
PACKET ANALYSIS DATA FOR CYBER ATTACKS IN NETWORK SYSTEMS

Jeyavim Sherin RC^{*1}, Parkavi K^{*2}

^{*1,2}School Of Computer Science And Engineering, Vellore Institute Of Technology, Chennai,
Tamilnadu, India.

DOI : <https://www.doi.org/10.56726/IRJMETS59899>

ABSTRACT

A comprehensive collection of real-time network data is necessary to evaluate any defense mechanism based on out-of-date software versions, unprotected ports, etc. The network forensics field uses packet analysis to trace back network traffic, which assuming an accurate packet detail. In addition to detecting online fraud, data breaches, and unauthorized activities, it can also help prevent them. Attempting intrusions, infecting websites with malware, and reconstructing image files, documents, and email attachments. The performance of a network is often of concern to network administrators. Network performance is affected by a variety of factors. This research aims to investigate a variety of factors related to home-based networks' integrity using Wireshark. During a network connection, different devices are captured. In addition to its basic statistical tools, Wireshark has advanced performance tools for analyzing captured traffic data in the network.

Keywords: Wire Shark, Packet Analysis, Packet Sniffing, TCP, Network Traffic, Network Monitoring.

I. INTRODUCTION

In recent years, the internet has become increasingly popular everywhere. Everyone has access to the internet today. Over-communication networks and online services typically transfer data as packets. There are two types of bits in these groups: data and control [1]. Packet-switched networks utilize protocol data units to capture and log network traffic flow. Intruders or hackers use packet sniffing to steal passwords or extract other valuable information from networks. There are two types of sniffer packets: active and passive [2]. Active sniffers send data into networks and can be detected by other systems, while passive sniffers collect data only. Wireshark is an example of a passive network sniffer. As well as packet sniffers, packet captures (pcap) are used to capture packets from all layers [3]. Packet analyzers analyze application layer protocols. Data is collected, converted, analyzed, and stolen by a packet sniffer.

If network packets are captured, stored, and processed efficiently, they can be used as evidence in forensic investigations. In this paper, packet analysis refers to all types of content, including frames, packets, datagrams, and sessions. Packet sniffers are useful for monitoring and troubleshooting legitimate network traffic [4]. It is possible to analyze network data and separate traffic based on the types of traffic with software designed specifically for this purpose. Passwords, usernames, and sensitive data can also be captured by sniffers besides incoming and outgoing traffic.

- Acquiring and Retaining Network Data Packets
- Analyses of traffic
- Utilize Wireshark for packet analysis

II. BACKGROUND

Protocols serve as frameworks for recognizing and initiating links, alongside defining the guidelines for transmitting data among networked devices. It is possible to separate traffic types using a specially designed software package that analyzes network data [5]. Analyzing network data and separating types of traffic can be done using software designed for this purpose. Using purpose-built software, network data can be analyzed and traffic can be separated according to type. Packet capture software intercepts and records network traffic over digital networks or parts of networks. By displaying various fields, raw data from captured packets can be analyzed and interpreted. To uncover the contents of the captured packets, we need to decode the raw data and display various fields. Packets pass through many intermediate devices as they travel from source to destination. Physical addresses are used to identify NICs in networks. Data packets are received by each device in the network during packet transmission. Every node with a promiscuous network interface receives network information. When promiscuous mode is enabled, a machine can view all traffic on the segment. It is possible to

sniff data if a network interface card sets a machine in promiscuous mode, which enables it to collect all packets and frames on the network, even if they weren't intended for it. Sniffers read all data put into a machine via the NIC [6].

A traffic analysis is the process of intercepting and examining packets to obtain information about communication parties [7]. Communication can still be performed even if it is encrypted and cannot be decrypted. The use of traffic analysis in computer security is a concern, as well as in military intelligence and counterintelligence [8]. There is a way for us to obtain helpful information, passwords, file names, etc., from the communication parties and time of conversations. Communication patterns between entities in a system are examined using traffic analysis as part of inference attacks.

It was invented by a scientist named Gerald Combs in 1997 to be used in trucking, to detect network problems, to monitor data traffic, and to identify network issues. The name of Ethereal was changed to Wireshark in May 2006. PCAP files are used for packet capture [9]. A hexadecimal and byte representation is provided for the different packet types and protocols. TCP streams can also be created from packet data by users.

Bindu Dodiya. et al. [10] Various attack signatures can be traced and categorized using Wireshark, a free open-source packet analysis tool. In addition to identifying malicious behavior online, Wireshark could also detect data breaches and identify malware indicators of compromise thanks to its capability of microscopic data capture. The advantage of Wireshark lies in its comprehensive analysis, to improve cybersecurity, it is necessary to understand the packets in the network and take proactive measures to improve their security. Despite of Wireshark's ability to uncover a wide range of security threats, it did not come with intrusion detection capabilities. The lack of real-time warnings or active prevention of unauthorized activity with Wireshark underscores the importance of proactive threat detection and response by using complementary security measures.

Muhammad Alfawareh [2] recommended optimizing traffic analysis performance, detecting network forensics and spam, performing penetration testing on networks, forming policies, and delivering data in integrated systems while also discussing countermeasures. Wireshark is used for network traffic analysis, its role in network forensics, and the risks associated with obtaining useful information for attacks or stealing data, as well as solutions for these issues.

Sujith Bebertta. et al. [11] developed a method to manage network traffic for various IoT devices. For a stable network and enhanced security, packet-level and flow-level analysis were used to identify and manage IoT devices for an efficient inter-arrival rate determination. Managing the expanding IoT landscape, this approach provided a precise understanding of network flows and insights into strengths, weaknesses, and future scopes. Further research is needed to explore proactive measures for identifying and isolating vulnerabilities, despite the lack of specific details.

Giovanni Barbieri et al. [12] conducted a comparison of Shodan-only assessments versus large-scale traffic analysis at an Internet Exchange Point (IXP). Shodan's limitations were overcome by this approach using sFlow sampling to identify ICS endpoints conducting legitimate industrial traffic. A more comprehensive view of the use of insecure industrial protocols was gained by identifying scanning behaviors as well as distinguishing between industrial and IT network traffic. Although the method offered benefits, a limitation of the study was its dependence on a 31-day sampled traffic capture, which could have overlooked transient industrial traffic patterns. In addition, while the proposed analytic framework is capable of determining legitimate industrial traffic, it may not address real-time or evolving cyber threats adequately, potentially reducing its ability to detect threats in real-time.

Wireshark was used as a network protocol analyzer to examine network security incidents by Ali Siddiqui et al. [13]. By analyzing Wireshark data, ethical hackers could uncover network intrusions and cybersecurity vulnerabilities at the user level, revealing evidence of network intrusions. A network sniffing tool such as Wireshark is capable of capturing and analyzing network transmissions in real-time. In this manner, HTTP, TCP, and UDP protocols were thoroughly examined, allowing a better understanding of network activities and vulnerability identification for websites. In this manner, HTTP, TCP, and UDP protocols were thoroughly examined, allowing a better understanding of network activities and vulnerability identification for websites.

According to Siswanto et al. [14] they examined the number of users who accessed the internet and used bandwidth in a vocational school. Based on its valuable features for network analysis, the Wireshark tool was found to be highly beneficial.

An analysis of high-speed fiber internet connections, specifically TM UniFi, in Malaysia was conducted by Ashaari, Kassim et al. [15] they examined the impact of multiple device connections or services provided by users. Throughput, Round-Trip Time (RTT), latency, and Quality of Service (QoS) were assessed using the Wireshark network packet analyzer.

III. DATA DESCRIPTION

A shared access point connected five devices, including two laptops and three mobile phones, to the system. The access point was designated with a static IP address of 172.16.21.43. Wireshark was installed on one of the laptops, referred to as the "sniffer laptop," which had the IP address 172.16.22.61, and network traffic was captured accordingly. Figure 1 illustrates how the laptop is connected to the router via Ethernet. Wireshark was running on the sniffer laptop's network interface card (NIC), which captured live packets within the network as it was configured in promiscuous mode. There were 28,678 packets captured and analyzed. Among the most notable strengths of Wireshark are its many statistical tools. A comprehensive analysis of network performance was conducted using both fundamental and advanced statistical tools.

Table 1. Specifications

Subject	Network Security
Specific subject area	Analyzing various attacks and defenses with a simulation dataset
Type of data	Table, Image, Chart, Graph, Figure Analyzed, Filtered
Data collection	A shared access point connected five devices, including two laptops and three mobile phones, to the system. The access point was designated with a static IP address of 172.16.21.43. Wireshark was installed on one of the laptops, referred to as the "sniffer laptop," which had the IP address 172.16.22.61, and network traffic was captured accordingly.
Data source location	Institution: Vellore Institute of Technology City/Town/Region: Chennai, Kelambakkam Country: India
Data Format	Packet analysis data CSV file Real-time Capture pcap
Data accessibility	Repository name: Packet analysis data using Wireshark Data identification number: 10.17632/5k6759w5jw.1 Direct URL to data: https://data.mendeley.com/datasets/5k6759w5jw/1 Instructions for accessing these data: Download the dataset from the above link. Which is located in mendeley data.

A brief description of the dataset presented in this paper is provided in Table 1. It outlines the dataset's subject matter, type, method of acquisition, format, description of data collection, source location, and accessibility, as well as the methodology used to collect the data.

An overview of the design process

Figure 1 illustrates how five devices are interconnected to a shared access point.

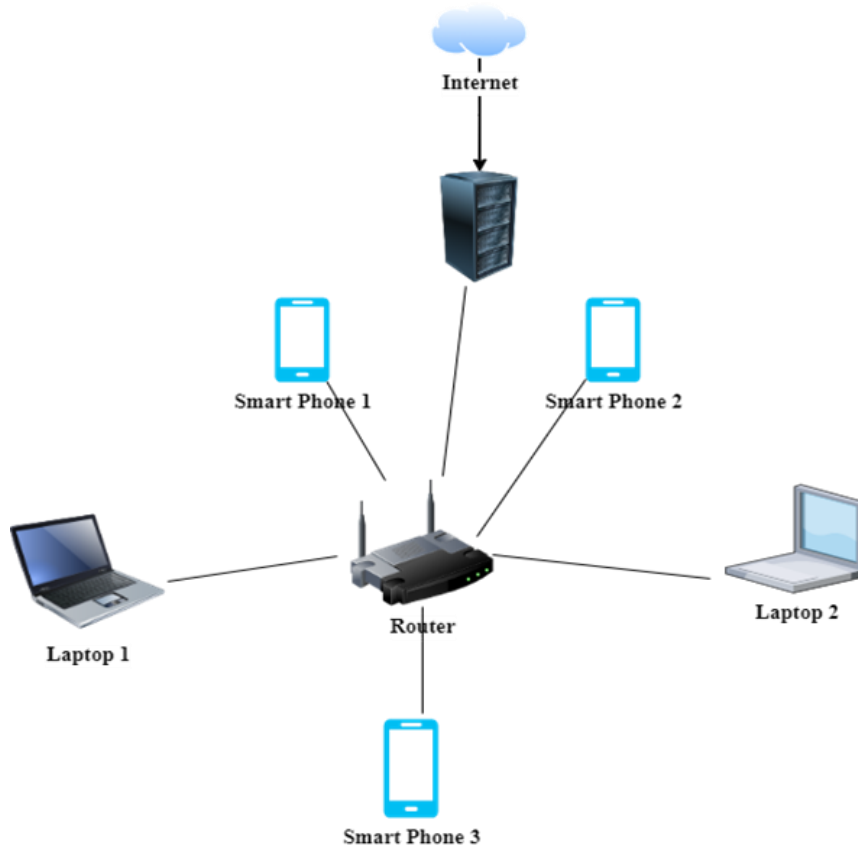


Figure 1: Network Topology.

Examination of TCP Internal Structure

The TCP/IP header encompasses distinct fields, which include:

- Source Port (16 bits) - This is the port where the traffic originates.
- A destination port (16 bits) indicates the intended destination.
- The initial sequence number is represented by the SYN flag when it is set to 1. This sequence number is further incremented for the first data byte and the acknowledged number in the corresponding ACK. Clearing the SYN flag (0) indicates the cumulative sequence number of the segment's initial data byte.
- (32 bits) Acknowledgment number - If the ACK flag is set, this field contains the next sequence number expected by the acknowledger.
- Data Offset (4 bits) - Measures the size of the TCP header in 32-bit words. In this case, 20 bytes is the minimum size and 60 bytes is the maximum size, calculated by multiplying 5 bytes by 15 words.
- The reserved (3 bits) field should remain unset in order to preserve its potential for future applications.
- The Urgent Pointer field is marked as URG.
- ACK: Indicates the importance of the Acknowledgement field. This flag should be set on all client packets after the initial SYN packet.

The following Figure 2 shows the TCP/IP Header

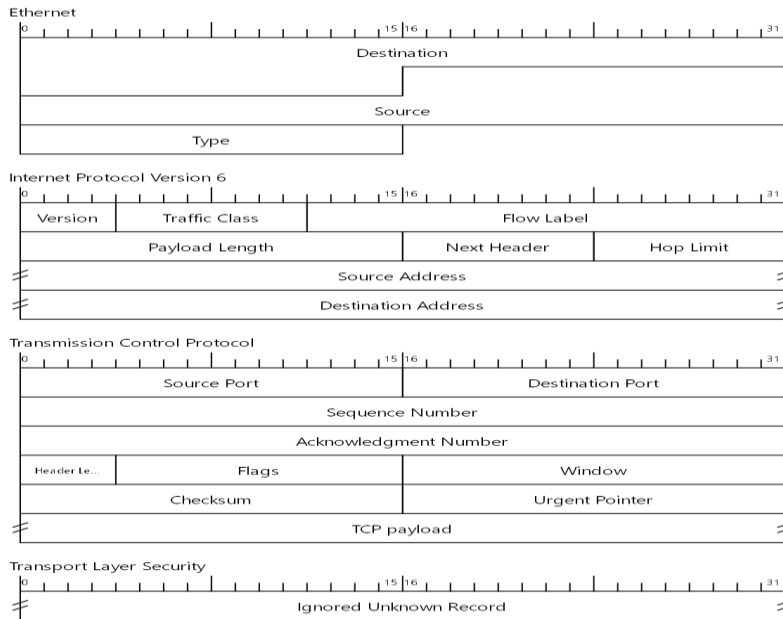


Figure 2: TCP/IP Header.

IV. METHODOLOGY

Experimental Analysis

Initially, we will explore some fundamental statistical tools, followed by an examination of advanced statistical methods. Statistical tools such as I/O graphs, Stevens' window, and TCP trace window are covered. Figure 3 presents an interface overview of the Wireshark Network Analyzer, while Figure 4 illustrates the total number of captured packets.

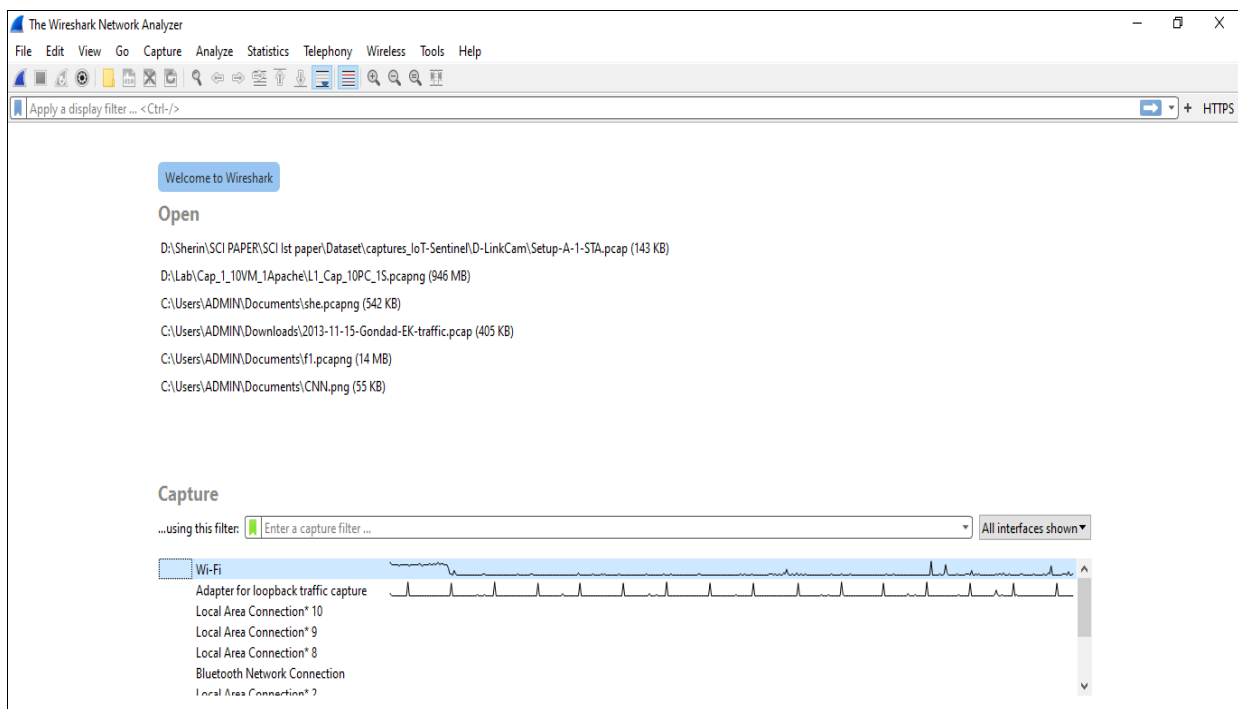


Figure 3: Overview of Wireshark Network Analyzer.

Packet sniffers possess the capability to intercept various forms of data. They can capture passwords, login credentials, as well as details about websites visited by a computer user, including the content viewed while browsing these sites. Businesses can employ packet sniffers to monitor employee network activities and inspect incoming traffic for potential malicious code. In certain instances, a packet sniffer can capture all traffic traversing a network. Subsequently, upon collecting the filtered traffic, one can proceed to identify and address

potential performance issues. For a more focused analysis, you can further filter the captured data based on source and destination ports, enabling targeted testing of specific network elements. The entirety of captured packet information can subsequently be utilized to troubleshoot network performance issues effectively.

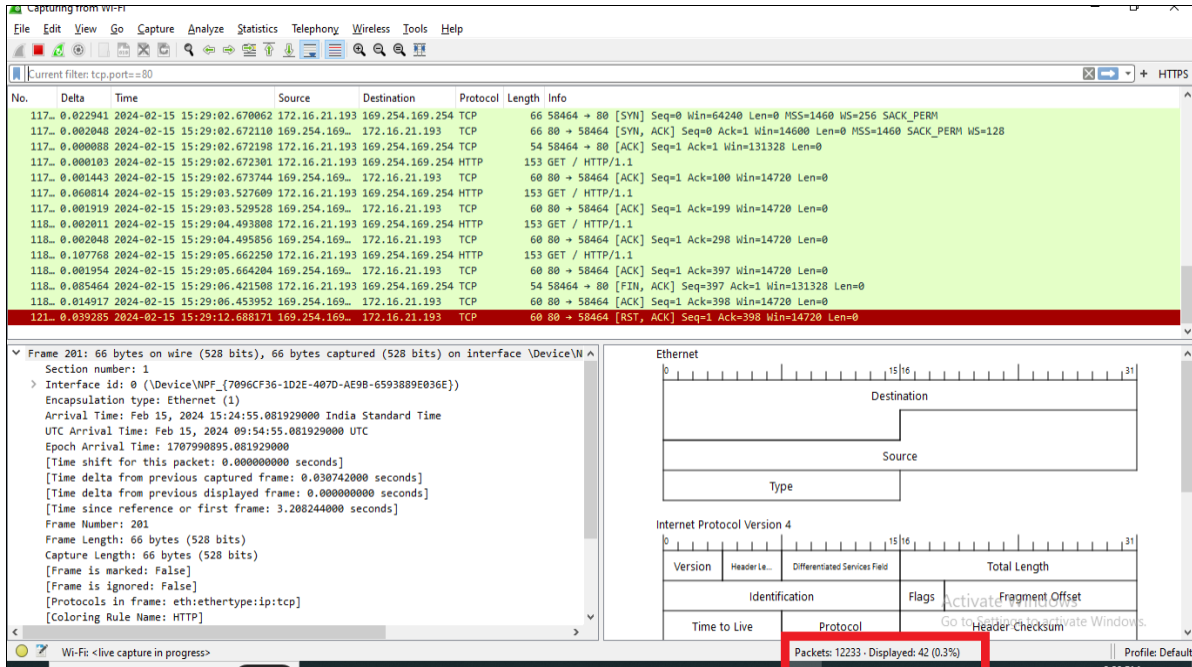


Figure 4: Total number of captured Packets.

The Packet Length refers to the overall size of the entire packet, encompassing the header, trailer, and the transmitted data contained within the packet. The packet length corresponds to the size of the captured frame. The size of the packet states the size of its header. In Wireshark, packet lengths serve as a valuable metric for identifying instances of small packet lengths, particularly relevant in diagnosing issues such as window size reduction, wherein the transmitted data may become smaller than the header.

The screenshot above of the Packet Lengths window exhibits various ranges of packet lengths, along with corresponding packet counts, minimum and maximum lengths, and percentages within each range. According to the provided screenshot, a significant portion of the packets falls within the range of 40 to 79 bytes, with approximately 10,752 packets falling within this length range. Typically, this range of packet lengths signifies the transmission of data, regardless of the specific content being transferred back and forth. Typically, this range of packet lengths signifies the transmission of data, regardless of the specific content being transferred back and forth. Figure 5 depicts the flow chart of the Analysis of captured packets. The packet analysis dataset is generated and accessible for download at <https://data.mendeley.com/datasets/5k6759w5jw/1>, with a corresponding DOI: 10.17632/5k6759w5jw.1.

- Packet Lengths: This section presents various ranges of packet lengths detected within the captured data.
- Count: This field indicates the quantity of packets falling within each respective length range.
- Average: This field exhibits the arithmetic mean of the packet lengths within the specified range.
- Min Val: This field shows the minimum length observed within the specified range.
- Max Val: This field indicates the maximum length observed within the specified range.
- Rate (ms): This field presents the average number of packets per millisecond for the packets within this range.
- Percent: This field indicates the percentage of packets within this range, based on count.
- Burst Rate: Burst occurrences of packets are identified by calculating the total number of packets within a specified time interval and comparing it to intervals within a window of time. By default, packet bursts are identified after 5 millisecond intervals, with comparisons made across 100-millisecond windows.
- Burst Start: This field indicates the start time (in seconds) from the beginning of the packet capture for the interval exhibiting the maximum number of packets.

- Display filter: In the “Filter” field, users can input the desired filter primitive and click on “Apply” to implement the filter and display the filtered packets accordingly.
- Copy: This function duplicates the current statistics, allowing them to be copied to the clipboard for further use or analysis.
- Save As: This feature enables the user to save the data in various formats such as plain text, CSV, YAML, or XML files.

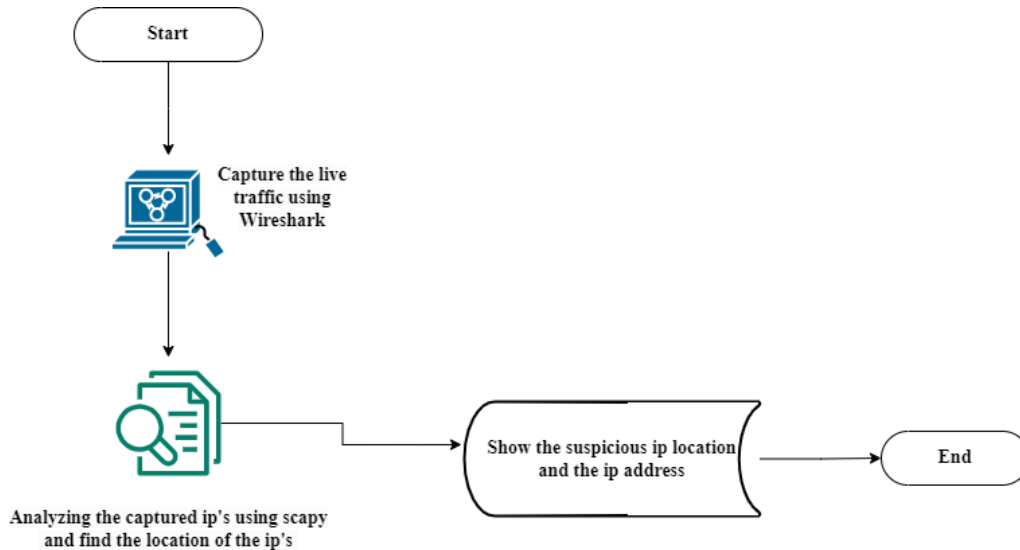
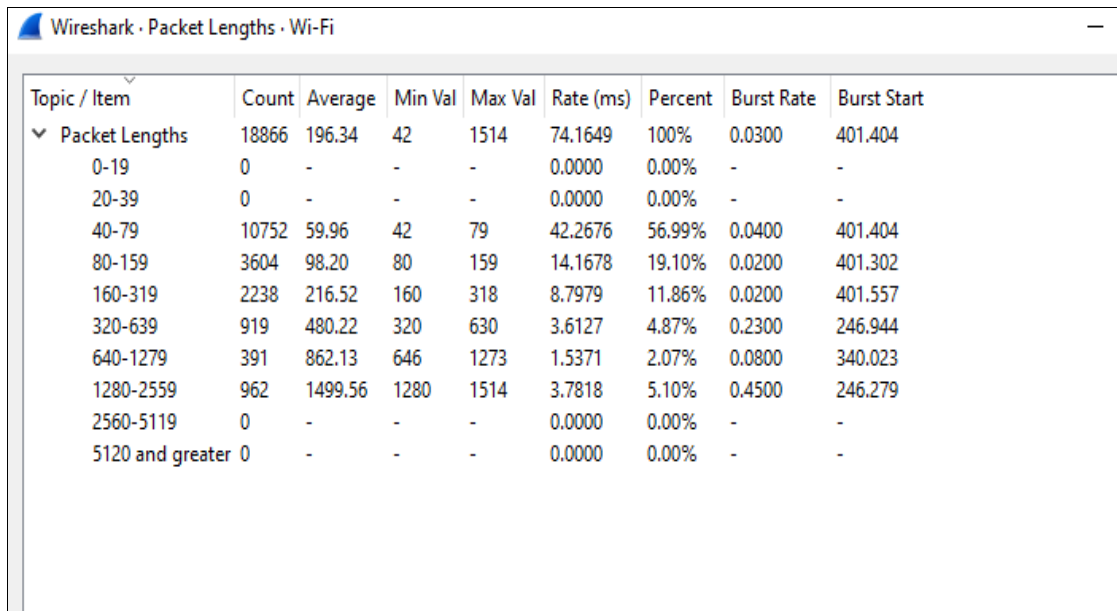


Figure 5: Flow chart of the Analysis of captured Packets.

The below Figure 6 shows the Packet Lengths



Topic / Item	Count	Average	Min Val	Max Val	Rate (ms)	Percent	Burst Rate	Burst Start
Packet Lengths	18866	196.34	42	1514	74.1649	100%	0.0300	401.404
0-19	0	-	-	-	0.0000	0.00%	-	-
20-39	0	-	-	-	0.0000	0.00%	-	-
40-79	10752	59.96	42	79	42.2676	56.99%	0.0400	401.404
80-159	3604	98.20	80	159	14.1678	19.10%	0.0200	401.302
160-319	2238	216.52	160	318	8.7979	11.86%	0.0200	401.557
320-639	919	480.22	320	630	3.6127	4.87%	0.2300	246.944
640-1279	391	862.13	646	1273	1.5371	2.07%	0.0800	340.023
1280-2559	962	1499.56	1280	1514	3.7818	5.10%	0.4500	246.279
2560-5119	0	-	-	-	0.0000	0.00%	-	-
5120 and greater	0	-	-	-	0.0000	0.00%	-	-

Figure 6: Packet Lengths.

V. MODELING AND ANALYSIS

Wireshark's packet filter focuses on filtering by protocol:

For example, Figure 7 indicates the packet filter ip. proto = 6 indicates that this filter will capture only TCP (Transmission Control Protocol) packets. Based on our requirements, the protocol number can be adjusted or different filters can be used.

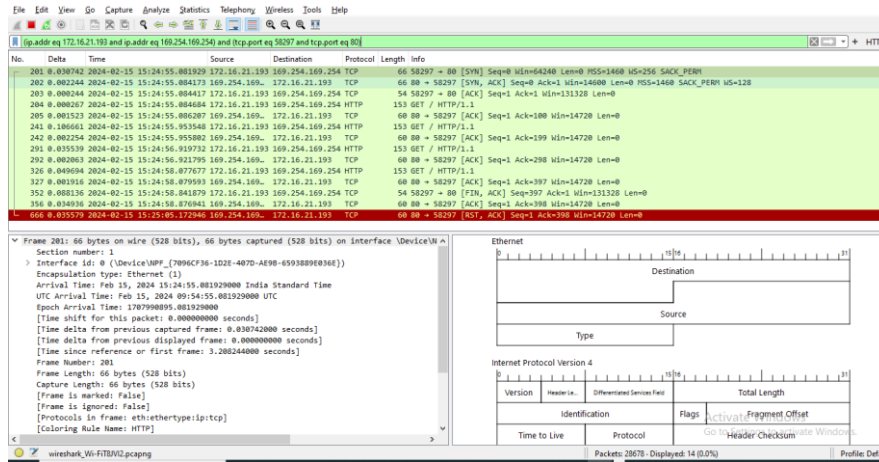


Figure 7: Packet Filter.

I/O Graph:

Displays how many packets are sent or how many bytes are sent per second for all packets matching the filter. The default display will display one graph displaying the number of packets per second.

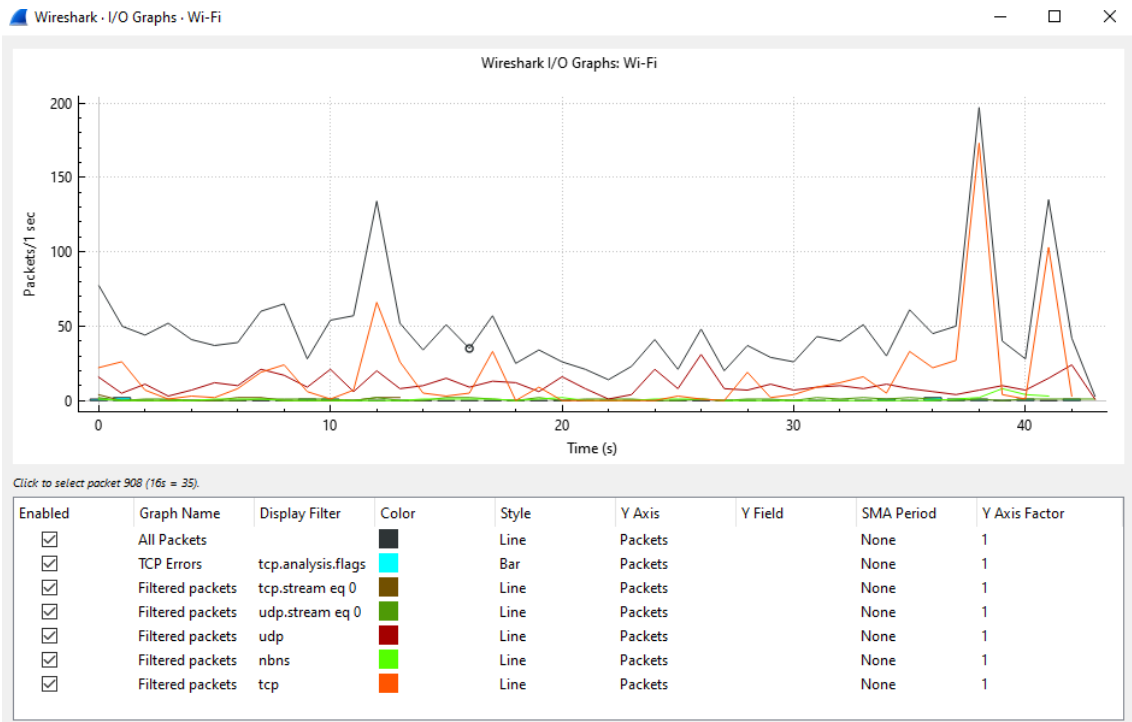


Figure 8a: I/O Graph with 1sec.

A screenshot of the above I/O Graph window shows a highly configurable graph of the captured network packets (Figure 8a). Traffic in a capture file is displayed as packets (bytes/bits) per second in the following Figure 8b, 8c and 8d graphs. In this chart, the time in seconds is shown on the x-axis and the number of packets is shown on the y-axis by default. A change in the scale can be made for the x-axis and the y-axis. We can change the time interval and the scale from linear to logarithmic.

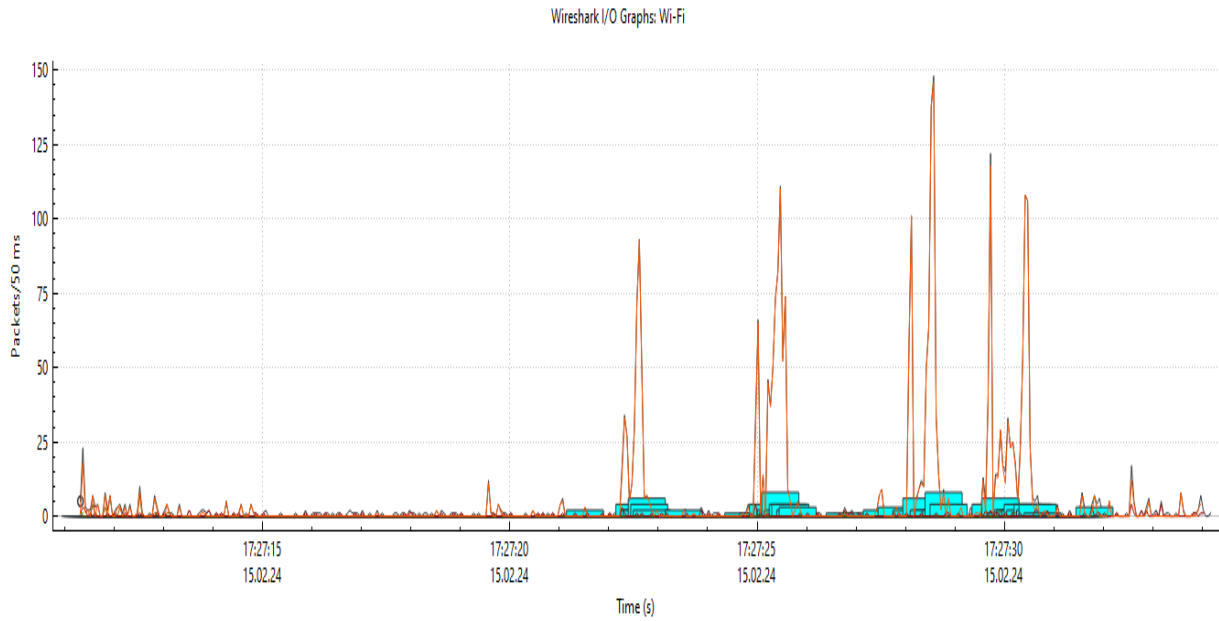


Figure 8b: I/O Graphs with 50ms and TCP Error.

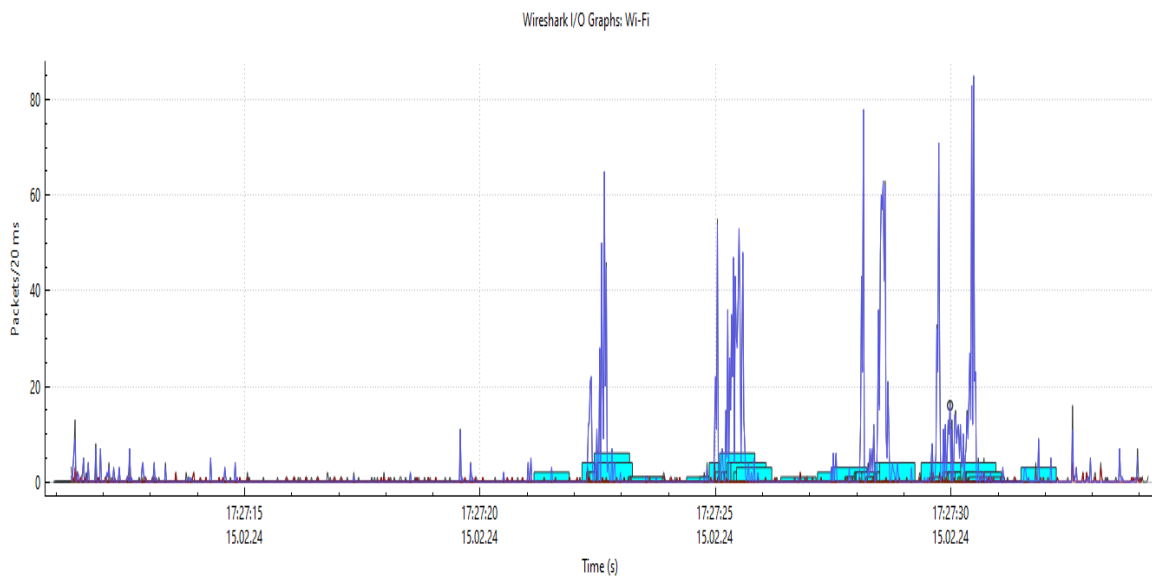


Figure 8c: I/O Graphs with 20ms.

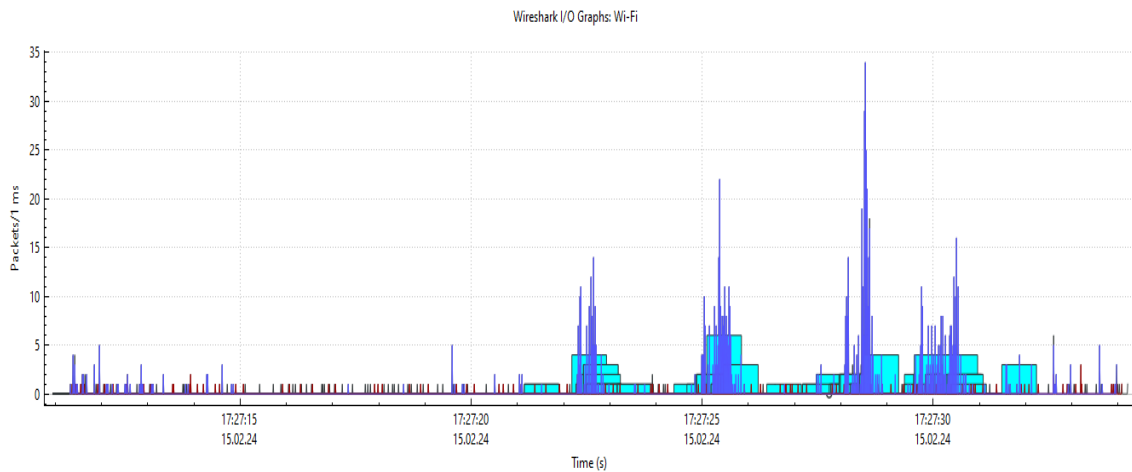


Figure 8d: I/O Graphs with 1ms with date and time.

Round trip time (RTT) Graph

Network admins use the graph below (Figure 9) to identify congestion or latency that may affect the performance of their network.

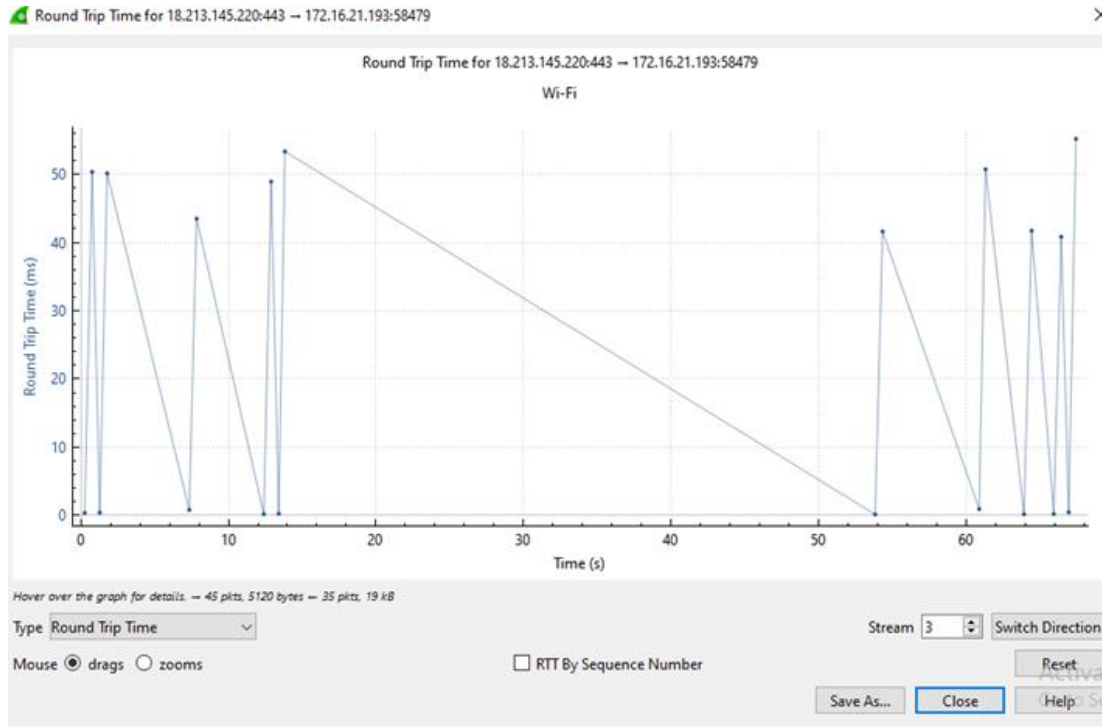


Figure 9: Round trip time.

The above graph shows some instances of latency.

Throughput Graph

Throughput graphs (Figure 10) are used to illustrate traffic flow, similar to I/O graphs.

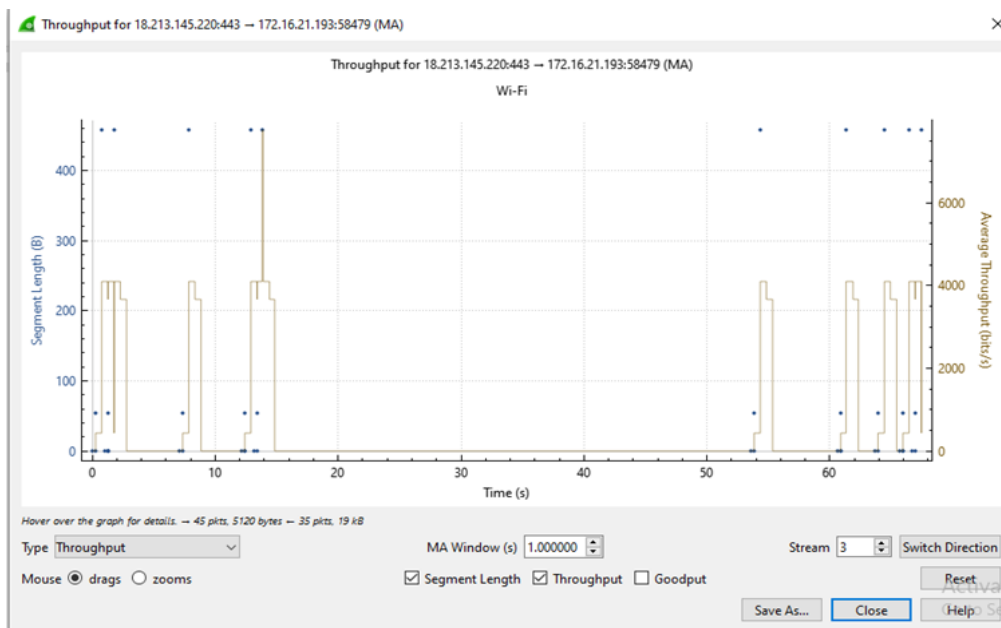


Figure 10: Throughput Graph.

Time-sequence graph (tcptrace)

A time-series graph of a TCP stream is shown in Figure 12. This graph illustrates unidirectional traffic. A time-sequence graph tells us what segments are traveling, when segments have been acknowledged, and what buffer area the client has available.

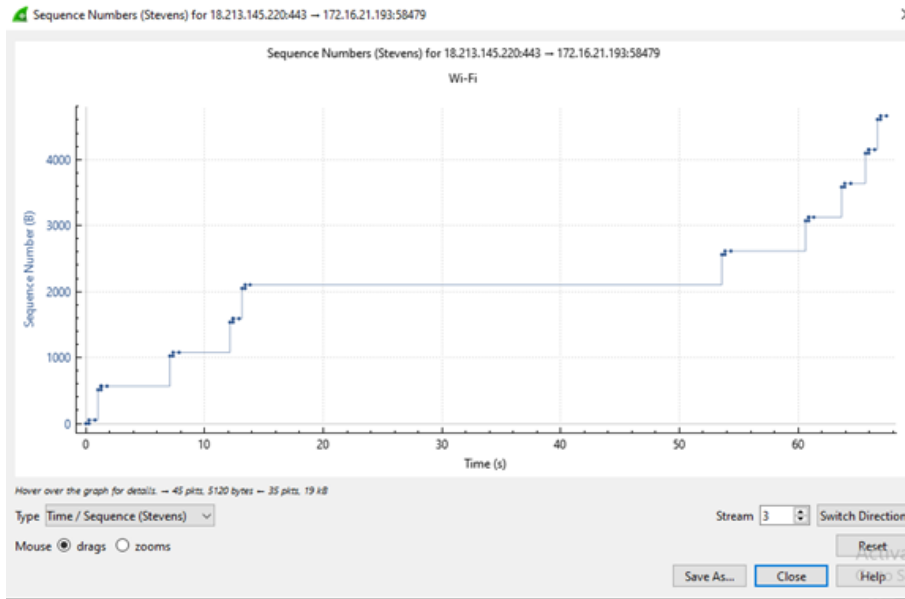


Figure 11: Time/Sequence (Stevens) Graph.

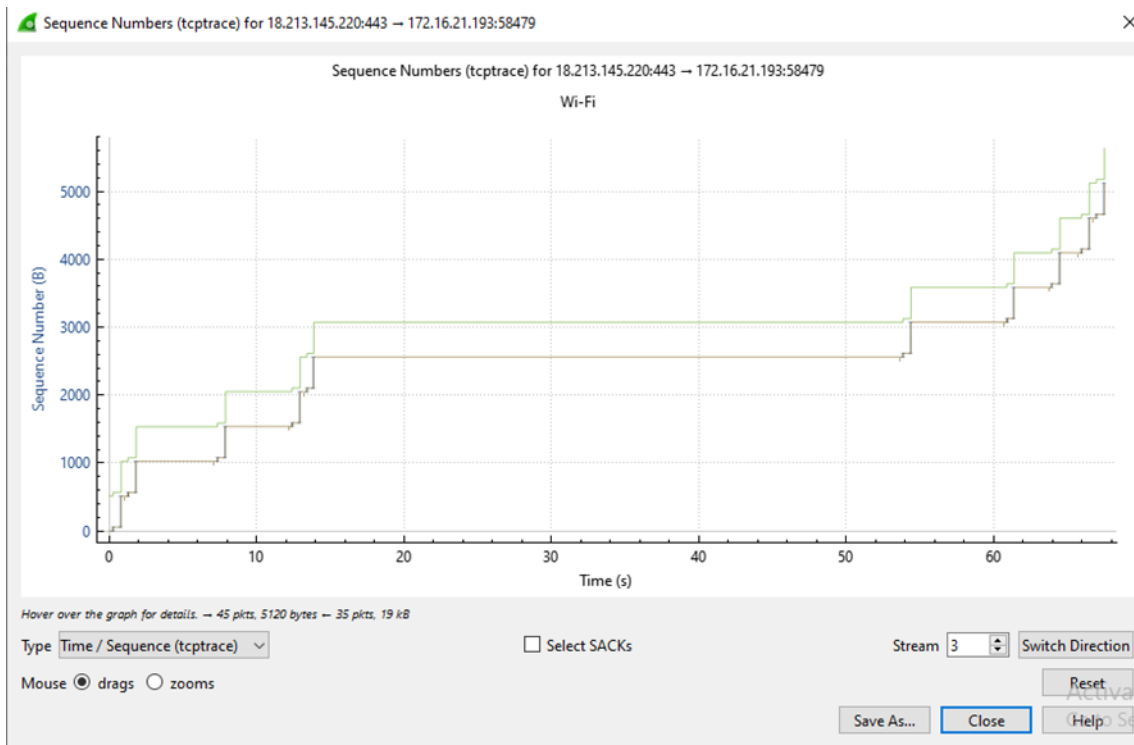


Figure 12: Time/Sequence (tcptrace) graph.

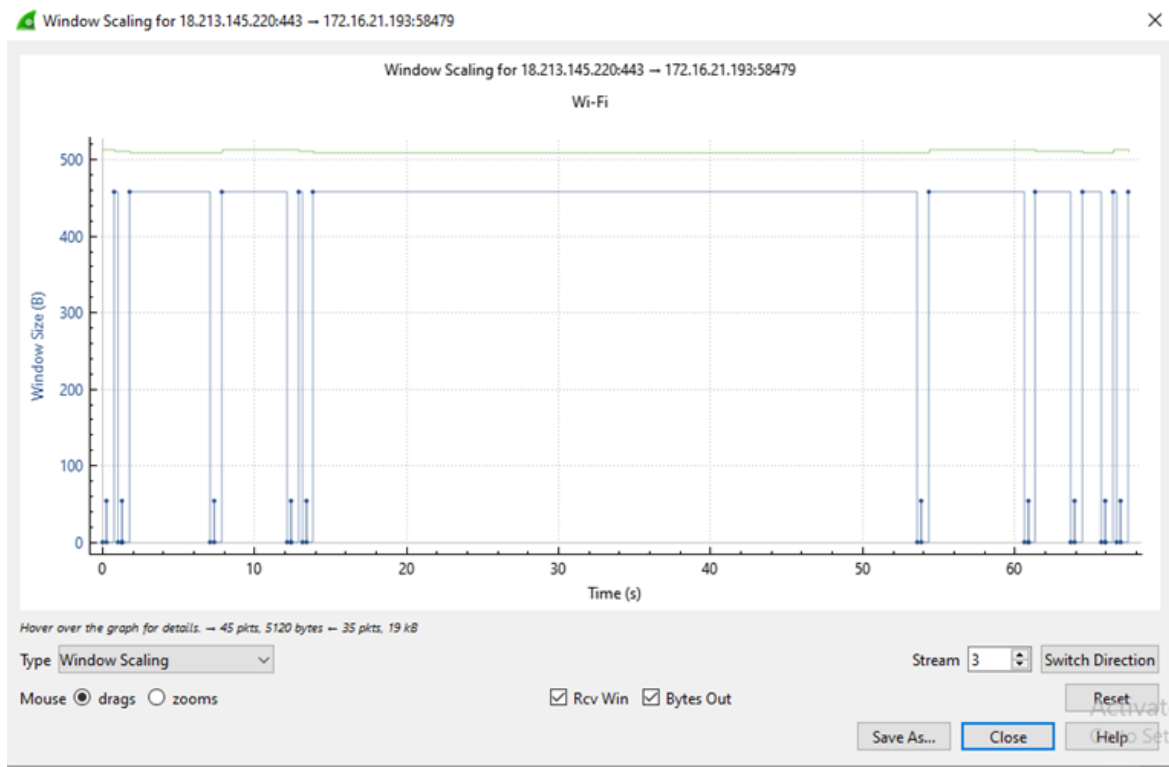


Figure 13: Window scaling Graph.

VI. RESULTS AND DISCUSSION

In the Internet Protocol Family, the Internet Protocol provides network layer transport functionality. Packets are transferred between IP addresses using the IP protocol. Protocol types are shown in Figure 14. A packet and an IP address are provided by the user, and the IP address is used to send the packet to the remote host. Protocols such as TCP and UDP are typically used over IP.

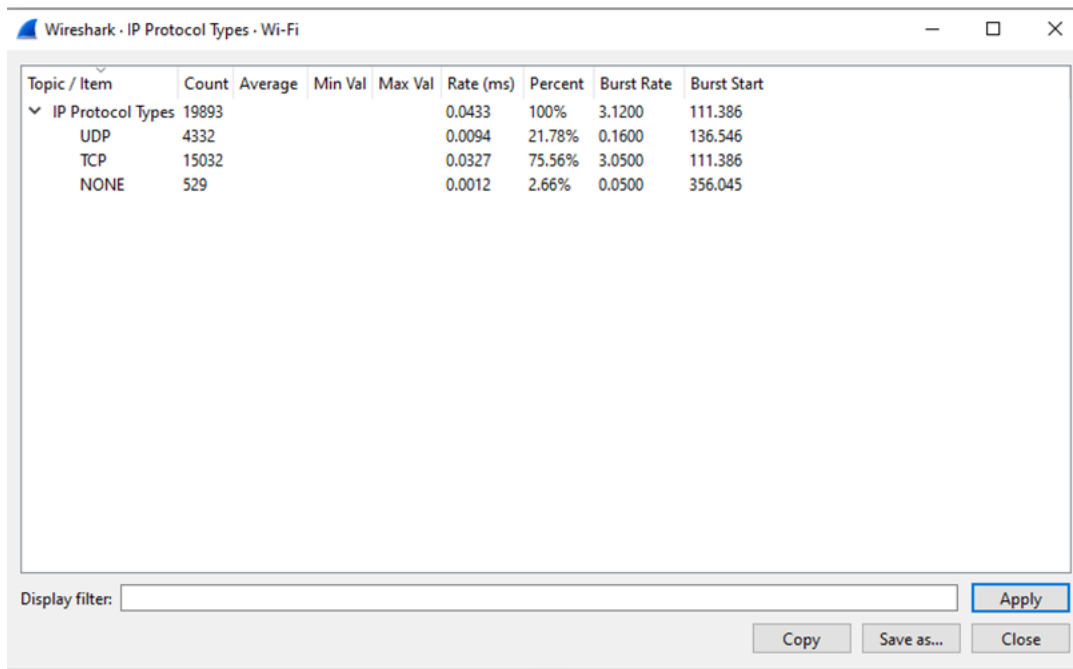


Figure 14: Protocol Types.

A Domain Name System identifies machines that are reachable over a network or via the Internet. An IP address is associated with a domain name, and a name request is resolved to the IP address of a reachable

machine. There are two types of resolution factors in the Domain Name System Resolution graph: packet count and burst rate. Figure 15 illustrates web-based packet capture statistics presented in the proposed dataset.

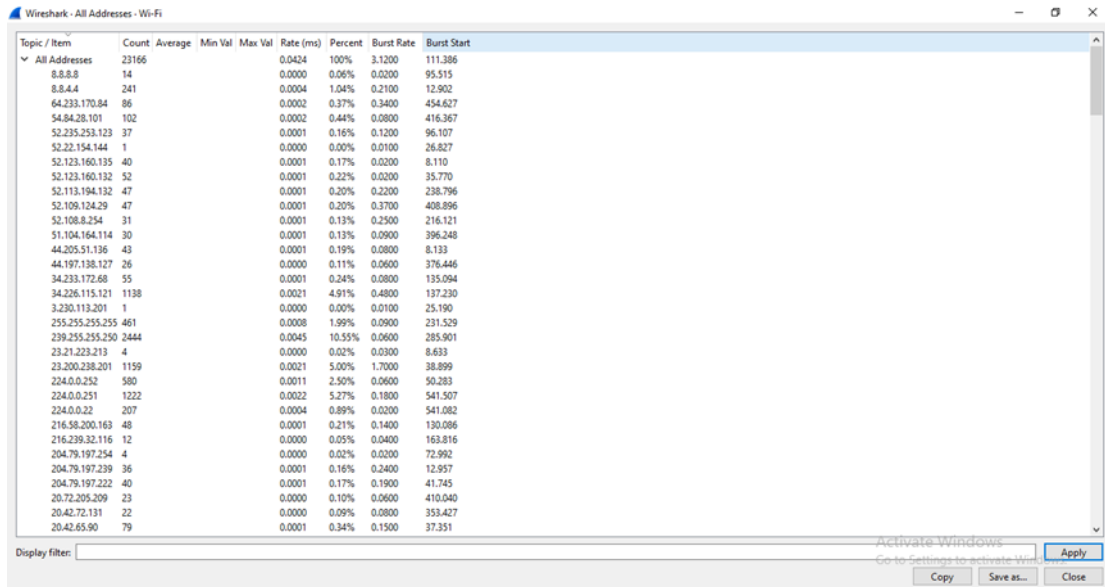


Figure 15: Web-based packet capture statistics.

Flow Graph

The connection between the hosts is shown in Figure 16. It shows the packet timing, direction, ports, and comments for each connection captured. A variety of filters are available, including ICMP (Internet Control Message Protocol) flows, ICMPv6 flows, UIM flows, and TCP flows. Accordingly, there are different controls in the flow graph window. Flow graphs are useful for finding out port numbers and IP addresses, and can help to identify unusual port numbers or IP addresses in traffic.

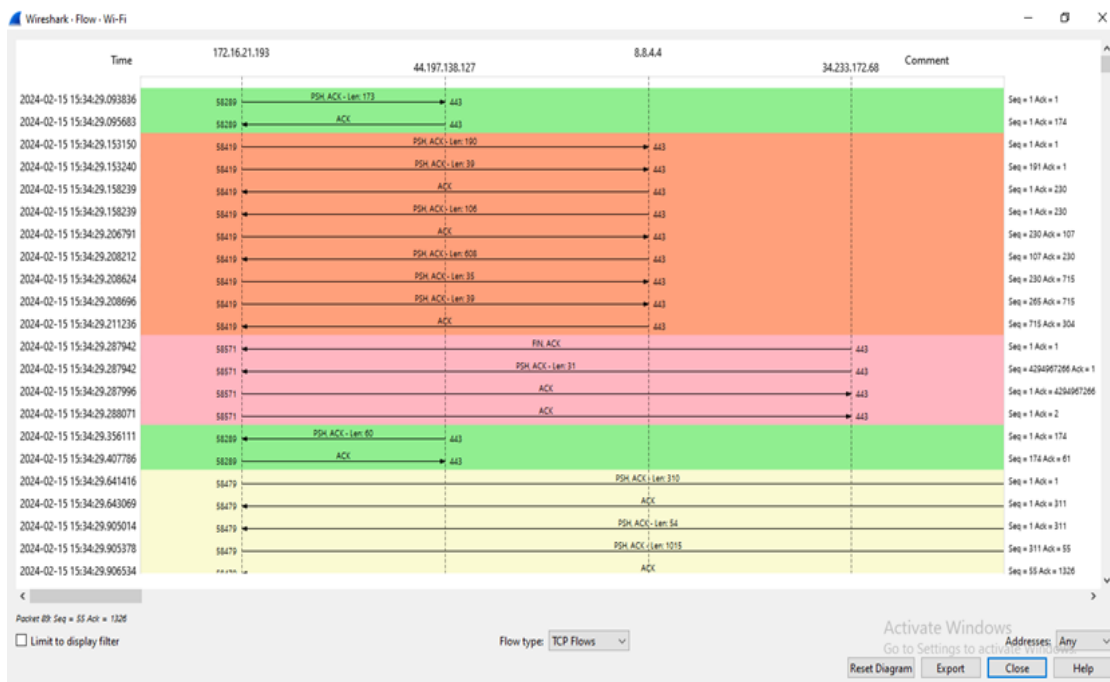


Figure 16: Flow Graph.

An example of a capture filter can be seen in Figure 17.

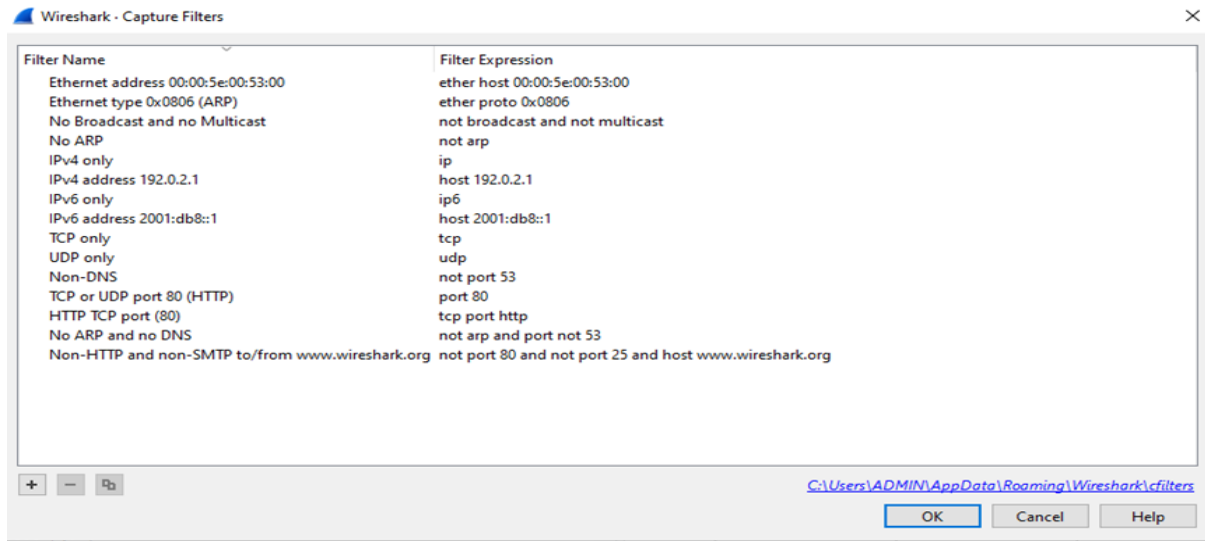


Figure 17: Capture Filters.

The Filtering of Data is shown in Figure 18.

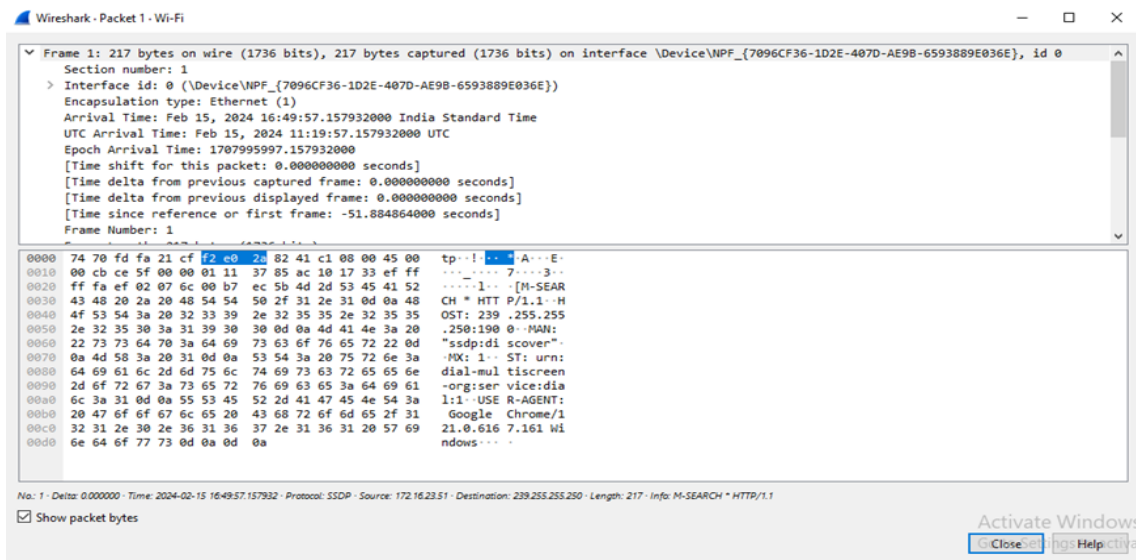


Figure 18: Filtering of Data.

The destination and port address are shown in Figure 19.

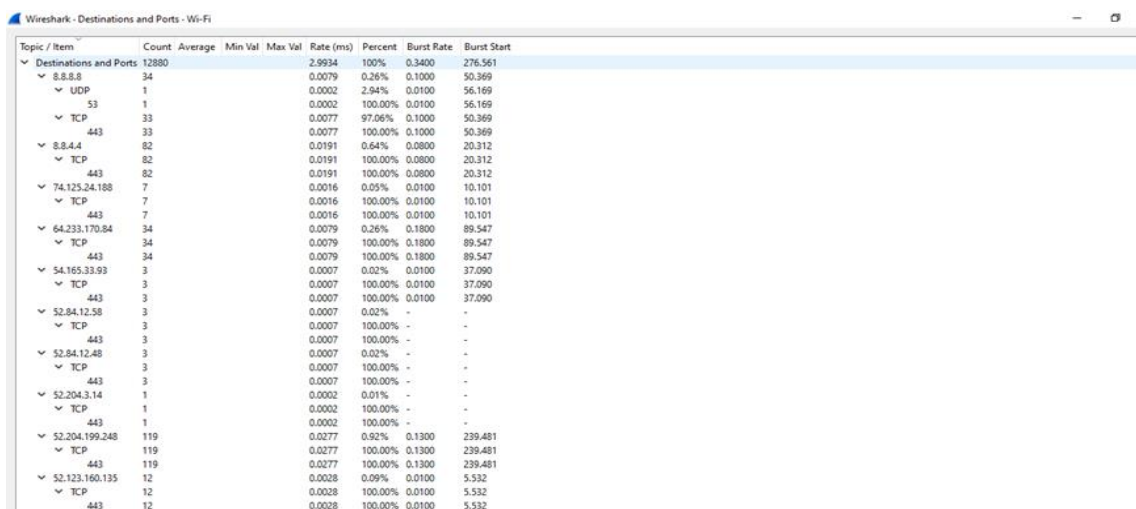


Figure 19: Destination and Ports.

VII. CONCLUSION

This paper shows case studies of packet analysis that highlight the importance of packet analyzers, particularly Wireshark, in network forensics. The current methods used by most users are not capable of detecting all computer attacks, particularly the most recent ones, without network packet analysis. Home and enterprise users have always favored antivirus software. Many antivirus programs use a signature-based detection method, but this method is inefficient for various reasons.

Packet analyzers were expensive and patented in the past. Open-source packet analyzers like Wireshark provide detailed packet data, and it is one of the best available. It is important to note that Wireshark is not an intrusion detection system, despite its powerful toolset. Network security professionals can use Wireshark to determine if strange things are happening on a network using its convenient and effective features. This paper demonstrate how Wireshark can detect security threats and attacks against networked computers.

VIII. REFERENCES

- [1] L. F. Sikos, "Packet analysis for network forensics: A comprehensive survey," *Forensic Sci. Int. Digit. Investig.*, vol. 32, p. 200892, 2020, doi: 10.1016/j.fsidi.2019.200892.
- [2] M. Alfawareh, "A Deeper Look into Network Traffic Analysis using Wireshark," *Academia.Edu*, vol. 1, pp. 4–7, [Online]. Available: https://www.academia.edu/download/58477339/LARGE_bf_A_Deeper_Look_into_Network_Traffic_Analysis_using_Wireshark.pdf
- [3] R. Tuli, "Analyzing Network Performance Parameters using Wireshark," *Int. J. Netw. Secur. Its Appl.*, vol. 15, no. 01, pp. 01–13, 2023, doi: 10.5121/ijnsa.2023.15101.
- [4] B. M. -, S. A. -, A. S. -, and R. K. -, "Exploring Wireshark For Network Traffic Analysis," *Int. J. Multidiscip. Res.*, vol. 5, no. 6, pp. 1–12, 2023, doi: 10.36948/ijfmr.2023.v05i06.8876.
- [5] P. Saxena and S. Kumar Sharma, "Analysis of Network Traffic by using Packet Sniffing Tool: Wireshark," *Int. J. Adv. Res.*, vol. 3, no. 6, pp. 804–808, 2017, [Online]. Available: www.ijariit.com
- [6] R. Kaur, "Investigating Network Traffic using Packet Sniffing Tool-Wireshark," *JETIR1901424 J. Emerg. Technol. Innov. Res.*, vol. 6, no. 1, pp. 181–186, 2019, [Online]. Available: www.jetir.org
- [7] S. K. Shandilya, C. Ganguli, I. Izonin, and P. A. K. Nagar, "Cyber attack evaluation dataset for deep packet inspection and analysis," *Data Br.*, vol. 46, p. 108771, 2023, doi: 10.1016/j.dib.2022.108771.
- [8] R. Soepeno, "Wireshark: An Effective Tool for Network Analysis," *ResearchGate*, vol. 1, no. 1, 2023, doi: 10.13140/RG.2.2.34444.69769.
- [9] V. Ndatinya, Z. Xiao, V. R. Manepalli, K. Meng, and Y. Xiao, "Network forensics analysis using Wireshark," *Int. J. Secur. Networks*, vol. 10, no. 2, pp. 91–106, 2015, doi: 10.1504/IJSN.2015.070421.
- [10] B. Dodiya and U. K. Singh, "Malicious Traffic analysis using Wireshark by collection of Indicators of Compromise," *Int. J. Comput. Appl.*, vol. 183, no. 53, pp. 1–6, 2022, doi: 10.5120/ijca2022921876.
- [11] S. Beborotta and D. Senapati, "Empirical Characterization of Network Traffic for Reliable Communication in IoT Devices," *Stud. Syst. Decis. Control*, vol. 339, no. October, pp. 67–90, 2021, doi: 10.1007/978-3-030-67361-1_3.
- [12] G. Barbieri, M. Conti, N. O. Tippenhauer, and F. Turrin, "Assessing the Use of Insecure ICS Protocols via IXP Network Traffic Analysis," *Proc. - Int. Conf. Comput. Commun. Networks, ICCCN*, vol. 2021-July, 2021, doi: 10.1109/ICCCN52240.2021.9522219.
- [13] A. Siddiqui, O. Olufunmilayo, H. Gohel, and B. Pandey, "Digital Healthcare System Vulnerability Analysis using Network Forensic Tool," *Proc. - 2021 IEEE 10th Int. Conf. Commun. Syst. Netw. Technol. CSNT 2021*, no. June 2021, pp. 881–885, 2021, doi: 10.1109/CSNT51715.2021.9509647.
- [14] A. Siswanto, A. Syukur, E. A. Kadir, and Suratin, "Network traffic monitoring and analysis using packet sniffer," *Proc. - 2019 Int. Conf. Adv. Commun. Technol. Networking, CommNet 2019*, pp. 1–4, 2019, doi: 10.1109/COMMNET.2019.8742369.
- [15] M. N. Ashaari, M. Kassim, R. A. Rahman, and A. R. Mahmud, "Performance Analysis on Multiple Device Connections of Small Office Home Office Network," *Baghdad Sci. J.*, vol. 18, no. 4, pp. 1457–1464, 2021, doi: 10.21123/bsj.2021.18.4(Suppl.).1457.