**SOFTWARE REQUIREMENTS SPECIFICATION**

**FOR**

**PROTOCOL ANALYST**



**BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI**

**CS F303**

**Computer Networks**

Submitted By : -

Group number 5

AKHIL KUMAR AGRAWAL 2014A7PS024P

GAURANG BANSAL 2014A7PS128P

SUBHAM KUMAR 2014A7PS121P

28 February 2017

**PROBLEM STATEMENT**

Networks have become an integral part of people’s lives for sharing and accessing resources and communication. As time progresses, networks keep expanding and become more complex in their design and implementation. This causes many situations and problems which need to be addressed quickly and effectively as soon as possible so that the network is not compromised critically. Packet Analyzers help network administrators to identify and diagnose a variety of network problems like network congestion quickly and in managing various modules present in the network properly. They are required to get a deeper and more profound knowledge of how the various protocols function and how communication takes place between various nodes present within it and with the outside world. They are used to monitor suspicious activities like unauthorized access of important files present in the network. They form the basis for Intrusion Detection Systems, which monitor and scan the network constantly for possible attacks.

[**TABLE OF CONTENTS**](#_2amixmbuo369) 1

[**1 INTRODUCTION**](#_oyh3owqji4um) 2

[1.1 Purpose](#_jxyq5k69y0l1) 2

[1.2 Scope](#_ppiw0q922bay) 2

[1.3 Definitions, acronyms and abbreviations](#_a5es7l3mwvo0) 2

[1.4 References](#_cizmupyqmsz9) 3

[1.5 Overview](#_1izlbbu7xijs) 3

[**2 OVERALL DESCRIPTION**](#_gfrisxer8gpb) 3

[2.1 Product Perspective](#_y1fad2mzlh3i) 3

[2.1.1 User Interfaces](#_mqz9wpkzj6be) 3

[2.1.2 Hardware Interfaces](#_5rzo8r5iobsq) 3

[2.1.3 Software & System Interfaces](#_dl4v1xfqukke) 3

[2.1.4 Memory Constraints](#_fqc3ydlgn588) 3

[2.1.5 Operations](#_3q5r4tzcek4e) 4

[2.2 Product Functions](#_y5ndadj3yhca) 4

[2.3 User Characteristics](#_srluq2bjhgt3) 4

[2.4 Constraints](#_nj9j2d4y3e05) 4

[2.5 Assumptions and Dependencies](#_iqgc1totioau) 4

[**3 SPECIFIC REQUIREMENTS**](#_tfnpj4jnsrjp) 4

[3.1 External Interfaces](#_udjmnj8p645r) 4

[3.1.1 User Interfaces](#_ijtgdngo2o70) 4

[3.1.2 Hardware Interfaces](#_1t0hdxa67vfn) 5

[3.1.3 Software & System Interfaces](#_977tderdptaa) 5

[3.2 Functions](#_77l3grtqqy6m) 5

[3.3 Performance Requirements](#_sw7b0oe28itq) 6

[3.4 Logical Database Requirements](#_ck7p1mfml4vu) 6

[3.6 Software System Attributes](#_8k077refn7sd) 6

[3.6.1 Reliability](#_qd3zv82u6zn0) 6

[3.6.2 Availability](#_1e5eidgxlcs1) 6

[3.6.3 Security](#_rconcrkucw64) 6

[3.6.4 Maintainability](#_ww45t9xhbnsn) 6

[3.6.5 Portability](#_4vz4dtvnm82p) 6

[4. **Appendix**](#_6ydvbd7j5aaj) 7

[4.1 Extensions](#_iher6a373dlk) 7

# **1 INTRODUCTION**

## **1.1 Purpose**

The purpose of this document is to design a Packet Analysis Tool. Packet analyzers can help us understand and ascertain network characteristics such as how data is being communicated , discover network congestions, identify possible attacks or malicious activity and much more.

The SRS document provides a detailed overview of our project, its parameters and goals. This document describes the project’s target audience and its user interface, hardware and software requirements.

This document is intended for enthusiastic developers/readers who aspire to understand and implement a packet analyzer working and use it as a stepping stone for further development.

## **1.2 Scope**

Protocol Analyst is an application which works with the objective of capturing and analyzing packets at various layers, all the way from application layer to link layer protocols. The capturing of packets is done in a passive and non-promiscuous manner. The objective of this project to understand and design a basic model of a Network Protocol Analyser which can help the user to understand the working of Network Communication using Protocol Capturing. It is not intended for unethical and malicious activities related to sniffing and violating any user’s privacy.

## **1.3 Definitions, acronyms and abbreviations**

Protocols are standards and policies comprised of rules and procedures that define communication between two or more network devices. Some of the protocols used in this project are as follows

HTTP Hypertext Transfer Protocol

DNS Domain Name System

[POP](https://en.wikipedia.org/wiki/Post_Office_Protocol) Post Office Protocol

TCP Transmission Control Protocol

UDP User Datagram Protocol

ICMP Internet Control Message Protocol

IP Internet Protocol

ARP Address Resolution Protocol

## **1.4 References**

[1] S. Ansari et al (2003) Packet sniffing: a brief introduction

[2] Mohammed Abdul Qadeer et al (2010) Network Traffic Analysis and Intrusion Detection Using Packet Sniffer

[3] Abes Dabir and Ashraf Matrawy (2007) Bottleneck Analysis of Traffic Monitoring using Wireshark

[4] Md. Kamrul Hasan et al (2012) IEEE 802.11b Packet Analysis to Improve

Network Performance

## **1.5 Overview**

The SRS follows the recommendation of IEEE standard. The first section contains a brief introduction and scope of the Project. The second section covers in specification description related to project perspectives, functionalities, constraints, hardware and software system requirements. The penultimate section covers the in depth design and implementation details of the Network Analyst that will enable readers to understand the proper functionalities of Project. Index, Appendix, References,Acronyms have been included to provide support for the readers.

## 

# **2 OVERALL DESCRIPTION**

## **2.1 Product Perspective**

The details of the various interfaces present in Protocol Analyst are listed briefly

### **2.1.1 User Interfaces**

* Command Line Interface with options for the user
* Log Text Files in Working Directory containing the required output

### **2.1.2 Hardware Interfaces**

* Ethernet Adapter

#### **2.1.3 Software & System Interfaces**

* Operating System Ubuntu 16.04
* Libraries: netinet.h, arpa/inet.h, socket.h

### **2.1.4 Memory Constraints**

RAM: Minimum Requirement of 3MB

Disk Space: Minimum 5 MB

These are memory constraints for the application only.

#### **2.1.5 Operations**

The user will be able to choose the protocols from all packets he wants to analyze by selecting options provided in the Command Line Interface. User must access network .The user will asked to specify a particular time period for which he wants to capture the packets, during that time he will be able to see the real time update of packets being captured and read.

### **2.2 Product Functions**

* It passively captures the packets coming and going out of the system, and analyzes them for particular protocol headers at different levels.
* It provides details like MAC addresses of the source and client, IP addresses, ports,various flags and data dumps of the packets.

### **2.3 User Characteristics**

This following SRS is intended for interested developers/researchers/readers who are aspiring to understand and implement a packet analyzer.This also provides a stepping stone for further development in the field.

### **2.4 Constraints**

* The Internet Network must have sufficient bandwidth to have real time packets transfer to and from the client.
* The scope is limited to non promiscuous mode as promiscuous mode may lead to security issues and leakage of sensitive data.
* The number of interfaces considered in the project are limited considering the time scale.

### **2.5 Assumptions and Dependencies**

* We have assumed that the user has all authoritative access
* User is able to send and receive packets over the network either Intra Network or via Internet.
* The user is in running application in Non promiscuous mode.

# **3 SPECIFIC REQUIREMENTS**

## **3.1 External Interfaces**

### **3.1.1 User Interfaces**

* Command Line Interface with options for the user to select which protocols’ packets he wishes to analyze the relevant details.
* The user interface for the software shall be compatible with any browser such as Internet Explorer, Mozilla Firefox or Google Chrome using which user can access the internet.
* Log Text Files in working directory containing the required output, the details of each packet which was captured during the specified time interval.

### **3.1.2 Hardware Interfaces**

* Since the application must run over the Internet, all the hardware required for Terminal to be connected to network and able to receive Protocols over network will be part of Hardware Interface. Eg. LAN
* Ethernet Adapter compatible with the software interfaces required by the application.

### **3.1.3 Software & System Interfaces**

* Operating System  
  Name: Ubuntu  
  Version number: 16.04  
   Source: https://www.ubuntu.com/download  
  An operating system (OS) is a program that is initially loaded into the computer by a boot program and manages all the other programs and hardware resources in a computer in an effective manner.
* Libraries   
  Netinet.h and arpa/inet.h contain various constants and data structures which will be required for implementation of the Protocol Analyst. Socket.h contains the necessary functions and data declarations for various operations related to sockets.

## **3.2 Functions**

* The system shall take user inputs based on which protocols he wishes to analyze. He will be given options to select his preferences. He will also get choices what details under each protocol he wants, for instance sequence number and acknowledge number in TCP packets.
* The user communicates to a web server which sends and receives packets of the desired protocols for him to examine.
* The system captures and parses packets in a passive manner, i.e. it just listens to the particular interface and captures the information without modifying or redirecting the packets. The system also captures packets in non-promiscuous mode, it captures only the packets which are coming to and going from the node through the socket it is monitoring.
* Once the packet is captured, it is taken for analysis. The link layer header, in case of both the wireless and wired connections is decoded from hexadecimal numbers and read for obtaining details of the ARP protocol.
* The rest of the payload proceeds to the network layer, where the payload is examined for the header of the IP protocol. The required details are extracted and the payload(NPDU) is transferred to the transport layer.
* The transport layer protocols’ headers are examined and the necessary details are obtained while the Transport Protocol Data Unit(TPDU) goes further.
* At the application layer the headers are removed and decoded to get the details of HTTP and \*\*\* protocols. The payload now contains the actual data which the packet was carrying. This can also be decoded as per the requirements and rightful privileges of the user.
* All the desired information is available to the user in the log file, which contains details of each and every packet captured during the time interval set by the user.

## **3.3 Performance Requirements**

Same as specified under memory constraints

## **3.4 Logical Database Requirements**

A log file containing all the relevant details is generated

## **3.6 Software System Attributes**

### **3.6.1 Reliability**

The results of the Packet Analyzer can be verified with the details provided by the client and the server. The system should has a working internet connection.

### **3.6.2 Availability**

In case the application crashes, it will have to be restarted.

### **3.6.3 Security**

* Protocol Analyst will require an username and a password to access the options available for analyzing the packets.
* The options for choosing the protocols and their respective details are provided in the form of options, if anything other than the available options is entered, then the application throws an error.

### **3.6.4 Maintainability**

Not Applicable

### **3.6.5 Portability**

Packet Analyst is a lightweight packet analyzer which requires minimalistic hardware and software

Resources.It can be installed and operated with ease on almost every machine.

# 

# 

# 

# 

# 

# 

# **4. Appendix**

## **4.1 Extensions**

* Increase in number of Protocols at each layer that are to be analysed.
* Use of other interfaces such as Wifi to capture packets.
* Extension to Windows Platform using WinSock Library
* Providing Additional Graphical User Interface

**TimeLine**

|  |  |  |
| --- | --- | --- |
| **Topic** | **Date** | **Mentor** |
| Literature Search and Review | 18th February - 3rd March | Akhil Gaurang Subham |
| Analysis and Modeling | 13th March - 16th March | Subham |
| Pseudo Code Generation & Socket Networking | 17th March - 20th March | Gaurang |
| Designing a Simple Prototype | 20th March - 24th March | Shubham & Gaurang |
| Network monitor & Protocol Capturing | 25th March - 29th March | Akhil |
| Packet Filtering | 27th March - 1st April | Gaurang |
| Network Utilities & Code Optimization | 1st April - 5th April | Subham |
| Packet Log File & User Display | 7th April - 10th April | Akhil |
| System testing and evaluation | 10th April - 16th April | Gaurang |
| Final Project Report | 16th April - 18th April | Akhil |