

Advanced Computer Networking (ACN)

QUIC Project – Description

Prof. Dr.-Ing. Georg Carle

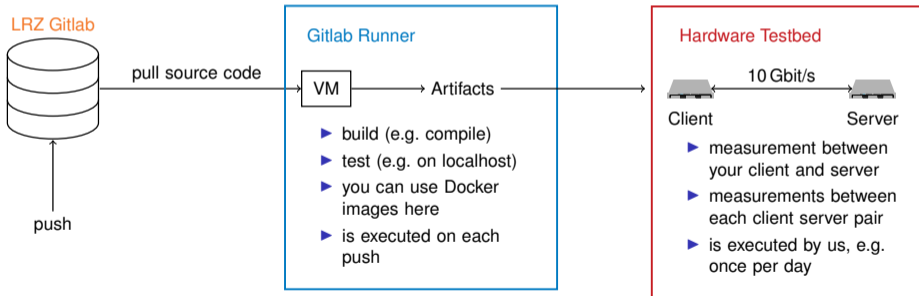
Benedikt Jaeger, Marcel Kempf

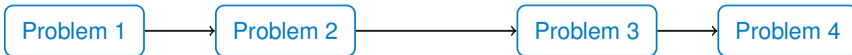
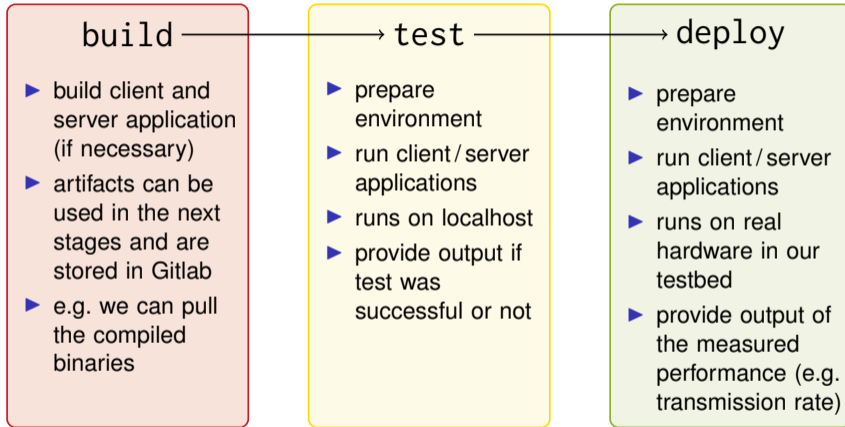
Chair of Network Architectures and Services
School of Computation, Information, and Technology
Technical University of Munich

- We use the LRZ Gitlab as infrastructure for this project
- You will get access to some shared repositories as well as a personal working repository

Gitlab CI

- This allows you to automatically compile your applications and access the artifacts via Gitlab





Project - Problem 3

Deadline: January 14, 2025, 4:00 PM

Tasks:

- Extend your HTTP/3 client and server to support different test cases

Testcase	Description
transfer	The client needs to send multiple requests and download all files using a single connection.
multihandshake	The client needs to send multiple requests and download all files using new connections for each request.
follow	The client requests a file from the server, which serves two files. The client only receives one request but has to download both files by parsing the second path out of the content of the first file.
chacha20	In this test, client and server are expected to offer only TLS_CHACHA20_POLY1305_SHA256 as a cipher suite.
retry	Tests that the server can generate a Retry, and that the client can act upon it (i.e. use the Token provided in the Retry packet in the Initial packet).
resumption	Tests QUIC session resumption (without 0-RTT). The client is expected to establish a connection and download the first file (first value in the <i>REQUESTS</i> variable). The server is expected to provide the client with a session ticket that allows it to resume the connection. After downloading the first file, the client has to close the first connection, establish a resumed connection using the session ticket, and use this connection to download the remaining file(s).
zerortt	Tests QUIC 0-RTT. The client is expected to establish a connection and download the first file. The server is expected to provide the client with a session ticket that allows the client to establish a 0-RTT connection on the next connection attempt. After downloading the first file, the client has to close the first connection, establish and request the remaining file(s) in 0-RTT.
transportparameter	Tests whether the server is able to set an <i>initial_max_streams_bidi</i> value of < 11 during the handshake.
