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

QUIC Measurements on Ark

Nikolas Gauder

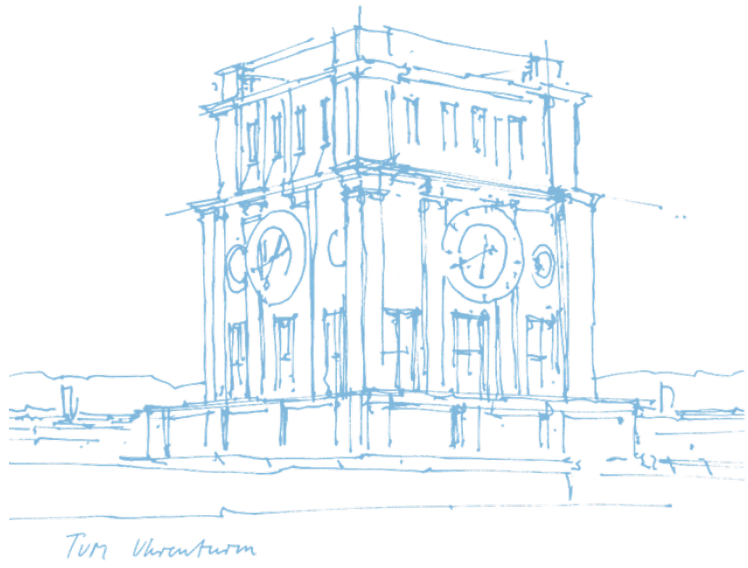
Johannes Zirngibl

Marcel Kempf

Johannes Späth

Tuesday 24th February, 2026

AIMS-19



QUIC Scanning and Its Challenges

- QUIC [1] is a transport protocol with growing adoption
  - Major web services (Google, Meta, etc.) are using QUIC
  - HTTP/3 [2] is built on QUIC
  - Compelling target for in-depth measurements

- Encrypted-by-default challenges
  - **Limited passive measurement opportunities**
  - Active measurements are essential to study QUIC behavior and its global deployment

- QScanner [3] is a feature-rich tool for large-scale QUIC scans
  - However, measurements are **limited to a single vantage point (VP)**
  - Restricts visibility in scenarios like geo-distributed CDN deployments or geo-blocking
  - Need globally distributed infrastructure $\rightarrow$ **Archipelago (Ark)** [4]

---

[1] Jana Iyengar and Martin Thomson. QUIC: A UDP-Based Multiplexed and Secure Transport. RFC 9000. 2021
[2] Mike Bishop. HTTP/3. RFC 9114. 2022
[3] TUM Chair of Network Architectures and Services. QScanner. https://github.com/tumi8/QScanner. 2022
[4] Archipelago (Ark) Measurement Infrastructure. https://www.caida.org/projects/ark/. Accessed: 2026-02-03

Bringing QUIC Measurements to Ark

- Goal: Deploy **QUIC scanning in the field-tested Ark infrastructure**
  - Transferring our proven approach and insights from QScanner
  - Unlock new opportunities to study QUIC's behavior
  - In **diverse geographic and network conditions**

+

Choosing a QUIC Stack: Design Trade-offs

- QUIC runs **entirely in user space**
  - No kernel implementation is currently available, but work is in progress[1]
  - A QUIC library must be chosen and integrated
- Avoid integrating a **full QUIC stack**
  - QUIC is complex: Flow & congestion control, loss recovery, ...
  - Rapidly evolving ecosystem
- Design choice: **Minimal integration overhead**
  - **OpenSSL $\geq$ 3.2** provides native QUIC client support (since Nov. 2023)
  - Interoperable with all major QUIC implementations[2]
  - Commonly available on most systems
  - Scamper already depends on OpenSSL
  - ➔ **No new external dependencies**

[1] https://lwn.net/Articles/1029851

[2] https://github.com/openssl/openssl/actions/workflows/run_quic_interop.yml

QUIC Integration in Scamper

- New Scamper command `quic`
  - Inspired by Scamper's `http` command
  - Similar structure and handling of connections and library calls

- Implemented **Python bindings**
  - Like Scamper does for other measurement primitives
  - Enables easy integration into existing Python-based analysis pipelines

- Perform a QUIC handshake with targets and **capture all QUIC packets**
  - Together with TLS keys for decryption
  - Results saved in **warts**[3] files
  - Can be converted to PCAPs using `sc_warts2pcap`
  - Analysis separated from measurement process
  - ➔ **Future compatibility**

[3] https://www.caida.org/catalog/software/scamper/man/warts.5.pdf

Scamper QUIC Command

```
./scamper -c “quic -d 443 -h ’example.com’ -a h3 -O insecure” -i 23.215.0.136
```

Usage string: `quic`

`[-d dport]`

- Destination port

`[-h host]`

- Set as Server Name Indication (SNI) and used for certificate verification

`[-a alpn-protos]`

- Comma-separated list of Application-Layer Protocol Negotiation (ALPN) names
- ALPN must be specified for QUIC handshakes according to RFC 9001 [5]

`[-c ciphersuites]`, `[-g groups]`, `[-s sigalgs]`

- Lists of TLS 1.3 cipher suites, groups, and signature algorithms to offer during the handshake
- Allows testing server behavior with different configurations, e.g., PQC algorithms

`[-O option]`

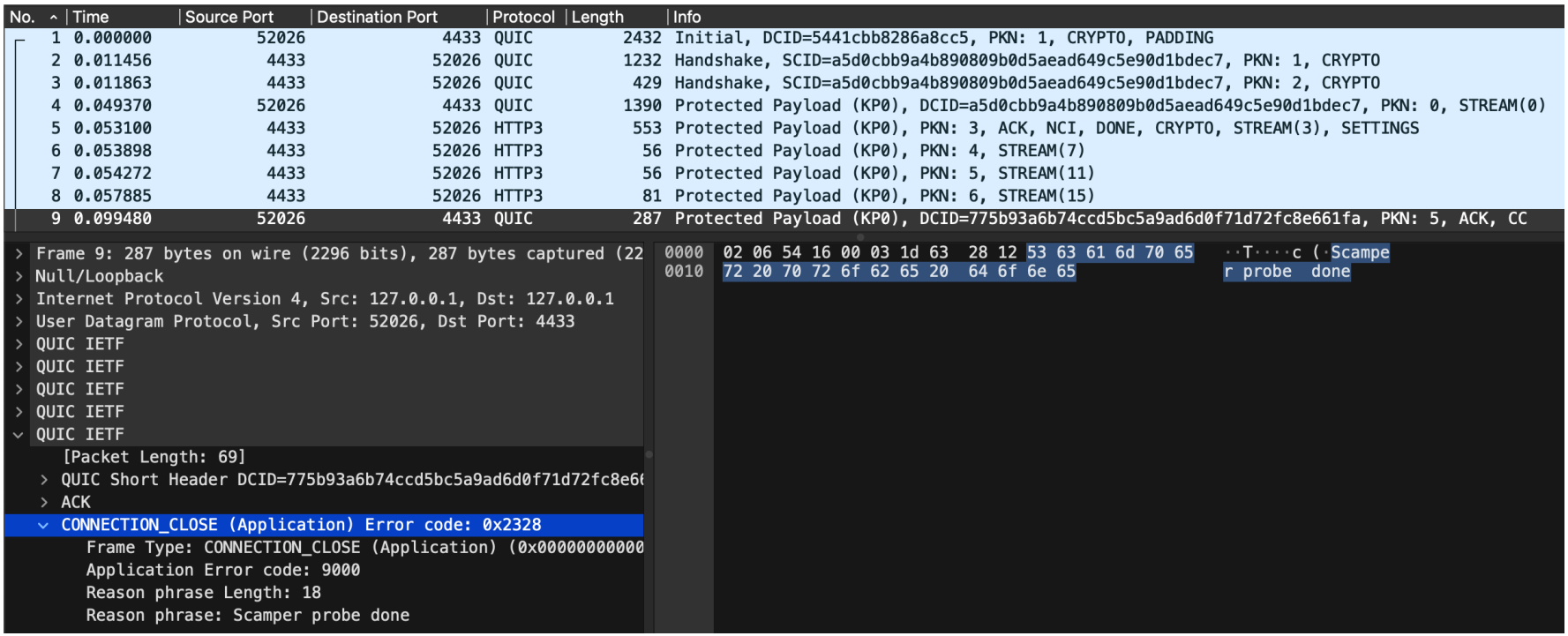
- Currently only `insecure` is implemented: disables certificate verification
- More options can be added in the future, e.g., QUIC transport parameters

[5] Martin Thomson and Sean Turner. Using TLS to Secure QUIC. RFC 9001. 2021

Scamper QUIC Measurement Packet Exchange

PCAP of a Scamper QUIC measurement against Cloudflare quiche on port 4433

- `CONNECTION_CLOSE` frame with reason phrase sent



Challenges Encountered & Lessons Learned

- Many systems have an **old OpenSSL version** installed
  - QUIC support was only added in OpenSSL 3.2 (Nov. 2023)
  - Must ensure that the target system has OpenSSL $\geq$ 3.2 installed
  - Debian Trixie ships with OpenSSL 3.5
  - ➔ **Test across multiple target distributions early** to uncover dependency mismatches

- QUIC transport parameters are currently **not configurable** in OpenSSL's QUIC stack
  - Restricts our ability to evaluate server behavior under varying transport parameter settings
  - A pull request has been submitted to OpenSSL[4]
  - Changes are expected to be merged in the immediate future
  - ➔ **Contribute to open-source projects** to close feature gaps and accelerate development

[4] https://github.com/openssl/openssl/pull/29664

Early Findings: Distributed QUIC Scans

- **Post-Quantum Cryptography**
  - Detection of PQ/hybrid key exchange (KE) groups in TLS 1.3
  - Measurable deployment share: ∼41 % support rate
  - ➔ Consistent across VPs

Library	ASes	PQ KE Support
Akamai	AKAMAI-AS, AKAMAI-ASN1	✗
LSQUIC	mixed	✗
MsQuic	mixed	✗
nginx	mixed	○
quic-go	mixed	○
Cloudflare quiche	CLOUDFLARE	✓
Google QUICHE	GOOGLE, GOOGLE-CLOUD-PLATFORM	✓[5]
mvfst	FACEBOOK	✓[6]
quicly	FASTLY	✓
s2n-quic	AMAZON-02	✓

[5] PQC support mainly in *GOOGLE* AS and not widely deployed in *GOOGLE-CLOUD-PLATFORM* AS

[6] We found 1 off-net deployment without PQC support

Why Did You Listen to This?

- We want to make QUIC measurements **easily accessible** to the research community!
  - QUIC-enabled Scamper currently deployed on test VPs: DE, NZ, US, UK, FR, PL, JP
  - Status: Merge into main Scamper codebase — *work in progress*
  - Available to the public soon!

- Currently working on my Master's thesis: **Analyzing QUIC deployments on the Internet**
  - **TLS features:** Cipher suites, certificates, PQC support
  - **QUIC features:** Deployed QUIC libraries, transport parameters
  - **Network features:** Handshake latency, anycast deployments

Which QUIC measurements would be most useful to you?

Which scanning features would add the most value for the community?

Bibliography

[1] Jana Iyengar and Martin Thomson. QUIC: A UDP-Based Multiplexed and Secure Transport. RFC 9000. 2021.

[2] Mike Bishop. HTTP/3. RFC 9114. 2022.

[3] TUM Chair of Network Architectures and Services. QScanner. https://github.com/tumi8/QScanner. 2022.

[4] Archipelago (Ark) Measurement Infrastructure. https://www.caida.org/projects/ark/. Accessed: 2026-02-03.

[5] Martin Thomson and Sean Turner. Using TLS to Secure QUIC. RFC 9001. 2021.