Advanced Computer Networking (ACN)

QUIC Project – Description

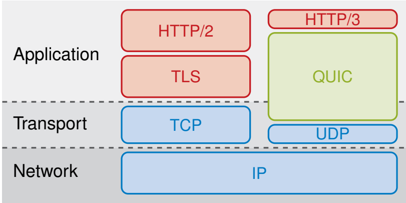
Prof. Dr.-Ing. Georg Carle

Benedikt Jaeger, Johannes Zirngibl

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

Introduction

- 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
- Detailed lecture about QUIC in the transport layer lecture



Design Goals of QUIC:

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

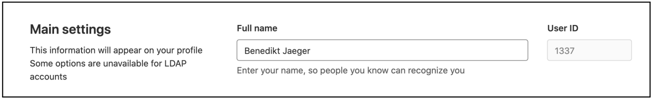
Goals of the Project:

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

Registration

How do you participate?

- Send a mail to acn.net.in.tum.de with subject starting with [QUIC] until November 16, 2021, 4:00 PM
- The mail has to contain your matriculation number as well as your LRZ Gitlab User ID
- To get the user ID, follow these steps:
  - Visit https://gitlab.lrz.de/-/profile
  - Main settings → User ID



Main settings

This information will appear on your profile
Some options are unavailable for LDAP accounts

Full name: Benedikt Jaeger

Enter your name, so people you know can recognize you

User ID: 1337

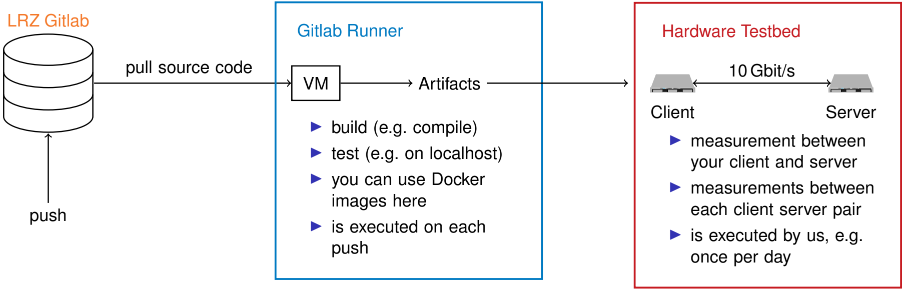
- After November 16, 2021, 4:00 PM we will randomly choose the participants in case more than 15 students want to participate

Infrastructure

- 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
- Don’t get confused, you then have two personal repositories for ACN

Gitlab CI

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



LRZ Gitlab

push

pull source code

Gitlab Runner

VM → Artifacts

- build (e.g. compile)
- test (e.g. on localhost)
- you can use Docker images here
- is executed on each push

Hardware Testbed

10 Gbit/s

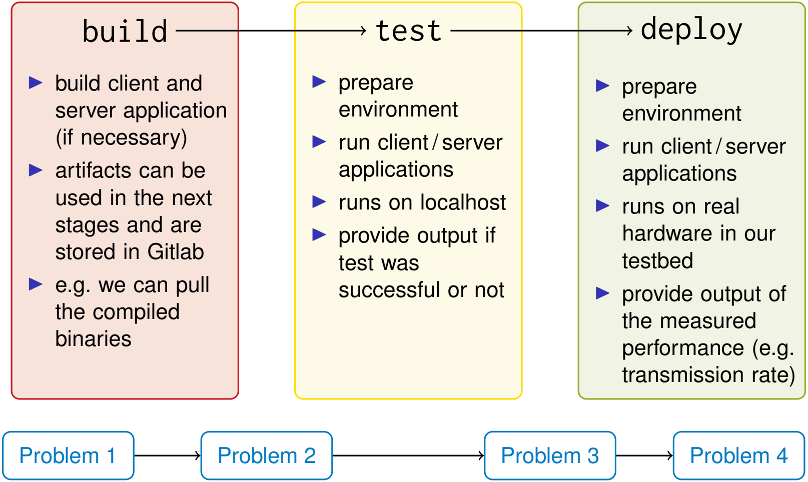
Client

Server

- measurement between your client and server
- measurements between each client server pair
- is executed by us, e.g. once per day

Infrastructure

Pipeline



build → test → deploy

build

- build client and server application (if necessary)
- artifacts can be used in the next stages and are stored in Gitlab
- e.g. we can pull the compiled binaries

test

- prepare environment
- run client / server applications
- runs on localhost
- provide output if test was successful or not

deploy

- prepare environment
- run client / server applications
- runs on real hardware in our testbed
- provide output of the measured performance (e.g. transmission rate)

Problem 1 → Problem 2 → Problem 3 → Problem 4

Schedule

Problem 1: until November 30, 2021, 4:00 PM

- Familiarize with the QUIC protocol
- Choose one implementation
- Answer basic questions about QUIC and the implementation

Problem 2: until December 21, 2021, 4:00 PM

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

Problem 3: until January 18, 2022, 4:00 PM

- Implement all functionality tests

Problem 4: until February 1, 2022, 4:00 PM

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

Project - Problem 1

Deadline: November 30, 2021, 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`)
  - aioquic (`https://github.com/aiortc/aioquic`)
- Answer some general questions about QUIC as well as implementation specific questions
- Optional: compile and execute example client server applications and try to get them running