

INTEGRATION OF THE QUIC PROTOCOL INTO THE SEASTAR FRAMEWORK

A native, user-space QUIC transport for the shared-nothing engine behind ScyllaDB.

Kamil Dalidowicz Stanisław Kalewski Adam Karaczewski Piotr Korcz • supervisor: dr hab. Jakub Pawlewicz



ScyllaDB

POWERED BY

SEASTAR.

THE PROBLEM

Distributed systems push their RPC over **TCP**: a single byte stream where one lost packet stalls everything (*head-of-line blocking*), plus a handshake on every connection. Seastar already bypasses the kernel for speed — yet still rides TCP.

OUR SOLUTION

The **first native, user-space QUIC** transport inside Seastar, built on the `ngtcp2` state machine and `GnuTLS`. A QUIC stream becomes an ordinary async I/O object; the **RPC** layer is adapted both as a drop-in and as a **multiplexing**, per-call design.

WHY IT MATTERS

An engineering bridge between a **callback-driven, sans-I/O** protocol engine and a **coroutine, shared-nothing** runtime. The *entire* RPC behavioural contract is preserved while the transport is swapped underneath — with a cost that is measured and **predictable**.

HOW IT FITS TOGETHER



RESULT. Correct, API-compatible, and predictable in cost — the QUIC transport stays within a constant of a mature TLS/TCP stack even in QUIC's *worst case* (a lossless single-host loopback), and was submitted upstream as a Seastar pull request.

BUILT WITH



SEASTAR.



QUIC

ngtcp2

GnuTLS

