

Advanced Computer Networking (ACN)

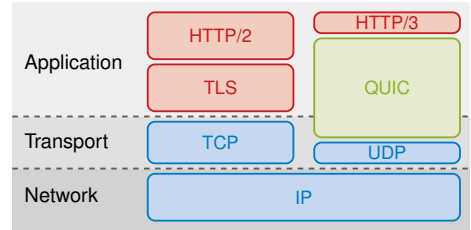
QUIC Project – Description

Prof. Dr.-Ing. Georg Carle

Benedikt Jaeger, Marcel Kempf, Johannes Zirngibl

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

- New transport protocol, originally developed by Google to replace the TCP / TLS stack
- Recently (May 2021) standardized by the IETF as RFC 9000
- Implemented on top of UDP in the user space
→ several implementations exist
- Includes features like flow and congestion control, stream multiplexing, encryption with TLS 1.3, and many more
- Detailed lecture about QUIC later this year



Design Goals of QUIC:

- Decrease handshake delay
- Get rid of head-of-line blocking
- Faster development cycles
- Middlebox resistance
- IP mobility

Goals of the Project:

- Familiarize with QUIC in general and a library
- Implement a working QUIC client and server
- Compare interoperability with other QUIC applications
- Optimize throughput

How do you participate?

- First, request a Gitlab repository if not done already for the exercise:
`https://acn.net.in.tum.de/auth`
- Merge requires resources from template repository:
`git remote add template git@gitlab.lrz.de:acn/terms/2023ws/template.git`
`git remote update`
`git merge --allow-unrelated-histories template/quic-project`
- You are only allowed to participate in one project (QUIC or Router)

How to make clear on which project you are working?

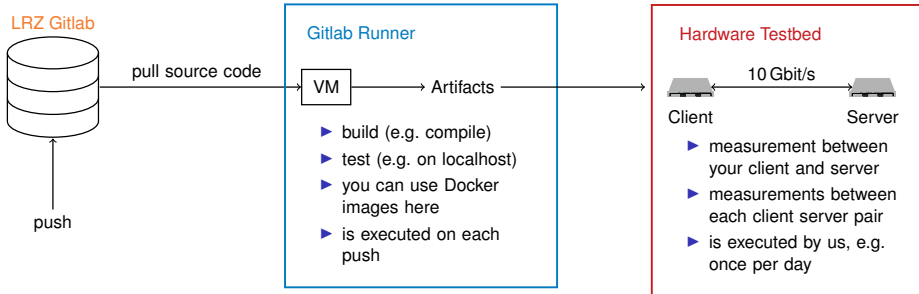
- In the root directory of your Git repository you find a file named `project.yml`
- We will only consider your submission for the QUIC project iff the file contains only the following line:

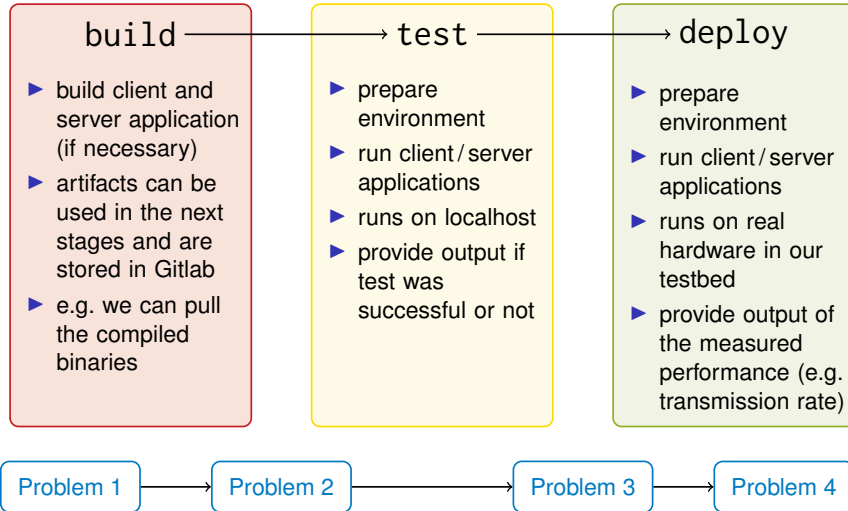
```
project: quic
```
- We use the content of this file to decide which project we correct for a certain deadline
- If you do not follow these instructions, we will not correct and grade your submission

- We use the LRZ Gitlab as infrastructure for this project (see `exercise/instructions.pdf`)
- 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





Problem 1: until November 28, 2023, 4:00 PM

- Familiarize with the QUIC protocol
- Choose one implementation
- Answer basic questions about QUIC and the implementation
- Deploy the Interop Runner on your VM and run a simple test

Problem 2: until December 19, 2023, 4:00 PM

- Setup client and server applications and environment
- Run basic functionality tests

Problem 3: until January 16, 2024, 4:00 PM

- Implement all functionality tests

Problem 4: until January 30, 2024, 4:00 PM

- Prepare testbed measurements
- Optimize parameters and compare results
- Summarize findings in a short report (2-3 pages)

Deadline: November 28, 2023, 4:00 PM

Tasks:

- Familiarize yourself with QUIC and the standard
- Select one of the offered implementations. You will work with the selected one for the rest of the project
 - lsquic (<https://github.com/litespeedtech/lsquic>)
 - quic-go (<https://github.com/lucas-clemente/quic-go>)
 - quiche (<https://github.com/cloudflare/quiche>)
- If you want to use another implementation:
 - ! Everything except *aioquic*¹ is allowed
 - ! You will not get support from our side if you use an implementation other than the suggested ones
- Answer some general questions about QUIC as well as implementation specific questions
- Deploy the Interop Runner on your VM and run a simple test
- **Optional:** compile and execute example client server applications and try to get them running

¹<https://github.com/aiortc/aioquic>