**PACKET SNIFFER**

**A Project Report**

***Submitted By:***

**Aditya Yadav (201B024)**

**Lakshya Srivastava (201B147)**

**Neha Yadav (201B166)**

**Name of Supervisor:- Dr. Neelesh Kumar Jain**

***In partial Fulfilment for the award of the degree of***

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

**at**



**JAYPEE UNIVERSITY OF ENGINEERING & TECHNOLOGY,**

**A-B ROAD, RAGHOGARH, DT. GUNA– 473226, M.P. INDIA**

**Aug 2022 – Dec 2022**

**DECLARATION**

We hereby declare that the work reported in 5th semester Minor project entitled **“PACKET SNIFFER”**, in partial fulfilment for the award of the degree of B.Tech (CSE) submitted at Jaypee University of Engineering and Technology, Guna, as per the best of our knowledge and belief there is no infringement of intellectual property rights and copyright. In case of any violation, we will solely be responsible.

**Aditya Yadav (201B024)**

**Lakshya Srivastava (201B147)**

**Neha Yadav (201B166)**

**Jaypee University of Engineering and Technology,**

**Raghogarh, Guna – 473226**

**Date: 4/12/2022**

**CERTIFICATE**

This is to certify that the project titled “**PACKET SNIFFER**” is the bona fide work carried out by **Aditya Yadav, Lakshya Srivastava** and **Neha Yadav**, a student of B Tech (CSE) of Jaypee University of Engineering and Technology, Guna (M.P) during the academic year 2020-21, in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology (Computer Science and Engineering) and that the project has not formed the basis for the award previously of any other degree, diploma, fellowship or any other similar title.

**Signature of the Guide**

**Jaypee University of Engineering and Technology, Raghogarh,**

**Guna – 473226**

**Date: 4/12/2022**

**ABSTRACT**

Packet sniffing or packet capture software is extensively used as tools for protocol analysis and security. In protocol design research, such a tool comes handy in analysing, debugging and testing of a new protocol implementation. In Security, as is true for any tools, it may be used both as a positive way to detect intrusions or attacks on a system as well as in the malicious way to hack for private and personal data of others. Even though use of upper layer encryption techniques make it difficult to gather data directly, yet these tools are important in learning about existing sessions, collecting encrypted data to launch offline attacks to generate the encryption key and any such attack limited only by ones imagination. Hence, packet sniffer software is one of the most essential tools required to get started to be able to perform any of the above mentioned activities. The goal of our project is to write a packet sniffer “Net Vigilant”, capable of sniffing across wired and wireless interfaces and providing additional packet aggregation, filtering and analysis capabilities. The goal of the project is not to provide a novel approach towards sniffing on the network but rather to provide a basic understanding to the challenges involved in writing such a software and also to build up from the knowledge and experience gained to design more advanced security tools.

**ACKNOWLEDGEMENT**

We would like to express our gratitude and appreciation to all those who gave us the opportunity to complete this project. Special thanks is due to our supervisor **Dr. Neelesh Kumar Jain** whose help, stimulating suggestions and encouragement helped us in all the time of development process and in writing this report. We also sincerely thank you for the time spent proofreading and correcting my many mistakes. We would also like to thank our parents and friends who helped us a lot in finalising this project within the limited period. Last but not the least I am grateful to all the team members of **PACKET SNIFFER.**

**THANKING YOU**

**Aditya Yadav (201B024)**

**Lakshya Srivastava (201B147)**

**Neha Yadav (201B166)**

**LIST OF FIGURES**

**TABLE OF CONTENTS**

Title Page 1

Declaration of the Student 2

Certificate of the guide 3

Abstract 4

Acknowledgement 5

List of Figure 6

**Chapter– 1 INTRODUCTION**

1.1 Project Overview

1.2 Existing System

1.3 Uses of Packet Sniffer

1.4 Sniffing Tools

1.5 Sniffing Methods

1.6 Working of Packet Sniffer

**Chapter– 2 LITERATURE SURVEY**

2.1 Problem Definition

2.2 Proposed System

2.3 Feasibility Study

**Chapter– 3 SYSTEM ANALYSIS & DESIGN**

3.1 Requirement Specification

3.1.1 C Language

3.1.2 Linux

3.1.3 Socket

3.1.4 recvFrom() method

3.2 Hardware Specification

3.3 Software Specification

3.4 Flowcharts

**Chapter– 4 RESULTS/OUTPUTS**

**Chapter– 5 CONCLUSIONS/RECOMMENDATIONS**

**Chapter– 6 REFERENCE**

**CHAPTER– 1**

**INTRODUCTION**

**1.1 Project Overview**

This project aims at developing a Network Packet Sniffer. Network Packet Sniffer is a piece of software that monitors all network traffic. This is unlike standard network hosts that only receive traffic sent specifically to them. As data streams flow across the network, the sniffer captures each packet and eventually decodes and analyses its content. For network monitoring purposes it may also be desirable to monitor all data packets in a LAN and to mirror all packets passing through a shared bus.

This project will consist of three modules namely the User Interface module, Statistics module and Packet Analysis module. The User Interface module provides all the Graphical Interface components necessary for the user to interact with the System. The Analysis Module will analyse the incoming packets into a computer, identify them and pass the information into the Statistics module. Finally the statistics module does the necessary calculation based on the information and produces information that can be understood by the user.

This system is thus very useful to the users and a network administrator in particular who is generally responsible for monitoring things on a network.

This system is a network analyzer (also known as protocol analyzer & packet sniffer), it performs real-time packet capturing, 24x7 network monitoring, advanced protocol analysing, in-depth packet decoding, and automatic expert diagnosing. It allows you to get a clear view of the complex network, conduct packet level analysis, and troubleshoot network problems.

This system is useful for a network administrator who needs to identify, diagnose, and solve network problems, a company manager who wants to monitor user activities on the network and ensure that the corporation's communications assets are safe, or a consultant who has to quickly solve network problems for clients.

This provides an administrator with a full set of reports such as Summary view, Endpoints view, and Protocols view.

**NETWORK PACKET SNIFFER** is a desktop application which facilitates online monitoring of packets which are travelling over the network..

NETWORK PACKET SNIFFER I is a tool in which the packets source and destination addresses and other information is captured

The source and destination address protocols that are used by a connection are also monitored and detected

The graph or charts displayed on the valuation and need .

The monitored packet source and destination address is captured. This shows the traffic on a network.

**1.2 Existing System**

As a network administrator who needs to identify, diagnose, and solve network problems, a company manager who wants to monitor user activities on the network and ensure that the corporation's communications assets are safe, or a consultant who has to quickly solve network problems for clients. It is difficult to identify the problems if the network traffic is not tracked, as an administrator in general we depend on the analyzer provided by the operating system (if any) or the anti virus software that is installed to provide real-time network security. However, it is identified that these systems provide a specific set of reports which may not be enough for an administrator to trace all the problems. To handle these types of issues we want to implement a specific network analyzer that can track all the incoming and outgoing calls.

The Conversations tab allows us to monitor network traffic by each conversation and figure out which conversation has generated the largest network traffic.

**Drawbacks with the Existing System:**

* Administrators need to put lot of efforts to identify the traffic
* Time taking process.
* No possibility of automatic network control.
* Presence of an administrator is compulsory.

**1.3 Uses of Packet Sniffer**

Sniffing programs are found in two forms.

1) Commercial packet sniffers are used to help maintain networks.

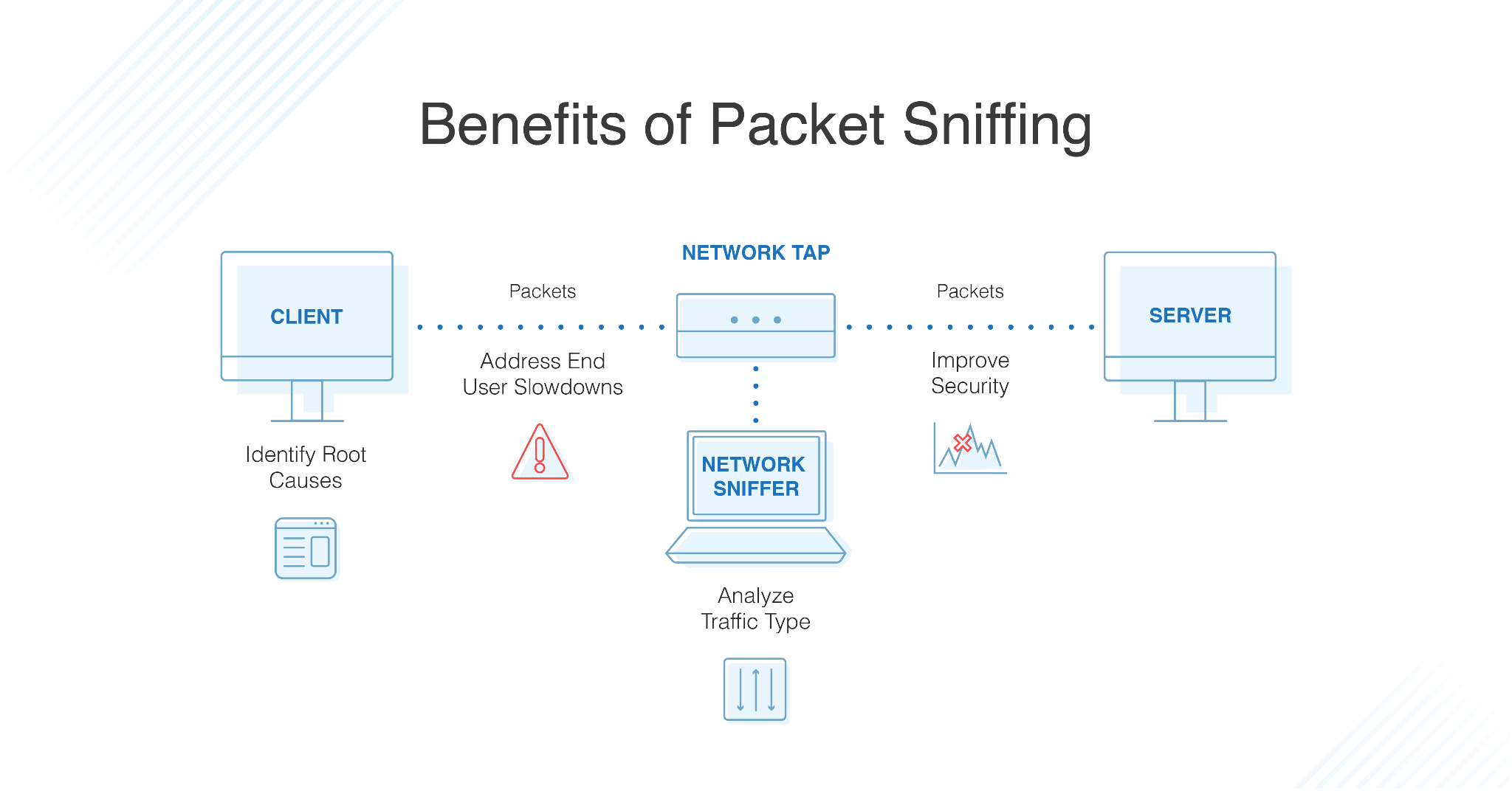
2) Underground packet sniffers are used by attackers to gain unauthorised access to remote hosts.

Listed below are some common uses of sniffing programs:

* Searching for clear-text usernames and passwords from the network.
* Conversion of network traffic into human readable form.
* Network analysis to find bottlenecks.
* Network intrusion detection to monitor for attackers.

Using a sniffer in an illegitimate way is considered a passive attack. It does not directly interface or connect to any other systems on the network. However, the computer that the sniffer is installed on could have been compromised using an active attack. The passive nature of sniffers is what makes detecting them so difficult. The following list describes a few reasons why intruders are using sniffers on the network:

* Capturing clear-text usernames and passwords
* Compromising proprietary information
* Capturing and replaying Voice over IP telephone conversations
* Mapping a network
* Passive OS fingerprinting



**Diagram 2 : Benefits of Packet Sniffing**

Obviously, these are illegal uses of a sniffer, unless you are a penetration tester whose job it is to find these types of weaknesses and report them to an organisation. For sniffing to occur, an intruder must first gain access to the communication cable of the systems that are of interest. This means being on the same shared network segment, or tapping into the cable somewhere between the paths of communications. If the intruder is not physically present at the target system or communications access point, there are still ways to sniff network traffic. These include:

* Breaking into a target computer and installing remotely controlled sniffing software.
* Breaking into a communications access point, such as an Internet Service Provider (ISP) and installing sniffing software.
* Locating/finding a system at the ISP that already has sniffing software installed.
* Using Social engineering to gain physical access at an ISP to install a packet sniffer.
* Redirecting communication to take a path that includes the intruder’s computer.

**1.4 Sniffing Tools**

* **tcpdump:**

Tcpdump is a powerful tool that allows us to sniff network packets and make some statistical analysis out of those dumps. One major drawback to tcpdump is the size of the flat file containing the text output. But tcpdump allows us to precisely see all the traffic and enables us to create statistical monitoring scripts.

* **sniffit:**

Robust packet sniffer with good filtering.

* **Ethereal:**

A free network protocol analyzer for UNIX and Windows. It allows you to examine data from a live network or from a capture file on disk.

* **Hunt:**

The main goal of the HUNT project is to develop tools for exploiting well- known weaknesses in the TCP/IP protocol suite.

* **Dsniff:**

Dsniff is a collection of tools for network auditing and penetration testing. dsniff, filesnarf, mailsnarf, msgsnarf, urlsnarf, and webspy passively monitor a network for interesting data (passwords, e-mail, files, etc.). arpspoof, dnsspoof, and macof facilitate the interception of network traffic normally unavailable to an attacker (e.g, due to layer-2 switching), sshmitm and webmitm implement active monkey-in-the-middle attacks against redirected SSH and HTTPS sessions by exploiting weak bindings in ad-hoc PKI.

* **IP spoofing:**

When the sniffing program is on a segment between two communicating end points, the intruder can impersonate one end in order to hijack the connection. This is often combined with a denial of service (DoS) attack against the forged address so they don't interfere anymore.

**1.5 Sniffing Methods**

There are three types of sniffing methods. Some methods work in non-switched networks while others work in switched networks. The sniffing methods are: IP-based sniffing, MAC-based sniffing, and ARP-based sniffing.

**IP-based sniffing:**

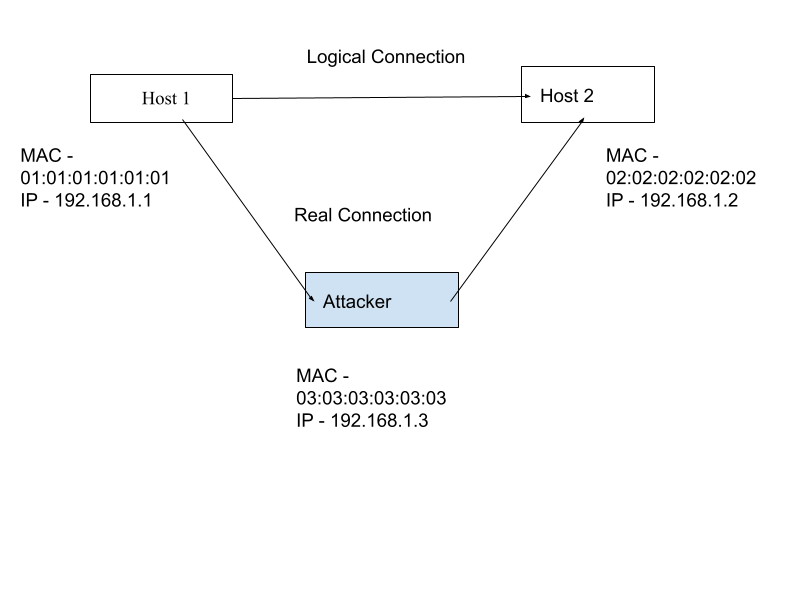
This is the original way of packet sniffing. It works by putting the network card into promiscuous mode and sniffing all packets matching the IP address filter. Normally, the IP address filter isn't set so it can capture all the packets. This method only works in non- switched networks.

**MAC-based sniffing:**

This method works by putting the network card into promiscuous mode and sniffing all packets matching the MAC address filter.

**ARP-based sniffing:**

This method works a little differently. It doesn't put the network card into promiscuous mode. This isn't necessary because ARP packets will be sent to us. This happens because the ARP protocol is stateless. Because of this, sniffing can be done on a switched network. To perform this kind of sniffing, you first have to poison the ARP cache of the two hosts that you want to sniff, identifying yourself as the other host in the connection. Once the ARP caches are poisoned, the two hosts start their connection, but instead of sending the traffic directly to the other host it gets sent to us. We then log the traffic and forward it to the real intended host on the other side of the connection. This is called a man-in-the-middle attack. See Diagram 1 for a general idea of the way it works.

 **Diagram 2 : ARP Sniffing Method**

**1.6 Working of Packet Sniffer**

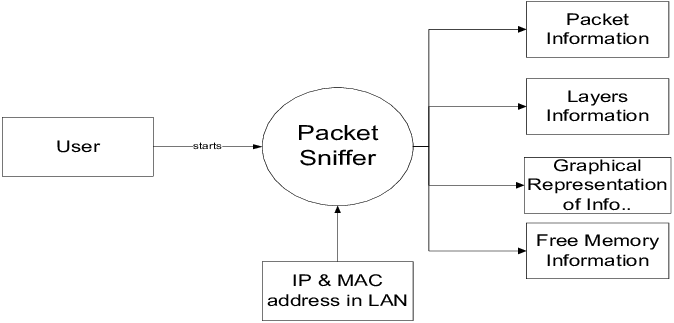
A packet sniffer works by looking at every packet sent in the network, including packets not intended for itself. This is accomplished in a variety of ways. These sniffing methods will be described below. Sniffers also work differently depending on the type of network they are in

**Shared Ethernet:**

In a shared Ethernet environment, all hosts are connected to the same bus and compete with one another for bandwidth. In such an environment packets meant for one machine are received by all the other machines. Thus, any machine in such an environment placed in promiscuous mode will be able to capture packets meant for other machines and can therefore listen to all the traffic on the network.

**Switched Ethernet:**

An Ethernet environment in which the hosts are connected to a switch instead of a hub is called a Switched Ethernet. The switch maintains a table keeping track of each computer's MAC address and delivers packets destined for a particular machine to the port on which that machine is connected. The switch is an intelligent device that sends packets to the destined computer only and does not broadcast to all the machines on the network, as in the previous case. This switched Ethernet environment was intended for better network performance, but as an added benefit, a machine in promiscuous mode will not work here. As a result of this, most network administrators assume that sniffers don't work in a Switched Environment.



**Diagram 3 : Working of Packet Sniffer**

**CHAPTER– 2**

**LITERATURE SURVEY**

**2.1 Problem Definition**

A packet sniffer — also known as a packet analyzer, protocol analyzer or network analyzer — is a piece of hardware or software used to monitor network traffic. Sniffers work by examining streams of data packets that flow between computers on a network as well as between networked computers and the larger Internet.

Packet sniffers can be used on both wired and wireless networks — their efficacy depends on how much they are able to "see" as a result of network security protocols. On a wired network, sniffers might have access to the packets of every connected machine or may be limited by the placement of network switches. On a wireless network, most sniffers can only scan one channel at a time, but the use of multiple wireless interfaces can expand this capability.

**2.2 Proposed System**

As a network analyzer (as a. packet sniffer), this system makes it easy for us to monitor and analyse network traffic in its intuitive and information-rich tab views. With this system network traffic monitor feature, we can quickly identify network bottlenecks and detect network abnormalities. This article is to discuss how we can monitor network traffic with this network traffic monitor feature.

This system provides a Summary view that provides general information of the entire network or the selected node in the explorer. In Summary view we can get a quick view of the total traffic, real-time traffic, broadcast traffic, multicast traffic and so on. When we switch among the nodes from the explorer, corresponding traffic information will be provided.

The Endpoints view; we can monitor network traffic information of each node, both local and remote. With its easy sorting feature we can easily find out which host is generating or has generated the largest traffic.

The Protocols view will list all protocols applied in network transmission. In Protocols view we can monitor network traffic by each protocol. By analysing network traffic by protocol, we can understand what applications are using the network bandwidth, for example "http" protocol stands for website browsing, "pop3" stands for email, etc.

**Advantages with the proposed system:**

* Network Admin can monitor the packets anywhere throughout the world.
* Traffic can be controlled
* System performance will be increased
* Immediate generation of reports on demand.
* Graphical data is available to analyse the network.

**2.3 Feasibility Study**

* *Reliability*

This software has been tested and found to be reliable.

* *Availability*

Since this system has been tested for defects and fixed, the downtime is low and therefore is available.

* *Security*

Security features are also provided by .NET.

* *Maintainability*

Ease of maintenance is one of the advantages of .NET.

* *Portability*

This application is supported by the following Operating Systems.

Windows 9X, Windows 2000, Windows Vista.

* *Performance*

Performance of this application is good on a small network. It has yet to be tested on a larger network.

**CHAPTER– 3**

**SYSTEM ANALYSIS & DESIGN**

**3.1 Requirement Specification**

**3.1.1 C Language**

[C](https://www.simplilearn.com/c-programming-article) is a procedural [programming language](https://www.simplilearn.com/tutorials/programming-tutorial/first-programming-language) with a static system that has the functionality of structured programming, recursion, and lexical variable scoping. C was created with constructs that transfer well to common hardware instructions. It has a long history of use in programs that were previously written in assembly language.

C programming language is a machine-independent programming language that is mainly used to create many types of applications and operating systems such as Windows, and other complicated programs such as the Oracle database, Git, [Python](https://www.simplilearn.com/learn-the-basics-of-python-article) interpreter, and games and is considered a programming foundation in the process of learning any other programming language.

**3.1.2 Linux**

Linux is a Unix-like, open source and community-developed operating system (OS) for computers, servers, mainframes, mobile devices and [embedded devices](https://www.techtarget.com/whatis/definition/embedded-device). It is supported on almost every major computer platform, including x86, ARM and [SPARC](https://searchservervirtualization.techtarget.com/definition/SPARC), making it one of the most widely supported operating systems.

### 

### 

### 

### 

### **The pros and cons of using Linux**

**Some advantages of using Linux include:**

* **Open source software.** The Linux kernel is released under the GNU GPL open source software licence. Most distros include hundreds of applications, with many options in almost every category. Many distributions also include proprietary software, such as device drivers provided by manufacturers, to support their hardware.
* **Licensing costs.** Unlike Microsoft Windows or Apple macOS, Linux has no explicit licensing fees. While system support is available for a fee from many Linux vendors, the OS itself is free to copy and use. Some IT organisations have increased their savings by switching their server software from a commercial OS to Linux.
* **Reliability.** Linux is considered a reliable OS and is well-supported with security patches. Linux is also considered to be stable, meaning it can run in most circumstances. Linux also copes with errors when running software and unexpected input.
* **Backward compatibility.** Linux and other open source software tend to be updated frequently for security and functional patches, while retaining core functionality. Configurations and shell scripts are likely to work unchanged even when software updates are applied. Unlike commercial software vendors that roll out new versions of their OSes along with new ways to work, Linux and open source applications generally don't change their modes of operation with new releases.
* **Many choices.** Between the hundreds of available distributions, thousands of applications and almost infinite options for configuring, compiling and running Linux on almost any hardware platform, it is possible to optimise Linux for almost any application.

**Some disadvantages of using Linux include:**

* **Lack of established standard.** There is no *standard* version of Linux, which may be good for optimising Linux for particular applications, but less so for deploying standardised server or desktop images. The wide range of options can complicate support as a result.
* **Support costs.** While an organisation can acquire Linux freely without licensing fees, support is not free. Most enterprise Linux distributors like SUSE and Red Hat offer support contracts. Depending on the circumstances, these licence fees can reduce savings significantly.
* **Proprietary software.** Desktop productivity software like Microsoft Office cannot be used on Linux desktops, and other proprietary software may be unavailable for Linux platforms.
* **Unsupported hardware.** While many hardware manufacturers make Linux device drivers available for their products, many do not.
* **Steep learning curve.** Many users struggle to learn to use the Linux desktop or Linux-based applications.

In some cases, the same Linux attribute can be either an advantage or disadvantage. For example, having many options for customising the Linux OS is advantageous for manufacturers looking for an embedded OS, but it is a disadvantage for enterprises that want a desktop OS that can be used by a wide range of end users.

**3.1.3 Socket**

A Socket is one endpoint of a two-way communication link between two programs running on a network.

**Why do we need a socket:**

Packet sockets are used to receive or send packets at the device driver (OSI layer 2) level. Here we are using **raw socket** as it provides direct access to lower layer protocols such as Ip or ICMP etc.

A raw socket is used to receive raw packets. This means the packet received at the ethernet layer will directly pass to the raw socket.

To create a raw socket,

sock\_raw = socket(AF\_INET , SOCK\_RAW , IPPROTO\_TCP);

Both reading and writing to a raw socket require creating a raw socket first.

-> Here the first parameter is the INET family raw socket. It describes the address of the family of the socket. Here we have used AF\_INET which is the address family constant of Ip version 4.

-> The next parameter passed is the type of the socket. Here we have used SOCK\_RAW that is used to define raw sockets.

-> The parameter is the protocol of the packet. The protocol number is defined by the Internet Assigned Numbers Authority(IANA).

We have to be aware of the family of the socket; then we can only choose a protocol.

Here we have used IPROTO\_TCP .i.e. our sniffer will currently only catch the TCP packets.

**3.1.4 recvFrom() method**

data\_size = recvfrom(sock\_raw , buffer , 65536 , 0 , &saddr , &saddr\_size);

-> The recvfrom() function reads the incoming data and captures the address from which the data was sent.

-> It is present in the socket module that helps us to receive all the data from the socket.

-> From recvfrom() the data is saved in the buffer from where it is passed further to print the data present inside the packet.

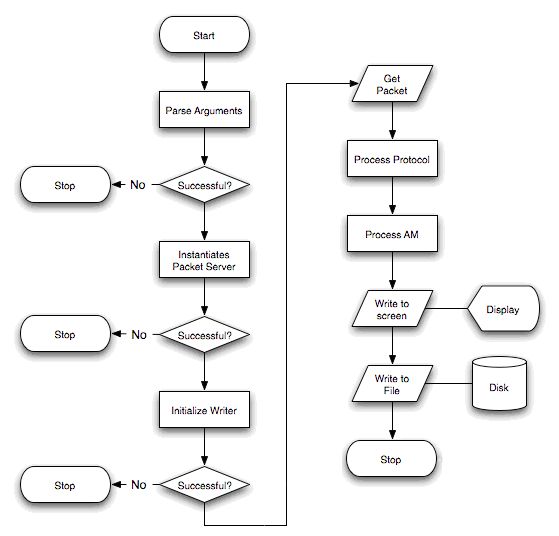
**3.2 Hardware Specification**

* Mouse is required for use of application
* Keyboard is required for use of application
* Monitor is required for use of application
* Network interface card is required for packet capture
* 1.5 MB of hard disk space.
* 1 GB RAM (Random Access Memory).

**3.3 Software Specification**

* C Interpreter
* Operating System: Linux- Ubuntu 16.04 to 17.10

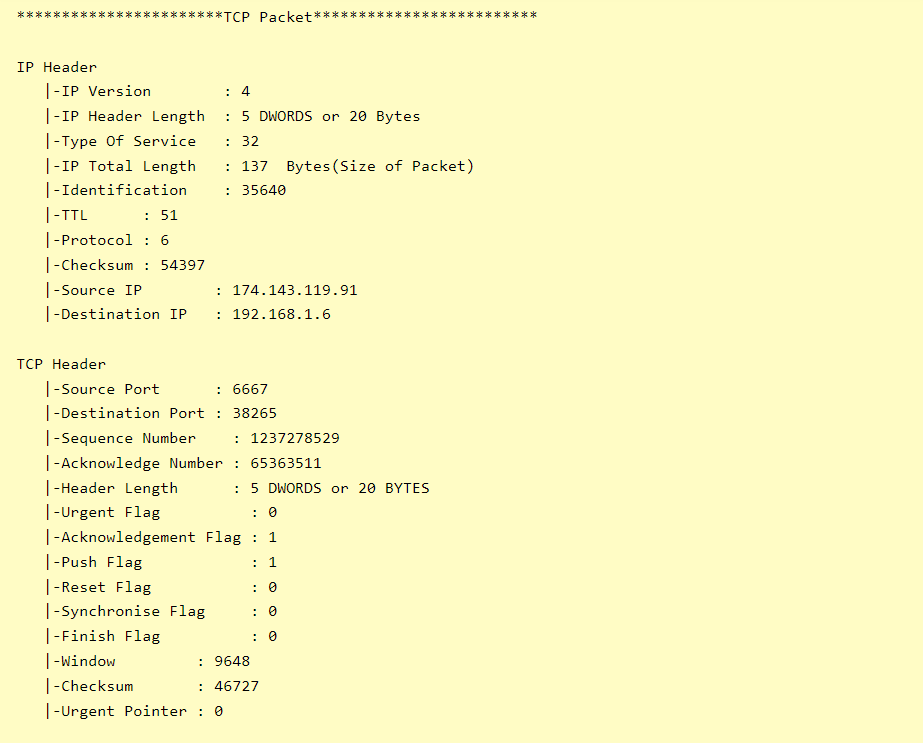
**3.4 Flowchart**

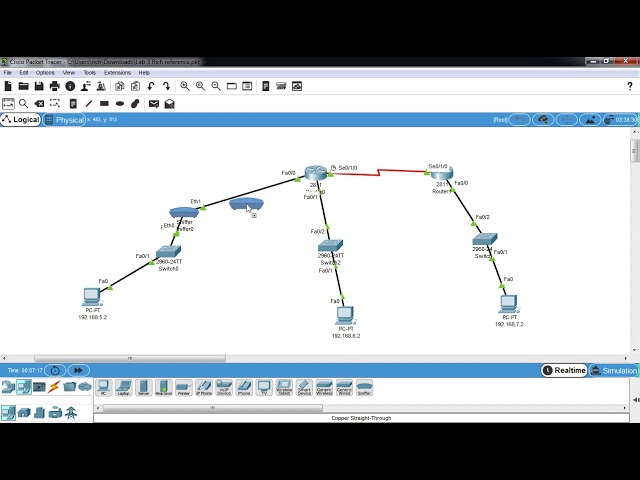
****

**Diagram 4 : Flowchart**

**Chapter– 4**

**RESULTS/OUTPUTS**

****

****

**Chapter– 5**

**CONCLUSIONS/RECOMMENDATIONS**

The term packet sniffer might sound a little fishy and suspicious, but it isn’t anything like that. In network management, packet sniffing plays a very crucial role. Network managers and technicians use packet sniffers to diagnose underlying problems in their networks. So, a packet sniffer is essentially a tool that aids in monitoring network traffic and troubleshooting a network.Proper use of packet sniffers can help clean up network traffic and limit malware infections; to protect against malicious use, however, intelligent security software is required.

**Chapter– 6**

**REFERENCE**