# PRACTICAL-12

## AIM: Study use the wireshark for the various network protocols.

### INTRODUCTION:

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. One could think of a network packet analyzer as a measuring device for examining what's happening inside a network cable, just like an electrician uses a voltmeter for examining what's happening inside an electric cable. Wireshark is available for free, is open source and is one of the best packet analyzer available today. Wireshark is available for windows, linux and mac operating system.

### PACKET SNIFFER:

The basic tool for observing the messages exchanged between executing protocol entities is called a packet sniffer. As the name suggests, a packet sniffer captures ("sniffs") messages being sent/received from/by your computer; it will also typically store and/or display the contents of the various protocol fields in these captured messages. A packet sniffer itself is passive. It observes messages being sent and received by applications and protocols running on your computer, but never sends packets itself. Similarly, received packets are never explicitly addressed to the packet sniffer. Instead, a packet sniffer receives a copy of packets that are sent / received from/by application and protocols executing on your machine.

## PURPOSES:

Here are some reasons people use Wireshark:

- Network administrators use it to troubleshoot network problems
- Network security engineers use it to examine security problems
- QA engineers use it to verify network applications
- Developers use it to debug protocol implementations
- People use it to learn network protocol internals.

## FEATURES:

The following are some of the many features Wireshark provides:

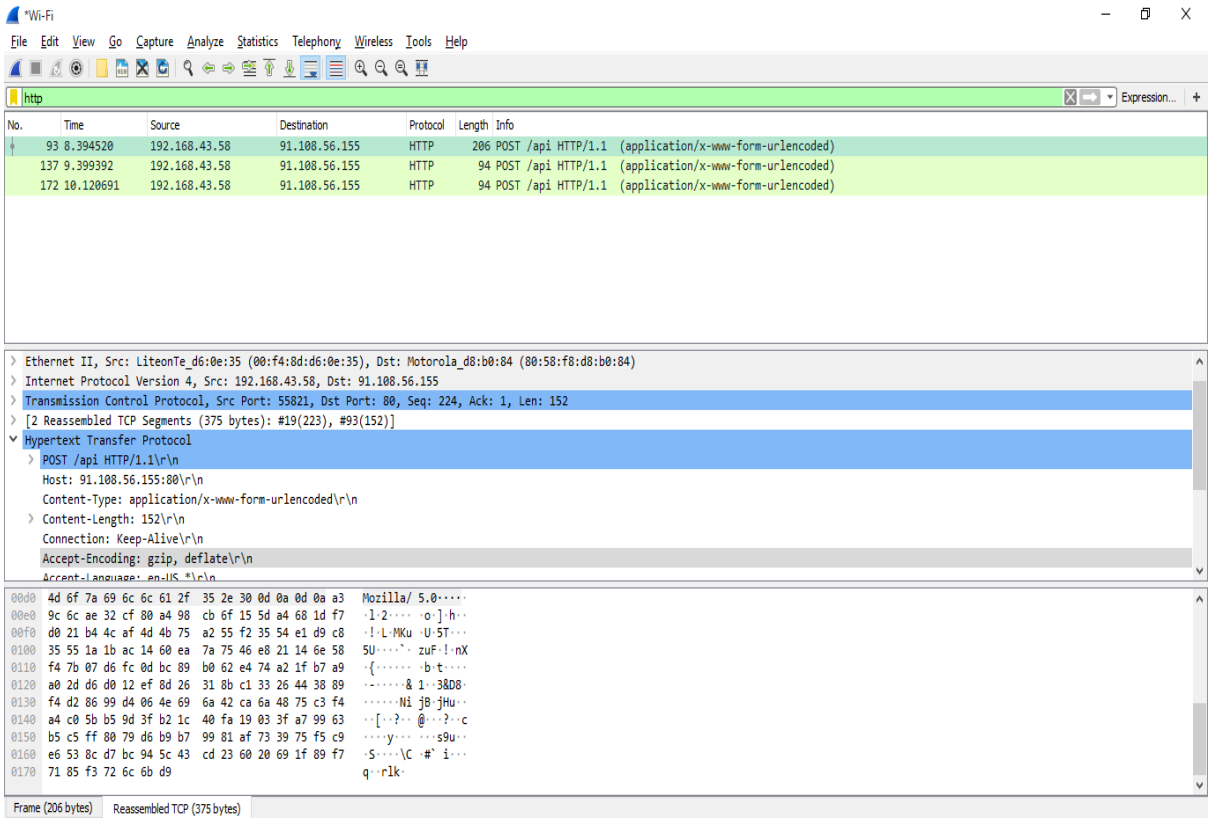
- Available for UNIX and Windows.
- Capture live packet data from a network interface.
- Open files containing packet data captured with tcpdump/WinDump, Wireshark, and many other packet capture programs.
- Import packets from text files containing hex dumps of packet data.
- Display packets with very detailed protocol information.
- Save packet data captured.
- Export some or all packets in a number of capture file formats.
- Filter packets on many criteria.
- Search for packets on many criteria.
- Colorize packet display based on filters.

- Create various statistics.
- ...and a lot more!

**Wireshark can be downloaded from following link:** https://wireshark.org

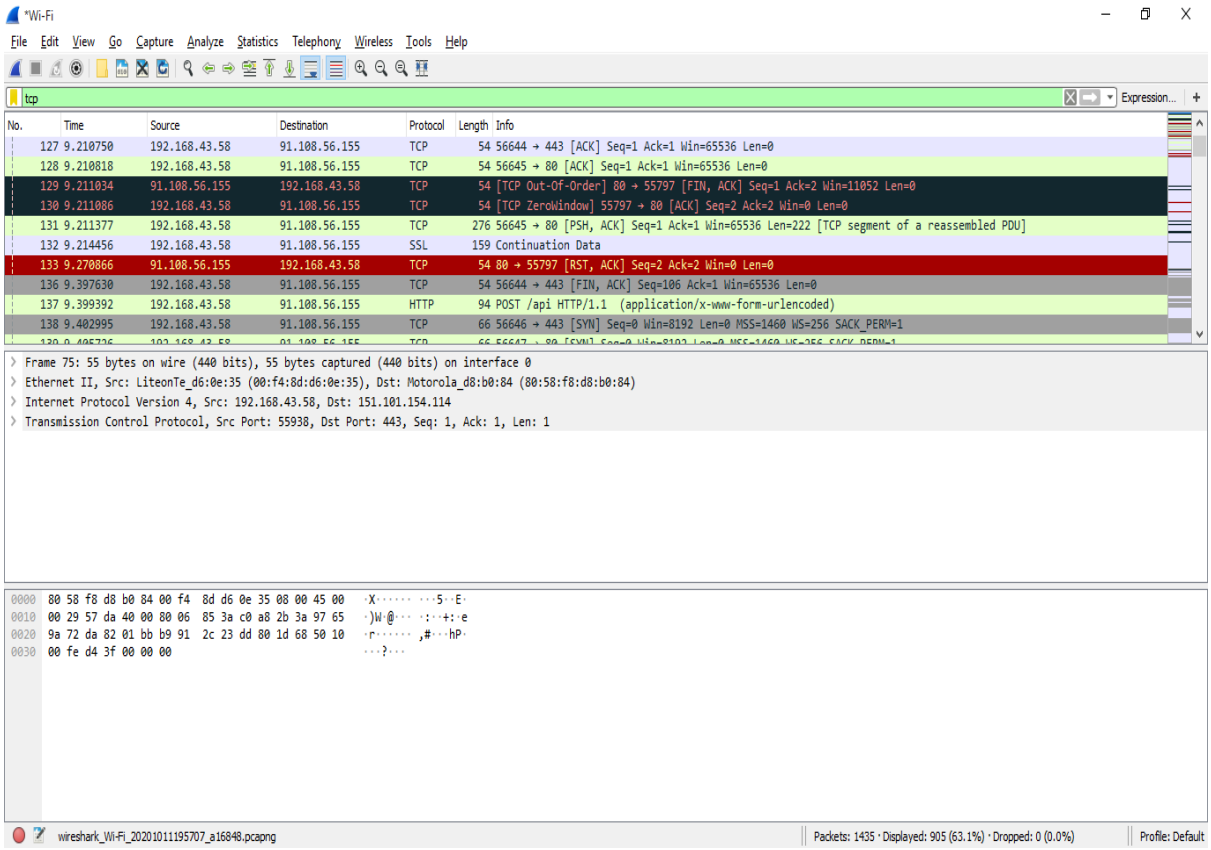
## HTTP Protocol :

The **Hypertext Transfer Protocol** (**HTTP**) is an application layer protocol for distributed, collaborative, hypermedia information systems. HTTP is the foundation of data communication for the World Wide Web, where hypertext documents include hyperlink to other resources that the user can easily access.



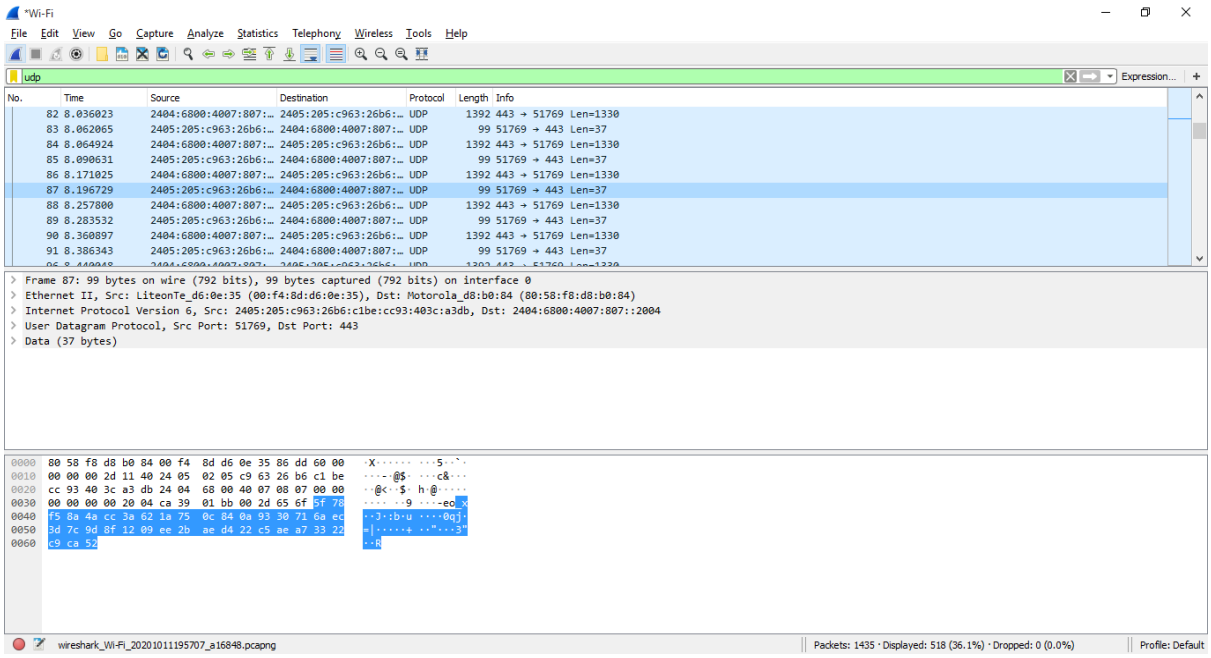

## TCP Protocol :

TCP(**Transmission Control Protocol**) is connection-oriented, and a connection between client and server is established before data can be sent. The server must be listening (passive open) for connection requests from clients before a connection is established. Three-way handshake (active open), retransmission, and error-detection adds to reliability but lengthens latency. Applications that do not require reliable data stream service may use the User Datagram Protocol (UDP), which provides a connectionless datagram service that prioritizes time over reliability. TCP employs network congestion avoidance. However, there are vulnerabilities to TCP including denial of service, connection hijacking, TCP veto, and reset attack. For network security, monitoring, and debugging, TCP traffic can be intercepted and logged with a packet sniffer.



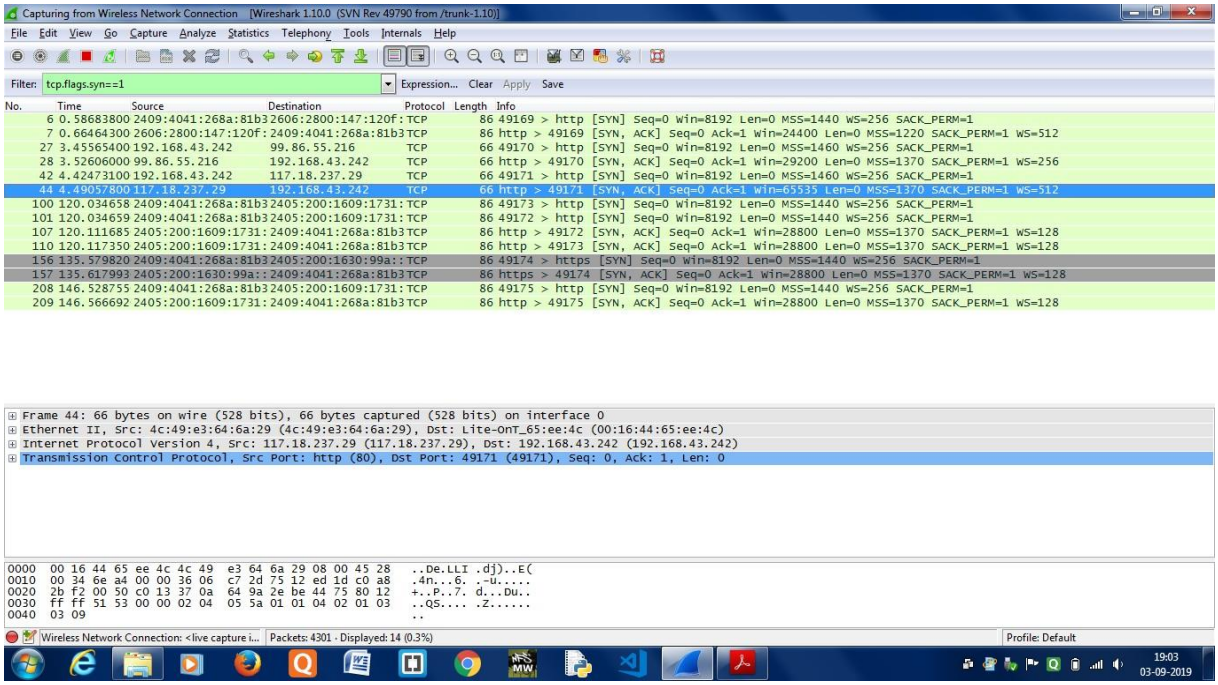

## UDP Protocol :

UDP(**User Datagram Protocol**) uses a simple connectionless communication model with a minimum of protocol mechanisms. UDP provides checksums for data integrity, and port numbers for addressing different functions at the source and destination of the datagram. It has no handshaking dialogues, and thus exposes the user's program to any unreliability of the underlying network; there is no guarantee of delivery, ordering, or duplicate protection.



# COMMANDS FOR WIRESHARK

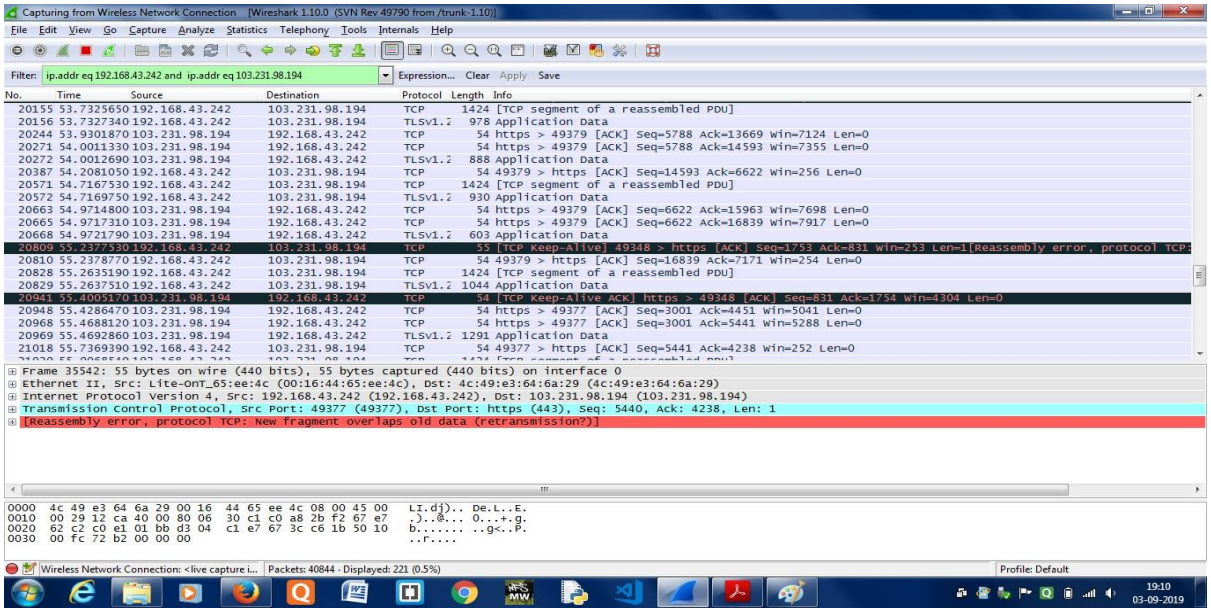
1) Analyzing TCP session using wireshark:
Find first SYN packet, sent from your PC to the web server. This signifies the start of a TCP 3-way handshake. Choose the correct flag and add==1.Hit the find button and first SYN packet in the trace should be highlighted.





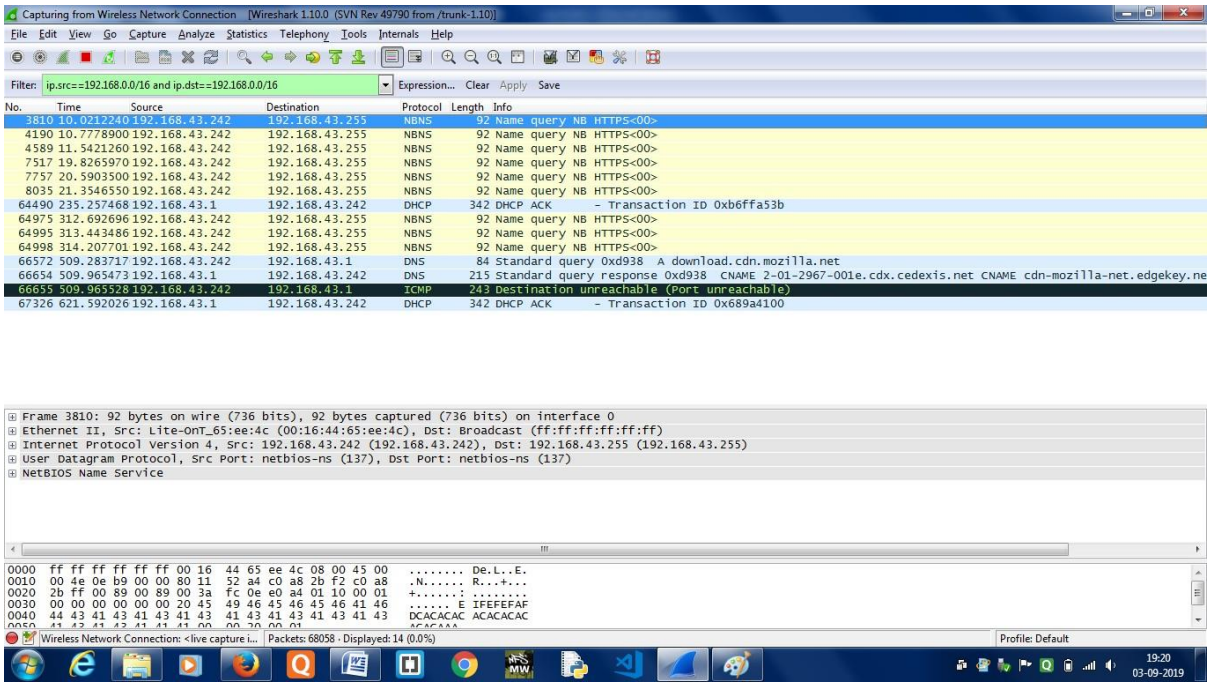


2) Filtering particular IP address from given IP addresses:
This command helps a programmer to filter packets according to its requirement from available packets which are being travelling from source port to destination port.Thus,by using wireshark we can analyse packets travelling from source to destination.
ip.addr eq 192.168.43.242 and ip.addr eq 103.231.98.194



3)Analysing packets travelling from a particular source to particular destination:
The given command helps a programmer to trace those packets which travelling from given source ip address to given destination ip address.

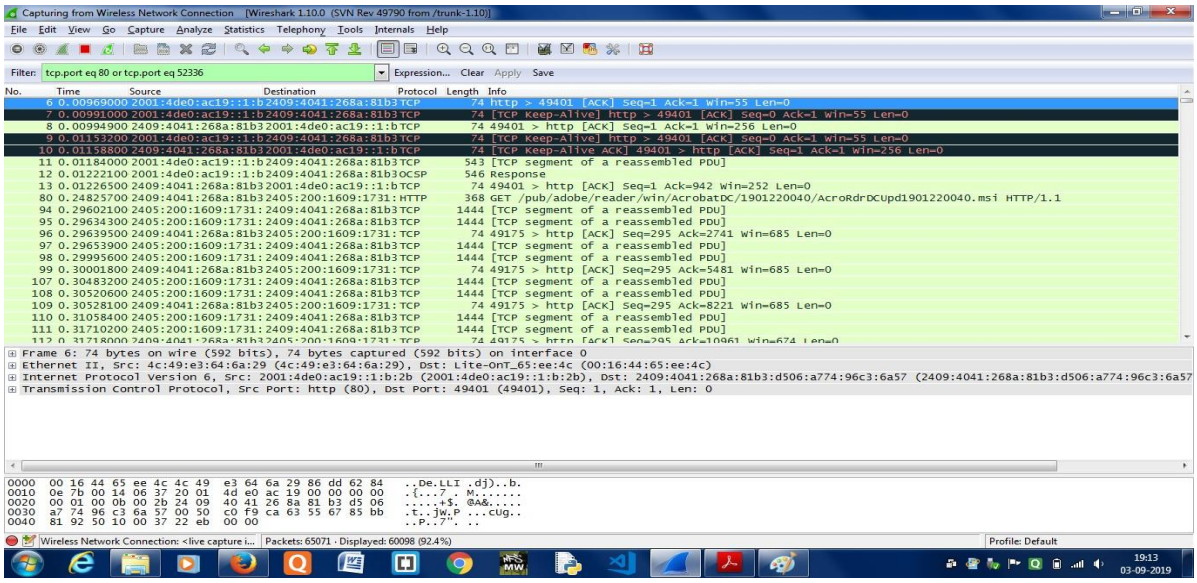
ip.src==192.168.0.0/16 and ip.dst==192.168.0.0/16




4)Monitoring traffic on more than two ports:

Sometimes you will be interested in inspecting traffic that matches either (or both) conditions whatsoever. For example if user wants to monitor tcp ports and other ports simultaneously then or relation can be used between them.

Tcp.port eq 80 or tcp.port eq 52336




## CONCLUSION:

We have concluded that wireshark is a program that is used to capture data packets to allow a more precise analysis. Such a tool allows the user to examine his/her own computer for protocol errors and problems within the network architecture.