



Modernizing HTTP Forward Proxies

Ben Schwartz
HTTPBIS & MASQUE @ IETF 115



Proxies: The Old Way

HTTP Request Proxies

```
GET http://foo.example/path HTTP/1.1
Host: proxy.example.com:8080
```

...

- One proxy per origin
- No virtual hosting of proxy origins in HTTP/2+ due to lack of dual-authority syntax.

TCP Transport Proxies

```
CONNECT 192.0.2.1:443 HTTP/1.1
Host: proxy.example.com:443
```

...

- Same issues, plus:
- No way to represent dual-stack destinations (except by hostname)



Proxies: The New Way

CONNECT-UDP

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

```
:method = CONNECT  
:protocol = connect-udp  
capsule-protocol = ?1  
:scheme = https  
:authority = proxy.example.org  
:path =  
/masque?target_host=192.0.2.1&targe  
t_port=443  
...
```

CONNECT-IP

```
https://proxy.example.org/masque  
{?target,ip_proto}
```

```
:method = CONNECT  
:protocol = connect-ip  
capsule-protocol = ?1  
:scheme = https  
:authority = proxy.example.org  
:path = /masque  
...
```



Proposal: Modern HTTP Proxies

Modern HTTP Request Proxy

```
https://proxy.example.org/http  
{?target_uri}
```

```
PATCH  
/http?target_uri=https%3A%2F%2Fapi.  
example%2Fresource HTTP/1.1  
Host: proxy.example.org:443  
...
```

Modern TCP Transport Proxy

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

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



Technical Tidbits

- Proxy-* headers work as usual.
- Both kinds of proxy are hypothetically reachable via a generic gateway.
- connect-tcp uses “Connection: Upgrade” for HTTP/1.1 (just like connect-udp).
- target_host is allowed to be a list of IP addresses for Happy Eyeballs and failover.
- UDP, TCP, IP, HTTP and even DoH can all be squeezed into a single URI Template.
 - TCP uses “tcp_port” instead of “target_port” to make this unambiguous.
- Proxy chaining is only partially supported.
 - No way to authenticate to further HTTP request proxies.
- The HTTP Request Proxy is intended to be usable as an Oblivious HTTP Relay.
 - ...apart from non-technical considerations.
- Doesn't POST Binary HTTP Messages to the HTTP Request Proxy.
 - ...but maybe it should!



Closing

- Needs design review (and coauthors!)
- Pairs nicely with draft-schwartz-masque-access-descriptions
- Seeking adoption (but where?)