Hop hop, hooray

Lucas Pardue

HTTP Workshop 2022

Everything over HTTP

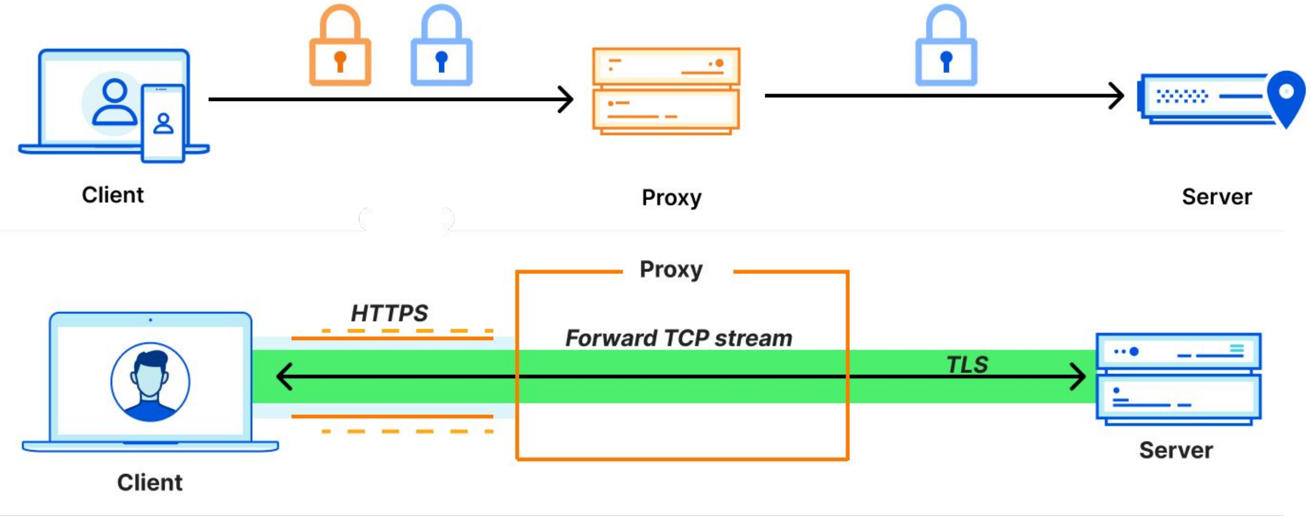
HTTP as a tunnel

HTTP as a substrate

Are these any different?

Intermediaries

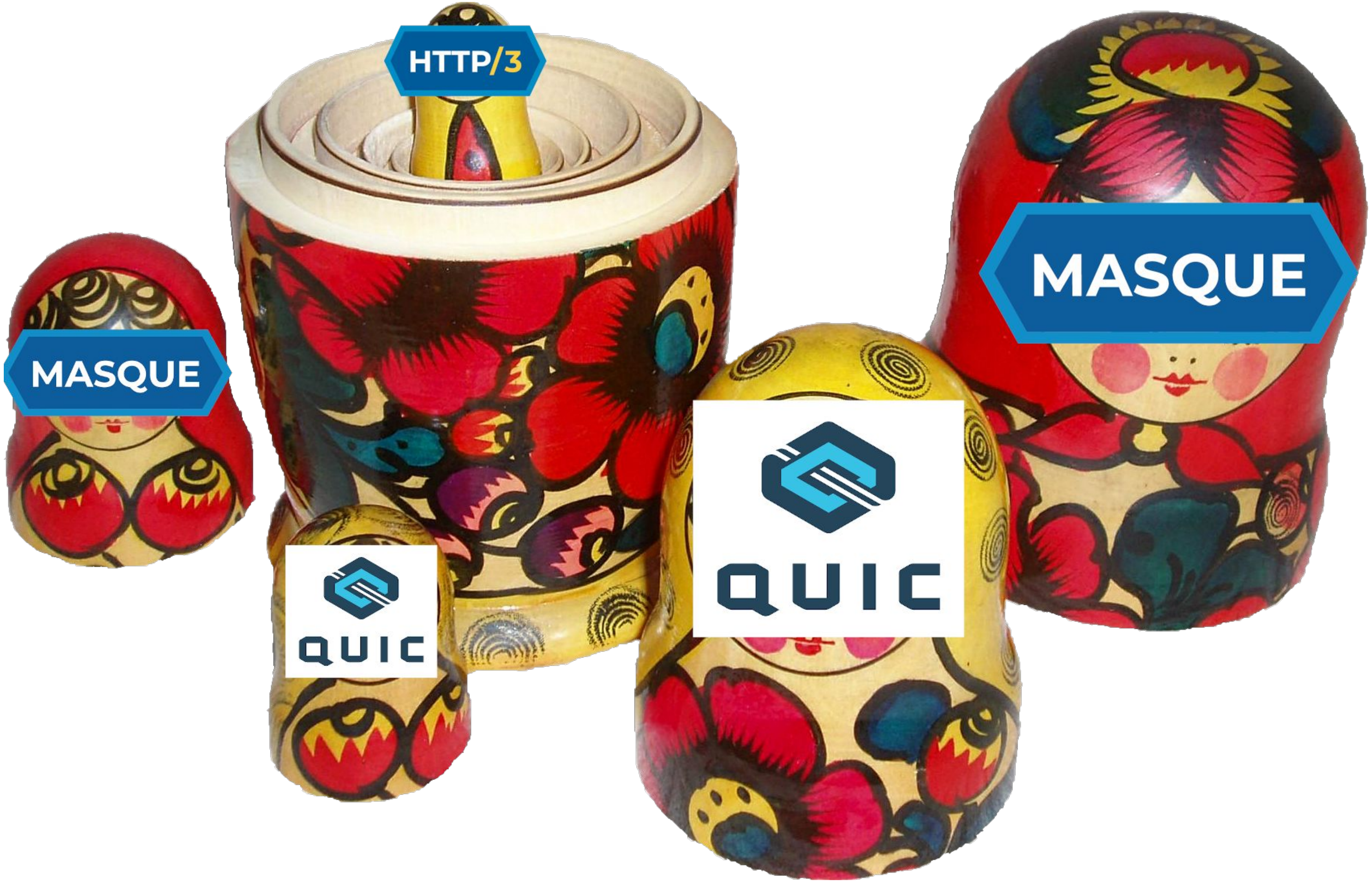
Lots of HTTP intermediaries. Less so the proxies of the “transparent” variety



```
CONNECT target.example.com:80 HTTP/1.1
Host: target.example.com
```

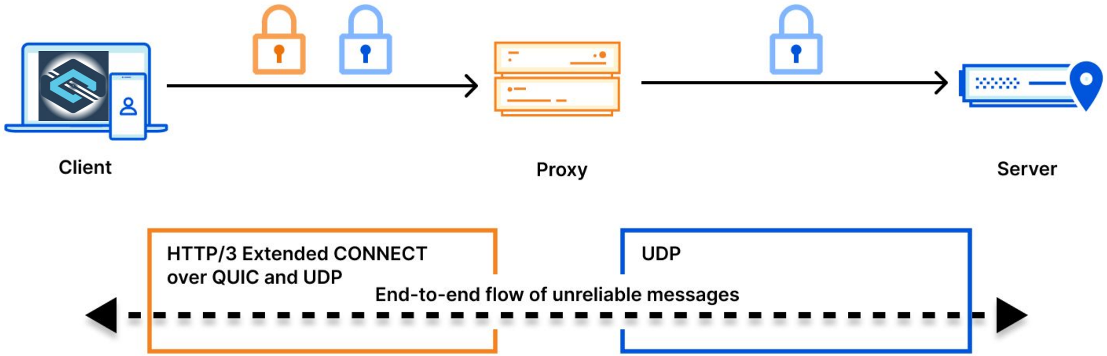
```
:method = CONNECT
:authority = target.example.com:443
```

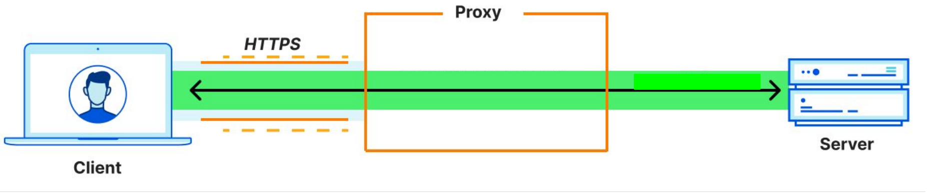
```
:method = CONNECT
:protocol = websocket
:scheme = https
:path = /chat
:authority = server.example.com
sec-websocket-protocol = chat, superchat
sec-websocket-extensions = permessage-deflate
sec-websocket-version = 13
origin = http://www.example.com
```


HTTP/3
MASQUE
MASQUE
QUIC
QUIC

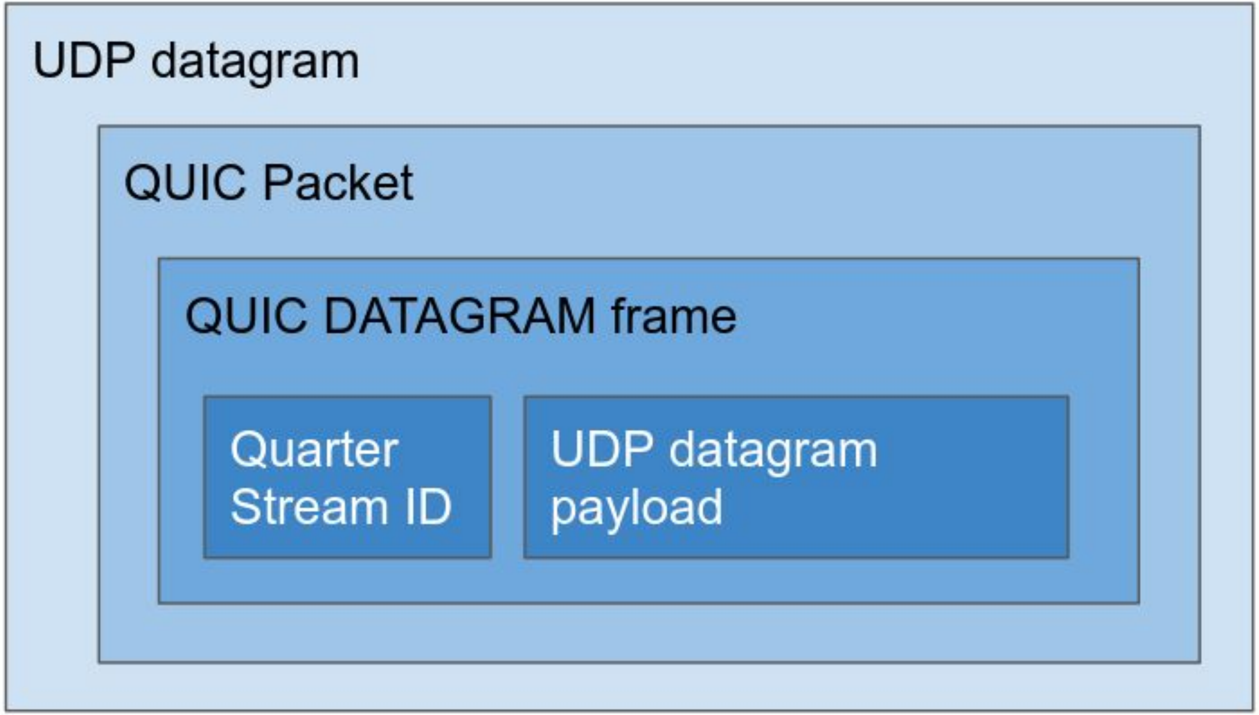
Proxying UDP in HTTP

RFC 9297





Boxes all the way down

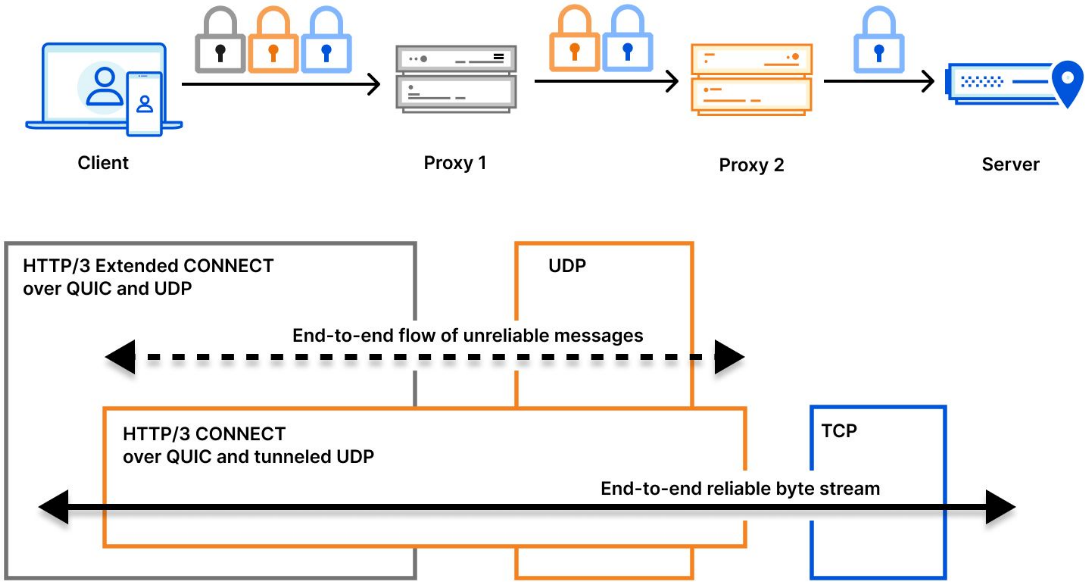


Proxying UDP in HTTP

RFC 9297

```
:method = CONNECT
:protocol = connect-udp
:scheme = https
:path = /.well-known/masque/udp/192.0.2.6/443/
:authority = example.org
capsule-protocol = ?1
```

Hop hop hop, hooray


Client
Proxy 1
Proxy 2
Server
HTTP/3 Extended CONNECT
over QUIC and UDP
UDP
End-to-end flow of unreliable messages
HTTP/3 CONNECT
over QUIC and tunneled UDP
TCP
End-to-end reliable byte stream

OHAI

OHAI - Oblivious HTTP Application Intermediation

oHTTP - Oblivious HTTP

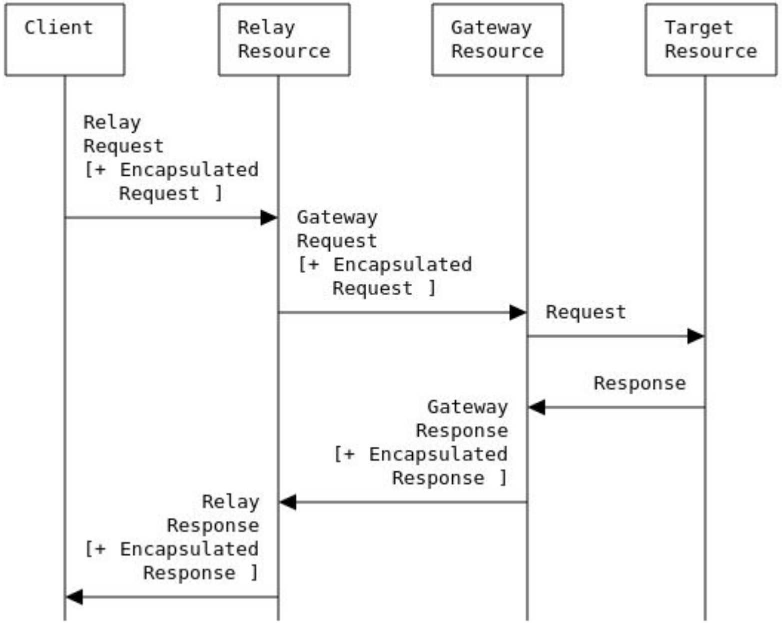
Avoid revealing aspects of client identity to servers

Not suitable for general-purpose HTTP (like web browsing)

draft-ietf-ohai-ohttp (in AD evaluation)

RFC 9292 - Binary Representation of HTTP Messages

oHTTP



Client

Relay Resource

Gateway Resource

Target Resource

Relay Request [+ Encapsulated Request]

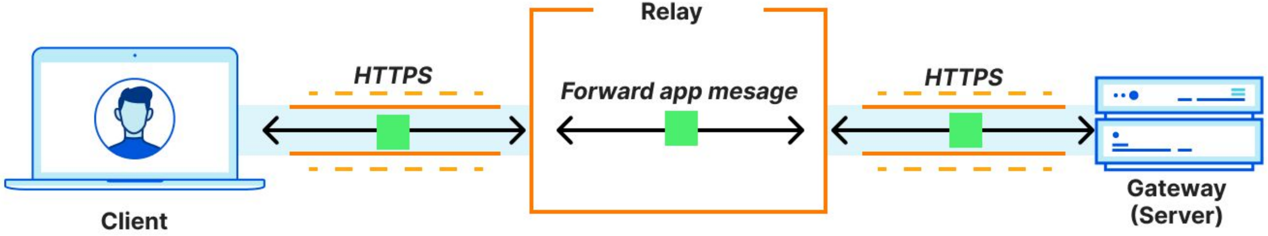
Gateway Request [+ Encapsulated Request]

Request

Response

Gateway Response [+ Encapsulated Response]

Relay Response [+ Encapsulated Response]



Client

HTTPS

Relay

Forward app mesage

HTTPS

Gateway (Server)

PPM

PPM - Privacy Preserving Measurements WG

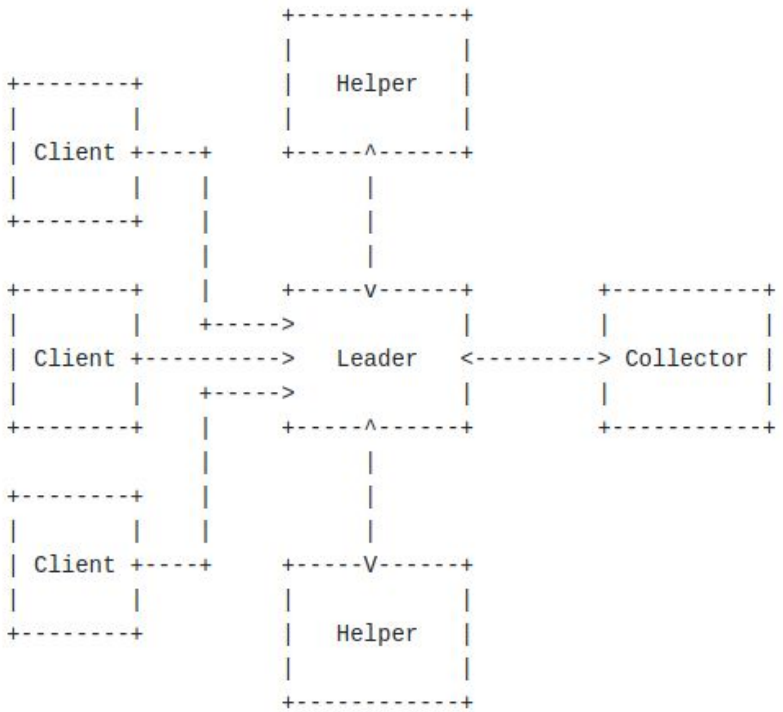
There are many situations in which it is desirable to take measurements of data which people consider sensitive. For instance, a browser company might want to measure web sites that do not render properly without learning which users visit those sites, or a public health authority might want to measure exposure to some disease without learning the identities of those exposed. In these cases, the entity taking the measurement is not interested in people's individual responses but rather in aggregated data (e.g., how many users had errors on site X). Conventional methods require collecting individual measurements in plaintext and then aggregating them, thus representing a threat to user privacy and rendering many such measurements difficult and impractical.

New cryptographic techniques address this gap through a variety of approaches, all of which aim to ensure that the server (or multiple, non-colluding servers) can compute the aggregated value without learning the value of individual measurements.

DAP

Distributed Aggregation Protocol

draft-ietf-ppm-dap



```
                    +------------+
                    |            |
+--------+          |   Helper   |
|        |          |            |
| Client +----+     +-----^------+
|        |    |           |
+--------+    |           |
              |           |
+--------+    |     +-----v------+         +-----------+
|        |    +----->            |         |           |
| Client +---------->   Leader   <---------> Collector |
|        |    +----->            |         |           |
+--------+    |     +-----^------+         +-----------+
              |           |
+--------+    |           |
|        |    |           |
| Client +----+     +-----V------+
|        |          |            |
+--------+          |   Helper   |
                    |            |
                    +------------+
```

Figure 1: System Architecture