structured event logging for (encrypted) protocols

Robin Marx  
robin.marx@kuleuven.be
What’s in a name?

\[ \text{qlog} = \text{QUIC Logging} \]

QUIC and HTTP/3 are complex
- Will need good debugging and analysis tools
- Tools need data to ingest

Typical network logging

get raw wire image from one location

wireshark
1. QUIC is almost entirely encrypted

Storing full packet captures and TLS secrets is bad for:
- scalability
- privacy
1. QUIC is almost entirely encrypted

TCP

<table>
<thead>
<tr>
<th>Src Port</th>
<th>Dest Port</th>
<th>Seq No</th>
<th>ACK No</th>
<th>Flags</th>
<th>Windows</th>
<th>Options</th>
<th>Encrypted</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Payload</td>
</tr>
</tbody>
</table>

UDP

<table>
<thead>
<tr>
<th>Src Port</th>
<th>Dest Port</th>
<th>Flags</th>
<th>Connection ID</th>
</tr>
</thead>
</table>

QUIC (open)

<table>
<thead>
<tr>
<th>Packet No</th>
<th>Frame</th>
<th>ACK</th>
<th>Window</th>
<th>Options</th>
<th>Payload</th>
</tr>
</thead>
</table>

QUIC (encrypted)

2. not everything is sent on the wire
   congestion control, decision making, internal errors, ...
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
    }
}
```
QUIC and HTTP/3 tools

https://qvis.quictools.info
“TCPtrace” for QUIC

https://qvis.quictools.info

https://github.com/quiclog/qvis
> 75% of QUIC/H3 stacks support direct qlog output:

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

mijoras 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).

https://qlog.edm.uhasselt.be/anrw
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
  - ...

Facilitate research

Easier to:
- Compare different implementations
- Share datasets
- Get access to production/deployment datasets

(Full) Factorial tests

Store
Process
Aggregate

Visualize
Analyze

Share

https://qlog.edm.uhasselt.be
https://github.com/quiclog/internet-drafts
Datamining (sanitized) production data

“In almost 60% of the events describing the loss of packet number N, the packet number N-1 was also lost”
Next steps

Come join us:
- 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