Study on QUIC Protocol

Anil Paudel (077BCT010)

December 2024

Overview

The QUIC protocol, pronounced "quick", stands for Quick UDP Internet Connections. It was developed by Google in 2012 to address latency issues and improve the speed of web applications that use Transmission Control Protocol (TCP). Initially introduced as an experimental protocol, QUIC aimed to optimize HTTP-based communication over the internet by combining the speed of UDP with the reliability and security features typically associated with TCP. Google integrated QUIC into its Chrome browser and servers, and its success led the Internet Engineering Task Force (IETF) to standardize it as RFC 9000 in May 2021. Although its name was initially derived from the acronym for 'Quick UDP Internet Connections', the IETF's use of the word QUIC is not an acronym: it is simply the name of the protocol. According to the statistics of W3Techs, about 25.5% of websites use HTTP/3 for the time being.

Introduction

QUIC is a transport-layer network protocol built on UDP (User Datagram Protocol) instead of the more traditional TCP (Transmission Control Protocol). Unlike TCP, which prioritizes reliability and ordered data delivery through mechanisms like handshake and retransmissions, QUIC leverages UDP's simplicity to provide faster, connection-oriented communication with reduced overhead. QUIC incorporates advanced features such as multiplexing, encryption, and flow control directly into the protocol, combining elements of TCP, TLS, and HTTP/2.

Major Milestones

- 2012: The design document of the QUIC protocol was released.
- 2013: Google began internal testing of QUIC and prepared to integrate it into the Chrome browser.
- 2014: Google considered gradually deploying QUIC on a large scale.
- 2017: QUIC was used by almost all Chrome users.
- 2021: The IETF announced the QUIC standard RFC9000, and HTTP/3 is based on the QUIC protocol

Before the release of QUIC, TCP was used as the underlying protocol for transferring data in HTTP. However, as the mobile internet continues to develop, there is an increasing demand for real-time interactions and more diverse network scenarios. Additionally, smartphones and portable devices have become increasingly mainstream, with over 60% of internet traffic currently transmitted wirelessly.

Why does QUIC seem important?

However, the traditional TCP, a transport-layer communication protocol that has been in use for over 40 years, has inherent performance bottlenecks in the current context of large-scale long-distance, poor mobile networks, and frequently network switching, which cannot meet the demands mainly due to the following three reasons:

- Large Handshake Delay in Establishing Connections
- Head-of-line blocking
- Updating TCP is not that easy

HOW QUIC differs from Traditional Protocols?

The main differences between QUIC and Traditional Protocol lie in their design goals, working principles and adaptability to modern network application needs.

Based on UDP

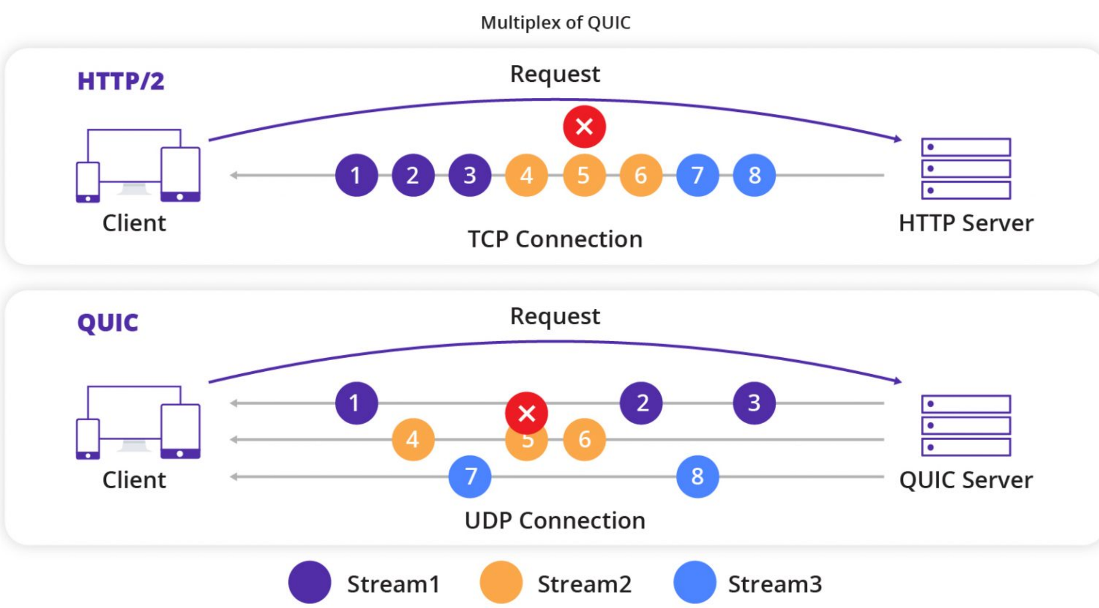
The QUIC protocol chooses UDP as the underlying protocol, rather than being based on IP like TCP. The simplicity of UDP allows QUIC to avoid the complex and sometimes unnecessary features of TCP, such as the three-way handshake and congestion control algorithms, thereby achieving lower network latency.

Multiplexing

In the TCP protocol, data streams on a TCP connection are processed sequentially, and the loss of a previous packet affects the transmission of subsequent packets, known as "head-of-line blocking." The QUIC protocol introduces the concept of "streams," allowing multiple data streams to be transmitted in parallel over the same connection, with each stream being independent and unaffected by packet loss in other streams.

Multiplexing in QUIC is achieved using stream identifiers, which are unique to each stream within a connection. Stream frames carry these identifiers, enabling endpoints to demultiplex data into the appropriate stream. This mechanism allows a single QUIC connection to support diverse application-level interactions simultaneously, such as transferring files, making API calls, or streaming multimedia content.

The multiplexing of QUIC is similar to HTTP/2. Multiple HTTP requests (streams) can be sent concurrently on a single QUIC connection. However, what makes QUIC's multiplexing surpass HTTP/2 is that there is no sequential dependency between each stream on a single connection. That means if stream 2 loses a UDP packet, it only affects the processing of stream 2, without blocking the data transmission for stream 1 and 3. As a result, this solution does not lead to Head-of-Line Blocking.



For example, if a client sends packets 1 and 2 using IP1 and then switches networks, changing to IP2 and sending packets 3 and 4, the server can recognize that all four packets are from the same client based on the Connection ID field in the packet header. The fundamental reason why QUIC can achieve connection migration is that the underlying UDP protocol is connectionless.

Fast Handshake and Closure

The handshake process of the QUIC protocol is very rapid because it combines multiple steps into a single message, reducing the number of round trips. Additionally, QUIC provides the ability to quickly close connections, allowing resources to be released immediately once data transmission is complete.

The QUIC handshake is designed to minimize latency and enable secure communication quickly. This is achieved by integrating the transport layer and cryptographic handshake into a single round-trip or even zero round-trips in certain cases.

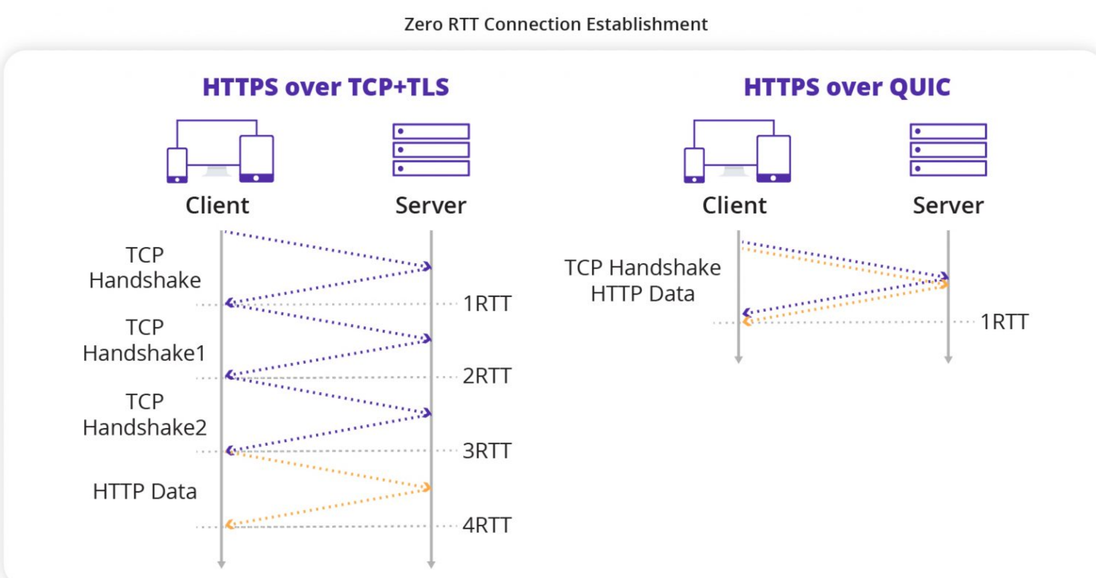
1-RTT Handshake

For a typical connection where the client has no prior interaction with the server, QUIC performs a 1-RTT (One Round-Trip Time) handshake. During this process:

- The client sends its initial packet, which includes cryptographic information using TLS 1.3, and the server responds with its cryptographic parameters.
- This simultaneous exchange of data allows the handshake to complete in one round-trip, after which application data can be sent securely.

0-RTT Handshake

QUIC supports 0-RTT (Zero Round-Trip Time) resumption for clients that reconnect to a server with which they have previously communicated. The client sends application data with its initial packet, leveraging previously shared cryptographic information. This drastically reduces latency for reconnecting clients but requires careful handling of replay attacks, as early data might be retransmitted maliciously.



QUIC provides mechanisms for graceful and abrupt connection termination, ensuring efficient resource cleanup and proper notification to both endpoints.

Connnection Migration

In unstable network environments, such as when a user switches from one Wi-Fi network to a mobile data network, the QUIC protocol can seamlessly migrate connections without needing to re-establish them. This feature is crucial for providing a seamless user experience. TCP connections are based on a 4-tuple: source IP, source port, destination IP, and destination port. If any of these changes, the connection must be reestablished. However, QUIC connections are based on a 64-bit Connection ID, which allows the connection to be maintained as long as the Connection ID remains the same without disconnection and reconnection. Each connection uses one or more connection identifiers, which are independently chosen by each endpoint. The key roles of CIDs include:

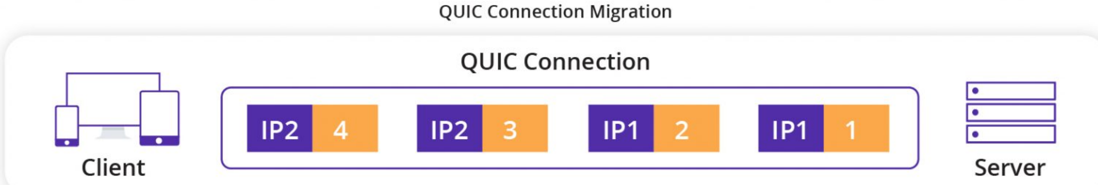
- **Network Path Migration:** Connection IDs allow a QUIC connection to seamlessly migrate to a new network path. For instance, a mobile device switching from Wi-Fi to cellular can maintain its connection without disruption.

- **Packet Routing:** CIDs ensure packets are routed correctly to the intended endpoint despite changes in addressing at lower protocol layers (e.g., UDP and IP).

- **Privacy**: Multiple connection IDs help conceal the association between packets and a connection, reducing the ability of observers to track connections across network changes.

For example, if a client sends packets 1 and 2 using IP1 and then switches networks, changing to IP2 and sending packets 3 and 4, the server can recognize that all four packets are from the same client based on the Connection ID field in the packet header. The fundamental reason why QUIC can achieve connection migration is that the underlying UDP protocol is connectionless.

QUIC Connection Migration



Security

The QUIC protocol integrates TLS 1.3, the most advanced encryption protocol currently available. TLS 1.3 provides robust encryption capabilities and fast and secure handshakes, enabling QUIC to maintain low latency while ensuring data security. Encryption by default: Every QUIC connection is encrypted, combining the features of TLS and transport protocols to ensure security from the start.

Error Handling and Recovery

The QUIC protocol has designed an efficient error handling and recovery mechanism. When packet loss occurs, QUIC does not immediately reduce the sending rate but attempts to recover lost packets by increasing retransmissions. This approach reduces performance degradation caused by network fluctuations. QUIC uses forward error correction (FEC) and intelligent retransmission to handle packet losses efficiently without disrupting the flow of unaffected streams.

Applications and Adoption

QUIC powers HTTP/3, the latest version of the Hypertext Transfer Protocol. It is widely adopted by major technology companies, including Google, Cloudflare, Facebook, and Microsoft. Popular applications like YouTube, Google Search, and Chrome already use QUIC to deliver faster and more reliable user experiences.

Web Application

In web applications, the QUIC protocol can significantly improve browser loading speeds and reduce page rendering times. This is important to improve user experience. Currently, many mainstream browsers support the QUIC protocol, such as Google Chrome and Mozilla Firefox.

Real-time Audio and Video Communication

Real-time audio and video communication require extremely low latency and high stability. The low latency characteristic of the QUIC protocol makes it highly valuable in the field of real-time audio and video communication. For example, videoconferencing and online voice calls can improve communication quality by adopting the QUIC protocol.

Online Gaming

Online gamers are very sensitive to network latency. The QUIC protocol can reduce the transmission latency of game data packets, improving the response speed and stability of games. This is important to improve the gaming experience of players.

Internet of Things (IoT) Devices

With the rapid development of IoT technology, more and more devices need to connect to the internet. The QUIC protocol is suitable for the connection needs of numerous IoT devices, reducing network congestion and device energy consumption. This is significant for promoting the development of the IoT industry.

Conclusion

QUIC is a game-changing protocol that overcomes the limitations of TCP, offering faster connections, built-in encryption, and efficient multiplexing. As the backbone of HTTP/3, it improves web performance, especially for latency-sensitive applications like streaming and gaming. Its ability to handle changes in the mobile network and improve reliability makes it vital for modern internet use. Looking ahead, QUIC is poised to drive greater adoption in industries, optimize edge computing, and inspire further innovations in network protocols. Despite challenges such as UDP compatibility and resource overhead, its future is bright, promising faster, safer, and more resilient communication across the web.