Exp No:- 08

Aim:- Study of packet sniffer tools Wireshark.
a) Observer performance in promiscuous as well as non-promiscuous mode.
b) Show the packets can be traced based on different filters.

Hardware / Software required:-
Wireshark, Ethernet & tcpdump.

Theory:-
Wireshark, a network analysis tool formerly known as Ethereal, captures packets in real time and display them in human-readable format.
Wireshark includes filters, color-coding & other features that let you dig deep into network traffic and inspect individual packets.

Applications:-
- Network administrators use it to troubleshoot network problems.
- Network Security engineers use it to examine Security problems.
- Developers use it to debug protocol implementations
- People use it to learn network protocol internals beside these examples can be helpful in many other situations too.

Features:-
- Available for UNIX and Windows.
- Capture live packet data from a network interface.

- Open files containing packet data captured with tcpdump / WinDump, Wireshark, and a number of other packet capture program.
- Import packets from text files containing hex dumps of packet data.
- Display packets with very detailed protocol information.
- 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 stastics.

Capturing Packets:-

After downloading and installing Wireshark, you can Launch it and click the name of an interface under Interface List to start capturing packets on that interface. for example, if you want to capture traffic on the Wireless network, click your Wireless interface. You can configure advanced features by clicking Capture Options.

As soon as you click the interface's name, you'll see the packets start to appear in real time. Wireshark captures each packet sent to or from your system. If you're capturing on a Wireless interface and have promiscuous mode enabled in your capture options. you'll also see other the other packets on the network.

click the stop capture button near the top left corner of the Window When you want to stop capturing traffic.

Wireshark uses colors to help you identify the types of traffic at a glance. By default, green is TCP traffic, dark blue is DNS traffic, light blue is UDP traffic, and black identifies TCP

Packets with problems — for example, they could have been delivered out-of-order.

Filtering Packets:-

If you're trying to inspect something specific, such as the traffic a program sends when phonning home, it helps to close down all other applications using the network so you can narrow down the traffic; still, you'll likely have a large amount of packets to shift through. That's where Wireshark's filters come in.

The most basic way to apply a filter is by typing it into the filter box at the top of the Window & clicking Apply (or pressing Enter). For Example, type "dns" & you'll see only DNS packets. When you start typing, Wireshark Will help you automated complete your filter.

Another interesting thing you can do is right-click a packet & select Follow Tcp Stream. You'll see the full conversation between the client & the server.

Choose the Window & you'll find a filter has been applied automatically - Wireshark is showing, you the packets that make up the Conversation.

Inspecting Packets:-

Click a packet to select it and you can dig down to view its details.

You can also create filter from here - just right click one of the details & use the Apply as filter submenu to create a filter based on it.

Wireshark is an extremely powerful tool, And this is just scraching the surface of what you can do with it. Proffessionals use it to debug network protocol implementations, examine security

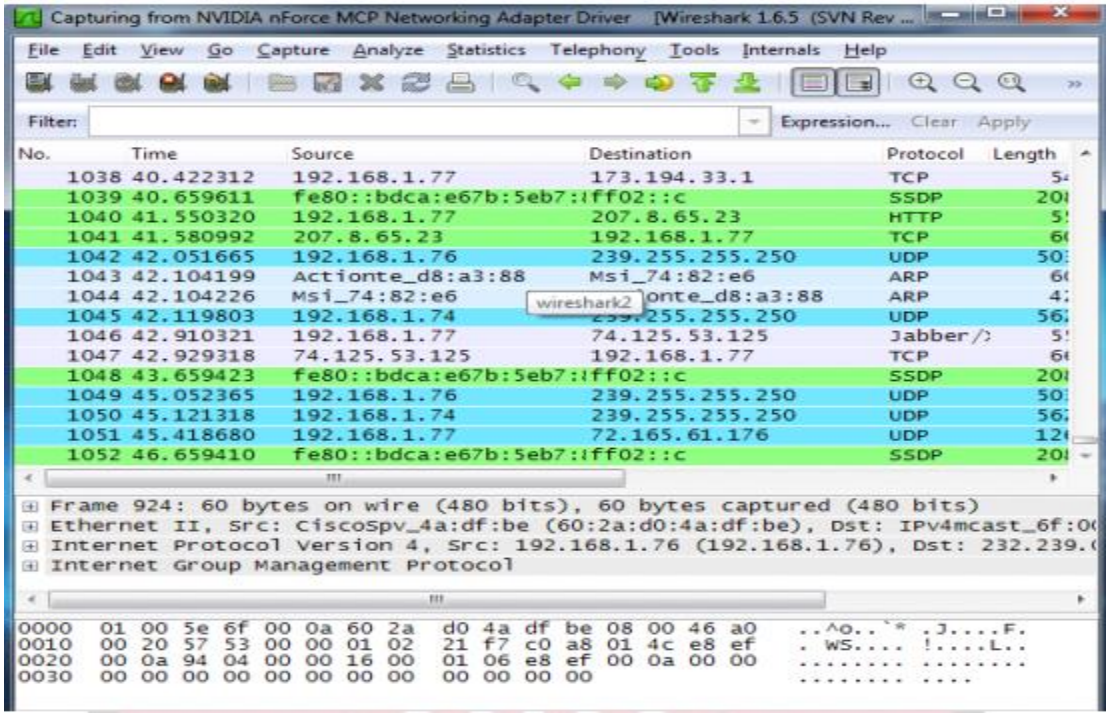
problems & inspect network protocol internals.

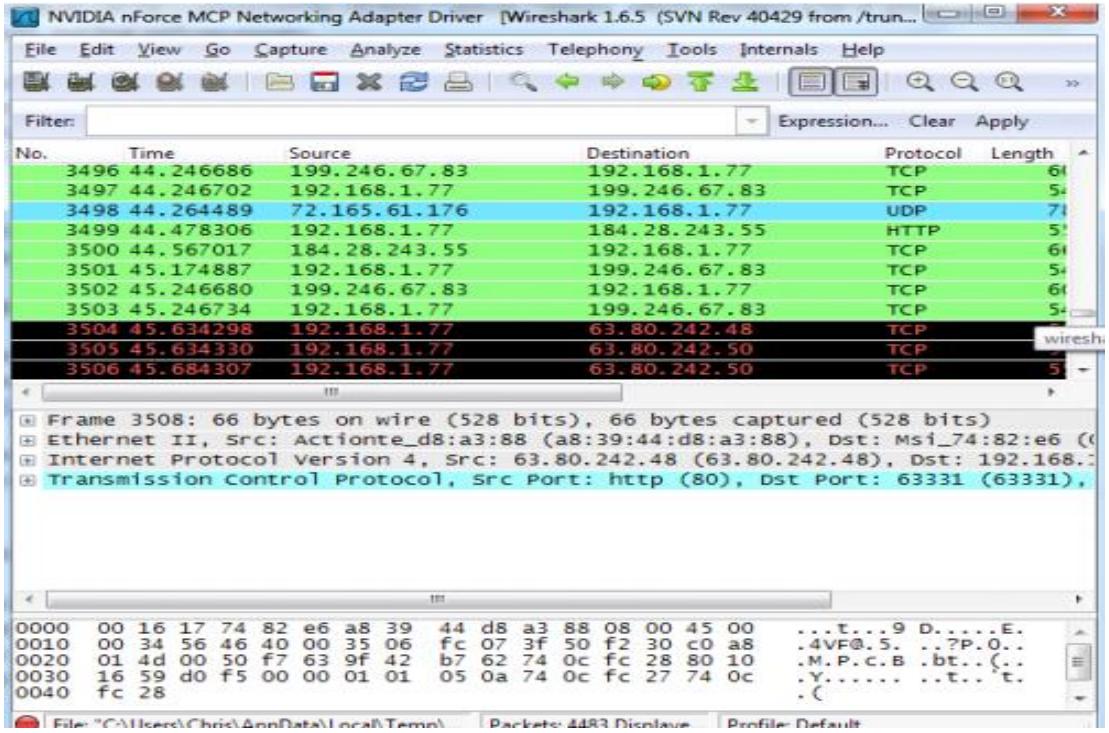
Conclusion:-

In this experiment, we analyze various packet sniffer tools that monitor network traffic transmitted between legitimate users or in the network. The packet sniffer is the network monitoring tools. It is opted for network monitoring, traffic analysis, troubleshooting, packet grappling, message, protocol analysis, penetration testing and many other purpose.

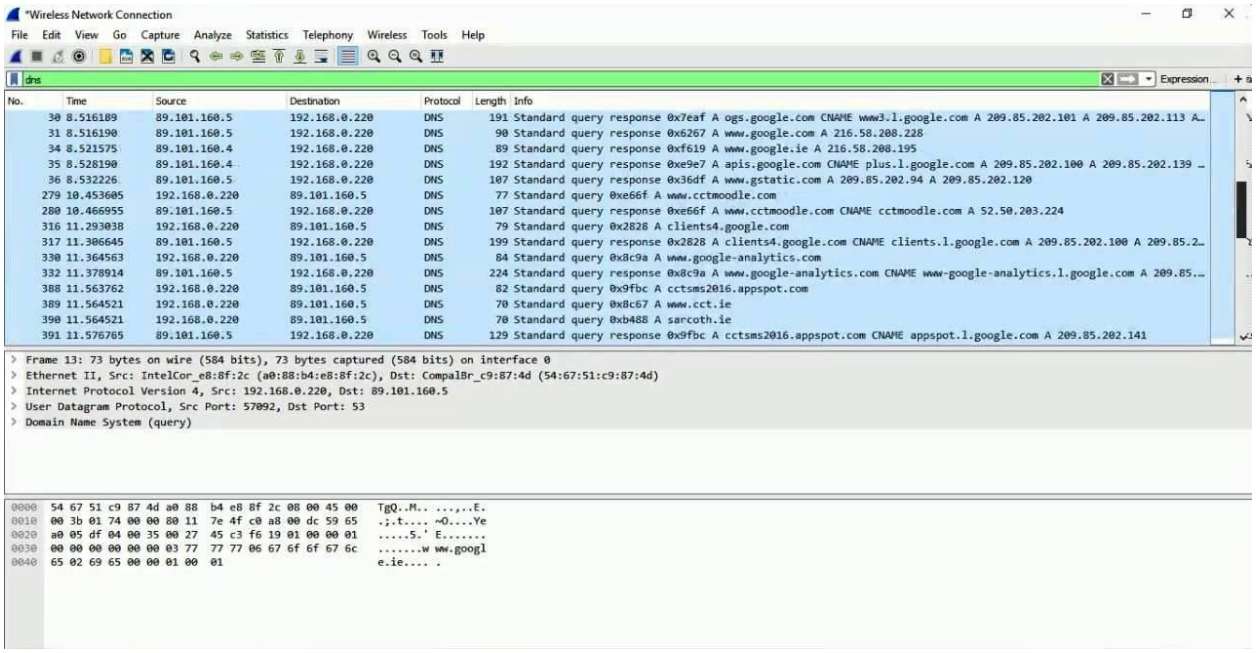
Output:

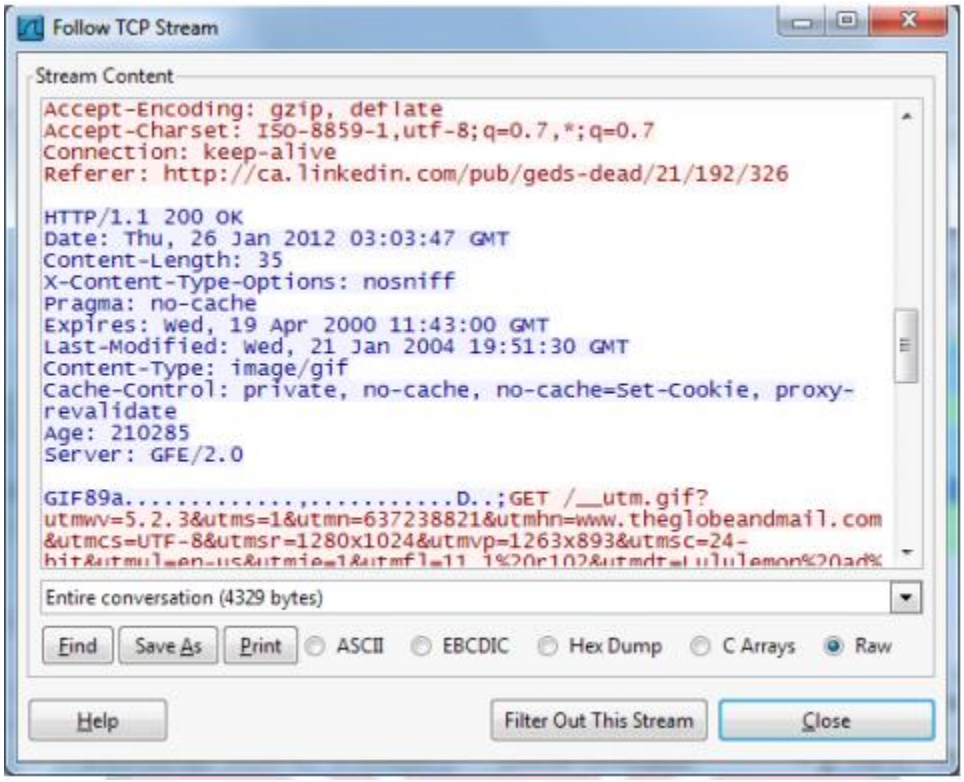
Capturing Packets

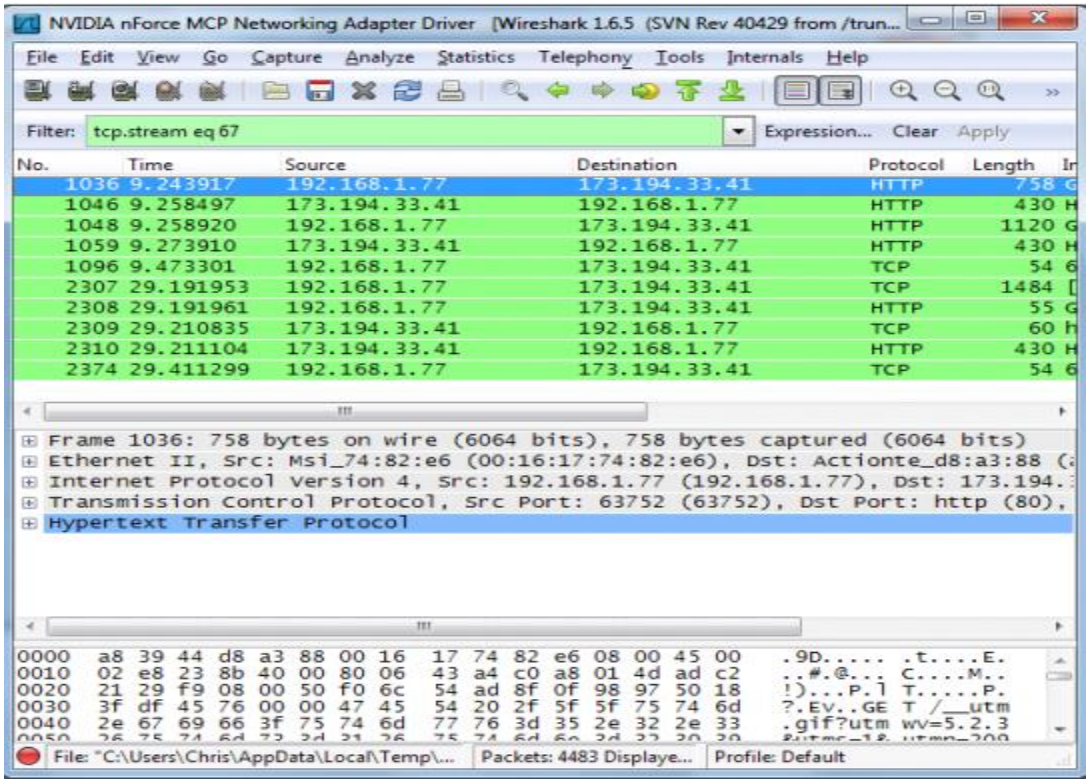




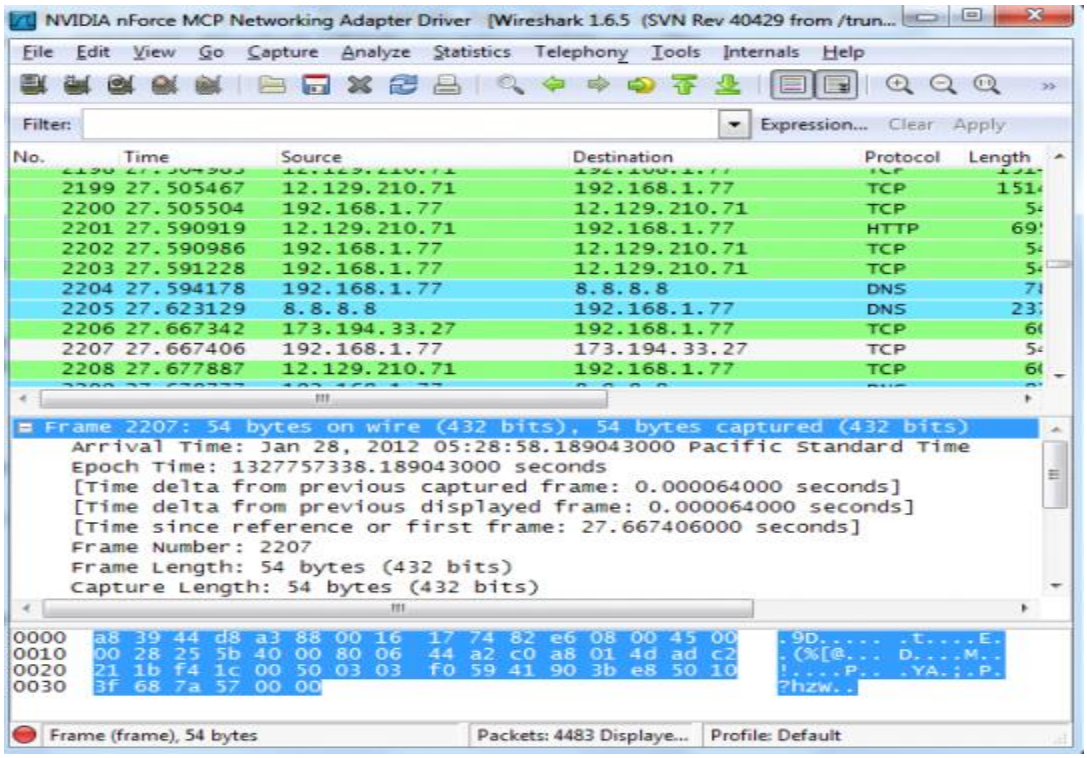
Filtering Packets

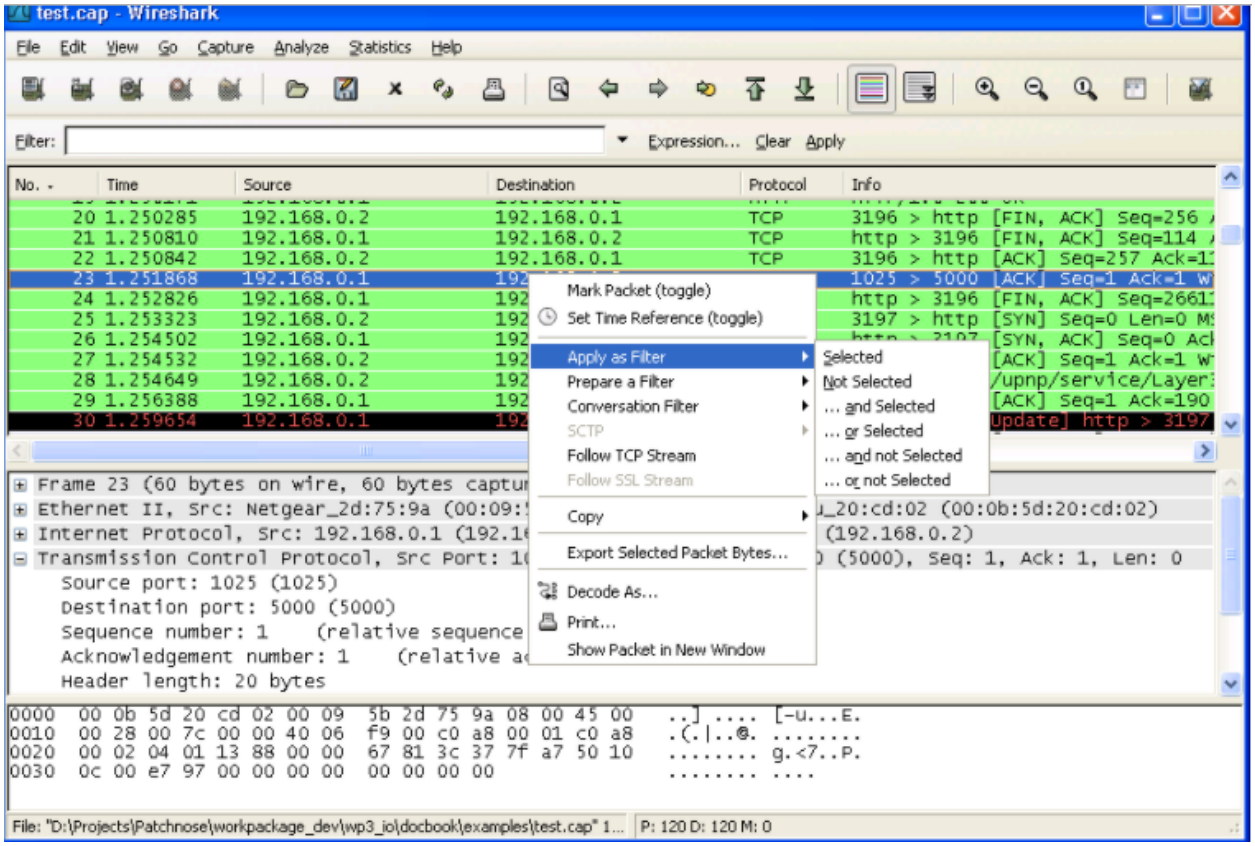






Inspecting Packets :





Conclusion:

In this experiment we analyze various packet sniffing tools that monitor network traffic transmitted between legitimate users or in the network. The packet sniffer is network monitoring tool. It is opted for network monitoring, traffic analysis, troubleshooting, Packet grapping, message, protocol analysis, penetration testing and many other purposes.