Design and Implementation of a Multipath Extension for QUIC

TECHNISCHE UNIVERSITÄT DARMSTADT

Tobias Viernickel
Betreuer: Alexander Frömmgen

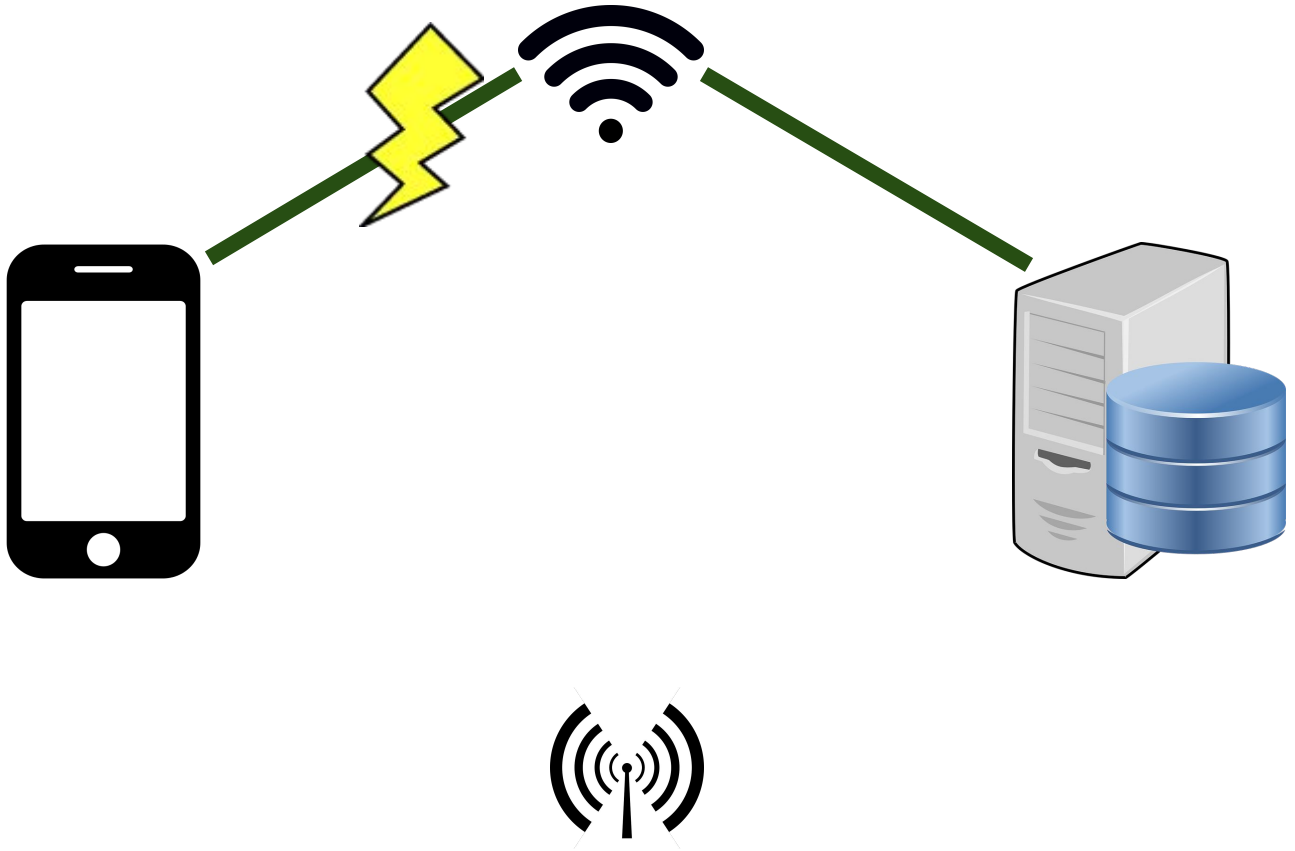
Prof. Dr.-Ing. Ralf Steinmetz
KOM - Multimedia Communications Lab

Motivation

TECHNISCHE UNIVERSITÄT DARMSTADT

Single-Path protocols

- TCP
- UDP
- QUIC

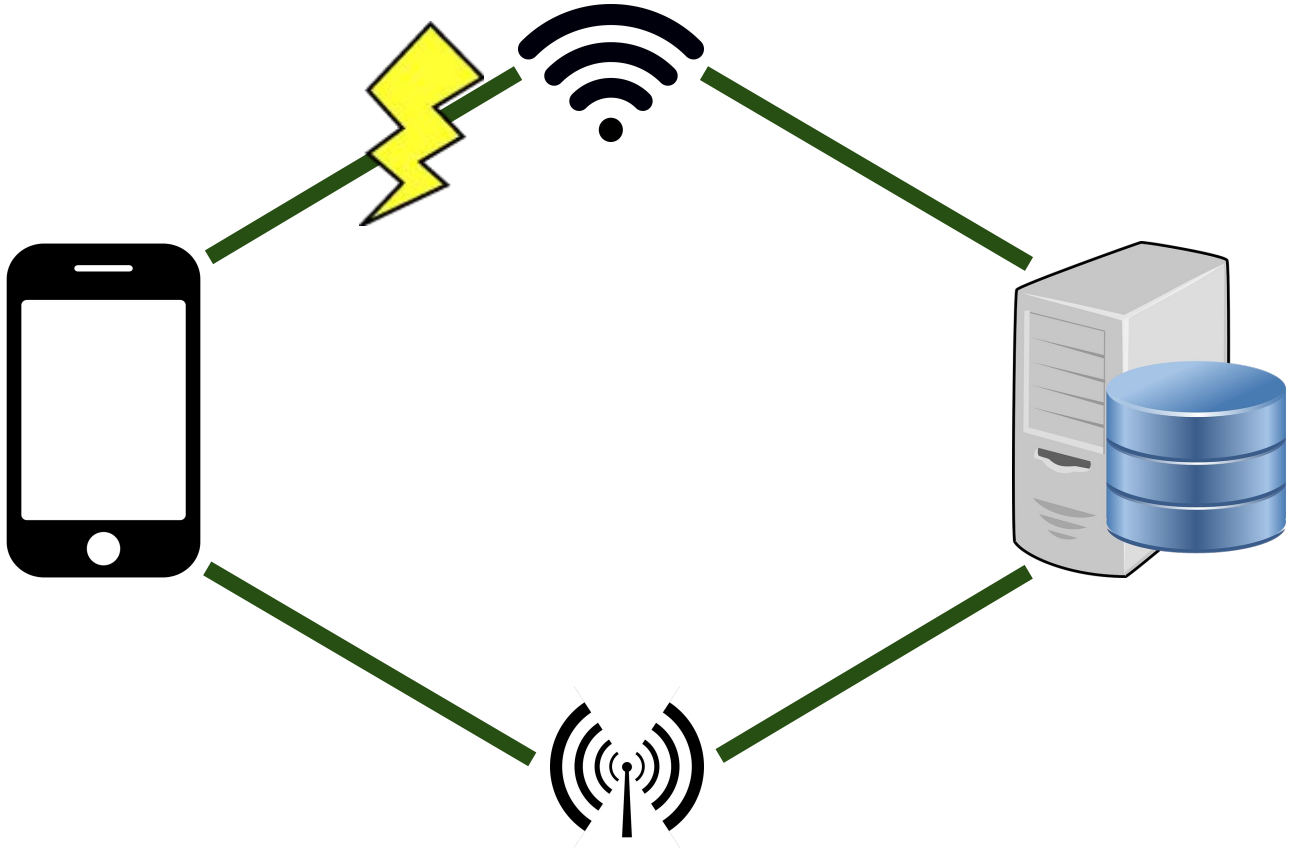


Motivation

TECHNISCHE UNIVERSITÄT DARMSTADT

Multipath protocols

- MPTCP
- MPUDP
- SCTP



Motivation

TECHNISCHE UNIVERSITÄT DARMSTADT

"… low-latency transport protocol …
enable rapid deployment ...
continued evolution of transport mechanisms …
deployed at Google on thousands of servers ..."[1]

No Multipath Support

The QUIC Transport Protocol: Design and Internet-Scale Deployment

Adam Langley, Alistair Riddoch, Alyssa Wilk, Antonio Vicente, Charles Krasic, Dan Zhang, Fan
Bailey, Jeremy Dorfman, Jim Roskind,
ade, Ryan Hamilton, Victor Vasiliev,
i Shi *

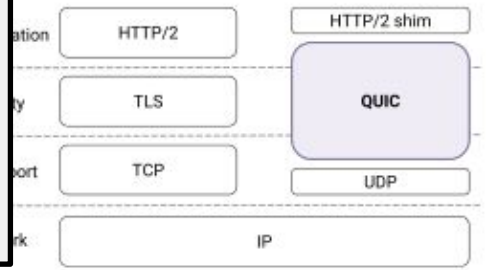


Figure 1: QUIC in the traditional HTTPS stack.

ples that guided our design, the Internet-scale process that we used to perform iterative experiments on QUIC, performance improvements seen by our various services, and our experience deploying QUIC globally. We also share lessons about transport design and the Internet ecosystem that we learned from our deployment.

TCP (Figure 1). We developed QUIC as a user-space transport with UDP as a substrate. Building QUIC in user-space facilitated its deployment as part of various applications and enabled iterative changes to occur at application update timescales. The use of UDP allows QUIC packets to traverse middleboxes. QUIC is an encrypted transport: packets are authenticated and encrypted, preventing modification and limiting ossification of the protocol by middleboxes. QUIC uses a cryptographic handshake that minimizes handshake latency for most connections by using known server credentials on repeat connections and by removing redundant handshake-overhead at multiple layers in the network stack. QUIC eliminates head-of-line blocking delays by using a lightweight data-structuring abstraction,

CCS CONCEPTS

• **Networks** → **Network protocol design**; **Transport protocols**; *Cross-layer protocols*;

ACM Reference format:
Adam Langley, Alistair Riddoch, Alyssa Wilk, Antonio Vicente, Charles Krasic, Dan Zhang, Fan Yang, Fedor Kouranov, Ian Swett, Janardhan Iyengar, Jeff Bailey, Jeremy Dorfman, Jim Roskind, Joanna Kulik, Patrik Westin, Raman Tenneti, Robbie Shade, Ryan Hamilton, Victor Vasiliev, Wan-Teh

[1] Langley et al. *The QUIC Transport Protocol: Design and Internet-Scale Deployment. In Proceedings of the Conference of the ACM Special Interest Group on Data Communication. ACM, 183–196.*

Research Question

TECHNISCHE UNIVERSITÄT DARMSTADT

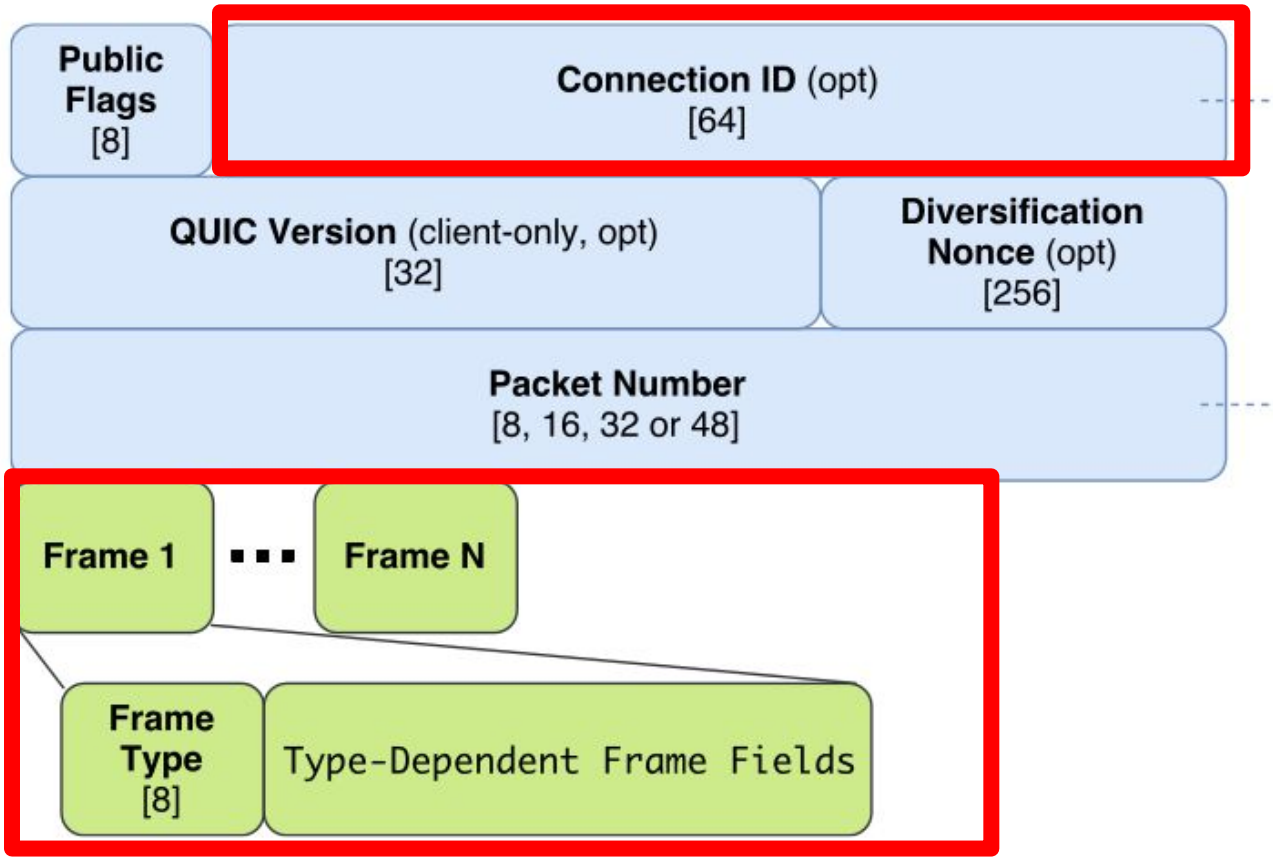
Can we make QUIC multipath capable?

Can MPQUIC improve the performance?

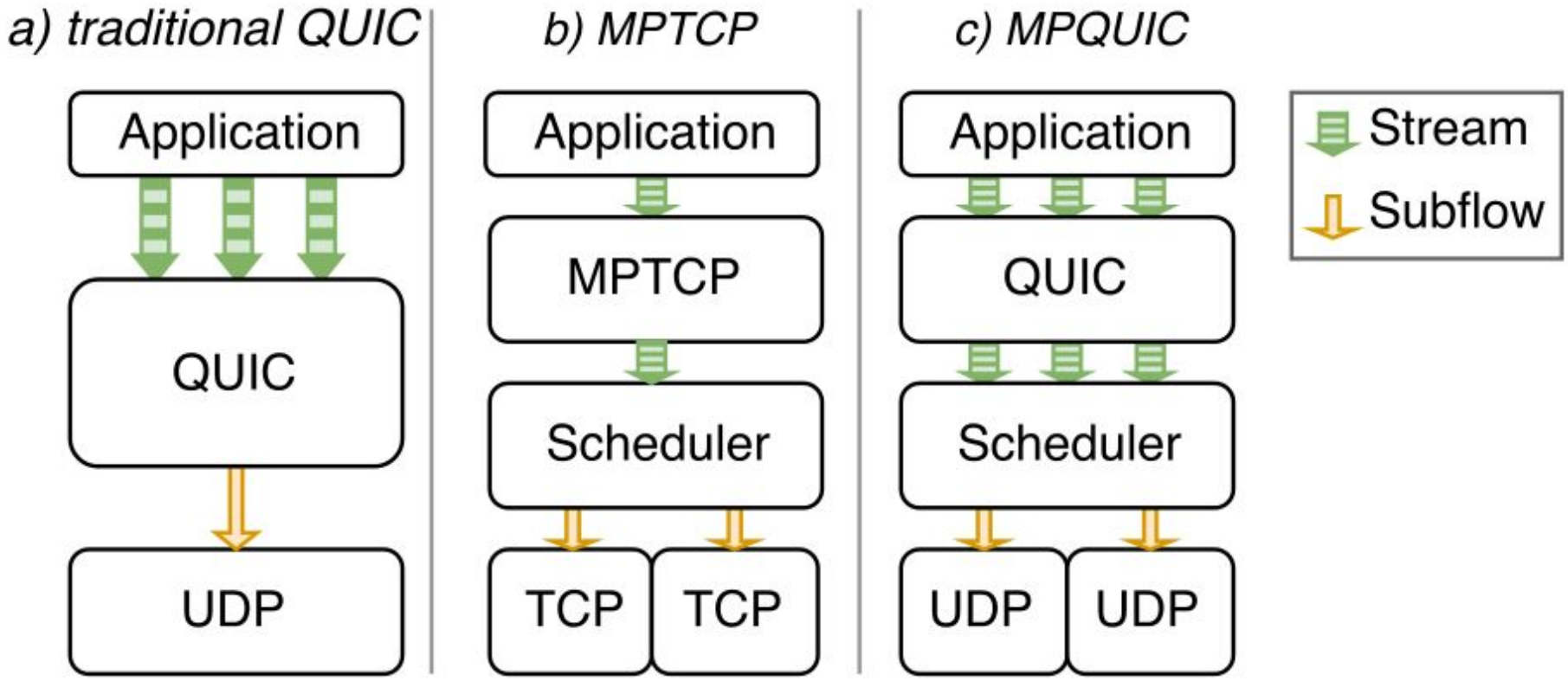
Has MPQUIC any fundamental advantages?

Background: QUIC

TECHNISCHE UNIVERSITÄT DARMSTADT

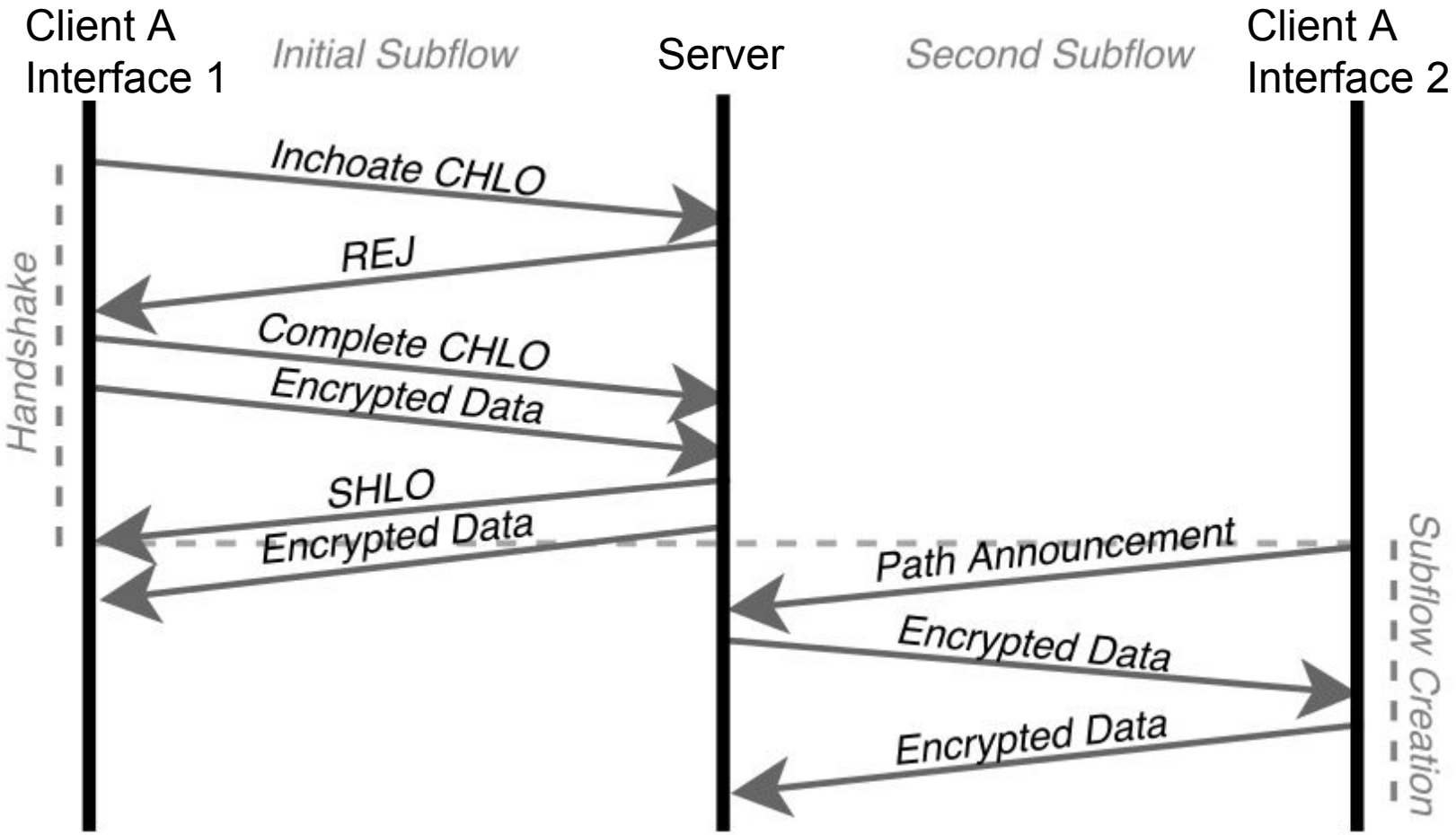


The Design of MPQUIC
Network Stack Comparison



The Design of MPQUIC
Subflow Establishment: Announcement

TECHNISCHE UNIVERSITÄT DARMSTADT

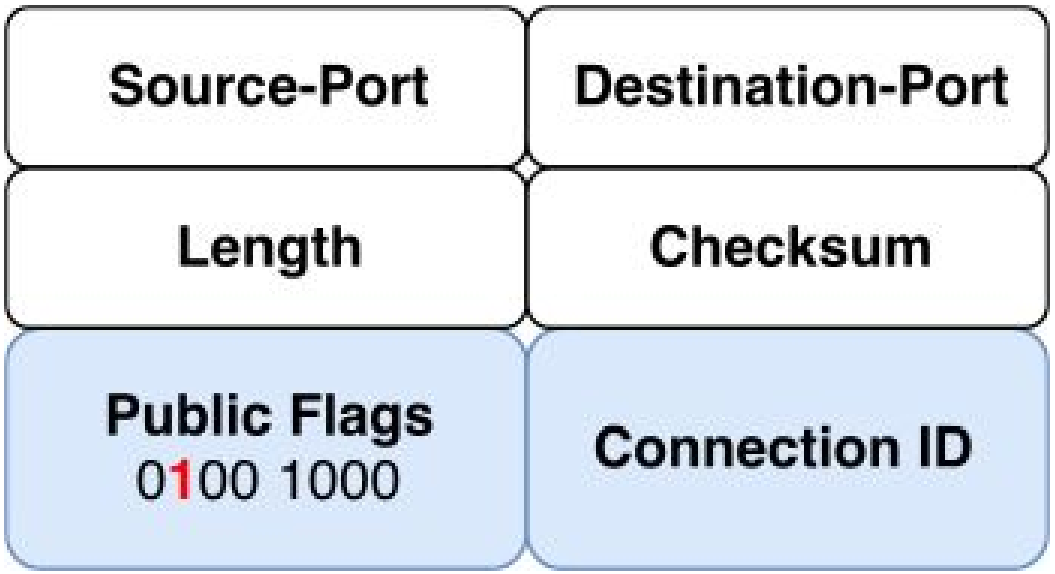


The Design of MPQUIC
Subflow Establishment: Announcement

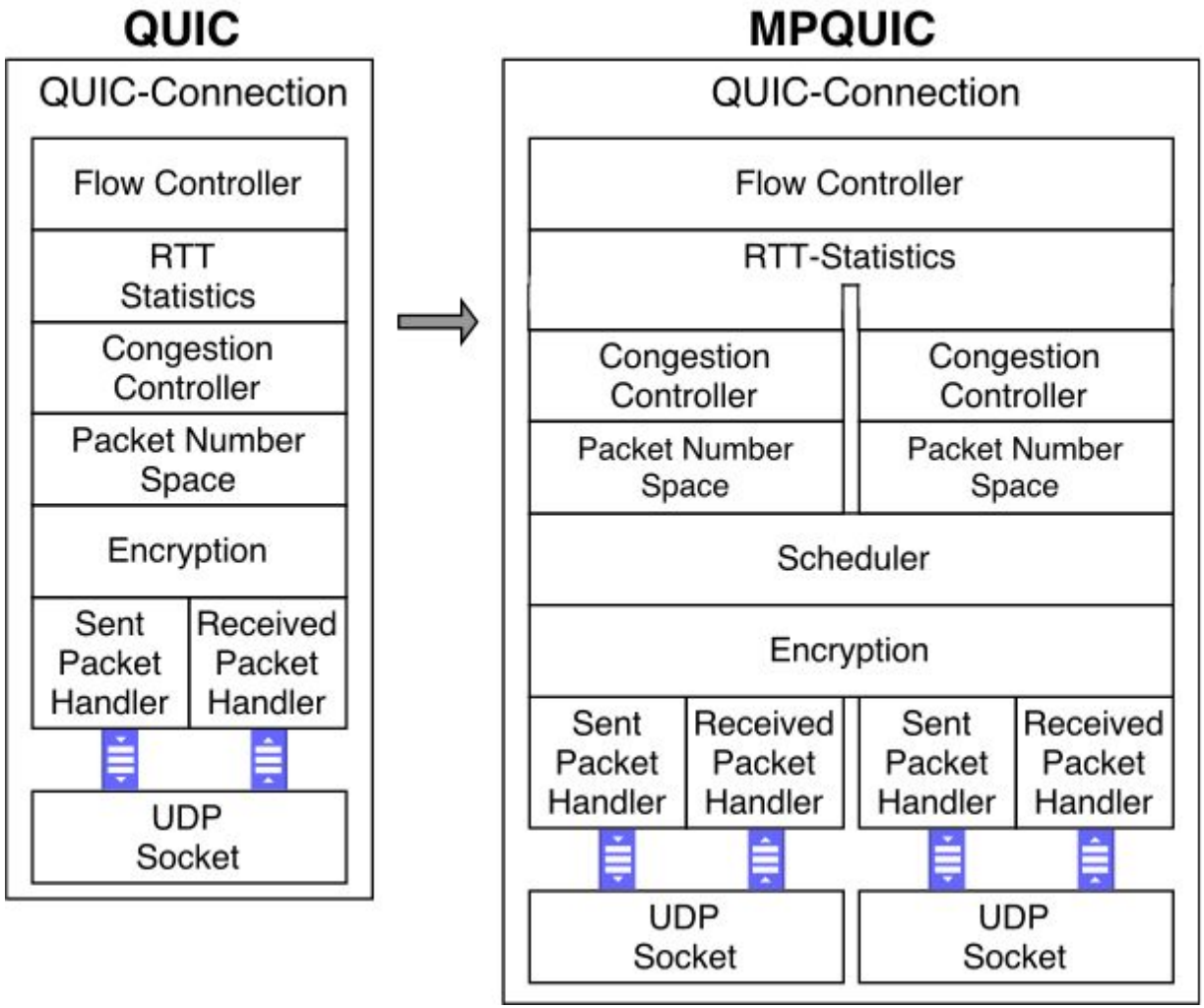
TECHNISCHE UNIVERSITÄT DARMSTADT

The Announcement Packet

- IP/Port inside the UDP/IP Header
- ConnectionID
- MultipathFlag



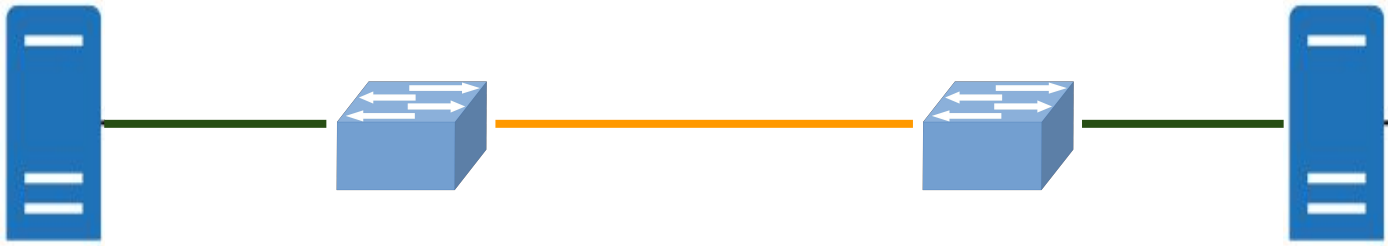
The Design of MPQUIC
Specifying the individual Components



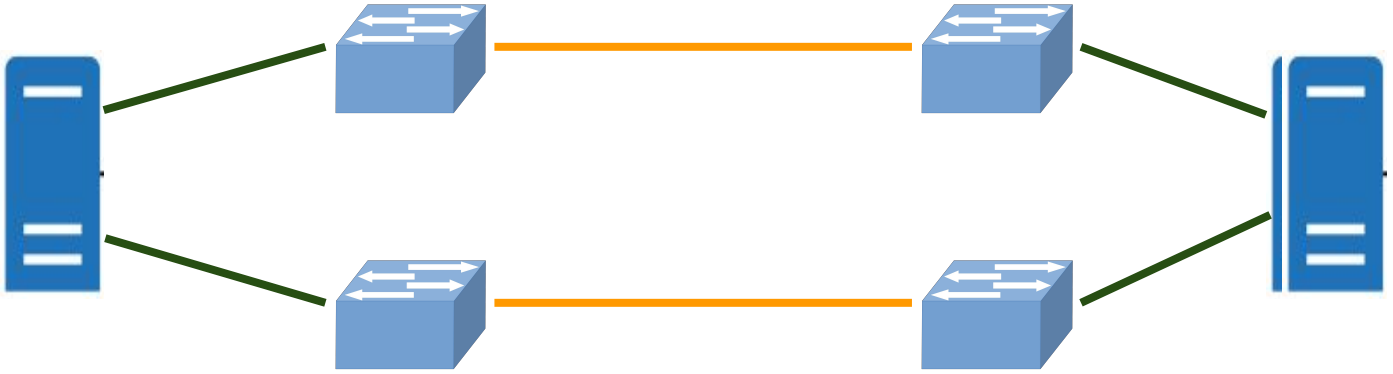
Evaluation
General Setup using Mininet

TECHNISCHE UNIVERSITÄT DARMSTADT

Single-Path Setup



Multipath Setup



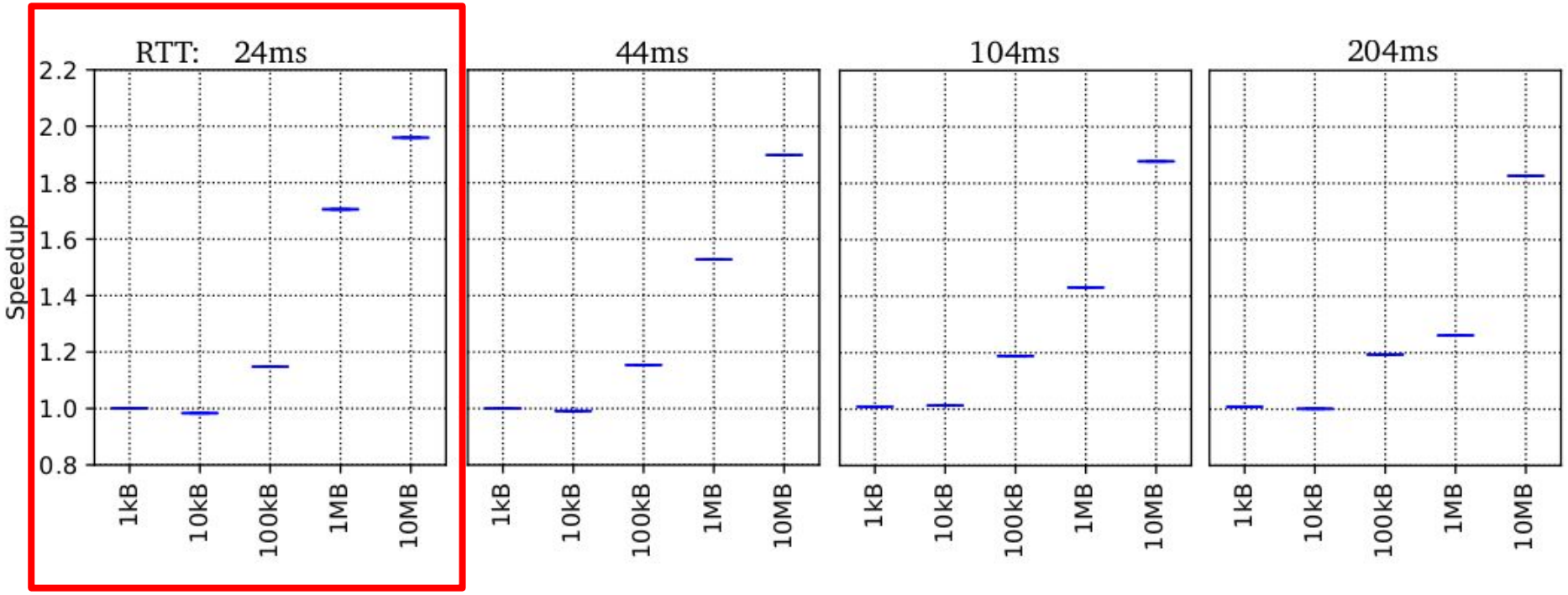
Evaluation
Speedup of multipath QUIC

TECHNISCHE UNIVERSITÄT DARMSTADT

HTTP/2 file requests of differenz sizes

Initial Path:
Bandwidth: 10Mbps

Second Path:
Bandwidth: 10Mbps



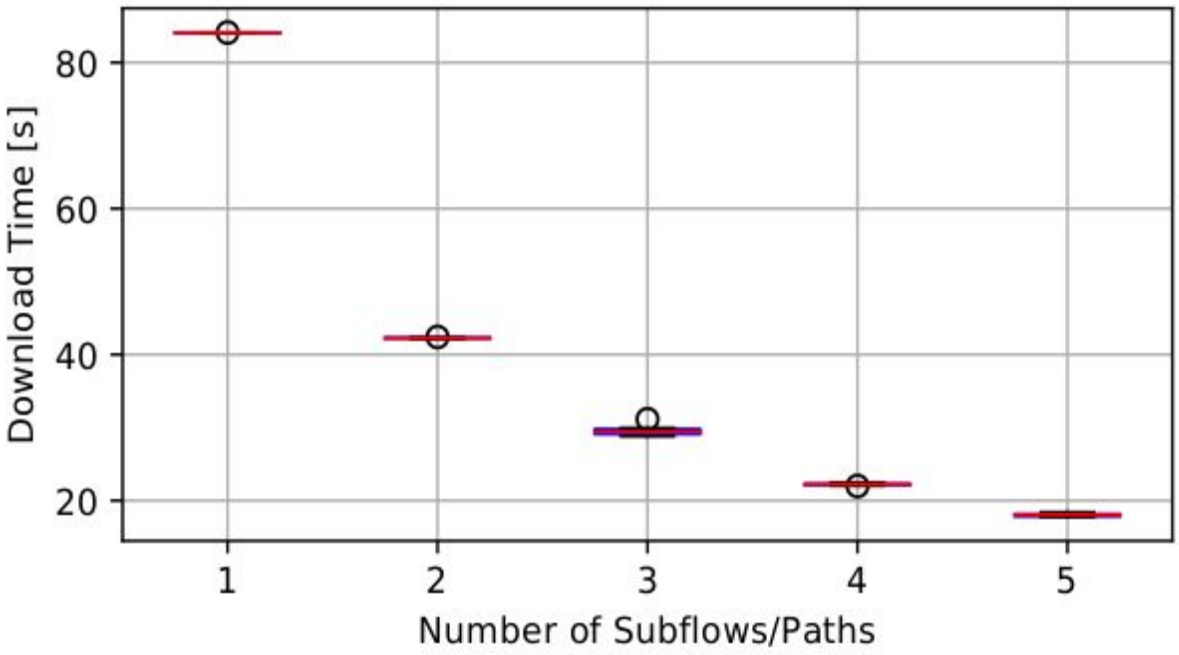
Evaluation
Speedup of multipath QUIC

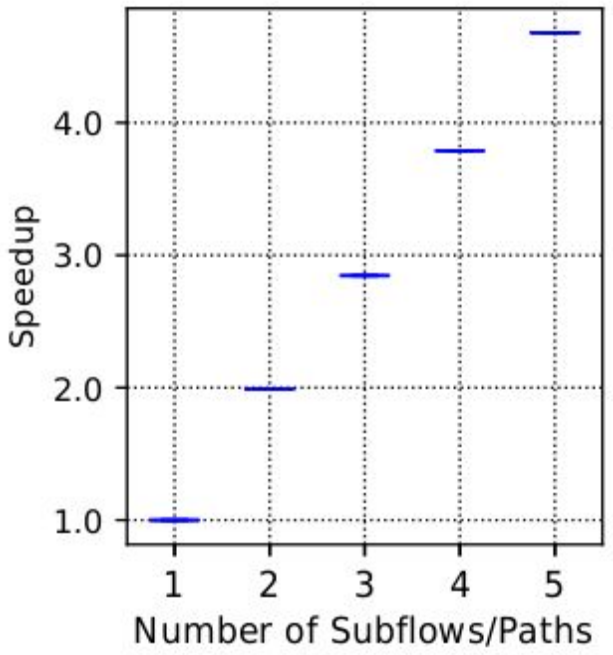
TECHNISCHE UNIVERSITÄT DARMSTADT

HTTP/2 file requests for a various number of paths

- File size: 100 MB

All Paths:
Bandwidth: 10Mbps
RTT: 44ms





Evaluation
Comparison of MPQUIC, QUIC, MPTCP, TCP

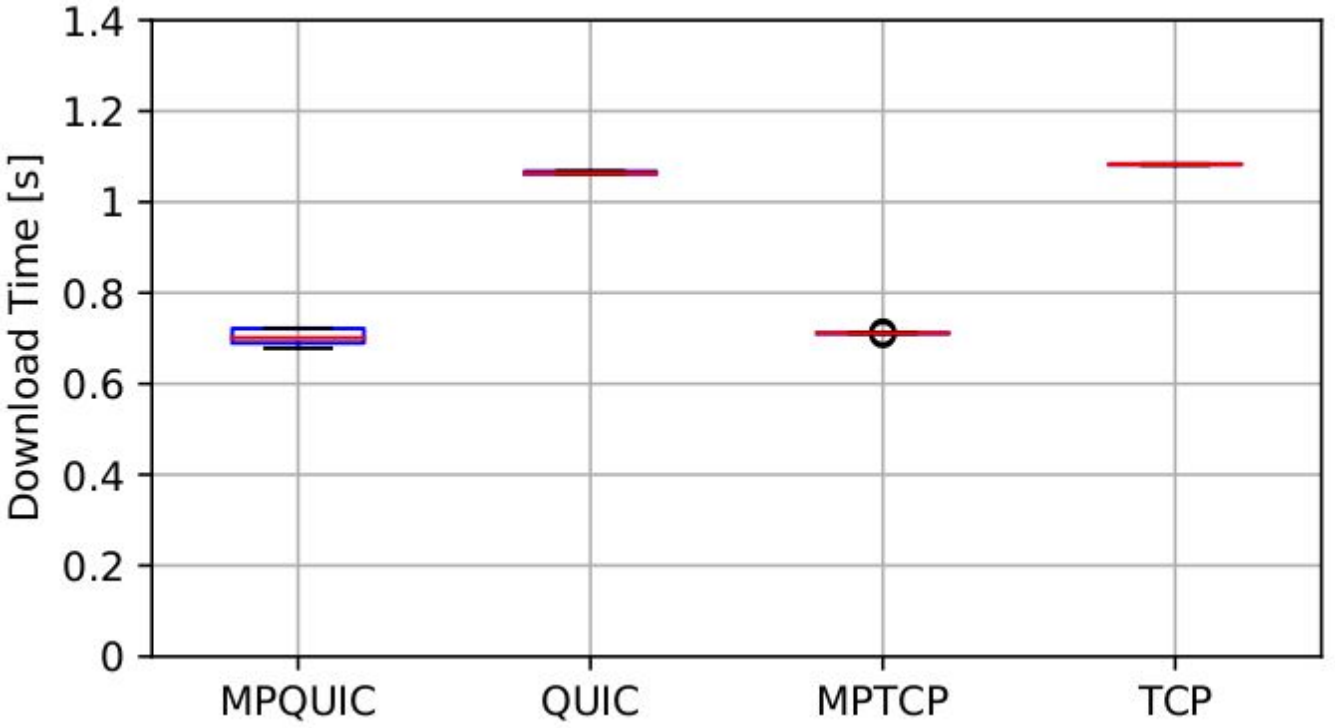
TECHNISCHE UNIVERSITÄT DARMSTADT

HTTP/2 file request

- File size: 1 MB

Initial Path
Bandwidth: 10Mbps
RTT: 44ms

Second Path
Bandwidth: 10Mbps
RTT: 44ms



Evaluation
Priority Scheduler

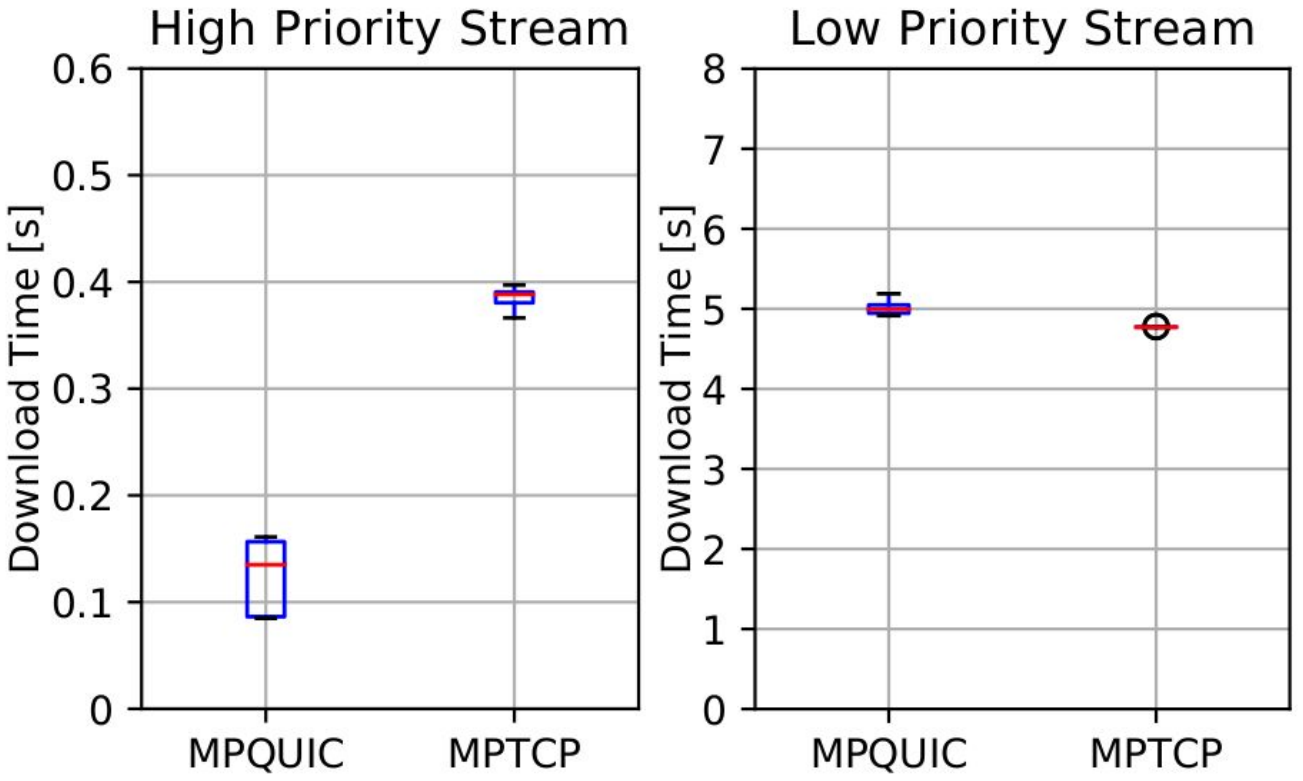
TECHNISCHE UNIVERSITÄT DARMSTADT

HTTP/2 file requests

- Low Priority File: 10 MB
- High Priority File: 1 KB (request after ~500ms)

Initial Path
Bandwidth: 10Mbps
RTT: 44ms

Second Path
Bandwidth: 10Mbps
RTT: 104ms



Conclusion and Outlook

TECHNISCHE UNIVERSITÄT DARMSTADT

Can we make QUIC multipath capable? ✓

Can MPQUIC improve the performance? ✓

Has MPQUIC any fundamental advantages? ✓

- *Deployability*
- *Middlebox interference*
- *Stream to Subflow Scheduling*

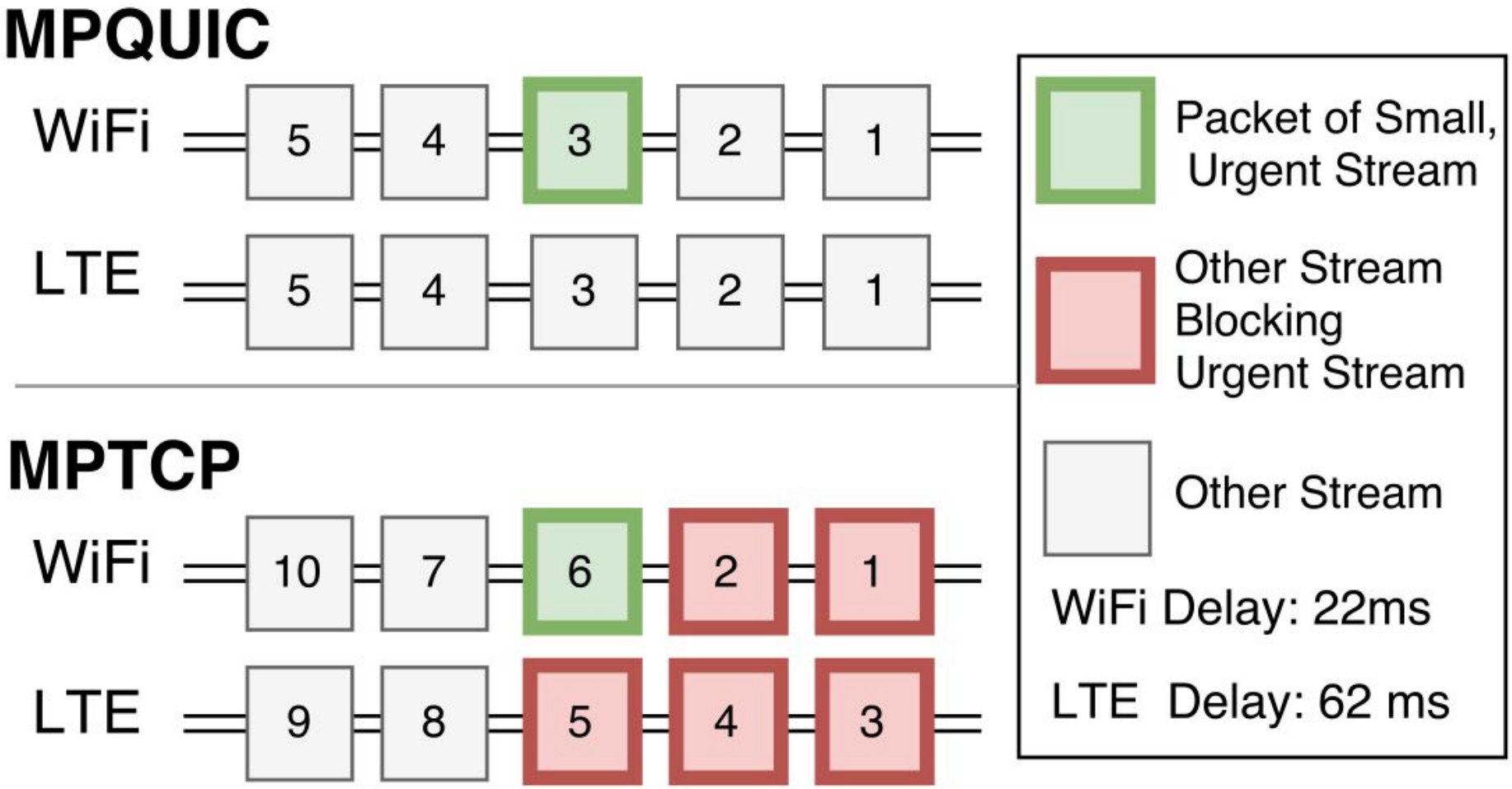
Outlook

- **Complete the design of MPQUIC**
- **Wide deployment and evaluation of various application scenarios**
- **Optimized stream to subflow scheduling (e.g., priorities)**

Thank you for your attention! Questions?

TECHNISCHE UNIVERSITÄT DARMSTADT

Evaluation
Experimantal Priority Scheduler

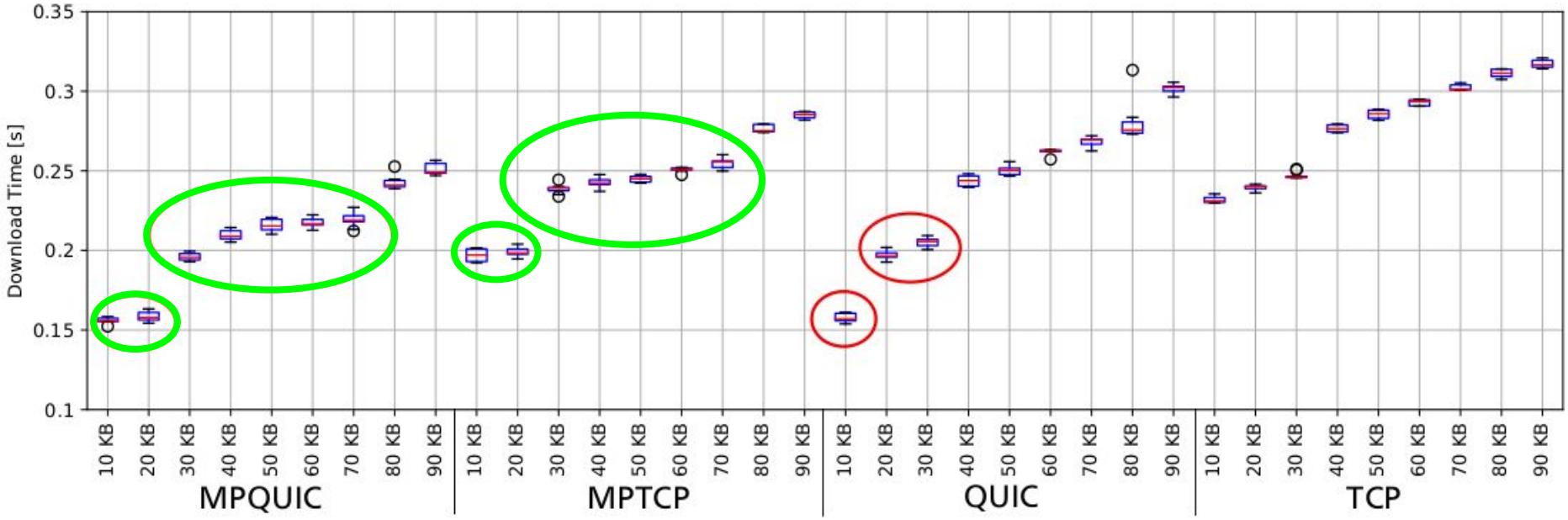


Evaluation
Comparison of MPQUIC, QUIC, MPTCP, TCP

TECHNISCHE UNIVERSITÄT DARMSTADT

HTTP/2 file requests of small sized files

- Bandwidth(const): 10 Mbps
- RTT(const):44ms

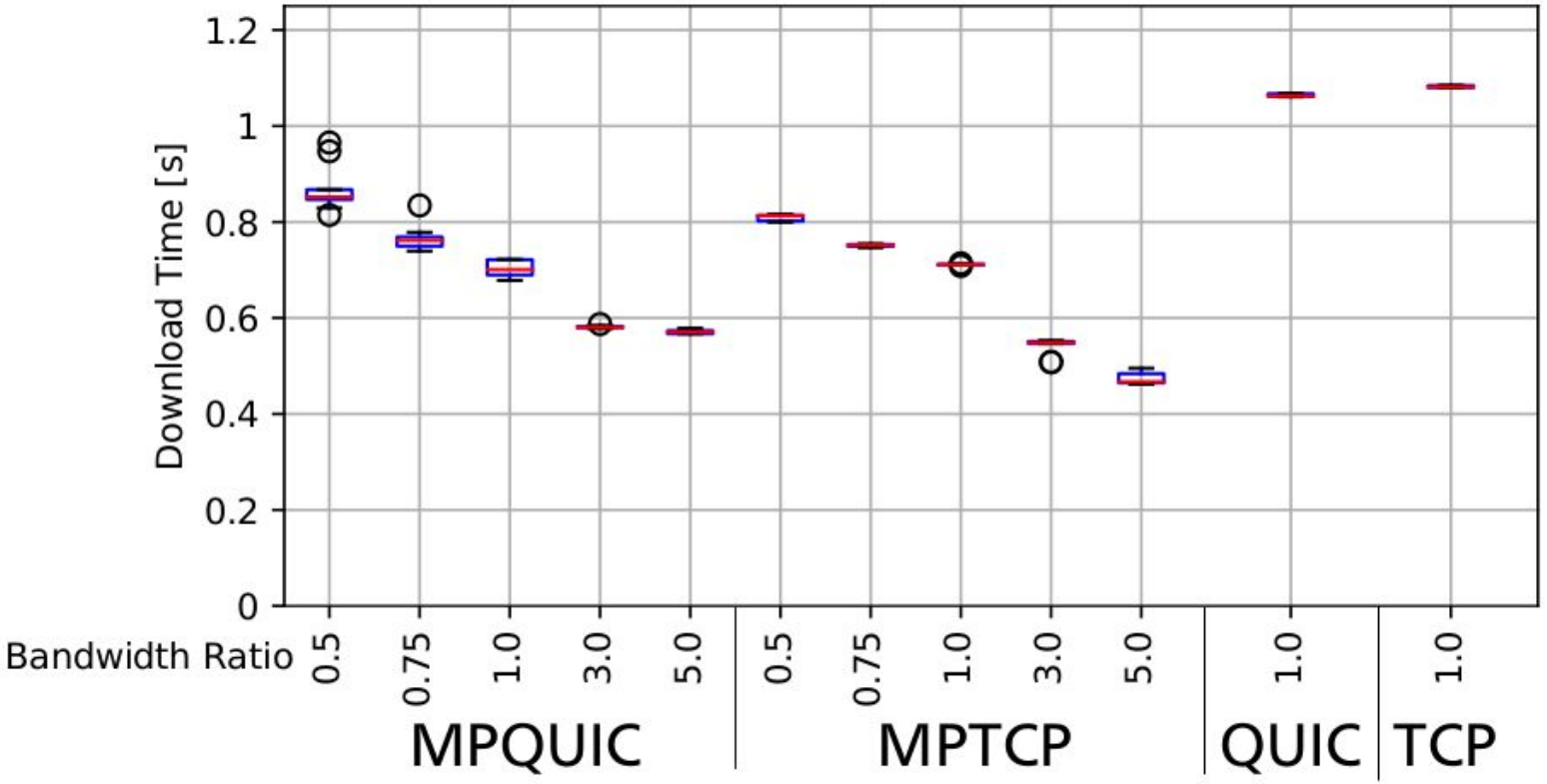


Evaluation
Comparison of MPQUIC, QUIC, MPTCP, TCP

TECHNISCHE UNIVERSITÄT DARMSTADT

HTTP/2 file requests for various bandwidth ratios

- Initial path bandwidth: 10 Mbps
- RTT(const): 44ms
- 1 MB

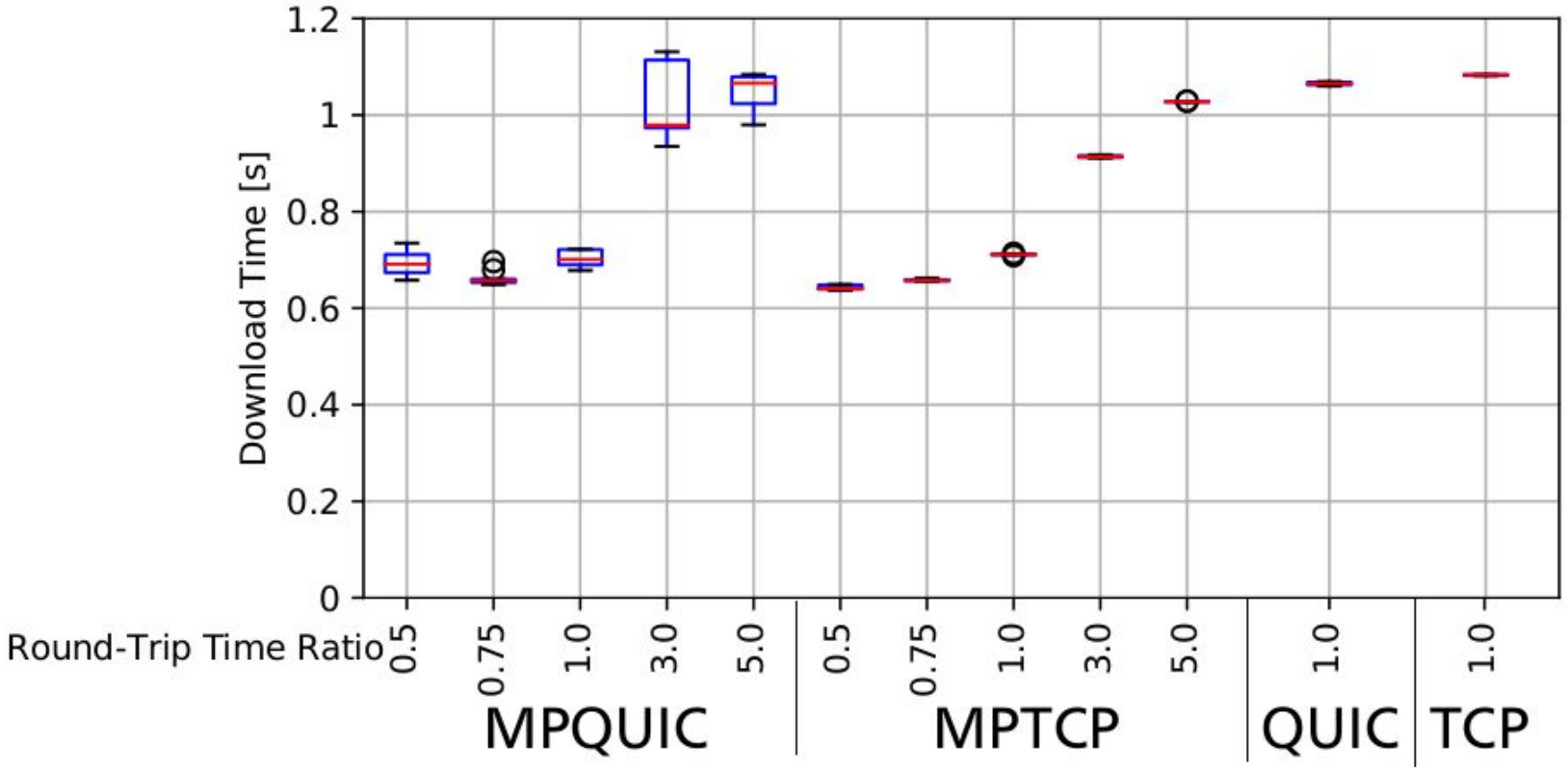


Evaluation
Comparison of MPQUIC, QUIC, MPTCP, TCP

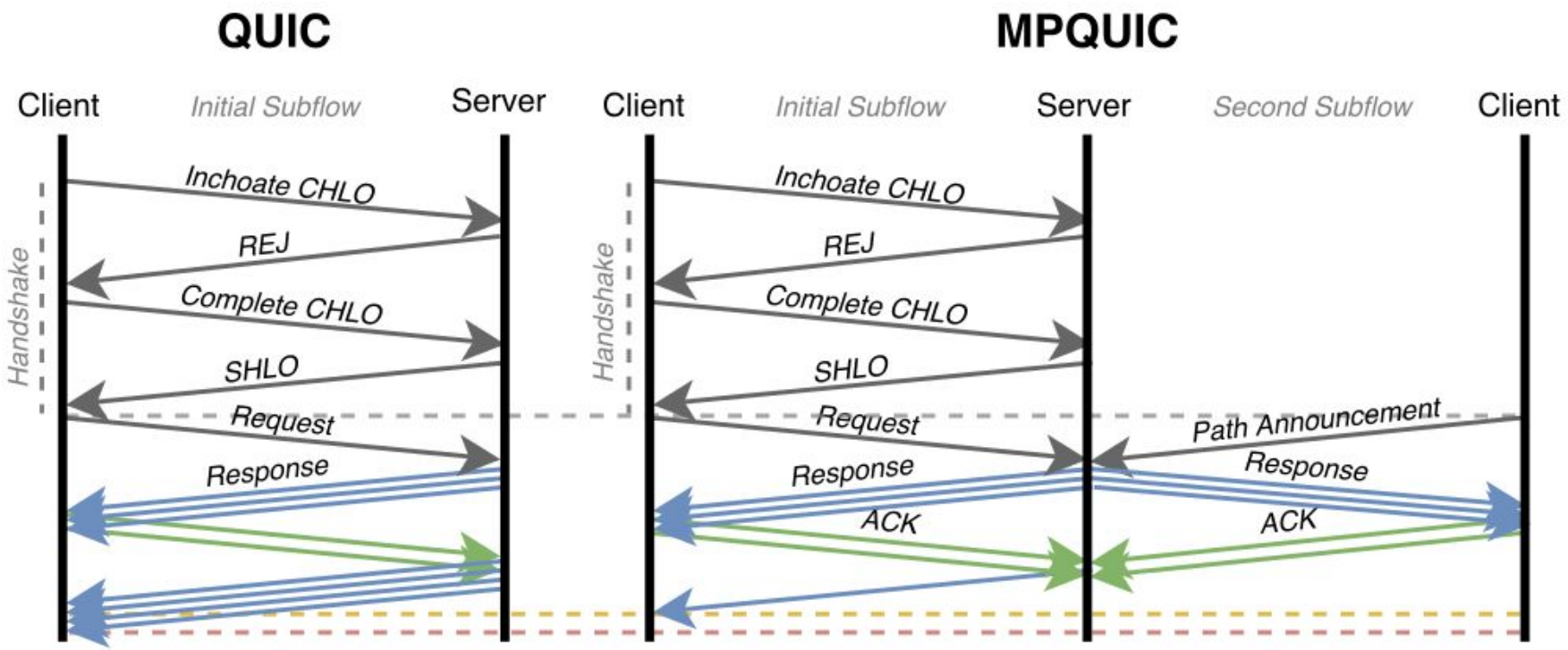
TECHNISCHE UNIVERSITÄT DARMSTADT

HTTP/2 file requests for various RTT ratios

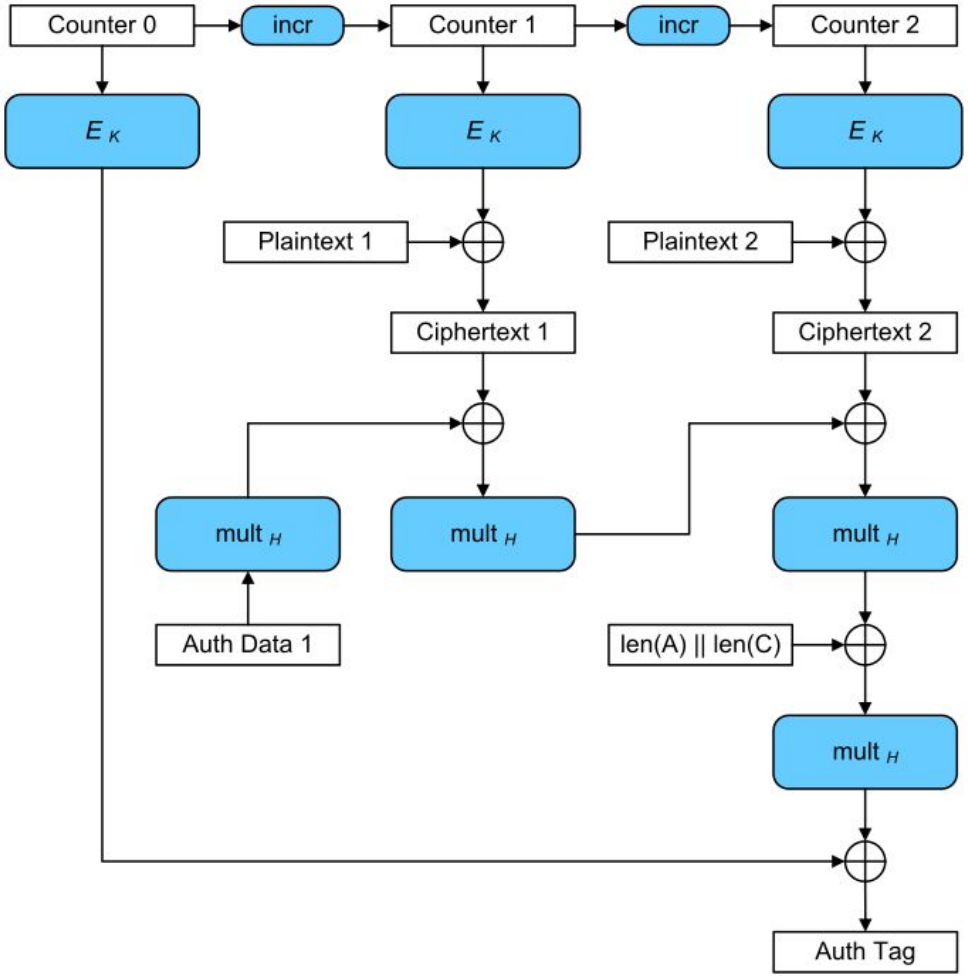
- Bandwidth(const): 10 Mbps
- Initial path RTT: 44ms
- 1 MB



30 KB file download



Encryption


Counter 0
incr
Counter 1
incr
Counter 2
E K
E K
E K
Plaintext 1
Plaintext 2
Ciphertext 1
Ciphertext 2
mult H
mult H
mult H
Auth Data 1
len(A) || len(C)
mult H
Auth Tag

Pubic Flags

TECHNISCHE UNIVERSITÄT DARMSTADT

Public Flag	Meaning
0x01	Public Flag Version
0x02	Public Reset Fag
0x04	Presence of Diversification Nonce
0x08	Presence of 8 byte Connection ID
0x30	(2 bit mask) Indicates length of Packet Number
0x40	Reserved for multipath use
0x80	Not used

RTT Estimate

$$RTT_1 = (r_1 - (r_0 - RTT_0 \times \frac{1}{2})) \times 2$$

Implementation of MPQUIC in quic-go

TECHNISCHE UNIVERSITÄT DARMSTADT

Extension for quic-go

Adjustments for Evaluation

- Initial congestion window size (10 packets)
- Default accept source-address-token
- Simple HTML Web-Scraper
- HTTP/2 TCP/TLS server and client

Plug and Play Scheduler

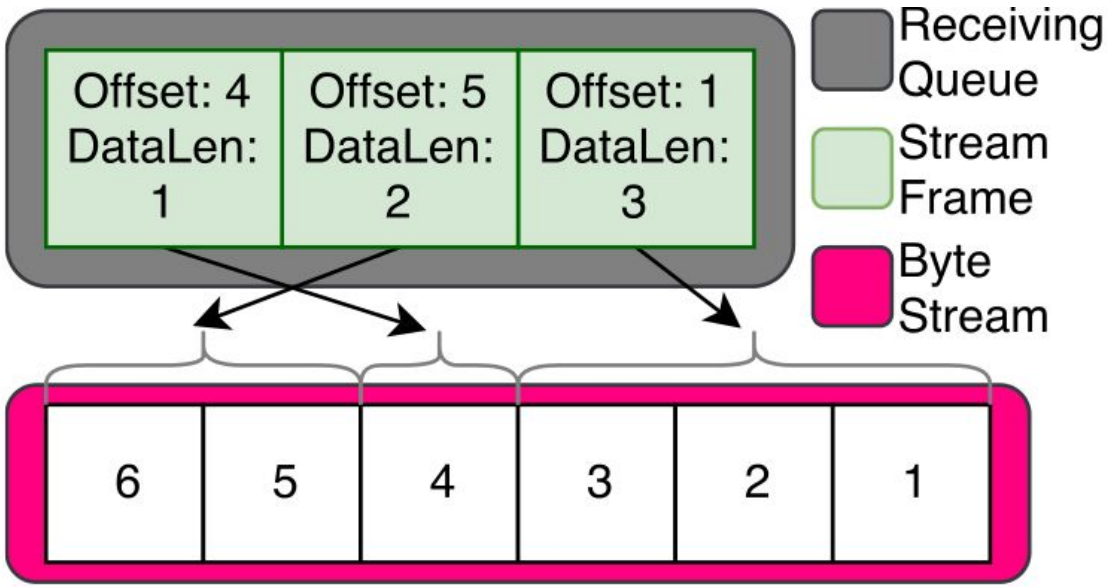
- Declaration of a stream and subflow scheduler lambda function
- Enabled easy modification of the scheduler
- Select scheduler via command-line parameter

Background: QUIC Streams

TECHNISCHE UNIVERSITÄT DARMSTADT

QUIC Streams:

- Independent sequences of data
- Multiple Streams per Connection
  - Identifier: StreamID
  - In-order delivery
    - Offset



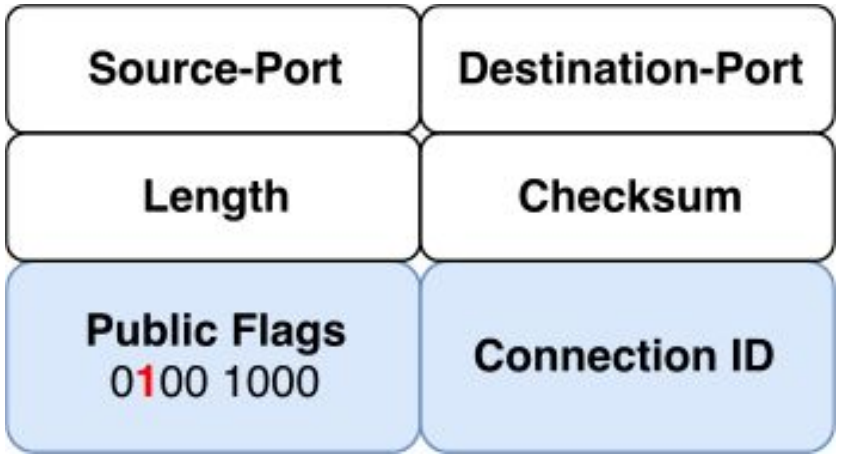
No data ordering based on Packet Numbers!

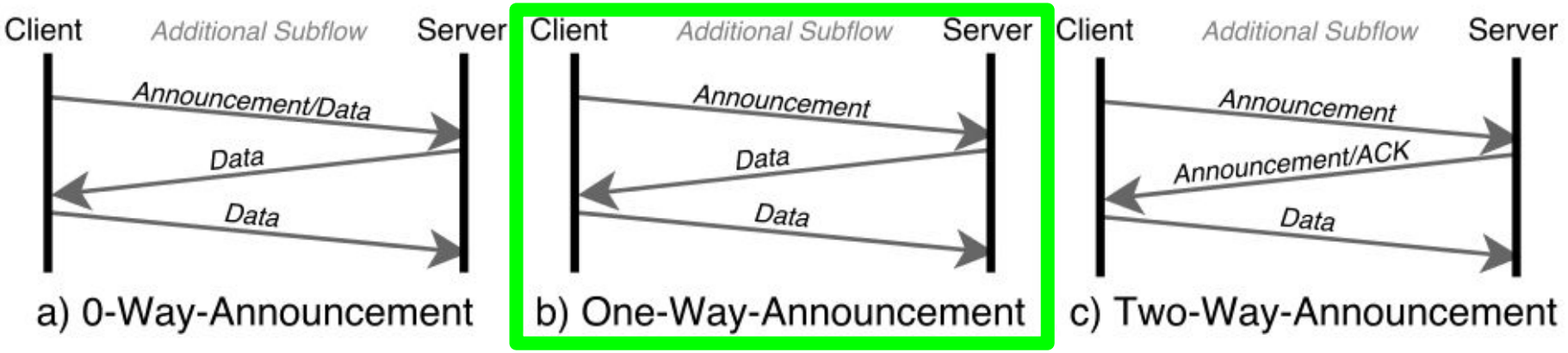
The Design of MPQUIC
Subflow Establishment: Announcement

TECHNISCHE UNIVERSITÄT DARMSTADT

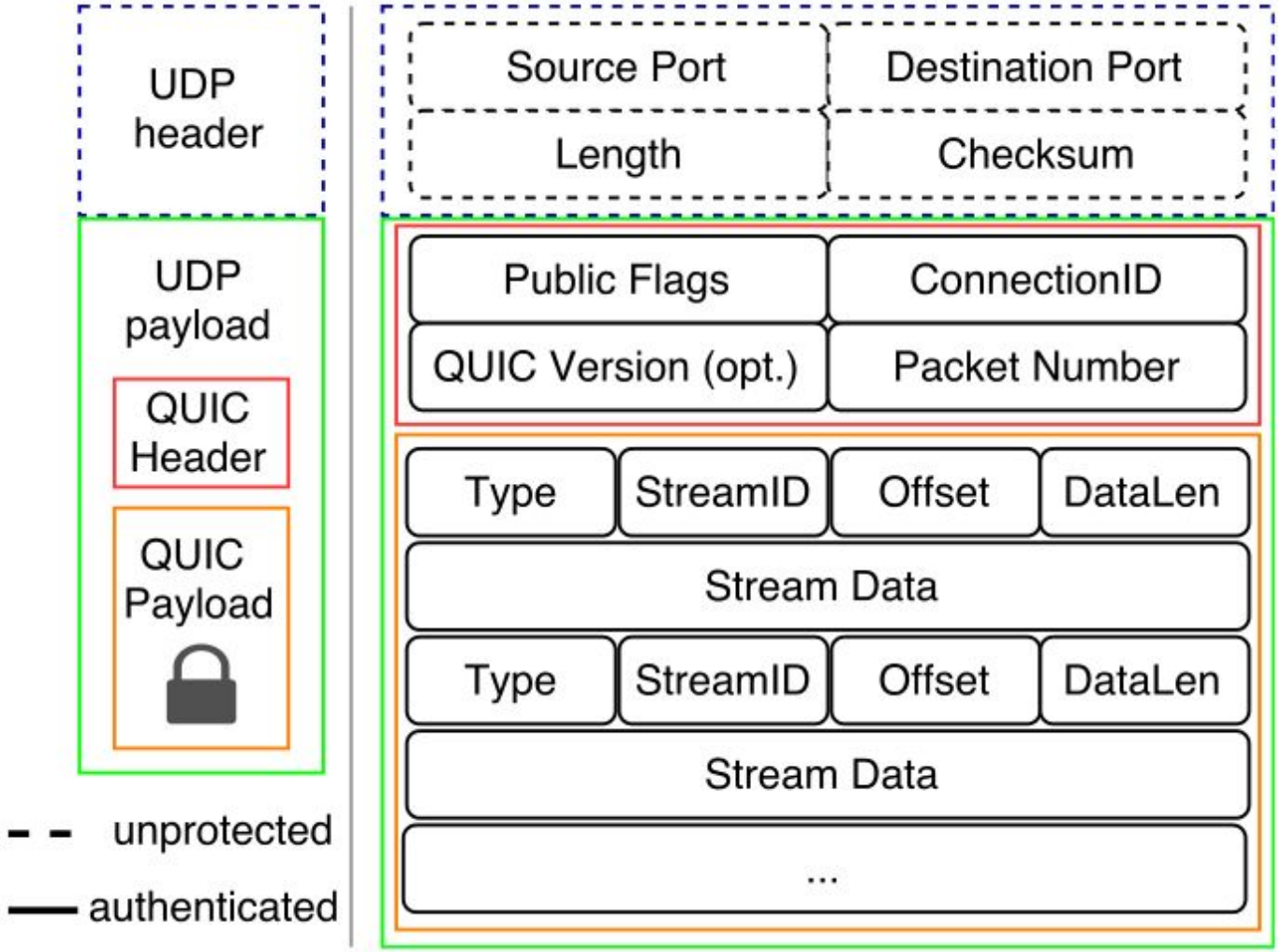
We call it Announcement

- Design of an
  - Announcement Packet
  - Announcement-Handshake

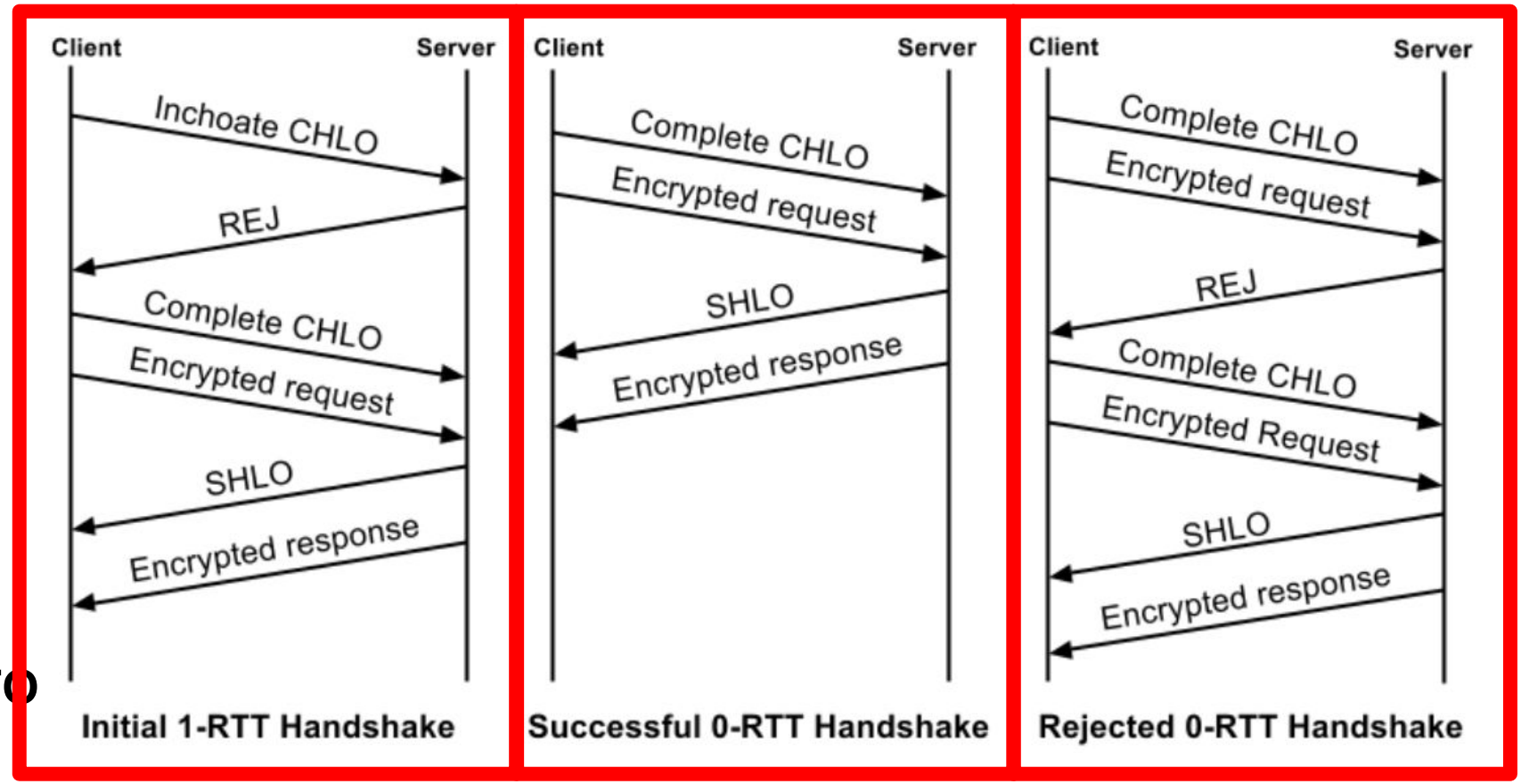




Background: QUIC Packets and Frames



QUIC Connection Establishment



TO

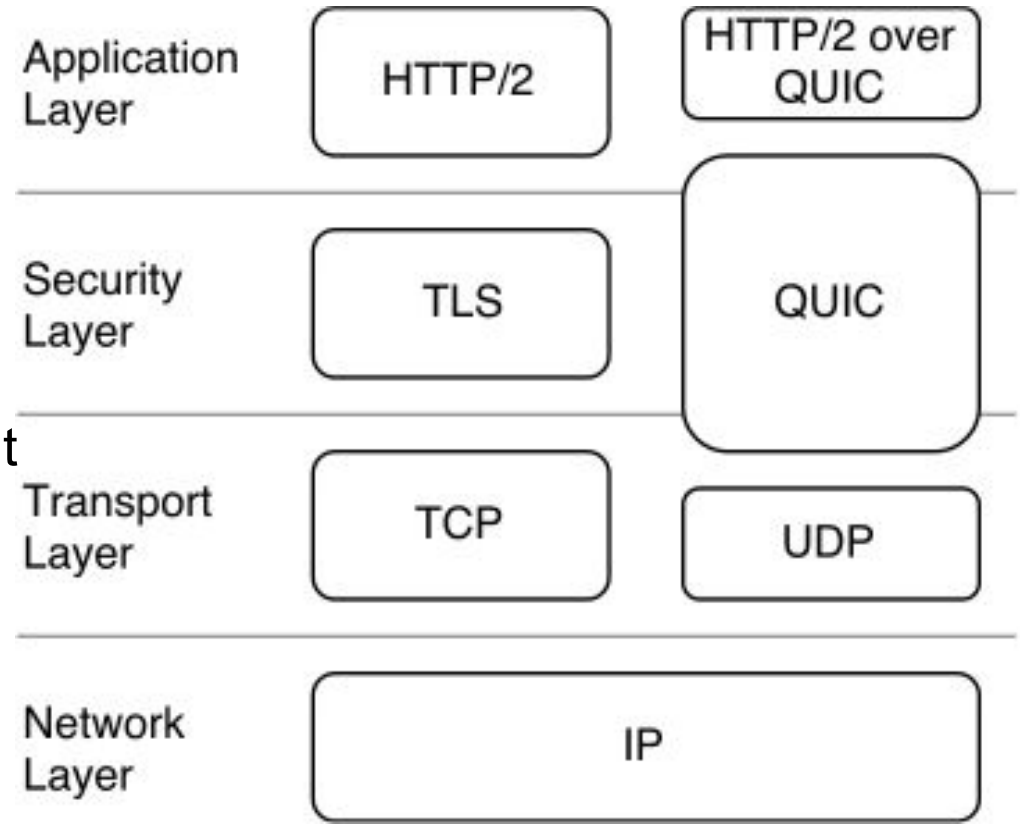
[1] Langley et al. *The QUIC Transport Protocol: Design and Internet-Scale Deployment.* *In Proceedings of the Conference of the ACM Special Interest Group on Data Communication. ACM, 183–196.*

Background: QUIC

TECHNISCHE UNIVERSITÄT DARMSTADT

QUIC Features:

- Reliable In-Order Delivery
- Congestion and Flow Control
- Fast Connection Establishment
- Stream Multiplexing
- ...



Is QUIC multipath suitable?

TECHNISCHE UNIVERSITÄT DARMSTADT

Multiplexed Stream

- In-order delivery is not based on Packet Numbers
  - No dependencies between subflows
- Benefits scheduling optimization

Identifying a Connection

- ConnectionID
  - Concept of Connection Migration
  - Fast Connection Establishment
  - Useful for subflows establishment

Deployability

- Shipment with application upgrades
- Inside the UDP Payload
- Authenticated Headers

Reserved Multipath Flag

The Design of MPQUIC
Specifying the individual Components