Zurich University
of Applied Sciences

School of
Engineering

InES Institute of
Embedded Systems

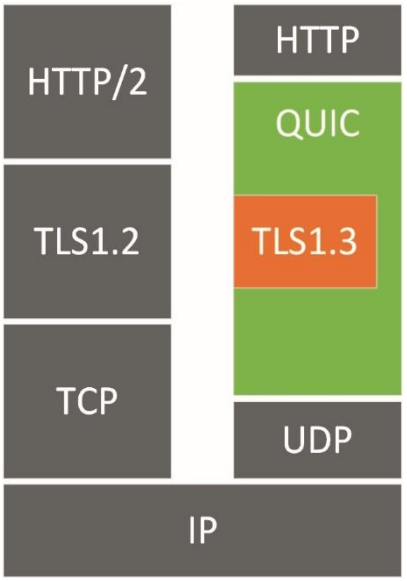
Google QUIC protocol engine for FPGA

Diplomierende
Sven Flachmüller
Richard Weiss

Dozent
Matthias Rosenthal

Enclustra's FPGA Manager allows simple and efficient data transfer via Ethernet between an FPGA and a host. The Ethernet solution should be improved by using a reliable protocol. The Google QUIC protocol therefore seems to be suitable for this application. The protocol was developed as a replacement for TCP/IP communication and supports multi-stream operation with low latency over a single UDP port.The focus of this work was to develop a FPGA-based QUIC engine that meets the required characteristics of reliability, performance and stream multiplexing. Based on the reference code from Google and the IETF QUIC Working Group and its documents, an independent concept was developed. The feasibility of the concept was tested and confirmed by means of an implementation in the programming language C on a Linux operating system. The test setup consists of two computers, a client and a server, whereby the server continuously streams data to the Client.After confirming the concept, the FPGA-based QUIC engine was implemented in blocks and tested with test benches at block level and at top level. A test setup with the developed C Client as host and the QUIC Engine as server was set up to check the reliability and determine the performance. The test data generated in the FPGA could be checked automatically, confirming its reliability. The performance was determined with Wireshark, a tool for analyzing data logs.The functionality of the implementation in C as well as the FPGA based QUIC engine was successfully confirmed. Data transmission is reliable, lost packets are sent again. The data rate of the FPGA-based QUIC engine is reduced by 2% for 1% packet loss and by 6.6% for 5% packet loss. The implementation in C reduces the data rate by 1.2 % for 1 % packet loss and by 9.8 % for 5 % packet loss. The FPGA based QUIC engine is much more efficient in case of a large packet loss due to the parallel processing of the QUIC logic and the data rate almost reaches the theoretically calculated expected value.

Stream multiplexing is not yet supported, but the framework for it has been created so that the possibility for expansion exists.



Classification of the Google QUIC protocol