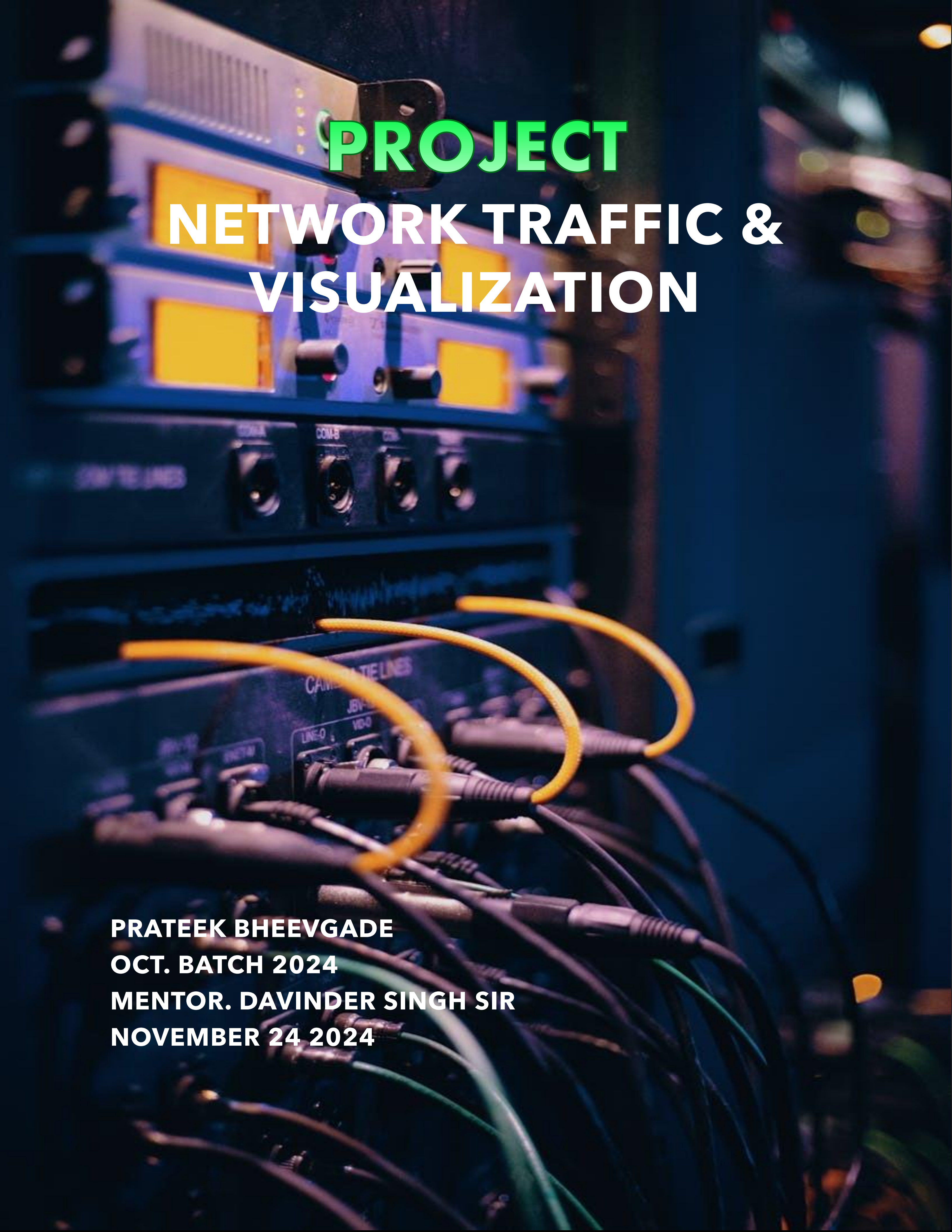


PROJECT
NETWORK TRAFFIC & VISUALIZATION

PRATEEK BHEEVGADE
OCT. BATCH 2024
MENTOR. DAVINDER SINGH SIR
NOVEMBER 24 2024

TABLE OF CONTENTS

Introduction

Network Traffic Analysis – Break Down

Results and Implementation

Conclusion

Project Overview

The Network Traffic Analysis Tool is a comprehensive Python-based application designed to provide real-time network packet capture, analysis, and visualization. The system utilizes modern libraries and frameworks to create an intuitive graphical interface that allows users to monitor and analyze network traffic patterns effectively.

System Requirements

- Python 3.8 or higher
- Operating System: Windows 10/11, Linux, macOS
- Minimum 4GB RAM
- Network interface card
- Administrative privileges for packet capture

NETWORK ANALYSIS TOOL

BREAK - DOWN

1. Core Architecture

1.1 Main Components

- PrateekNetworkAnalyzer: Core class implementing the main application logic and GUI
- PacketFilter: Handles packet filtering based on various criteria
- ProtocolAnalyzer: Manages protocol-specific analysis
- AdvancedVisualizer: Provides advanced visualization capabilities

1.2 Dependencies

- Scapy: Network packet manipulation
- Pandas: Data analysis and manipulation
- Matplotlib: Data visualization
- Tkinter: GUI framework
- Seaborn: Statistical data visualization

2. Detailed Component Analysis

2.1 GUI Implementation (PrateekNetworkAnalyzer)

- Layout: Uses a two-panel design with controls/packet list on the left and visualizations on the right

Key Features:

- Real-time packet capture display
- Color-coded packet list (TCP: light blue, UDP: light green, ICMP: yellow)
- Statistical summary display
- Interactive control buttons
- Multi-threaded packet capture to maintain UI responsiveness

2.2 Packet Processing

- Captures packets using Scapy's sniffing capabilities
- Extracts key information:
    - Timestamp
    - Source/Destination IP
    - Protocol
    - Packet length
- Implements real-time updates to both GUI and internal data structures

2.3 Data Visualization

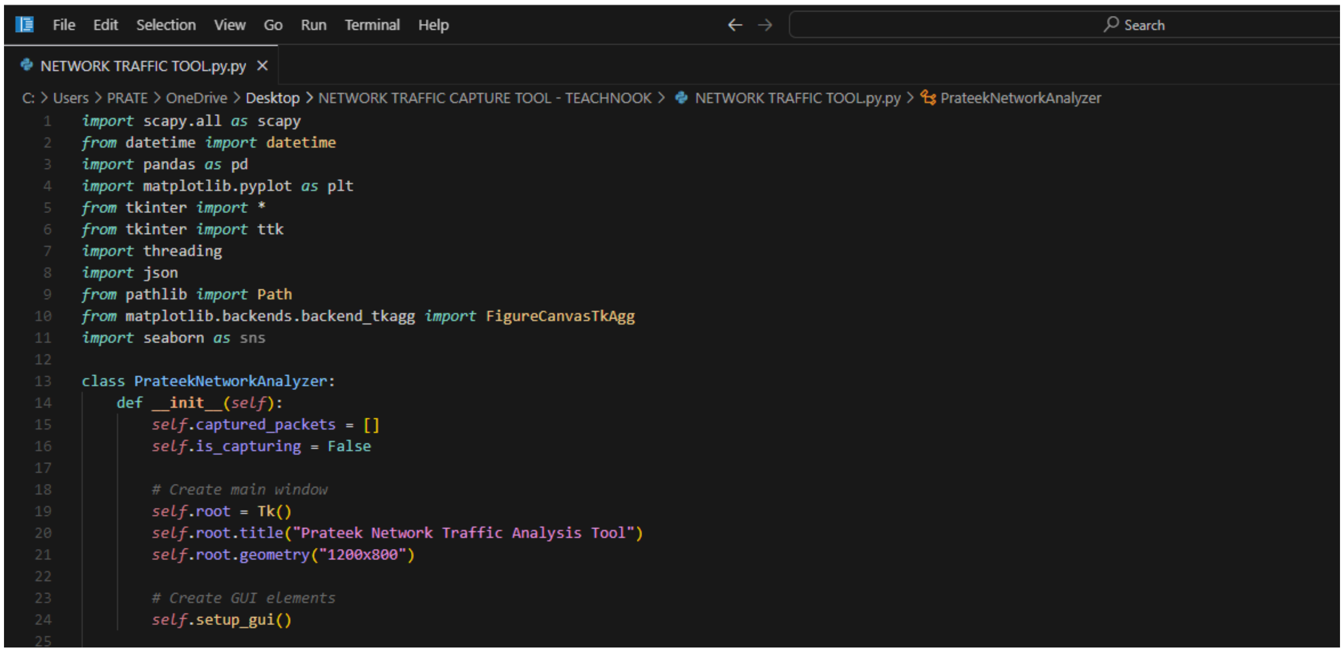
- Three main visualization types:
    1. Protocol Distribution (*Pie Chart*)
    2. Packet Length Distribution (*Histogram*)
    3. Traffic Flow Over Time (*Line Plot*)
- Uses FigureCanvasTkAgg for embedding Matplotlib plots in Tkinter
- Implements automatic updates and layout adjustments

3. Data Export Capabilities

- JSON export of captured packets
- Excel report generation with summary statistics
- Visualization export as PNG files
- Automated file naming with timestamps

Result & Implementation

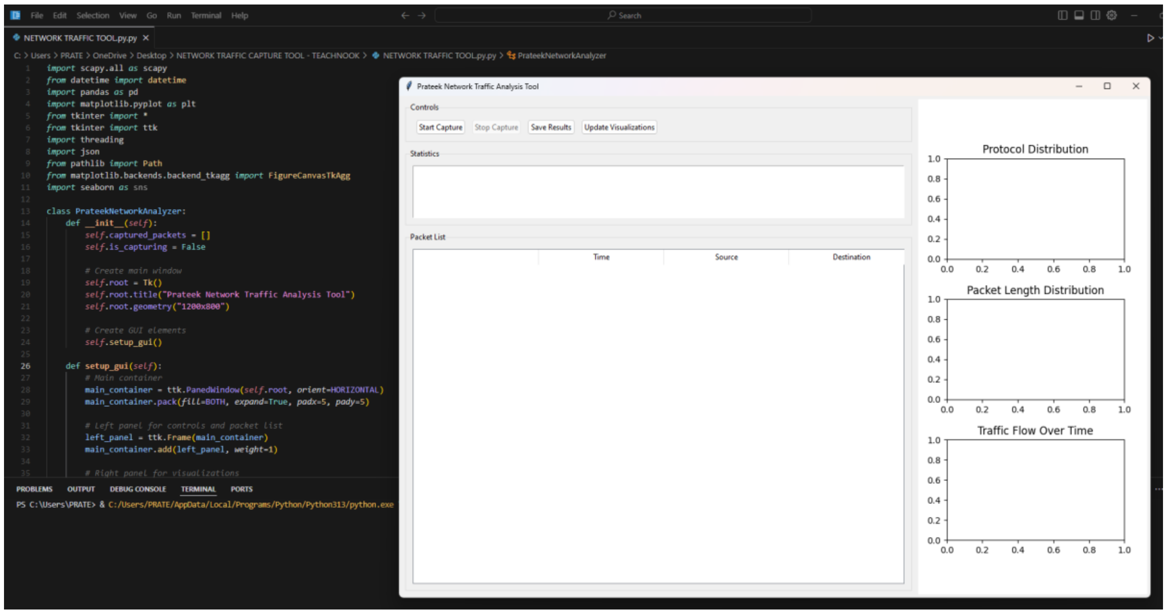
Python Code – (Available @ GitHub)


File Edit Selection View Go Run Terminal Help
Search
NETWORK TRAFFIC TOOL.py.py
C: > Users > PRATE > OneDrive > Desktop > NETWORK TRAFFIC CAPTURE TOOL - TEACHNOOK > NETWORK TRAFFIC TOOL.py.py > PrateekNetworkAnalyzer

```python
import scapy.all as scapy
from datetime import datetime
import pandas as pd
import matplotlib.pyplot as plt
from tkinter import *
from tkinter import ttk
import threading
import json
from pathlib import Path
from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg
import seaborn as sns

class PrateekNetworkAnalyzer:
    def __init__(self):
        self.captured_packets = []
        self.is_capturing = False

        # Create main window
        self.root = Tk()
        self.root.title("Prateek Network Traffic Analysis Tool")
        self.root.geometry("1200x800")

        # Create GUI elements
        self.setup_gui()
```

*Full Code available @ GitHub

Implementation & Result:



```python
import scapy.all as scapy
from datetime import datetime
import pandas as pd
import matplotlib.pyplot as plt
from tkinter import *
from tkinter import ttk
import threading
import json
from pathlib import Path
from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg
import seaborn as sns

class PrateekNetworkAnalyzer:
    def __init__(self):
        self.captured_packets = []
        self.is_capturing = False

        # Create main window
        self.root = Tk()
        self.root.title("Prateek Network Traffic Analysis Tool")
        self.root.geometry("1200x800")

        # Create GUI elements
        self.setup_gui()

    def setup_gui(self):
        # Main container
        main_container = ttk.PanedWindow(self.root, orient=HORIZONTAL)
        main_container.pack(fill=BOTH, expand=True, padx=5, pady=5)

        # Left panel for controls and packet list
        left_panel = ttk.Frame(main_container)
        main_container.add(left_panel, weight=1)

        # Right panel for visualizations
```

```
PS C:\Users\PRATE> & C:/Users/PRATE/AppData/Local/Programs/Python/Python313/python.exe
```

File Edit Selection View Go Run Terminal Help
NETWORK TRAFFIC TOOL.py.py
C: > Users > PRATE > OneDrive > Desktop > NETWORK TRAFFIC CAPTURE TOOL - TEACHNOOK > NETWORK TRAFFIC TOOL.py.py > PrateekNetworkAnalyzer
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Prateek Network Traffic Analysis Tool
Controls
Start Capture
Stop Capture
Save Results
Update Visualizations
Statistics
Packet List
Time
Source
Destination
Protocol Distribution
Packet Length Distribution
Traffic Flow Over Time

Fig. 1. Running Program & Overview of Network Traffic Tool

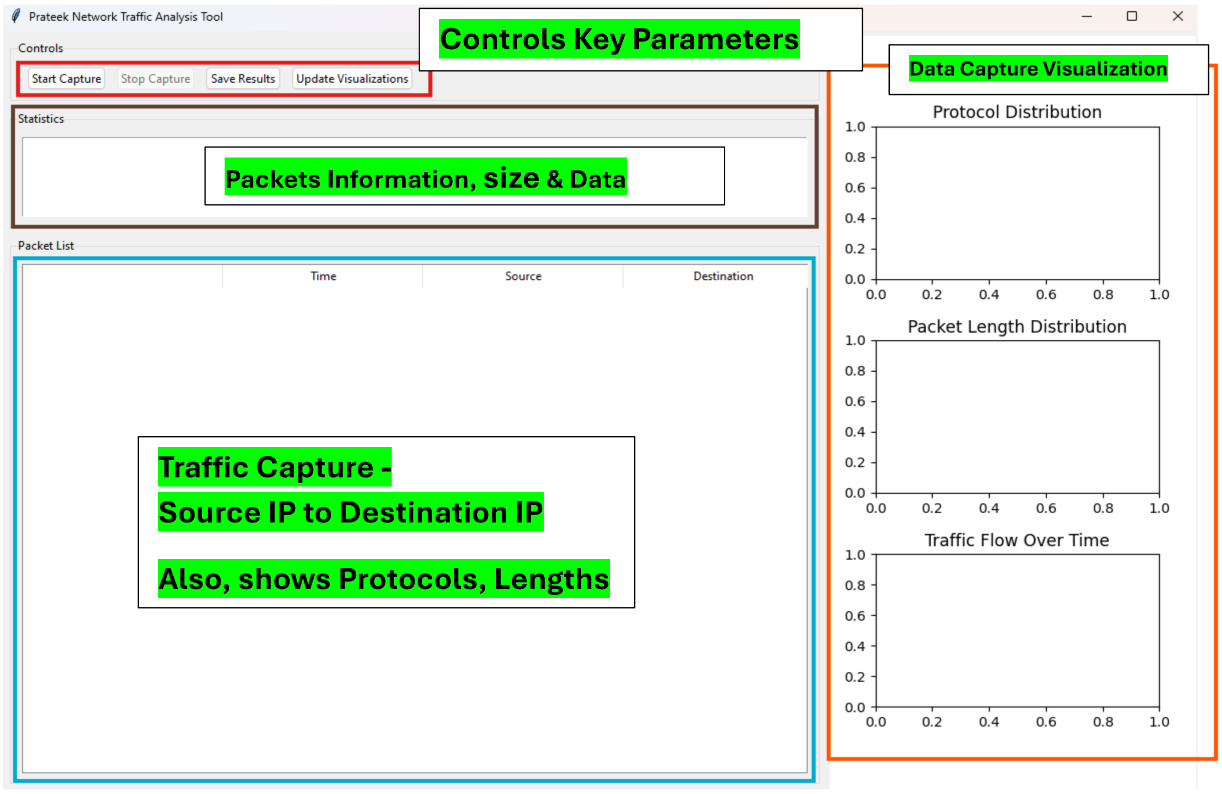
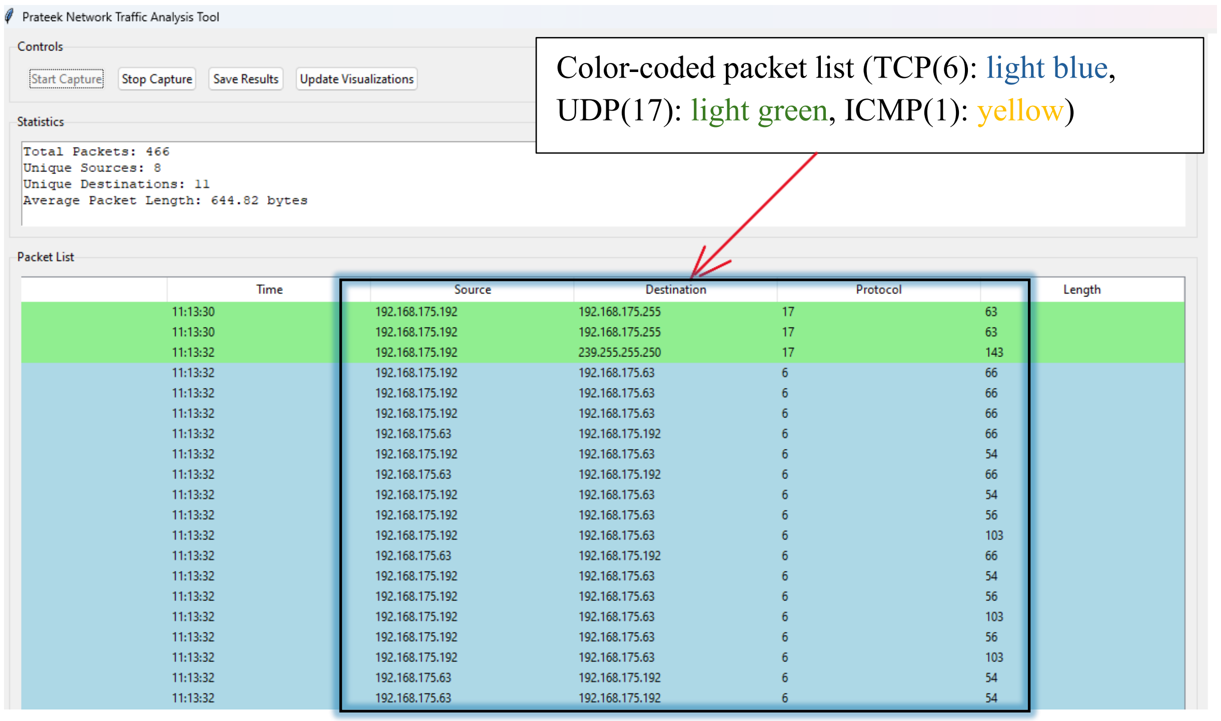


Fig. 2. Explaining every Controls and parameters



Color-coded packet list (TCP(6): light blue, UDP(17): light green, ICMP(1): yellow)

```
Total Packets: 466
Unique Sources: 8
Unique Destinations: 11
Average Packet Length: 644.82 bytes
```

Time	Source	Destination	Protocol	Length
11:13:30	192.168.175.192	192.168.175.255	17	63
11:13:30	192.168.175.192	192.168.175.255	17	63
11:13:32	192.168.175.192	239.255.255.250	17	143
11:13:32	192.168.175.192	192.168.175.63	6	66
11:13:32	192.168.175.192	192.168.175.63	6	66
11:13:32	192.168.175.192	192.168.175.63	6	66
11:13:32	192.168.175.63	192.168.175.192	6	66
11:13:32	192.168.175.192	192.168.175.63	6	54
11:13:32	192.168.175.63	192.168.175.192	6	66
11:13:32	192.168.175.192	192.168.175.63	6	54
11:13:32	192.168.175.192	192.168.175.63	6	56
11:13:32	192.168.175.192	192.168.175.63	6	103
11:13:32	192.168.175.63	192.168.175.192	6	66
11:13:32	192.168.175.192	192.168.175.63	6	54
11:13:32	192.168.175.192	192.168.175.63	6	56
11:13:32	192.168.175.192	192.168.175.63	6	103
11:13:32	192.168.175.192	192.168.175.63	6	56
11:13:32	192.168.175.192	192.168.175.63	6	103
11:13:32	192.168.175.63	192.168.175.192	6	54
11:13:32	192.168.175.63	192.168.175.192	6	54

Fig. 3. Network Traffic Capturing

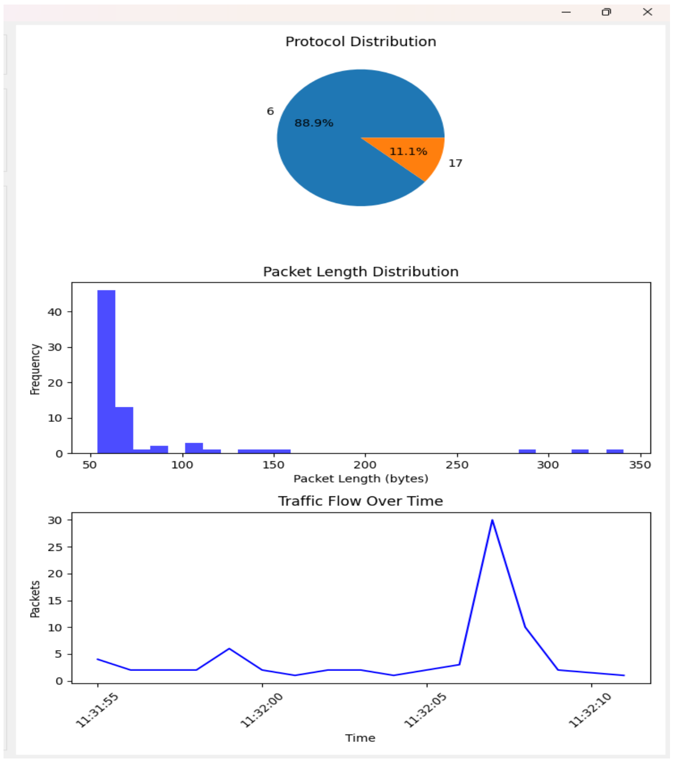
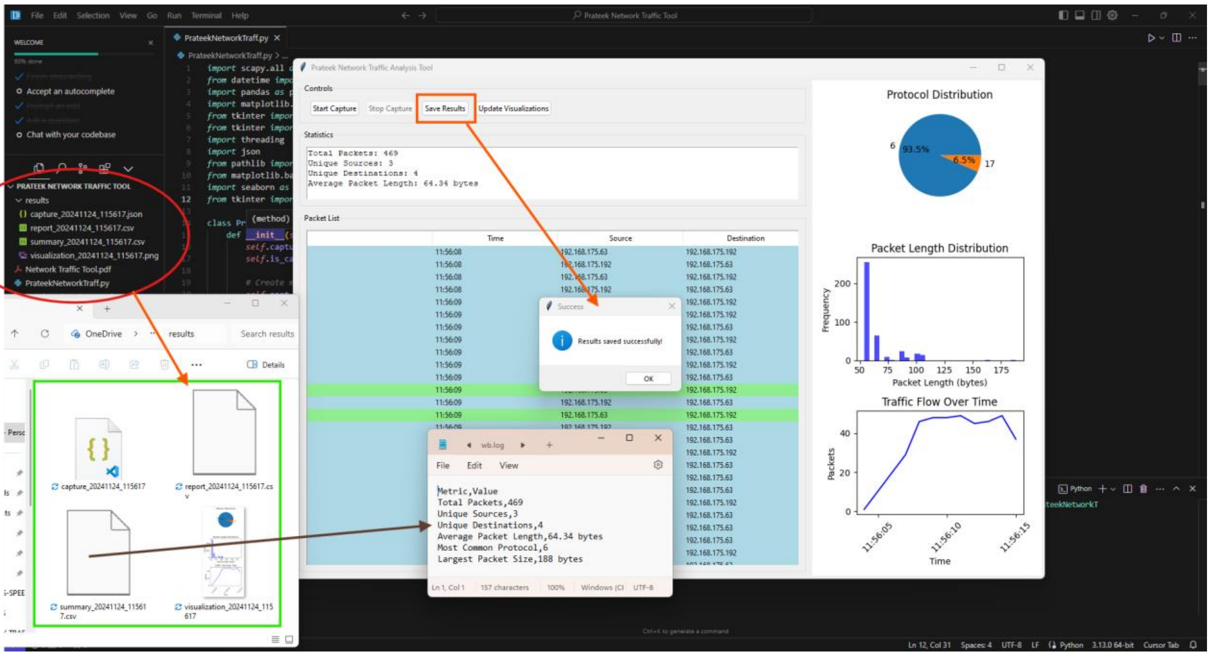


Fig .4. Data Visualization according to Traffic Capture

Save Results & Data Visualization Information



```
Total Packets: 469
Unique Sources: 3
Unique Destinations: 4
Average Packet Length: 64.34 bytes
```

Results saved successfully!

```
Metric,Value
Total Packets,469
Unique Sources,3
Unique Destinations,4
Average Packet Length,64.34 bytes
Most Common Protocol,6
Largest Packet Size,188 bytes
```

Fig.5. Data Visualization information

Conclusion

The analyzed network traffic analyzer demonstrates a robust foundation for network monitoring and analysis. Its modular design allows for easy extension and modification, while the integration of advanced features like machine learning and security analysis makes it a valuable tool for network administrators. The identified areas for improvement provide clear paths for future development while maintaining the existing strengths of the application.