MAWLANA BHASHANI SCIENCE AND TECHNOLOGY UNIVERSITY

LAB-REPORT

Report No : 04

Course Code : ICT-4202

Course Title : Wireless and Mobile Communication Lab.

Date of Performance :11-09-2020

Date of Submission : 18-09-2020

Submitted By

Name: Md. Fahim Al Mamun

ID: IT-15006

4th year 2nd Semester

Session: 2014-15/2015-16

Dept. of ICT

MBSTU.

Submitted To

Nazrul Islam

Assistant Professor

Dept. of ICT

MBSTU.

Experiment No: 04

Experiment Name: Protocol Analysis with Wireshark.

Objectives:

- To become familiarized with the wireshark application environment.
- To perform basic PDU capture using wireshark.
- To perform basic PDU analysis.
- To perform experiment with wireshark features and options such as PDU capture,display filtering and following TCP streams.
- To define the purpose of network protocol analyzer such as wireshark.

Wireshark core features:

- Capture live packet data.
- Import packets from text files.
- View packet data and protocol information.
- Save captured packet data.
- Display packets.
- Filter packets.
- Search packets.
- Colorize packets.
- Generate Statistics.

Installing and running wireshark:

In order to run Wireshark we need a couple of other tools installed on our system first. These are:

1.GTK+, The GIMP Tool Kit and Glib, both from the same source.

2.You will also need Glib.

3.Libpcap.

After installing the software for wireshark, we need to extract it from the tar file.

```
gzip -d wireshark-1.2-tar.gz
tar xvf wireshark-1.2-tar
```

Change to the Wireshark directory and then issue the following commands:

```
./configure
make
make install
```

Starting with wireshark:

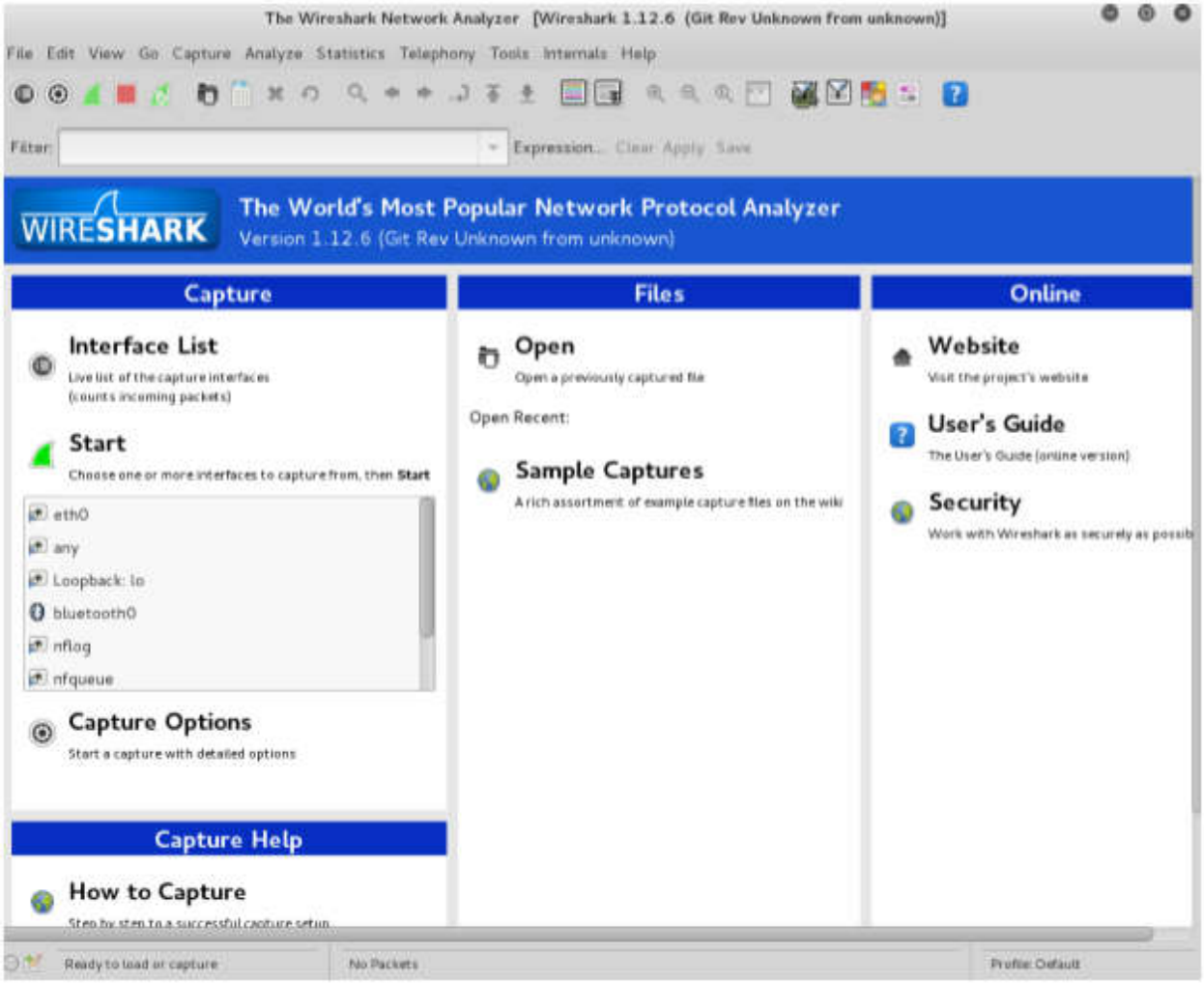

Figure 01: Initial graphic user interface of wireshark.

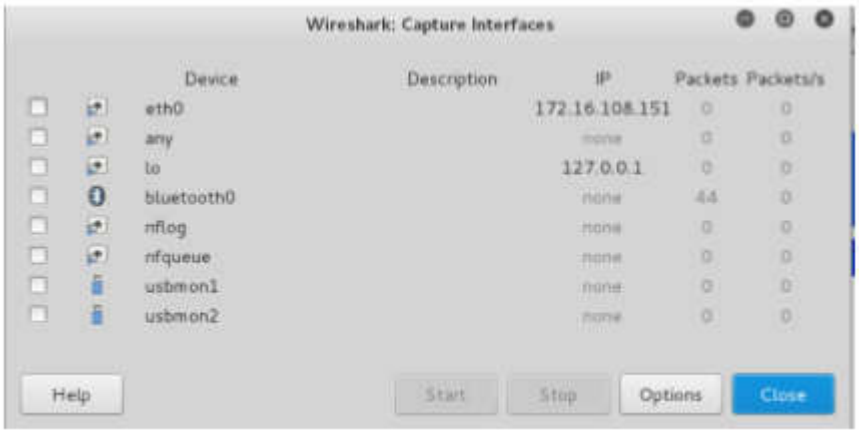

Figure 02: Wireshark Interface list.

How to capture data packets:

1. Make sure that you have the administrative privileges to start a live capture on your device.
2. Choose the correct network interface to capture packet data from.
3. Capture packet data from the correct location in your network.

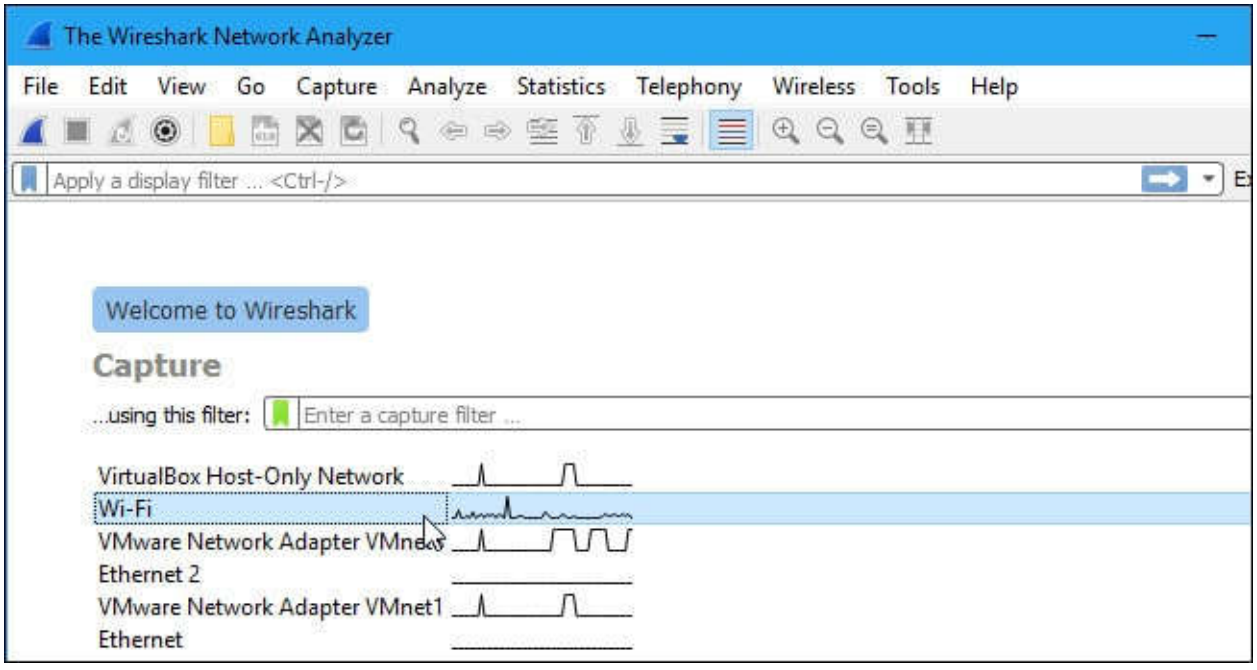

Figure 03: The wireshark network analyzer.

On Unix or Linux, the dialog box is shown in a similar style like this:

Figure 04: Wireshark capture interfaces in linux.

We can also start Wireshark by using the following command line:

```
<¢ wireshark -i   eth0 —k>
```

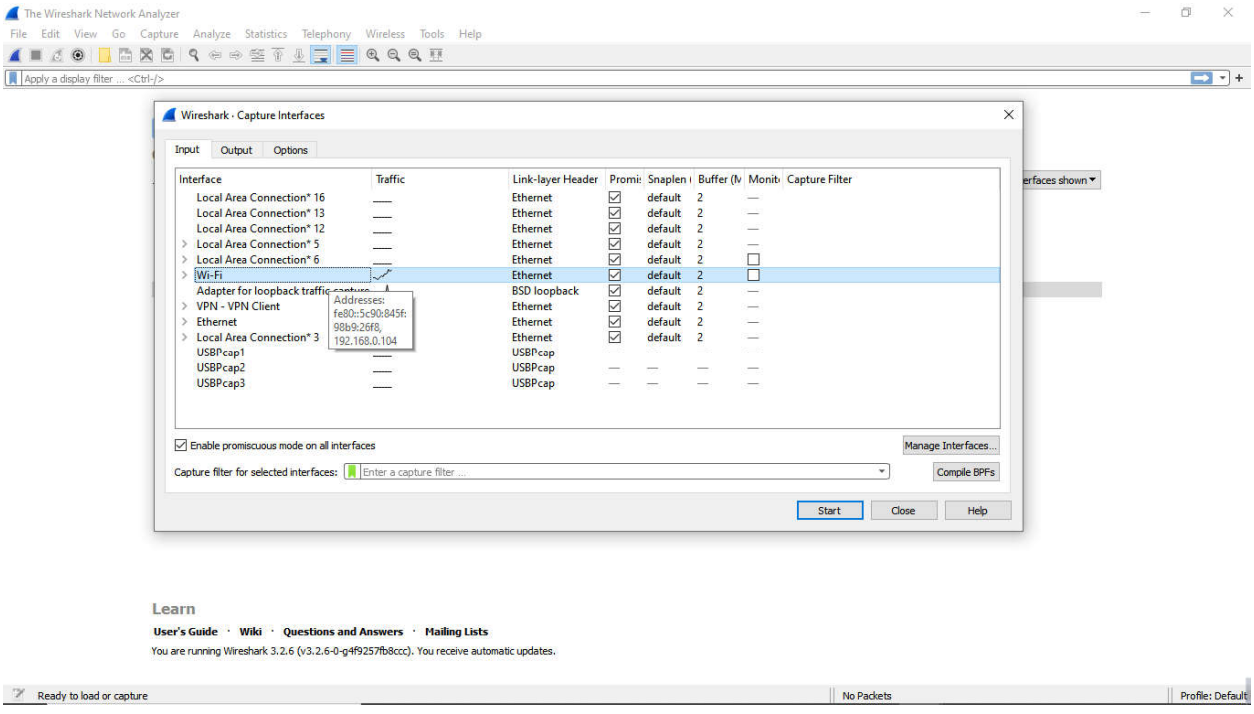

Figure 05: Start Capturing Interface that has IP address

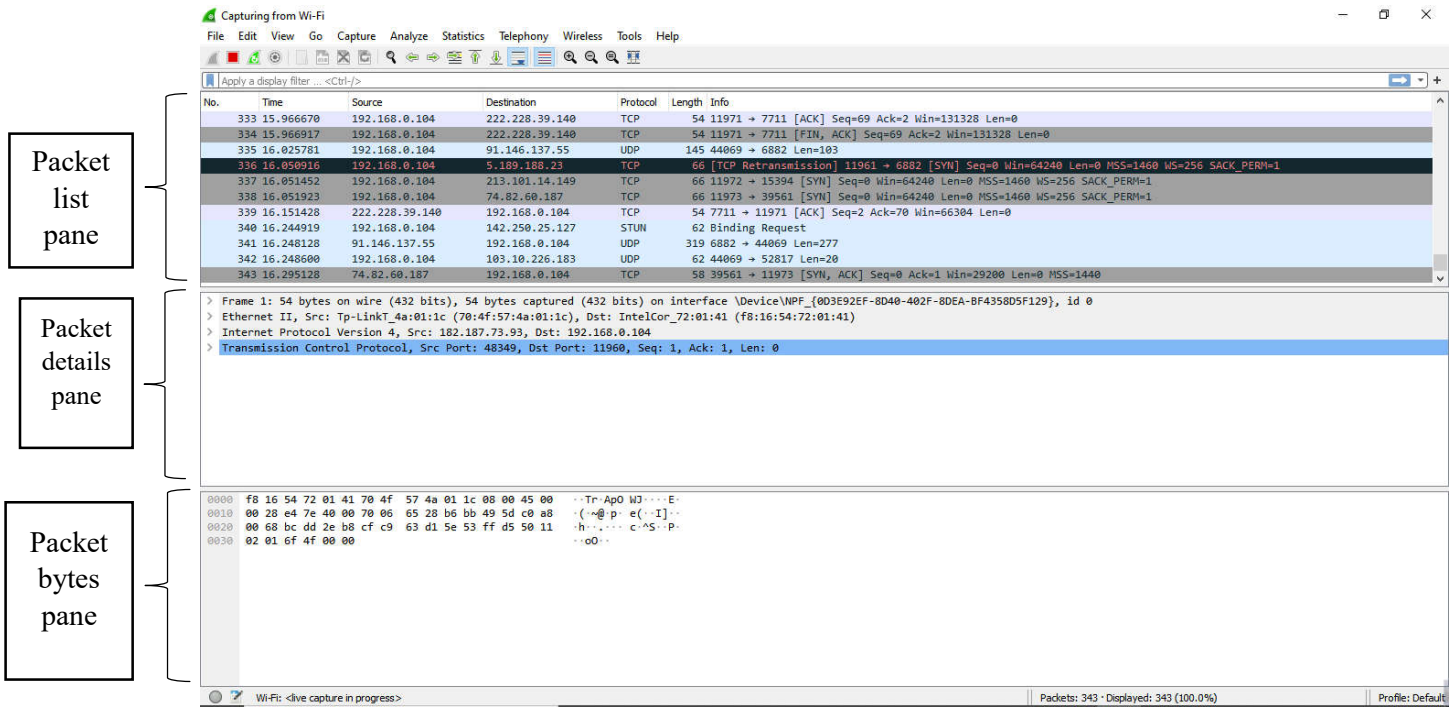

Figure 06: A sample packet capture window.

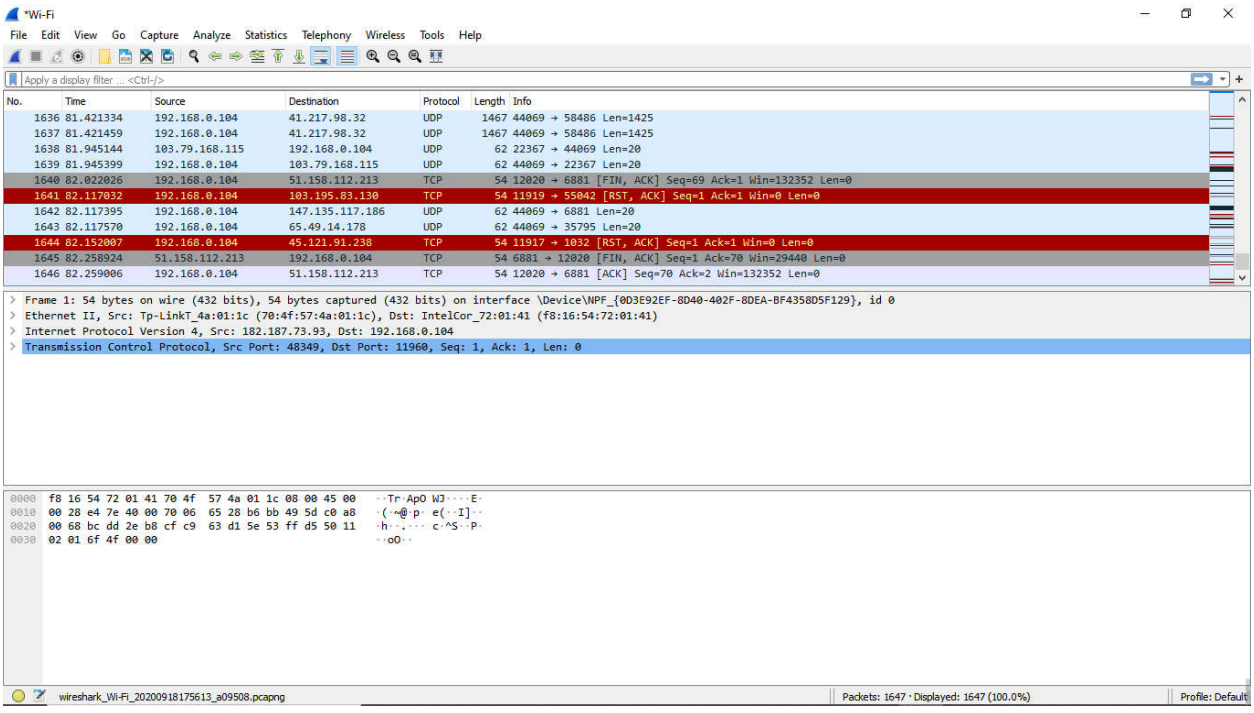

Figure 07: Stopping Capture.

Filtering:

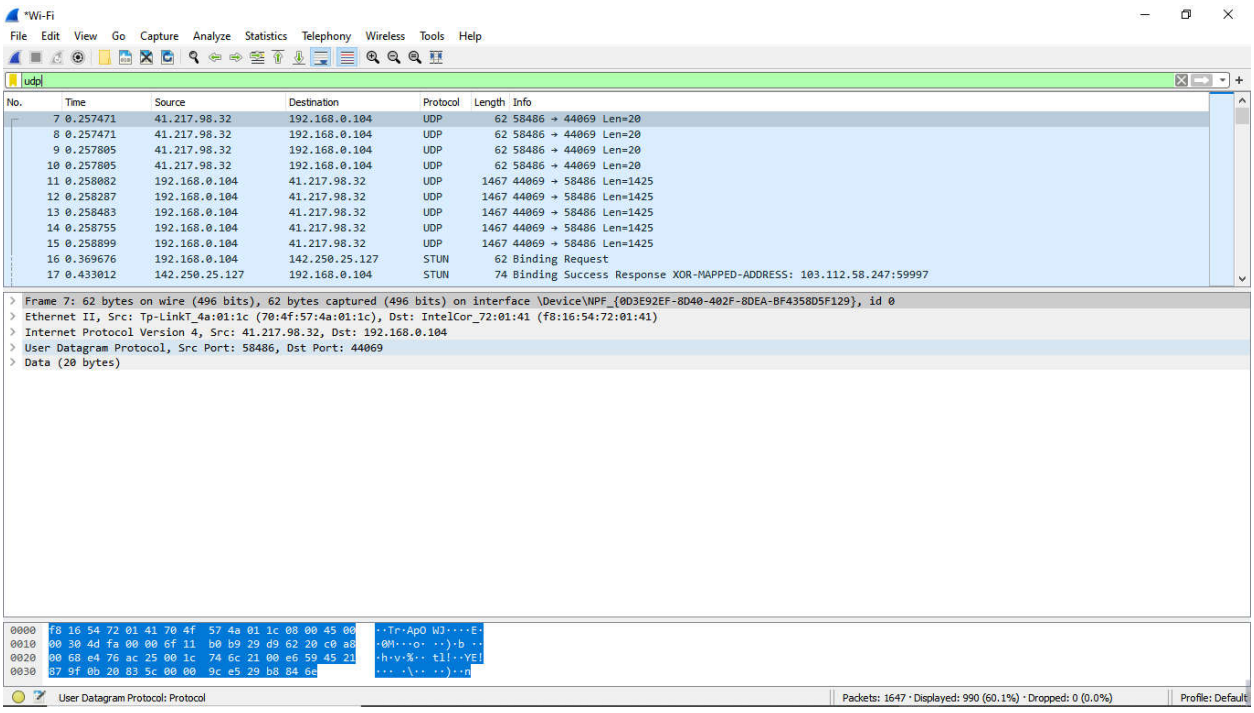

Figure 08: Filter by Protocol

A source filter can be applied to restrict the packet view in wireshark to only those packets that have source IP as mentioned in the filter.

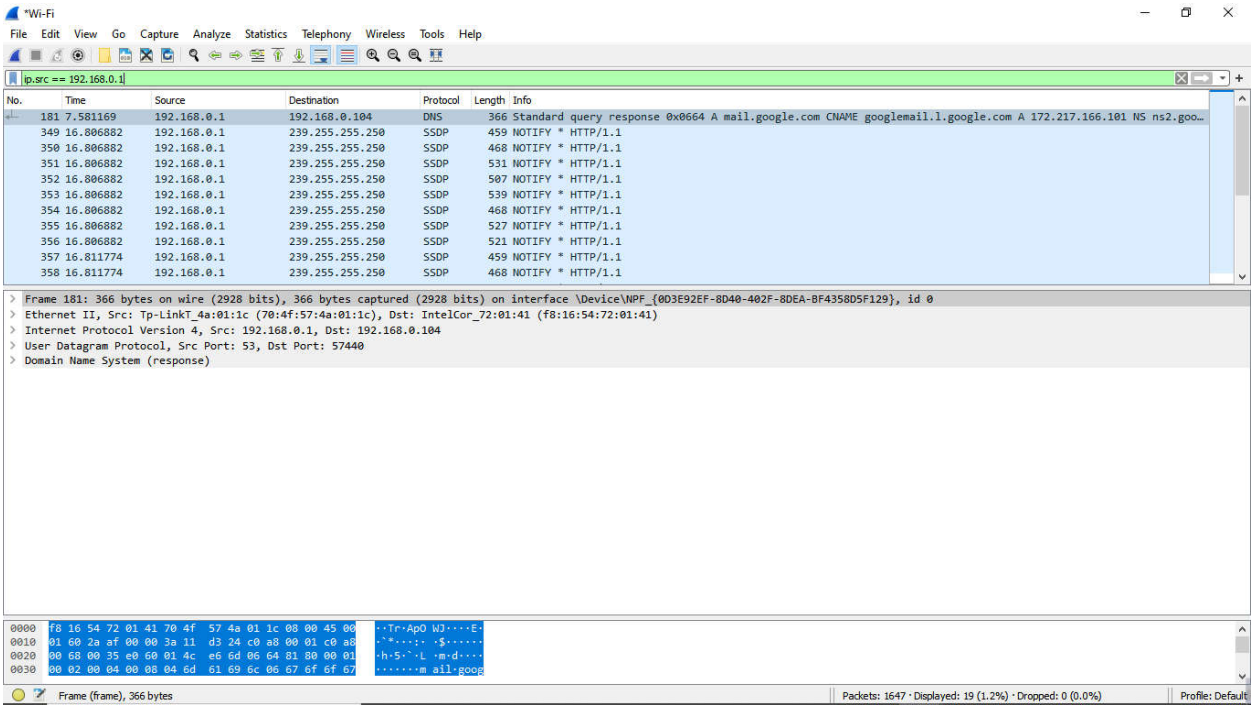

Figure 09: Source IP filter.

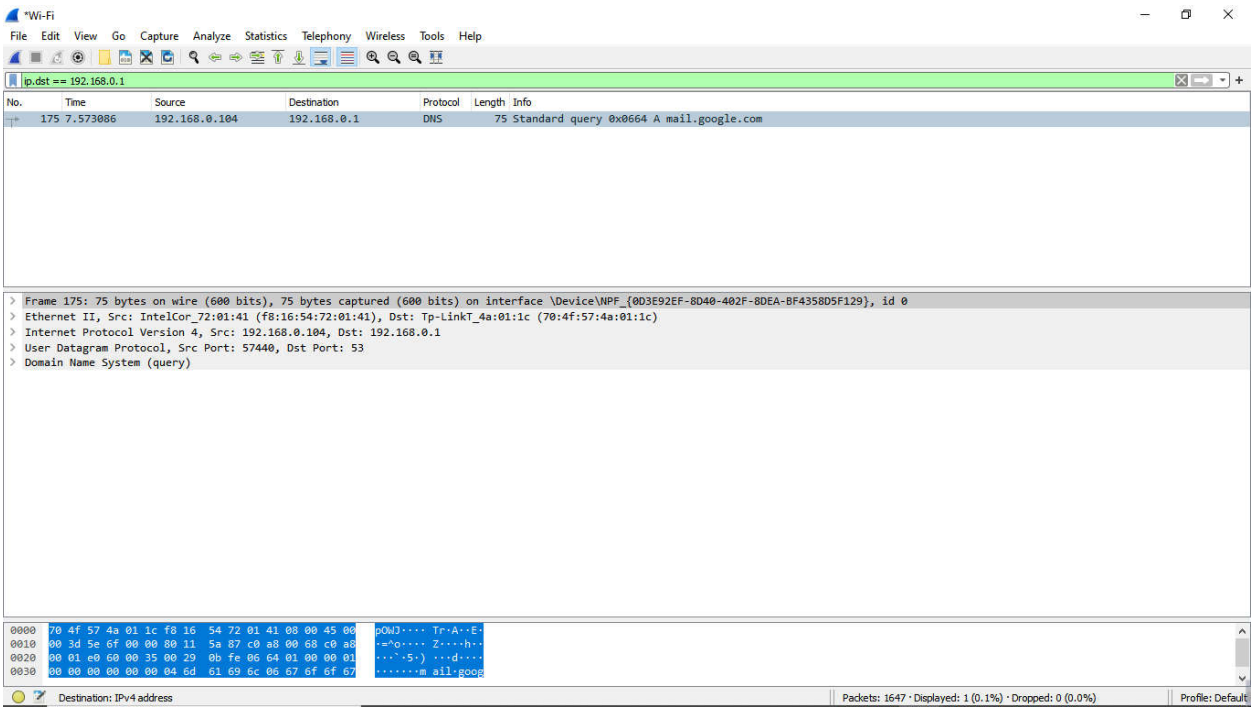

Figure 10: Destination IP filter.

• Packets and protocols can be analyzed after capture

• Individual fields in protocols can be easily seen

• Graphs and flow diagrams can be helpful in analysis

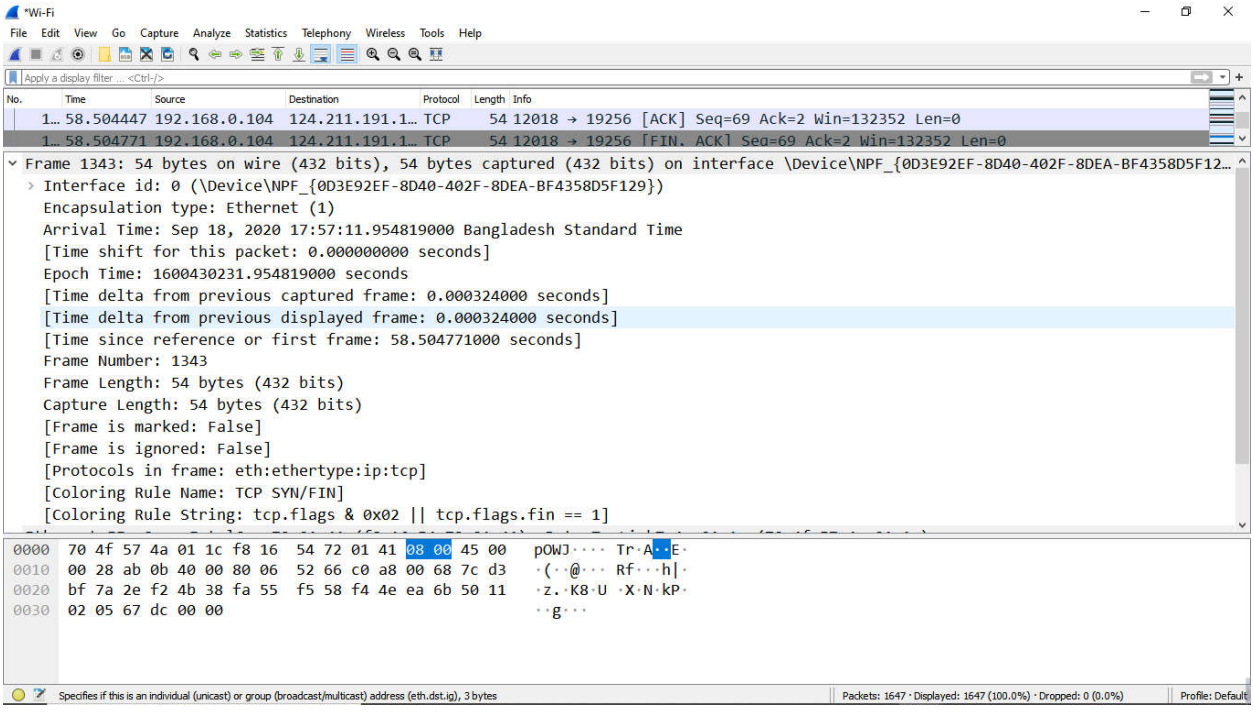

Figure 11: Packet Details Pane(Frame segment).

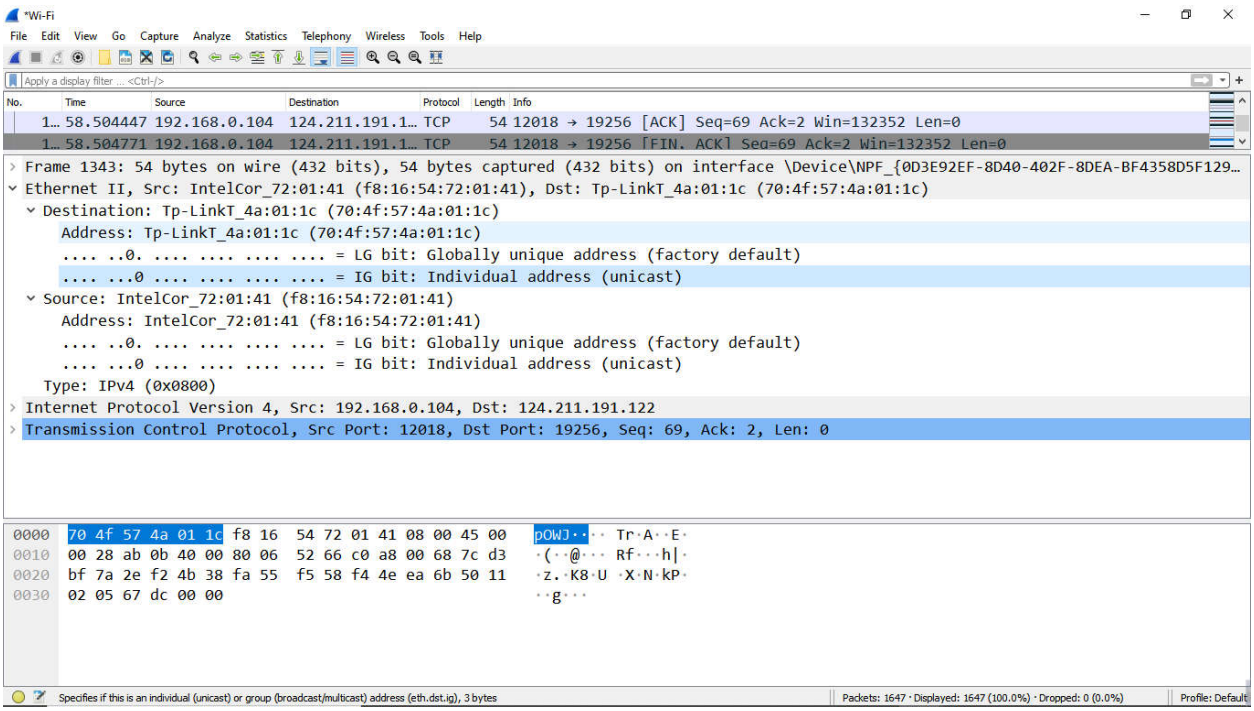

Figure 12: Packet Details Pane (Ethernet Segment).

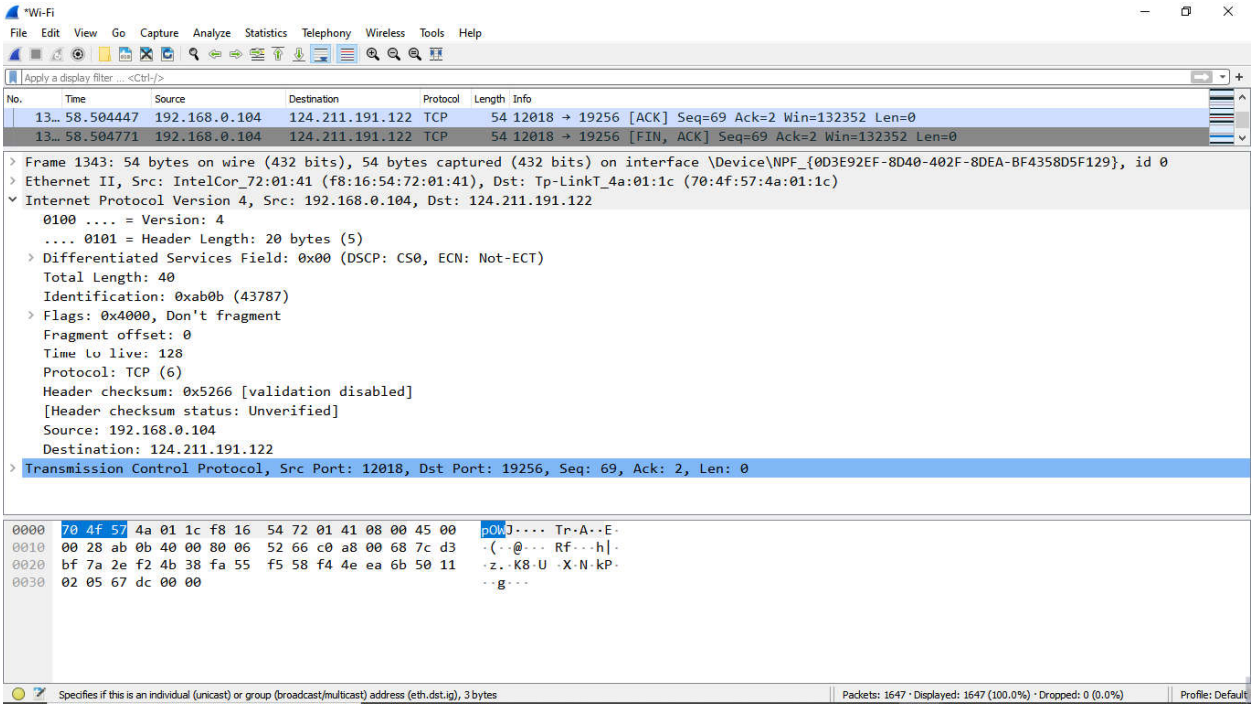

Figure13: Packet Details Pane(IP segment).

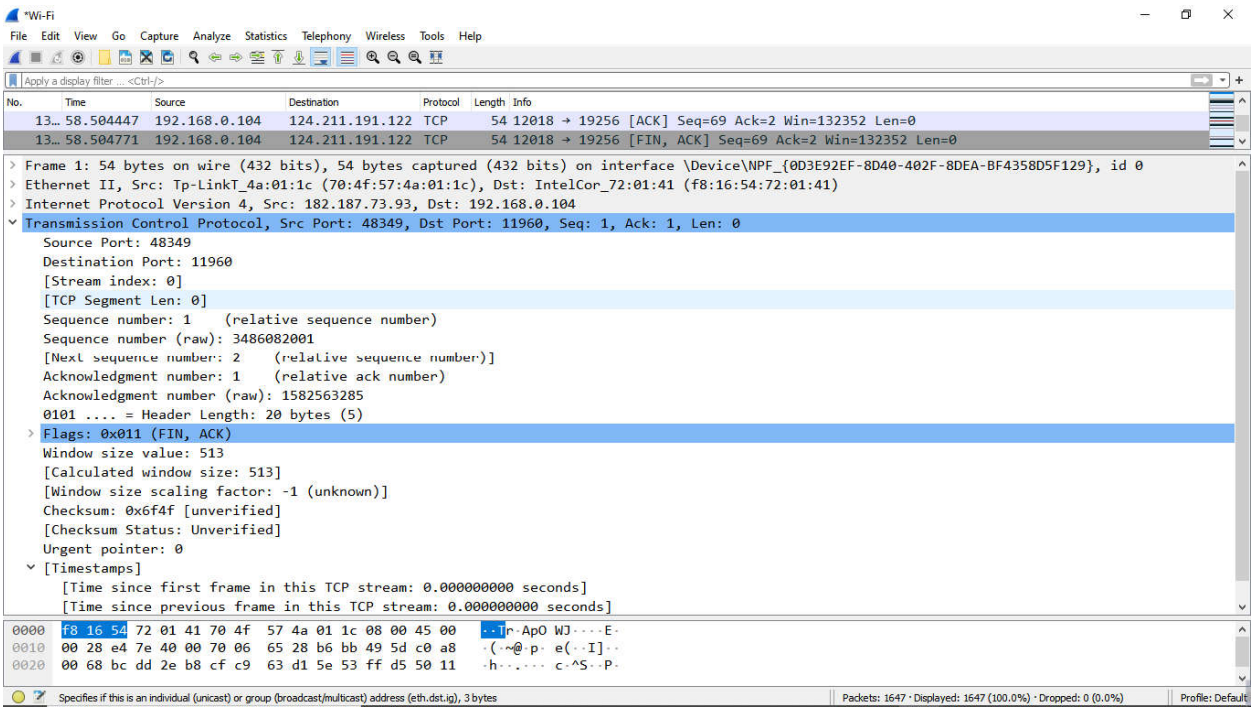

Figure 14: Packet Details Pane (TCP Segment).

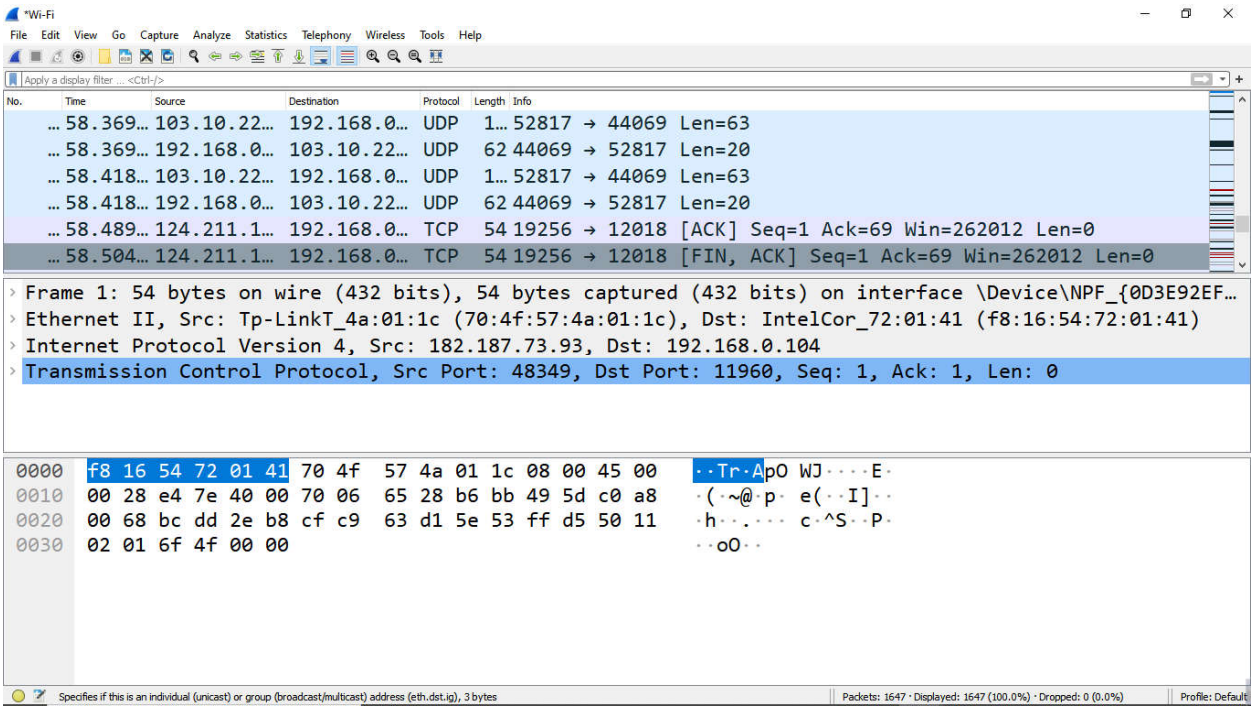

Figure 15: Packet Byte Pane.

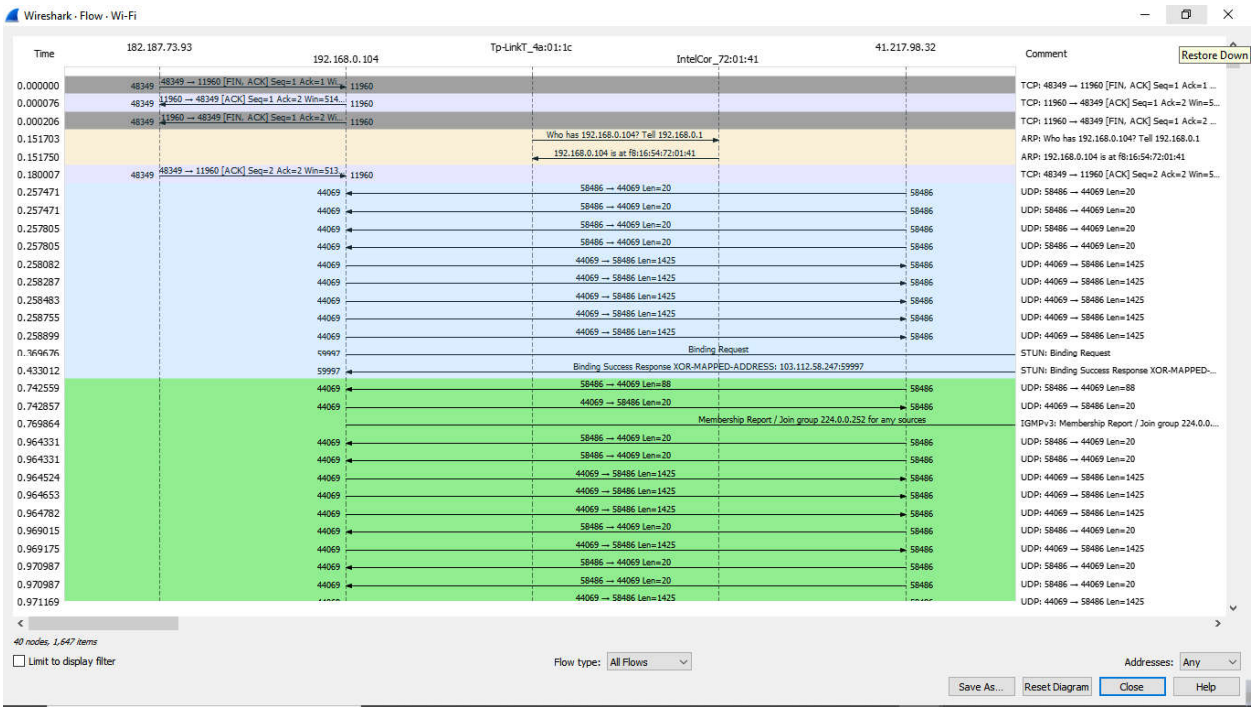

Figure 16: Statistics- Flow Graph(All Flows).

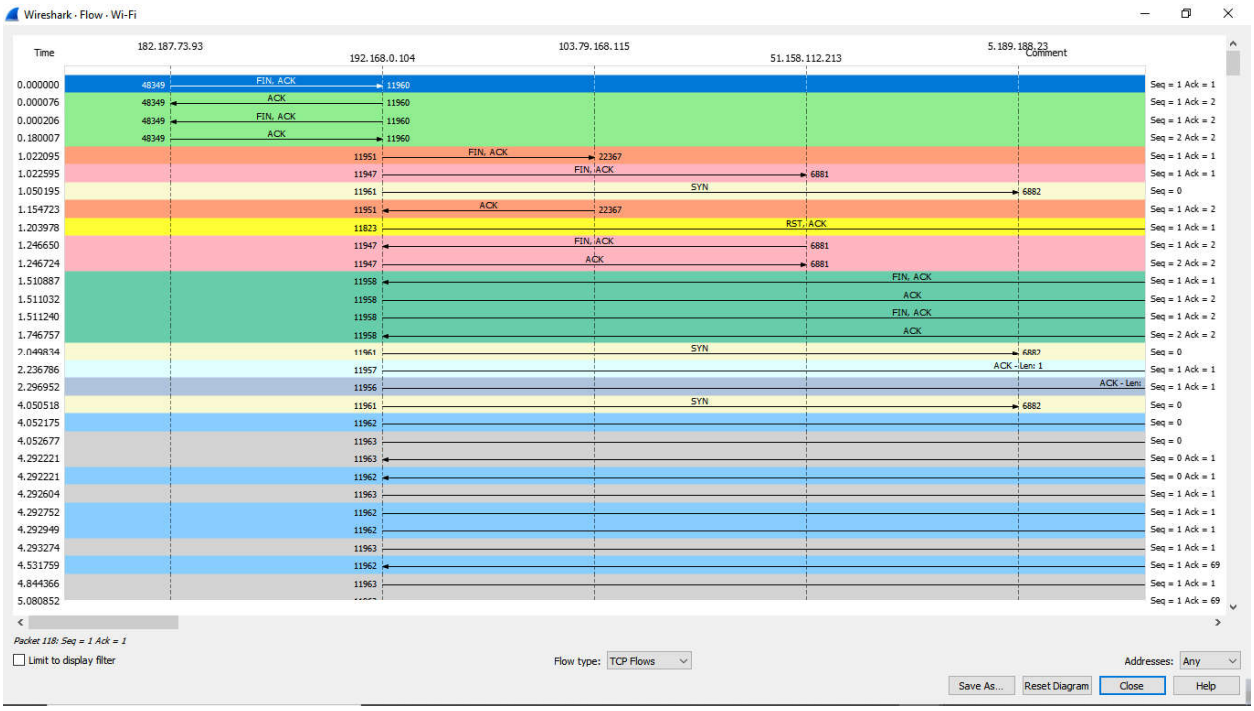

Figure 17: Statistics- Flow Graph(TCP Flows).

Conclusion:Wireshark is a network protocol analysis tool. At its core, Wireshark was designed to break down packets of data being transferred across different networks. The user can search and filter for specific packets of data and analyze how they are transferred across their network. These packets can be used for analysis on a real-time or offline basis. You will find out how to start up a packet capture and what information you can expect to get out of it. The Wireshark tutorial will also show you how to get the best out of the data manipulation functions within the interface. You will also learn how you can get better data analysis functions than those that are native to Wireshark. Wireshark, users can troubleshoot network problems, examine network security issues, debug protocols, and learn network processes.