

Lovely Professional University

Punjab, India

INT 301: Open-Source Technologies

Project Report

Use a Wireshark tool to analyse your network at the microscopic level and investigate at least 10 protocols, read the live data from Bluetooth and USB.

**Faculty**: Rajeshwar Sharma - 29484

**Name**: Kartikey Sahu

**Reg No**: 11915124

**Roll No**: 50

**Section**:KE002

1. **Introduction**

Bluetooth and USB devices have become ubiquitous in our daily lives, used for various purposes such as transferring files, connecting to wireless headphones, and charging our smartphones. To better understand the behaviour of these devices, it is important to investigate the various protocols involved in their communication.

In this project, we will use Wireshark, an open-source software, to read live data from Bluetooth and USB devices and investigate 10 protocols. By reading live data, we can gain insights into how Bluetooth and USB communication works, as well as the various protocols involved in their communication. The project aims to gain a deeper understanding of Bluetooth and USB devices, as well as the protocols involved in their communication.

The report will describe the methodology used to capture and analyze Bluetooth and USB traffic, as well as the protocols investigated using Wireshark.

* 1. **Objective**

The objective of this project is to use Wireshark to analyze a computer network at the microscopic level and investigate at least 10 protocols. In addition to analyzing network traffic, the project will also involve reading live data from Bluetooth and USB devices. The project aims to gain insights into how network protocols work, identify potential security risks, and improve the performance of the network. The report will describe the methodology used to capture and analyze network traffic, the tools and equipment used, and the results obtained from the analysis. Additionally, the report will discuss the limitations of the project and future work that could be done to improve or expand upon the results. By gaining a deeper understanding of network protocols and USB/Bluetooth communication, we can better secure and optimize computer networks for a variety of applications.

* 1. **Description**

The project aims to gain a better understanding of Bluetooth and USB devices, as well as the various protocols involved in their communication. By using Wireshark, an open-source software, we will read live data from Bluetooth and USB devices and investigate 10 different communication protocols.

The project will start with an overview of Bluetooth and USB devices, their common use cases, and how they work. We will then describe the methodology used to capture and read live data from these devices, including the equipment and tools used.

Using Wireshark, we will investigate 10 different protocols involved in Bluetooth and USB communication, such as Bluetooth Low Energy (BLE), USB 3.0, and Wireless USB.

* 1. **Scope**

The scope of this project is to investigate the protocols involved in Bluetooth and USB communication using Wireshark. We will focus on reading live data from Bluetooth and USB devices and analysing 10 different communication protocols.

1. **System Description**

The system used for this project includes a computer running the Wireshark software and a network interface card (NIC) connected to the network being analysed. Wireshark is an open-source software tool that allows for the capturing and analysis of network traffic. The NIC used for capturing traffic should be capable of promiscuous mode, which allows it to capture all traffic on the network, even if it's not intended for the host computer.

Additionally, for investigating Bluetooth traffic, a Bluetooth adapter is required to capture Bluetooth packets. The adapter should be compatible with the Bluetooth version being used and should support the Bluetooth protocol analyser in Wireshark.

For investigating USB traffic, a USB protocol analyser should be used. The protocol analyser can be a hardware device or a software tool that captures USB packets and sends them to Wireshark for analysis. The USB device being analysed should be connected to the analyser using a USB cable. Thus, the tool that I used is USBPcap which worked as an extension to the Wireshark.

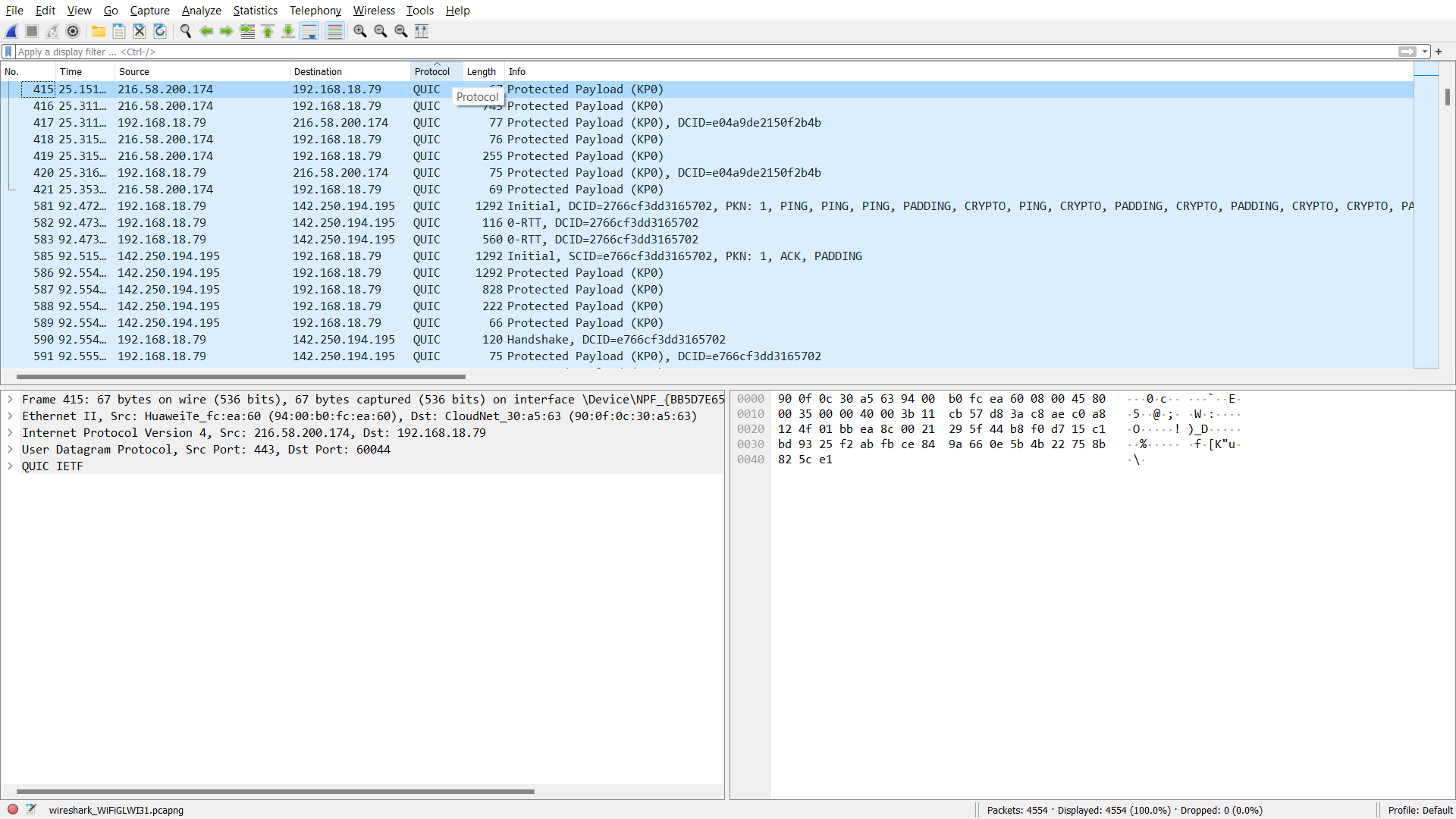
The system requirements for running Wireshark are minimal, with the software being compatible with various operating systems, including Windows, which is the default OS my laptop have.

1. **Analysis Report**

**Investigate 10 protocols using Wireshark**

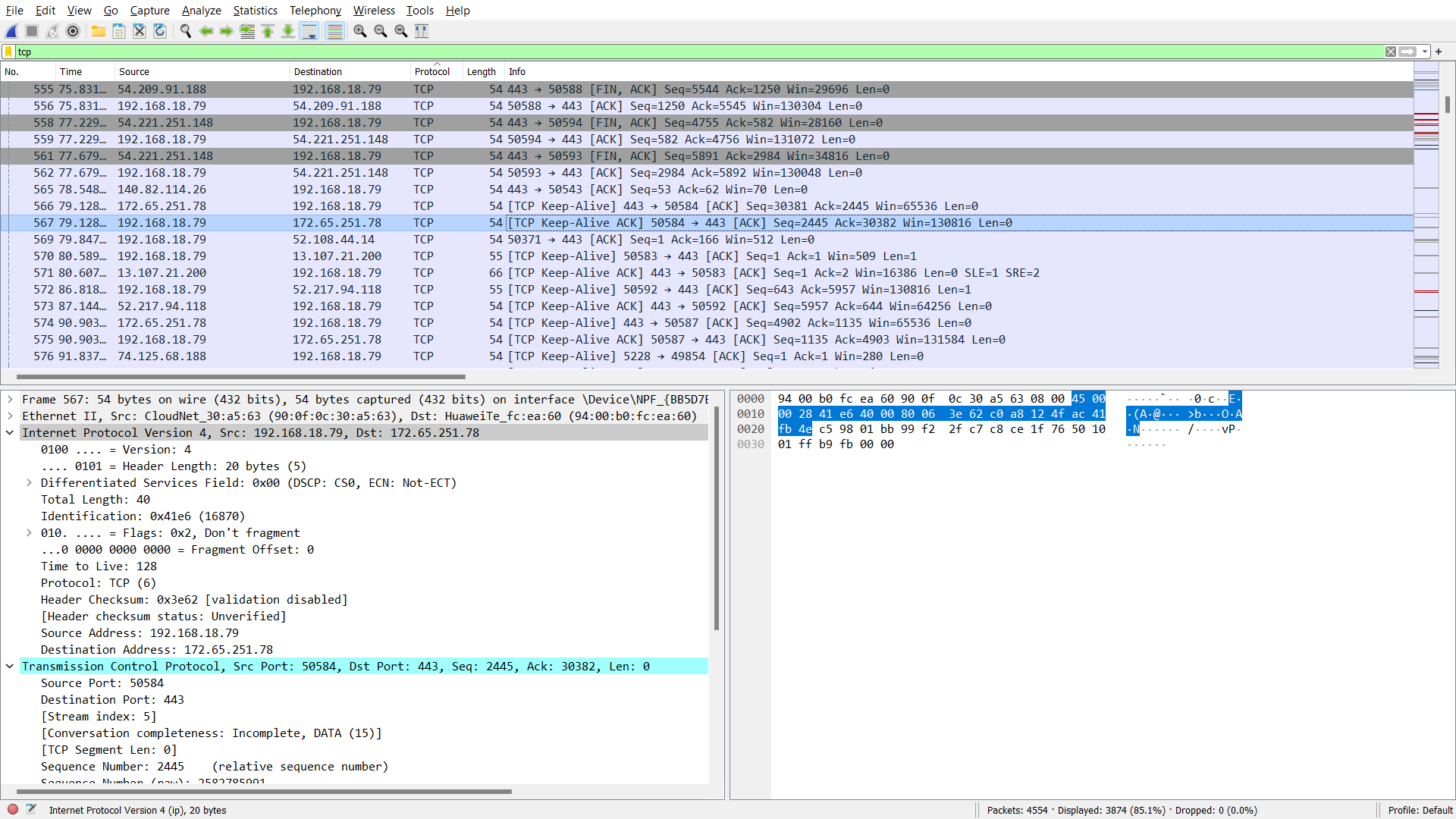
To investigate 10 protocols using Wireshark, we can follow these steps:

1. Launch Wireshark and select the network interface we want to capture traffic on. In my case, since I’m connected to wi-fi, I’ll select the same.
2. Double click on wi-fi to begin the packet capture.



**Figure 1**: Packets captured on wi-fi

1. In figure 1, I’ve sorted the packets via protocol, we can also filter them using the text box which is given just above the titles of the columns.
2. We could Analyse the packets for each protocol by examining the details of the protocol fields in each packet.



**Figure 2**: Packet Info

1. Selecting a particular packet will show us all the details related to that packet, which include headers, payload (if any) and more.
2. We could also look for patterns or anomalies in the traffic for each protocol. For example, we might look for unusually large or small packet sizes, high error rates, or unexpected behaviour.

**Read data from Bluetooth**

This section involves capturing and analysing live Bluetooth data using open-source software tool, Wireshark. The purpose of this is to gain insights into Bluetooth communication protocols and improve our understanding of how Bluetooth works. This section will describe the tools and equipment used for reading live data from Bluetooth, the methodology for capturing Bluetooth data.

1. **Introduction**

Bluetooth is a wireless communication technology that was first introduced in 1994. It uses short-range radio waves to connect devices within a range of about 10 meters. Bluetooth devices communicate using a variety of protocols, including the Bluetooth Core Specification, which defines the basic architecture and functionality of Bluetooth devices, and various application-specific profiles, which define how different types of devices communicate and what types of data can be exchanged. Bluetooth uses frequency-hopping spread spectrum (FHSS) to avoid interference from other wireless devices and ensure reliable communication.

Bluetooth is a wireless communication technology that is widely used in various devices, such as smartphones, laptops, and smartwatches. The purpose of this project is to capture and analyse live Bluetooth data using open-source software tools. By doing so, we hope to gain insights into Bluetooth communication protocols and improve our understanding of how Bluetooth works.

1. **Tool used**

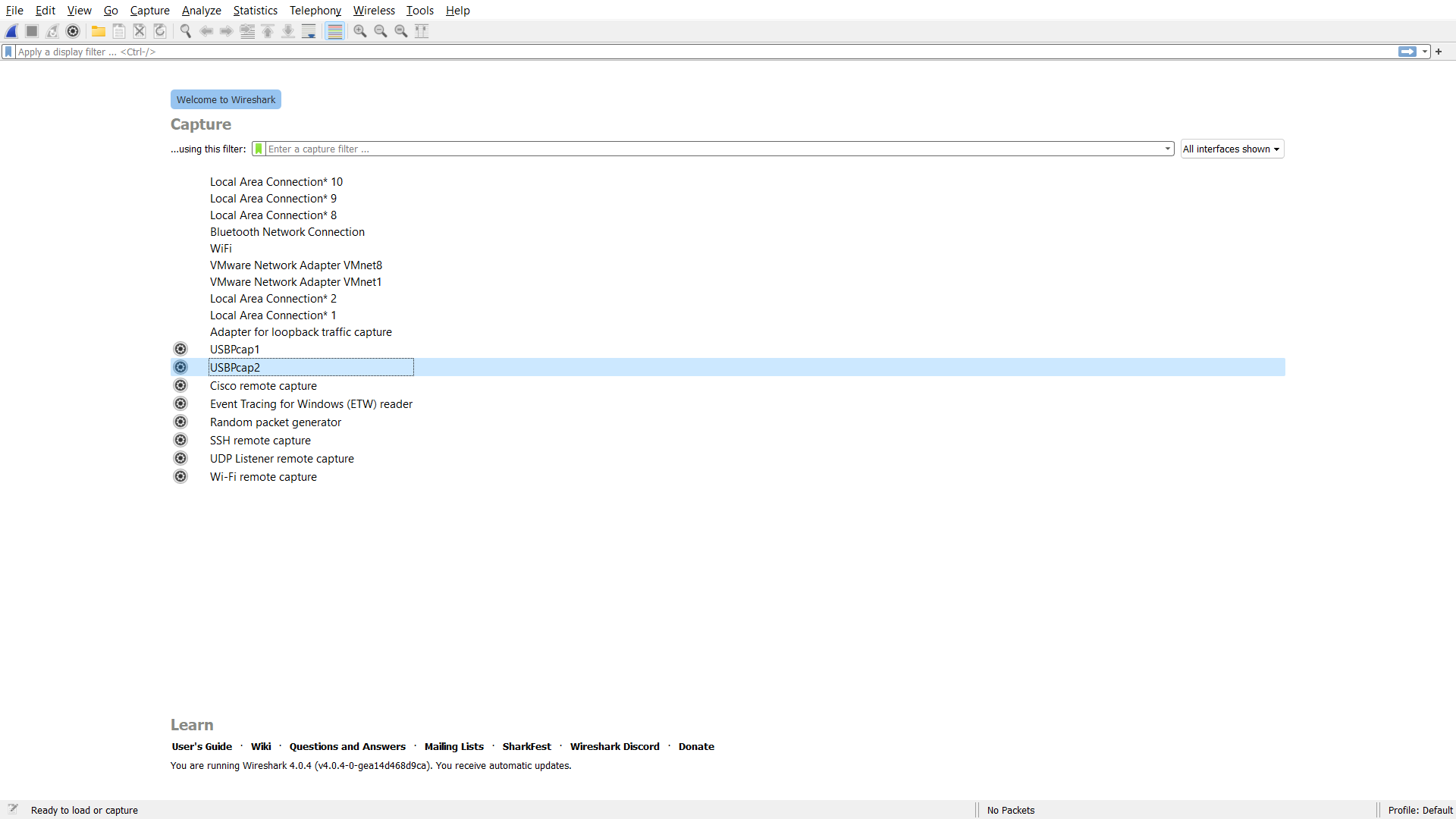
For this, I used a Bluetooth adapter that is compatible with the open-source software tool, Wireshark. Wireshark is a network protocol analyser that allows us to capture and analyse live Bluetooth data. We also used a laptop computer with Bluetooth capability**.**

1. **Methodology**

To capture live Bluetooth data, I first installed Wireshark on my laptop computer and connected a Bluetooth device, my phone, to my laptop. I then used Wireshark to capture Bluetooth traffic on the appropriate interface. I initiated Bluetooth communication between two devices and monitored the data exchange using Wireshark. I faced some challenges in setting up the capture interface, like missing driver and not knowing which interface refers to the Bluetooth connection, but I was able to overcome them by consulting the Wireshark documentation and online forums.

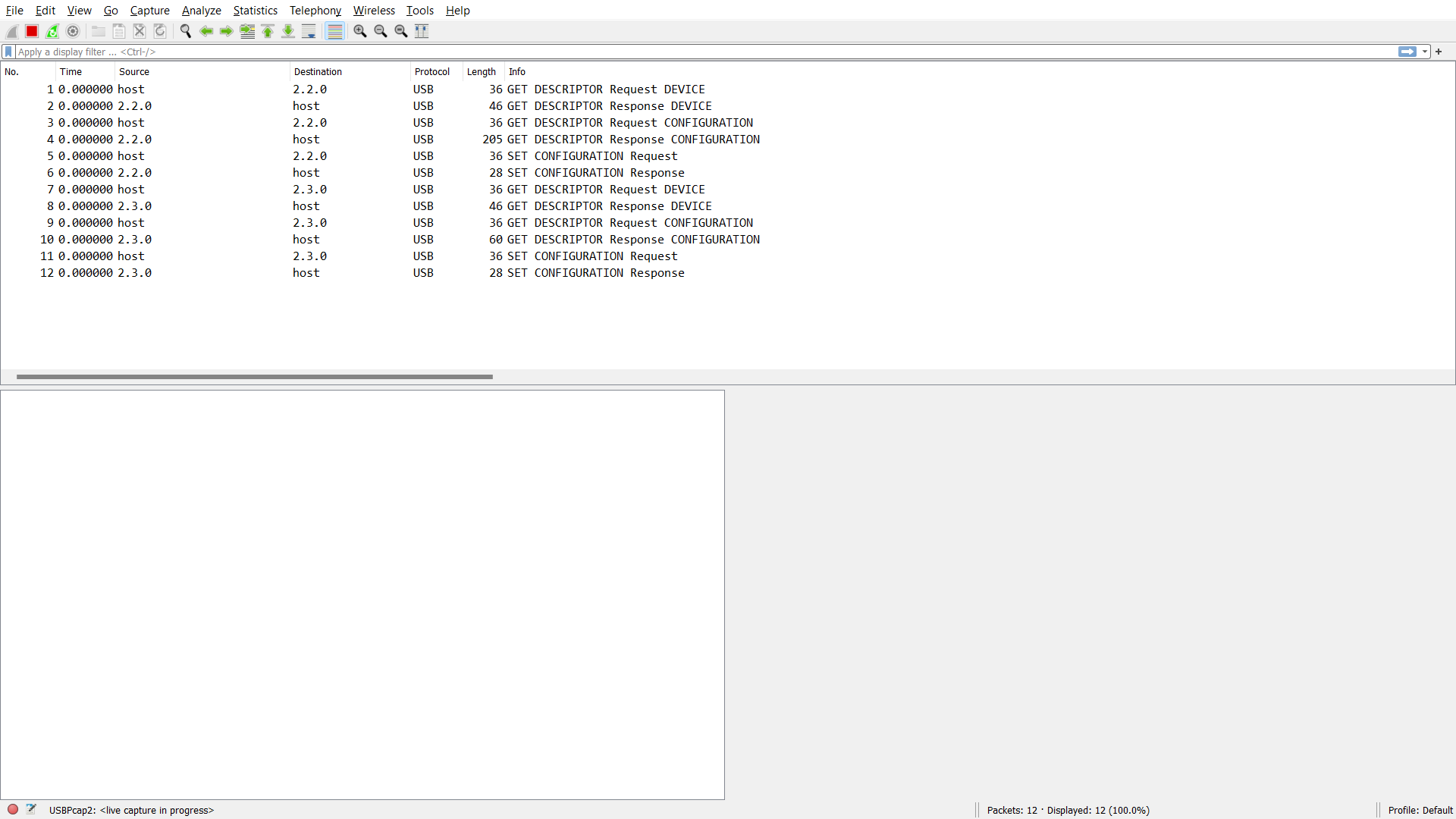
Step-wise approach to read live data from Bluetooth is as follows:

1. Connect my phone with my laptop through Bluetooth.
2. The USBPcap2 interface in the **Figure 3** is the interface that we could use to read the Bluetooth traffic.



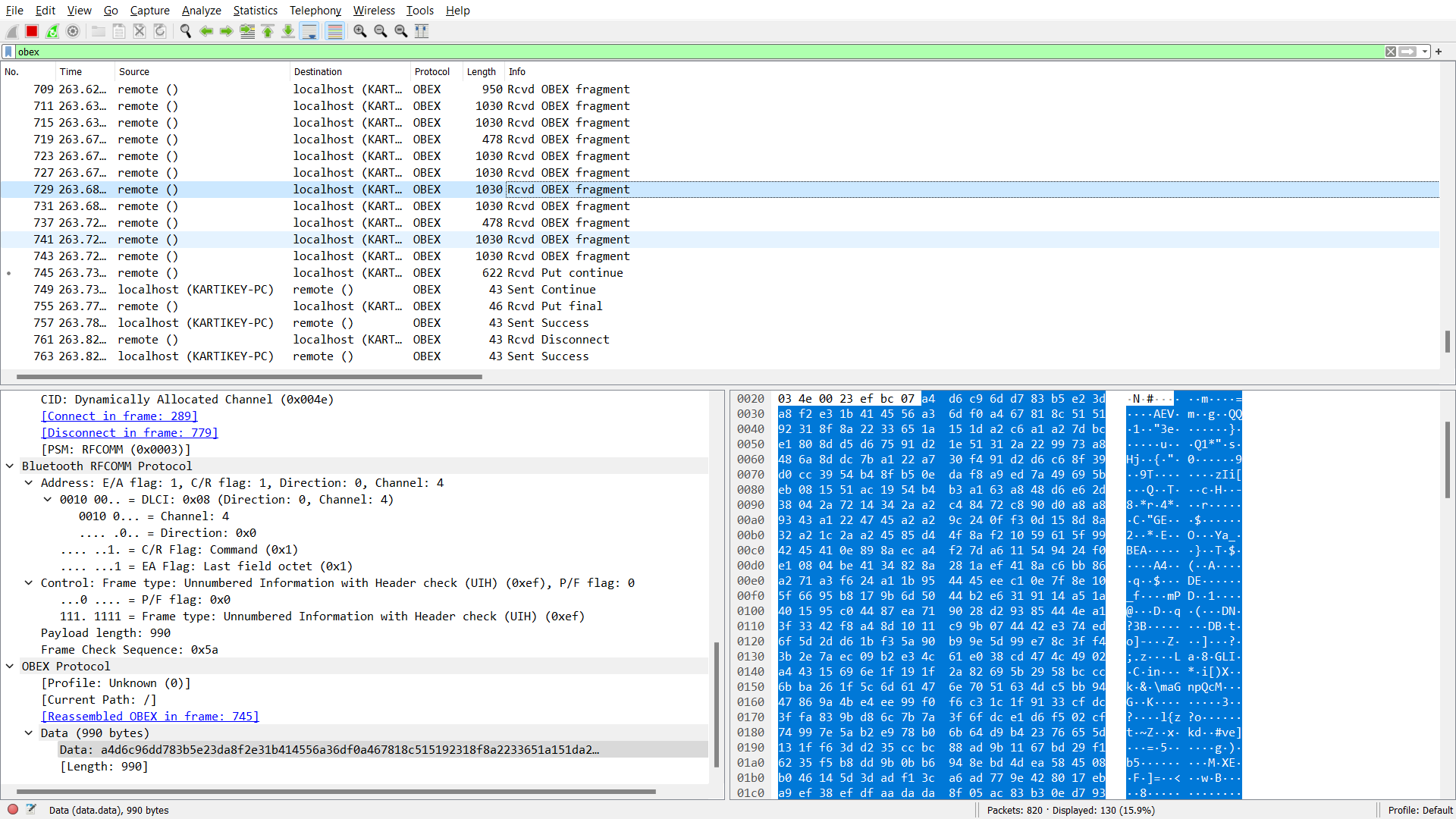
**Figure 3**: Main screen of Wireshark displaying the interfaces where we can capture packets.

1. Double click on USBPcap2 and we can see the live capturing of packets on Bluetooth. **Figure #** shows some packets which are not captured from Bluetooth, because I opened this interface after plugging in a USB pen drive otherwise it wouldn’t have opened, at least for me.



**Figure 4**: Packets captured before any data transfer is initiated through Bluetooth.

1. As soon as I initiated a connection i.e., transferring files between my phone and my laptop, we can see the packets captured as the files are transferred. I’m filtering the packets by OBEX protocol, which is a communications protocol that facilitates binary transfers between Bluetooth enabled devices. Last few packets with info “Sent Success” represents that the data was transferred successfully.



**Figure 5**: Packets filtered by OBEX protocol indicating data transfer between Bluetooth-enabled devices.

The data highlighted in blue on right-bottom panel is the data that was transferred in the selected packet.

**Read data from USB**

This section involves capturing and analysing live data from USB devices using the open-source software tool Wireshark. The purpose of the project is to gain insights into how USB devices communicate with the host system, and to improve our understanding of USB communication protocols. This section will describe the tools and equipment used for the project, the methodology for capturing and analysing USB data.

1. **Introduction**

Universal Serial Bus (USB) is a widely used technology for connecting devices to computers and other host systems. USB devices are used for a wide range of purposes, including data storage, communication, and peripheral input/output. To gain insights into how USB devices communicate with host systems, it is important to capture and analyse live USB data. This project aims to use the open-source software tool Wireshark to capture and read live data from USB devices, with the objective of improving our understanding of USB communication protocols and device performance.

1. **Tool Used**

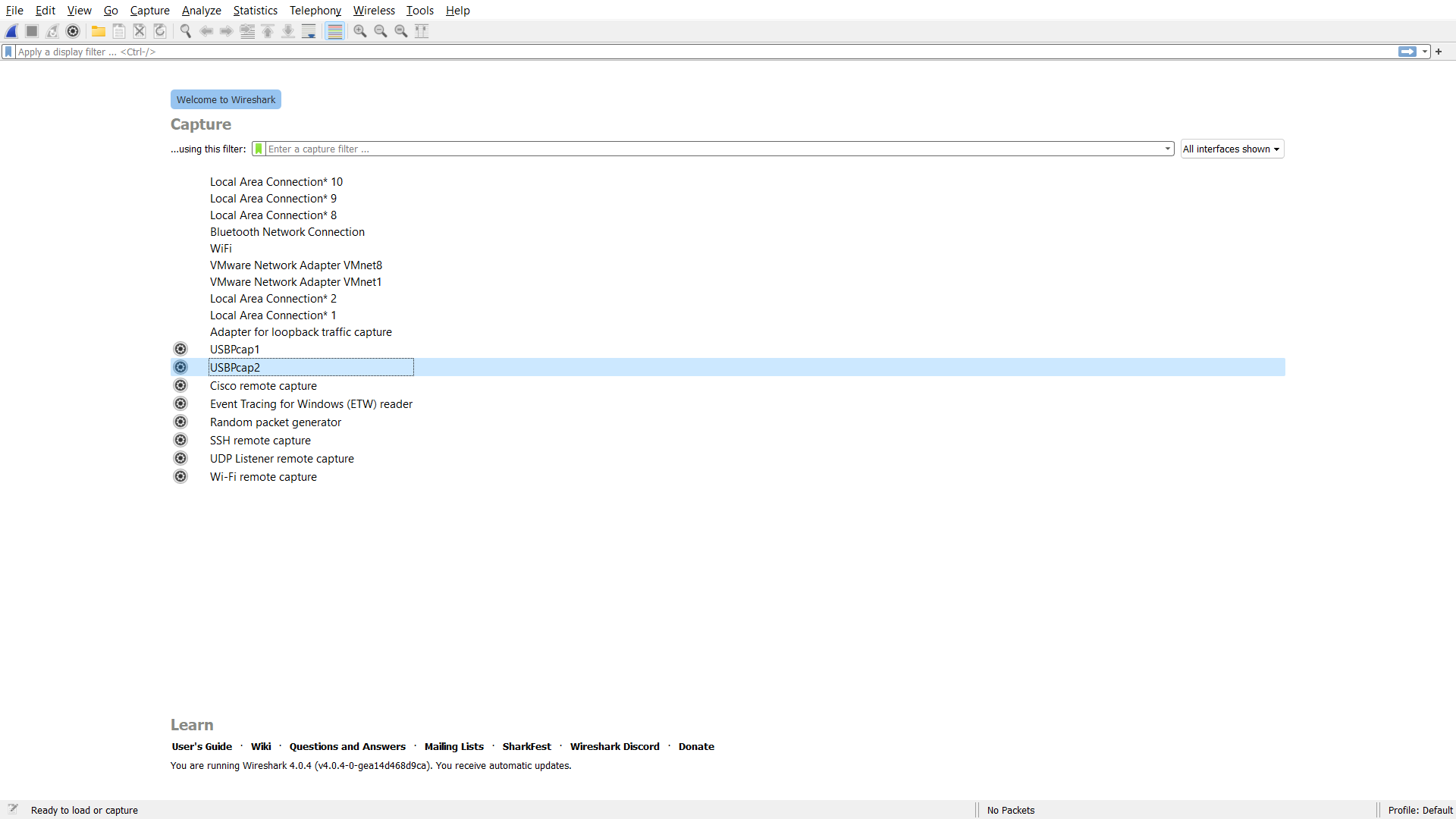
For this, I used a pen drive to transfer data to my computer and capture the packets while it is being transferred and the open-source software tool, Wireshark. Since, it can capture the USB traffic too along with Bluetooth and other network interfaces.

1. **Methodology**

Similar to what I did while reading live data from Bluetooth, to capture live USB data, I first installed Wireshark on my laptop computer and connected a USB pen drive to my laptop. I then used Wireshark to capture USB traffic on the appropriate interface. I initiated communication between the pen drive and my computer; and monitored the data exchange using Wireshark.

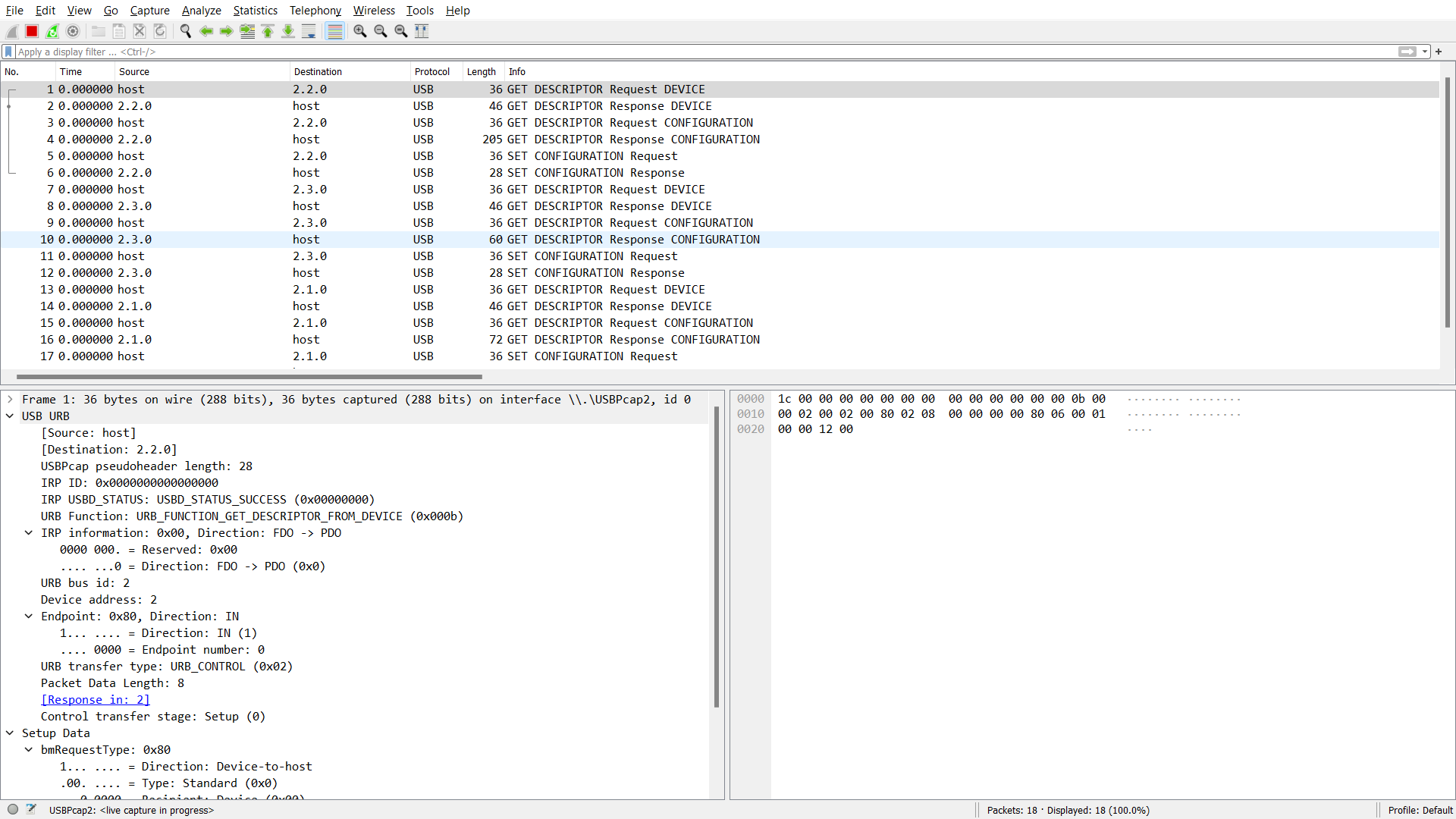
Step-wise approach to read live data from USB is as follows:

1. Connect a pen drive to my system.
2. Just like Bluetooth, the USBPcap2 interface in the **Figure 6** is the interface that could be used to read the USB traffic.



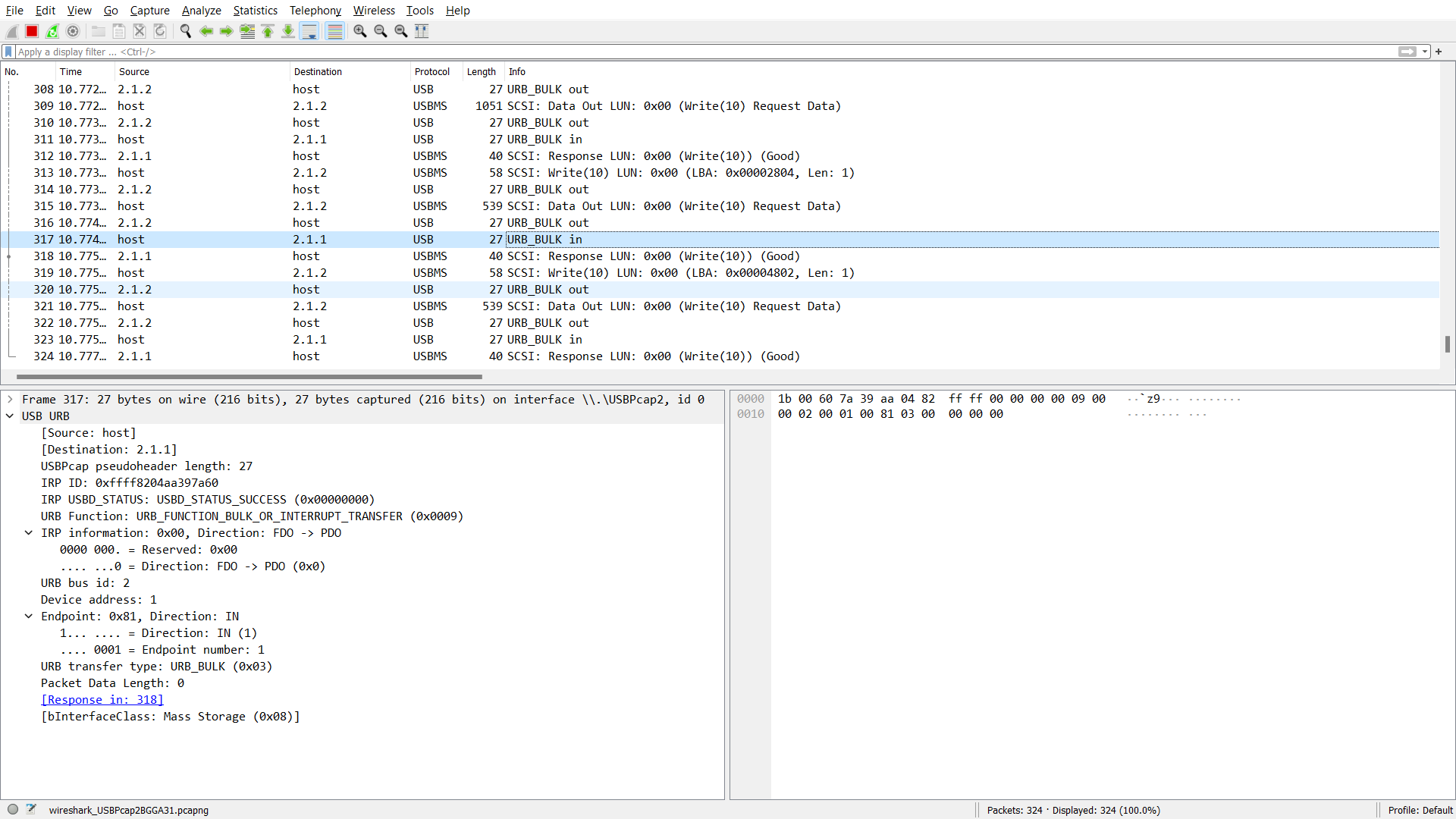
**Figure 6**: Main screen of Wireshark displaying the interfaces where we can capture packets.

1. Double click on USBPcap2 and we can see the live capturing of packets on USB.



**Figure 7**: Packets captured before any data transfer is initiated through USB.

1. As soon as I initiated a connection i.e., transferring files between my phone and my laptop, we can see the packets captured as the files are transferred.



**Figure 8**: Packets which carried data on transferring files to USB

1. **References**
2. Bluetooth Technology Website. (2021). Retrieved from https://www.bluetooth.com/
3. USB Implementers Forum Website. (2021). Retrieved from https://www.usb.org/
4. Wireshark Website. (2021). Retrieved from https://www.wireshark.org/
5. Bluetooth Low Energy (BLE) Overview. (2021). Retrieved from https://www.bluetooth.com/bluetooth-resources/bluetooth-technology-basics/bluetooth-low-energy/
6. USB 3.0 Specification. (2021). Retrieved from https://www.usb.org/document-library/usb-30-specification
7. Bluetooth Core Specification Version 5.2. (2020). Retrieved from https://www.bluetooth.com/specifications/bluetooth-core-specification/
8. USB Type-C Specification. (2021). Retrieved from https://www.usb.org/document-library/usb-type-c-specification-revision-20
9. Bluetooth Low Energy: A Primer. (2017). Retrieved from https://www.nxp.com/docs/en/white-paper/BLUETOOTH-LOW-ENERGY-A-PRIMER.pdf
10. USB 2.0 Specification. (2021). Retrieved from https://www.usb.org/document-library/usb-20-specification
11. Bluetooth Mesh Networking: An Introduction. (2018). Retrieved from https://www.bluetooth.com/bluetooth-resources/bluetooth-mesh-networking-an-introduction/
12. USB Power Delivery Specification. (2021). Retrieved from https://www.usb.org/document-library/usb-power-delivery
13. Bluetooth 5: What's New and What's Next? (2017). Retrieved from https://www.eetimes.com/bluetooth-5-whats-new-and-whats-next/
14. USB Audio Device Class Specification. (2021). Retrieved from https://www.usb.org/document-library/audio-devices-20
15. Bluetooth Location Services. (2018). Retrieved from https://www.bluetooth.com/bluetooth-resources/bluetooth-technology-basics/bluetooth-location-services/
16. USB Mass Storage Class Specification. (2021). Retrieved from https://www.usb.org/document-library/mass-storage-class-10-specification.

**Github Link:**

https://github.com/heykartikey/INT301-Report