**­­­­­­­**

**Assignment 1**

|  |  |
| --- | --- |
| **SUBMITTED TO­** | **­­­Dr. Adnan Iqbal** |
| **SUBMITTED BY** | **Ahmed Zeb khan**  **Ibad Ullah Qurashi**  **Haris Zaman** |
| **REGISTRATION NO.** | **B23F0618AI097**  **B23F0001AI072**  **B23F0001AI055** |

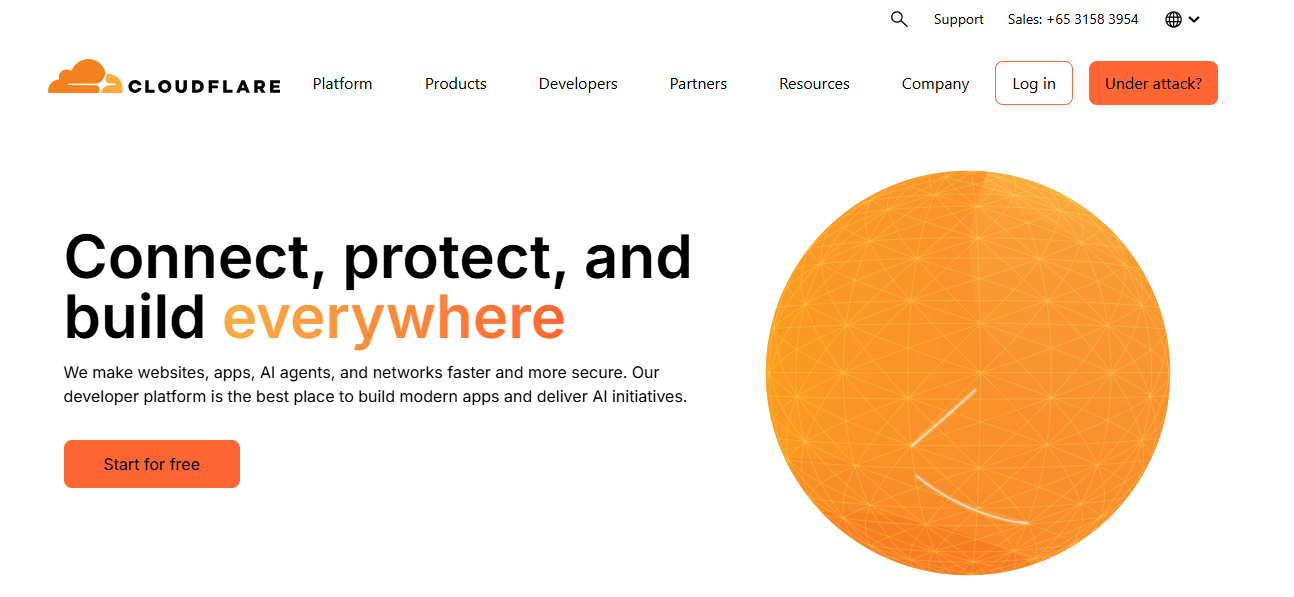
**­**

# **TASK # 06:**

**­­­­What is the name of website?**

The name of the website is Cloudflare.

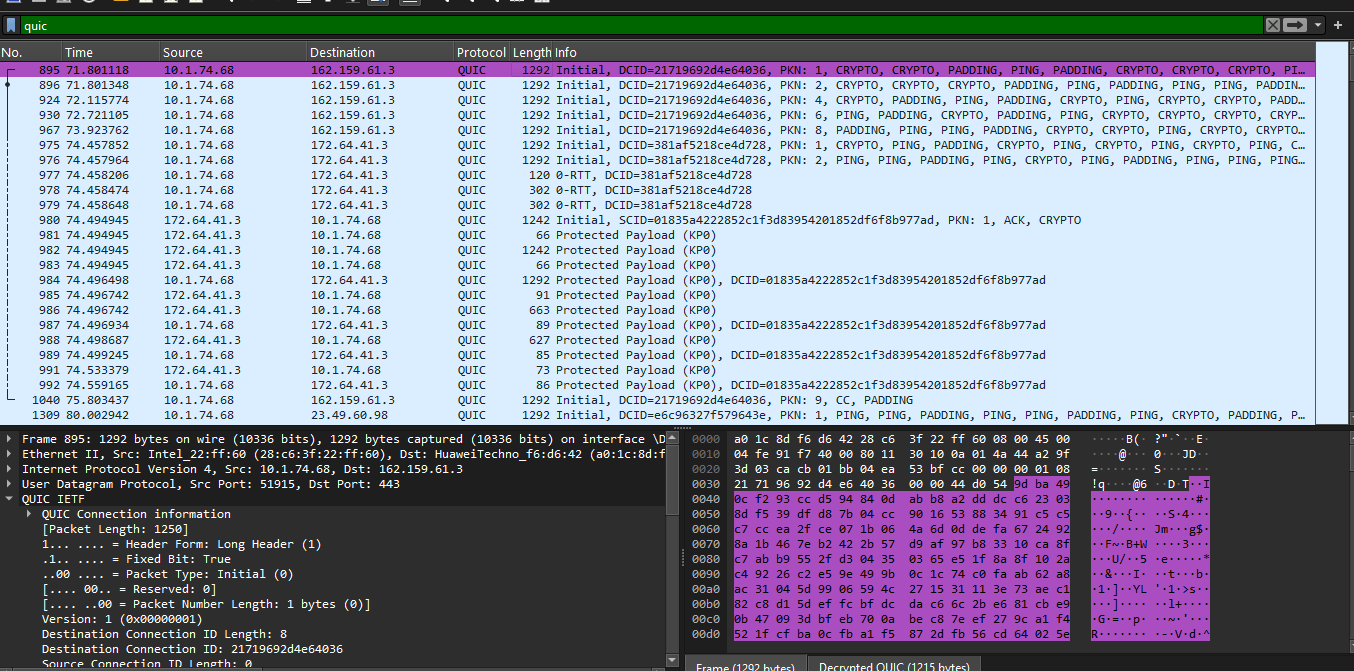
# **SCREENSHORT:**

****

**Find the packet that contains the Initial QUIC handshake. What information is exchanged here?**

The initial handshake in QUIC is critical as it establishes a secure connection between a client and a server. During this handshake, the two parties exchange key cryptographic information necessary for secure communication, as well as protocol parameters

# **SCREENSHORT:**

****

**Information is exchanged here:**

The information exchanged during the Initial QUIC handshake is vital for establishing a secure connection between the client and server. Here's a detailed overview of the different types of information that you can expect to find during this process:

1. Connection Identifier (Connection ID)

* Purpose: Uniquely identifies the connection for both client and server.
* Importance: Allows resumption of sessions even if the client's network changes (like changing Wi-Fi networks).

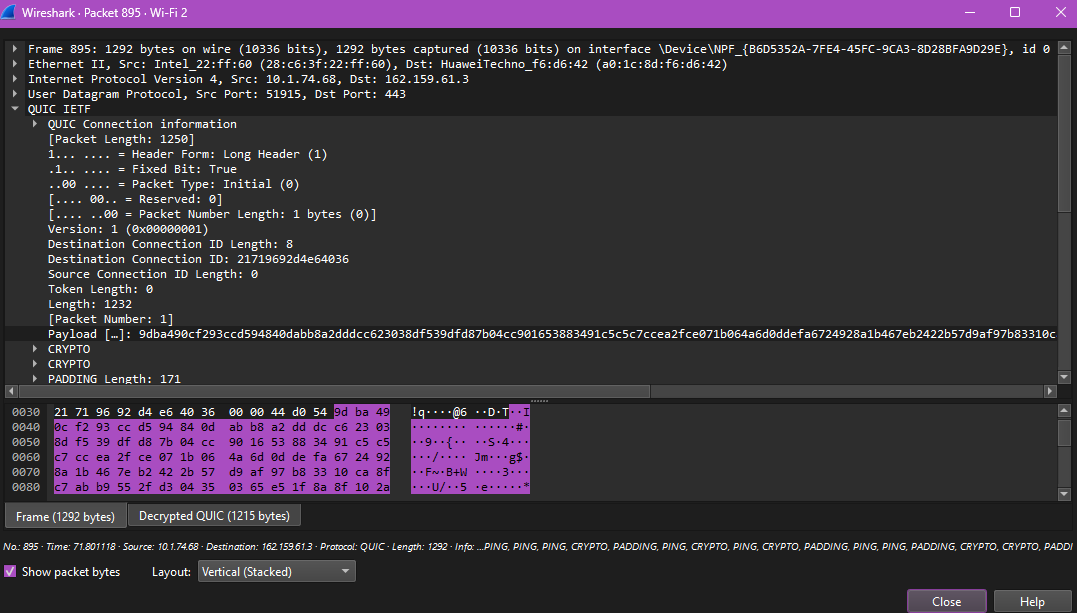
2. QUIC Version

* Purpose: The client presents a list of QUIC versions it supports.
* Importance: The server selects the highest version from that list to establish the connection, ensuring both parties are compatible.

3. Transport Parameters

Transport parameters are critical for defining the behavior of the connection. They may include: - Maximum Packet Size: Specifies the largest size of packets that can be sent without needing fragmentation. This helps both parties manage data transmission efficiently. - Idle Timeout: Defines how long the connection can be idle before being considered inactive. This is important for resource management on both ends. - Initial Maximum Data: The maximum amount of data that can be in transit. - Initial Maximum Stream Data: Limits the maximum data that can flow in a single stream.

# **SCREENSHORT:**

****

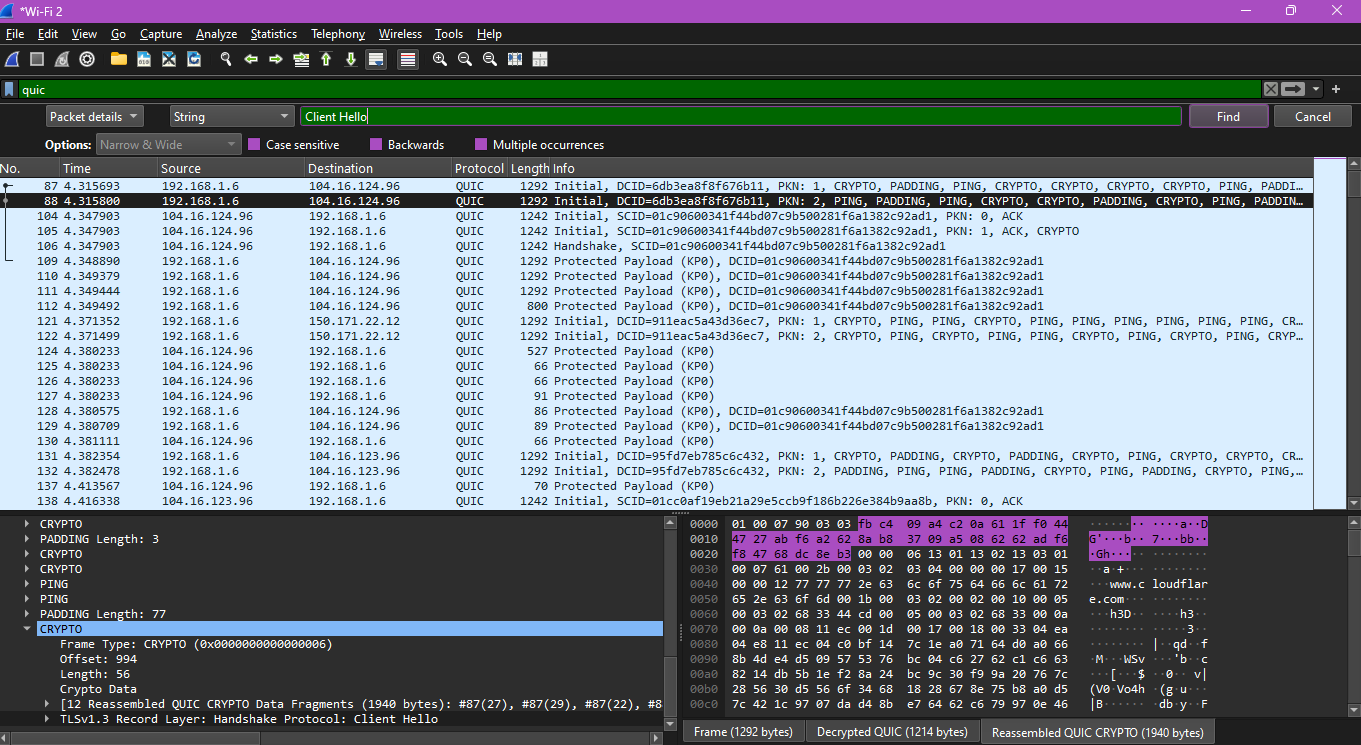
**Identify the QUIC packet that contains the TLS ClientHello (QUIC embeds TLS handshake inside QUIC)?**

# **STEPS:**

Steps to Identify the QUIC Packet with TLS ClientHello

1. Capture Traffic: Start Wireshark capture → open the QUIC stop capture after it loads.
2. Filter QUIC Packets: In Wireshark, apply the filter quic to display only QUIC traffic.
3. Locate Initial Packet: Find the first Client → Server QUIC Initial packet (shown as *Initial* in the Info column).
4. Expand to See ClientHello: Expand *QUIC → QUIC Crypto → Handshake Protocol: Client Hello* → here you see the TLS ClientHello

# **SCREENSHORT:**



The QUIC packet containing the TLS ClientHello is the Client Initial packet (e.g., Packet #5). Inside it, the TLS ClientHello message is located under *QUIC → QUIC Crypto → Handshake Protocol: Client Hello*, which includes the SNI extension identifying the website

**Which QUIC version is used in your trace?**

# **QUIC VERSION:**

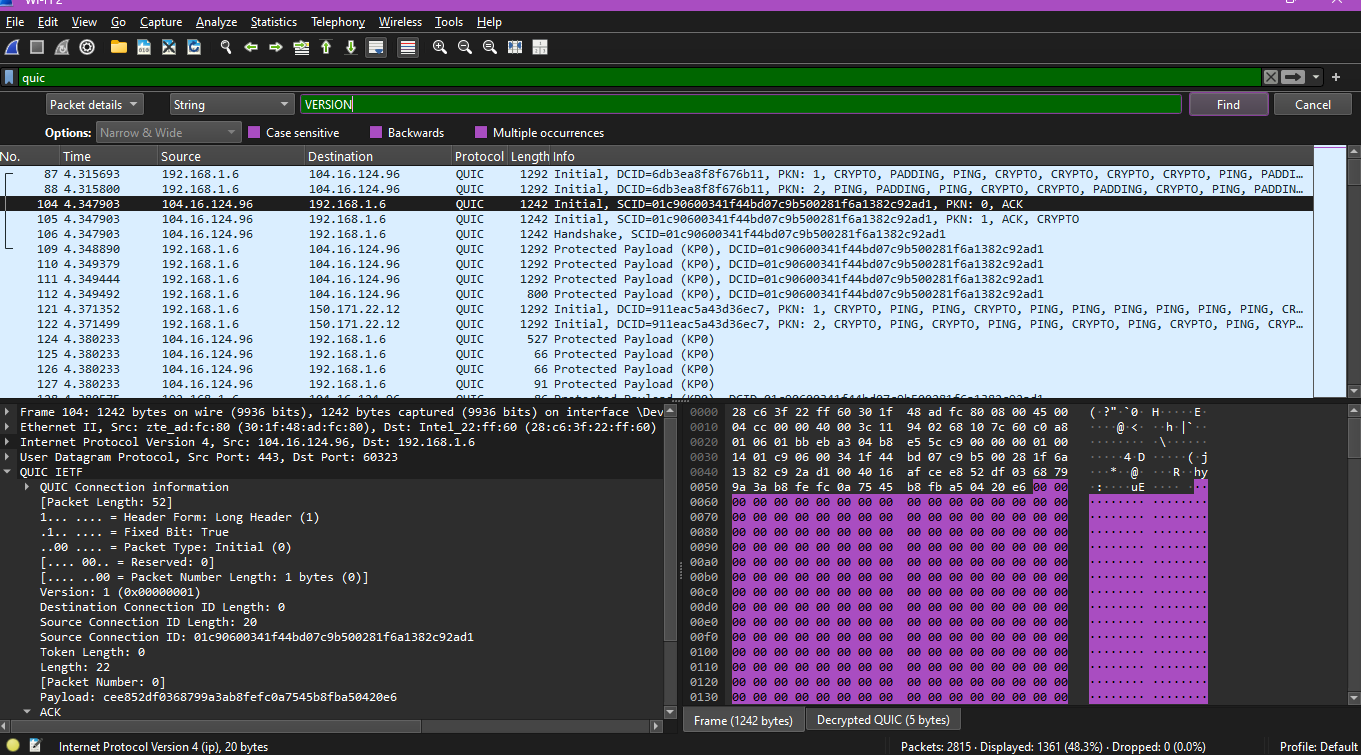
QUIC version is a number in the QUIC packet header that tells which version of the protocol is being used.

Different versions of QUIC may support new features, improved performance, or security fixes.

In Wireshark, it is usually in the Client Initial packet → QUIC Long Header → Version.

Example: 0x00000001 corresponds to QUIC v1, which is the standard version used by most websites like Cloudflare.

# **SCREENSHORT:**

****

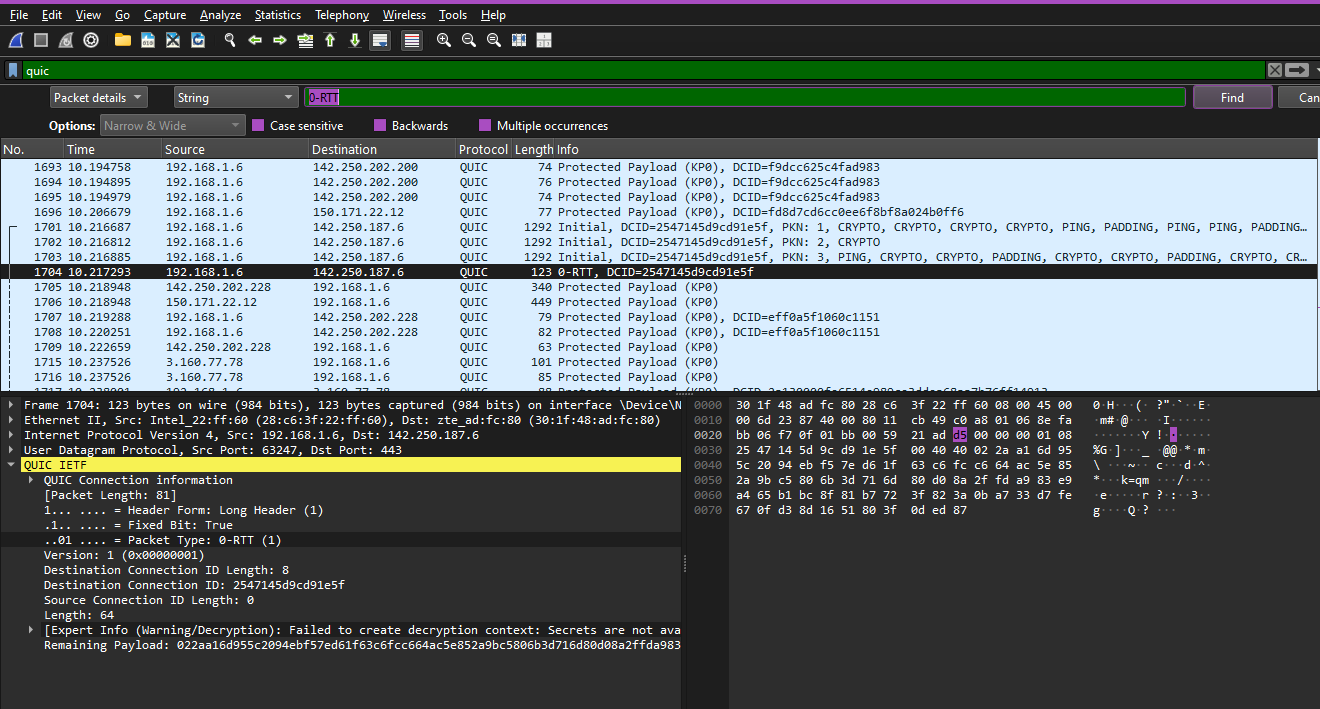
**Locate the packet where 0-RTT or 1-RTT keys are first used?**

# **First Use of 0-RTT / 1-RTT Keys:**

**What Are 0-RTT and 1-RTT Keys?**

* **0-RTT (Zero Round Trip Time) Keys**:
  + Optional early data encryption used **before the TLS handshake is fully completed**.
  + Allows the client to send data immediately to the server, reducing latency.
  + Not always supported, and data may be **replayed**, so it’s less secure.
* **1-RTT (One Round Trip Time) Keys**:
  + Fully established encryption keys used **after the handshake is complete**.
  + These keys protect all subsequent communication between client and server.
  + Indicates that secure communication has started.

# **SCREENSHORT:**

****

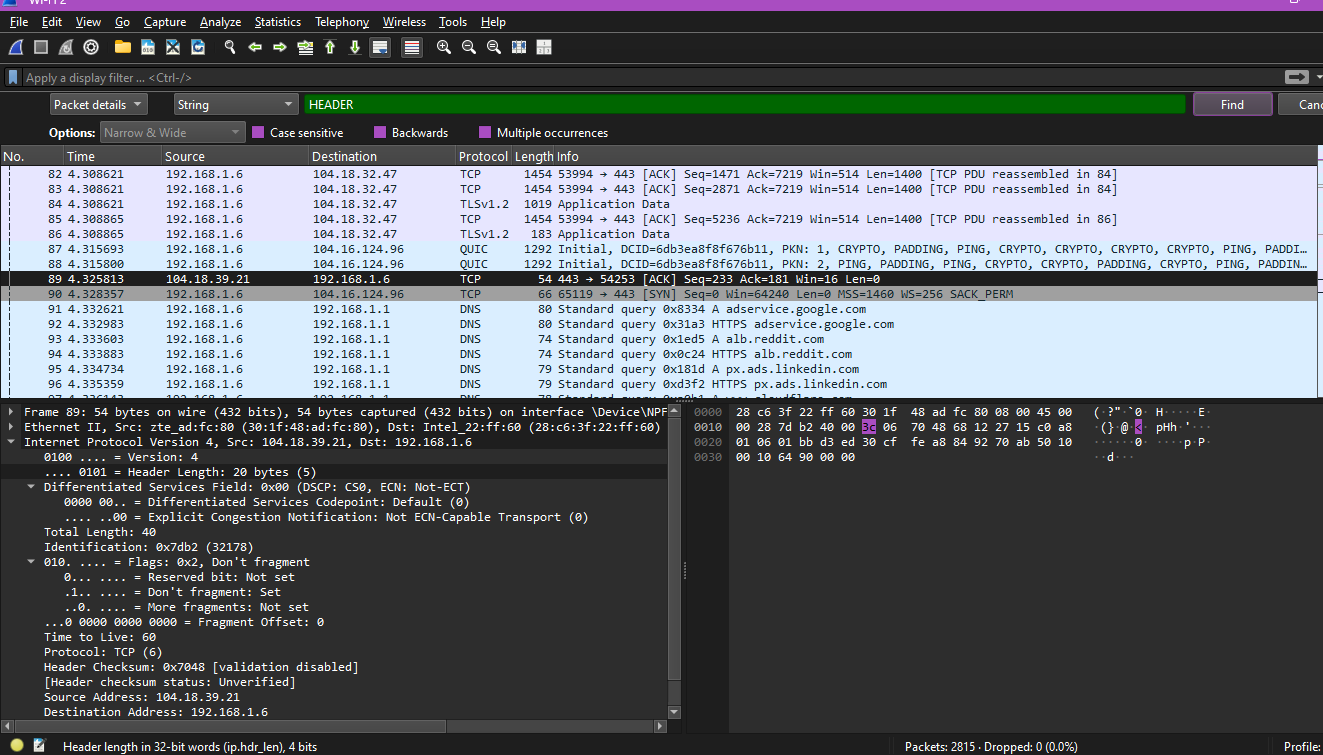
After the Client Initial and Handshake packets, the first QUIC STREAM packet or any packet labeled Protected (1-RTT) indicates the use of 1-RTT keys.

If 0-RTT is used, it will appear in the packet list as Protected (0-RTT).

You can use filters like quic.encryption\_level == 1-RTT or quic.encryption\_level == 0-RTT to locate these packets quickly.

**Find the first packet that carries application data (HTTP/3). How does this differ from HTTP over TCP?**

# **SCREENSHORT:**

****

The first packet carrying HTTP/3 application data is the first QUIC STREAM packet after the 1-RTT handshake (Packet #\_\_\_ on Stream #\_\_\_). Unlike HTTP over TCP, HTTP/3 runs on QUIC/UDP, integrates encryption with transport, supports multiple independent streams, reduces latency, and avoids head-of-line blocking.

# **CONCLUSION:**

In this analysis of QUIC-based website access ([www.cloudflare.com](http://www.cloudflare.com)), the **Client Initial packet** was identified as the first handshake packet, exchanging protocol versions, cipher suites, and extensions including the SNI (server name). The **TLS ClientHello** is embedded inside this QUIC Initial packet, initiating the secure handshake. The capture revealed that **QUIC v1 (0x00000001)** was used. The first packets using **1-RTT keys** mark the start of fully encrypted communication, while no 0-RTT packets were observed. The **first application data packet** occurs on a QUIC stream after 1-RTT keys are established, carrying HTTP/3 headers and data. Unlike HTTP over TCP, HTTP/3 over QUIC runs on UDP, integrates TLS, supports multiplexed streams, and avoids head-of-line blocking, providing faster and more reliable data transfer. Overall, QUIC significantly reduces latency while ensuring secure communication.