DEMO: Debugging QUIC with qlog and QUICvis

Robin Marx – Jonas Reynders – Kevin Pittevils – Peter Quax – Wim Lamotte
QUIC and HTTP/3

- **Very** complex
  - Congestion control, flow control, handshake, 0-RTT, migration, ...

- Everything is re-implemented from scratch, so
- There will be:
  - Bugs
  - Suboptimal performance
  - Incomplete implementations
  - Consciously differing implementation choices and trade-offs
QUIC and HTTP/3

- **Very** complex
  - Congestion control, flow control, handshake, 0-RTT, migration, ...

- Many people will be looking into the behavior
  - Initial implementations + conformance testing (current stage)
  - Early and at-scale deployments
  - Academic research (and teaching!)

- Cycle starts over with new features in v2
  - multipath, FEC, unreliability, ...
QUIC timeline
QUIC timeline: comparing traces

- Transmitted packets (TX)
- Received packets (RX)
- Frames per stream (TX + RX)

Frames:
- STREAM (data)
- MAX_STREAM_DATA, MAX_DATA, BLOCKED (flow control)
- ACK (congestion control)
- PATH_CHALLENGE, PATH_RESPONSE (security)
QUIC sequence diagram

- Client + Server logs
  - Exact latency
    - Flight + processing!
QUIC sequence diagram

- Client + Server logs
  - Exact latency
    - Flight + processing!
  - Many extra goodies

- Re-ordering

- Loss

- Retransmits
QUIC Flow and congestion control diagram

10% packet loss

Stream: 4 8 12 16 20
- : MAX_STREAM_DATA (stream level)
I : Data segment (STREAM frame)

data (bytes)

0 4K 8K 12K 16K 20K

0 135 270 405 540 675 810 945 1080 1215 1350

time (ms)

- : MAX_DATA (connection level)
- : Congestion window (CWND)
* : Data allowance (CWND - bytes in flight)

L

H

UHASSELT EDM
QUIC logging: qlog

```json
{
    "connectionid": "0x763f8eaf61aa3ffe84270c0644bddd20b0d", "starttime": 1543917600,
    "fields":
        ["time", "category", "type", "trigger", "data"],
    "events": [
        [50, "TLS", "0RTT_KEY", "PACKET_RX", {"key": ...}],
        [51, "HTTP", "STREAM_OPEN", "PUSH", {"id": 0, "headers": ...}],
        ...
        [200, "TRANSPORT", "PACKET_RX", "STREAM", {"nr": 50, "contents": "GET /ping.html", ...}],
        [201, "HTTP", "STREAM_OPEN", "GET", {"id": 16, "headers": ...}],
        [201, "TRANSPORT", "STREAMFRAME_NEW", "PACKET_RX", {"id": 16, "contents": "pong", ...}],
        ...
        [250, "TRANSPORT", "ACK_NEW", "PACKET_RX", {"nr": 51, "acked": 60, ...}],
        [251, "RECOVERY", "CWND_UPDATE", "ACK_NEW", {"nr": 51, "cwnd": 20780, ...}],
        ...
        [1001, "RECOVERY", "LOSS_DETected", "ACK_NEW", {"nr": a, "frames": ...}],
        [2002, "RECOVERY", "PACKET_NEW", "EARLY_RETRAN\", {"nr": x, "frames": ...}],
        [3003, "RECOVERY", "PACKET_NEW", "TAIL\_LOSS\_PROBE", {"nr": y, "frames": ...}],
        [4004, "RECOVERY", "PACKET_NEW", "TIMEOUT", {"nr": z, "frames": ...}]
    ]
}
```
QUIC logging: qlog

```json
{
  "connectionid": "0x763f8eaf61aa3ffe84270c0644b6bd2b0d", "starttime": 1543917600,
  "fields": {
    "time", "category", "type",
  },
  "events": [
    [50, "TLS", "0RTT_KEY", "STREAM_OPEN"],
    [51, "HTTP", "STREAM_OPEN"],
    ...
    [200, "TRANSPORT", "PACKET_RX", "STREAM"],
    [201, "HTTP", "PACKET_RX"],
    [201, "TRANSPORT", "PACKET_RX", "GET"],
    [201, "TRANSPORT", "PACKET_RX", "PACKET_RX"],
    [201, "TRANSPORT", "PACKET_RX", "CWND_EXCEEDED"],
    [201, "TRANSPORT", "ACK_NEW"],
    [250, "TRANSPORT", "ACK_NEW"],
    [251, "RECOVERY", "ACK_NEW"],
    [252, "TRANSPORT", "ACK_NEW"],
    ...
    [1001, "RECOVERY", "LOSS_DETECTED"],
    [1002, "RECOVERY", "PACKET_RX"],
    [1003, "RECOVERY", "PACKET_RX"],
    [1004, "RECOVERY", "PACKET_RX"],
    [1004, "RECOVERY", "PACKET_RX"],
    "trigger",
    "data",
    "PACKET_RX",
    "PUSH",
    "STREAM",
    "GET",
    "PACKET_RX",
    "PACKET_RX",
    "CWND_EXCEEDED",
    "PACKET_RX",
    "ACK_NEW",
    "CWND_UPDATE",
    "ACK_NEW",
    "EARLY_RETRANS",
    "TAILLOSS_PROBE",
    "TIMEOUT",
    "key": ...],
    "id": 0, "headers": ...],
    "nr": 50, "contents": "GET /ping.html",...
    "id": 16, "headers": ...],
    "nr": 67, "frames": [16, ...], ...],
    "nr": 67, "cwnd": 14600, ...],
    "nr": 51, "acked": 60, ...],
    "nr": 51, "cwnd": 20780, ...],
    "nr": 67, "frames": [16, ...], ...],
    "nr": a, "frames": ...],
    "nr": x, "frames": ...],
    "nr": y, "frames": ...],
    "nr": z, "frames": ...]
}
QUIC logging: standardized

- **Easy to access**
  - [https://example.com/.well-known/h3/state](https://example.com/.well-known/h3/state) (this connection)
  - [https://example.com/.well-known/h3/state/list](https://example.com/.well-known/h3/state/list) (list of all connections)

- chrome://net-internals/h3/state/{connID}
- about:networking/h3/state/list

- WebPageTest.org
  - Simply fetch server-log after test is done (vs needing to let browser do it)
  - Get browser log via devtools integration
QUIC logging: standardized

- Easy and secure to access
  - `/h3/state/{connID}?token=53CR3T`
    - Server config file
    - Passed as QUIC transport parameter?

- Disable logging of sensitive info
  - Only congestion info, no packet contents, keys, ...
  - Interesting for live deployments

- Encrypt logs themselves
  - If attacker obtains logs, cannot access

- Make it non-trivial to enable (by accident)
  - Sensible defaults
QUIC debugging: logging + visualizations

(Full) Factorial tests

Store
Process
Aggregate

Visualize
Analyse

Share