CS471 – Web Technologies (Laboratory)	Qassim University College of Computer جامعة القصيم كلية الحاسب	Lab Week 2 / The Internet Protocols

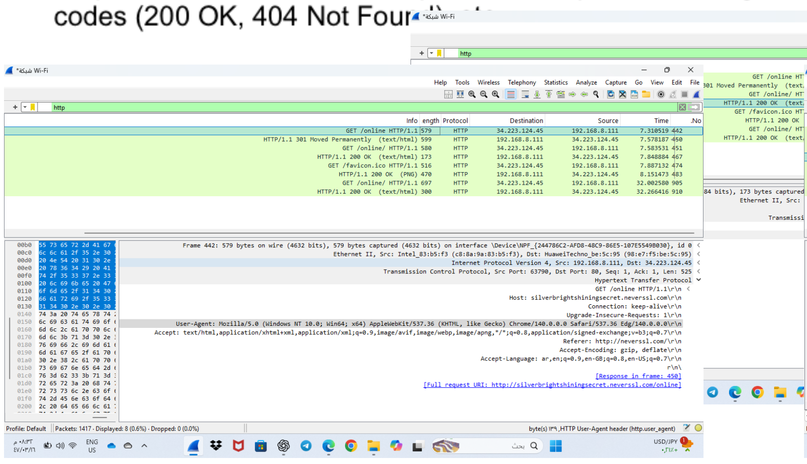
This lab session covers the usage of the Wireshark application to monitor and capture the outgoing and incoming packets from a network connection (WIFI, ethernet, etc.). Specifically, students should be able to analyze HTTP, HTTPS, TCP/IP, and UDP protocols using Wireshark, a network protocol analyzer, and draw conclusions.

Pre-lab Preparation:

1. Review the basics and the structure of HTTP, TCP/IP, and UDP protocols,
2. Install Wireshark and ensure it is running on your computer,
3. Create an online, *publically accessible* Git repository to host and upload your work in the labs. We recommend you use GitHub or GitLab.

Task 2: Filter HTTP packets and analyze them.

Step 1: In the filter bar, type http and press Enter. This filters out only the HTTP packets from the capture.
Step 2: Select any HTTP packet to view its details.
Step 3: Observe the HTTP request and response messages. Note the method (GET, POST), URL, response codes (200 OK, 404 Not Four

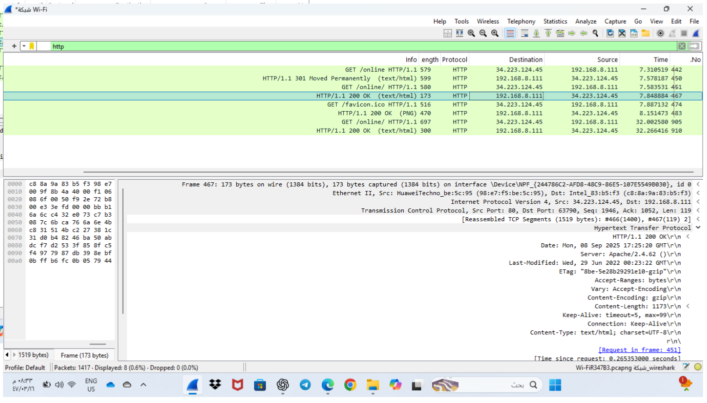


.No	Time	Source	Destination	Protocol	Length	Info
442	7.310519	192.168.8.111	34.223.124.45	HTTP	579	GET /online HTTP/1.1
450	7.578187	34.223.124.45	192.168.8.111	HTTP	599	HTTP/1.1 301 Moved Permanently (text/html)
451	7.583531	192.168.8.111	34.223.124.45	HTTP	580	GET /online/ HTTP/1.1
467	7.848884	34.223.124.45	192.168.8.111	HTTP	173	HTTP/1.1 200 OK (text/html)
474	7.887132	192.168.8.111	34.223.124.45	HTTP	516	GET /favicon.ico HTTP/1.1
483	8.151473	34.223.124.45	192.168.8.111	HTTP	470	HTTP/1.1 200 OK (PNG)
905	32.002580	192.168.8.111	34.223.124.45	HTTP	697	GET /online/ HTTP/1.1
910	32.266416	34.223.124.45	192.168.8.111	HTTP	300	HTTP/1.1 200 OK (text/html)

```
Frame 442: 579 bytes on wire (4632 bits), 579 bytes captured (4632 bits) on interface \Device\NPF_{244786C2-AFD8-48C9-86E5-107E5549B030}, id 0
Ethernet II, Src: Intel_83:b5:f3 (c8:8a:9a:83:b5:f3), Dst: HuaweiTechno_be:5c:95 (98:e7:f5:be:5c:95)
Internet Protocol Version 4, Src: 192.168.8.111, Dst: 34.223.124.45
Transmission Control Protocol, Src Port: 63790, Dst Port: 80, Seq: 1, Ack: 1, Len: 525
Hypertext Transfer Protocol
    GET /online HTTP/1.1\r\n
    Host: silverbrightshiningsecret.neverssl.com\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/140.0.0.0 Safari/537.36 Edg/140.0.0.0\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7\r\n
    Referer: http://neverssl.com/\r\n
    Accept-Encoding: gzip, deflate\r\n
    Accept-Language: ar,en;q=0.9,en-GB;q=0.8,en-US;q=0.7\r\n
    r\n\
    [Response in frame: 450]
    [Full request URI: http://silverbrightshiningsecret.neverssl.com/online]
```

Profile: Default | Packets: 1417 · Displayed: 8 (0.6%) · Dropped: 0 (0.0%) | byte(s) ١٣٩ ,HTTP User-Agent header (http.user_agent)

HTTP Request Packet



```
Frame 467: 173 bytes on wire (1384 bits), 173 bytes captured (1384 bits) on interface \Device\NPF_{244786C2-AFD8-48C9-86E5-107E5549B030}, id 0
Ethernet II, Src: HuaweiTechno_be:5c:95 (98:e7:f5:be:5c:95), Dst: Intel_83:b5:f3 (c8:8a:9a:83:b5:f3)
Internet Protocol Version 4, Src: 34.223.124.45, Dst: 192.168.8.111
Transmission Control Protocol, Src Port: 80, Dst Port: 63790, Seq: 1946, Ack: 1052, Len: 119
[Reassembled TCP Segments (1519 bytes): #466(1400), #467(119) 2]
Hypertext Transfer Protocol
    HTTP/1.1 200 OK\r\n
    Date: Mon, 08 Sep 2025 17:25:20 GMT\r\n
    Server: Apache/2.4.62 ()\r\n
    Last-Modified: Wed, 29 Jun 2022 00:23:22 GMT\r\n
    ETag: "8be-5e28b29291e10-gzip"\r\n
    Accept-Ranges: bytes\r\n
    Vary: Accept-Encoding\r\n
    Content-Encoding: gzip\r\n
    Content-Length: 1173\r\n
    Keep-Alive: timeout=5, max=99\r\n
    Connection: Keep-Alive\r\n
    Content-Type: text/html; charset=UTF-8\r\n
    r\n\
    [Request in frame: 451]
    [Time since request: 0.265353000 seconds]
```

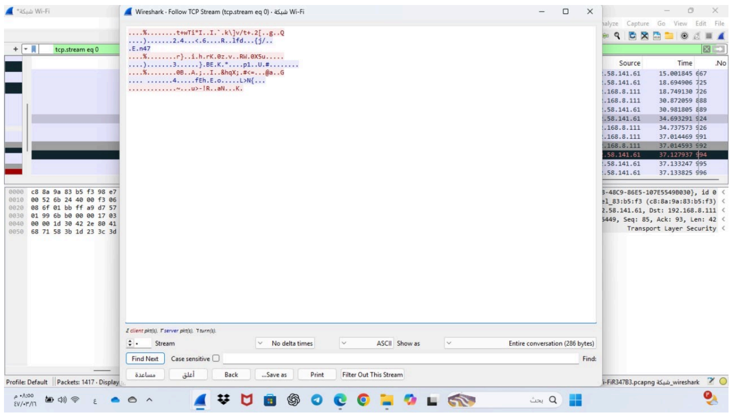
Profile: Default | Packets: 1417 · Displayed: 8 (0.6%) · Dropped: 0 (0.0%) | Wi-FiR347B3.pcapng شبكة_wireshark

HTTP Response Packet

Part 2: Analyzing TCP/IP Traffic.

Task 1: Filter TCP packets

Step 1: Clear the previous filter and type TCP to focus on TCP packets.
Step 2: Select a TCP packet related to your HTTP request/response.
Step 3: Right-click on the packet and select "Follow" -> "TCP Stream".
Step 4: This shows the entire conversation between the client and server.



Wireshark · Follow TCP Stream (tcp.stream eq 0) · شبكة Wi-Fi

```
....%........t+wTi*I..I.`.k\]v/t+.2[..g..Q
....).......2.4...<.6....R..lfd...{j/..
.E.n47
....%........r}..i.h.rK.0z.v..RW.0X5u.....
....).......3......}.BE.K.*....p1..U.#........
....%........0B..A.;..I..&hqX;.#<=...@a..G
.... .......4.....fEh.E.o.....L>N{...
............~...u>-!R..aN...K.
```

٤ client pkt(s), ٣ server pkt(s), ٦ turn(s).

Stream · | No delta times | Show as ASCII | Entire conversation (286 bytes)

.No	Time	Source
667	15.001845	.58.141.61
725	18.694906	.58.141.61
726	18.749130	.168.8.111
888	30.872059	.168.8.111
889	30.981805	.58.141.61
924	34.693291	.58.141.61
926	34.737573	.168.8.111
991	37.014469	.168.8.111
992	37.014593	.168.8.111
994	37.127937	.58.141.61
995	37.133247	.58.141.61
996	37.133825	.58.141.61

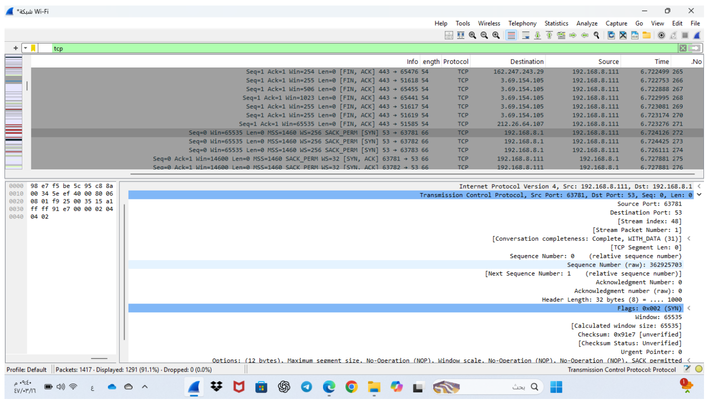
```
3-48C9-86E5-107E5549B030}, id 0
el_83:b5:f3 (c8:8a:9a:83:b5:f3)
2.58.141.61, Dst: 192.168.8.111
5449, Seq: 85, Ack: 93, Len: 42
Transport Layer Security
```

“TCP Stream for HTTPS traffic (data is encrypted, not readable).”

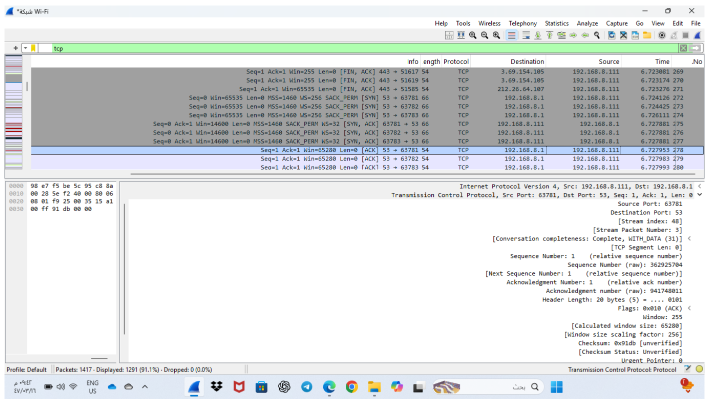
Task 2: Analyze TCP handshake and investigate Data Transfer and Termination
Step 1: Find and select packets related to the TCP three-way handshake:

- o SYN: Initiates a connection.
- o SYN-ACK: Acknowledges and responds to the SYN.
- o ACK: Acknowledges the SYN-ACK and establishes the connection.

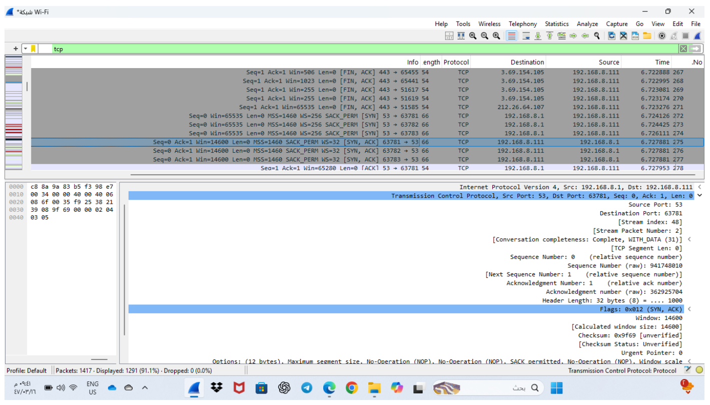
Step 2: Note the sequence and acknowledgment numbers. Screenshot and upload your image to your online git repository.
Step 3: Observe the data packets exchanged between the client and server. Take a screenshot and upload it to your online git repo.
Step 4: Look at the TCP termination process (FIN, ACK packets).



```
Info  .ength Protocol  Destination  Source  Time  .No
Seq=1 Ack=1 Win=254 Len=0 [FIN, ACK] 443 → 65476 54  TCP  162.247.243.29  192.168.8.111  6.722499 265
Seq=1 Ack=1 Win=255 Len=0 [FIN, ACK] 443 → 51618 54  TCP  3.69.154.105  192.168.8.111  6.722753 266
Seq=1 Ack=1 Win=506 Len=0 [FIN, ACK] 443 → 65455 54  TCP  3.69.154.105  192.168.8.111  6.722888 267
Seq=1 Ack=1 Win=1023 Len=0 [FIN, ACK] 443 → 65441 54  TCP  3.69.154.105  192.168.8.111  6.722995 268
Seq=1 Ack=1 Win=255 Len=0 [FIN, ACK] 443 → 51617 54  TCP  3.69.154.105  192.168.8.111  6.723081 269
Seq=1 Ack=1 Win=255 Len=0 [FIN, ACK] 443 → 51619 54  TCP  3.69.154.105  192.168.8.111  6.723174 270
Seq=1 Ack=1 Win=65535 Len=0 [FIN, ACK] 443 → 51585 54  TCP  212.26.64.107  192.168.8.111  6.723276 271
Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM [SYN] 53 → 63781 66  TCP  192.168.8.1  192.168.8.111  6.724126 272
Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM [SYN] 53 → 63782 66  TCP  192.168.8.1  192.168.8.111  6.724425 273
Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM [SYN] 53 → 63783 66  TCP  192.168.8.1  192.168.8.111  6.726111 274
Seq=0 Ack=1 Win=14600 Len=0 MSS=1460 SACK_PERM WS=32 [SYN, ACK] 63781 → 53 66  TCP  192.168.8.111  192.168.8.1  6.727881 275
Seq=0 Ack=1 Win=14600 Len=0 MSS=1460 SACK_PERM WS=32 [SYN, ACK] 63782 → 53 66  TCP  192.168.8.111  192.168.8.1  6.727881 276

0000  98 e7 f5 be 5c 95 c8 8a
0010  00 34 5e ef 40 00 80 06
0020  08 01 f9 25 00 35 15 a1
0030  ff ff 91 e7 00 00 02 04
0040  04 02

Internet Protocol Version 4, Src: 192.168.8.111, Dst: 192.168.8.1
Transmission Control Protocol, Src Port: 63781, Dst Port: 53, Seq: 0, Len: 0
    Source Port: 63781
    Destination Port: 53
    [Stream index: 48]
    [Stream Packet Number: 1]
    [Conversation completeness: Complete, WITH_DATA (31)]
    [TCP Segment Len: 0]
    Sequence Number: 0    (relative sequence number)
    Sequence Number (raw): 362925703
    [Next Sequence Number: 1    (relative sequence number)]
    Acknowledgment Number: 0
    Acknowledgment number (raw): 0
    Header Length: 32 bytes (8) = .... 1000
    Flags: 0x002 (SYN)
    Window: 65535
    [Calculated window size: 65535]
    Checksum: 0x91e7 [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0
    Options: (12 bytes), Maximum segment size, No-Operation (NOP), Window scale, No-Operation (NOP), No-Operation (NOP), SACK permitted

Profile: Default | Packets: 1417 · Displayed: 1291 (91.1%) · Dropped: 0 (0.0%) | Transmission Control Protocol: Protocol
```

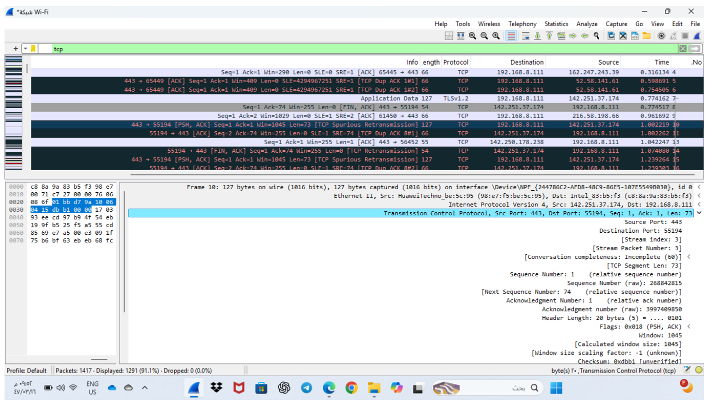


```
Info  .ength Protocol  Destination  Source  Time  .No
Seq=1 Ack=1 Win=255 Len=0 [FIN, ACK] 443 → 51617 54  TCP  3.69.154.105  192.168.8.111  6.723081 269
Seq=1 Ack=1 Win=255 Len=0 [FIN, ACK] 443 → 51619 54  TCP  3.69.154.105  192.168.8.111  6.723174 270
Seq=1 Ack=1 Win=65535 Len=0 [FIN, ACK] 443 → 51585 54  TCP  212.26.64.107  192.168.8.111  6.723276 271
Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM [SYN] 53 → 63781 66  TCP  192.168.8.1  192.168.8.111  6.724126 272
Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM [SYN] 53 → 63782 66  TCP  192.168.8.1  192.168.8.111  6.724425 273
Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM [SYN] 53 → 63783 66  TCP  192.168.8.1  192.168.8.111  6.726111 274
Seq=0 Ack=1 Win=14600 Len=0 MSS=1460 SACK_PERM WS=32 [SYN, ACK] 63781 → 53 66  TCP  192.168.8.111  192.168.8.1  6.727881 275
Seq=0 Ack=1 Win=14600 Len=0 MSS=1460 SACK_PERM WS=32 [SYN, ACK] 63782 → 53 66  TCP  192.168.8.111  192.168.8.1  6.727881 276
Seq=0 Ack=1 Win=14600 Len=0 MSS=1460 SACK_PERM WS=32 [SYN, ACK] 63783 → 53 66  TCP  192.168.8.111  192.168.8.1  6.727881 277
Seq=1 Ack=1 Win=65280 Len=0 [ACK] 53 → 63781 54  TCP  192.168.8.1  192.168.8.111  6.727953 278
Seq=1 Ack=1 Win=65280 Len=0 [ACK] 53 → 63782 54  TCP  192.168.8.1  192.168.8.111  6.727983 279
Seq=1 Ack=1 Win=65280 Len=0 [ACK] 53 → 63783 54  TCP  192.168.8.1  192.168.8.111  6.727993 280

0000  98 e7 f5 be 5c 95 c8 8a
0010  00 28 5e f2 40 00 80 06
0020  08 01 f9 25 00 35 15 a1
0030  00 ff 91 db 00 00

Internet Protocol Version 4, Src: 192.168.8.111, Dst: 192.168.8.1
Transmission Control Protocol, Src Port: 63781, Dst Port: 53, Seq: 1, Ack: 1, Len: 0
    Source Port: 63781
    Destination Port: 53
    [Stream index: 48]
    [Stream Packet Number: 3]
    [Conversation completeness: Complete, WITH_DATA (31)]
    [TCP Segment Len: 0]
    Sequence Number: 1    (relative sequence number)
    Sequence Number (raw): 362925704
    [Next Sequence Number: 1    (relative sequence number)]
    Acknowledgment Number: 1    (relative ack number)
    Acknowledgment number (raw): 941748011
    Header Length: 20 bytes (5) = .... 0101
    Flags: 0x010 (ACK)
    Window: 255
    [Calculated window size: 65280]
    [Window size scaling factor: 256]
    Checksum: 0x91db [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0

Profile: Default | Packets: 1417 · Displayed: 1291 (91.1%) · Dropped: 0 (0.0%) | Transmission Control Protocol: Protocol
```



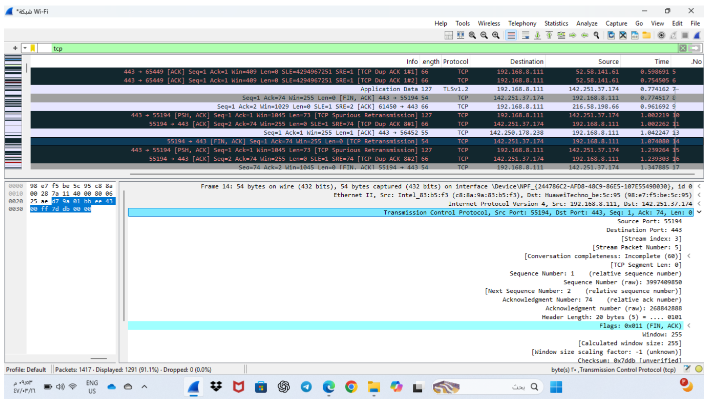
```
Info  .ength Protocol  Destination  Source  Time  .No
Seq=1 Ack=1 Win=506 Len=0 [FIN, ACK] 443 → 65455 54  TCP  3.69.154.105  192.168.8.111  6.722888 267
Seq=1 Ack=1 Win=1023 Len=0 [FIN, ACK] 443 → 65441 54  TCP  3.69.154.105  192.168.8.111  6.722995 268
Seq=1 Ack=1 Win=255 Len=0 [FIN, ACK] 443 → 51617 54  TCP  3.69.154.105  192.168.8.111  6.723081 269
Seq=1 Ack=1 Win=255 Len=0 [FIN, ACK] 443 → 51619 54  TCP  3.69.154.105  192.168.8.111  6.723174 270
Seq=1 Ack=1 Win=65535 Len=0 [FIN, ACK] 443 → 51585 54  TCP  212.26.64.107  192.168.8.111  6.723276 271
Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM [SYN] 53 → 63781 66  TCP  192.168.8.1  192.168.8.111  6.724126 272
Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM [SYN] 53 → 63782 66  TCP  192.168.8.1  192.168.8.111  6.724425 273
Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM [SYN] 53 → 63783 66  TCP  192.168.8.1  192.168.8.111  6.726111 274
Seq=0 Ack=1 Win=14600 Len=0 MSS=1460 SACK_PERM WS=32 [SYN, ACK] 63781 → 53 66  TCP  192.168.8.111  192.168.8.1  6.727881 275
Seq=0 Ack=1 Win=14600 Len=0 MSS=1460 SACK_PERM WS=32 [SYN, ACK] 63782 → 53 66  TCP  192.168.8.111  192.168.8.1  6.727881 276
Seq=0 Ack=1 Win=14600 Len=0 MSS=1460 SACK_PERM WS=32 [SYN, ACK] 63783 → 53 66  TCP  192.168.8.111  192.168.8.1  6.727881 277
Seq=1 Ack=1 Win=65280 Len=0 [ACK] 53 → 63781 54  TCP  192.168.8.1  192.168.8.111  6.727953 278

0000  c8 8a 9a 83 b5 f3 98 e7
0010  00 34 00 00 40 00 40 06
0020  08 6f 00 35 f9 25 38 21
0030  39 08 9f 69 00 00 02 04
0040  03 05

Internet Protocol Version 4, Src: 192.168.8.1, Dst: 192.168.8.111
Transmission Control Protocol, Src Port: 53, Dst Port: 63781, Seq: 0, Ack: 1, Len: 0
    Source Port: 53
    Destination Port: 63781
    [Stream index: 48]
    [Stream Packet Number: 2]
    [Conversation completeness: Complete, WITH_DATA (31)]
    [TCP Segment Len: 0]
    Sequence Number: 0    (relative sequence number)
    Sequence Number (raw): 941748010
    [Next Sequence Number: 1    (relative sequence number)]
    Acknowledgment Number: 1    (relative ack number)
    Acknowledgment number (raw): 362925704
    Header Length: 32 bytes (8) = .... 1000
    Flags: 0x012 (SYN, ACK)
    Window: 14600
    [Calculated window size: 14600]
    Checksum: 0x9f69 [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0
    Options: (12 bytes), Maximum segment size, No-Operation (NOP), No-Operation (NOP), SACK permitted, No-Operation (NOP), Window scale

Profile: Default | Packets: 1417 · Displayed: 1291 (91.1%) · Dropped: 0 (0.0%) | Transmission Control Protocol: Protocol
```

“TCP three-way handshake: SYN, SYN-ACK, ACK establishing the connection.”



```
Info  .ength Protocol  Destination  Source  Time  .No
Seq=1 Ack=1 Win=290 Len=0 SLE=0 SRE=1 [ACK] 65445 → 443 66  TCP  192.168.8.111  162.247.243.39  0.316134 4
443 → 65449 [ACK] Seq=1 Ack=1 Win=409 Len=0 SLE=4294967251 SRE=1 [TCP Dup ACK 1#1] 66  TCP  192.168.8.111  52.58.141.61  0.598691 5
443 → 65449 [ACK] Seq=1 Ack=1 Win=409 Len=0 SLE=4294967251 SRE=1 [TCP Dup ACK 1#2] 66  TCP  192.168.8.111  52.58.141.61  0.754505 6
Application Data 127  TLSv1.2  192.168.8.111  142.251.37.174  0.774162 7
Seq=1 Ack=74 Win=255 Len=0 [FIN, ACK] 443 → 55194 54  TCP  142.251.37.174  192.168.8.111  0.774517 8
Seq=1 Ack=2 Win=1029 Len=0 SLE=1 SRE=2 [ACK] 61450 → 443 66  TCP  192.168.8.111  216.58.198.66  0.961692 9
443 → 55194 [PSH, ACK] Seq=1 Ack=1 Win=1045 Len=73 [TCP Spurious Retransmission] 127  TCP  192.168.8.111  142.251.37.174  1.002219 10
55194 → 443 [ACK] Seq=2 Ack=74 Win=255 Len=0 SLE=1 SRE=74 [TCP Dup ACK 8#1] 66  TCP  142.251.37.174  192.168.8.111  1.002262 11
Seq=1 Ack=1 Win=255 Len=1 [ACK] 443 → 56452 55  TCP  142.250.178.238  192.168.8.111  1.042247 13
55194 → 443 [FIN, ACK] Seq=1 Ack=74 Win=255 Len=0 [TCP Retransmission] 54  TCP  142.251.37.174  192.168.8.111  1.074080 14
443 → 55194 [PSH, ACK] Seq=1 Ack=1 Win=1045 Len=73 [TCP Spurious Retransmission] 127  TCP  192.168.8.111  142.251.37.174  1.239264 15
55194 → 443 [ACK] Seq=2 Ack=74 Win=255 Len=0 SLE=1 SRE=74 [TCP Dup ACK 8#2] 66  TCP  142.251.37.174  192.168.8.111  1.239303 16

0000  c8 8a 9a 83 b5 f3 98 e7
0010  00 71 c7 27 00 00 76 06
0020  08 6f 01 bb d7 9a 10 06
0030  04 15 db b1 00 00 17 03
0040  93 ee cd 97 b9 4f 54 eb
0050  19 9f b5 25 f5 a5 55 cd
0060  85 69 e7 a5 00 e3 09 1f
0070  75 b6 bf 63 eb eb 68 fc

Frame 10: 127 bytes on wire (1016 bits), 127 bytes captured (1016 bits) on interface \Device\NPF_{244786C2-AFD8-48C9-86E5-107E5549B030}, id 0
Ethernet II, Src: HuaweiTechno_be:5c:95 (98:e7:f5:be:5c:95), Dst: Intel_83:b5:f3 (c8:8a:9a:83:b5:f3)
Internet Protocol Version 4, Src: 142.251.37.174, Dst: 192.168.8.111
Transmission Control Protocol, Src Port: 443, Dst Port: 55194, Seq: 1, Ack: 1, Len: 73
    Source Port: 443
    Destination Port: 55194
    [Stream index: 3]
    [Stream Packet Number: 3]
    [Conversation completeness: Incomplete (60)]
    [TCP Segment Len: 73]
    Sequence Number: 1    (relative sequence number)
    Sequence Number (raw): 268842815
    [Next Sequence Number: 74    (relative sequence number)]
    Acknowledgment Number: 1    (relative ack number)
    Acknowledgment number (raw): 3997409850
    Header Length: 20 bytes (5) = .... 0101
    Flags: 0x018 (PSH, ACK)
    Window: 1045
    [Calculated window size: 1045]
    [Window size scaling factor: -1 (unknown)]
    Checksum: 0xdbb1 [unverified]

Profile: Default | Packets: 1417 · Displayed: 1291 (91.1%) · Dropped: 0 (0.0%) | byte(s) ٢٠ ,Transmission Control Protocol (tcp)
```

"TCP Data Transfer between client and server"



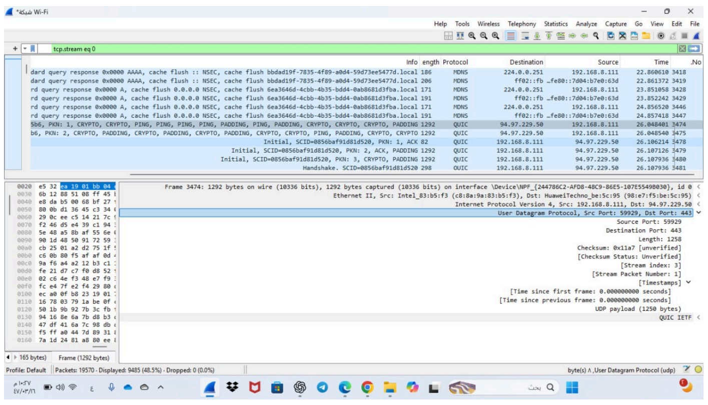
```
*شبكة Wi-Fi
tcp
                                                                  Info .ength Protocol    Destination         Source        Time .No
    443 → 65449 [ACK] Seq=1 Ack=1 Win=409 Len=0 SLE=4294967251 SRE=1 [TCP Dup ACK 1#1] 66   TCP   192.168.8.111    52.58.141.61    0.598691 5
    443 → 65449 [ACK] Seq=1 Ack=1 Win=409 Len=0 SLE=4294967251 SRE=1 [TCP Dup ACK 1#2] 66   TCP   192.168.8.111    52.58.141.61    0.754505 6
                                                               Application Data 127 TLSv1.2   192.168.8.111    142.251.37.174  0.774162 7
                                       Seq=1 Ack=74 Win=255 Len=0 [FIN, ACK] 443 → 55194 54   TCP   142.251.37.174   192.168.8.111   0.774517 8
                                 Seq=1 Ack=2 Win=1029 Len=0 SLE=1 SRE=2 [ACK] 61450 → 443 66   TCP   192.168.8.111    216.58.198.66   0.961692 9
      443 → 55194 [PSH, ACK] Seq=1 Ack=1 Win=1045 Len=73 [TCP Spurious Retransmission] 127   TCP   192.168.8.111    142.251.37.174  1.002219 10
          55194 → 443 [ACK] Seq=2 Ack=74 Win=255 Len=0 SLE=1 SRE=74 [TCP Dup ACK 8#1] 66   TCP   142.251.37.174   192.168.8.111   1.002262 11
                                              Seq=1 Ack=1 Win=255 Len=1 [ACK] 443 → 56452 55   TCP   142.250.178.238  192.168.8.111   1.042247 13
              55194 → 443 [FIN, ACK] Seq=1 Ack=74 Win=255 Len=0 [TCP Retransmission] 54   TCP   142.251.37.174   192.168.8.111   1.074080 14
      443 → 55194 [PSH, ACK] Seq=1 Ack=1 Win=1045 Len=73 [TCP Spurious Retransmission] 127   TCP   192.168.8.111    142.251.37.174  1.239264 15
          55194 → 443 [ACK] Seq=2 Ack=74 Win=255 Len=0 SLE=1 SRE=74 [TCP Dup ACK 8#2] 66   TCP   142.251.37.174   192.168.8.111   1.239303 16
                                       Seq=74 Ack=2 Win=1045 Len=0 [FIN, ACK] 55194 → 443 54   TCP   192.168.8.111    142.251.37.174  1.347885 17

0000  98 e7 f5 be 5c 95 c8 8a
0010  00 28 7a 11 40 00 80 06
0020  25 ae d7 9a 01 bb ee 43
0030  00 ff 7d db 00 00

Frame 14: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface \Device\NPF_{244786C2-AFD8-48C9-86E5-107E5549B030}, id 0
Ethernet II, Src: Intel_83:b5:f3 (c8:8a:9a:83:b5:f3), Dst: HuaweiTechno_be:5c:95 (98:e7:f5:be:5c:95)
Internet Protocol Version 4, Src: 192.168.8.111, Dst: 142.251.37.174
Transmission Control Protocol, Src Port: 55194, Dst Port: 443, Seq: 1, Ack: 74, Len: 0
    Source Port: 55194
    Destination Port: 443
    [Stream index: 3]
    [Stream Packet Number: 5]
    [Conversation completeness: Incomplete (60)]
    [TCP Segment Len: 0]
    Sequence Number: 1    (relative sequence number)
    Sequence Number (raw): 3997409850
    [Next Sequence Number: 2    (relative sequence number)]
    Acknowledgment Number: 74    (relative ack number)
    Acknowledgment number (raw): 268842888
    Header Length: 20 bytes (5) = .... 0101
    Flags: 0x011 (FIN, ACK)
    Window: 255
    [Calculated window size: 255]
    [Window size scaling factor: -1 (unknown)]
    Checksum: 0x7ddb [unverified]

Profile: Default | Packets: 1417 · Displayed: 1291 (91.1%) · Dropped: 0 (0.0%)    byte(s) ٢٠ ,Transmission Control Protocol (tcp)
```

"TCP connection termination using FIN and ACK packets."

Task 1: Generate UDP traffic and capture packets

Step 1: Open a network application that uses UDP (e.g., streaming video, VoIP software, or custom script).
Step 2: Start the application to generate UDP traffic.
Step 3: Start capturing packets in Wireshark while the UDP application is running.
Step 4: After sufficient traffic is generated, stop capturing packets.

Task 2: Filter and analysis UDP Packets
Step 1: In the filter bar, type UDP and press Enter.
Step 2: This filters out only the UDP packets from the capture.
Step 3: Select any UDP packet to view its details.
Step 4: Observe the source and destination ports, length, and data.
Step 5: Compare the simplicity of UDP headers with TCP headers.



```
*شبكة Wi-Fi
tcp.stream eq 0
                                                                                              Info .ength Protocol  Destination          Source        Time .No
dard query response 0x0000 AAAA, cache flush :: NSEC, cache flush bbdad19f-7835-4f89-a0d4-59d73ee5477d.local 186  MDNS  224.0.0.251   192.168.8.111        22.860610 3418
dard query response 0x0000 AAAA, cache flush :: NSEC, cache flush bbdad19f-7835-4f89-a0d4-59d73ee5477d.local 206  MDNS  ff02::fb      …fe80::7d04:b7e0:63d 22.861372 3419
rd query response 0x0000 A, cache flush 0.0.0.0 NSEC, cache flush 6ea3646d-4cbb-4b35-bdd4-0ab8681d3fba.local 171  MDNS  224.0.0.251   192.168.8.111        23.851058 3428
rd query response 0x0000 A, cache flush 0.0.0.0 NSEC, cache flush 6ea3646d-4cbb-4b35-bdd4-0ab8681d3fba.local 191  MDNS  ff02::fb      …fe80::7d04:b7e0:63d 23.852242 3429
rd query response 0x0000 A, cache flush 0.0.0.0 NSEC, cache flush 6ea3646d-4cbb-4b35-bdd4-0ab8681d3fba.local 171  MDNS  224.0.0.251   192.168.8.111        24.856520 3446
rd query response 0x0000 A, cache flush 0.0.0.0 NSEC, cache flush 6ea3646d-4cbb-4b35-bdd4-0ab8681d3fba.local 191  MDNS  ff02::fb      …fe80::7d04:b7e0:63d 24.857418 3447
5b6, PKN: 1, CRYPTO, CRYPTO, PING, PING, PING, PING, PADDING, PING, PADDING, CRYPTO, CRYPTO, CRYPTO, PADDING 1292  QUIC  94.97.229.50  192.168.8.111        26.048401 3474
b6, PKN: 2, CRYPTO, PADDING, CRYPTO, PADDING, CRYPTO, PADDING, CRYPTO, CRYPTO, PING, PADDING, CRYPTO, CRYPTO 1292  QUIC  94.97.229.50  192.168.8.111        26.048540 3475
                                                                Initial, SCID=0856baf91d81d520, PKN: 1, ACK 82  QUIC  192.168.8.111 94.97.229.50         26.106214 3478
                                                      Initial, SCID=0856baf91d81d520, PKN: 2, ACK, PADDING 1292  QUIC  192.168.8.111 94.97.229.50         26.107126 3479
                                                   Initial, SCID=0856baf91d81d520, PKN: 3, CRYPTO, PADDING 1292  QUIC  192.168.8.111 94.97.229.50         26.107936 3480
                                                                      Handshake. SCID=0856baf91d81d520 298  QUIC  192.168.8.111 94.97.229.50         26.107936 3481

0020  e5 32 ea 19 01 bb 04
0030  6b 12 88 51 08 ff 45
0040  e8 da b5 00 68 bf 27
0050  80 0b d1 36 45 c3 34
0060  29 0c ee c5 14 21 7c
0070  f2 46 d5 e4 39 c1 94
0080  5e 48 a5 8b af 55 6e
0090  90 1d 48 50 91 72 59
00a0  cb 25 01 a2 d2 75 1f
00b0  c6 0b 80 f5 af af 0d
00c0  9a f6 a4 a2 12 b3 c1
00d0  fe 21 d7 c7 f0 d8 52
00e0  02 c6 4e f3 48 e7 f9
00f0  fc e4 7f e2 f4 29 80
0100  ec a0 0f b8 23 19 01
0110  16 78 03 79 1a be 0f
0120  50 1b 9b 92 7b 3c fb
0130  94 16 8e 6a 7b d8 b3
0140  47 df 41 6a 7c 98 db
0150  f5 ff a0 44 7d 89 31
0160  7a 1d 24 81 a8 80 ee

Frame 3474: 1292 bytes on wire (10336 bits), 1292 bytes captured (10336 bits) on interface \Device\NPF_{244786C2-AFD8-48C9-86E5-107E5549B030}, id 0
Ethernet II, Src: Intel_83:b5:f3 (c8:8a:9a:83:b5:f3), Dst: HuaweiTechno_be:5c:95 (98:e7:f5:be:5c:95)
Internet Protocol Version 4, Src: 192.168.8.111, Dst: 94.97.229.50
User Datagram Protocol, Src Port: 59929, Dst Port: 443
    Source Port: 59929
    Destination Port: 443
    Length: 1258
    Checksum: 0x11a7 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 3]
    [Stream Packet Number: 1]
    [Timestamps]
        [Time since first frame: 0.000000000 seconds]
        [Time since previous frame: 0.000000000 seconds]
    UDP payload (1250 bytes)
QUIC IETF

165 bytes)   Frame (1292 bytes)
Profile: Default | Packets: 19570 · Displayed: 9485 (48.5%) · Dropped: 0 (0.0%)    byte(s) ٨ ,User Datagram Protocol (udp)
```

"UDP packet captured showing Source Port, Destination Port, and Length. Payload data may be encrypted."

Step 4: Observe the source and destination ports, length, and data.
Step 5: Compare the simplicity of UDP headers with TCP headers.

Comparison of UDP and TCP headers:
The captured UDP packet (QUIC) demonstrates the simplicity of UDP headers, which are only 8 bytes and contain the source port, destination port, length, and checksum. In contrast, TCP headers are larger (at least 20 bytes) and include additional fields such as sequence numbers, acknowledgment numbers, flags (SYN, ACK, FIN), window size, and options. This shows that UDP has a much simpler structure compared to TCP.

Part 4: Comparing TCP and UDP by filling in the following tables. Save your work (e.g., in an MS Word document), and upload it to your online git repo.

Task 1: Fill in the following table and provide reasons.

	TCP or UDP	Reasons
Reliability and Connection Establishment		
Data Integrity and Ordering		

	TCP or UDP	Reasons
Reliability and connection Establishment	TCP	TCP is a reliable Protocol that establishes connection using a three-way hand shake (SYN, SYN-ACK, ACK). UDP is connectionless and doesn't not guarentee reliability
Data Integrity and Ordering	TCP	TCP ensures data integrity and correct ordering : all Packets arrive and in order ; lost Packets are retransmitted UDP does not gurantee order or integrity

Task 2: Identify the use Cases and Performance of TCP and UDP.

	TCP	UDP
Use cases		
Performance		

	TCP	UDP
Use Cases	Web browsing (http(https) remile (Smtp, IMAP, POP3) , file transfer (ftp) , remote login (SSh, telnet)	Streaming Video / audio, online gaming , VoIP calles , DNS Queries
Performance	Slower than UDP due to connection Setup, ack, and retransmission , but reliable and ordered	faster than TCP because it has no connection setup or retransmission but less reliable ; Suitable for real-time application