DEMO: Debugging QUIC with qlog and QUICvis

Robin Marx – Jonas Reynders – Kevin Pittevils – Peter Quax – Wim Lamotte

UHASSELT EDM

https://quic.edm.uhasselt.be
EPIQ workshop – December 2018

QUIC and HTTP/3

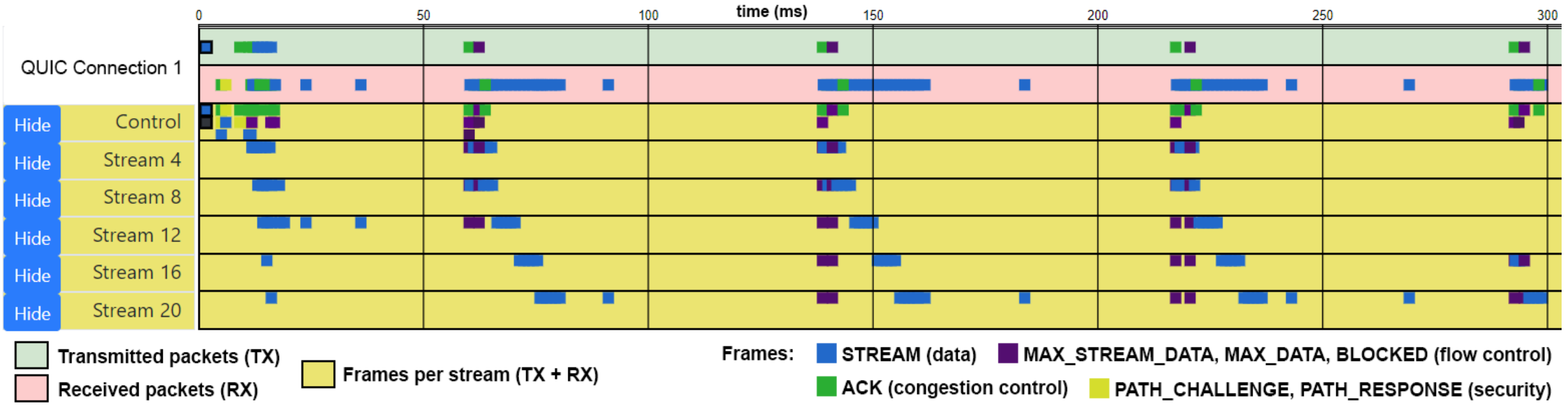
- **Very** complex
  - Congestion control, flow control, handshake, 0-RTT, migration, …

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

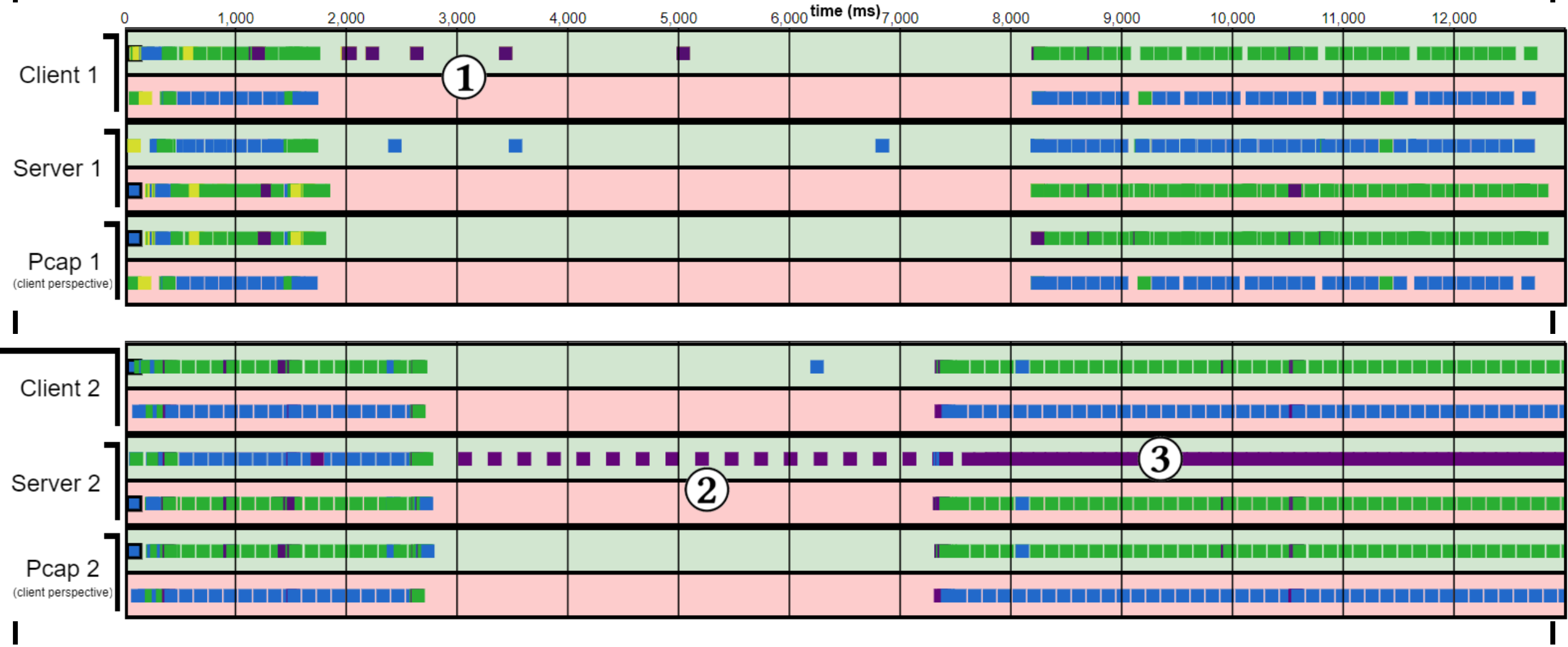
QUIC and HTTP/3

- **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

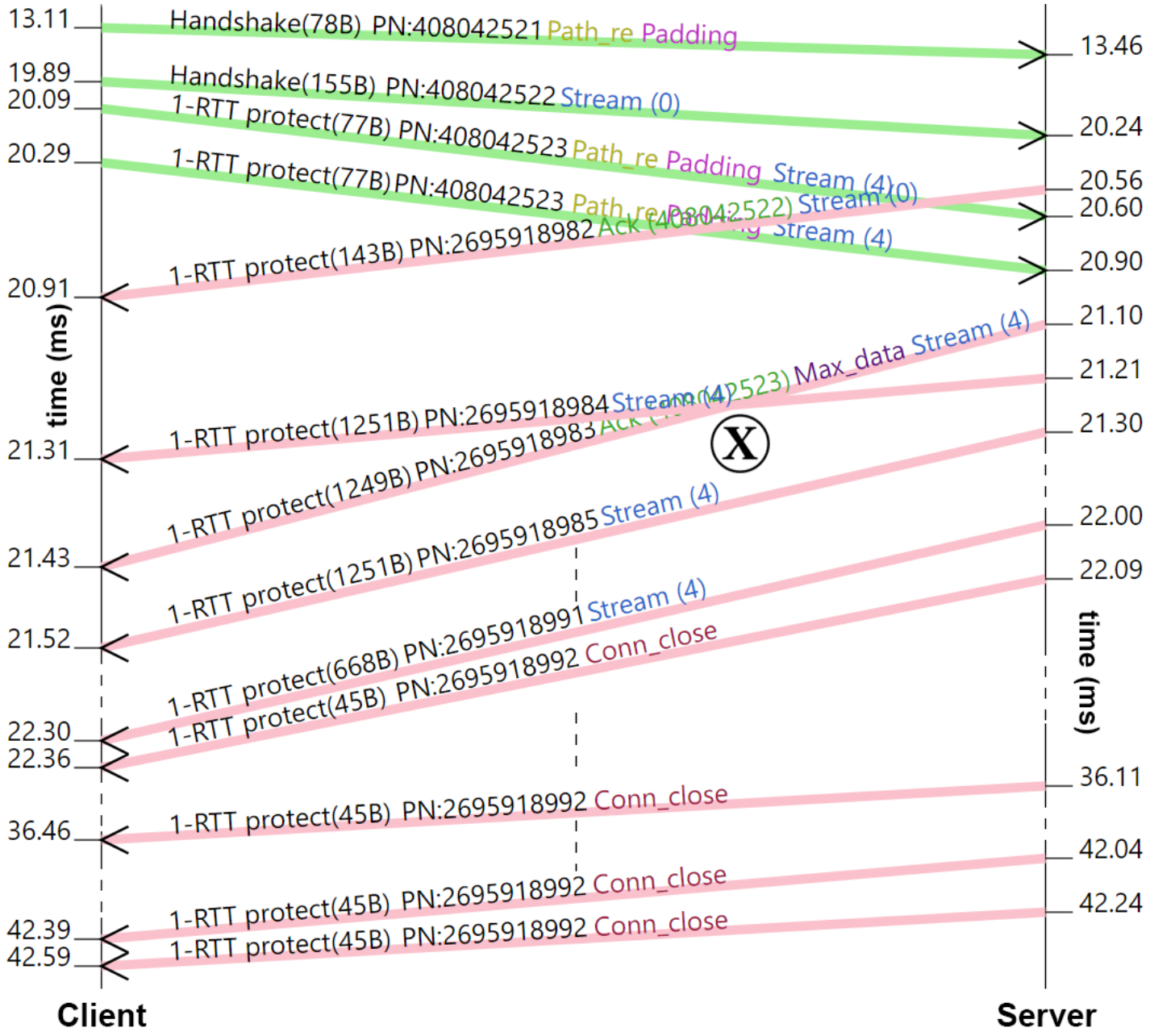


QUIC timeline : comparing traces



QUIC sequence diagram

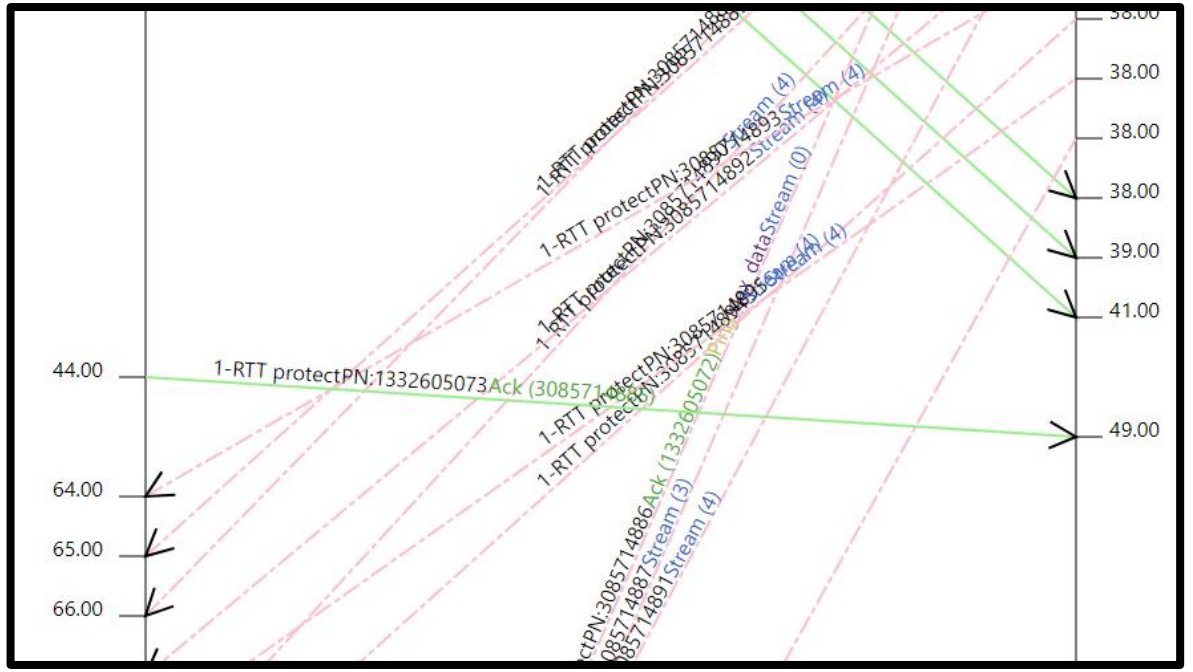
- Client + Server logs
  - Exact latency
    - Flight + processing!



QUIC sequence diagram

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

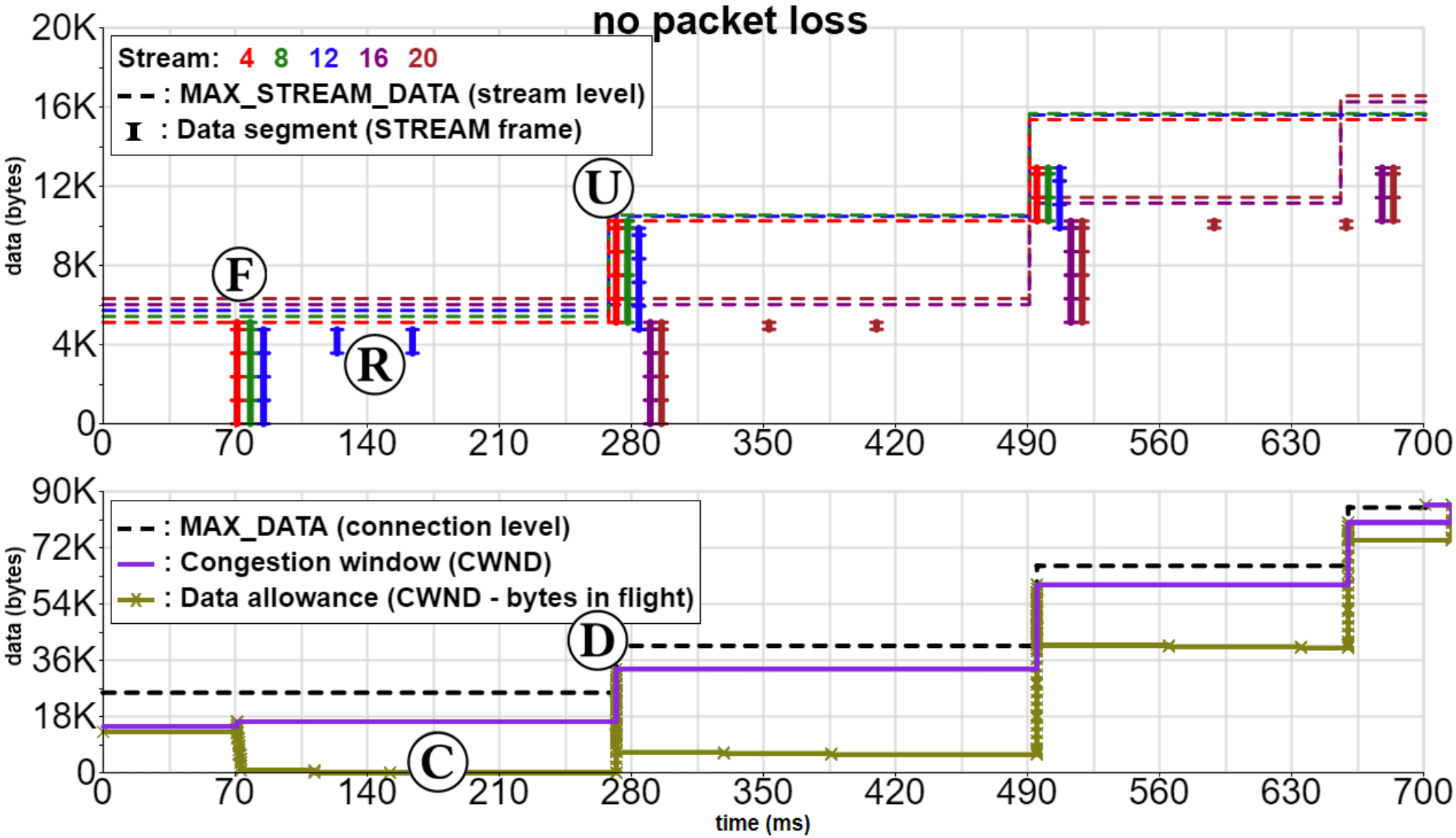
Re-ordering



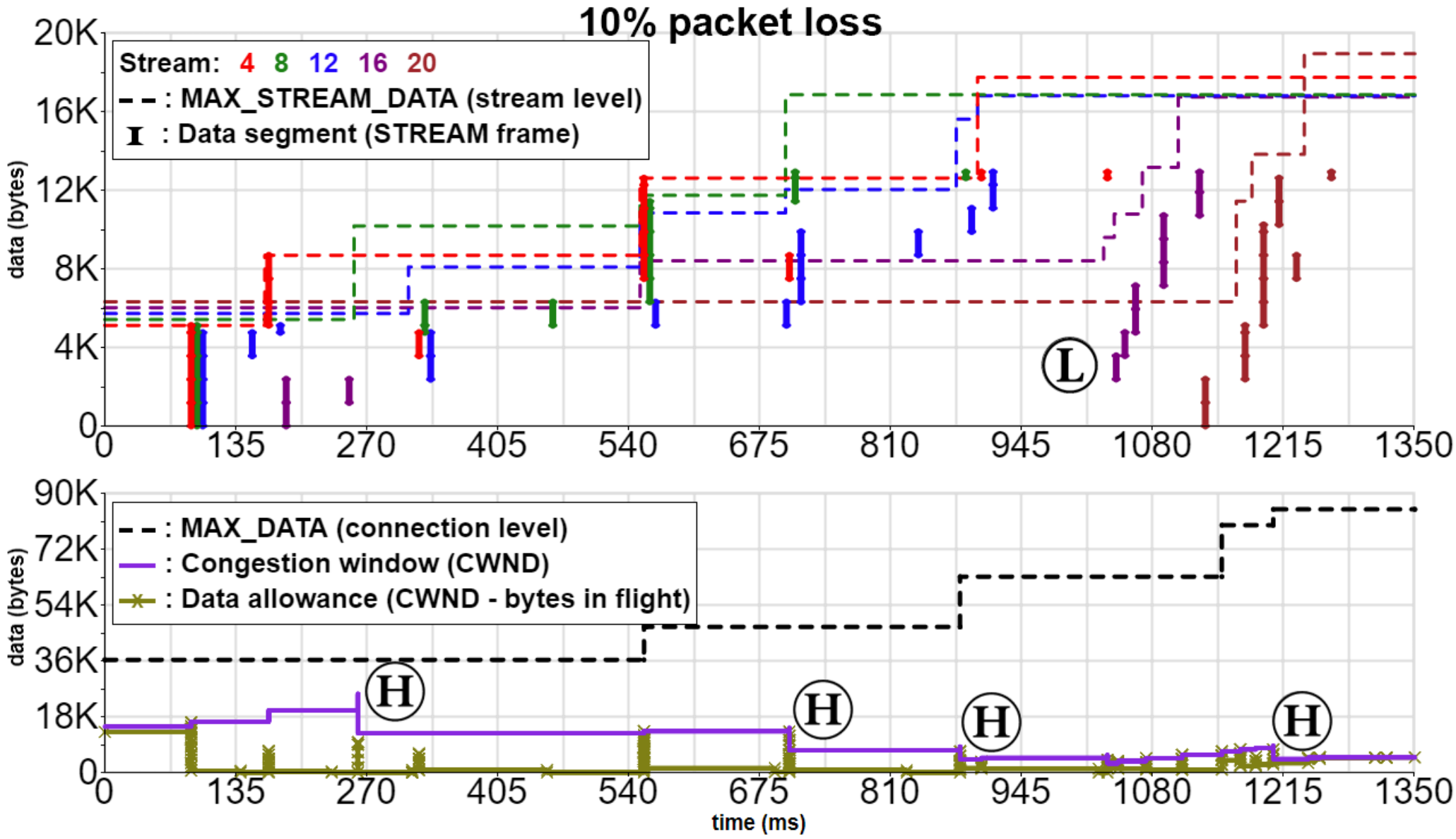
Loss

Retransmits

QUIC Flow and congestion control diagram



QUIC Flow and congestion control diagram



QUIC logging: qlog

```
{"connectionid": "0x763f8eaf61aa3ffe84270c0644bdbd2b0d", "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", ...}],
    [201,   "TRANSPORT", "PACKET_NEW",      "PACKET_RX",       {"nr": 67, "frames": [16, ...], ...}],
    [203,   "RECOVERY",  "PACKET_QUEUED",   "CWND_EXCEEDED",   {"nr": 67, "cwnd": 14600, ...}],
    [250,   "TRANSPORT", "ACK_NEW",         "PACKET_RX",       {"nr": 51, "acked": 60, ...}],
    [251,   "RECOVERY",  "CWND_UPDATE",     "ACK_NEW",         {"nr": 51, "cwnd": 20780, ...}],
    [252,   "TRANSPORT", "PACKET_TX",       "CWND_UPDATE",     {"nr": 67, "frames": [16, ...], ...}],
    ...
    [1001,  "RECOVERY",  "LOSS_DETECTED",   "ACK_NEW",         {"nr": a, "frames": ...}],
    [2002,  "RECOVERY",  "PACKET_NEW",      "EARLY_RETRANS",   {"nr": x, "frames": ...}],
    [3003,  "RECOVERY",  "PACKET_NEW",      "TAIL_LOSS_PROBE", {"nr": y, "frames": ...}],
    [4004,  "RECOVERY",  "PACKET_NEW",      "TIMEOUT",         {"nr": z, "frames": ...}]
 ]}
```

QUIC logging: qlog

```
{"connectionid": "0x763f8eaf61aa3ffe84270c0644bdbd2b0d", "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", ...}],
    [201,   "TRANSPORT", "PACKET_NEW",      "PACKET_RX",        {"nr": 67, "frames": [16, ...], ...}],
    [203,   "RECOVERY",  "PACKET_QUEUED",   "CWND_EXCEEDED",    {"nr": 67, "cwnd": 14600, ...}],
    [250,   "TRANSPORT", "ACK_NEW",         "PACKET_RX",        {"nr": 51, "acked": 60, ...}],
    [251,   "RECOVERY",  "CWND_UPDATE",     "ACK_NEW",          {"nr": 51, "cwnd": 20780, ...}],
    [252,   "TRANSPORT", "PACKET_TX",       "CWND_UPDATE",      {"nr": 67, "frames": [16, ...], ...}],
    ...
    [1001,  "RECOVERY",  "LOSS_DETECTED",   "ACK_NEW",          {"nr": a, "frames": ...}],
    [2002,  "RECOVERY",  "PACKET_NEW",      "EARLY_RETRANS",    {"nr": x, "frames": ...}],
    [3003,  "RECOVERY",  "PACKET_NEW",      "TAIL_LOSS_PROBE",  {"nr": y, "frames": ...}],
    [4004,  "RECOVERY",  "PACKET_NEW",      "TIMEOUT",          {"nr": z, "frames": ...}]
 ]}
```

QUIC logging: standardized

- Easy to access
  - https://example.com/.well-known/h3/state (this connection)
  - https://example.com/.well-known/h3/state/{connID} (other connection)
  - 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