
Faculty of Computer Science and Information Technology

Perpustakaan SKTM

**WXES 3182: PROJEK ILMIAH TAHAP AKHIR II
SESSION 2004/2005**

**Muhammad Izwan Bin Idris
WEK 020144**

**LAN Monitoring and Management Tool
(LMMT)**

**Supervisor : Pn. Fazidah bt Othman
Moderator : En. Nor Badrul Anuar Juma'at**



UNIVERSITY OF MALAYA

ABSTRACT

LAN Monitoring and Management Tool (LMMT) is an application that monitors and manages interconnected computer via a network. This tool is design for small to medium company that is running their performance in Local Area Network (LAN).

The goal is to create a remote application, which covers the management of the connected computer and monitor the traffic across a network. This application obtains a maximum of capabilities and efficiency, which allows the administrator to monitor all the traffic flow in their network, controls other computer in a systematic way.

This application development based on the users need, which is focus on network engineer and administrator. It may help these target users to gain the information of the activity running in their network environment and manage it efficiently.

This application will be developing base on **Windows Packet Capturing (WinPcap)**, **Windows Socket (Winsock)** and **Windows Management Instrumentation (WMI)** technology using Visual Basic. All are rich in queries and provide powerful technologies.

ACKNOWLEDGEMENT

The success completion of this project is related to the contributions of many people. I would like to take this opportunity to thank some of the key people.

First of all, I would like to thank Puan Fazidah bt Othman, my supervisor for giving me this opportunity to develop this project. Secondly, I would like to thank for her constructive advice, generous guidance, encouragement, support, dedication and supervision along the progress of this project. Her diligence and kindness in helping me throughout the project is deeply appreciated.

I also would like to acknowledge En. Nor Badrul Anuar Juma'at, as the project moderator who contributed comments, suggestions and ideas to further enhance value of this project.

Lastly, the appreciation is also dedicated the others who have giving me valuable supports and encouragements and provide important information for the project.

Thank you very much!

TABLE OF CONTENTS

ABSTRACT..... i

ACKNOWLEDGEMENT..... ii

TABLE OF CONTENTS..... iii

LIST OF FIGURES..... vii

LIST OF TABLES..... viii

CHAPTER 1 INTRODUCTION

1.1 Project Overview..... 1

1.2 Problem Statements..... 3

1.3 Project Objectives..... 4

1.4 Project Scopes..... 5

1.5 Limitation of Project..... 6

1.6 Expected Outcomes..... 7

CHAPTER 2 LITERATURE REVIEW

2.1 Literature Review Overview..... 8

2.2 Domain Studies..... 9

2.3 Overview of Networking..... 10

2.3.1 Area Network..... 10

2.3.1.1 LAN Basic..... 11

2.3.1.2 WAN Basic..... 11

2.3.2 Network Topology..... 12

2.3.2.1 Bus Topology..... 13

2.3.2.2 Ring Topology..... 14

2.3.2.3 Star Topology..... 15

2.4 Technologies Consideration..... 16

2.4.1 WMI..... 16

2.4.2	WinPcap.....	18
2.4.2.1	Component of Packet Filtering.....	19
2.4.3	Winsock.....	26
2.5	Programming Language Consideration.....	26
2.5.1	Visual Basic.....	26
2.5.2	Java.....	27
2.6	Operating System Consideration.....	29
2.6.1	Windows NT.....	29
2.6.2	Windows 2000.....	30
2.6.3	Windows XP.....	32
2.7	Existing System Review.....	33
2.7.1	Ipswitch WhatUP Gold 8.0.....	33
2.7.2	GFI Network Server Monitor 5.0.....	34
2.7.3	Monitor Magic.....	36
2.7.4	Comparison Table.....	37
2.8	Conclusions.....	38

CHAPTER 3 METHODOLOGY

3.1	Introduction.....	39
3.2	System Development Methodology.....	40
3.3	Methodology Suitability to Project Domain.....	43
3.4	Information Gathering Method.....	44
3.4.1	White Document and Reference Book.....	44
3.4.2	Internet Research.....	45
3.4.3	Brainstorm.....	45
3.4.4	Observation.....	46

CHAPTER 4 SYSTEM ANALYSIS

4.1	Introduction.....	47
4.2	Objective of System Analysis.....	47
4.3	System Requirement Analysis.....	48

4.3.1 Functional Requirement.....	48
4.3.2 Non Functional Requirement.....	49
4.4 Limitation of LMMT.....	51
4.5 Choices of Programming Language and Technology.....	52
4.5.1 Programming Software Chosen.....	52
4.5.2 Monitoring and Management Technologies Chosen.....	53
4.5.3 Development Platform Chosen.....	55
4.6 Hardware Requirement.....	56
4.7 Summary.....	57

CHAPTER 5 SYSTEM DESIGN

5.1 Introduction.....	58
5.2 System Architecture Design.....	59
5.2.1 Context DFD.....	59
5.2.2 System Structure Chart.....	60
5.2.3 WinPcap Architecture.....	61
5.2.4 WMI Architecture.....	64
5.3 Graphical User Interface Design.....	66
5.4 Summary.....	71

CHAPTER 6 SYSTEM IMPLEMENTATION

6.1 Introduction.....	72
6.2 Development Environment.....	72
6.2.1 Hardware Requirement.....	72
6.2.2 Software Requirement.....	73
6.3 Platform Development.....	73
6.3.1 Development Environment Setting.....	74
6.4 Development of The System.....	74
6.4.1 Coding Approach.....	74
6.5 Coding Documentation.....	75
6.6 Module Implementation.....	76

6.6.1 Monitoring Module.....	76
6.6.2 Management Module.....	77
6.7 Summary.....	78

CHAPTER 7 SYSTEM TESTING

7.1 Introduction.....	79
7.2 Type of Fault.....	80
7.3 Testing Strategy.....	81
7.3.1 Unit Testing.....	81
7.3.2 Integration Testing.....	83
7.3.3 System Testing.....	85
7.3.3.1 Performance Testing.....	85
7.3.3.2 Load Testing.....	85
7.4 Summary.....	86

CHAPTER 8 SYSTEM EVALUATION

8.1 Introduction.....	87
8.2 System Strength.....	87
8.3 System Limitation and Constraint.....	89
8.4 Future Enhancement.....	90
8.5 Problem Encountered.....	91
8.6 Knowledge Gained.....	92
8.7 Summary.....	93

REFERENCES.....	94
-----------------	----

APPENDIX

Source Code Sample

User Manual

LIST OF FIGURES

<u>Figure</u>	<u>Page</u>
Figure 2.1 Bus Topology	13
Figure 2.2 Ring Topology	14
Figure 2.3 Star Topology	15
Figure 2.4 Segment Format	20
Figure 2.5 User Datagram Header Format	23
Figure 2.6 GFI Network Server Monitor	34
Figure 2.7 MonitorMagic	36
Figure 3.1 The Waterfall Model	40
Figure 5.1 Context DFD	59
Figure 5.2 System Structure Chart	60
Figure 5.3 Structure of Capture Stack	62
Figure 5.4 WMI Architecture	64
Figure 5.5 Administrator Login Dialogue Box	66
Figure 5.6 LMMT Dashboard	67
Figure 5.7 Computer Browser	68
Figure 5.8 Port Monitor	69
Figure 5.9 Sniffer	70

LIST OF TABLES

<u>Table</u>	<u>Page</u>
Table 2.1 Enhanced WMI Features	17
Table 2.2 Windows 2000 Professional Foundation	35
Table 2.3 Product Comparison Table	41
Table 6.1 Development Tools	73
Table 7.1 Unit Testing Sample 1	82
Table 7.2 Unit Testing Sample 2	82
Table 7.3 Integration Testing	84

CHAPTER 1: INTRODUCTION

1.1 Project Overview

Nowadays, human beings are depending on complex computer-based system. As business, education, and government become more global and competition among organization increase, there is no end for the growing use of information technologies. Therefore, we will realize that information technology must be well monitored and managed in order to support business process to running smoothly.

The monitoring and management system in small and medium company tend to be difficult for the administrator in maintaining fraudulent of computer by the internal staff due to the growing use of hardware and software technologies. Therefore, **LAN Monitoring and Management Tool (LMMT)** is propose as to allow the network administrator or even engineer to cover the monitoring and management of computers via network.

LMMT is develops for the administrator to interact with the system without intervention of other users. This application must be run under Windows server family (2000, NT, XP) operating system as those platform are supporting **Windows Management Instrumentation (WMI)**, **WindowsSocket (Winsock)** and **Windows Packet Capturing (WinPcap)** technologies. Using WMI, Winsock and WinPcap, we can create monitoring and management application that implement a number of features such as

traffic monitoring, generating an inventory of network resources, displaying system information and controlling computers remotely.

The proposed system is more efficient and simple to operate in order to allow the network administrator in monitoring and controlling fraudulent of computers by the users. LMMT provide a numerous of capabilities such as traffic filtering, packet capturing, port listening, traffic analysis, viewing remote computer information, remote management (process and services) and shutting down and rebooting computer in the range of the LAN.

At the end of the project, the new system with the economical, efficient, flexible, reliable and expandable features will be develops.

1.2 Problem Statements

Nowadays, many systems provide remote monitoring, managing which can control, and monitor computer through a main server. However, there are several problems and some reasons that motivate the project to be develop as the solution for the problems:

- **The increase of number of users and fraudulent**

In a LAN environment, not only one user using the computer but it consists of many users sharing the same network. The numbers of user are expecting to contribute a lot of fraudulent to network, which can distract the network flows that can cause to business loss.

- **Management method are less effective**

Managing the computers one-by-one either to obtain specification and functions or working with system resource and services are too time consuming and not effective. Thus, there is a need to have effective method for managing the computers in shorter time.

- **More computer ports are open to enable business function**

Because of nowadays computing requires a lot of new technologies; more ports in a computer are open to make sure the technology are running. This can decrease the computer security itself which more attack is exposing to the open ports. That makes a responsibility to monitor all port over the network.

1.3 Project Objectives

The proposed system is a better and more efficient in monitoring and managing a network. It also can overcome the problems encountered in the present system and cater the needs of user. The new system being design is to achieve the following objectives:

- **Security**

Unauthorized access of the system may cause hazardous to the whole LAN. Thus, it was equipped with log in authentication where only authorized personnel with the correct username and password are allows to access into the system.

- **Clientless**

To develop an application that allows the administrator to manage the computer without intervention of other users

- **Ease of Monitoring**

Ongoing monitoring from time to time on the desire LAN can be simplify and fraudulent can be easily trace.

- **Reduce time**

To speed up the management time for data gathering and remote managing.

1.4 Project Scopes

The main purpose of the project is to develop an application that covers the monitoring session, management, and run the vital utilities remotely across the network. The following scopes are required as to meet the requirements:

- **Secure Login System**

The system should be able to set the parameters for authorized personnel to access into the system.

- **Remote Monitoring System**

Able to monitor all activities within the LAN including port monitoring, TCP/IP and UDP monitoring.

- **Accessible System**

The system shall able to allow the administrator to access and control other computer in the same LAN through a main computer whereby the computers must pose Windows Management Instrumentation (WMI).

- **Develop a simple and user-friendly system that is easy to operate and menu driven.**

1.5 Limitation of Project

There is a few limitation of this project as below:

- **Suitable Environment and Tool**

This system is just available for the small to medium-scale organization that required the Local Area Network, network interface card (NIC) and operating system which support WinPcap and WMI technology such as Windows 2000/NT/XP.

- **Limited of Accessibility**

The user cannot view the desktop of other personal computer. He/She can access and control it by rebooting, and shutting down the other computer.

- **Limited of Remote Monitoring**

Administrator can only monitor the computers within the LAN.

- **Limited of User Involvement**

This system is clientless. Thus, it is aims to authorized personnel such as network administrator and system administrator with the correct identification and password to access into the system.

1.6 Expected Outcomes

The expected outcomes of this project are summarized as below:

- The system achieved all functionality and objectives of the project within the scopes of the project.
- System usability – a simple, user friendly, menu driven and easy to use where the administrator can interact with the system easily.
- System reliability – system should have errors handling mechanism and bugs-free design

CHAPTER 2: LITERATURE REVIEW

2.1 Literature Review Overview

Literature review is a systematic approach, the review of literature works in various topics performed in the scope of the undertaken thesis project. A review of literature is essential in uncovering knowledge required before decisions are made upon certain aspect of the project.

Literature review of a project is important as it places the project in the content of others, which might have similar characteristic. It helps the developer to know some of the existing features offered by similar system.

Another important purpose of a literature review is to sufficiently equip the developer with some knowledge of the strength and limitations of several developments tools. This can help the developer to choose the right tool to develop the system.

2.2 Domain Studies

In this project, below terms are refers to:-

LAN

A Local Area Network (LAN) provides networking capability to a group of computers in close distance to each other such as in an office building, a school, or a home.

Monitoring

Ability to verify the network connectivity of TCP/IP hosts on the Internet and LAN, checking TCP and UDP ports of each connected computer on the LAN and analyze the traffic flow and monitor the resource of other connected computer.

Management

Ability to control and monitor a computer network from a central location. It includes controlling the computer resource such as services, running application and many more. Restarting and shutting down connected computer also one of the capabilities.

Tool

A program that helps the user analyzes or search for data. It normally an on-screen function in a graphics program.

2.3 Overview of Networking

In the computer world, networking is a practice of linking two or more computing devices for the purpose of sharing data. Networks are built with a mix of computer software and hardware.

2.3.1 Area Network

Network can be categorized in several different ways. For historical reasons, the industry refers to nearly every type of network as an "area network." The most commonly-discussed categories of computer networks include the following:-

- Local Area Network (LAN)
- Wide Area Network (WAN)
- Metropolitan Area Network (MAN)
- Storage Area Network (SAN)
- System Area Network (SAN)
- Server Area Network (SAN)
- Small Area Network (SAN)
- Personal Area Network (PAN)
- Desk Area Network (DAN)
- Controller Area Network (CAN)
- Cluster Area Network (CAN)

LANs and WANs were the original taste of network design. The concept of "area" made good sense at this time, because a key distinction between a LAN and a WAN involves the physical distance that the network spans. As technology improved, new types of networks appeared on the scene. These, too, became known as various types of "area networks" for consistency, although distance no longer proved a useful differentiator.

2.3.1.1 LAN Basic

A Local Area Network (LAN) connects network devices over a short distance. A networked office building, school, or home usually contains a single LAN, though sometimes one building will contain a few small LANs, and occasionally a LAN will span a group of nearby buildings.

Besides in a limited space, LANs include several other distinctive features. LANs are typically owned, controlled, and managed by a single person or organization. They also use certain specific connectivity technologies, primarily Ethernet and Token Ring.

2.3.1.2 WAN Basic

As the term implies, a Wide-Area Network (WAN) covers a large physical distance. A WAN is a collection of LANs. A network device called a router connects LANs to a WAN. WANs differ from LANs in several ways. Like the Internet, most WANs are not owned by any one organization but rather exist under collective or distributed ownership

and management. WANs use technology like ATM, Frame Relay and X.25 for connectivity.

2.3.2 Network Topology

A network topology represents its layout or structure from the point of view of data flow. One can think of a topology as a network's "shape." This shape does not necessarily correspond to the actual physical layout of the devices on the network.

Network topologies are categorized into the following basic types:

- bus
- ring
- star
- tree
- mesh

In this subchapter, only 3 most important topologies will be discussed.

2.3.2.1 Bus Topology

Bus networks use a common backbone to connect all devices. A single cable, the backbone functions as a shared communication medium, that device attach or tap into with an interface connector. A device wanting to communicate with another device on the network sends a broadcast message onto the wire that all other devices see, but only the anticipated recipient actually accepts and processes the message.

Ethernet bus topologies are relatively easy to install and do not require much cabling compared to the alternatives. Bus networks work best with a limited number of devices. If more than a few dozen computers are added to a bus, performance problems will be occurred. In addition, if the backbone cable fails, the entire network will be down.

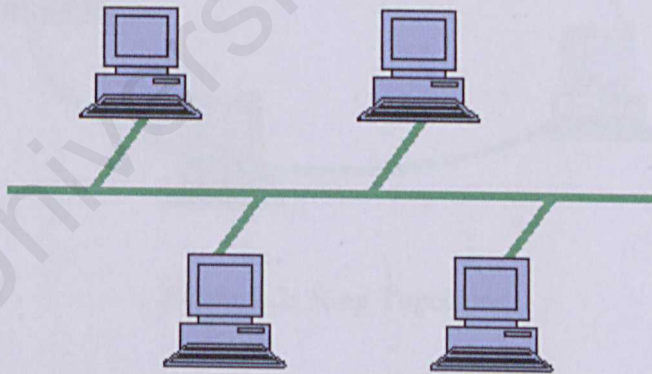


Figure 2.1: Bus Topology

2.3.2.2 Ring Topology

In a ring network, every device has exactly two neighbors for communication purposes. All messages travel through a ring in the same direction. A failure in any cable or device breaks the loop and can break down the entire network.

To implement a ring network, we can use FDDI, SONET, or Token Ring technology. Rings are found in some office buildings or university campuses.

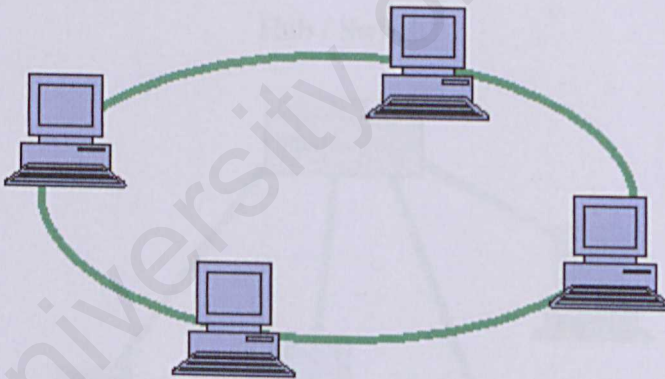


Figure 2.2: Ring Topology

2.3.2.3 Star Topology

Many home networks use the star topology. A star network features a central connection point called a "hub" that may be an actual hub or a switch. Devices typically connect to the hub with Unshielded Twisted Pair (UTP) Ethernet.

Compared to the bus topology, a star network generally requires more cable, but a failure in any star network cable will only take down one computer's network access and not the entire LAN. If the hub fails, the entire network also fails.

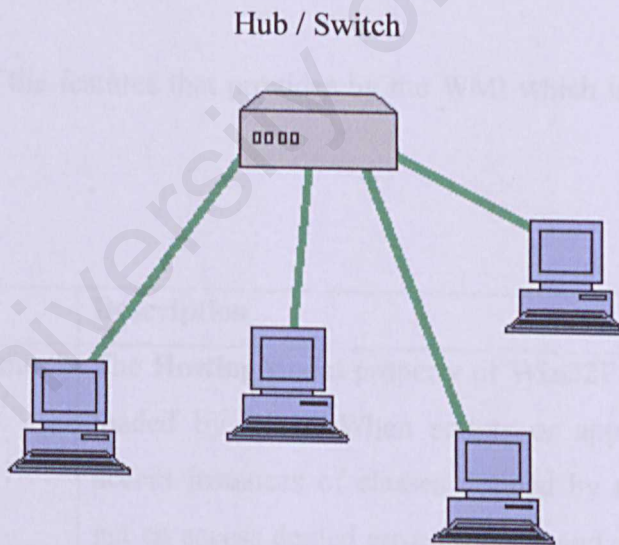


Figure 2.3: Star Topology

2.4 Technologies Consideration

2.4.1 WMI

Window Management Instrumentation (WMI) is a component of Microsoft Windows Operating System that provides management information and control. WMI is use to query and set information on desktop systems, applications, networks, and other enterprise components. Using an appropriate architecture, WMI can communicate with management application by using variety of interfaces such as Visual Basic and C++.

WMI can integrate with Windows components, such as directory service to allow for a unified management experience. WMI is preinstalled into Windows Server 2003, Windows XP, Windows Me, and Windows 2000.

Below are some of the features that provides by the WMI which is enhanced from the previous version:

Features	Description
Provider hosting more secure	The HostingModel property of Win32Provider is no longer loaded by WMI. When scripts or applications attempt to access instances of classes defined by those providers they get an access denied error message and an entry is written to the NT Event Log
Prevention of code execution on memory data pages.	Win32_OperatingSystem has three new properties to help prevent buffer overrun attacks by indicating whether code can execute on memory data pages. The properties are DataExecutionPrevention_32BitApplications , DataExecutionPrevention_Available ,

	DataExecutionPrevention_Drivers
Remote connections	WMI is designed to obtain data from remote computers in a network
Administrator and non-administrator access rights.	WMI security is based on requiring administrator permissions for most access and operations.
Shadow Copy Provider and Storage Volume Provider	Two new storage providers have been added. The Shadow Copy provider supplies management functions for the Windows Server 2003 Shared Folders feature. The Storage Volume provider enumerates all volumes known to the Windows Mount Manager, and manages volume drive letters, mount points, and storage quotas.

Table 2.1: Enhanced WMI features

2.4.2 WinPcap

WinPcap is an architecture for packet capture and network analysis for the Windows platforms. The packet filter is a device driver that adds to Windows 95, 98, ME, NT, 2000, XP the ability to capture and send raw data from a network card, with the possibility to filter and store the captured packets.

WinPcap includes a low-level dynamic link library (**packet.dll**), and a high-level and system-independent library (**wpcap.dll**).

Packet.dll is an API that can be used to directly access the functions of the packet driver, offering a programming interface independent from the Microsoft Operating System.

While, **wpcap.dll** exports a set of high level capture that are compatible with the well known UNIX capture library, libpcap. These functions allow capturing packets in a way independent from the underlying network hardware and operating system.

For the time being, WinPcap can be integrated into many programming language such as C++, C# and Visual Basic. To implement WinPcap into visual basic, we need an extra dynamic link library called **vbpcap.dll** in order to make sure it can be connected to **packet.dll**.

WinPcap is released under a BSD license. So, the source code is freely distributed.

2.4.2.1 Component of packet filtering

Protocol

- **TCP**

The Transmission Control Protocol (TCP) is a means for building a reliable and safe communications stream of data transfer on top of the unreliable packet Internet Protocol (IP). TCP handle the safe delivery of packages across the network.

The basic method of TCP operation involves:-

- ❖ Wrapping higher level application data in segments
- ❖ Wrapping the segments into IP datagrams
- ❖ Associating port numbers with particular application
- ❖ Associating a sequence number with every byte in the data stream
- ❖ Exchanging special segments to start up and close down a data flow between two hosts
- ❖ Using acknowledgments and timeouts to ensure the integrity of the data flow

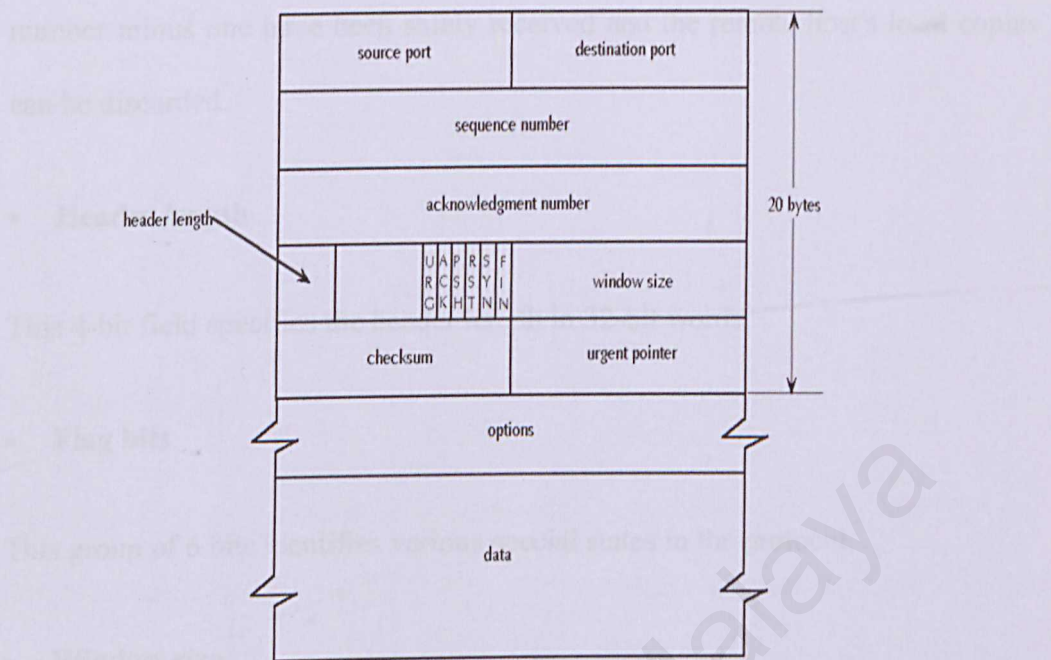


Figure 2.4: Segment Format

TCP segments are constructed from 32 bit words and include a 20 byte header.

- **Source and Destination port number**

The source (and destination) port numbers are used for demultiplexing the data stream to applications.

- **Sequence number**

This 32-bit number identifies the **first** byte of the data in the segment.

- **Acknowledgment number**

This 32 bit number is the byte number of the next byte that the sender expects to receive from the remote host. The remote host can infer that all bytes up to this

number minus one have been safely received and the remote host's local copies can be discarded.

- **Header length**

This 4-bit field specifies the header length in 32-bit words.

- **Flag bits**

This group of 6 bits identifies various special states in the protocol.

- **Window size**

The amount of space that the receiver has available for the storage of unacknowledged data. The units are bytes unless the window scale factor option is used. The maximum value is 65535.

- **Checksum**

This covers both the header and the data. It is calculated by prepending a **pseudo-header** to the TCP segment, this consists of 3 32 bit words which contain the source and destination IP addresses, a byte set to 0, a byte set to 6 (the protocol number for TCP in an IP datagram header) and the segment length (in words). The checksum field of the TCP segment is set to zero and the following algorithm applied to the prepended segment treated as a sequence of 16 bit (unsigned) words.


```
unsignedlong    cksum = 0;
unsignedshort   *sptr;
while(sptr points to part of prepended segment)
    cksum += *sptr++;
cksum = (cksum >> 16) + (cksum & 0xffff);
cksum += (cksum >> 16);
cksum = (short)(~cksum & 0xffff);
```

- **Urgent pointer**

This is part of TCP's mechanism for sending urgent data that will overtake the normal data stream. If the URG flag bit is set this field indicates the position within the data of the last byte of the urgent data. There is no way of indicating where the urgent data starts.

- **Options**

There are a large number of options. The most useful is the Maximum Segment Size (MSS) specification facility.

- **UDP**

The User Datagram Protocol (UDP) supports network applications that need to transport data between computers. Applications that use UDP include client/server programs like video conferencing systems.

UDP's main purpose is to abstract network traffic in the form of *datagrams*. A datagram comprises one single "unit" of binary data. The first eight (8) bytes of a datagram contain the *header information* and the remaining bytes contain the data itself.

UDP Headers

The UDP header consists of four (4) fields of two bytes each:

- source port number
- destination port number
- datagram size
- checksum

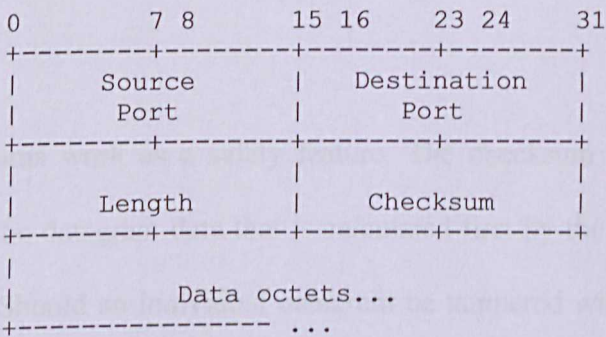


Figure 2.5: User Datagram Header Format

UDP port number

UDP port numbers allow different applications to maintain their own "channels" for data. Both UDP and TCP use this mechanism to support multiple applications sending and receiving data concurrently. The sending application (that could be a client or a server) sends UDP datagrams through the source port, and the recipient of the packet accepts this datagram through the destination port. Some applications use *static port numbers* that are reserved for or registered to the application. Other applications use *dynamic* (unregistered) *port numbers*. Because the UDP port headers are two bytes long, valid port numbers range from 0 to 65535. The values above 49151 represent dynamic ports.

Datagram Size

The UDP datagram size is a simple count of the number of bytes contained in the header and data sections. Because the header length is a fixed size, this field refers to the length of the variable-sized data portion. The maximum size of a datagram varies depending on the operating environment. With a two-byte size field, the theoretical maximum size is 65535 bytes.

Checksum

UDP checksums work as a safety feature. The checksum value represents an encoding of the datagram data that is calculated first by the sender and later by the receiver. Should an individual datagram be tampered with (due to a hacker)

or get corrupted during transmission (due to line noise), the calculations of the sender and receiver will not match, and the UDP protocol will detect this error.

IP Address

An IP address is the logical address of a network adapter. The IP address uniquely identifies computers on a network. An IP address can be private, for use on a LAN, or public, for use on the Internet or other WAN. IP addresses can be determined statically (assigned to a computer by a system administrator) or dynamically (assigned by another device on the network on demand). IP addresses consist of four bytes (32 bits). Each byte of an IP address is known as an octet. Octets can take any value between 0 and 255, but various rules exist for ensuring IP addresses are valid.

Port Number

A port number represents an endpoint or "channel" for network communications. Port numbers allow different applications on the same computer to utilize network resources without interfering with each other.

Sometimes, port numbers are made visible to the casual user. For example, some Web sites a person visits on the Internet use a URL like the following:

`http://www.tnbserver.com.my:8080/`

In this example, the number 8080 refers to the port number used by the Web browser to connect to the Web server. Normally, a Web site uses port number 80 and this number need not be included with the URL.

2.4.3 Windows Socket (Winsock)

Winsock is an application programming interface (API) standard for software that provides a TCP/IP interface under Windows. It is a set of routines that an application uses to request and carry out lower-level services performed by a computer's operating system. These routines usually carry out maintenance tasks such as managing files and displaying information. Winsock can be integrate with variety of programming language and work perfect with visual basic.

2.5 Programming Language Consideration

2.5.1 Visual Basic

The Visual refers to the method used to create the graphical user interface (GUI). Rather than writing numerous lines of code to describe the appearance and location of interface elements, you simply add prebuilt objects into place on screen

The Basic refers to the BASIC (Beginners All-Purpose Symbolic Instruction Code) language, a language used by more programmers than any other language in the history of computing.

Visual Basic contains several hundred statements, functions, and keywords, many of which relate directly to the Windows GUI. Beginners can create useful applications by learning just a few of the keywords, yet the power of the language allows professionals to accomplish anything that can be accomplished using any other Windows programming language.

Listed below are the features of Visual Basic:

- Data access features allow you to create databases, front-end applications, and scalable server-side components for most popular database formats, including Microsoft SQL Server and other enterprise-level databases.
- ActiveX technologies allow you to use the functionality provided by other applications, such as Microsoft Word word processor, Microsoft Excel spreadsheet, and other Windows applications. You can even automate applications and objects created using the Professional or Enterprise editions of Visual Basic.
- Internet capabilities make it easy to provide access to documents and applications across the Internet or intranet from within your application, or to create Internet server applications.

2.5.2 JAVA

The JAVA programming language is a high-level language that can be characterized as Object-oriented, architecture-neutral, portable, interpreted, distributed, high performance, secure, dynamic, and robust.

The Java platform has 2 components:

- 1 The Java Virtual Machine (JVM)
- 2 The Java Application Programming Interface (API)

JVM base for the Java platform and is ported onto various hardware-based platforms. The Java API is a large collection of ready-made software components that provide many useful capabilities, such as Graphical User Interface (GUI) widgets...etc. The API supports all these kinds of program within packages of software components. Every full implementation of the Java platform gives the following features:

- The essentials: Object, strings, threads, numbers, input and output etc.
- Applets: The set of conventions used by applets.
- Networking: URLs, TCP (Transmission Control Protocol), UDP (User Datagram Protocol) sockets, and IP (Internet Protocol) addresses.
- Internationalization: Help for writing programs that can be localized for users worldwide.
- Security: Both low level and high level, including electronic signatures, public and private key management, access control and certificates.
- Object serialization: Allows light weight persistence and communication via Remote Method Invocation (RMI).
- Java Database Connectivity (JDBCTM): Provide uniform access to a wide range of relational database.

2.6 Operating System Consideration

An operating system is a program that needs to load into the computer to manage all other programs (application) in computer. The application making request for services through a defined application program interface (API) and users can interact directly with the operating system through an interface such as a command language.

An operating system can perform many services for application and determines the sequence of the application should running in multitasking mode. Manage the sharing of internal memory among multiple applications and the computer's resource such as CPU, memory, disk drives, printers and etc also become the responsibility of the operating system.

2.6.1 Windows NT Server 4.0

Microsoft Windows NT Server 4.0 is one of the leading operating system in the networking worlds. Microsoft still position Windows NT Server 4.0 as a corporate solution and standardized as both a development and deployment platform. Java is supported in this operating system through Microsoft Java 1.1 Virtual Machine. Windows NT Server 4.0 support virtual domains and the ability to delegate administration to other users. In term of management, there is a Windows-based management console, browser-based administration and command-line scripting. In term of security, Windows NT Server 4.0 features user authorization via username and password as configured by the administrator. In addition, it was integrated certificate server for issuing x.509 digital certificates which can be mapped to user accounts.

Advantages of Windows NT Server 4.0

- Inclusion of Internet Information Server (IIS).
- Great integration with other Microsoft Internet / Web development and deployment tools.
- Load balancing is available in Enterprise Edition.

Disadvantages of Windows NT Server 4.0

- Not robust for high-traffic situations.
- Closed architecture.
- Expensive.

2.6.2 Windows 2000 Professional

Windows 2000 Professional was designed for the ground up to the most integrated, comprehensive and easy server operating system and to provide several premises such as scalability, reliability, and manageability necessary for mission-critical applications. It also provides total solution in Intranet and Internet services. Windows 2000 Professional is a fully web-aware operating system, with a built in web server (Microsoft IIS 5.0).

Below is the foundation of Windows 2000 Professional with the premises of its design:

Premise	Description
Scalability	Support symmetric multiprocessing (SMP).

Portability	Would need to run on different hardware platform with minimal change.
Security	It could be locked down through software, meeting NSA's C2-level criteria.
Extensibility	Writing to a well-defined application programming interface (API) could easily expand it on.
Compliance and Compatibility	It would be POSIX-compliant, run existing Windows applications, and support open international standard.
Internationalization	It could easily be ported to run in numerous different language and writing systems with minimal modification to the software.

Table 2.2: Windows 2000 Professional foundation

Moreover, Windows 2000 Professional supports Redundant Array of Inexpensive Disk (RAID) technology that provides data protection. This can help prevent Web Server from easily crash due to hard disk failure. Furthermore, integration of Internet Information Server (IIS) with Windows 2000 Professional provides a fast and secure platform of HTTP, FTP, WWW and Gopher Services.

2.6.3 Windows XP Professional

Windows XP Professional is built on the code base of Windows 2000. It features a 32-bit computing architecture with a fully protected memory model. Thus, it makes Windows XP Professional as the most reliable version yet.

Windows XP also introduces several new features, which one of them is the System Restore. The System Restore enables users and administrators to restore their computer to a previous state without the risk of losing data. It also provides internet connection firewall which can protect small businesses from common internet attacks.

Windows XP can be regard as having the highest security level compared to the previous Windows version. It also has increased level of application compatibility.

2.7 Existing System Review

2.7.1 Ipswitch WhatsUp Gold 8.0

Ipswitch WhatsUp Gold 8.0 is a network monitoring and management tool that to ensure the network services is up and running. It also able to monitor the protocol within the network. Ipswitch WhatsUp is a powerful and intuitive tool that gives network administrators greater control and understanding of their networks and helps keep mission-critical networks up and running. It can automatically map the entire network and monitor critical devices and services. When WhatsUp Gold detects a problem, it immediately notifies you via pager, beeper, email, or other methods.

Key features:

- **Event monitoring system** - Allows troubleshooting network problems by relating device downtime or network strain to a specific event that has recently.
- **Enhanced TCP/UDP Monitoring** – Allow to better pinpoint existing system an existing problem regarding TCP/UDP service monitoring.
- **SMS Notification** - provides broad support for SMS providers to send direct alerts to a pager or cell phone when the networks occur and problem no network.
- **Enhanced NT Service Monitoring** - Adds the convenient option to restart a failed NT/2000/XP service in conjunction with sending a notification of this service failure.

2.7.2 GFI Network Server Monitor

GFI Network Server Monitor is powerful and convenient tool that consists of a network monitoring service (called the Network Monitor Engine) and a separate management interface (the Network Manager). The Network Monitor Engine is multi-threaded and can run 16 checks at a time. This software architecture allows for high reliability and scalability to monitor both large and small networks.

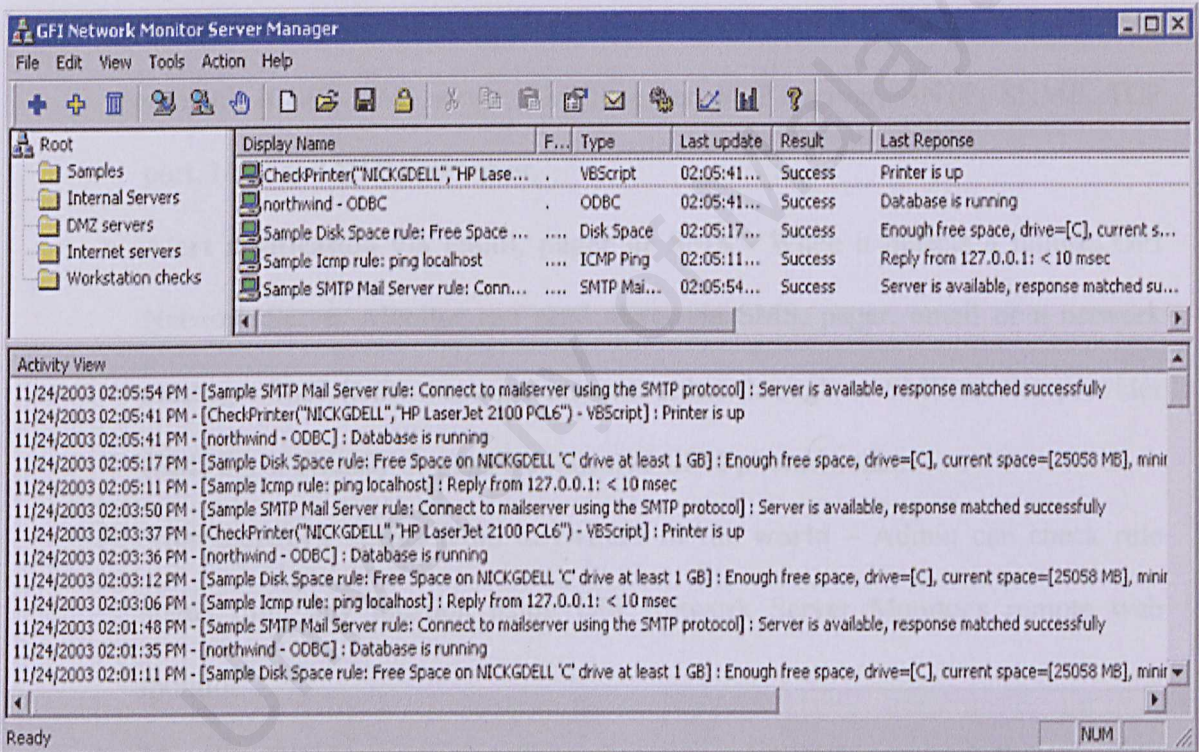


Figure 2.6: GFI Network Server Manager

Key Features:

- **In-built Exchange 2000/2003 monitoring** - monitors the status of your Exchange Server by monitoring critical Exchange services and performance.

- **Monitor database servers (SQL/ORACLE/ODBC)** - check the availability of all leading database applications.
- **Monitor remote event logs** - scan Windows event logs on local or remote computers and look for specific event sources, categories, and event IDs and for patterns in the description of the event.
- **Built-in computer monitor functions** - monitor CPU usage, directory size, disk drive, disk space, file existence and file size.
- **Built in Internet service monitor functions** – monitor HTTP, FTP, Web content, ICMP, DNS server, SMTP server, POP3 server, NNTP, SNMP, TCP port, UDP and NTP timeserver.
- **Alert notification via email, pager or SMS** - When it detects a failure, GFI Network Server Monitor can send alerts via SMS, pager, email or a network message. SMS (text) messages are sent either through an SMS service provider (SMSC), or directly through a connected GSM phone/modem
- **View network status from anywhere in the world** – Admin can check rule status from any location using GFI Network Server Monitor's remote web monitor.
- **Monitor Windows servers & workstations, UNIX/Linux and Novell** - GFI Network Server Monitor can monitor multiple platforms simultaneously including UNIX/Linux and Novell.
- **Monitor users, groups and other Active Directory information** - monitor group membership of the domain group. It can also check user accounts (locked

out, disabled, etc.), computer accounts, groups, group membership, organizational units, and so on.

- **Reporting** - create reports that detail the availability of network resources. Reports are directly created in HTML, or generate XML/CSV reports.

2.7.3 MonitorMagic

MonitorMagic is a proactive monitoring/alerting tool for Windows 2003, 2000 and NT servers, workstations and SNMP devices. It provides flexibility to quickly manage various components of an enterprise level or single server network. With MonitorMagic, we can monitor various objects like disks, services, performance counters, and so on. It uses monitors that correspond with these network devices. When a rule is triggered, this tool can execute alarm actions, for instance: send an E-mail, send an SNMP trap, or restart a service.

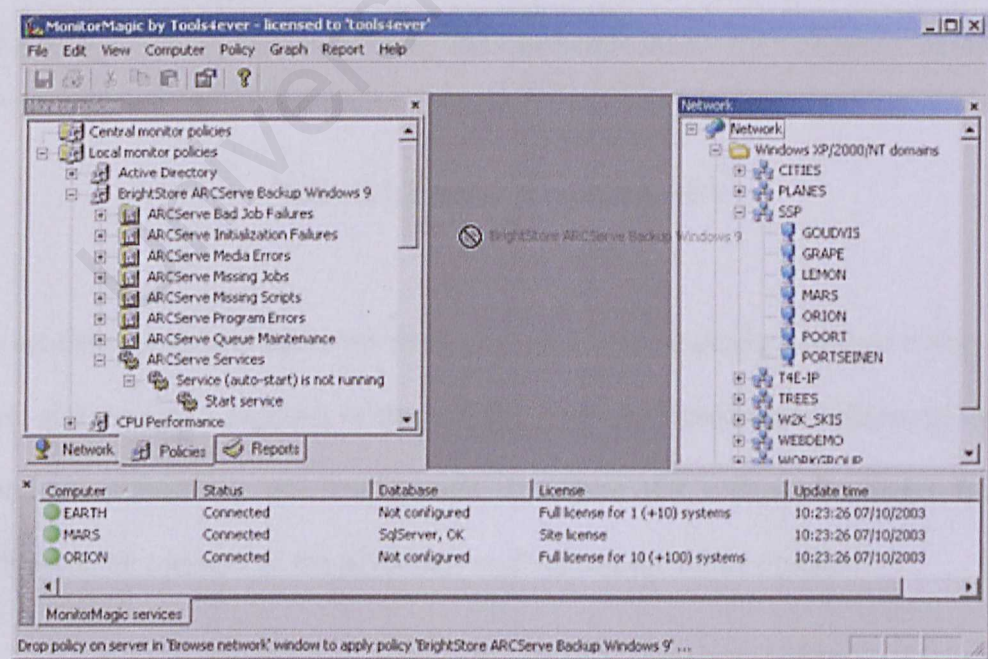


Figure 2.7: MonitorMagic

2.7.4 Comparison table

Attributes\Product	Ipswitch WhatsUp Gold 8.0	GFI Network Server Monitor	MonitorMagic
Business Scale	Medium	Large	Small/Medium
Event monitoring	Yes	Yes	No
TCP/UDP monitoring	Yes	Yes	No
SNMP Trap	Yes	Yes	Yes
System resources monitoring	Yes	Yes	Yes
System resources management	Yes	Yes	Yes
Server monitoring & management	Small range	Broad range	Small range
Different platform server monitoring	No	Yes	No
Active Directory monitoring	Yes	Yes	Yes
Active Directory management	No	Yes	No
Remote Web monitoring	No	Yes	No
Alert Notification	E-mail,pager, SMS, beeper	E-mail,pager,SMS, beeper,network message	E-mail
Report	Instant, HTML, Txt file	Instant,HTML, XML/CSV	Instant, Txt file

Table 2.3: Product comparison table

Base on the comparison table, we manage to differentiate product compatibilities, ease of use, and the target segment of the product. Even each product development is focus on network monitoring and management, but there still a dissimilar target function which is can be consider as the advantage or disadvantage of the product.

2.8 Conclusion

Literature review on various aspects has gained many ideas for me to develop this project. This literature review is the stepping stone for me to consider the requirements for my proposed project. By reviewing the existing system, I managed to find some important features from those reference and the features gave me some ideas to enhance my proposed system. Reviewing such technologies and products gave me ideas on how to build a complete system. With the information gathered, I am able to choose the most suitable technologies for my project.

CHAPTER 3: METHODOLOGY

3.1 Introduction

In the development of any software, one important aspect that we should consider is the aspect of software engineering. The method that we choose and used should be relevant and according to the system requirement of what are we going to develop. In software engineering, there are few software process models that can be implemented to develop a system. Among of the model are the 'waterfall' model, reuse-oriented development, spiral development, incremental development and many more.

In general, software engineers adopt a systematic and organized approach to their work as this is often the most effective way to produce high-quality software. However, engineering is all about selecting the most appropriate method for a set of circumstances.

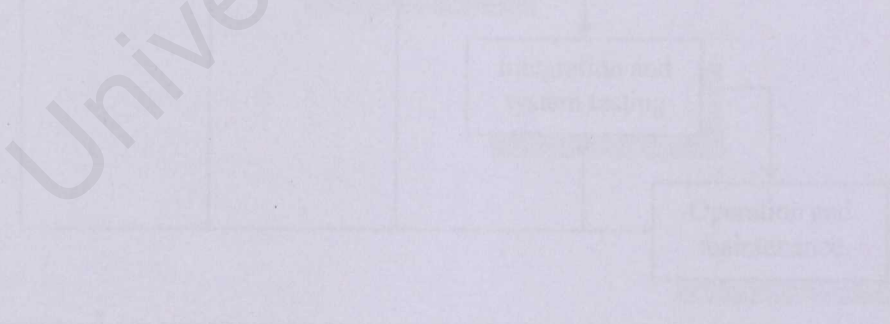


Figure 3.1: The Waterfall Model

3.2 System Development Methodology

The methodology used in the development of LAN Monitoring and Management Tool (LMMT) is the Waterfall model. The Waterfall model offers the benefit of a structured development, in addition to good visibility and proper documentation for each development stage. Figure 3.1 shows the stages included in the model.

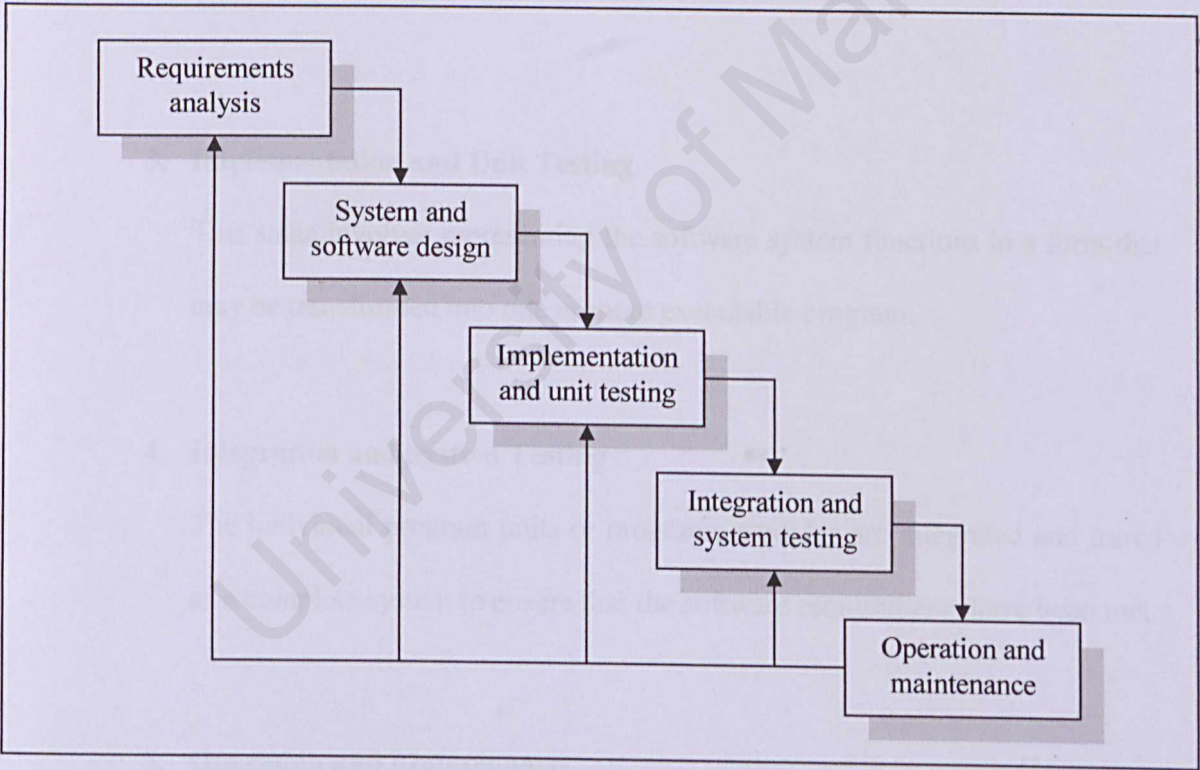


Figure 3.1: The Waterfall Model

The 5 stages of the Waterfall Model are discussed below:

1. Requirement Analysis

The concept, purpose and functionality of LMMT are identified and defined. After finish this stage, it comes to system design.

2. System and Software Design

Under this stage, we will begin the software design stage, where it will establish overall system architecture.

3. Implementation and Unit Testing

This stage involves representing the software system functions in a form that may be transformed into one or more executable program.

4. Integration and System Testing

The individual program units or programs modules are integrated and tested as a complete system to ensure that the software requirements have been met.

5. Operation and Maintenance

After testing, the system can be installed to be fully utilized. Maintenance for the system has to be done from time to time.

The reasons for choosing Waterfall model as the development methodology are:

- **System are visible**

For a relatively big system, documentation of the project is important. It can be used as a reference in the future. The software engineer can be very clearly stated at what stage the software process is currently in. This will let them easily manage the software process.

- **System are well structured**

As what we know, the software processes are designed at the beginning of the software development. What we needs to then are developed the system step by step according to the structure of the outlined.

- **System are predictable**

Waterfall model is emphasizing on planning rather than rapid development. For this purpose, it is easier to estimate the cost needed to develop the system.

- **System are ease to use**

No special skills are required in this approach. This is because all the outlined of the system is draft out. What the developer needs to do then is done according to what was already structured out.

3.3 Methodology Suitability to Project Domain

If we look back, the waterfall model main strength is on its simplicity. Simplicity means that the development methodology is simple and easier to understand. Hence we can have a better and clearer guideline on what we shall do during development process. Imagine if we are going to adopt more complex methodology which eventually will lead to more confusion and therefore we cannot have a clear understanding on what we should do during the development process. With the adoption of the waterfall model, we just need to emphasize on one stage at one time and do not need to think of the next stage before the current stage is completed and therefore allowing us to have more focus and attention on what we are currently doing instead of over burdening ourselves with the upcoming stage.

By looking back at the model, we can say that although we have proceed to the next stage, if we found that there is something in the previous stage which has been left out we can still go back to the previous stage to rectify the problem. Such feature is important especially if the user at the development time wanted to add or remove certain functionality which other methodology cannot offer. So we can say that the waterfall methodology offers us a backward feature whereby we can rectify errors or incomplete of certain criteria in the previous stage although we have proceeded into the next stage.

Simplicity is also essential when we are to present or explain or development progress to the user especially those who are not familiar with software development cycle.

3.4 Information Gathering Method

For information-gathering phase, there is no underlying standard or procedure to be followed strictly as each single project is unique and data-gathering may be vary to suit the needs of each particular project. However, there are a certain number of methods that are commonly used in gathering-information such as collecting hard data like written documents or reports, interviewing, using questionnaires, observation and sampling.

As for this project, due to cost and tight schedule constraints as well as the difficulties in finding and getting domain experts whom are willing to help, method such as interviewing becomes the intermediary who obtained the user's requirement from the real estate company. The main data sources for system analysis were written documents, reference books, observation and other sources from the Internet.

3.4.1 Written Documents and Reference Books

I have done some reading through printed documents such as books, magazine and journal to gather information about users' needs, system requirement and also technical requirement for the proposed system. I have go to the library and find some books, magazine and journals that are related to network monitoring and management tool to learn about the background in that can help me to understand more about the system proposed.

Besides that, I also get more information about today computer technology and updated computing knowledge from PC Magazine and also from the newspaper such as In-Tech from Star.

3.4.2 Internet Research

Research for this project was also done via the Internet. The result from the research from this research has been elaborated in more detail in Chapter 2. From this research, the Internet is used as the main resource for referring to any ambiguities that might arise during the entire development period. Through the Internet, I manage to collect some idea from the similar system such as LMMT that are developed by other company

From these researches, I get a lot of software, tools, programming languages and server-client computing knowledge. From the help file that can be found prepared, I can learn more about those software and programming tools and this help me to apply it in the proposed system. All of these have been described in more detail in Chapter 2.

3.4.3 Brainstorming

During the requirement elicitation, I meet together with my supervisor to discuss about the proposed system. During this stage, we generate as many ideas as possible without any analysis until all the idea has been exhausted.

3.4.4 Observation

Several existing monitoring and management tool have been reviewed. By using this technique, the methods used by those systems have been observed. Each system uses different method to give users more flexibility.

I also tested using those existing system and learn about their functionality and also found out the problems faced by each system. By reviewing those current systems, I learn more about the users' requirement, systems requirement and also the technology needed to develop the proposed new system.

CHAPTER 4: SYSTEM ANALYSIS

4.1 Introduction

The next step after the literature search and review is to perform a detailed analysis. The main purpose of the system analysis phase is to learn exactly what takes place in the current system, and to determine and fully document in detail what should be included.

Through system analysis, the programmer may manipulate system components toward the goal of improving the overall system. Furthermore, the programmer can determine types of functional requirements and non-functional requirements for the system.

4.2 Objectives of System Analysis

Following are some of the objectives of the analysis:

- To study the problem faced by the user.
- To study the problem and find out the best solution to reduced it.
- To acquire knowledge on how this system will be developed with the new emerging technology.
- To choose the development tools for the new system.
- To identify the major modules to be included in the system.

4.3 System Requirement Analysis

A requirement can be categorized as functional requirements and non-functional requirements.

4.3.1 Functional Requirements

Functional requirements describe the system's services and functions that provide for the users.

❖ General module

- *Administrator login*: administrator login through this dialogue box in order to enter the main system. Authentication is emphasizing here. Therefore, each administrator must have their own login ID and password.
- *Password setting*: administrator can change their password here.

❖ Monitoring module

- *Bandwidth monitor*: monitor the bandwidth of local network.
- *Computer browser*: list the connected PCs within the local network.
- *Port monitor*: monitor all active port.
- *TCP/IP & UDP monitor*: statistic of TCP/IP and UDP
- *Packet Sniffer*: Sniff incoming and outgoing packet within the local network.

❖ Management module

- *TCP Servers Eye*: remotely monitor and manage TCP service port.
- *Remote Shutdown/Reboot*: Shutdown or reboot PCs within the network
- *Remote User Information*: retrieved remote user information
- *Remote Management*: remote processes and services management

4.3.2 Non-functional Requirements

Non-functional requirements are those constraints placed on the services or functions offered by the system (for instance, the required response time), or on the development process (such as the use of a specific language standard). These are important to ensure the quality of the system.

- **Flexibility**

The system should have the capability to take advantage of the new technologies and resources. The system should be able to adapt and implement in the changing environment.

- **Reliability**

A system is said to have reliability if it does not produce dangerous or costly failures when it is used in a reasonable manner. The system should be reliable in perform its intended functions and operations accurately. For example, whenever a button is clicked, the system should able to respond and execute particular

function accordingly such as generate some messages to inform the user what is happening.

- **Usability**

The system should be developed in such a way that it is easy to use. It should be able to enhance and support rather than restrict the processes. Besides, the interface should be self-explanatory and consistent with other application in the system environment.

- **User friendliness**

A good flow of navigation is important to help and guide users on navigating with little effort through hyperlinks and procedure steps. Good interface is able to improve interaction between the users and the system.

- **Efficiency**

The system should be called or accessed in an unlimited number time to produce expected outcome or output at a creditable pace or speed.

- **Manageability**

The system should be easy to manage and handle to ensure that maintenance can be done regularly. Besides, it should enable the evolutionary of the system easy to be done and making the enhancement works simpler.

- **Correctness**

The system should be built according to the user requirements and specifications. It must meet its objectives and mission.

- **Maintainability**

The system where the software should be able to be understood, corrected, adapted and allow enhanced in the future.

- **Expandability**

The degree to where the architectural, data, or procedural design can be extended.

- **Security**

The system should involve the capability in performing authentication and authorization of valid users. The system should mandate a user to input a username and password before being allowed access into the system. This is important to provide security characteristic to privacy and confidential data, in order to avoid improper admission of eavesdroppers or hackers to the system.

4.4 Limitation of the LMMT

- Does not produce detail reports.
- Sniffing and monitoring session can not be recorded regarding specific time.
- All connected computer must comprise shutdown/restart privilege in order to remotely done.
- Remote processes and services management require local credential authorization.
- Monitoring and management limited only to local area network.

4.5 Choices of Programming and Technologies

An analysis has been carried out in order to select the most appropriate programming and technologies tools that suit the requirements of this system. These tools include the entire platform, technologies and programming languages. Besides considering the suitability of the tools to the requirement, the tools used must be able to support each other.

4.5.1 Programming Software Chosen

Programming software is one of the most important features that have to be thought off by the developers in developing a system. There are many software available in the market and each of these development software have their own strength and weaknesses. The ability and constraints of those programming software is an important factor for us to decide which software is the most suitable to be used.

By choosing suitable software to develop the system, it will simplify the work done by the developer as well as reduce the time needed to build the system. After doing some research on the software available in the market, the programming software that will be used to develop the system is:

Visual Basic 6.0

Visual Basic is an extremely powerful, full-featured application development tool that exploits the key features of Microsoft Windows. It is easy to use through a graphical interface. Applications can be built in the short time by using it. It can be used to create several types of applications; such as Client applications, Office applications, Client-Side applications, Group-Ware applications, ASP Web-based applications and server side applications. Therefore, it can be widely used in developing different types of applications.

Using Visual Basic, it will remove the need to do traditional window programming styles. Programmers only need to plan their program's logic and the design of the codes well without needed to know how to build the interface components, such as frames, command buttons and so on.

4.5.2 Monitoring and Management Technologies Chosen

Monitoring and management technologies consider the most important part in the development of LMMT. After doing the research, the best technologies that suit the proposed system is:

WMI

Windows Management Instrumentation (WMI) can be used in all Windows-based applications, and is most useful in enterprise applications. WMI is designed for

programmers who use C/C++, the Microsoft Visual Basic application, or a scripting language that has an engine on Windows and handles Microsoft ActiveX objects.

WinPcap

WinPcap is a free, public system for direct network access under Windows. The purpose of WinPcap is to give this kind of access to Win32 applications; it provides facilities to:

- capture raw packets, both the ones destined to the machine where it's running and the ones exchanged by other hosts (on shared media)
- filter the packets according to user-specified rules before dispatching them to the application
- transmit raw packets to the network
- gather statistical values on the network traffic

In order to use WinPcap in Visual Basic, we need a wrapper. **VB.PCAP** is an OPEN SOURCE project that aims to develop a fast and simple Packet Capture engine for Visual Basic. So, **vbpcap.dll** is a wrapper for the **packet.dll** in WinPcap.

Windows Socket (Winsock)

Winsock control is great application programming interfaces (API) provide by Microsoft. It's meant for the development of monitoring and management application which covers variety of implementation regarding the networking environment.

4.5.3 Development Platform Chosen

Development Platform is one of the most important features in the development of the proposed system. There are many different type of development platform. Each of these development platforms has their own strength and weaknesses. In choosing the development platform, the function of the development platform have to been reviewed in detail to make sure it can support all the tools used and also function that will be carried out by the proposed system. After doing some research, Windows XP Professional is acknowledged to be the development platform in this project.

Windows XP Professional

Windows XP Professional is provided with the technology foundation to grow business, whether the requirements are simply sharing information or a printer with co-workers or more advanced needs such as accessing office computer remotely and hosting a business Web site.

Reasons why Windows XP Professional is chosen:

- **A networked foundation**
- **Sharing Information**
- **Keeping data safe**
- **Connecting remotely**

4.6 Hardware Requirement

Basically, the requirement for LMMT is as follow:

Windows XP Professional Operating System

- ❖ Pentium 400MHz and above (or equivalent)
- ❖ Memory – 128Mb RAM or above
- ❖ 4.0GB hard disk or above
- ❖ Network card

Windows 2000 Professional

- ❖ Installed with latest service pack
- ❖ Pentium 266Mhz and above (or equivalent)
- ❖ Memory – 64Mb RAM or above
- ❖ 4.0GB hard disk or above
- ❖ Network card

4.7 Summary

In this chapter, study has been carried out on the proposed LAN Monitoring and Management Tool (LMMT) to find out the suitable methodology used. Various techniques of gathering information have been reviewed to gather sound and good information as guidance to the proposed system. System requirements such as functional requirements and non-functional requirements have been investigated and analysed.

The development programming and technologies tools have been rectified as well. As a result, Windows XP Professional has been chosen as the system platform, Visual Basic as the programming language, WinPcap as the monitoring technology and WMI as the management technology.

CHAPTER 5: SYSTEM DESIGN

5.1 Introduction

System design is a creative process of transforming the problem into solution and the description of the solution. The goal of system design is to translate the requirements defined during the system analysis phase into a working model or representation of an entity that will be built later. During this phase, quality is fostered. System design involves designing of program and user interfaces. System design has to go through a thorough modification and testing before coming to a complete system. Amendment has to be done on every occurrence of mistakes especially in coding and user interfaces design. Under this chapter, the system design will be discussed into the following few components:

- System Architecture Design
- Graphical User Interface Design

5.2 System Architecture Design

5.2.1 Context data flow diagram

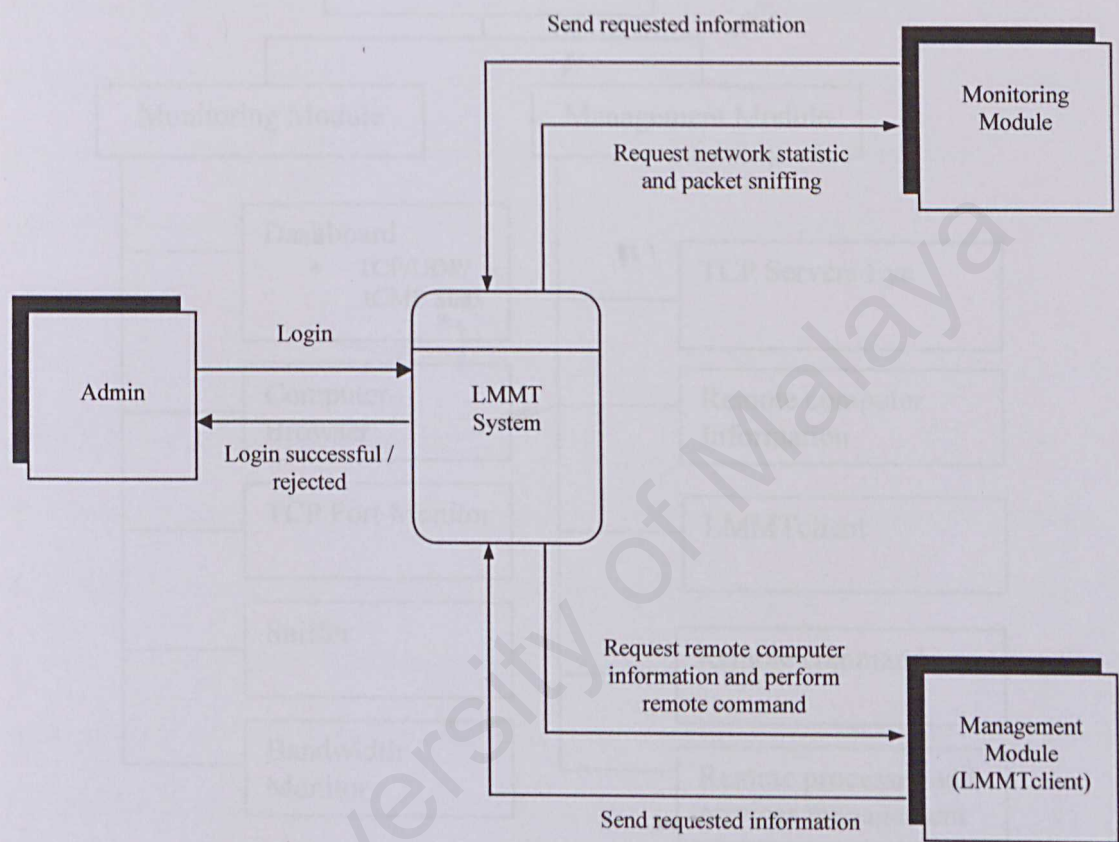


Figure 5.1: Context DFD

Figure 5.1 shows the context diagram of the LAN Monitoring and Management Tool System. The LMMT system is a system, which create the interaction between the administrator and the monitoring and management module. The modules that will be developed in this system will act as the method to create this interaction in order to make the system request become smooth and easy.

5.2.2 System structure chart

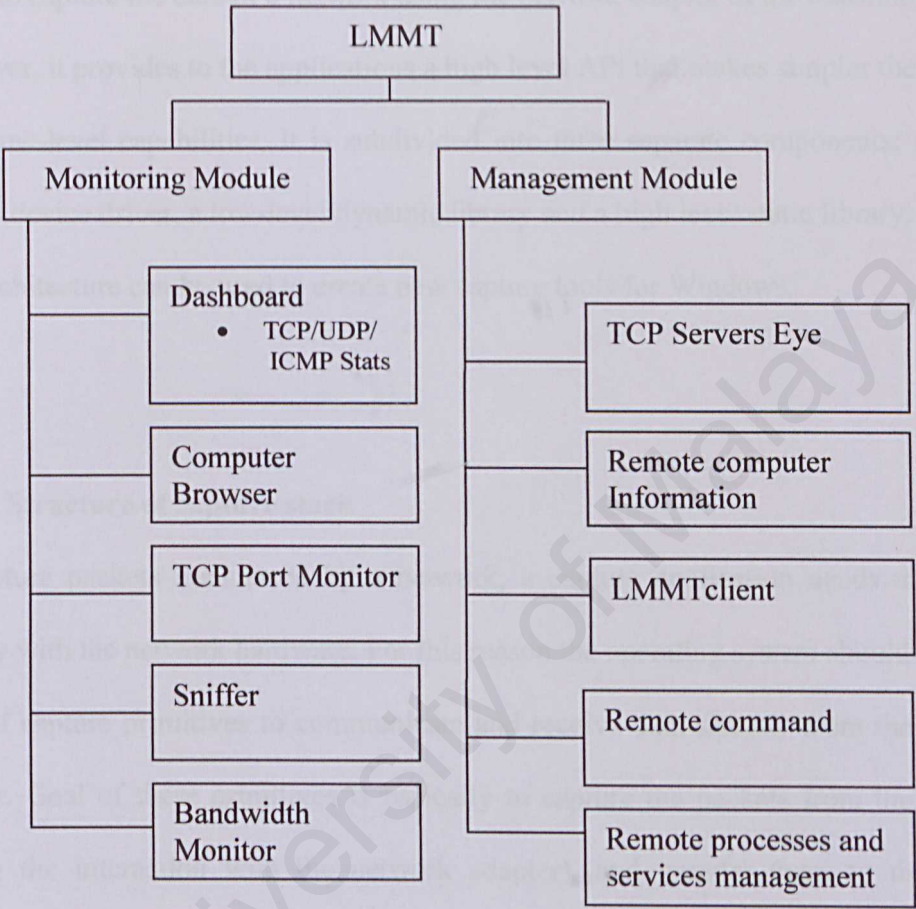


Figure 5.2: System Structure Chart for LMMT

This system structure chart performs different system functions in LAN Monitoring and Management Tool System.

5.2.3 WinPcap Architecture

WinPcap is an architecture that adds to the operating systems of the Win32 family the ability to capture the data of a network using the network adapter of the machine.

Moreover, it provides to the applications a high level API that makes simpler the use of its low-level capabilities. It is subdivided into three separate components: a packet capture device driver, a low-level dynamic library and a high level static library.

This architecture can be used to create new capture tools for Windows.

5.2.3.1 Structure of capture stack

To capture packets transferred by a network, a capture application needs to interact directly with the network hardware. For this reason the operating system should offer a set of capture primitives to communicate and receive data directly from the network adapter. Goal of these primitives is basically to capture the packets from the network (hiding the interaction with the network adapter), and transfer them to the calling programs.

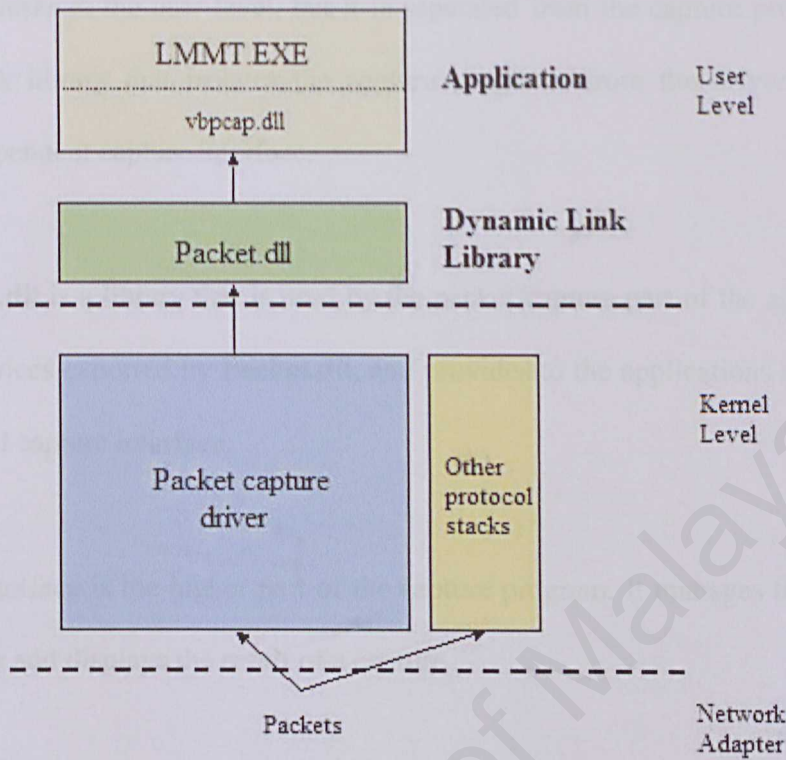


Figure 5.3: Structure of the capture stack

At the lowest level there is the network adapter. It is used to capture the packets that circulate in the network. During a capture the network adapter usually works in a particular mode (‘promiscuous mode’) that forces it to accept all the packets instead of the ones directed to it only.

Packet Capture Driver is the lowest level software module of the capture stack. It is the part that works at kernel level and interacts with the network adapter to obtain the packets. It supplies the applications a set of functions used to read and write data from the network at data-link level.

Packet.dll works at the user level, but it is separated from the capture program. It is a dynamic link library that isolates the capture programs from the driver providing a system-independent capture interface.

The **vbpcap.dll** is a library that is used by the packet capture part of the applications. It uses the services exported by **Packet.dll**, and provides to the applications a higher level and powerful capture interface.

The user interface is the higher part of the capture program. It manages the interaction with the user and displays the result of a capture

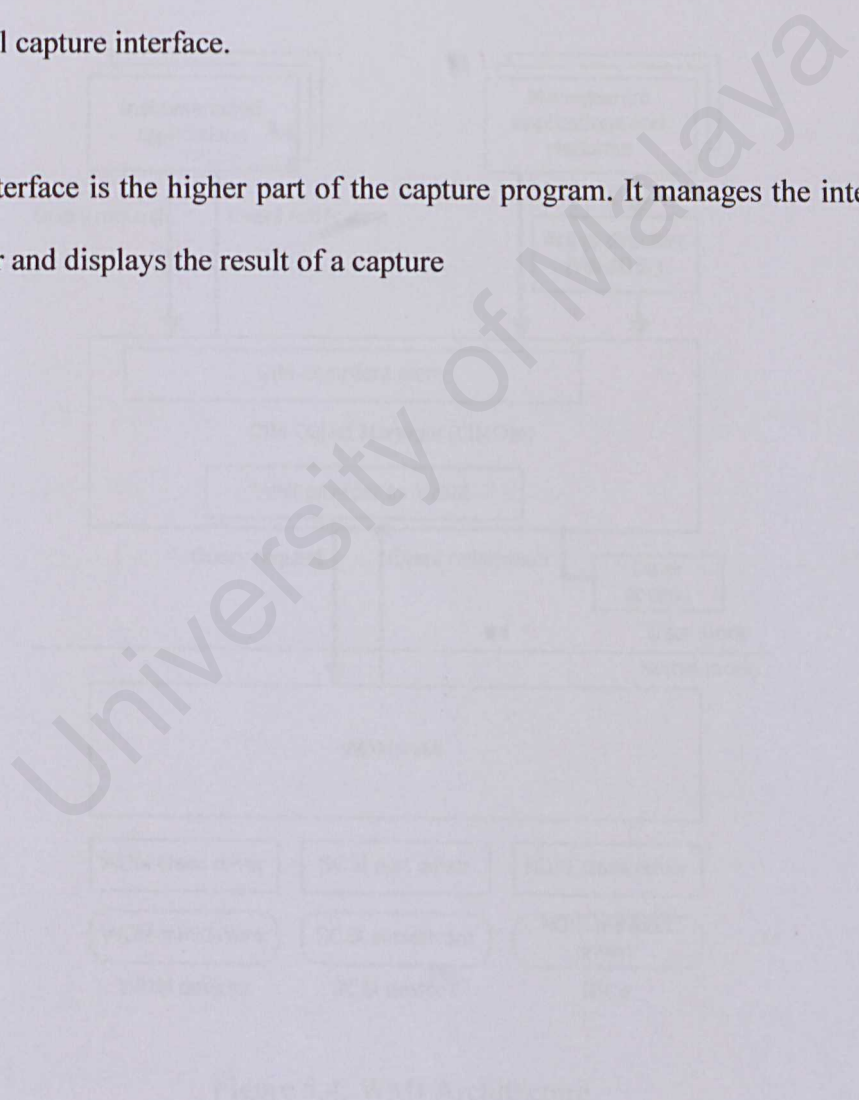


Figure 5.4: WPA Architecture

5.2.4 WMI Architecture

WMI is a unifying architecture that allows access to data from a variety of underlying technologies. These technologies include, for example, WMI extensions to WDM, the Desktop Management Interface (DMI), and the Simple Network Management Protocol (SNMP).

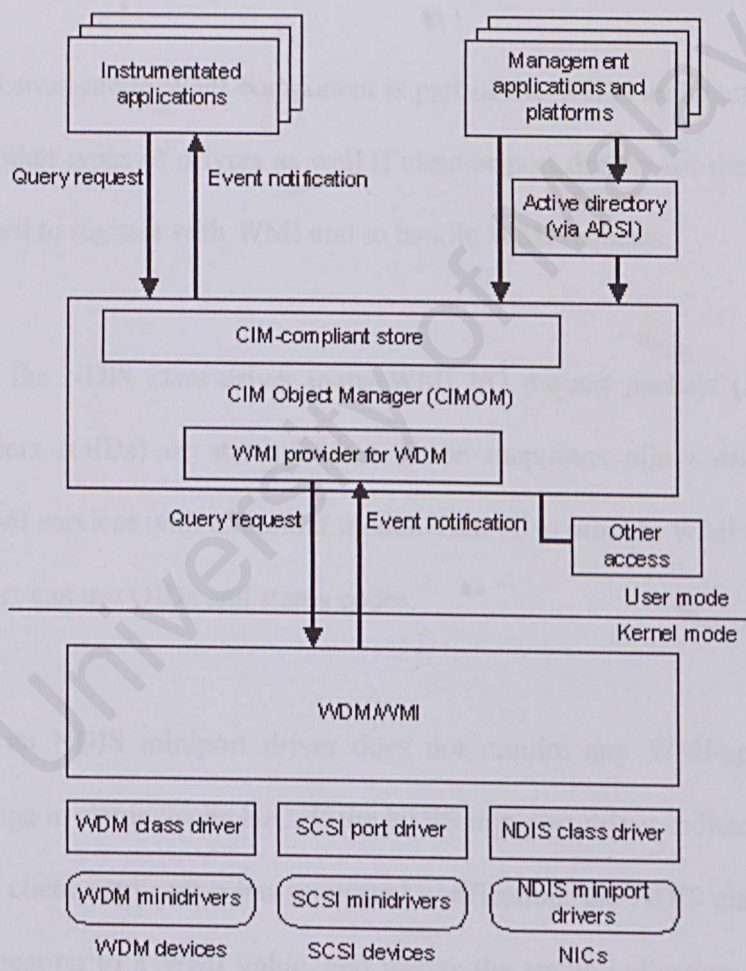


Figure 5.4: WMI Architecture

WMI provides a three-tiered approach for collecting and providing management data:

- A standard mechanism for storing data (a CIM-compliant data repository)
- A standard protocol for obtaining and distributing management data (for example, COM and DCOM)
- A WMI provider (a Win32 dynamic-link library (DLL) that supplies instrumentation data)

Although the kernel-mode WMI component is part of the WDM architecture, it can also be used with other types of drivers as well if class or port drivers for those driver types are implemented to register with WMI and to handle WMI requests.

For example, the NDIS class driver maps WMI I/O request packets (IRPs) to either object identifiers (OIDs) or status codes. These mappings allow driver writers to implement WMI services with a familiar model. That is, to provide WMI services, NDIS miniport drivers can use OIDs and status codes.

For example, an NDIS miniport driver does not require any WMI-specific code to indicate a change in status for its NIC. If the NDIS miniport driver indicates status about which a WMI client application has requested notification, the NDIS class driver maps that status indication to a WMI value, and passes the status indication to WMI. WMI then passes the status indication to the WMI client application.

5.3 Graphical User Interface Design

The LAN Monitoring and Management Tool (LMMT) is a stand-alone based system. Thus the Graphical User Interfaces (GUIs) play an increasingly important role of the system. The system will be designed so that it is easier to use, and a user do not have to require training and minimal support.

Among the important points that are crucial in developing a good GUI:

- Easy to navigate
- Consistency, clarify and easy to comprehend
- Color and font tips

LMMT expected outcome (GUI)

Administrator login



The image shows a screenshot of a Windows-style dialog box titled "Login". Inside the dialog, there is a section titled "LOGIN" and a main heading "LMMT Administrator Login". Below this, there are two text input fields: one labeled "Username" and one labeled "Password". At the bottom right of the dialog, there are two buttons: "Proceed" and "Exit". The dialog box has a blue border and standard Windows window controls (minimize, maximize, close) in the top right corner.

Figure 5.5: Administrator login dialogue box

LMMT main screen (LMMT Dashboard)

This is the main screen of LMMT. It will appear if the authentication is successful. From this screen, we manage to get the information of the local network setting, the statistic of TCP/IP/UDP/ICMP, and the network bandwidth.

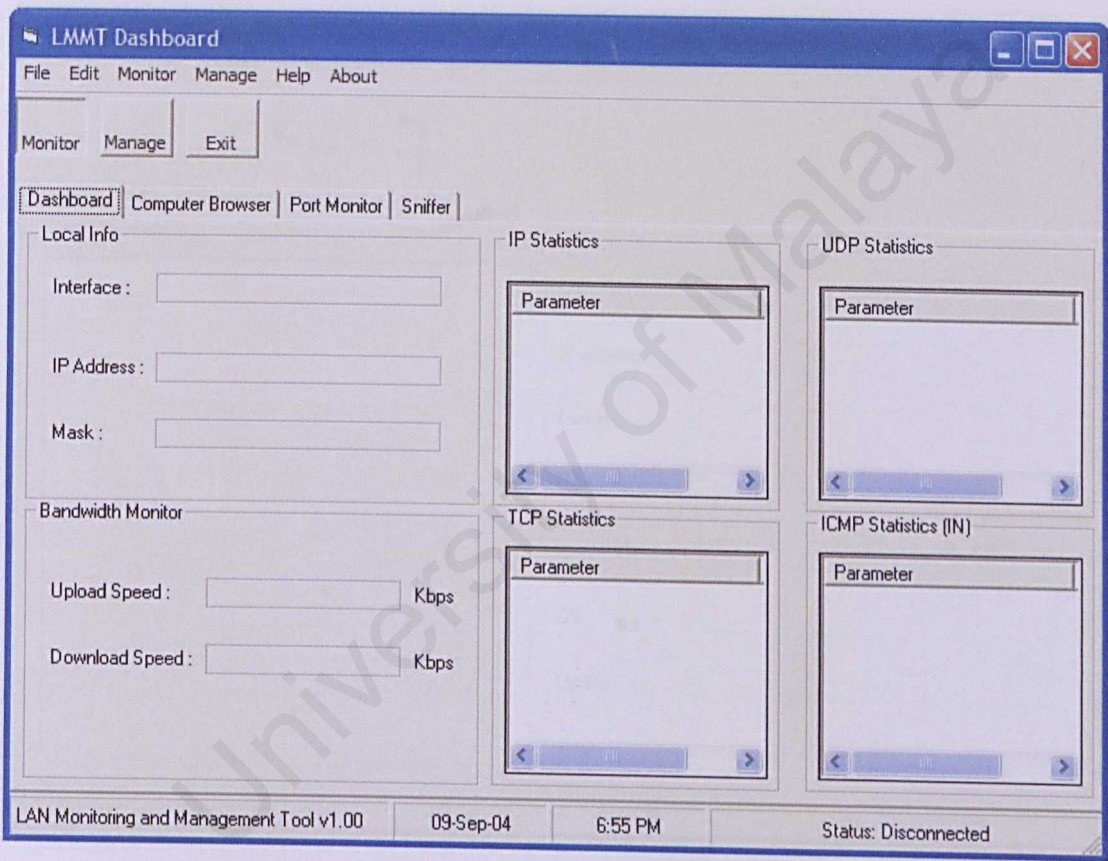


Figure 5.6: LMMT Dashboard

Computer browser screen

This screen will enumerate the entire connected computer within the LAN in a tree-view list. Beside that, it also provide the information of each connected computer such as the computer name, IP address, operating system and many more.

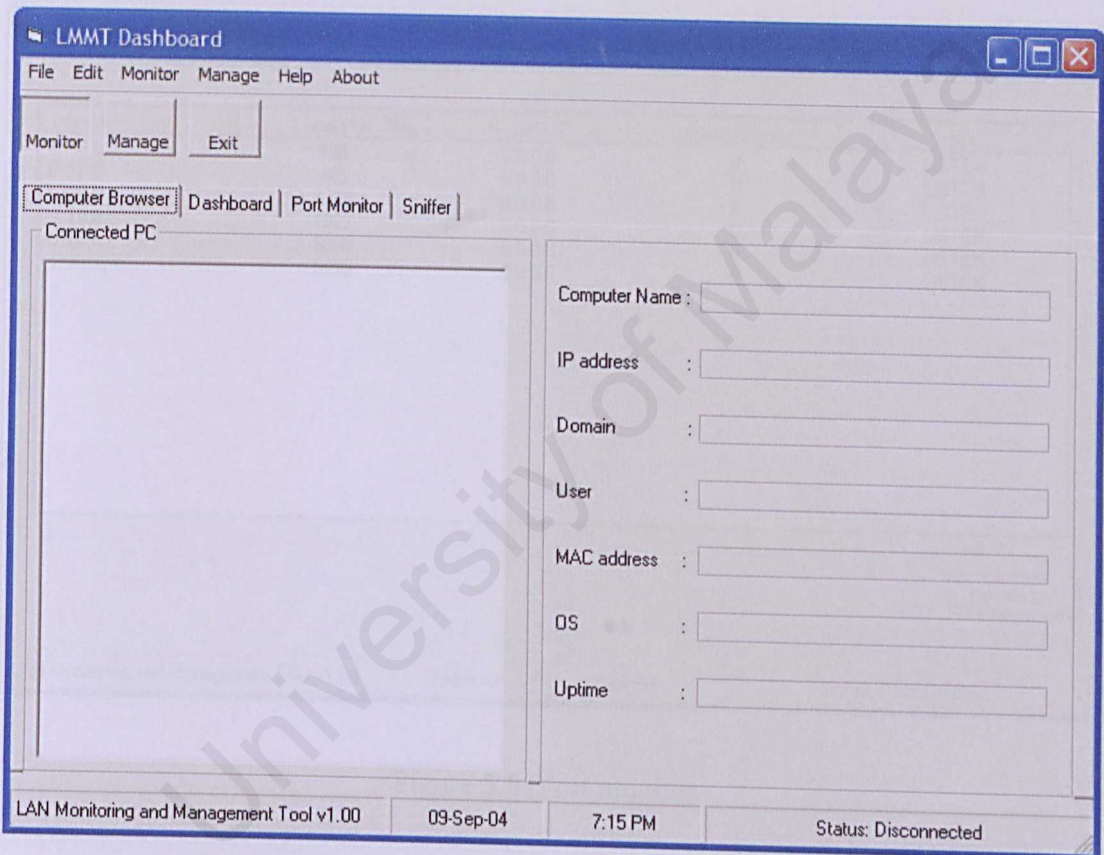


Figure 5.7: Computer browser

Port monitor screen

This screen will provide information on local and network port.

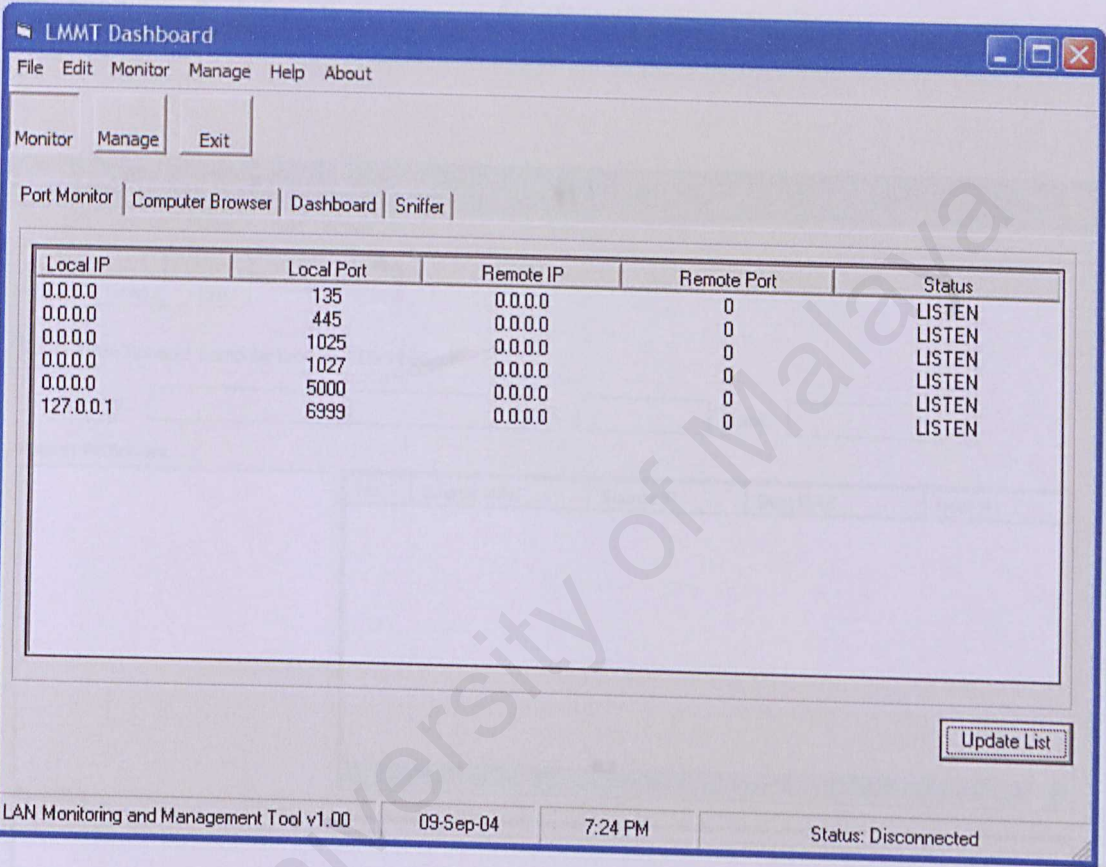


Figure 5.8: Port monitor

Sniffer screen

Sniffer screen is one of the critical functions in this project. The sniffer can perform the sniffing activity of the whole LAN. It will read the packet header and the raw data and display it in the proper way.

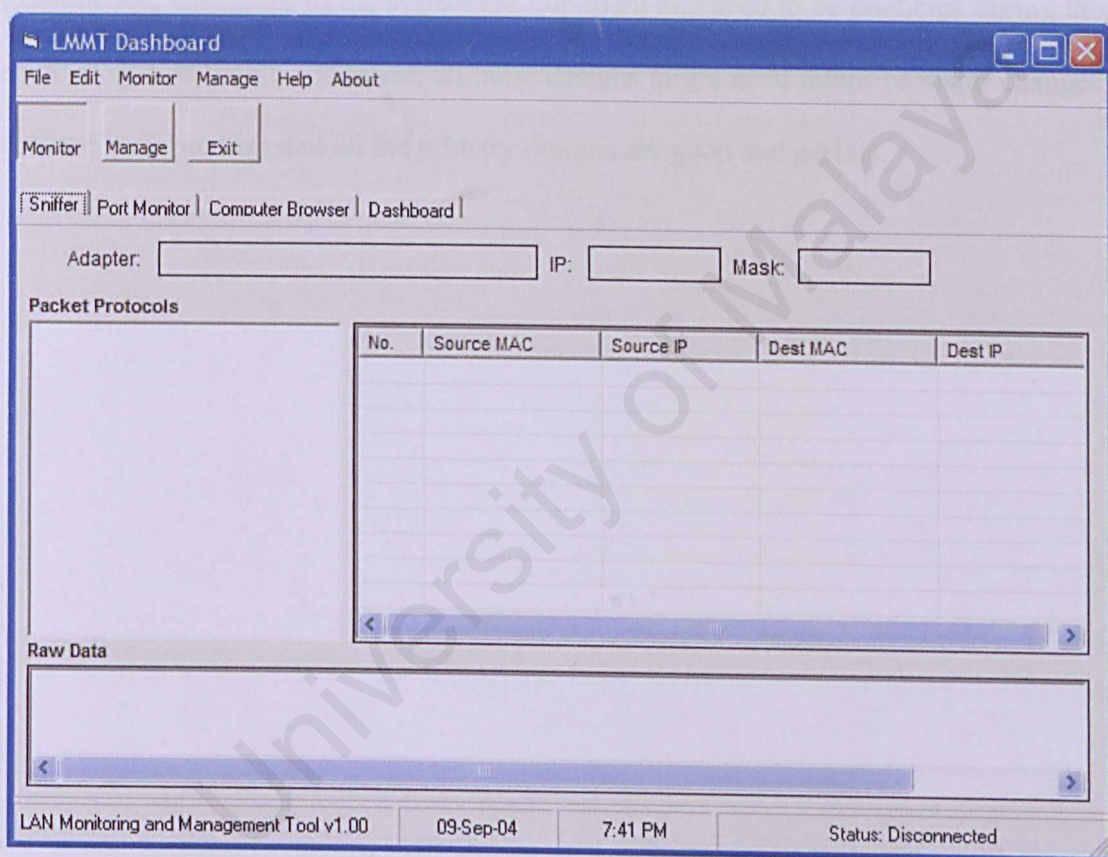


Figure 5.9: Sniffer

5.4 Summary

System design is an important aspect in system development cycle. Things that need to be taking care are program design, which comprises of many modules defined by their functionality such as monitoring module and management module. Graphical user interface design is other important parts in system design, which required extra examination. Outcomes of the system are important and need to be predicted during this stage of system design. However, all these designs might need minor or major changes, as there is no promise that all the primary designs are good and perfect.

University of Malaysia

CHAPTER 6: SYSTEM IMPLEMENTATION

6.1 Introduction

System implementation is the process that converts the system requirements and designs into program codes using selected programming language. It focuses on implementing the solution as software. This stage involves the application implementation.

6.2 Development Environment

Development environment has certain impact on the development of a system. Appropriate hardware and software chosen will not only help to speed up the system development but also determine the success of the project. Below are the lists of hardware and software tools being used to develop the entire system.

6.2.1 Hardware Requirement

Hardware requirement during system development process are as below:

- Pentium 400 MHz processor and above
- 128MB RAM and above
- Hard Disk capacity 4 GB and above
- Other standard desktop PC components
- Network connection

6.2.2 Software Requirement

During the system development process, software tools as determine during system analysis are being used. However, there is minor modification at the software version and also few additional tools are selected. Below depict the software tools used to develop the system.

Technologies/Software	Description	Purpose
Windows XP Professional	Operating System (OS)	System Requirement
Microsoft Visual Basic 6.0 (SP5)	Coding	System Development
Adobe Photoshop 7.0	Skinning	System Development
WinPcap 3.0	Packet Capturing Driver	System Requirement

Table 6.1: Development Tools

6.3 Platform Development

Platform Development involves all the development environment set-up processes from the operating system, driver to the software development tool. A brief installation steps and requirements are discussed here.

6.3.1 Development Environment Setting

As mention before, Microsoft Windows XP Professional was selected as the operating system (OS) Firstly, the OS is install then follow by the Adobe Photoshop 7.0. After these steps, WinPcap 3.0 is installed to provide the driver for packet capturing. Finally, Microsoft Visual Studio 6.0 (included Visual Basic 6.0) was installed as the programming tools for this system.

6.4 Development of the System

6.4.1 Coding Approach

There are properly two types of coding approach, namely top-down and bottom-up. The bottom-up coding is based on coding the lower-level modules initially and leaving the high-level modules merely as skeletons that are used to call the lower modules, whereas the top-down approach is the reverse of the bottom-up approach.

The LMMT system was developed modularly using the bottom-up approach. Each lower-level function and procedure was developed individually which are then integrated into appropriate high-level modules accordingly. Some bottom-up advantages:

- Testing can be conduct on some of the modules while the others are still under construction.
- Critical functions can be coded initially to test their efficiency.
- Faults are easier to be detected.

6.5 Coding Documentation

Program documentation is a set of written descriptions that explain to readers about what the programs do and how they do it. This documentation may categorize as internal documentation whereas the others are known as external documentation. Internal documentation is descriptive material written directly within the code. It contains information directed at someone who will be reading the source code of the program. Comment within the code is the example of internal documentation. It provides line-by-line explanation of what the program is doing such as the description about data structures, algorithms and so on. Besides, it also breaks the code into phases that represent major activities.

Different programming language uses distinguish comment syntax. For instances, in Visual Basic coding, each comment tag is proceeding with a single code ('). Any statement at the same line after a single code will be ignored during execution time. This comment line will appear in green colour.

Example:

```
' This is a comment.
```

Apart from the comment usage, meaningful variable names and statement labels also being used to increase code readability.

6.6 Module Implementation

There are two main modules in LMMT System: Monitoring Module and Management Module. Each module is consists of several sub-module or functions.

Beside those module, administrator authentication also been implemented. Mandate for the administrators before they are granted the access to the functions in Administrator, users have to enter their password. The system also provides the password setting for the administrator to change their password.

6.6.1 Monitoring Module

Below are five sub-modules in Monitoring Module.

- i. **Dashboard:** As the main tab of the LMMT System. It shows the statistic for TCP, UDP and ICMP.
- ii. **Computer Browser:** Tab that show the connected PC in a LAN.
- iii. **TCP Port Monitor:** Tab showing the port connection between PC's in the LAN and internet and the state for the connection.
- iv. **Sniffer:** Sniff packet in the LAN and provide information and raw data of the packet.
- v. **Bandwidth Monitor:** Show the incoming and outgoing bandwidth for the local PC and record the highest incoming and outgoing bandwidth.

6.6.2 Management Module

- i. **TCP Servers Eye:** Administrator needs to input the server IP address and the services to monitor. This form will inform the administrator if the server services are alive or down.
- ii. **Remote Computer Information:** Here, administrator can get the managed PC's username, OS platform and Windows running time. The remote PC's must be installed with LMMTclient before LMMT can retrieve the information.
- iii. **Remote Command:** Administrator can perform the remote command here. Commands that are available in this version of the system are remote shutdown and reboot. The remote PC's must be installed with LMMTclient before the command can be performed.
- iv. **LMMTclient Module:** In this module, the complete coding of the client will be place on the remote computer to be managed. In contain the WMI Script for reboot and shutdown the PC's, and coding for retrieve the information from the managed PC's.
- v. **Remote Processes and Services Management:** This module cover up the service and process management of remote PC. User can kill and create process and start or stop services on remote computer.

6.7 Summary

During system implementation, system requirements and designs were converted into program codes. Besides, it also involves development environment setting such as the operating system and the driver installation. Several software tools were used to deploy the design into machine-readable language and then in turn to produce the required applications.

CHAPTER 7: SYSTEM TESTING

7.1 Introduction

After coding the program components, we usually examine the code to spot faults and eliminate them right away. Testing is a process that focused on finding faults. Therefore, we consider a test successful only when a fault is discovered or a failure occurs as a result of our testing procedures. Fault identification is the process of determining what fault or faults caused the failure, and fault correction or removal is the process of making changes to the system so that the fault are removed.

Software testing is crucial for software quality assurance and represents the ultimate review of specification, design, and code generation. In this chapter, software testing fundamentals, testing strategies and debugging methods will be presented.

7.2 Type of Fault

Faults can be categorized as algorithmic faults, syntax faults and documentation faults. Algorithmic fault occurs when a program algorithm or logic does not produce the proper output for a given input because something is wrong with the processing steps. These faults are sometime easy to spot by reading through the program (call desk checking) or by submitting input data from each of the different classes of data that we expect the program to receive during its regular working.

Syntax fault can be checked while parsing for algorithmic faults. This will ensure that the construct of programming language is used properly. On the other hand, documentation fault occurs if the documentation does not match what the application does, and such faults can lead to other faults later because of the wrong implementation.

7.3 Testing Strategy

Testing is a process of exercising or evaluating a system by manual or automatic means to verify that it has satisfied requirements or to identify differences expected and actual results. Testing can uncover different classes of errors in a minimum amount of time and with a minimum amount of effort. There are three types of testing strategies, namely, unit testing, integration testing and system testing.

7.3.1 Unit Testing

Unit testing is the first approach in system testing. Each program component is tested on its own, isolated from the other components in the system. This process verifies that the component functions properly with the types of input expected from studying the component's design. Testing can start by examining the program code by reading through it, trying to spot algorithm, data and syntax faults. Test also can be performed by comparing the code with the predefined specifications and design to ensure that all relevant cases have been considered. Finally, test cases are developed ensure that the input is properly converted to the desired output.

7.3.1.1 Unit Testing Example

Here are few examples among a great number of unit testing cases that have been performed for this system.

Example 1

On system startup, LMMT should show the local hostname and IP address (not loopback address, 127.0.0.1).

Step	Test Procedure	Expected Output	Test Result Analyzing
1	Connect the network card to the LAN	Unit shows the correct hostname and IP address for local system	Successfully shows local hostname and IP address

Table 7.1: Unit Testing Sample 1

Example 2

Before allowing administrator to access the system, he or she has to undertake an authentication by entering password. Testing is carried out to ensure that the system perform verification properly and granted the access only to proper and valid users.

Step	Test Procedure	Expected Output	Test Result Analyzing
1	Click on the LMMT executable file	Display login dialogue box.	Link successful
2	Enter the invalid user password. Click on Proceed button.	Access denied and “Wrong Password. Please try again” message will be displayed.	Verification was successfully performed and expected output was accomplished.
3	Enter the valid password. Click on Proceed button.	Access granted and LMMT Dashboard (main page) will be shown.	Verification was successfully performed and LMMT Dashboard was displayed.

Table 7.2: Unit Testing Sample 2

7.3.2 Integration Testing

As each unit was tested successfully, these components are then being combined into a working system. In other words, integration testing is the process of verifying that the system components work together as described in the system and program design specifications. This integration is planned and coordinated so that when a failure occurs, idea of the cause can be gained.

Integration testing is used on this system to uncover errors associated with remote command. This testing will ensure that the command send from LMMT to the LMMTclient are successfully done.

Several integration testing approaches exists. However, Sandwich integration was used for this system. This approach combines top-down integration with bottom-up integration.

Testing started from the main management module section to the lowest level of the functions in the remote command and back to main management module section. This testing was performed several times to ensure the system working properly.

7.3.2.1 Integration Testing Example

Testing was performed several times on the same scenario as well as different situation. Therefore a lot of test case involved.

Example

For the TCP Servers Eye implementation, the user need to input the IP address or Hostname and the service port to monitor in the provide list. Testing was carried out to ensure that input is successfully added in the list. Table below shows the test case for this integration testing.

Step	Test Procedure	Expected Output	Test Result Analyzing
1	Click on “TCP Servers Eye” button in the main menu.	The form for TCP Servers Eye will be shown.	Link succeeded.
2	Input the IP address and select the port .Click “Add” button	Input will be listed in the list box.	Input successfully listed.
3	Click the selected item in the list. Click “Delete Entry” button	Item selected will be deleted from the list.	Item successfully deleted.
4	Click on “Monitor” button.	All item(IP address and Port) in the list will be monitor and result the State(either Alive or Down)	Monitoring succeeded and shows the state.
5	Click on “Stop” button.	Monitoring will be stopped and shows the last state.	Monitoring successfully stopped and shows the last state.
6	Click on “Close” button.	Close the TCP Servers Eye form.	Form successfully closed.

Table 7.3: Integration Testing

7.3.3 System Testing

Final testing procedure done is system testing. However, testing the system at whole is very different from previous unit testing and integration testing. The objective of unit testing and integration testing is to ensure that the code has implemented the design properly. In other words, the code is written to do what the design specifications intended. There are several testing scenario exists. Yet, this system was tested with Performance Testing and Load Testing.

7.3.3.1 Performance Testing

Performance Testing addresses the non-functional requirements of the application. Once the functions are convinced work as specified, the performance test compares the integrated components with the non-functional system requirements. The types of performance tests carried out for this application are:

- i. **Compatibility Tests**

This test was performed to find out that the module functions perform according to the requirements. Results clarify that the accuracy of the system.

- ii. **Security Tests**

This test ensures that the application fulfils the security requirements. Only the valid users are granted the access to the secure zone.

7.3.3.2 Load Testing

This test was carried out to make sure the system able to adapt in the higher load of bandwidth and user. Each functional module is tested to fulfill this requirement.

7.4 Summary

During system testing phase, several testing strategies were being used to ensure the system is integrated and developed successfully. Approaches were employed to recover faults in the system. Unit, integration and system testing has been carried out for this system. These testing approaches lead to delivering a quality system to users. The objective of a system will only achieve after all the careful testing done by different user with different aspects.

CHAPTER 8: SYSTEM EVALUATION

8.1 Introduction

At this point, the software development cycle of LMMT System is considered successfully achieved and implemented. The system is now ready for the evaluation and assessment concern. Several issues and reviews on the final system are explained in this section. System's features and strengths, system's limitation and constraints and lastly the future enhancement will be described.

8.2 System Strength

Followings are the features and strengths that can be found in the LMMT system:

- **Simplicity of user interface**

The graphic interface design of the system was designed to let the users feel comfortable and easy-to-use. The GUI ensured user friendliness.

- **Remote Monitoring**

TCP Servers Eye is a powerful remote monitoring tool which can monitor your TCP based services (FTP,HTTP and etc) without any interference from user and client program.

➤ Packet Capturing

LMMT perform well with the packet sniffer which can filter all the network traffic in the promiscuous mode. This packet sniffer works well if in the proxy server or connected to the span port of your main switch. It can monitor the distribution of packet between within the LAN and Internet.

➤ TCP Port Monitoring

TCP Port Monitor, monitors all your connection with the user within the LAN and their connection to internet. It shows the remote IP and remote port that established between the connections. The strength is, we can kill the connection as we wish.

➤ Security

Only valid administrators can gain access to the system and only he/she will have the power to perform the remote command (reboot, shutdown and kill connection).

➤ Event Logging

LMMT provide the capability of event logging for several crucial information that collected by the system for the purposes of reference.

➤ Remote Processes and Services Management

User can control services and processes on the remote computer.

8.3 System Limitation and Constraint

Even though there are many features provided by the system, it is still not perfect. Due to the problem of time constraint and technologies, some of the feature cannot be implemented. The limitation is as listed below:

➤ Multiple Account For Administrator

➤ Don't Support Multiple Administrator Account

Only one person with one correct password can access the system. LMMT just provide one administrator account.

➤ Specific User Packet Distribution

➤ Packet Distribution per Connected Computer Are Not Recorded

LMMT does not provide the function to record the packet distribution of each connected computer in the LAN.

➤ Alert Mechanism Will Be Local

➤ No Remote Alert Mechanism

Alert type for the TCP Servers Eye and the Bandwidth Monitor are categorized as local alert. It does not support remote alert (E-mail or SMS).

➤ Enable Local Service

➤ Save Function is Disable

Save function for details on sniffing and port monitoring are not available in this version of this system.

➤ Provide Remote Information

➤ Require Local Authorization to Perform Remote Process and Service Management.

8.4 Future Enhancement

Due to the limitations found in the LMMT System, enhancement will be applied to the system to improve its ability in the future.

➤ **Multiple Account For Administrator**

System can provide more than one account for the administrator. That's mean it can be more than one user who can interact with the system.

➤ **Specific User Packet Distribution**

System able to identify each sent packet by each computer to be recorded in the professional manner due to prevent from Denial of Service (DOS) attack.

➤ **Alert Mechanism Will Be Upgrade**

Local alert still be implemented but with the enhancement of remote alert. Alert through Email forwarding and Short Message Service will be available.

➤ **Enable Save Function**

System can save details on sniffing and tcp port monitoring for the purpose of proves and revisions.

➤ **Provide More Remote Information**

WMI features will be enhanced to gain more information on remote computer.

8.5 Problem Encountered

➤ Requirement Change From User

It is very difficult to develop and implement the system when the requirement changes very frequently. Sometimes it is easy to change the requirement, however, the coding need to be changed a lot in order to follow the new requirement.

➤ No End User Evaluation

Because of time limitation, to get an appropriate end user evaluation is a major problem. Only a few colleges and my supervisor has test end evaluate this system.

➤ Difficulties During Project Study and Requirement Analysis

This system involves a lot of business rules and model. Therefore, basic knowledge is needed as a foundation in building an application of this nature.

➤ Integration Problem

Due to the different approaches of the third party (WinPcap and WMI), constraints exist and integration is difficult to done in especially in the coding part.

8.6 Knowledge Gained

➤ The Important of SDLC and Software Engineering

System Development Life Cycle (SDLC) provides an effective guideline for software development. Each phase is highlighted and crucial to the development process. For example, system analysis is important in capturing user requirements, objective and the goal of the system. Any faulty occurs in this stage may delay or cause the failure to the whole system. The same vital role is applied to other phase as well.

➤ Development Tool Knowledge

During the system coding and implementation, a lot of knowledge and techniques in VBScript and Visual Basic are gained. By practically apply them in the application, it is able to improve the understanding about the languages themselves as well as their integration.

➤ Self Expression

Involvement and experiences gained during system development have provided the change for self-improvement and evaluation. System design and coding give a great chance to express my own opinions and ideas.

➤ Network Programming

Major coding implementations in this system are related to network programming. It strengthens my knowledge in the related programming field.

8.7 Summary

System evaluation is needed to ensure its objectives and intended functions have been achieved. This chapter has covered all the aspect of evaluating application software. At this point, it also implies the conclusion of the project.

REFERENCES

1. Chappell, David. (2000). Understanding Microsoft Windows 2000 distributed services. USA: Microsoft Press.
2. Sommerville, Ian. (1995). Software engineering (5th ed.). England: Addison-Wesley Publishers Ltd.
3. Pressman, Roger. S. (1998). Software engineering: A beginner's guide. USA: McGraw-Hill.
4. Whitten, J.L., Bentley, L.D., & Dittman, K.C. (2000). System analysis and design methods (5th ed.). New York: McGraw-Hill.
5. Deitel, H.M, Deitel, P.J, & Nieto, T.R (1999). Visual Basic 6: How to program. New Jersey: Prentice Hall
6. <http://www.winpcap.polito.it>
7. <http://www.microsoft.com/technet/prodtechnol/winxppro/evaluate/netwkxp.mspx>
8. http://msdn.microsoft.com/library/en-us/wmisdk/wmi/what_s_new_in_wmi.asp
9. <http://about.com/compute/od/basicnetworkinconcepts/>
10. <http://www.scit.wlv.ac.uk/rfc>
11. http://searchsecurity.techtarget.com/bestweblink/0,,sid14_tax281934,00.html
12. <http://netgroup-serv.polito.it/winpcap>

Appendix

1 Example Code: Encryption and Decryption of Password

Encrypt

Function Encrypt(What As String) As String

Dim Before\$, After\$, EpN%, Dracula%, Aeneima\$, DeMoNs\$

Before\$ = " ?!@#\$\$%^&*()_|0123456789abcdefghijklmnopqrstuvwxyz,-
~ABCDEFGHIJKLMNOPQRSTUVWXYZ¿¡²³ÀÁÂÃÄÅÖÓÔÕÖÙÚÛäåääää
ØŒ\$Ú¥"

After\$ = " ¿¡²³@#\$\$%^&*()_|01²³456789ÀbÁdÂÃghÄjklmÅÒÓqÔÕÖÙvwÛÜz.-
~,AääääFGHäJKåMNØŒQR\$TÚVWX¥Z?!23acefinoprstuxyBCDEILOPSUY"

For EpN% = 1 To Len(What)

Dracula% = InStr(Before\$, Mid(What, EpN%, 1))

If Not Dracula% = 0 Then

Aeneima\$ = Mid(After\$, Dracula%, 1)

DeMoNs\$ = DeMoNs\$ + Aeneima\$

End If

Next

Encrypt = DeMoNs\$

End Function

Decrypt

Function DeEncrypt(What As String) As String

Dim Before\$, After\$, EpN%, Dracula%, Aeneima\$, DeMoNs\$

Before\$ = " ĺi@#\$\$%^&*()_|01²³456789ÀbÁdÂÃghÄjklmÅÒÓqÔÕÖÙvwÛÜz.-
~,AàáâãFGHäJKåMNØŒQR\$TÚVWX¥Z?!23acefinoprstuxyBCDEILOPSUY"

After\$ = " ?!@#\$\$%^&*()_|0123456789abcdefghijklmnopqrstuvwxy.,-
~ABCDEFGHGIJKLMNOPQRSTUVWXYZĺi²³ÀÁÂÃÄÅÖÓÔÕÖÙÛÜäåääää
ØŒ\$Ú¥"

For EpN% = 1 To Len(What)

Dracula% = InStr(Before\$, Mid(What, EpN%, 1))

If Not Dracula% = 0 Then

Aeneima\$ = Mid(After\$, Dracula%, 1)

DeMoNs\$ = DeMoNs\$ + Aeneima\$

End If

Next

DeEncrypt = DeMoNs\$

End Function

2 Example Code: Event Logging

Sub

Public Sub LogEvent(WhatHappened As String)

Dim WhatToPrint As String

On Error GoTo ErrorH

Open "LMMTlog.Log" For Append As #500

WhatToPrint = Date & " " & time & " " & WhatHappened

Print #500, WhatToPrint

Close #500

Exit Sub

ErrorH:

Open "LMMTlog.Log" For Input As #500

Resume Next

End Sub

Implementation

```
LogEvent "Highest TCP Segment Received :" & ListView1.ListItems(10).SubItems(1)
& " , " & "Highest TCP Segment Sent :" & ListView1.ListItems(11).SubItems(1)
LogEvent "Highest UDP Datagram Received :" & ListView3.ListItems(1).SubItems(1)
& " , " & "Highest UDP Datagram Sent :" & ListView3.ListItems(4).SubItems(1)
LogEvent "Highest ICMP Message Received :" & ListView4.ListItems(1).SubItems(1)
& " , " & "Highest ICMP Message Sent :" & ListView9.ListItems(1).SubItems(1)
```



```

LogEvent "Highest Incoming Bandwidth  :" &
bandwidthform.RecordIncoming.Caption & " KBPS" & " , " & "Highest Outgoing
Bandwidth :" & bandwidthform.RecordOutgoing.Caption & " KBPS"
LogEvent "LMMT Exit"
LogEvent
"*****"

```

3 Example Code: TCP Port Monitor

```

Dim arrBuffer() As Byte
    Dim lngSize As Long
    Dim lngRetVal As Long
    Dim lngRows As Long
    Dim i As Long
    Dim TcpTableRow As MIB_TCPCROW
    Dim lvItem As ListItem

    ListView5.ListItems.Clear

    Me.MousePointer = vbHourglass

    lngSize = 0

    lngRetVal = GetTcpTable(ByVal 0&, lngSize, 0)

    If lngRetVal = ERROR_NOT_SUPPORTED Then

        'This API works only on Win 98//2000 and NT4 with SP4
        MsgBox "IP Helper is not supported by this system."
        Exit Sub
    End If

```

End If

ReDim arrBuffer(0 To lngSize - 1) As Byte

lngRetVal = GetTcpTable(arrBuffer(0), lngSize, 0)

If lngRetVal = ERROR_SUCCESS Then

CopyMemory lngRows, arrBuffer(0), 4

ReDim arrTcpTableRows(1 To lngRows)

For i = 1 To lngRows

DoEvents

CopyMemory TcpTableRow, arrBuffer(4 + (i - 1) * Len(TcpTableRow)),
Len(TcpTableRow)

If Not ((Check1.Value = vbUnchecked) And

(GetIpFromLong(TcpTableRow.dwRemoteAddr) = "0.0.0.0" Or

GetIpFromLong(TcpTableRow.dwLocalAddr) = "0.0.0.0" Or

GetIpFromLong(TcpTableRow.dwLocalAddr) = "127.0.0.1")) Then

With TcpTableRow

Set lvItem = ListView5.ListItems.Add(, , GetIpFromLong(.dwLocalAddr))

lvItem.SubItems(1) = GetPortNumber(.dwLocalPort)

lvItem.SubItems(2) = GetIpFromLong(.dwRemoteAddr)

lvItem.SubItems(3) = GetPortNumber(.dwRemotePort)

lvItem.SubItems(4) = GetServiceByPort(.dwRemotePort)

lvItem.SubItems(5) = GetState(.dwState)

End With


```

End If
arrTcpTableRows(i) = TcpTableRow

Next i
End If
End Sub

```

4 Example Code: Packer Receive and Parsing

```

Private Sub vpacket_PacketReceived(numbytes As Long)

Dim SrcMacAddr As String
Dim DestMacAddr As String
Dim ipHead As IPv4Header
Dim PosInBuffer As Long
Dim ThisEtherNetHeader As EtherNetHeader
Dim NextProto As Long
Dim EHeader As EtherNetHeader

PacketCount = PacketCount + 1
StatusBar1.Panels(StatusBar1.Panels.Count - 1).Text = "Packets: " & PacketCount

vpacket.GetRawPacketData ByteArray
PosInBuffer = 0
ThisPacketHeader = GetFrameHeader(ByteArray, NextProto)

ReDim Preserve ThisPacketBytes(PacketCount + 1)
ReDim Preserve ThisPacketDataSize(PacketCount + 1)
ThisPacketDataSize(PacketCount) = numbytes

```

```
ReDim ThisPacketBytes(PacketCount).ByteArray(numbytes)
```

```
For i = PosInBuffer To PosInBuffer + numbytes
```

```
    ThisPacketBytes(PacketCount).ByteArray(i) = ByteArray(i)
```

```
Next i
```

```
StatusBar1.Panels(StatusBar1.Panels.Count).Text = "Bytes: " &
```

```
ThisPacketHeader.CapLength
```

```
EHeader = GetEtherNetHeader(ByteArray, NextProto, NextProto)
```

```
lsvPackets.ListItems.Add , , PacketCount
```

```
ipHead = GetIpV4Proto(ByteArray, NextProto)
```

```
lsvPackets.ListItems(PacketCount).ListSubItems.Add , , EHeader.SrcMacAddr
```

```
lsvPackets.ListItems(PacketCount).ListSubItems.Add , ,
```

```
GetIpFromLongRev(ipHead.Source)
```

```
'lsvPackets.ListItems(PacketCount).ListSubItems.Add , , ""
```

```
lsvPackets.ListItems(PacketCount).ListSubItems.Add , , EHeader.DestMacAddr
```

```
lsvPackets.ListItems(PacketCount).ListSubItems.Add , ,
```

```
GetIpFromLongRev(ipHead.Destination)
```

```
End Sub
```


5 Example Code: LMMT Client (WMI Script)

Shutdown

```
Dim p_objSet As SWbemObjectSet
Dim p_objSWbemObj As Object

Set p_objSet = GetObject("winmgmts:{(Shutdown)}//./root/cimv2")._
    ExecQuery("select * from Win32_OperatingSystem where Primary=true")

For Each p_objSWbemObj In p_objSet
    p_objSWbemObj.shutdown
    Debug.Print
Next p_objSWbemObj
```

Reboot

```
Dim p_objSet2 As SWbemObjectSet
Dim p_objSWbemObj2 As Object

Set p_objSet2 = GetObject("winmgmts:{(Shutdown)}//./root/cimv2")._
    ExecQuery("select * from Win32_OperatingSystem where Primary=true")

For Each p_objSWbemObj2 In p_objSet2
    p_objSWbemObj2.reboot
    Debug.Print
Next p_objSWbemObj2
```

6 Example Code: TCP Servers Eye (Monitor Code) (Winsock Implementation)

```
Private Sub Monitor()
```

```
Dim port As Ports
```

```
Dim RowPos As Integer
```

```
grdLMMTGrid.Enabled = False
```

```
For RowPos = 1 To grdLMMTGrid.Rows - 1
```

```
    If AbruptStop = True Then Exit Sub
```

```
    If sckLMMT.State <> 0 Then sckLMMT.Close
```

```
    grdLMMTGrid.Col = TIpAddress
```

```
    grdLMMTGrid.Row = RowPos
```

```
    sckLMMT.RemoteHost = grdLMMTGrid.Text
```

```
    StatusBar10.Panels.Item(1).Text = grdLMMTGrid.Text
```

```
    grdLMMTGrid.Col = TPortNum
```

```
    sckLMMT.RemotePort = Val(grdLMMTGrid.Text)
```

```
    sckLMMT.Connect
```

```
    grdLMMTGrid.Col = TIpAddress
```

```
While sckLMMT.State = 4
```

```
    StatusBar10.Panels.Item(2).Text = "Resolving..."
```

```
    DoEvents
```

```
    If sckLMMT.State = 5 Then StatusBar10.Panels.Item(2).Text = "Resolved"
```

```
Wend
```

```
While sckLMMT.State = 6
```

```
    StatusBar10.Panels.Item(2).Text = "Connecting..."
```

```
    DoEvents
```

```
Wend
```


Select Case sckLMMT.State

Case 0:

StatusBar10.Panels.Item(2).Text = "Closed"

Case 1:

grdLMMTGrid.Col = TLastState

grdLMMTGrid.Text = "ALIVE! (1)"

grdLMMTGrid.Col = TIpAddress

StatusBar10.Panels.Item(2).Text = "ALIVE! (1)"

LogEvent grdLMMTGrid.Text & "(" & sckLMMT.RemotePort & ")" & " is
ALIVE! (1)"

Case 2:

StatusBar10.Panels.Item(2).Text = "Listening"

Case 3:

StatusBar10.Panels.Item(2).Text = "Pending"

Case 4:

StatusBar10.Panels.Item(2).Text = "Resolving"

Case 5:

StatusBar10.Panels.Item(2).Text = "Resolved"

Case 6:

StatusBar10.Panels.Item(2).Text = "Connecting"

While sckLMMT.State = 6

DoEvents

Wend

Case 7:

grdLMMTGrid.Col = TLastState

grdLMMTGrid.Text = "ALIVE! (7)"

grdLMMTGrid.Col = TIpAddress

StatusBar10.Panels.Item(2).Text = "Connected"

StatusBar10.Panels.Item(2).Text = "ALIVE! (7)"

LogEvent grdLMMTGrid.Text & "(" & sckLMMT.RemotePort & ")" & " is

ALIVE! (7)"

Case 8:

StatusBar10.Panels.Item(2).Text = "Closing"

Case 9:

grdLMMTGrid.Col = TLastState

grdLMMTGrid.Text = "DOWN! (9)"

grdLMMTGrid.Col = TIpAddress

StatusBar10.Panels.Item(2).Text = "DOWN! (9)"

LogEvent grdLMMTGrid.Text & "(" & sckLMMT.RemotePort & ")" & " is
Down! (9)"

End Select

sckLMMT.Close

Next

grdLMMTGrid.Enabled = True

End Sub

7 Event Logging Example Output

LMMT Log

2/21/2005 12:36:23 PM LMMT Start

2/21/2005 12:36:23 PM Hostname:ANX801 , IP Address:202.185.109.129

2/21/2005 12:36:50 PM Highest TCP Segment Received :9380 , Highest TCP Segment Sent :8430

2/21/2005 12:36:50 PM Highest UDP Datagram Received :3609 , Highest UDP Datagram Sent :3318

2/21/2005 12:36:50 PM Highest ICMP Message Received :32 , Highest ICMP Message Sent :7

2/21/2005 12:36:50 PM Highest Incoming Bandwidth :0.275 KBPS , Highest Outgoing Bandwidth :0.54 KBPS

2/21/2005 12:36:50 PM LMMT Exit

2/21/2005 12:36:50 PM

2/21/2005 12:46:30 PM LMMT Start

2/21/2005 12:46:30 PM Hostname:ANX801 , IP Address:202.185.109.129

2/21/2005 12:51:07 PM Highest TCP Segment Received :9800 , Highest TCP Segment Sent :8779

2/21/2005 12:51:07 PM Highest UDP Datagram Received :3680 , Highest UDP Datagram Sent :3374

2/21/2005 12:51:07 PM Highest ICMP Message Received :48 , Highest ICMP Message Sent :11

2/21/2005 12:51:07 PM Highest Incoming Bandwidth :10.179 KBPS , Highest Outgoing Bandwidth :2.433 KBPS

2/21/2005 12:51:07 PM LMMT Exit

2/21/2005 12:51:07 PM

ServerEye Log

2/21/2005 12:25:07 PM TCP Servers Eye is loading...
2/21/2005 12:25:07 PM TCP Servers Eye Configuraton Loaded
2/21/2005 12:25:07 PM TCP Servers Eye load complete!
2/21/2005 12:29:29 PM Monitoring Started
2/21/2005 12:29:30 PM 202.185.109.157(80) is Down! (9)
2/21/2005 12:29:31 PM 202.185.109.157(21) is Down! (9)
2/21/2005 12:29:31 PM 202.185.109.157(23) is ALIVE! (7)
2/21/2005 12:29:32 PM 202.185.109.157(25) is Down! (9)
2/21/2005 12:29:33 PM 202.185.109.157(53) is Down! (9)
2/21/2005 12:29:34 PM 202.185.109.157(110) is Down! (9)
2/21/2005 12:29:35 PM 202.185.109.157(443) is Down! (9)
2/21/2005 12:29:45 PM Monitoring Stopped
2/21/2005 12:36:50 PM TCP Servers Eye Exited.
2/21/2005 12:36:50 PM -----

Table of Contents

Table of Contents

List of Figures

Introduction

System Requirements

Chapter 1: Login to System

1.1. Backend Setting

Chapter 2: Monitoring Module

2.1. LMMT Dashboard

2.2. Using Computer

2.3. Working With TCP Port Monitor

2.4. Configuration Server

2.5. View Bandwidth Monitor

Chapter 3: Management module

3.1. Installing LMMT

3.2. Monitor Computer Information and Perform Remote Command

3.3. Monitor From IP Address

3.4. Configuring TCP Services Eye

3.5. Remote Processes and Services Management

Setting Log File

USER MANUAL

Table of Contents

Table of Contents.....	1
List of Figures.....	2
Introduction.....	3
System Requirement/Predecessor.....	4
Chapter 1 Login to System.....	5
1.1 Password Setting.....	6
Chapter 2 Monitoring Module	
2.1 LMMT Dashboard.....	7
2.2 Using Computer Browser.....	8
2.3 Working With TCP Port Monitor.....	9
2.4 Configuring Sniffer.....	10
2.5 View Bandwidth Monitor.....	11
Chapter 3 Management Module	
3.1 Installing LMMTclient.....	12
3.2 Retrieve Remote Computer Information and Perform Remote Command.	13
*** Get Hostname From IP Address.....	14
3.3 Configuring TCP Servers Eye.....	15
3.4 Remote Processes and Services Management.....	17
Viewing Log File.....	18

List of Figures

Figure 1. Login Dialogue Box	5
Figure 2. Password Setting Dialogue Box	6
Figure 3. LMMT Dashboard	7
Figure 4. Computer Browser	8
Figure 5. TCP Port Monitor	9
Figure 6. Sniffer	10
Figure 7. Bandwidth Monitor	11
Figure 8. Remote Computer Information and Remote Command	13
Figure 9. TCP Servers Eye	15
Figure 10. Remote Process and Service Form	17

Introduction

The purpose of this user manual is to provide some helpful guideline and usage about this LAN Monitoring and Management Tool (LMMT) system to the user. LMMT can be divided into two main modules – **Monitoring Module** and **Management Module**.

The first chapter covers the system authentication using password and how to change the administrator password.

The second chapter tells the user about the monitoring module contain in this system. The sub-module to be covers is the **LMMT Dashboard** which show the statistic for IP, TCP, UDP and ICMP, **Computer Browser** where show the connected computer in a LAN with their IP address and share folder, **TCP Port Monitor** that shows the connection between computer in a LAN and internet, **Sniffer** that capture the packet in a LAN and provide the information about the packet and the **Bandwidth Monitor** that shows the incoming and outgoing packets in the format of Bytes.

The third chapter completes this system with the management module. This will cover up the **Remote Computer Information** and **Remote Command** that can be performed on the remote computer with the LMMTclient installed.

System Requirement/Predecessor

Before you can start using this system, please make sure the following stuffs are installed in your computer.

1. WinPcap 3.0

It is important to make sure the sniffing engine in this system to work successfully. It is a freeware driver that can be download at:

<http://www.winpcap.polito.it>

2. WMI Core

To make sure the remote command to be accomplished. It is for the system that does not pre-installed with the Windows Management Instrumentation (WMI) (Windows 95, 98 and NT). It comes free and can be download at the Microsoft Website.

<http://www.microsoft.com>

Chapter 1 Login to System

1. When enter the system, a password authentication dialogue box will be appears:

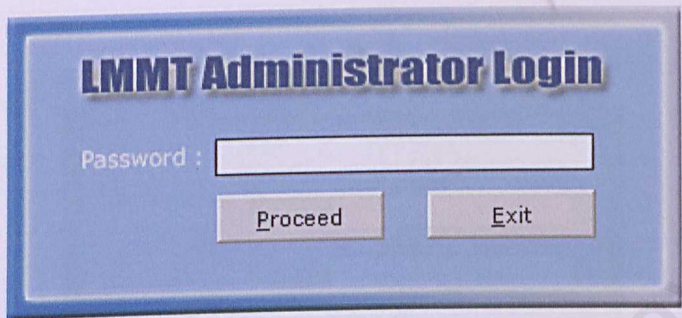


Figure 1: Login Dialogue Box

3. User needs to enter the password to proceed. Wrong password will abort the user from enter the system.

4. Click **Proceed** to enter the system otherwise click **Exit** to quit.

! For the first time user, password is set to blank. Once enter the system, please change your password for the security purposes.

1.1 Password Setting

1. Click on the **Password Setting** button to change your password. Once click, a Password setting dialogue box will be appears:

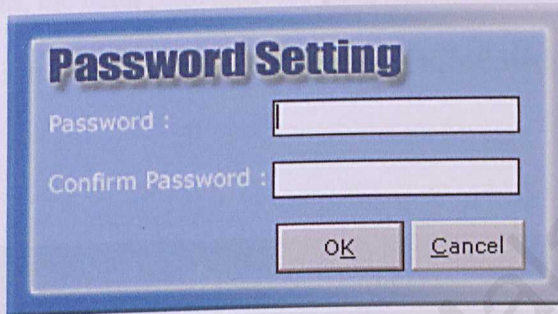
A screenshot of a 'Password Setting' dialog box. The dialog box has a blue title bar with the text 'Password Setting' in bold. Below the title bar, there are two text input fields. The first field is labeled 'Password :' and the second field is labeled 'Confirm Password :'. At the bottom of the dialog box, there are two buttons: 'OK' and 'Cancel'.

Figure 2: Password Setting Dialogue Box

2. Enter your new password in the password field and again in the confirm password field. Both fields must be identical. Click **OK** to proceed with password change, otherwise click **Cancel**.

Chapter 2 Monitoring Module

2.1 LMMT Dashboard

Once enter the system, the main page that will be appear is the LMMT Dashboard. Here, user can get various information on IP, TCP, UDP and ICMP statistic.

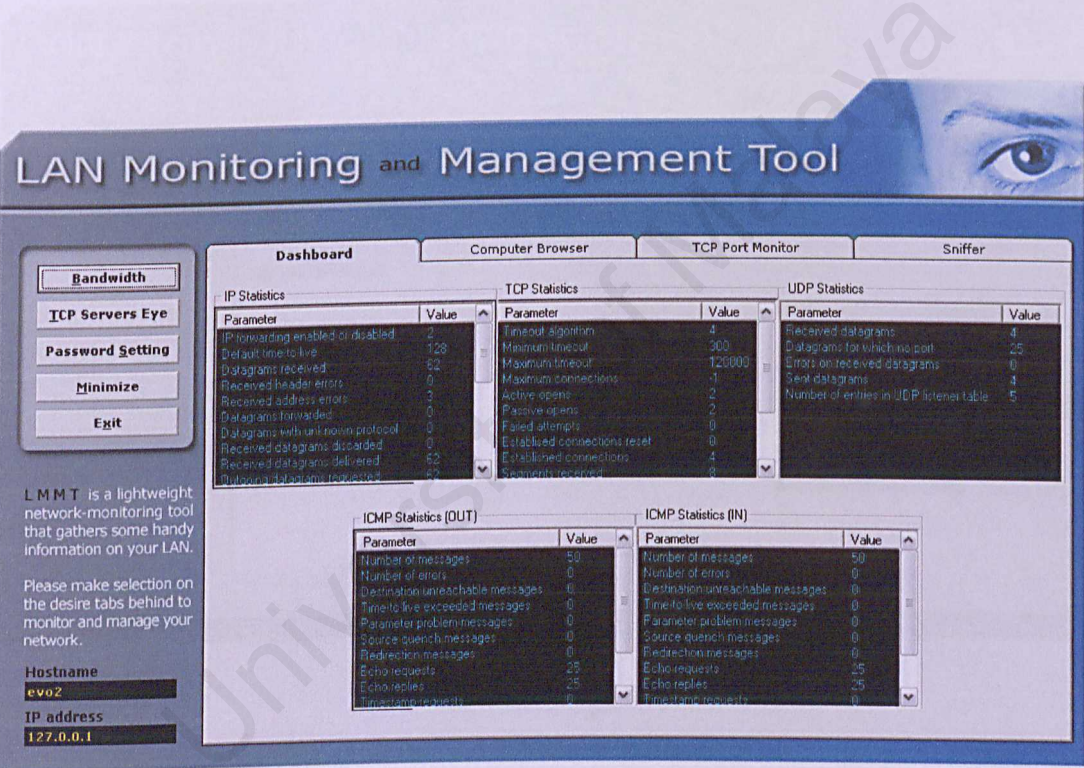


Figure 3: LMMT Dashboard

On the bottom left of the system, shows user the Hostname for the current computer and its IP address. While, on the left side (located in the shadowed frame) is the button that are available for the user. Each button perform different task base on the caption of the button.

2.2 Using Computer Browser

1. Click on the Computer Browser tab to switch to computer browser section.

* User can browse the connected computer in the LAN. The computer browser also shows the share document/path for the connected computers.

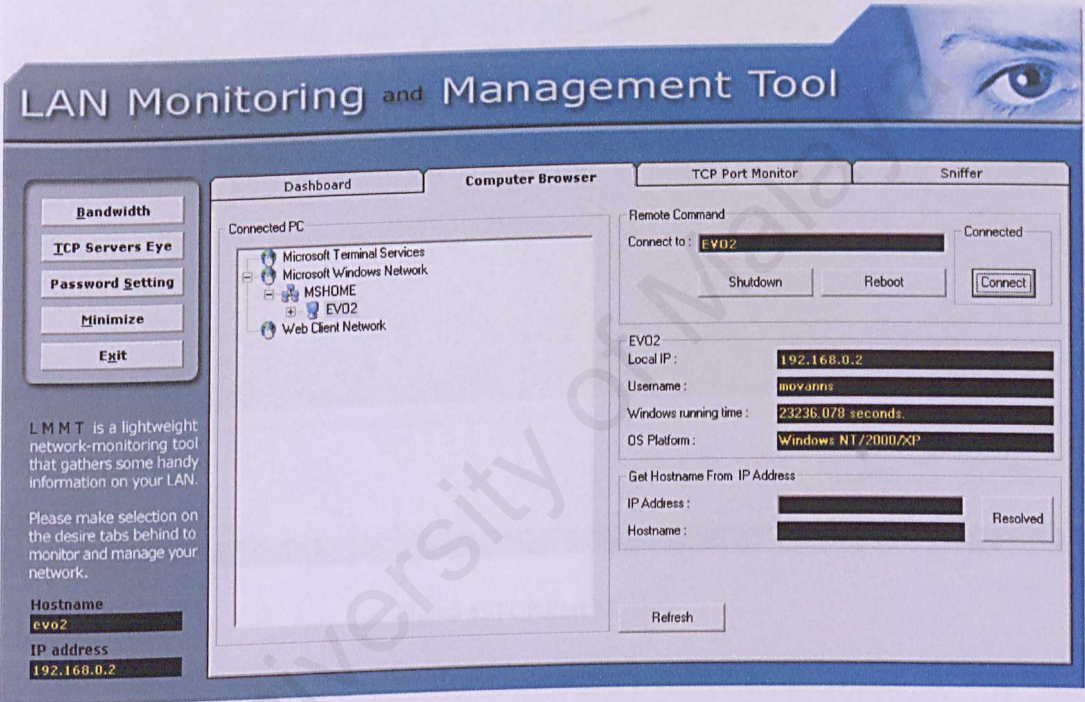


Figure 4: Computer Browser

2. Click on the connected computer name on the list view to resolved the IP address for the computer. The address will be shown on the Local IP field.

3. Click **Refresh** button to refresh the list view.

2.3 Working With TCP Port Monitor

1. Click on TCP Port Monitor tab to access the TCP Port Monitor section.

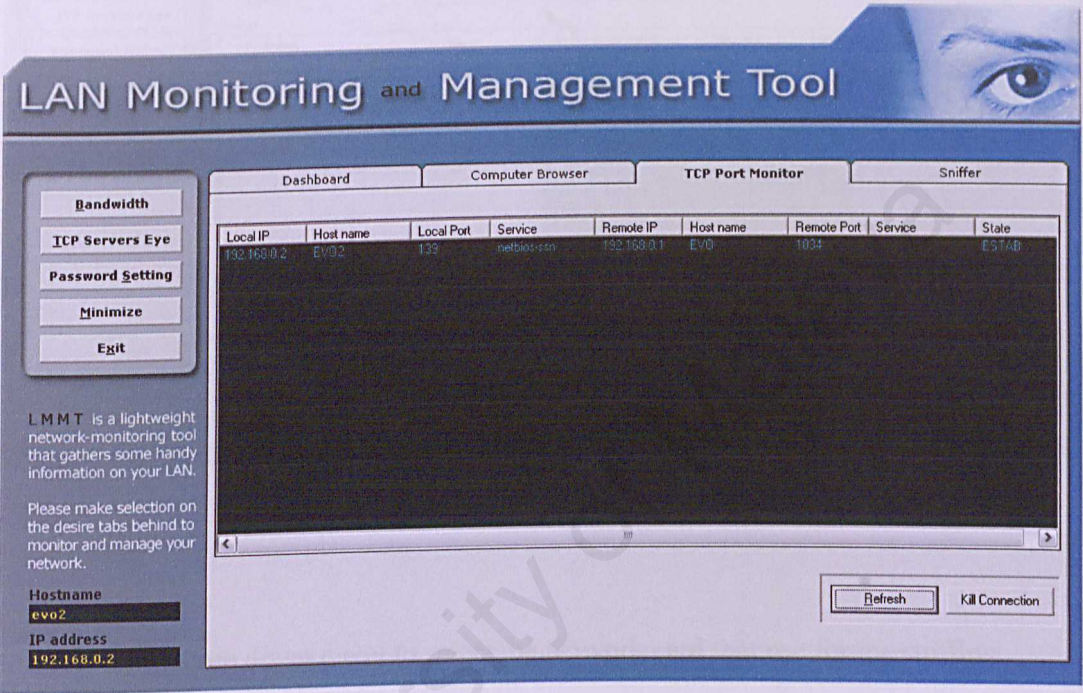


Figure 5: TCP Port Monitor

* The list box is empty when access for the first time

2. Click on the **Refresh** button to fill up the list box with the connection and the state for entire connection.
3. To kill the unwanted connection to the computer or system, click the **Kill Connection** button. If the connection is successful or vice versa, a message box will be appear to show the result.

2.4 Configuring Sniffer

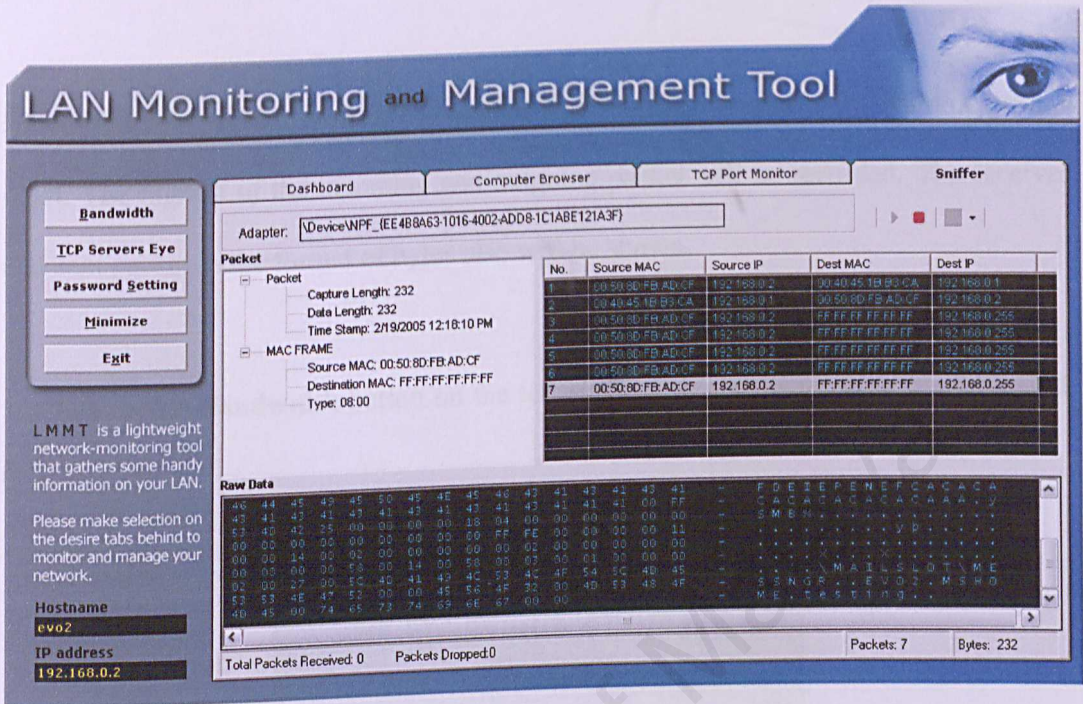


Figure 6: Sniffer

1. Click on the drop down menu to select the network card that use for the sniffing.
2. Once select, the network card detail will be appear at the adapter field.
3. Click play button (green colour) to start sniffing the network and click stop button (red colour) to stop.

* Click on the desire row on the list box to shows the information and raw data of the captured packet. For best result of packet capturing, place this system either on the proxy server of you LAN or connect to the span port in your main switch.

2.5 View Bandwidth Monitor

Bandwidth monitor will show the incoming and outgoing packets in the form of column chart. The highest of the incoming and outgoing packet will be recorded. The received and sent packet in the format of bytes also will be shown.

1. Click on the **Bandwidth** button on the left side of the system to open the bandwidth box.

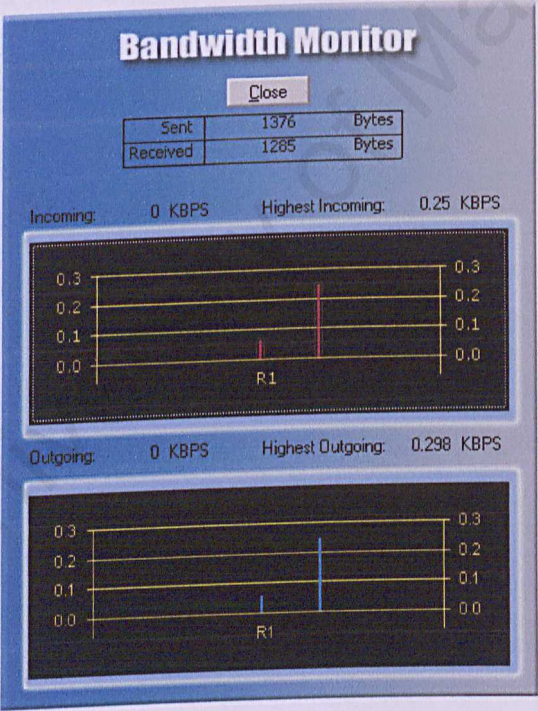


Figure 7: Bandwidth Monitor

2. Click the **Close** button to close the Bandwidth Monitor box.

Chapter 3 Management Module

3.1 Installing LMMTclient

LMMTclient is a program that needs to be installed in the client computer in order to make sure the remote computer information and remote command working properly.

1. Copy the LMMTclient.exe to Windows folder in client computer
2. Create a shortcut to LMMTclient.exe and put the shortcut in the windows startup folder.

3.2 Retrieve Remote Computer Information and Perform Remote Command

Remote computer information field is located on the **Computer Browser** section.

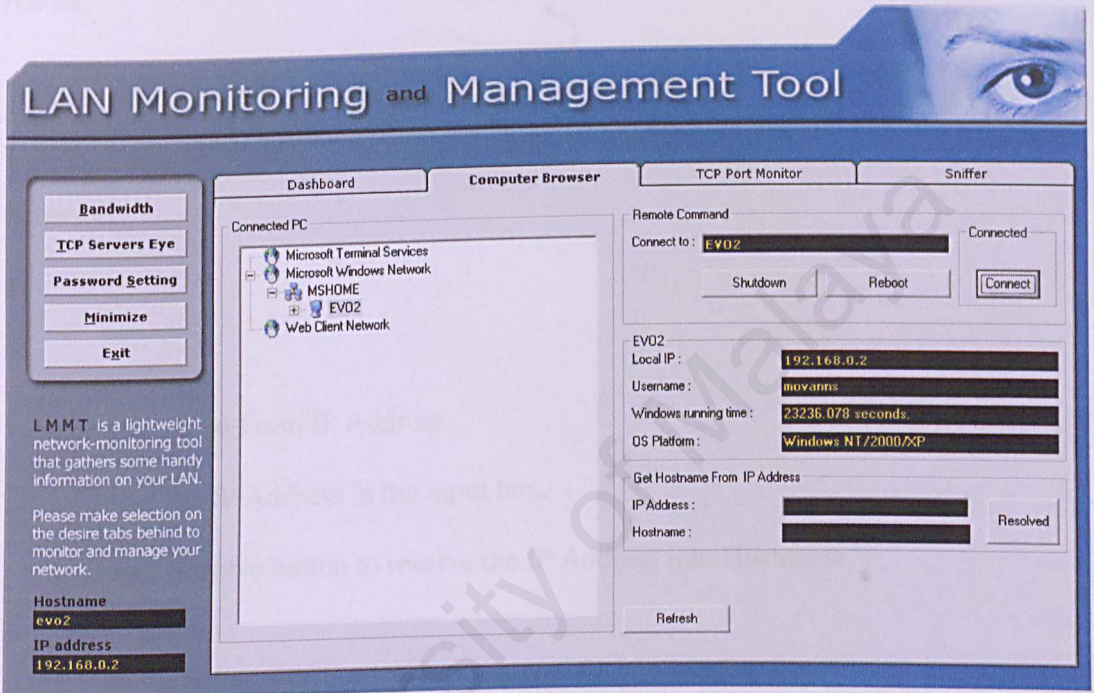


Figure 8: Remote Computer Information and Remote Command

1. Click on the **Connect** button on the right side of the system to connect the system to the remote computer installed with the LMMTclient.

* Once connected, the frame caption for connect button change to “Connected” and information on remote computer will be appear (Username of current user using the computer, Windows running time and the OS platform for the computer)

2. When the remote computer are connected, remote command can be send and will be perform on the remote computer.
3. Click either **Shutdown** button to shutdown the remote computer or **Reboot** button to reboot.
4. To disconnect from remote computer, simply click to other computer on the Computer Browser.

*** Get Hostname From IP Address.

1. Input the IP Address in the input box.
2. Click Resolve button to resolve the IP Address into Hostname.

3.3 Configuring TCP Servers Eye

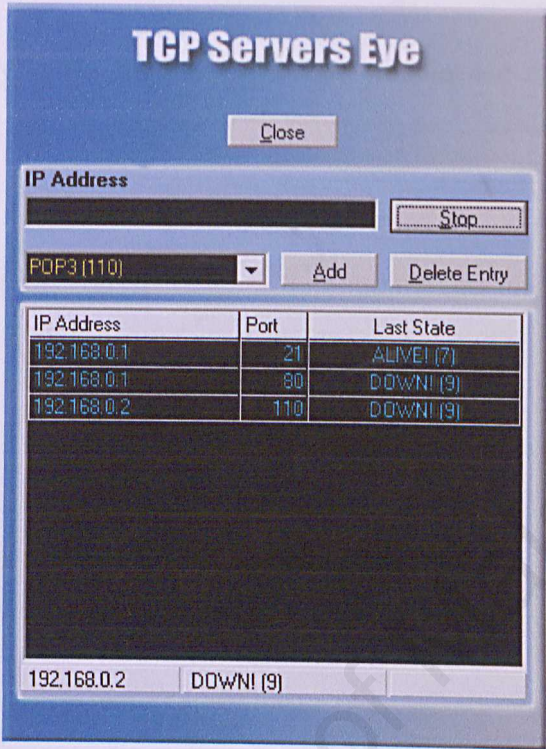


Figure 9: TCP Servers Eye

TCP Servers Eye is a server management that monitors your server in the LAN by checking the service port used by the server. Only server that running their service on the TCP port are monitored.

1. Enter the IP Address for the server that you want to monitor.
2. Select the service port.
3. Click **Add** button to add the entry in the list box. Repeat to enter more servers.

4. To monitor the servers, click **Monitor**, and simply click **Stop** to stop the monitoring.
5. To delete the entry in the list box, select the column and click the **Delete Entry** button.

*While monitor the server, system will inform the user either the service are
ALIVE or DOWN.

3.4 Remote Processes and Services Management

- 1. On the computer browser tab, click remote management button to show the remote management form.

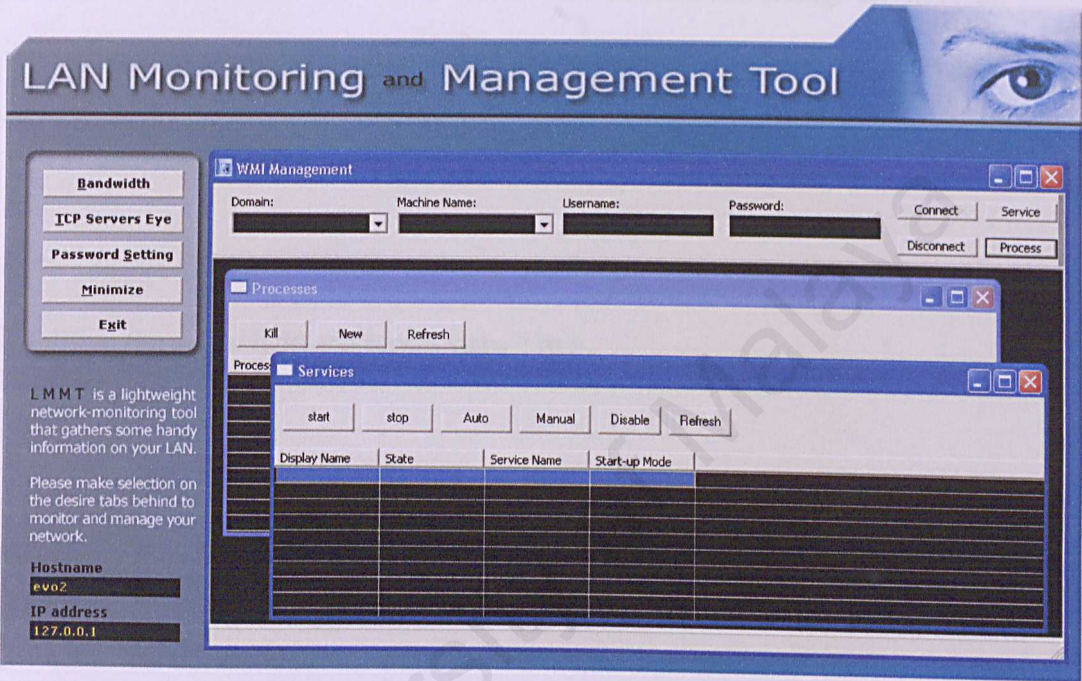


Figure 10: Remote Process and Service Management Form

- 2. Enter the domain, computer name/IP address, username and the password for the remote computer that you want to connect. Make sure it is correct. Otherwise, you won't be connected.

- 3. Click the connect button.

* Once connected, the process and service form will be appear with the list of the remote service and process. Otherwise, it will not appear.

4. Select the process or service that you want to control. For processes, the commands that are available is “KILL”, “NEW” and “REFRESH”. Click the desire button to remotely done the command.
5. For services, user can “START”, “STOP”, the services and change the services routine with the command button “AUTO”, “MANUAL” and “DISABLE”. Click the “REFRESH” button to refresh the list.
6. To connect to other computer, click the disconnect button and enter the new computer details on the desire field and repeat steps 2 to 5.

Viewing Log File

Every session the user start or exit the system, a log is created. It contains some useful information. TCP Server Eye also generates a log file for the monitoring of the servers.

To view the log file:

1. Go to the installation folder where the LMMT.EXE is located.
2. Find for the textfile:
 - LMMTlog.txt - LMMT log file
 - ServerEyelog.txt - TCP Servers Eye log file
3. Double click or open the file with Notepad.