Experiment No. 9

Aim: To install and study Network Protocol Analyzer, analyse packet header using **Wireshark.**

API: Packet Analyzer, Ethereal, Wireshark, Ntop etc.

Theory :-

Version 3.6.2 (v3.6.2-0-g626020d9b3c3)

Copyright 1998-2022 Gerald Combs <gerald@wireshark.org> and contributors. License GPLv2+: GNU GPL version 2 or later <https://www.gnu.org/licenses/gpl-2.0.html> This is free software; see the source for copying conditions. There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

Compiled (64-bit) using Microsoft Visual Studio 2019 (VC++ 14.29, build 30139), with Qt 5.15.2, with libpcap, with GLib 2.66.4, with zlib 1.2.11, with Lua 5.2.4, with GnuTLS 3.6.3 and PKCS #11 support, with Gcrypt 1.8.3, with MIT Kerberos, with MaxMind DB resolver, with nghttp2 1.44.0, with brotli, with LZ4, with Zstandard, with Snappy, with libxml2 2.9.10, with libsmi 0.4.8, with QtMultimedia, with automatic updates using WinSparkle 0.5.7, with AirPcap, with SpeexDSP (using bundled resampler), with Minizip.

Running on 64-bit Windows 10 (21H2), build 19044, with Intel(R) Core(TM) i5-1035G1 CPU @ 1.00GHz (with SSE4.2), with 7973 MB of physical memory, with GLib 2.66.4, with Qt 5.15.2, with Npcap version 1.55, based on libpcap version 1.10.2-PRE-GIT, with c-ares 1.17.0, with GnuTLS 3.6.3, with Gcrypt 1.8.3, with nghttp2 1.44.0, with brotli 1.0.9, with LZ4 1.9.3, with Zstandard 1.4.0, without AirPcap, with light display mode, without HiDPI, with LC_TYPE=English_India.utf8, binary plugins supported (21 loaded).

Wireshark is Open Source Software released under the GNU General Public License.

Check the man page and https://www.wireshark.org for more information.

Wireshark is the world's foremost and widely-used network protocol analyzer. It lets you see what's happening on your network at a microscopic level and is the de facto (and often de jure) standard across many commercial and non-profit enterprises, government agencies, and educational institutions. Wireshark development thrives thanks to the volunteer contributions of networking experts around the globe and is the continuation of a project started by Gerald Combs in 1998.

Wireshark has a rich feature set which includes the following:

- Deep inspection of hundreds of protocols, with more being added all the time
- Live capture and offline analysis

- Standard three-pane packet browser
- Multi-platform: Runs on Windows, Linux, macOS, Solaris, FreeBSD, NetBSD, and many others
- Captured network data can be browsed via a GUI, or via the TTY-mode TShark utility
- The most powerful display filters in the industry
- Rich VoIP analysis

Read/write many different capture file formats: tcpdump (libpcap), Pcap NG, Catapult DCT2000, Cisco Secure IDS iplog, Microsoft Network Monitor, Network General Sniffer® (compressed and uncompressed), Sniffer® Pro, and NetXray®, Network Instruments Observer, NetScreen snoop, Novell LANalyzer, RADCOM WAN/LAN Analyzer, Shomiti/Finisar Surveyor, Tektronix K12xx, Visual Networks Visual UpTime, WildPackets EtherPeek/TokenPeek/AiroPeek, and many others

Capture files compressed with gzip can be decompressed on the fly

Live data can be read from Ethernet, IEEE 802.11, PPP/HDLC, ATM, Bluetooth, USB, Token Ring, Frame Relay, FDDI, and others (depending on your platform)

Decryption support for many protocols, including IPsec, ISAKMP, Kerberos, SNMPv3, SSL/TLS, WEP, and WPA/WPA2

Coloring rules can be applied to the packet list for quick, intuitive analysis

Output can be exported to XML, PostScript®, CSV, or plain text

Procedure: To download Wireshark:

1.Open a web browser.

2.Navigate to http://www.wireshark.org.

3.Select Download Wireshark.

4.Select the Wireshark Windows Installer matching your system type, either 32-bit or 64-bit as determined in Activity 1. Save the program in the Downloads folder.

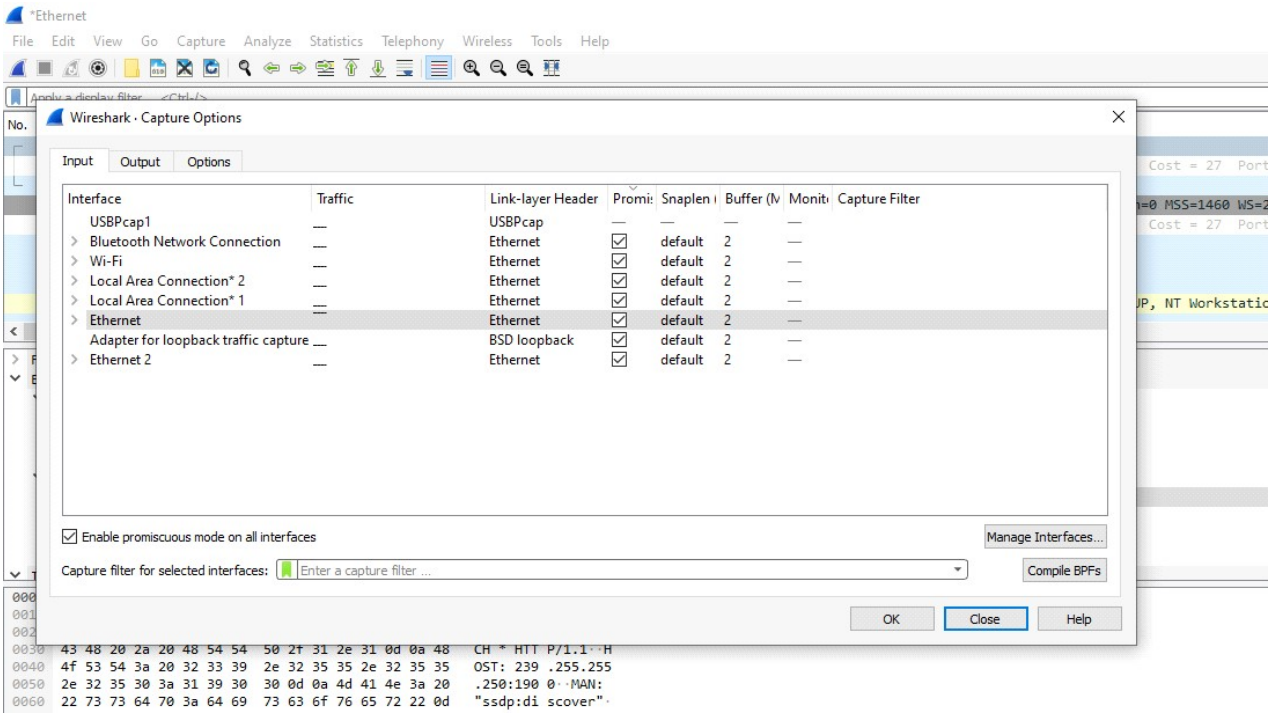
5.Close the web browser.

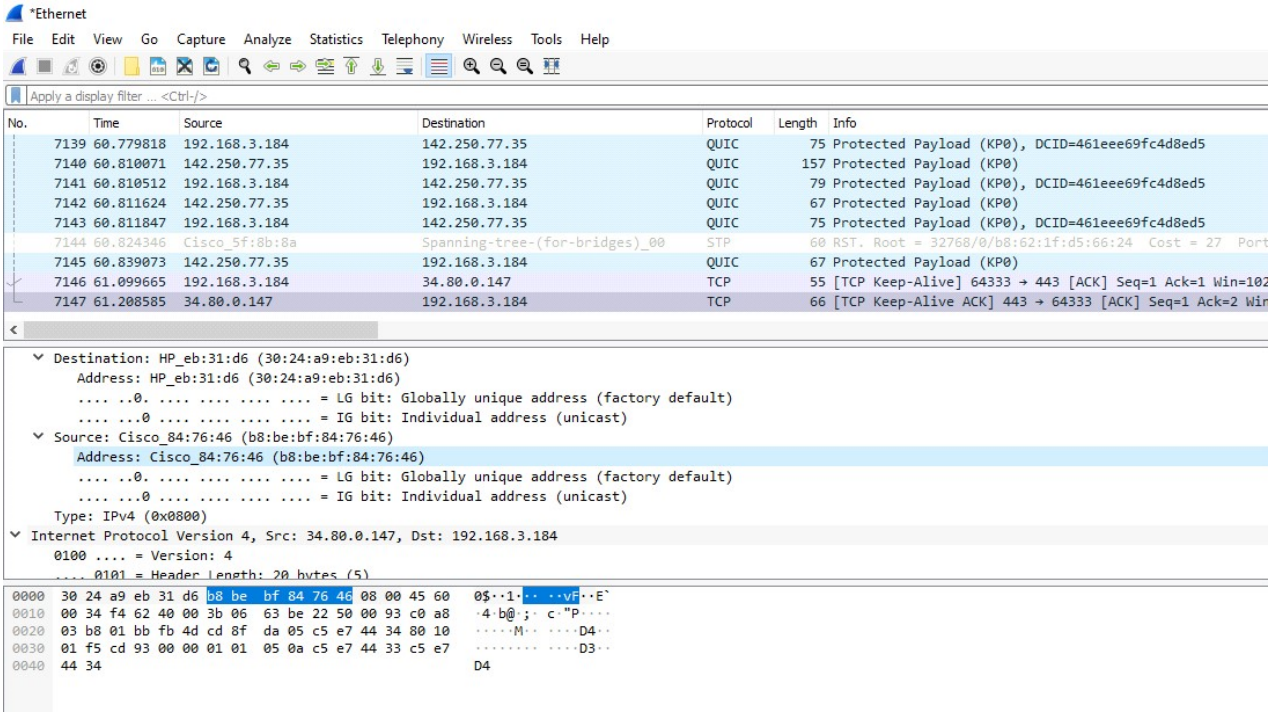
To install Wireshark:

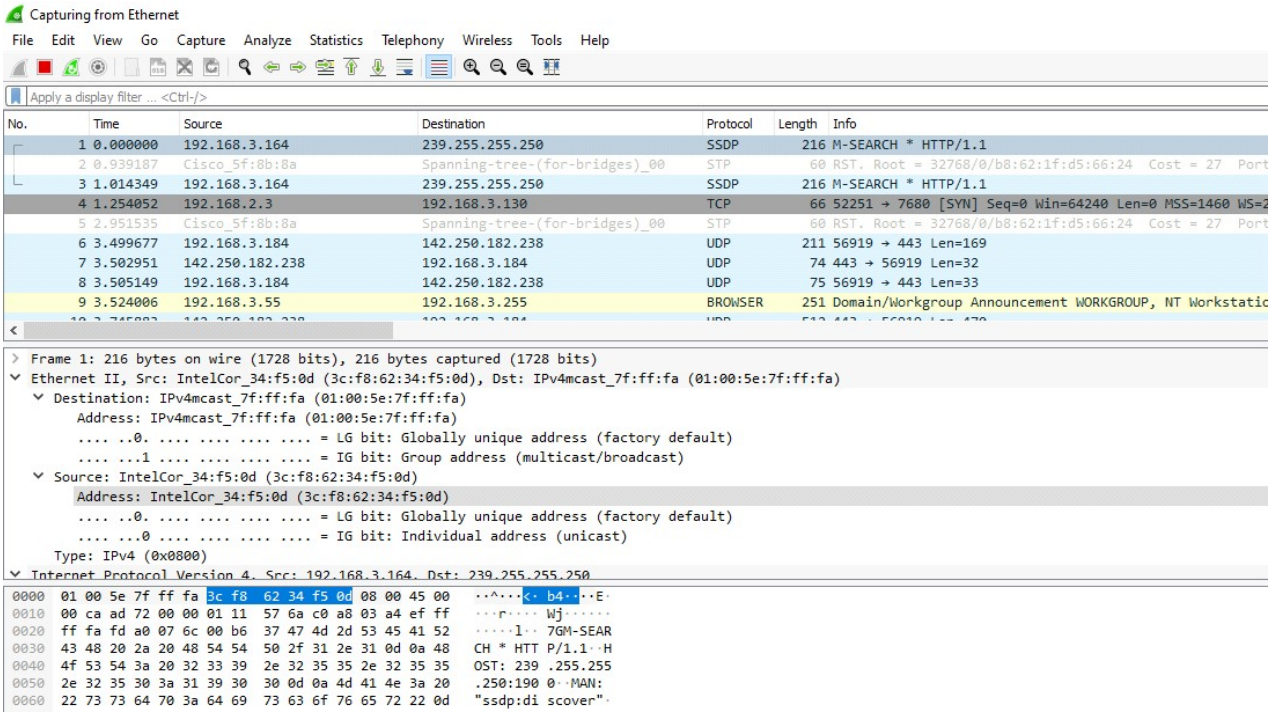
1.Open Windows Explorer.
2.Select the Downloads folder.
3.Locate the version of Wireshark you downloaded in Activity 2. Double-click on the file to open it.
4.If you see a User Account Control dialog box, select Yes to allow the program to make changes to this computer.

5.Select Next > to start the Setup Wizard.
6.Review the license agreement. If you agree, select I Agree to continue.
7.Select Next > to accept the default components.
8.Select the shortcuts you would like to have created. Leave the file extensions selected. Select Next > to continue.
9.Select Next > to accept the default install location.
10.Select Install to begin installation.
11.Select Next > to install WinPcap.
12.Select Next > to start the Setup Wizard.
13.Review the license agreement. If you agree, select I Agree to continue.
14.Select Install to begin installation.
15.Select Finish to complete the installation of WinPcap.
16.Select Next > to continue with the installation of Wireshark.
17.Select Finish to complete the installation of Wireshark.

Conclusion: Learnt how to install and configure Wireshark.







----------X-0-X----------

APIs: Packet Analyzer

Procedure: ***To capture packer: Set up the Packet Capture:***

1. Click View > Wireless Toolbar. The Wireless Toolbar will appear just below the Main toolbar.

2. Use the Wireless Toolbar to configure the desired channel and channel width.

3. Under Capture, click on AirPcap USB wireless capture adapter to select the capture interface.

Note: If the AirPcap isn't listed, press F5 to refresh the list of available packet capture interfaces. The AirPcap has been discontinued by RiverBed and is 802.11n only.

4. Click the Start Capture button to begin the capture.

5. When you are finished capturing, click the Stop button.

•***Saving the Capture:***

1. To save the capture, click File > Save.

2. Name the file, and click Save.

Note: .Pcap and .Pcap-ng are good filetypes to use for the capture if you plan to use Eye P.A. to open the capture.

3. Eye P.A. can now open the capture file.

Theory:

Packet sniffer \ Packet analyzer:

A packet analyzer (also known as a network analyzer, protocol analyzer or packet sniffer or for particular types of networks, an Ethernet sniffer or wireless sniffer) is a computer program or a piece of computer hardware that can intercept and log traffic passing over a digital network or part of a network. As data streams own across the network, the sniffer captures each packet and, if needed, decodes the packet's raw data, showing the values of various fields in the packet, and analyzes its content according to the appropriate RFC or other specifications.

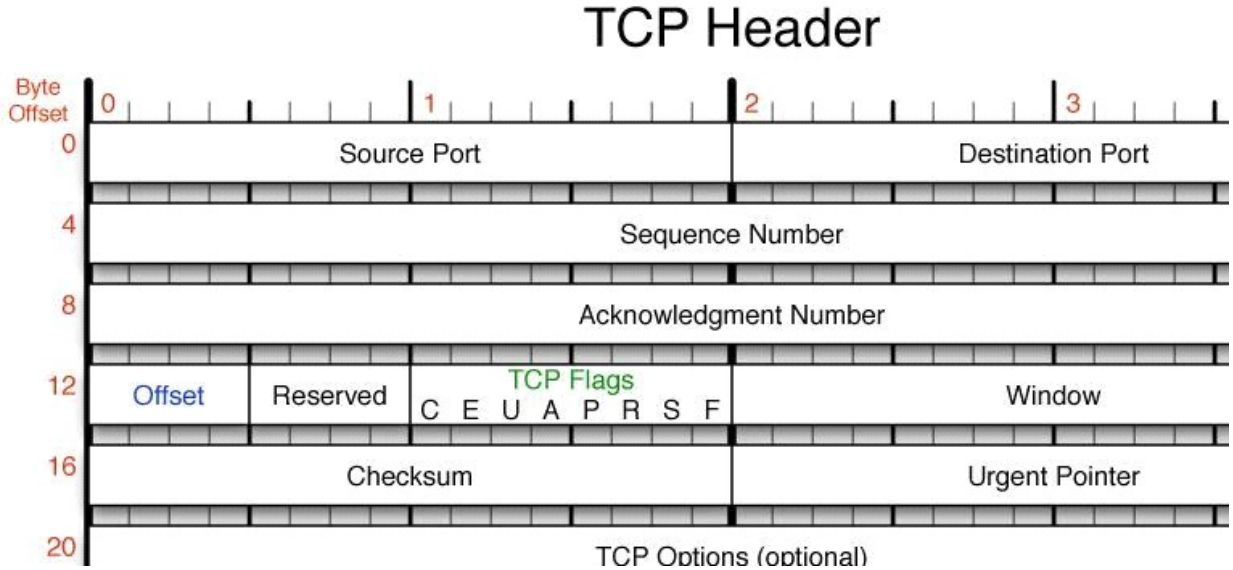
Different types of packet:

1. TCP:

The Transmission Control Protocol (TCP) is one of the core protocols of the Internet protocol suite (IP), and is so common that the entire suite is often called TCP/IP. TCP provides reliable, ordered and error-checked delivery (or notification of failure to deliver) of a stream of octets between programs running on computers connected to a local area network, intranet or the public Internet. It resides at the transport layer. Web browsers use TCP when they connect to servers on the World

Wide Web, and it is used to deliver email and transfer files from one location to another. The protocol corresponds to the transport layer of TCP/IP suite. TCP provides a communication service at an intermediate level between an application program and the Internet Protocol (IP). That is, when an application program desires to send a large chunk of data across the Internet using IP, instead of breaking the data into IP-sized pieces and issuing a series of IP requests, the software can issue a single request to TCP and let TCP handle the

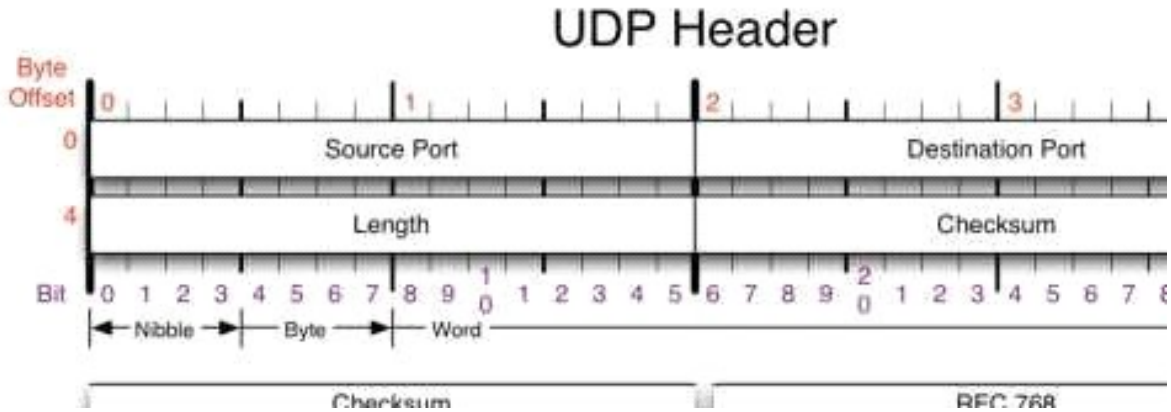
IP works by exchanging pieces of information called packets. A packet is a sequence of octets (bytes) and consists of a header followed by a body. The header describes the packet's source, destination and control information. The body contains the data IP is transmitting. Due to network congestion, traffic load balancing, or other unpredictable network behavior, IP packets can be lost, duplicated, or delivered out of order. TCP detects these problems, requests retransmission of lost data, rearranges out-of-order data, and even helps minimize network congestion to reduce the occurrence of the other problems. If the data still remains undelivered, its source is notified of this failure. Once the TCP receiver has reassembled the sequence of octets originally transmitted, it passes them to the receiving application. Thus, TCP abstracts the application's communication from the underlying networking details. TCP is a reliable stream delivery service that guarantees that all bytes received will be identical with bytes sent and in the correct order. Since packet transfer over many networks is not reliable, a technique known as positive acknowledgment with retransmission is used to guarantee reliability of packet transfers. This fundamental technique requires the receiver to respond with an acknowledgment message as it receives the data. The sender keeps a record of each packet it sends. The sender also maintains a timer from when the packet was sent, and retransmits a packet if the timer expires before the message has been acknowledged. The timer is needed in case a packet gets lost or corrupted.



2. UDP:

The User Datagram Protocol (UDP) is one of the core members of the Internet protocol Suite. UDP uses a simple connectionless transmission model with a minimum of protocol mechanism. It

has no handshaking dialogues, and thus exposes any unreliability of the underlying network protocol to the user's program. There is no guarantee of delivery, ordering, or duplicate protection. UDP provides checksums for data integrity, and port numbers for addressing different functions at the source and destination of the datagram. With UDP, computer applications can send messages, in this case referred to as datagrams, to other hosts on an Internet Protocol (IP) network without prior communications to set up special transmission channels or data paths. UDP is suitable for purposes where error checking and correction is either not necessary or is performed in the application, avoiding the overhead of such processing at the network interface level. Time-sensitive applications often use UDP because dropping packets is preferable to waiting for delayed packets, which may not be an option in a real-time system.



3.ICMP:

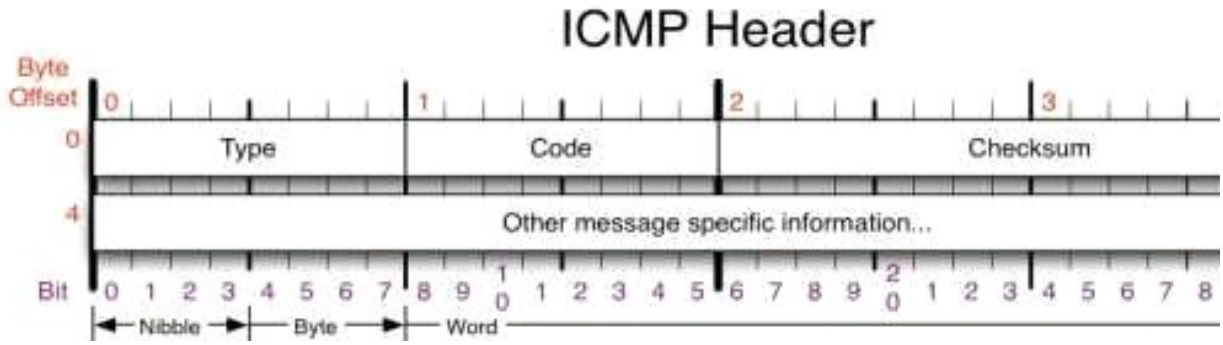
The Internet Control Message Protocol (ICMP) is one of the main protocols of the Internet Protocol Suite. It is used by network devices, like routers, to send error messages indicating, for example, that a requested service is not available or that a host or router could not be reached.

ICMP can also be used to relay query messages. It is assigned protocol number 1.ICMP differs from transport protocols such as TCP and UDP in that it is not typically used to exchange data between systems, nor is it regularly employed by end-user network applications (with the exception of some diagnostic tools like ping and trace route). ICMP for Internet Protocol version 4 (IPv4) is also known as ICMPv4. IPv6 has a similar protocol, ICMPv6. The Internet Control Message Protocol is part of the Internet Protocol Suite, as defined in RFC 792. ICMP messages

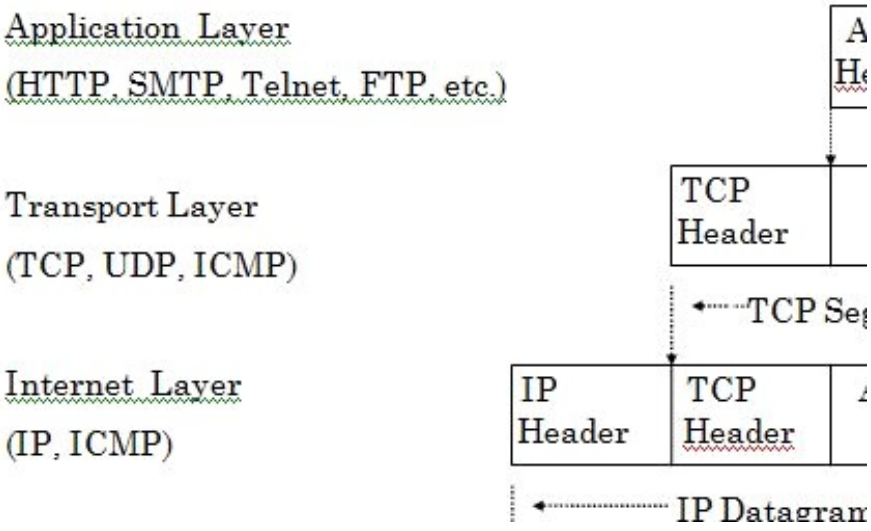
4.IGMP:

The Internet Group Management Protocol (IGMP) is a communications protocol used by hosts and adjacent routers on IP networks to establish multicast group memberships. IGMP is an integral part of IP multicast.IGMP can be used for one-to-many networking applications such as online streaming video and gaming, and allows more efficient use of resources when supporting these

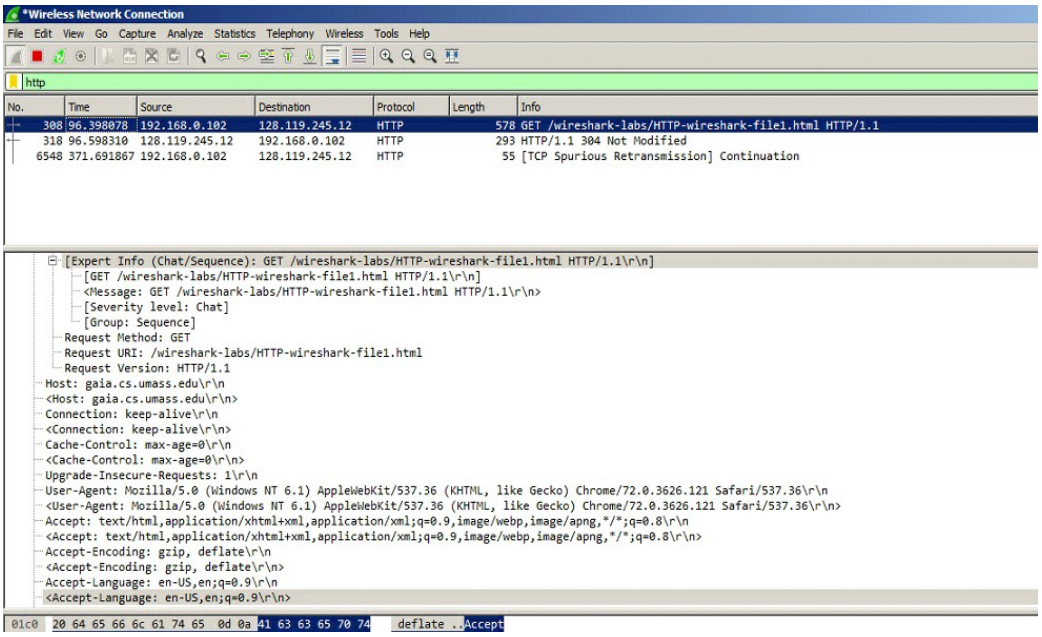
types of applications. IGMP messages are carried in bare IP packets with IP protocol. There is no transport layer used with IGMP messaging, similar to the Internet Control Message Protocol. Membership Queries are sent by multicast routers to determine which multicast addresses are of interest to systems attached to its network. Routers periodically send General Queries to refresh the group membership state for all systems on its network. Group-Specific Queries are used for determining the reception



state for a particular multicast address.



Output:



Conclusion: We have successfully analysed the captured packets from Wireshark.