HTTP/3 and QUIC

Table of Content

- What is HTTP/3?
- What is QUIC?
- How Does HTTP/3 Work?
- HTTP/3 vs HTTP/2 vs HTTP/1
- Benefits of HTTP/3 and QUIC
- Limitations of HTTP/3 and QUIC
- Is HTTP/3 Available Now?

What is HTTP/3

- HTTP/3 is the latest version of the HTTP (Hypertext Transfer Protocol), Released in June 2022.

- HTTP/2 introduces features like **multiplexing**, **header compression**, and **server push** but was still limited by the underlying TCP.

- HTTP/3 addresses the limitation of using TCP, by using **QUIC** which promises faster, more reliable connections, particularly over networks with variable performance, such as **mobile** and **wireless networks**.

- The Internet Engineering Task Force (IETF) enhances web performance and security (RFC 9114).

What is QUIC

- Stands for **Quick UDP Internet Connections**
- Transport layer protocol developed by Google.
- Initially developed to address TCP’s shortcomings, especially in speed and performance (*slow start*, *multiple round trips*, *series of handshakes and setup processes*).
- QUIC is a connectionless protocol that aims to establish connections **faster** and **reduce latency** by utilizing UDP.
- Google started working on the QUIC protocol around 2012
- And then adopted by the IETF as the basis for HTTP/3.
- Integrates TLS for end-to-end encryption, simplifying the protocol stack and enhancing security.
- Advanced congestion control algorithms and mechanisms for handling packet loss without causing significant delays.

HTTP/3 History

version	Date	Specification	Status	Key features
HTTP/3	2022	RFC 9114	Proposed Standard	makes HTTP compatible with QUIC, moves from TCP to UDP transport
HTTP/2	2015	RFC 9113	Proposed Standard	introduces a new binary framing layer that's not compatible with HTTP/1.1 and request and response multiplexing, stream prioritization, automatic header compression(HPACK), connection reset, server push
HTTP/1.1	1997	RFC 9112	Internet Standard	update to HTTP/1; introduces the Host header, the 100 Continue status, persistent connections, and new HTTP methods (PUT, PATCH, DELETE, CONNECT, TRACE, OPTIONS)
HTTP/1	1996	RFC 1945	historical (not in use)	introduced HTTP status codes, Content-Type, the POST and HEAD methods, and request headers
HTTP/0.9	1991	has no RFC number; see the original doc created by Tim Berners-Lee	historical (not in use)	only raw data transfer introduced the TCP/IP model and GET requests (also called the 'one-line protocol')

HTTP/3 – Protocol Stack

Robin Marx: H3 Protocol Stack; GitHub

HTTP/3 – How it Works

1. **Connection Establishment**: When a client wants to communicate with a server, a handshake is initiated using the QUIC transport protocol. This handshake is designed to establish a connection faster than traditional methods.

2. **Connection IDs**: During the handshake, the client and server exchange connection IDs. These IDs help maintain the connection state and ensure continuity even when IP addresses change.

3. **Single TCP Connection**: Unlike HTTP/2 (head-of-line blocking), HTTP/3 processes each stream independently using the QUIC transport protocol. This allows each data stream to be processed separately without waiting for others => enhancing efficiency.

4. **Multiple Requests**: After establishing the connection, the client can send various requests simultaneously over the same single TCP connection. Each request is multiplexed, transmitting in parallel, reducing latency and improving load times.

5. **Connection Migration**: If the client changes networks (such as switching from Wi-Fi to mobile data), connection migration ensures the connection persists without interruption (due to connection IDs, which help maintain the session even when IP addresses change).

HTTP/3 & QUIC– Benefits

HTTP/3 and QUIC encryption provides

- Faster page load times due to reduced handshake latency & multiplexed streams.
- Enhanced security features like built-in encryption provide robust protection against common threats.
- Additionally, improved connection reliability and efficient resource utilization ensure optimal performance even in challenging network conditions.

Faster Page Load Time

- Using QUIC’s faster connection establishment => bypass the conventional TCP handshake process => reducing initial load times.
- QUIC’s connection establishment occurs within a **single round-trip time (RTT)**, drastically lowering latency.
- QUIC uses UDP => allows more efficient error correction and packet retransmission strategies.
- Lost packets are retransmitted without disrupting the order of other packets, further reducing latency.
- When combined with HTTP/3’s optimized header compression and prioritization, this significantly enhances page load performance, making web applications more responsive and user-friendly.

HTTP/3 & QUIC– 0 RTT

- First connection (1 RTT)

  1. Client and server perform a full handshake
  2. Server sends session ticket, transport parameters
  3. Client stores them

- Reconnection (0 RTT)

  1. Client sends initial ticket, encrypted early data
  2. Server decrypts using saved session info
  3. Normal connection continues

Enhanced Security Features

- Transport layer security in QUIC is seamless and robust. It employs forward secrecy to ensure that past session data remains secure even if long-term keys are compromised.
- The encryption algorithms used are state-of-the-art, providing strong cryptographic assurances continually updated to counter emerging threats.
- QUIC uses authenticated encryption that ensures the confidentiality and integrity of data.
- HTTP/3 prioritizes security, making it resilient to various attacks, such as man-in-the-middle and replay attacks.

Improved Connection Reliability

- HTTP/3 and QUIC significantly enhance connection reliability by minimizing latency and optimizing packet delivery.

- Connection migration: switching between networks (Wi-Fi -> cellular), QUIC maintains active connections without requiring a complete handshake renegotiation.

- QUIC also employs **forward error correction (FEC)** techniques to enhance reliability. FEC allows the receiver to reconstruct lost packets without retransmissions, thus maintaining a smooth data stream.

Efficient Resource Utilization

- HTTP/3 and QUIC optimize resource utilization by minimizing redundant data transmissions and enhancing congestion control mechanisms.
- These protocols employ multiplexing to handle multiple requests simultaneously over a single connection, avoiding the head-of-line blocking that plagued HTTP/2.
- QUIC’s advanced congestion control algorithms dynamically adjust data flow based on real-time network conditions, ensuring optimal throughput without overwhelming the network.
- This intelligent management of data packets enhances reliability and speed, particularly in fluctuating network environments.
- HTTP/3 uses the **QPACK compression algorithm** => significantly reducing the overhead associated with HTTP headers.

  Compressing headers more efficiently => minimizes the amount of data transmitted, conserving bandwidth and decreasing the time required to establish connections.

Limitations of HTTP/3 and QUIC

Compatibility With Existing Systems

- QUIC's integration into existing network infrastructures may be tricky due to its reliance on UDP rather than the more traditionally used TCP

Network Congestion Impact

- QUIC's rapid retransmission strategies aim to minimize latency, yet these same strategies can lead to increased network congestion (e.g. retransmission).
- Eliminating the head-of-line blocking at the transport layer, but its congestion control mechanisms may still introduce inefficiencies (e.g. out of order)

Performance Under Load

- Can exhibit bottlenecks when establishing numerous simultaneous connections
- Latency reduction, but it also introduces potential packet loss issues, which can degrade performance under network congestion.

Is HTTP/3 Available Now?

Web Servers

1. NGINX
2. Apache (via mod_http3)
3. LiteSpeed
4. Caddy

Content Delivery Networks

1. Cloudflare
2. Akamai
3. Fastly
4. Amazon CloudFront
5. Google Cloud CDN
6. Microsoft Azure CDN