Chair of Network Architectures and Services
Department of Informatics
Technical University of Munich

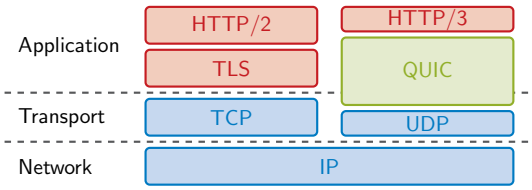
Thesis M.Sc.

IDP, Guided Research

Bringing QUIC to High-speed Networks

Motivation

QUIC is a protocol developed by Google [1] and is the designated successor of the TCP / TLS stack. It is implemented on top of UDP and improves several issues with TCP and was recently standardized by the IETF as RFC 9000 [2]. QUIC allows low latency handshakes, gets rid of head-of-line blocking, enables fast development and allows switching IP addresses within one connection.



Since QUIC is implemented in user space, which provides several advantages, it is lacking several optimizations within the kernel which are utilized for example by TCP. For example, early QUIC implementations suffered from the problem of higher power consumption than normal TCP and TLS.

In this thesis you try to get the most out of QUIC regarding bandwidth utilization on 10 G links. For this you first survey appropriate efficient implementations among existing ones,setup a measurement infrastructure in our hardware testbed, identify performance bottleneck and try to improve. Performance bottlenecks can for example reside within the hardware, the programming language, the implementation, the UDP interface offered by the kernel.

[1] A. Langley, A. Riddoch, A. Wilk, A. Vicente, C. Krasic, D. Zhang, F. Yang, F. Kouranov, I. Swett, J. Iyengar, *et al.*, "The quic transport protocol: Design and internet-scale deployment," in *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*, pp. 183–196, 2017.

[2] J. Iyengar and M. Thomson, "QUIC: A UDP-Based Multiplexed and Secure Transport," RFC 9000, RFC Editor, May 2021.

Requirements

- Advanced knowledge about basic networking protocols (IP, TCP, UDP) and their implementation in Linux (recommended: understanding of the QUIC protocol)
- Understanding how packet processing in the Linux kernel works
- Experience with working on Linux and via `ssh`

Your Task

- Setup a measurement setup within our hardware testbed including different HTTP server and QUIC implementations
- Design baseline measurements: e.g. different number of requests per second
- Monitor performance metrics: e.g. CPU utilization, response time, ...
- Analyze and evaluate results to further improve the link utilization

Contact

Benedikt Jaeger jaeger@net.in.tum.de
Johannes Zirngibl zirngibl@net.in.tum.de