

PACKET SNIFFING SYSTEM

**CHAN HAI CHAI
WEK 020026**

**Under the Supervision of
MR. ANG TAN FONG**

**Moderator
MR. LIEW CHEE SUN**

Perpustakaan SKTM

**This project is submitted to the
Faculty of Computer Science and Information Technology
University of Malaya**

**In partial fulfilment of the requirement for the degree of
Bachelor of Computer Science**

Session 2004/2005

Abstract

Network security tool such as packet sniffing is a standalone system that use to monitoring and capture network traffic of computers in a LAN environment. It is capable of filtering out a certain port, IP address and protocol and generates a network traffic graph. It is a useful tool for network administrator and network engineer to monitoring the network traffic.

The aim of this project to build this security tool to monitoring a network, trace and identifies misuse of computer, improve network manageability and overcome some network problem. This system can show the summary report after the packets is captured from all the users in the network, that can make network administrator and network engineer easy to identify which users have use more bandwidth in the network. This system is target to be implementing in a small organization LAN such as Faculty of Computer Science and Information Technology in University Malaya.

The Waterfall with prototyping model is chosen as the methodology to develop this system. This system is a standalone solution that needs special development tools and languages such as C#, Microsoft Access, Winpcap and some relevant tools. This system is developed using Object-Oriented Analysis and Design which provides a particularly fine basis for internal or structural design. The target users for this system are network administrator and network engineer.

This system hope can achieve the objective of providing a user friendly packet sniffing system that easy to use and effective in assisting network administrator to perform their works.

Acknowledgement

The development of this network security tool that is packet sniffing was carried along with advices, assistance, contributions and ideas from many individual.

First and foremost, I would like to express my utmost gratitude to my supervisor, Mr. Ang Tan Fong for the guidance throughout the development of my project. Not forgetting, my special thanks to Mr. Liew Chee Sun, my project moderator for spending precious time to moderate this project

Special thanks also go to my fellow course mates, friends and senior for sharing their time and knowledge with me. Their supports and motivations are deeply appreciated. I feel grateful to have their encouragements whenever I encounter difficulty.

Last but not least, I would like to express my heartiest appreciation to my beloved parents and siblings for all their love, moral supports and encouragement throughout this period. Their kindness will always get a special place in the bottom of my heart.

Table of Content

ABSTRACT	i
ACKNOWLEDGEMENT	ii
CONTENTS	iii
LIST OF FIGURES	vi
LIST OF TABLES	viii
1 INTRODUCTION	1
1.1 PROJECT OVERVIEW	2
1.2 STATEMENT OF PROBLEMS	3
1.3 PROJECT OBJECTIVE	4
1.4 PROJECT SCOPE	5
1.4.1 ENVIRONMENT	5
1.4.2 TARGET USER	6
1.4.3 USAGE TIME LIMITATION	6
1.5 PROJECT EXPECTATION	6
1.6 PROJECT SCHEDULE	7
1.7 REPORT LAYOUT	8
1.8 CHAPTER SUMMARY	9
2 LITERATURE REVIEW	10
2.1 NETWORK SECURITY THREAT	10
2.1.1 DENIAL OF SERVICE (DoS)	10
2.1.2 EAVESDROPPING	11
2.1.3 TROJAN HORSE	12
2.1.4 IP SPOOFING	12
2.1.5 PASSWORD ATTACKS	13
2.1.6 PORT SCANNING	14
2.1.7 PACKET SNIFFING	15
2.1.7.1 FIVE BASIC COMPONENTS	16
2.1.7.2 SHARED ETHERNET	18
2.1.7.3 SWITCHED ETHERNET	19
2.1.7.4 METHODS OF PACKET SNIFFING	19
2.1.7.5 METHODS OF DETECTING SNIFFER	23
2.1.7.6 METHODS OF DEFENDING SNIFFER	23
2.2 TCP / IP	25
2.2.1 SOCKET	29
2.2.2 PORT	31
2.3 ANALYSIS STUDIES	32
2.3.1 CASE STUDY 1 – DISTINCT NETWORK MONITOR	32
2.3.2 CASE STUDY 2 – ANOLOGX PACKET MON	38
2.1.3 CASE STUDY 3 – THE ETHEREAL NETWORK ANALYZER	40
2.4 LANGUAGE	43
2.4.1 VISUAL BASIC	43
2.4.2 C / C++	43
2.4.3 C#	44
2.4.4 JAVA	45

2.5	OPERATING SYSTEM	47
2.5.1	UNIX	47
2.5.2	LINUX	48
2.5.3	MACINTOSH	49
2.5.4	WINDOWS XP	50
2.6	AUTHORING TOOLS	51
2.6.1	MICROSOFT VISUAL STUDIO .NET 2003	51
2.6.2	MICROSOFT VISUAL BASIC 6.0	53
2.7	UNICAST, BROADCAST, AND MULTICAST	53
2.7.1	UNICAST	53
2.7.2	BROADCAST	54
2.7.3	MULTICAST	54
2.8	DATABASE SERVER	56
2.8.1	MICROSOFT SQL SERVER 2000	57
2.8.2	ORACLE	59
2.8.3	MICROSOFT ACCESS 2003	60
2.8.4	MYSQL	62
2.9	SUMMARY CHAPTER	63
3	SYSTEM REQUIREMENTS ANALYSIS	64
3.1	METHODOLOGY	64
3.1.1	CONCLUSION ON DEVELOPMENT METHODOLOGY	65
3.1.2	WATERFALL WITH PHOTOTYPING	65
3.1.3	JUSTIFICATION OF METHODOLOGY	71
3.2	INFORMATION GATHERING METHODS	72
3.3	CHAPTER SUMMARY	73
4	SYSTEM ANALYSIS	74
4.1	SYSTEM ANALYSIS REQUIREMENTS	74
4.1.1	FUNCTIONAL REQUIREMENT	74
4.1.1.1	CAPTURING THE PACKETS MODULE	75
4.1.1.2	FILTERING DATA MODULE	75
4.1.1.3	ANALYZE THE PACKETS MODULE	75
4.1.1.4	GENERATE STATISTIC GRAPH MODULE	76
4.1.1.5	PRESENT THE RESULT MODULE	76
4.1.2	NON-FUNCTIONAL REQUIREMENTS	76
4.2	CONCLUSION ON TOOLS AND TECHNOLOGY	78
4.3	HARDWARE AND SOFTWARE REQUIREMENTS	80
4.4	CHAPTER SUMMARY	81
5	SYSTEM DESIGN	82
5.1	STRUCTURE CHART	83
5.2	FLOW CHART	86
5.3	DATA FLOW DIAGRAM (DFD)	89
5.4	USER INTERFACE DESIGN	92
5.5	CHAPTER SUMMARY	95

6	SYSTEM IMPLEMENTATION	96
6.1	DEVELOPMENT ENVIRONMENT	96
6.1.1	HARDWARE CONFIGURATION	96
6.1.2	SOFTWARE CONFIGURATION	97
6.2	PLATFORM DEVELOPMENT	97
6.2.1	SETTING UP OPERATION SYSTEM	97
6.3	PROGRAM IMPLEMENTATION	98
6.3.1	IMPLEMENTATION OF CAPTURING PACKET MODULE	98
6.3.2	IMPLEMENTATION OF FILTERING PACKET MODULE	98
6.3.3	IMPLEMENTATION OF ANALYZING PACKET MODULE	99
6.3.4	IMPLEMENTATION OF PRESENT RESULT MODULE	100
6.4	SYSTEM DEBUGGING	101
6.5	CHAPTER SUMMARY	102
7	SYSTEM TESTING	103
7.1	TESTING STRATEGIES	105
7.1.1	UNIT TESTING	105
7.1.2	INTEGRATION TESTING	106
7.1.3	SYSTEM TESTING	107
7.2	TEST CASES	108
7.2.1	UNIT TEST CASE	108
7.3	CHAPTER SUMMARY	110
8	SYSTEM EVALUATION	111
8.1	INTRODUCTION	111
8.2	SYSTEM STRENGTHS	111
8.3	SYSTEM CONSTRAINTS AND LIMITATIONS	113
8.4	FUTURE ENHANCEMENTS	114
8.5	CHAPTER SUMMARY	115
	REFERENCE	116
	APPENDIX A : USER MANUAL	117

List of Figures

Figure 1-1: Project Schedule	7
Figure 2-1: Packets sent from PC to hub	18
Figure 2-2: Hub forwards the traffic to all PC	19
Figure 2-3: ARP Sniffing Method	21
Figure 2-4: TCP / IP Model	26
Figure 2-5: TCP / IP Model	28
Figure 2-6: Client make connection request	30
Figure 2-7: Server accept connection request	30
Figure 2-8: Distinct Network Monitor	32
Figure 2-9: Configuration	33
Figure 2-10: Network Protocols Distribution	34
Figure 2-11: IP Protocols Distribution	35
Figure 2-12: IP Traffic Distribution	35
Figure 2-13: Subnet Traffic Distribution	36
Figure 2-14: Packet Size Distribution	37
Figure 2-15: Summary Distribution	37
Figure 2-16: AnalogX PacketMon	38
Figure 2-17: Detailed information for packets	39
Figure 2-18: Include Raw Header function	40
Figure 2-19: The Ethereal Network Analyzer	40
Figure 2-20: IP Multicasting Works	56
Figure 3-1: System Development Process Model	64
Figure 3-2: Waterfall Model	66
Figure 3-3: Prototype Model	67
Figure 3-4: Waterfall Model with Prototyping	70
Figure 5-1: Structure Chart for Packet Sniffing	83
Figure 5-2: Structure Chart for Capturing Packets Module	84
Figure 5-3: Structure Chart for Filtering Packets Module	84
Figure 5-4: Structure Chart for Analyzing Packets Module	85
Figure 5-5: Structure Chart for Generate Statistic Module	85
Figure 5-6: Structure Chart for Present Result Module	86
Figure 5-7: Flow Chart of Packet Sniffing System	88
Figure 5-8: Context Diagram of Packet Sniffing System	90
Figure 5-9: Diagram 0 of Packet Sniffing	91

Figure 5-10: Main interface of Packet Sniffing	92
Figure 5-11: Configuration Filter Menu	93
Figure 5-12: Interface of Protocol Statistic	94
Figure 6-1: Function ToUint	99
Figure 6-2: Members in the Struct	100
Figure 6-3: Function CalculateHost	101
Figure 7-1: Testing Process	103
Figure 7-2: Top-Down Testing	107
Figure 7-3: Unit Test Case	109

University of Malaya

List of Tables

Table 2-1:	Reserved Ports Assigned Lists	32
Table 2-2:	Comparison Among Three Languages	46
Table 2-3:	Operating System Statistic	47
Table 4-1:	Tools and Technology Chosen	80
Table 4-2:	Hardware and Software Requirements	81
Table 5-1:	Flow Chart Symbol	87
Table 5-2:	Data Flow Diagram (DFD) Objects	89
Table 6-1:	Software Used	97

Chapter 1

Introduction

University of Malaya

Chapter 1 - Introduction

Nowadays, many systems are connected to the internet because internet is a huge network and has no boundaries some more business opportunities also increasing dramatically. Information on the internet can be accessed from anywhere in the world in real time, however is also allowed for the increasing of network security threats. Hacker tools now very easily to get it through the internet. Some web site even provides the guideline on how to hack into a system, giving details of the vulnerabilities of the different kinds of systems. So, anyone can download the tools from internet and use it to break into a system which is not properly secured. The various kind of network security threats are packet sniffing, Denial of Service (Dos) attacks, IP spoofing and password cracker.

Thus, develop network security tools are very important to protect our system from attacking by hackers through the internet or LAN. It is because network security tools are the program of preventing and detecting use of your computer. Prevention measures help you to stop unauthorized users from accessing any part of your computer system. Detection help you to determine whether or not someone attempted to break into your system, if got some people success access to your system and will monitor what they may have done.

Many network security tools are provide to users for protecting their system such as port – scanning, Intrusion Detection System (IDS), firewall, packet analyzer, packet sniffing for monitoring network and some more. Different tools have their different

function to prevent hacker attacking on your system. So, we must analysis and determine which sections should be protected. After that, the network security tools will be created based on that analysis.

1.1 Project Overview

Ethernet technology was built on the concept of sharing computer resources, where all the computers on the local network are connected by the same switches. Packet sniffing is a basic network security tool that is used to monitoring the network traffic. The tools will monitor all incoming and outgoing packets from users on the network and captures packets which detect as the danger packets to the network. Beside that, this tool also can summary the report about how many users on the network and how many packets are sent out and receive to the users.

The purpose of built packet sniffing is to assist network engineer and network administrator in performing their daily tasks that monitoring and analyze the network traffic. This tool is specially design to be use under real time network environment where it is able to capture the current network packets and perform the summary like statistic of IP traffic or protocols in the graphical form. Network engineer also can analyze content of packets which captured from the network and displays in a human readable format with this system. This is useful for a network engineer to maintain and discover any problem will occur in the network.

The abilities of this security tool to show the protocol, port and IP address upon the summary report will help the network engineer to discover and maintain the network from any possible threat.

1.2 Statement of Problems

Network is become more important in any organization to do their business operations than ever before. It is because of network can help users transfer files and do the business transaction via Internet easily. If the network is can't work properly then business can be impacted significantly, resulting in loss of revenue as well as end user productivity. Therefore there was a need for network professionals to cover every aspect of network management and network security. The following problems are faced by network administrators and network engineers without packet sniffing system:

➤ Can't monitor the network traffic

Network Administrator can't monitor the network traffic in the LAN environment. They don't know the activities are done by users in the network such as chatting, downloading and entertainments. They also don't know the usage of bandwidth by each user in the network. So, when problems are occurred in the network, network administrator can't easily trace and identify the problem.

➤ Security problem

Network Administrator also doesn't know unauthorized users access the network to gain the useful information. This is due to they no have tool to protect and sniff the packets which are sent out and receive from users

without the packet sniffing system. So, some packets are sniffed by hackers for their use without any acknowledgement to network administrator.

1.3 Project Objective

This system is developed under purpose of helping network engineer and network administrator. Objective to develop this network security tool are:

- ❖ To monitor and analyze network traffic in LAN environment is done easily without any extra work.
 - One of main targets of this tool is ability of monitoring activities being done by the users in the LAN. When a packet is sent from a workstation to another workstation through the network by applying the 'listen' technique, the packet can be capture and the data being transfer can be copy down. By capturing the packet, network engineer can know the activities done by user.
- ❖ To easily trace and identify the problem of network.
 - With the continuing monitoring the network activities, network engineer can trace and identify the problem of the network as soon as they start it.
- ❖ To study the trend and usage of network traffic
 - The graphical report shows that the current network traffic is useful for network engineer to study the trend of the network traffic and maintain the network. So that can always capable to support user's command.

- ❖ By analyzing the existing and pass monitoring report can help network engineer to discover the problem of network.
- ❖ To be more expert in the network environment and beware of the safety of the network.

1.4 Project Scope

- ❖ Aimed to provide study and report on the user's network traffic and know how the packets are captured in the network.
- ❖ Can be study about analysis content of the packets which captured from the network.
- ❖ Suitable for use of network engineer and network administrator of small organization like Faculty of Computer Science and Information Technology in University Malaya in perform their daily duties.

1.4.1 Environment

To monitor user's network activities and generate a graphical report that summaries the monitoring result after the packets are captured by this tool. This is security tool is targeted to use in the LAN environment. LAN is normally belonging to an organization and all the hardware is arrange in a single location such as office, building or campus. LAN structure enables resources such as hardware, software or data can be share in the network.

1.4.2 Target User

This tool is help network engineer and network administrator perform their task in manage the network. These tasks such as stabilize the network traffic, make sure the network traffic always smooth and available, monitor user's network activities, make sure the network always free from virus attack and hacking by illegal user.

1.4.3 Usage Time Limitation

Normally users are not limited to use the network. That is mean user can use the network anytime. This is also brought the meaning that this tool has run in 24 hours and 7 days non-stop. This tool must able to work when the network is available and working. Practically, the network administrator must determine when the time to run this tool. To determine this, we have to know what kind of organization is holding this LAN system. For example, if this tool is applied for an office, network administrator can do the analysis about the network traffic that has been recorded after 1 week this tool is implementing. He can do this at non-office hour to cause the minimum affect.

1.5 Project Expectation

This tool is expected to fulfill the requirement of networking system now and development of the information technology nowadays.

This tool is expected to help the network engineer in performing their tasks especially in troubleshooting and find the cause of the problem such as identify the unstable situation of network traffic, capture the packet that is come in and out and finally fulfill user's request as long as it don't exceed the rules that have been set.

Meanwhile, this tool also hope can show the correct statistic about the network activities such as transferring in and out of data, internet connectivity. This is to make sure the statistic collected is not a random amount which can't show the .real situation of network. The up-to-date are useful to plan, organize and maintain the system. The graphical report can easily show the trend of the network traffic to network engineer and to prevent congestion and enhance performances.

1.6 Project Schedule

A project schedule describes the software development cycle for a particular project by enumerating the phase of a project and breaking each of the phases into discrete tasks that need to be carried out. It is essential as it acted as a time management and control to the developer to determine what tasks to be carried out and what goals should be achieved when a certain milestone is met. The project schedule is as shown in **Error!**

Reference source not found.

ID	Task Name	Start	Finish	Duration	2004						2005	
					Jul	Aug	Sep	Oct	Nov	Dec	Jan	Feb
1	System Study	7/2/2004	8/10/2004	28d								
2	Literature Review	7/22/2004	8/20/2004	22d								
3	System Requirement Analysis	8/16/2004	9/20/2004	26d								
4	VIVA	8/23/2004	9/10/2004	15d								
5	System Design	8/19/2004	9/23/2004	26d								
6	System Module Coding	8/23/2004	12/23/2004	89d								
7	Evaluation and Testing	8/25/2004	1/25/2005	110d								
8	Final Evaluation	1/20/2005	2/18/2005	22d								
9	Supervisor Consultation	7/2/2004	2/24/2005	170d								
10	Documentation	7/9/2004	2/24/2005	165d								

Figure 1-1: Project schedule

1.7 Report Layout

The purpose of the report layout is to give an overview of all the phases involved during development of the project. This report consists of eight chapters. Below is the report layout:

Chapter One: Introduction

This chapter is an introduction of the project overview, statement of problems, project objectives, project scopes, expected outcome and project schedule.

Chapter Two: Literature Review

This chapter carries out the research before the project can be implemented. It consists of domain studies, security threats and technology review where research and analysis on the currently available system and techniques used are carried out.

Chapter Three: Methodology

This chapter emphasizes on the conclusion on development methodology and the justification of the chosen methodology. It also discusses the information gathering methods and the explanation about the development tools.

Chapter Four: System Analysis

This chapter describes the system analysis that contains the requirements needed such as functional requirement, non-functional requirement, and hardware and software requirements, and technologies chosen to develop this system.

Chapter Five: System Design

This chapter explains the conceptual and technical design of the system which covers the system architecture design, system functionality design, and interface design.

Chapter Six: System Implementation

This chapter explains the implementation of the system. It discusses on the system development that convert the modules and algorithm that have been designed into programming language that can be implemented

Chapter Seven: System Testing

This chapter presents various type of system testing to find system error and fault. This is also important to make sure that the system fulfills the requirements and specifications that have been planned.

Chapter Eight: System Evaluation

This chapter presents the system evaluation that reveals the problem encountered and solutions, system strength and limitation, future enhancements and others.

1.8 Chapter Summary

This chapter focuses on the introduction of the proposed project, packet sniffing system. Overview of packet sniffing is explained at the beginning of this chapter. This chapter also covers statement of problem, project objective, project scope, expected outcome, project schedule and report layout. The duration for this project that includes research and development will take about 8 months.

The next chapter gives brief explanation on topics researched and studies that are relevant to this project. It is the combination between literature search and literature review about tools and technologies.

Chapter 2

Literature Review

Chapter 2 – Literature Review

2.1 Network Security Threat

2.1.1 Denial of Service (DoS)

Denial of service (DoS) attacks where one user can render the system unusable for legitimate users by "hogging" a resource or destroying resources using multi-user, multi-tasking operating systems. DOS attacks are denied you access to your own system or network and loss of service is related to a specific network service, such as e-mail or DNS. The most popular DOS attack is the flood attack which much of properly formatted packets are directed sent to your systems. Typically, the malicious traffic consists of ICMP control packets sent through the TCP or UDP protocol. This attack is become popular because of the easily, effectiveness, and anonymity associated with some activity. The hackers use pushbutton scripts which available download from the Internet to run the flood attack. The effectiveness of these attacks can be amplified through the use of D-DOS attack methods, via hacked computers or improperly configured networks. Some examples of available tools for carrying out such D-DOS attacks are smurf, fraggle, and SYNflood.

Nowadays, hackers have identified weaknesses in the implementation of many TCP/IP implementations that cause systems to crash when subjected to very specific packet formats. There are even tools that many weaknesses package into a single attack script, so that at the push of a button an attacker can launch a stream of packets that are known to disrupt common TCP/IP implementations. Fortunately, when these weaknesses come to light, the product developers rush to make a patch available to fix the vulnerability.

With this reason, it is so important to keep all of the equipment connected to your network patched with the latest software or firmware updates. It is important to note that we cannot totally eliminate the affects of a DOS attack. DOS attacks are sequence used to make the system lag to response to IP-address spoofing during an attempt to gain access to a related system. As a result, a DOS attack on one part of a network may signal an intrusion attempt on another part. DOS activity should never be ignored by the system administrator. it should be logged and tracked to its source.

2.1.2 Eavesdropping

Cracker use eavesdropping to make a complete transcript of network activity to gain some sensitive information, such as passwords, data, and procedures for performing functions. It is possible for cracker to eavesdrop by wiretapping, eavesdropping by radio and eavesdropping via helping ports on terminals. It is also possible to eavesdrop using software that monitors packets sent over the network. It is difficult to detect that a cracker is eavesdropping.

Many network programs, such as *telnet* and *ftp* are vulnerable to eavesdroppers gaining passwords which are often sent across the network unencrypted. Network programs which involve file transfer are easily to eavesdroppers gaining the contents of files. Encryption can be used to prevent eavesdroppers from obtaining data traveling over unsecured networks.

2.1.3 Trojan Horse

Trojan is a small program which runs on a workstation. Many Trojan tools can download from thousands of places around the globe like internet, computer bulletin boards, "underground" computer society's hacker BBS sites, computer user groups, mail order and friends. Trojan is using for record every key that u press, including your username and password when you access any computer network. Hackers use this recording of your username and password will be retrieved for illegal purposes.

Trojan can easy access into your network by many ways. A common method is physically load onto the workstation by insert the diskette which containing the Trojan. Besides that, Trojan can be installed by email where Trojan is sent to that workstation and debug the scripts. Trojan also can directly install it into the PC when that PC using the internet, FTP, and telnet. A Trojan can also be copied from one workstation to another using these or a variety of other methods.

You can take some methods to protect yourself against Trojan threats. Security policy prohibiting the installation or execution of unauthorized software can help prevent Trojan attack. More stringent policy which prohibits the introduction or removal of any diskettes or electronic data media better prevents these methods of installation. Besides that, using the security tools to scan all the e-mail to prevent the Trojan access your PC system which sent by e-mail.

2.1.4 IP Spoofing

IP spoofing is a one of method of attacking network to gain unauthorized access.

IP spoofing attacks based on internet communication between distant computers is handled by routers which find the best route by finding the destination address, but generally ignore the origination address.

In spoofing attack, the hacker sends the message to a computer indicating the message is from a trusted system. First, the hacker must determine the IP address of a trusted system, then modify the packet headers to make that packets are coming from trusted system. The hacker is fooling the distant computer to believe that they are authorized user on the network. Then the hacker has established a connection that will allow them to gain access into the target system.

The solution to preventing IP spoofing is encrypting all network traffic to avoid source and destinations from being compromised. We also can implement cryptographic authentication system wide. We also can using router to configure network to reject the packets from internet that claim to originate from a local address.

2.1.5 Password Attacks

The use of password is a major point of vulnerability in computer security because password is most common method of authenticating user. Password attacks is the hacker gain the user id and password for accessing user system. The most common password attacks are guessing, brute force, and cracking.

Password guessing is the hacker entering common password either manually or using programmed scripts. Password guessing is usually not so powerful to gain the password

because it is laborious process. Besides that, password guessing need to take more to guess the password and may be detected by user if user got powerful computer security.

Brute – force is hacker attacks password with follow the basic logic as password guessing but faster and more powerful. This tool is automatic to quickly find a working password and username before processing is detected by user. Brute – force attacks are more efficient than password guessing even both techniques are nearly same.

Password cracking is more effective method for discovering password to unlock a resource that has been secured with a password. These attacks need significant level of access to launch, so the best defense is restricting and monitoring access privileges. Password cracking tools such as Lp0phtcrack, LC3 and John The Ripper have easily to lunch the process of cracking passwords unless users are very careful to use hard to crack passwords using special characters such as "@&#.

2.1.6 Port Scanning

Port scanning is similar to a thief going through your neighborhood and checking every door and window on each house to see which ones are open and which ones are locked. Port-scanning is a useful tool for the hacker to use in enumerating access points to a network by identifying open ports. Port scanning software is very easy to use, simply sends out a request to connect to the target computer on each port sequentially and makes a note of which ports responded or seem open to more in-depth probing. There are many powerful scanners available that have function for anti-intrusion detection, the

best being NMAP. Most of port scanning software is running in transparency mode, can enumerate and map whole networks prior to attack.

Some port scanning software also consists of using vulnerability scanners. Many of the scanners used by companies to monitor weaknesses in networks are also used by crackers and hackers to access of those networks.

There are a variety of different scanners available and they all have different features or exploit functions. The most common types are:

- Port Scanners - these just scan for the existence of ports
- General Vulnerability Scanners - these will scan whole networks looking for devices and will test for general networking vulnerabilities.
- Application Specific Vulnerability Scanners - these are usually used after a network has been mapped out and specific applications detected. They related to HTTP, FTP, SMTP, POP3, DNS, etc.

2.1.7 Packet Sniffing

Packet sniffing is a technique of monitoring every packet that crosses the network and analyzing network traffic, detecting bottleneck and problem. A packet sniffing is a tool that running in computer network and monitor all network traffic. This is unlike standard network hosts that only receive traffic sent specifically to them. The security threat presented by sniffer is their ability to capture all incoming and outgoing packet, including clear-text passwords and usernames or other sensitive material. Actually,

sniffer is impossible to detect because they are passive in nature that mean they are only collect data. However, some sniffer is not fully passive, so they can be detected by security tools. Packet sniffing can be run on both switched and non-switched networks.

Packet sniffing has found two form to use. First is commercial packet sniffing is used to help maintain networks and monitoring networks. While underground packet sniffing is used by attackers to gain unauthorized access to remote hosts such as searching for clear-text usernames and passwords from the network. It also can be conversion of network traffic to human readable form. Besides that, packet sniffing also can be analysis the network to find the bottlenecks. It also can act as network intrusion detection of monitor for hackers or unauthorized users.

Packet sniffing is capturing and monitoring every packet sent in the network including packets not intended for it. It has a variety of ways to complete and depending on the type of network they are in. Basically, in network have two types of Ethernet environments there are Shared Ethernet and Switched Ethernet.

2.1.7.1 Five Basic Components

Packet sniffing is a combination of hardware and software, so this tool is composed of five basic components:

➤ Hardware

Most packet sniffing tool is software-based and work with standard operating system and NIC. However, there are some special hardware are offered for additional benefits such as Cyclic Redundancy Check(CRC) errors, voltage problems, cable problems, jitter,

jabber, negotiation errors. Packet sniffing only support Ethernet or wireless adapters, so sometimes you will also need a hub or a cable taps to connect to the existing cable.

➤ **Capture Driver**

This is the part of the packet sniffer that is responsible for actually capturing the network traffic from the network. It also filter out the traffic that you want and store the data in a buffer. This is the main of packet sniffer to capture packets.

➤ **Buffer**

This is component stores the captured data. Data can be stored in the buffer until it is full, or in a rotation method such as 'round robin' where the latest data replaces the oldest data. Buffer can be disk-based or memory-based.

➤ **Real-time analysis**

This feature analyzes the data as it comes off the cable. This help the packet sniffer to find network performance issues and network intrusion detection systems do this to look for signs of intruder activity.

➤ **Decode**

This component displays the contents of the packets which captured from the network with descriptions so that it is human-readable. Decodes are specific to each protocol, so packet sniffer tend to vary in the number of decodes they currently support. However, new decodes are constantly being added to network analyzers.

2.1.7.2 Shared Ethernet

In shared Ethernet environment, all hosts are connected to the hubs and competition with one another for bandwidth. In here, one user sent packets are received by all other users because hubs allow for a single broadcast domain, such as when PC A wants to send information to PC B, the information will pass to each computer on the network and say: “I am from PC A and looking for PC B.” Because the information is transmitted to the computers on this LAN, sniffing the packets can be easily launched.

When one of users on this LAN puts a network card in promiscuous mode and installs a sniffing program on a shared Ethernet, that user will able to collect all the packets on the network. The following figure will describe the shared Ethernet environment.

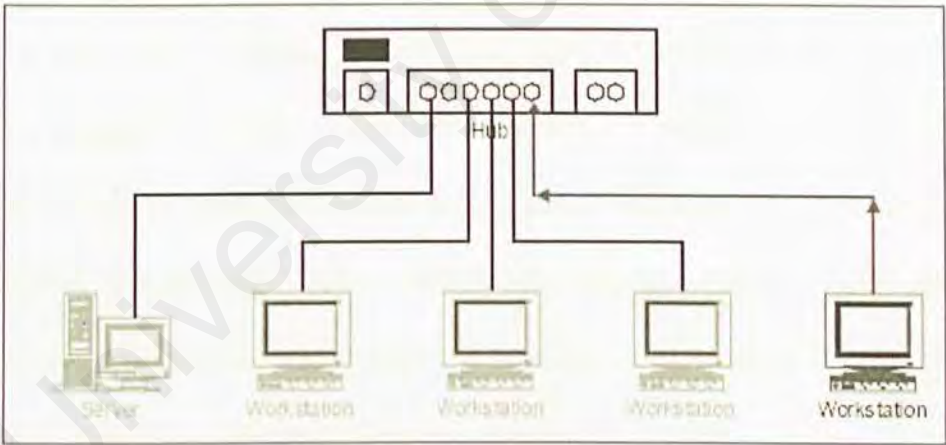


Figure 2-1: Packets sent from PC to hub

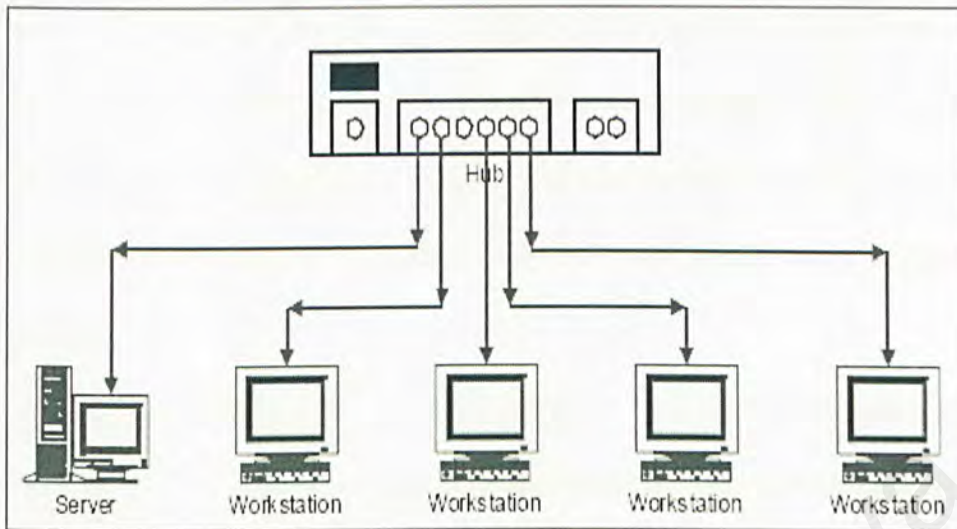


Figure 2-2: Hub forwards the traffic to all PC

2.1.7.3 Switched Ethernet

In Switched Ethernet environment, all hosts are connected to a switch. The switch maintains a table keeping tracks of each computer's MAC address and delivers the packets to a particular user's pc to the port on which that user's pc is connected. The switch is an intelligent device that sends the packets to the destined computer only and does not broadcast to all the pc on the network. This Switched Ethernet environment has a best network performance. Thus, most the network administrators assume attackers can't capture packet in a Switched Ethernet environment.

2.1.7.4 Methods of Packet Sniffing

Packet sniffing has three type of sniffing methods. Some methods work in non-switched and others work in switched networks. The sniffing methods are IP-based sniffing, MAC-based sniffing, and ARP-based sniffing.

- ❖ IP-based sniffing – this is original way of sniffing packet. It work by plug-in the network card into promiscuous mode and sniffing all packets which match with IP address. This way is hard works in switched networks because switch has IP address filter. However, without IP address filter, it can easily capture all the packets.
- ❖ MAC-based sniffing – this methods work by putting network card into promiscuous mode and sniffing all packets matching all the MAC address.
- ❖ ARP-based sniffing – this methods is different because it not plug-in the network card into promiscuous mode because ARP packets will be sent to us. It is due to ARP protocol is random, so sniffing can be done on the switched network. To launching this attacks, first must have poison the ARP cache of the two hosts that you want to sniff and identifying yourself as the other host in the network. Once the ARP caches are poisoned, the two host start their connection then sending the traffic directly to the other hosts. However, when in the middle sending the packets got other hosts to get the packets. Then modify the packets and send it to real intended host. This is called man-in-the-middle attack. See the Figure 2-1 in below to understand the way it works.

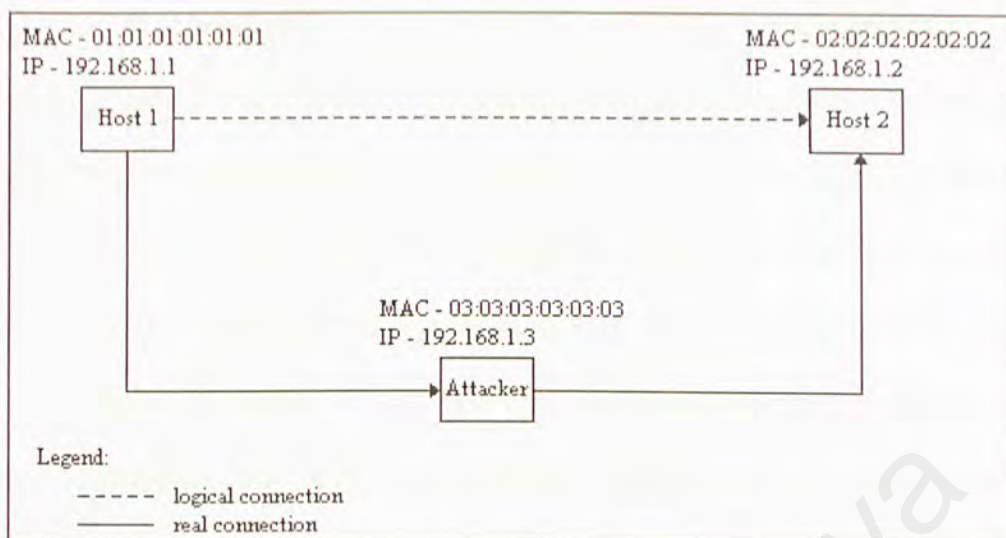


Figure 2-3: ARP Sniffing Method Reference [1]

Some attackers using sniffing methods to do unauthorized access, so some anti-sniffing tool is made to detect hosts on an Ethernet / IP network segment when attackers promiscuous gathering data. However, current anti-sniffing version is only work on non-switched network. Anti-sniffing perform different types of tests to determine whether a host is in promiscuous mode. Anti-sniffing has 3 tests to detect the attackers. There are DNS tests and operating system specific tests.

- ❖ DNS tests – this test is based on packet sniffing tools perform IP address to name lookups to provide DNS names in place of IP addresses. Attackers use this information to sniff packet because most of the time hosts are named for what they provide such as mail server being named mail.abc.com. To test this, anti-sniffer place the network card into promiscuous mode and sends the packets out onto the network aimed to bogus hosts. If any name lookups from bogus hosts are seen, a sniffer might be in action on the host performing the lookups.

❖ Operating system specific tests – this test is aimed at certain operating system.

There is the ARP test that is designed for Microsoft windows 95, 98, and NT.

Another is called Ether Ping test which designed for Linux and NetBSD kernel.

- ARP test – this test is to exploit the flaw found in Microsoft operating system analyze broadcast ARP packets. The network card driver checks for MAC address being that of the network card for unicast packet when in promiscuous mode. But, that only checks the first octet of the MAC address to determine that packet is broadcast or not. To test this flaw, send a packet with MAC address of FF: 00: 00: 00: 00: 00 and the correct destination IP address of the host. Microsoft OS using the flawed driver to respond while in promiscuous mode.
- Ether Ping test – In Linux kernels there is a specific condition that allows users to determine whether a hosts is in promiscuous mode or not. Every packet is passed on to OS when a network card is placed in promiscuous mode. Some Linux kernel is looking at IP address only in the packets to determine whether they should be processed or not. Anti sniffer always send the packets with a bogus MAC address and valid IP address to test for this flaw. To get a response, an ICMP echo request message is sent within the bogus packet leading to vulnerable hosts in promiscuous mode to respond.

2.1.7.5 Methods of Detecting Sniffer Reference [2]

In the network which runs at promiscuous mode is very hard to detect sniffer. It is because sniffer is used the powerful tool to capture packets from the network. There are some ways may help you to find a sniffer on the network:

- Run your own sniffer and monitor the DNS traffic of nominated host;
- Judge from some status, for example, if the rate of lost packets on your network communication is abnormally high, or one machine on network occupies biggish bandwidth for a long time, it may imply that a sniffer has been existed on your network;
- Check whether your system is in promiscuous mode, if yes, a sniffer may be running at the same time;
- Use anti-sniffer software to search sniffer in our system.

2.1.7.5 Methods of Defending Sniffer

Nowadays, still no any effective solution can be used to defense sniffer against its installation and attack to our systems. A lot of methods will be done by Network administrators to reduce the chances of attacked by sniffers. The most popular methods are always used as follows:

➤ Switch

Mostly sniffer prefer working on hub, it may make sniffer disable to replace the hub in your computer with a switch which transfers packets according as

destinations on network layer. To date with the cost and price decreasing greatly, switch is becoming a main sniffer defense tool both effective and economic.

➤ **Encryption**

Encrypting your data can reduce the effects of sniffer to your private information for that even a sniffer can capture all important data from you. It is due to they can not decode and read data which captured from you.

➤ **SSH (Secure Shell)**

SSH is a protocol offering secure communication for application programs, based on client / server mode. The distributive port of SSH server is 22, and links are built on RSA method. When authorization is complete, data transmitting will be encrypted with IDEA technique which is powerful generally.

F-SSH is the higher level of SSH, usually used by military communication. It offers the most powerful encryption for all purposes. That means if F-SSH is used on a site point, username and password will be not very important. At present, F-SSH is still the most advanced encryption and no one can pierce into it.

➤ **SSL (Secure Sockets layer)**

SSL initially presented by Netscape Corporation for the purpose of transferring data secretly and confidentially on Internet and has been applied widely on web.

SSL provides services from three aspects mainly:

- Identify user and server to make sure data will be sent to right client and server;

- Encrypt data to hide transmitted data;
- Keep data's integrity and prevent them from being modified during transferring.

Except above encryption techniques, there are some other tools you can try, like Kerberos, Deslogin, VPN, and SMB/CIFS.

Sniffer can work only in promiscuous mode, so it is crucial whether your system is in such mode or not. In the past, most network interface cards of DOS compatible computers did not support promiscuous mode but now it is the reverse. You shall enquiry system provider about the mode of your network interface.

2.2 TCP/ IP

TCP/IP (Transmission Control Protocol and Internet Protocol) is a name given to the collection or *suite* of networking protocols that have been used to construct the global Internet. TCP/IP is a set of protocols developed to allow all computers to share resources across a network. It was developed by a community of researchers centered on the Arpanet. Arpanet is the best-known TCP/IP network.

TCP/IP is consists two of the protocols in this suite called TCP (Transmission Control Protocol) and IP (Internet Protocol). These protocols work together to provide a basic networking framework that is used by much different application protocols. TCP/IP protocols are used on the internet and intranet.

Although the OSI reference model is universally recognized, the historical and technical open standard of the Internet is Transmission Control Protocol/Internet Protocol (TCP/IP). The TCP/IP reference model and the TCP/IP protocol stack make data communication possible between any two computers, anywhere in the world. Normally, networking protocols consist of different layers, with each layer responsible for a different aspect of the communication. The TCP/IP protocol suite is the combination of different protocols at various layers. TCP/IP is normally considered to be comprised of 4 layers, where each layer has its own specific responsibility

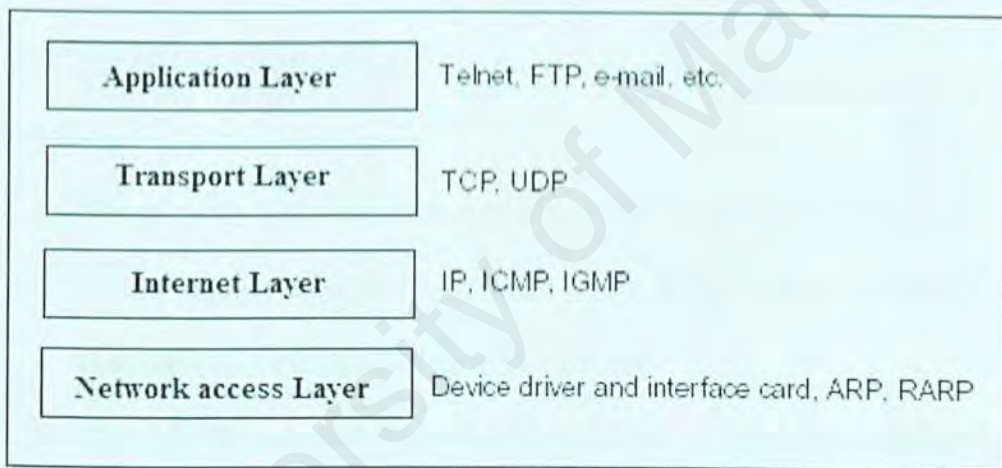


Figure 2-4: TCP/IP model

Application Layer

- Handles high level protocols, issues of representation, encoding, and dialog control.
- The TCP/IP combines all issues related to application into one layer, and assures this data is properly packaged for the next layer.

Transport Layer

- Deals with the quality of service issues of reliability, flow control, and error correction.
- The transmission control protocol (TCP), provides excellent and flexible ways to create reliable, well-flowing, low-error network communications.
- TCP is a connection-oriented protocol. It dialogues between source and destination while packaging application layer information into units called segments.
- Connection-oriented means that Layer 4 segments travel back and forth between two hosts to acknowledge the connection exists logically for some period (packet switching).

Internet Layer

- The purpose of the internet layer is to send source packets from any network on the internet work and have them arrive at the destination independent of the path and networks they took to get there.
- The specific protocol that governs this layer is called the Internet protocol (IP). Best path determination and packet switching occur at this layer.
- In an analogy of the postal system. When you mail a letter, you don't care how it gets there (there are various possible routes), but you do care that it arrives.

Network Access Layer

- The name of this layer is very broad and somewhat confusing. It is also called the host-to-network layer.

- It is the layer that is concerned with all of the issues that an IP packet requires to actually make a physical link. It includes the LAN and WAN technology details, and all the details in the OSI physical and data link layers.

In the TCP/IP model, regardless of which application requests network services, and regardless of which transport protocol is used, there is only one network protocol - internet protocol, or IP. This is a deliberate design decision. *IP* serves as a universal protocol that allows any computer, anywhere, to communicate at any time.

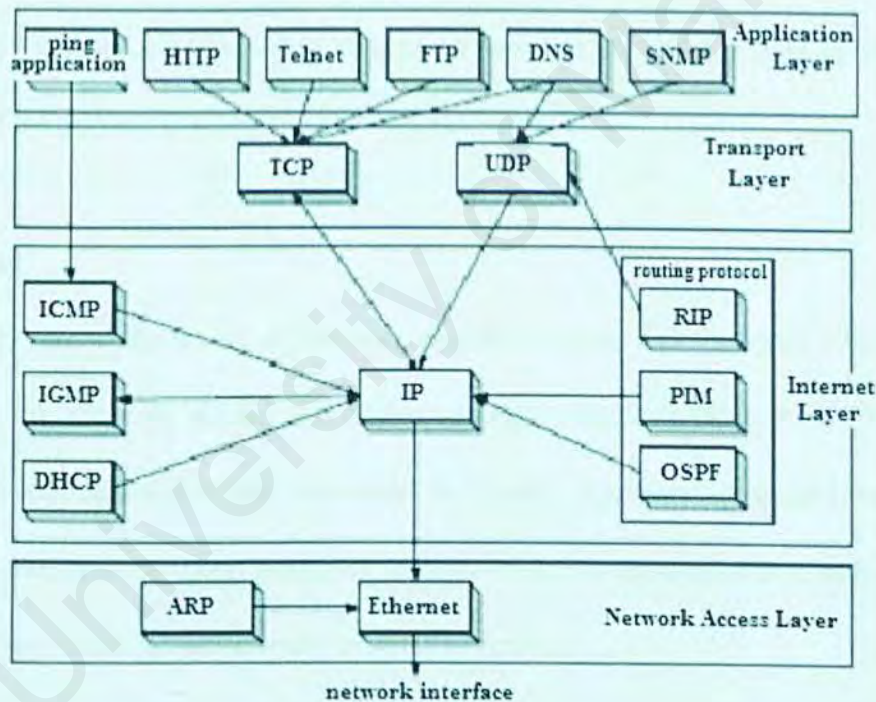


Figure 2-5: TCP/IP model

There are many different protocols in the TCP/IP protocol suite.

- TCP is one of the two predominant transport layer protocols. It uses IP as the network layer. TCP provide reliable transmission of data over a logical connection to HTTP

- IP is the main protocol at the internet layer. Every piece of data that gets transferred around an internet layer goes through the IP layer at both end systems and at every intermediate router. IP provide unreliable transmission of IP datagram across an IP network.
- ICMP is an attachment to IP. It is used by the IP layer to exchange error messages and other vital information with the IP layer in another host or router.
- Ethernet provide service to IP by giving transmission of a frame across an Ethernet segment for IP.
- ARP is a specialized protocol used only with certain types of network interfaces like Ethernet and token ring, to convert between the address used by the IP layer and the address used by the network interface.

2.2.1 Socket

A socket is one endpoint of a two-way communication link between two programs running on the network. A socket is bound to a port number so that the TCP layer can identify the application that data is destined to be sent. A socket is the combination of an IP address and a port number. A socket is also a virtual communication conduit between two processes. These processes may be local or remote.

Basically, a server runs on specific computer and has a socket that is bound to a specific port number. Then the server just waits for listening to the socket for client to make connection request.

On the client side, the client only knows the server hostname and the port number to which the server is connected. The client tries to contact with the server on the server machine and port when connection request is establish.

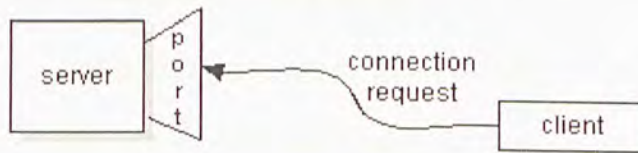


Figure 2-6: Client make connection request

On the server side, server will accepts connection if everything goes well. When server accepts the connection, it gets new socket bound to a different port. It needs new socket which consequently a different port number so that it can continue to listen to the original socket for connection requests while tending to the needs of the connected client.



Figure 2-7: Server accept connection request

On the client side, if connection is accepted, a socket is created successful and the client can use the socket to communicate with the server. The client and server can communicate each other by writing or reading from their socket.

2.2.2 Port

Ports are used in the TCP to name the ends of logical connections which carry long term conversations. For the purpose of providing services to unknown callers, a service contact port is defined. This list specifies the port used by the server process as its contact port. The contact port is sometimes called the "well-known port". A connected application must be bound to at least one port. *Binding* means that a port is assigned to a socket used by an application. The application is registered with the system. All incoming packets that contain the port number of the application in the packet header are given to the application socket.

The port numbers are divided into three ranges: the Well Known Ports, the Registered Ports, and the Dynamic and/or Private Ports. The Well Known Ports are those from 0 through 1023. The Registered Ports are those from 1024 through 49151. The Dynamic and/or Private Ports are those from 49152 through 65535. The Well Known Ports are assigned by the IANA and on most systems can only be used by system (or root) processes or by programs executed by privileged users. To the extent possible, these same port assignments are used with the UDP. The range for assigned ports managed by the IANA is 0-1023. There are lists of ports that defined as mentioned above and some examples are given as follow:

Service	TCP	Notes
FTP	21	File Transfer Protocol [Control]
SSH	22	Secure Shell
TELNET	23	Telnet

SMTP	25	Simple Mail Transfer Protocol
DNS	53	Domain Name Server
Whois++	63	Whois++
TFTP	69	Trivial File Transfer
HTTP	80	HyperText Transfer Protocol (e.g. for web browsing).
Kerberos	88	Kerberos
POP3	101	Post Office Protocol - Version 3
SNMP	161	Simple Network Management Protocol

Table 2-1: reserved ports assigned lists

2.3 Analysis Studies

2.3.1 Case Study 1 – Distinct Network Monitor Reference [3]

Packet	Length	Time (s)	Src IP	Dst IP	Src Port	Dst Port	T..	Description
14	243	5.263	202.185.109.172	202.185.109.191	138	138		SMB Command SMB_COM_TRANSACTION Re...
15	60	6.201						STP Configuration
16	175	6.358	202.185.110.128	239.255.255.250	1030	1900		UDP iadl -> ssdp len: 141
17	62	7.057	202.185.109.188	224.0.0.2	1985	1985		UDP hsrp -> hsrp len: 28
18	62	7.122	202.185.109.189	224.0.0.2	1985	1985		UDP hsrp -> hsrp len: 28
19	74	7.150	202.185.109.188	224.0.0.10				IGRP Request
20	60	8.082						Ethernet 00:0D:BD:EB:A3:AD -> 00:0...
21	60	8.195						STP Configuration
22	74	9.378	202.185.109.189	224.0.0.10				IGRP Request
23	62	9.929	202.185.109.188	224.0.0.2	1985	1985		UDP hsrp -> hsrp len: 28
24	62	10.062	202.185.109.189	224.0.0.2	1985	1985		UDP hsrp -> hsrp len: 28

<ul style="list-style-type: none"> UDP: Source Port 1985 (hsrp) -> Destination Port 1985 (hsrp). The Source IP Address 202.185.109.188 contacts the destination IP Address 224.0.0.2 with 20 bytes of higher protocol data Source Port: 1985 (hsrp) Destination Port: 1985 (hsrp) The length of this datagram including the header: 28 bytes Checksum: 0x4705 IP: Source IP 202.185.109.188 -> Destination IP 224.0.0.2 Destination address is a multicast address. This datagram is not fragmented. IP contains 28 bytes of higher protocol data Ethernet: Source MAC 00:04:C0:F8:02:44 -> Destination MAC 01:00:5E:00:00:02 IPv4 	<pre> 0000 01 00 5E 00 00 02 00 04 C0 ..^..... 0009 F8 02 44 08 00 45 C0 00 30 e.D..E.A.0 0012 00 00 00 00 00 02 11 9F 85 CA Y.. 001B B9 6D BC E0 00 00 02 07 C1 'm&a.... 0024 07 C1 00 1C 47 05 00 00 08 .A..G.... 002D 03 0A 6E 00 00 63 69 73 63 ...n..cisc 0036 6F 00 00 00 CA B9 6D BE o....'m& </pre>
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Figure 2-8: Distinct Network Monitor

The Distinct Network Monitor is a network security tool which combines the packet sniffing tools and network monitoring. This tool as well as normal network monitoring because it can capture network traffic and translates the protocol into simple English and provide great details of the packets plus the analysis of the network by gathering the information provided by the packets at a specific period of time.

This program show the basic information of the packets such as length of the packets, time, source IP address, destination IP address, source port, destination port, and brief description of the segment like protocol and direction of the packet.

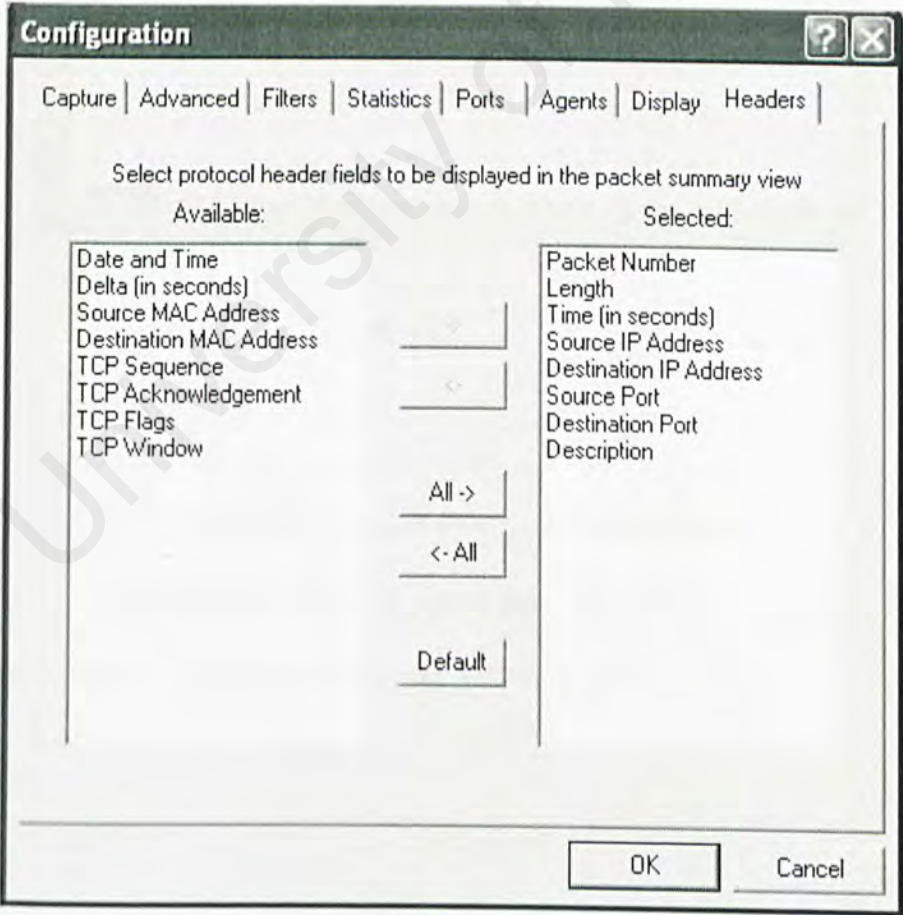


Figure 2-9: Configuration

Before running this program, configuration must be done first to make sure what the details information of the packets user need. Normally user is want to see the information are packet number, length, time, source and destination IP address, source and destination port and description. These features must be adding before starting to capture the packets and just can show the result same with Figure 2-9.

User can view the statistic of the traffic by categories it with distribution like IP Protocols Distribution, IP Traffic Distribution by IP Address, Subnet Traffic Distribution by MAC address, Packet Size Distribution and Summary.

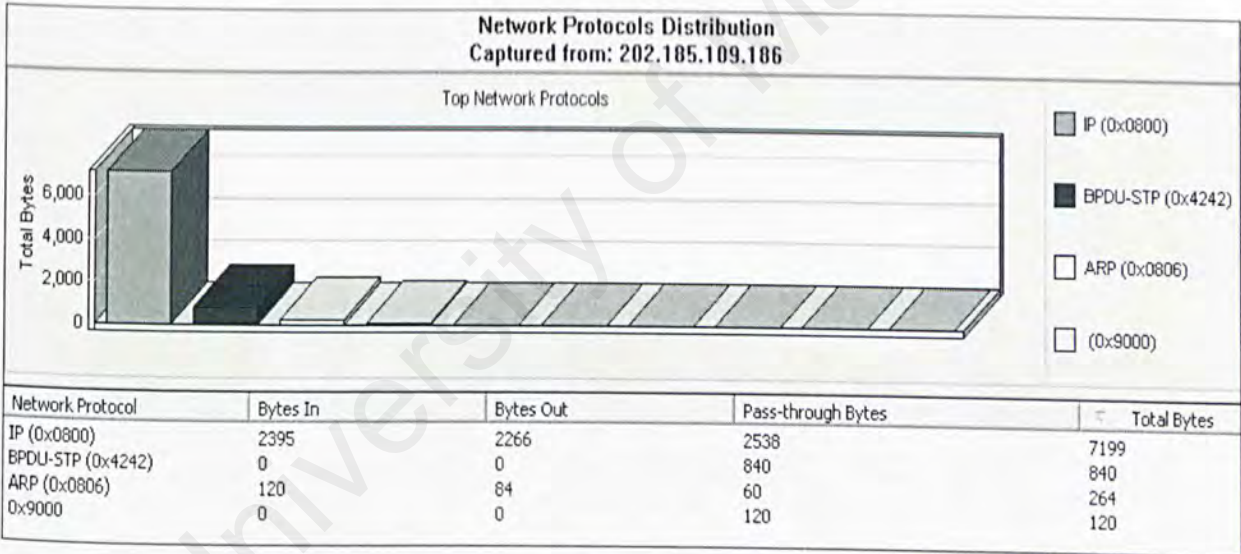


Figure 2-10: Network Protocols Distribution

As shown in figure 2-10, this distribution shows the total bytes pass through the network categorized by network protocols like IP and ARP in a graph.

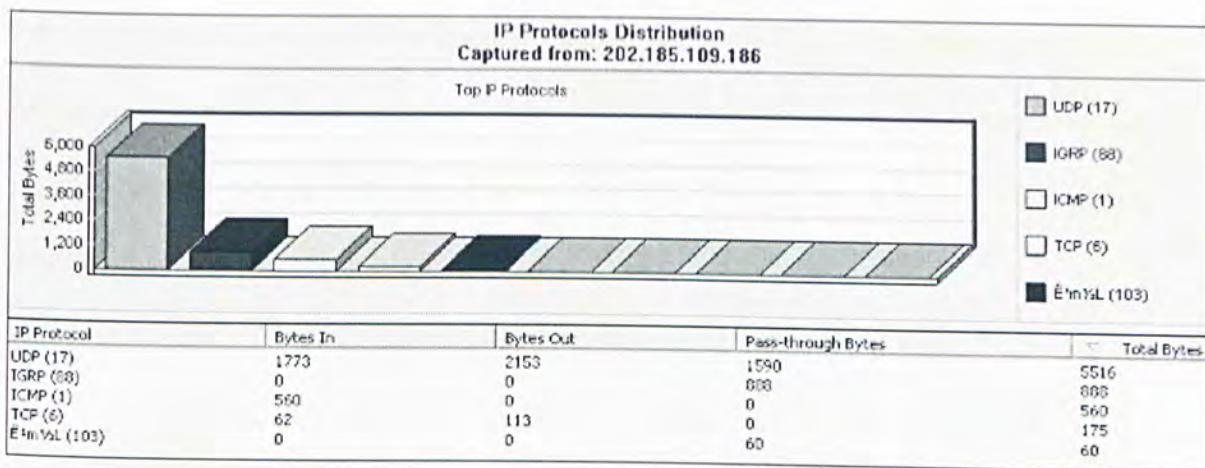


Figure 2-11: IP Protocols Distribution

This distribution shows a list of transport protocols and the total number of bytes and packets transmitted for each one (figure 2-11).

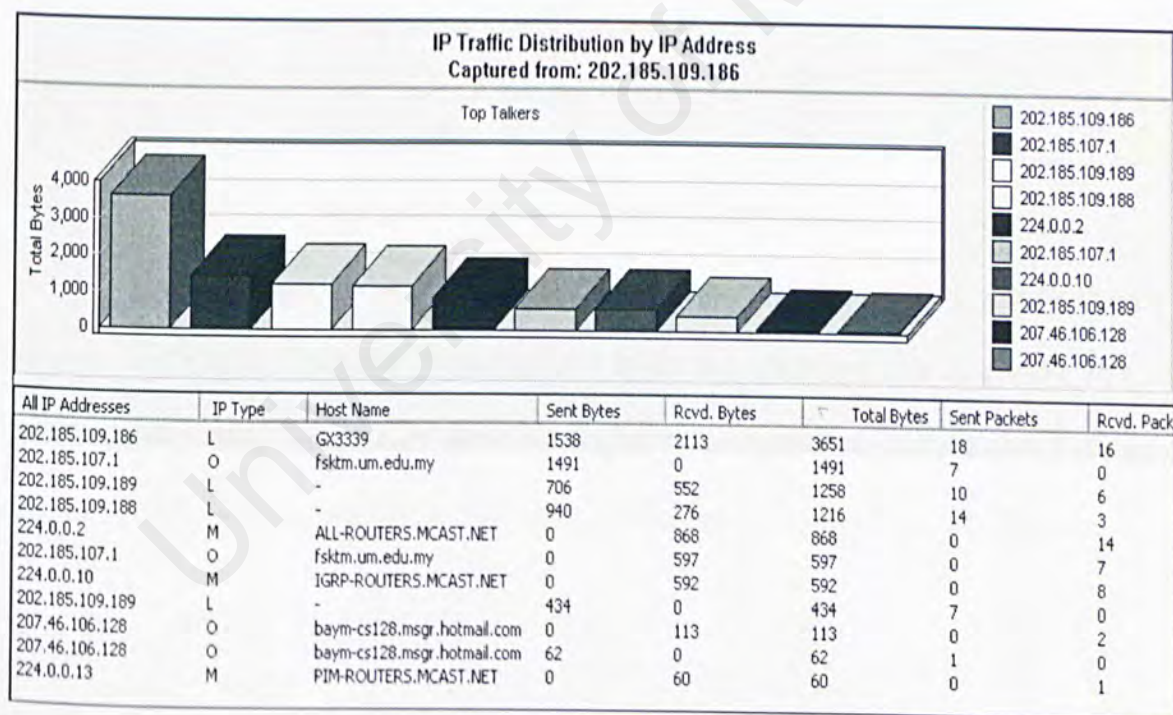


Figure 2-12: IP Traffic Distribution

This distribution shows the list of IP addresses that are active on the network segment, showing the total number of bytes sent and received by each IP address. The type of IP address is shown which may be L for an IP address on the local network, B

for a broadcast address, or M for multicast addresses. Note that if the system listed is not on the same hub, the traffic numbers do not indicate the total traffic for that system, but just the traffic created between it and other systems on the hub or switch being monitored.

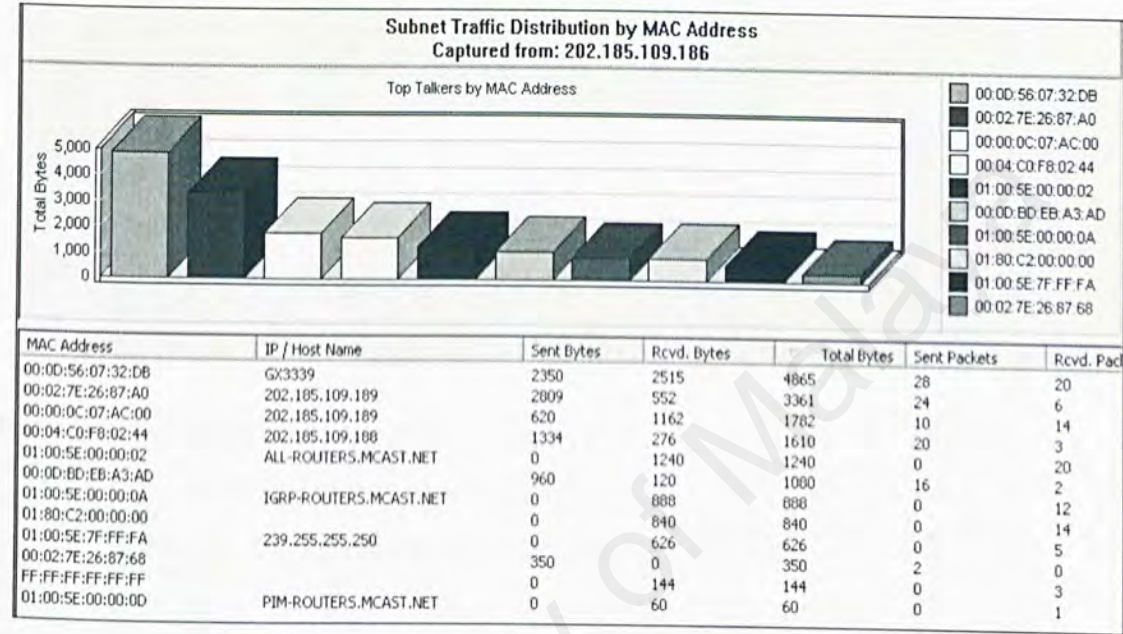


Figure 2-13: Subnet Traffic Distribution

This distribution shows the list of MAC addresses that are active on the network segment and the total number of packets and bytes that were sent and received by each MAC address. This includes all packets whether IP or otherwise that are over Ethernet or Token Ring and may include packets that are not parsed by the Network Monitor

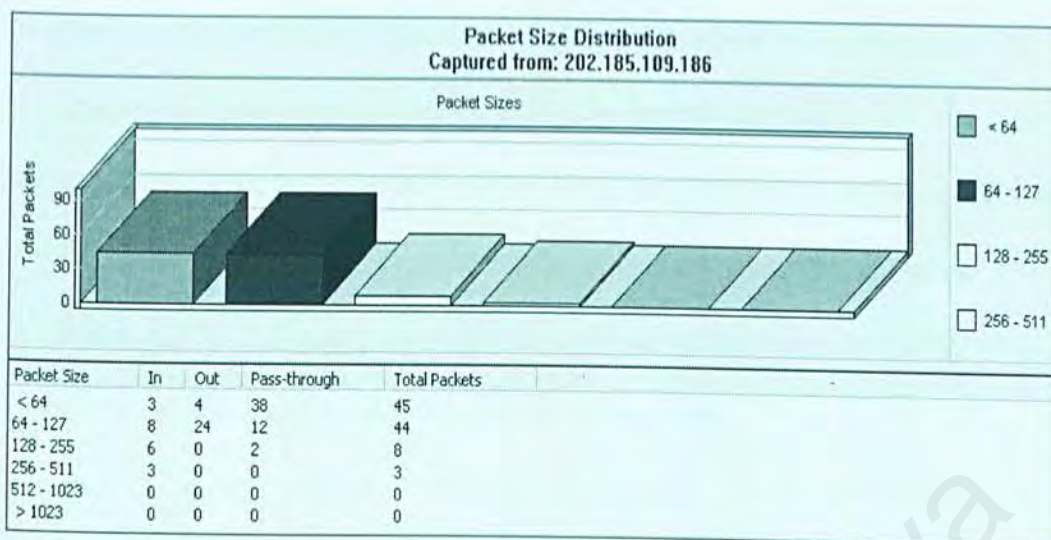


Figure 2-14: Packet Size Distribution

The Analysis of Packet size distribution showing the number of packets transmitted in various size ranges.

Summary	
Captured from: 202.185.109.186	
Summary Name	Value
Start time	03:02:02 PM on Friday, July 30, 2004
End Time	03:02:33 PM on Friday, July 30, 2004
Elapsed Time	0 days, 0 hours, 0 minutes and 30 seconds
Paused Time	0 days, 0 hours, 0 minutes and 0 seconds
Capture Time	0 days, 0 hours, 0 minutes and 30 seconds
Total packets/bytes seen by network adapter(s)	100 / 8423
Packets/bytes filtered out by the current capture filter	0 / 0
Actual packets/bytes written to file	100 / 8423
Packets dropped because of collisions	0
Incoming packets	20
Outgoing packets	28
Pass-through packets	52
Filter On 202.185.109.186	none

Figure 2-15: Summary Distribution

This summary of the statistics recorded during the session are shown in this distribution.

Strength:

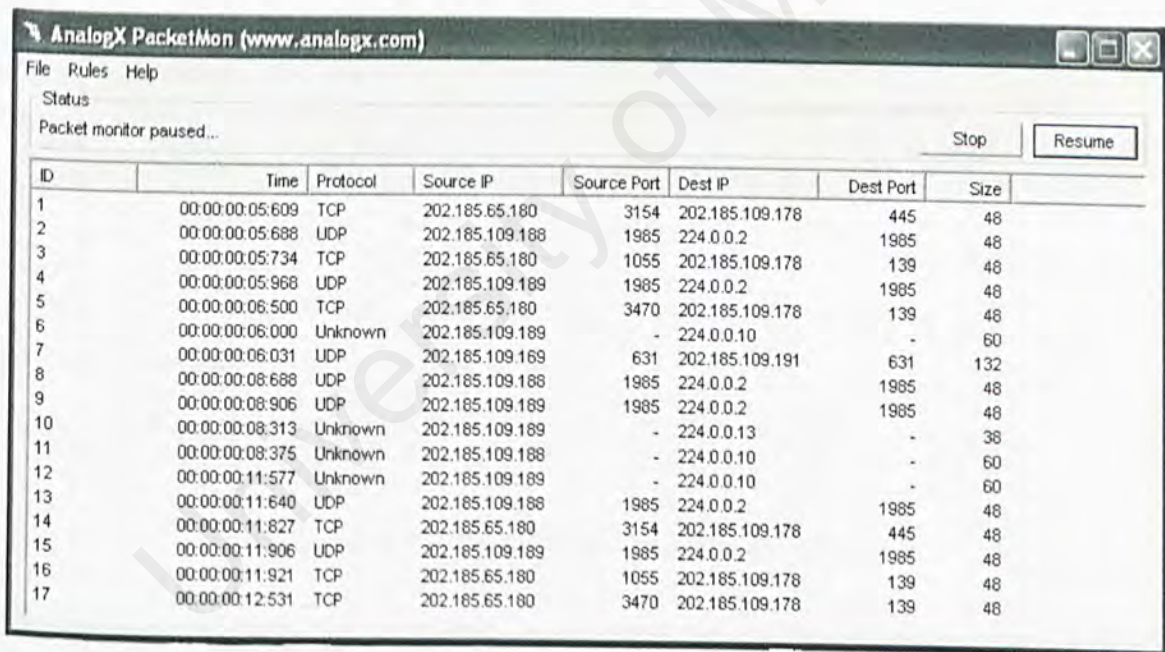
- This program has a friendly graphic user-interface for user to use it to capture the packet in small organization.

- The result has to show all the details information about the packet like all IP address and hostname are listed for sent and receive the packet.
- Total size of the packet which captured also has listed in the report. It can make the network administrator can know about the user's network activities.
- It also has the function to print put the summary distribution for the network administrator when use it to monitoring the network.

Weakness:

- Very complicated software and certain users not easy to use it.

2.3.2 Case Study 2 – AnalogX PacketMon Reference [4]



The screenshot shows the AnalogX PacketMon application window. The title bar reads 'AnalogX PacketMon (www.analogx.com)'. The menu bar includes 'File', 'Rules', and 'Help'. The status bar indicates 'Status' and 'Packet monitor paused...'. There are 'Stop' and 'Resume' buttons. The main display area contains a table of captured packets.

ID	Time	Protocol	Source IP	Source Port	Dest IP	Dest Port	Size
1	00:00:00:05:609	TCP	202.185.65.180	3154	202.185.109.178	445	48
2	00:00:00:05:688	UDP	202.185.109.188	1985	224.0.0.2	1985	48
3	00:00:00:05:734	TCP	202.185.65.180	1055	202.185.109.178	139	48
4	00:00:00:05:968	UDP	202.185.109.189	1985	224.0.0.2	1985	48
5	00:00:00:06:500	TCP	202.185.65.180	3470	202.185.109.178	139	48
6	00:00:00:06:000	Unknown	202.185.109.189	-	224.0.0.10	-	60
7	00:00:00:06:031	UDP	202.185.109.189	631	202.185.109.191	631	132
8	00:00:00:08:688	UDP	202.185.109.188	1985	224.0.0.2	1985	48
9	00:00:00:08:906	UDP	202.185.109.189	1985	224.0.0.2	1985	48
10	00:00:00:08:313	Unknown	202.185.109.189	-	224.0.0.13	-	38
11	00:00:00:08:375	Unknown	202.185.109.188	-	224.0.0.10	-	60
12	00:00:00:11:577	Unknown	202.185.109.189	-	224.0.0.10	-	60
13	00:00:00:11:640	UDP	202.185.109.188	1985	224.0.0.2	1985	48
14	00:00:00:11:827	TCP	202.185.65.180	3154	202.185.109.178	445	48
15	00:00:00:11:906	UDP	202.185.109.189	1985	224.0.0.2	1985	48
16	00:00:00:11:921	TCP	202.185.65.180	1055	202.185.109.178	139	48
17	00:00:00:12:531	TCP	202.185.65.180	3470	202.185.109.178	139	48

Figure 2-16: AnalogX PacketMon

The PacketMon is a packet monitoring that only can run in Windows 2000 and Windows XP platform. A test on the program as was shown in above on interface 202.185.109.186 and the program was working properly to sniff the packets flow on the

network. Within five minutes this program was able to capture 17 packets and these packets are shown in a list of table. These packets are named with an ID and the time it was captured was recorded. Other information like protocol, source IP address, source port, destination IP address, destination port and packet size also can be resolved.

The full detailed information of the packet in a new table will pop-up by double-clicks the specific packet in the table. The more detailed information is about fields in the Internet header like identifier, fragment offset, time to live, checksum, control flags and type of service. Besides the content of the packet in hexadecimal and ASCII code is also available to be displayed.

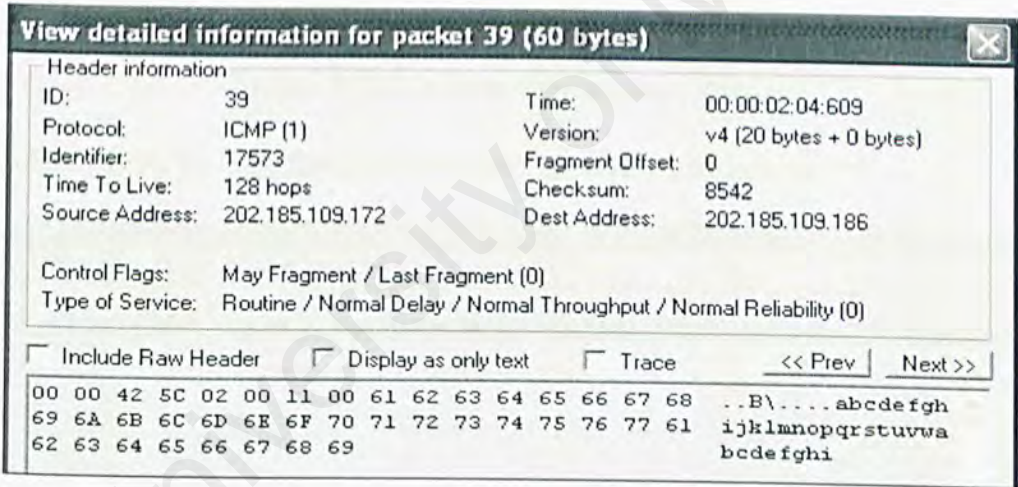


Figure 2-17: Detailed information for packet

Strength:

1. Shows the information of each packet in a well organized list of table sorted with ID and time captured.
2. Have stop and resume function for sniffing.

3. Able to generate an “Unable to open raw socket, packet monitoring cancelled..” error message when there was an error message and cancel the sniffing when unable to open the raw socket or the adapter does not exist.
4. User can easily analyse the content of the packet in the Internet layer of the TCP/IP model (IP header) through the program.
5. The “Include Raw Header” function enable user to view the packet separately the IP header and IP datagram.

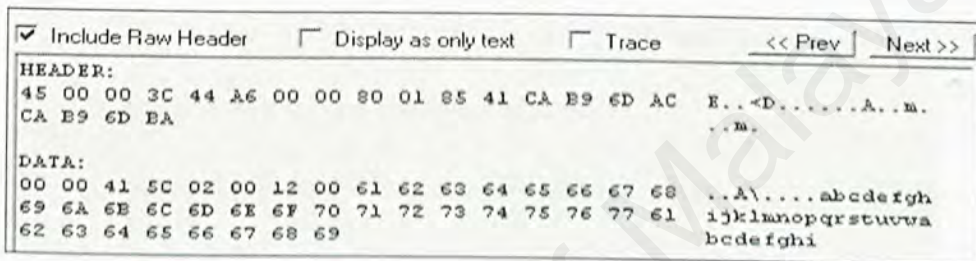


Figure 2-18: Include raw header function

2.3.3 Case Study 3 – The Ethereal Network Analyzer Reference [5]

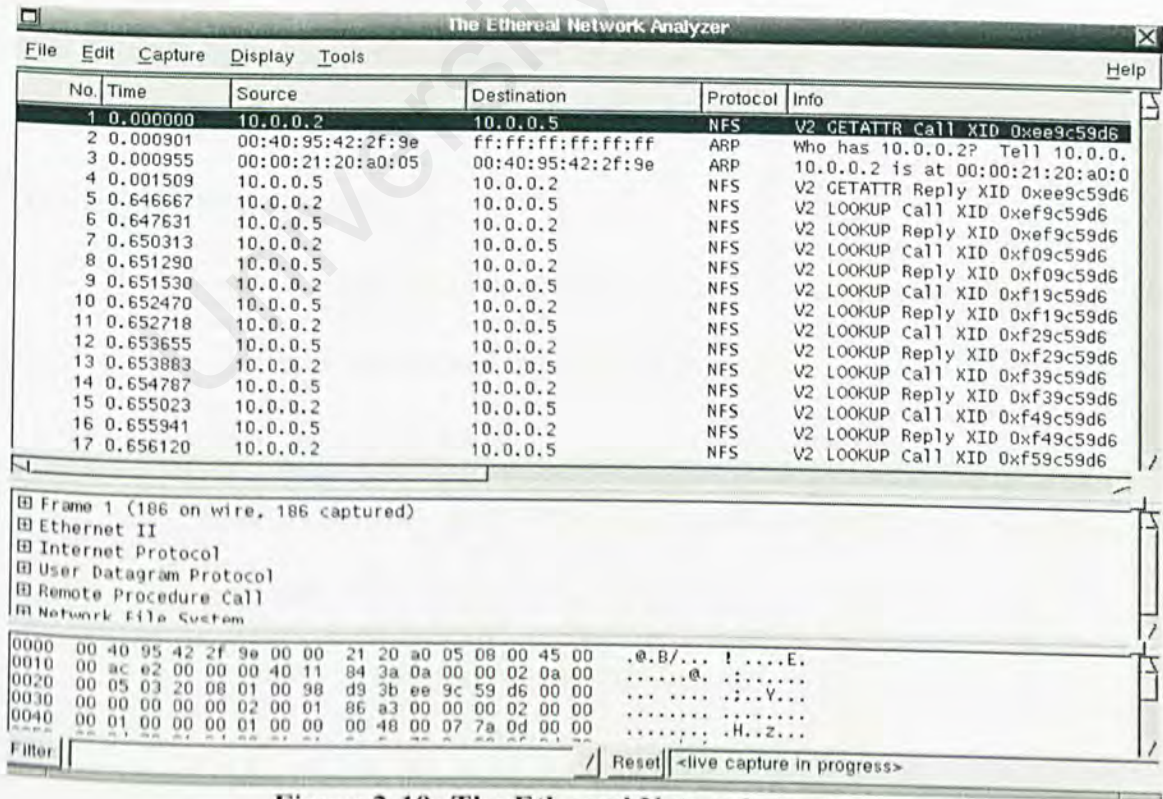


Figure 2-19: The Ethereal Network Analyzer

The Ethereal Network Analyzer is the software used to capture the packets off the network and analyze the packets. The Ethereal is the best open source packet sniffing tool. It is available for use on Windows and UNIX and captures and displays the packets like shown in figure 2-19.

From that screen shot, Ethereal contains three windows:

- First window displays the summary information of the packets captured. It displays some features such as time, source and destination IP address and MAC address, protocol, and info. By clicking on the packets, the other windows will appear.
- Second window displays the details information of the packets captured.
- Third window displays the data of the packets captured.

The Ethereal Network Analyzer has the menu on the top of the screen. The menu contains file, edit, capture, display, tools, and help items and each of them also has their function. The following items:

- File – to open and reload capture files, save capture files, print capture files, print packets and quit Ethereal program.
- Edit – to find the frame and go to the frame, mark one or more frames, set your preference, create filters, and enable or disable the dissection of protocols.
- Capture – to start and stop the capture.

- Displays – to modify display options, match selected frames, colorize frames, expand all frames, collapse all frames, show the packet in a separate window, and configure user specified decode.
- Tools – to display loaded plug in, follow TCP stream, display hierarchy statistics.
- Help – show the basic help.

Strength:

1. Friendly – Users interface and shows the information of each packet in list of table and time captured.
2. Easy to start and stop capture the packets on the network with clicking capture button.
3. Each packet can view the details easily by clicking the packets which captured off the network.
4. It can save the packets are captured easily with clicking save button.

Weakness:

1. It not have display how many packets will sent and receive by the users.
2. It combines the packet's IP address and MAC address and will see more complex.
3. It no shows the summary report for all the packets are captured.

2.4 Language

2.4.1 Visual Basic

A programming language and environment developed by Microsoft. Based on the BASIC language, Visual Basic was the first products to provide a graphical programming environment and a paint metaphor for developing user interfaces. Instead of worrying about syntax details, the Visual Basic programmer can add a substantial amount of code simply by dragging and dropping *controls*, such as buttons and dialog boxes, and then defining their appearance and behavior.

Since its launch in 1990, the Visual Basic approach has become the norm for programming languages. Now there are visual environments for many programming languages, including C, C++, Pascal, and Java. Visual Basic is sometimes called a Rapid Application Development (RAD) system because it enables programmers to quickly build prototype applications.

2.4.2 C / C++

C programming language was developed at Bell Labs during the early 1970's. C is rather like Pascal or FORTRAN. Values are stored in variables. Programs are structured by defining and calling functions. Program flow is controlled using loops, if statements and function calls. Input and output can be directed to the terminal or to files. Related data can be stored together in arrays or structures.

C++ programming language is an object-oriented programming. C++ is replacing the more traditional structured programming techniques. Because C++ retains C as a subset,

it gains many of the attractive features of the C language, such as efficiency, closeness to the machine, and a variety of built-in types. A number of new features were added to C++ to make the language even more robust, many of which are not used by novice programmers. By introducing these new features here, we hope that you will begin to use them in your own programs early on and gain their benefits. Some of the features we will look at are the role of constants, inline expansion, references, declaration statements, user defined types, overloading, and the free store.

2.4.3 C#

C# (pronounced C-Sharp, just like in musical notation) is a new language for Windows applications, intended as an alternative to the main previous languages, C++ and VB. C# is Developed at Microsoft by a team led by Anders Hejlsberg and Scott Wiltamuth. It incorporated into .NET platform which can use as Web based applications and. C# also as network programming by quickly prototype and deploy network applications using C# classes. Combining the C# Forms library to write the graphical code with the C# Socket library to write the networking code makes creating professional network applications simple. With C# network classes, what used to take a day to write often only takes an hour or less. The C# language, a close cousin to Java, is a new object-oriented programming language (OOPL) designed to work within the .NET framework. It improves upon many of the vague or ill-defined areas of C++ that frequently lead programmers into trouble. C# is a strongly-typed, object-oriented language designed to give the optimum blend of simplicity, expressiveness, and performance.

2.4.4 JAVA

Java is an object-oriented programming language with a built-in application programming interface (API) that can handle graphics and user interfaces. It can be used to create applications or applets. Because of its rich set of API's, similar to Macintosh and Windows, and its platform independence, Java can also be thought of as a platform in itself. Java also has standard libraries for doing mathematics.

Java's syntax most is the same as C and C++. One major difference is that Java does not have pointers but the biggest difference is that you must write object oriented code in Java. Procedural pieces of code can only be embedded in objects. In the following we assume that the reader has some familiarity with a programming language. In particular, some familiarity with the syntax of C/C++ is useful.

In Java we distinguish between applications, which are programs that perform the same functions as those written in other programming languages. Java can be embedded in a Web page and accessed over the Internet with using applets.

Comparison of C/C++, C# and JAVA Language

	Java	C#	C++
Object-Orientation	Hybrid	Hybrid	Hybrid / Multi-Paradigm
Static / Dynamic Typing	Static	Static	Static
Generic Classes	No	No	Yes
Inheritance	Single class, multiple interfaces	Single class, multiple interfaces	Multiple
Feature Renaming	No	No	No

	Java	C#	C++
Method Overloading	Yes	Yes	Yes
Operator Overloading	No	Yes	Yes
Higher Order Functions	No	No	No
Lexical Closures	No	No	No
Garbage Collection	Mark and Sweep or Generational	Mark and Sweep or Generational	None
Class Variables / Methods	Yes	Yes	Yes
Reflection	Yes	Yes	No
Access Control	public, protected, "package", private	public, protected, private, internal, protected internal	public, protected, private, "friends"
Design by Contract	No	No	No
Multithreading	Yes	Yes	Libraries
Regular Expressions	Standard Library	Standard Library	No
Pointer Arithmetic	No	Yes	Yes
Language Integration	C, some C++	All .NET Languages	C, Assembler
Built-In Security	Yes	Yes	No
Object-Oriented Features			
Encapsulation / Information Hiding	Yes	Yes	Yes
Inheritance	Yes	Yes	Yes
Polymorphism / Dynamic Binding	Yes	Yes	Yes
All pre-defined types are Objects	No	No	No
All operations are messages to Objects	No	No	No
All user-defined types are Objects	Yes	Yes	No

Table 2-2: Comparison among the three Languages

2.5 Operating System











Operating System	Unique	Reload	Total	Share	Graph
Windows 98	117,796	77,470	195,266	55.84%	
Windows 2000 / XP	47,576	35,597	83,173	22.55%	
Other/unknown	16,818	8,910	25,728	7.97%	
Windows 95	9,287	6,802	16,089	4.40%	
Windows ME	8,302	7,864	16,166	3.94%	
Windows NT	8,248	5,596	13,844	3.91%	
Linux 2	1,567	598	2,165	0.74%	
MacOS	1,014	278	1,292	0.48%	
Unix/unknown	32	3	35	0.02%	
OS/2	3	0	3	0.00%	

Table 2-3: Operating System Statistic Reference [6]

An operating system is a set of programs that controls the hardware of a computer how to works. An operating system deals with the loading of software and management of memory. The operating system enables applications to save files to disk or print them via the printer. The operating system also determines what applications software will run on it. For security, OS ensures that unauthorized users do not access the system. The most widely used operating systems are **Windows**, **LINUX** and **UNIX**.

2.5.1 UNIX

UNIX is a much older operating system that was created in the late 1960s. UNIX is a layered OS. UNIX is interacting directly with the hardware and provides the services to the user programs. Unix provide a set of standard system calls for accessing a file, starting or updating accounting records, changing ownership of a file or directory, changing to a new directory, creating, suspending, or killing a process, enabling access to hardware devices, and setting limits on system resources.

UNIX is a multi-user, multi-tasking operating system. You can have many users logged into a system simultaneously, each running many programs. UNIX keep each process and user separate and to regulate access to system hardware, including CPU, memory, disk and other I/O devices.

Pros: - was written for powerful microcomputer systems and has strong multi-tasking capability

- manage large quantities of memory well
- not frequently crash
- performs very well in a networking environment
- support for remote management

Cons: - Unix industry standards are not uniform

- requires powerful, large microcomputer system
- no user-friendly GUI
- does not include some customized applications development and web publishing features that Windows has

2.5.2 Linux

Linux is a free open source operating system originally developed by Linus Torvalds in the early 1990s. It was released as open source and remains a worldwide developed operating system mainly distributed via the Internet.

Linux become a popular operating system for Internet/ intranet serving purposes. With a host of performance enhancements that will benefit Web sites and Internet sites of all sizes. Linux has made progress, primarily in functionality important to Internet

infrastructure and Web server capabilities, including a greater selection of drivers, easier installation, and GUI-based front ends for Web administration and window management.

Pros: - better network management tools

- new support for SMP and RAID arrays
- improved compiler performance

Cons: - reliance on Red Hat tools to the detriment of system administrators

- Administration tools are less sophisticated than those found in more mature Unix version

2.5.3 Macintosh

The Macintosh operating system runs on Macintosh computer. It offers a high-quality graphical user interface and is very easy to use. The Macintosh operating system is contained in two primary files – the system file and the Finder which work together to perform the standard operating system procedures, such as copying files, erasing files and running application programs. In addition, the system file and the Finder also manage the user interface, display menu and activate tasks that chosen by the user from the menus.

Pros: - multi-tasking system that allows more than one application to run at one time

- supported Graphical User Interface (GUI)
- easy to learn and use
- manage large quantities of memory
- mouse skills vital to react with the operating system

Cons: - compatible software not always easily available

- requires expensive hardware
- historically not viewed as a professional computer but more towards education and game playing

2.5.4 Windows XP

Windows XP is Microsoft latest version of popular windows operating system. Windows XP has two version, home and professional version. Both of two versions have a sleek and clean new interface that is more customizable than before and it looks great, with rounded window corners, larger and more detailed icons, and a clean-look desktop that on first installation shows only the taskbar and recycle bin.

Both of two versions provide Fast User Switching is a great feature for computers used by more than one person. It lets another user log on without killing the other user's session. When you switch back, running applications and open documents are there as you left them. This is impressive, but what really counts is that *XP* properly understands how to deal with multiple users. Each user has their own special folders, like My Documents, which cannot be seen by other users.

XP professional version has best performance in LAN, the Network Setup wizard simplifies setting up a network, and there is built-in support for 802.11. This is peer-to-peer networking, and you can also connect to a domain resource such as a shared directory or printer by entering a user name and password. Business users' note: unlike Windows 98 or ME, cannot join a Windows server domain, so the networking is peer-to-peer only.

- Pros:
- provides better integration of Windows 9x and Windows NT that did Windows 2000
 - uses slightly more total memory for the OS to add features than does Windows 2000
 - offers significant GUI enhancements
 - built-in support for compressed files
 - advanced file sorting options
 - more stable and improved troubleshooting tools
- Cons:
- requires nearly 1GB hard drive space
 - Programs used with Windows XP may need more than minimum system specification nearly eliminates support for device drivers not approved by Microsoft. Security concerns with centralized storage of on-line information is Microsoft Passport, a repository of the user IDs and passwords that users use on the internet

2.6 Authoring Tools

2.6.1 Microsoft Visual Studio .NET 2003

Microsoft Visual Studio .NET 2003 is a software development tool designed for programmers to build software and application tools. It combines VS.NET and programming language C# together and provides a very intuitive and user-friendly interface for beginning programmers.

Microsoft Visual Studio .NET 2003 enhances, further refines, and is highly compatible with its predecessor. With Visual Studio .NET 2003, the integrated development

environment (IDE) provides a consistent interface for all languages, including Microsoft Visual Basic .NET, Microsoft Visual C++ .NET, Microsoft Visual C# .NET, and Microsoft Visual J# .NET. Using the language best suited to your skill set, you can take advantage of shared visual designers to build rich Windows-based applications and dynamic Web applications that render in any browser.

Microsoft Visual Studio .NET 2003 enables developers to build Internet applications. It also provides developers with the tools for integrating solutions across operating systems and languages. With Visual Studio .NET, developers can easily convert existing business logic into reusable XML Web Services, encapsulating processes and making them available to applications on any platform.

- Pros:
- offers multiple language support.
 - supports the Microsoft .NET Framework, which provides the common language runtime and unified programming classes
 - include MSDN Library, which contains all the documentation for these development tools
 - intuitive tools for working with XML and XSD files
 - cross-language, cross-process, and remote debugging
 - statement completion and syntax notification for HTML and XML tags

Cons:

- only available on Windows.

2.6.2 Microsoft Visual Basic Studio 6.0

The Microsoft Visual Studio 6.0 is a comprehensive suite of industry-leading development tools for software applications for the Windows operating system, including client/server, multi-tier and Web-based solutions. Visual Studio 6.0 Professional Edition features Visual Basic 6.0, the Visual C++ 6.0 development system, the Visual J++ 6.0 development system for Java, the Visual InterDev 6.0 for Web development system, the Visual FoxPro 6.0 for database development system, and the Microsoft Developer Network (MSDN) Library.

The Microsoft Visual Studio 6.0 is the complete suite for rapidly building scalable enterprise solutions. All components of this tool are designed to help developers rapidly build scalable distributed applications that can be easily integrated with existing enterprise systems and applications.

2.7 Unicast, Broadcast, and Multicast Reference [7]

2.7.1 Unicast

Unicast is the term used to describe communication where a chunk of information is from one point to another point. This is called one-to-one communication that there is just one sender and one receiver. Unicast transmission is a packet sent from a single source to a specific destination, is still the predominant form of transmission on LANs and within the Internet. So, sometimes unicast wastes bandwidth by sending multiple copies of the data to only a portion of the clients on the network.

All LANs support the unicast transfer mode such as Ethernet and IP network. The most users are familiar with the standard unicast applications such as HTTP, FTP, SMTP and Telnet which employ the TCP transport protocol.

2.7.2 Broadcast

Broadcast is the term used to describe communication where a chunk of information is sent from one point to all other point such as one sender sending frames simultaneously to all hosts on the network. Broadcast is waste the bandwidth in the network by sending the data to the whole network whether or not the data is wanted by users.

In the Ethernet, broadcast transmission is supported on most LANs. Broadcast normally used to send the same message to all users on the network and use Address Resolution Protocol (ARP) to send an address resolution query to all users on the network. A form of broadcast is supported by IP address in network layer protocols which allows the same packet to be sent to every user in the network. Each user must process the broadcast data whether the broadcast is of interest or not.

2.7.3 Multicast

Multicast is the term used to describe communication where a chunk of information is sent from one or more point to a set of other points. In this case there is may be one or more senders, and the information is distributed to a set of receivers.

Video server is one of example which use the multicast for sending out networked TV channels. Simultaneous delivery of high quality video to each of the large number of delivery platforms will exhaust the capability of even a high bandwidth network with a

powerful video clip server. This is for applications which required sustained high bandwidth.

Multicasting is the networking technique of delivering the same packet simultaneously to clients. IP multicast provides dynamic many-to-many connectivity between senders and receivers. Multicast applications must use the UDP transfer protocol because TCP only supports the unicast mode.

The majority of installed LANs such as Ethernet are able to support the multicast transmission mode. Shared LANs inherently support multicast which using hubs and repeaters, since all packets reach all NIC connected to the LAN. The earliest LAN network interface cards had no specific support for multicast and introduced a big performance penalty by forcing the adaptor to receive all packets in promiscuous mode and perform software filtering to remove all unwanted packets. Most modern network interface cards implement a set of multicast filters, relieving the host of the burden of performing excessive software filtering.

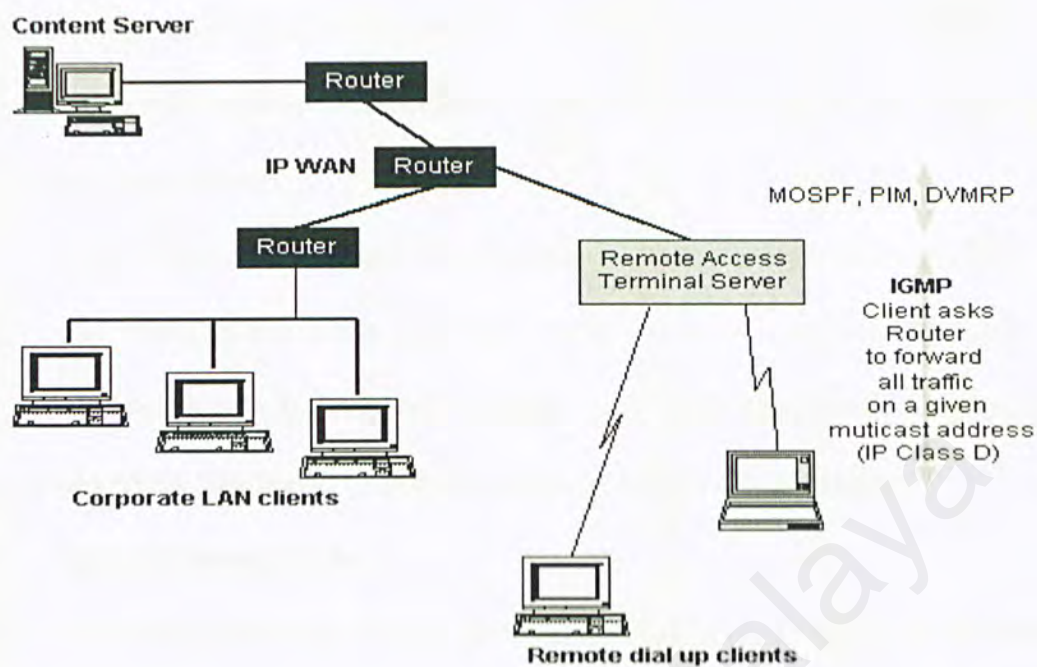


Figure 2-20: IP Multicasting Works

2.8 Database Management Systems

A database management system (DBMS), sometime just called a database manager, is a program that lets one or more computer users create and access data in the database. DBMS available today can be group into four different models: Hierarchical, Network, Relational and Object oriented.

➤ Hierarchical Model

Data is represented in a tree structure that originates from a root. Each class of data is located at different levels along a particular branch which stems from the roots.

➤ Network Model

Conceptually describes database in which many-to-many relationship exists.

Data is represented as records linked together, forming intersecting sets of data.

➤ **Relational Model**

Data is represented in logical mathematical sets in a tabular structure. Each data field becomes a column and each record becomes a row in the table. The relational system is completely flexible in describing the relationships between the various data items. Its primary goal is to preserve data integrity.

➤ **Object Oriented Model**

Document is stored as a single object in stead of several tables, and would have properties whose state would be maintained. Object oriented model often provided object-oriented concepts, for example inheritance and encapsulation.

2.8.1 Microsoft SQL Server 2000

Microsoft SQL Server is Microsoft's leading Windows database and data-warehousing package. Microsoft SQL Server 2000 is the most robust database for the windows family, the Relational Database Management System (DBMS) of choice for a wide range of corporate customers and Independent Software Vendors (ISVs) building business application. It extends the performance, quality, ease-to-use and reliability of Microsoft SQL Server 7.0.

SQL Server is a traditional database management system and is designed to understand Structure Query Language (SQL). It can manage a large amount of data in a multi-user distributed client-server environment. SQL Server is highly integrated with the Windows

NT operating system to make use of native operating system thread that can results in better performance and stability of SQL Server.

Microsoft SQL Server supports a set of features that result in the following benefits:

3 Fully Web Enable

3.7 Rich integrated XML support

3.8 User-friendly queries

3.9 Powerful search capability

3.10 Web enable analysis

3.11 Web access to data

3.12 Ensure the application are secure at any network environment

3.12.2 Highly Scalable and Reliable

3.12.2.1 High availability

3.12.2.2 Scalability

3.12.2.3 Large memory support

3.12.2.4 Gain performance from existing hardware by storing query results and reducing

➤ Faster Time to Market

- Simplified Database Administration
- Improved Developer Productivity
- Analysis OLSAP services
- OLAP Flexibility
- Ease of installation, deployment and use

2.8.2 Oracle

Oracle is the world's leading vendor of database software and has the distinction of being the first company to create and sell a commercial RDBMS that used SQL. Oracle databases run on computers ranging from mainframes to personal computers. It is designed to support and leverage the capabilities of the internet. It does not only provide extensive functionality to support business running on the World Wide Web but also the traditional mission critical OLTP and data warehousing applications.

The most powerful features of oracle are its portability and scalability. Oracle is capable of handling all types of information for all types of applications. It also provides outstanding, across-the-board, and transparent scalability from low-end uniprocessor to high-end symmetric multi-processor system and multi-node clustered configurations.

Oracle is an open system. It adheres to industry-accepted standards for data access language, user interface, operating system and network communication protocols. The key features of Oracle are:

- Transaction Processing
 - Data partitioning
 - Material views
 - Query optimization
- Reliability
 - Complete data protection

- Online data evolution
- Self service error correction
- Security
 - Single sign-on oracle advance security
 - Selective data encryption
 - Secure data sharing
- Data warehousing
 - Data mining
 - Data warehousing Extraction-Transformation-Loading (ETL)
- Management tools
 - Intelligent self managing and tuning
 - Manage the entire stack
 - Pin point diagnostics
 - Database resource management
- Data integration and massaging
 - Message queuing
 - Database replication
 - Legacy database data gateways.

2.8.3 Microsoft Access 2003

Microsoft Access 2003 is a relational database management system created by Microsoft for small office or home user to store data in relational format. MS Access offers an

easy-to-use database for managing and sharing data, regardless of creating a stand-alone desktop database for personal use, departmental use or for an entire organization. Other than the traditional broad range of easy data management tools, Microsoft Access also adds increased integration with the web for easier sharing of data across a variety of platform and user levels and additional ease-of-use enhancements to assist with personal productivity.

Microsoft Access 2003 can be used as a database in a client/server or an n-tier architectural system, with data access interface paradigm such as Remote Data Object (RDO) and Data Access Object (DAO). It enables sharing of database among the co-worker over the internet and taking advantages of automated, pre-packaged solutions to quickly create databases. Users can also search and retrieve the information quickly through MS Access. Data in Microsoft Access can be migrated to the MS SQL Server.

Below are some of the benefits for Microsoft Access:

- Information easy to find and use

MS Access offer an easy-to-use tool for easily finding information that provides consistency and integration with the other applications in the office suite.

- Web-enabled information sharing

MS Access allows easily sharing information via the corporate internet and the ability to easily host a database within the browser. This is the combination of the power of a desktop database and the power of the web.

- Powerful solutions tools for managing information

Power users and developers can create solution that combine the easy-to-use of the Access interface (client) with the scalability and reliability of SQL server

2.8.4 MySQL

MySQL is an open source relational database management system (RDBMS) for Linux, Unix and Windows platform that uses Structured Query Language (SQL). It is the most popular language for adding, accessing, and processing data in a database. The MySQL database management system contains a large amount of functionality and power, and is noted mainly for its speed, reliability and flexibility. However, it works best when managing content and not executing transactions.

MySQL provides application programs interfaces (APIs) for C, C++, Eiffel, Java, Perl, PHP, Python and Tcl. It also allows for many column types and offers full operator and function support in the SELECT and WHERE parts of queries. A complex set of databases and tables can be developed just using a simple set of command for inserting, retrieving, deleting and updating data.

MySQL also provide or plan to provide features that include a table definition file format, enhanced replication, more function for a full-text search, fail-safe replication, a port of MySQL to BeOS and an option to periodically flush key pages for tables with delayed key. Below are some key features for MySQL :

- MySQL is free download and comes complete with all tools that need to get started.
- MySQL can be accessed and manipulated from many popular programming languages.
- MySQL is completely optimized for both Unix and Win32 platform. It uses in-memory hash tables, thread-based memory allocation and kernel threads that are

capable of utilizing multiple processors, and highly optimized individual pre-compiled class libraries.

- MySQL contains built-in support for common field type.
- MySQL supports a subset of advanced querying and grouping functions
- MySQL allows per-server password allocation
- MySQL supports a variety of connection methods, for example unit sockets and TCP/IP sockets.

2.9 Chapter Summary

This chapter is mainly focus on the research of the problem encountered before project can be done. Through literature review on various aspects, many ideas have been gained to develop the network security tool that is Packet Sniffing. Research on this tools concepts and strategies is the stepping stone to have a better understanding on the requirements of this project. By reviewing the existing system that match the proposed system, the strong features gave some ideas to enhance the proposed system. Review on tools and methods were also carried out. Development tools and other useful technology need to be considered in order to develop a quality and useful system.

Chapter Three is the research on the methodology and techniques that used to gain information.

Chapter 3

System Requirements Analysis

Chapter 3 - System Requirements Analysis

3.1 Methodology

Methodology is early phase in development system. Methodology is the study of methods or a body of method to create a system with a series of procedures, techniques, tools and documentation aids. These can help the software developers to speed up the software development process. A methodology consists of a set of phases that in turn may consist of sub-phases. These phases are important in guiding the developers to the choice of techniques at various stages in the project. Additionally, a methodology helps the developers to plan, manage, control and evaluate information system project.

The software development process can be defined as system life cycle model and every system development process model includes system requirements (user, needs, resource) as input and a finished product as output.

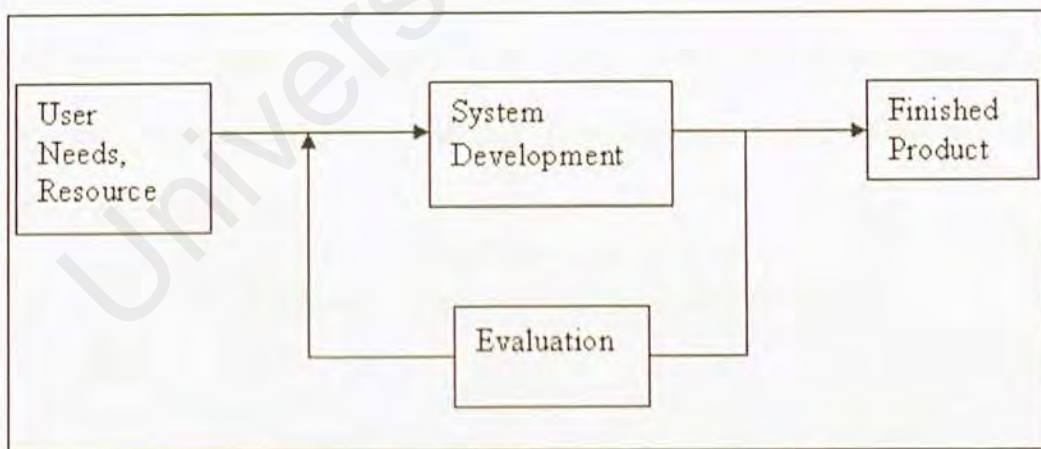


Figure 3-1 : System Development Process Model

There are several types of development model in software engineering:

1. Waterfall Model with prototyping
2. V- Model

- 3. Spiral Model
- 4. Iterative and Incremental Model

3.1.1 Conclusion on Development Methodology

The methodology used in development of network sniffing tools is modified waterfall model with prototyping. Waterfall model with prototyping is chosen because the strength of both the waterfall model and prototyping model can be combined in a single project in which waterfall model support interactive design while the prototyping model helps to gain user requirement. Beside that, this model also easy to use, systematic, scopes of project well understand and project risks are considered to be low.

3.1.2 Waterfall Model with prototyping

Waterfall Model

This methodology is so called waterfall because each phase flows naturally into the next phase like water over a series of fall. Each phase in the waterfall method should be completed before moving on to the next. There are five stages in Waterfall model as illustrated in Table 0-1.

Table 0-1: Five Stages in Waterfall Model

Reference [8]

Requirements Analysis and Definition	What is the problem? Functions to be developed Possible future extensions Amount and kind of documentation Performance characteristics for functions
--------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------

	Feasibility Study Technical Social Economic Political
System and Software Design	What is the solution? A system model which solves the problem for the user.
Implementation and Unit Testing	How is the solution constructed? A transformation of the design into an executable form.
Integration and System Testing	Is the problem solved? Determining if the solution as constructed meets the requirements.
Operation and Maintenance	Are enhancements/changes needed? Corrective - repair errors Adaptive - modify software to adapt to changes in environment Perfective - providing new functionality for new requirements Preventive - improving the system's maintainability

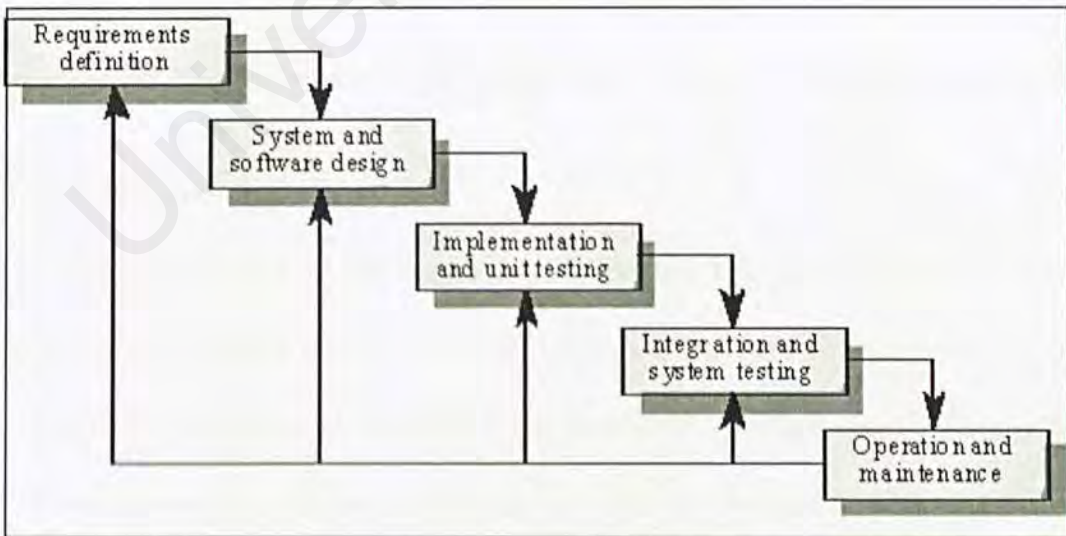


Figure 3-2: Waterfall Model

Prototype

Prototype is partially developed product that enables customers and developers to examine some aspect of the proposed system and decide if it is suitable for the final product. It also can be said as a smaller-scale, representative or working model of a proposed design for an information system.

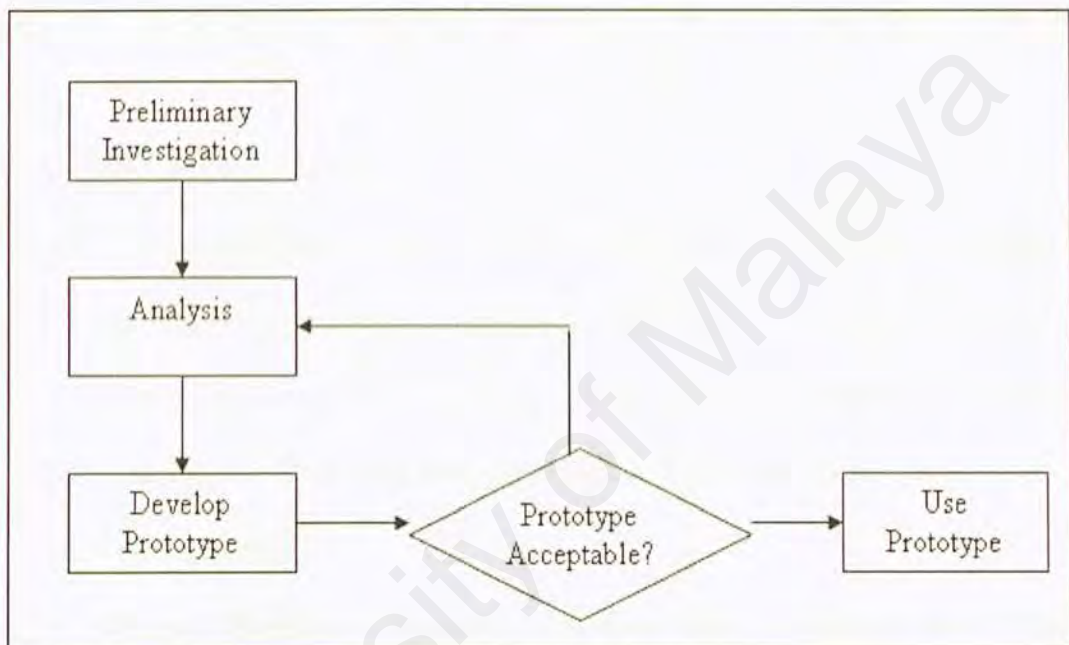


Figure 0-3: Prototype Model

Prototyping can be combined with the waterfall methodology to make it more accurate.

Waterfall Model with prototyping

When prototyping is implemented into the waterfall model, it can solve a lot of problems which cause in traditional model. That is, a software development will run a lots of iteration processes as listed from the traditional waterfall model. The waterfall model with prototyping has some advantages to improve the quality of the software life cycle process. Those advantages are:

- 1 Allows all or part of the system to be constructed quickly to understand or clarify the requirement
- 2 Understands the feasibility of a design or approach
- 3 Reduces risk and uncertainty

Waterfall Model with prototyping has eight stages in this software development process. Each development stage should be finished before the next stage starts. The eight stages are:

i. Requirement Analysis

Understanding and determining users need by having brainstorming, eliciting and analyzing user requirements by having interview, survey or questionnaire session, collecting and specifying all the user requirements and validating requirements. This stage also can represent the 'What' stage.

ii. System Design

Diagram the system functional by analysis case studies on current system. Determine and specify hardware and software architecture to develop the system and verifying system design. This stage represents the 'How' stage.

iii. Program Design

Determining and specifying program design and database design and verifying program design.

iv. Coding

Involve programming, personal planning, tool acquisition, database development, component level documentation and programming management.

v. Unit and Integration Testing

Test units separately and integrate the tested units.

vi. System Testing

Combine the integrated units into system and testing on the system.

Specifying, reviewing and updating of the system test and validating of system.

vii. Acceptance Testing

Finalize testing on the system and deliver the system.

viii. Operation and Maintenance

Control and maintain system and revalidating of system.

The system has to be validated and verified during the stage of system testing.

The validation is to determine that Packet Sniffing Tools has implemented all the requirements in the specification. The verification is to make sure that the functions in the Packet Sniffing Tools can work correctly and to check the quality of implementation.

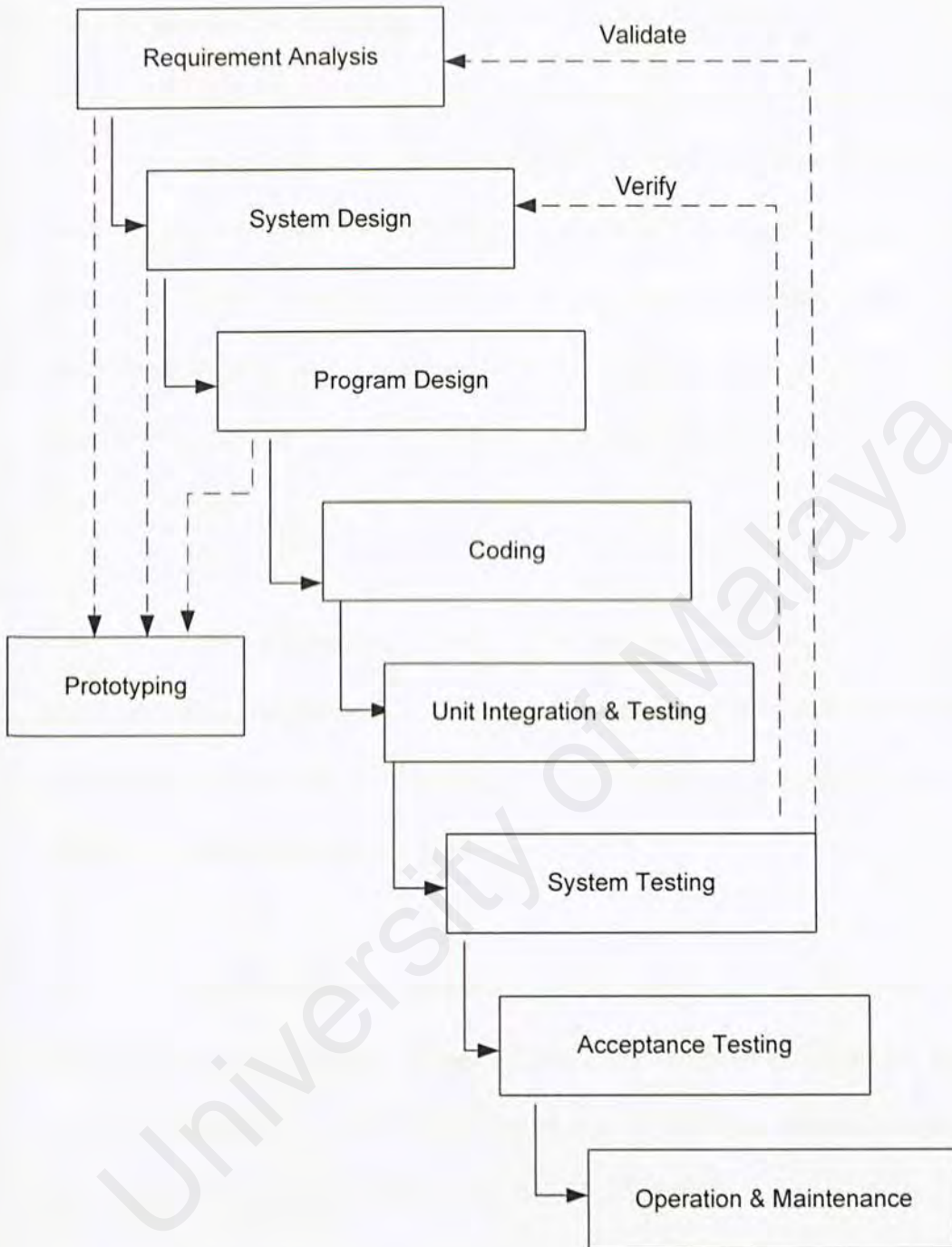


Figure 3-4: Waterfall Model with Prototyping

3.1.3 Justification of Methodology

The modified waterfall model with prototyping was chosen for the following reasons:

- The development methodology is simple and easy to understand. It helps the developers to lay out what they need to do easily. Therefore in the process, they no need to burden themselves with the upcoming stage. In addition, one can has a better understanding and clearer guideline on what he or she should do during the development process. Moreover, it is easier to associate and identify each milestone with its deliverer.
- With prototyping, users' participations are involved where user requirements can be gathered. It helps the developers to ensure that the requirements are feasible and practical. Furthermore, it helps to access alternative design strategies and decide which is best for the project.
- This development methodology will ensure that the developer building the right system according to the specification and verification checks the quality of the implementation. It also enables developers to develop more accurate system according to the user's discretion. This would help the developers to learn about the system and gain better understanding of the entire system.
- The arrows between adjacent stages are bi-directional. Bi-directional means that there is feedback between stages. If a problem is discovered in one stage, developers can return to the previous stages so that suitable corrective action can be

taken. There is a cascading effect where developers can go back further and further up the waterfall until the problem can be corrected properly

➤ As the methodology is simple and easy to understand, it will be easier to present or explain to the users, especially those who are not familiar with software development. Therefore, developers can give the users or customers a clearer view on what is going on.

3.2 Information Gathering Methods

Information gathering methods are essential step for all system analysis. It is a classical set of techniques used to collect information about system problems, opportunities, solution, requirement and priorities. Information gathering methods that usually used are internet surfing, reading materials, and discussion.

➤ Internet Research

I have surf around the net for gain the information about Packet Sniffing architecture and download the demo version which available on the net. With development of each search engine such Google, yahoo and relevant information site can be view with a click away. Lots of information can be collected for literature review, especially information about analysis case studies.

➤ Reading Materials

A lot of published literatures have been read in order to gather information of the users' need, system development needs and technical issues of the proposed system. All these can be categorized into printed material such as book and journal and non-printed material such as electronic document. Ideas are managed to get from books,

magazines and journal through reading. These ideas can be implemented in the proposed system.

➤ Discussion

Discussions were made from time to time with supervisor to get helps and advices on the project and much of the details and idea on how to develop the system.

3.3 Chapter Summary

This chapter discuss about the methodology of the system development. Waterfall model with prototyping methodology has been chosen as the framework to develop the system. Various techniques of information gathering such as online surfing, reading materials, observation, discussion and questionnaire have been practiced. This chapter also includes the conclusion on tools and technology.

The next chapter will focus on the system requirement including functional requirement, non-functional requirement, and software and hardware requirement

Chapter 4

System Analysis

Chapter 4 - System Analysis

System analysis is an essential and important phase in software life cycle. The main purpose of this phase is to determine clearly of all necessary requirement before proceeding into subsequent phase. In shorts, system design is the process of gathering and interpreting facts, diagnosing problems, and using the information to recommend improvement to the system.

4.1 System Requirement Analysis

A software specification definition is an abstract description of the services, which the system should provide, and the constraints under which the system must operate. Functional requirement and non-functional requirement are two types of system requirement analysis for the system.

4.1.1 Functional Requirement

A functional requirement is a function or service that must be included in the system to satisfy the users need. Functional requirement are statements of services the system should provide, how the system should react to particular inputs and how the system should behave in particular situation.

- Capturing the packets module
- Filtering data module
- Analyze the packets module
- Generate statistic graph module
- Present the result module

4.1.1.1 Capturing the packets module

This module is built for the packet capturing in the promiscuous mode that is copy the packet which users have to sent it out or receive it from other place through the network wire. This module has user-friendly interface and capturing option like stop capture when reach a number of packet or bytes will be issued in this module. It should have the “start”, “stop” and “continue” button to handle the capturing process. Besides the “save” function should exist to enable the saving of packets captured into a notepad file.

4.1.1.2 Filtering data module

Before process capture the packets is done, users can filter and categorize the information about the packets are captured such as packets from certain destination/source or packets that match the protocols like TCP, UDP, ARP and ICMP. This purpose of this process is we can left out the packets that are not desired. Operator “OR”, “AND” “NOT” “ALL” will be used in the filter module.

4.1.1.3 Analyze the packets module

Each packet captured will be analyzed to get the value in the header field like Port number, length, IP, checksums. Different type of packet have different header, for example a TCP packet has TCP header while UDP packet has UDP header with different fields. Thus there will be different ways to analyze the packets

4.1.1.4 Generate statistic graph module

This module will analyze all the packets and generates the IP traffic statistic which shows the total number of bytes and packets sent by one station to another station. This will show to user about the size of packets generated by both stations in a single connection. Another statistic generated is shows the total packets in a ranges of packets size.

4.1.1.5 Present the result module

This module display the packets captured in two different ways. First shows the packets in a list view with source/destination IP address, protocols etc. The second is the display of the raw packet in the rich-text box when the packet in list-view is selected. A click on the certain root or child in the tree will highlight the match portion of packet in the rich-text box.

4.1.2 Non-functional requirement

A non-functional Requirement is a description of features, characteristic and attributes of the system. It also decrypts the constraints that may limit the boundaries of the proposed system. In order to ensure the quality of the system produced, the role of non-functional requirements is as important as functional requirements. The following are the non-functional requirement that must be fulfilled.

➤ Reliability

A system is said to be reliable if a system performs its functions with required precision and accuracy. It is also important for the system to not to produce dangerous and costly failures to the viewers when it is used.

➤ **Efficiency**

Undeniable, efficiency is the main key for implementing the new meetings management system. Efficiency is understood as the ability of a process procedure to be called or accessed unlimitedly to produce similar performance outcomes at an acceptable or credible speed. Efficiency is measured base on response time performance, report generation speed and graphic generation speed.

➤ **Simplicity and User friendly**

The system by itself should be made simple. Users will not need to know so much about how the system goes. They should only be made to understand what they need to do when they want to generate a function. Moreover, usage of suitable and meaningful caption will help the user to consume the system easier. Therefore, an attractive and easily understand user interface is needed.

➤ **Manageability**

Maintenance should be easy to done. The same goes to evolutionary. Furthermore, it could be adapt to new demand and requirement or enhanced in the future. New records should be easy to be implemented into the system with this non-functional requirement.

➤ **Expandability**

The system should be able to be extended to accommodate more functionality in the future.

4.2 Conclusion on Tools and Technology

After reviewing and analyzing on all the tools and technologies, the most appropriate tools and technologies are chosen to develop this project.

Selected Programming Language

For this project, C# is chosen as the main programming language to develop this system.

The advantages of this software are:

- It is based on GUI (graphical user interface)
- safer, simpler, and more productive
- it is .NET Framework (can access thousand of classes that you won't have re-create)
- it is also object-oriented language and can designed to give the optimum blend of simplicity
- compatible with window platform
- Module oriental, easy error discovery, can focus on the module that cause error only.

Selected Application Platform

After reviewing on Windows XP, Unix, RedHat Linux and Macintosh, **Windows XP** is chosen to be the project application platform as it support all the tools and technologies that will be used in this project.

Advantages of Window XP

- provides better integration of Windows 9x and Window NT that did Windows 2000
- uses slightly more total memory for the OS to add features than does Windows 2000
- offers significant GUI enhancements
- built-in support for compresses files
- advanced file sorting options
- more stable and improve troubleshooting tools

Selected Database Management System

Microsoft Access 2003 is the suitable choice for the development of this system as it works well with databases of any size. Microsoft Access can be use as database at the server or multi-system. it provides a user friendly interface that helps the users to create a database in the easiest way.

Microsoft Access 2003 is a easy tool for system developer to add or to delete data from the database because this will not involve the programming part. The changes can do on the database itself without disturb the programming part.

The advantages to choose the Microsoft Access 2003 to create database for this system are:

- To create a record using Microsoft Access is easier and faster compare to coding.
- Data type is easy to specify.
- Relation between records can easily manage and create.
- Error easy to be detected and easy to make the correction
- Compatible with the system

Development Model	Waterfall Model with Prototyping
Application Platform	Windows XP
Programming Language	C#
Authoring Tool	Microsoft Visual Studio .NET 2003
Database Management System	Microsoft Access 2003

Table 4-1: Tools and Technology chosen

4.3 Hardware and Software Requirements

Hardware and software requirements describe the constraints on computers and peripheral equipments. Hardware and software requirements need to be decided to determine the performance requirement’s feasibility. Both hardware and software requirements are divided into runtime and development requirements.

	Run Time	Development
Hardware	<ul style="list-style-type: none">• 233 MHz Pentium / higher microprocessor / or	<ul style="list-style-type: none">• Pentium IV 1.6 Gigabyte• Random Access

Requirements	equivalent <ul style="list-style-type: none"> • Random Access Memory : 64 MB and above (128MB recommended) • Hard disk : 2.5 GB and above • Standard input and output • Others standard computer peripherals 	Memory : 256 MB <ul style="list-style-type: none"> • Hard disk : 40GB • Display : VGC display card • Others standard computer peripherals
Software Requirements	<ul style="list-style-type: none"> • Windows XP 	<ul style="list-style-type: none"> • Windows XP • Microsoft Access 2003 • Microsoft Visual Studio .NET 2003

Table 4-2: Hardware and Software Requirements

4.4 Chapter Summary

This chapter discusses about the evaluation of the requirement analysis, which consists of functional requirements and non-functional requirements. The functional requirements describe the functionality and the services that the system is expected to provide while the nonfunctional requirements will affect the overall quality of the system. The summary of hardware and software is presented in this chapter. It is divided into run-time requirements and development

Chapter 5

System Design

University of Malaya

Chapter 5 - System Design

System design is considered as an important part of the system development process.

System design is a process to arrange the structure of the system in details to realize the system in purpose. The entire requirements of the systems are translate into system characteristics. In previous chapter, the requirements for the system are regarding to the analysis that had been discussed. In shorts, system design is a process to convert the conceptual ideas from requirement specification in system analysis into more technical specification.

A system outline that contain the design of user interface, database design, system functionality and other related details how the system can work and be able to implement for the further development. System design must be very details and errorless for the each phases.

Under this chapter, the system design will be discuss in the following categories:

- Structure Chart
- System Functionality Design
- User Interface Design

5.1 Structure Chart

The main purpose of structure chart is shows all the relation between modules in this security tool. It also identifies the activities that structure of this system. Structure chart is used to depict high-level abstraction of a specified system and describe the interaction between independent modules. In structure chart, the major functions form the initial component part which can be split into detailed sub-components.

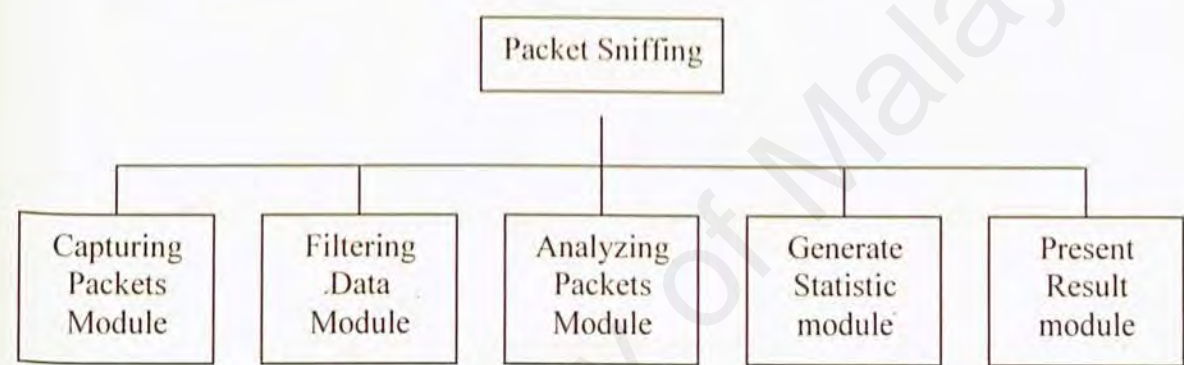


Figure 5-1: Structure chart for Packet Sniffing

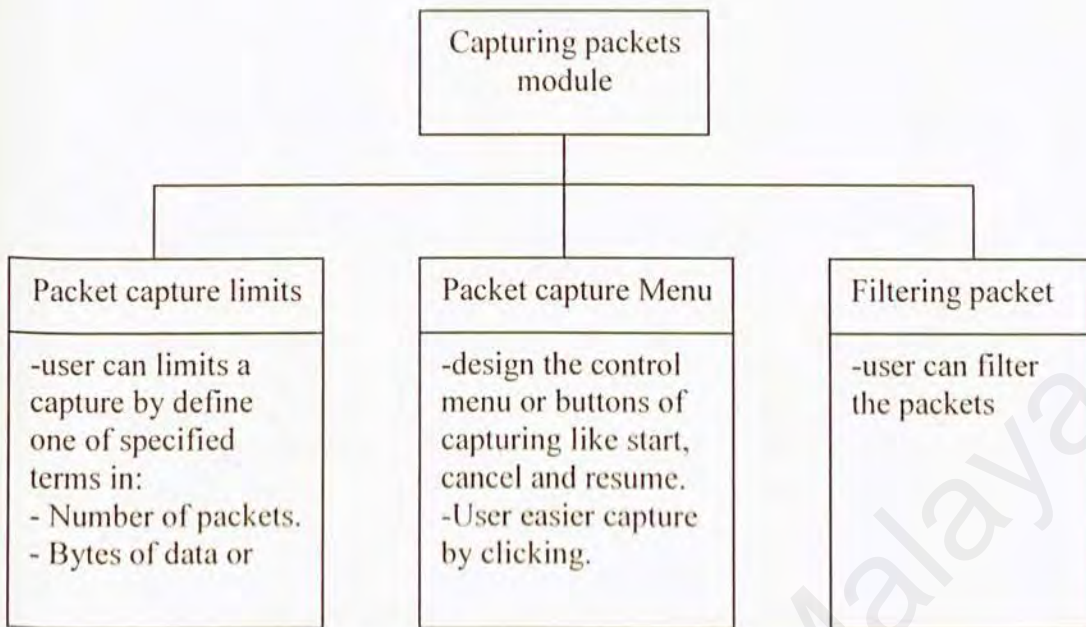


Figure 5-2: Structure chart for Capturing packets module

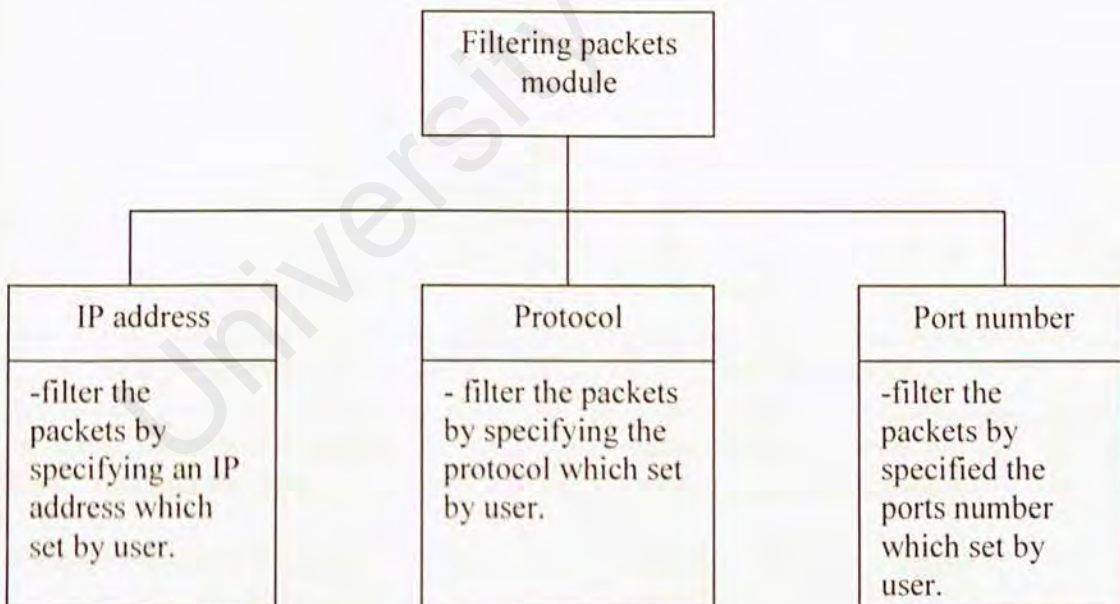


Figure 5-3: Structure chart for Filtering packets module

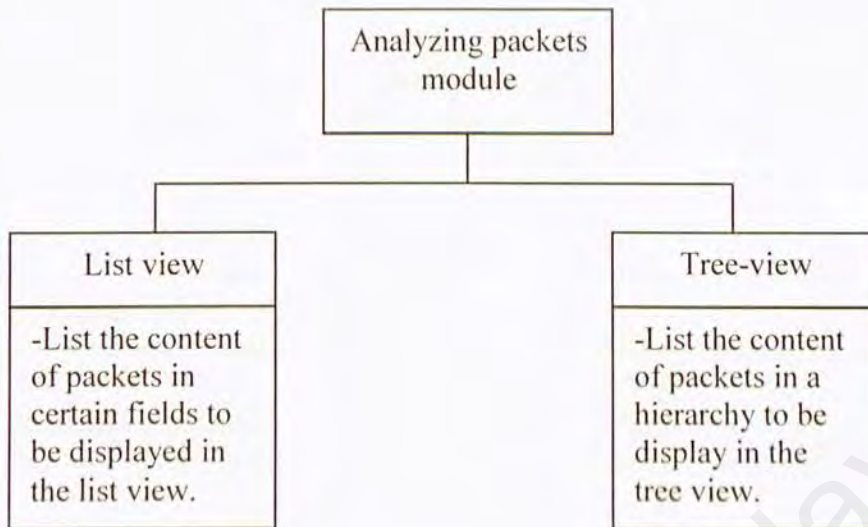


Figure 5-4: Structure chart for Analyzing packets module

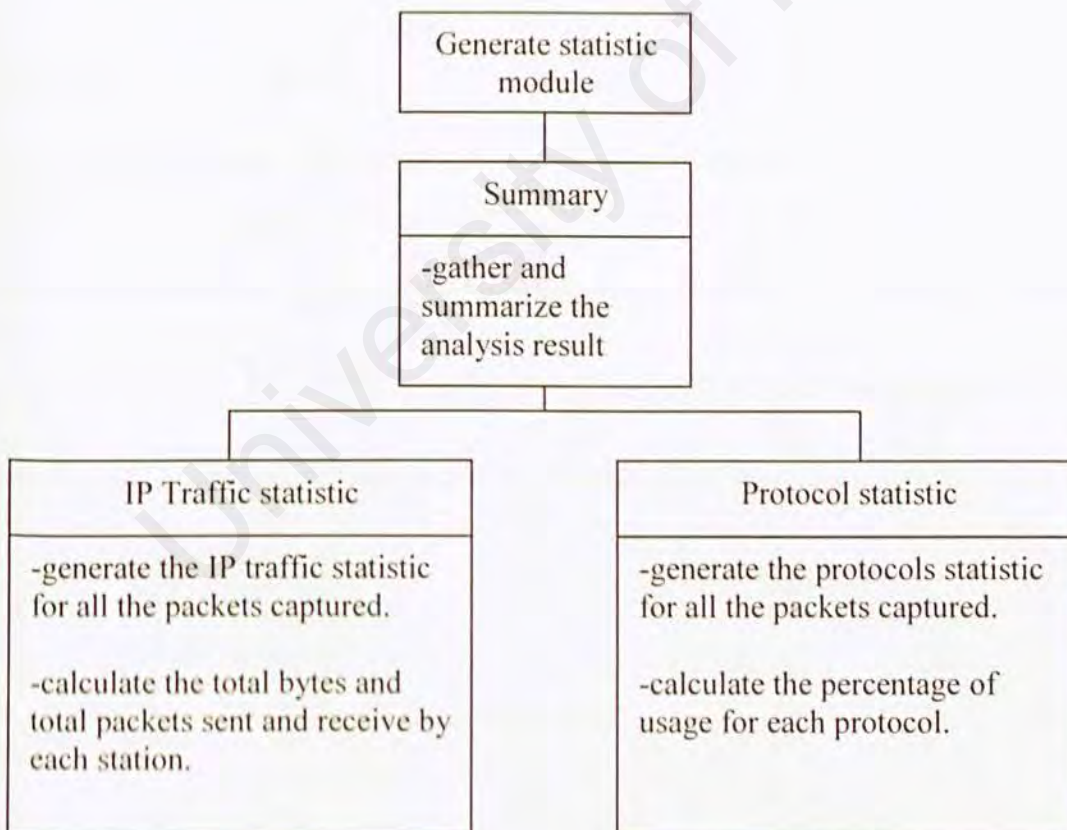


Figure 5-5: Structure chart for Generate statistic module

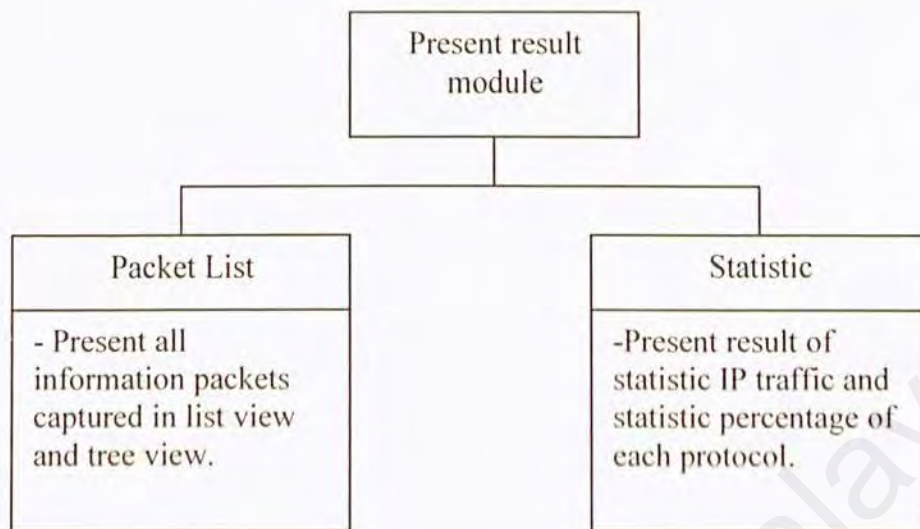





Figure 5-6: Structure chart for Present result module

5.2 System Flow Chart

System flow chart represents a complete algorithm in Packet Sniffing System.

Symbols	Attributes
	Begin and End process
	Process
	Decision


	Flow of process
-----------------------------------------------------------------------------------	-----------------

Table 5-1: Flow Chart Symbol

University of Malaya

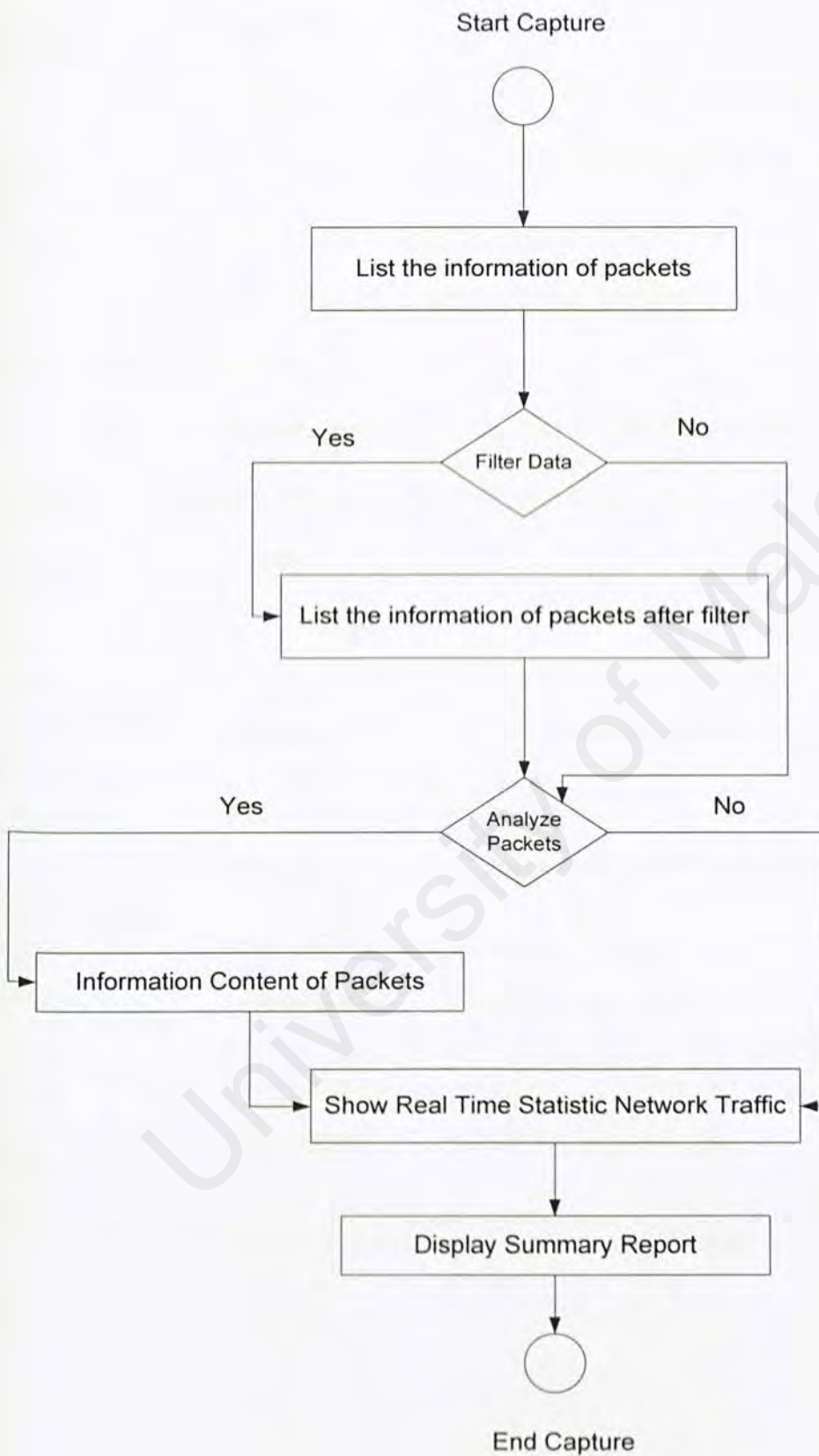


Figure 5-7: Flow Chart of Packet Sniffing System

5.3 Data Flow Diagram

Data Flow Diagram (DFD) is a technique used to present the graphical characterization of the data process and flows in the system. The DFD gives an overview of system inputs and outputs, processes and flows of data through each process. The DFD is a simple to use and easy to learn and represents the flows of the data through each process in a right sequence.

Data Flow Diagram usually is made after a Context Diagram has been created. The Context Diagram functions as the basic of a Data Flow Diagram. The following is the basic symbols of a DFD.





Symbol	Attribute	Definition
	Process	Transformation of data into another data
	Entity	Source and destination of data
	Flow of data	Data on the move
	Data source	Data in static storage

Table 5-2: Data Flow Diagrams (DFD) objects

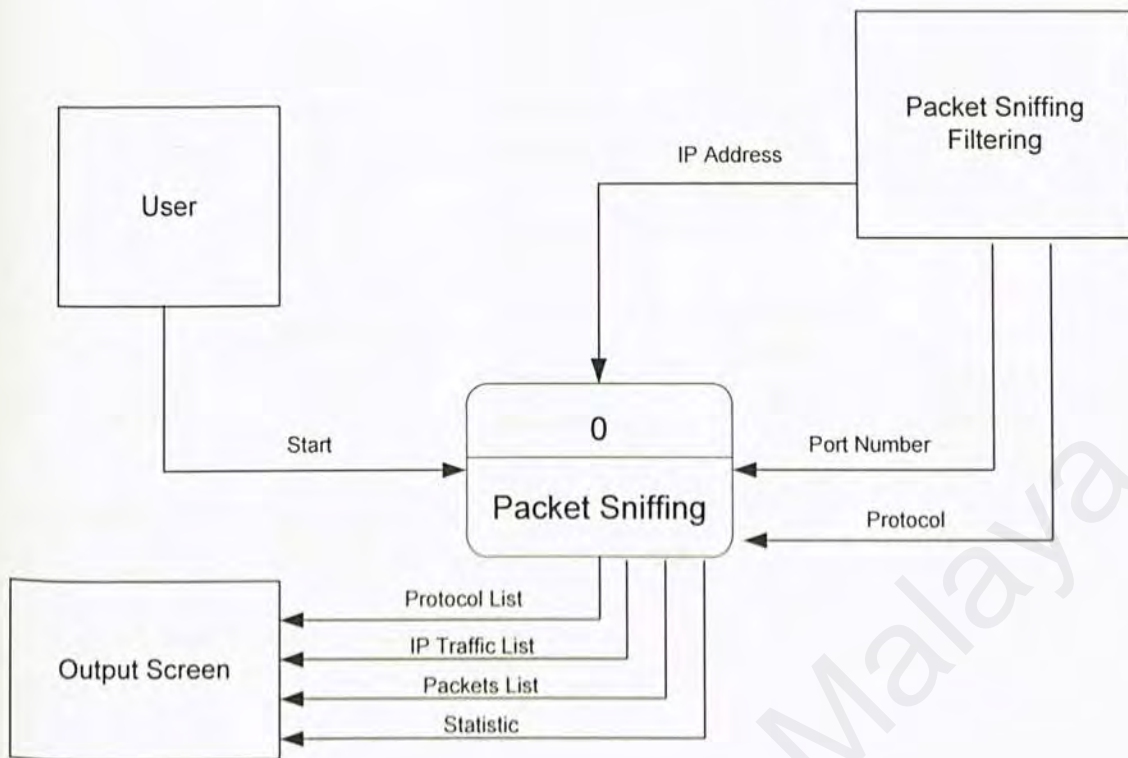


Figure 5-8: Context Diagram of Packet Sniffing

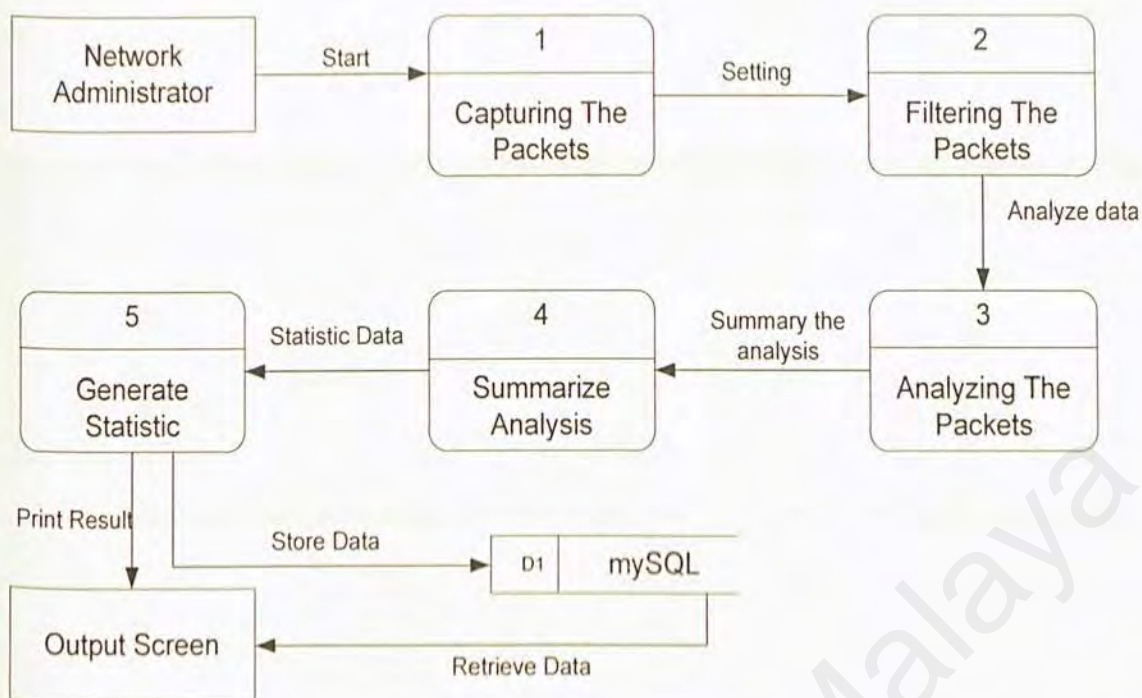


Figure 5-9: Diagram 0 of Packet Sniffing

5.3 User Interface Design

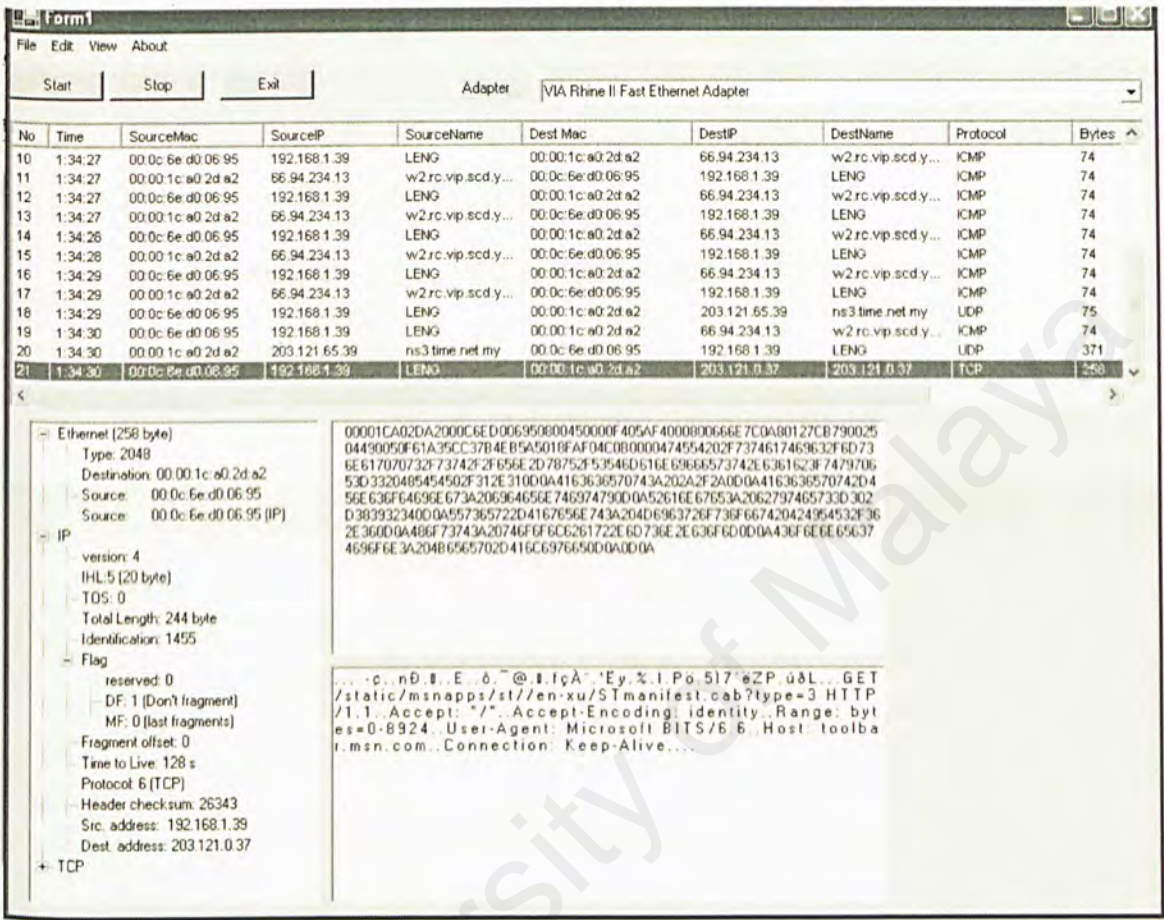


Figure 5-10: Main interface of Packet Sniffing

At the top of main interface is the menu bar which has five menu items.

- File
 - Save As – save the result
 - Save Selected Packet As – save the result in directory chosen
 - Open – Open the result which save to the disk
 - Exit - close the whole program

- Edit
 - Filter Manager - load the configuration filter menu and configure it
 - View Filtering – View what the filtering is configured.
- View
 - IP statistic - display the IP statistic
 - Protocol Statistic - display the Protocol statistic
- About – shows the user manual

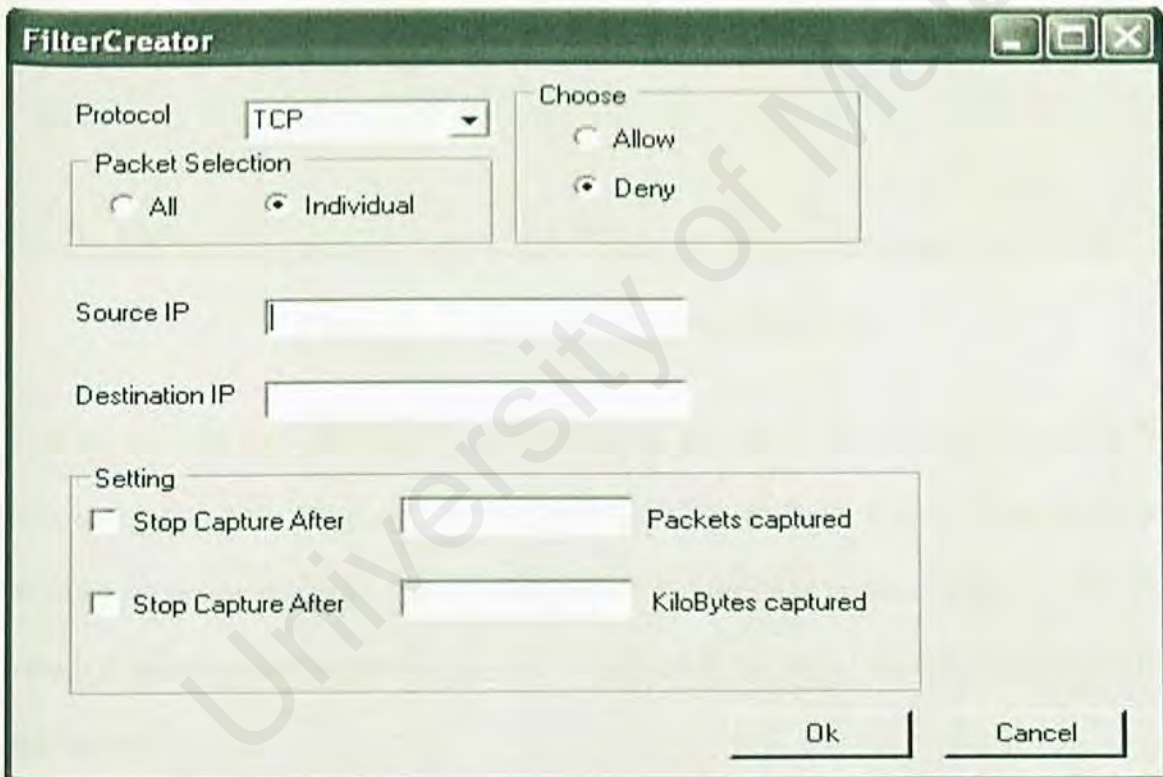


Figure 5-11: Interface of Packet Sniffing Configuration Filter menu

The configuration filter interface provides three types filter choice that are filter Protocol, filter IP address and Setting with each can be selected by a click on the checkbox. In the filter IP address, users can choose the range of IP address from start and finish then will

display in the text box. In the setting group, we can set how many packets are wanted to capture and stop the system and the size of packets we want to capture from the network.

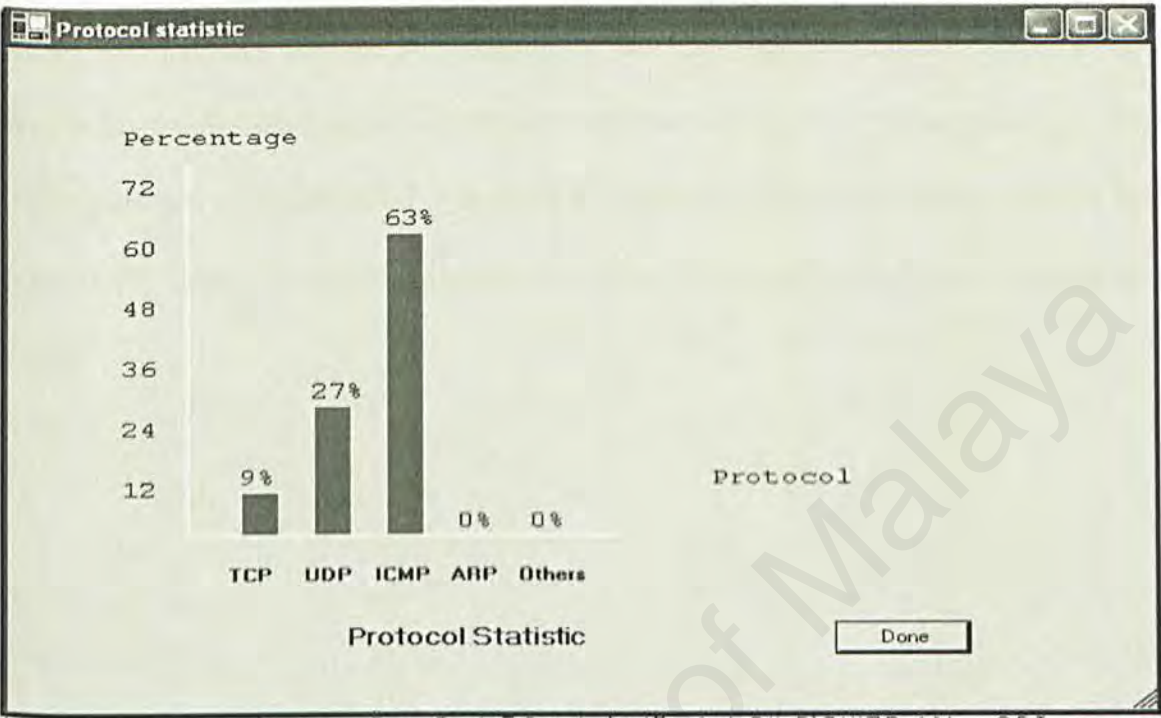


Figure 5-12: Interface of Protocol Statistic

This page shows the statistic of the protocols in a bar chart. The value for Y-axis is the percentage (%) and the X-axis is the Protocols. The scale in Y-axis is dynamic that accordingly to the maximum value of the statistic but the label on the X-axis is static. The value of each protocol in percentage will be labeled at the top of the bar respectively for ease of view.

5.5 Chapter Summary

Chapter Five present the system architecture, system functionality design, database design and interface design. This chapter is show the system analysis translated into system design. System functionality design explains the process in packet sniffing system where structure chart and DFD was used to graphically characterize data process and flows in the system. Some proposed interface designs were included for the user interface design.

Chapter 6

System Implementation

University of Malaya

Chapter 6 – System Implementation

During this phase, the design model of packet sniffing system is transformed into workable product. Therefore, system implementation involved the translation of the software representation produces by the design into a computer understandable form. It involves coding of the program by using the appropriate language and coding approach, testing of the system to ensure every function work properly and debugging the code, which will identify and correct bug within program.

6.1 Development Environment

The initial stage of system implementation involves setting up the development environment. Development environment is very important to the development of a system as suitable hardware and software will determine the success of the project.

6.1.1 Hardware Configuration

The following hardware specifications have been used to develop the system:

- Intel Pentium IV 1.6 Ghz
- 256MB DDR RAM
- 40 GB Hard Disk
- 15" color monitor capable of 1024 x 768 resolution
- Standard Input and Output
- Others standard computer peripherals

6.1.2 Software Configuration

There are a lot of software tools, which are used in designing and writing report. Below is a listing of software used throughout the development process as pertaining to the specific usage:

Software	Usage	Description
Microsoft Windows XP	System Development	Operating System
Microsoft Visual Studio.Net	System Development	Authoring Tools
Microsoft Word	System Development	Documentation

Table 6-1: Software Used

6.2 Platform Development

Services and tools installations may be the very first step in order to start the development. Platform development includes setting up the operating system and web server.

6.2.1 Setting Up Operating System

Microsoft Windows XP is used as the operating system for this project. Before the installation begins, the hard disk need to be formatted. This is to ensure a more stable and secure environment. Moreover, it can also prevent the environment being affected by previous settings or configurations. Windows XP’s installation is very easy as it provides user friendly and descriptive interface guide. User just need to follow the step by step instruction appear on the installation’s menu interface.

6.3 Program Implementation

During program development, program is written and user interface is being developed is initialized with data.

6.3.1 Implementation of Capturing Packet Module

To implement the capturing packet module, I install the NDIS driver in my system to capture and send raw packets. This driver is that the ability to work in without the TCP/IP protocols environment. This feature enable the program runs as sniffer that without having IP configuration such as IP address, network domain and subnet mask.

After the NDIS driver installed, we can handle the device driver like a file where we can write to and read from a file. The receiving and sending of packets in this application is done through the ReadFile API and WriteFile API respectively. The Windows Driver Development Kits (DDKs) provides the NDIS driver but in manners that only send packets with own source address and receive packets that destined for our own.

6.3.2 Implementation of Filtering Packet Module

With NDIS driver, the packets are captured and passed to the filtering packet module before display the packets in the list view in the program. Filtering packet module is setting the packets which will be chosen or dropped to display in the list view. The filtering packet module is made in sort of statement where each statement is an object of a class called *FilterItem* that keep in the ArrayList named *FilterList* which is an attribute

of class *FilterManager*. The class *FilterItem* is composed of attributes like protocols, source and destination IP address and etc.

6.3.3 Implementation of Analyzing Packet Module

The packets captured is keep in an array of type byte (byte[]), therefore the size of the packet in bytes is determined by the size of the array. Each element of the array is an 8-bit integer where the maximum value is 255 in decimal or FF in hexadecimal. Each element in the array can be converted in ASCII code by performing casting with data type char. To analyze the packet for more readable format, a function called *ToUInt* is used to convert to unsigned integer.

```
public static uint ToUInt(byte[] datagram, int offset, int length)
{
    uint total = 0;
    int byte_index;
    int bit_offset;
    int bit;
    byte b;

    for ( int i = 0; i < length; i++ )
    {
        bit_offset = (offset+i) % 8;
        byte_index = (offset+i-bit_offset)/8;
        b = datagram[byte_index];
        bit = (int)(b >> (7 - bit_offset));
        bit = bit & 0x0001;

        if ( bit > 0 )
            total += (uint)Math.Pow(2,length-i-1);
    }
    return total;
}
```

Figure 6-1: Function ToUInt

The function receives three arguments. The argument offset represents the index in bits and the argument length represents the length of bits to analyze from the offset.

To handle variable kind of return type we can simply add the casting to the *ToUInt* function.

6.3.4 Implementation of Present Result Module

I define one Struct where it contains a static ArrayList and others public members are fields to be displayed in the ListView. ArrayList is used to store the information from the packets are captured and retrieve the information from ArrayList to displayed in the ListView.

```
public struct ItemCollect
{
    public static ArrayList ItemDisplay = new ArrayList();
    public static ArrayList ItemSave = new ArrayList();
    public string pacSource;
    public string sourceIP;
    public string sourceName;
    public string destIP;
    public string destName;
    public string pacDest;
    public string pacTime;
    public string pacProtocol;
    public int pacNumber;
    public int pacByte;
    public int etherType;
}
```

Figure 6-2: Members in the Struct

6.3.5 Implementation of Statistic Generation Module

In this module, the system will collect the IP address from the hosts and generate statistic in the format graphical graph. I use ArrayList to store the information for connection statistic. I had used some functions such as *GenerateHosts* and *CalculateHosts* to collect all the information from the users in the network and use these information to generate statistic.

```

private void calculateHost()
{
    bool foundsrc =true;
    bool founddes =true;
    if(normalized.Count==0 )
    {
        normalized.Add(source[0]);
        amount.Add(counter[0]);
        sentByte.Add(senderSend[0]);
        recByte.Add(senderRec[0]);
        talkerId++;

        normalized.Add(destination[0]);
        amount.Add(counter[0]);
        sentByte.Add(senderRec[0]);
        recByte.Add(senderSend[0]);
        talkerId++;
    }
    for(int i=1;i<connectionId;i++)
    {
        foundsrc=checkSrcNormalized(i);
        if(foundsrc==false)
        {
            normalized.Add(source[i]);
            sentByte.Add(senderSend[i]);
            recByte.Add(senderRec[i]);
            amount.Add(counter[i]);
            talkerId++;
        }
        founddes=checkDesNormalized(i);
        if(founddes==false)
        {
            normalized.Add(destination[i]);
            sentByte.Add(senderRec[i]);
            recByte.Add(senderSend[i]);
            amount.Add(counter[i]);
            talkerId++;
        }
    }
}

```

Figure 6-3: Function CalculateHost

6.4 System Debugging

System debugging is a necessary part in development process because every application must have bugs or errors which most of the time is logical errors that cause the application giving the unwanted result. Thus debugging is done in order to track and correct program bugs before a system is lunch. When test case is uncovers an error, debugging is the essential to perform for remove the error.

With the Microsoft Visual Studio.Net debugger, you can take advantage of the following capabilities to debug your code:

- The debugger gives you the ability to control aspects of run-state program execution, such as the ability to step through function execution, run to a specified location or run until a designated message is encountered, and so on.
- The kernel debugger captures exceptions and breakpoints, and provides additional functionality that allows controlling break-state behavior.
For example, you can set the next step to execute in source code or disassembly, or you can place the cursor at the next location where you want execution to stop.
- In the IDE, the debugger provides access to kernel information. You can view process, thread, and other debugging information by using debugging windows and menu items.

6.5 Chapter Summary

In this system implementation phase, nearly all the design phases that have been presented and directed toward a final objective that needs to translate representation of system into a form that can be understood by computer.

Chapter Seven presents the various type of system testing that includes the unit testing, integration testing and the system testing.

Chapter 7

System Testing

University of Malaya

Chapter 7 – System Testing

Testing is critical in uncovering logical error and to test the system reliability. The main objective of testing is to uncover different types of errors that exist while executing the system. System testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. However, testing can only show that software defects are present.

In developing a system, testing usually involves several stages. An example of testing process is shown as below:

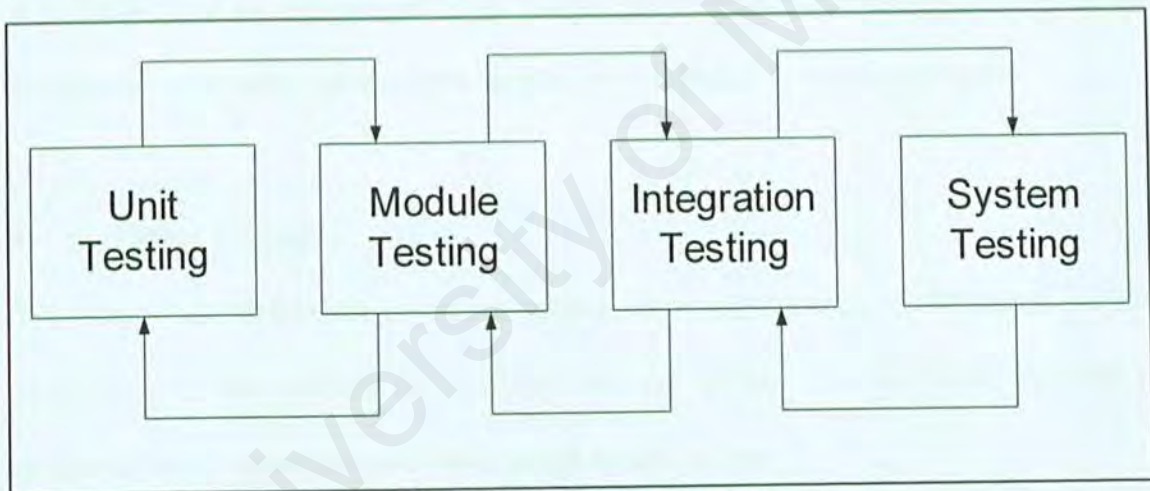


Figure 7-1: Testing Process

Generally, there were 3 stages involve altogether and were listed down as below:

➤ Unit Testing

This is the first stage of testing where each program component is tested on its own and is isolated from the other components in the system. It verifies that the component functions work properly with the types of input and output expected from studying the

component's design. After each component has been tested, the interaction between these components must be tested again to ensure that the components can be integrated.

➤ Module Testing

Module testing is performed without other system modules. A module consists of a collection of dependent components to perform a particular task or function. Different possible test cases are applied to the module and the test results would be verified. Unusual results will be analyzed and they would help in debugging sub-modules in order to produce the desired output. The module testing can be done by creating a new project with the defined sources and predicted results. Besides, the accuracy of the program can increase by comparing the results with that one generated by existing program.

➤ Integration Testing

This stage ensures that the interfaces among the components are defined and handled properly. It is the process of verifying that the system modules work together as described in the system and program design specifications

➤ System Testing

This is the last stage which is performed to find out errors, which result from unanticipated interactions of system components or units. It is to ensure that the whole system works according to users' specifications.

7.1 Testing Strategies

There are a few testing strategies such as unit, integration and system testing are done in order to test the reliability of packet sniffing system.

7.1.1 Unit Testing

Unit testing is done to uncover errors in each module. The primary goal of unit testing is to confirm that the unit is correctly coded and that it carries out the function as it is supposed to perform. Each unit is tested independently in order to assure their accuracy. For this system – packet sniffing system, each module may contain sub modules and the sub modules may consist of functions. The functions are individually tested before the entire module is tested. In the development of packet sniffing, unit testing was conducted after development of each of the component and it is a continuous process throughout the coding phase.

➤ Packet Sniffing System Unit Testing

Below are some of the units testing being done on packet sniffing system:

- Test whether the system can successfully capture the packets from the network which switch monitoring port.
- Test whether the filtering packet can successfully filter the packets which want to display in the listview.
- Test whether the records being displayed is correct and matches the search criteria.
- Test whether the column listview can be sorted.
- Test whether the password can be sniff using html to display content of the packets.
- Test whether the statistic graph can successfully displayed.

➤ Packet Sniffing System Debugging Strategies

Debugging is actually of finding and fixing the errors. There are several debugging strategies that applied in e-veterinary such as:

- Built-in Error Detection

Error will be discovered if a program is not performing well. VS.NET has built-in error detection where an error message together with the lines number where the error occurred will be debugged. With this features, the debugging work becomes much easier and faster.

- Reviewing the Algorithm Used

Reviewing algorithm and computations for the correctness and efficiency will help to discover logic error or database error. Usage of different algorithms will sometime increase the efficiency of the program.

- Display the Passing Value On Screen

By displaying the passing value on screen, it helps to ensure that the correct value has been passed to the next program for processing.

- Check Success Status

The success status is checked to determine whether to continue the process or exit from the program and display error message whenever there is failure in the previous process.

7.1.2 Integration Testing

The purpose of the integration testing is to know whether the entire software is able to work as one program. It will also verify that each module will be able to function together. Integration testing concentrates on module interaction and the detection of

interface errors. The design specification is referred for the purpose of verification and helps to test the software according to the dependencies present in particular module that being tested. For packet sniffing system integration testing, the system is viewed as a hierarchy of components, where each component belongs to a layer of design. The approach applied in testing the packet sniffing system is referred as Top-Down Integration where integration will start at the highest level of main program or module or sub modules are gradually added until the bottom is reached.

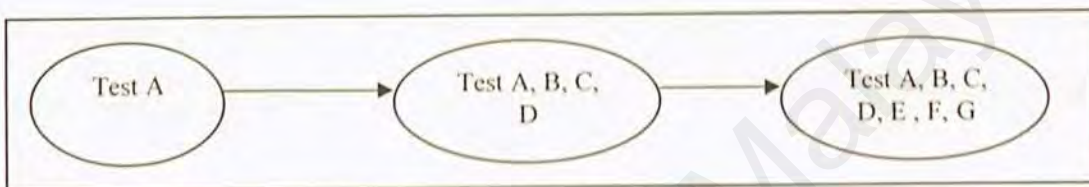


Figure 7-2: Top-Down Testing

7.1.3 System Testing

A system testing is a series of different test designed to fully exercise the system to uncover its limitation and to measure its capabilities. The objective is to test an integrated system and verify that it meets the specified requirements. Several steps were taken in testing packet sniffing system such as function testing and performance testing.

➤ Function Testing

System testing begins with function testing that focus completely on functionality. The system structure is being ignored. The testing is based on the system's functional requirements which are stated in the early chapter.

➤ **Performance Testing**

Performance testing aims at testing the run-time performance. Response time of the event triggered was checked to verify the performance of the system.

7.2 Test Cases

Test cases are developed to show that the input is properly converted to the desired output. They are used as some set of structural input is given and the output is observed. The test cases are design to perform unit testing till integration testing with the specific results. Repetitive testing is done on a single test case to prove the consistency of the results.

7.2.1 Unit Test Cases

Unit test cases aim to test an individual independent component. Below is an example of unit test case that being done for packet sniffing system.

Test Case : 1				
Module : Main				
Sub Module : Save				
Unit : Main – Edit				
Scenario : Save a packet to the disk				
No	Steps/Fields	Test Data	Expected Result	Test Result
1	Click “Save” in the menu item.		System prompts one window	New Window is prompted.

			information	
2	Put the "File Name" in the text field in save window.		System displays text field to write the file name in the save dialog	Text Field is displayed.
3	Click Ok button on save window		System save the file in the save dialog.	File is save to the disk.
4	Click Cancel button on the save window		System closes the save window	Save window is closed.
Status : Pass Date : 1/25/2005				

Figure 7-3: Unit Test Case

7.5 Chapter Summary

Testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. Unit, integration and system testing has been carried out for e-veterinary. At the end of the testing phase, the system should be able to perform the tasks required and free of some errors.

Chapter eight present the system evaluation which reveals the problem encountered and solution, system strength and system constraints, future enhancements, knowledge and experience gained.

Chapter 8

System Evaluation

Chapter 8 – System Evaluation

8.1 Introduction

System evaluation is a process where system developer evaluates the system after the system has been fully developed. Normally, the developer will evaluate the system from many aspects, which will summarize the system strengths and limitation. . In addition, the problems encountered during process developing were listed down with the solutions figure out during the developing process.

After all the handwork of designing and developing as well as implementing packet sniffing system, the end product of the project is brought up for evaluation. They were many evaluation techniques that used to evaluate the final system. This chapter also includes the future enhancements for performance the system in order to get a more satisfactory system.

8.2 System Strengths

Below are the strengths of Packet Sniffing System application:

➤ Simple, Standard and User-Friendly Interface

Packet sniffing system application is specially designed on the principle for ease to use. So, all forms are kept simple. The inclusion of graphic user interface has contributed vastly to aid users. The learning curve is foreseen to be short and a user will be able to use the system with ease within minutes.

➤ Scalability

Hardware and applications could be easily added to the existing system without influence the existing applications. This was because the system was not hardware-dependent.

➤ Advance Functions

Application not only designs for packet sniffer and analyzer but with advance functions such as below:

- sniffing sensitive word such as passwords which use for the login to the email system and save it in html file.
- showing the computer name from each users in the network.

➤ Graph for Statistic

Packet sniffing system is with capability to show the graph statistic for protocol, the most active ten host using TCP and UDP. This capability can help user in analysis the most using protocol in the traffic and host the using most TCP and UDP and the traffic.

➤ Provides Filter To Capture Specific Packets

Packet sniffing system provides filter to capture specific packets by protocol, IP address, and custom filters. Packet filtering will help user filter to accept packet if matched and explicitly for gathering desire information about the traffics.

8.3 System Constraints and Limitations

Below are the constraints and limitations of packet sniffing system:

➤ Lack of Supported Protocol in Encoding Packet

Packet sniffing system only supports five general protocols which are ARP, TCP, UDP, ICMP and IGMP. These five general protocols are enough to encoding most of the packet on the network but for future enhancement is better to support more protocols for details information about the packet and the network.

➤ Limitation of Threading Programming Concept

Packet sniffing system can't perform a function like follow TCP stream if the application. In future enhancement, this system must apply threading programming concept to perform the TCP stream.

➤ Lack of Functionality To Assist User In Creating Configuration File

This limitation occur when more configuration need to be set before to start capture packet which means a user need to repeat configuration for the same configuration setting for each time using the application.

8.4 Future Enhancements

As there are many types of packets with its own protocol and different data type, the analyzing of the packets still limited in my project which only focus only the regularly used protocols such as TCP and UDP, ARP, ICMP and EIGRP. The analyzing of packets still limited until transport layer of the OSI model. It can be further enhanced to analyze until the application layer such as HTTP and to follow the TCP stream.

The existing system can be used to sniffing for an authentication process. If there is no encryption applied, the user login's information can be seen in the text area. But it requires us to manually do trace the packets one by one. It can be enhanced to automatically search for the information and generate reports to us.

In the future, packet sniffing suppose must capture the packet with duration time in day by day. This is a challenge for all packet sniffer application because for a LAN network, packet sniffer not only monitoring its own packet but for the whole network. Therefore, packet sniffer will receive many packets and can not help in user to analysis the network because of to many packets. This problem can solve by develop a database for the captured packet on the traffic by time or by total of packets.

Packet sniffing system also must generate various type of report in the future. Because of, the analysis of the network traffic is more useful for the user if the application allows the users to generate various type of report about the network traffic. Besides that, for more advances function user can customized the report to generate more various reports which all of this report will give the user valuable information about the network.

8.5 Chapter Summary

Overall, the Packet Sniffing system has achieved and fulfilled the objectives and functional requirements as a packet sniff and analyzer as determined during system analysis. This application enable user to capture network traffic packet and analysis the network traffic.

This project gives me an opportunity to build a full application which theories and knowledge gained throughout the course of computer science studies like system analysis, design and software engineering were literally put into practice. The knowledge also can gain in programming, concepts and challenge to develop a system alone. Even though programming skills and techniques are important in development, good software engineering techniques must also be applied. It gives me a strong foundation to take this project as long as to complete it.

Finally, there are much more rooms for improvement in this system, especially in terms of implementing a more perfect system. With the first step taken, enhancements could still be made with more features and function added for future version.

References

- [1] <http://www.node99.org/projects/arpspoof/arpspoof.pdf>
- [2] <http://www.colasoft.com/resources/network-sniffer.php>
- [3] www.dictinct.com
- [4] www.packetmon.com
- [5] www.ethereal.com
- [6] <http://counter.search.bg>
- [7] <http://www.erg.abdn.ac.uk/users/gorry/course/intro-pages/uni-b-mcast.html>
- [8] <http://www.bruegge.in.tum.de/teaching/ss02/DistComp/Slides/Dmitri.ppt>