

raf

TK5105.5.H34 2005



0000037752

Monitoring network by capturing packets using C++ /  
Halida Mohd Zahari.

## **MONITORING NETWORK BY CAPTURING PACKETS USING C++**

**HALIDA MOHD ZAHARI**

This report is submitted in partial fulfillment of the requirements for the Bachelor of  
Computer Science (Computer Network)

**KOLEJ UNIVERSITI TEKNIKAL KEBANGSAAN MALAYSIA**

## BORANG PENGESAHAN STATUS TESIS<sup>^</sup>

JUDUL: MONITORING NETWORK BY CAPTURING PACKETS USING C++

SESI PENGAJIAN: 2005/2006

Saya HALIDA MOHD ZAHARI

(HURUF BESAR)

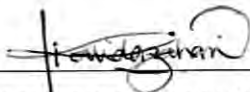
mengaku membenarkan tesis (PSM/Sarjana/Doktor Falsafah) ini disimpan di Perpustakaan Fakulti Teknologi Maklumat dan Komunikasi dengan syarat-syarat kegunaan seperti berikut:

1. Tesis adalah hakmilik Kolej Universiti Teknikal Kebangsaan Malaysia.
2. Perpustakaan Fakulti Teknologi Maklumat dan Komunikasi dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan Fakulti Teknologi Maklumat dan Komunikasi dibenarkan membuat salinan tesis ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. \*\* Sila tandakan (/)

           SULIT (Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)

           TERHAD (Mengandungi maklumat TERHAD yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan)

  /   TIDAK TERHAD



(TANDATANGAN PENULIS)

Alamat tetap : NO 16, JLN BENDAHARA 10/7,  
BANDAR MAHKOTA CHERAS, 43200 CHERAS

Tarikh : 23 NOVEMBER 2005



(TANDATANGAN PENYELIA)

ASLINDA HASSAN

Nama Penyelia

Tarikh : 23 NOV 2005

CATATAN: \*\* Jika tesis ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa.

<sup>^</sup> Tesis dimaksudkan sebagai Laporan Projek Sarjana Muda (PSM)

## DECLARATION

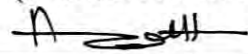
I hereby declare that this project report entitled  
**MONITORING NETWORK BY CAPTURING PACKETS USING C++**

is written by me and is my own effort and that no part has been plagiarized  
without citations.

STUDENT :   
\_\_\_\_\_

(HALIDA MOHD ZAHARI)

Date : 23 NOVEMBER 2005

SUPERVISOR :   
\_\_\_\_\_

(PUAN ASLINDA HASSAN)

Date : 23 NOV 2005

## **DEDICATION**

Specially dedicated to my beloved parents, family and fellow friends, who had encouraged and supported me in my entire journey of learning, thanks a lot.

## ACKNOWLEDGEMENT

First of all, I want to say syukur alhamdulillah and thanks to God because I can complete my thesis during my short semester.

Secondly, I would like to thank to my supervisor Puan Aslinda Hassan for giving me guidance and encouragement. I am really appreciated all the cooperation and guidance to gain knowledge in order to complete my thesis. Thanks to all your support and trust.

I also want to thank to my family who gave me supportive morale during finish my thesis. They always gave me spirit in doing my thesis successfully. This acknowledgement also goes to my previous practical company which is Telekom Training College. A big thank to my industrial supervisor, Puan Aibealiza, Puan Wan Fariza and all the staff for their encouragement and cooperation.

Thank you also to my friends, housemates, classmates and all people who have supported me. Thanks for the cooperation and support in doing my thesis.

## ABSTRACT

The main purpose of this project is to capture packets that pass through a computer to which connects Internet. All the data in the packets can be viewed and analyzed.

The system supports 78 types of protocol from Application Layer of the Open System Interconnection model. 15 types of protocol from User Datagram Protocol and 63 types of protocol from Transmission Control Protocol of the Transport Layer.

This program uses Windows as platform and the raw socket method to get direct access to the DataLink layer.

This application is developed using the Application Programming Interface sockets and Microsoft Visual C++ 6.0.

This project is a helpful monitoring tool. It also is very helpful for network administrator and anyone who interested in network traffic going through the computer.

## ABSTRAK

Tujuan utama projek ini adalah mendapatkan paket yang melepasi komputer. Semua data di dalam paket boleh dilihat dan kemudian data ini akan dianalisa.

Sistem ini dapat menganalisa 78 jenis protokol dari *Application Layer*. 15 jenis protokol dari *User Datagram Protocol* dan 63 jenis protocol dari *Transmission Control Protocol*.

Program ini menggunakan Windows sebagai landasan dan menggunakan *raw socket* untuk berhubung terus dengan *datalink layer*.

Program ini menggunakan soket Aplikasi Pengaturcaraan Antaramuka dan ia dibina di dalam Microsoft Visual C++ 6.0.

Kesimpulannya, projek ini merupakan alat pemantauan yang sangat membantu. Ia juga sangat membantu bagi pentadbir rangkaian dan sesiapa sahaja yang berminat dalam trafik rangkaian komputer yang melepasi komputer tersebut.



## TABLE OF CONTENT

| CHAPTER          | SUBJECT                         | PAGE        |
|------------------|---------------------------------|-------------|
|                  | <b>DECLARATION</b>              | <b>ii</b>   |
|                  | <b>DEDICATION</b>               | <b>iii</b>  |
|                  | <b>ACKNOWLEDGEMENTS</b>         | <b>iv</b>   |
|                  | <b>ABSTRACT</b>                 | <b>v</b>    |
|                  | <b>ABSTRAK</b>                  | <b>vi</b>   |
|                  | <b>TABLE OF CONTENTS</b>        | <b>vii</b>  |
|                  | <b>LIST OF TABLES</b>           | <b>xi</b>   |
|                  | <b>LIST OF DIAGRAMS</b>         | <b>xii</b>  |
|                  | <b>LIST OF SYMBOLS/ACRONYMS</b> | <b>xiii</b> |
|                  | <b>LIST OF ATTACHMENTS</b>      | <b>xiv</b>  |
| <b>CHAPTER I</b> | <b>INTRODUCTION</b>             |             |
|                  | <b>1.1 Project Background</b>   | <b>1</b>    |
|                  | <b>1.2 Problem Statements</b>   | <b>2</b>    |
|                  | <b>1.3 Objectives</b>           | <b>3</b>    |
|                  | <b>1.4 Scopes</b>               | <b>3</b>    |
|                  | <b>1.5 Project Significance</b> | <b>3</b>    |
|                  | <b>1.6 Expected Output</b>      | <b>4</b>    |
|                  | <b>1.7 Conclusion</b>           | <b>4</b>    |



|                    |  |    |
|--------------------|--|----|
| <b>CHAPTER II</b>  | <b>LITERATURE REVIEW AND PROJECT<br/>METHODOLOGY</b> |    |
| 2.1                | Introduction   | 5  |
| 2.2                | Fact and Finding                                     | 5  |
|                    | 2.2.1 Network Monitoring                             | 5  |
|                    | 2.2.2 Ethereal                                       | 6  |
| 2.3                | Project Methodology                                  | 7  |
| 2.4                | Project Requirement                                  | 10 |
|                    | 2.4.1 Software Requirement                           | 10 |
|                    | 2.4.2 Hardware Requirement                           | 10 |
| 2.5                | Project Schedule and Milestone                       | 10 |
|                    | 2.5.1 Project Schedule                               | 10 |
|                    | 2.5.2 Project Milestones                             | 11 |
|                    | 2.6 Conclusion                                       | 11 |
| <b>CHAPTER III</b> | <b>ANALYSIS</b>                                      |    |
| 3.1                | Introduction   | 12 |
| 3.2                | Problem Analysis                                     | 12 |
| 3.3                | Requirement Analysis                                 | 13 |
|                    | 3.3.1 Functional Requirements                        | 13 |
|                    | 3.3.2 Software Requirements                          | 15 |
|                    | 3.3.3 Hardware Requirements                          | 16 |
|                    | 3.3.4 Network Requirement                            | 16 |
| 3.4                | Conclusion   | 16 |
| <b>CHAPTER IV</b>  | <b>DESIGN</b>  |    |
| 4.1                | Introduction   | 17 |
| 4.2                | High Level Design                                    | 17 |
|                    | 4.2.1 System Architecture                            | 18 |
|                    | 4.2.2 User Interface Design                          | 19 |
|                    | 4.2.2.1 Navigation Design                            | 20 |
|                    | 4.2.2.2 Output Design                                | 21 |
| 4.3                | Detailed Design                                      | 23 |

|                    |       |   |    |
|--------------------|-------|---|----|
|                    | 4.3.2 | Physical Design                           | 23 |
|                    | 4.4   | Conclusion                                | 24 |
| <b>CHAPTER V</b>   |       | <b>IMPLEMENTATION</b>                     |    |
|                    | 5.1   | Introduction                              | 25 |
|                    | 5.2   | Software Development Environment          |    |
|                    |       | Setup                                     | 26 |
|                    | 5.3   | Software Management Configuration         | 27 |
|                    |       | 5.3.1 Configuration Environment           |    |
|                    |       | Setup                                     | 27 |
|                    |       | 5.3.2 Version Control Procedure           | 27 |
|                    | 5.4   | Implementation Status                     | 27 |
|                    | 5.5   | Conclusion                                | 28 |
| <b>CHAPTER VI</b>  |       | <b>TESTING</b>                            |    |
|                    | 6.1   | Introduction                              | 29 |
|                    | 6.2   | Test Plan                                 | 30 |
|                    |       | 6.2.1 Test Organization                   | 30 |
|                    |       | 6.2.2 Test Environment                    | 31 |
|                    |       | 6.2.3 Test Schedule                       | 31 |
|                    | 6.4   | Test Design                               | 32 |
|                    |       | 6.4.1 Test Description                    | 32 |
|                    | 6.5   | Test Results and Analysis                 | 32 |
|                    |       | 6.5.1 Capturing Packet Test Case          | 33 |
|                    |       | 6.5.2 Capturing Packet Protocol Test Case | 35 |
|                    |       | 6.5.3 Analyzing Packet Test Case          | 36 |
|                    | 6.6   | Conclusion                                | 36 |
| <b>CHAPTER VII</b> |       | <b>PROJECT CONCLUSION</b>                 |    |
|                    | 7.1   | Observation on Weakness and Strengths     | 38 |
|                    |       | 7.1.1 Weakness                            | 39 |
|                    |       | 7.1.2 Strengths                           | 40 |
|                    | 7.2   | Proposition for Improvement               | 41 |

|                       |           |
|-----------------------|-----------|
| <b>7.4 Conclusion</b> | <b>42</b> |
| <b>BIBLIOGRAPHY</b>   | <b>43</b> |
| <b>REFERENCES</b>     | <b>44</b> |
| <b>APPENDIX A</b>     | <b>45</b> |
| <b>APPENDIX B</b>     | <b>52</b> |

**LIST OF TABLES**

| <b>TABLE</b> | <b>TITLE</b>                                | <b>PAGE</b> |
|--------------|---|-------------|
| 5.1          | <b>Implementation Status</b>                | 28          |
| 6.1          | <b>Test Organization</b>                    | 30          |
| 6.2          | <b>Test Schedule for Functional Process</b> | 31          |
| 6.3          | <b>Test Cases Description</b>               | 33          |
| 6.4          | <b>Test Case – Capturing Packets</b>        | 34          |
| 6.5          | <b>Test Case – Analyzing Packets</b>        | 35          |

**LIST OF DIAGRAMS**

| <b>FIGURE</b> | <b>TITLE</b>                 | <b>PAGE</b> |
|---------------|------------------------------|-------------|
| 2.1           | <b>The Frame Methodology</b> | <b>9</b>    |
| 3.1           | <b>Use Case Diagram</b>      | <b>14</b>   |
| 3.2           | <b>Sequence Diagram</b>      | <b>15</b>   |
| 4.1           | <b>System Architecture</b>   | <b>18</b>   |
| 4.2           | <b>User Interface</b>        | <b>19</b>   |
| 4.3           | <b>Navigation Design</b>     | <b>20</b>   |
| 4.4           | <b>Output(1)</b>             | <b>21</b>   |
| 4.5           | <b>Output(2)</b>             | <b>22</b>   |
| 4.6           | <b>Physical Design</b>       | <b>23</b>   |
| 5.1           | <b>Software Environment</b>  | <b>26</b>   |

## LIST OF SYMBOLS/ACRONYMS

|       |   |   |
|-------|---|---|
| IP    | - | Internet Protocol                           |
| FBI   | - | Federal Bureau of Investigation             |
| GUI   | - | Graphic User Interface                      |
| TCP   | - | Transmission Control Protocol               |
| UDP   | - | User Datagram Protocol.                     |
| OOAD  | - | Object Oriented and Design                  |
| SSADM | - | Structure System Analysis and Design Method |
| SDLC  | - | Software Development Life Cycle             |
| LAN   | - | Local Area Network                          |

**LIST OF ATTACHMENTS**

| <b>ATTACHMENT</b> | <b>TITLE</b>       | <b>PAGE</b> |
|-------------------|--------------------|-------------|
| <b>APPENDIX A</b> | <b>Gantt Chart</b> | <b>45</b>   |
| <b>APPENDIX B</b> | <b>User Manual</b> | <b>52</b>   |



# CHAPTER I

## INTRODUCTION

### 1.0 INTRODUCTION

Network monitoring is an important tool for network. Without monitoring, it is difficult to judge whether changes made to the network can either improve the network performance or degraded it. Consequently, there are several reasons for performing network monitoring.

Perform some kind of network monitoring help a given company to be more efficient. For example, some kind of warnings or errors on the network may be detected and corrected before some bad situations arise and cause a loss of business productivity.

Network monitoring can also improve the security of a company. This will ensure that the business can run more efficiently and smoothly. If any problem arises, it can be determined and solved quickly.

Therefore, network monitoring is quite an important task that needs to be done by corporate network administrators, managers or other people who involve in its charge.

## 1.1 PROJECT BACKGROUND

The purpose of this project is to develop a program that monitors network traffic that passes through a computer and eavesdrops on the network traffic. Similar to a telephone wiretap that allows the FBI to listen on other people's conversations, a sniffing program lets someone listen in on computer conversations. However, computer conversations consist of apparently random binary data. Therefore, this program also comes with a feature known as protocol analysis, which allows user to decode the network traffic and turn it into network information that can be used by network administrator.

This program which runs on the computer connected to the Internet using a modem, which can tell user's IP address as well as the IP addresses of the web servers whose sites are visiting. Also, it can view all the unencrypted data that travels from the computer, to the Internet which includes passwords and other sensitive data which are not secured by encryption that pass through that computer.

This project is developed in Microsoft Windows XP.

## 1.2 PROBLEM STATEMENT

By using this system, user can capture packets and view the data packets. The system views the data packets through a console window. User can view the entire data packets that travel from the computer to the Internet or vice versa which includes passwords and other sensitive data that are not secured by encryption.

### 1.3 OBJECTIVES

- i. To capture packets and IP headers
- ii. To view type of protocol of the packets
- iii. To analyze the packets in term of network protocols

### 1.4 SCOPES

- i. To view the data in the packets and IP headers such as port number, packet's size and IP header's size.
- ii. To view the type of protocol for the packets from Transport Layer and Application Layer of the OSI model.
- iii. To count and view the total of each type of protocol that has been captured

### 1.5 PROJECT SIGNIFICANCE

This system is helpful for:

- Network administrators in troubleshooting, analysis, software and protocol development by analyzing the type of protocol for the packets
- Programmers to check the packets that they are trying to send
- Students to learn internal network protocol

## 1.6 EXPECTED OUTPUT

Some assumptions are made for this system to make sure its' smoothness and effectiveness. Assumption made during developing this system is:

- i. The program is able to capture packets and IP header
- ii. The program is able to view type of protocol of the packets
- iii. The program is able to analyze the packets

At the end of the development phase, this program must be able to capture and analyzed the data in the packets.

## 1.7 CONCLUSION

In conclusion, this chapter has explained in detail all the information and studies that are related to the development planning for the program. Studies on the concept of the system gives a lot of information and helps to understand better the development of the program. This is essential to ensure all the development tools that are going to be used are correct and suitable. Next chapter will describe about literature review.

## **CHAPTER II**

### **LITERATURE REVIEW AND PROJECT METHODOLOGY**

#### **2.1 INTRODUCTION**

This program is using Windows XP as its platform and it does not use the sock packet method which is restricted to Windows, but uses the packet filter method which is the newer way to get direct access to the datalink layer. This application is developed using the API sockets and Microsoft Visual C++ 6.0.

#### **2.2 FACT AND FINDING**

##### **2.2.1 Network Monitoring**

Network monitoring is a tool to monitor the network. It is a process of viewing and analyzing network traffic.



Network monitoring software packages are used to capture network data frames and examine them. A software package that can analyze protocol information in a data frame often referred to as a protocol sniffer (*Joseph W Habraken, 2003, Absolute Beginner's Guide to Networking, 395*). The Monitoring Network by Capturing Packets Using C++ application is built as a network monitoring tool which is able to capture packets and analyze the protocol of the packets.

Most network monitoring and packet sniffing are geared for Ethernet network because it is the most commonly used network architecture (*Joseph W Habraken, 2003, Absolute Beginner's Guide to Networking, 395*).

Networking monitoring is an important tool to detect failure in network system. This can notify the network administrator in case of outages via email, pager or other alarms.

### 2.2.2 Ethereal

The Monitoring Network by Capturing Packets Using C++ application is based on Ethereal. Ethereal is a network packet analyzer. A network packet analyzer will try to capture network packets and tries to display that data packets as detailed as possible. It can be a standalone hardware device.

A network analyzer is a combination of software and hardware (*Angela D Orebaugh, 2004, Ethereal Packet Sniffing, 4*). It is composed of five basic parts:

- Hardware  
It is a software-based and work with operating system and network interface cards (NICs)

- Capture driver  
It is a part to capture the raw network traffic from the cable. This is a core of network analyzer and capturing data can't work without it.
- Buffer  
It is a component where captured data are stored.
- Real-time analysis  
It is a feature that analyzes the captured data.
- Decode  
It is a component used to display the captured data.

## 2.3 PROJECT METHODOLOGY

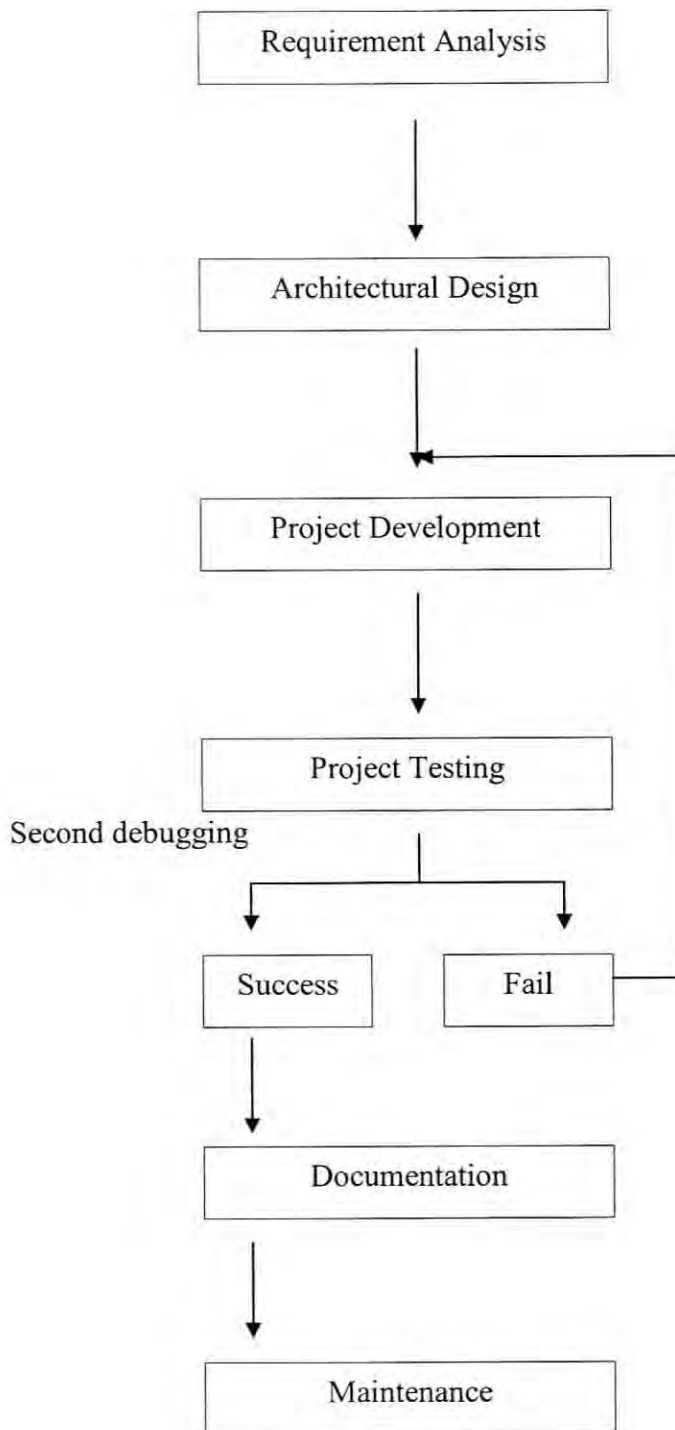
Methodology is a solution tree, or in a more general case, a directed acyclic graph, rooted at the problem statement and includes the system acceptance test that satisfies all of the goals in the problem statement. In terms of software methodology definition, a methodology is a study how to navigate through each phase of the software process model (determining data, control, or uses hierarchies, partitioning functions, and allocating requirements) and how to represent phase products (structure charts, stimulus-response threads, and state transition diagrams). Methodology is really important while developing certain software that act as a guidance that may affect the entire progress of the project. A suitable methodology usage may guide the developer through the whole project in order to meet the user requirement.



There's a lot of methodology type that has been created by the noble, such as OOAD (Object Oriented and Design), SSADM (Structured System Analysis and Design Method), SDLC (Software Development Life Cycle) and others that are not listed here. But in the network scheme, there's no specific methodology that can be referred as physical guidance on creating a good system. So, this methodology is created based on the planned project, Monitoring Network by Capturing Packets Using C++. Using the frame methodology, there will be some phases in the system that suits with the planned progress. 6 phases of the methodology:

1. Requirement Analysis
2. Architectural Design
3. Project Development
4. Testing
5. Documentation
6. Maintenance

**Figure 2.7** below shows the exact graphical methodology of the proposed system.



**Figure 2.1** : The Frame Methodology