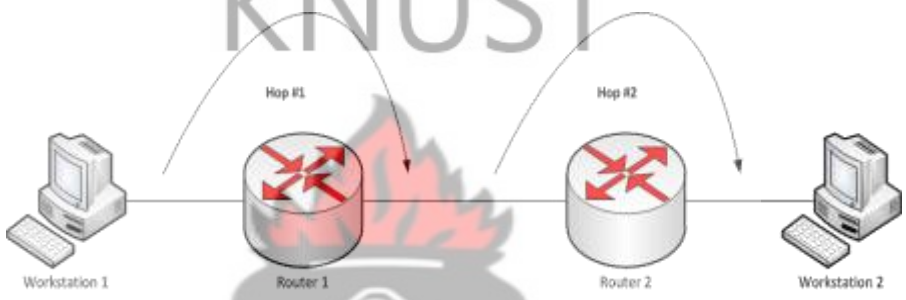


Figure 2.3 Movement of packets between two routers bounded by two workstations (Source: adopted from Schonwalder, 2005).

Figure 2.3 has two work stations, workstation 1, and workstation 2, two routers: roiuter1 and router 2. Packets moves from workstation 1, traversing the two routers to workstation 2.

Port Numbers

Ports are transport layer (TCP and UDP) connection points numbered from 0 to 65,535. According to Internet Assigned Number Authority (IANA) classification, the port space of 0 to 65,535 actually breaks down into three ranges (Caliskan, 2011).

Ports are used by the operating system to make connections to remote systems.

Through ports, the system can identify which service it is requesting from another system. In addition, the work can use the port number system to trace the source of

traffic.

2.4. Network Packet capture

Network packet capturing application, commonly are programs or libraries that obtain data packets flowing through a certain network segment in which the system is connected to by means of a network card (Clos, 2010). These captured packets from a network are processed, for instance decoding headers information and showing it or extracting data from headers for later calculations. In this work, the major task will be obtaining packets from a network card or network interface, and analyze it (Srikanth, 2004).

Network packet capture has the ability to capture packet data from the data link layer of the ISO-OSI Reference model. This includes headers and payload. Thus, Packet capture encompasses every packet that crosses a network segment, regardless of source, protocol (Casad, 2011). The captured packet serves as the base for the monitoring, and therefore this research work adopts this technique.

a. Packet Monitoring in an Ethernet Network

Packet monitoring can be achieved in an Ethernet Network because of its broadcasting nature. That is any packet sent over an Ethernet Network is broadcasted to all machines in the network. The Ethernet card in every machine checks whether the particular packet is destined to it, if so accepts it else rejects it. This basic functionality of an Ethernet Network is exploited to capture all packets that travel through the network. In addition, the Ethernet card can be placed in a number of modes using drivers and can be used to capture packets as desired. (wiki.wireshark.org, 2012).

b. **Network Interface Card(NIC) and Driver**

Drivers are developed by programmers to fulfil the needs of a particular application. A device driver is the glue between the Operating system and its input/output devices. Drivers act as translators, converting the generic requests received from the Operating System into commands that specific peripheral controllers can understand. For network device communication, network interface card is needed (Calsoft, 2012).

NIC drivers interface directly to the hardware (NIC) at its lower edge and at their upper edge these provide an interface that helps the upper level drivers to: Send and receive packets (Calsoftlabs.com, 2012). Thus any defect of the NIC may impair the capturing process, since device selections are based on it.

c. **Modes in an Ethernet Interface Card**

The various modes in which an Ethernet card can be placed is as follows:

i. **Directed**

A frame that is destined to a particular machine has the destination machine's physical address (Ethernet address) specified as its destination address. The machine with that physical address accepts the frame while all the others reject it. The card can also be set to only receive directed frames programmatically (Degioanni, 2000) .

ii. **Promiscuous**

Any card when placed in this mode accepts any packet that comes to it. This mode along with the broadcasting nature of the Ethernet is the key to the packet monitoring application. The PACKET.SYS driver helps to place the network card in all modes. The application with the aid of the PACKET.SYS places the card in the promiscuous mode to capture all the packets that travel in the network (wisegeek.com, 2012).

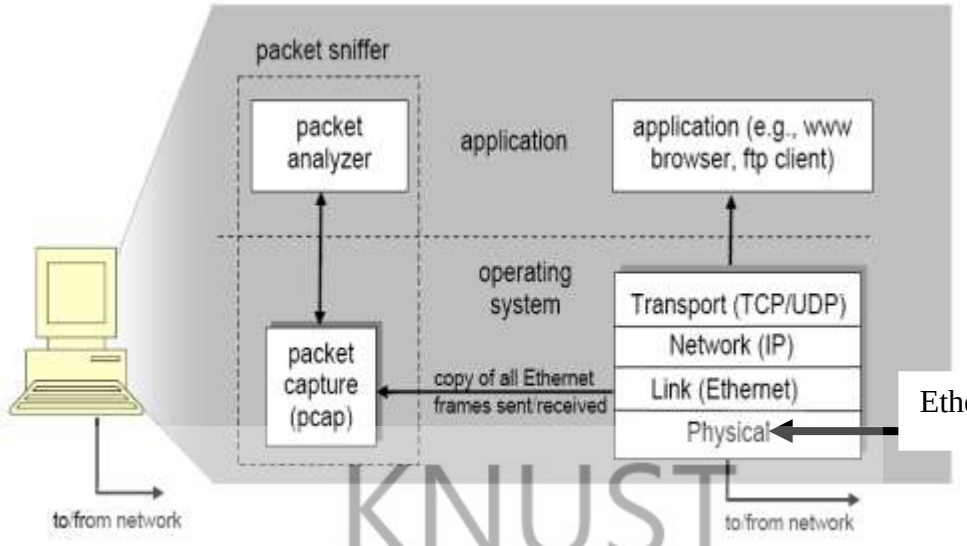


Figure. 2.4: Packet capture structure

(Source: adopted from Network Research Laboratory)

The above diagram in figure 2.4 shows how packet is captured from the network through the physical layer using wincap and forwards it to Jpcap which is Java native interface for further analysis using Java (Antoniou, 2008).

Capturing packets from a network is highly dependent on the type of the network used and of the topology and configuration of the network. LAN (Local Area Network) networks based on the IEEE 802.XX (physical and link layer protocols) protocol family, for instance in the IEEE 802.3 (Hancock, 1999) protocol based networks, also known as *Ethernet networks*, and in the IEEE 802.11(Odom, 2006) based networks, called *Wifi or Wireless* networks demonstrate this.

In IEEE 802.3 LAN network, a star topology is used, so all the nodes in the network are connected through their own cable to switch. Switches send packets to the port where the destination host is connected, by previously identifying all the hosts connected to each port.

IEEE 802.11 based networks share access medium, so it may be easier than IEEE 802.3 switched networks to capture packets, as having a network card being

able to be set to promiscuous mode (actually monitor mode) is all the hardware required. However, some considerations have to be kept in mind. When placing a capturing system in wireless network, some packets or even all the packets sent by a certain host may be lost, due to environment conditions (shadowing) and the physical position of the capturing tool host and the other hosts in the network (Clos, 2010).

2.5. **Network Traffic Monitoring and Analysis Techniques**

Network traffic analysis could be fundamentally summarized with the history of network monitoring on one hand and network intrusion detection on the other. Both of them have been the main areas in which network analysis engineering efforts have been centered in due to their interest and outcome (Schmidt et al, 2005). Here, the study, examines router based monitoring techniques and non-router based monitoring techniques (Cecil, 2006).

Two Monitoring Techniques are discussed in the following sections: Router Based and Non-Router Based (Wei et al, 2011). Monitoring functionalities that are built-into the routers themselves and do not require additional installation of hardware or software are referred to as Router Based techniques. Non-Router based techniques require additional hardware and software to be installed and provide greater flexibility. Both techniques are further discussed in the following paragraphs

a. **Router Based Monitoring Techniques**

Router Based Monitoring Techniques are hard-coded into the routers and therefore offer little flexibility. A brief explanation of the most commonly used monitoring techniques is given below. (Kevin et al).

b. **Non-Router Based Techniques**

Although non-router based techniques still have some limitations in their abilities, they do offer more flexibility than the router based techniques. These techniques are classified as either active or passive (Caliskan, 2011).

i. **Active Monitoring**

Active monitoring transmits probes into the network to collect measurements between at least two endpoints in the network. Active measurement systems deal with metrics such as:

- Routes
- Packet Delay

Some of commonly used tools such as ping, which measures delay and loss of packets, and traceroute which helps determine topology of the network, are examples of basic active measurement tools. They both send Internet Control Message Protocol (ICMP) packets (probes) to a designated host and wait for the host to respond back to the sender. Figure 2.5 is an example of the ping command that uses active measurements by sending an Echo Request from the source host through the network to a specified destination (Domingo-Pascual et al, 2011).

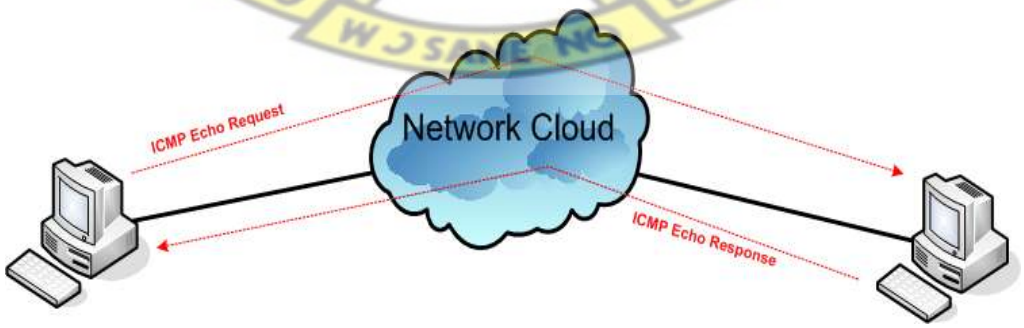


Fig. 2.5 ICMP ping command (Active Measurement)

The problem that exists with active monitoring is that introducing probes into the network can be interference to the normal traffic on the network. Often times the active probes are treated differently than normal traffic as well, which causes the validity of the information provided from these probes to be questioned.

As a result of the information detailed above, active monitoring is very rarely implemented as a stand-alone method of monitoring as a good deal of overhead is introduced. On the other hand, passive monitoring does not introduce much if any overhead into the network.

ii. **Passive Monitoring**

Passive monitoring unlike active monitoring does not inject traffic into the network or modify the traffic that is already on the network. Also, unlike active monitoring, passive monitoring collects information about only one point in the network that is being measured rather than between two endpoints as active monitoring measures (wand.cs.waikato.ac.nz, 2012). Figure 2.6 shows the setup of a passive monitoring system where the monitor is placed on a single link between two endpoints and monitors traffic as it passes along the link.

Figure 2.6 Passive Monitoring

(Source: adopted from Cecil, 2006)

Passive measurements deal with information such as: Traffic and protocol mixes Accurate bit or packet rates Packet timing and inter-arrival timing.

Passive monitoring can be achieved with the assistance of any packet sniffing program.

Although passive monitoring does not have the overhead that active monitoring has, it has its own set of downfalls. With passive monitoring, measurements can only be analyzed **off-line** and not as they are collected. This creates another problem with processing the huge data sets that are collected (inet.tu-berlin.de, 2012).

As one can see passive monitoring might be better than active monitoring in that **overhead data** is not added into the network but post-processing time can take a large amount of time. This is why a combination of the two monitoring methods seems to be the best option. This work combines both the active and passive monitoring techniques to achieve its objectives.

a. **Network traffic analysis theory**.

Network traffic analysis could be defined as: "the inference of information from observation of the network traffic data flow". Analysis in network traffic can be categorized by time criteria and by the purpose of the analysis. In this work, time and purpose is of greater interest, as packet traverses the network with time as well as for what purpose Works (Clos, 20107).

b. **Network traffic data inspection techniques**

Network data inspection techniques obtain information of network data by inspecting network header fields of each packet, compute them and produce outputs or results.

The simplest network data inspection possible is ***packet decoding***, also called packet analysis, in which all header's field are decoded and presented in a human readable way. Network analyzers like tcpdump (Garcia, 2010), or Wireshark (Chappell, 2012), are some examples of packet decoding applications. The work captures the packet

block decode it and extract the parameters, present it graphically and draw statistical information and pattern extraction.

Graphical representation of packet data is of interest in this work, principally in network metrics on user movement.

2.6. **Libraries**

To grab a copy of packets off the wire before they are processed by the operating system, the work needed packet capture library and others. Packets arrive at the network card; it verifies the checksum, extracts link layer data and triggers an interrupt (Dainotti et al, 2004). The interrupt then calls the corresponding kernel driver for packet capturing. Figure 2.7 below shows the flow diagram.

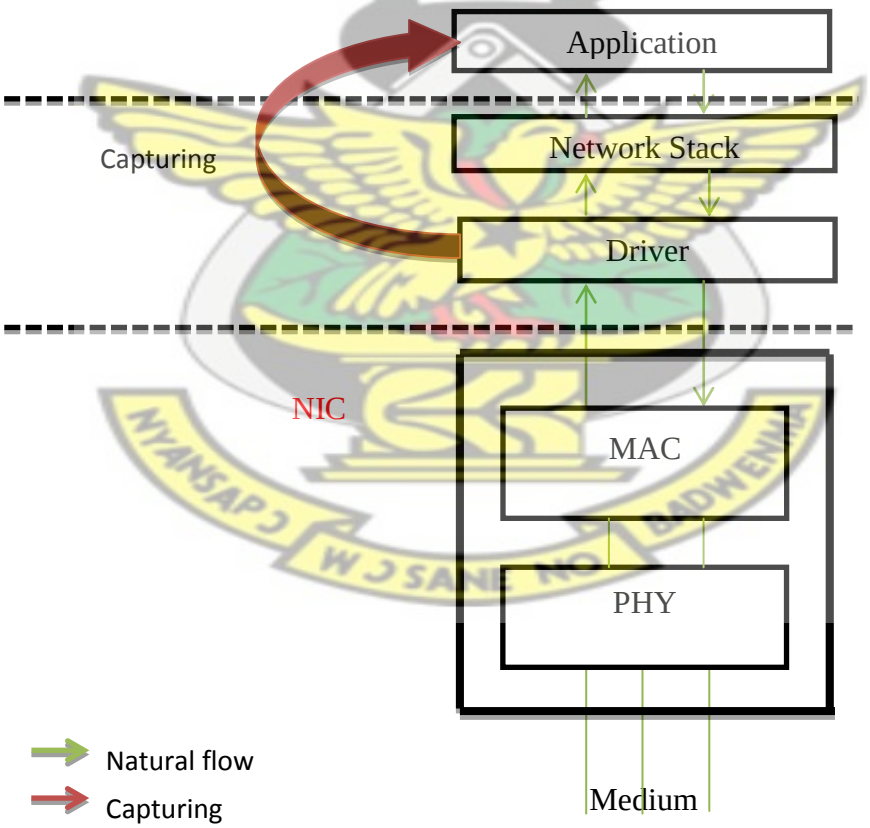


Figure 2.7 Application and driver communication (exa.unicen.edu.ar,2012)

The figure 2.7 shows how the packet capturing by pass the protocol stack.

A major function of many packet analyzers is to capture network traffic and extract information from the packets flowing across the network. What makes this possible is the packet capture library.

This section provides a quick and practical introduction to packet capture, wincap and jpcap library on both wired and wireless networks. It is intended to accelerate and simplify the process of creating a packet-capturing application.

To implement a software component for each protocol participant that is capable of processing all IEEE 802.11 and Ethernet incoming packets and of sending new packets, the packets need to be captured at the lower level and forwarded to the user or application layer.

Usually the applications that are running in the user level are able to operate only on network packets that were passed through the protocol stack to the application layer. With regard to the requirements of this work for the implementation, the application need to be able to capture the network traffic on a lower level in order to get the original packets that are being processed by the kernel stack (tcpdump.org, 2007) .

a.**Wincap:** JPcap supports using the windows port of winpcap, which means that the system can still be used on most unicies and windows.

The WinPcap for Microsoft Windows environments allows Windows applications to capture and transmit packets on low-level network layers (winpcap.org, 2012)

b. **JCap: This** library offers the work an API to create user-level network capturing programs. It offers the following capabilities: packet capturing from a network card

However, Jpcap is a *Java Native Interface (JNI)* implementation of the WinPcap libraries described above. Hence, Jpcap supports any operating system that is compatible with WinPcap (CACE Technologies, 2008).

Furthermore, to capture network packets in the Java program, This API was the better option because no parts of the core Java API give access to low-level network data. In addition, Jpcap is a Java API that provides access on Windows.

c. **Sigar:** It is an API that provides a portable interface for gathering system information such as:

- System memory and CPU.
- Per-process memory, cpu credential info and, environment
- Network interface detection, configuration info and metrics

This information can be found in most operating systems, but each OS has their own way(s) of providing it. SIGAR provides this work with **one** API to access this information regardless of the underlying platform (hyperic.com, 2012).

d .**Primefaces:** The work shall draw on primeface graph features for its graph presentation analysis, because is a lightweight library with one jar, zero-configuration and no required dependencies (primefaces.org, 2011).

e. Persistent **Storage:** In order for the data recorded by the system to be available after restarting the server, it requires some form of persistent storage. MySQL was chosen to provide the persistent storage, to provide a standalone server that can be embedded into Java applications. Alongside the MySQL server, mysql-connector-java-5.1.7-bin.jar was chosen to provide the JDBC drivers to access the MySQL database from within the application (Ardila, 2008)

In all the works above none of them deals with metrics like delay, bandwidth consumption, congestion and in addition, providing a means of feedback to the network users as the study has graphs, internal help desk and live communicator as well as database for persistent.

CHAPTER 3

METHODOLOGY

3.1. Overview

The problem observed in the study on networks is that users are able to abuse time and resources allotted to them during peak working hours, thus visits other domains/Internet that they are not suppose to do so, resulting in additional load. Therefore, it becomes difficult for administrators to identify the sources of problem like traffic performance degradation. The designed application is a monitoring tool; it is aimed to analyze traffic metrics like bandwidth consumed, delay, packet captured, filtering of packets on network segments using packet analysis, to visually determine if users' of network are adding additional traffic with their activity on the Internet.

The basic architecture of the monitoring tool consists of a client application that end user uses to communicate with the administrators. It is designed to receive network packet findings from administrators, using internal helpdesk. Considering the nature of network environment client application is able to detect, show basic operating system and network configuration information as well as driver status. In addition, the client application is able to ensure basic security mechanisms to provide a degree of anonymity, including display of IP address.

The server consists of different logical instances; the monitor server is in charge to implement the technique to capture and measure user connection characteristics for analysis. Finally, the server has database to store all the captured packets and gathered data for graph.

The server verify the identity and the integrity of the captured packet, classifying the packets according to their subnet or domain of the communicating host, and establish if packets obtained are valid and not affected by external or remote sources.

However, upon visual analysis on the packet length and subnet, the server sends information to the clients concerning their actions on the Internet through the helpdesk.

A number of procedures were put together to formulate the methods, and in addition adopted part of standard waterfall model to help the application design. Secondary resources derived from various publications, including books and journals, internets were included to support our design, analysis and findings.

In the quest to get the major data to buttress the solution, a monitoring application was designed, so that the captured packets in traffic are used as base in answering some of the formulated questions and part of the designed application also used.

Generally, the best and cheapest way to get answers to non-router based problems is passive and active monitoring as is seen in the section 5.1. Using this approach cost less as open source packages (Agyepong, 2010) products are inexpensive as compared to router based.

3.2. **Research Method and Design**

In order to acquire, analyze, and capture the network traffic generated in the monitoring application, a series of methods were adopted to help in answering the questions.

To obtain packet data, an application is designed, to capture packet passing through a network card. In addition, as part of the design requirements, certain tools were also used and these are briefly described in section 3.3 and tabulated in table 3.1.

The requirement analysis and design of the application were done in section 3.5 and 3.7. The design of the application was based on part of waterfall model, which divides the life cycle of the application development process into phases as shown in figure 3.2.

Once the design of packet capture was complete, the next step was to choose the source that would supply this traffic data. Hence, WAN/Internet became the source to collect traffic data. Before analyzing the data, the captured packets were stored into a database and each information/parameter extracted from the packet block captured.

To monitor the activities of users' in a test network and the effects of the chosen metrics on the network parameters from remote sources, an application designed in section 3.8, a test-bed illustrated in Fig. 3.1 were used in turn and a series of packets were captured and analyzed using the designed application.

The capturing was performed in two cases: homogeneous case, where the entire hosts are in the same subnet or domain and heterogeneous case, where some of the entire hosts were in different subnets or domain.. This was done in order to make meaningful analysis, by gathering data from actual network traffic, taking packets from the network and reading them.

The time, destination and source IP address, packet length, hop count, identification field packet type and windows size of packets were noted for every capture session through both the homogeneous and heterogeneous case in turn. Each capture was run once in which case the values for delay, captured packets, bandwidth consumption were analyzed based on the relations in section 3.9.

The following assumptions were made in the course of the demonstration

1. The bandwidth was assumed to be the same for all connections route.
2. Transmission delay between sources and destination pair was measure by packet length (bits) per link bandwidth (bit/s).

A set of requirements were specified as in section 3.5. The application had to record packets to permanent storage so that analysis could be performing on the data. In

addition, the study maintained records of various packets captured so that analysis was made out of the traffic pattern..

In order to determine what kind of traffic pattern each node generated, we took a number of steps. First and foremost, there was gathering of traffic data. With this data in hand, there was analysis of the data from the output obtained with regards to the chosen metrics. Using this data, the next step was to determine if the data gathered was enough to proceed to graphing.

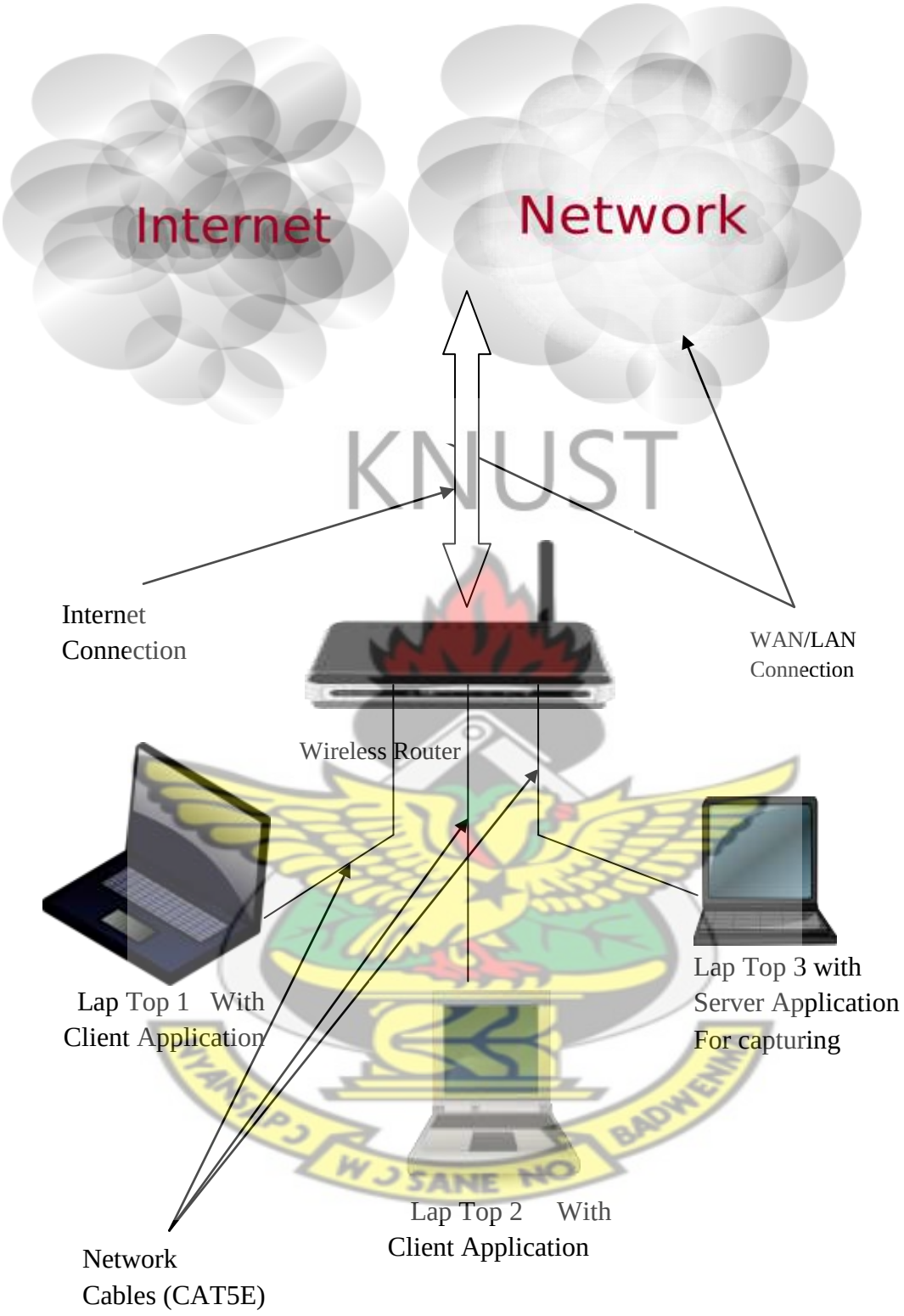


Figure. 3.1 Test and Demonstration Setup

(Source: adopted from Clker.com, 2012)

3.3. **Development Tools and Technologies Used**

One of the most common flaws of the open source is repeatedly duplicating software packages because the existing packages do not have the set of features required. In this work, it was decided that instead of writing several components from the scratch it would be faster and efficient to make minor modifications to existing packages rather than implementing the same thing again. The tools and technologies used are tabulated in table 3.1 as shown below.

Table 3.1. showing Tools and Technologies

Resource/Tool	**Purpose**
Microsoft Windows 7	Operating System
Microsoft Office 2007	Documentation
Netbeans 6.9.1	IDE
Mozilla Firefox	Default Browser
JDK (Version 6 update 21)	Programming Language
Prime Faces	User Interface Development/Graphs
Wincap & JPcap	Windows & Java packet capture API
Mysql	For Database/ Persistence

(Source: adopted from Clemm, 2007)

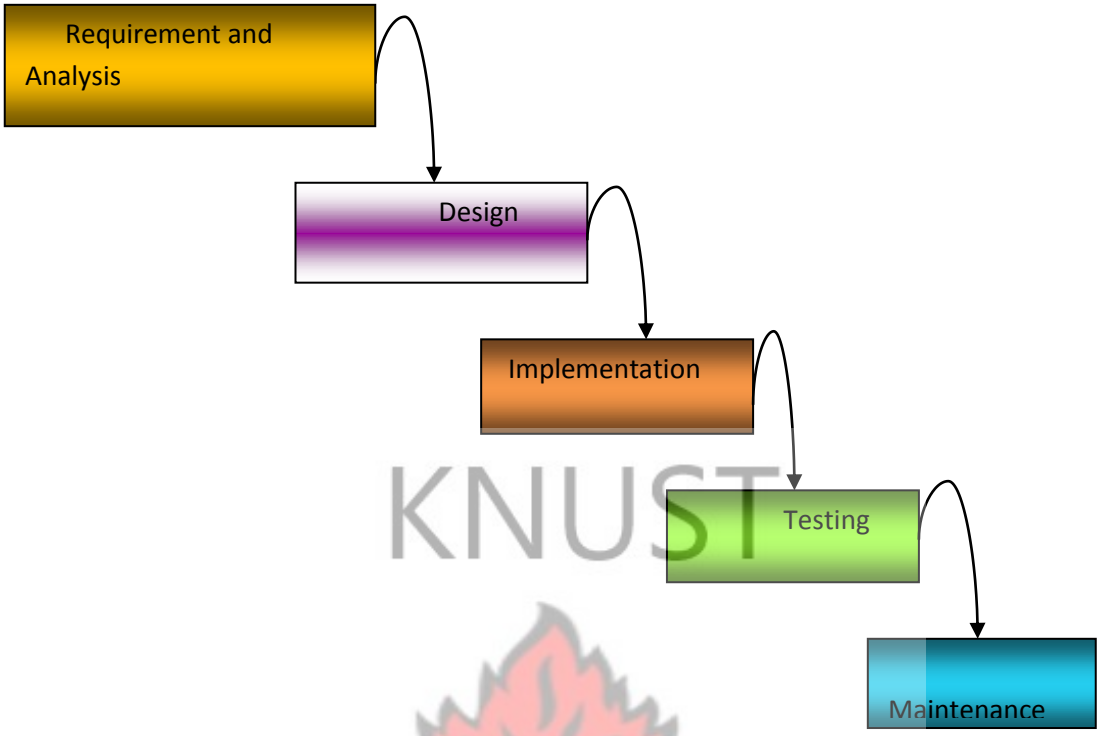


Figure 3.2: Classical Waterfall Model (en.wikipedia.org, 2012)

3.4. **Description of Modules**

a. **Login/logout for Administrator and Clients:**

Require login and logout credentials before and after accessing the Application for both Administrator and users. Require username and password to identify authentic users for the purposes of security. Third party encryption was adopted to make it work successfully.

b. **Packet Capture:** Input from the network is handled by jpcap and is responsible for capturing the packets emanating from the card through air or wire. This component is responsible for doing the basic first level of analysis of all packets captured. In addition, it has the following sub module, capture view, packet capture analysis, load captured packets, view packet block, view graph and view packet details as its sub modules.

c. **Utility:** The network utility module has the following sub modules to do active monitoring:

- Network statistics
- Routing table
- Ping

d. **Internal Helpdesk:** A typical help desk provides the users a single point of contact, to receive help on various computer issues. The help desk typically manages its requests via help desk software, such as an issue tracking system that allows them to track user requests with a unique number (Wikipedia, 2012). This part takes care of the communication issues and sends messages to users when they go online. Also they assist them in resolving network issues. This module bridges the gap between the server side and client side.

e. **Network Interfaces**: To be sure, of interface status, this module always gives information about all hardware and software network interfaces, whether they are up and running or not.

3.5. **Requirements Specification**

This is the requirements document for the application. The system to be developed is for capturing the packets flowing in the network and analyzes them, use help desk to communicate messages between users, and monitor the status of the network interfaces as well as visualize network configuration information with ease. The information in the packets is analyzed, and saved into a database.

3.5.1. **Scope**

The target groups are network administrators and organizations who are meant to know the network flow, in and out, of the system and provide accountability for network use.

3.5.2. **Functional Specifications Requirement**

The functional specification is described in table 3.2 where its main components are shown, including a brief description of the component and the major function. The idea is to state the main objectives and the functions related to the overview of the system, stating the main system blocks necessary to build the monitoring system able to characterize packet traversing in a network. The system through the Server and client performs the functions in the table.

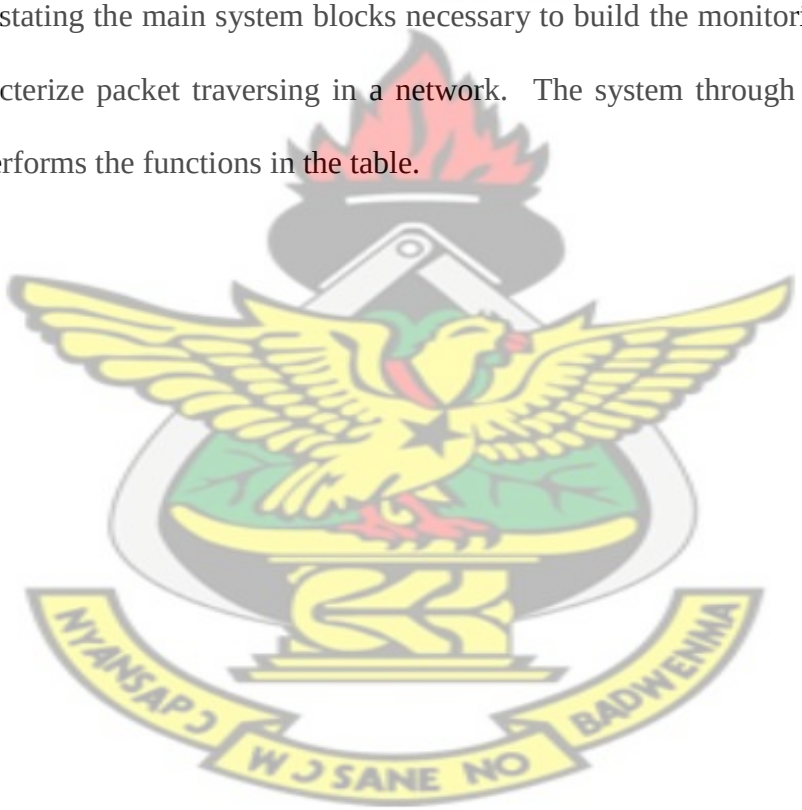


Table 3.2 Functional Specification (Source: adopted from Arjun et al, 2008)

Component	Description	Related Components
Packet capture (Server)	Captures packet from the NIC	Analysis and GUI user component
View Packet block (Server)	View the captured packet block before Separating into parameters	Analysis and GUI user component
View packet parameters (Server)	Views detailed packet parameters Required for analysis	Analysis and GUI user component
Packet analysis (Server)	Present analysis of the packet captured	Analysis and GUI user component
View graph (Server)	Allows you to view the graphs obtained from the metrics	Analysis and GUI user component
Internal helpdesk (Client)	Allows communication between server and client, users of network request, create, receive messages	Both client and Server
Network configuration on client (Client)	Allows easy access to network configuration	Client application
System specification on client (Client)	Allows easy access to basic operating system specification (OS type and version, CPU, DNS and memory details.	Client Application

3.5.3. **User Requirement Specification**

These are specification of requirements the users expects from the system. These are as follows:

a. User Characteristics: The users of the system are in two categories that is the systems administrator who creates user accounts, access rights, and controls, monitors the network traffic through the server, and also monitors users activities as well responds to users' needs**.**

The second category is the client whose activity is monitored and sort for help in times of IT related problems. These two groups communicate using internal helpdesk.

b**. General Constraints and Assumptions:** The assumption is that the packets moving in the network are coming from only wired or wireless network.

3.5.4 **Non Functional Requirements**

A non functional requirement specifies criteria that can be used to judge the performance or operation of a system, rather than behaviours.

Non-functional requirements are often called qualities of a system. It can be divided into two main categories.

- Execution qualities, such as security and usability, which are observable at run time.
- Clarification of Portability and Accessibility Requirement

In today's geographically distributed network environment, the ability to gain access to monitored objects from a number of different locations is becoming increasingly important. The application uses web based which makes accessibility easier.

Graphical User Interface

Specification of interface requirements that is clarification of user groups and levels of access.

The user interface for this type of system should present the required information in a clear and concise format, giving accurate and timely information to the user when requested.

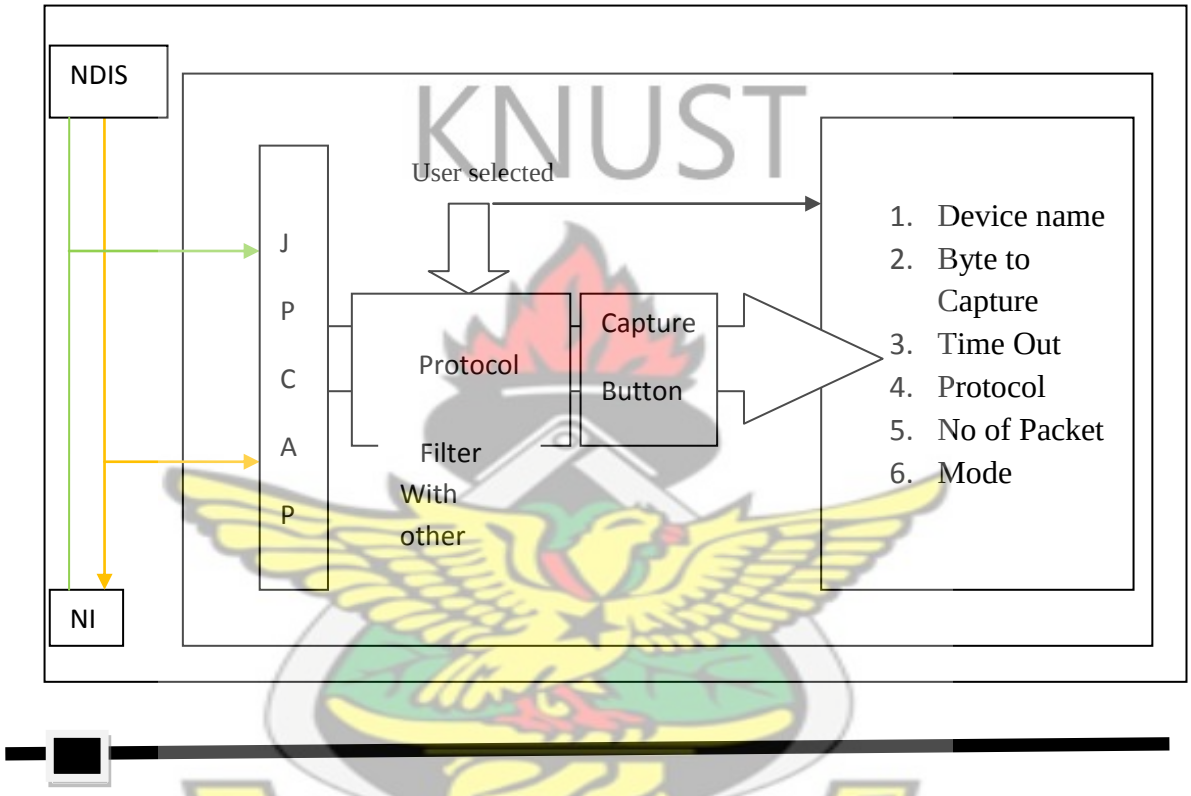


Figure 3.3: Block Diagram of the packet Capture component interface

(Source: adopted from Shintaro et al , 2009)

Figure 3.3 is the block diagram of the capturing component interface. In the diagram, an external site or packet traverses the network with thick dark line. The packet through the network is captured by selecting the parameters in the figure.

a. **Software Environment**

This involves the unseen side of the system. This is the side which supports the system as shown in table 3.1

b. **Hardware Environment**

This involves what the system runs on. This allows the user to interact with the system hardware. It is also known as the physical components of the system. They are:

Processor	:	Intel Pentium P6200 @ 2.13 GHz
RAM	:	2.0 GB RAM or higher
HDD	:	50 GB or higher
LAN	:	Enabled

Performance Constraints

The speed of the networks should not exceed 10/100Mbps for LAN.

3.6. **Design and Operation Requirements**

3.6.1. **Conceptual design**

The work used UML modeling for the design which shows different system components and how data flow from one component to another to achieve the systems goal.

3.6.2. **Use Case Diagram and Description**

The use case below gives the general overview of design and operational requirement.

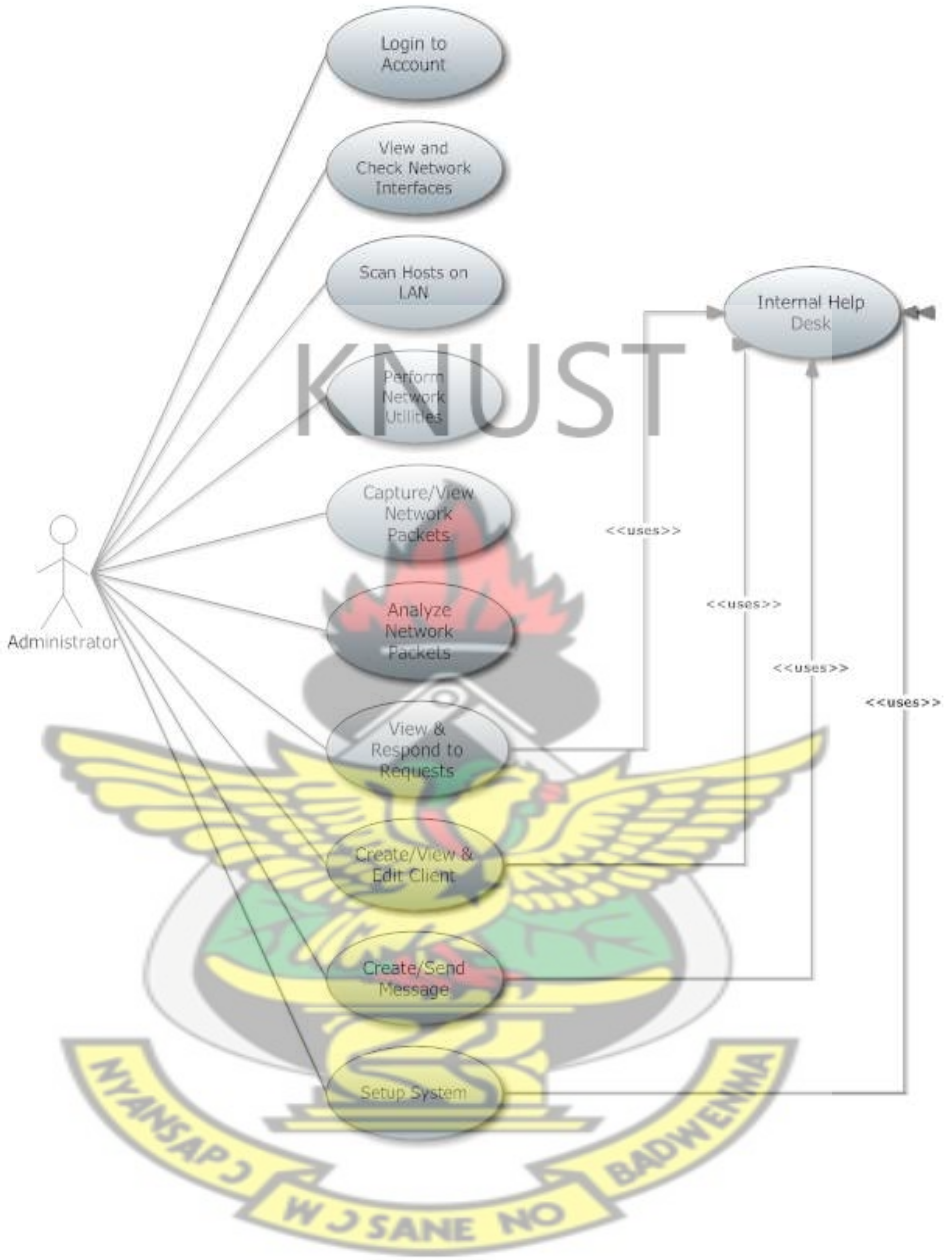


Figure. 3.4 Use case Administrator and system (designed by author)

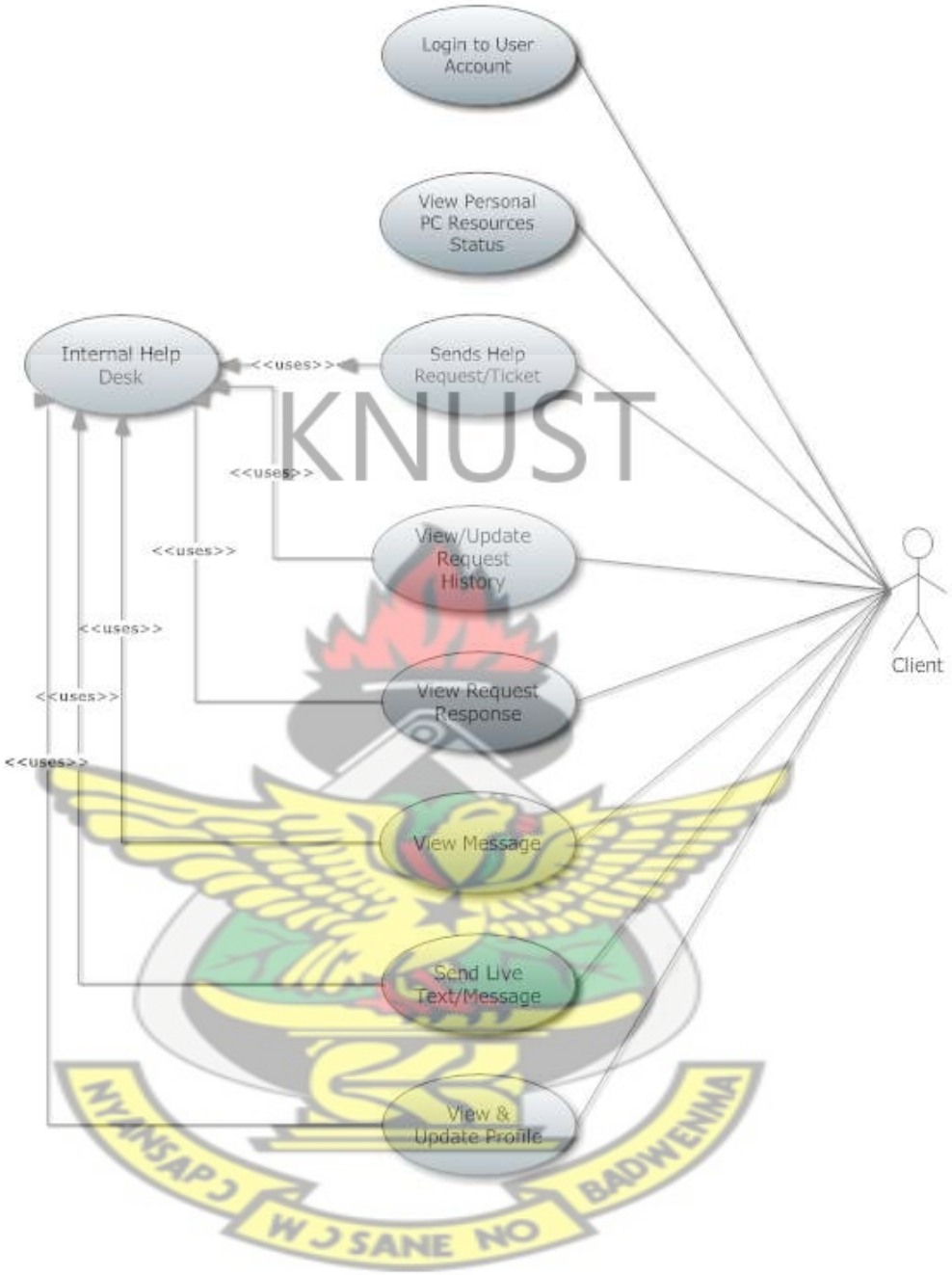


Figure. 3.5 Use case Client and system (designed by author)

Table 3.3 **Use case Description**

ACTOR	ACTIVITY
Administrator	• Manages the user account and login as well. • View and check network interfaces • Scan host on LAN • Selects interface card • Filters • Perform network utilities • Capture packets • View and respond to internal help desk from clients. • Create, view and edit clients accounts • Create and send messages • Setup system
Client	• Login to user Account • Views personal pc resources • Send helpdesk request or ticket • View update request history • View request responds. • View message • Send live text/ message • View and update profile

(Source: adopted from agilemodeling.com, 2009)

3.7. **Design**

The study looks at the following application design objectives:

3.7.1 UML Diagrams

a. Use Case

The diagram below represents the Use Cases of the capturing and monitoring tool. There are two main actors: the External network or packet feeder that is different websites and subnets as source of packet to perform the capturing and users that use the Database that forms the captured data store (sparxsystems.com, 2011)

Considering the main requirement of obtaining network parameters and measures, the tool architecture in figure 3.6 has been designed to perform varying operations in order to suite a monitoring tool (Arjun et al, 2008).

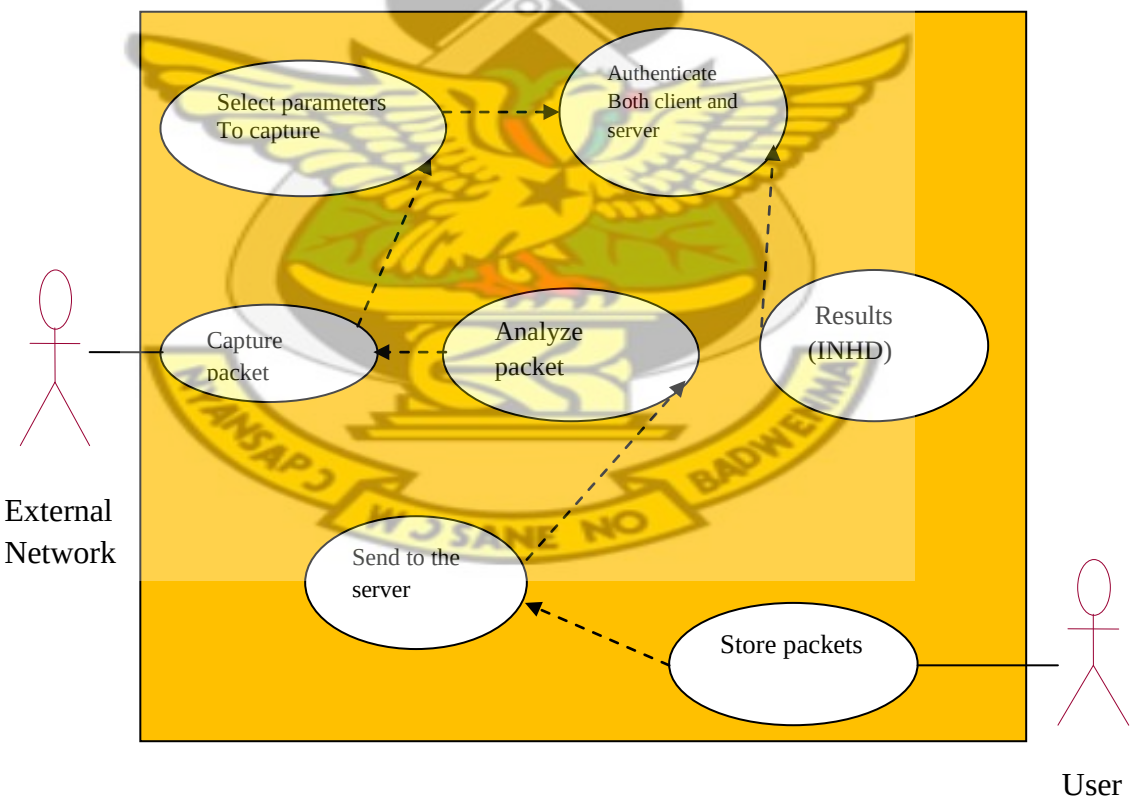


Figure 3.6 The Architecture: using Use Case

b. **Data Flow Diagrams**

Data flow diagrams are made up of a number of symbols, which represent system components. Most data flow modelling methods use four kinds of symbols. These symbols are used to represent four kinds of system components: processes, data stores, data flows and external entities (visual-paradigm.com, 2012).

The data flow diagrams for this work are shown in figure 3.7, 3.8 and figure 3.9 below. It is the data flow diagram for the entire packet capture process. It specifies the major change centers in the approach to be followed for producing the application. This is the first step in the design method. In this work, the inputs are the packets that are flowing on the Internet/ network that are captured through the network interface card set to promiscuous or direct mode. The output is the information contained in the packets in human readable form, which is stored in the database.

The context diagram and data flow diagram of the application are given as follows:

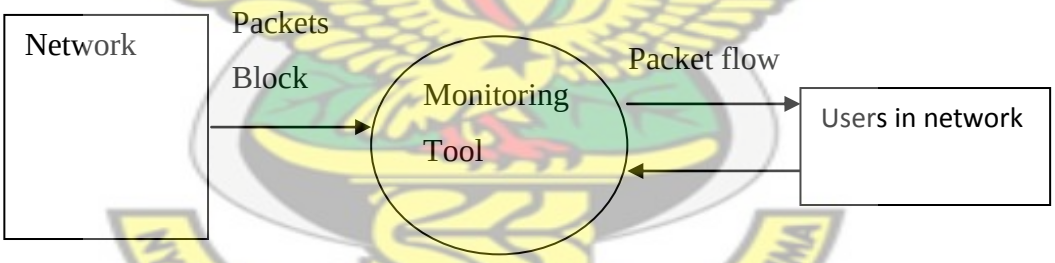


Figure 3.7 Context Diagram

(Source: adopted from modernanalyst.com, 2011)

The diagram above which is figure 3.7 explains the principle behind the work with which object the network monitoring tool interacts.

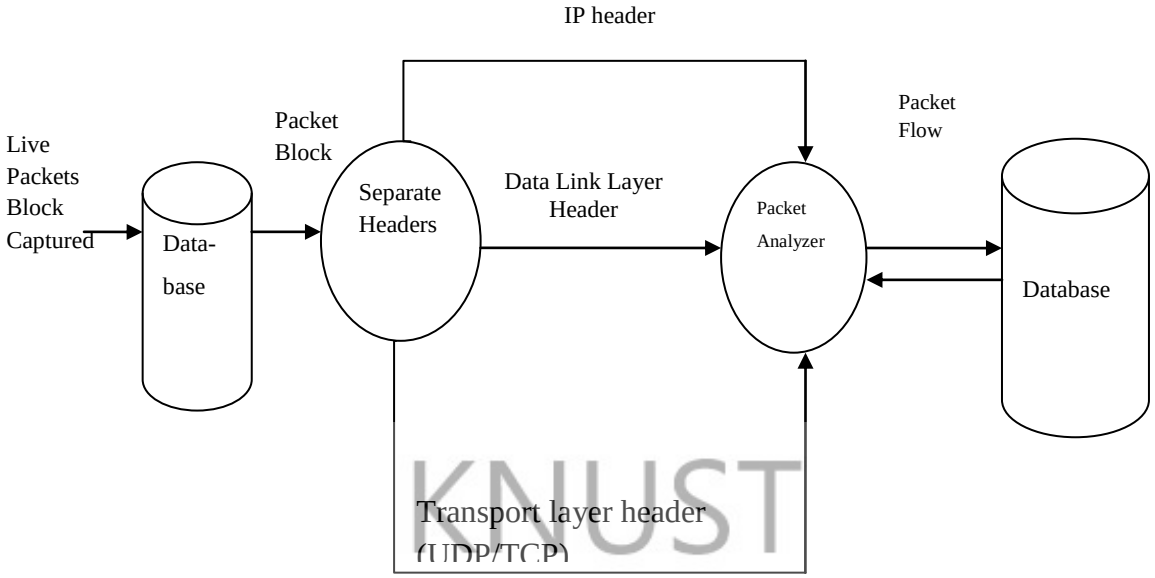


Fig 3.8: DFD for the Packet Analyzer process

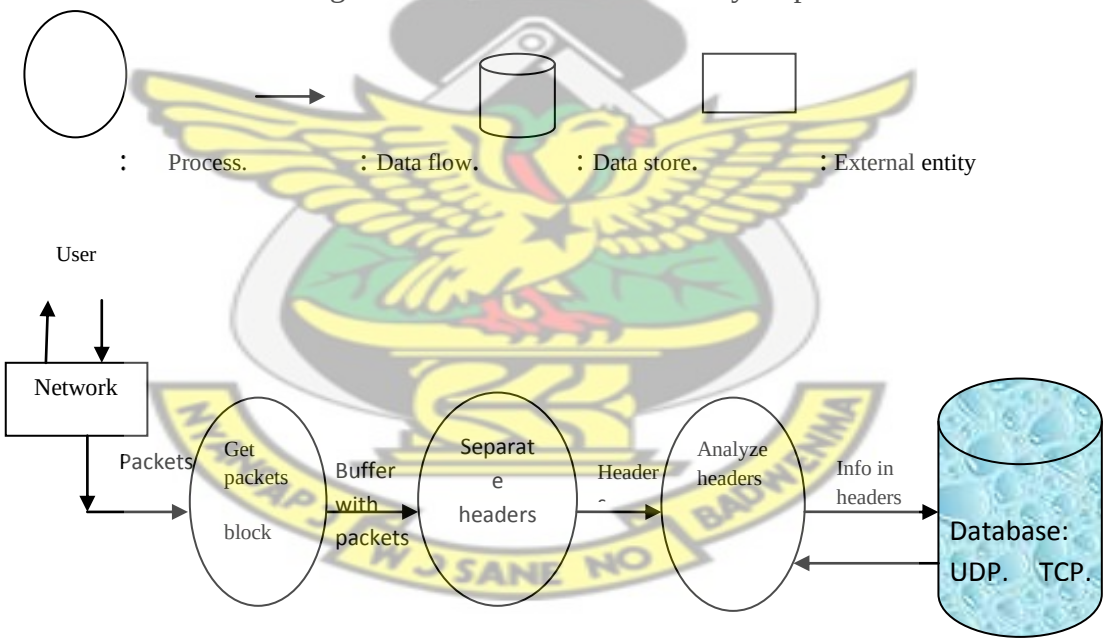


Figure 3.9 Data flow diagram for Packet Analyzer

(Source: adopted from sparxsystems.com, 2012)

From the diagram the input is obtained as packets through the network interface by the 'Get packets' process. This process defines a list of interface devices, obtains the raw packets from the network interface device selected, and stores them into a buffer

using JPcap. The buffer containing the packets is passed to the 'separate header' process, which strips off various headers of the packet and passes them to 'analyze headers' process where they will be analyzed and the information is passed on to the database process and store in a database. Here the output file is be updated with the latest information obtained from the subsequent captures (packetlife.net, 2011).

c. **Application Structure chart**

For a function-oriented design, the structure chart represents the design graphically. The structure of the tool is made up of modules and methods together with the interconnections between them. The structure charts of the application are a graphical representation of its structure. In this structure chart, a box represents a module or method with the module name written in the box and an arrow with an empty arrow represents the packets data one module or method passes to each other. The parameters returned as output by a module or method (hit.ac.il, 2012).

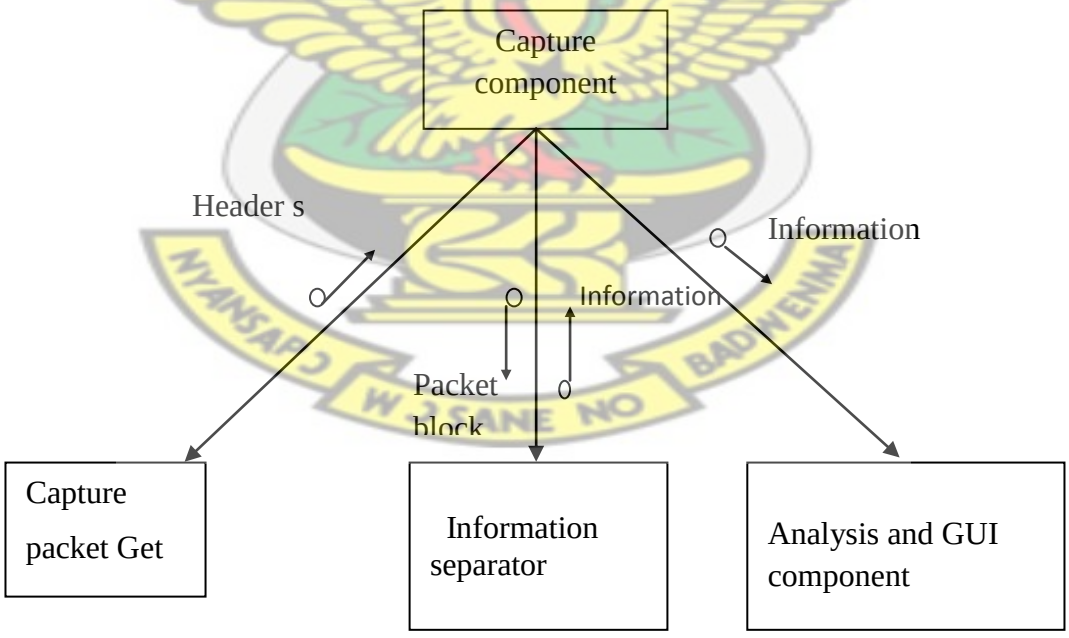


Figure 3.10 application structure (rational.com, 1997)

From the application structure chart in figure 3.10, there are three modules/methods, one for input, one for output and main control module which is capture component which controls the basic capturing of packets in the system. The main module's job is to invoke the sub modules/methods (hit.ac.il, 2012).

Here, there is one input packet capture, which returns the packet block in the packet to the capture component. The capture component passes this block to the information separator which processed them into human readable information. This information is passed to the capture component. The capture component module passes this information to the Analysis and Graphical User Interface which is made up of view packet block; view packet details and view graph analysis as output from the database.

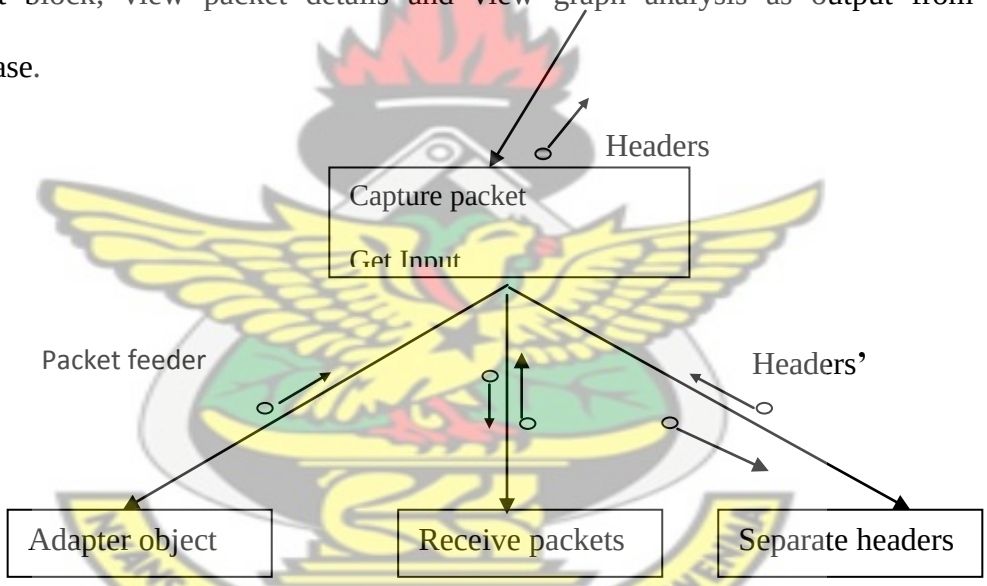


Figure 3.11 Capture input structure (rational.com, 1997)

From 3.11, the packet capture input, the network interface is turned into promiscuous mode or direct mode so that all the packets can be captured even though they are not intended to it. This is done by defining an adapter object as shown in figure 3.11 and reading all the packets which comes in a form of block using JPcap. Then each packet is taken and the various headers are separated by separator method and sent back to

the capture component. The protocol/packet analysis module is shown in figure 3.12. In this module, the process is split into the detail packet parameters that are IP, ARP, TCP and UDP. The modules are named after the type of headers they handle. Each module knows the specified format in which the information in that particular header is stored, so they convert it into required format by which we can easily understand and know about the packets in detail. This information is passed to the capture component, with the database support.

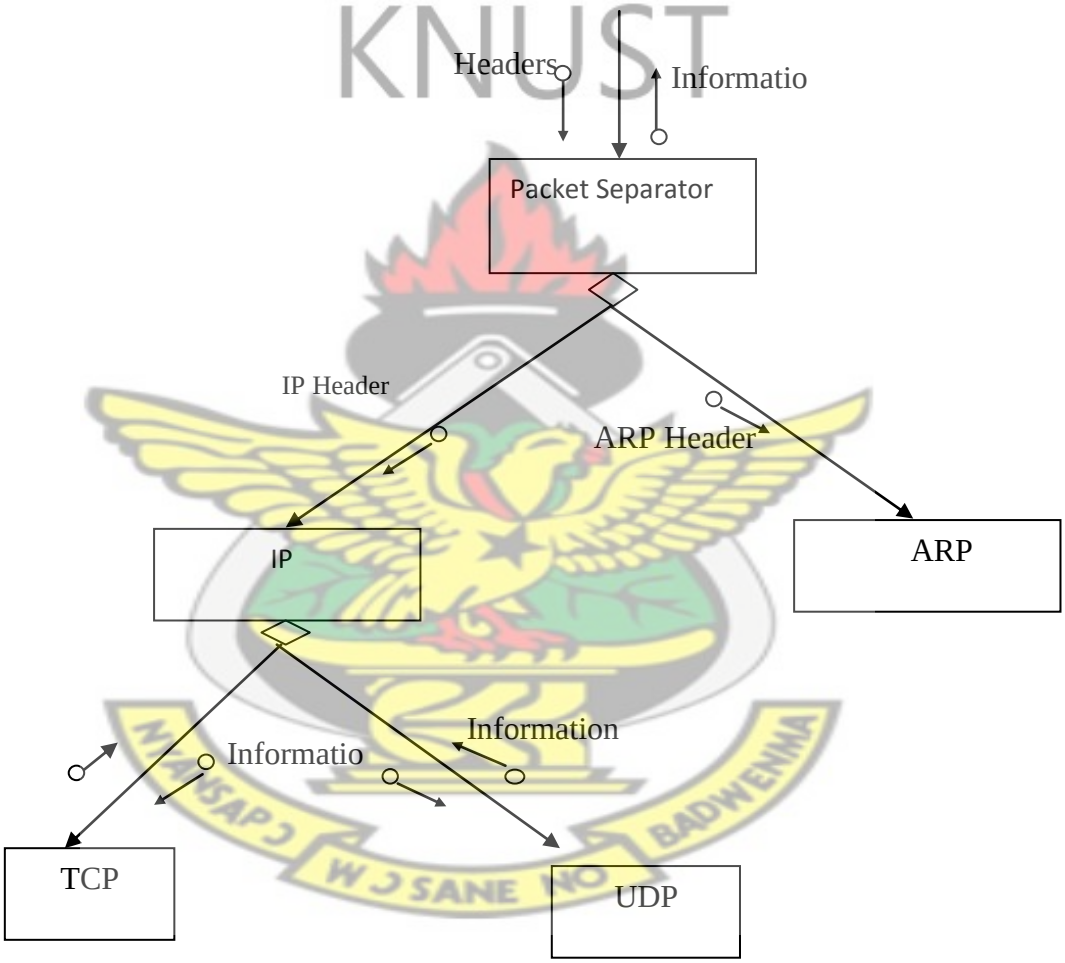


Figure 3.12 Packet Separator (rational.com, 1997)

The view packet block, load captured packet and view packet graph analysis gets the information stored in the headers of the packets as input from capture component. The output module is Analysis and GUI which contains the load captured packet view

which controls view packetInfo, view packet block, and view graph. The load captured packet lets you load packet from the database, view packet block and detail packet parameters and view graph analysis lets you view the analysis in graphs form.

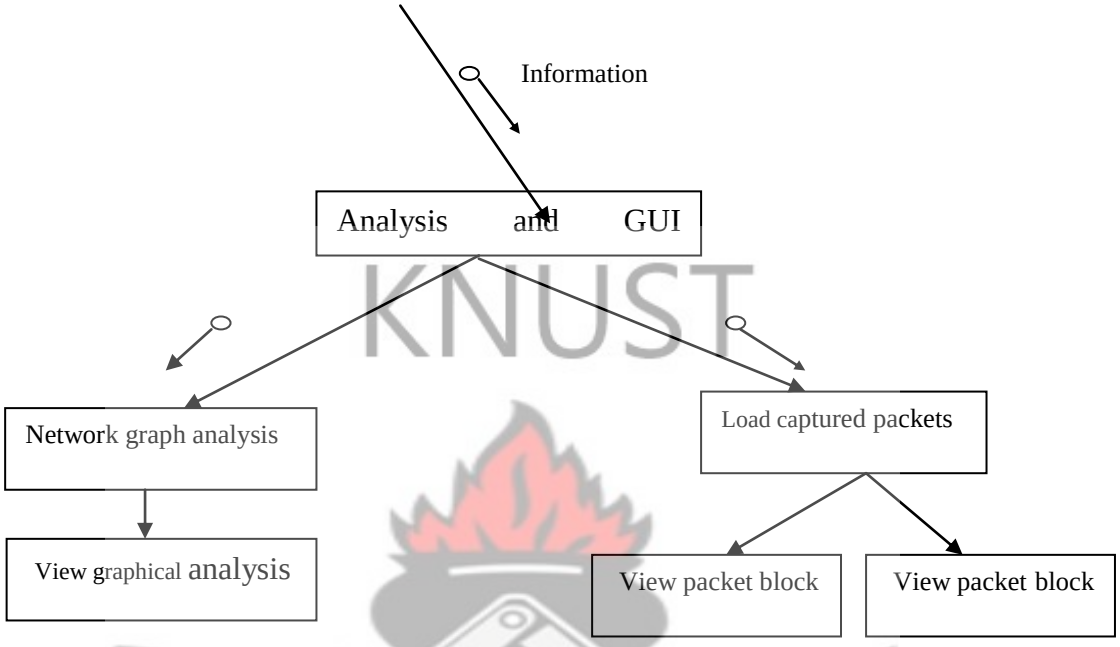


Figure 3.13 Analyses and Graphical User Interface

(Source: adopted from en.wikipedia.org, 2012)

The figure 3.13 above is the output that aid in the analysis of the detail packet parameters, where the performance metrics are applied. In this module, source and destination IP address, packet type, Date, packet length, hop count, identification number, window size and time are display by the view detail packet information as the expected results for the monitoring (tcpguide.com). However, the View graph implements the performance metrics on the packet parameters.

a. **Activity Diagram**

The activity diagram in figure 3.14 shows the high-level action the tool does. The diagram flow process allows designing a systematic path with little user interaction, and the actions followed in sequence to produce an application that does not consume many resources to avoid disturbing the user when the monitoring tool is running (tutorialspoint.com, 2012).

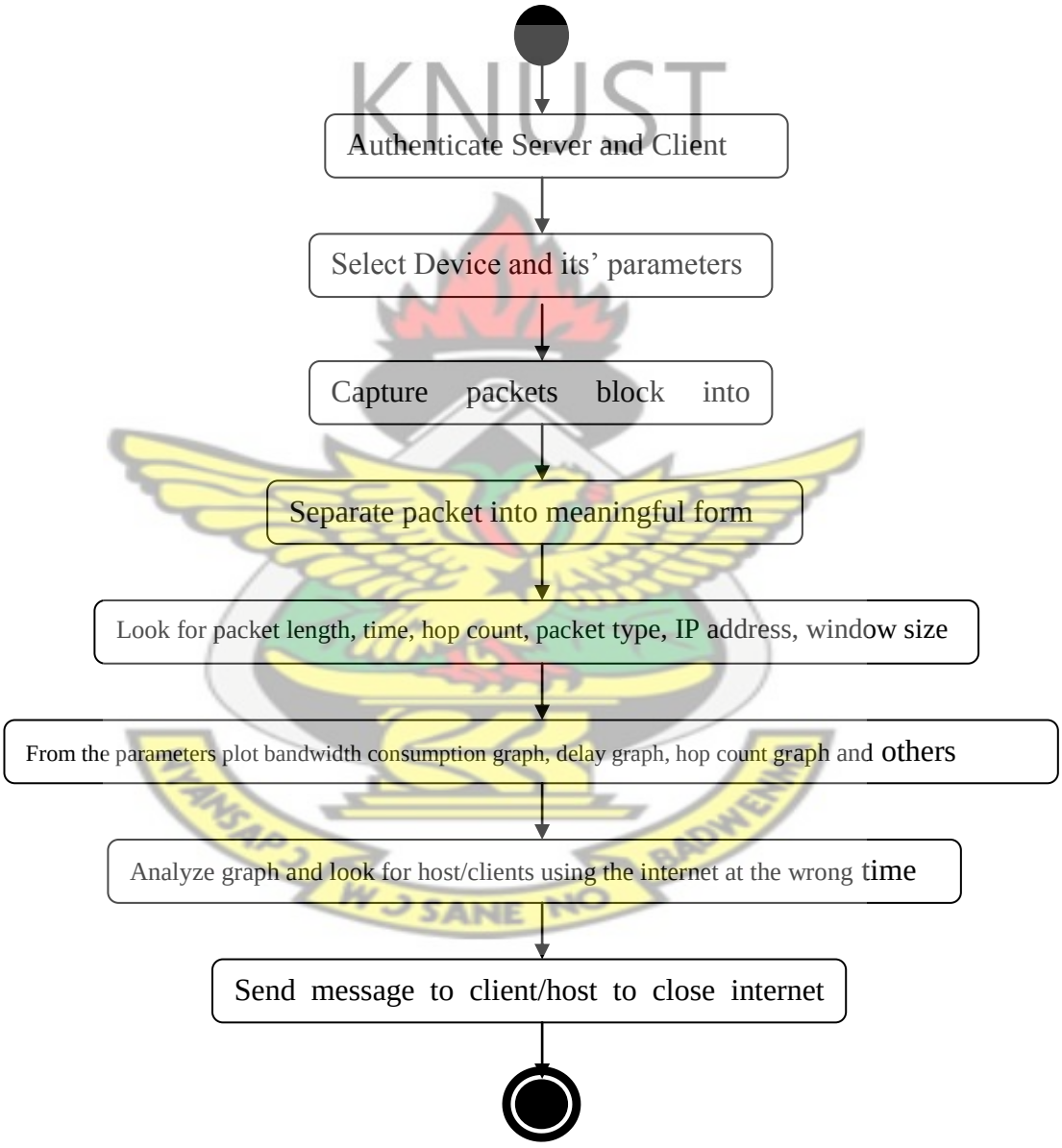


Figure 3.14 Activity Diagram (tutorialspoint.com, 2012)

b. **Class Diagram**

The design of the class diagram in figure 3.15 is based on clarity, modularity, and scalability to allow the system to be easily understood, grow and add more features (agilemodeling.com, 2011). The detailed class diagram is on the attached CD.

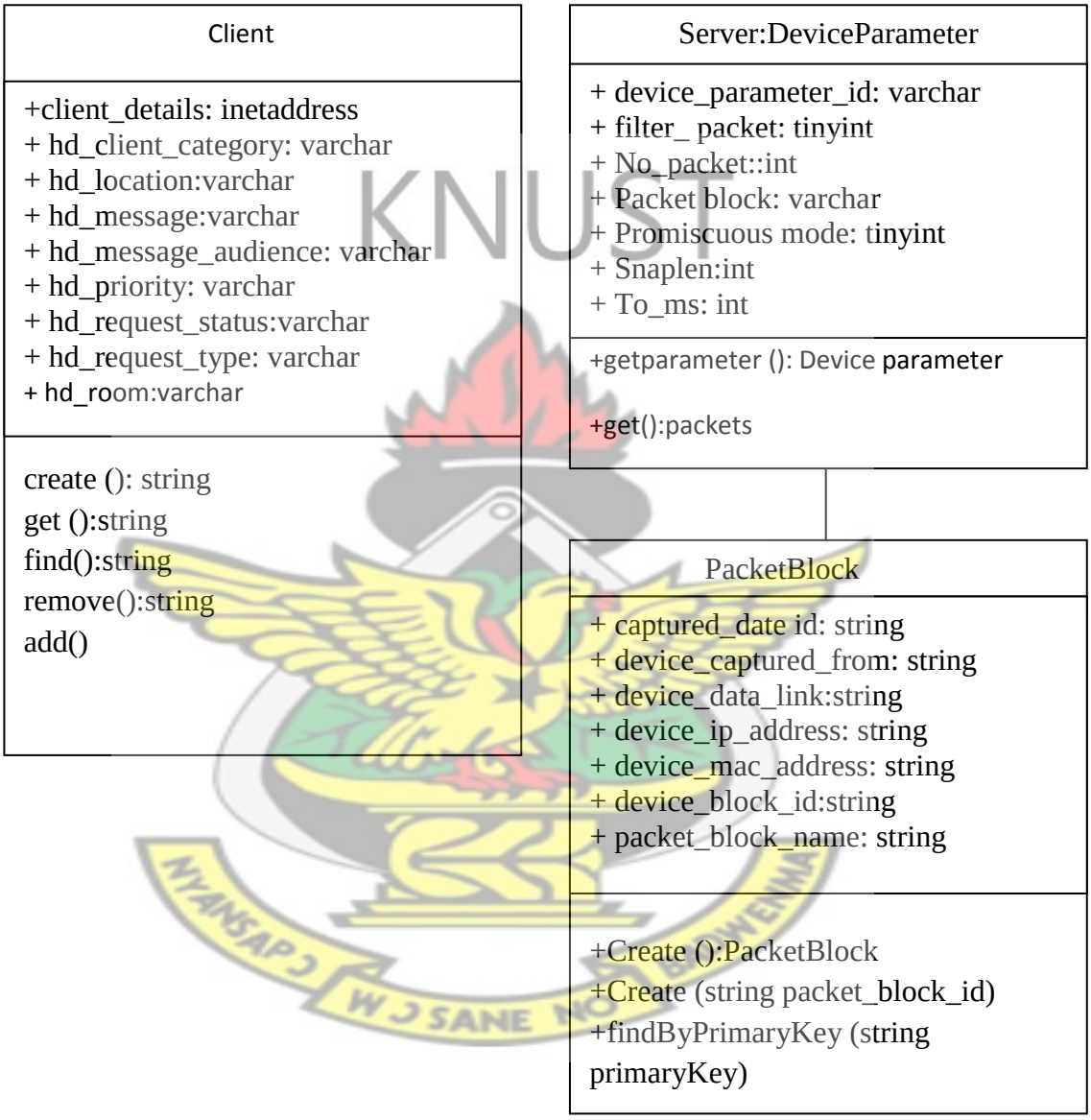


Figure 3.15. Class Diagram

3.8. **Database**

In this part, some of the needs of the database system are stated in order to gather the captured data and messages for the monitoring system. In addition, the detailed schema is shown on the CD, tables and entity relationship diagram for the work are shown on the CD.

MYSQL version 5.0 community server edition databases is used for the following purposes namely, storing generated packets, helpdesk messages, captured packets and comparing generated and captured packets as an evaluator. For database connection via javaee platform, mysql-connector-java version 5.1.7 driver is used. Generated packets are stored in a database table, created with sql for device_parameter, packet_info, helpdesk messages as seen in Tables on CD. Furthermore, all generated graphs are also stored in the database.

a. **Server Side:** The Database provides information security to avoid unnecessary activities over the data. At the application level, authentication and authorization features should be implemented to control user login.

b. **Client Side: The** security at the client side is limited to the communication between the server and the client in the network. To avoid masquerading from both client and server there should be two-way authentication.

To connect to the database with the monitoring application, both server and client, database driver is used. However, the use of an Application Programmable Interface (API) help the tool to connect directly with the database to insert the data needed.

c. **Entity Relationship Diagram**

The entity relationship diagram of the entire work is on the CD.

It describes how the entities in both the client side interact as well as server side.

3.9. **Performance Indicators**

The concept of bandwidth is a key metric in networks, is related to the amount of data a link can deliver per unit time. In data-hungry applications such as file transfers (batch processing) and multimedia applications, the available bandwidth directly affects its performance. Furthermore, an important bandwidth metric is the data rate or throughput of TCP connection. TCP throughput metrics are very important for end users, because it measures the data rate it can achieve. However, various factors influence TCP throughput, but the work looks at transfer size and TCP window size at sender or receiver sides, (Ardila, 2008).

Each transmission unit consists of header and actual data. The actual data is referred to as MSS (Maximum Segment Size), which defines the largest segment of TCP data that can be transmitted. Essentially,

MTU=MSS + TCP and **IP headers.**

For the purposes of this work and all comparisons the following assumptions are made (Filipov, 2000):

- MSS=MTU-40 , thus , a standard 40 byte header (20 byte IP and 20 byte TCP)
- packets are not being fragmented
- no packet loss
- no router congestion

a. **Packet size delay and throughput**

Using scenario transfer of 1,500,000 bytes of data using different packet size over a line (link speed=1,544,000 bits/sec) using the following formula:

(MSS+header)*8bits/byte/1,544,000 bits/sec =delay (per hop)

Thus, using different MTU values, we can calculate the relevance of packet size to delay for the work.

Let TD represent packet transmission delay

Packet size be PS

Transmission speed be TS

Packet transmission delay, TD = packet size/transmission speed (1)

TD=PS/TS

From relation 1, higher TS means smaller delay value

P =TD/TS, which implies, that bigger PS higher TD (2).

Bandwidth=packet transmitted in byte /time in seconds (3)

Let BW represent bandwidth

Let PT be Packet transmitted

T be time

Hence, BW=PT/T, an increase in T affects BW by wasting BW

This also means lagger values directly affect BW

b. **Calculating TCP throughput**

Let TW be TCP Window size in bits and Round Trip Time RTT

Then throughput, TP = TW/RTT (4)

This means an increase in time will decrease the TP.

From relation (4), it means an increase in TW, increase the TP.

It can be conclusively, established that, TW affects TP.

CHAPTER 4

IMPLEMENTATION

4.1. Overview

This research was conducted to monitor network users and to give accounts of Internet use, using packet analysis. The design and descriptive method of research was applied. Through packet analysis and qualitative approaches, an application was designed and developed to gather data.

Literatures to support the objectives and findings were also reviewed and included.

After gathering the captured packet from the implemented application in section 4.2 using the demonstration setup in chapter three, the data for the analysis in graph was obtained. The data captured from the application were presented in tables to facilitate the analysis. The results obtained were plotted in graph in order to start the analysis using ***cause*** and ***effects*** (source***:*** thinkreliability.com) of ***packet analysis*** (Coull, 2007). Furthermore, the information in the packets captured by the application was the base data to plot graphs for analysis and findings on the data.

Finally, tests designed to measure the accuracy of the captured data were ran as seen in setup in chapter 3.

With a solid understanding of the patterns found in the data and graph, the building of conclusion was made in chapter 5.

Once the logical design is constructed leading to the proposed tool constructs, there is the need to implement the features involved during the design phase in this chapter. However, the implementation of the application is to capture data for the analysis and communicate the results using helpdesk.

4.2. **Analysis**

This section describes implementation of the packet monitoring tool. It describes the two major components developed for the application used.

The main components of the system, capture component, Analysis and GUI component, and internal helpdesk consists of login interface, dashboard for both server and client, and graphical user interface, with the back end performing the necessary class and method operations through the database.

As already stated in earlier chapter concerning requirement, this work runs in a web based environment and therefore there is the need to adhere to the stated tools in chapter three (3) required by the system.

The major class structure of the monitoring application being implemented is the capturing component and internal helpdesk.

Findings and Packet Analysis

This section deals with the packet block and the information in the block.

Figure 4.1, figure 4.2 below aids and shows the packet block captured in the network and figure 4.2 shows the information extracted from the block through the Analysis and GUI.

Information in figure 4.3 and figure 4.4 are the source document for the table 4.1 and table 4.2 in which the graphs are plotted using absolute packet parameters. The tables were used for the purposes of clarity.

a. **Network Hardware and Software Interfaces Verification**

At any point in time, the network hardware and software interfaces gives you the configuration status as whether is up and running or not.. The software part takes care of drivers and hardware takes care of interface card. Before any packet capture is performed this component helped us to verify if both the driver and hardware are working properly. From the screen capture in the smaller window below figure 4.1 , Atheros AR8152 PCI-E Fast Ethernet controller(NDIS 6.20) with multicast support, maximum transmission unit of 1500 megabyte and MAC address of 18-03-73-60-28 were verified as whether it was running or not before the selection were made. This occurs at the administrator's side.

Figure 4.1 Network interface card and driver verification status

b. **Device Parameter Selection**

The figure 4.2 below allows the administrator to select network interface to capture packet from. In this implantation, Atheros LIC PCI-E Ethernet Controller card was chosen as shown in the screen capture below. However, maximum number of byte to capture was 65535 and a timeout value of 200ms

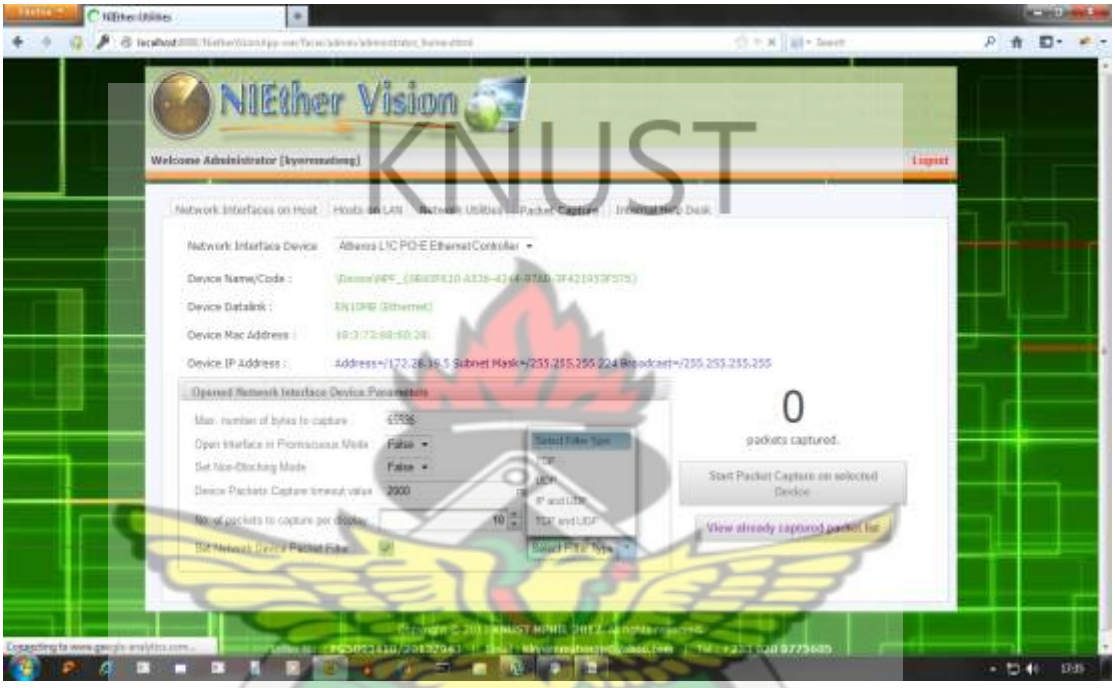


Figure 4.2 Network interface device and device parameters

c. **Packet block**

The screen capture below is the raw packet from the wired or wireless card before extraction

Figure 4.3 Packet block captured

d. **Separated packet or Extracted Information**

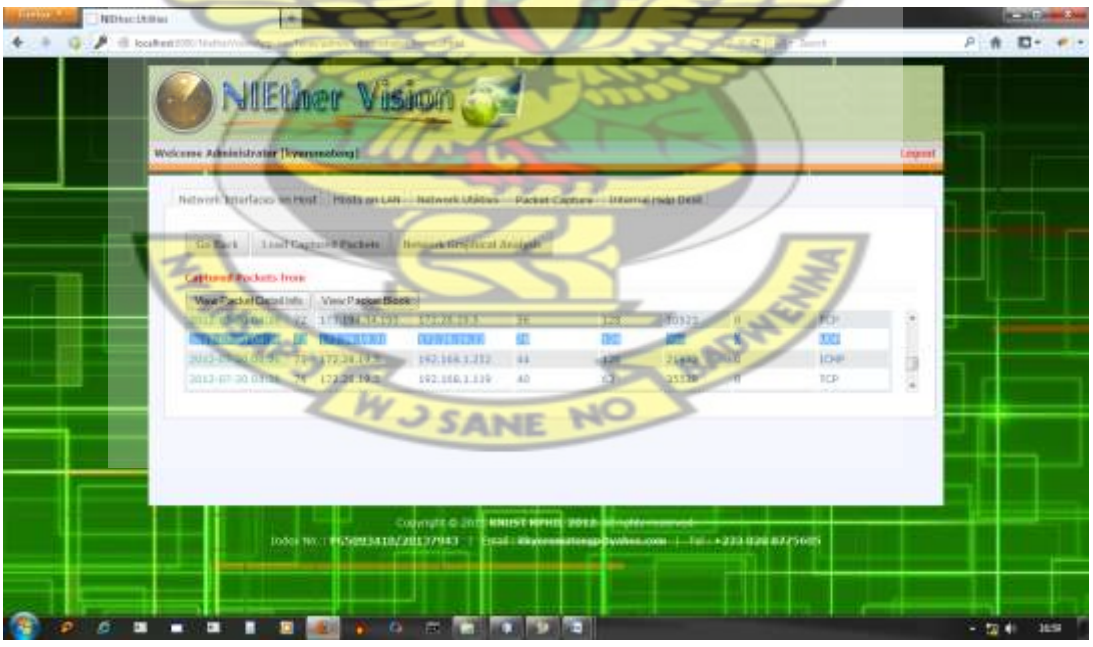


Figure 4.4 Extracted information with packet highlighted

As was stated in the design stage, more specifically, for each capture, the packet contains the following sets of data as shown in both tables below. The data in tables 4.1 and 4.2 were captured on 20/07/12 at 4:26: pm. Even though one hundred and fourteen (114) packets were captured within 2000 milliseconds, only ten (10) are shown in both tables, all others are shown in the graph.

4.2.1. **Part of Packets captured Results**

Table 4.1 Packets traversing in the same subnet within the same WAN

SOURCE IP ADDRESS	DESTINATION IP ADDRESS	PACKET LENGTH	HOP COUNT	PACKET TYPE	IDENT NUMBER
172.28.19.8	172.28.19.31	78	128	UDP	420
172.18.19.11	172.18.19.31	78	128	UDP	45889
172.18.19.12	172.18.19.31	60	128	TCP	10543
172.18.19.22	172.18.19.31	78	128	UDP	1003
172.194.34.195	172.28.19.5	60	128	TCP	1049

From the table 4.1 each packet has an identity number that aid in fragmentation and defragmentation. The variation in number means they packet of their own.

At the TCP/P or OSI model there are different packet types that aids data transmission, during the capture, the packet types shown in the table were in transits.

The IP address with network addresses 72.28.19, are within the same LAN or network

Table 4.2 Packet Traverses other subnet/ Domain within the same WAN

SOURCE IP ADDRESS	DESTINATION IP ADDRESS	PACKET LENGTH	HOP COUNT	PACKET TYPE	IDENT NUMBER
172.28.19.5	173.194.34.196	60	128	TCP	10491
192.168.1.10	172.28.19.5	56	62	ICMP	12152
172.28.19.8	224.0.0.252	54	1	UDP	425
172.28.19.5	192.168.1.170	44	255	ICMP	10499
192.168.1.119	172.28.19.5	44	63	TCP	35533

Considering the captured packets from the output in table 4.1 and table 4.2, Table 4.1 represents packets traversing LAN and table 4.2 represent packets traversing WAN. In addition, table 4.3 shows the constituent packet type of the total captured. Tables 4.4 to table 4.6 were also randomly selected from the packets captured to represent packet size, time, hop count window size and delay. All the tabulated parameters below are shown in the graphs.

From table 4.2,all the IP addresses with network address 172.28.19, are in the same subnet, 192.168.1 are in the same subnet, 224.0.0 is also in different subnets and 173.194.34 is also in different subnets.

From the same table 4.2, Packet lengths are varied because the nomenclature of packets varies because of the payload size, header size and packet type padding. It comes as results of what data is being sent across the network and what packet type.

Table 4.3 Packet type and number captured

PACKET TYPE	NUMBER CAPTURED
ICMP	31
TCP	35
UDP	48
TOTAL	114

From table 4.3, the total packets captured were 114, out of the total ICMP-31, TCP-35, and UDP-48. Variation in number is as a results of what is been transmitted from the source. This means that, you can only capture packets that are traversing in the network in the application is installed.

Table 4.4 Packet size and time

PACKET SIZE	TIME
60	32
78	33
40	37
44	39
40	58

From table 4.4, the variation in arrival time indicates that packets can have different arrival times based on factors like, hop count, TCP window sze , speed of the link, and the packet size itself.

Table 4.5 Hop Count and Delay

HOP COUNT	DELAY
1	48
59	35
62	44
128	62
252	35

From table 4.5, the variation in hop count values means that the packets traverses the stated number of routers in the table before arriving at their destination. However, this number of routers can affect packets as a result of per hop delays posed by the routers buffers.

From the same table 4.5, the delay values, means that the various packet experiences different delay levels.

Table 4.6 Window Size Time

WINDOW SIZE	TIME
8	32
12	37
64	37
64	42
8	54

From table 4.6 the variation in the figures came as a results of the fact that, TCP window size are not static and can be varied , where larger windows size shows better performance, and enhance throughput. Smaller windows size increase time of delivery and increase fragmentation.

4.2.2. **Packet Parameters**

1**. Captured Date**

From the captured results in the interface above, the date on which the packet was captured, is 20 Julyl, 2012 at 4:26 pm. This indicates date and time an activity was carried out on that particular subnet that packet traverses and therefore if the source is known an inference can be made as whether at that time the user was suppose to work or not. In that case, date and timing of user's activity can be monitored.

2. **Destination and Source IP Address:** From the captured results in the tables, source and destination IP address inform us, where the packet is been generated from as well as where it is going. This means that by knowing the source and destination, you can easily identify the user of that particular environment. From table 4.2 in the output, the source 192.168.1.10 point to address 172.28.19.5.

4. **Packet length:** From both table 4.1 and 4.2 above, the packet length gives us information on packet in byte. By inference, it tells us how long it will stay during transmission and processing if the maximum transfer unit is less than the packet size. The bigger the size, the higher the transmissions and queuing delay based on speed of link and buffer size of routers or nodes. Transmission delay is a function of the packet's length and has nothing to do with the distance between the two nodes. This delay is proportional to the packet's length in bits. Queuing delay is also a function of maximum segment size.

This means that as users download bigger packet sizes there will be more delays, which may affect productivity. From the output in the table 4.2, data in row two has packet length of 56.

5. **Hop count:** From table 4.2, the hop count 62 is the number of routers through which ICMP packet traverses between source 192.168.1.10 and destination 172.28.19 within the network.

The more hops a packet length of 56 traverses to reach their destination, the greater the queuing and transmission delay incurred.

6. **Identification:** This field contains a value that is common to fragments belonging to a particular message; it can be used if the datagram must be fragmented by a router during delivery. This field is used by the destination to reassemble messages without accidentally mixing fragments from different messages. This is needed as results of the fact that fragments may arrive from multiple messages mixed together, since IP datagram's can be received out of order from any device. From the output, the identification number is 12152.

4.3. **Network graphical analysis:** The network graphical analysis was implemented using pie chart for number of packets, time series graph for packet size and time graph as well as congestion, bar for delay –hop count graph.

a. **Packet type graph:** The packet type graph as shown in the graph below gives the number of individual packet captured at a particular time.

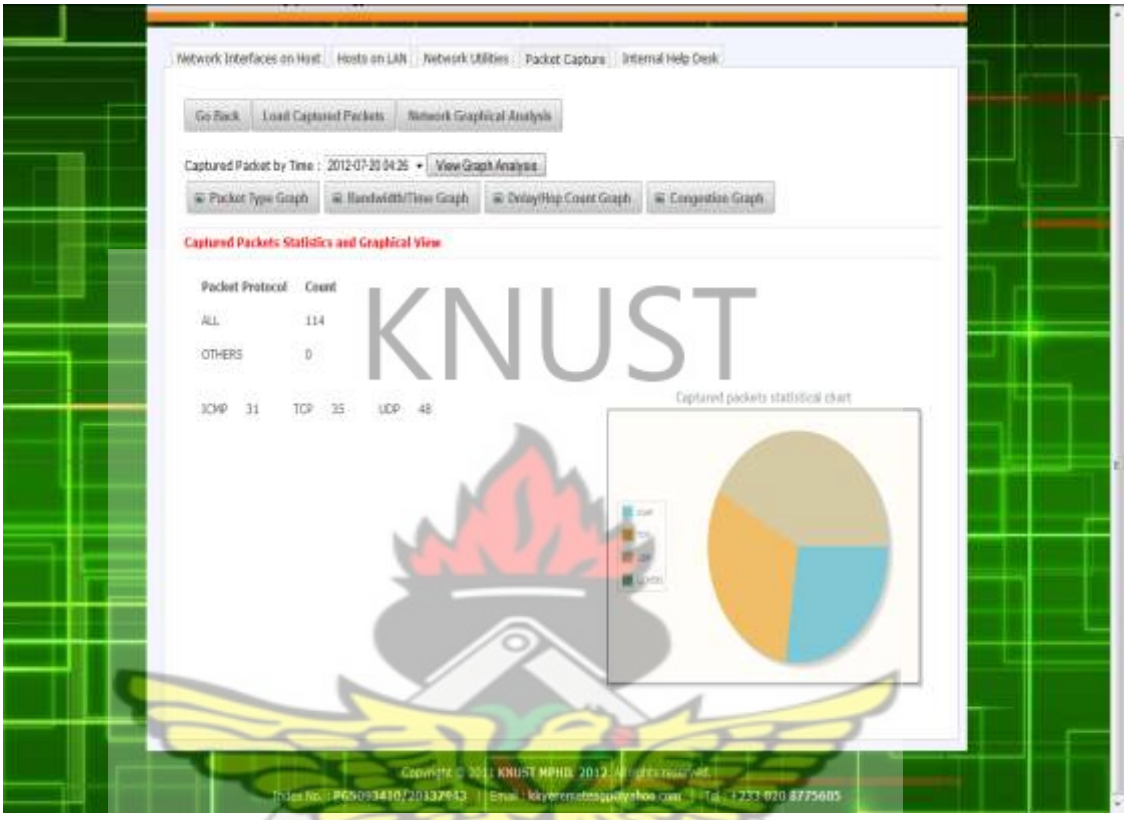


Figure 4.5 Packet captured Statistical Chart

From Figure 4.5, a total of 114 packets were captured at a time and out of the total, specific packet numbers are ICMP: 31, TCP: 35 and UDP: 48. The various colour codes show the different type of packet representation and statistical value. The value will occupy more space.

b. **Packet Time graph:** An arrival time is plotted against packet size. From figure 4.6 the largest packet captured during the implementation in the graph is 78 byte with arrival time of 32 milliseconds and the smallest packet size is 40byte. This means that such packets will consume more time. However, other factors like hop count may also affect it arrival as a results of queuing delay.

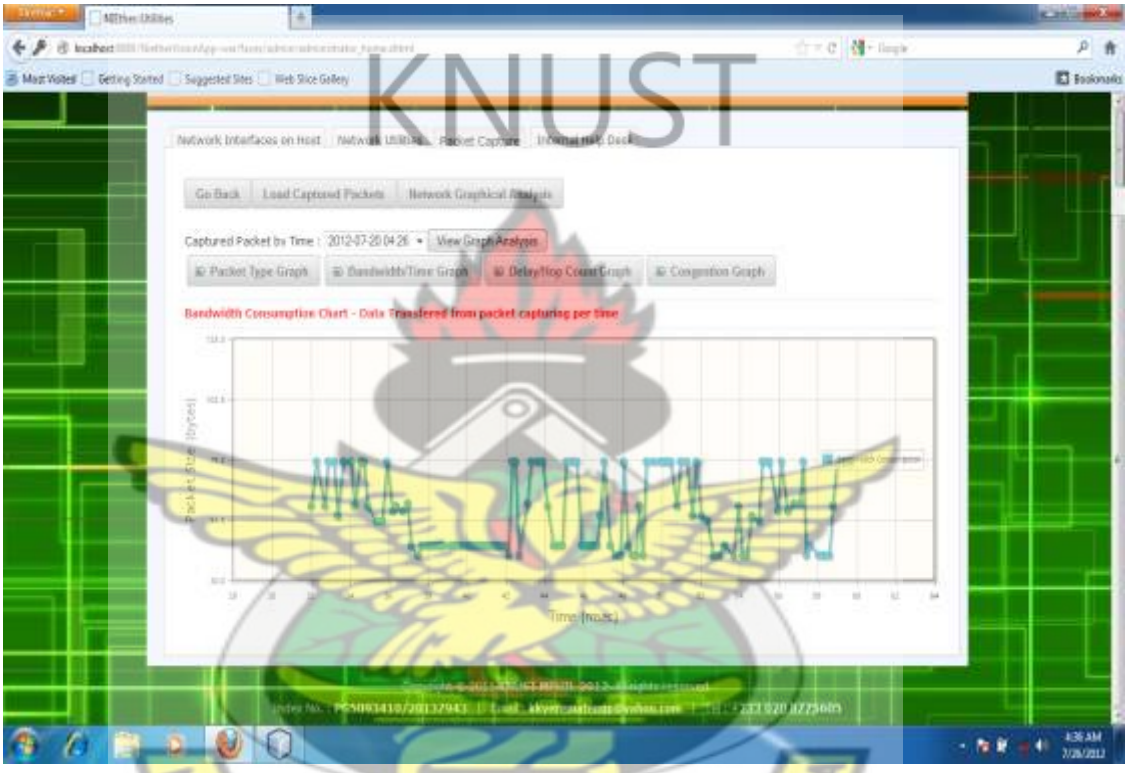


Figure 4.6 Packet captured time graph

From figure 4.6 (screen capture) the trend in the packet time graph is that , packets have different arrival time. Even though, some packets are of similar sizes, yet their arrival times are not the same. This means those with higher time values arrived late and therefore workers who use such time will have to wait and waste organizational time if they are not to use such services that demand those times.

c. **Window size Time graph:** TCP window size in byte is plotted against time in milliseconds, which represents the throughput of the TCP window size. The higher the window sizes the better throughput. However, higher arrival time values size may lead to congestion.

From figure 4.7, the larger window size with less time will offer better TCP performance throughput and less congestion. Considering the graph, window size of 64 kilobyte with a time of 37ms will experience better throughput than window size 64kilobyte with time of 58ms

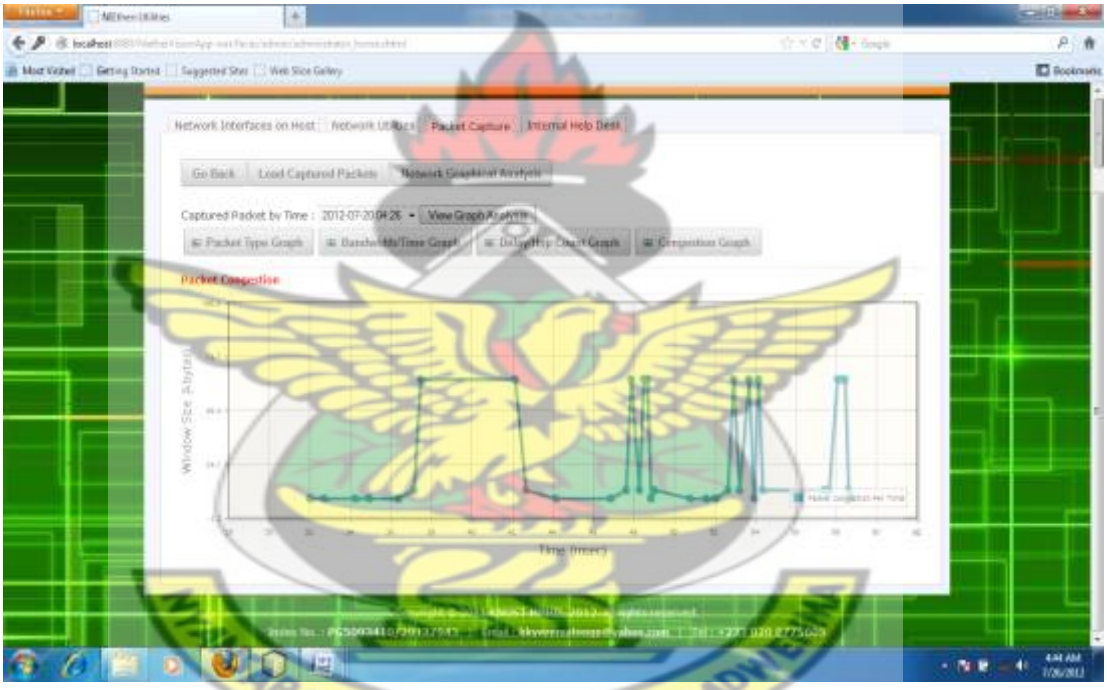


Figure 4.7 Window size time graph

d. **Hop count delay graph:** Hop count is plotted against the delay. From the two parameters, the higher the hop counts the higher the delay. This is as a result of queuing delay within the buffers in the routers.

Considering the graph below, the second highest hop count is 128 with the highest queuing delay of 62ms. The minimum hop count is 63 with delay of 32ms. The highest hop count is 255 with a delay of 32ms. Comparing the 128 and 255, the 128 experiences more delay as results of packet size.

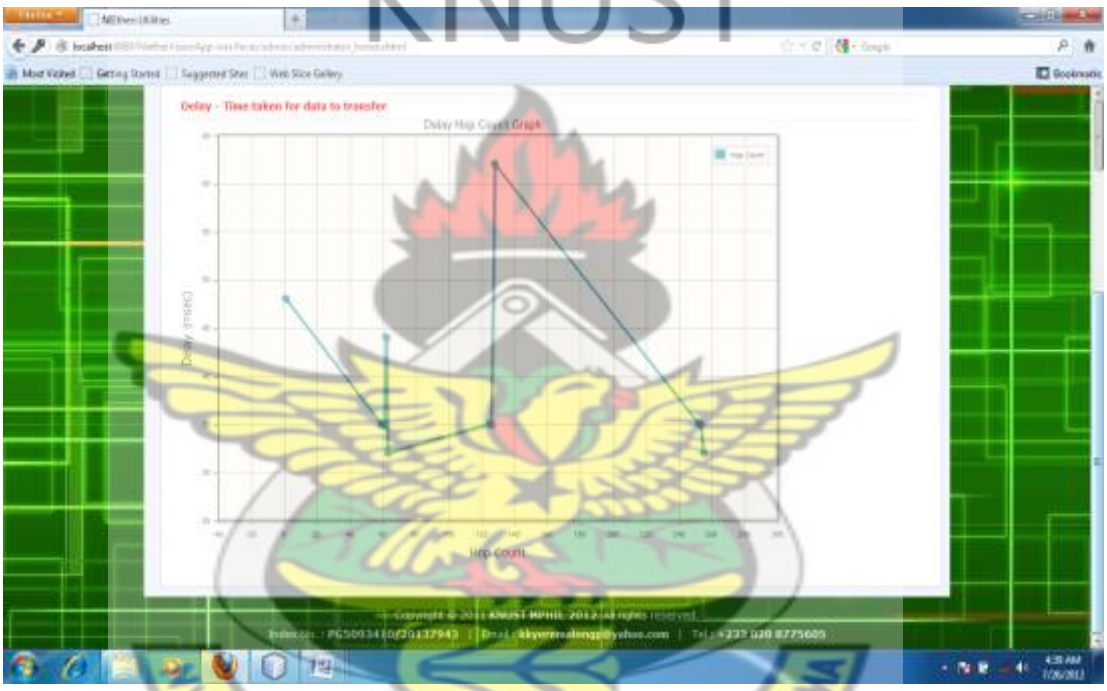


Figure 4.8 Hop count delay graph

From figure 4.8, which shows delay hop count graph, indicates the higher the hop count, the higher the delay per hop. From the graph, it can be infer that the hop count delay varies. These variations may be due to the speed of the various links in which the packets traverses and packet size. From the graph the delay value is not the highest hop count value.

4.3. **Client side**: This side takes care of the internal help desk, general information and network configuration

a. **System Specification and Network Configuration (General Information)**

For installation of other applications, it gives an easy access to the host resource specification, and based on that information, you can easily visualize as whether to install or not. And as security check you can attend to client's request with certainty. Figure 4.9 shows screens of both general personal computer specification and network configuration information, and it was made possible by the use sigar API.

Client Dashboard (General Information)

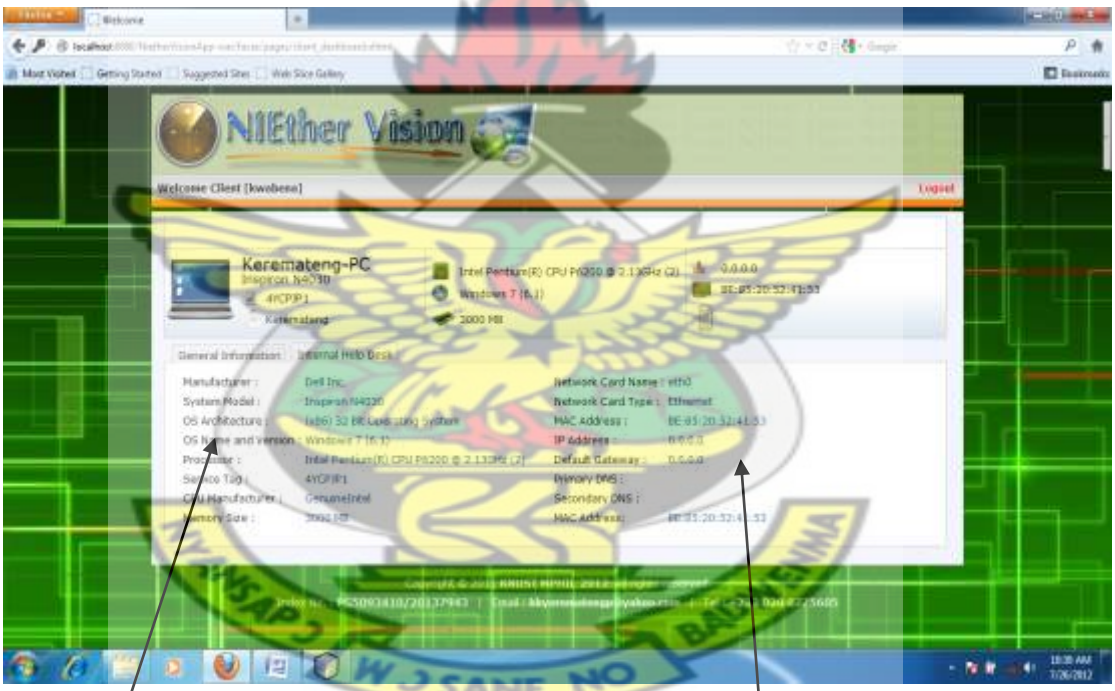


Figure.4.9 System and network information on client

System Specification

Network Configuration on clients

Clients use figure 4.10 pages to request for help and also use it to responds to queries from administrator after sending monitored results from packets captured.

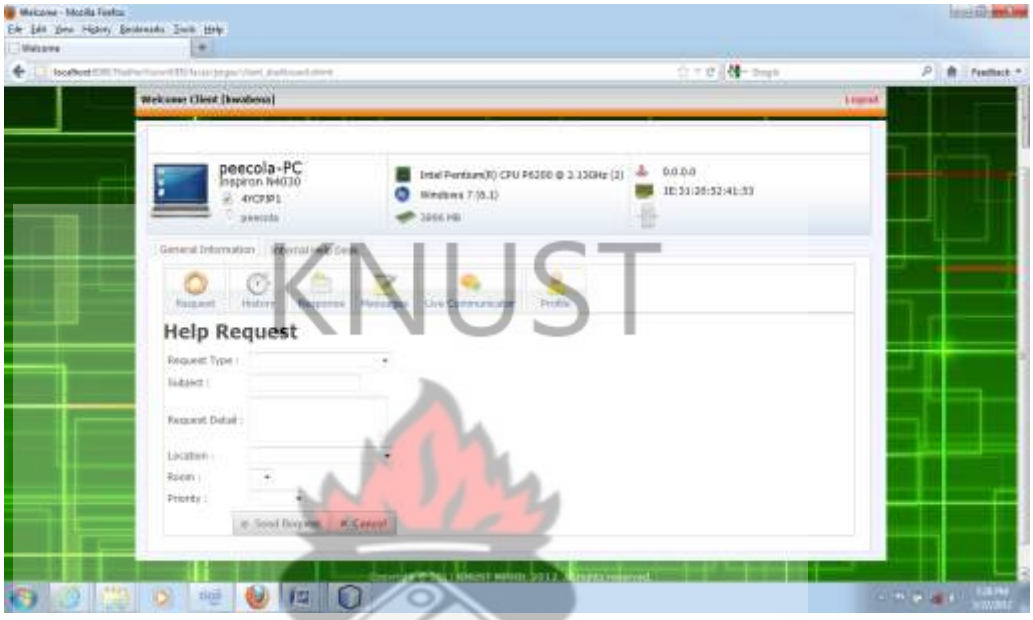


Figure 4.10 Request page

b. **Request History Page**

Al l requests histories are logged for the administrator as shown below.

Figure 4.10 History page

In order to ensure the functionality requirements, all the necessary tests were done in section 4.4.

4.4. **Testing**

During development, several testing procedures have been used in order to test the various subcomponents of the work.

The network card has been tested using standard testing procedures provided by manufacturers (ehow.com, 2012). These procedures include capturing packets from network through network interface card, and the source of this packets generation are obtained by visiting and downloading music/audio/files from different networks or web site on the internet.

The receiving and monitoring part of the system has been tested as shown in the table 4.3. The outputs of the tool were also tested when a single downloaded music was played at the server side of the system.

Also the tool provides timing information about each packet. More specifically every captured packet is time stamped in relation to the first packet with a resolution of a microsecond.

The main testing infrastructure for the complete system is based on the deployment of services to a LAN, WAN with wireless router and LAN setup as well as wired setup. A diagram of the testing infrastructure is given in chapter 3.

The application was able to capture packets from data link layer to the TCP,IP and UDP level, presented the graph in clear and readable form, captured and display all network drivers and their interfaces as well as both general personal computer system and network configuration information.

The internal help desk test was performed when a message was sent from one of the clients to the server in the setup.

In table 4.3, a total of 114 packets were captured as test with two additional final year IT students, and analyzed, it was able to give the various individual packets, thus UDP-48, TCP-35 and ICMP-31 summing up to 144.

Test case

During testing, the application is executed with a set of test cases and the output of the program for the test cases is evaluated to determine if the program is performing as it is expected to perform. These tests are designed to validate functional requirements without regard to working of the program.

Testing Levels

In order to make sure that the system does not have errors, the different levels of testing strategies that are applied at differing phases of software development are.

Unit Testing: Unit testing is done on individual modules as they are completed and become executable. It is confined only to the designer's requirements. Unit Testing focuses on verification of the smallest unit of software design module.

Integrating Testing: Integration testing ensures that software and subsystems work together as a whole. It tests the interface of all the modules to make sure that the modules behave properly when integrated together.

System Testing: System testing involves testing of the entire system before delivery. Its aim is to satisfy the user the system meets all requirements of the specifications.

Table 4.7 Test case

No.	Test case Title	Description	Expected Outcome	Result
1	Test the functionality of a capturing from internet	User saves the packets in database and views it later	packets is saved in database	Passed
2	Test that GUI is able to display captured traffic	User sees the packets in graphical form.	Successful presentation of traffic.	Passed
3	Test the mapping of files to the IP address	User views the attached files to each destination IP address	Correct IP address is mapped to each file	Passed
4	Test the functionality of a graph plotting	User sees graph and its' parameters	Graph is opened From database	Passed
5	Test the decoded information of the selected packet block	User chooses from the list to view packet information	The output is displayed correctly	Passed
6	Test system information	User changes memory size	The output memory size changes	passed
7	Test helpdesk message	User sends message	Responds came from admin	passed
8	Test hard and software interfaces	User clicks the interface to view	Interface opens with status showing	passed

(Source: adopted from Arjun et al, 2008)

4.5. Discussion

The results of the findings and analysis are discussed in relation to the objectives of the research. Specifically, easy access to interface information, users' movement, effects of users' traffic.

Users Movement, help desk and Interface Status

Table 4.1 and table 4.2 shows that, it is easier to visually monitor network users' communication or movement through packets loaded from database.

Figure 4.1 shows that network driver and interface card status can easily be verified before capturing packets instead verifying from registry or using commands

Figure 4.3 above shows that there are pieces of information that can be extracted from packet block captured from the wire.

Figure 4.4 above shows the various information extracted from the block as follows:

- the date of capturing
- sequence of capturing or position in the queue
- source IP address
- destination IP address
- packet length,
- hop count,
- identification
- packet type
- Window size

Figure 4.5 Shows that packet can be grouped on statistical bases in a single capture

Figure 4.6 Shows packet can be captured based on their arrival time and analyzed using graph

Figure 4.7 Shows how packet can easily helps in analyzing factors that contributes to delay based on the hop count.

Table 4.1 and table 4.2 Shows that we can easily predict user's activity or host activity outside its network or on the internet.

Screen capture 4.9 shows how easy it is to get access to system and network information on host.

Screen capture 4.10 shows that in packet analysis environment help desk can be used for exchanging information between users of network and administrator.

Based on the results of the captured packets it can be clearly seen that users of WAN, LAN and Internet in any environment that goes outside their domain or subnet, their activities or movement can visually be monitored.

The findings indicated that users of networks can traverse any domain or website once giving the access and therefore does not care about what they bring into the environment or its effect. However, as far as they can traverse in the network their activities can be traced.

Graphs

Based on the gathered trend and results from the graph, user's activities outside their subnet or domain add additional load to the existing environment. However, depending on how far they traverses hop count, delay, TCP window size and packet size may affect the link in which they work.

CHAPTER 5

CONCLUSION

This study is focussed on monitoring user's activities in network using non-router based approach and packet analysis so that administrators will make meanings of what packets come into their network at the right time and the effects of additional traffic.

Again, the study made it possible for both clients and administrators to communicate on issues relating to both monitored results from packet analysis and also responds to clients needs using internal help desk. However, for the purposes of network card and driver status verification, network card and driver status verifier were included to test such status before packet capturing.

This research also aimed to monitor the movement of hosts or users in a network that can have adverse effects on performance as a results of additional traffic they bring into the environment, whether LAN, WAN or internet.

Based on the results of the analysis graph, how far network users' search for information can affect the performance. In order to maintain undisturbed links in organizations, users must be monitored to find out what they do with their personal computers that is connected to the office network and made account for what they do.

The number of routers users' crosses or traverses also affects performance as a result of transmission delay, in addition, window size and packet size also affects the TCP throughput. Finally, using helpdesk in such an environment helps to let network users know that they cannot hide activities.

5.1. **Research Question**

The research questions to be answered are:

How can packet parameters be used to visually monitor network hosts' or users' movement in the network and on the Internet based for management decision?

In chapter two, there was a review of packet parameters and some of the possible tools as well as some of the router based monitoring systems. Based on the review, the design of the application was done in chapter 3.

In chapter 4, based on the implantation analysis, one can easily visually determine hosts' or users' movement based on their source and destination IP address.

How can the monitored results be communicated to clients and their network problems communicated back to administrators?

At the implementation in chapter 4, results were sent to host using internal help desk as well as client sending information back to the administrator in the network environment.

In addition, access to network configuration and system information were made possible and easy. This also makes it easy to monitor man in the middle issues.

What are the effects of users' activities on bandwidth consumption, delay, and congestion characterized by packet parameters?

During the design phase after the review in chapter 2, bandwidth, delay and TCP measures were included to facilitate the effect of the users' movement.

In chapter 4, hop count was captured and based on the analysis results from the graph its effect were made known.

Thus the results established that the more routers in the communication distance, the higher the hop count which affects per hop transmission delay.

The designed application gave packet captured statistical information to the network administrator that can be used for management decision.

In chapter 4, in the implementation, packets were saved into database which made it possible to get information to do the analysis for monitoring and management decision.

Window size and arrival time were also analyze to see the effects of TCP window on throughput.

How can administrators determine the network interfaces on the server with ease as whether it is up and running or not?

In chapter 4, based on the analysis from the output it was very to have access to all the software and hardware interfaces, which made it possible to determine the interfaces as it is up and running or not.

The monitoring of information being done via different ways till date as checking the remote communication between network hosts is always best for economic purposes.

A tool has been designed, network monitoring tool that aids management in using packet analysis with the goal to ease the work of administrators and other users. The tool is easy to use and also acts as a guide for administrators.

The concept of this work allows administrators and users to access the remote information via the same environment the user will not have to go here and there to access and collect the information but the same is available on his own system. This means there is reduction in time, stress, and increased flexibility, and information availability. More importantly, users activities on the internet or outside their subnet adds additional traffic.

Based on the tables, graphs analysis, administrators must be able to monitor network users and should make them to give accounts of network used.

5.2. **Discussion**

There were some limitations in this study, the packets captured and used was limited to NHIS network, even though it worked in an internet environment as well. Once the test or the experiment was done in a live environment, it means that it can be used in every network environment for management purposes.

5.5 **. Challenges**

There were some challenges faced during the course of the study, they are: Getting access to the required API for the packet capture was a problem as other options were there on the internet, and the wrong ones crashing the application.

However, building applications to capture packets from network involves downloading windows packet capture library (winpcap) and Sigar whose process of downloading was always interrupted.

Furthermore, using JPCAP was difficult because the native language is C and was built using AMD processor, and therefore using Intel processor was difficult.

Besides all the above, building the application in windows7 was a daunting task because some of the APIs' needed to be kept in particular folders in either system 32 or just windows and some directories of glassfish.

Finally, incorporating internal help desk was not easy because this was the first time such thing was been done.

5.4. **Recommendation**

Since the tool is able to monitor the user's movement in the LAN using Hop count and IP address as well as the bandwidth consumption peculiar to a particular user, I recommend that it should be adopted

5.5. **Future Research**

The next step is using packet analysis for threat analysis, where contents of packet be used in identifying hackers.

However, other enhancement like detailed packet sender's side analysis worth included.

And it is hoped that real time analysis will be used as the packets are being captured.

In addition, internal help desk should be automatic sending results to users without administrator's intervention.

As results of the advent of USB modems, it is hoped that packets emanating from such communication medium be monitored. Driver and card status verification should be automatic.

These additions will make things better for the current emerging technologies.

For documentation and installation see attached CD

References

Agyepong, S. (2010). Msearch: A mobile Search Service

Akindeinde, O. (2009). Security Analysis and Data Visualization

Ardila, G. (2008). Design and Implementation of a Distributed System to Evaluate NetNeutrality.

Antoniou, P. (2008). Network Research Lab, High Speed Multimedia and Multiservice Networks.

Arjun, S., Vaibhav, G., Hitansh, V. (2008). Network Intrusion Detection System., (http://dcm.uhcl.edu/c56330211spsomala/nids.pdf) (accessed: 10/10/11)

Asante, M., Sherrat (2010). Journal of Science and Technology KNUST, Volume 30 Number 3, Page 87 to 100

Booch, G. (1994). Object oriented analysis and design.

Bradley,M.,(http://compnetworking.about.com/od/networkadapters/g/bldef_nic.htm), (accessed :12/05/12)

Buck, G. (2001. Tcp/Ip Addressing: Designing and Optimizing Your Ip Addressing Scheme.

CACE Technologies (2008). (http://www.winpcap.org/)

Caliskan, E. (2011). Campus network topology discovery and distributed firewall policy generation.

Cardigliano, A. (2010). Towards wire-speed network monitoring using Virtual Machine

Casad, J. (2004) Sams Teach Yourself TCP/IP in 24 Hours

Cecil, A. (2006), 'Router based and Non-Router Based ... Monitoring Techniques, Internetworking Technologies Handbook, Chpt 55, 1992—2006'

Chuck,J.,Wei,L., (2006).(Ibm.com/redbooks,TCP/IP Tutorial and Technical Overview).

Claire, B. (2004). Request for Comments: 3954, Cisco Systems.

Clemm, A.(2007). Network Management fundamentals, Cisci Press, ISBN: 1-58720-137-2

Clos, M. (2010). A framework for network traffic analysis using GPUs (http://upcommons.upc.edu/pfc/bitstream/2099.1/8800/1/Thesis.pdf)

Cole, R. (2003). (http://www.cs.jhu.edu/~rgcole/publications/rfc3577_2003.txt)

Comer, D. (2006). Principles, protocols, and architecture, Prentice Hall

Coull,E (2007). Traffic analysis

Craig, H. (2002), TCP/IP Network Administration, 3rd Edition, O'Reilly Media.

Degioanni, L. (2000). Development of Architecture for Packet Capture and Network Traffic Analysis.

Ding, J. (2010). Advances in Network Management, Auerbach Publications, ISBN-13/ EAN: 9781420064520.

Domingo-P.(2011).ISBN 978-3-642-20304-6.(http://www.springer.com/computer), (accessed 22/04/12)

Domingo-Pascual, J. (2001). Traffic Monitoring and Analysis.

Dainotti, A., Pescap, A. (2004). Plab: packet capture and analysis architecture, (http://www.grid.unina.it/Traffic/pub/TR-DIS-122004.pdf)

Fall, K. (2011). TCP/IP Illustrated, Volume 1, The protocols Second Edition.

Garcia,M.(2010). (www. tcpdump.org)

Hamzah, N. (2001). Developing a Packet Capturing Program

Hartpence, B. (2011). Packet Guide to Core Network Protocols, O'Reilly Media, Inc.

http://compnetworking.about.com/od/networkprotocols/l/bldef_packet.htm

http://computer.howstuffworks.com/question525.htm, packets

http://en.wikdownlodipedia.org/wiki/Software_engineering (accessed: 25/05/12)

http://en.wikipedia.org/wiki/Help_desk, (accessed: 02/03/12)

http://en.wikipedia.org/wiki/Unified_Modeling_Language (accessed: 20/02/12)

http://en.wikipedia.org/wiki/Waterfall_model (accessed: 23/03/12)

http://packetlife.net/captures/ (accessed: 12/11/11)

http://primefaces.org/gettingStarted.html, (accessed: 26/11/11)

http://wiki.wireshark.org/CaptureSetup/Ethernet, (accessed: 30/06/11)

http://www.agilemodeling.com/artifacts/classDiagram.htm (accessed: 05/12/11)

http://www.agilemodeling.com/essays/umlDiagrams.htm (2009) (accessed: 23/01/12)

http://www.calsoftlabs.com/whitepapers/ethernet-network.html (accessed: 12/05/12)

http://www.cisco.com/en/US/docs/internetworking/troubleshooting/guide/tr1907.html , (accessed: 10/04/12)

http://www.clker.com/clipart-1933.html (accessed: 20/05/12)

http://www.ehow.com/how_5943787_test-nic-card.html,(accessed:20/05/12)

http://www.firewall.cx/networking-topics/protocols/icmp-protocol/156-icmp-time-exceeded.html (accessed: 20/07/11)

http://www.hit.ac.il/staff/leonidm/information-systems/ch63.html(accessed: 03/03/12)

http://www.exa.unicen.edu.ar/catedras/comdat1/material/TP1-Ejercicio5-ingles.pdf

http://www.hyperic.com/products/sigar (accessed 12/03/11)

http://www.inet.tuberlin.de/fileadmin/fg234_teaching/SS12/IM_SS12/im12_09_how_to_analyze_the_app_layer.pdf (accessed: 20/01/12)

http://www.modernanalyst.com/Careers/InterviewQuestions/tabid/128/articleType/ArticleView/articleId/1433/What-is-a-Context-Diagram-and-what-are-the-benefits-of-creating-one.aspx, (accessed: 27/04/2011)

http://www.myipaddressinfo.com/ (accessed: 11/07/11)

http://www.sparxsystems.com/resources/tutorial/use_case_model.htm l(accessed, 24/04/12)

http://www.thinkreliability.com/Root-Cause-Analysis-CM-Basics.aspx (accessed:06/05/11)

http://www.tutorialspoint.com/uml/uml_activity_diagram.htm (accessed: 07/05/12)

http://www.visual-paradigm.com/product/bpva/tutorials/dfd.jsp (accessed: 28/01/12, 2pm)

(http://www.wisegeek.com/what-is-promiscuous-mode.htm), (accessed: 12/05/12)

(http://wand.cs.waikato.ac.nz/old/wand/publications/jamie_420/final/node9.html)

Judge, T. (2005), Passive Distributed Network Analysis Using Remote Packet Capture in Java

Kevin, J,Schmidt, E. (2005). SNMP, Help for System and Network Administrators, 2nd Edition, O'Reilly Media, Inc.

Kozierok,M.(2005).(http://www.tcpipguide.com/free/t_TCPIPTransportLayerProtocolsTransmissionControlPro.htm) (accessed: 12/05/12)

Kristoff, J. (2009). http://condor.depaul.edu/jkristof/, (accessed: 12/03/ 2012)

Lucas, M. (2010). Network Flow Analysis, No Starch publishers,ISBN 1593272030, 9781593272036.

Mashitah, G. (2003). Network Traffic Monitoring Analysis on Quality of Service

Miller, P. (2010). TCP/IP - The Ultimate Protocol Guide: Complete 2 Volume.

Mohammed, S. (2006). Developing a Web-Based Packet Monitoring Tool.

Nordby, G. (2012).(http://www.webopedia.com/TERM/S/software_engineer.html), (accessed 27/04/12)

Odom,W., Thomas, A. (2006). Networking basics: CCNA 1 companion

Pahl, J. (2011). Network Optimization.

Papadopouli, M. (2009). Traffic Monitoring and Analysis First International Workshop, TMA

Perkins, D. (2012). RMON: Remote Monitoring of SNMP-Managed LANs, 2007, ISBN-13: 9780130961631, Publisher: Pearson Education, (http://www.barnesandnoble.com/w/rmon-david-t-perkins/1003086523), (accessed: 23/04/12)

Rafiq, C. (2005). Developing TCP/IP and UDP Traffic Monitoring Tool

Rashmi,S.(2010).(http://seminarprojects.com/Thread-rmon-remote-monitoring-presentation), (accessed 23/04/12)

Robert, C. Martin, (1997). http://www.rational.com

Rooney,T. (2011). John Wiley & Sons, IP Address Management Principles and Practice,

Sanders,C. (2007). Practical Packet Analysis, ISBN 1593271492, 9781593271497, Publisher No Starch Press

Satya, Srikanth,P. (2004). Gigabit PickPacket: A network Monitoring Tool For Gigabit Networks.

Schonwalder,J.(2005). Networks and Protocols,School of Engineering and Science,International University Bremenll

Shintaro,M.,Akihiko,S.,Takuya,S.,Hideyuki,H.,Noriichi,K (2009). Versatile Network Stream Capture Tool Using Java For High Energy Accelerator Control Systems (http://accelconf.web.cern.ch/accelconf/icalepcs2009/papers/tup030.pdf)(accessed: 10/02/12)

Tamara, D. (2009). Network+ Guide to Networks.

Tejparkash,(2010).(http://tejparkash.wordpress.com/2010/12/05/paws-tcp-sequence-number-wrapping-explained/), (accessed 11/05/12)

Todd, L. (2007). John Wiley & Sons,:CCNA: Cisco Certified Network Associate.

Yang, C. (2010). Department of Electrical Engineering, National Cheng Kung University,

(http://140.116.177.177/courses/NM_97/doc_Network%20Monitoring.pdf).