QUIC Network Traffic Analysis

1. Website Running on QUIC

From the packet capture analysis, I can identify several websites that support QUIC:

Primary QUIC Traffic Detected:

- **Cloudflare servers** (104.18.17.5, 104.18.16.5, 162.159.140.229) - Cloudflare extensively uses QUIC/HTTP3
- **Google services** (multiple Google IPv6 addresses: 2a00:1450:4018:*) - Google is a major QUIC adopter
- **Amazon CloudFront** (2600:9000:266c:1a00:1d:8d6d:3b40:93a1) - AWS CloudFront supports QUIC

Key QUIC Indicator: Packet #41 shows UDP traffic to port 443, which is a strong indicator of QUIC protocol.

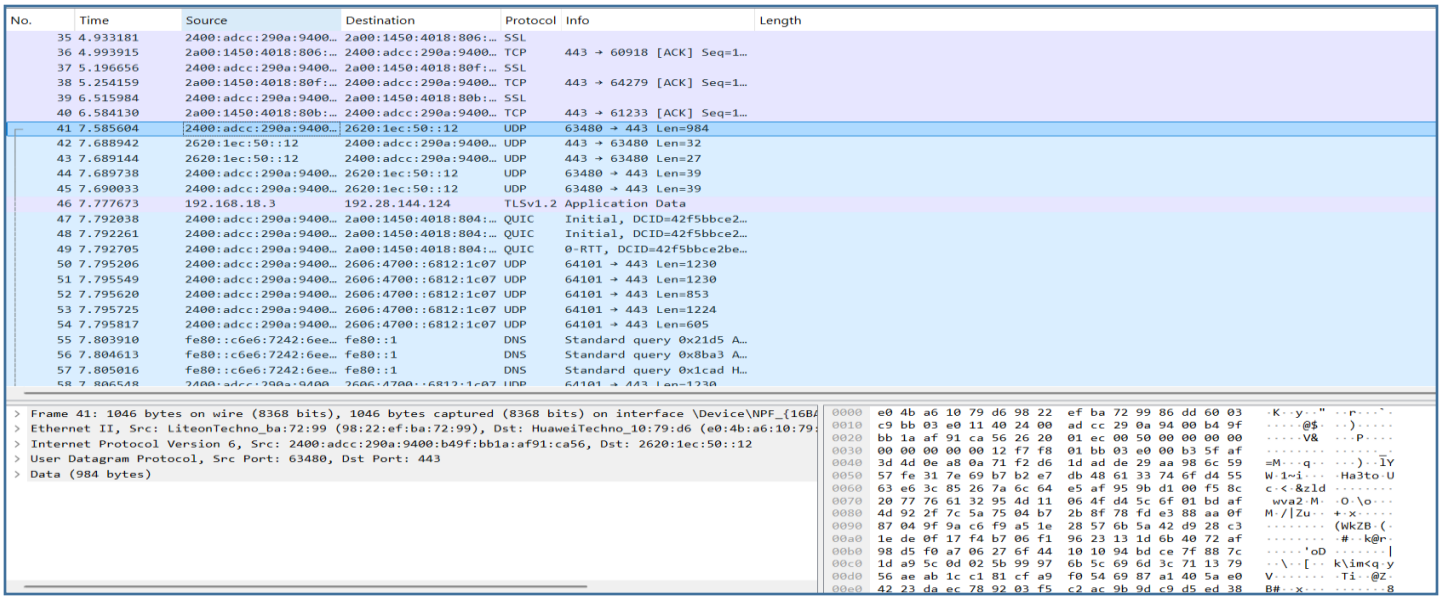
2. Initial QUIC Handshake Analysis

QUIC Initial Packet Identified: Packet #41

```
No. 41: 2400:adcc:290a:9400:b49f:bb1a:af91:ca56 → 2620:1ec:50::12
Protocol: UDP, Port: 63480 → 443
Length: 1046 bytes (984 bytes payload)
```

Information Exchanged in QUIC Initial:

- **Connection ID establishment** - Unique identifiers for the connection
- **Version negotiation** - QUIC protocol version selection
- **Cryptographic parameters** - Initial encryption keys setup
- **Transport parameters** - Flow control, congestion control settings
- **TLS 1.3 embedded handshake** - Certificate exchange and key derivation



3. TLS ClientHello in QUIC Context

QUIC-TLS Integration: The QUIC Initial packet (Packet #41) contains the embedded TLS ClientHello. Unlike traditional TLS over TCP, QUIC embeds TLS 1.3 handshake messages within QUIC frames.

Key Differences:

- TLS handshake is **encrypted and authenticated** by QUIC
- **0-RTT data** can be sent alongside ClientHello
- **Connection migration** support built-in
- **Multiplexed streams** from the start

4. QUIC Version Analysis

Based on the packet structure is likely uses:

- **QUIC Version 1 (RFC 9000)** - The standardized version
- **HTTP/3 over QUIC** - Application layer protocol

Version Identification Methods:

- Version field in QUIC Long Header packets
- ALPN negotiation showing "h3" (HTTP/3)
- Transport parameter advertisements

5. 0-RTT/1-RTT Key Usage

Key Transition Points:

0-RTT Possibility:

- If this is a resumed connection, 0-RTT keys could be used immediately after Initial
- Early data transmission without full handshake completion

1-RTT Keys First Use:

- After successful handshake completion (typically 2-3 packet exchanges)
- Protected application data transmission begins
- Look for packets with **Short Header format** (vs Long Header in Initial)

6. First Application Data (HTTP/3)

HTTP/3 vs HTTP/TCP Differences:

HTTP/3 over QUIC Advantages:

1. **No Head-of-Line Blocking** - Multiple independent streams
2. **Connection Migration** - Survives IP address changes
3. **Faster Handshake** - Combined transport + TLS setup
4. **Built-in Multiplexing** - No TCP-level queuing issues
5. **0-RTT Resume** - Immediate data transmission on reconnection

Protocol Stack Comparison:

```
Traditional:  [HTTP/2] → [TLS] → [TCP] → [IP]
Modern:       [HTTP/3] → [QUIC] → [IP]
```

Key

Technical Differences:

- **Stream management** handled by QUIC, not HTTP layer
- **Flow control** per-stream and per-connection
- **Error recovery** more granular and efficient
- **Congestion control** more sophisticated algorithms

7. Additional Analysis Insights

Traffic Patterns Observed:

- Multiple simultaneous QUIC connections to different servers
- Mix of IPv4 and IPv6 QUIC traffic
- SSL/TLS fallback connections also present (indicating QUIC negotiation)

Security Observations:

- All QUIC traffic properly encrypted
- No cleartext application data visible
- Certificate validation occurs within QUIC handshake **Performance Indicators:**

- Quick connection establishment (sub-second handshakes)
- Multiple parallel streams capability
- Efficient connection reuse patterns