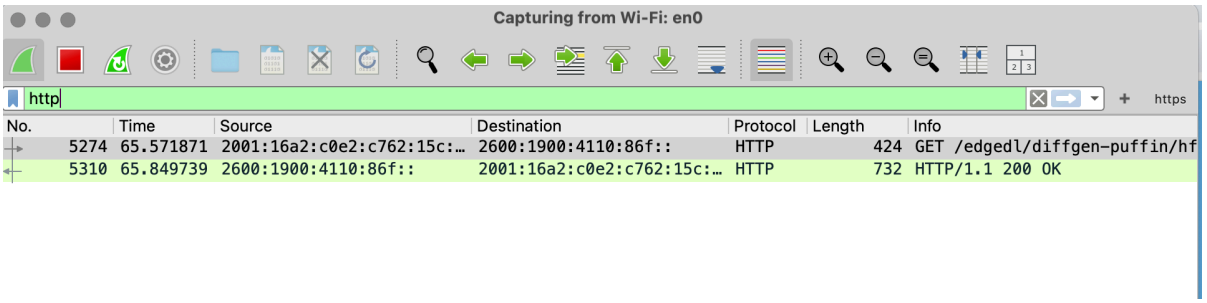
# Wire Shark

## Capturing HTTP Traffic.

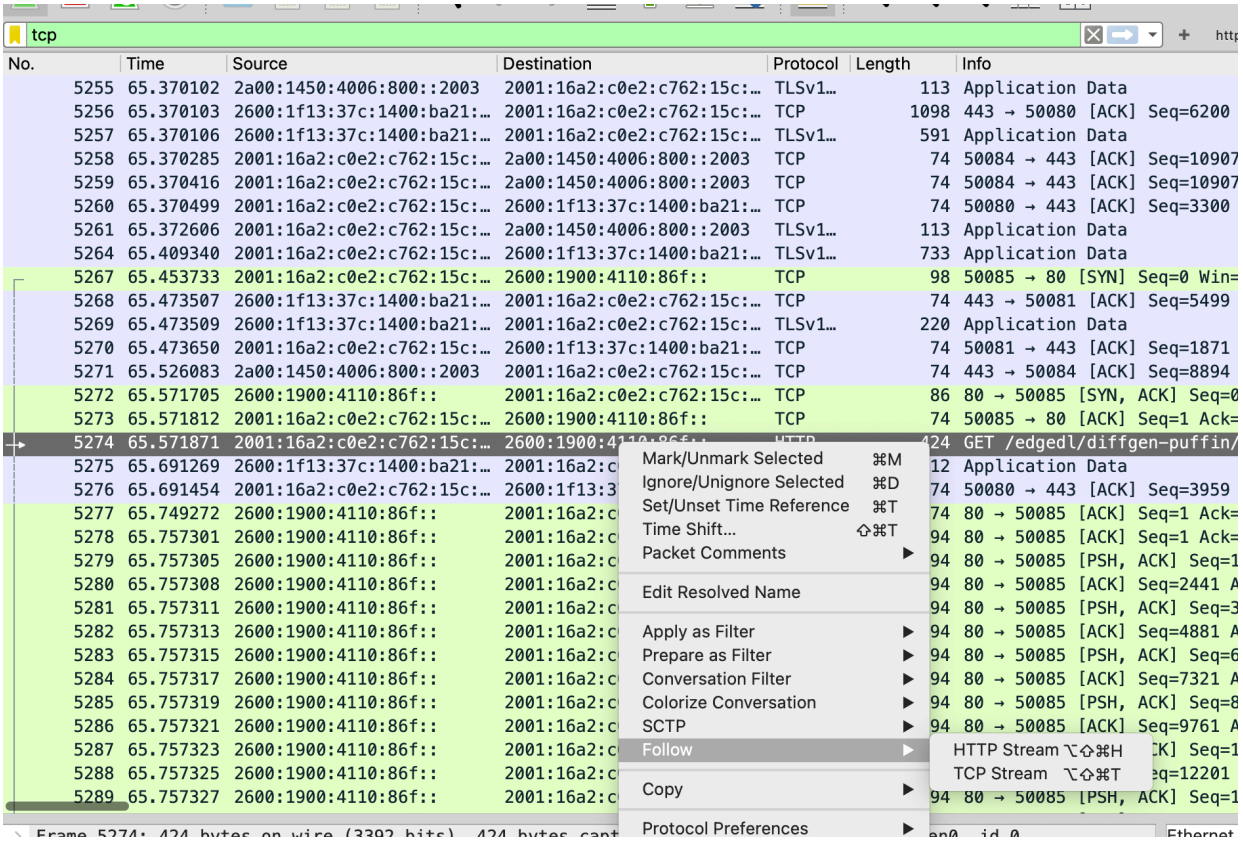
## Task 1: Start Wireshark and capture packets.

## Task 2: Filter HTTP packets and analyze them.

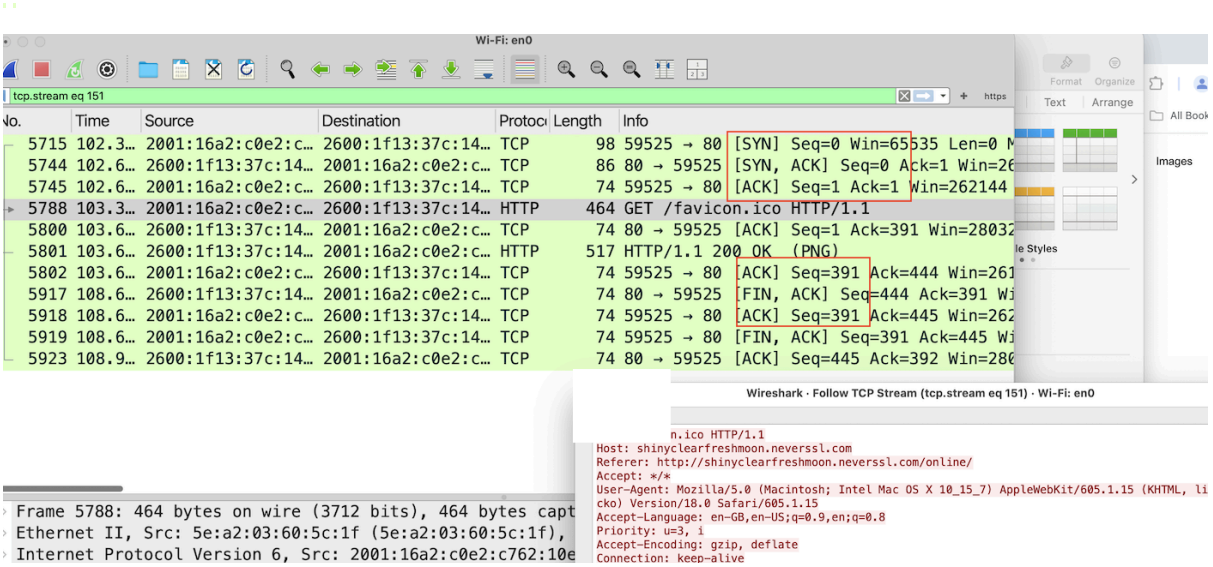



## Part 2: Analyzing TCP/IP Traffic.

## Task 1: Filter TCP packets.



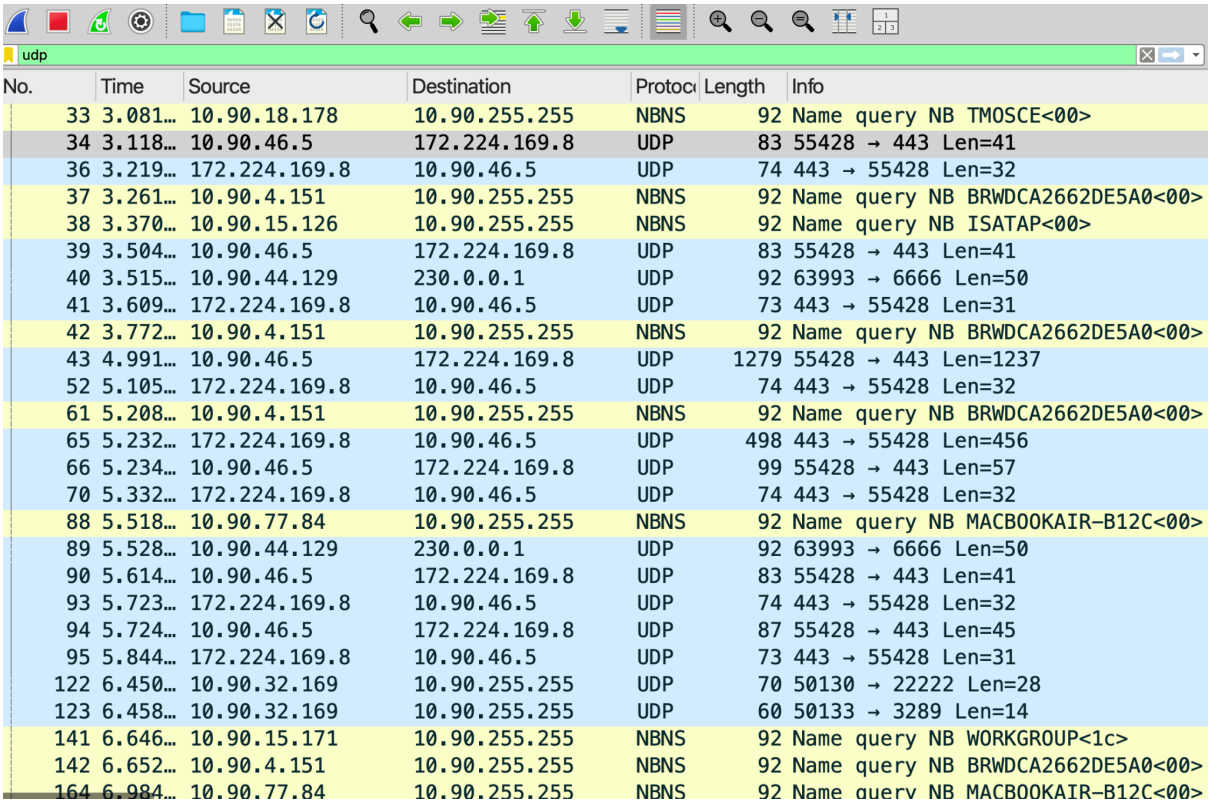
## Task 2: Analyze TCP handshake and investigate Data Transfer and Termination.




## Part 3: Capturing and Analyzing UDP Traffic

## Task 1: Generate UDP traffic and capture packets

## Task 2: Filter and analysis UDP Packets



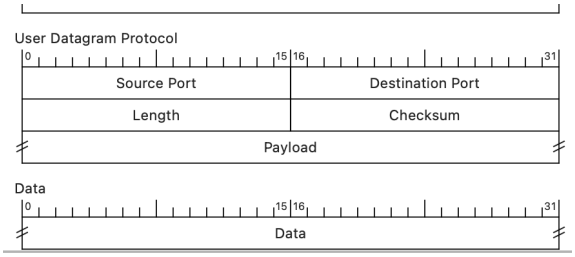
Step 4: Observe the source and destination ports, length, and data

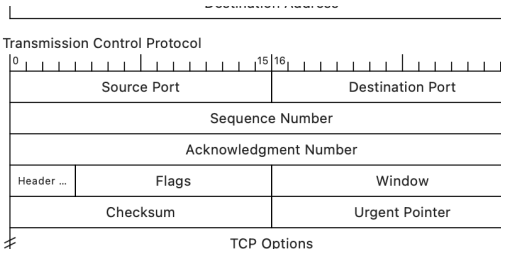
```
   24 0.984… 10.90.46.5        172.224.169.8      UDP      83 55428 → 443 Len=41
   25 1.117… 172.224.169.8     10.90.46.5         UDP      74 443 → 55428 Len=32
   26 1.308… 10.90.46.5        172.224.169.8      UDP      83 55428 → 443 Len=41
   27 1.499… 172.224.169.8     10.90.46.5         UDP      73 443 → 55428 Len=31
   28 1.534… 10.90.1.208       255.255.255.255    UDP      60 63299 → 8610 Len=16
   29 1.536… 10.90.1.208       255.255.255.255    UDP      60 63299 → 8610 Len=16
Destination Address: 10.90.46.5
[Stream index: 5]
User Datagram Protocol, Src Port: 443, Dst Port: 55428
  Source Port: 443
  Destination Port: 55428
  Length: 40
  Checksum: 0xbc73 [unverified]
  [Checksum Status: Unverified]
  [Stream index: 5]
  [Stream Packet Number: 2]
  [Timestamps]
  UDP payload (32 bytes)
Data (32 bytes)
  Data: 51530eed6b9cc4426ae92ccd9876ca564b0264c5009d3079422d814106b4a4fe
  [Length: 32]
```

.

Step 5: Compare the simplicity of UDP headers with TCP headers.

| | UDP | TCP |
|---|---|---|
| **Header Size** | 8 bytes | 20-60 bytes |




## Part 4: Comparing TCP and UDP by filling in the following tables. Save your work (e.g., in an MS Word document)

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

| Feature | TCP or UDP | Reasons |
|---|---|---|
| **Reliability and Connection Establishment** | TCP | TCP is a connection-oriented protocol that ensures reliable communication. It establishes a connection using a three-way handshake |

| | | |
|---|---|---|
| **Data Integrity and Ordering** | TCP | TCP provides error checking, retransmission, and ordering of packets to ensure data integrity. It ensures that data is received in the correct sequence. UDP, does not guarantee ordering or delivery. |

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

| Feature | TCP | UDP |
|---|---|---|
| **Use Cases** | Web browsing (HTTP/HTTPS), email (SMTP, IMAP, POP3), file transfer (FTP) | Real-time applications like VoIP, video streaming, online gaming, and DNS |
| **Performance** | Slower due to connection establishment, error checking, and retransmissions | Faster because it is connectionless, has low overhead, and does not wait for acknowledgments or retransmit lost packets. |

Raghad Alsuhaibani