

# QUIC-Aware Proxying Using CONNECT-UDP

Tommy Pauly & David Schinazi  
MASQUE  
IETF 110, March 2021, Virtual

# Why QUIC-aware?

Allow proxies to reuse upstream ports

Improve performance!

- Processing overhead on clients and proxy
- Loss of MTU due to encapsulation

# Applicability

Useful for enabling multiple proxy hops

- First N proxy hops can use forwarding mode, and a last hop can re-encapsulate

Useful for proxies that exist for routing optimization, access control, or IP obfuscation without needing extra payload security

Both client and proxy must consent to use forwarding

# Test Setup

## QUIC-aware CONNECT-UDP proxy

- Quiche H3 and QUIC
- NIC forwarding implemented with XDP<sup>1</sup> and eBPF<sup>2</sup> rules

## Target HTTP/3 server

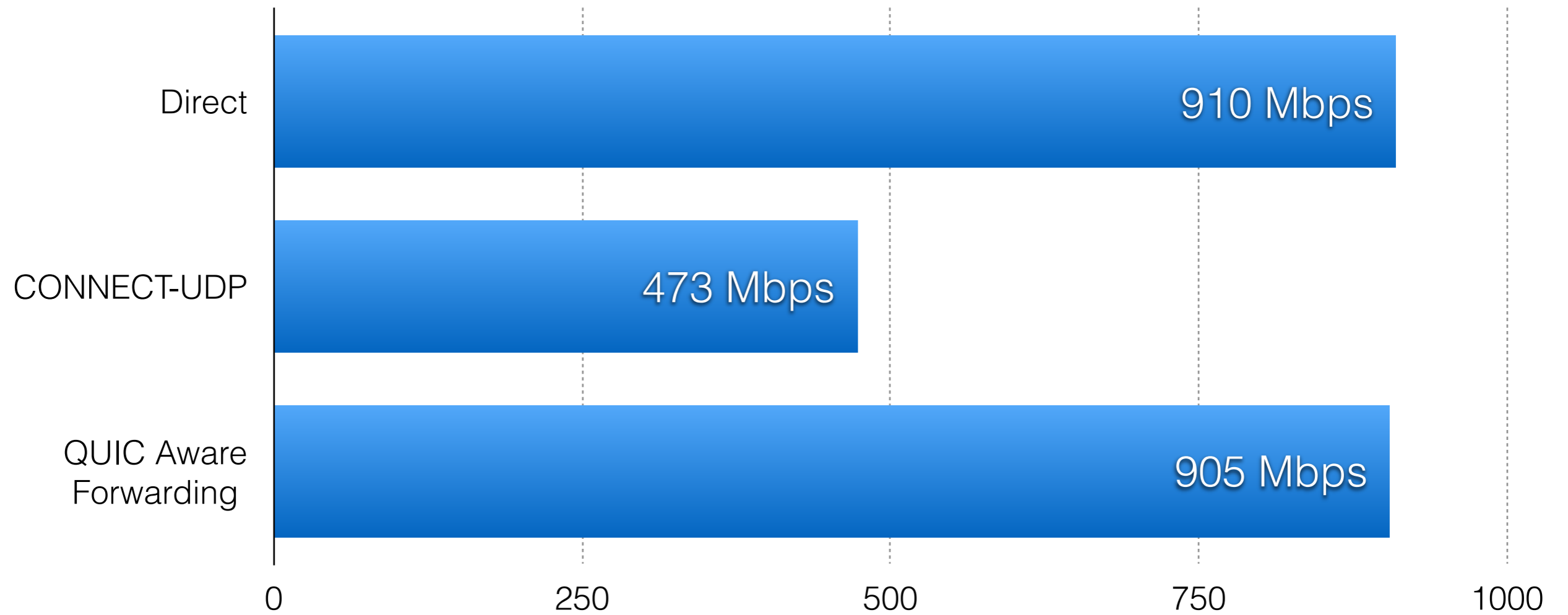
- NGINX with quiche

## 1g Ethernet link

1. eXpress Data Path, <https://www.iovisor.org/technology/xdp>

2. Extended Berkeley Packet Filter, <https://www.kernel.org/doc/html/latest/bpf/index.html>

# Measurements



# Interesting Issues

Changes in MTU due to forwarding mode

Servers can better use the link if they do PMTUD, or else get an explicit signal

Advice for connection migration

Similar to CONNECT-UDP in general, but forwarding mode must be sure to pace/drop packets on unvalidated paths

Prohibit sharing server-facing sockets with non-QUIC flows

