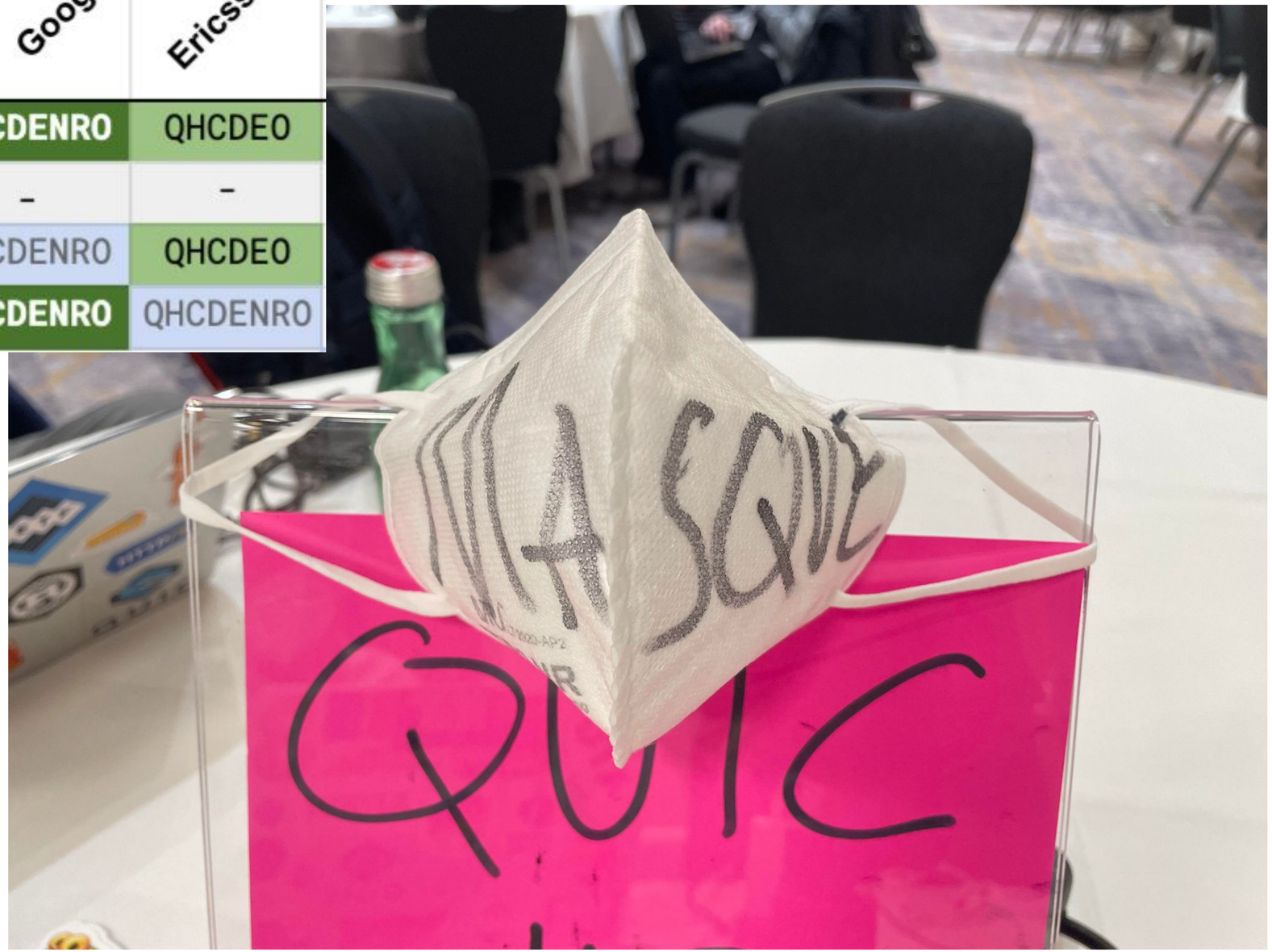
MASQUE
HTTP Datagrams and CONNECT-UDP

draft-ietf-masque-h3-datagram
draft-ietf-masque-connect-udp

IETF 113 – Vienna – 2022-03-21

David Schinazi – dschinazi.ietf@gmail.com
Lucas Pardue – lucaspardue.24.7@gmail.com

server → / client ↓	quiche	Apple QUIC	Google	Ericsson
quiche	–	–	**QHCDENRO**	QHCDEO
Apple QUIC	–	–	–	–
Google	–	–	QHCDENRO	QHCDEO
Ericsson	–	–	**QHCDENRO**	QHCDENRO



Previously, on MASQUE...

We are building CONNECT-UDP, like CONNECT but for UDP!

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

We redesigned everything, and then re-redesigned multiple times

We didn't agree on a solution at IETF 112, so we formed a design team

Design team output reached WG consensus 3 weeks ago, merged PRs

Also resolved minor issues, and posted new drafts

Capsule Protocol

Sequence of TLVs on data stream, carries end-to-end messages even with intermediaries

New methods / upgrade tokens can choose to use the Capsule Protocol

CONNECT-UDP uses it

```
Capsule Protocol {
  Capsule (..) ...,
}

Capsule {
  Capsule Type (i),
  Capsule Length (i),
  Capsule Value (..),
}
```

New `Capsule-Protocol` HTTP header that allows endpoints to inform intermediaries that the capsule protocol is in use

Intermediaries can translate between different versions of HTTP

HTTP Datagrams

HTTP Datagrams are associated with a currently open HTTP request stream

HTTP Datagrams can be sent either using the QUIC DATAGRAM frame, or inside a new DATAGRAM capsule

HTTP Datagrams are supported for all existing versions of HTTP

```
DATAGRAM QUIC Frame {
  Type (i) = 0x30..0x31,
  [Length (i)],
  Quarter Stream ID (i),
  HTTP Datagram Payload (..),
}
```

QUIC

HTTP/3

```
Datagram Capsule {
  Type (i) = DATAGRAM,
  Length (i),
  HTTP Datagram Payload (..),
}
```

Extensibility / Demultiplexing

Moved from HTTP Datagrams to CONNECT-UDP

HTTP Datagrams

All issues closed

Unblocks WebTransport

WGLC?

CONNECT-UDP

`connect-udp` HTTP Upgrade Token

`:method = CONNECT`

`:protocol = connect-udp`

Context ID

- 0 means UDP Payload
- non-zero is dynamically allocated
- even IDs are client-allocated
- odd IDs are server-allocated

```
DATAGRAM QUIC Frame {
  Type (i) = 0x30..0x31,
  [Length (i)],
  Quarter Stream ID (i),
  Context ID (i),
  Payload (..),
}
```

QUIC

HTTP/3

CONNECT-UDP

```
Datagram Capsule {
  Type (i) = DATAGRAM,
  Length (i),
  Context ID (i),
  Payload (..),
}
```

CONNECT-UDP Context ID Registration

CONNECT-UDP draft defines concept of registration:

means informing the peer of the semantics/format of the payload

Registration can use HTTP headers and/or capsules

Specifics of how registration is done is currently not defined and left to extensions

→ Are we happy with this?

#58: Well-known prefix for default URI template

Client configured with URI Template and sends target in `:path`

```
https://masque.example.org/{target_host}/{target_port}/
https://proxy.example.org:4443/masque?h={target_host}&p={target_port}
https://proxy.example.org:4443/masque{?target_host,target_port}
```

We've tightened the requirements on the URI template

When configuration UI only accepts `$PROXY_HOST` and `$PROXY_PORT`, use "default URI template":

```
https://$PROXY_HOST:$PROXY_PORT/{target_host}/{target_port}/
```

Should its path start with `.well-known`? e.g.:

```
/.well-known/masque/udp/{target_host}/{target_port}/
```

#57: HTTP/1.1 Method for Upgrade

Current draft says to use CONNECT with Upgrade

WebSocket uses GET

There were no strong opinions either way at 112 but we decided to ask HTTPBIS

Deployment concerns raised with CONNECT

- Proposal: use GET #89

CONNECT-UDP

Next Steps?

MASQUE
HTTP Datagrams and CONNECT-UDP

draft-ietf-masque-h3-datagram

draft-ietf-masque-connect-udp

IETF 113 – Vienna – 2022-03-21

David Schinazi – dschinazi.ietf@gmail.com
Lucas Pardue – lucaspardue.24.7@gmail.com