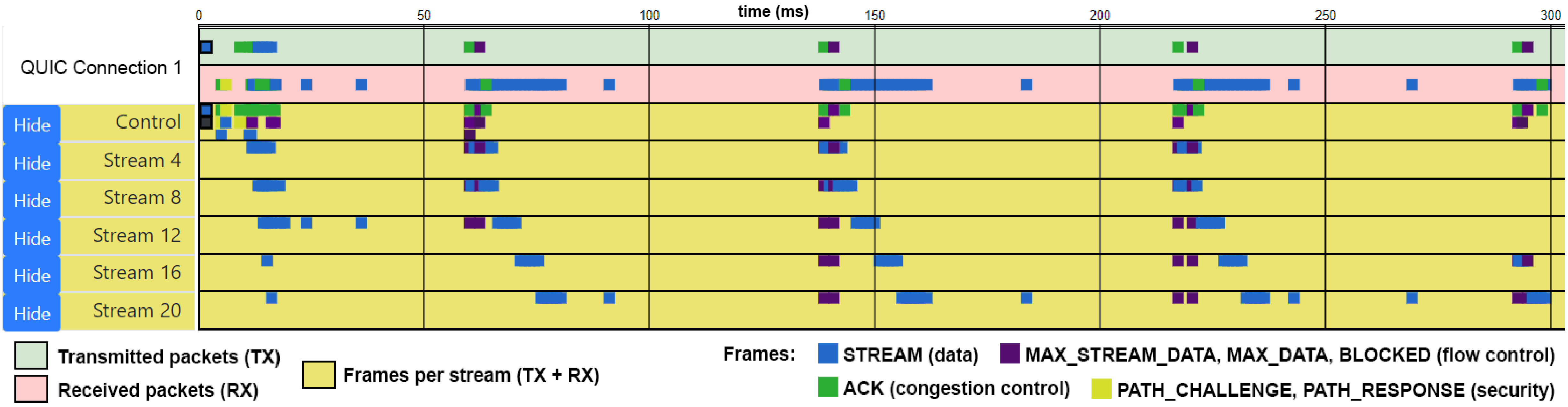
QUIC Logging, debugging and tooling

Robin Marx

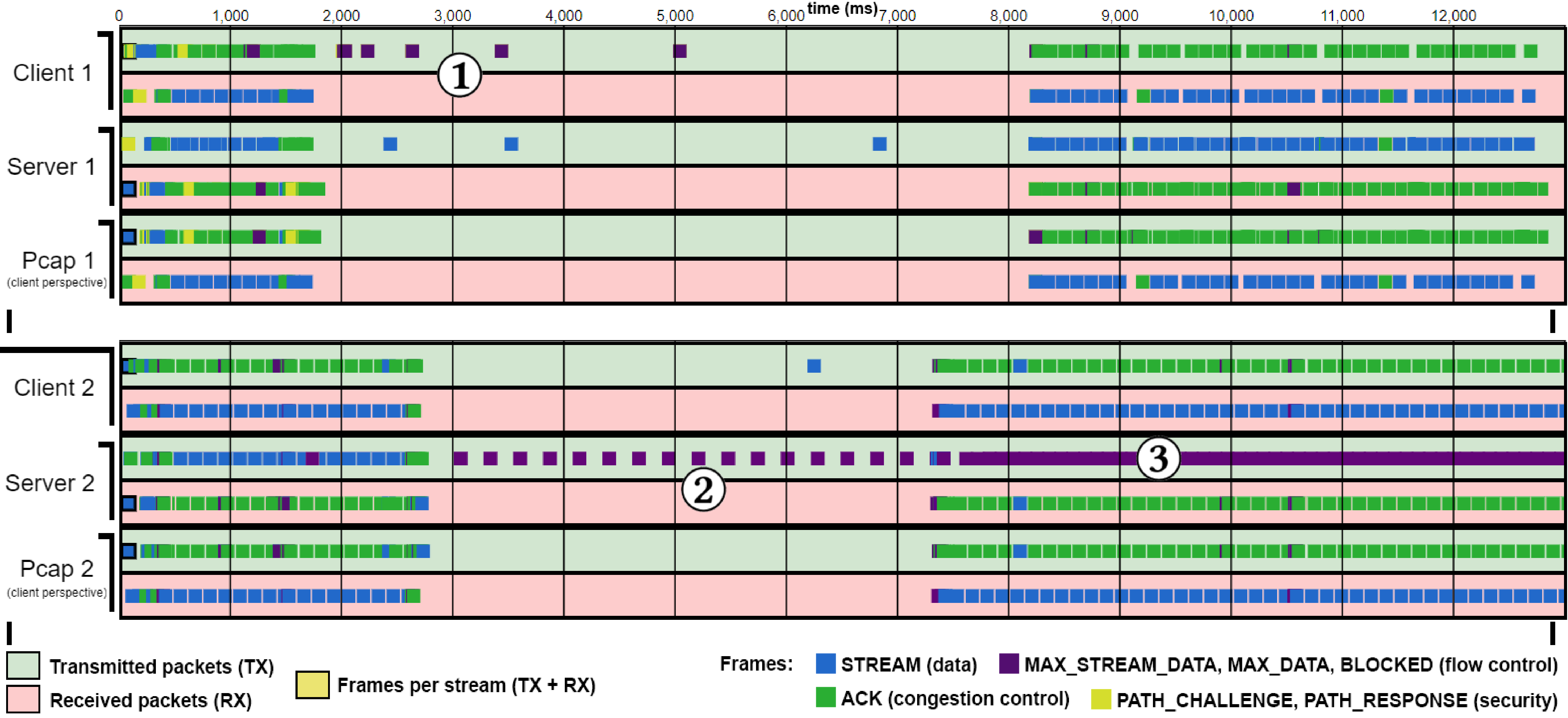
▶▶ UHASSELT EDM

https://quic.edm.uhasselt.be
IETF interim Tokyo – January 2019

QUIC timeline : split per-stream

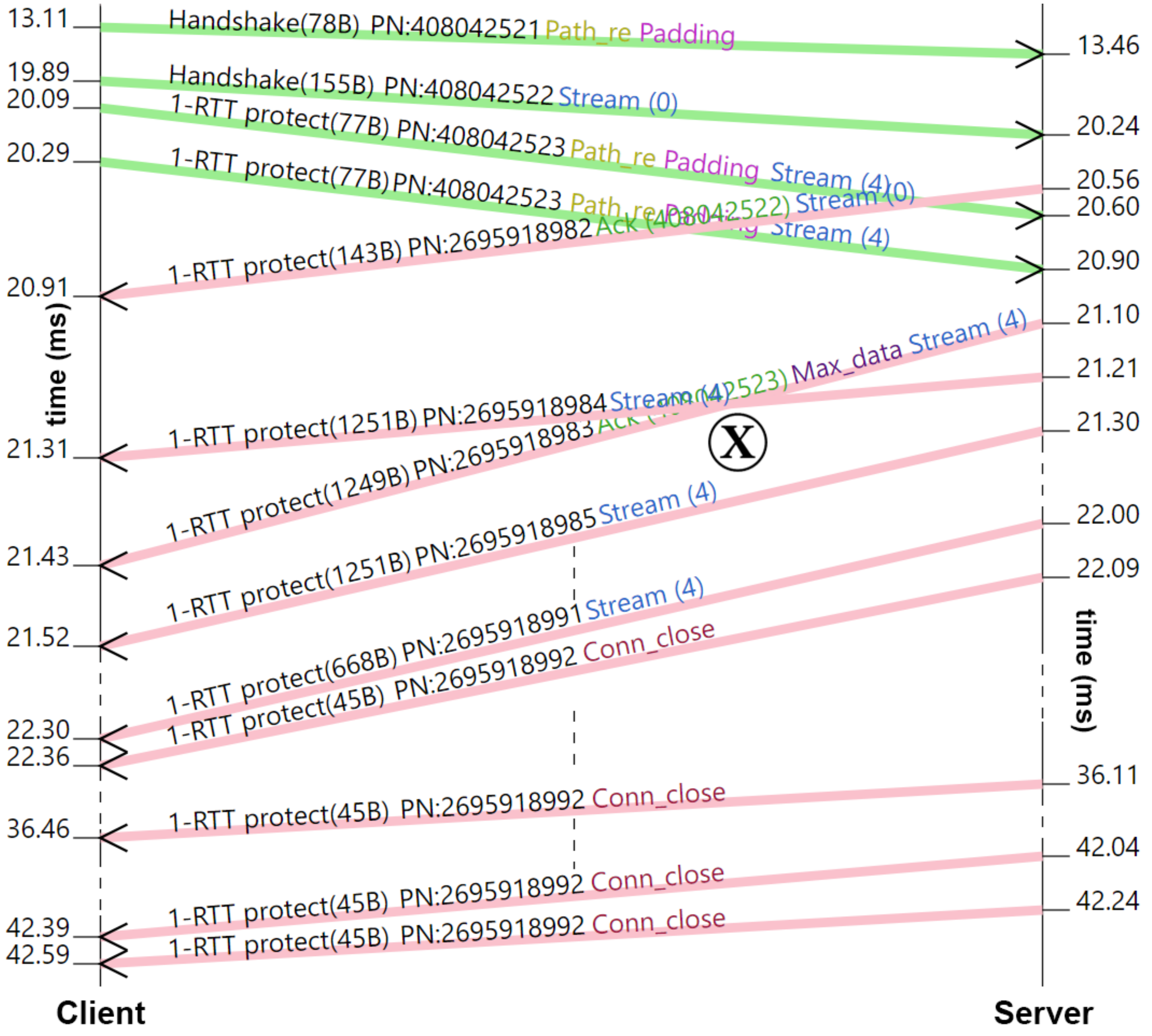


QUIC timeline : compare connections : network drop



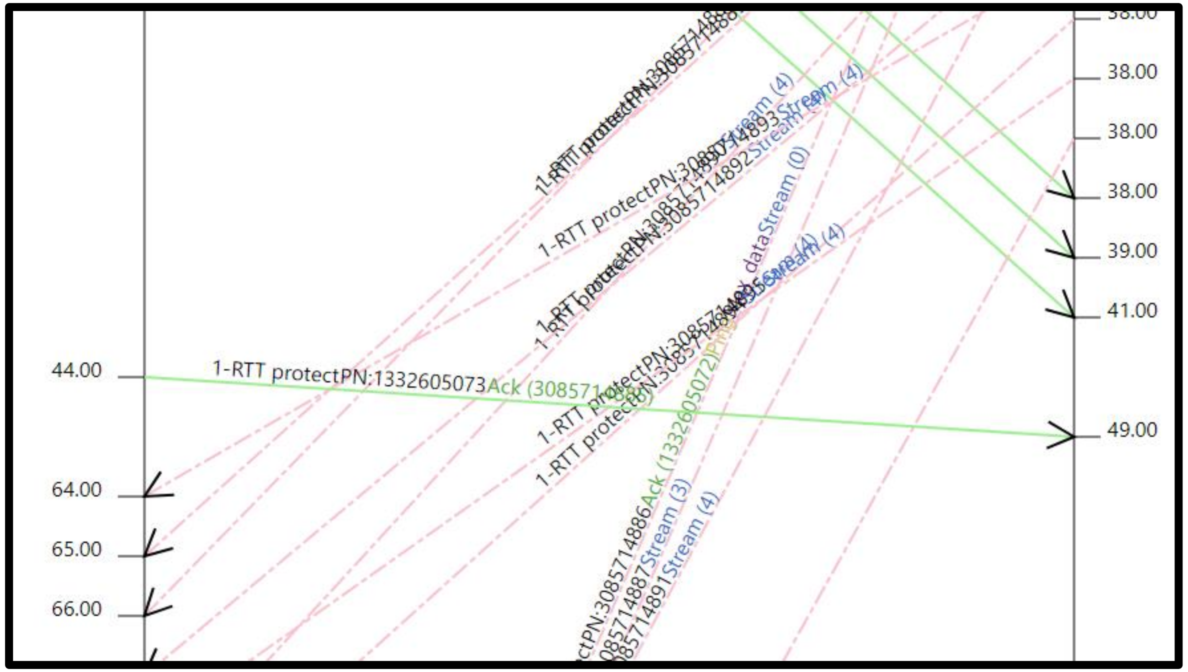
▶▶ UHASSELT EDM

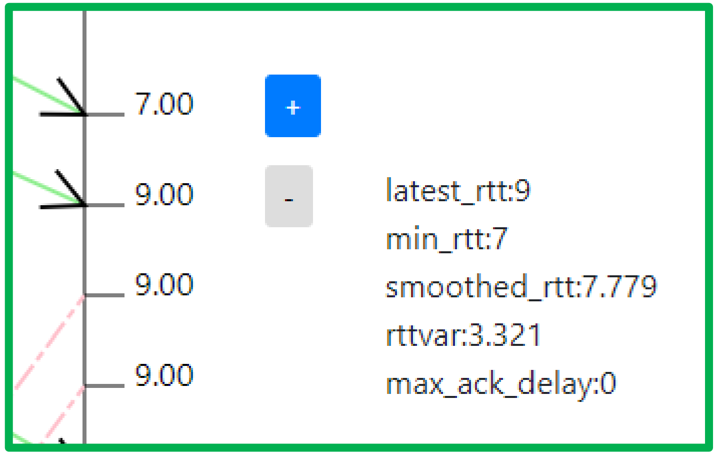
QUIC sequence diagram


13.11
Handshake(78B) PN:408042521 Path_re Padding
13.46
19.89
Handshake(155B) PN:408042522 Stream (0)
20.24
20.09
1-RTT protect(77B) PN:408042523 Path_re Padding Stream (4)
20.60
20.29
1-RTT protect(77B) PN:408042523 Path_re Padding Stream (4)
20.90
20.56
1-RTT protect(143B) PN:2695918982 Ack (408042522) Stream (0)
20.91
21.10
1-RTT protect(1249B) PN:2695918983 Ack (4...2523) Max_data Stream (4)
21.43
21.21
1-RTT protect(1251B) PN:2695918984 Stream (4)
21.31
X
21.30
1-RTT protect(1251B) PN:2695918985 Stream (4)
21.52
22.00
1-RTT protect(668B) PN:2695918991 Stream (4)
22.30
22.09
1-RTT protect(45B) PN:2695918992 Conn_close
22.36
36.11
1-RTT protect(45B) PN:2695918992 Conn_close
36.46
42.04
1-RTT protect(45B) PN:2695918992 Conn_close
42.39
42.24
1-RTT protect(45B) PN:2695918992 Conn_close
42.59
time (ms)
time (ms)
Client
Server

QUIC sequence diagram

Re-ordering

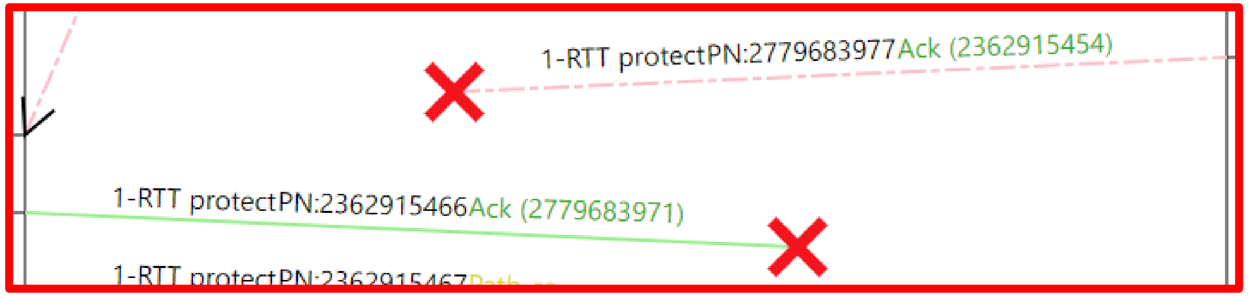




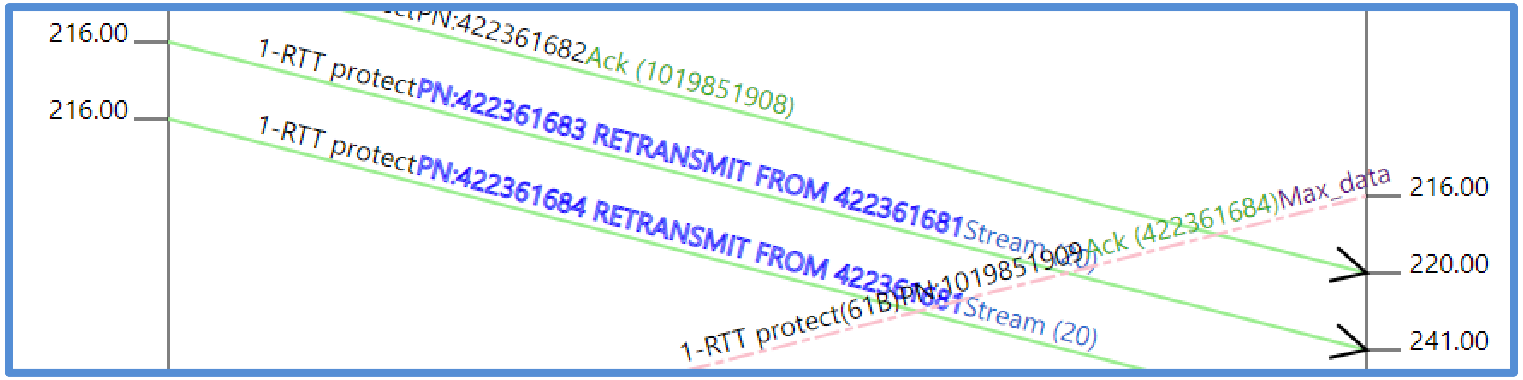
latest_rtt:9
min_rtt:7
smoothed_rtt:7.779
rttvar:3.321
max_ack_delay:0

RTT estimates

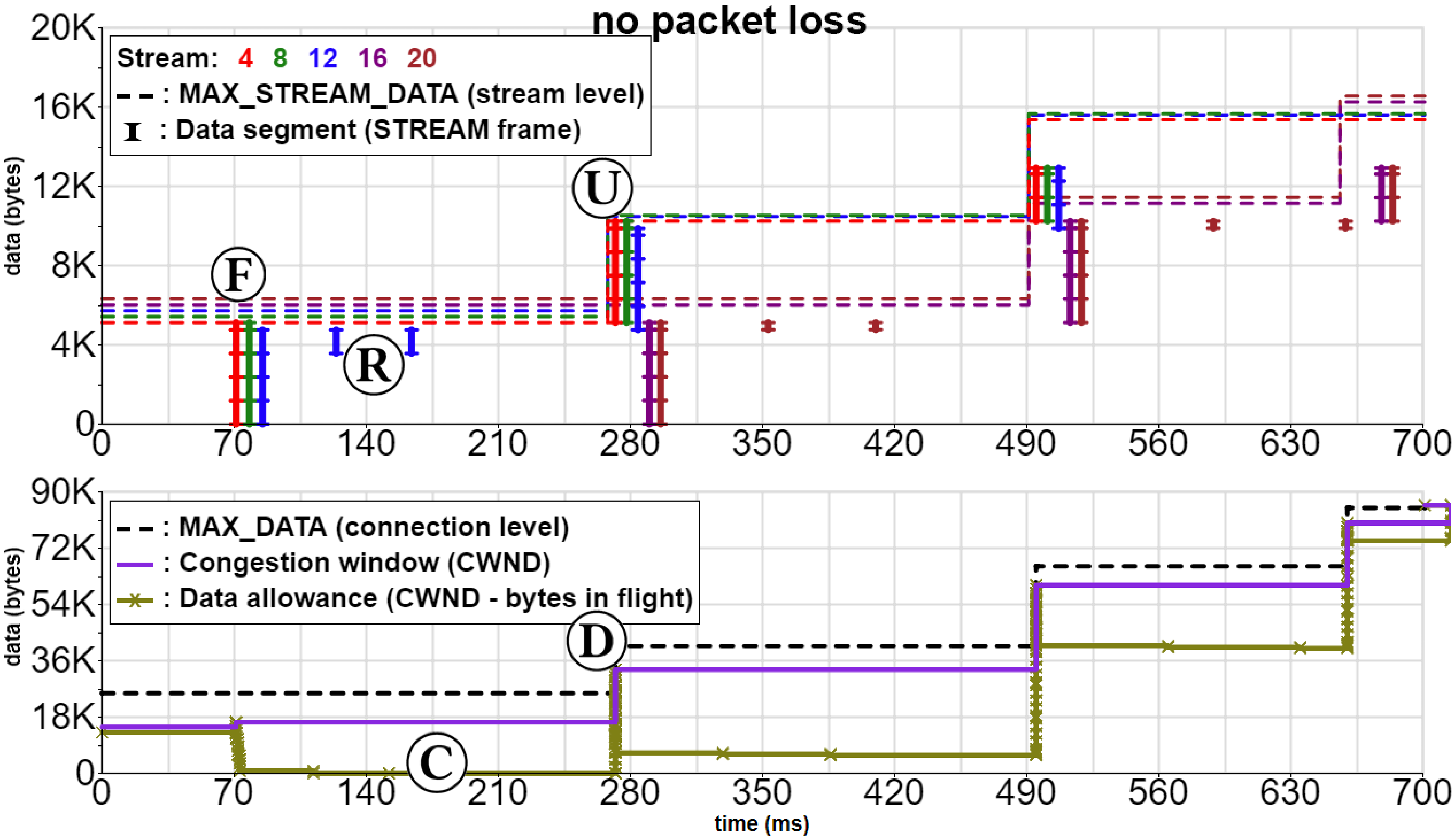
Loss


1-RTT protectPN:2779683977Ack (2362915454)
1-RTT protectPN:2362915466Ack (2779683971)
1-RTT protectPN:2362915467Path

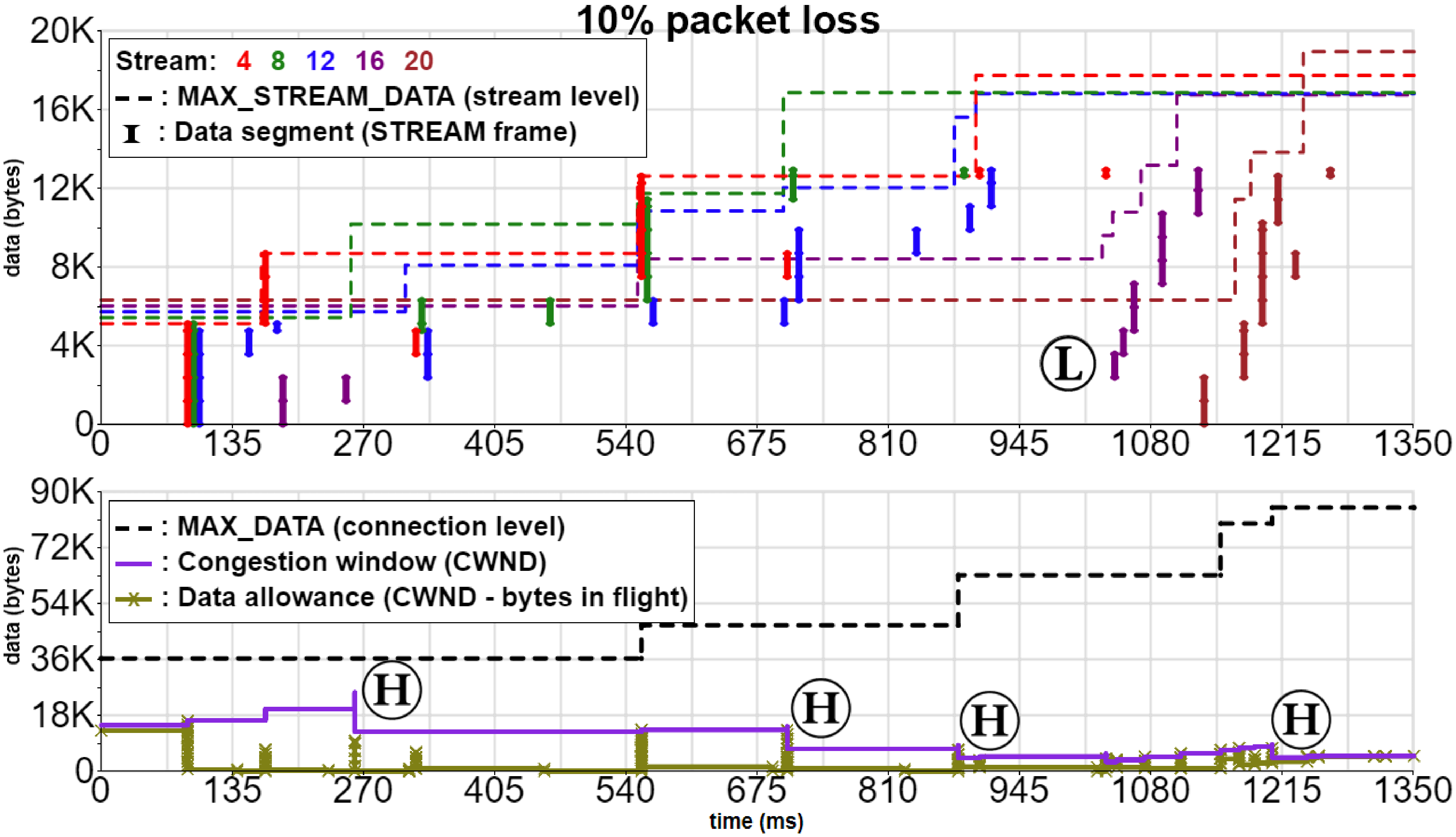
Retransmits


216.00
216.00
ctPN:422361682Ack (1019851908)
1-RTT protectPN:422361683 RETRANSMIT FROM 422361681Stream (20)
1-RTT protectPN:422361684 RETRANSMIT FROM 422361681Stream (20)
1-RTT protect(61B)PN:1019851909Ack (422361684)Max_data
216.00
220.00
241.00

QUIC flow and congestion control

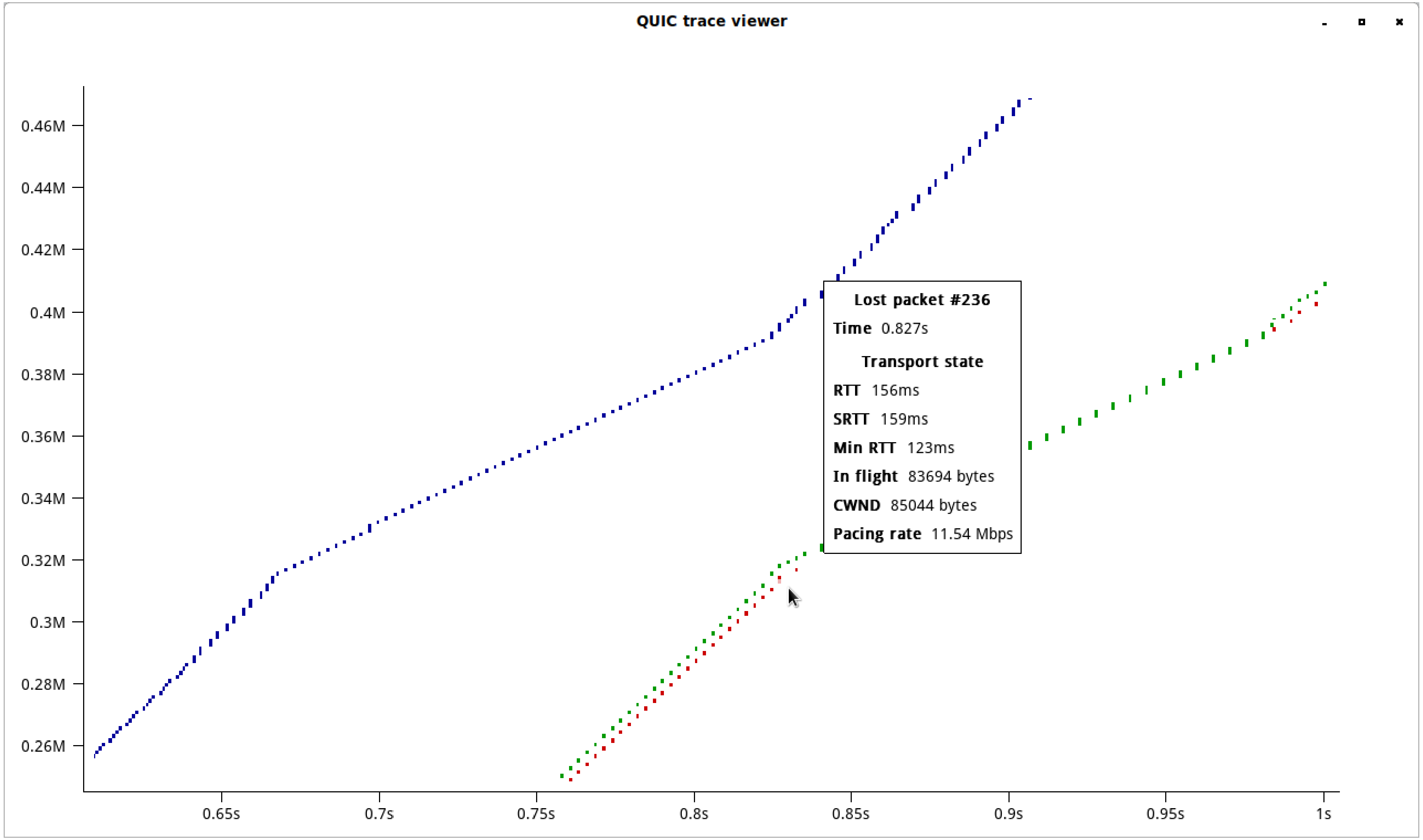


QUIC flow and congestion control



▸▸ UHASSELT EDM

QUIC congestion control : quic-trace tool (Google)



Lost packet #236

Time 0.827s

Transport state

RTT 156ms

SRTT 159ms

Min RTT 123ms

In flight 83694 bytes

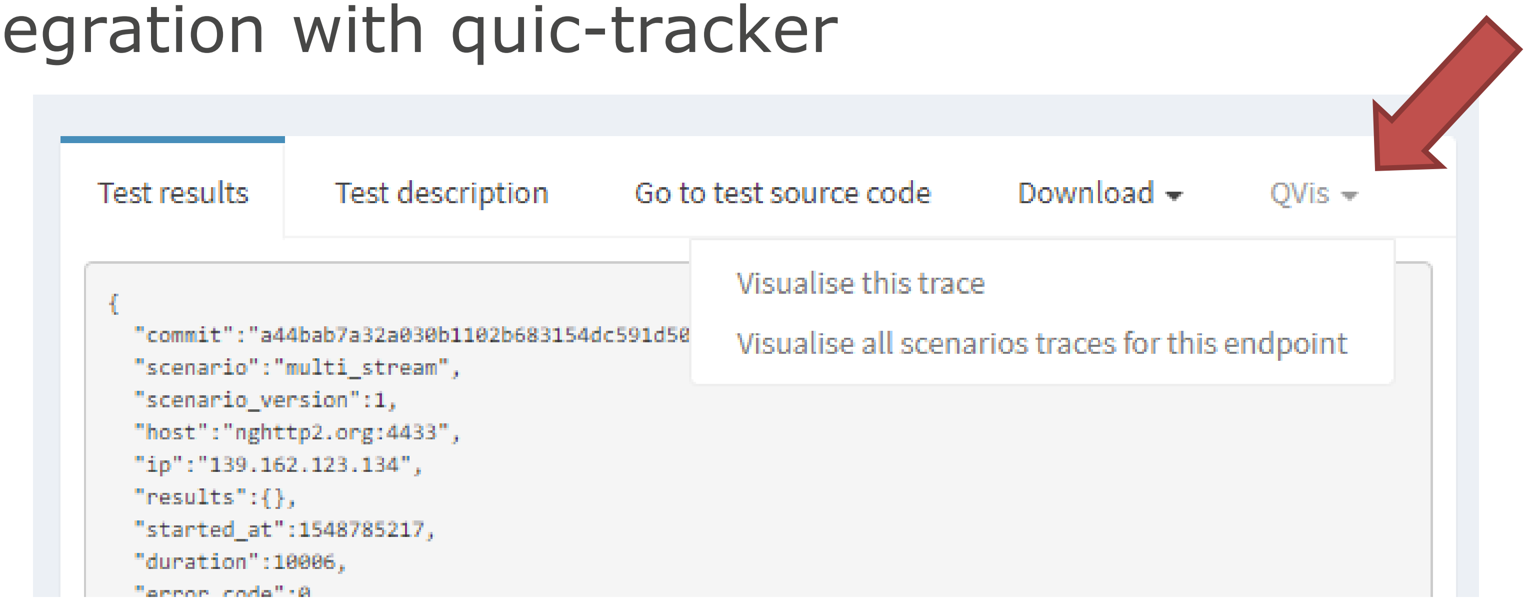
CWND 85044 bytes

Pacing rate 11.54 Mbps

https://github.com/google/quic-trace

QUICvis

- POCs and demos (hardcoded data) available at
  - https://quic.edm.uhasselt.be
- Being reworked into a more flexible framework
  - Something usable in coming weeks
  - Prime-time ready by Prague (end of March)
  - V1: integration with quic-tracker

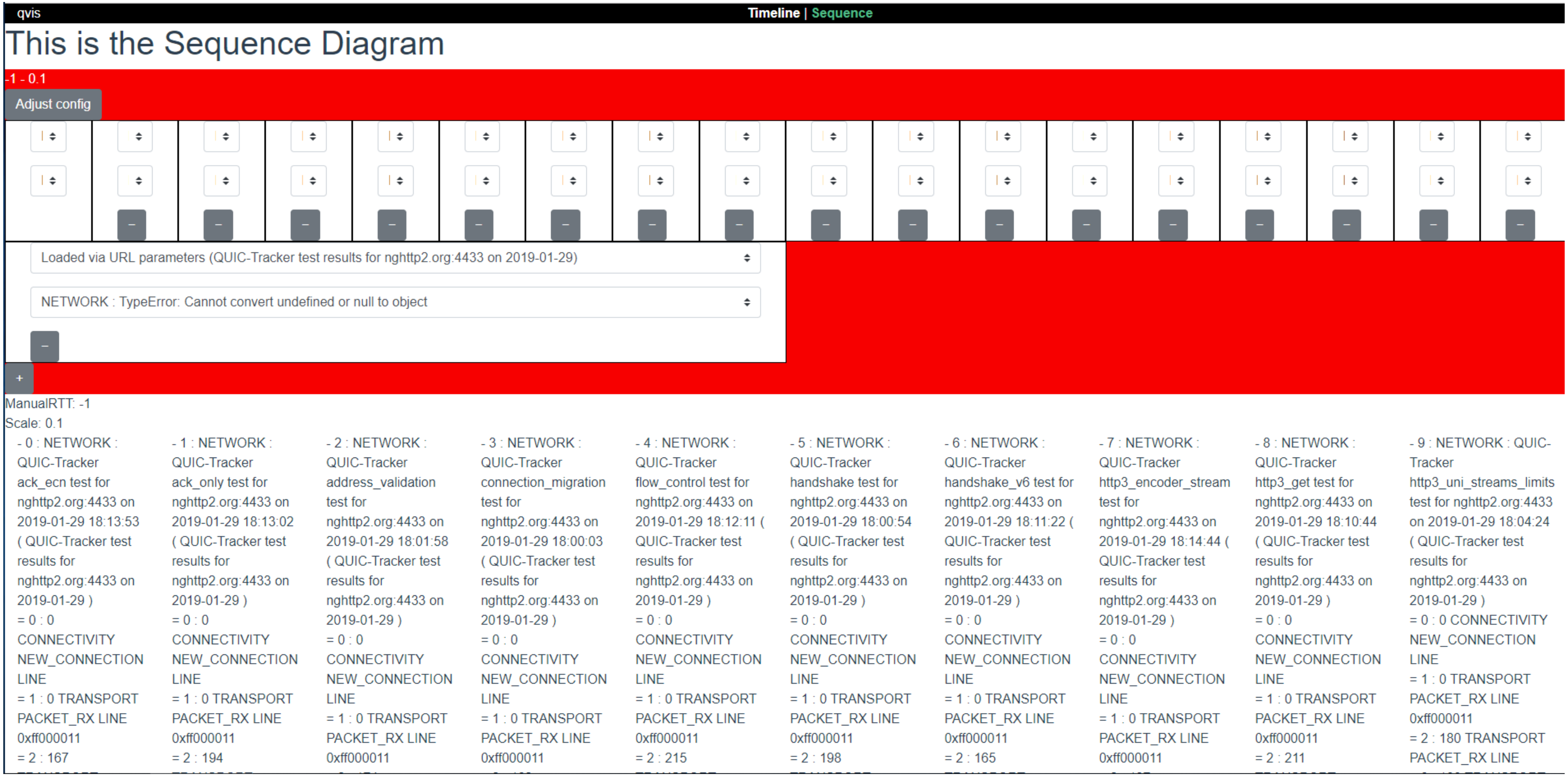


```
{
  "commit":"a44bab7a32a030b1102b683154dc591d50
  "scenario":"multi_stream",
  "scenario_version":1,
  "host":"nghttp2.org:4433",
  "ip":"139.162.123.134",
  "results":{},
  "started_at":1548785217,
  "duration":10006,
  "error_code":0
```

https://quic-tracker.info.ucl.ac.be/traces/20190129/229#

https://quicvis.edm.uhasselt.be:8443/#?list=https://quic-tracker.info.ucl.ac.be/traces/20190129/list/nghttp2.org:4433

QUICvis v1



This is the Sequence Diagram

Loaded via URL parameters (QUIC-Tracker test results for nghttp2.org:4433 on 2019-01-29)

NETWORK : TypeError: Cannot convert undefined or null to object

ManualRTT: -1

Scale: 0.1

- 0 : NETWORK : QUIC-Tracker ack_ecn test for nghttp2.org:4433 on 2019-01-29 18:13:53 (QUIC-Tracker test results for nghttp2.org:4433 on 2019-01-29) = 0 : 0 CONNECTIVITY NEW_CONNECTION LINE = 1 : 0 TRANSPORT PACKET_RX LINE 0xff000011 = 2 : 167

- 1 : NETWORK : QUIC-Tracker ack_only test for nghttp2.org:4433 on 2019-01-29 18:13:02 (QUIC-Tracker test results for nghttp2.org:4433 on 2019-01-29) = 0 : 0 CONNECTIVITY NEW_CONNECTION LINE = 1 : 0 TRANSPORT PACKET_RX LINE 0xff000011 = 2 : 194

- 2 : NETWORK : QUIC-Tracker address_validation test for nghttp2.org:4433 on 2019-01-29 18:01:58 (QUIC-Tracker test results for nghttp2.org:4433 on 2019-01-29) = 0 : 0 CONNECTIVITY NEW_CONNECTION LINE = 1 : 0 TRANSPORT PACKET_RX LINE 0xff000011

- 3 : NETWORK : QUIC-Tracker connection_migration test for nghttp2.org:4433 on 2019-01-29 18:00:03 (QUIC-Tracker test results for nghttp2.org:4433 on 2019-01-29) = 0 : 0 CONNECTIVITY NEW_CONNECTION LINE = 1 : 0 TRANSPORT PACKET_RX LINE 0xff000011

- 4 : NETWORK : QUIC-Tracker flow_control test for nghttp2.org:4433 on 2019-01-29 18:12:11 (QUIC-Tracker test results for nghttp2.org:4433 on 2019-01-29) = 0 : 0 CONNECTIVITY NEW_CONNECTION LINE = 1 : 0 TRANSPORT PACKET_RX LINE 0xff000011 = 2 : 215

- 5 : NETWORK : QUIC-Tracker handshake test for nghttp2.org:4433 on 2019-01-29 18:00:54 (QUIC-Tracker test results for nghttp2.org:4433 on 2019-01-29) = 0 : 0 CONNECTIVITY NEW_CONNECTION LINE = 1 : 0 TRANSPORT PACKET_RX LINE 0xff000011 = 2 : 198

- 6 : NETWORK : QUIC-Tracker handshake_v6 test for nghttp2.org:4433 on 2019-01-29 18:11:22 (QUIC-Tracker test results for nghttp2.org:4433 on 2019-01-29) = 0 : 0 CONNECTIVITY NEW_CONNECTION LINE = 1 : 0 TRANSPORT PACKET_RX LINE 0xff000011 = 2 : 165

- 7 : NETWORK : QUIC-Tracker http3_encoder_stream test for nghttp2.org:4433 on 2019-01-29 18:14:44 (QUIC-Tracker test results for nghttp2.org:4433 on 2019-01-29) = 0 : 0 CONNECTIVITY NEW_CONNECTION LINE = 1 : 0 TRANSPORT PACKET_RX LINE 0xff000011

- 8 : NETWORK : QUIC-Tracker http3_get test for nghttp2.org:4433 on 2019-01-29 18:10:44 (QUIC-Tracker test results for nghttp2.org:4433 on 2019-01-29) = 0 : 0 CONNECTIVITY NEW_CONNECTION LINE = 1 : 0 TRANSPORT PACKET_RX LINE 0xff000011 = 2 : 211

- 9 : NETWORK : QUIC-Tracker http3_uni_streams_limits test for nghttp2.org:4433 on 2019-01-29 18:04:24 (QUIC-Tracker test results for nghttp2.org:4433 on 2019-01-29) = 0 : 0 CONNECTIVITY NEW_CONNECTION LINE = 1 : 0 TRANSPORT PACKET_RX LINE 0xff000011 = 2 : 180 TRANSPORT PACKET_RX LINE

Logging format

- Current setup: transform .pcap to .json

- Problem: .pcap does not contain everything
  - Retransmit details (what was resent when and why?)
  - Exact latency information (Ack Delay on steroids)
  - HTTP/3 and QPACK state
  - Congestion control logic and state
  - *-> some can be inferred by heuristics, but not everything*

- "Solution": Parse client/server logs from # implementations
  - Problem: extremely tedious and error-prone

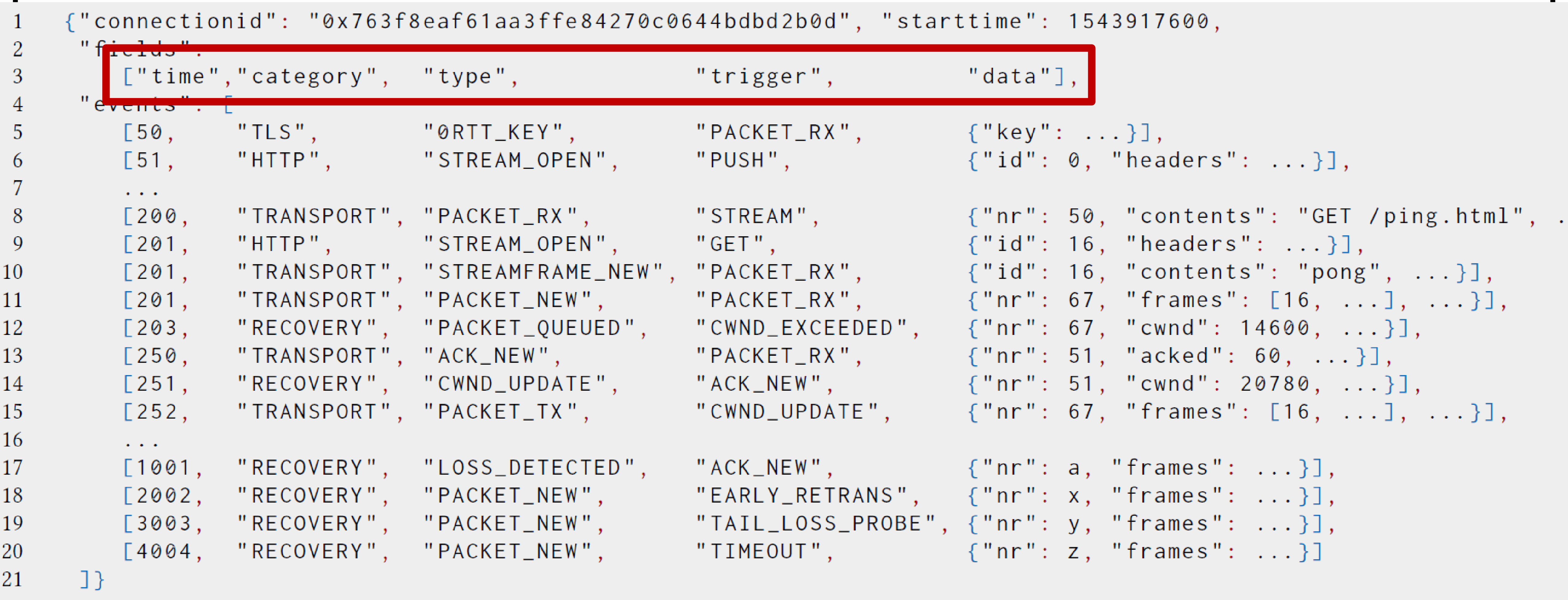
Logging format

- The real solution: (de-facto) standard logging schema

- quic-trace has one
    - Focused on congestion-related events
    - Protocol-buffers

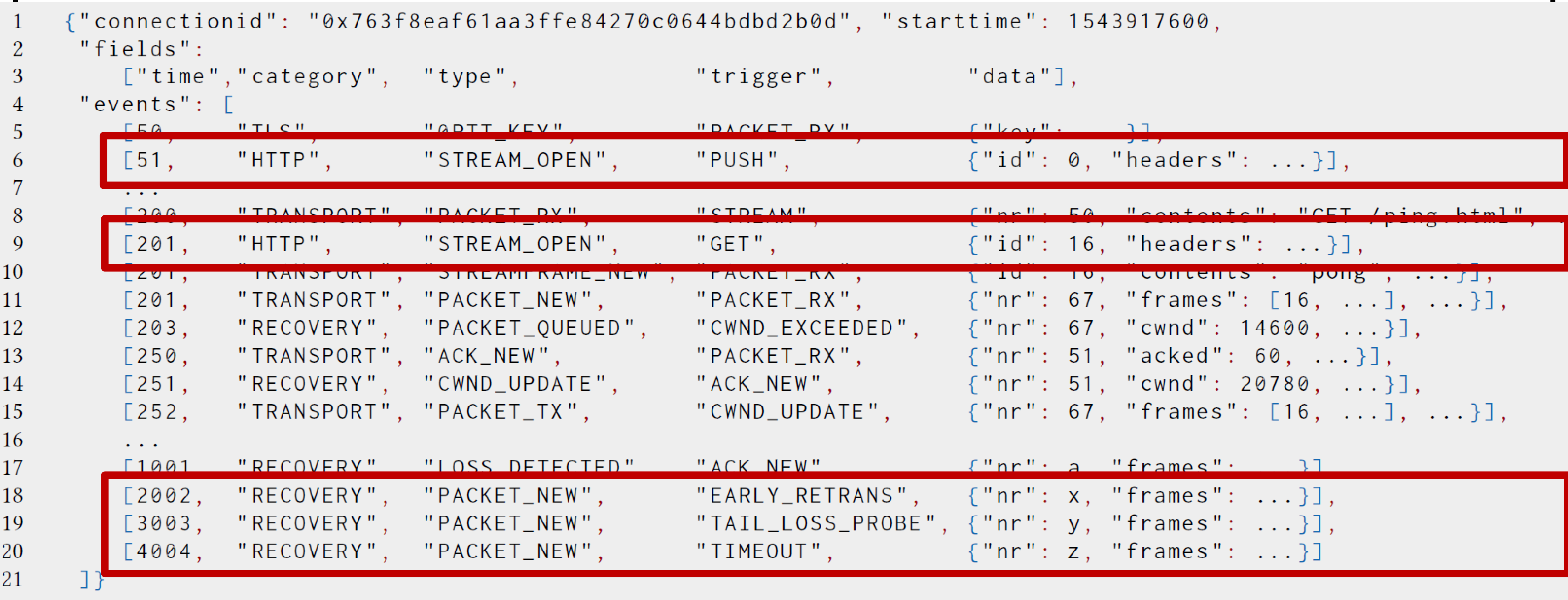
- We proposed one: qlog
    - Logs everything and more
    - .json-based (for easy integration with web-based tools)

qlog example : easy filtering and progressive enhancement



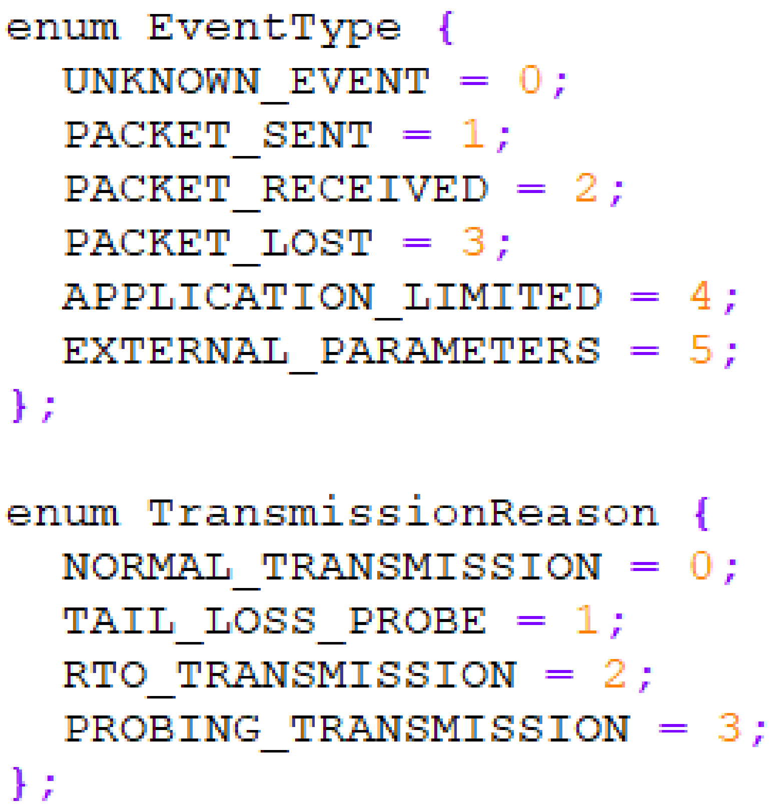
```
{"connectionid": "0x763f8eaf61aa3ffe84270c0644bdbd2b0d", "starttime": 1543917600,
 "fields":
    ["time","category", "type",             "trigger",          "data"],
 "events": [
    [50,    "TLS",       "0RTT_KEY",         "PACKET_RX",        {"key": ...}],
    [51,    "HTTP",      "STREAM_OPEN",      "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": ...}]
 ]}
```

qlog example : clear cause and effect



```
{"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-trace example



```
enum EventType {
  UNKNOWN_EVENT = 0;
  PACKET_SENT = 1;
  PACKET_RECEIVED = 2;
  PACKET_LOST = 3;
  APPLICATION_LIMITED = 4;
  EXTERNAL_PARAMETERS = 5;
};

enum TransmissionReason {
  NORMAL_TRANSMISSION = 0;
  TAIL_LOSS_PROBE = 1;
  RTO_TRANSMISSION = 2;
  PROBING_TRANSMISSION = 3;
};
```

Triggers / Reasons

- Better than heuristics
- Can be tricky to implement (pass around additional state)

- Worth the effort?
- Not for everything?

Standard schema

- Options for integration
  - Directly implemented (maybe replaces current logging?)
  - Transformed from other format
    - E.g., Facebook database, Fastly also setting up logging infra

- Flexibility is important
  - Define minimal subset, rest is icing on the cake
  - You can support only the stuff you need (initially ☺)
  - Different levels of verbosity (debugging vs production)

Let us have some feedback

- By Prague
  - Tools and visualizations workable
  - Extensive logging schema (maybe as a draft?)
  - Results of people using the tools

  - Hopefully one of you also has qlog support built-in by then ☺

Let us have some feedback

- What will make you consider qlog?
  - Auto-validator?
  - Schema in certain format?
  - I send you a PR that integrates logging functions?
  - A full-blown logging library in C?
  - Also: what will make you dismiss it outright?
    - “json should burn in hell”

- Which features and use cases are most important to you?
  - Short term and longer term (E.g., debugging stack vs production tracing)
  - What would you use it for? What makes the extra work worth it?