

QUIC is a general-purpose transport layer network protocol proposed as the replacement for TCP and TLS in HTTP/3. QUIC is based on UDP and provides always-encrypted connections able to transmit multiple streams of data in parallel. Compared to TCP, QUIC promises lower latency, better congestion control flexibility, and a solution to head-of-line blocking occurring in multiplexed HTTP/2 connections.

The latest release of .NET — .NET 5 — has shipped with experimental support for QUIC based on the MsQuic C library. However, when implementing new features in standard .NET libraries, purely .NET implementations are preferable to adding dependencies on native libraries because .NET implementations offer better maintainability and — in some cases — even better performance. This thesis explores the viability of a purely C# QUIC implementation as a future replacement for .NET 6 or later release.

This thesis's result is a fork of the official .NET runtime repository with partial C# implementation of the QUIC protocol. We implemented a subset of the QUIC specification, which is sufficient for a basic performance evaluation. As part of the thesis, we have benchmarked the throughput and latency of our and the MsQuic-based QUIC implementation and compared them to TCP's performance in two environments: LAN and simulated cellular network. While the benchmarking results show that our implementation is slower than the MsQuic-based one, we identified the primary performance limiting factors in our implementation and suggested the course for future development.