National University of Computer and Emerging Sciences

Internet Traffic Classification Tool

Project Advisor:

Sir Ibrahim Nadir

Group Member:

Mairaj Muhammad	17L-4222
Malik Abdulaziz	17L-4227
Usama Arshad	17L-4244

B.S. Computer Science

Computer Network Project: May 2020

Department of Computer Science

FAST NU, Lahore, Pakistan

Table of Contents

Abstract:

This report is written to brief about the internet traffic classification tool we developed as a part of our final semester project. The project proposed was an internet traffic classification tool. It demanded the monitoring of all sorts of packets the ethernet uses in order to establish connection with different hosts and eventually convey the data. A software we developed that is able to identify and quantize all sorts of packets. It possesses a user friendly GUI (Graphical User Interface). The traffic captured through tcpdump/windump is required to be classified on the basis of different port numbers or types (HTTP/s,VoIP,DNS,FTP,RTP etc.). Radio buttons were created to classify the packets and provided a custom port number to capture traffic from. A thread was utilized to make the graphical user interface interactive while actually capturing and updating traffic on the output text area (Java Swing). We made several easy to use functionality based buttons to start, pause, clear the traffic. Finally, the functionality of showing details of each packet (source, destination, ip, header etc.) was shown upon entering the packet number of each packet.

Chapter 1: Introduction

A packet analyzer computer program that runs through the command line interface is tcpdump. It provides its users with flexibility to see the TCP/IP and some other types of packets being received or transferred over the internet through which the system is connected usually. TCPdump usually works on Unix-like operating systems. Some of which are Linux, FreeBSD, macOS, OpenWrt, macOS, NetBSD, Solaris. In these systems, tcpdump usually uses the libpcap library to get started with capturing packets. The similar version for tcpdump incase of windows in WinPcap, the windows version of libpcap. Tcpdump was originally written in the C programming language. The contents of network packets are printed by Tcpdump. It directly can read packets from network interface cards or from packet files. It can also be used to write packets. If necessary privileges are given tcpdump can read encrypted traffic such as to view login IDs and password etc. A user must have superuser privileges to use tcpdump in unix like operating systems to get started with packet capturing

mechanisms. It provides its users the flexibility to classify packets on the basis of port numbers and process the packets and resolve into its types like TCPpacket, UDPPacket etc.

Chapter 2: Project Requirements

2.1 Java Development Kit:

It is a development kit to develop java softwares. It comprises of following components:

- A Java Runtime Environment
- Javac which is Java Compiler
- Javadoc which is a document generator
- An interpreter

Along with other things as well.

2.2 Java RunTime Environment:

JRE may also be written as “Java RTE”, stands for “Java Runtime Environment” and for executing a Java application the Java Runtime Environment provides the minimum requirements; it consists of:

- The supporting files
- Core classes
- The Java Virtual Machine (JVM).

2.3 Jpcap:

It is a very popular library for packet sniffing and capturing in java. Jpcap can be used to capture packets from a network interface. Not only capturing but this library can also send packets of any format to the network layer. It also provides users with the functionalities like applying filters (like WireShark), interpreting TCP/UDP or Ethernet/Arp packets and saving and loading a packet capture from/to a file.

2.4 WinPcap:

For link-layer network access in Windows environments, WinPcap has been recognized as the industry's standard tool from a number of years, as it allows applications to transmit and capture packets bypassing the protocol stack, from network, and including kernel-level packet filtering, support for remote packet capture and a network statistics engine. For providing low-level network access, a driver that extends the operating system and a library that is used to easily access low-level network layers is used in WinPcap. The well-known libpcap Unix API's Windows version is also included in this library. WinPcap has been the filtering engine and packet capture for many commercial and open source networking tools, which includes network testers, traffic generators, sniffers, network intrusion detection systems, network monitors and protocol analyzers, thanks to WinPcap's set of features. like ntop, Snort, Nmap, and Wireshark, a number of these networking tools are used and known throughout the whole networking community. WinDump is the Windows version of the popular tcpdump tool, Winpcap is also the home of it. According to various complex rules WinDump can be used to save network traffic to disk, diagnose and watch network traffic.

2.5 Swing:

Swing is a GUI widget toolkit for Java. To provide a graphical user interface for Java programs there is an API known as Oracle's Java Foundation Classes, Swing is also a part of it .

Chapter 3: Methodology

We use windows operating system for development and installed Apache NetBeans on it, which is an integrated development environment (IDE) for Java. Then we included Jpcap and WinPcap libraries in the project, and their provided functions helped us in capturing the internet traffic packets. We analyzed the captured packets using functions provided by WinPcap library. Furthermore, after capturing and analyzing the internet traffic packets, we were concerned about the visualizing of our work, thanks to JAVA Swing library that includes a rich set of widgets which is a lightweight Graphical User Interface (GUI) toolkit. It

includes a package that lets us make GUI components for our Java application, and It is platform independent.

Chapter 4: Implementation

Implementation of this project was done in several steps which includes the installing the suitable internet development environment, adding the Java Development Kit and Java Runtime Environment. After that the project dependencies were added in the project properties. Finally, after all this work development phase starts and has been discussed as below:

4.1 Setting the Network Interface:

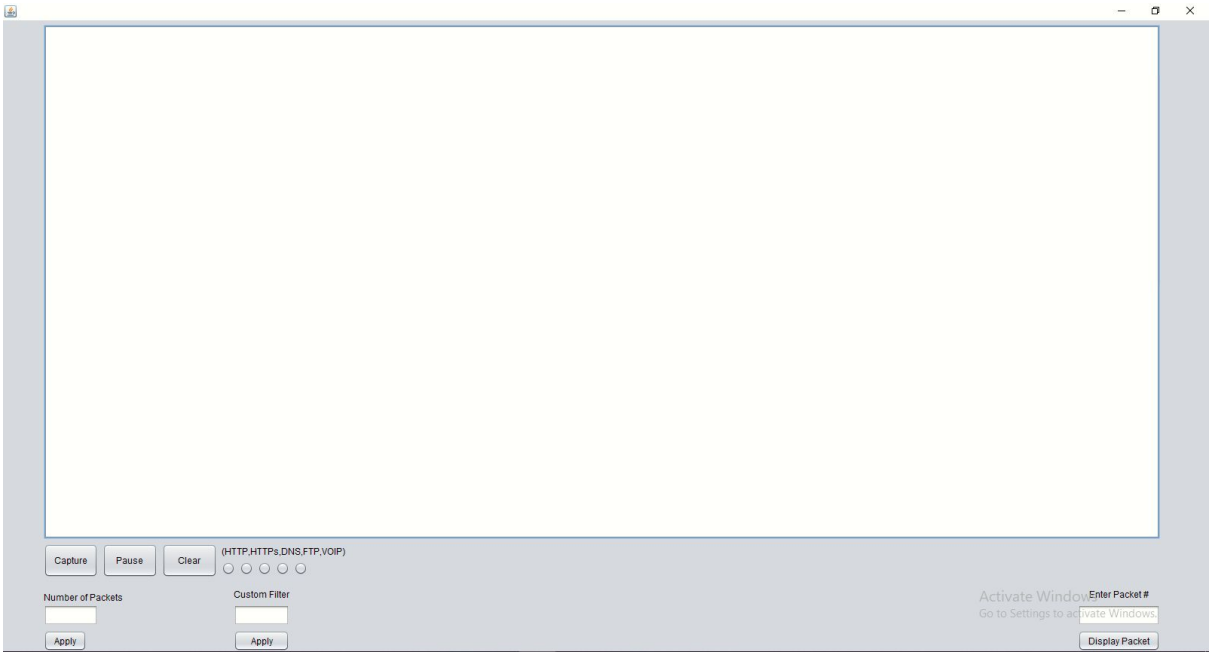
It was the first in the project development. In this step, we got the network interface from a builtin method of JpcapCaptor which is JpcapCaptor.getDeviceList(). It returns an array which consists of Network Interface Objects. Then to open a specific network interface, we use another method which is JpcapCaptor.openDevice (Network Interface Object, int snaplen, boolean promics, int to_ms). It accepts a network interface object, 'snaplen' which is the maximum number of data(in bytes) it can capture in one go, 'promics' which is if set true can capture packets even if host mac address is not same as the current PC and if set false it can capture only those packets which were sent or received by our PC only. The final parameter it accepts is to_ms which is to set a capture timeout in milliseconds.

4.2 Capturing Packets:

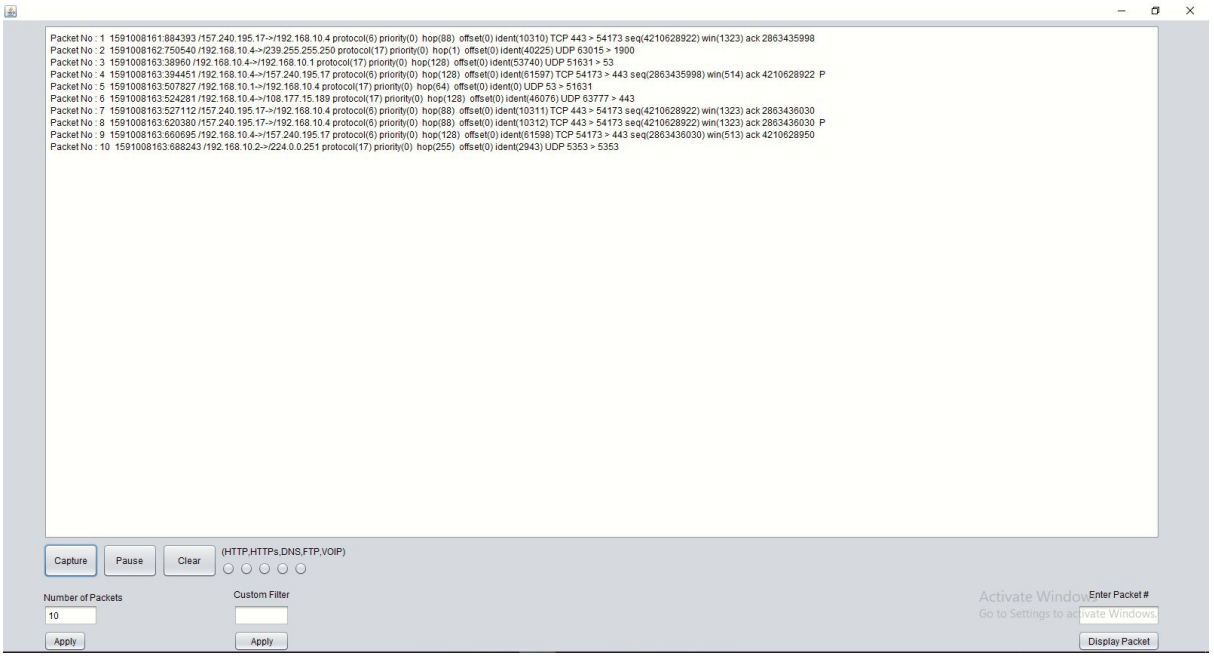
Packet Capturing is done by using a JpcapCaptor method which is getPacket(). It simply returns a captured packet which is after some processing displayed on the screen.

4.3 GUI Main Frame:

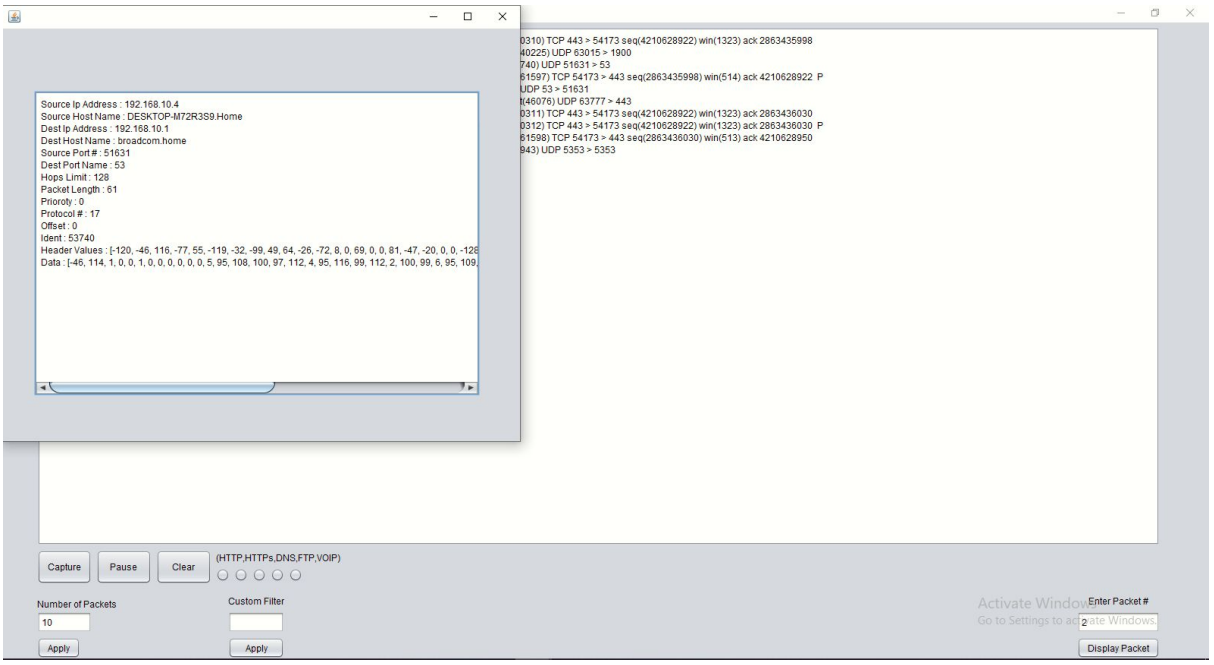
GUI was developed in JFrame. A java frame was created and a text area was created on the main frame on which the captured packets are to be displayed. To make GUI responsive, we made text fields and buttons, in which users can input various commands and can access various project functionalities like displaying some specific packet or applying some filter to it. Given below is the image of the GUI main frame.



In the above frame first user will input the maximum number of packets captured in one go, then the user selects some filter (if required) and presses the capture button. Then it starts displaying the packets till the limit of the maximum number of packets is reached.



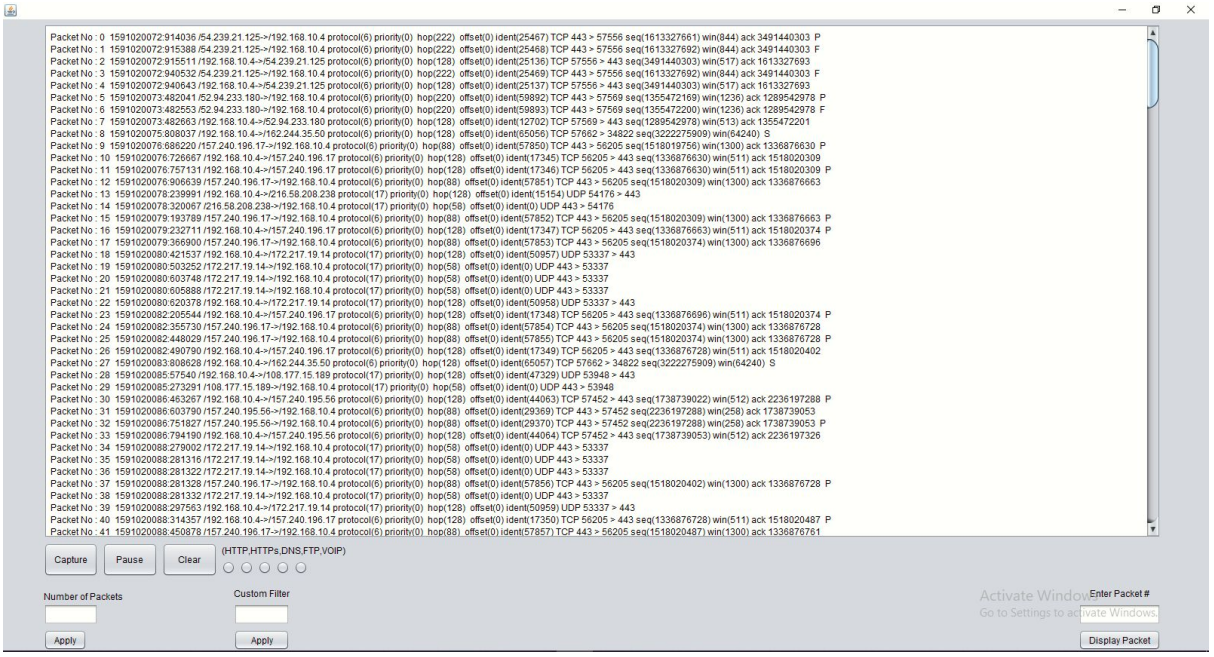
If some filter is selected it only captures the packets of that filter and then displays it. There is also a feature of printing the details of a specific packet. For this user inputs a packet number and then presses on the display packet button. It displays that packet on another frame as displayed below.



Chapter 5: Test Cases

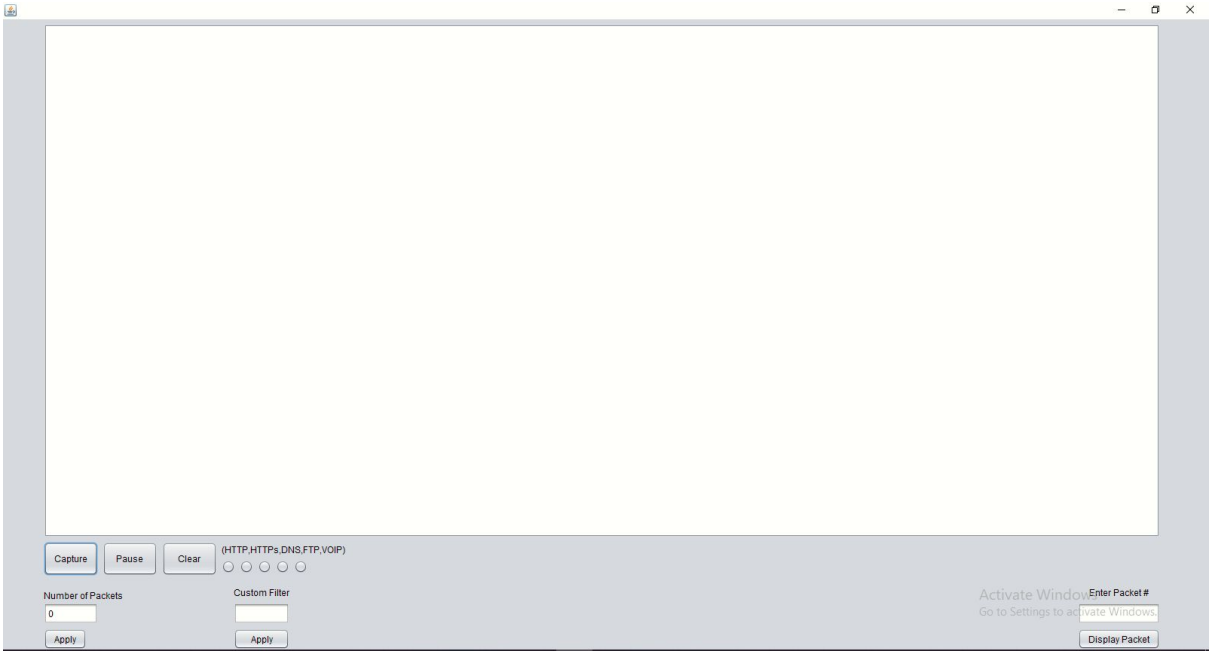
5.1 Test case 1:

Capturing packets when no maximum packet input is given:



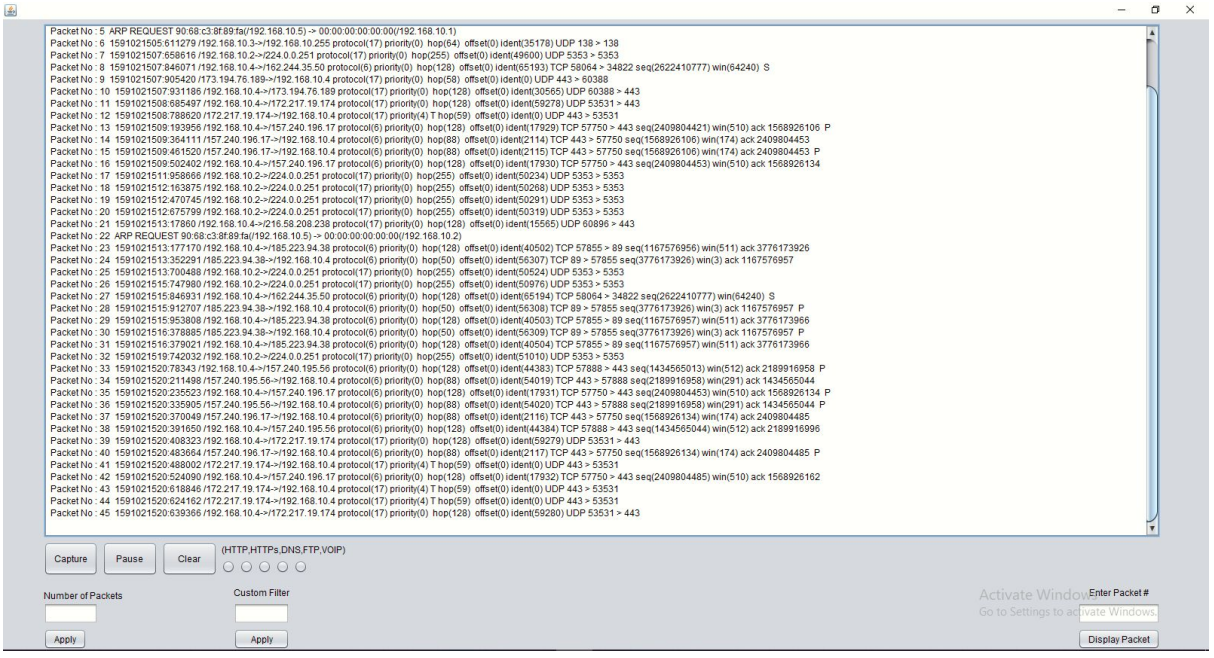
5.2 Test case 2:

Capture When Maximum Packet Input Given is 0.



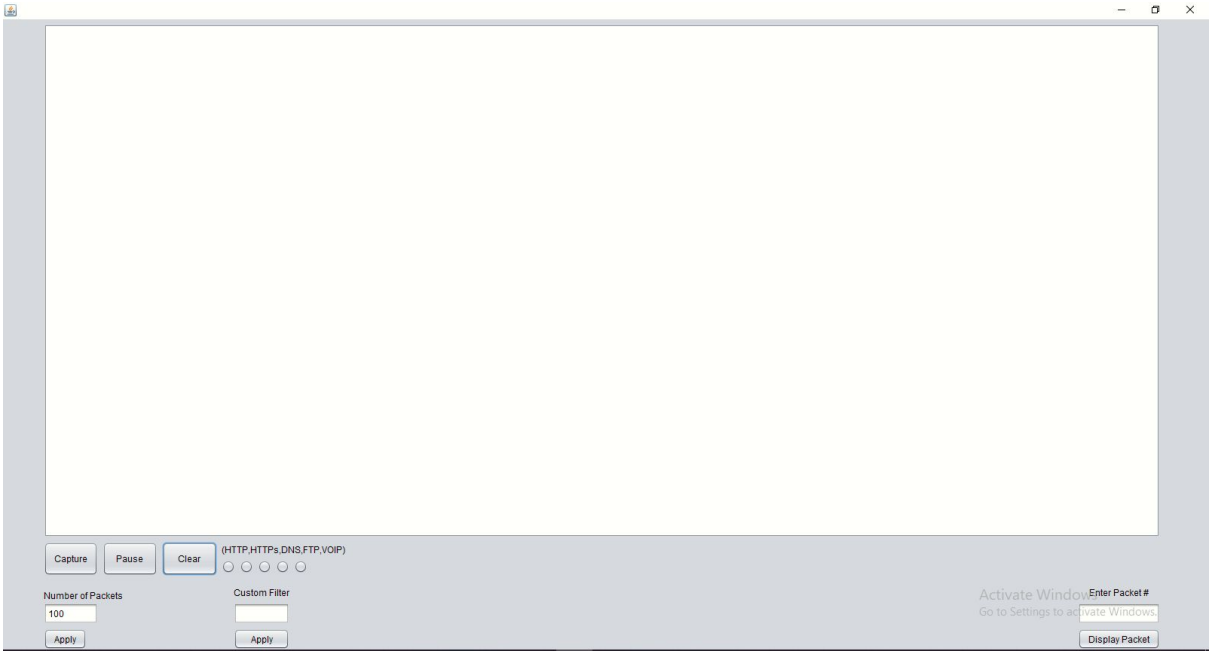
5.3 Test case 3:

Pressing the Pause button while packet capturing.



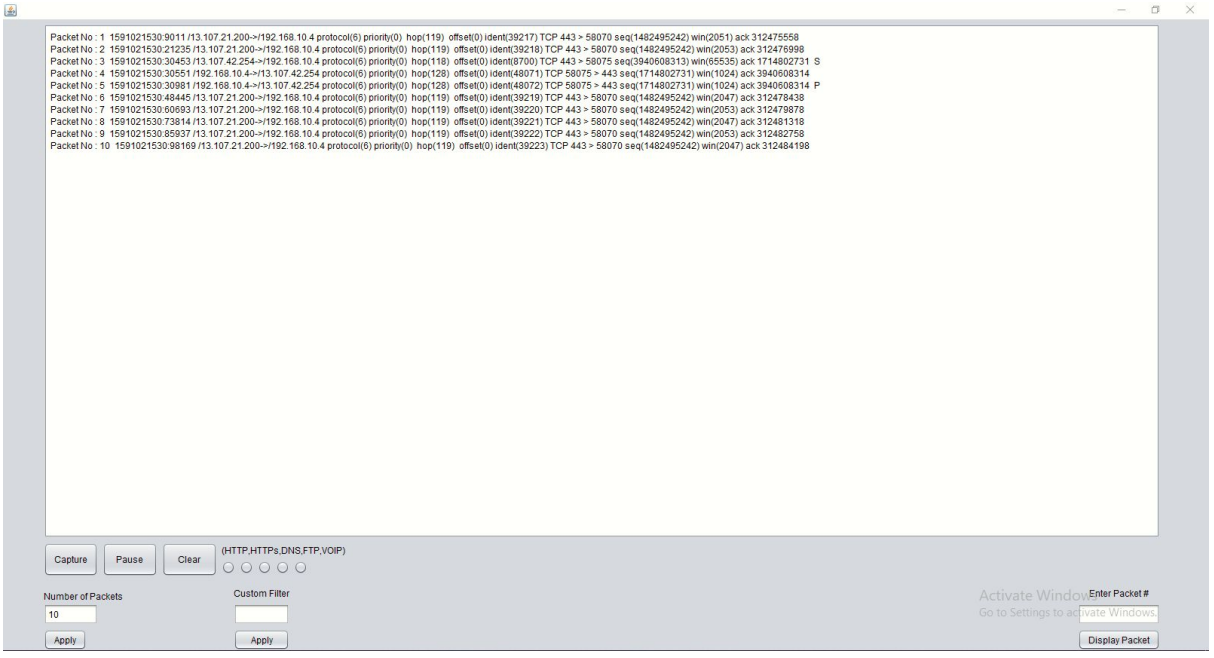
5.4 Test case 4:

Pressing the Clear button while capturing.



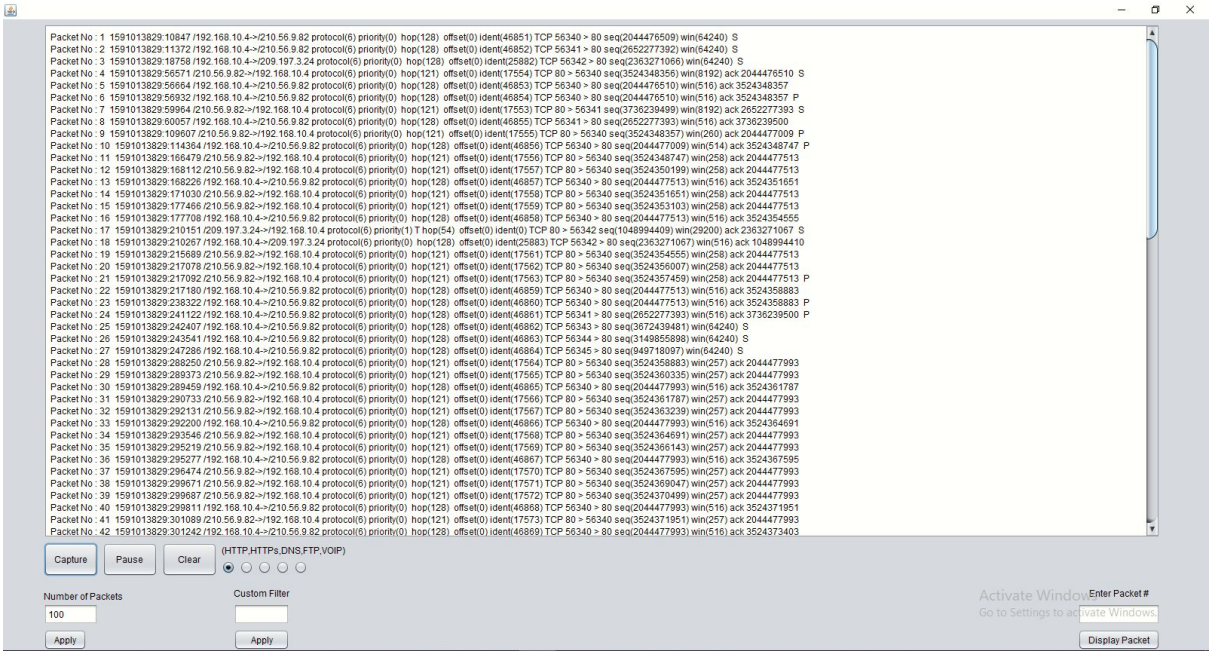
5.5 Test case 5:

Pressing the capture button after giving the inputs of the maximum packet limit.



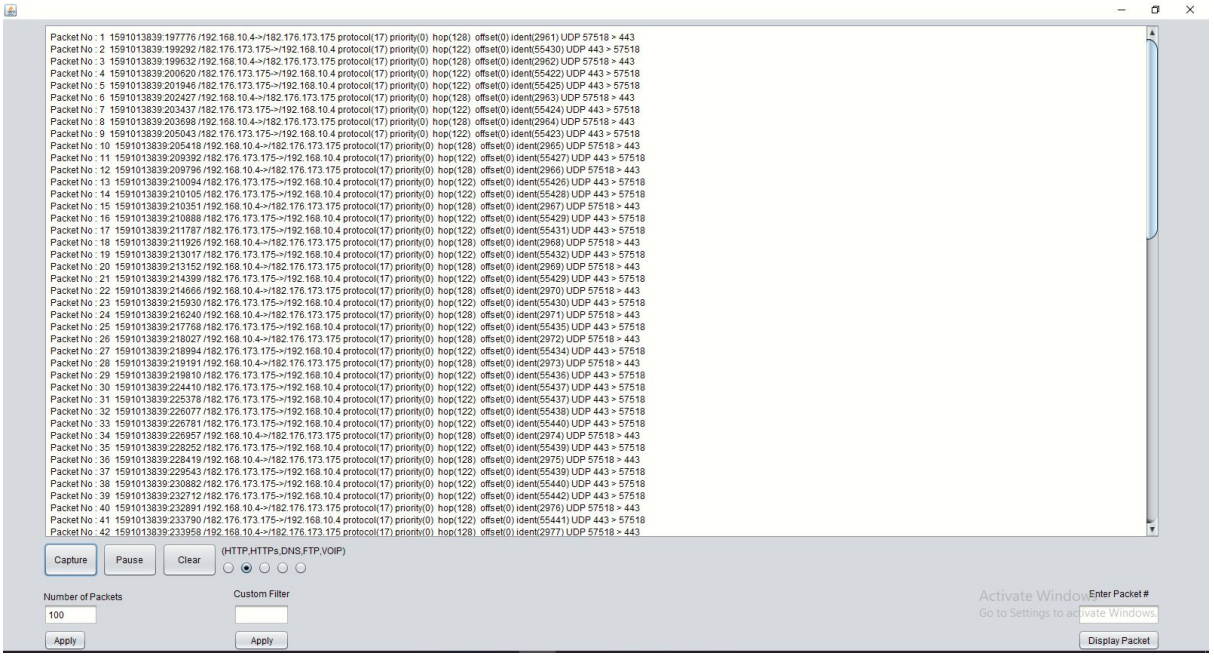
5.6 Test case 6:

Pressing capture while HTTP filter is set.



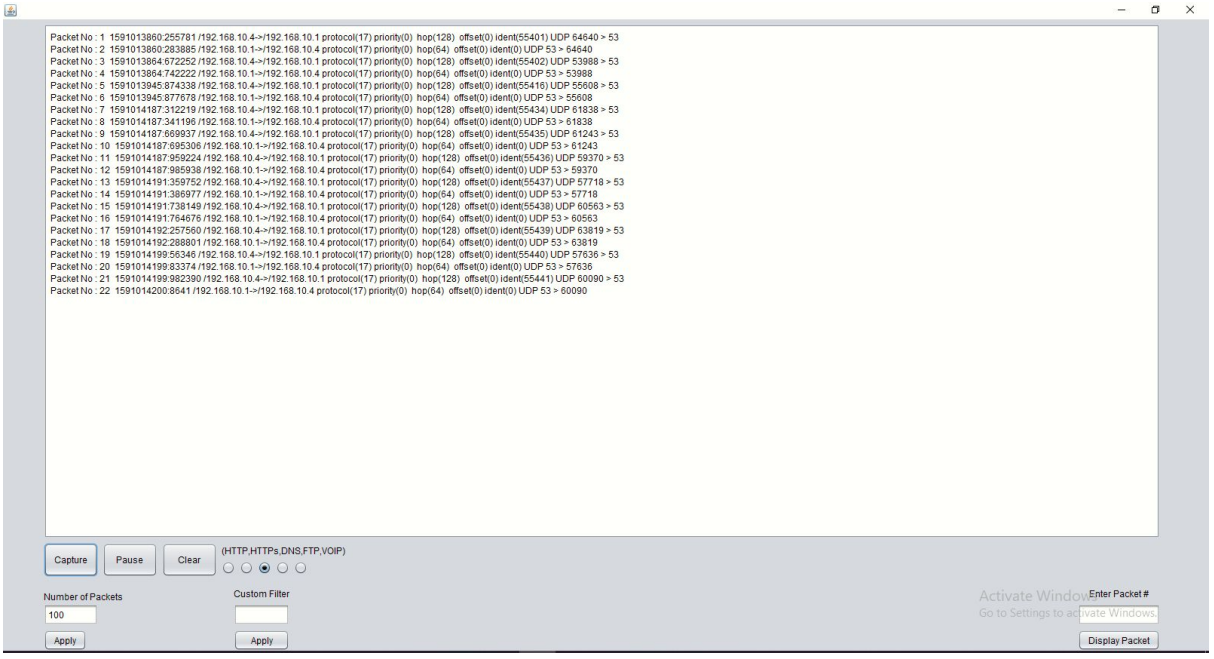
5.7 Test case 7:

Pressing capture while HTTPS filter is set.



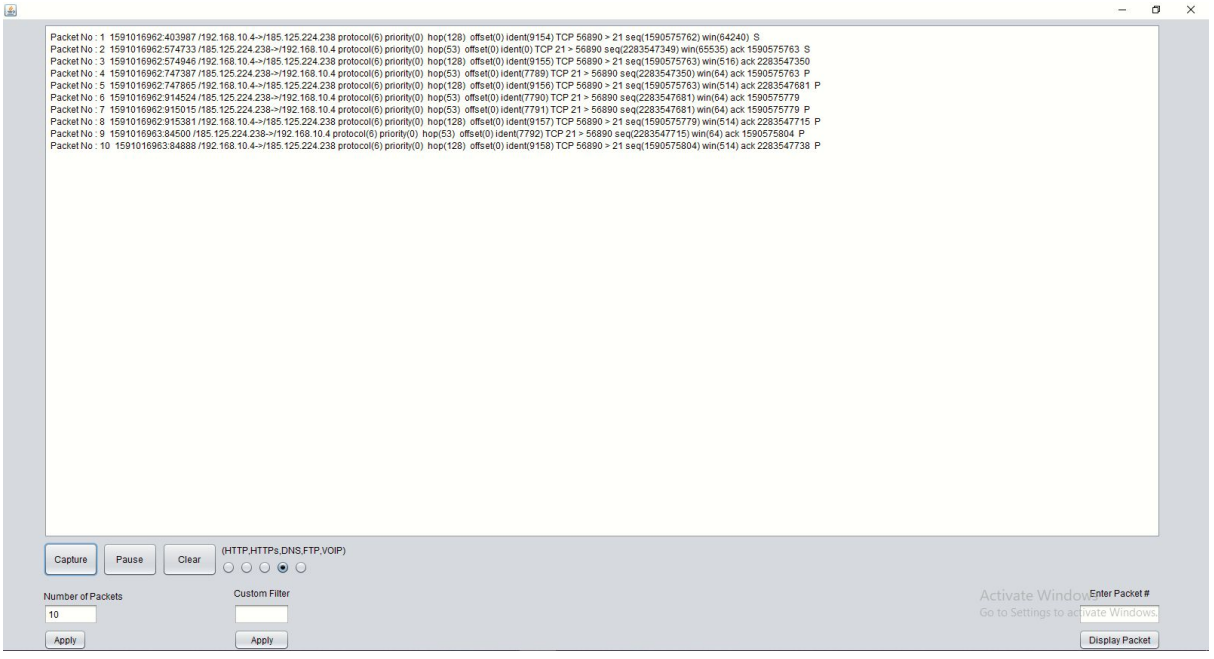
5,8 Test case 8:

Pressing capture while DNS filter is set.



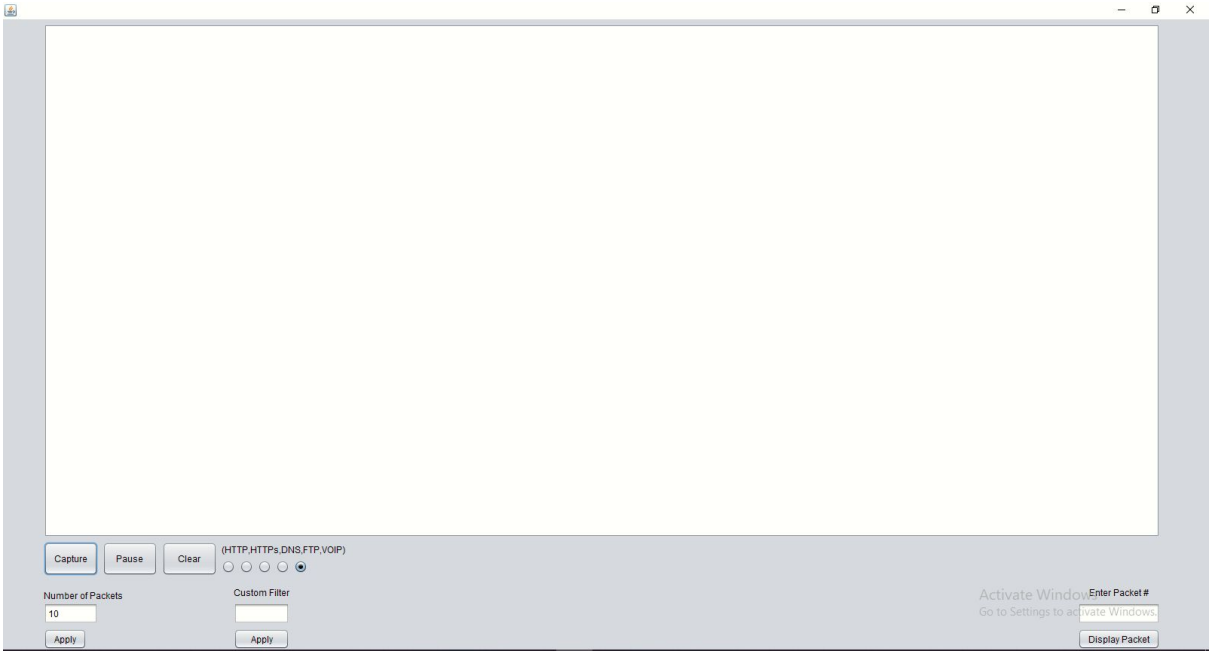
5.9 Test case 9:

Pressing capture while FTP filter is set.



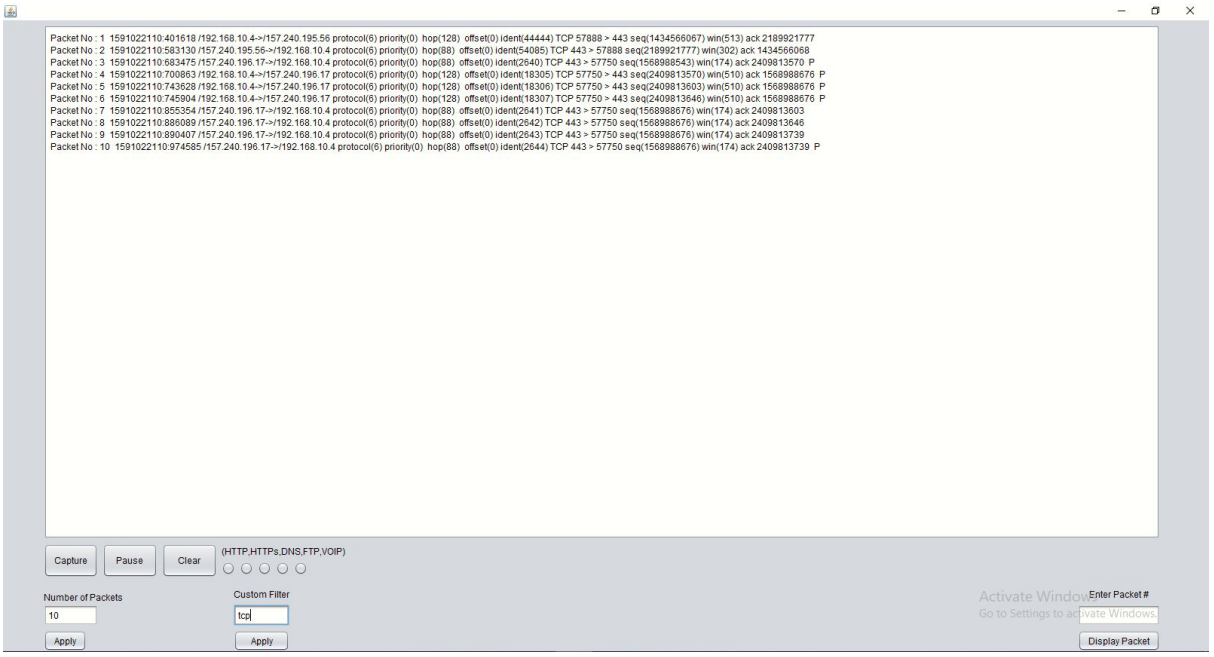
5.10 Test case 10:

Pressing capture while VoIP filter is set (No packet received at our machine).



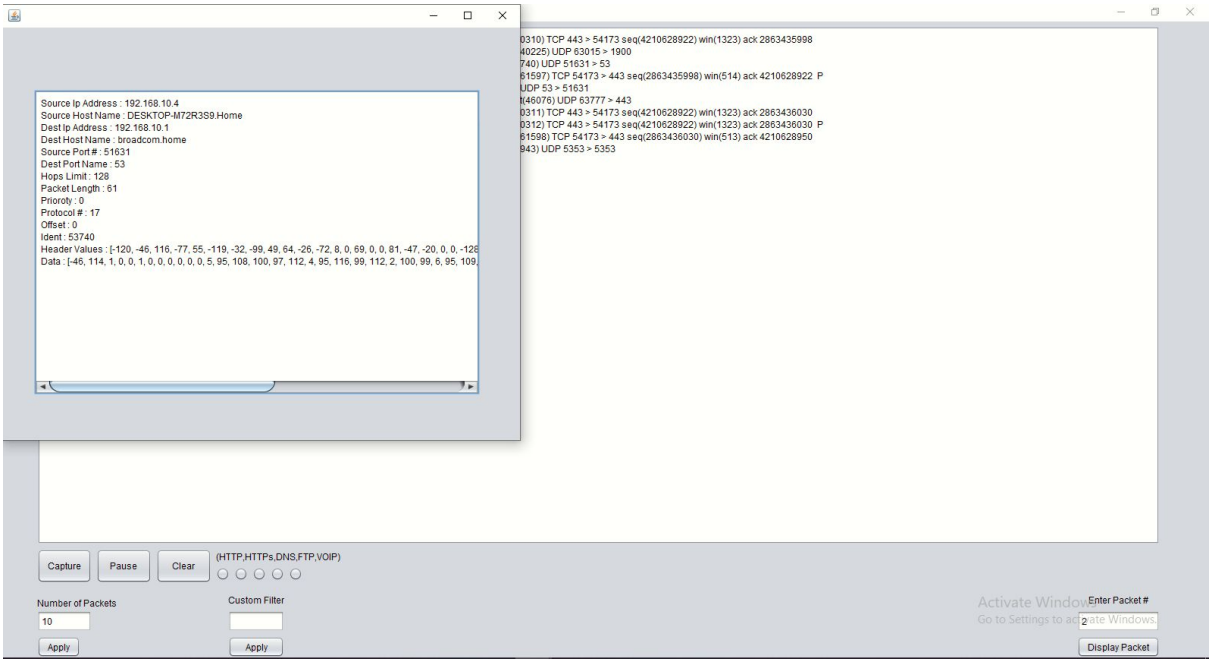
5.11 Test case 11:

Pressing capture while custom filter is set.



5.12 Test case 12:

Displaying a packet separately.



Chapter 6: Conclusion

A fully functioning Java Software was successfully developed and made to function based on its GUI (Graphical User Interface). Traffic has been successfully obtained / captured through the internet using Java Packet Capture (Jpcap) class. An instance of Java Packet Capture was made utilizable to capture the traffic from and resolve it into Packet types like TCP, UDP etc. Different types of filters were applied using custom port numbers and radio buttons. Network Interface list was obtained programmatically and passed as a parameter to find the networks within the system and then relevant network was passed to Java Packet Capture (Jpcap) to capture packets from. GUI (Graphical User Interface) was implemented to get the user started with. The GUI (Graphical User Interface) provides the functionality of starting the live capture and stopping it with a specific number of packets to be captured as input. It supports the functionality of watching details of each packet given the packet number as input. The details of each packet were shown on a separate java frame. Java Swing was used to develop the Graphical User Interface. Data types like Textarea, Text Fields, JFrame, Radio Buttons, Buttons were made functional to get the work done. Finally, GUI (Graphical User Interface)

was made responsive using thread as the main process provided with flexibility of starting, stopping the packet capturing process and also to clear the text area and filter the traffic based on radio buttons, and thread itself was used to update the Text Area with details of captured packets traffic through internet. In this way, the software was made functional.

References:

- “Jpcap_tutorial - SIP Inspector,” Jpcap_tutorial - SIP Inspector. [Online]. Available: http://www.sipinspector.com/download/jpcap/JPcap_tutorial. [Accessed: 01-Jun-2020].
- “What is Java Development Kit (JDK)? - Definition from Techopedia,” Techopedia.com. [Online]. Available: https://www.techopedia.com/definition/5594/java-development-kit-jdk. [Accessed: 01-Jun-2020].
- Filip, “jpcap,” Java Tutorial Network, 20-Oct-2017. [Online]. Available: https://javatutorial.net/tag/jpcap. [Accessed: 01-Jun-2020].
- PacketCapture (jpcap System Class Documentation). [Online]. Available: http://jpcap.sourceforge.net/javadoc/net/sourceforge/jpcap/capture/PacketCapture.html. [Accessed: 01-Jun-2020].
- “NetBeans,” Wikipedia, 28-May-2020. [Online]. Available: https://en.wikipedia.org/wiki/NetBeans. [Accessed: 01-Jun-2020].
- “Swing (Java),” Wikipedia, 11-May-2020. [Online]. Available: https://en.wikipedia.org/wiki/Swing_(Java). [Accessed: 01-Jun-2020].
- “WinPcap,” WinPcap. [Online]. Available: https://www.winpcap.org/. [Accessed: 01-Jun-2020].
- “WinPcap,” WinPcap - The Wireshark Wiki. [Online]. Available: https://wiki.wireshark.org/WinPcap. [Accessed: 01-Jun-2020].
- “Differences between JDK, JRE and JVM,” GeeksforGeeks, 10-Aug-2018. [Online]. Available: https://www.geeksforgeeks.org/differences-jdk-jre-jvm/. [Accessed: 01-Jun-2020].