Batch: D2 Roll No.:16010123325

Experiment / assignment / tutorial No.10

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

Signature of the Staff In-charge with date

Experiment No.:10

TITLE: Study of Packet Analyzer tool: Wireshark

AIM: To study and analyse various Protocols using Packet Analyzer tool: Wireshark

Expected Outcome of Experiment:

CO5: Describe various features and operations of application layer protocols such as Telnet, HTTP, DNS, SMTP.

Books/ Journals/ Websites referred:

1. A. S. Tanenbaum, “Computer Networks”, Pearson Education, Fourth Edition
2. B. A. Forouzan, “Data Communications and Networking”, TMH, Fourth Edition

Pre Lab/ Prior Concepts:

IPv4 Addressing, Subnetting, Link State Protocol, Router configuration Commands

New Concepts to be learned: Packet Analyzer tool: Wireshark.

THEORY:

Wireshark Overview:

Wireshark is a free and open-source network protocol analyzer used to capture and examine data packets in real-time. It helps network engineers and students analyze the functioning of different network layers and protocols.

Working Principle:

Wireshark captures data packets traveling over a network interface and displays detailed information such as source/destination IP addresses, MAC addresses, port numbers, and protocol types. Each packet can be expanded to view headers from all OSI layers.

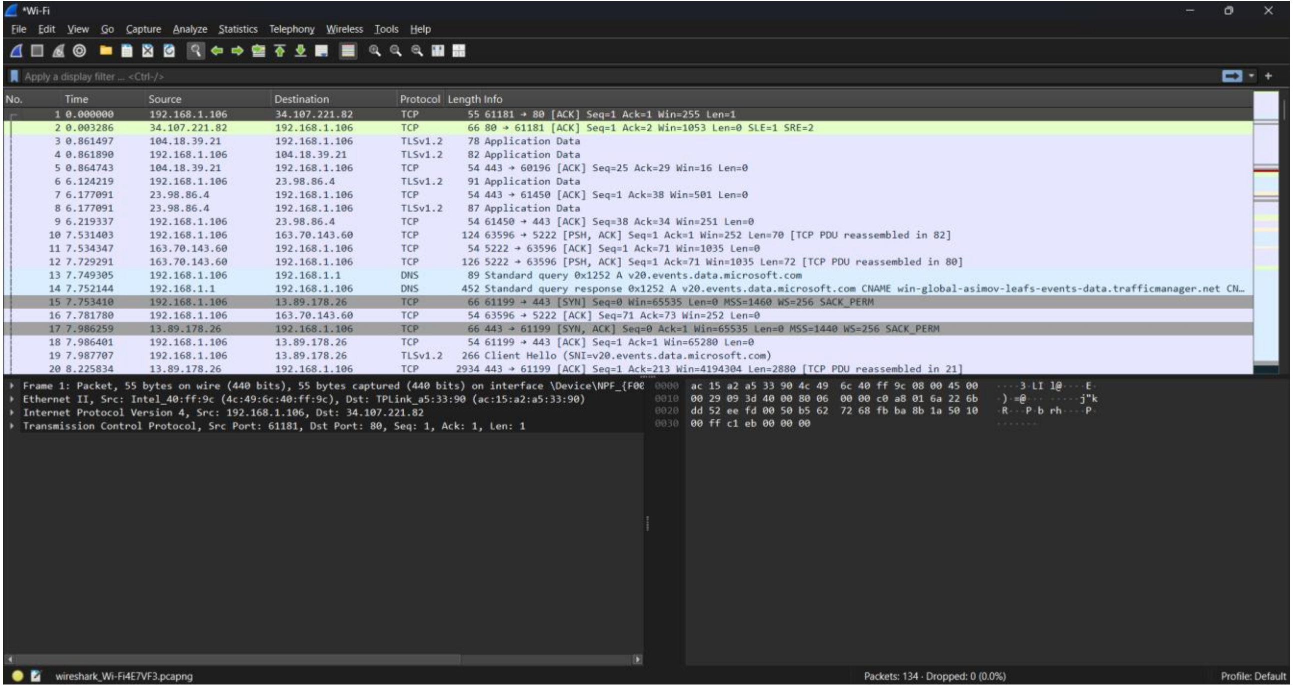
Protocols Analysed:

- Application Layer: HTTP, DNS, SMTP, FTP, Telnet
- Transport Layer: TCP, UDP
- Network Layer: IP, ICMP
- Data Link Layer: Ethernet (MAC addresses)

Uses of Wireshark:

- Network troubleshooting and performance analysis
- Studying protocol behavior (handshakes, queries, responses)
- Detecting network anomalies and attacks
- Learning how real data communication occurs between devices

IMPLEMENTATION:

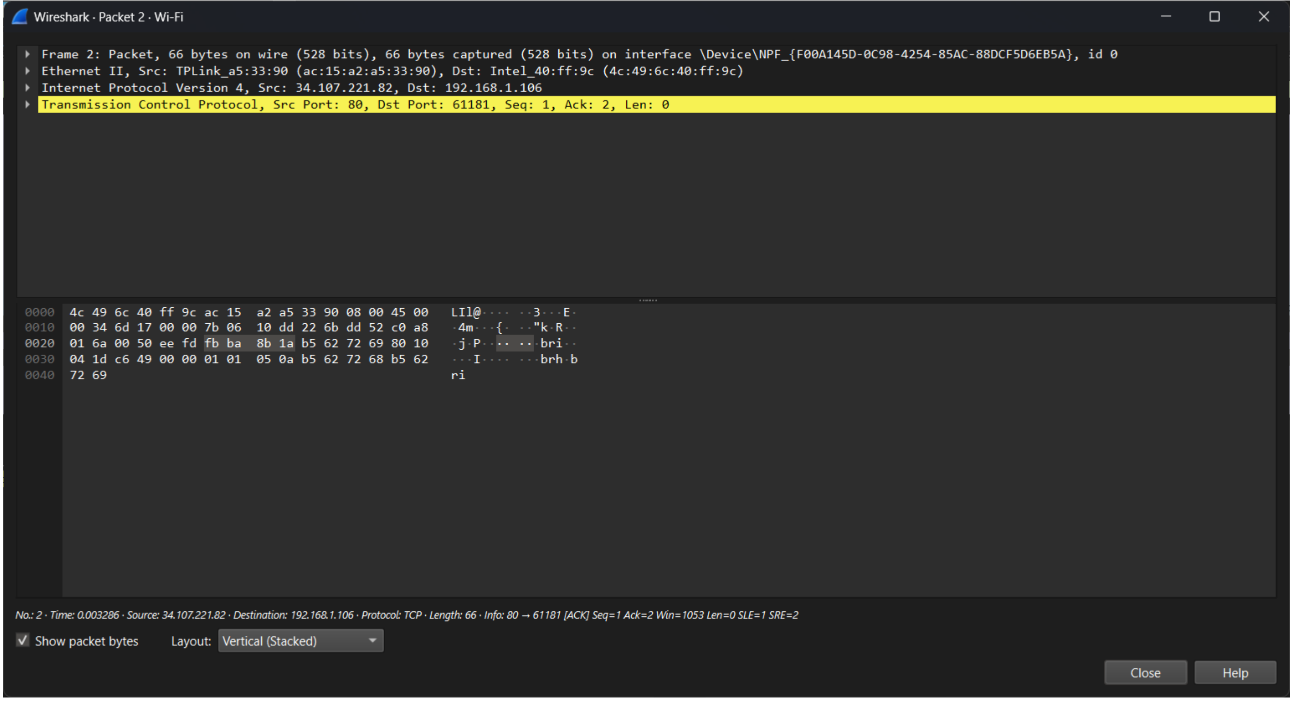

*Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>

```
No.  Time      Source          Destination     Protocol Length Info
   1 0.000000  192.168.1.106   34.107.221.82   TCP        55 61181 → 80 [ACK] Seq=1 Ack=1 Win=255 Len=1
   2 0.003286  34.107.221.82   192.168.1.106   TCP        66 80 → 61181 [ACK] Seq=1 Ack=2 Win=1053 Len=0 SLE=1 SRE=2
   3 0.861497  104.18.39.21    192.168.1.106   TLSv1.2    78 Application Data
   4 0.861890  192.168.1.106   104.18.39.21    TLSv1.2    82 Application Data
   5 0.864743  104.18.39.21    192.168.1.106   TCP        54 443 → 60196 [ACK] Seq=25 Ack=29 Win=16 Len=0
   6 6.124219  192.168.1.106   23.98.86.4      TLSv1.2    91 Application Data
   7 6.177091  23.98.86.4      192.168.1.106   TCP        54 443 → 61450 [ACK] Seq=1 Ack=38 Win=501 Len=0
   8 6.177091  23.98.86.4      192.168.1.106   TLSv1.2    87 Application Data
   9 6.219337  192.168.1.106   23.98.86.4      TCP        54 61450 → 443 [ACK] Seq=38 Ack=34 Win=251 Len=0
  10 7.531403  192.168.1.106   163.70.143.60   TCP       124 63596 → 5222 [PSH, ACK] Seq=1 Ack=1 Win=252 Len=70 [TCP PDU reassembled in 82]
  11 7.534347  163.70.143.60   192.168.1.106   TCP        54 5222 → 63596 [ACK] Seq=1 Ack=71 Win=1035 Len=0
  12 7.729291  163.70.143.60   192.168.1.106   TCP       126 5222 → 63596 [PSH, ACK] Seq=1 Ack=71 Win=1035 Len=72 [TCP PDU reassembled in 80]
  13 7.749305  192.168.1.106   192.168.1.1     DNS        89 Standard query 0x1252 A v20.events.data.microsoft.com
  14 7.752144  192.168.1.1     192.168.1.106   DNS       452 Standard query response 0x1252 A v20.events.data.microsoft.com CNAME win-global-asimov-leafs-events-data.trafficmanager.net CN…
  15 7.753410  192.168.1.106   13.89.178.26    TCP        66 61199 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
  16 7.781780  192.168.1.106   163.70.143.60   TCP        54 63596 → 5222 [ACK] Seq=71 Ack=73 Win=252 Len=0
  17 7.986259  13.89.178.26    192.168.1.106   TCP        66 443 → 61199 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1440 WS=256 SACK_PERM
  18 7.986401  192.168.1.106   13.89.178.26    TCP        54 61199 → 443 [ACK] Seq=1 Ack=1 Win=65280 Len=0
  19 7.987707  192.168.1.106   13.89.178.26    TLSv1.2   266 Client Hello (SNI=v20.events.data.microsoft.com)
  20 8.225834  13.89.178.26    192.168.1.106   TCP      2934 443 → 61199 [ACK] Seq=1 Ack=213 Win=4194304 Len=2880 [TCP PDU reassembled in 21]
```

```
› Frame 1: Packet, 55 bytes on wire (440 bits), 55 bytes captured (440 bits) on interface \Device\NPF_{F0€
› Ethernet II, Src: Intel_40:ff:9c (4c:49:6c:40:ff:9c), Dst: TPLink_a5:33:90 (ac:15:a2:a5:33:90)
› Internet Protocol Version 4, Src: 192.168.1.106, Dst: 34.107.221.82
› Transmission Control Protocol, Src Port: 61181, Dst Port: 80, Seq: 1, Ack: 1, Len: 1
```

```
0000  ac 15 a2 a5 33 90 4c 49  6c 40 ff 9c 08 00 45 00   ····3·LI l@····E·
0010  00 29 09 3d 40 00 80 06  00 00 c0 a8 01 6a 22 6b   ·)·=@··· ·····j"k
0020  dd 52 ee fd 00 50 b5 62  72 68 fb ba 8b 1a 50 10   ·R···P·b rh····P·
0030  00 ff c1 eb 00 00 00                               ·······
```

wireshark_Wi-Fi4E7VF3.pcapng   Packets: 134 · Dropped: 0 (0.0%)   Profile: Default

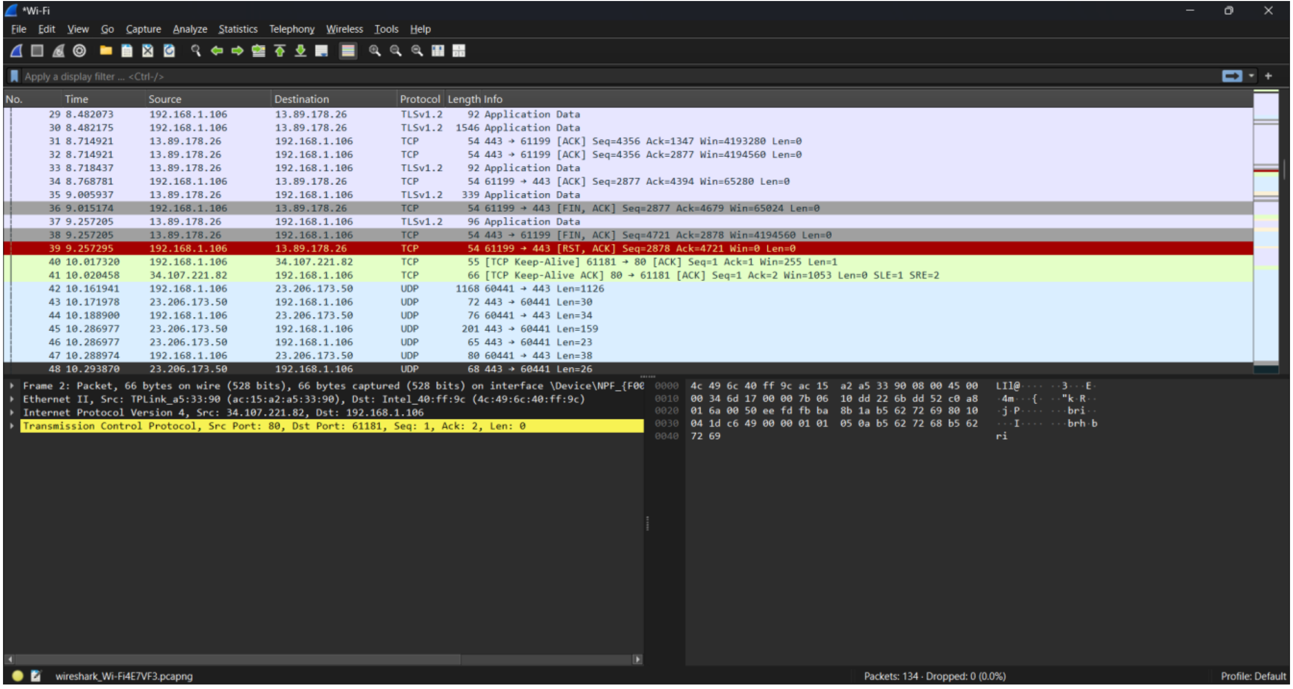


```
› Frame 2: Packet, 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface \Device\NPF_{F00A145D-0C98-4254-85AC-88DCF5D6EB5A}, id 0
› Ethernet II, Src: TPLink_a5:33:90 (ac:15:a2:a5:33:90), Dst: Intel_40:ff:9c (4c:49:6c:40:ff:9c)
› Internet Protocol Version 4, Src: 34.107.221.82, Dst: 192.168.1.106
› Transmission Control Protocol, Src Port: 80, Dst Port: 61181, Seq: 1, Ack: 2, Len: 0
```

```
0000  4c 49 6c 40 ff 9c ac 15  a2 a5 33 90 08 00 45 00   LIl@···· ··3···E·
0010  00 34 6d 17 00 00 7b 06  10 dd 22 6b dd 52 c0 a8   ·4m···{· ··"k·R··
0020  01 6a 00 50 ee fd fb ba  8b 1a b5 62 72 69 80 10   ·j·P···· ···bri··
0030  04 1d c6 49 00 00 01 01  05 0a b5 62 72 68 b5 62   ···I···· ···brh·b
0040  72 69                                              ri
```

No.: 2 · Time: 0.003286 · Source: 34.107.221.82 · Destination: 192.168.1.106 · Protocol: TCP · Length: 66 · Info: 80 → 61181 [ACK] Seq=1 Ack=2 Win=1053 Len=0 SLE=1 SRE=2

☑ Show packet bytes   Layout: Vertical (Stacked)


*Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>

```
No.  Time       Source          Destination     Protocol Length Info
  29 8.482073   192.168.1.106   13.89.178.26    TLSv1.2    92 Application Data
  30 8.482175   192.168.1.106   13.89.178.26    TLSv1.2  1546 Application Data
  31 8.714921   13.89.178.26    192.168.1.106   TCP        54 443 → 61199 [ACK] Seq=4356 Ack=1347 Win=4193280 Len=0
  32 8.714921   13.89.178.26    192.168.1.106   TCP        54 443 → 61199 [ACK] Seq=4356 Ack=2877 Win=4194560 Len=0
  33 8.718437   13.89.178.26    192.168.1.106   TLSv1.2    92 Application Data
  34 8.768781   192.168.1.106   13.89.178.26    TCP        54 61199 → 443 [ACK] Seq=2877 Ack=4394 Win=65280 Len=0
  35 9.005937   13.89.178.26    192.168.1.106   TLSv1.2   339 Application Data
  36 9.015174   192.168.1.106   13.89.178.26    TCP        54 61199 → 443 [FIN, ACK] Seq=2877 Ack=4679 Win=65024 Len=0
  37 9.257205   13.89.178.26    192.168.1.106   TLSv1.2    96 Application Data
  38 9.257205   13.89.178.26    192.168.1.106   TCP        54 443 → 61199 [FIN, ACK] Seq=4721 Ack=2878 Win=4194560 Len=0
  39 9.257295   192.168.1.106   13.89.178.26    TCP        54 61199 → 443 [RST, ACK] Seq=2878 Ack=4721 Win=0 Len=0
  40 10.017320  192.168.1.106   34.107.221.82   TCP        55 [TCP Keep-Alive] 61181 → 80 [ACK] Seq=1 Ack=1 Win=255 Len=1
  41 10.020458  34.107.221.82   192.168.1.106   TCP        66 [TCP Keep-Alive ACK] 80 → 61181 [ACK] Seq=1 Ack=2 Win=1053 Len=0 SLE=1 SRE=2
  42 10.161941  192.168.1.106   23.206.173.50   UDP      1168 60441 → 443 Len=1126
  43 10.171978  23.206.173.50   192.168.1.106   UDP        72 443 → 60441 Len=30
  44 10.188900  192.168.1.106   23.206.173.50   UDP        76 60441 → 443 Len=34
  45 10.286977  23.206.173.50   192.168.1.106   UDP       201 443 → 60441 Len=159
  46 10.286977  23.206.173.50   192.168.1.106   UDP        65 443 → 60441 Len=23
  47 10.288974  192.168.1.106   23.206.173.50   UDP        80 60441 → 443 Len=38
  48 10.293870  23.206.173.50   192.168.1.106   UDP        68 443 → 60441 Len=26
```

```
› Frame 2: Packet, 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface \Device\NPF_{F0€
› Ethernet II, Src: TPLink_a5:33:90 (ac:15:a2:a5:33:90), Dst: Intel_40:ff:9c (4c:49:6c:40:ff:9c)
› Internet Protocol Version 4, Src: 34.107.221.82, Dst: 192.168.1.106
› Transmission Control Protocol, Src Port: 80, Dst Port: 61181, Seq: 1, Ack: 2, Len: 0
```

```
0000  4c 49 6c 40 ff 9c ac 15  a2 a5 33 90 08 00 45 00   LIl@···· ··3···E·
0010  00 34 6d 17 00 00 7b 06  10 dd 22 6b dd 52 c0 a8   ·4m···{· ··"k·R··
0020  01 6a 00 50 ee fd fb ba  8b 1a b5 62 72 69 80 10   ·j·P···· ···bri··
0030  04 1d c6 49 00 00 01 01  05 0a b5 62 72 68 b5 62   ···I···· ···brh·b
0040  72 69                                              ri
```

wireshark_Wi-Fi4E7VF3.pcapng   Packets: 134 · Dropped: 0 (0.0%)   Profile: Default

CONCLUSION:

In this experiment, we successfully used Wireshark to capture and analyze various network protocols at different OSI layers. We observed real-time data exchange, including HTTP, DNS, TCP, and ICMP packets. This helped us understand how application layer protocols function and how packets are transmitted, routed, and received in a network.

Date: __________   Signature of faculty in-charge