Template-Driven HTTP CONNECT Proxying for TCP

Ben Schwartz, Meta Platforms Inc.

HTTPBIS @ IETF 121

Reminder: Template-driven TCP Transport Proxy (i.e. MASQUE for TCP)

Proxy is identified by a template:

```
https://proxy.example/tcp
{?target_host,target_port}
```

In HTTP/1.1:

```
GET /tcp?
    target_host=192.0.2.1&
    tcp_port=443 HTTP/1.1
Host: proxy.example:443
Connection: Upgrade
Upgrade: connect-tcp
```

In HTTP/2 & HTTP/3:

```
:method = CONNECT
:protocol = connect-tcp
:scheme = https
:authority = proxy.example:443
:path = /tcp?
        target_host=192.0.2.1&
        target_port=443
...
```

Changes since IETF 119 (draft 02→06)

- s/tcp_port/target_port/
  - “connect-tcp” templates are now identical to (and indistinguishable from) “connect-udp” templates.
- Removed target_host list capability
  - If the client does its own DNS resolution, it can no longer delegate Happy Eyeballs or TCP failover.
- Added “connect-tcp-capsule”
  - Principally motivated by WRAP_UP.
  - See next slide
- Various editorial improvements and clarifications

CONNECT-TCP - Now With Capsules

```
:method = CONNECT
:protocol = connect-tcp-capsule
:scheme = https
:authority = proxy.example:443
:path = /tcp?
        target_host=192.0.2.1&
        target_port=443
capsule-protocol = ?1
```

- New capsule type: “DATA”
  - ○ *The ordered concatenation of DATA capsule payloads represents the main payload data stream in any protocol where this is well-defined. Intermediaries MAY split or merge DATA capsules.*
- *Clients of this specification MAY implement "connect-tcp", "connect-tcp-capsule", or both. Accordingly, a templated TCP proxy server MUST implement both Upgrade Tokens unless its use is restricted to a subset of compatible clients.*

FIN