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

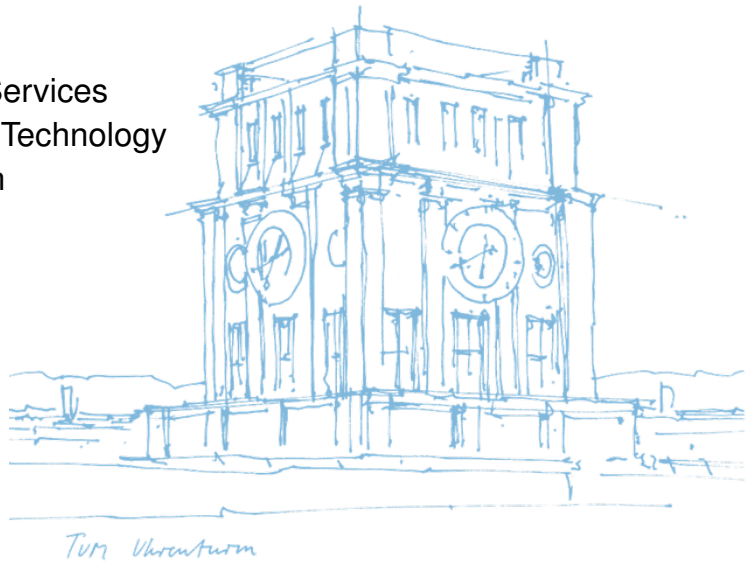
C++-based MASQUE-Proxying for Lower OSI-Layer Protocol Traffic

Intermediate talk for the IDP by

Christoph Rotte
advised by Lion Steger and Richard von Seck

Wednesday 19th April, 2023

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



MASQUE-Proxying

Current Status

- **Idea**: Adapt general approach of **HTTP CONNECT** for HTTP/3
- HTTP CONNECT: TCP/IP tunnel over HTTP
- HTTP **CONNECT-UDP** (MASQUE): UDP/IP tunnel over HTTP
- HTTP **CONNECT-IP** (MASQUE): IP tunnel over HTTP

MASQUE-Proxying

Current Status

- **Idea**: Adapt general approach of **HTTP CONNECT** for HTTP/3
- HTTP CONNECT: TCP/IP tunnel over HTTP
- HTTP **CONNECT-UDP** (MASQUE): UDP/IP tunnel over HTTP
- HTTP **CONNECT-IP** (MASQUE): IP tunnel over HTTP
- MASQUE Working Group
- Current Status:
  - CONNECT-UDP: `rfc9298`
  - CONNECT-IP: `draft-ietf-masque-connect-ip-10`
- Early stage regarding implementation

Background

TCP Proxying

- Many proxy protocols (SOCKS, PEPs, HTTP CONNECT) are based on TCP
- HTTP CONNECT: **several logical streams** over **one TCP connection**

→ performance problems like **head-of-line blocking**

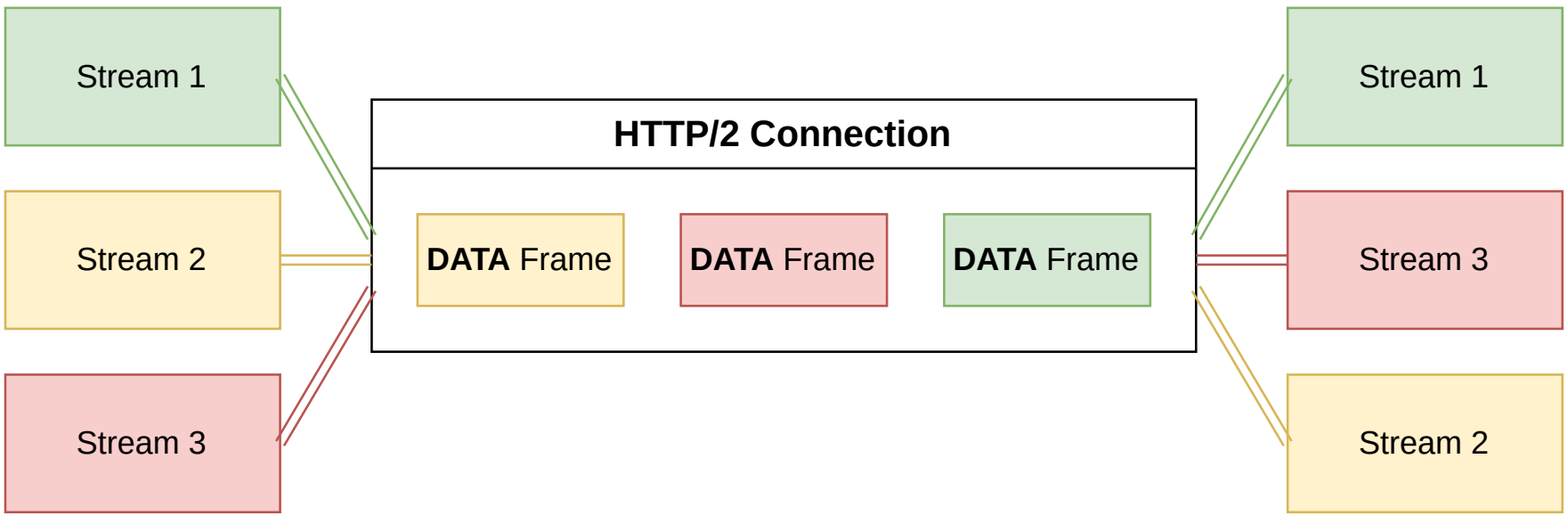


Figure 1: HTTP/2 Proxying via CONNECT [5]

Background

QUIC and QUIC Datagrams

- QUIC: UDP-based TCP alternative (used for HTTP/3)
- Logical streams consist of **reliable** STREAM frames [1]
- **RFC9221**: Unreliable QUIC Datagram Extension [2]

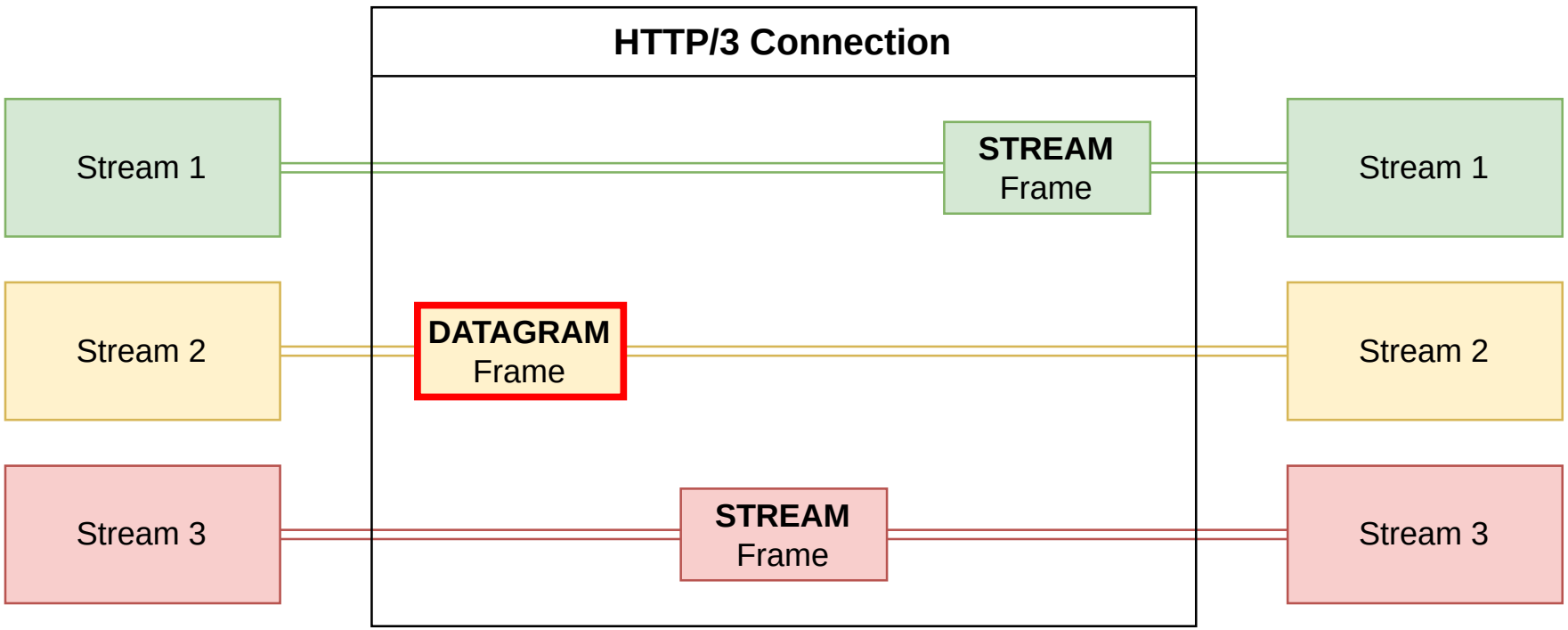


Figure 2: QUIC Streams Using Unreliable QUIC Datagrams

MASQUE-Proxying

CONNECT-UDP Method

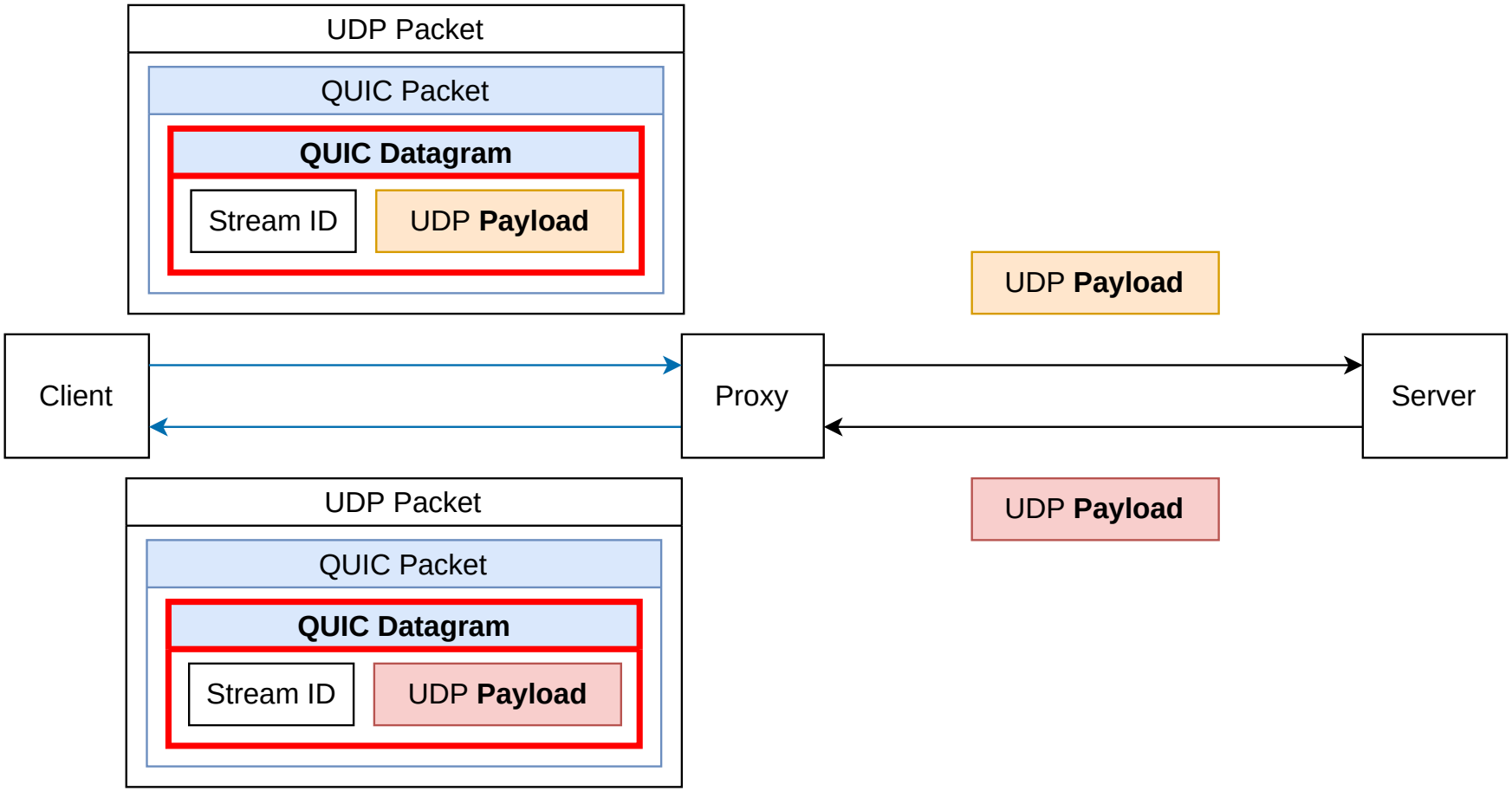


Figure 3: Proxying via QUIC and CONNECT-UDP [4]

MASQUE-Proxying

CONNECT-IP Method (DRAFT)

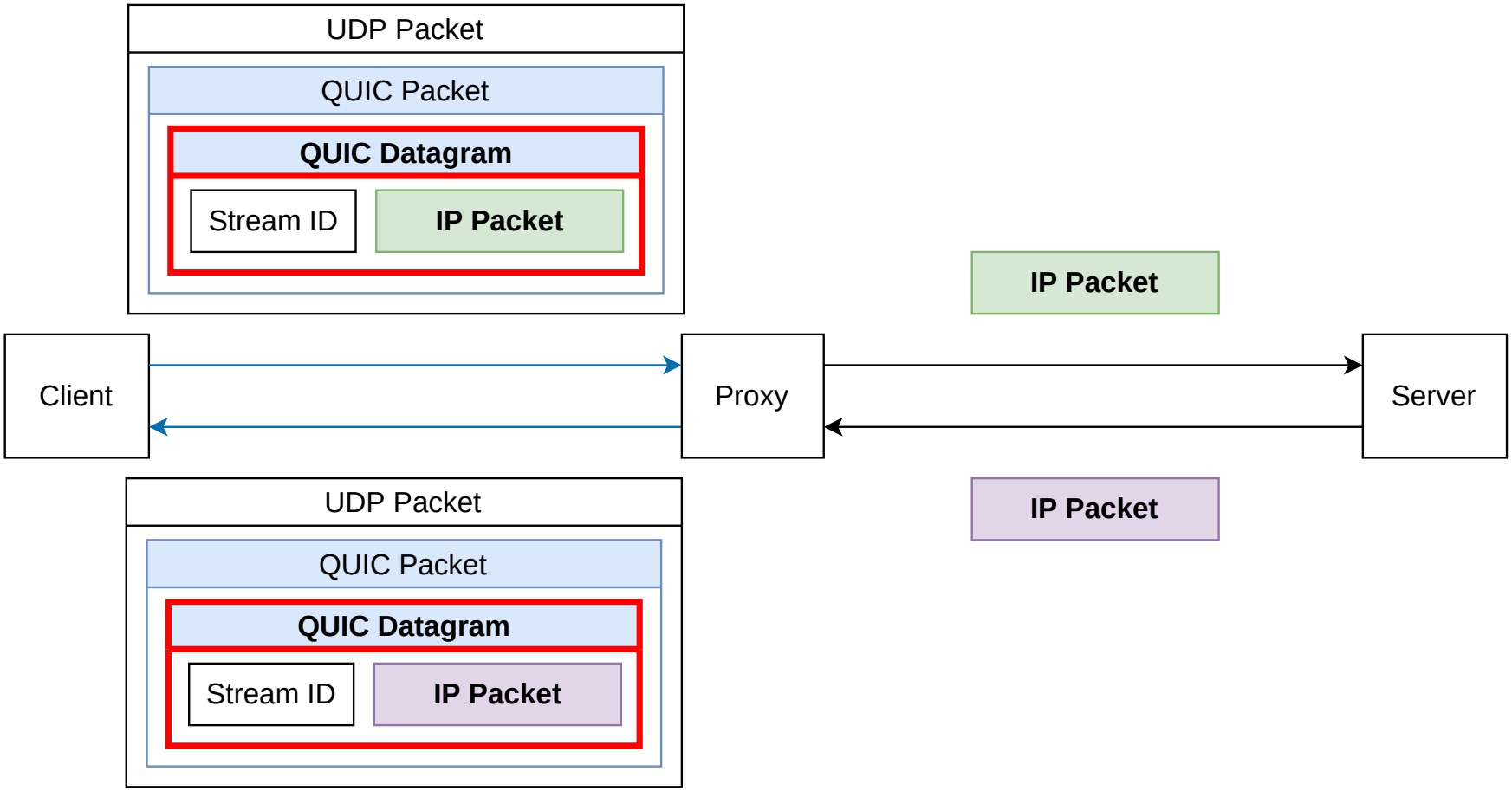


Figure 4: Proxying via QUIC and CONNECT-IP [3]

MASQUE-Proxying

Motivation: Research Questions

RQ 1 What are the implications of using CONNECT-IP in the context of overhead caused by encapsulation?

RQ 2 How does CONNECT-IP behave in the context of transmission performance compared to CONNECT-UDP?

RQ 3 What areas may cause library-specific differences and which challenges may occur because of this?

Encapsulation Overhead (CONNECT-IP)

- **Reduced available packet size** due to headers and MTU
  - Typical MTU: **1500B** $\leftrightarrow$ Minimum required MTU for Masque: **1200B**
  - Overhead for encapsulated packets: **49B - 94B**
    - Context ID: 1B - 8B
    - HTTP/3 Datagram Frame header: 1B - 8B
    - QUIC Datagram Frame header: 1B - 9B
    - QUIC (short) header: 2B - 25B
    - QUIC MAC: 16B
    - UDP header: 8B
    - IPv4 header: 20B

  $\rightarrow$ Conservative bound for multi-hop setups: **3** CONNECT-IP connections

MASQUE-Proxying

Research Questions: Preliminary Analysis

Encapsulation Overhead (CONNECT-IP)

- **Reduced available packet size** due to headers and MTU
  - Typical MTU: **1500B** $\leftrightarrow$ Minimum required MTU for Masque: **1200B**
  - Overhead for encapsulated packets: **49B - 94B**
    - Context ID: 1B - 8B
    - HTTP/3 Datagram Frame header: 1B - 8B
    - QUIC Datagram Frame header: 1B - 9B
    - QUIC (short) header: 2B - 25B
    - QUIC MAC: 16B
    - UDP header: 8B
    - IPv4 header: 20B

  $\rightarrow$ Conservative bound for multi-hop setups: **3** CONNECT-IP connections
- Preliminary measurement (proxy on the same host):

  ```
  $ curl https://speed.hetzner.de/10GB.bin --interface tun0
  ```

  CONNECT-IP: $\sim$ **18.1 MiB/s**

  w/o CONNECT-IP: $\sim$ **19 MiB/s**

CONNECT-IP vs. CONNECT-UDP

- Only difference in terms of packet structure:
  Absence of IP header + UDP header (20B + 8B)

→ Expected similar performance characteristics in relative terms of latency and **absence of head-of-line blocking**

→ Logical streams theoretically **scale on one** QUIC connection

Library-Specific Differences

- **Parameterization**, i.e. exchanging proxy configuration information (like available IP routes)
- Implementations **following different draft versions**
  $\rightarrow$ transport parameter sizes, different handling of methods like CONNECT, etc.
- Example (QUIC):
  - facebookincubator/mvfst[1] seems to be based on **draft-ietf-quic-transport version $\leq$ 23**
  - google/quiche[2] seems to be based on the finalized **RFC 9000**

  $\rightarrow$ Sending a datagram size of 2^{16} leads to an overflow in mvfst

[1] https://github.com/facebookincubator/mvfst
[2] https://github.com/google/quiche

Implementation

Library

- Motivation:
  - Evaluation / measurements ($\rightarrow$ RQs)
  - Current MASQUE implementations do not contain all features
- facebook/proxygen[3] (HTTP/3 Library)
- facebookincubator/mvfst[4] (QUIC Library)

[3] https://github.com/facebook/proxygen
[4] https://github.com/facebookincubator/mvfst

Implementation

Library

- Motivation:
  - Evaluation / measurements ($\rightarrow$ RQs)
  - Current MASQUE implementations do not contain all features
- facebook/proxygen[3] (HTTP/3 Library)
- facebookincubator/mvfst[4] (QUIC Library)

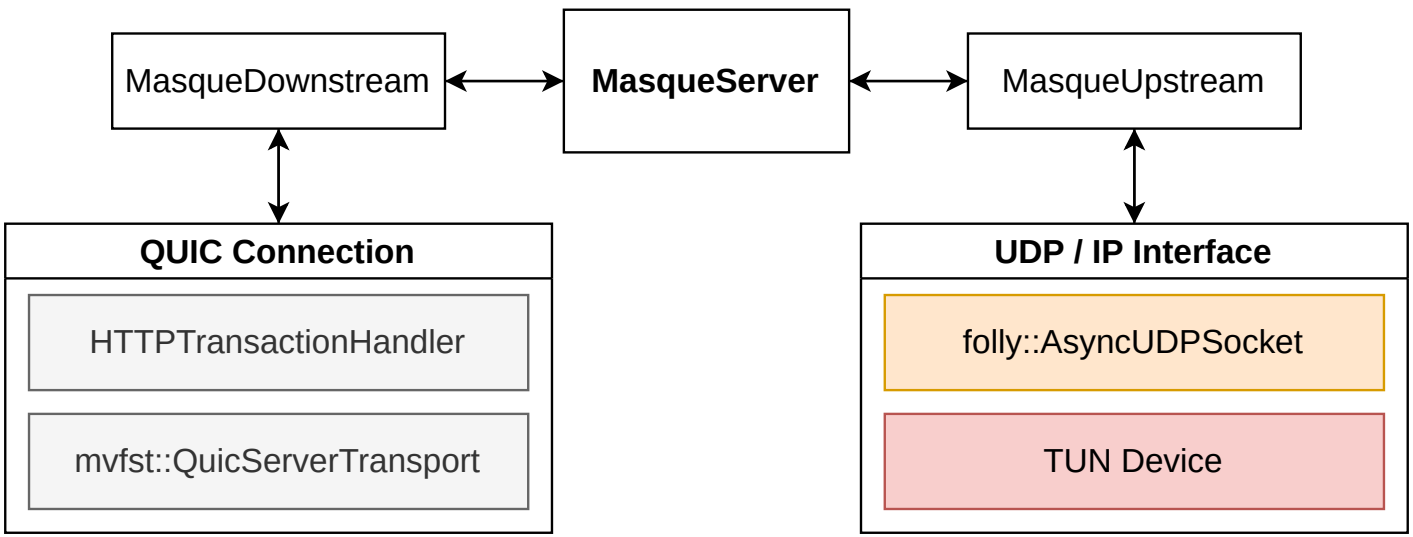


Figure 5: Architectural Overview

[3] https://github.com/facebook/proxygen
[4] https://github.com/facebookincubator/mvfst

Next Steps

Progress + Planned

- CONNECT-UDP implementation (client + server) ✓
- CONNECT-IP implementation (client + server) ✓
- CONNECT-UDP implementation cross-tested ✓

- CONNECT-IP Implementation cross-tested
- Evaluation / Analysis of the theoretic assumptions
  - Testbed
  - Multihop setup
  - ...
- Library Comparison (Cross-Evaluation)
  - Google QUICHE[5]
  - ...

[5] https://github.com/google/quiche

Bibliography

[1] J. Iyengar and M. Thomson.
QUIC: A UDP-Based Multiplexed and Secure Transport, 2021.
http://tools.ietf.org/html/rfc9000.

[2] T. Pauly, E. Kinnear, and D. Schinazi.
An Unreliable Datagram Extension to QUIC, 2022.
http://tools.ietf.org/html/rfc9221.

[3] T. Pauly, D. Schinazi, A. Chernyakhovsky, M. Kühlewind, and M. Westerlund.
Proxying UDP in HTTP, 2023.
https://www.ietf.org/archive/id/draft-ietf-masque-connect-ip-10.html.

[4] D. Schinazi.
Proxying UDP in HTTP, 2022.
http://tools.ietf.org/html/rfc9298.

[5] M. Thomson and C. Benfield.
HTTP/2, 2022.
http://tools.ietf.org/html/rfc9113.

Appendix

CONNECT-IP Implementation

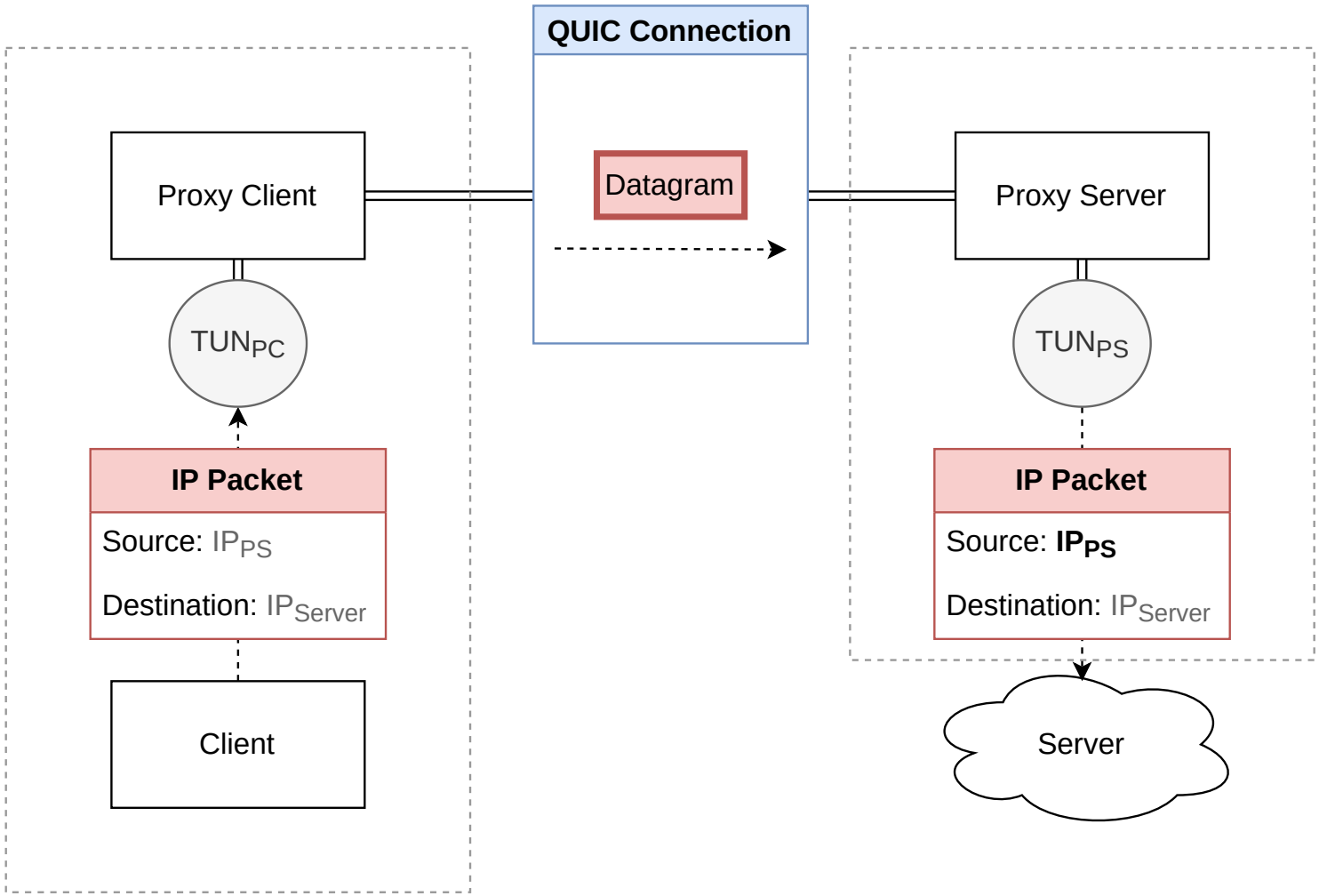


Figure 6: CONNECT-IP Implementation Overview

Appendix

Multithreading

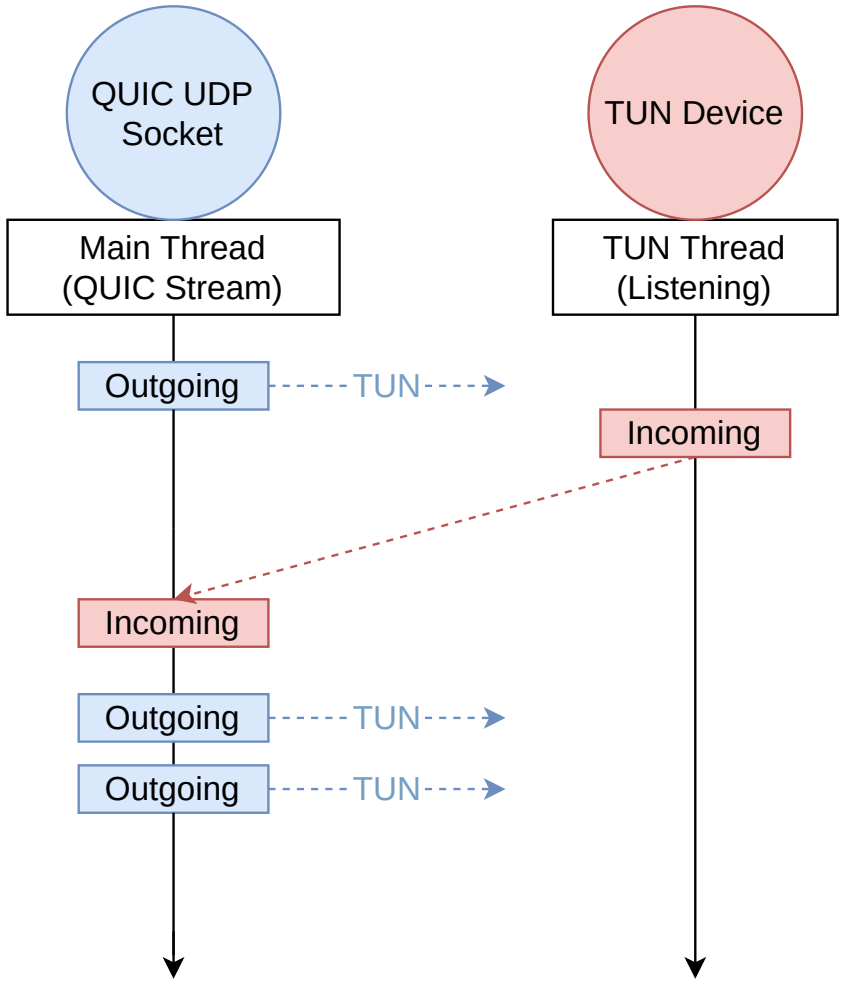


Figure 7: Proxygen Multithreading Overview for CONNECT-IP