


Template-Driven HTTP CONNECT Proxying for TCP

Ben Schwartz, Meta Platforms Inc.
HTTPBIS @ IETF 117



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

Proxy is identified by a template:

```
https://proxy.example/tcp  
{?target_host,tcp_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&  
    tcp_port=443
```

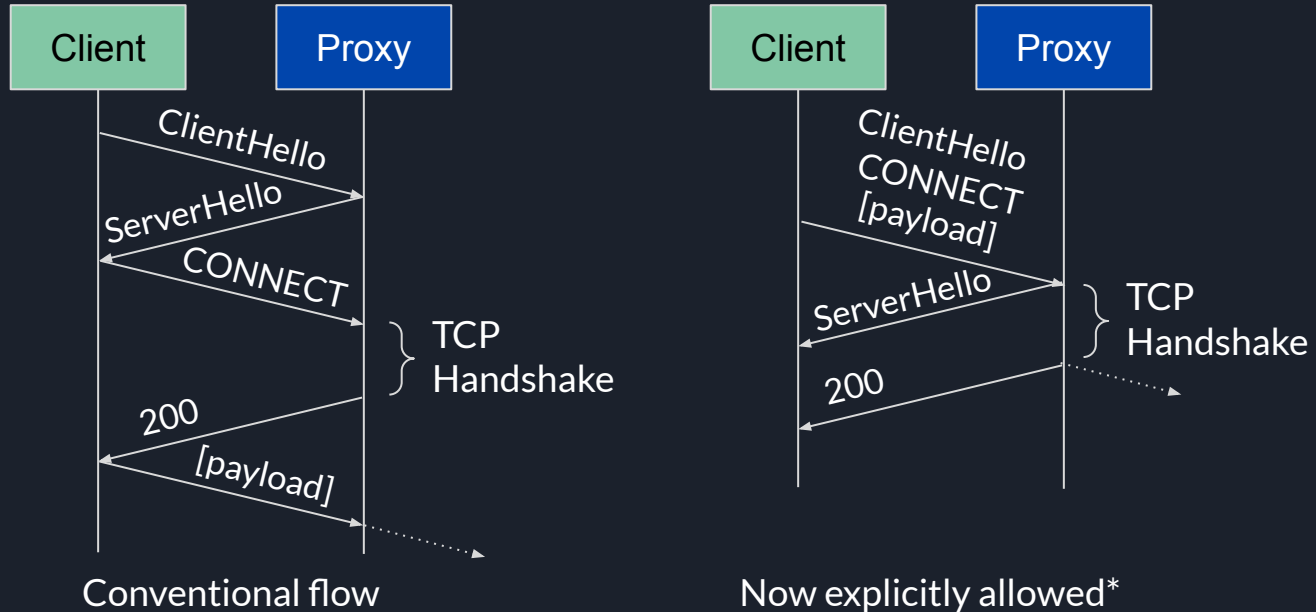
...



Status

- Adopted in HTTPBIS as draft-ietf-httpbis-connect-tcp.
- Technical content seems basically stable.
- Security section is currently TODO...
- Needs implementation and interop!

New since 116: Text on False Start & 0-RTT

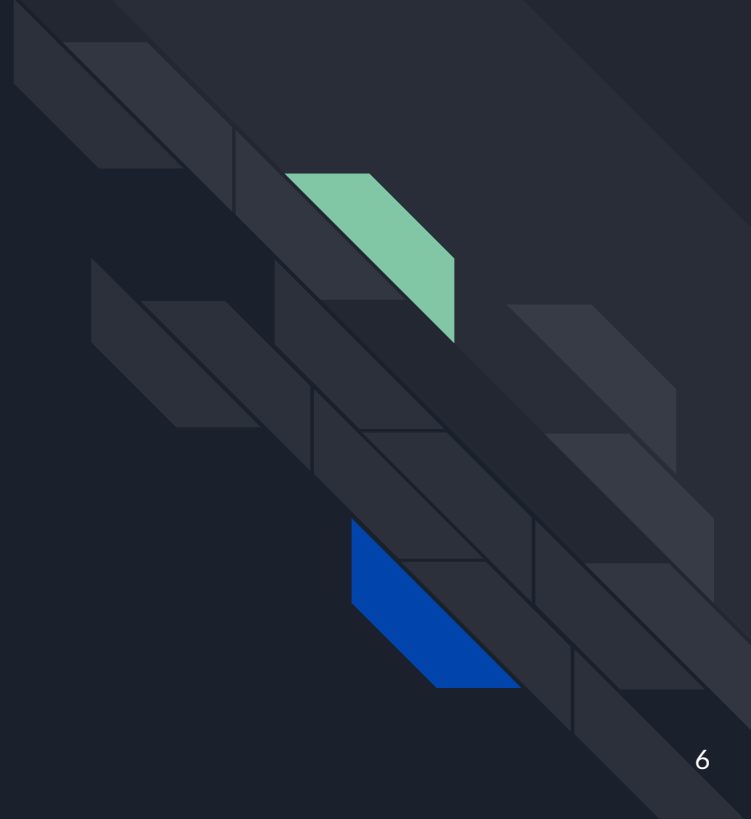


*Servers MUST support False Start and MAY support 0-RTT.
In classic HTTP CONNECT it's not clear whether either is allowed.

False Start vs. connection recovery in HTTP/1.1

- **RFC 9110:** “A server MAY ignore a received Upgrade header field if it wishes to continue using the current protocol on that connection.”
 - Implies that False Start is not allowed with Upgrade due to ambiguity after Upgrade is ignored.
- **This draft:** “This “false start” behavior is not permitted in HTTP/1.1 because it would prevent reuse of the connection after an error response such as 407 (Proxy Authentication Required).”
 - This is a fairly common case, and disabling connection reuse might make it slower.
 - False Start+connection recovery would create trivial attacks (e.g., HTTP POST injection).
- **CONNECT-UDP (RFC 9298):**
 - “the client MUST treat this proxying attempt as failed and **abort the connection**”.
 - The HTTP/1.1 connection or the inner (i.e. UDP) connection?
 - “A client MAY optimistically start sending UDP packets in HTTP Datagrams before receiving the response to its UDP proxying request”
 - ...sounds like False Start is allowed? HTTP/1.1 isn’t explicitly excluded...
 - Attack: **Capsule{type: 20559, len: 21332, payload: “ /foo...”} => “POST /foo...”**
 - **!!!Ban Capsule Type IDs that are valid HTTP method characters!!!**
- **Should we allow False Start or connection recovery for “Upgrade: connect-tcp”?**
 - **!!!Allow both but add a null byte before the start of the TCP payload data!!!**

Appendix





HTTP Proxying Overview

Classic HTTP CONNECT (TCP):

`https://proxy.example`

`CONNECT 192.0.2.1:443 HTTP/1.1`

`Host: 192.0.2.1:443`

...

- No path -> One proxy per origin
- No "Host" -> One origin per IP:port
 - Cannot use the recommended defenses against origin identity misbinding.

MASQUE (UDP, IP):

`https://proxy.example/path{?target_host,target_port,target,ip_proto}`

`:method = CONNECT`

`:protocol = connect-udp`

`capsule-protocol = ?1`

`:scheme = https`

`:authority = proxy.example`

`:path = /masque?`

`target_host=192.0.2.1&`

`target_port=443`

...



New text on False Start and 0-RTT

[§3.1 HTTP/1.1] *If a TCP connection was not established, the proxy MUST NOT switch protocols to "connect-tcp", and the client MAY reuse this connection for additional HTTP requests.*

[§4.1 Latency Optimizations] *Proxies MUST buffer this "false start" content until the TCP stream becomes writable, and discard it if the TCP connection fails. (This "false start" behavior is not permitted in HTTP/1.1 because it would prevent reuse of the connection after an error response such as 407 (Proxy Authentication Required).)*

Servers that host a proxy under this specification MAY offer support for TLS early data in accordance with [RFC8470]. Clients MAY send "connect-tcp" requests in early data, and MAY include "false start" content in early data (in HTTP/2 and HTTP/3). Proxies MAY accept, reject, or delay processing of this early data. For example, a proxy with limited anti-replay defenses might choose to perform DNS resolution of the target_host when a request arrives in early data, but delay the TCP connection until the TLS handshake completes.



New text on Proxy-Status

Proxies implementing this specification **SHOULD include a Proxy-Status response header [RFC9209] in any success or failure response (i.e., status codes 101, 2XX, 4XX, or 5XX) to support advanced client behaviors and diagnostics. In HTTP/2 or HTTP/3, proxies MAY additionally send a Proxy-Status trailer in the event of an unclean shutdown.**

[RFC 9209] Proxy-Status MAY be sent as an HTTP trailer field.