**LOVELY PROFESSIONAL UNIVERSITY**



**Assignment**

**On**

**Open Source & Technologies**

**Course Title: Wireshark**

**Course Code: INT-301**

**Submitted to:**

**Rajeshwar Sharma**

**Submitted by:**

**Chirag Khemchandani**

**Date of submission- 10/04/23**

**Table of Content:**

1. **Introduction**
   1. **Objective**
   2. **Description**
   3. **Scope**
2. **System Description**
   1. **What is Wireshark?**
   2. **How does Wireshark work?**
   3. **Assumptions and Dependencies**
   4. **Functional/Non-Functional Dependencies**
3. **Analysis Report**
   1. **Capturing packet from the network connection**
   2. **Tracing connections**
   3. **Viewing the contents of suspect network transactions in Wireshark**
   4. **Identifying bursts of network traffic in Wireshark**
4. **Conclusion**
5. **Github link**
6. **References/Bibliography**
7. **Introduction:**

**1.1 Objective:**

The objective is to assess our ability to work with open source software tools to capture and analyze network traffic. It requires us to demonstrate our knowledge and skills in using the Wireshark tool to capture packets from a network connection, trace connections between hosts, view the contents of suspect network transactions, and identify bursts of network traffic. By successfully completing this task, we’ll demonstrate our proficiency in using Wireshark to troubleshoot network issues, identify performance problems, and detect security threats on a network.

* 1. **Description:**

The project involves using open source software to capture, analyze, and report on network traffic. Specifically, the project requires using Wireshark, an open source packet analyzer, to capture packets from a network connection, trace connections between hosts, view the contents of suspect network transactions, and identify bursts of network traffic.

To accomplish this task, the project may involve setting up a test network or capturing packets from an existing network connection. The captured packets can then be analyzed using Wireshark's various filters and analysis tools to identify suspicious or anomalous traffic patterns.

Once the analysis is complete, the project requires generating a report that summarizes the findings and identifies any issues or threats on the network. The report may include visualizations, such as graphs or charts, to help illustrate the traffic patterns and identify areas of concern. Additionally, the report may include recommendations for remediation or further investigation.

Overall, the project aims to demonstrate the ability to use open source tools to effectively capture, analyze, and report on network traffic in order to identify potential issues or threats and take appropriate action to address them.

* 1. **Scope:**

The scope of the project involves using open source software to capture, analyze, and report on network traffic. Specifically, the project requires using Wireshark, an open source packet analyzer, to capture packets from a network connection, trace connections between hosts, view the contents of suspect network transactions, and identify bursts of network traffic.

The project scope includes the following:

1. Setting up a test network or capturing packets from an existing network connection.
2. Using Wireshark to capture packets from the network connection.
3. Analyzing the captured packets using Wireshark's filters and analysis tools to identify suspicious or anomalous traffic patterns.
4. Generating a report that summarizes the findings and identifies any issues or threats on the network.
5. Including visualizations, such as graphs or charts, to help illustrate the traffic patterns and identify areas of concern.
6. Providing recommendations for remediation or further investigation.

The project scope does not include the actual remediation of any identified issues or threats, as this may require additional resources and expertise.

Overall, the scope of the project is focused on using open source software to capture, analyze, and report on network traffic in order to identify potential issues or threats and take appropriate action to address them.

1. **System Description:**
   1. **What is Wireshark?**

Open-source software for network traffic analysis called Wireshark is widely regarded as the industry standard. Wireshark is supported and updated for new network technologies and encryption techniques by a multinational organisation of network specialists and software developers.

Wireshark is used by businesses, non-profits, government organizations, and educational institutions for teaching and troubleshooting purposes. Looking at traffic using Wireshark is the best method to learn low-level networking, in my opinion.

Wireshark should only be used on networks where inspecting network packets is permitted. It is prohibited to use Wireshark to examine packets without authorization.

* 1. **How does Wireshark work?**

A packet sniffer and analysis tool is called Wireshark. It records network traffic from a variety of connections, including Ethernet, Bluetooth, wireless (IEEE.802.11), Token Ring, and Frame Relay connections, and stores the information for further analysis.

Editor's note: A single message from any network protocol is referred to as a "packet." (e.g., TCP, DNS, etc.).

Editor's Note 2: Because LAN traffic is broadcast, a machine running Wireshark can see traffic going between two other computers. You must capture the packets on the local computer in order to see communication to an external site.

In order to focus in and narrow down on what you're searching for in the network trace, Wireshark gives you the option to filter the log either before the capture begins or during analysis. For instance, you can configure a filter to only display packets coming from one computer or to see TCP activity between two IP addresses. One of the main factors contributing to Wireshark's status as the de facto packet analysis tool is its filters.

* 1. **Assumptions and Dependencies:**
* **Assumptions:**

1. The test network or the existing network connection being used for the project is representative of the actual network environment where the open source software will be implemented.
2. The network being analyzed is not encrypted, as this would make it difficult to analyze the contents of the packets.
3. There are no legal or regulatory restrictions that prevent the capture and analysis of network traffic.
4. The users who will be working with the open source software have the necessary knowledge and skills to effectively use Wireshark and other tools to capture and analyze network traffic.

* **Dependencies:**

1. The successful completion of the project is dependent on the availability of a network connection or test network that can be used for capturing packets.
2. The project is dependent on the reliability and accuracy of the Wireshark tool, as any inaccuracies or errors in the captured packets could affect the accuracy of the analysis.
3. The ability to generate accurate and actionable reports is dependent on the quality of the captured data and the analysis performed on it.

4.The project may be dependent on the availability of additional resources or expertise, such as network engineers or security experts, to address any identified issues or threats.

Overall, the success of the project is dependent on the accuracy of the captured data, the reliability of the Wireshark tool, and the ability of the project team to effectively analyze and report on the network traffic.

* 1. **Functional/Non-Functional Dependencies:**
* **Functional Dependencies:**

1. The successful completion of the project is dependent on the functionality of the Wireshark tool to capture and analyze network traffic.
2. The ability to trace connections and identify bursts of network traffic is dependent on the functionality of the network protocol being used.
3. The ability to view the contents of suspect network transactions is dependent on the ability to decode the captured packets.

* **Non-Functional Dependencies:**

1. The performance of the Wireshark tool may be affected by the amount of network traffic being captured, which could impact the speed of the analysis and reporting.
2. The accuracy of the analysis and reporting may be affected by the expertise and experience of the project team in using the Wireshark tool and interpreting the captured data.
3. The ability to generate actionable reports may be impacted by the quality of the captured data and the completeness of the analysis.
4. The security and privacy of the captured data must be ensured throughout the project, to prevent any unauthorized access or disclosure.
5. The project may be subject to any legal or regulatory requirements related to the capture and analysis of network traffic, which may impact the scope and methodology of the project.

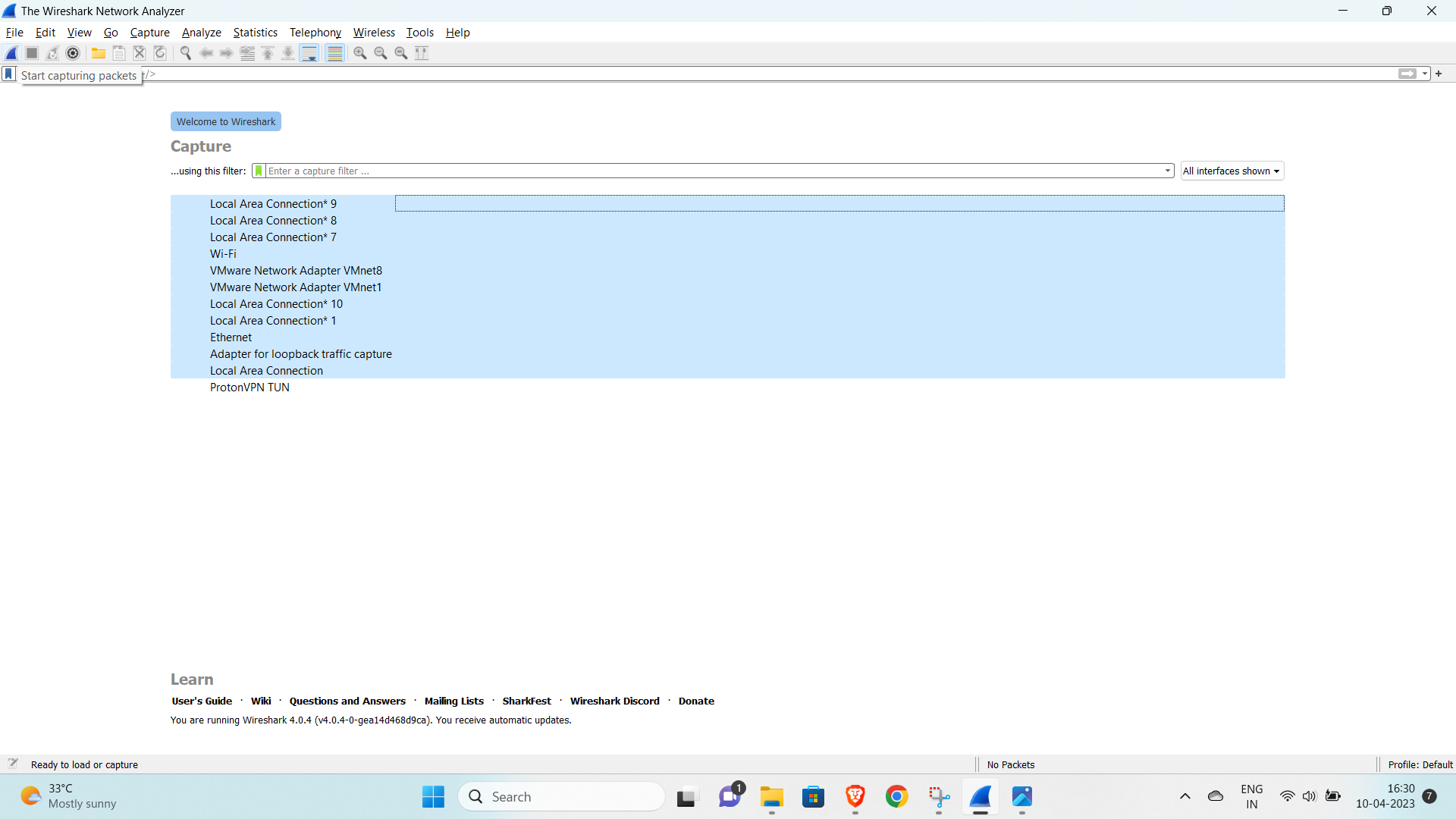
Overall, the success of the project is dependent on both functional and non-functional factors, including the performance and accuracy of the Wireshark tool, the expertise and experience of the project team, and the adherence to security, privacy, and legal requirements.

**III Analysis report:**

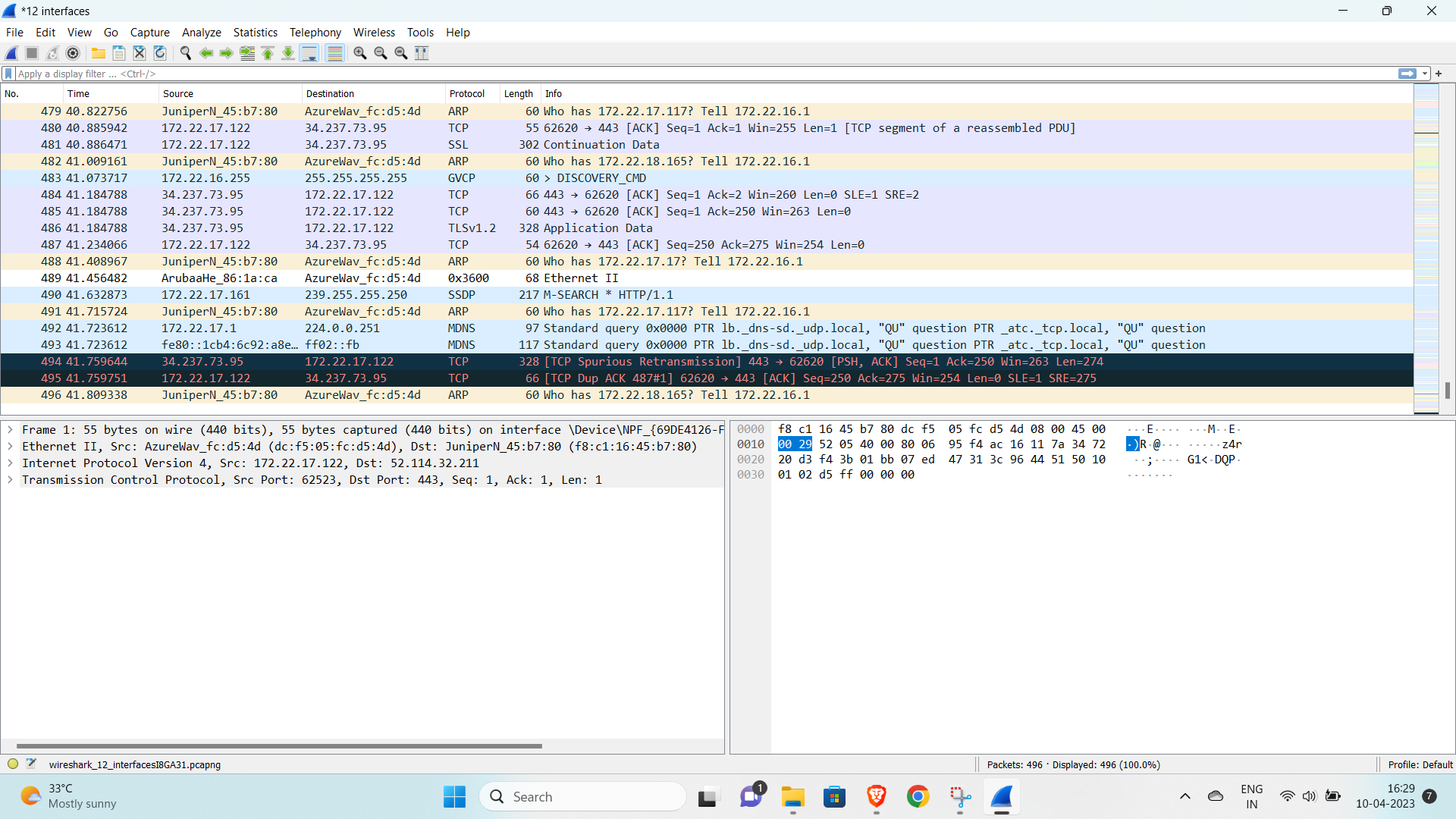
* 1. **Capturing packet from the network connection:**

To capture packets using Wireshark, follow these steps:

1. Launch Wireshark on your computer.
2. Select the network interface you want to capture packets from. You can do this by clicking on the drop-down menu next to the "Capture" button in the top toolbar.



1. Once you've selected the appropriate network interface, click on the "Capture" button in the top toolbar to start capturing packets.
2. Wireshark will begin capturing packets and displaying them in real-time. You can stop the capture at any time by clicking on the "Stop" button in the top toolbar.
3. Once you've captured packets, you can analyze them in Wireshark by applying filters or navigating through the various tabs and columns in the user interface.
4. To save the captured packets for later analysis, click on the "File" menu and select "Save As". Choose a file format (such as PCAP or PCAPNG) and a location to save the file.

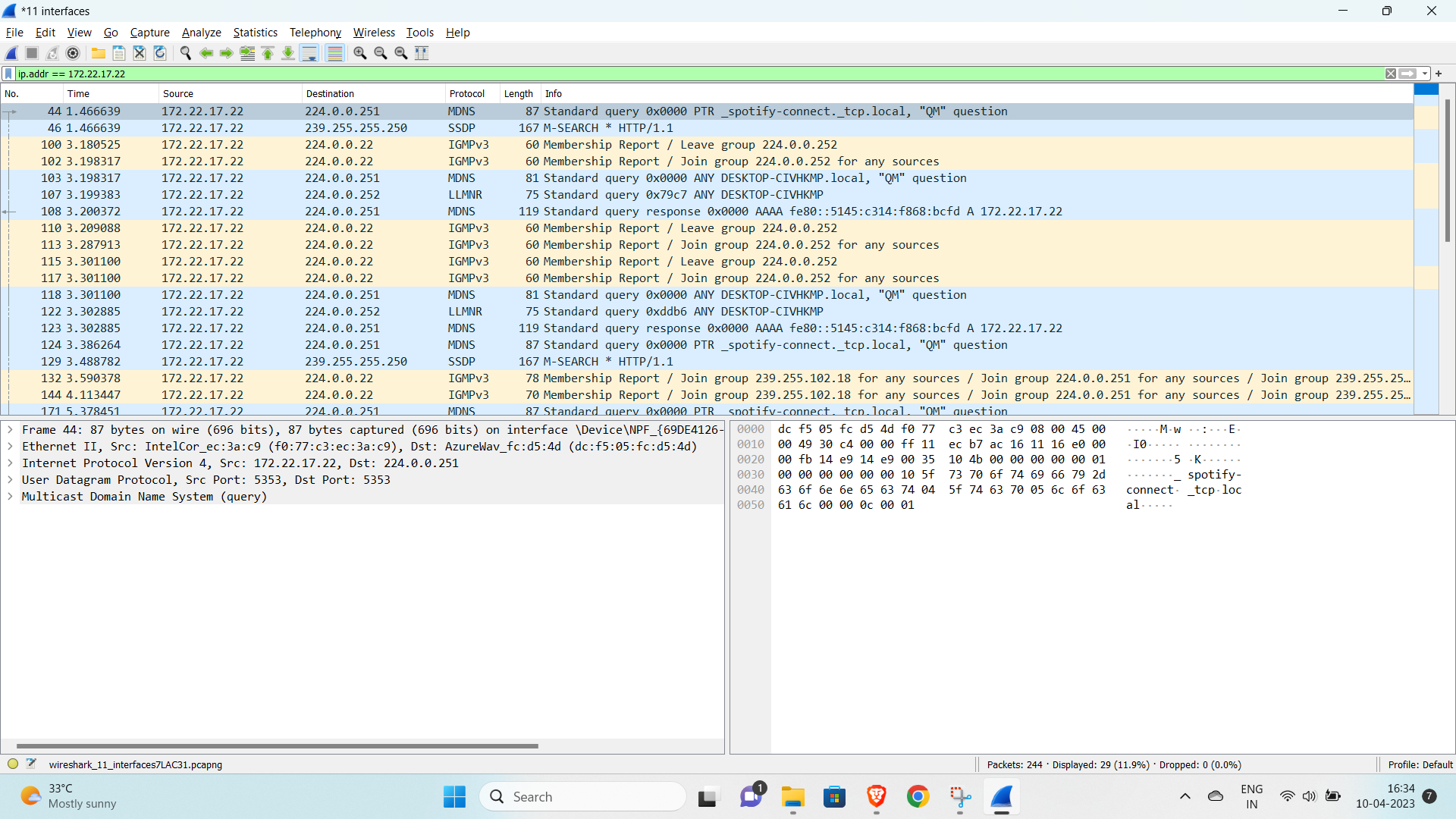


Note that in order to capture packets using Wireshark, you may need to have administrative privileges on your computer, depending on the operating system and network interface you're using. Additionally, capturing packets on a busy network can generate a lot of traffic, so be sure to limit your capture to the specific packets or protocols you're interested in.

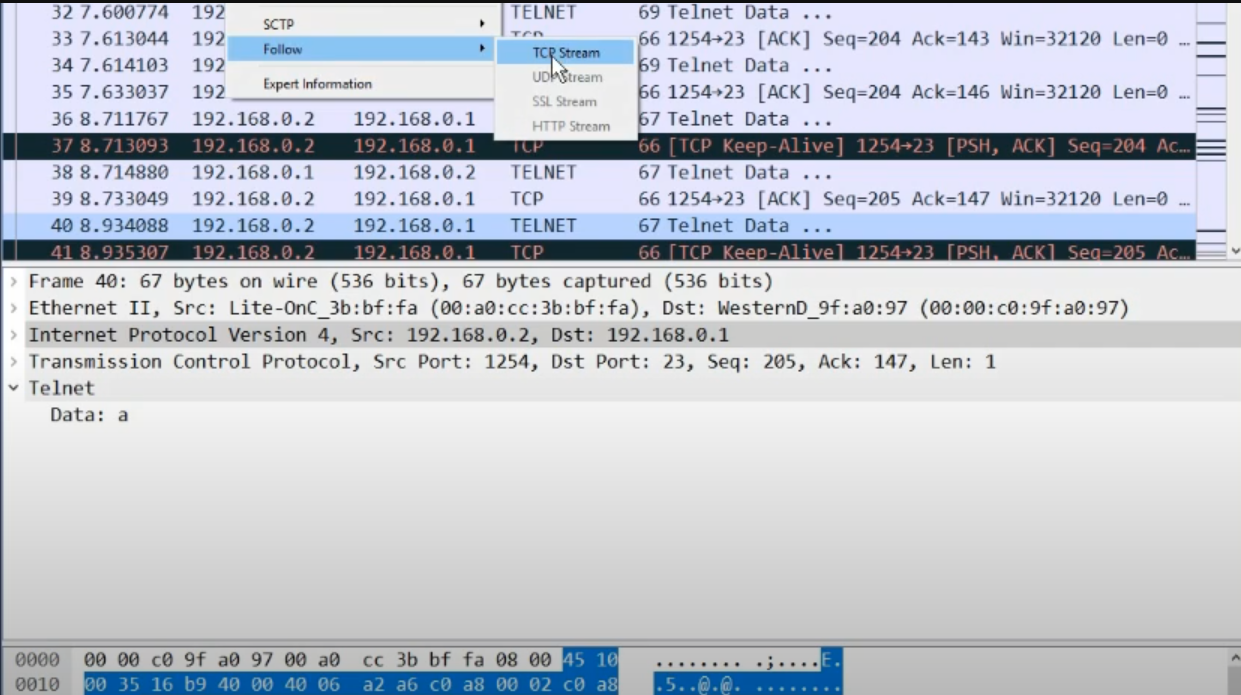
* 1. **Tracing connections:**

To trace a network connection using Wireshark, you can follow these steps:

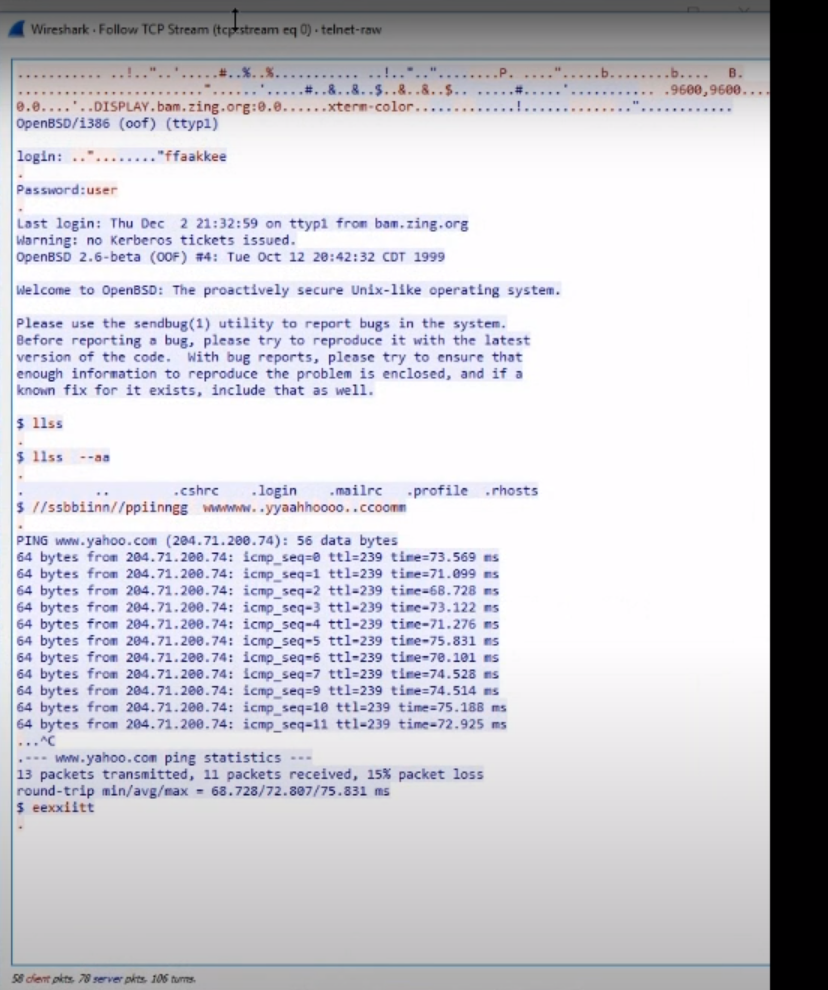
1. Start a packet capture in Wireshark by selecting the appropriate network interface and clicking on the "Capture" button.
2. Identify the IP addresses or hostnames of the two endpoints of the connection you want to trace. You can do this by looking for packets with source and destination IP addresses that match the endpoints.
3. Apply a display filter to the packet capture to show only packets that are part of the connection you want to trace. For example, you can use a filter like "ip.addr == 192.168.1.100 && ip.addr == 192.168.1.200" to show only packets between two IP addresses.



1. Once you've filtered the packet capture to show only the packets you're interested in, you can follow the connection through the sequence of packets by looking at the TCP sequence and acknowledgement numbers. These numbers are included in the TCP header of each packet and are used to keep track of the sequence of packets in a connection.
2. Use Wireshark's "Follow TCP Stream" feature to view the contents of the connection in a human-readable format. This feature will show you the complete sequence of packets in the connection and will reassemble the packets into their original order.



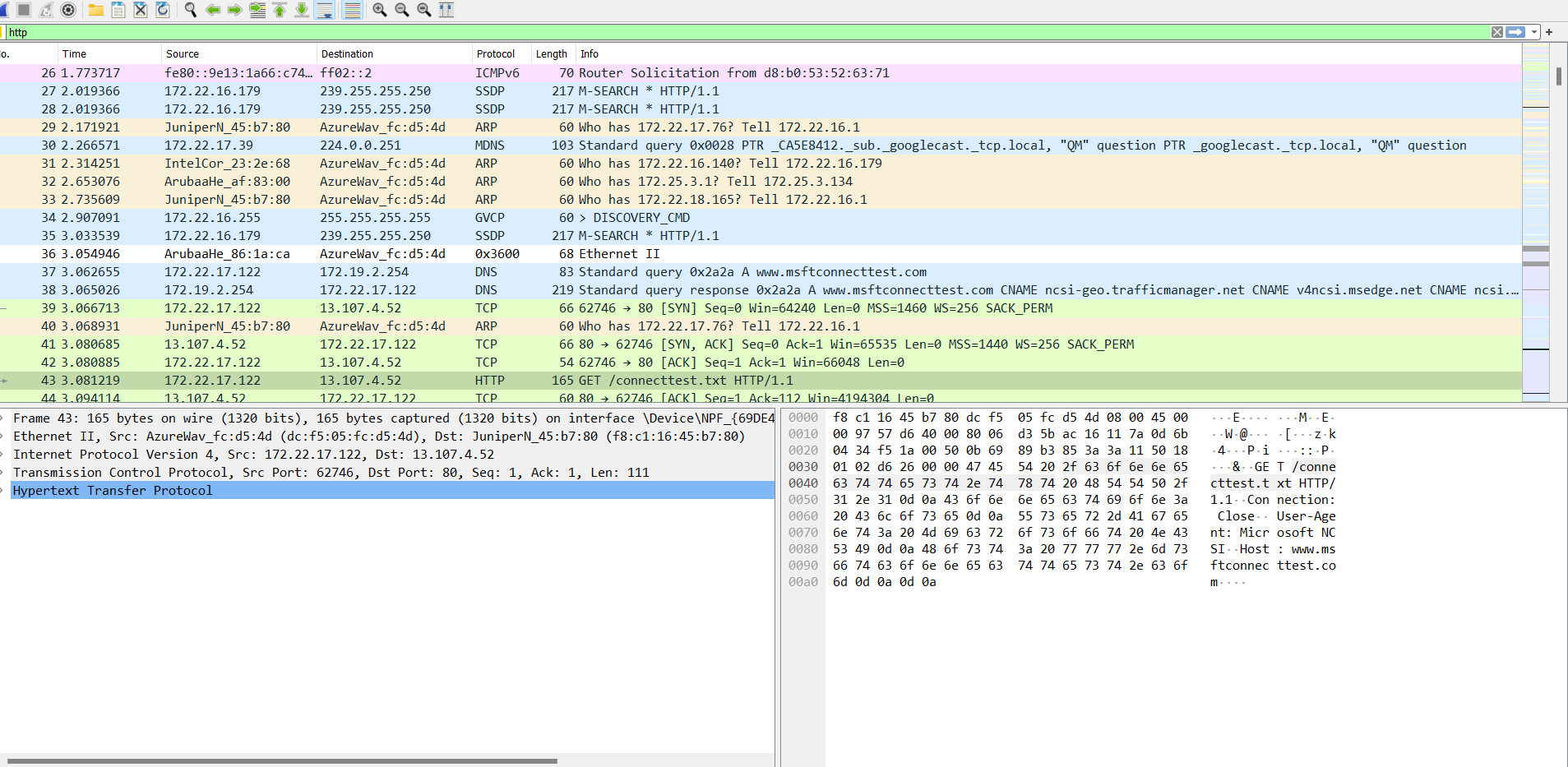
By tracing a network connection using Wireshark, you can gain insight into the behavior of the connection and identify any issues or anomalies that may be affecting the connection's performance.



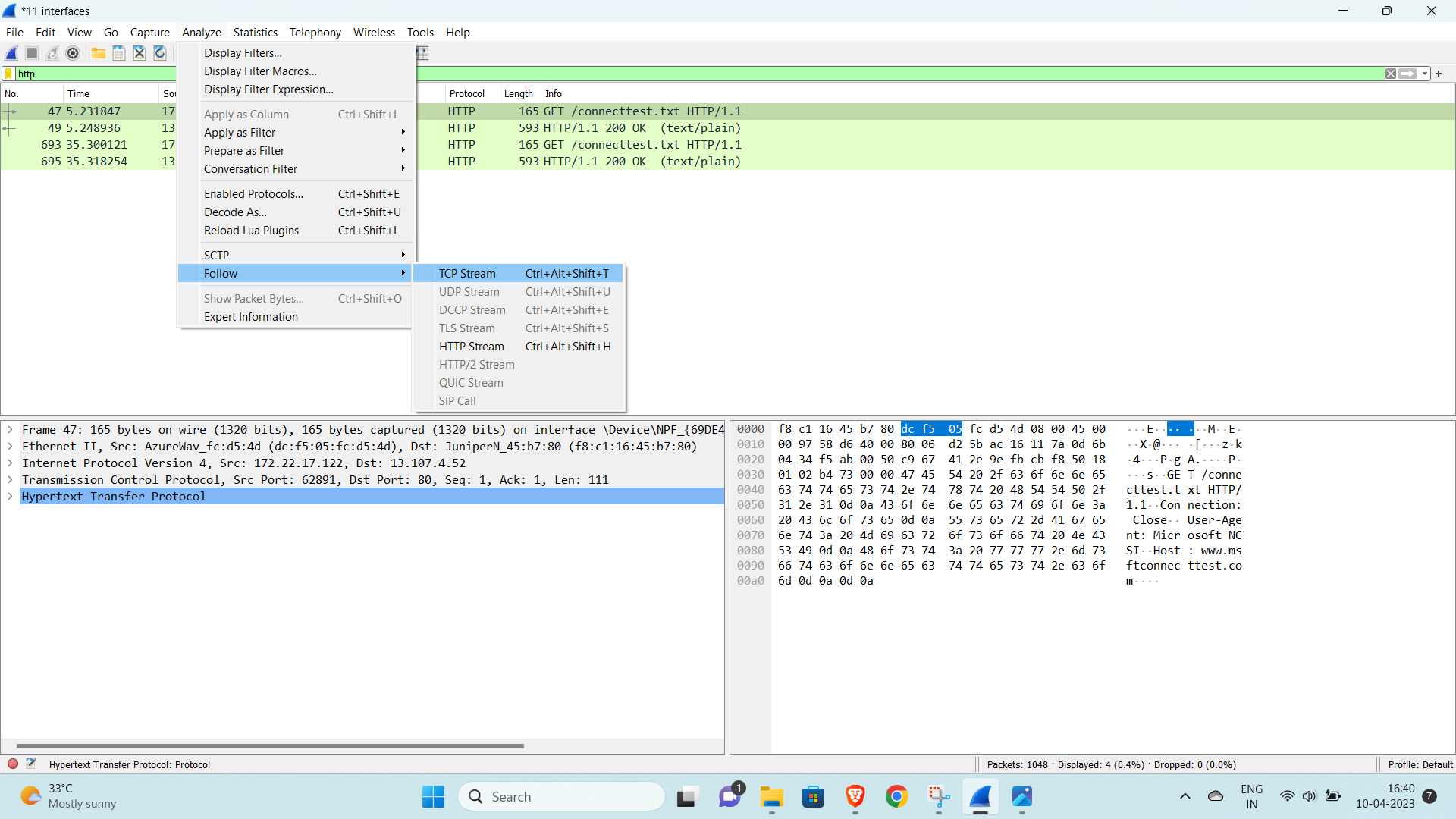
* 1. **Viewing the contents of suspect network transactions in Wireshark:**

To view the contents of suspect network transactions in Wireshark, you can follow these steps:

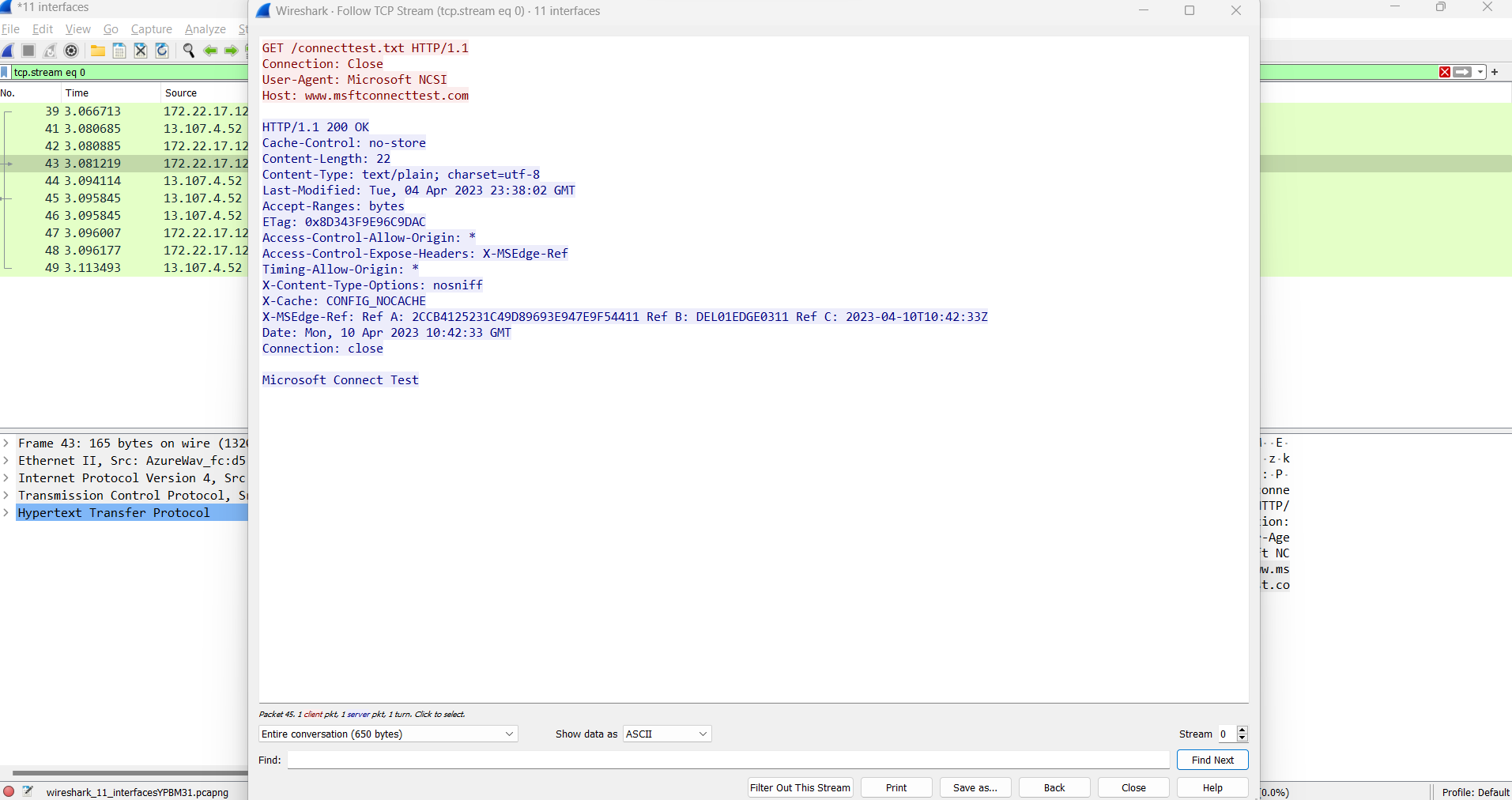
1. Start a packet capture in Wireshark by selecting the appropriate network interface and clicking on the "Capture" button.
2. Use filters to narrow down the packet capture to the suspect network transactions you're interested in. For example, you can filter by source or destination IP address, or by protocol (such as HTTP or FTP).



1. Once you've narrowed down the packet capture to the transactions you're interested in, you can view the contents of each transaction by selecting the packet in Wireshark and looking at the packet details in the bottom pane of the Wireshark window.
2. Use the "Follow TCP Stream" feature in Wireshark to view the complete contents of a TCP connection, including all packets in the connection and their contents.



It will give the output:



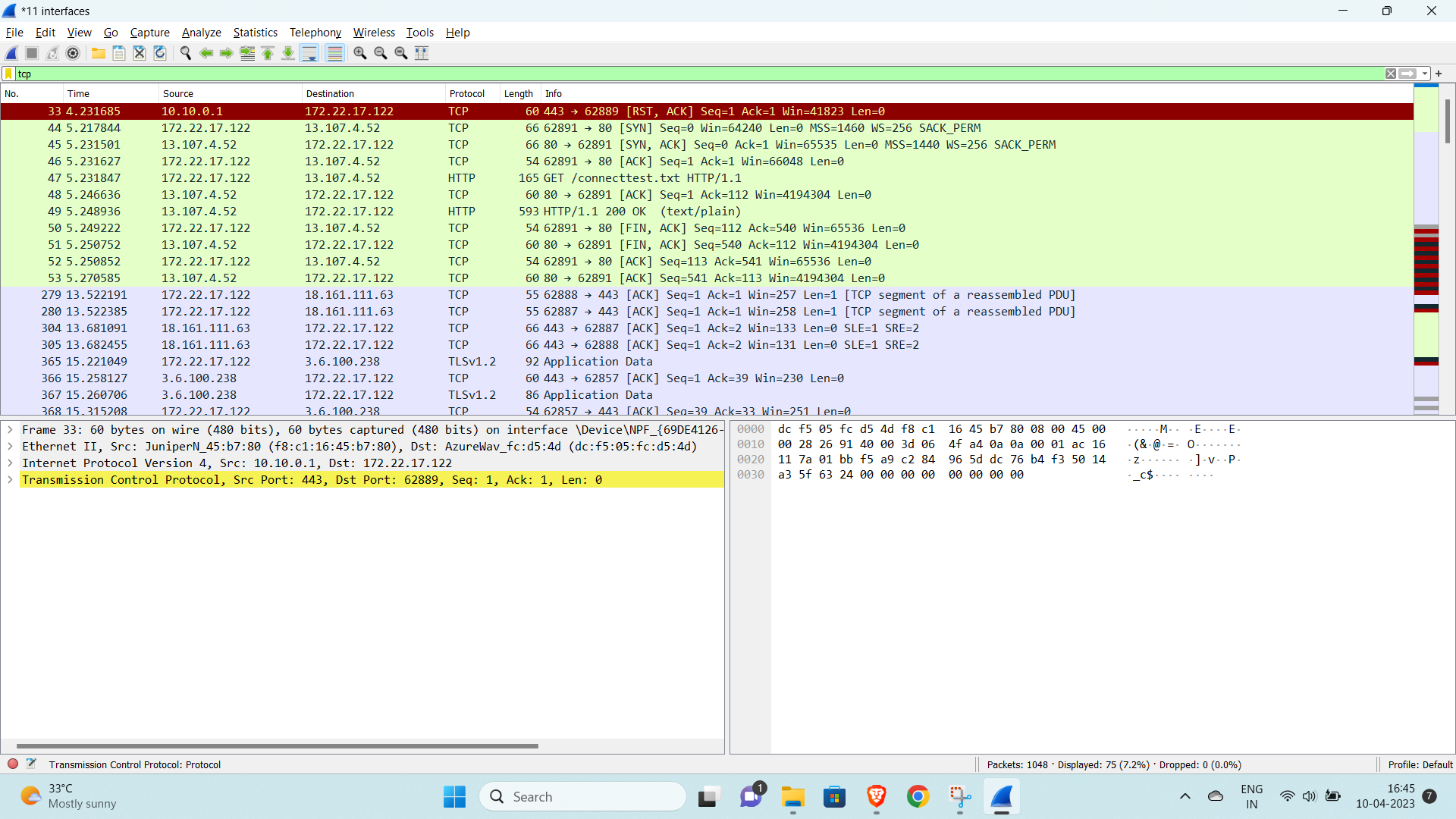
1. For HTTP transactions, Wireshark has a built-in HTTP analyzer that can display the contents of HTTP requests and responses in a human-readable format. To use the HTTP analyzer, right-click on an HTTP packet in the packet list and select "Follow > HTTP Stream".

By viewing the contents of suspect network transactions in Wireshark, you can gain insight into the behavior of the transactions and identify any issues or anomalies that may be affecting network performance or security.

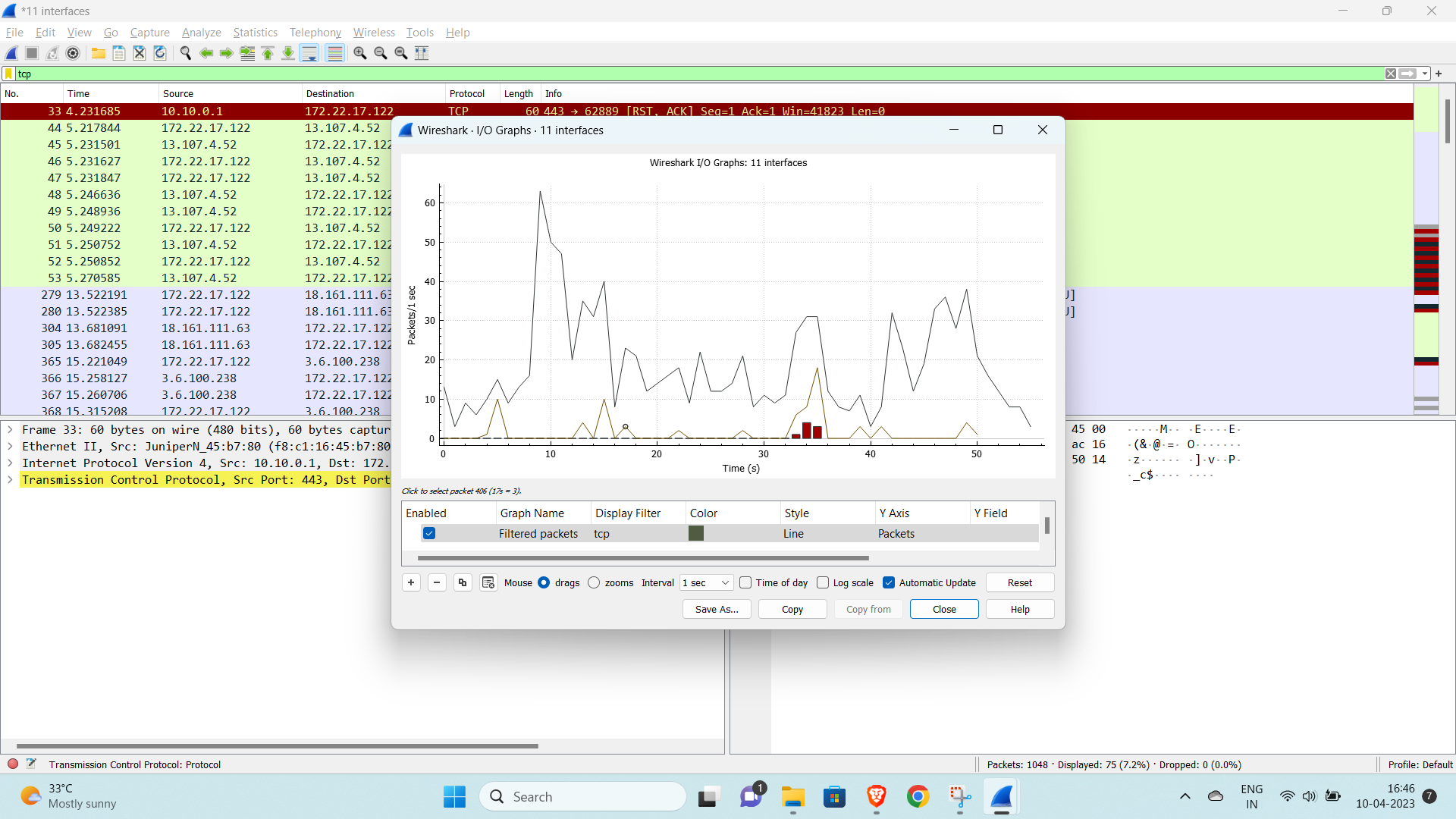
* 1. **Identifying bursts of network traffic in Wireshark:**

To identify bursts of network traffic in Wireshark, you can follow these steps:

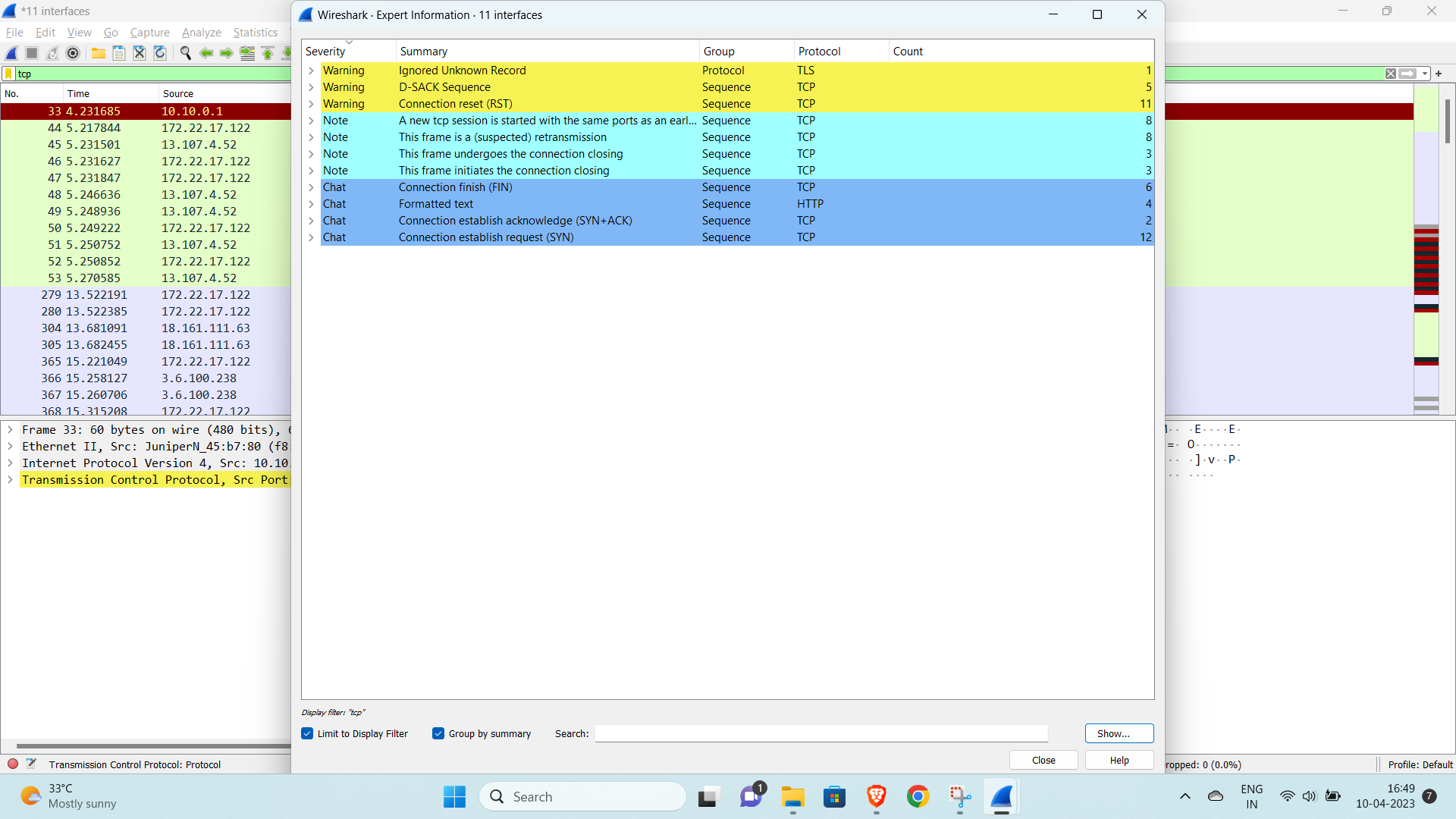
1. Start a packet capture in Wireshark by selecting the appropriate network interface and clicking on the "Capture" button.
2. Use filters to narrow down the packet capture to the traffic you're interested in analyzing. For example, you can filter by source or destination IP address, or by protocol (such as TCP or UDP).



1. Once you've narrowed down the packet capture to the traffic you're interested in, use Wireshark's "IO Graph" feature to plot the packet count or data size over time. You can access the IO Graph by selecting "Statistics" from the Wireshark menu bar and then selecting "IO Graph".
2. Use the IO Graph to visualize the network traffic over time and identify any spikes or bursts of traffic. You can adjust the time scale and graph properties to focus on specific time ranges or traffic characteristics.



1. Use Wireshark's "Expert Info" feature to identify any issues or anomalies in the traffic. Expert Info provides detailed analysis and recommendations based on Wireshark's built-in heuristics and statistical analysis.



By identifying bursts of network traffic in Wireshark, you can gain insight into the behavior of the network and identify any performance or security issues that may be affecting the network.

**IV Conclusion:**

In conclusion, the project to use open source software such as Wireshark to capture and analyze network traffic can be a valuable tool for identifying potential issues or threats in a network environment. By capturing packets from a network connection, tracing connections, viewing the contents of suspect network transactions, and identifying bursts of network traffic, the project team can gain insights into the performance and security of the network.

However, the success of the project is dependent on a number of factors, including the accuracy and reliability of the Wireshark tool, the expertise and experience of the project team, and the ability to generate accurate and actionable reports. Additionally, the project may be subject to legal or regulatory requirements related to the capture and analysis of network traffic, which must be taken into account during the project planning and execution.

Overall, the use of open source software such as Wireshark can be a valuable tool for network analysis, but it is important to carefully plan and execute the project to ensure the accuracy and completeness of the results.

**V. GitHub Link**

[**https://github.com/chiragkhemchandni/INT-301-CA-3-Report.git**](https://github.com/chiragkhemchandni/INT-301-CA-3-Report.git)

**VI . Reference/Bibliography:**

1. Varonis.com
2. Wikepedia.com
3. Wireshark.org
4. Youtube.com