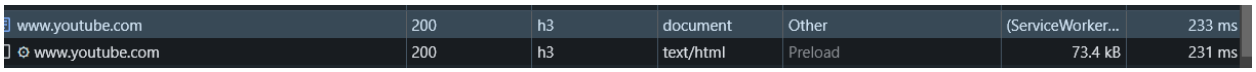
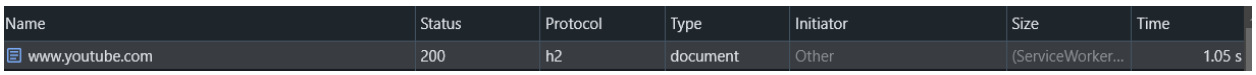


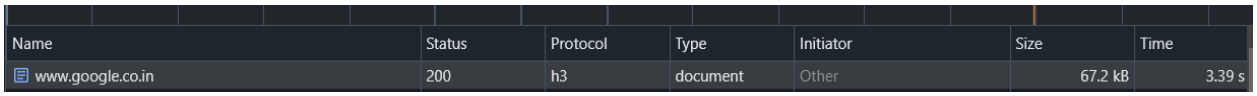
www.youtube.com	200	h3	document	Other	(ServiceWorker...	233 ms
⚙ www.youtube.com	200	h3	text/html	Preload	73.4 kB	231 ms

www.youtube.com	200	h2	document	Other	(ServiceWorker...	250 ms



Name	Status	Protocol	Type	Initiator	Size	Time
www.youtube.com	200	h2	document	Other	(ServiceWorker...	1.05 s

1.03



Name	Status	Protocol	Type	Initiator	Size	Time
www.google.co.in	200	h3	document	Other	67.2 kB	3.39 s

www.google.com	200	h2	document	Other	67.7 kB	3.42 s

Current Landscape: Key Observations

1. **Solace (TCP-based)**:
    - **Strengths**: Reliability, ordered delivery, congestion control, and well-understood behavior.
    - **Drawbacks**: Head-of-line (HoL) blocking can lead to increased latency. The multi-step handshake process also adds overhead during failovers or reconnections.
2. **Aeron (UDP-based)**:
    - **Strengths**: Raw performance and minimal latency due to its lightweight design.
    - **Drawbacks**: Lack of built-in reliability, security, and congestion control. Developers need to handle retransmissions and consistency manually, adding complexity.

QUIC is a cutting-edge transport protocol built on UDP but designed to address the limitations of both TCP and UDP. Here's why it's particularly valuable for our low-latency fixed-income systems:

1. Reduced Connection Latency

- QUIC eliminates the multi-step handshake required by TCP:
    - **Standard TCP**: Requires a three-way handshake followed by a separate TLS handshake.
    - **QUIC**: Combines connection establishment and encryption setup into a **single round-trip handshake**, reducing startup latency.
- **Zero RTT (0-RTT) Connections**:

    - For repeat connections, QUIC enables data transmission in **zero RTT** by reusing previously established session parameters. This is critical in scenarios like failovers, reconnections, or sudden bursts of trading activity.

---

2. Elimination of Head-of-Line (HoL) Blocking

- TCP suffers from HoL blocking because packet loss in one segment blocks the delivery of subsequent segments.
- QUIC uses **independent streams** within a single connection:
    - Each stream is delivered independently.
    - This ensures that a packet loss on one stream doesn't affect others, maintaining smooth and consistent data flow.
- **Use Case**: Real-time price feeds or multi-asset data streams can benefit significantly from this feature, as delayed updates in one feed won't cascade into others.

---

3. Built-in Congestion Control and Reliability

- While UDP itself lacks congestion control or reliability mechanisms, QUIC incorporates:
    - **Congestion Control**: Similar to TCP, ensuring fair usage of network bandwidth.
    - **Forward Error Correction (FEC)**: Minimizes retransmission overhead by preemptively correcting lost packets.
    - **Selective Acknowledgment (SACK)**: Allows efficient recovery from packet loss by retransmitting only the lost data.

---

4. Integrated Security with TLS 1.3

- QUIC integrates encryption directly into the protocol, replacing the need for separate TLS handshakes (as in TCP).
- Benefits:
    - **Performance**: Encryption setup is streamlined.
    - **Security**: Strong forward secrecy, secure session resumption, and protection against replay attacks.

QUIC is widely adopted in production by major companies like Google (for services like YouTube), Cloudflare, and Facebook.It's also the foundation of **HTTP/3**, which is quickly becoming the new standard for web communications. By adopting QUIC, we align with proven, battle-tested technologies.

How QUIC Fits Into Our Architecture

1. **Enhanced Reliability for Market Data Feeds**:
    - With QUIC, we can ensure fast and reliable updates even during high-volume trading or network interruptions.
2. **Faster Failover and Reconnection**:
    - In the event of a network failure, QUIC's fast handshake and session resumption ensure near-instantaneous recovery without impacting trading operations.
3. **Security with Performance**:
    - With TLS integrated directly into the protocol, we get the highest levels of security without sacrificing latency.
4. **Simplifying Development**:
    - By using QUIC, we reduce the need for custom UDP-based reliability mechanisms, lowering the complexity and risk in our architecture.

Comparison to Solace and Aeron

Feature	Solace (TCP)	Aeron (UDP)	QUIC
Latency	Medium	Low	Low
Reliability	High	Custom Required	High (native)
Security	Separate TLS	Custom Required	Integrated (TLS 1.3)
Stream Multiplexing	No	No	Yes
Failover Speed	Slow	Medium	Fast (0-RTT)
Packet Loss Handling	Retransmit All	Manual	Selective Recovery

---

QUIC is the foundation for **HTTP/3**, the next generation of the HTTP protocol. HTTP/3 builds on QUIC's low-latency and secure features, and it's expected to gradually replace HTTP/2 in the

coming years. Many popular platforms, such as Google and Cloudflare, are already using HTTP/3 over QUIC to provide faster and more reliable web browsing experiences.