structured event logging for (encrypted) protocols

Robin Marx  robin.marx@kuleuven.be
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
Typical network logging

get raw wire image from one location
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>
</table>

UDP

<table>
<thead>
<tr>
<th>Src Port</th>
<th>Dest Port</th>
<th>Flags</th>
<th>Connection ID</th>
<th>QUIC (open)</th>
<th>QUIC (encrypted)</th>
</tr>
</thead>
</table>

<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>

2. not everything is sent on the wire
congestion control, decision making, internal errors, ...
[qlog] structured endpoint logging

get data from (all) vantagepoints directly
Event examples

```json
{
  "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

@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 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
  - ...

Main
Protocol-agnostic
- Container / metadata
- Format (JSON)
- Best practices / guidelines

QUIC
- Connectivity
- Transport
- Recovery

HTTP/3
- HTTP/3
- QPACK

Hopefully more to come

Plenty of challenges
- Formats and datatypes
- Privacy aspects
- Operational aspects
- Event definitions
- Cross-protocol tooling
- Protocol overlaps (e.g., TCP and QUIC, HTTP/3 vs HTTP/2 and 1, DoX, …)
- …
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

```python
class StreamFrame{
    frame_type: string = "stream";
    stream_id: uint64;
    offset: uint64;
    length: uint64;
    fin?: boolean;
    raw?: bytes;
}
```
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
Could help with (QUIC) manageability?

- Spinbit/lossbit woes
- Share qlogs between systems (e.g., network vs server operators)
- How to?
  - Request
  - Transport
  - Store
  - Aggregate
  - Control access

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