**Objective:**

To capture and analyze network traffic generated when accessing a website using the QUIC protocol (HTTP/3) in order to understand its key differences from traditional TCP-based HTTP.

**Executive Summary:**

This report documents the analysis of a QUIC-based connection to `www.youtube.com` using Wireshark. The capture was analyzed to identify key components of the QUIC handshake, the embedded TLS negotiation, and the exchange of application data. The analysis confirms QUIC's core advantages, such as combining the transport and cryptographic handshakes and eliminating head-of-line blocking at the transport layer.

**Task 1:**

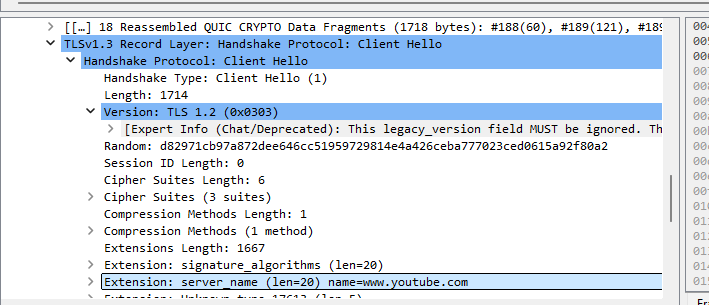
What is the name of the website?

**Answer:**

The website accessed was `**www.youtube.com`**

**Explanation:**

The website name was identified by locating the Server **Name Indication (SNI)** within the TLS Client Hello message. This message is embedded inside a QUIC CRYPTO frame. The SNI extension explicitly listed the target hostname as **`www.youtube.com**`.



**Task 2:**

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

**Answer:**

The Initial QUIC handshake packet was found in **Frame 188**. The information exchanged includes connection identifiers and the beginning of the TLS cryptographic handshake.

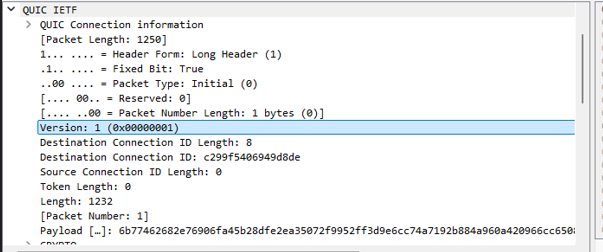
**Explanation:**

The Initial packet (identified by the Long Header with Packet Type: Initial (0)) contains:

- **Version Negotiation:** QUIC Version 1 (`0x00000001`).

- **Connection IDs:** A Destination Connection ID (`c299f54d6949d86e`) for connection identification.

- **TLS Handshake Initiation:** A CRYPTO frame carrying the initial portion of the TLS 1.3 Client Hello message, which starts the secure session negotiation.



**Task 3:**

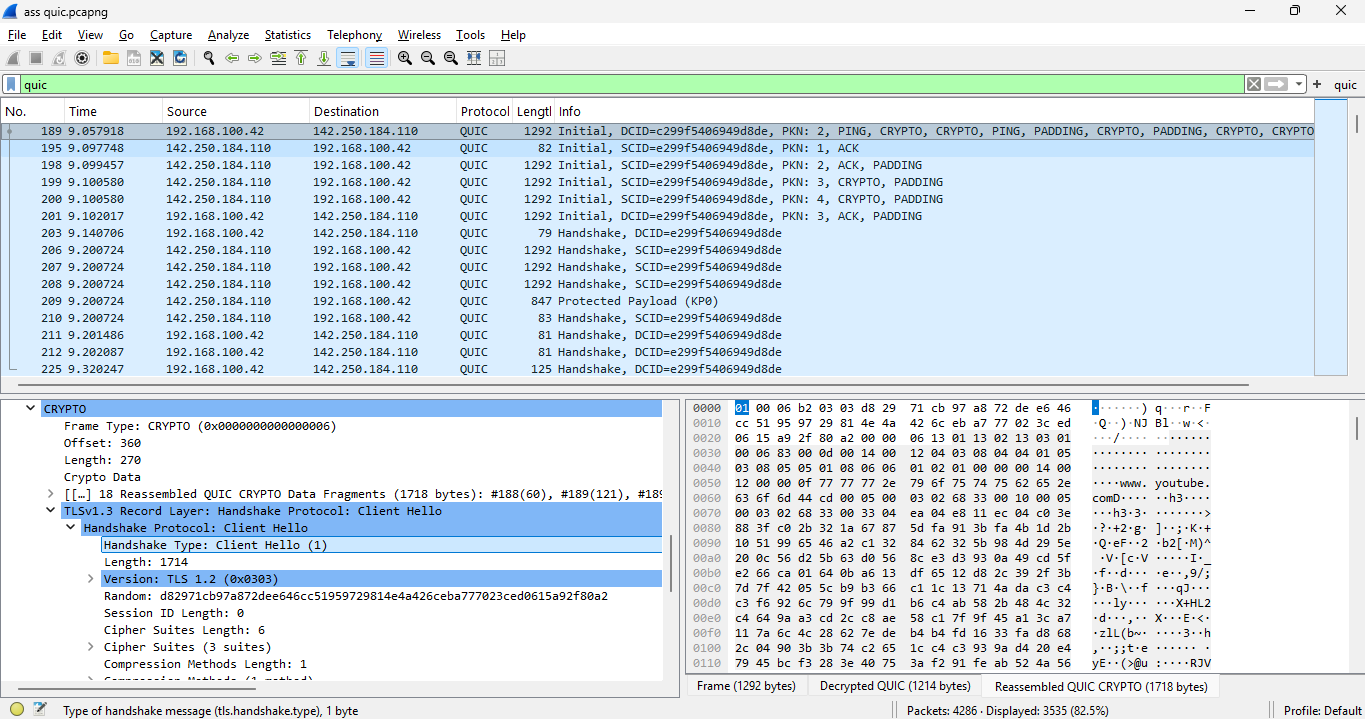
Identify the QUIC packet that contains the TLS ClientHello.

**Answer:**

The TLS ClientHello is contained within the same **Initial QUIC handshake packet (Frame 188)** identified in Question 2.

**Explanation:**

QUIC encapsulates the TLS handshake within its own protocol frames. The CRYPTO frame payload in the Initial packet was reassembled by Wireshark to reveal the complete TLSv1.3 Client Hello record. This demonstrates that QUIC integrates the cryptographic handshake directly into the transport layer connection setup.



**Task 4.**

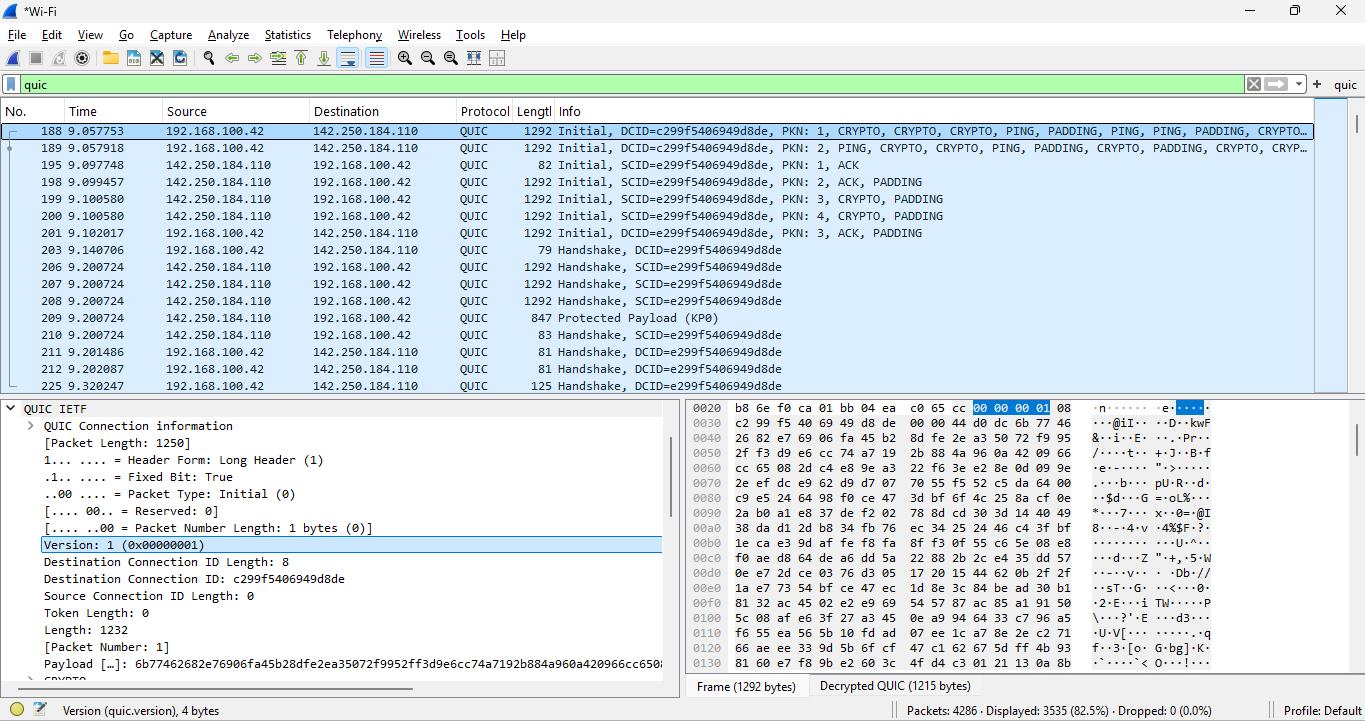
Which QUIC version is used in your trace?

**Answer:**

The QUIC version used is **IETF QUIC Version 1 (`0x00000001`).**

**Explanation:**

The version field in the Long Header of the Initial packet (Frame 188) clearly specifies the standardized version number `0x00000001`, indicating the use of the final IETF-standardized QUIC protocol.



**Task 5:**

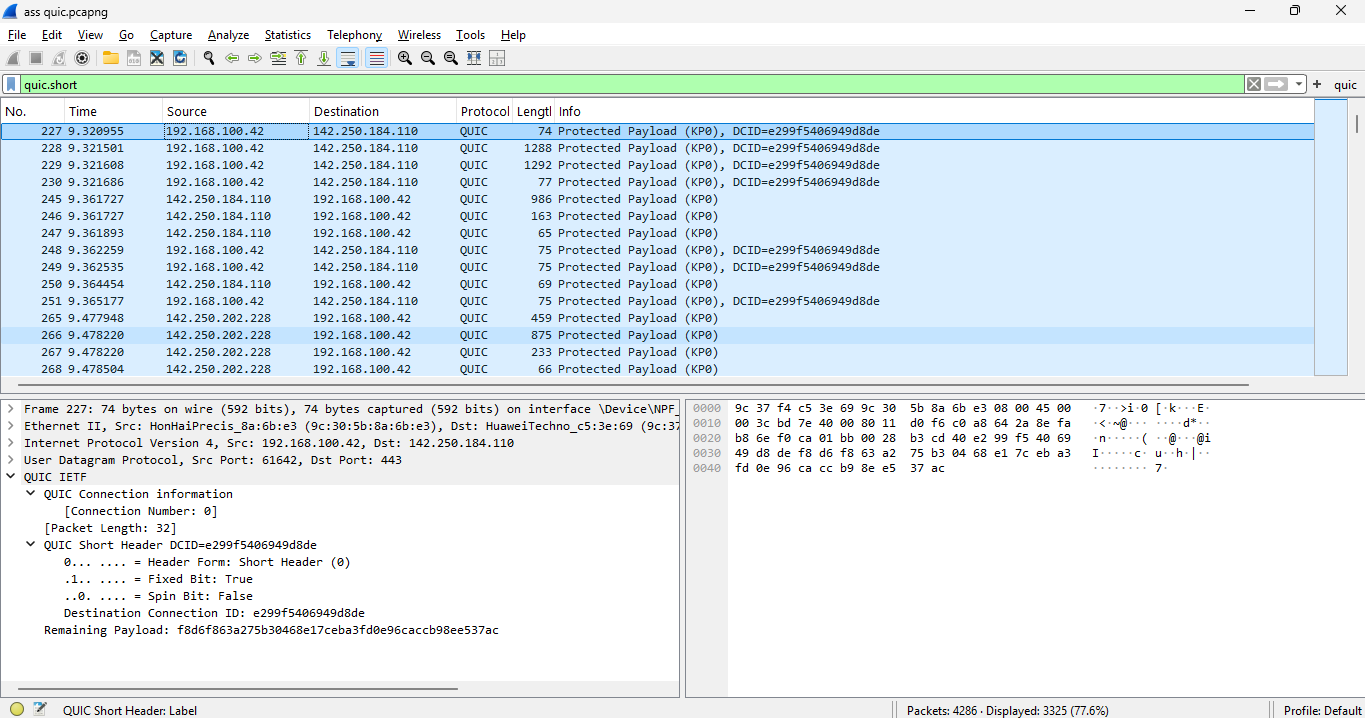
Locate the packet where 0-RTT or 1-RTT keys are first used.

**Answer:**

The **1-RTT keys are first used in Frame 227.**

**Explanation:**

This packet is identified by its **Short Header** (`Header Form: Short Header (0)`), which is used exclusively for packets protected with the final, forward-secure 1-RTT keys. This packet is sent by the client immediately after the cryptographic handshake is completed and contains the encrypted application data (HTTP/3 frames).



**Task 6**

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

**Answer:**

The first packet carrying HTTP/3 application data is **Frame 227**. The differences from HTTP over TCP are significant and fundamental to QUIC's design.

**Explanation:**

- **First Data Packet:** The application data is contained within the encrypted payload of the 1-RTT packet (Frame 227).

**- Key Differences from HTTP over TCP:**

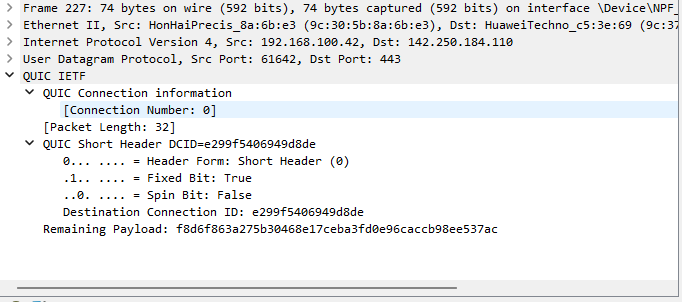
1**. Integrated Handshake:** QUIC combines the transport and cryptographic handshakes, reducing connection establishment latency (often to 1-RTT or even 0-RTT). In contrast, TCP requires a separate 3-way handshake followed by a TLS handshake.

2. **Transport Protocol:** QUIC is built on UDP, making it more flexible and faster to deploy than TCP, which is a more complex, kernel-space protocol.

3. **Elimination of Head-of-Line (HOL) Blocking:** QUIC provides per-stream reliability. A lost packet only affects the specific stream it belongs to, not the entire connection. In TCP, a single lost packet blocks all multiplexed HTTP/2 streams until it is recovered.

4. **Improved Connection Migration:** QUIC connections are identified by a Connection ID, allowing them to survive network changes (e.g., switching from Wi-Fi to mobile data), which breaks TCP connections.

5. **Enhanced Security:** QUIC encrypts most of its header fields by default, providing better privacy and protection against interference than TCP, where headers are sent in clear text.



**Conclusion:**

This practical analysis successfully captured and dissected a QUIC connection. The findings confirm the protocol's core architectural advancements: the integration of TLS for reduced latency, the use of streams to eliminate HOL blocking, and the overall efficiency gained by building a modern transport protocol on UDP. The transition from the initial handshake to secured application data in a single, streamlined process highlights the significant performance improvements QUIC offers over traditional TCP-based HTTP.