MASQUE HTTP Datagrams and CONNECT-UDP draft-ietf-masque-h3-datagram draft-jetf-masque-connect-udp IETF 111 – Virtual – 2021-07-26 <u>David Schinazi – dschinazi@google.com</u> Lucas Pardue – lucaspardue.24.7@gmail.com draft-ietf-masque-(h3-dgram|connect-udp) – IETF 111 – Virtual – 2021-07-26

Previously, on MASQUE...

We are building CONNECT-UDP, like CONNECT but for UDP! We want it to work over all versions of HTTP, and across intermediaries When over HTTP/3, we want to leverage the QUIC DATAGRAM frame

There is interest in datagrams beyond CONNECT-UDP, so we split the draft into HTTP Datagrams + CONNECT-UDP

We had an interim in 2021-04, focused on the design of HTTP Datagrams

We redesigned everything, and after discussion on the list, merged some PRs

Interlude: interop results

serv

- Latest draft:
 - qu
 - Apple Q
 - Go
 - Erics

Also: iCloud Private Relay means Apple and CDN partners have CONNECT-UDP deployed in production!

draft-ietf-masque-(h3-dgram|connect-udp) – IETF 111 – Virtual – 2021-07-26

https://github.com/ietf-wg-masque/draft-ietf-masque-connect-udp/wiki/Interop

lerop results			
er→	500	e culo	
ent ↓	d'il	<i>b</i> .86.	
iche			
JUIC			
ogle			QH
sson			QH

ICDENRO QHCDENRO **ICDENRO** QHCDENRO

Google









evilmilk.com

What do HTTP Datagrams look like now?

HTTP Datagram { Quarter Stream ID (i), [Context ID (i)], HTTP Datagram Payload (..),

HTTP Datagrams are now strongly associated with HTTP request streams

communicated to the peer

draft-ietf-masque-(h3-dgram|connect-udp) – IETF 111 – Virtual – 2021-07-26

Since you can have different formats or encodings, those need to be

They can optionally carry a Context ID which allows multiplexing multiple flows of datagrams for the same request (e.g. compression, priorities)

Stream Chunks went away In previous drafts, CONNECT-UDP would use a sequence of "Stream Chunks" in the DATA stream – sequence of TLVs

Client

draft-ietf-masque-(h3-dgram|connect-udp) – IETF 111 – Virtual – 2021-07-26

HTTP/3 HTTP/2Intermediary

CONNECT-UDP

But intermediaries need to convert from QUIC DATAGRAM to Stream Chunk Since this broke the architectural purity of intermediaries not changing the DATA stream, we went with a different approach...



Target

Capsules!

In the presence of HTTP intermediaries,

For HTTP/1, we'll define a new protocol that is a sequence of capsules

HTTP/2 and HTTP/3 frames are hop-by-hop, not end-to-end How do we get information across end-to-end if the DATA stream isn't suitable? Capsules are a new HTTP/2 and HTTP/3 frame that is forwarded by intermediaries, even if the intermediary doesn't know the capsule type This allows defining new capsules without having to modify intermediaries



We should make sure we have consensus on this Are folks OK with the change from Stream Chunks to Capsules? Both would work One main difference though: capsules mean the DATA stream is still available Both require modifying intermediaries the first time around Neither plays nicely with existing HTTP semantics But the architecture implications are different



Of datagrams and capsules DATAGRAM capsule carries an HTTP datagram Allows sending HTTP Datagrams over HTTP/2 (and HTTP/1)



Contexts and registrations Contexts are end-to-end, they are opaque to intermediaries Each request has its own namespace of context IDs – 62bit integers Even ones are client-initiated, odd ones are server-initiated Both can be used bidirectionally "Registration" is the act of unilaterally declaring the semantics of a context, uses REGISTER DATAGRAM (NO)CONTEXT capsule Context IDs can be closed by either endpoint, uses CLOSE DATAGRAM CONTEXT capsule draft-ietf-masque-(h3-dgram|connect-udp) – IETF 111 – Virtual – 2021-07-26

9

Client

STREAM (44): CAPSULE ---->Capsule Type = REGISTER DATAGRAM CONTEXT Context ID = 0Context $Extension = \{\}$

/* Endpoint happily exchange encapsulated IP packets * / /* using Quarter Stream ID 11 and Context ID 0. /* After performing some analysis on traffic patterns, */ /* the client decides it wants to compress a 5-tuple. */

STREAM (44): CAPSULE ---->Capsule Type = REGISTER DATAGRAM CONTEXT Context ID = 2Context Extension = {IP COMPRESSION=tcp, 192.0.2.6:9876, 192.0.2.7:443}

draft-ietf-masque-(h3-dgram|connect-udp) – IETF 111 – Virtual – 2021-07-26

Example of a context extension

Server

10

CONNECT-UDP Draft was on hold while we were redesigning HTTP Datagrams Now that HTTP Datagrams support all versions of HTTP, **CONNECT-UDP** is much simpler Now reduces to: send a CONNECT-UDP request, then encode UDP payloads as HTTP Datagrams

Now everyone's favorite: open GitHub issues!





#66: Context ID extensibility Register context capsules carry a Context Extensions field Meant to allow conveying semantics of a context It's a sequence of TLVs where types are registered in an IANA registry (designed to resemble QUIC transport parameters) This is maximal and can encode pretty much any extension under the sun, but is it what we want?

What about extension composability? If extension A adds a timestamp and B adds an ECN bit, do we want A+B?

draft-ietf-masque-(h3-dgram|connect-udp) – IETF 111 – Virtual – 2021-07-26

13

<u>#65</u>: HTTP Datagrams and MTU

If an application tries to send an HTTP Datagram payload that won't fit inside a QUIC packet on this path, it can either get dropped or be sent as a capsule Current draft tells intermediaries to avoid converting from datagram to capsule as

that prevents end-to-end PMUTD

We could build a protocol feature that communicates the MTU Or this could make for a good extension to HTTP Datagrams



#23: CONNECT-UDP scheme CONNECT-UDP is not CONNECT so it must carry the :scheme pseudo-header The value really does not matter That makes it the perfect bikeshed Can we use https and move on?

Disclaimer: though many WebTransport enthusiasts are here today, anything we agree on in MASQUE would need to be discussed in WEBTRANS on Friday, and on both mailing lists.

CONNECT-UDP or Extended CONNECT CONNECT-UDP currently defines its own method WebTransport currently uses Extended CONNECT (RFC 8441) Since the two are pretty similar, should they make the same choice here?

MASQUE HTTP Datagrams and CONNECT-UDP draft-ietf-masque-h3-datagram draft-jetf-masque-connect-udp IETF 111 – Virtual – 2021-07-26 <u>David Schinazi – dschinazi@google.com</u> Lucas Pardue – lucaspardue.24.7@gmail.com draft-ietf-masque-(h3-dgram|connect-udp) – IETF 111 – Virtual – 2021-07-26

