LAB Manual

PART A

(PART A : TO BE REFFERED BY STUDENTS)

**Experiment No. 5**

**A.1 Aim:**

To install network sniffer (Wireshark) , analyze the working of packet sniffing and study other features of Wireshark

**A.2 Prerequisite:**

Role of packet sniffers in network security

**A.3 Outcome:**

**After successful completion of this experiment students will be able to** 1. Appreciate wireshark as a tool to analyze the packets travelling in a network.

2. Know how this tool can be used by malicious intruders to capture and analyze network traffic.

**A.4 Theory:**

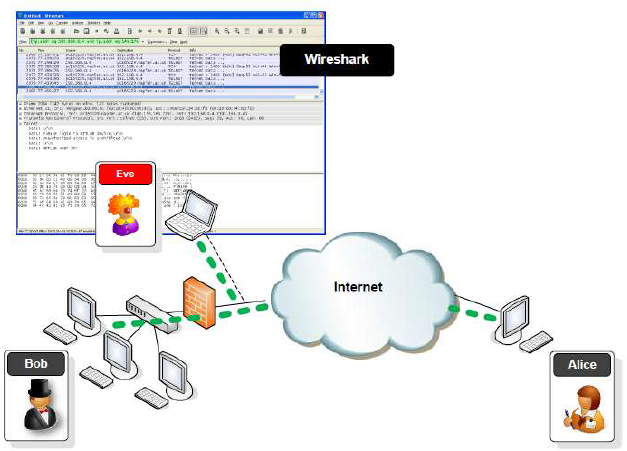
A packet sniffer, sometimes referred to as a network monitor or network analyzer, can be used by a network or system administrator to monitor and troubleshoot network traffic. Using the information captured by the packet sniffer an administrator can identify erroneous packets and use the data to pinpoint bottlenecks and help maintain efficient network data transmission. In its simple form a packet sniffer simply captures all of the packets of data that pass through a given network interface. By placing a packet sniffer on a network in promiscuous mode, a malicious intruder can capture and analyze all of the network traffic.

Wireshark is a network packet analyzer. A network packet analyzer will try to capture

network packets and tries to display that packet data as detailed as possible.

**Packet Capture (Packet Sniffing)**

A **packet sniffer** is an application which can capture and analyse network traffic which is passing through a system’s Network Interface Card (NIC). The sniffer sets the card to **promiscuous mode** which means all traffic is read, whether it is addressed to that machine or not. The figure below shows an attacker sniffing packets from the network, and the **Wireshark** packet sniffer/analyser (formerly known as ethereal).

****

**Packet Analysis**

Wireshark is an open source cross-platform packet capture and analysis tool, with versions for

Windows and Linux. The GUI window gives a detailed breakdown of the network protocol stack

for each packet, colorising packet details based on protocol, as well as having functionality to

filter and search the traffic, and pick out TCP streams. Wireshark can also save packet data to

files for offline analysis and export/import packet captures to/from other tools. Statistics can also

be generated for packet capture files.

Wireshark can be used for **network troubleshooting**, to **investigate security issues**, and to

**analyse and understand network protocols**. The packet sniffer can exploit information passed

in plaintext, i.e. not encrypted. Examples of **protocols** which pass information in plaintext are

**Telnet, FTP, SNMP, POP, and HTTP**.

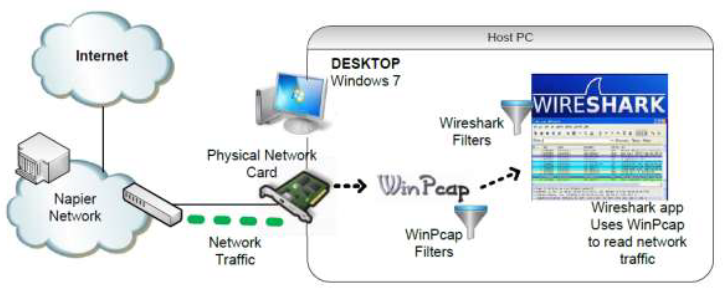
Wireshark is a GUI based network capture tool. There is a command line based version of the

packet capture utility, called **TShark**. TShark provides many of the same features as it’s big

brother, but is console-based. It can be a good alternative if only command line access is

available, and also uses less resources as it has no GUI to generate.

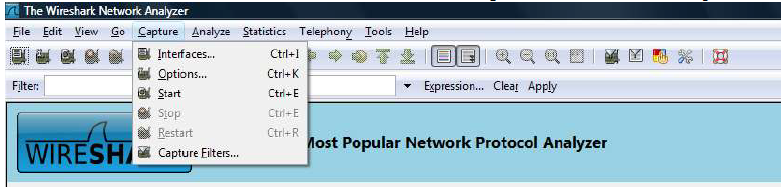
**Using Wireshark to Capture Traffic**

****

***Select a Network Interface to Capture Packets through.***

Start the Wireshark application. When Wireshark is first run, a default, or blank window is

shown. To list the available network interfaces, select the **Capture->Interfaces** menu option.

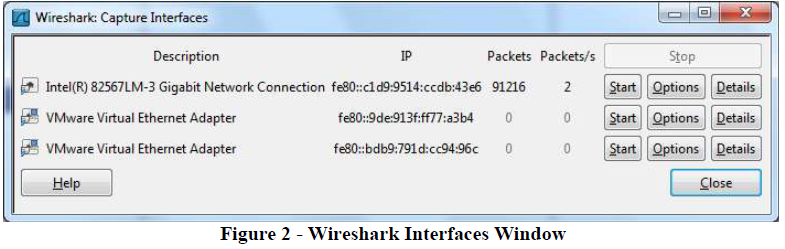


Wireshark should display a popup window such as the one shown in Figure 2. To capture

network traffic click the **Start** button for the network interface you want to capture traffic on.

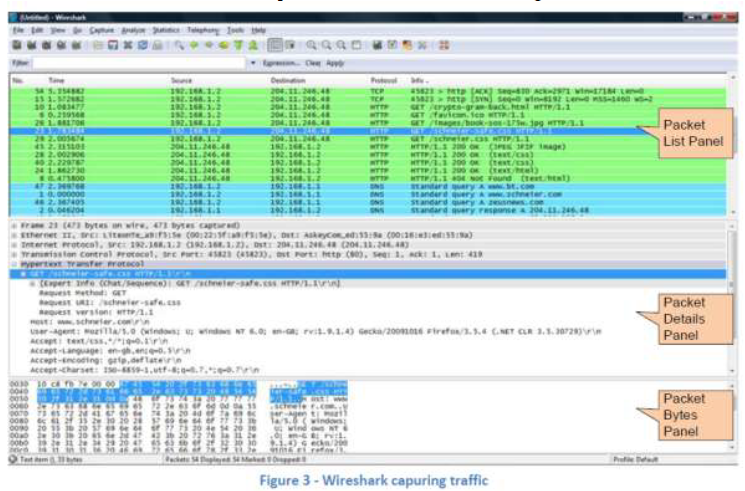
Windows can have a long list of virtual interfaces, before the Ethernet Network Interface Card

(NIC).



Generate some network traffic with a Web Browser, such as Internet Explorer or Chrome. Your

Wireshark window should show the packets, and now look something like.



To stop the capture, select the **Capture->Stop** menu option, Ctrl+E, or the Stop toolbar button.

What you have created is a Packet Capture or **„*pcap’***, which you can now view and analyse using

the Wireshark interface, or save to disk to analyse later.

The capture is split into 3 parts:

1. **Packet List Panel –** this is a list of packets in the current capture. It colours the packets based

on the protocol type. When a packet is selected, the details are shown in the two panels below.

2. **Packet Details Panel –** this shows the details of the selected packet. It shows the different

protocols making up the layers of data for this packet. Layers include Frame, Ethernet, IP,

TCP/UDP/ICMP, and application protocols such as HTTP.

3. **Packet Bytes Panel –** shows the packet bytes in Hex and ASCII encodings.

To select more detailed options when starting a capture, select the **Capture->Options** menu

option, or **Ctrl+K**, or the Capture Options button on the toolbar (the wrench). This should show a window such as shown in Figure 4.

Some of the more interesting options are:

 ***Capture Options > Interface* -** Again the important thing is to select the correct Network

Interface to capture traffic through.

 ***Capture Options > Capture File* –** useful to save a file of the packet capture in real time, in

case of a system crash.

 ***Display Options > Update list of packets in real time* –** A display option, which should be

checked if you want to view the capture as it happens (typically switched off to capture straight to a file, for later analysis).

 ***Name Resolution > MAC name resolution –*** resolves the first 3 bytes of the MAC Address,

the Organisation Unique Identifier (OUI), which represents the Manufacturer of the Card.

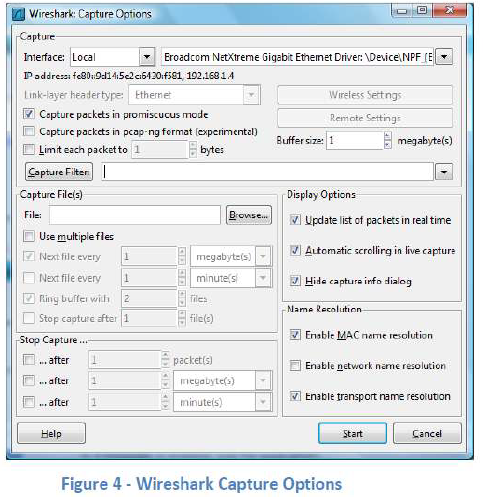
 ***Name Resolution > Network name resolution –*** does a DNS lookup for the IP Addresses

captured, to display the network name. Set to off by default, so covert scans do not generate this

DNS traffic, and tip off who’s packets you are sniffing.

Make sure the **MAC name resolution** is selected. Start the capture, and generate some Web

traffic again, then stop the capture.



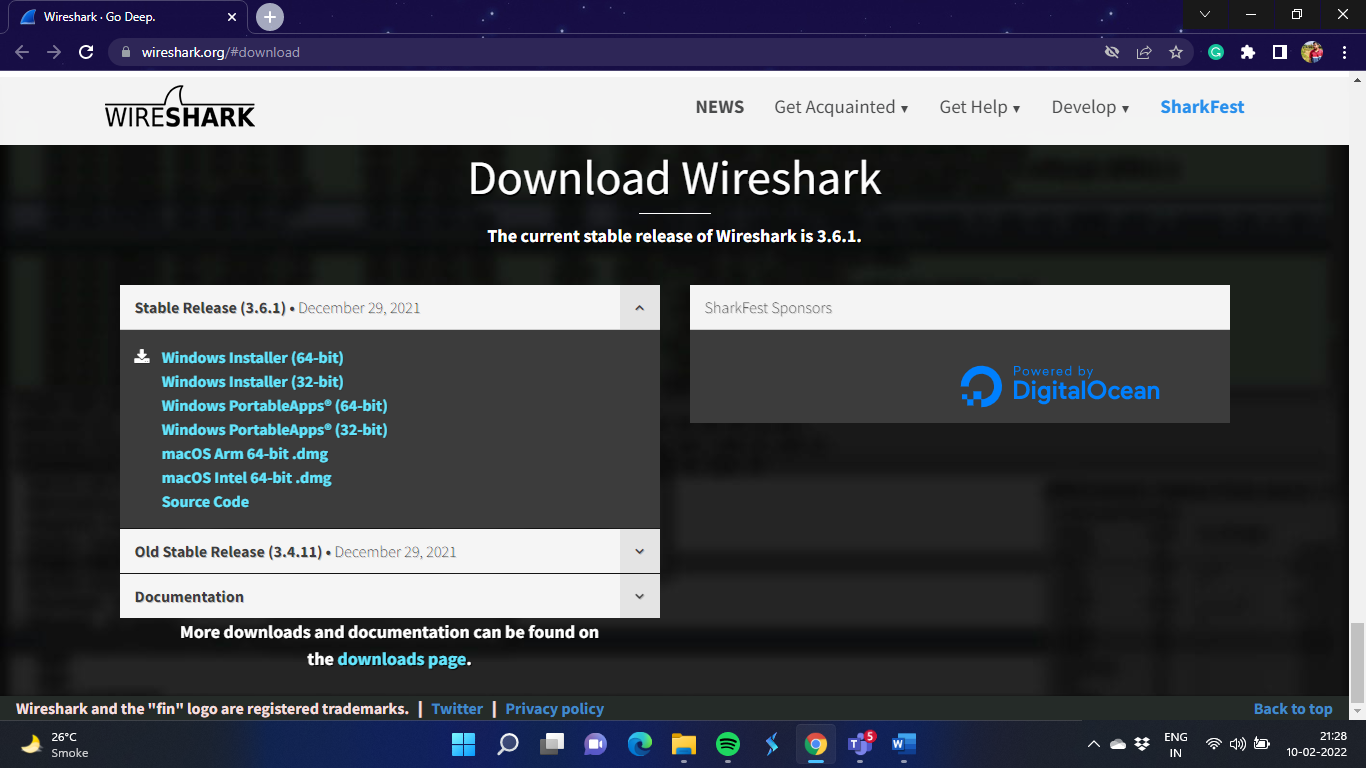
PART B

(PART B: TO BE COMPLETED BY STUDENTS)

***(Students must submit the soft copy as per following segments within two hours of the practical. The soft copy must be uploaded on the Blackboard or emailed to the concerned lab in charge faculties at the end of the practical in case the there is no Black board access available)***

|  |  |
| --- | --- |
| Roll. No. A16 | Name: Varun Khadayate |
| Class B.Tech CsBs | Batch: 1 |
| Date of Experiment: 04-02-2022 | Date of Submission:10-02-2022 |
| Grade: | |

**B.1 Software installation issues faced:**



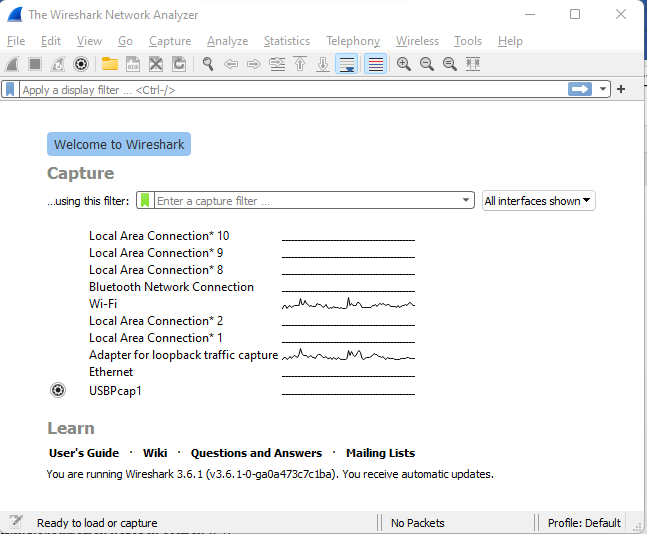
1. Download Wireshark Application from the link: <https://www.wireshark.org/#download>
2. Once the download is complete, open the installer and install the application on the Device.
3. Note: Once installed you’ll have to restart your device to make Wireshark work efficiently.
4. Some devices have trouble even after restarting so for that, run Wireshark as administrator.

**B.2 Input and Output:**

***(Paste your program input and output in following format, If there is error then paste the specific error in the output part. In case of error with due permission of the faculty extension can be given to submit the error free code with output in due course of time. Students will be graded accordingly.)***

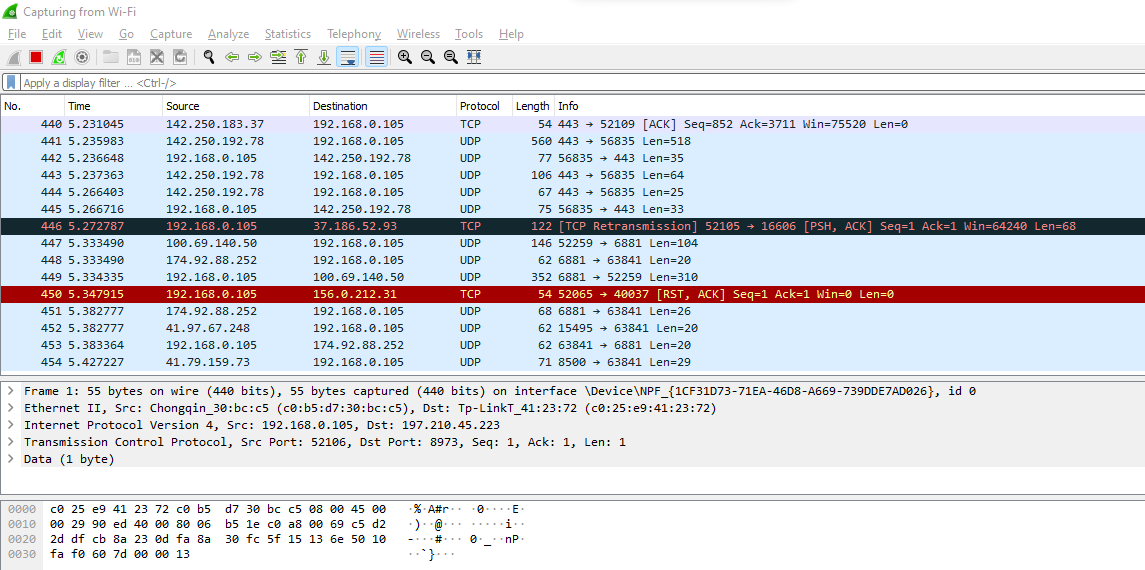
**Input:**

Below shows the wireshark screen when nothing is running except the **Wifi** and **Adapter for Loopback Traffic capture** in the device.

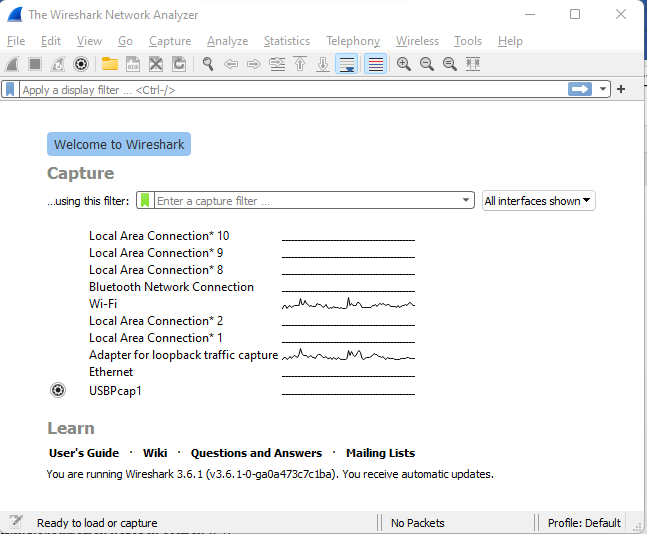
****

**Output:**

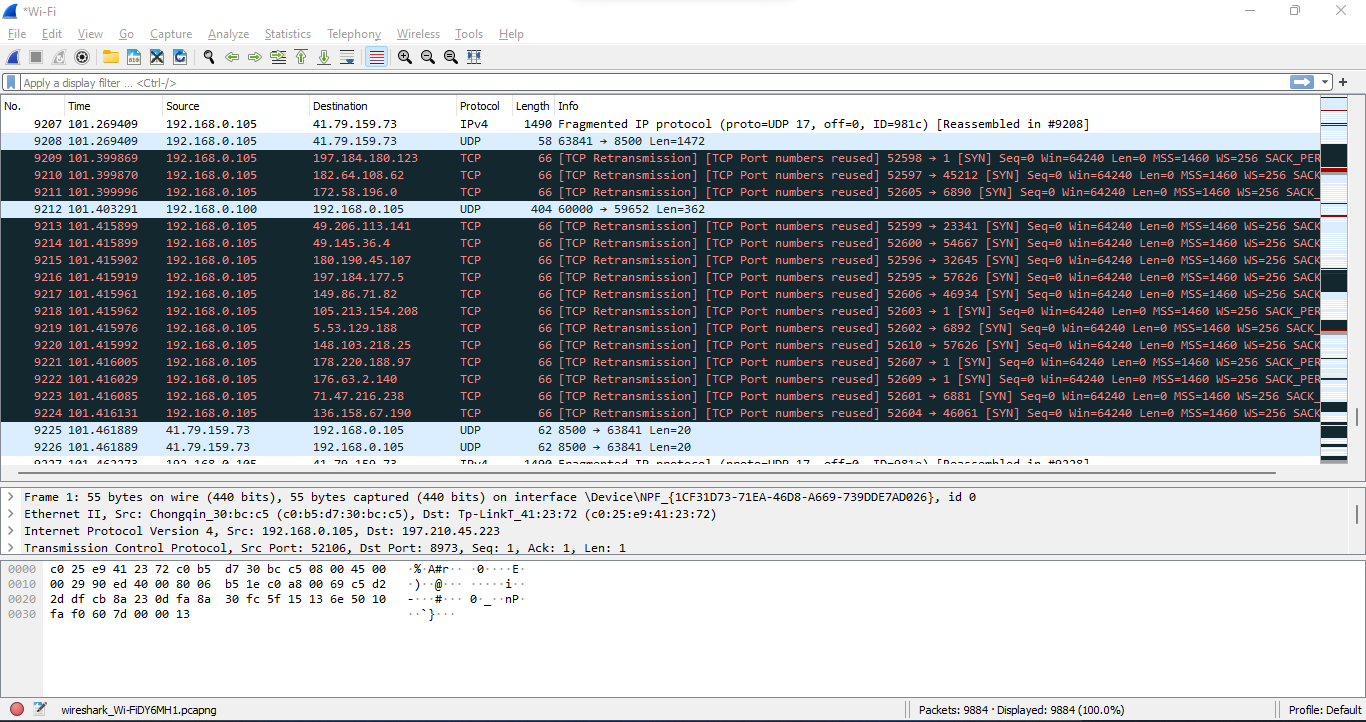
## Below is the IP Addresses, Hex codes and their corresponding ASCII Codes and some frame information about the packet that are parsing from WiFi.



## If you start your Google Chrome Browser and open your GMAIL Account, the Wireshark tool shows some data.

****

## This also shows the IP Adresses, the hexa-decimal code for my username as well.



**B.3 Observations and learning:**

***(Students are expected to comment on the output obtained with clear observations and learning for each task/ sub part assigned)***

From the above experiment, we can observe that device’s WiFi passes some default packets to the device even when there is nothing else using the WiFi. When we open GMAIL via Google Chrome Browser, we see some waves of network in Traffic Capture

**B.4 Conclusion:**

*(****Students must write the conclusion as per the attainment of individual outcome listed above and learning/observation noted in section B.3)***

We were able to observe the working of Wireshark Tool when there is only wifi working and one when we use the browser, successfully.

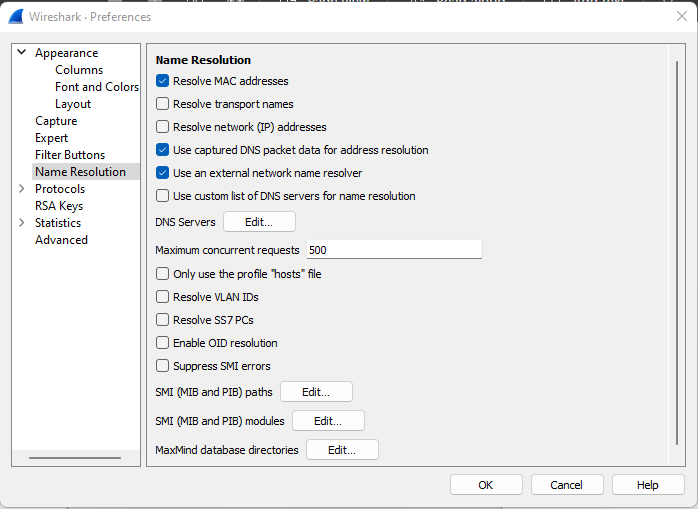
# Questions of Curiosity

***(To be answered by student based on the practical performed and learning/observations)***

## Q1: Give the uses of wireshark tool

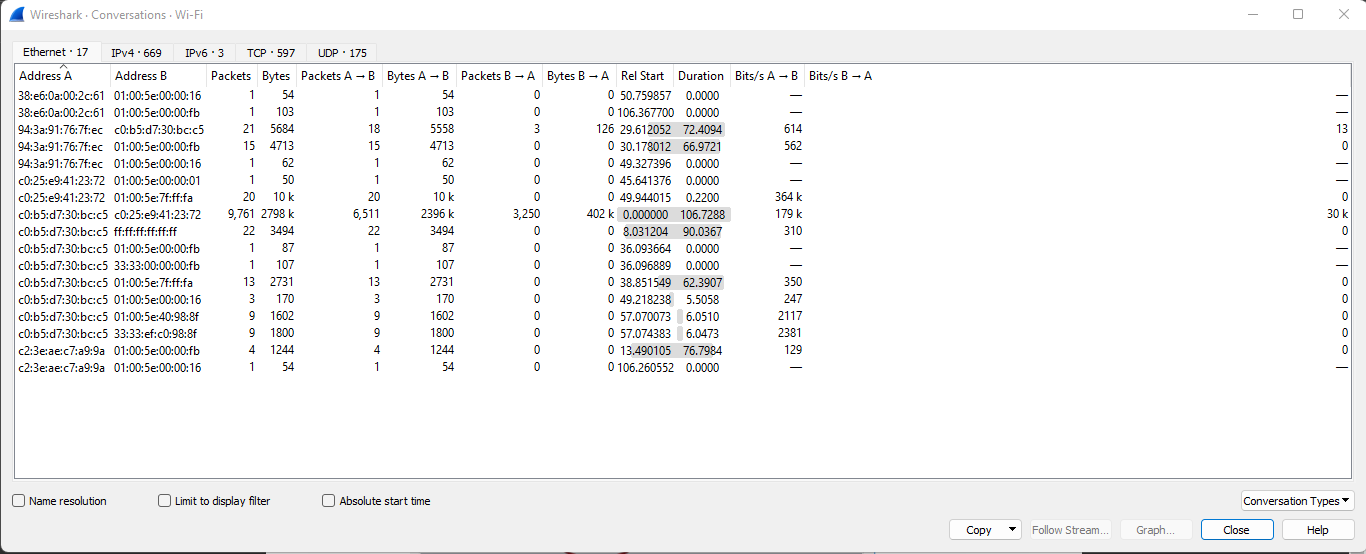
### Baselining your traffic:

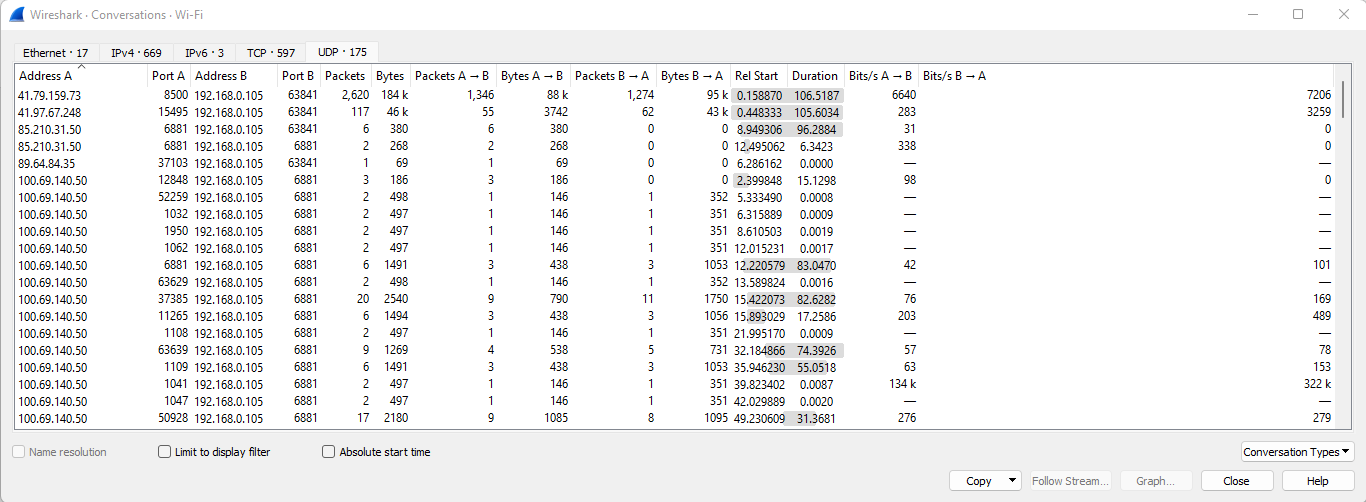
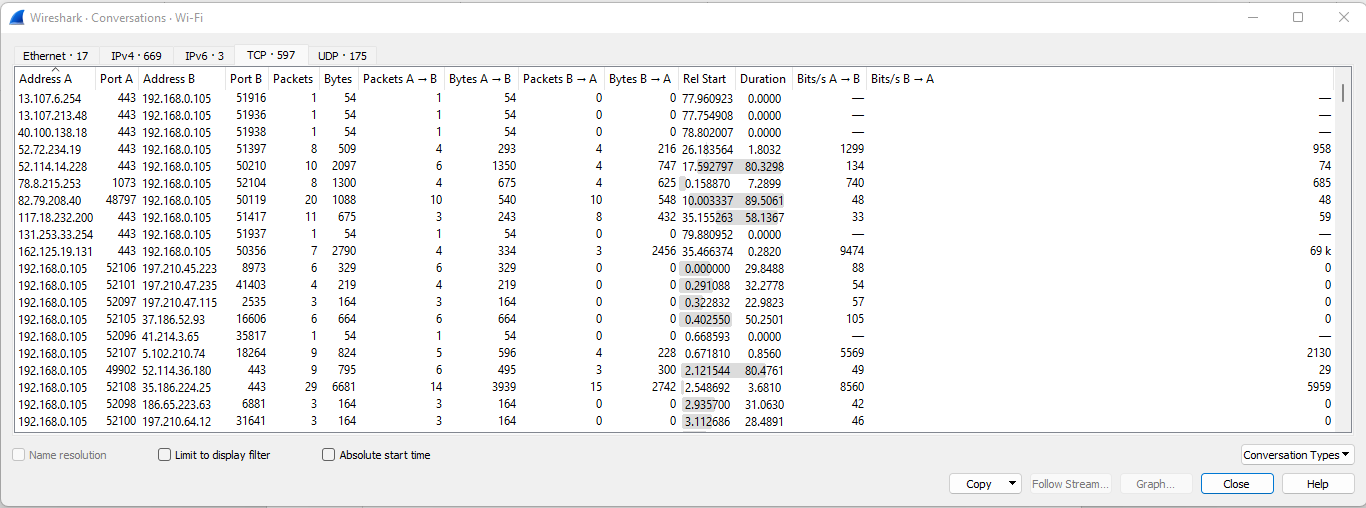
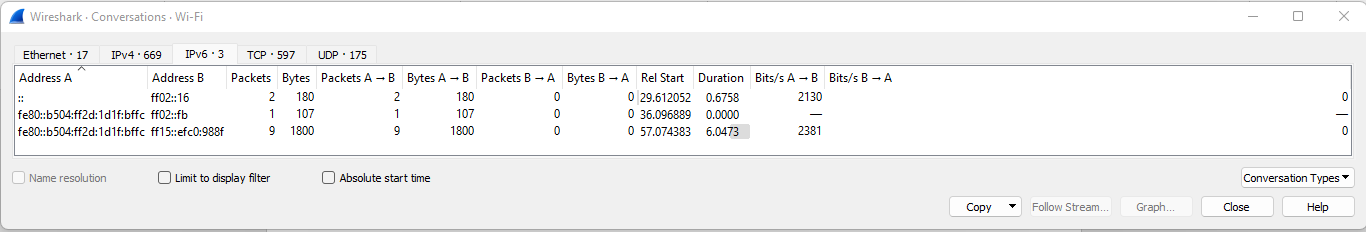
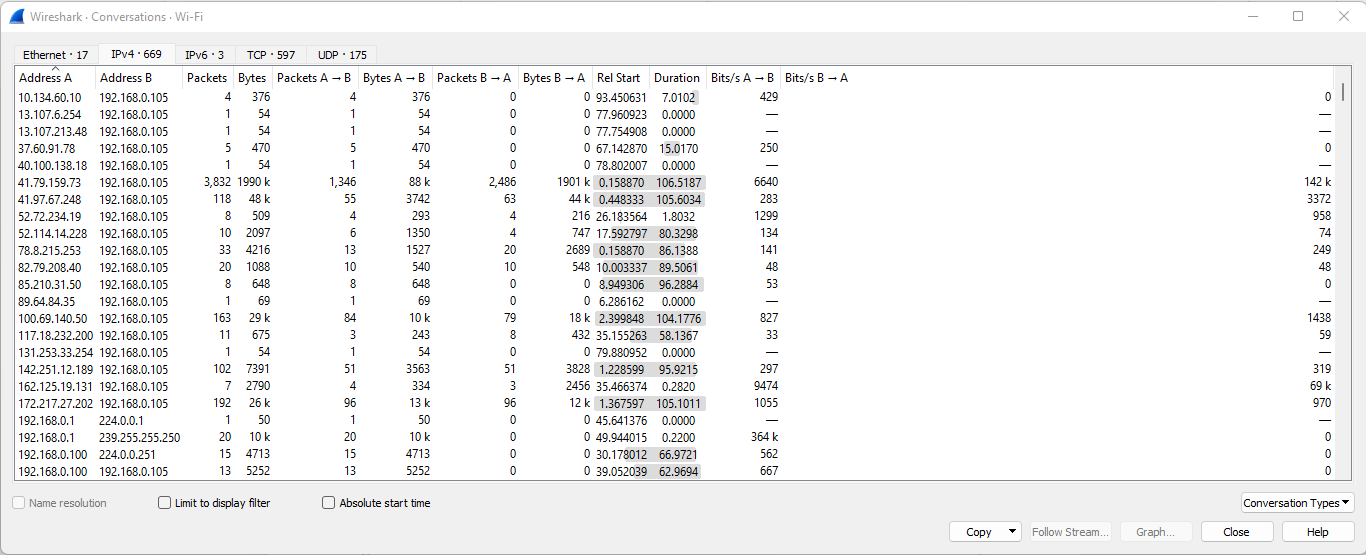
Baselining is the process of capturing and identifying the “normal” traffic on a network. This traffic may include the auto-update applications on a network, a myriad of broadcast and multicast traffic streams, auto-detect applications scrounging around the network unnecessarily, and more.



### Perform Passive Discovery:

Passive discovery is the process of building a map of the network based on what you hear while listening to the traffic. From the Statistics menu, Wireshark can provide a list of visible hosts, conversations (pairs of hosts communicating with each other), resolved addresses, port numbers, and more.





### Detect Unsecured Application

### Detect Suspicious Protocols and Application

## Q2: List some other packet sniffing tools.

### SolarWinds Deep Packet Inspection and Analysis Tool

Gives detailed insights into what causes network slowness and uses deep packet inspection to allow you to resolve the root causes. You can identify traffic by application, category and risk level to eliminate and filter problem traffic. With a great user interface, this excellent packet sniffing software is perfect for network analysis.

### ManageEngine NetFlow Analyzer

A traffic analysis tool that works with NetFlow, J-Flow, sFlow Netstream, IPFIX, and AppFlow

Paessler Packet Capture Tool A packet sniffer, a NetFlow sensor, an sFlow sensor, and a J-Flow sensor built into Paessler PRTG.

### Omnipeek Network Protocol Analyzer

A network monitor that can be extended to capture packets.

### tcpdump

The essential free packet capture tool that every network manager needs in his toolkit.

### tshark

A lightweight answer to those who want the functionality of Wireshark, but the slim profile of tcpdump.

### NetworkMiner

A Windows-based network analyzer with a no-frills free version.

### Fiddler

A packet capture tool that focuses on HTTP traffic.

### Capsa

Written for Windows, the free packet capture tool can be upgraded for payment to add on analytical features.