[qlog]

structured event logging

The search for the one streaming format

Robin Marx   robin.marx@kuleuven.be

IETF 112

draft-01: Moved from NDJSON to RFC7464

Normal JSON

```
[
    {
        "title": "Dune",
        "the_one": "Kwisatz Haderach"
    },
    {
        "title": "The Wheel of Time",
        "the_one": "Dragon Reborn"
    }
]
```

Pros

- Well-defined
- Broadly supported

Cons

- Non-streamable

Main Discussion: https://github.com/quicwg/qlog/issues/172

draft-01: Moved from NDJSON to RFC7464

Newline Delimited JSON (NDJSON)

```
{ "title": "Dune", "the_one": "Kwisatz Haderach" }
{ "title": "The Wheel of Time", "the_one": "Dragon Reborn" }
```

Pros

- Streamable

Cons

- Not defined in RFC
- Not broadly supported
- No newlines in data...

Main Discussion: https://github.com/quicwg/qlog/issues/172

draft-01: Moved from NDJSON to RFC7464

JSON Text Sequences (RFC7464)

```
<RS>{ "title": "Dune",
      "the_one": "Kwisatz Haderach" }
<RS>{ "title": "The Wheel of Time",
      "the_one": "Dragon Reborn" }
```

Pros

- Streamable
- Well-defined in RFC
- Newlines in event

Cons

- Not broadly supported
- Annoying **<RS>** character
    - Record separator, U+001E

3

Main Discussion: https://github.com/quicwg/qlog/issues/172

draft-01: Moved from NDJSON to RFC7464

Early tests

- Very similar to NDJSON to parse/use
- Existing text-based tools deal with it just fine (jq, sed/awk/grep/...)

qlog-main-schema-draft-01

- Moved from NDJSON to JSON Text Sequences
- New `qlog_version` value: 0.3
- New `qlog_format` value: JSON-SEQ
- New file extension: .sqlog
- Added media types: application/qlog+json(-seq)

Currently supported by

- qvis
- quic-go
- ngtcp2
- Cloudflare quiche (pending PR)

Main takeaways:

- **Both** normal **JSON and JSON-SEQ** are specced
- Unlikely to change much in future, so **no need to wait**

4

Plans for draft-02: editorial changes

How structures/events are defined *in the drafts* (mainly editorial)

TypeScript

```
class Prophecy {
  title: string,
  the_one: string,
  occurrences?: number,
  incarnation?: Array<Manifestation>,
  avatar?: bytes
}
```

CDDL
(Concise Data Definition Language)

```
Prophecy = {
  title: tstr,
  the_one: tstr,
  ? occurrences: uint,
  ? incarnations: [* Manifestation],
  ? avatar: bstr
}
```

- JSON-mapping needs to be defined in qlog

- Well-defined in RFCs
- Automated tooling for generating **schema-definitions and validators**

Main Discussion: https://github.com/quicwg/qlog/issues/143
Editorial Issues: https://github.com/quicwg/qlog/issues?q=is%3Aissue+is%3Aopen+label%3Aeditorial
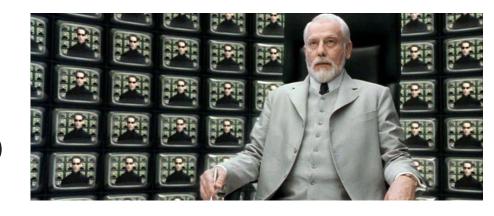
draft-03 and beyond

Flesh out

- Extend TLS + QPACK definitions
- Fix semantic details (e.g., tokens, RawInfo, 0-RTT, congestion states)
- Add additional high-level events/fields (e.g., CPU/threading info, blocking indicators)

Versioning and extensions

- Main schema versions vs protocol event versions
- protocol_type indication
- Built-in Transport Parameters vs extension TPs
- "Official" extensions (Datagram, Masque, WebTransport, …)
- Multipath support, aggregated metrics, …

Some of this is practical (new drafts for extensions? where?),
some is design

→ Could use some extra eyeballs on these
   in the coming months

Versioning: https://github.com/quicwg/qlog/issues/156
Extensions: https://github.com/quicwg/qlog/issues/170
TP's: https://github.com/quicwg/qlog/issues/176
protocol_type: https://github.com/quicwg/qlog/issues/146
Multipath: https://github.com/quicwg/qlog/issues/134
CPU/threading info: https://github.com/quicwg/qlog/issues/152

6

draft-03 and beyond

Flesh out

- Extend TLS + QPACK definitions
- Fix semantic details (e.g., tokens, RawInfo, 0-RTT, congestion states)
- Add additional high-level events/fields (e.g., CPU/threading info, blocking indicators)

Versioning and extensions

- Main schema versions vs protocol event versions
- protocol_type indication
- Built-in Transport Parameters vs extension TPs
- "Official" extensions (Datagram, Masque, WebTransport, …)
- Multipath support, aggregated metrics, …

Some of this is practical (new drafts for extensions? where?), some is design

→ Could use some extra eyeballs on these in the coming months

Overall goal:

- Focus on QUIC/H3 for RFC
  - Hopefully even -04 or -05
- Look into TCP etc. later
  - Re-usable tooling seems like *utopia* atm
  - Consistent design philosophy seems *unattainable*

Versioning: https://github.com/quicwg/qlog/issues/156
Extensions: https://github.com/quicwg/qlog/issues/170
TP's: https://github.com/quicwg/qlog/issues/176
protocol_type: https://github.com/quicwg/qlog/issues/146
Multipath: https://github.com/quicwg/qlog/issues/134
CPU/threading info: https://github.com/quicwg/qlog/issues/152