Experiment No. 4

Title: Exploring Application layer protocols

Batch: B1 **Roll No.: 16010420133** **Experiment No.4**

Aim: To explore the application layer protocols using Wireshark.

Resources needed: Internet, Wireshark software (downloaded from the official site)

Theory

Background of Wireshark

Wireshark is a network packet analyzer. Any network packet analyzer will try to capture network packets and will try to display that packet data as detailed as possible in human readable format. Wireshark is an open source software project, and is released under the GNU General Public License (GPL). We can freely use Wireshark on any number of computers, without worrying about license keys. In addition, all source code is freely available under the GPL. Because of that, it is very easy for people to add new protocols to Wireshark, either as plug-in, or built into the source code. In the past, such tools were either very expensive, proprietary. However, with the advent of Wire-shark, all that has changed. Wireshark is perhaps one of the best open source packet analyzers available today.

What Wireshark is not

Here are some things Wireshark does not provide:

1. Wireshark isn't an intrusion detection system. It will not warn us when someone does strange things on our network that he/she isn't allowed to do. However, if strange things happen, Wireshark might help you figure out what is really going on.
2. Wireshark will not manipulate things on the network, it will only "measure" things from it. Wireshark doesn't send packets on the network or do other active things.

Applications

Here are some applications. Many people use Wireshark for doing following things,

- Network administrators use it to ***troubleshoot network problems.***
- Network security engineers use it to ***examine security problems (Network Forensics.)***
- Developers use it to ***debug protocol implementations.***
- People use it to ***learn network protocol*** **internals/analysis.**

Beside these examples Wireshark can be helpful in many other situations too.

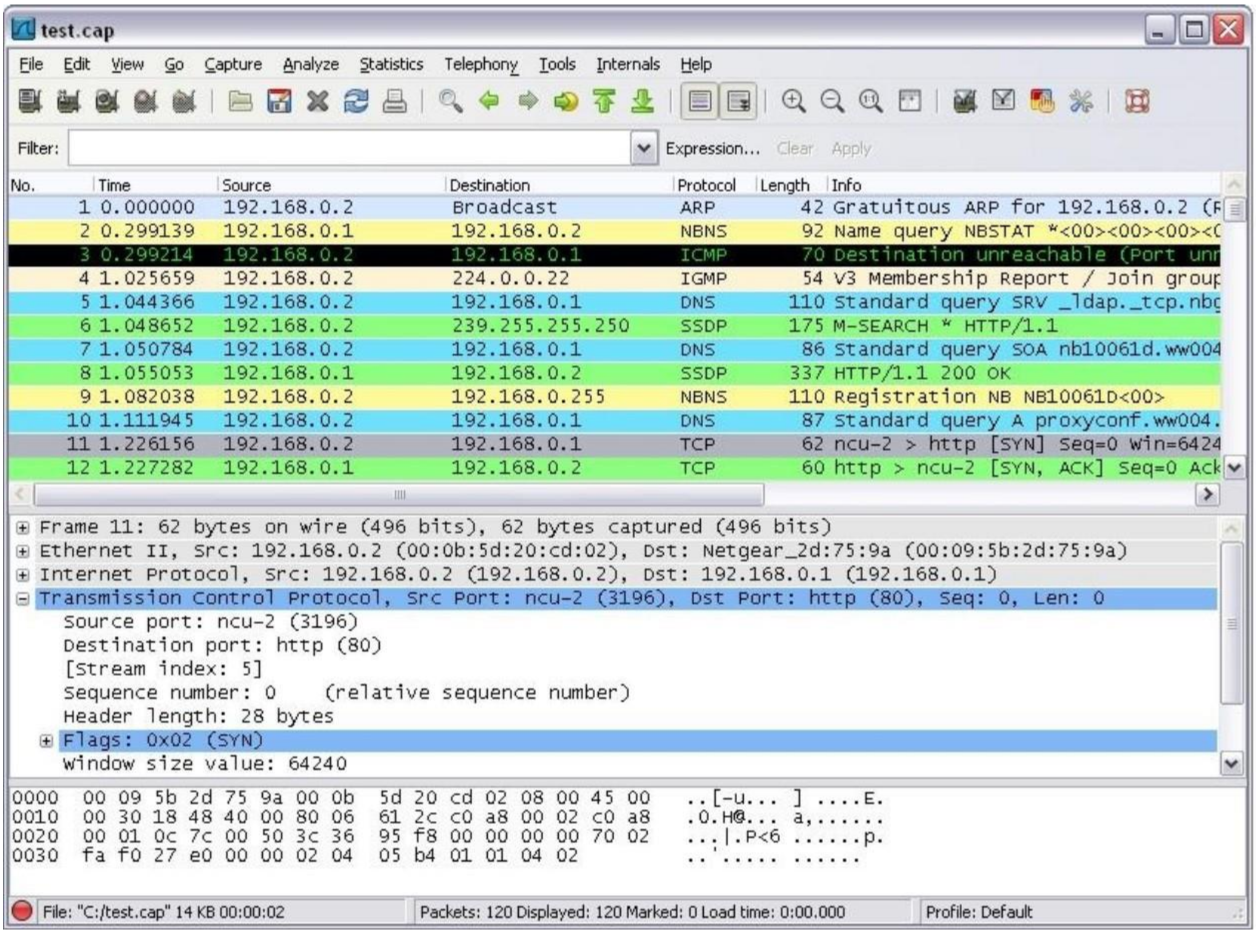
The following are some of the features Wireshark has:

- Available for*UNIX* and *Windows operating systems.*
- *Capture* live packet data from a chosen network interface.
- *Open* files containing packet data captured with tcpdump/WinDump and a number of 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!*

However, to really appreciate its power we have to start

using it. Here is a snapshot of Wireshark main menu.



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.0.2	Broadcast	ARP	42	Gratuitous ARP for 192.168.0.2 (R
2	0.299139	192.168.0.1	192.168.0.2	NBNS	92	Name query NBSTAT *<00><00><00><C
3	0.299214	192.168.0.2	192.168.0.1	ICMP	70	Destination unreachable (Port unr
4	1.025659	192.168.0.2	224.0.0.22	IGMP	54	V3 Membership Report / Join grour
5	1.044366	192.168.0.2	192.168.0.1	DNS	110	Standard query SRV _ldap._tcp.nbg
6	1.048652	192.168.0.2	239.255.255.250	SSDP	175	M-SEARCH * HTTP/1.1
7	1.050784	192.168.0.2	192.168.0.1	DNS	86	Standard query SOA nb10061d.ww004
8	1.055053	192.168.0.1	192.168.0.2	SSDP	337	HTTP/1.1 200 OK
9	1.082038	192.168.0.2	192.168.0.255	NBNS	110	Registration NB NB10061D<00>
10	1.111945	192.168.0.2	192.168.0.1	DNS	87	Standard query A proxyconf.ww004.
11	1.226156	192.168.0.2	192.168.0.1	TCP	62	ncu-2 > http [SYN] Seq=0 Win=6424
12	1.227282	192.168.0.1	192.168.0.2	TCP	60	http > ncu-2 [SYN, ACK] Seq=0 Ack

```
⊞ Frame 11: 62 bytes on wire (496 bits), 62 bytes captured (496 bits)
⊞ Ethernet II, Src: 192.168.0.2 (00:0b:5d:20:cd:02), Dst: Netgear_2d:75:9a (00:09:5b:2d:75:9a)
⊞ Internet Protocol, Src: 192.168.0.2 (192.168.0.2), Dst: 192.168.0.1 (192.168.0.1)
⊟ Transmission Control Protocol, Src Port: ncu-2 (3196), Dst Port: http (80), Seq: 0, Len: 0
    Source port: ncu-2 (3196)
    Destination port: http (80)
    [Stream index: 5]
    Sequence number: 0    (relative sequence number)
    Header length: 28 bytes
  ⊞ Flags: 0x02 (SYN)
    Window size value: 64240
```

```
0000  00 09 5b 2d 75 9a 00 0b  5d 20 cd 02 08 00 45 00   ..[-u... ] ....E.
0010  00 30 18 48 40 00 80 06  61 2c c0 a8 00 02 c0 a8   .0.H@... a,......
0020  00 01 0c 7c 00 50 3c 36  95 f8 00 00 00 00 70 02   ...|.P<6 ......p.
0030  fa f0 27 e0 00 00 02 04  05 b4 01 01 04 02         ..'..... ......
```

File: "C:/test.cap" 14 KB 00:00:02 | Packets: 120 Displayed: 120 Marked: 0 Load time: 0:00.000 | Profile: Default

Most important menus are : 1) Capture 2) Analyze 3) Statistics
Students are expected to explore all these menus and sub-menus in details.

Wireshark can capture traffic from many different network media types including wireless LAN as well. Which media types are supported, depends on many things like the operating system we are using and the hardware support.

Physical Interfaces support

A. ATM - capture ATM traffic

B. Bluetooth- capture Bluetooth traffic .

C. Cisco HDLC links - capture on synchronous links using Cisco HDLC encapsulation. D. Ethernet- capture on different topologies, including

switched networks.
E. Framerelay – captures framerelay traffic.
F. IrDA capture IrDA traffic - currently limited to Linux.
G. PPP links - capture on dial-up lines, ISDN connections and PPP-over-Ethernet (PPPoe, e.g. ADSL)
H. Tokenring - capture on Tokenring adapters, promiscuous mode and switched networks
I. USB- capture of raw USB traffic
J. WLAN- capture on 802.11 (WLAN, Wi-Fi) interfaces, including "monitor mode" , raw 802.11 headers and radio information

Virtual interfaces :

1. Loopback - capture traffic from a machine to itself, including the IP address 127.0.0.1
2. Pipes - use UNIX pipes to capture from other applications (even remote!)
3. VLAN – capture VLAN traffic, including VLAN tags.

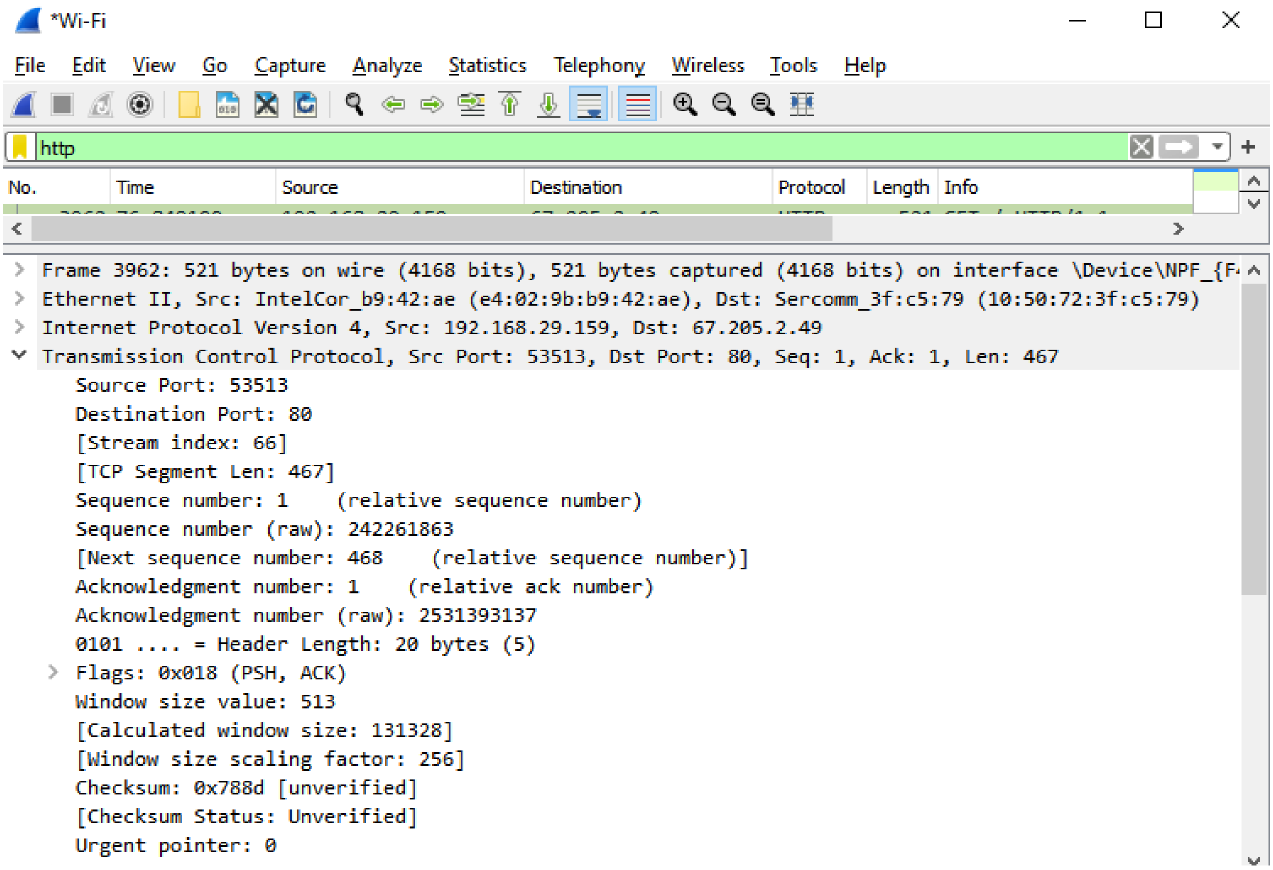
In addition to this, Wireshark can do following things.

1. Import files from many other capture programs.
2. Wireshark can open packets captured from a large number of other capture programs.
3. Export files for many other capture programs.
4. Wireshark can save packets captured in a large number of formats of other capture programs.
5. Can be used as a protocol decoder.
6. Few application layer protocols(HTTP,FTP,SMTP etc..) make use of TCP to deliver their purpose so you see tcp as their backbone.

Procedure:

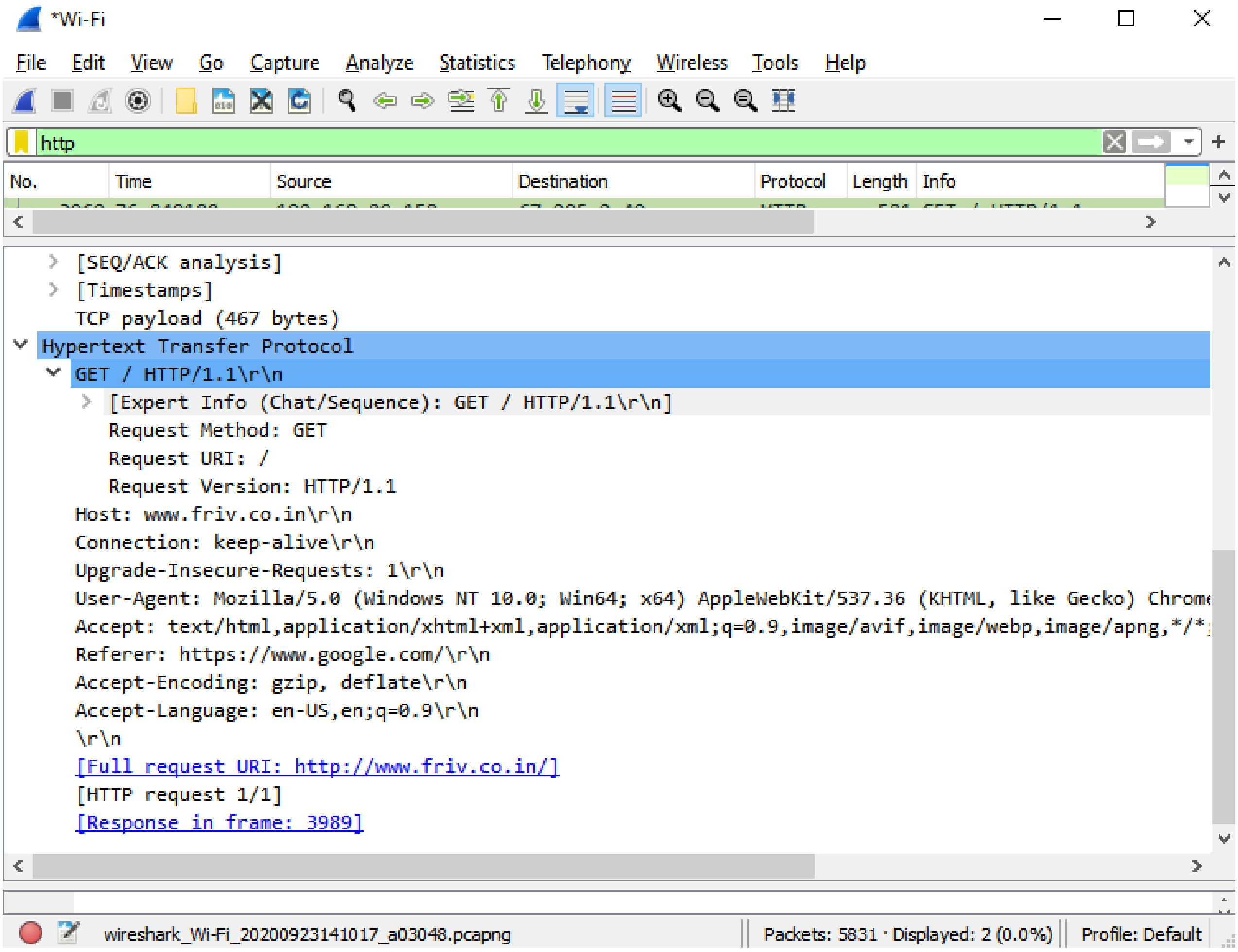
1. Start the machine as an administrator.
2. Start internet.
3. Go to the official website of Wireshark . (www.wireshark.org) and download the old stable version of Wireshark for 32 bit windows operating system.
4. After successful installation you will get the blue icon of Wireshark on the desktop.
5. Click on the icon and start the software.
6. Choose an interface and start capturing the packets.
7. Study the packet details of all the protocols of Application layer(DNS,HTTP and FTP etc.)
8. Perform the statistics for a particular protocol. (Every student should perform for different protocol.)
10. Show the output to the teacher and get it approved.

Activity:
Identify one application layer protocol , study analyse and interpret the same along with snapshots.



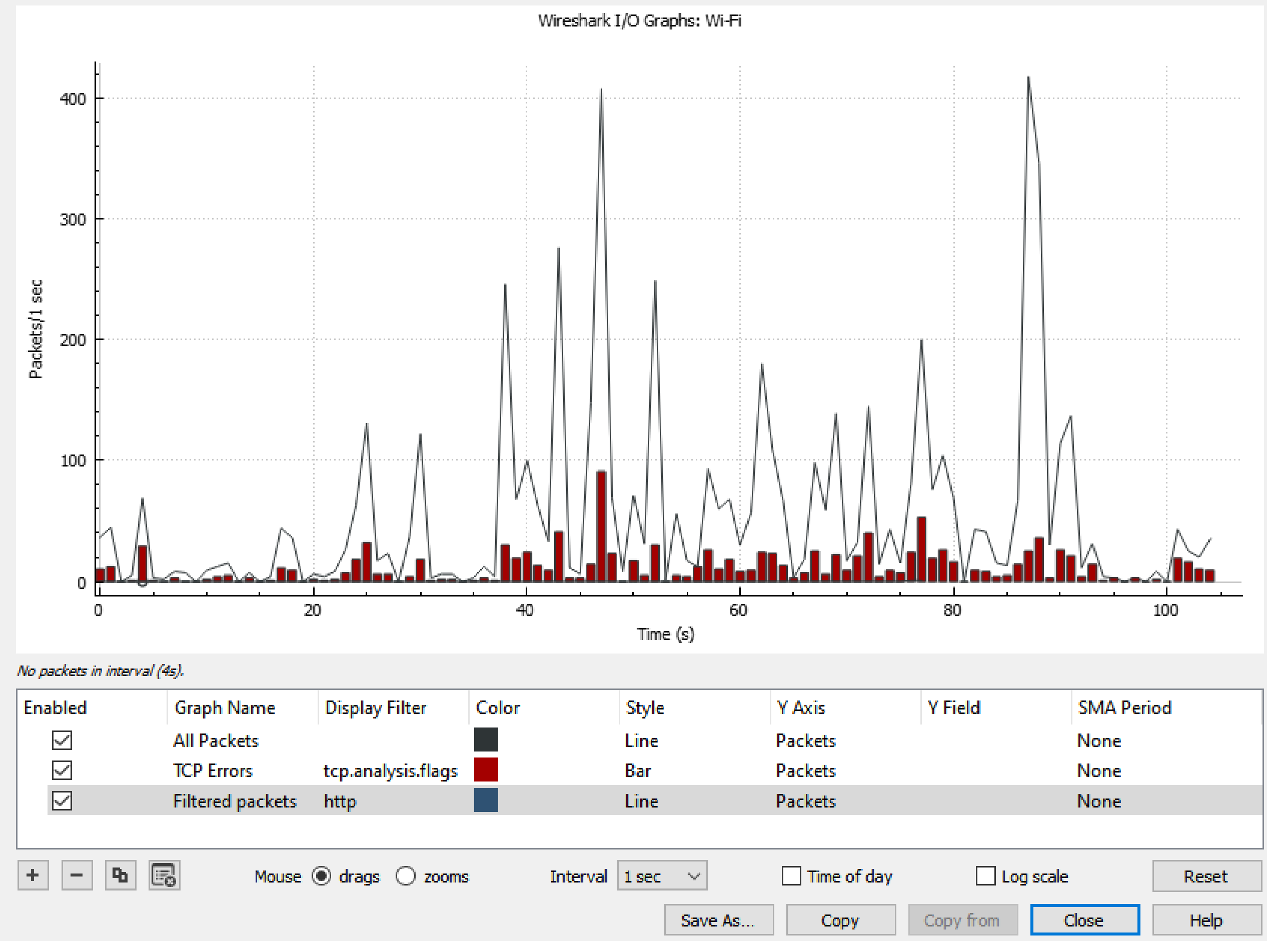
```
> Frame 3962: 521 bytes on wire (4168 bits), 521 bytes captured (4168 bits) on interface \Device\NPF_{F
> Ethernet II, Src: IntelCor_b9:42:ae (e4:02:9b:b9:42:ae), Dst: Sercomm_3f:c5:79 (10:50:72:3f:c5:79)
> Internet Protocol Version 4, Src: 192.168.29.159, Dst: 67.205.2.49
v Transmission Control Protocol, Src Port: 53513, Dst Port: 80, Seq: 1, Ack: 1, Len: 467
     Source Port: 53513
     Destination Port: 80
     [Stream index: 66]
     [TCP Segment Len: 467]
     Sequence number: 1    (relative sequence number)
     Sequence number (raw): 242261863
     [Next sequence number: 468    (relative sequence number)]
     Acknowledgment number: 1    (relative ack number)
     Acknowledgment number (raw): 2531393137
     0101 .... = Header Length: 20 bytes (5)
   > Flags: 0x018 (PSH, ACK)
     Window size value: 513
     [Calculated window size: 131328]
     [Window size scaling factor: 256]
     Checksum: 0x788d [unverified]
     [Checksum Status: Unverified]
     Urgent pointer: 0
```

Opened friv.com
Source Port: 53513
Destination Port: 80
Source: 192.168.29.159
Destination: 67.205.2.49
Protocol: Hypertext Transfer Protocol
Request: GET
Full request URI: http://www.friv.co.in/
Host: www.friv.co.in\r\n
Packet number: 3989



```
  > [SEQ/ACK analysis]
  > [Timestamps]
    TCP payload (467 bytes)
v Hypertext Transfer Protocol
  v GET / HTTP/1.1\r\n
    > [Expert Info (Chat/Sequence): GET / HTTP/1.1\r\n]
      Request Method: GET
      Request URI: /
      Request Version: HTTP/1.1
    Host: www.friv.co.in\r\n
    Connection: keep-alive\r\n
    Upgrade-Insecure-Requests: 1\r\n
    User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*
    Referer: https://www.google.com/\r\n
    Accept-Encoding: gzip, deflate\r\n
    Accept-Language: en-US,en;q=0.9\r\n
    \r\n
    [Full request URI: http://www.friv.co.in/]
    [HTTP request 1/1]
    [Response in frame: 3989]
```

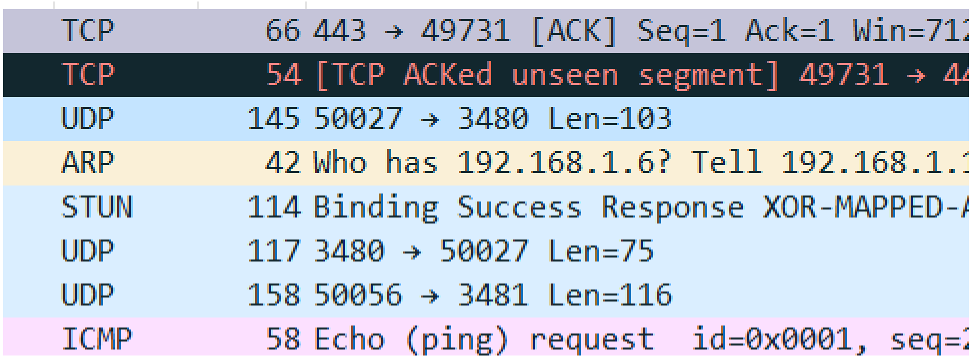
wireshark_Wi-Fi_20200923141017_a03048.pcapng | Packets: 5831 · Displayed: 2 (0.0%) | Profile: Default



No packets in interval (4s).

Enabled	Graph Name	Display Filter	Color	Style	Y Axis	Y Field	SMA Period
☑	All Packets		⬛	Line	Packets		None
☑	TCP Errors	tcp.analysis.flags	🟥	Bar	Packets		None
☑	Filtered packets	http	🟦	Line	Packets		None

Mouse ● drags ○ zooms Interval 1 sec □ Time of day □ Log scale



```
TCP      66 443 → 49731 [ACK] Seq=1 Ack=1 Win=712
TCP      54 [TCP ACKed unseen segment] 49731 → 44
UDP     145 50027 → 3480 Len=103
ARP      42 Who has 192.168.1.6? Tell 192.168.1.1
STUN    114 Binding Success Response XOR-MAPPED-A
UDP     117 3480 → 50027 Len=75
UDP     158 50056 → 3481 Len=116
ICMP     58 Echo (ping) request  id=0x0001, seq=2
```

Questions:

1. What is MIME? List the MIME headers.

Multipurpose Internet Mail Extensions (MIME) is an Internet standard that extends the format of email messages to support text in character sets other than ASCII, as well as attachments of audio, video, images, and application programs.
MIME defines five media types for discrete content: text, image , audio , video , and application
The following headers are defined in MIME:

- MIME-Version
- Content-Type
- Content-Transfer-Encoding
- Content-ID
- Content-Description
- Content-Disposition

2. Why do we need POP3 for electronic mail?

→ Post Office Protocol version 3 (POP3) is a standard mail protocol used to receive emails from a remote server to a local email client. If you use POP3, your messages are stored on your local computer, which reduces the space your email account uses on your web server.

→ POP3 allows you to download email messages on your local computer and read them even when you are offline.

3. What are the FTP transmission modes?

→ Transmission mode refers to the mechanism of transferring of data between two devices connected over a network.

→ FTP defines three different transmission modes (also called transfer modes) that specify exactly how data is sent from one device to another over an opened data channel:
stream mode, block mode, and compressed mode.

4. Name the three components of a browser.

→ A browser consists of a controller, client program, and interpreters.

5.What is URL and what are its components?

→ URL stands for Uniform Resource Locator. A URL is nothing more than the address of a given unique resource on the Web.

→ It is the mechanism used by browsers to retrieve any published resource on the web.

→ Example:
http://www.example.com:80/path/to/myfile.html?key1=value1&key2=value2#SomewhereInTheDocument

→ It's components are:

• Protocol:	http
• domain name:	www.example.com
• port:	:80

- path to the resource : path/to/myfile.html
- parameters : key1=value1&key2=value2
- anchor:#SomewhereInTheDocument

Outcomes: To enumerate the layers of the OSI/TCP model, their functions and ptotocols (Application layer)

Conclusion: We explored the application layer protocols using Wireshark.

Grade: AA / AB / BB / BC / CC / CD /DD

Signature of faculty in-charge with date

References:

Books and Websites:

- Behrouz A Forouzan, Data Communication and networking,Tata Mc Graw hill, India, 4th Edition
- http://www.wireshark.org
- Wireshark user manual