**SSN COLLEGE OF ENGINEERING KALAVAKKAM -603110**

**Department of Computer Science and Engineering**

**Assignment 2**

**UCS2501 – Computer Networks**

**Network performance evaluation using Wireshark**

**TEAM MEMBERS:**

1. **Adithya Muthukumar:- 3122 21 5001 003**
2. **Ashwin R - 3122 21 5001 014**
3. **Gayathri Venkatesan:- 3122 21 5001 026**
4. **Kakarla Tejaswi:- 3122 21 5001 039**

**INDEX**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **CONTENT** | **PAGE NO** |
| 1 | Problem Statement | 3 |
| **2** | Protocol / Method Explanation | 4 |
| **3** | Code | 8 |
| **4** | Output | 8 |
| **5** | Learning outcomes |  |
| **6** | Readme File |  |
| **7** | GitHub link |  |

PROBLEM DEFINITION:-

* In this project, we aim to evaluate the network performance like LAN, Wi-Fi and Bluetooth using Wireshark.
* The primary goals include capturing network packets, filtering the packets , evaluating their performance using statistical methods.

PROJECT OVERVIEW:-

**1. Setup of Wireshark**

Installing Wireshark is a simple process. Download the appropriate installer from the official Wireshark website and follow the on-screen instructions. After installation, launch Wireshark, select the network interface, apply filters (optional), and start capturing traffic. Analyze captured packets using Wireshark's tools and save traffic for later analysis.

**2. Capturing a network’s packets**

After installing Wireshark, choose a network to capture the packets. After choosing, you will get the real time simulation of the packets and their protocols with it’s time and source and destination IP address.

**3. Filtering the packets according to the filters to be measured.**

Before filtering, stop the capturing of the packets, right click on any packet, and apply filter to it. The filter will show all the types of available conversation filters and it will filter it accordingly. Do it for all the other packets and create separate pcapng files.

**4. Apply statistical measure for the captured packets in pcapng file**

After filtering and creating the file, now go to statistics and apply the required statistic comparison for the performance evaluation. Mostly, IO graph is used for the evaluation.

**5. Modify the IO graph Y-axis measure for packets and bytes**

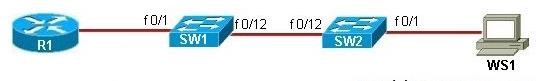
After generating the graphs, change the Y-axis of IO graph for packets to bytes and vice versa according to the needs of the user(you). This will give you an idea of the graphical analysis of the Packets-Time and Bytes-Time analysis with filtered and unfiltered packets.

**6. Documentation**

After we have completed our analysis for HTTP, IMCPv4,v6, SMB, SMTP, etc. with the pcapng file and IO graph, document the analysis in a report.

PROTOCOL/METHOD EXPLANATION:-

**Device configuration:**

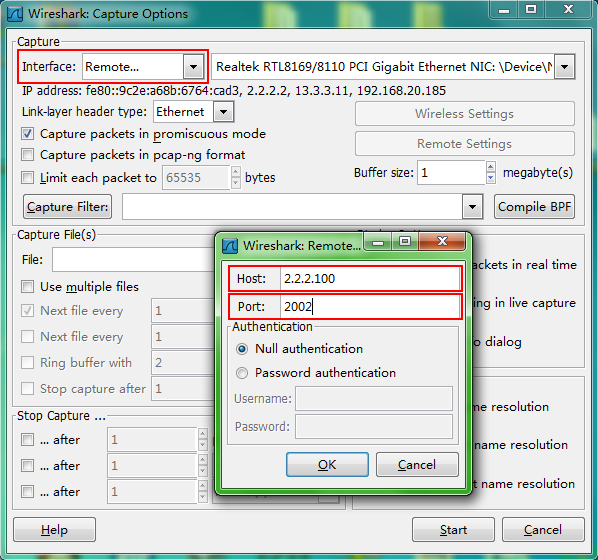


What is Packet capture in Wireshark?

Packet capture, an integral component of network analysis, entails intercepting and recording data packets as they traverse a network. Wireshark, a widely acclaimed network protocol analyzer, excels in capturing and analyzing packets, enabling you to delve into the intricacies of network traffic patterns, both in real time and from saved capture files. This process unveils invaluable insights into network behavior, empowering you to troubleshoot network issues, identify security threats, and optimize network performance.

Features of Packet Capture include:

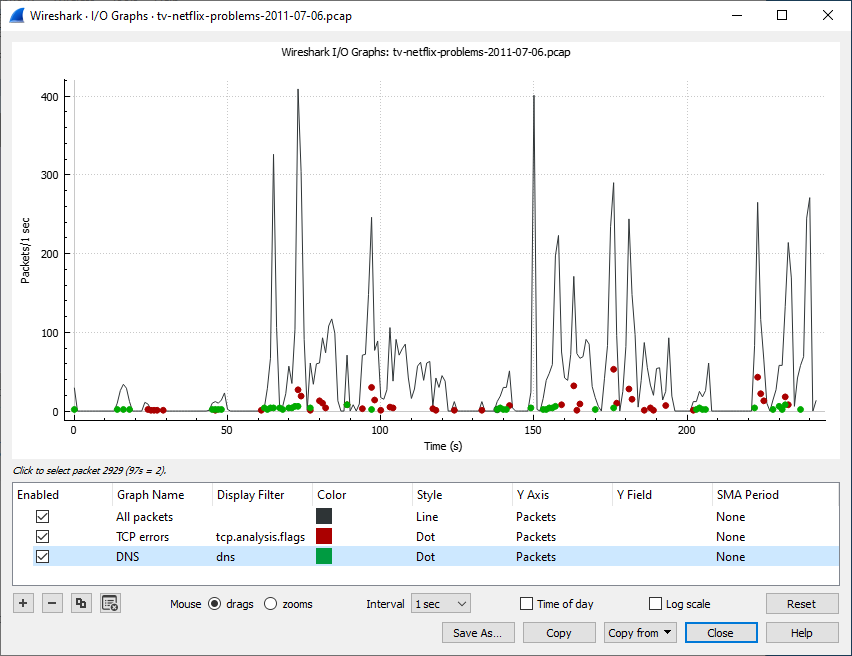
* **Live Capture:-** Wireshark can capture packets in real time from a network interface. This feature is useful for troubleshooting network problems or monitoring network activity.
* **Offline Analysis:-** Wireshark can also analyze packets that have been previously captured and stored in a capture file. This feature is useful for analyzing large amounts of traffic or for conducting in-depth analysis of specific network events.
* **Filtering:-** Wireshark provides a powerful filtering engine that allows users to filter captured packets based on a variety of criteria, such as protocol, IP address, port number, and content. This feature is useful for isolating specific traffic patterns for analysis.
* **Color Coding:-** Wireshark can colorize captured packets based on a variety of criteria, such as protocol, packet type, and error status. This feature makes it easier to identify patterns in the captured traffic.
* **Statistics:-** Wireshark provides a variety of statistics that can be used to analyze network traffic. These statistics can be used to identify trends in the traffic, such as the most common protocols used or the amount of traffic being generated by different applications.
* **Conversation Tracking:-** Wireshark can track TCP conversations, which allows users to see the entire exchange of data between two devices. This feature is useful for troubleshooting TCP-based applications.
* **Expert Tools:-** Wireshark includes a variety of expert tools that can be used to analyze network traffic in more detail. These tools include packet dissection, protocol analysis, and traffic scripting.



IO Graph in Wireshark:-

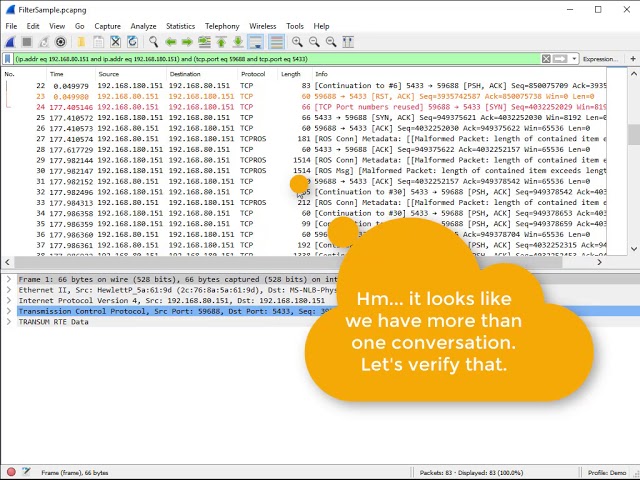
The IO Graph in Wireshark is a visual representation of network traffic over time. It displays the amount of data transmitted and received on each network interface, as well as the overall network load. The IO Graph can be used to identify trends in network traffic, such as periods of high or low activity, and to pinpoint the source of network bottlenecks.

The IO Graph is divided into two sections: the upper section displays the amount of data transmitted on each network interface, and the lower section displays the amount of data received on each network interface. The vertical axis of the graph represents the amount of data in bytes, and the horizontal axis represents time.



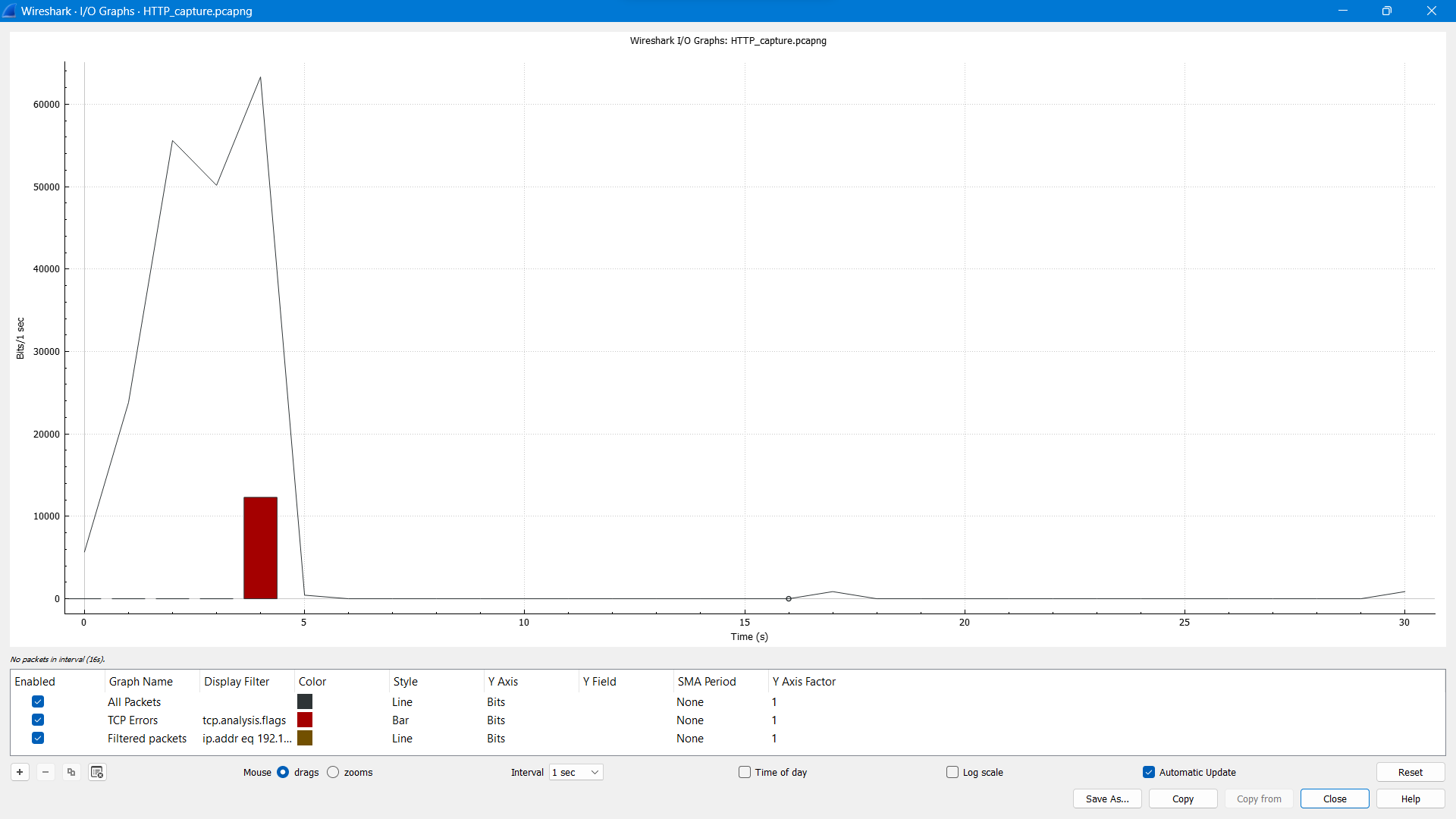
Conversational filter in Wireshark:-

* The Conversational Filter in Wireshark is a powerful tool for analyzing network traffic by identifying and isolating specific conversations between two devices. It allows users to focus on a particular conversation and view all the packets associated with that exchange, making it easier to troubleshoot network issues or identify suspicious activity.
* To apply the Conversational Filter, simply select a packet from the capture list that belongs to the conversation you want to analyze. Then, right-click on the packet and select "Follow TCP Stream" or "Follow UDP Stream" from the context menu. This will create a display filter that only shows packets that belong to the selected conversation.
* The Conversational Filter is particularly useful for analyzing TCP-based conversations, as it allows users to view the entire exchange of data between two devices. This can be helpful for troubleshooting TCP-based applications, such as web browsers or email clients.

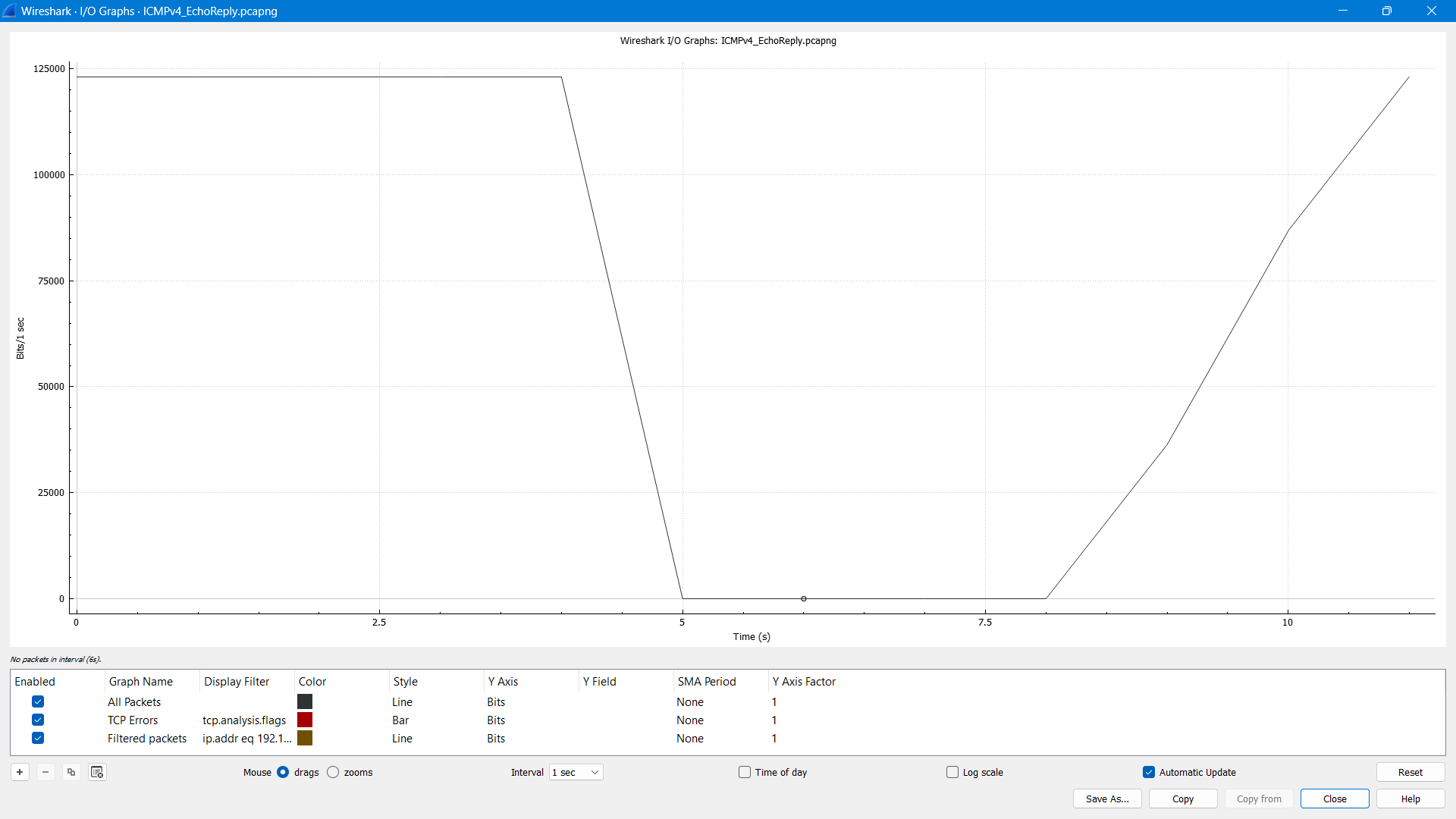


Code, Output Analysis and Conclusion:-

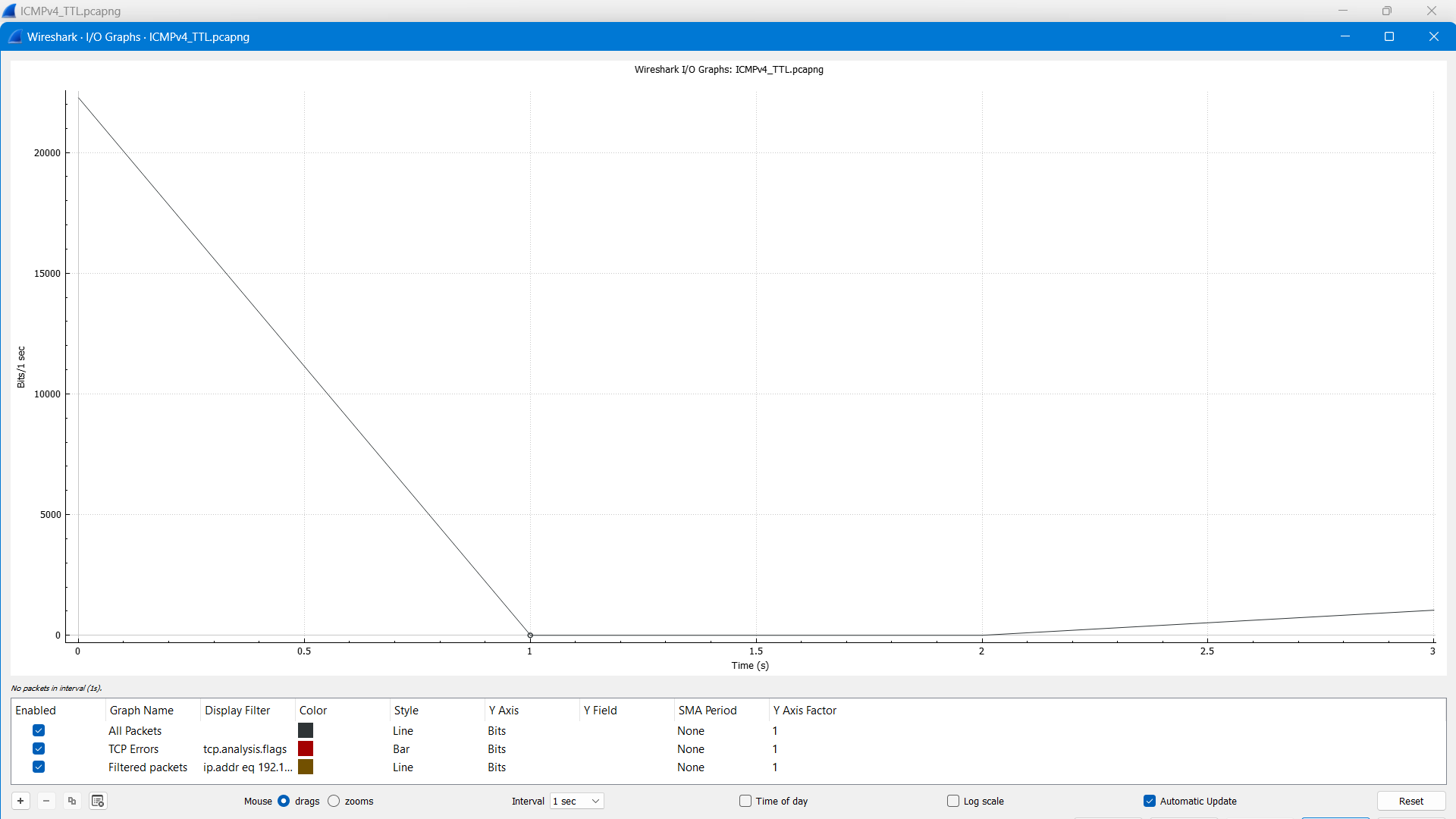
HTTP\_capture



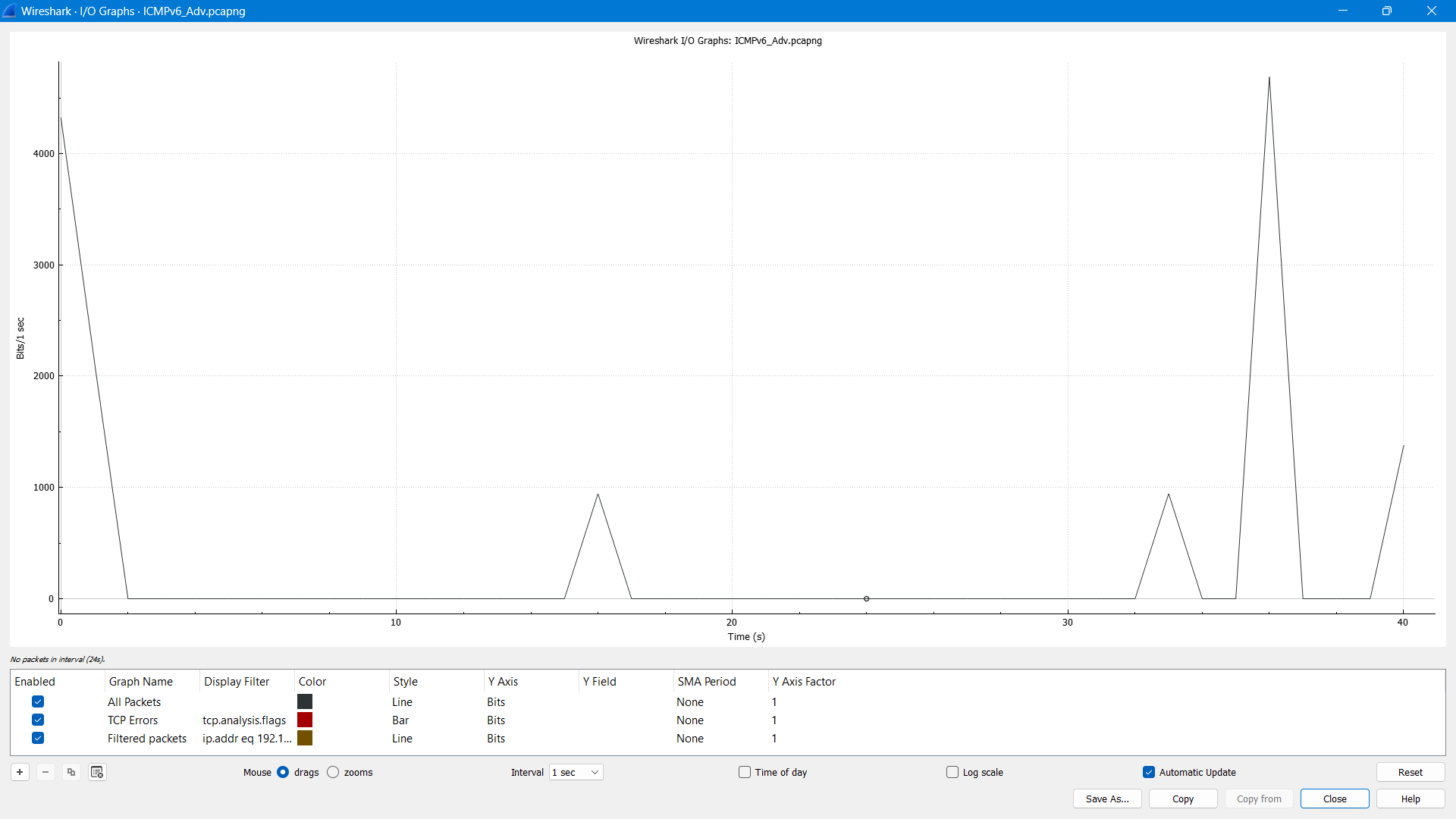
ICMPv4\_EchoReply



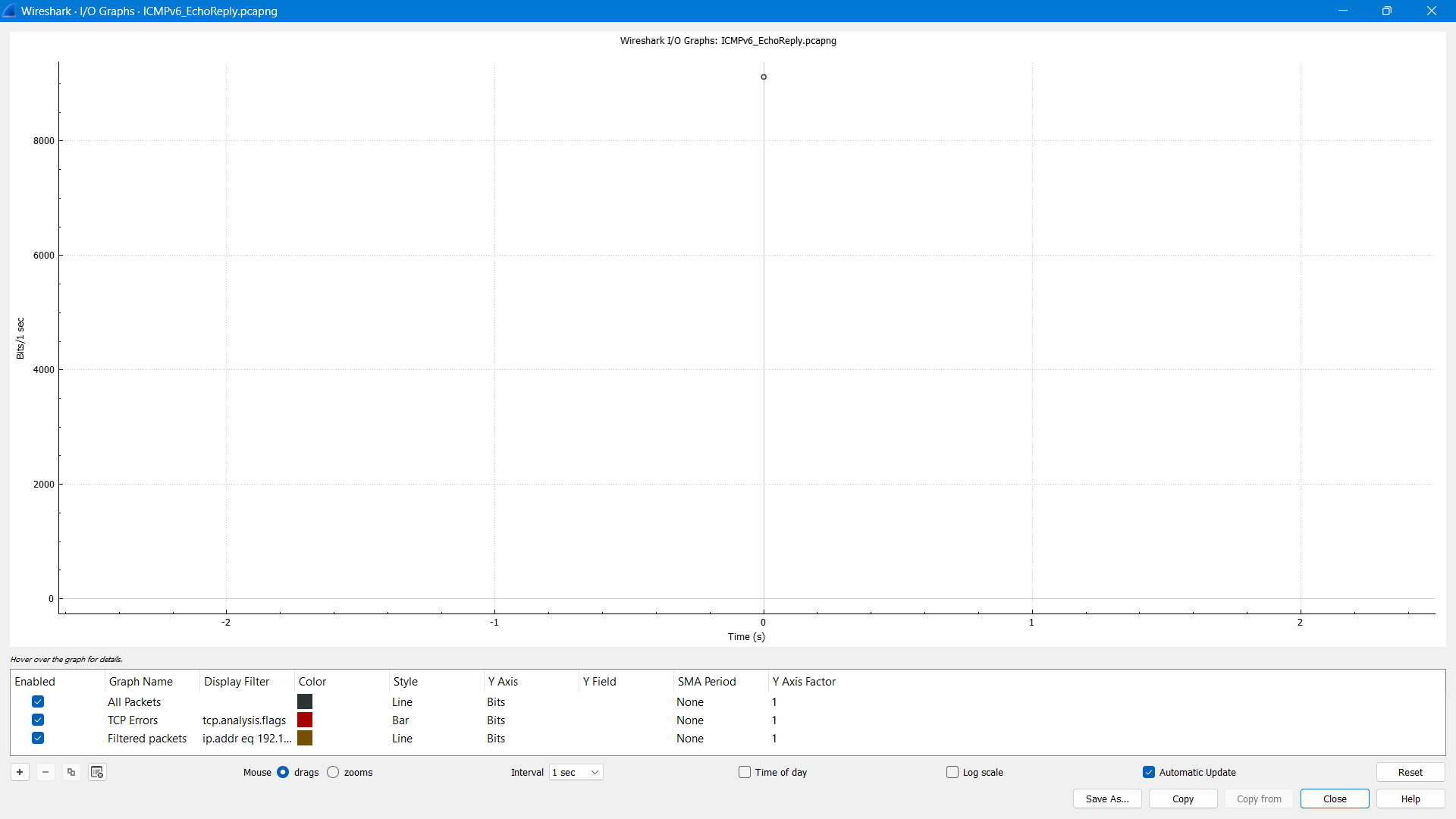
ICMPv4\_TTL



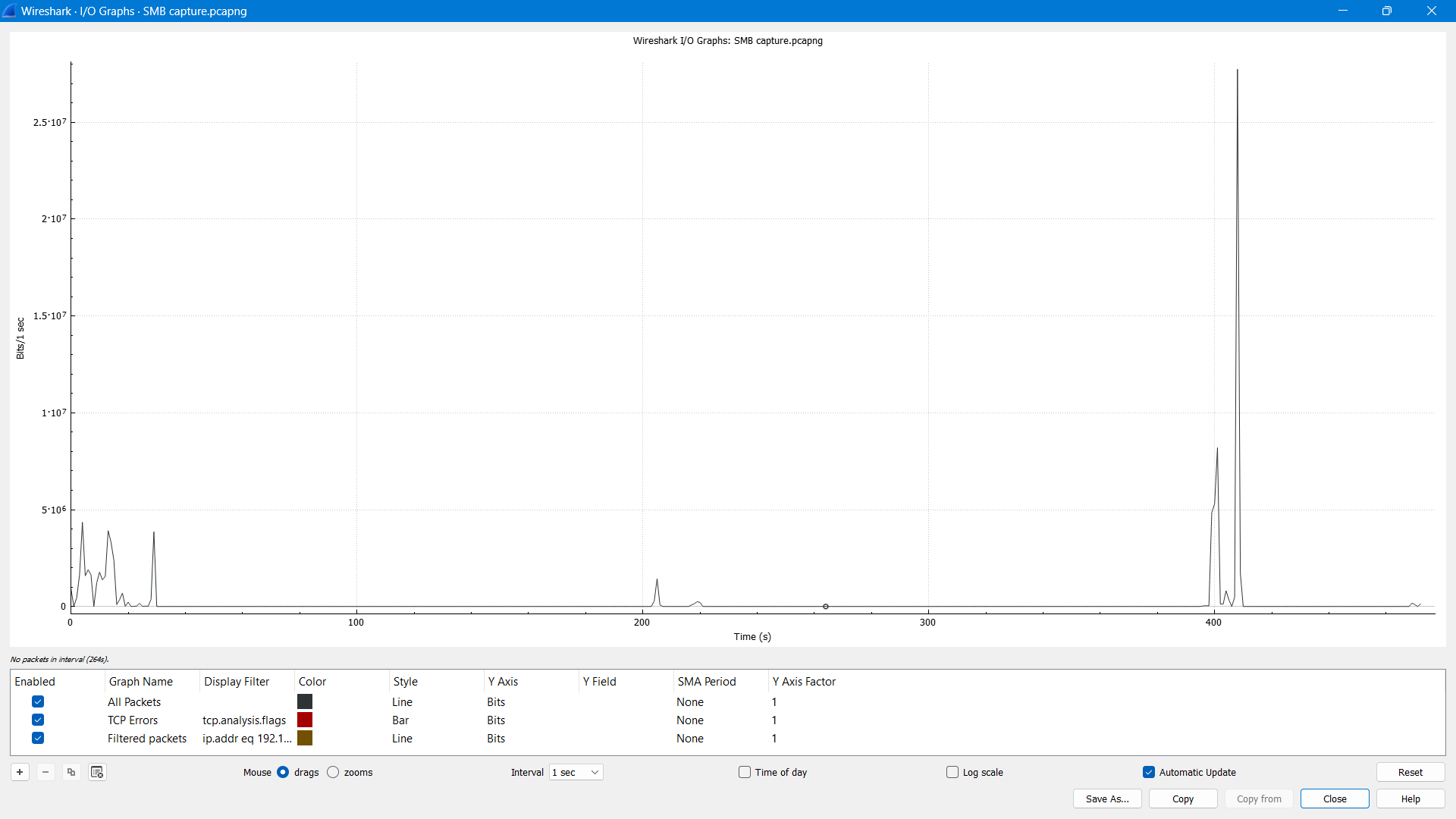
ICMPv6\_Adv



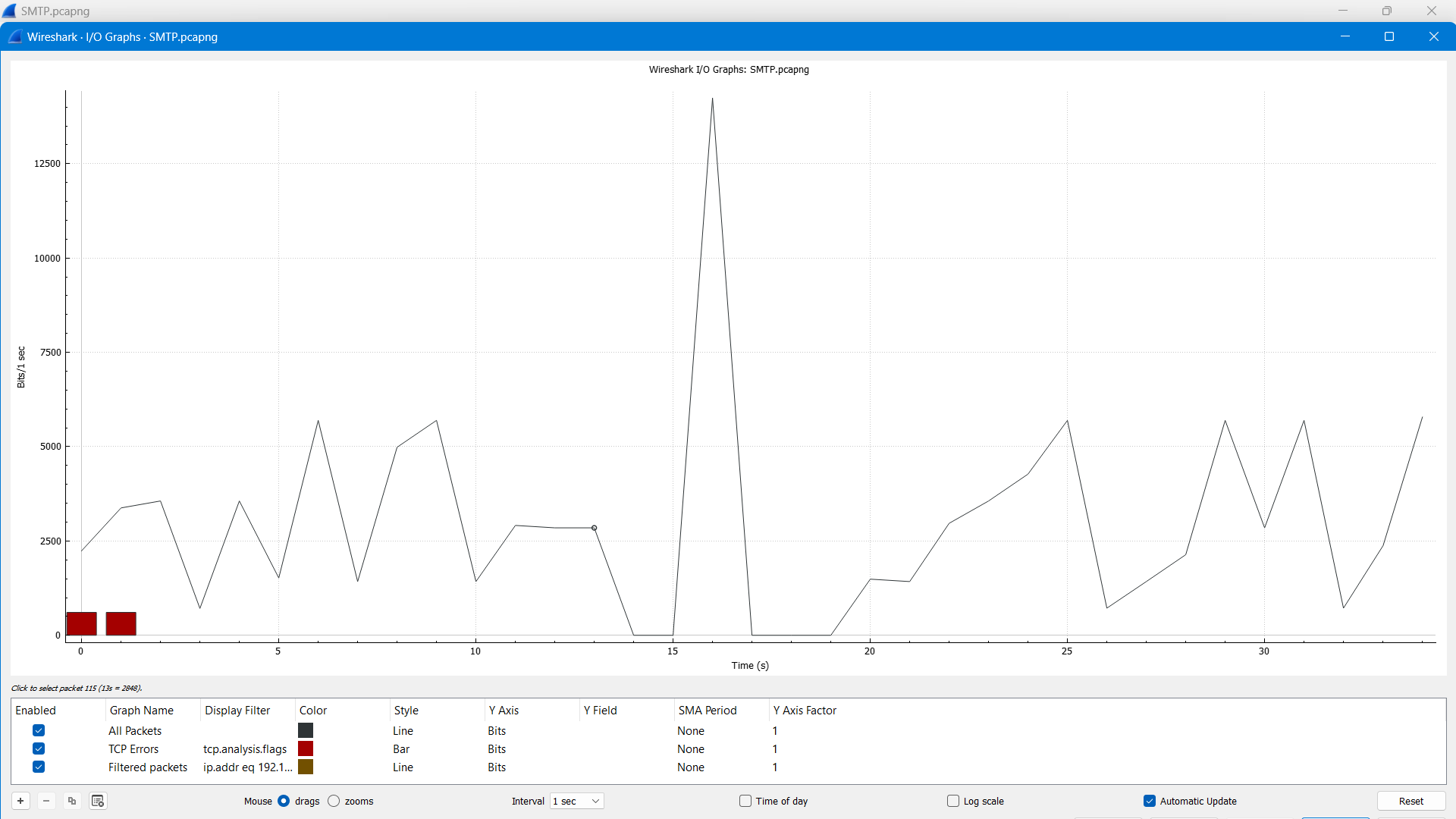
ICMPv6\_EchoReply



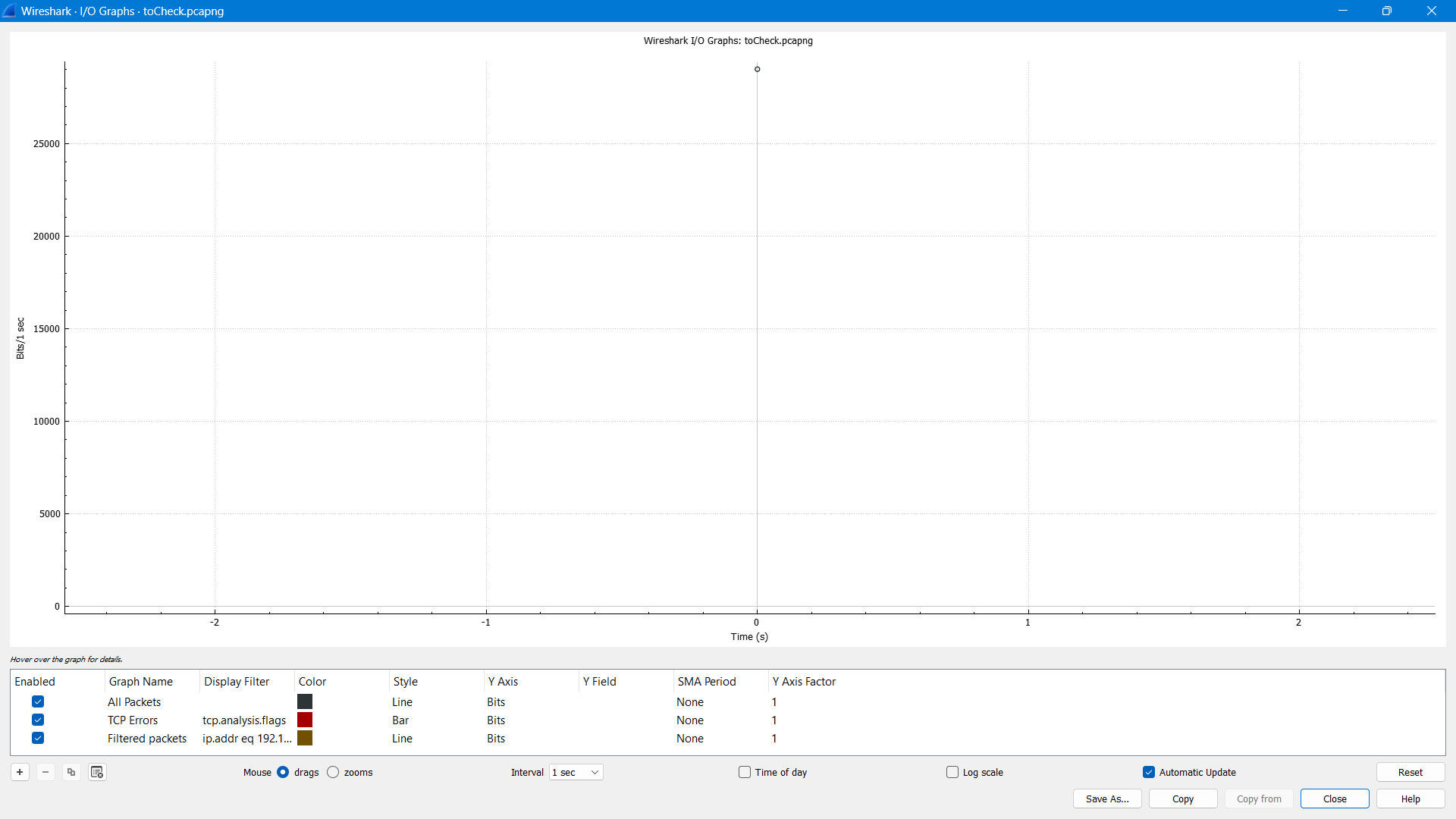
SMB capture



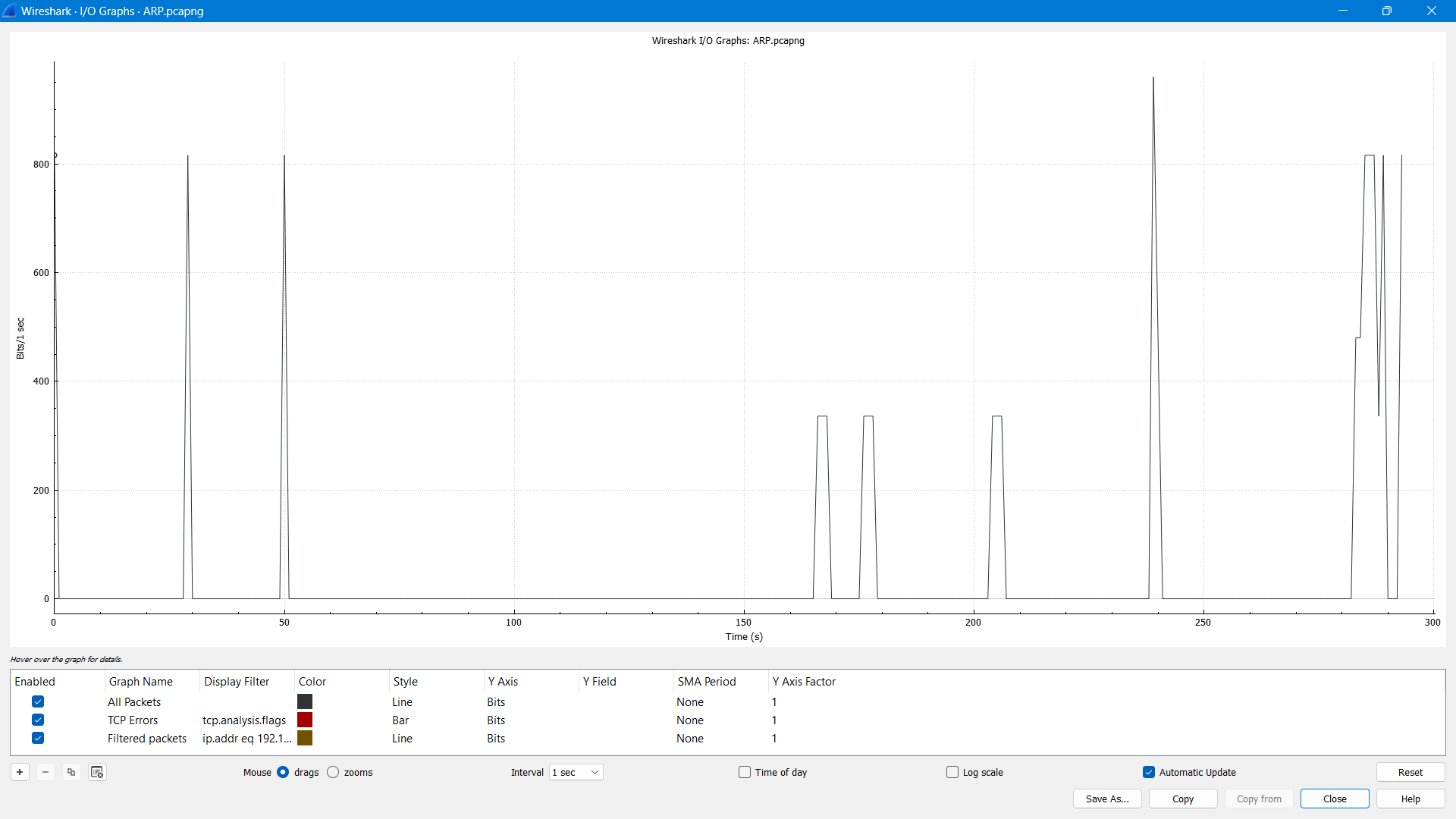
SMTP



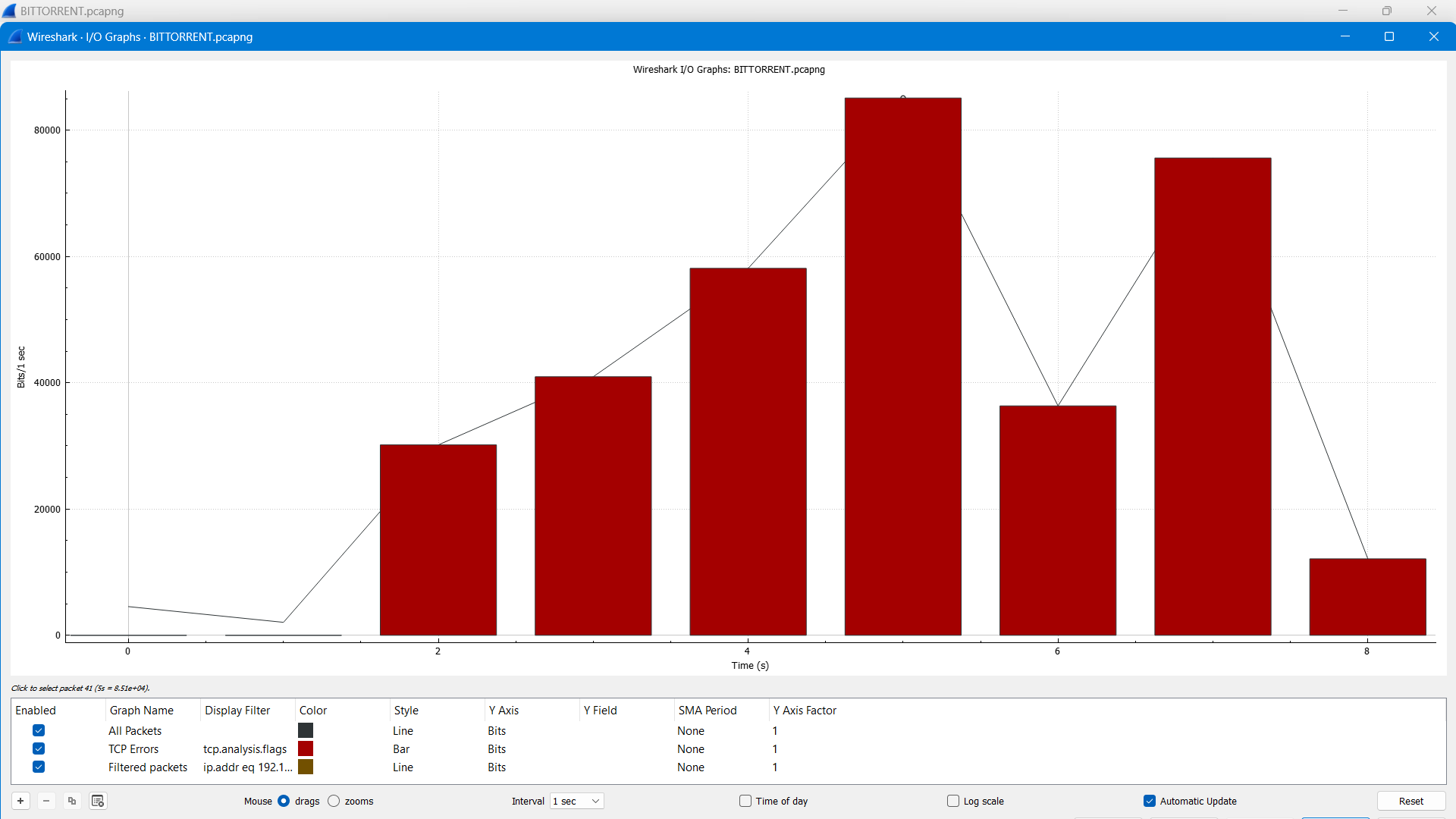
toCheck



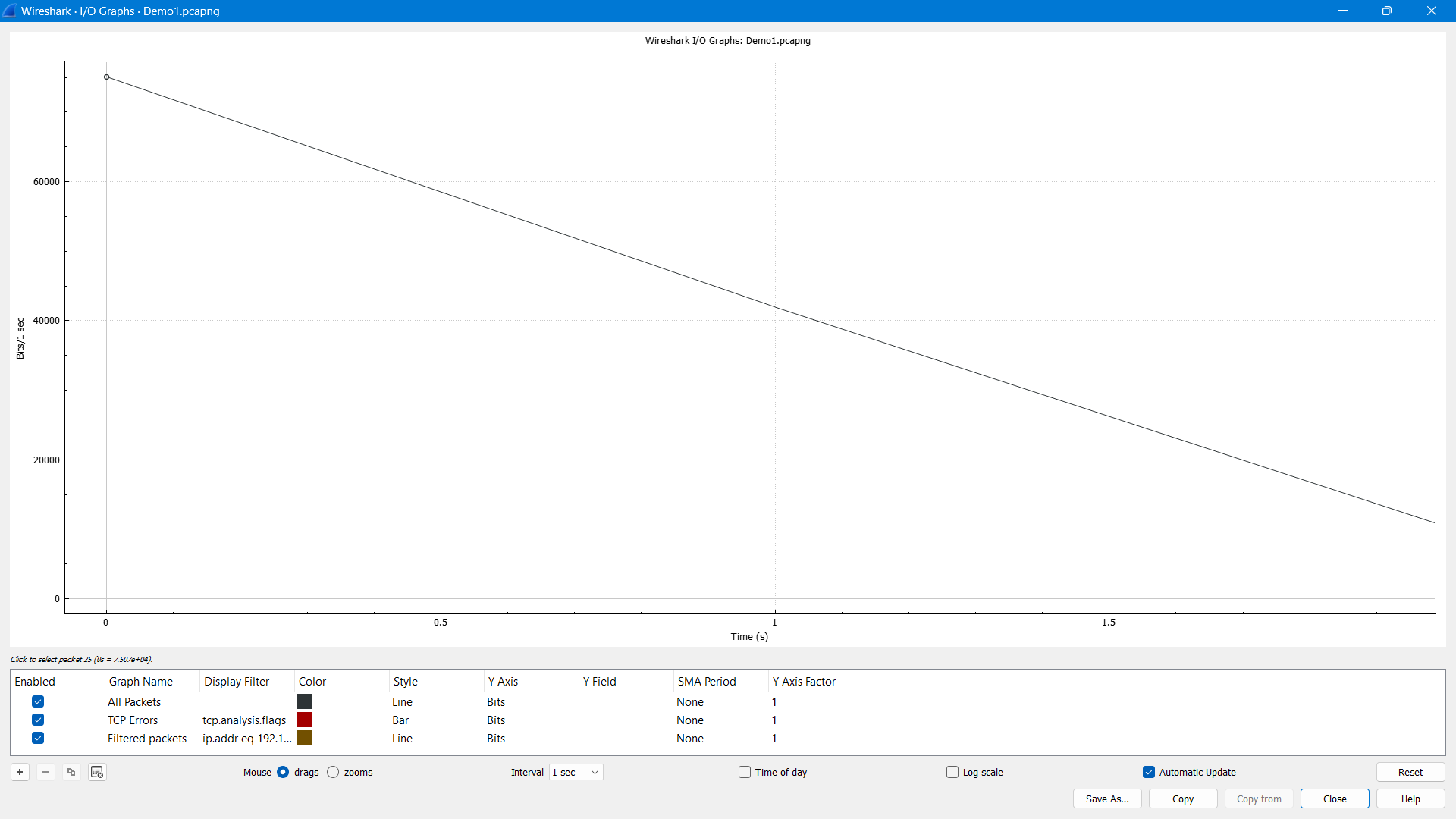
ARP



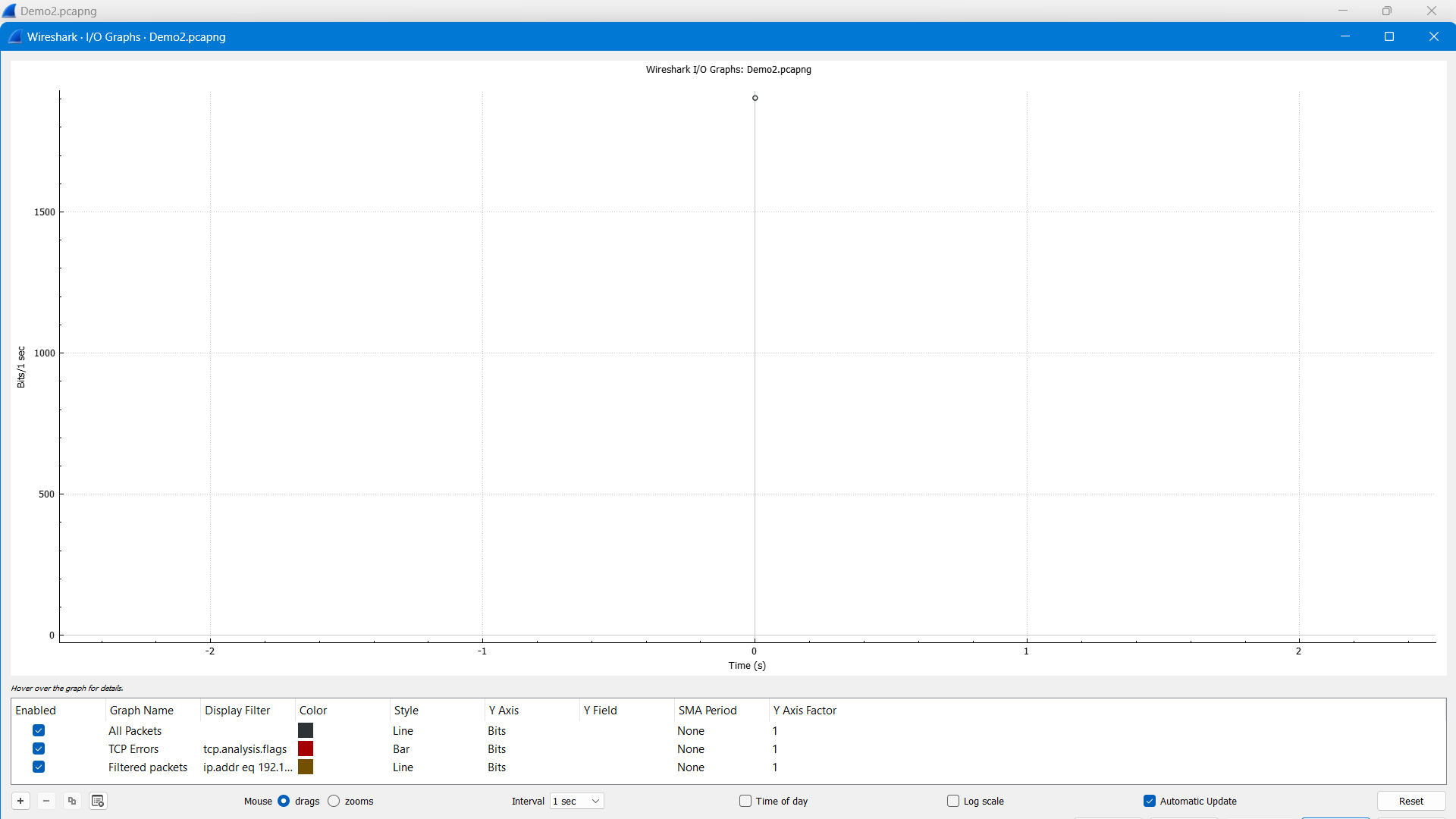
BITTORRENT



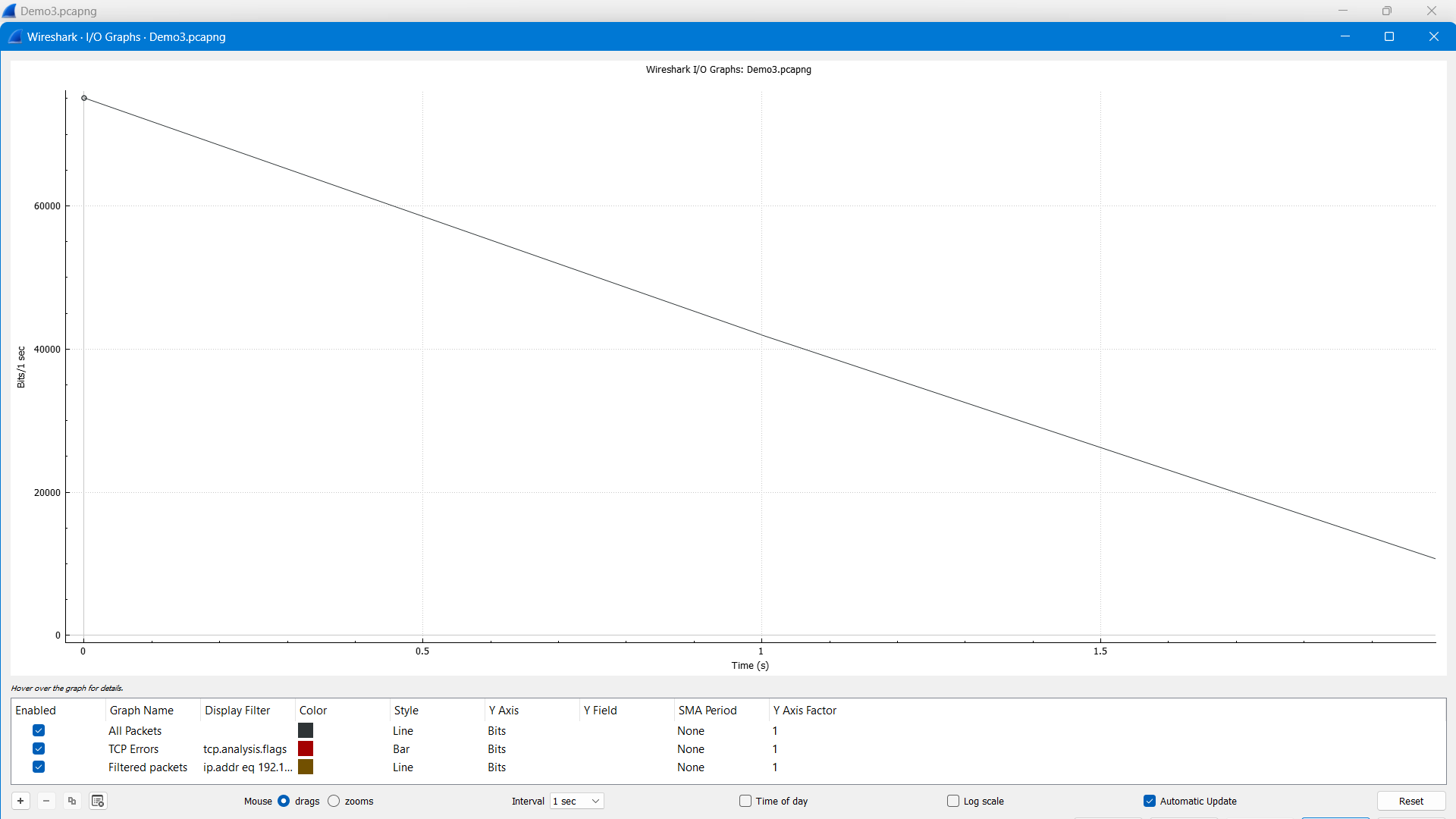
Demo1



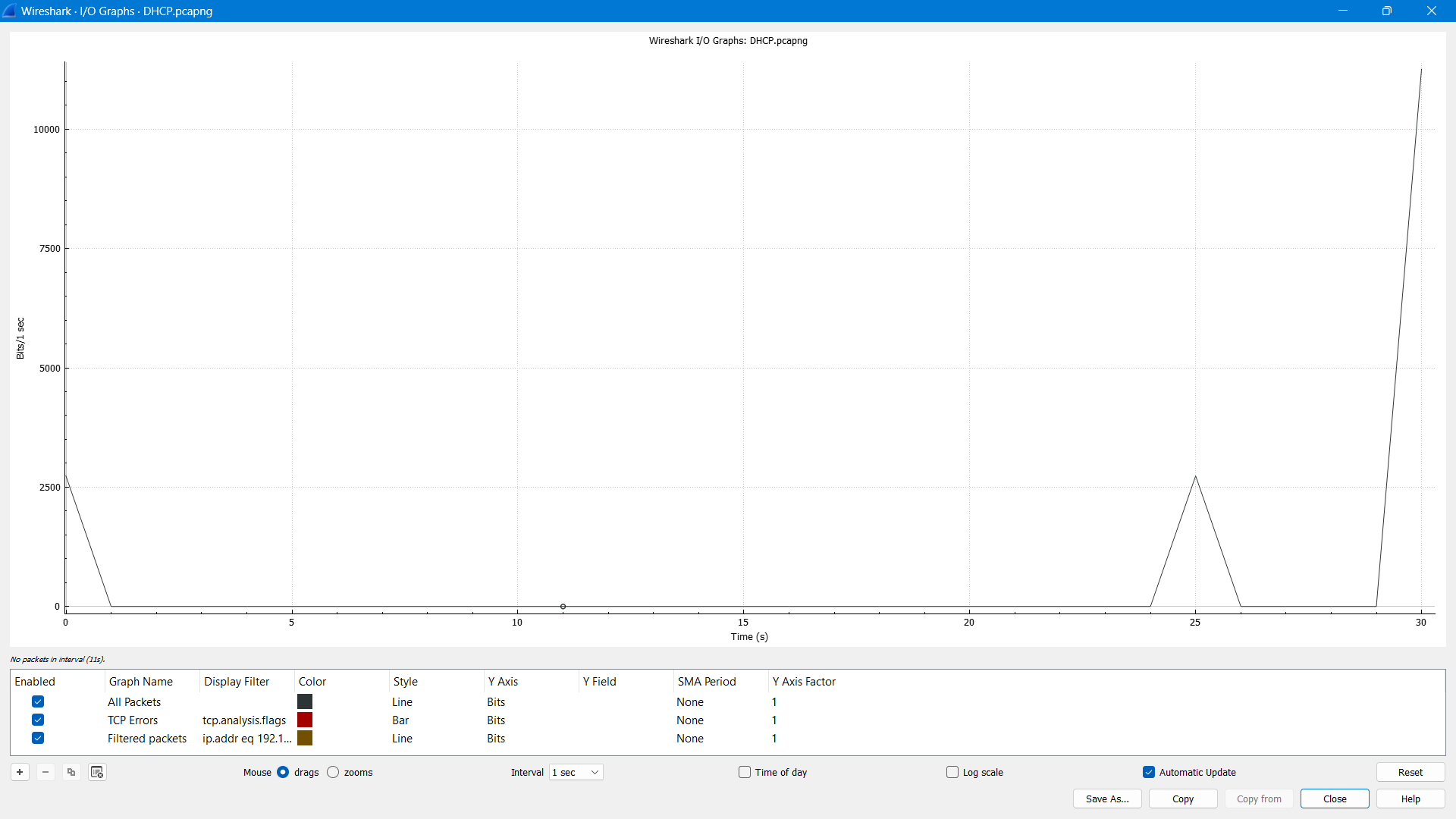
Demo2



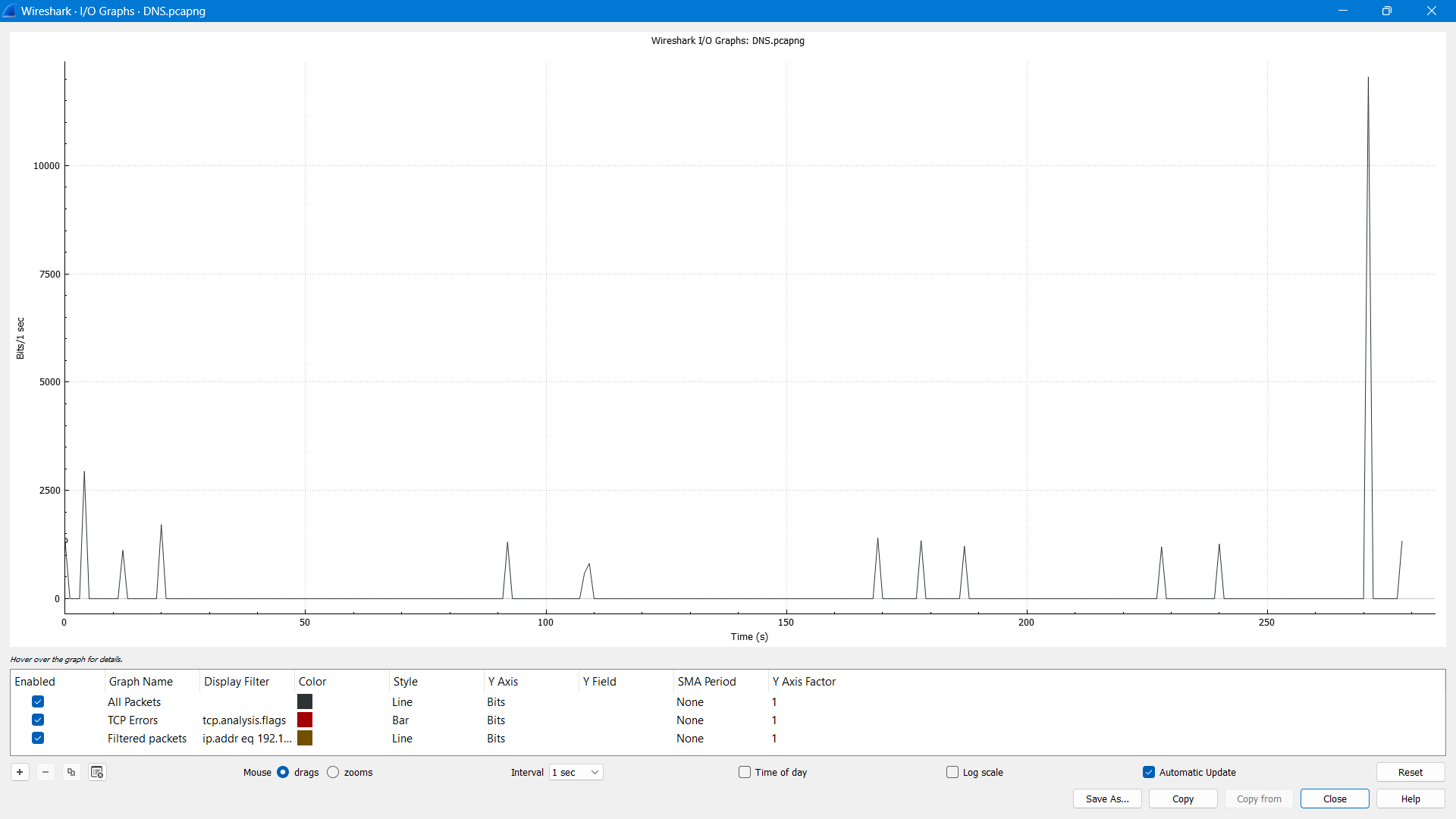
Demo3



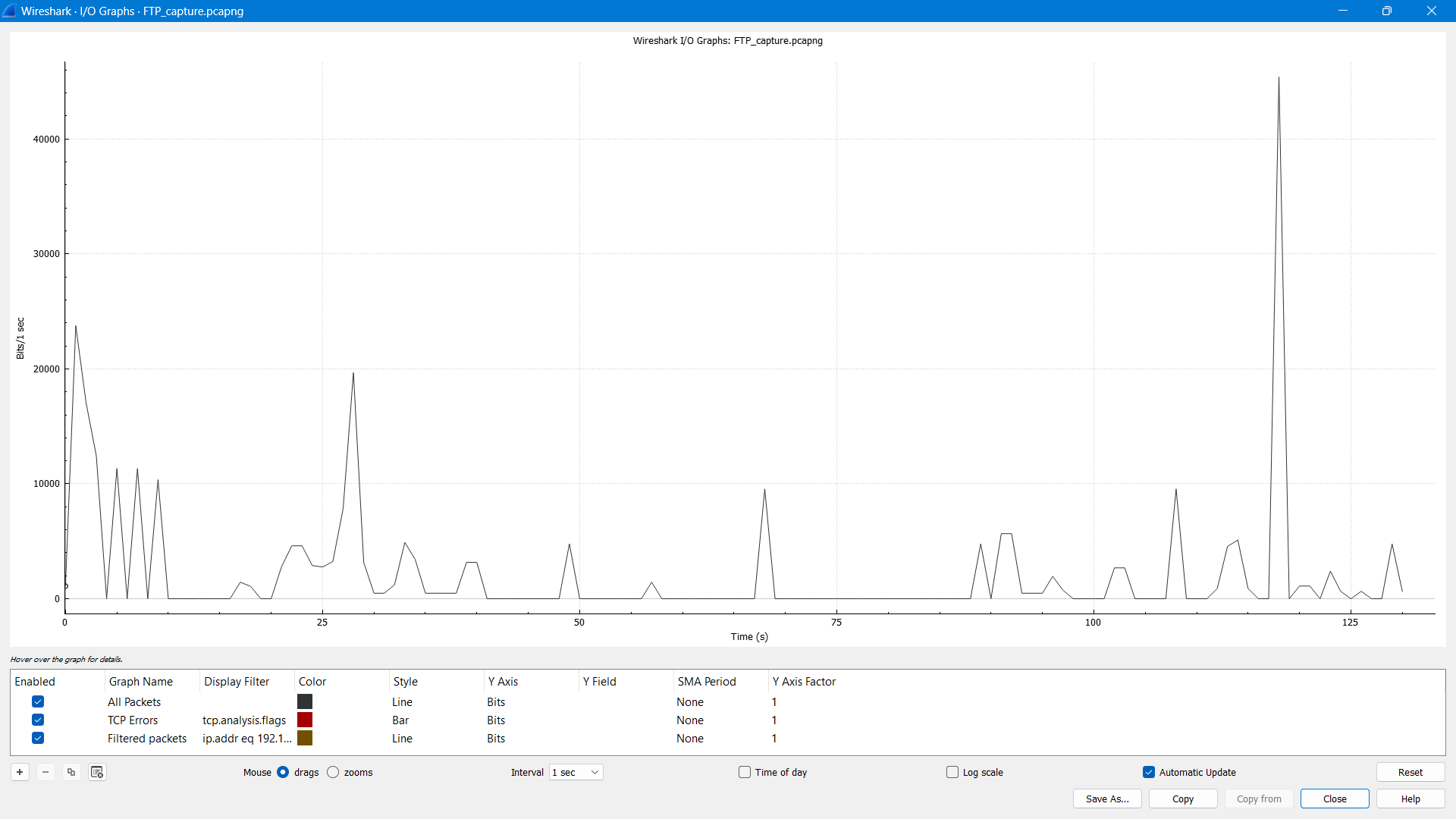
DHCP



DNS



FTP\_capture



**Conclusion:-**

Thus, the efficiency of the above-mentioned network can be found using Wireshark. This allows us to access the performance and on a larger scale, can access the performance of larger networks.

**Learning Outcomes:-**

1. Comprehensive Understanding of Network Performance Metrics

* Thoroughly define and explain key network performance metrics such as latency, throughput, packet loss, and jitter.
* Analyse and interpret trends in network performance metrics using statistical analysis and graphical representations.

2. Proficient Utilization of Wireshark for Network Traffic Capture and Analysis

* Confidently configure Wireshark to capture network traffic from various sources and interfaces.
* Employ filtering techniques strategically to isolate specific traffic patterns for in-depth analysis.
* Leverage Wireshark's colouring rules effectively to visualize network traffic patterns intuitively.
* Utilize Wireshark's statistical tools adeptly to gain insights into network performance trends.

3. Effective Network Performance Troubleshooting Using Wireshark's Capabilities

* Diagnose network connectivity problems accurately by identifying packet loss, latency, or routing issues.
* Analyse application performance bottlenecks skilfully by examining TCP and UDP conversations.
* Utilize Wireshark's statistical and graphical tools expertly to identify and resolve network bottlenecks.
* Detect and mitigate security threats promptly by analyzing suspicious network activity captured in Wireshark.

4. Network Performance Optimization through Comprehensive Analysis and Implementation

* Establish network performance baselines confidently and identify areas for improvement.
* Analyse network traffic patterns meticulously to optimize resource allocation and network configuration.
* Implement network performance optimization strategies effectively based on insights gained from Wireshark analysis.
* Validate the effectiveness of network performance optimization techniques through rigorous testing using Wireshark.

5. Advanced Network Performance Analysis with Expertise

* Utilize advanced Wireshark features proficiently, such as protocol dissection, expert tools, and scripting for in-depth analysis.
* Apply knowledge of network protocols and troubleshooting techniques judiciously to interpret complex network traffic patterns.
* Effectively communicate network performance findings and recommendations to stakeholders in a clear, concise, and professional manner.
* Continuously stay up-to-date with the latest Wireshark features and network performance evaluation methodologies through ongoing learning and industry engagement.

**Readme file and GitHub link:-**

Network Performance Evaluation Using Wireshark

This repository provides a comprehensive guide to evaluating network performance using Wireshark, a powerful network protocol analyzer. This guide is designed to empower network administrators, security analysts, and network engineers with the knowledge and skills to effectively capture, analyze, and optimize network traffic, ensuring a well-performing and secure network infrastructure.

Prerequisites

* Basic understanding of networking concepts and protocols
* Familiarity with Wireshark installation and operation
* Access to a network environment for capturing and analyzing traffic

GitHub link:- [SolituteAsh/Wireshark: Network performance evaluation using Wireshark (github.com)](https://github.com/SolituteAsh/Wireshark)