Performance Comparison of QUIC and TCL+TLS Protocols in a File Transfer Application

1 Introduction

1.1 Background

Secure communication is a cornerstone of the modern Internet, enabling privacy, authentication, and data integrity. Two key protocols, Transport Layer Security (TLS) and QUIC, play vital roles in this domain.

Transport Layer Security (TLS). TLS is a cryptographic protocol designed to provide secure communication over the Internet by encrypting data and authenticating the identity of communicating parties. It operates on top of the Transport Control Protocol (TCP) and is used in a wide range of applications, including web browsing, email, messaging, and file transfers.

There are three main components to what the TLS protocol accomplishes:

- Encryption: ensures the confidentiality of data exchanged between a client and a server.
- Authentication: verifies the identity of the server, and optionally the client, through digital certificates.
- Integrity: protects against data tampering by using message authentication codes (MACs).

While TLS has undergone significant improvements since its inception (from SSL to TLS 1.3), its reliance on TCP introduces performance limitations, such as latency caused by the sequential handshakes of both protocols.

QUIC. QUIC (Quick UDP Internet Connections) is a modern transport protocol developed by Google and later standardized by the IETF. Unlike TLS, which operates as an independent protocol layered on top of TCP, QUIC integrates both the transport and security layers directly into its design, building on User Datagram Protocol (UDP) as its foundation. QUIC is the default transport protocol for HTTP/3, the latest version of the HTTP protocol.

The key features of QUIC include:

- Built-in Security: incorporates TLS 1.3 for encryption and authentication, eliminating the need for a separate handshake layer.
- Reduced Latency: combines the transport and security handshake into a single exchange, significantly reducing connection setup time.
- Multiplexing Streams: unlike TCP, QUIC allows multiple data streams within a single connection, avoiding head-of-line blocking when packet loss occurs.

Both TLS and QUIC aim to ensure secure communication, but they achieve this in fundamentally different ways. TLS focuses solely on encryption and authentication, relying on an underlying transport protocol like TCP. In contrast, QUIC integrates security and transport into a single protocol, offering not only robust encryption but also performance optimizations. The differences between these approaches have profound implications for latency, throughput and resilience, which will be explored in the later sections.