Thesis M.Sc.

IDP, Guided Research

In Depth Analysis of QUIC's Lack of Kernel Optimizations

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 is currently standardized by the IETF [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 will investigate the following question: what are the drawbacks of the decision of QUIC being a user space protocol? For this purpose you will compare the performance of a HTTPs server based on TCP and QUIC to get further insights into how the kernel is involved (e.g. [3]).

[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] "IETF QUIC Working Group (quic)." https://datatracker.ietf.org/wg/quic/documents/. Accessed: 2020-05-20.

[3] P. Wang, C. Bianco, J. Riihijärvi, and M. Petrova, "Implementation and performance evaluation of the quic protocol in linux kernel," in *Proceedings of the 21st ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, pp. 227–234, 2018.

Requirements

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

Your Task

- Identify optimizations which were added to the Linux kernel to improve TCP / TLS
- 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

Contact

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