The **DUNE: PART TWO** Update

IETF 119 – Brisbane – 2024-03

Robin Marx, Luca Niccolini, Marten Seemann, Lucas Pardue
Since IETF 118

- Published 3 new drafts
  - Removed QPACK
  - transport:datagrams_sent → transport:udp_datagrams_sent
  - Editorial updates
  - Lots of clarifications / RFC alignment
  - Groundwork for extensibility (today)

- Big thanks to Hugo Landau
  - OpenSSL QUIC/qlog implementer
Since IETF 118: Merged (Multi)path/Migration support!

Simple but extensible approach:

```
{time: 12456, path: "my_first_path", name: "quic:packet_sent", data: {...}}
```

```plaintext
PathAssigned = {
    path_id: text
    ? path_remote: PathEndpointInfo
    ? path_local: PathEndpointInfo
}

PathEndpointInfo = {
    ? ip: IPAddress
    ? port: uint16
    ? connection_ids: [+ ConnectionID]
}
```

Feedback/experience still welcome!
Extensibility: which events are you using exactly #415

Taking inspiration from RFC8285:

```
QlogFile = {
    ...
    "additional_event_schemas": [
        "urn:ietf:params:qlog:http3",
        "urn:ietf:params:qlog:quic#transport",
        "urn:ietf:params:qlog:quic#connectivity",
        "https://atreides.com/~paul/032024/dune_name_system.html"
    ]
    ...
}
```

New documents register URNs with IANA with urn:ietf:params:qlog prefix

? Should absence of #category modifiers indicate all categories are used?
Extensibility: properly add new types #417

*Without* proper extensibility:

MaxDataFrame = {
    frame_type: "max_data"
    maximum: uint64
}

PacketSent = {
    frames: [* MaxDataFrame / StreamFrame / ...]
    ...
}

**Too rigid**: impossible to add new frame types
Extensibility: properly add new types #417

Using CDDL “type sockets”:

MaxDataFrame = {
    frame_type: "max_data"
    maximum: uint64
}

PacketSent = {
    frames: [* $QuicFrame]
    ...
}

$QuicFrame /= MaxDataFrame

Later extension, separate document:
$QuicFrame /= AckFrequencyFrame
Extensibility: extend **existing** things #417

What we had BEFORE:

```plaintext
QUICParametersSet = {
    ? ack_delay_exponent: uint16
    ? max_ack_delay: uint16
    ...

    ; to support later defined parameters
    * text => any
}
```

**Too flexible**: impossible to really type-check
Extensibility: extend **existing** things #417

Using CDDL “group sockets”:

```cddl
QUICParametersSet = {
   ? ack_delay_exponent: uint16
   ? max_ack_delay: uint16
   ...

   ; to support later defined parameters
   * $$quic-parametersset-extension
}
```

**Separate document for Ack Frequency Extension:**

```cddl
$$quic-parametersset-extension //= ( 
   ? min_ack_delay: uint64
 )
```
Focus: Extensibility for main RFC extension points

Mostly IANA-registered extensions (with some additions):

- Packets
  - packetheader-extension, $PacketType
- Frames
  - $QuicFrame, $H3Frame, $H3Datagram
- Transport Parameters, Settings
  - $quic-parameterset-extension, $h3-parameters-extension
- Stream types
  - $H3StreamType
- Error codes
  - $TransportError, $ApplicationError
- Protocol identifiers
  - $ProtocolType

Good time to **try and exercise** these (Multipath + Media-over-QUIC: we’re looking at you ;)
How to communicate fin, stream_reset, stop_sending

Signals not always *immediately* communicated to application layer

E.g., only bubbled up when there’s a read from QUIC layer

```plaintext
QUICStreamDataMoved = {
  ? stream_id: uint64
  ? offset: uint64
  ? length: uint64

  ? from: Layer
  ? to: Layer

  ? additional_info: [+ text]
}
```

Examples: "fin_set", "stream_reset", "stop_sending"

Look at other events for more details
Moving towards WGLC by end of year

No open major design issues!

Should be below 30 issues and 10 PRs soon

Fixes #379.
Also closes #261, #176, #170, #124, #192, #297.

How to help:

- Comprehensive document reviews
- Exercise extension points
- Create a qlog issue today!