Using QUIC Datagrams with HTTP/3

draft-schinazi-masque-h3-datagram
Virtual Interim - 2021-01

Lucas Pardue - lucaspardue.24.7@gmail.com
David Schinazi - dschinazi@google.com

Recap from where we were at IETF 109

When QUIC is in use and ALPN=h3,

> Every QUIC DATAGRAM frame starts with a Flow Identifier (62-bit integer)
> Both endpoints provide a flow allocation service to get unique identifiers
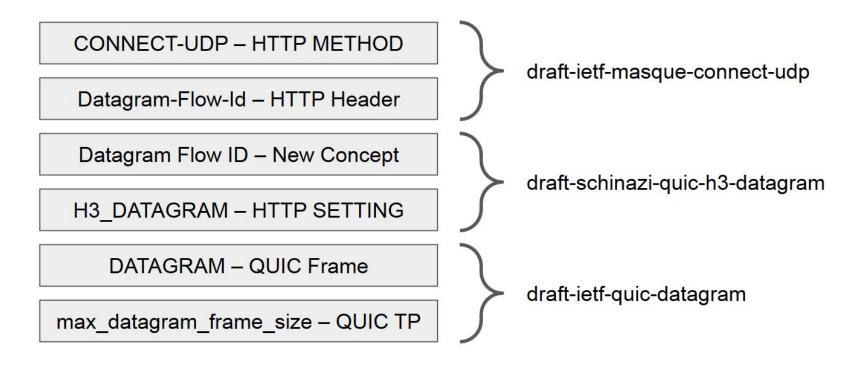> The protocol to negotiate these flow IDs is not defined in that draft

CONNECT-UDP carries the new "Datagram-Flow-Id" header to indicate flow ID

```
:method = CONNECT-UDP
:authority = server.example.com:443
Datagram-Flow-Id = 42
```
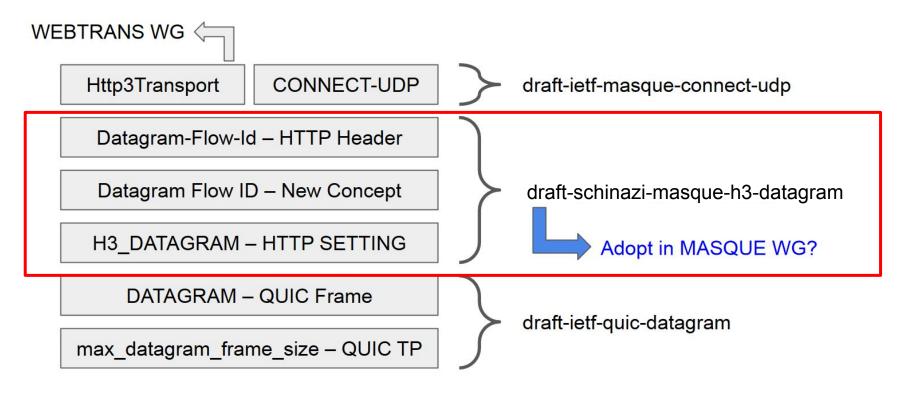
draft-schinazi-masque-h3-datagram - Virtual Interim - 2021-01

What was in the drafts at IETF 109

CONNECT-UDP – HTTP METHOD	draft-ietf-masque-connect-udp
Datagram-Flow-Id – HTTP Header	
Datagram Flow ID – New Concept	draft-schinazi-quic-h3-datagram
H3_DATAGRAM – HTTP SETTING	
DATAGRAM – QUIC Frame	draft-ietf-quic-datagram
max_datagram_frame_size – QUIC TP	

What we have today

| Http3Transport | CONNECT-UDP | draft-ietf-masque-connect-udp |

| Datagram-Flow-Id – HTTP Header |
| Datagram Flow ID – New Concept | draft-schinazi-masque-h3-datagram |
| H3_DATAGRAM – HTTP SETTING | → Adopt in MASQUE WG? |

| DATAGRAM – QUIC Frame | draft-ietf-quic-datagram |
| max_datagram_frame_size – QUIC TP |

draft-schinazi-masque-h3-datagram - Virtual Interim - 2021-01

Headline changes

- draft-schinazi-quic-h3-datagram has been renamed to draft-schinazi-masque-h3-datagram
- Datagram-Flow-Id header has been moved from draft-ietf-masque-connect-udp to draft-schinazi-masque-h3-datagram
- Datagram-Flow-Id has been changed from an sf-item to an sf-list
- RfcDiff - https://tools.ietf.org/rfcdiff?url1=draft-schinazi-quic-h3-datagram-05.txt&url2=draft-schinazi-masque-h3-datagram-04.txt

Supporting Flow ID extensibility and multiplicity

Previously, Datagram-Flow-Id was a singular sf-integer, which was fine for the simple CONNECT-UDP case, but not easily extensible.

Active WG discussion about using multiple flow IDs for more-advanced use cases.

Draft-schinazi-masque-h3-datagram "owns" the definition of Datagram-Flow-Id, independent of use case.

We've changed to an sf-list to allows multiple flow IDs, that can each be parameterized.

Basic example

Datagram-Flow-Id = sf-list

- List members are flow identifier elements, which can be named or unnamed.
- Name is encoded in the key of the first parameter of that element.
- Each name MUST NOT appear more than once in the list. The value of the first parameter of each named element (whose corresponding key conveys the element name) MUST be of type Boolean and equal to true.

```
Datagram-Flow-Id = 2 ✅
Datagram-Flow-Id = 2; my-cool-flow ✅
Datagram-Flow-Id = 2; my-cool-flow, 4; my-cool-flow ❌
```

The one with the ECN example

An HTTP method that wishes to use four datagram flow identifiers:

```
Datagram-Flow-Id = 42, 44; ecn-ect0, 46; ecn-ect1, 48; ecn-ce ✓
```

Where 42 is the unnamed flow identifier, 44 represents the
name "ecn-ect0", 46 represents "ecn-ect1", and 48 represents "ecn-ce".

List order has no meaning, equally valid example:

```
Datagram-Flow-Id = 44; ecn-ect0, 42, 48; ecn-ce, 46; ecn-ect1 ✓
```

There are other changes in the draft

Add a parameter registry to help avoid conflicts

Allow the possibility of flow ID retirement and reuse

Change an error code in a specific error case

More guidance on use with intermediaries

Adopt this document to the MASQUE WG?

Draft-schinazi-masque-h3-datagram has a **normative** dependency of draft-ietf-masque-connect-udp-01

The authors believe draft-schinazi-masque-h3-datagram-04 captures changes that reflect the discussions from IETF 109 until now.

Can we now adopt the document to solve the paradox?

Questions?