eBPF-Assisted Relays for Multimedia Streaming

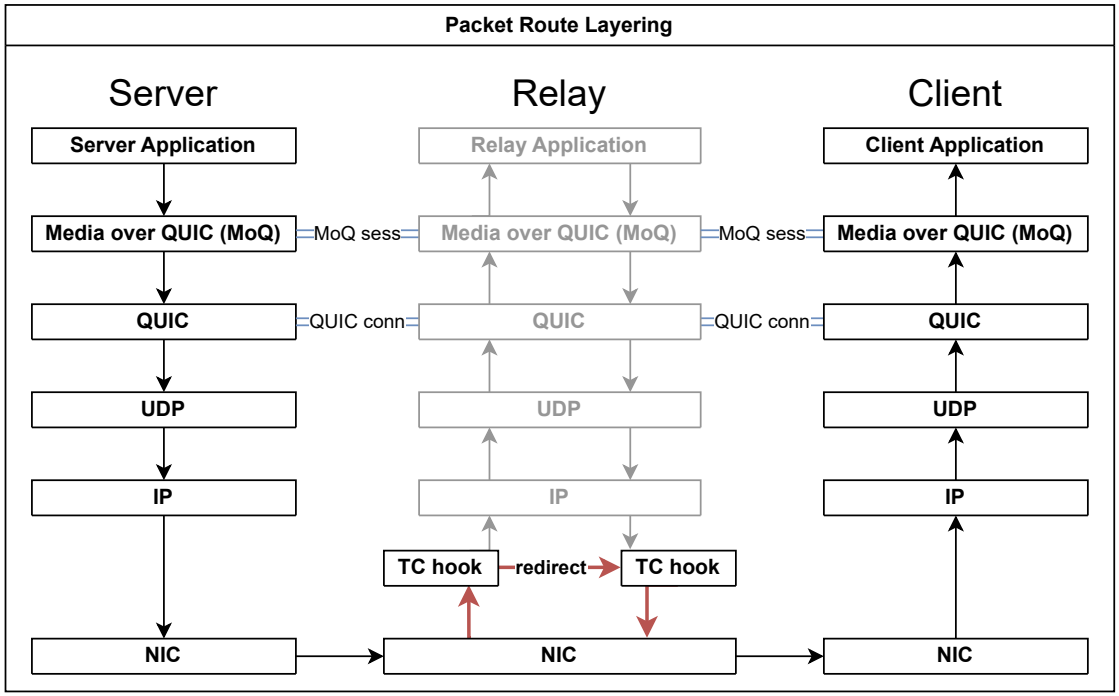
Daniel Alexander Antonius Pfeifer

Technical University of Munich

August 12, 2024

Motivation

- Shorten Critical Path
- Avoid Network Stack Traversal
- Reduce Forwarding Delay

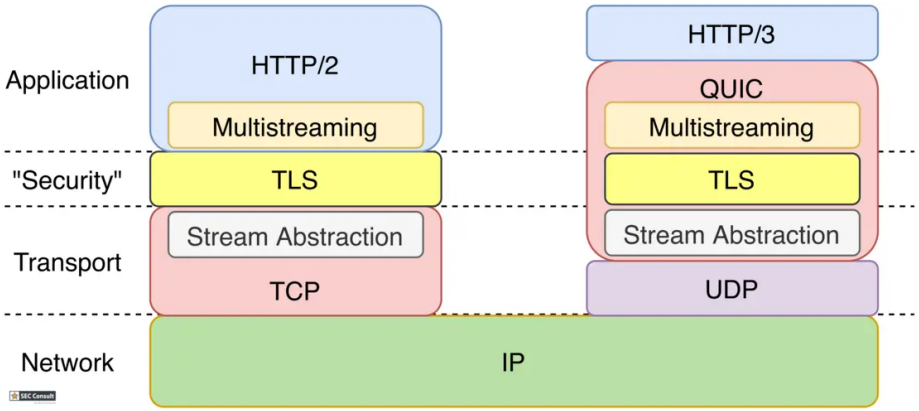


Research Question

- *Improve relay performance by using eBPF technology?*
  - *Remove userspace packet-processing from critical path?*
  - *Handle packet en- and decryption?*
  - *Communication between userspace and the eBPF program?*
  - *Generalize to support other protocols?*

QUIC

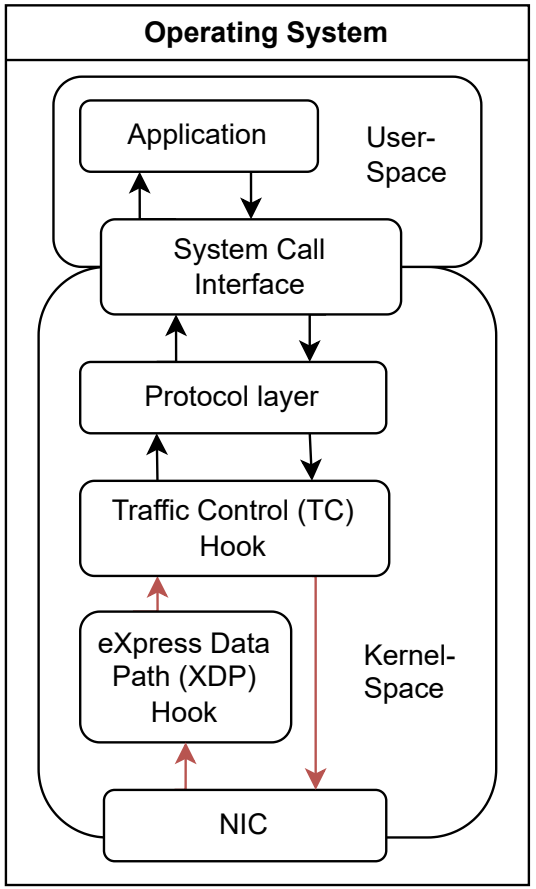
- Started by Google as *Quick UDP Internet Connections*
- Standardized by IETF
- Fast Development Cycle since Userspace Implementation
- Gets rid of Issues like Head-of-Line Blocking



Source: https://sec-consult.com/blog/detail/better-dont-be-too-quick/

eBPF

- Kernel-Internal Virtual Machine
- Used for Packet Filtering and Tracing
- Multiple Hook-Points in the Kernel (e.g. XDP and TC)
- Userspace Communication via Maps



QUIC Adaptations

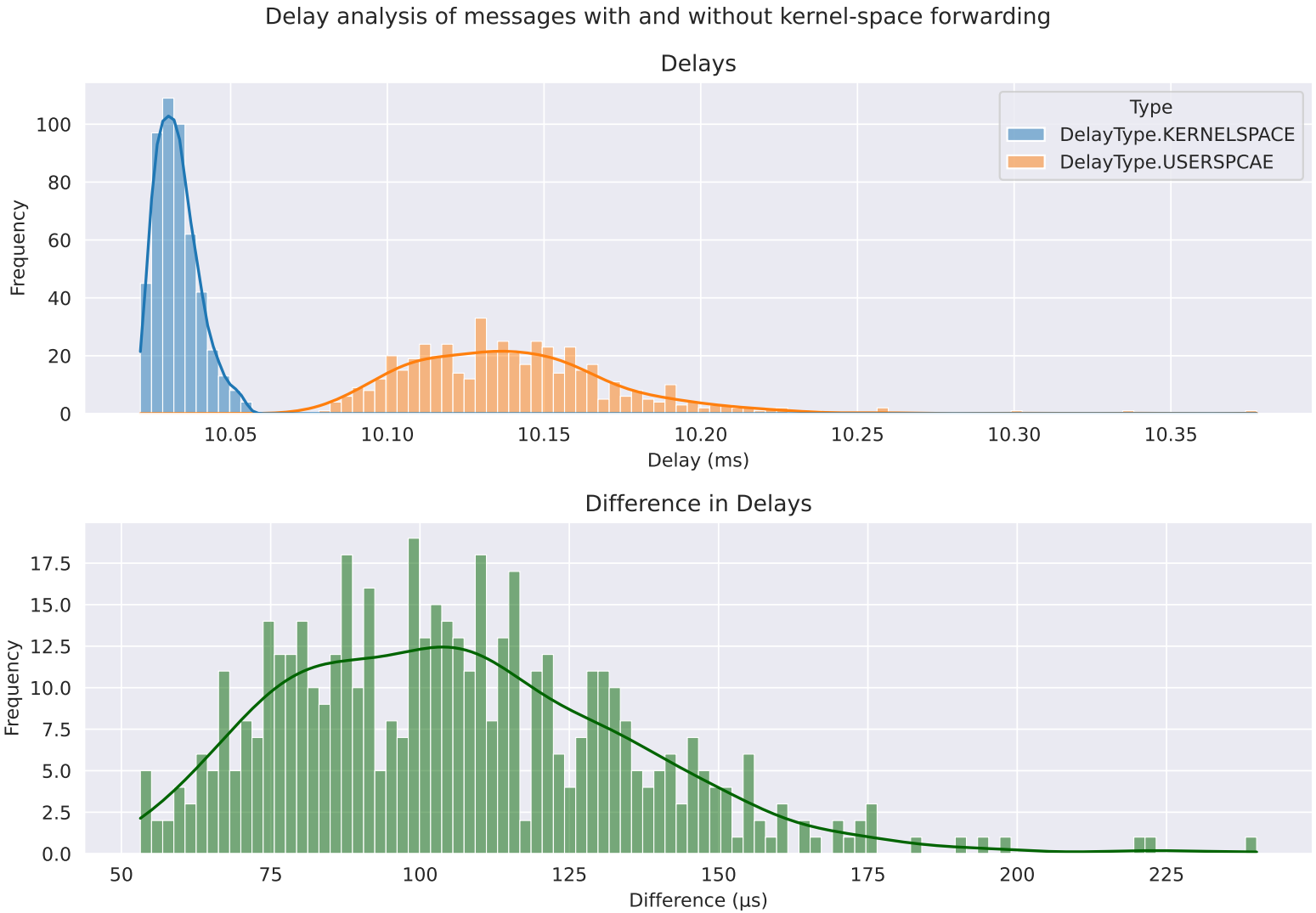
eBPF Setup

Userspace Synchonization

Congestion Considerations

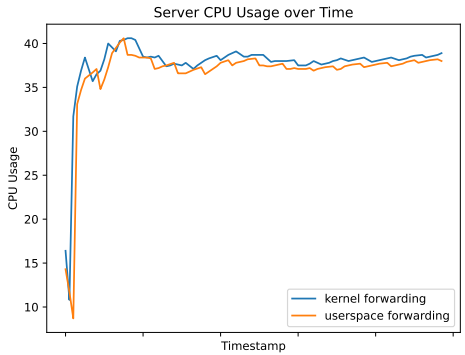
Test Setup

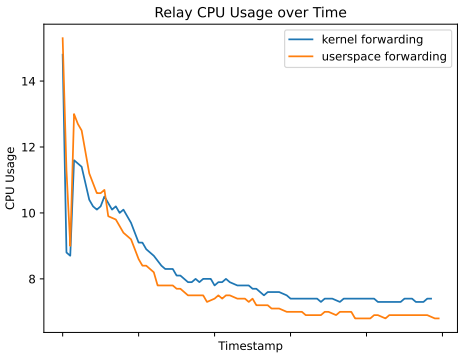
Test Results Delay Reduction

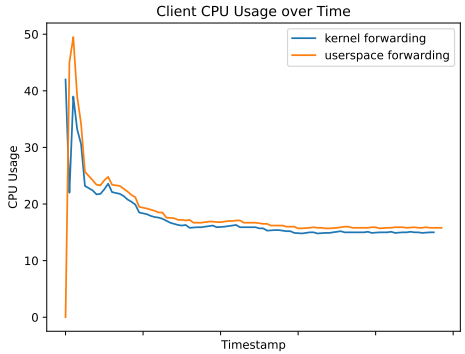


Test Results CPU Usage

- No Impact on CPU Usage
- Fewer System Calls
  - Mainly due to reduced Userspace Synchronization







System Calls

- Example Stream of 30 Seconds
- Overall System Calls
  - Userspace forwarding: 296132 calls
  - eBPF forwarding: 225674 calls
  - Reduction of 24%
- *futex*
  - Reduction of 34%
  - 21666 calls instead of 32940
- *nanosleep*
  - Reduction of 42%
  - 14293 calls instead of 24716
- *epoll_wait*
  - Reduction of 67%
  - 11289 calls instead of 34149

Conclusion

Future Work