[qlog]

structured event logging for (encrypted) protocols

Robin Marx robin.marx@kuleuven.be

IETF 110

What’s in a name?

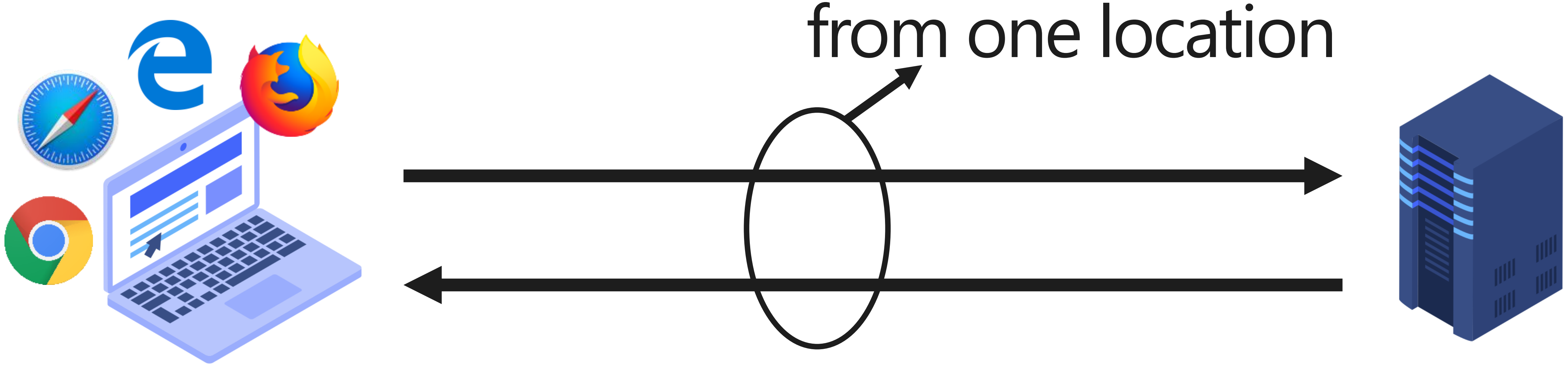
[qlog] = QUIC Logging

QUIC and HTTP/3 are complex

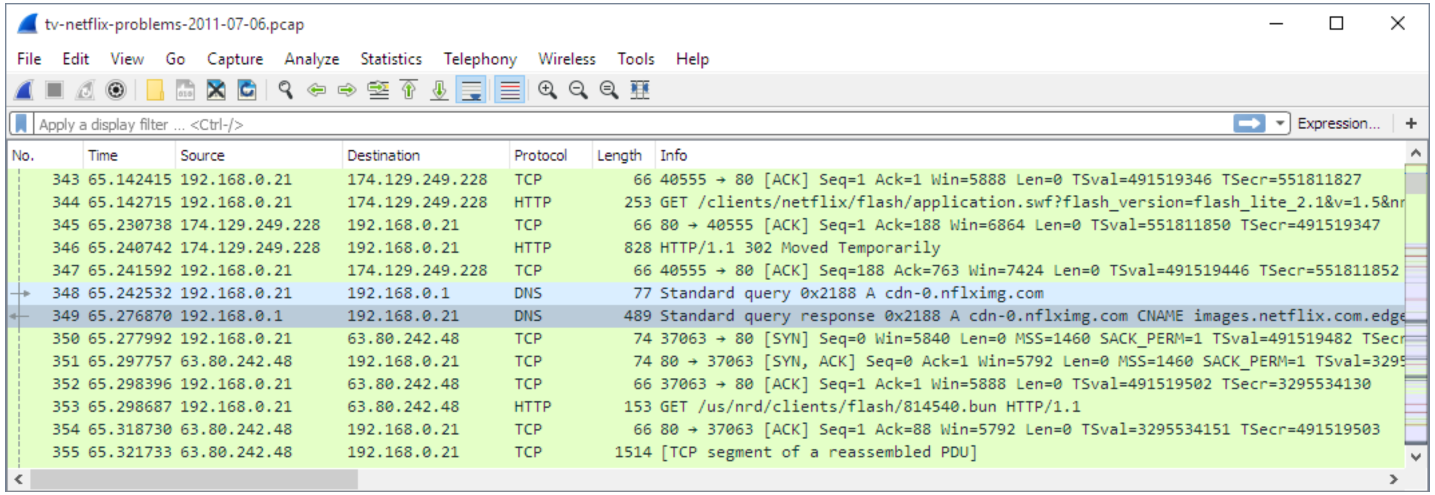
- Will need good debugging and analysis **tools**
- Tools need **data** to ingest

https://tools.ietf.org/html/draft-marx-qlog-main-schema-02
https://tools.ietf.org/html/draft-marx-qlog-event-definitions-quic-h3-02

Typical network logging



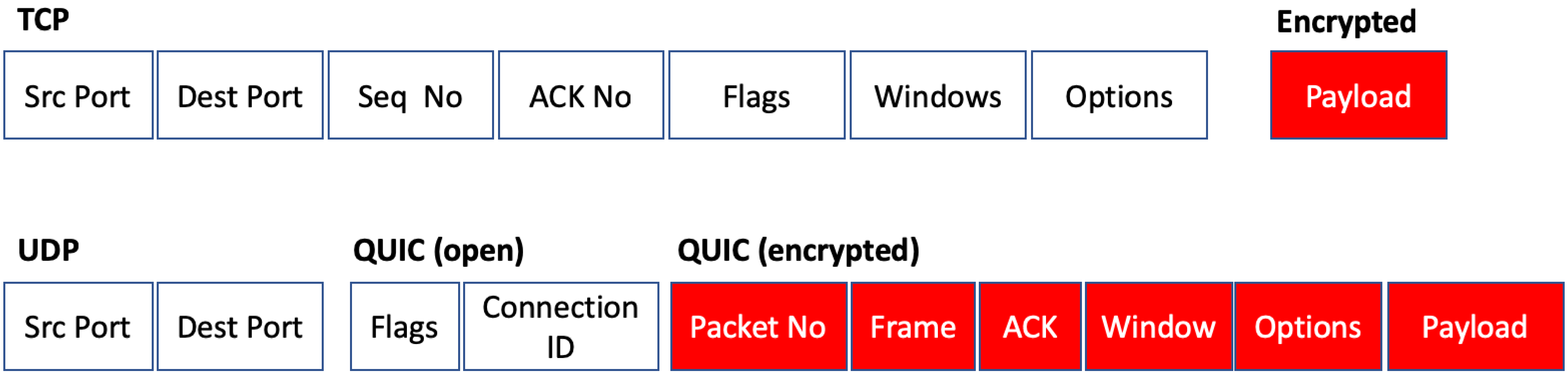
get raw wire image from one location



```
No.     Time      Source          Destination     Protocol Length Info
    343 65.142415 192.168.0.21    174.129.249.228 TCP        66 40555 → 80 [ACK] Seq=1 Ack=1 Win=5888 Len=0 TSval=491519346 TSecr=551811827
    344 65.142715 192.168.0.21    174.129.249.228 HTTP      253 GET /clients/netflix/flash/application.swf?flash_version=flash_lite_2.1&v=1.5&nr
    345 65.230738 174.129.249.228 192.168.0.21    TCP        66 80 → 40555 [ACK] Seq=1 Ack=188 Win=6864 Len=0 TSval=551811850 TSecr=491519347
    346 65.240742 174.129.249.228 192.168.0.21    HTTP      828 HTTP/1.1 302 Moved Temporarily
    347 65.241592 192.168.0.21    174.129.249.228 TCP        66 40555 → 80 [ACK] Seq=188 Ack=763 Win=7424 Len=0 TSval=491519446 TSecr=551811852
    348 65.242532 192.168.0.21    192.168.0.1     DNS        77 Standard query 0x2188 A cdn-0.nflximg.com
    349 65.276870 192.168.0.1     192.168.0.21    DNS       489 Standard query response 0x2188 A cdn-0.nflximg.com CNAME images.netflix.com.edge
    350 65.277992 192.168.0.21    63.80.242.48    TCP        74 37063 → 80 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 SACK_PERM=1 TSval=491519482 TSecr
    351 65.297757 63.80.242.48    192.168.0.21    TCP        74 80 → 37063 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM=1 TSval=3295
    352 65.298396 192.168.0.21    63.80.242.48    TCP        66 37063 → 80 [ACK] Seq=1 Ack=1 Win=5888 Len=0 TSval=491519502 TSecr=3295534130
    353 65.298687 192.168.0.21    63.80.242.48    HTTP      153 GET /us/nrd/clients/flash/814540.bun HTTP/1.1
    354 65.318730 63.80.242.48    192.168.0.21    TCP        66 80 → 37063 [ACK] Seq=1 Ack=88 Win=5792 Len=0 TSval=3295534151 TSecr=491519503
    355 65.321733 63.80.242.48    192.168.0.21    TCP      1514 [TCP segment of a reassembled PDU]
```

wireshark

1. QUIC is almost entirely encrypted



TCP

Src Port | Dest Port | Seq No | ACK No | Flags | Windows | Options

Encrypted

Payload

UDP

Src Port | Dest Port

QUIC (open)

Flags | Connection ID

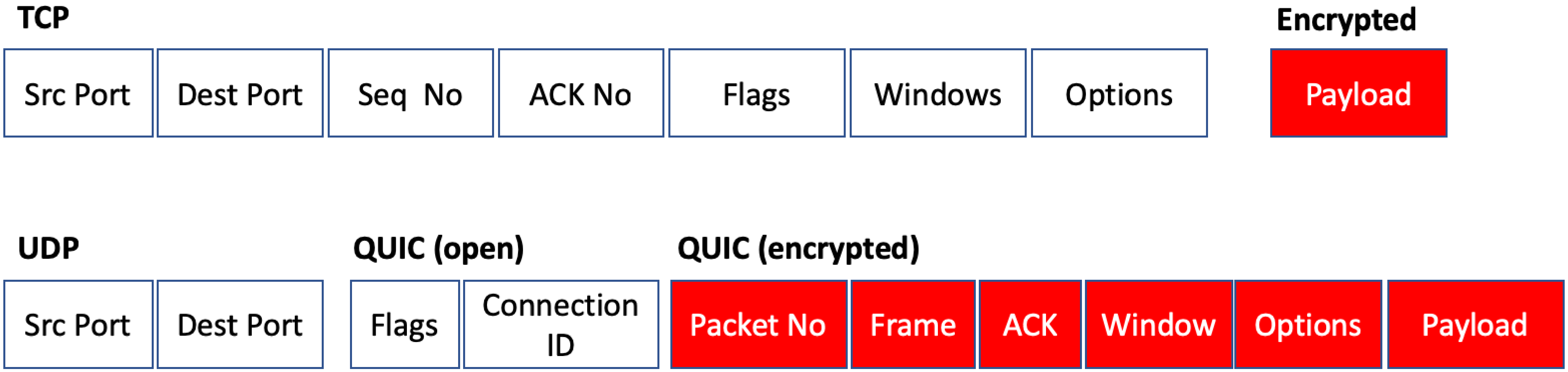
QUIC (encrypted)

Packet No | Frame | ACK | Window | Options | Payload

Storing full packet captures and TLS secrets is bad for:
- scalability
- privacy

img src: https://labs.apnic.net/?p=1207

1. QUIC is almost entirely encrypted



TCP

Src Port | Dest Port | Seq No | ACK No | Flags | Windows | Options

Encrypted

Payload

UDP

Src Port | Dest Port

QUIC (open)

Flags | Connection ID

QUIC (encrypted)

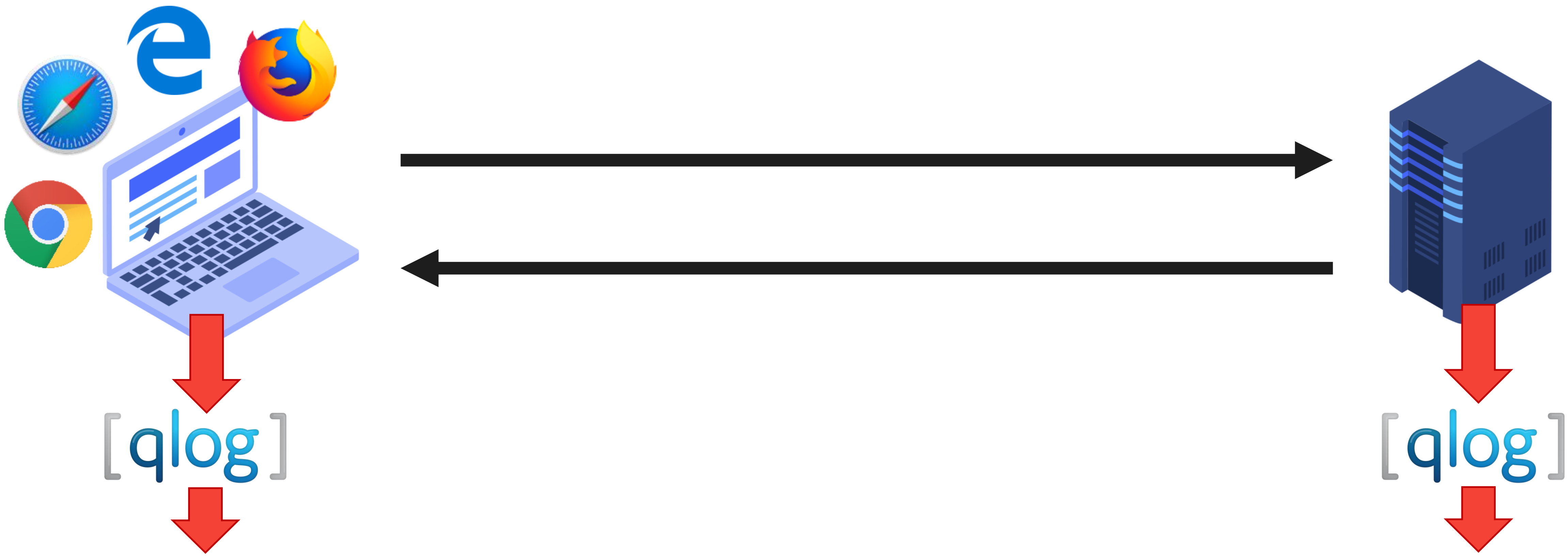
Packet No | Frame | ACK | Window | Options | Payload

2. not everything is sent on the wire

congestion control, decision making, internal errors, ...

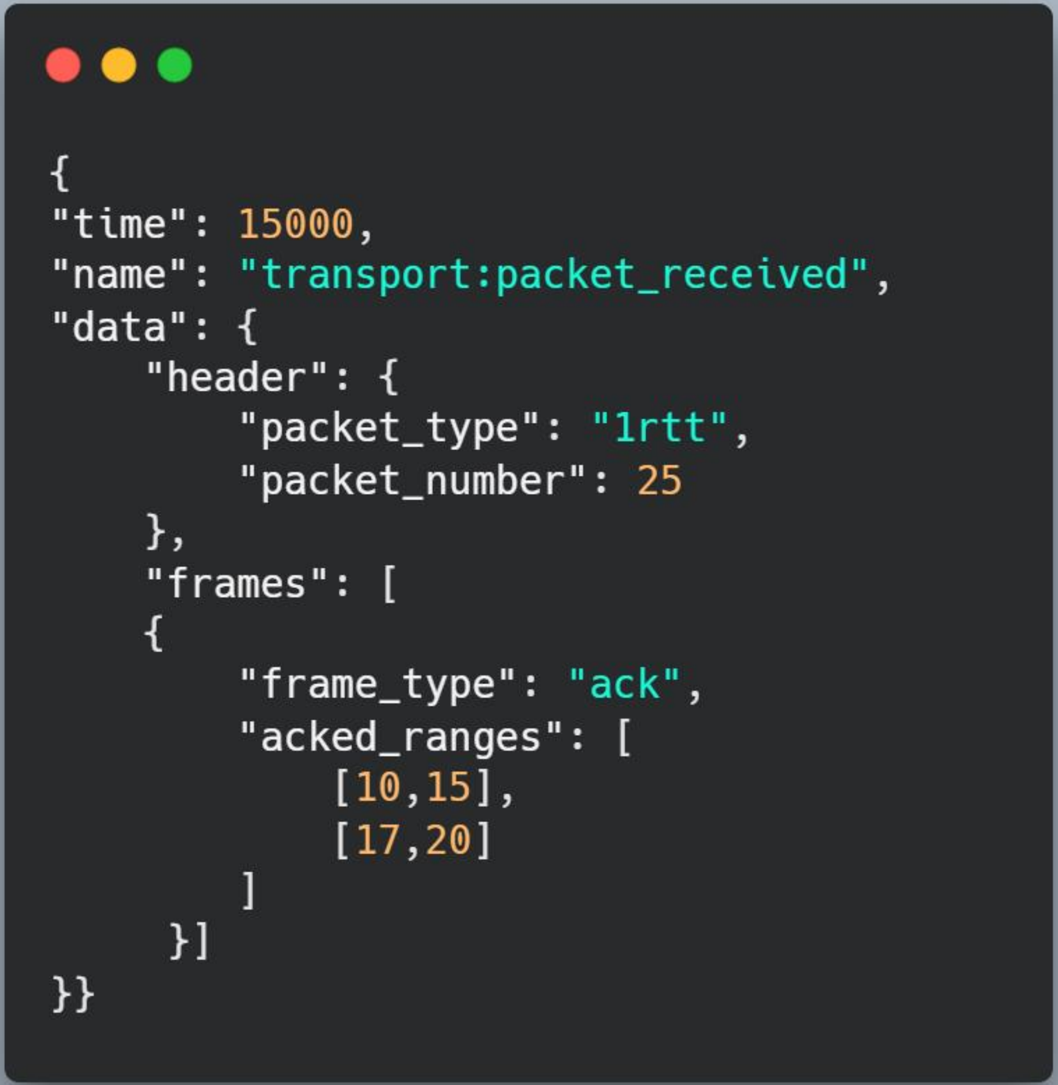
img src: https://labs.apnic.net/?p=1207

[qlog] structured endpoint logging

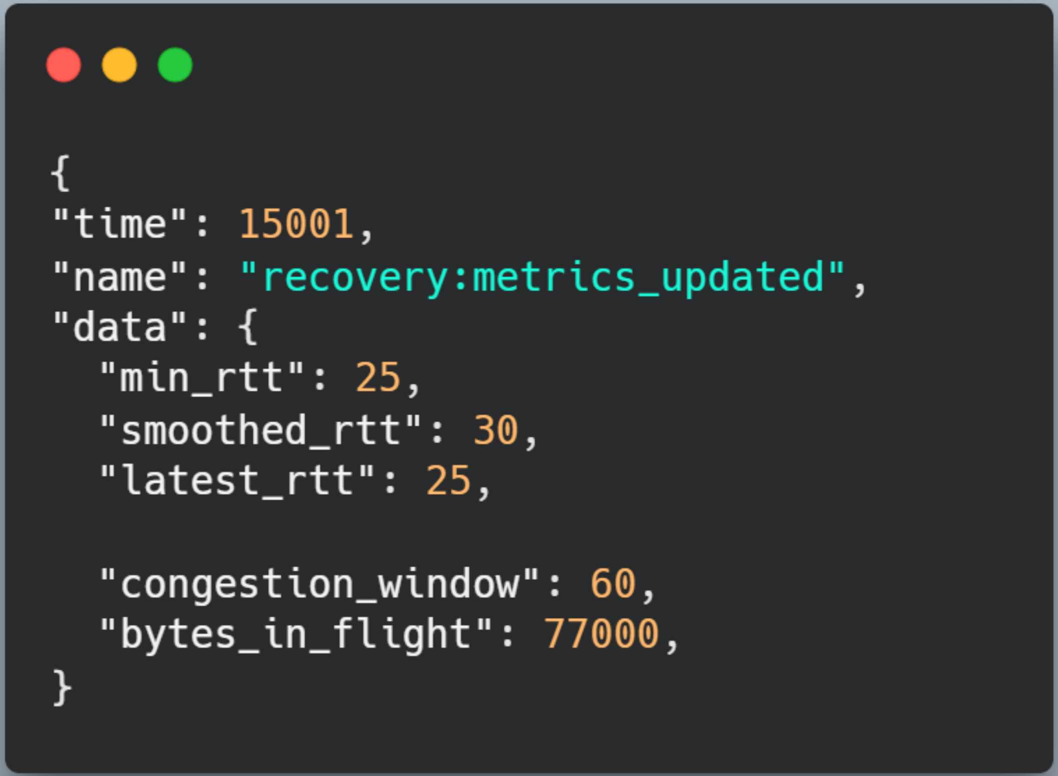


get data from (both) implementations directly

Event examples



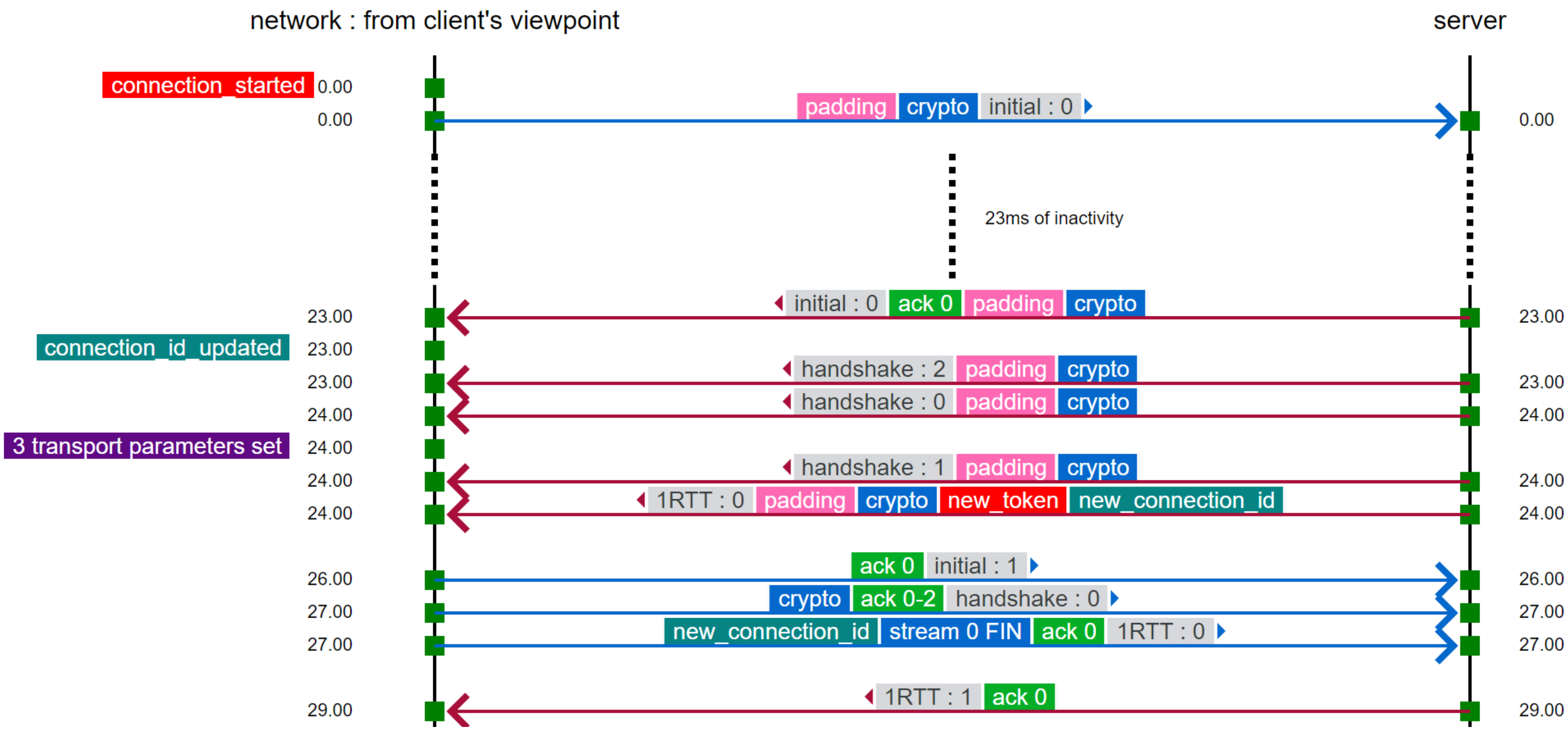
```
{
"time": 15000,
"name": "transport:packet_received",
"data": {
    "header": {
        "packet_type": "1rtt",
        "packet_number": 25
    },
    "frames": [
    {
        "frame_type": "ack",
        "acked_ranges": [
            [10,15],
            [17,20]
        ]
     }]
}}
```



```
{
"time": 15001,
"name": "recovery:metrics_updated",
"data": {
  "min_rtt": 25,
  "smoothed_rtt": 30,
  "latest_rtt": 25,

  "congestion_window": 60,
  "bytes_in_flight": 77000,
}
```

<qvis> QUIC and HTTP/3 tools



network : from client's viewpoint

server

connection_started 0.00

0.00 padding crypto initial : 0 0.00

23ms of inactivity

23.00 initial : 0 ack 0 padding crypto 23.00

connection_id_updated 23.00

23.00 handshake : 2 padding crypto 23.00

24.00 handshake : 0 padding crypto 24.00

3 transport parameters set 24.00

24.00 handshake : 1 padding crypto 24.00

24.00 1RTT : 0 padding crypto new_token new_connection_id 24.00

26.00 ack 0 initial : 1 26.00

27.00 crypto ack 0-2 handshake : 0 27.00

27.00 new_connection_id stream 0 FIN ack 0 1RTT : 0 27.00

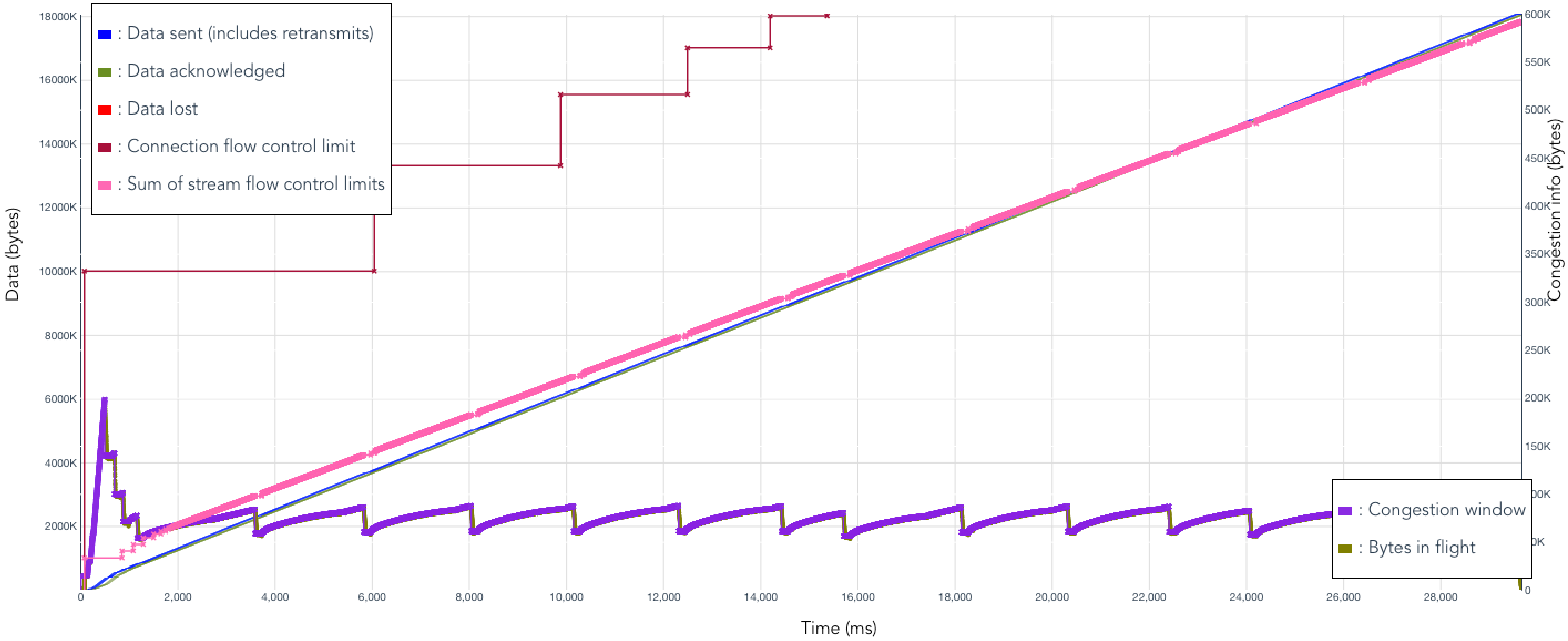
29.00 1RTT : 1 ack 0 29.00

https://qvis.quictools.info

https://github.com/quiclog/qvis

<qvis>

“TCPtrace” for QUIC



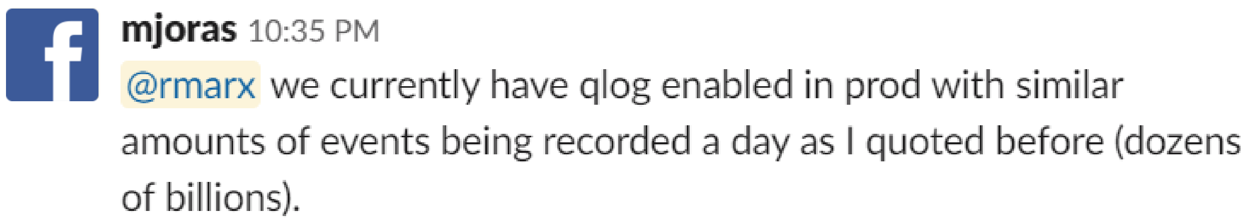
https://qvis.quictools.info

https://github.com/quiclog/qvis
https://blog.cloudflare.com/cubic-and-hystart-support-in-quiche

[qlog] support

> **75%** of QUIC/H3 stacks support direct qlog output:

- mvfst
- ngtcp2
- quiche
- quic-go
- aioquic
- quicly / H2O
- neqo
- picoquic
- ...



mjoras 10:35 PM

@rmarx we currently have qlog enabled in prod with similar amounts of events being recorded a day as I quoted before (dozens of billions).

[qlog] adoption

qlog draft adoption in QUIC wg

- Expected before or during IETF 111
- Part of recharter

Goals

- Flesh out schema’s for QUIC and HTTP/3

- **Prepare qlog for broader use with other protocols / applications**
    - TCP + TLS + HTTP/x
    - DNS, BGP, WebTransport
    - Multipath TCP and QUIC, MASQUE
    - Adaptive BitRate (ABR) video streaming logic
    - …

https://tools.ietf.org/html/draft-marx-qlog-main-schema-02
https://tools.ietf.org/html/draft-marx-qlog-event-definitions-quic-h3-02
https://research.edm.uhasselt.be/~mwijnants/pdf/herbotsCONEXT2020.pdf

[qlog] drafts

Main

Protocol-agnostic

- Container / metadata
- Format (JSON)
- *Best practices / guidelines*

QUIC

- Connectivity
- Transport
- Recovery

HTTP/3

- HTTP/3
- QPACK

…

Hopefully more to come

https://tools.ietf.org/html/draft-marx-qlog-main-schema-02
https://tools.ietf.org/html/draft-marx-qlog-event-definitions-quic-h3-02

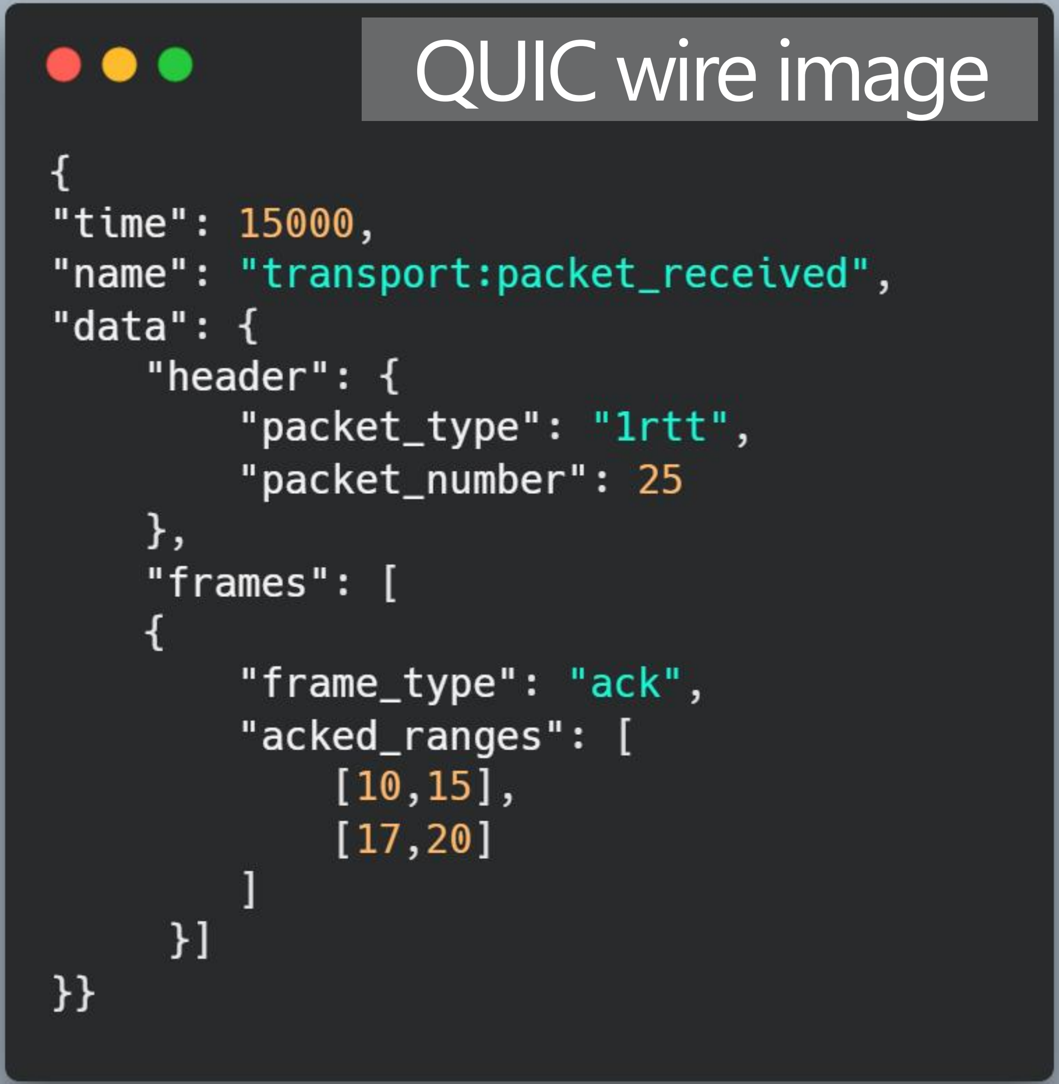
[qlog] for more than QUIC/H3

Plenty of challenges

- Event definitions
- Formats and datatypes
- Privacy and security aspects

- Operational aspects
- Cross-protocol tooling
- Protocol overlaps (e.g., TCP and QUIC, HTTP/3 vs HTTP/2 and 1, DoX, …)
- …

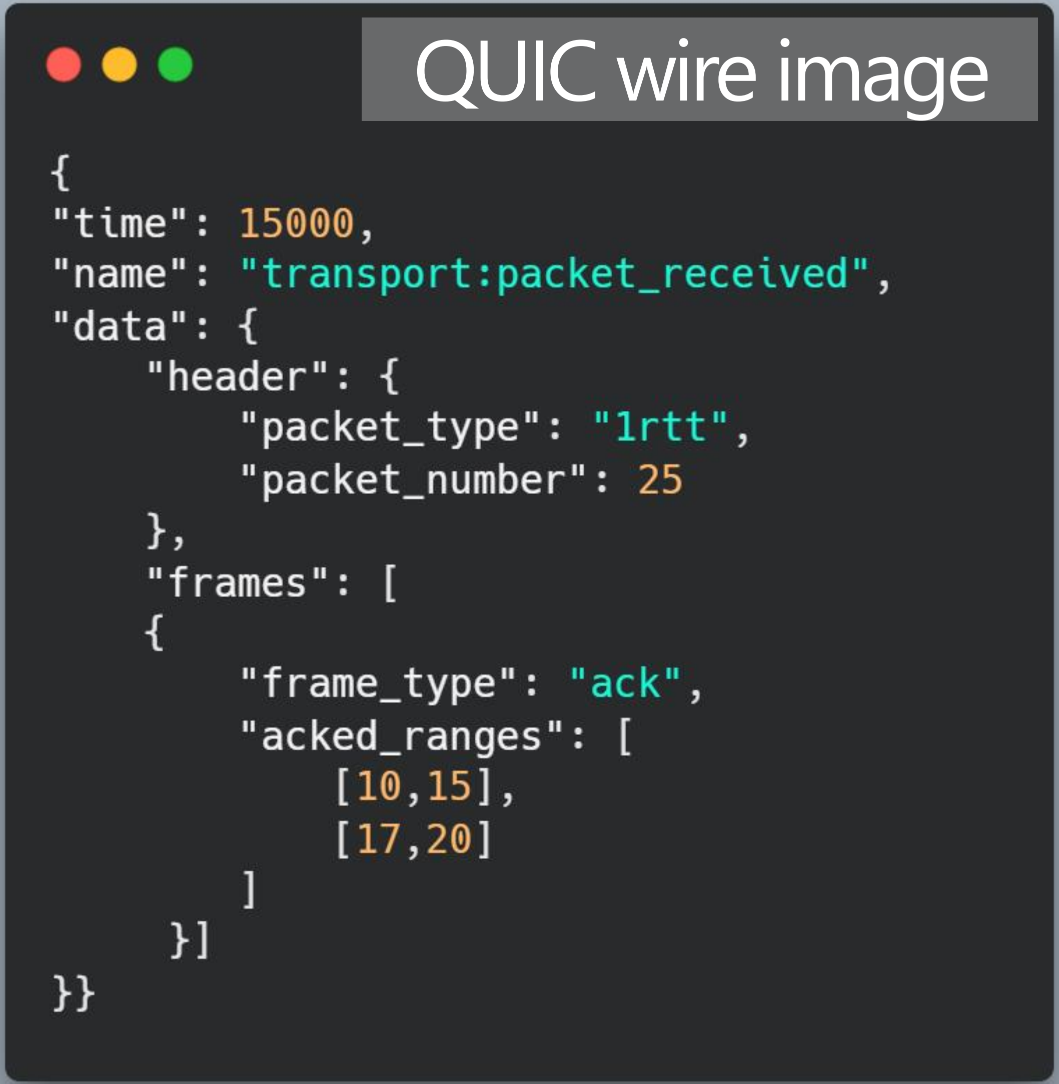
Event definitions



QUIC wire image

```
{
"time": 15000,
"name": "transport:packet_received",
"data": {
    "header": {
        "packet_type": "1rtt",
        "packet_number": 25
    },
    "frames": [
    {
        "frame_type": "ack",
        "acked_ranges": [
            [10,15],
            [17,20]
        ]
     }]
}}
```

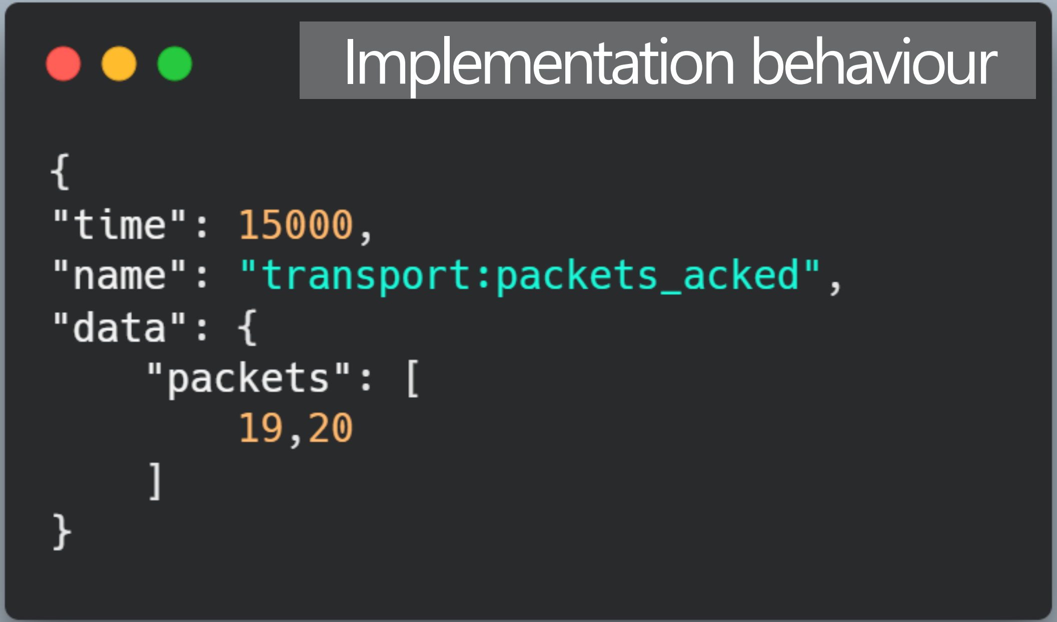
Event definitions



QUIC wire image

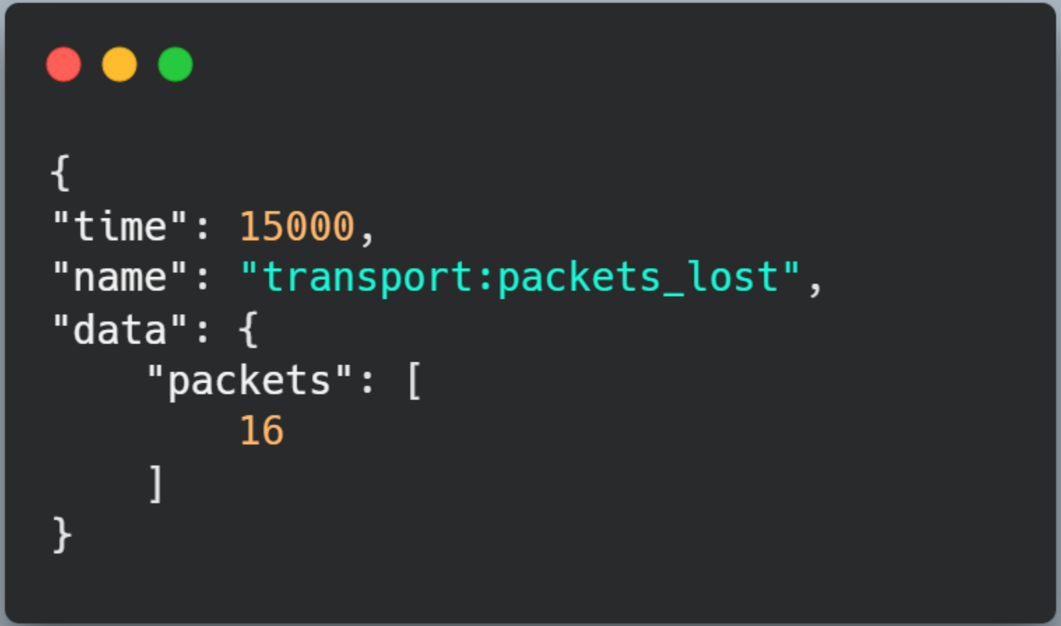
```
{
"time": 15000,
"name": "transport:packet_received",
"data": {
    "header": {
        "packet_type": "1rtt",
        "packet_number": 25
    },
    "frames": [
    {
        "frame_type": "ack",
        "acked_ranges": [
            [10,15],
            [17,20]
        ]
     }]
}}
```

and / or?



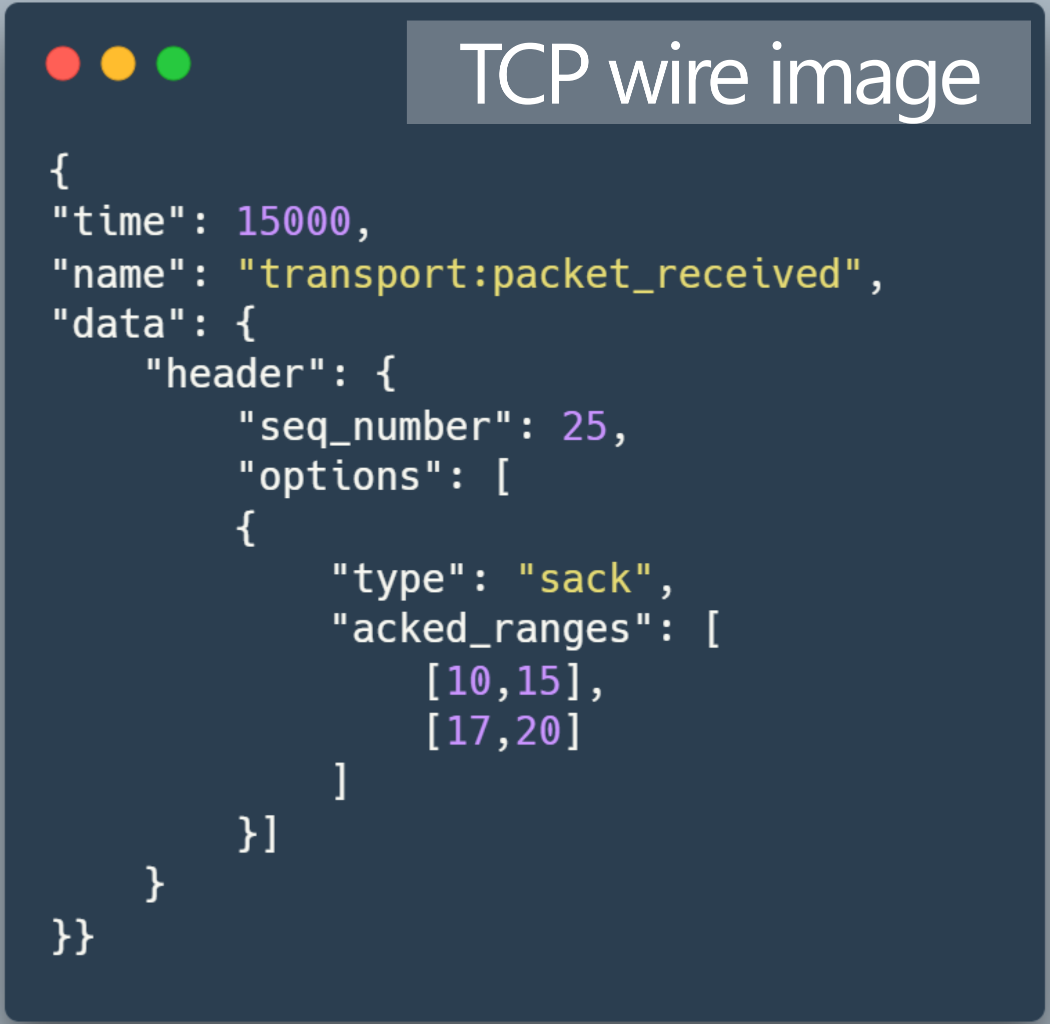
Implementation behaviour

```
{
"time": 15000,
"name": "transport:packets_acked",
"data": {
    "packets": [
        19,20
    ]
}
```



```
{
"time": 15000,
"name": "transport:packets_lost",
"data": {
    "packets": [
        16
    ]
}
```

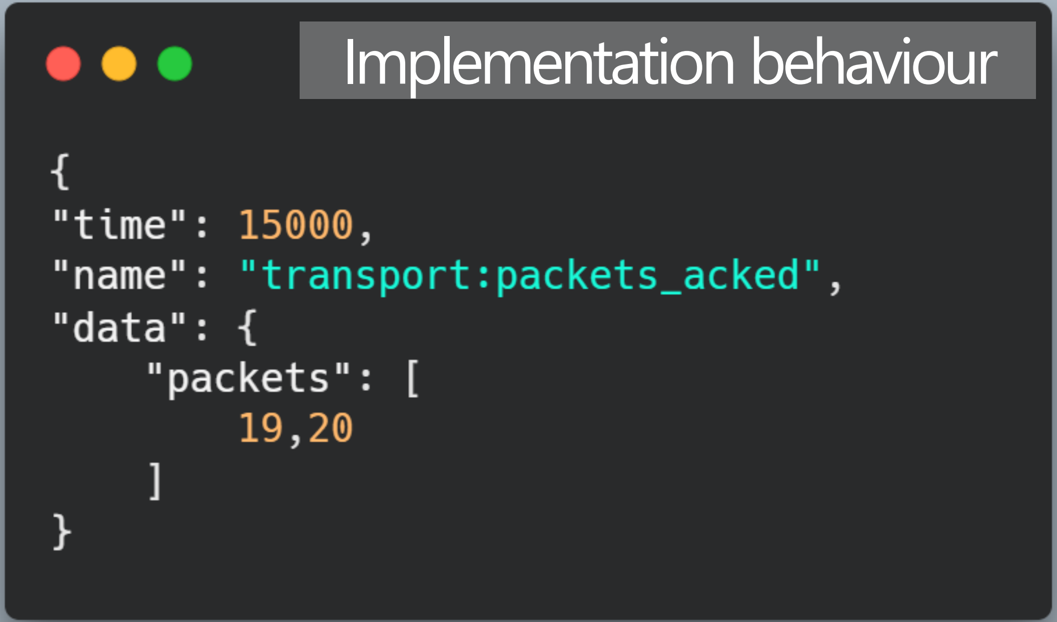
Event definitions



TCP wire image

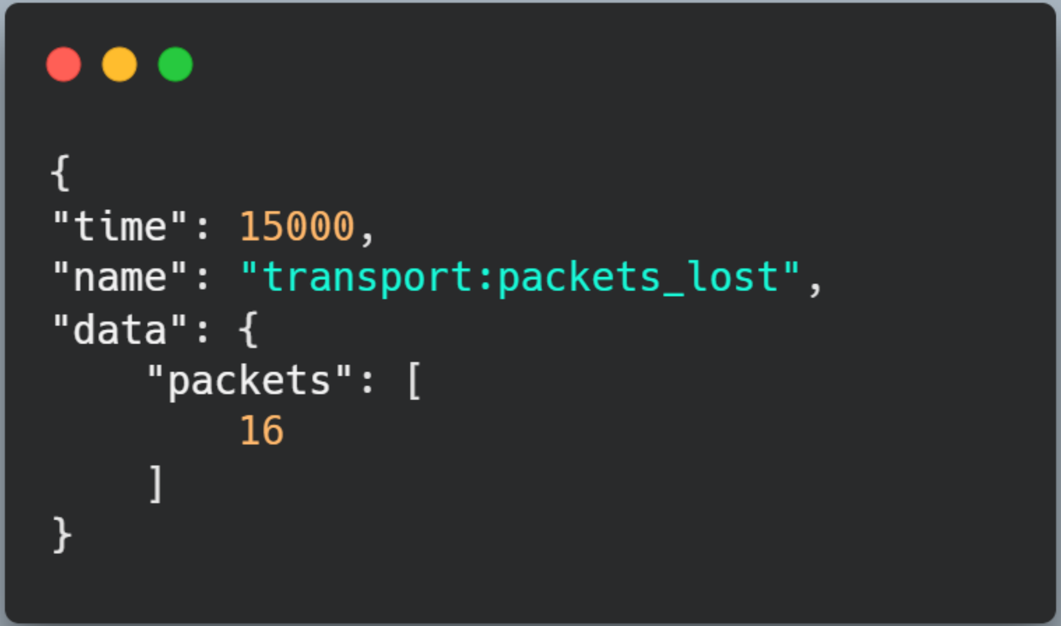
```
{
"time": 15000,
"name": "transport:packet_received",
"data": {
    "header": {
        "seq_number": 25,
        "options": [
        {
            "type": "sack",
            "acked_ranges": [
                [10,15],
                [17,20]
            ]
        }]
    }
}}
```

and / or?



Implementation behaviour

```
{
"time": 15000,
"name": "transport:packets_acked",
"data": {
    "packets": [
        19,20
    ]
}
```



```
{
"time": 15000,
"name": "transport:packets_lost",
"data": {
    "packets": [
        16
    ]
}
```

[qlog] serialization format

qlog is currently JSON-based

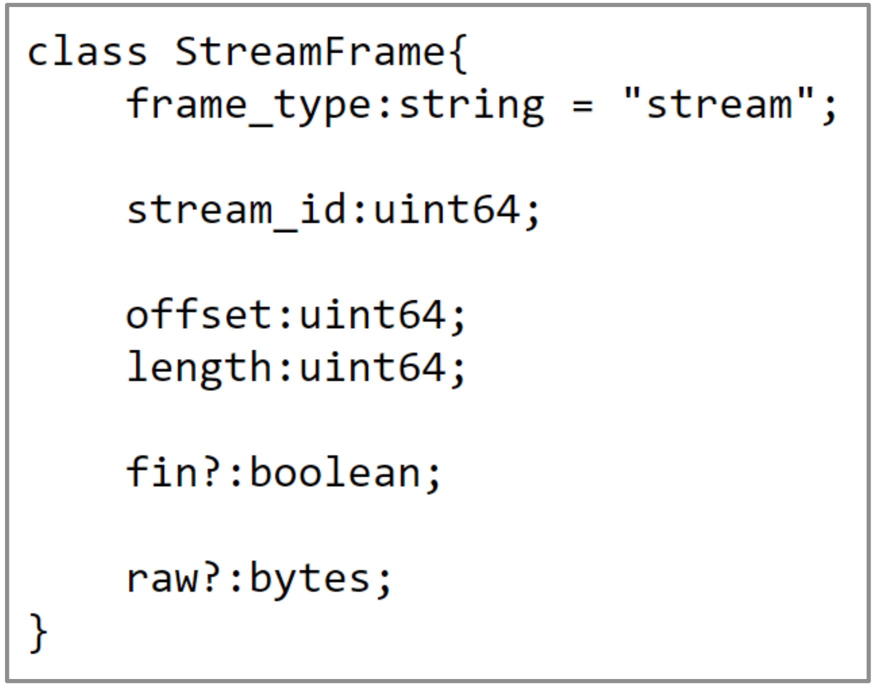
- 500 MB transfer → 300 MB qlog
- With compression: 18 MB

Format agnostic

- Define datatypes and schema
- Can be mapped to multiple serialization formats
  - Which one(s) should we focus on?
  - Automated generation from text?

Stream vs file-based

- Typical ingestion/storage/analysis pipelines



```
class StreamFrame{
    frame_type:string = "stream";

    stream_id:uint64;

    offset:uint64;
    length:uint64;

    fin?:boolean;

    raw?:bytes;
}
```

https://github.com/quiclog/internet-drafts/issues/30

[qlog] privacy and security

Lots of sensitive data

- IP addresses / Connection IDs
- HTTP payloads, SNIs
- Timestamps?

“Sanitization levels”

- From loose to strict
- Concrete guidelines and rules
- Tagging of individual fields

Log Storage/Transport/Sharing

- Encrypt logs themselves?
- Safe access to external log sources (e.g., QUIC manageability, research datasets)

https://huitema.wordpress.com/2020/07/21/scrubbing-quic-logs-for-privacy

Next steps

Eventually:

- Separate qlog wg for main aspects?
- Individual (protocol) wg's define new qlog documents?

First step:

- Drafts adoption in the QUIC wg (part of recharter)
- Expected before or during IETF 111

In the mean time

- Join us on github.com/quiclog/internet-drafts
- Join the qlog IETF mailing list ietf.org/mailman/listinfo/qlog

Give feedback now!