

Better QUIC Implementations with Nesquic

Conference Poster**Author(s):**

[Brandner, Laurin](#) ; Marti, Kevin; Niekel, Bas; [Mishra, Ayush](#) ; Antichi, Gianni; Vanbever, Laurent

Publication date:

2025-09-10

Permanent link:

<https://doi.org/https://doi.org/10.3929/ethz-c-000784375>

Rights / license:

[Creative Commons Attribution 4.0 International](#)

Better QUIC Implementations with *Nesquic*

Laurin Brandner, Kevin Marti, Bas Niekel, Ayush Mishra, Gianni Antichi, Laurent Vanbever

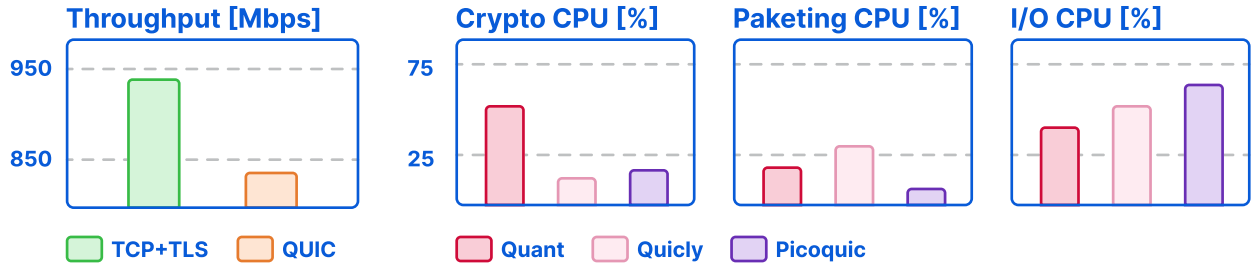


ETH zürich

QUIC could be quicker and lighter

In high-throughput scenarios, QUIC performs worse than TCP+TLS.

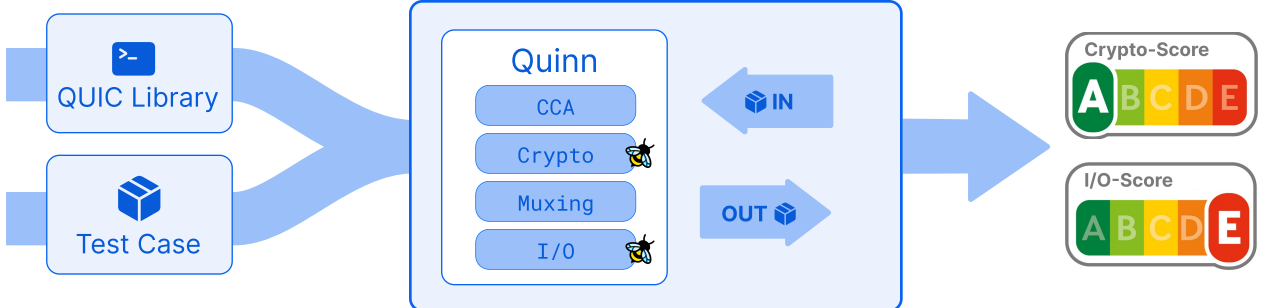
QUIC libraries implement **diverse design choices**, not all of which are production ready.



Examine the design choices of QUIC libraries with *Nesquic*

Nesquic is a **testing infrastructure** that produces **actionable insights**.

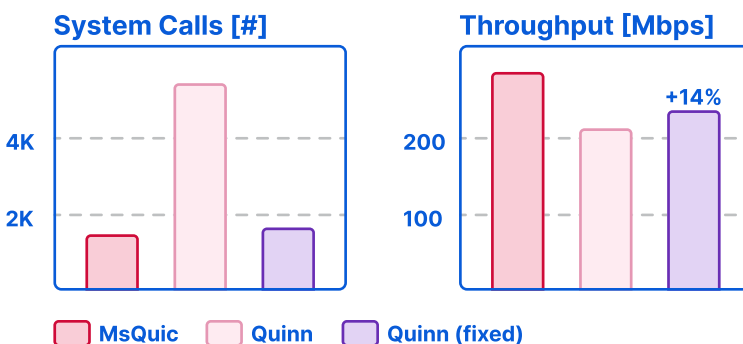
It leverages **eBPF** to produce **library-internal metrics** without code changes.



Quinn's I/O layer is inefficient

Quinn only writes **16KB/syscall**, but could be writing **64KB**.

Fixing this increases throughput and reduces CPU usage.



What's Next?

- reduce **monitoring overhead** of probes
- **meta analysis** to find QUIC stack to rule them all
- add more libraries, metrics and workloads