Secure Middlebox-Assisted QUIC

Mike Kosek, **Benedikt Spies**, Jörg Ott

Technical University of Munich
School of Computation, Information and Technology
Department of Computer Engineering
Chair of Connected Mobility

IFIP Networking 2023
Session 5: Transport Protocols
2023-06-13

Motivation

- Evolution of the Internet was driven by the end-to-end model
- Reliability, congestion control, and security are usually realized end-to-end

- Breaking up end-to-end connections to improve transport
  - Content Distribution Networks
  - Overlays like Media Over QUIC (moq)
  - Performance Enhancing Proxies

⇒ **Middleboxes can be really useful**

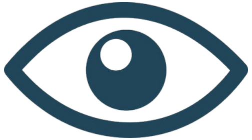
- Middleboxes rely on the ability to read and modify protocol exchanges

⚡ **QUIC protects most protocol information from the network**

QUIC Recap

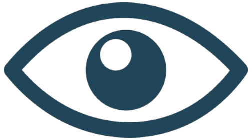
- General Purpose Transport Protocol
- Standardized in 2021 by RFC 9000
- Advantages over TCP+TLS
  - Improved security
  - Stream multiplexing
  - Faster connection establishment
  - Network path migration
  - Extensible by design
- Over 25 % of websites use QUIC

What we want



- No transparent middleboxes ossifying the internet
- Consciously added by endpoints

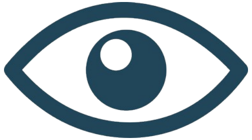
What we want



- No transparent middleboxes ossifying the internet
- Consciously added by endpoints

- Middleboxes with the least privileges to perform a certain task
- End-to-end protected application data

What we want

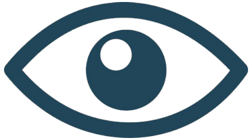


- No transparent middleboxes ossifying the internet
- Consciously added by endpoints

- Middleboxes with the least privileges to perform a certain task
- End-to-end protected application data

- Flexible and effective
- Minimal changes to QUIC

What we want



- No transparent middleboxes ossifying the internet
- Consciously added by endpoints

- Middleboxes with the least privileges to perform a certain task
- End-to-end protected application data

- Flexible and effective
- Minimal changes to QUIC

- Middlebox usage on demand
- Support for off-path middleboxes

Motivation

Middleboxes require a conscious decision and consent by either or both endpoints

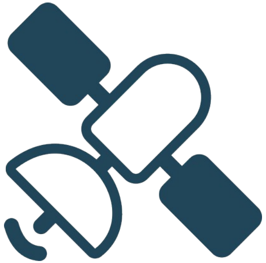
- Out-of-band
  - Establish an independent signaling channel between one or more endpoints and the middleboxes
  - Information is explicitly sent to the middleboxes

- In-band
  - Include middleboxes en route during connection setup, or insert them later
  - Different levels of encryption expose selective information to the middleboxes

What we did

SMAQ

Secure Middlebox-Assisted QUIC



Design and implement in-band mechanism to insert middleboxes into an end-to-end encrypted QUIC connection

Case study on using SMAQ for building Performance Enhancing Proxies for long-delay satellite networks

Secure Middlebox-Assisted QUIC

SMAQ

State Handover

enabling connection state transfer and restoration across machines

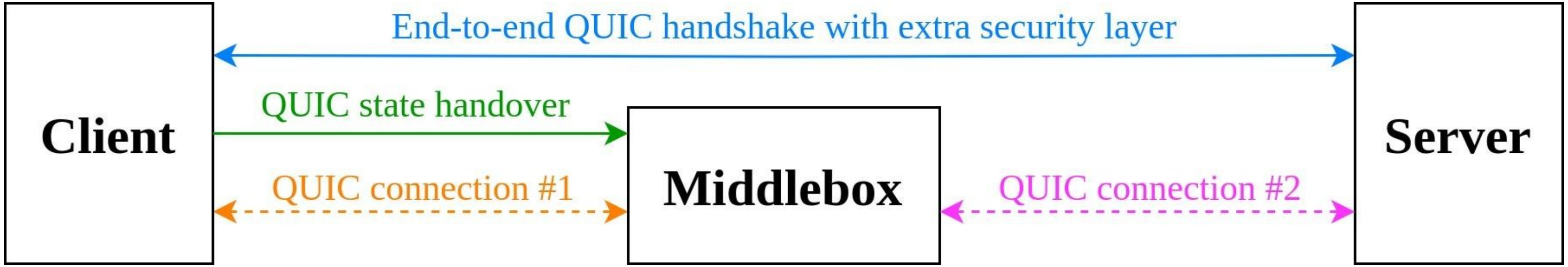
Enhanced Migration

enabling server address migration and address validation across connections

Extra Security Layer

granting only selected privileges to a middlebox

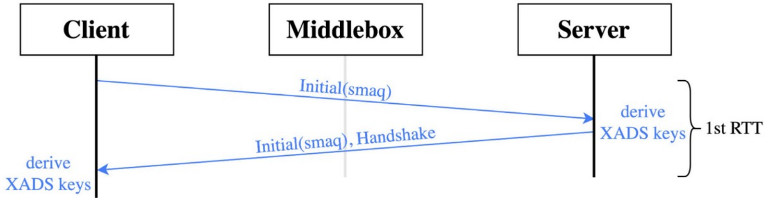
Secure Middlebox-Assisted QUIC



- End-to-end QUIC handshake with setup of additional security layer (Blue)
- Client involves middleboxe by sharing state, including selected key material (Green)
- Split connection by address migration to support off-path middleboxes

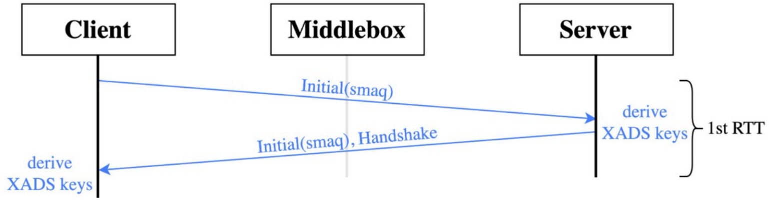
Design: Connection Setup

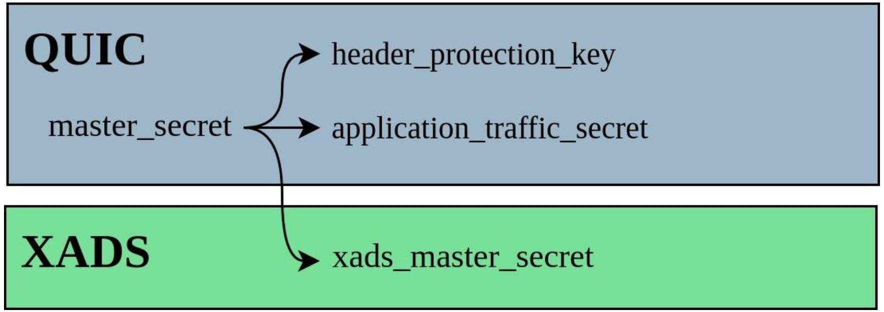
Design: Connection Setup



Design: Connection Setup

- XADS: Extra Application Data Security
- TLS 1.3 record protocol



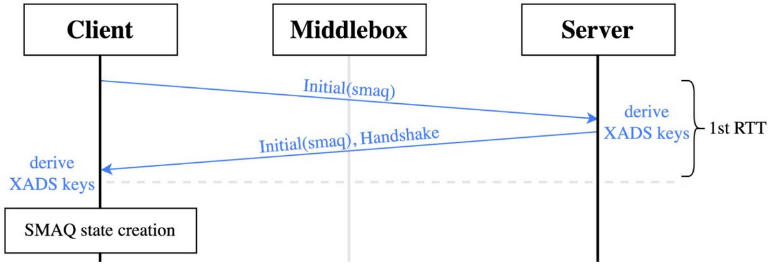


Design: Connection Setup

- Client creates SMAQ state

SMAQ state: QUIC connection properties and cryptographic information, e.g.:

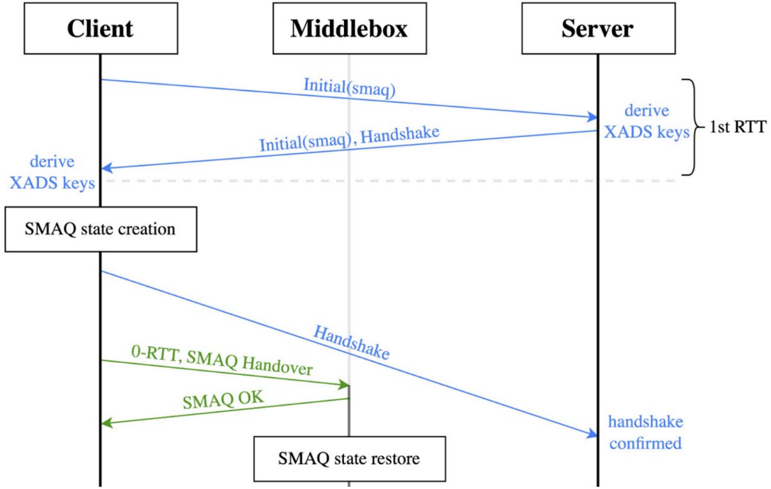
- QUIC version
- Connection IDs
- Endpoint addresses
- Traffic secrets
- Header protection keys



Design: Connection Setup

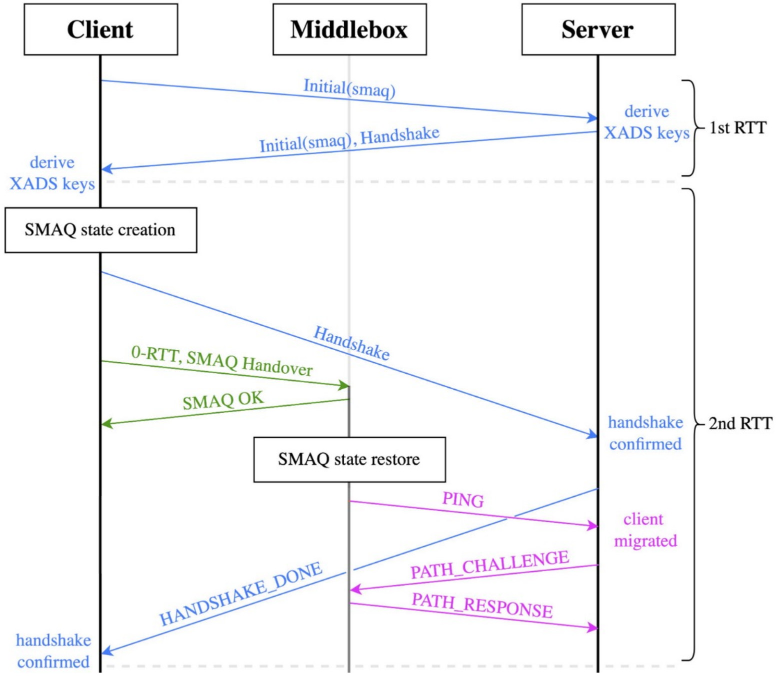
- SMAQ state is transferred and restored a the middlebox

⚠ **State does not contain XADS keys**



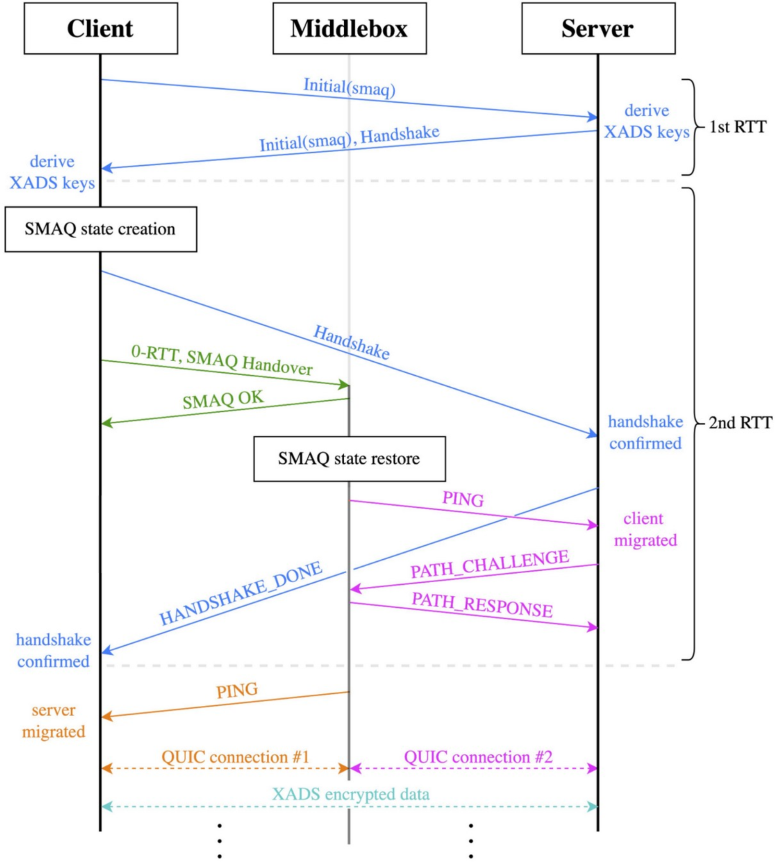
Design: Connection Setup

- Middlebox sends PING frame to the server, while acting as the client
- Triggering standard QUIC path migration and address validation



Design: Connection Setup

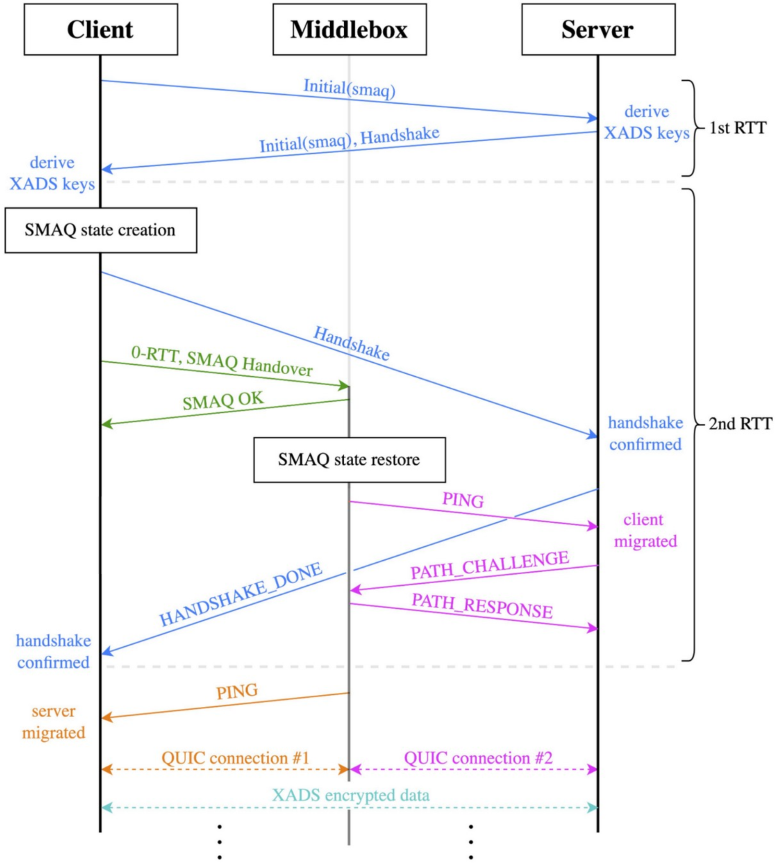
- Middlebox sends PING frame to the client, while acting as the server
- Triggering path migration
    - Address is already validated
- Split connection in two control loops
    - Spliced by middlebox
- Client and server no longer exchange data directly
- Forwarded streams remain E2E encrypted



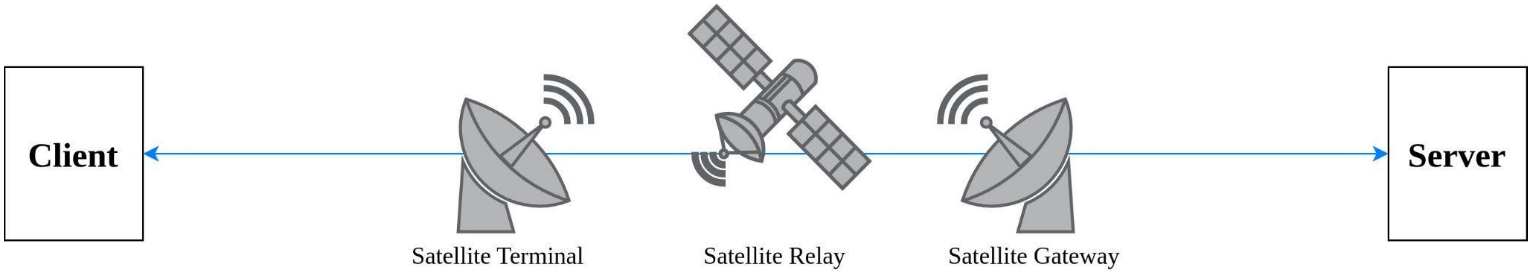
Design: Connection Setup

- Middlebox sends PING frame to the client, while acting as the server
- Triggering path migration
  - Address is already validated
- Split connection in two control loops
  - Spliced by middlebox
- Client and server no longer exchange data directly
- Forwarded streams remain E2E encrypted

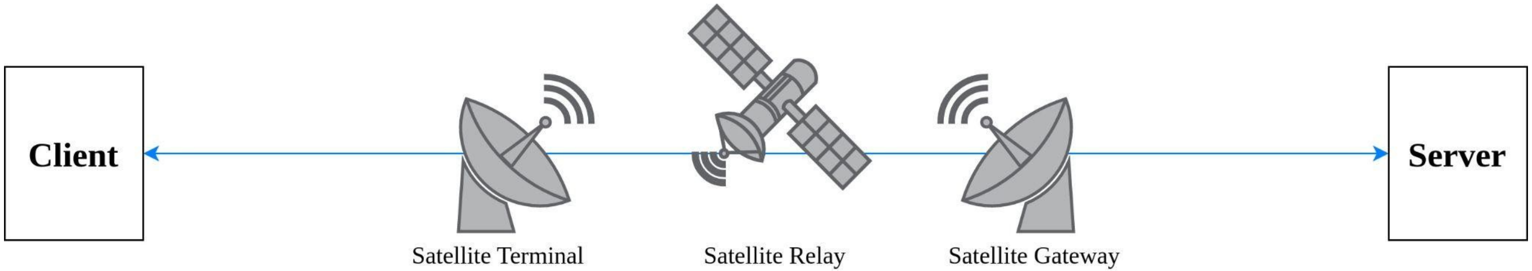
⇒ **Prolongs handshake by one RTT**

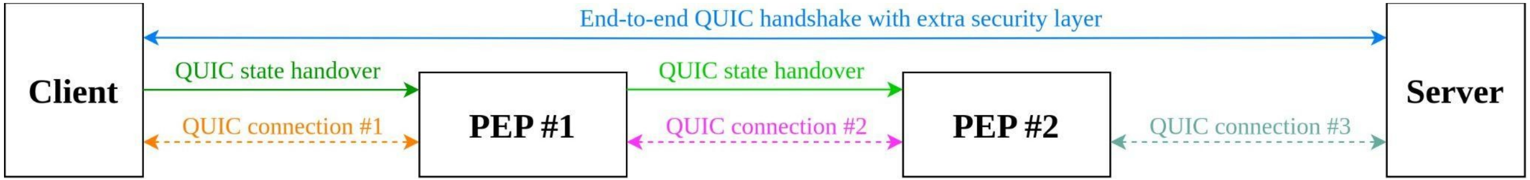


Case Study: Distributed PEPs



Case Study: Distributed PEPs





Case Study: Distributed PEPs

- Satellite Communication Emulation Testbed
  - Reproducible measurements over SATCOM networks
  - PEP #1 is placed on ingress, PEP #2 on egress of SATCOM network
  - Transport connection in between is optimized

- Evaluation
  - 20 Mbps link-layer goodput Server to Client
  - Loss: 0.01 % and 0.1 %
  - Orbits: LEO (112 ms RTT) and GEO (580 ms RTT)

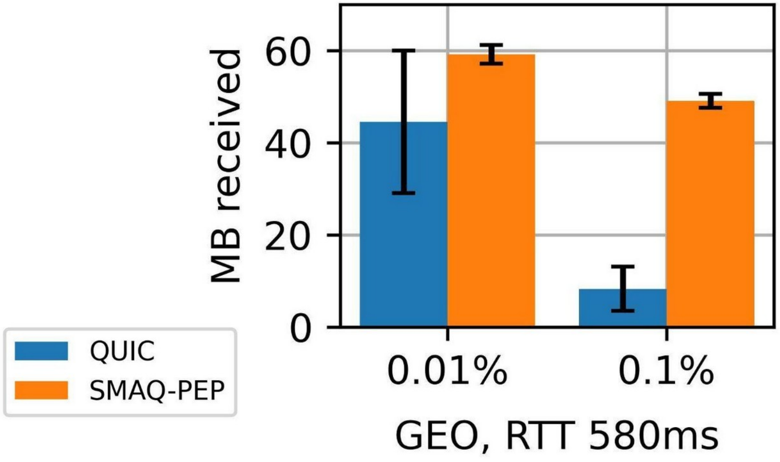
Case Study: Distributed PEPs

- End-to-end QUIC
  - Default QUIC CCA NewReno with IW 10

- PEP-optimized SMAQ-PEP
  - CCA Hybla-Westwood on PEPs
  - Hybla improves congestion window increase on high latency connections
  - Westwood improves goodput over links with high packet loss

- Measurements
  - Bulk Download
  - Web Performance

Case Study: Bulk Download (GEO)

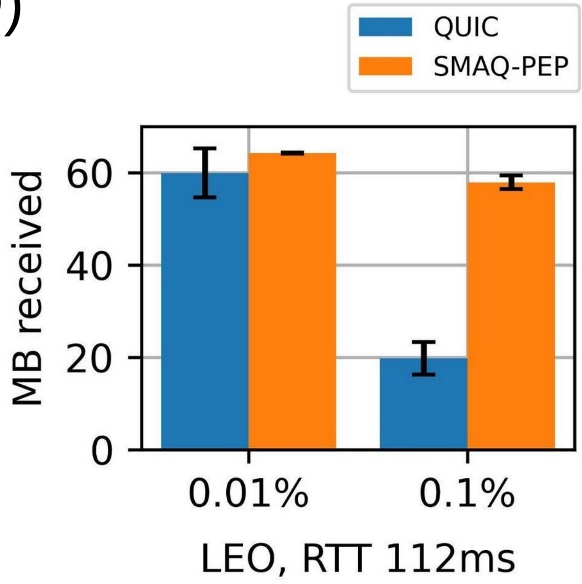


- ~ 30 % more bytes transferred for 0.01 % packet loss in 30 s
- ~ 6 x improvement for 0.1 % loss in 30 s

Median received bytes after 30 seconds bulk download

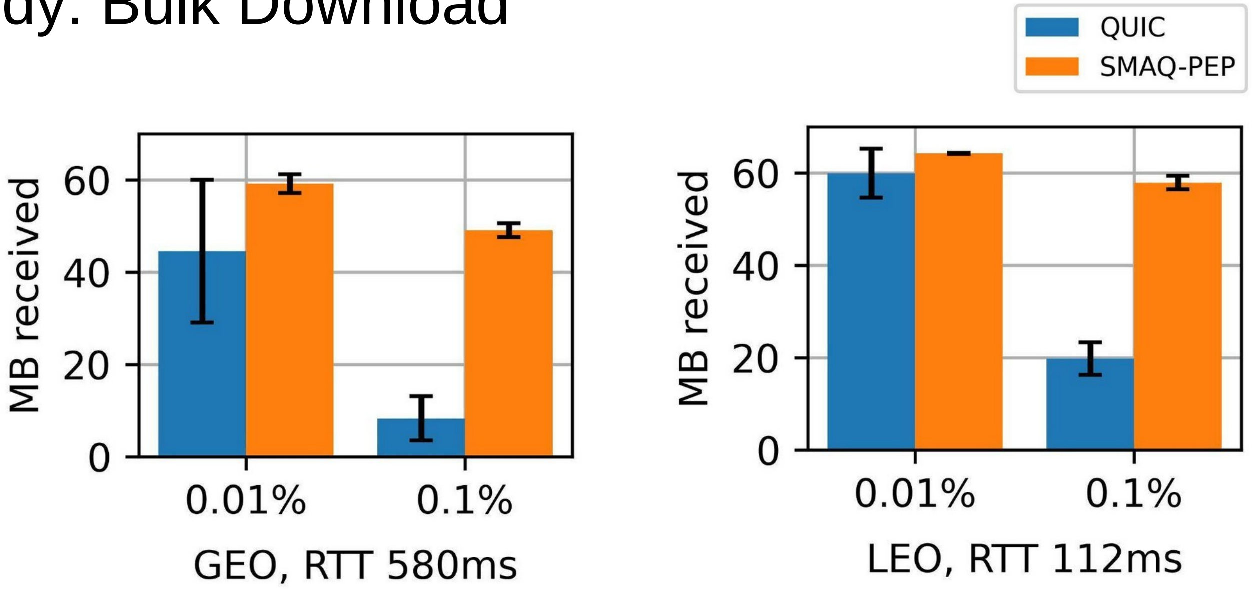
Case Study: Bulk Download (LEO)

- ~ 7 % more bytes transferred for 0.01 % packet loss in 30 s
- ~ 3 x improvement for 0.1 % loss in 30 s



Median received bytes after 30 seconds bulk download

Case Study: Bulk Download

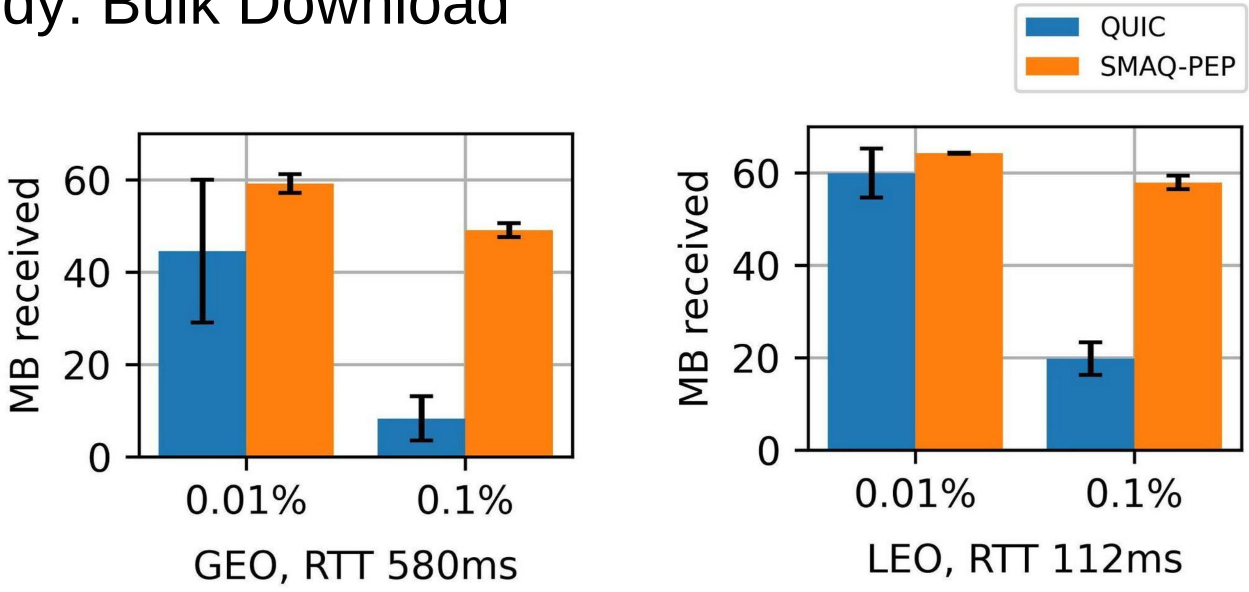


Median received bytes after 30 seconds bulk download

⇒ **The higher the RTT and loss, the higher the benefits of SMAQ-PEP**

Case Study: Bulk Download

beneficial
after ~ 1.9 s
~ 3.3 RTT



beneficial
after ~ 0.6 s
~ 5.4 RTT

Median received bytes after 30 seconds bulk download

⇒ **The higher the RTT and loss, the higher the benefits of SMAQ-PEP**

Case Study: Web Performance

- Top 10 most popular Webpages from Tranco toplist as of December 2022
- Cloned webpages to use on emulated testbed

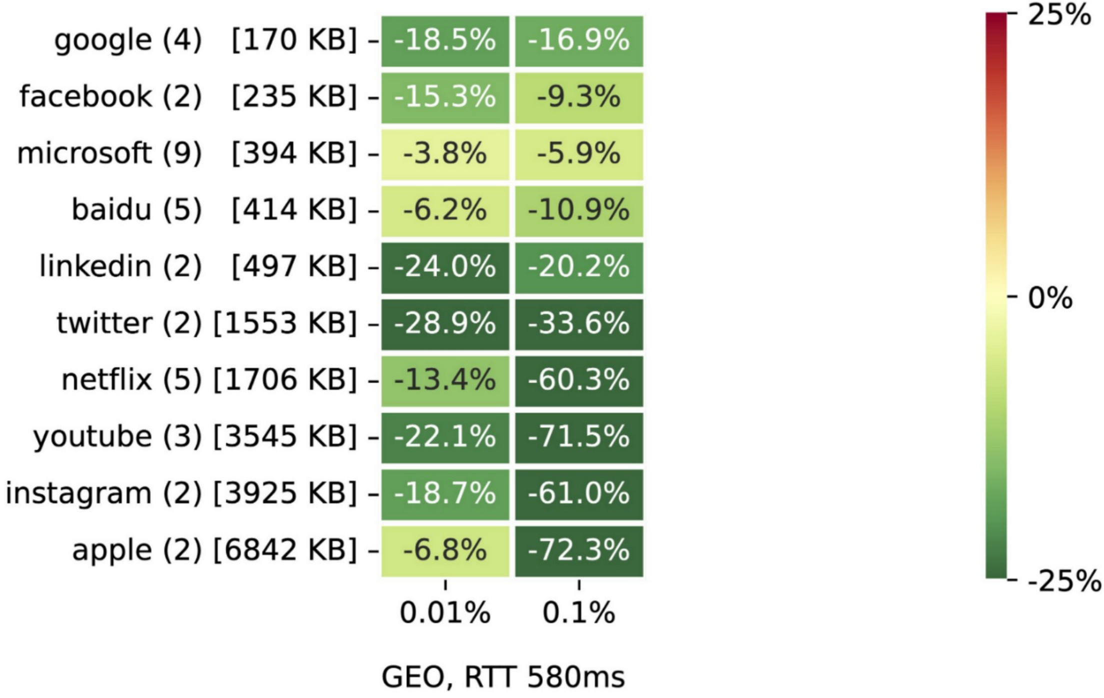
- Webpages are consisting of multiple elements
- Elements are served from different hosts

⇒ Client is required to establish a new connection (+ SMAQ) for each hostname

- Page Load Time using HTTP/3 over QUIC and SMAQ-PEP
- Assess the potential benefits of SMAQ-PEP in a typical web browsing use-case

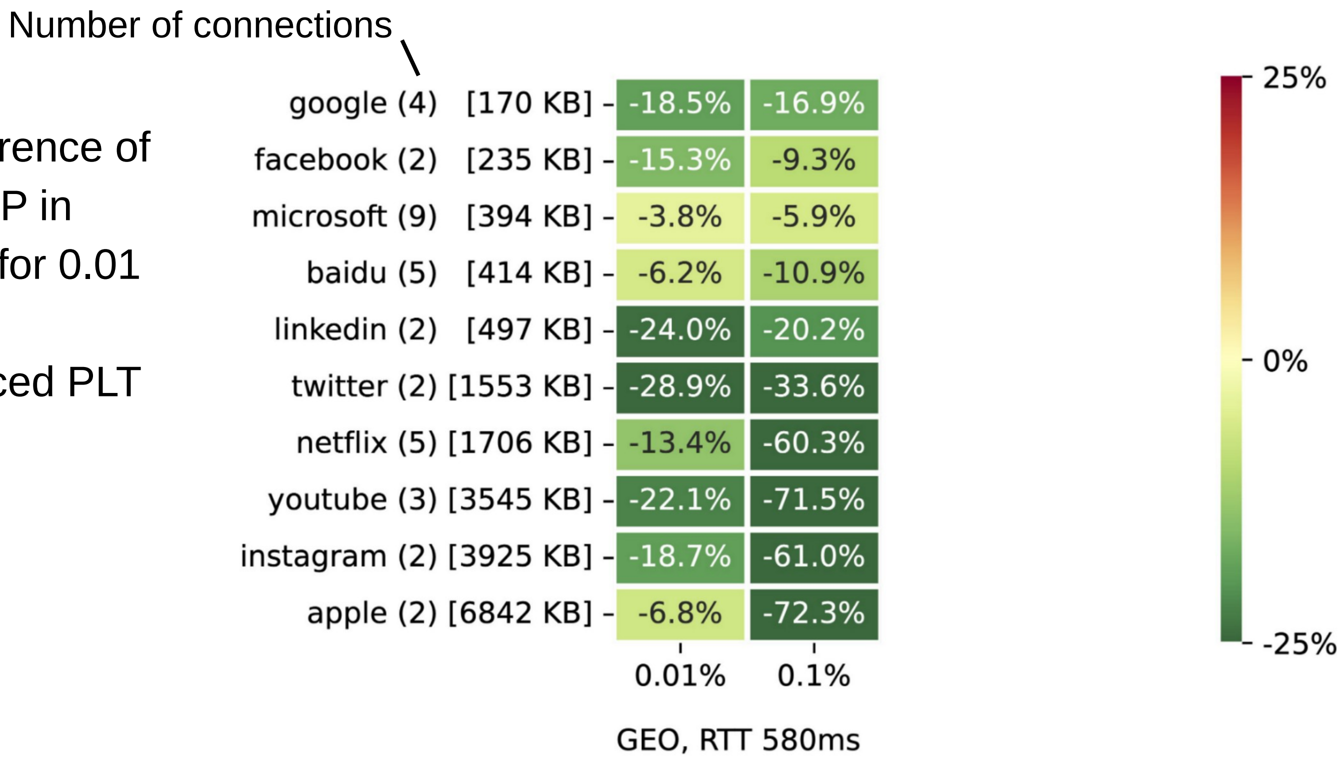
Case Study: Web Performance (GEO)

- Relative median difference of the PLT of SMAQ-PEP in comparison to QUIC for 0.01 and 0.1 % loss
- ~ 4 % to ~ 72 % reduced PLT



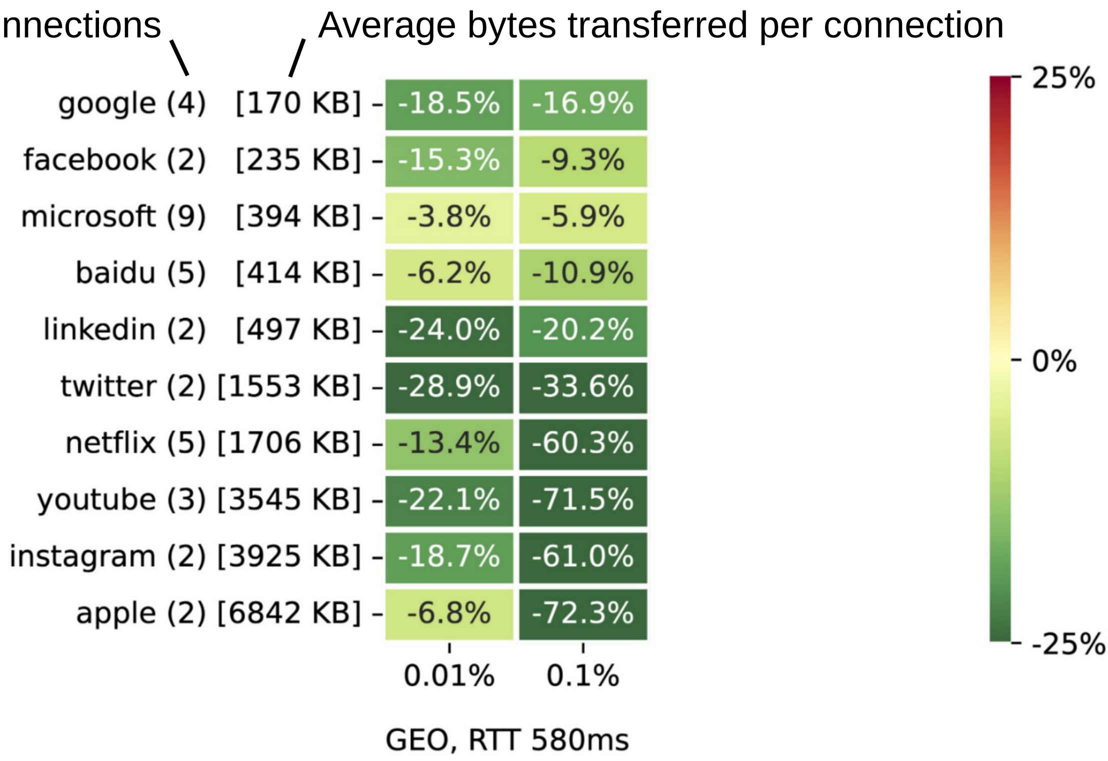
Case Study: Web Performance (GEO)

- Relative median difference of the PLT of SMAQ-PEP in comparison to QUIC for 0.01 and 0.1 % loss
- ~ 4 % to ~ 72 % reduced PLT



Case Study: Web Performance (GEO)

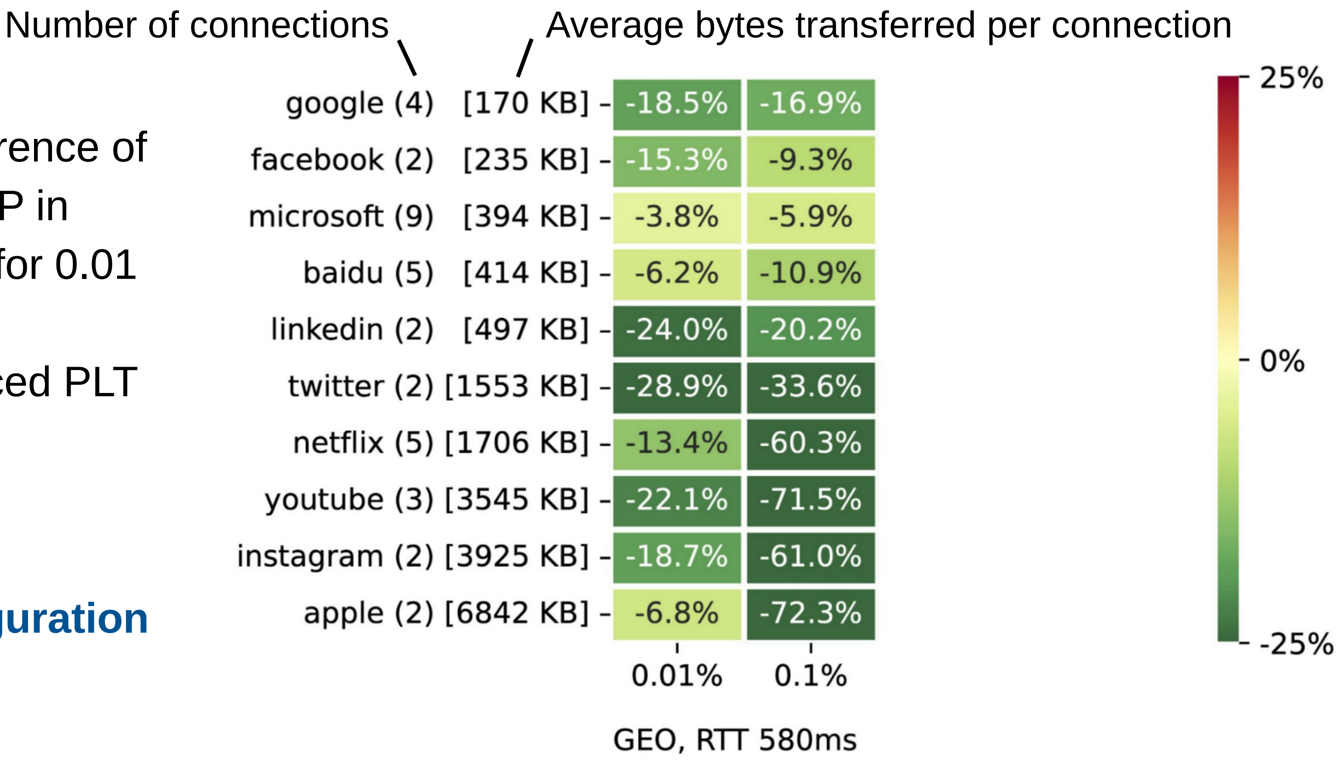
- Relative median difference of the PLT of SMAQ-PEP in comparison to QUIC for 0.01 and 0.1 % loss
- ~ 4 % to ~ 72 % reduced PLT



Case Study: Web Performance (GEO)

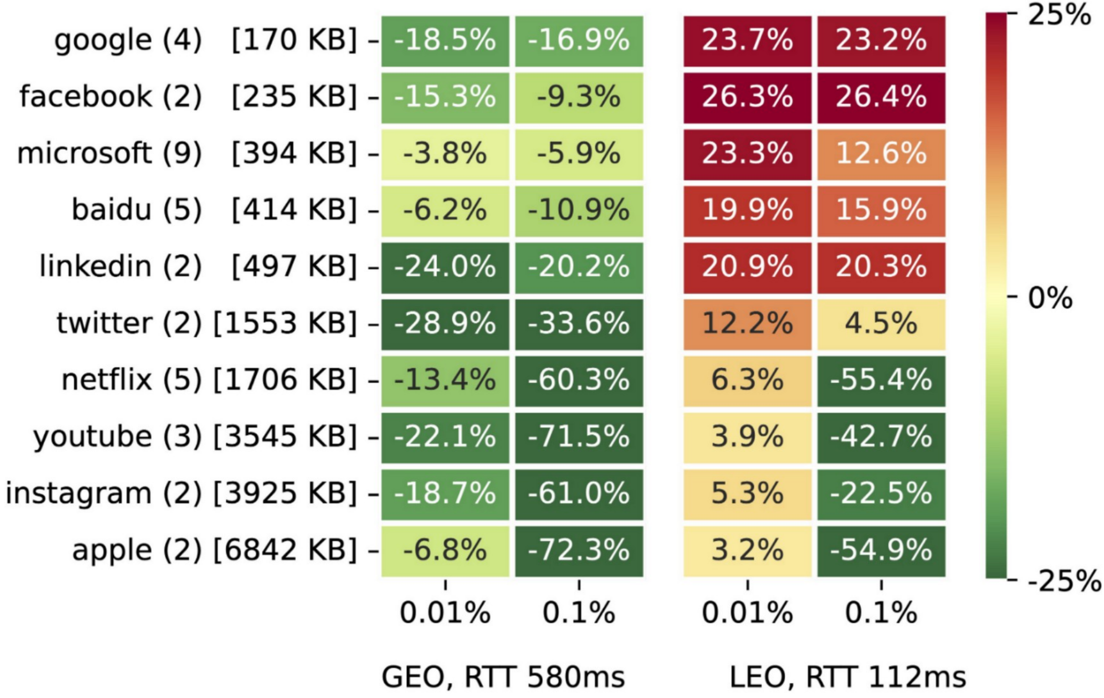
- Relative median difference of the PLT of SMAQ-PEP in comparison to QUIC for 0.01 and 0.1 % loss
- ~ 4 % to ~ 72 % reduced PLT

⇒ **PLT using SMAQ-PEP improves QUIC for every webpage and loss configuration**



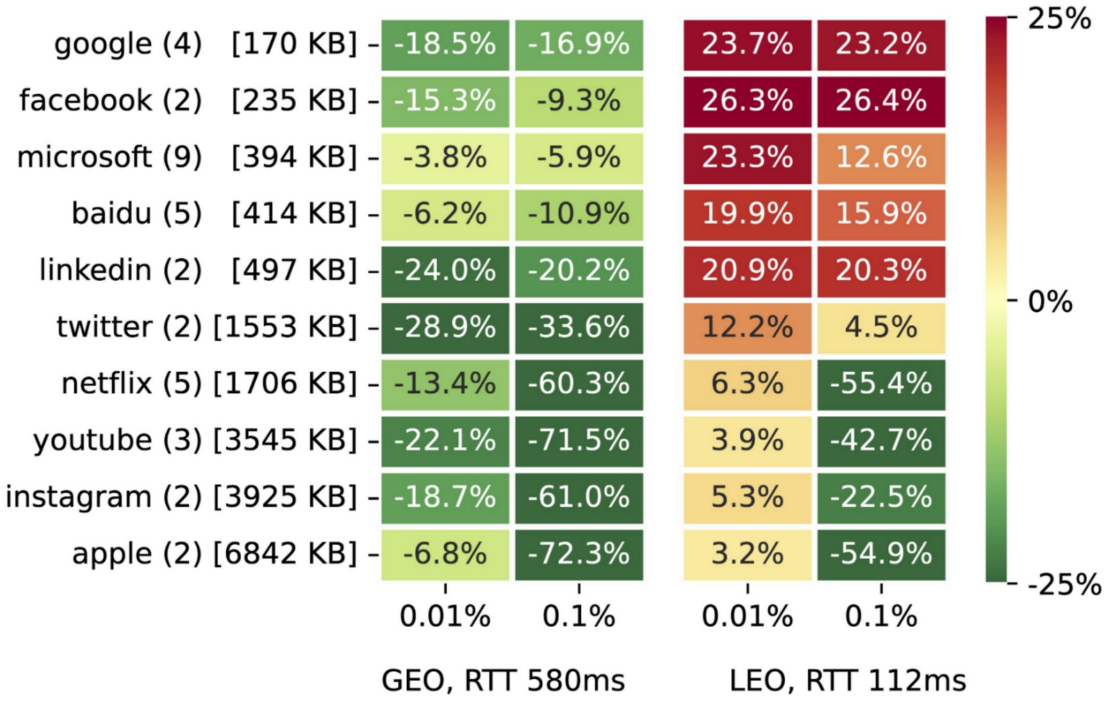
Case Study: Web Performance (LEO)

- SMAQ-PEP does prolong the page load for every webpage for 0.01 % loss
- For 0.1 % loss, SMAQ-PEP does improve over QUIC in 4 out of the 10 webpages



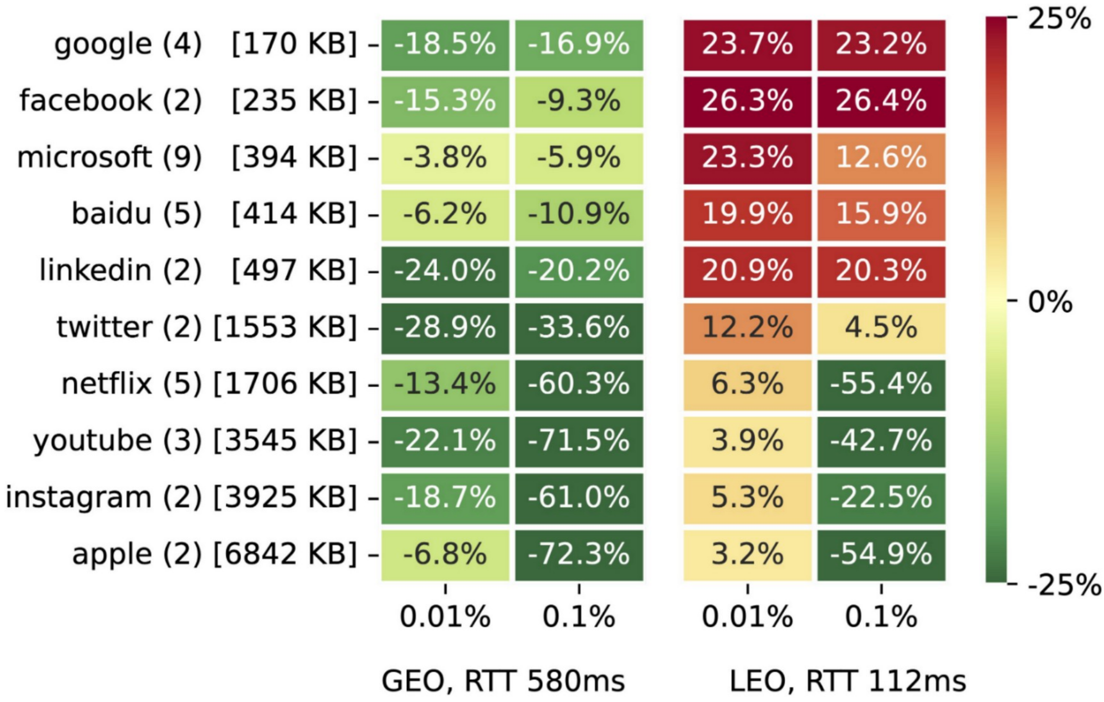
Case Study: Web Performance

- Benefits of SMAQ-PEP increase with
  - packet loss
  - latency
  - transferred bytes per connection



Case Study: Web Performance

- Benefits of SMAQ-PEP increase with
  - packet loss
  - latency
  - transferred bytes per connection



⇒ **Architecture of the webpages is the most decisive factor for the observed relative differences**

Conclusion

- Enhanced QUIC to expose selected information to middleboxes
  - in-band setup during QUIC handshake

- Evaluated the design in a distributed satellite PEP environment
- Improved end-to-end performance by splitting control loops and applying the Hybla-Westwood CCA on the satellite segment

- The higher the RTT and loss, and the more data is transferred over a connection, the higher the benefits of SMAQ-PEP

- Our findings highlight the potential of SMAQ, warranting further exploration

Future Work

- Add middleboxes at any time during the connection
- State migration with open streams

- More fine grained control about what to expose to the middlebox
- Extra protect for further application and control data

- Explore other user cases
- e.g. load balancing or live service migration

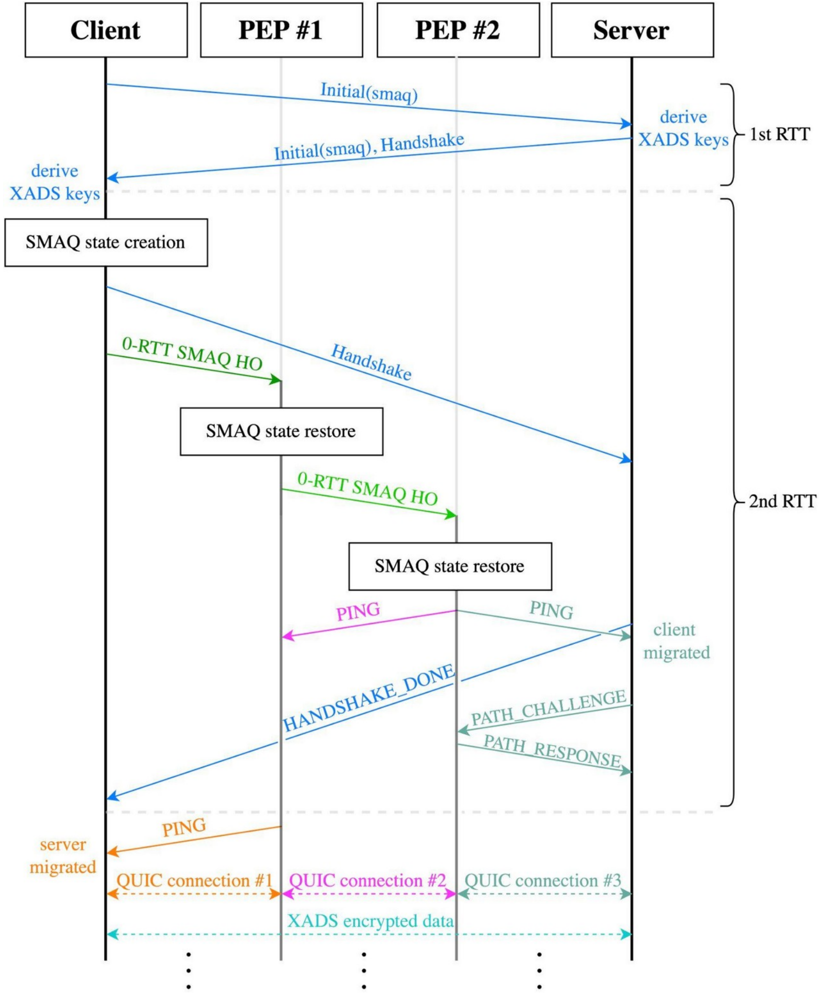
Q&A

SMAQ State

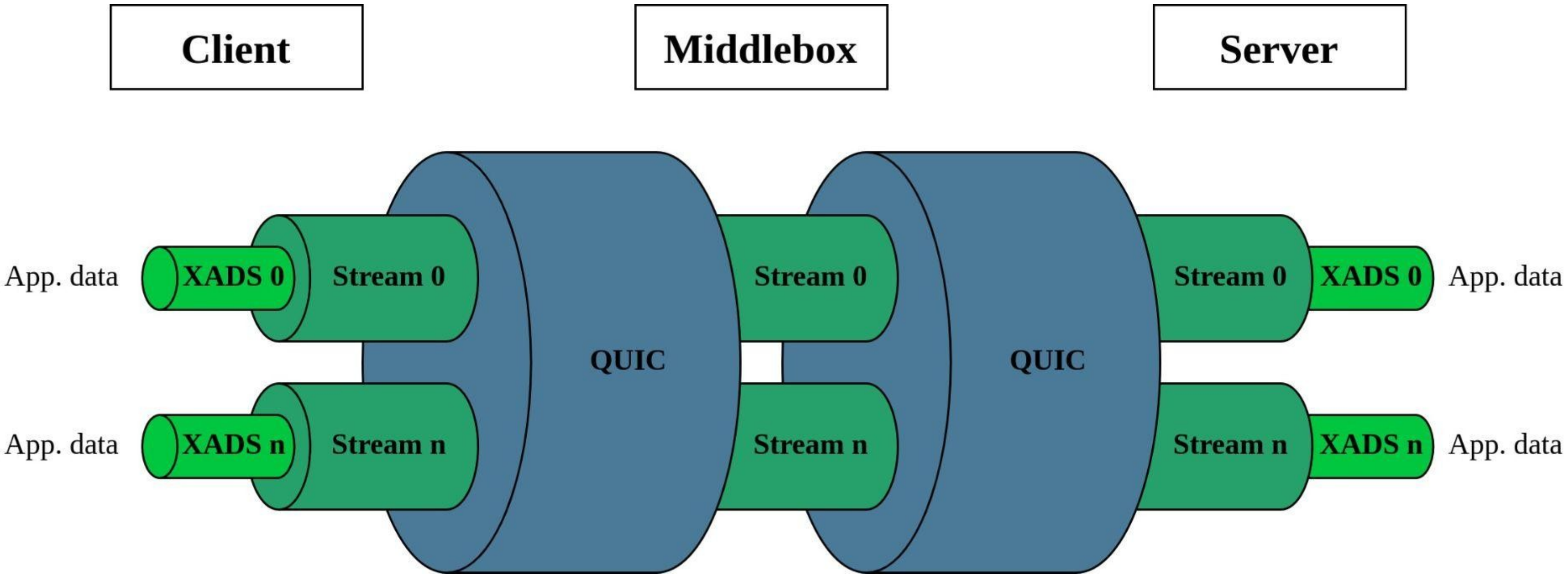

```json
{
    "ClientConnectionIDs": [
        {
            "SequenceNumber": 0,
            "ConnectionID": "",
            "StatelessResetToken": null
        }
    ],
    "ServerConnectionIDs": [
        {
            "SequenceNumber": 10,
            "ConnectionID": "uONQ8g==",
            "StatelessResetToken": "UA6xLMzDdUDqvlI9ss66dw=="
        },
        {
            "SequenceNumber": 8,
            "ConnectionID": "N0tI3Q==",
            "StatelessResetToken": "x0X0+ETTc+gEdJjiKBAcrQ=="
        },
        {
            "SequenceNumber": 9,
            "ConnectionID": "wXMRbA==",
            "StatelessResetToken": "KD7IcrIBoAJKYvNKLdgwTw=="
        }
    ],
    "Version": 1,
    "KeyPhase": 0,
    "CipherSuiteId": 4865,
    "ServerHeaderProtectionKey": "GIN6+qwud88+vBkWRfF41Q==",
    "ClientHeaderProtectionKey": "d64BtZ86HettbKjnYFBCPg==",
    "ServerTrafficSecret": "QHyGPstGzvRCXJ7XQbTWPP91+OcFTukWuqODGya0bLo=",
    "ClientTrafficSecret": "1NDdqomNiE5BPSAlFXGcrhU76oChDQhRDTgcjE1goXA=",
    "ServerAddress": "10.0.0.1:18080",
    "ClientAddress": "10.0.0.3:35746",
    "ClientTransportParameters": "BQSACAAABgSACAAABwSACAAABASADAAACAJAZAkCQGQBBIAAdTADAkWsCwEaDgEEDwAgAkSw",
    "ServerTransportParameters": "BQSACAAABgSACAAABwSACAAABASADAAACAJAZAkCQGQBBIAAdTADAkWsCwEaAhCbGVJVvOAIanAf92Om7FwFABLAsIi5Gkod2+WXpc66f4QNbOwOAQQPBI8fY50gAkSw",
    "ClientHighestSentPacketNumber": 74186,
    "ServerHighestSentPacketNumber": 78670,
}
```


Concise, serialized state containing only the essential pieces to restore the connection on another machine

Distributed PEP Connection Setup



XADS Encapsulation



PLT GEO

orbit	loss	page	median PEP PLT (s)	median QUIC PLT (s)
geo	0.01	apple.com	20.172	21.649
geo	0.01	baidu.com	8.576	9.142
geo	0.01	facebook.com	4.448	5.255
geo	0.01	google.com	4.528	5.554
geo	0.01	instagram.com	8.961	11.025
geo	0.01	linkedin.com	4.728	6.223
geo	0.01	microsoft.com	10.927	11.355
geo	0.01	netflix.com	8.936	10.318
geo	0.01	twitter.com	5.684	7.997
geo	0.01	youtube.com	9.551	12.254
geo	0.10	apple.com	21.538	77.694
geo	0.10	baidu.com	8.731	9.800
geo	0.10	facebook.com	4.774	5.261
geo	0.10	google.com	4.727	5.687
geo	0.10	instagram.com	10.222	26.207
geo	0.10	linkedin.com	5.145	6.447
geo	0.10	microsoft.com	11.409	12.124
geo	0.10	netflix.com	10.005	25.186
geo	0.10	twitter.com	6.214	9.352
geo	0.10	youtube.com	11.129	38.988

PLT LEO

orbit	loss	page	median PEP PLT (s)	median QUIC PLT (s)
leo	0.01	apple.com	8.581	8.315
leo	0.01	baidu.com	2.602	2.171
leo	0.01	facebook.com	1.414	1.120
leo	0.01	google.com	1.491	1.206
leo	0.01	instagram.com	4.840	4.596
leo	0.01	linkedin.com	1.698	1.405
leo	0.01	microsoft.com	3.421	2.776
leo	0.01	netflix.com	5.058	4.756
leo	0.01	twitter.com	2.653	2.365
leo	0.01	youtube.com	6.058	5.832
leo	0.10	apple.com	9.082	20.157
leo	0.10	baidu.com	2.640	2.279
leo	0.10	facebook.com	1.421	1.124
leo	0.10	google.com	1.498	1.216
leo	0.10	instagram.com	5.016	6.470
leo	0.10	linkedin.com	1.704	1.417
leo	0.10	microsoft.com	3.406	3.026
leo	0.10	netflix.com	5.168	11.580
leo	0.10	twitter.com	2.693	2.578
leo	0.10	youtube.com	6.350	11.083