HiNT and HELIUM for UDP (and IP?) tunnelling

Presentation to HTTPbis WG at IETF 102

17th July 2018
Lucas Pardue
Internet-Drafts

- HiNT - HTTP-initiated Network Tunnelling
  - draft-pardue-httpbis-http-network-tunnelling-00

- HELIUM - Hybrid Encapsulation Layer for IP and UDP Messages
  - draft-schwartz-httpbis-helium-00

- Discussion is framed in terms of client-server proxying but tunnelling can be applied to other use cases.
HTTP/1.1 via forward proxy

* Typically configured with http_proxy variable
HTTP/1.1 over TLS via HTTP/1.1 proxy

HTTP/1.1 Client*

TCP
CONNECT example.com

TLS
GET /foo

HTTP/1.1 Proxy*

TCP

HTTPS/1.1 Server
eexample.com

* Typically configured with https_proxy variable
HTTP/2 over TLS via HTTP/1.1 forward proxy

HTTP/2 Client

TCP

CONNECT example.com

TLS

HTTP/2 stream

GET /foo

HTTP/1.1 Proxy

TCP

HTTP/2 over TLS via HTTP/1.1 forward proxy

HTTP/2 Server

example.com
HTTP/2 over TLS via secure HTTP/QUIC forward proxy

One QUIC context plus one TLS context in the same UDP association. TLS over QUIC on one stream. Streams within streams.

HTTP/2 over TLS via secure HTTP/QUIC forward proxy

HTTP/2

HTTP/QUIC

Client*

UDP

QUIC transport security

QUIC stream

CONNECT example.com

QUIC stream

TCP

TLS

HTTP/2 stream

GET /foo

HTTP/2

Server example.com

HTTP/QUIC

Proxy*

HTTP/QUIC advertised by the HTTP/QUIC proxy, or set up using prior knowledge (proxy.pac)
HTTP/QUIC to server via HTTP proxy is not standardised today. TURN / SOCKS5-UDP could be used...
Hypothetical: HTTP over QUIC via secure HTTP/QUIC forward proxy

*HTTP/QUIC advertised by the HTTP/QUIC proxy, or set up using prior knowledge (proxy.pac)
HTTP-initiated Network Tunnelling (HiNT)

• Generalise the existing CONNECT-based tunnelling.
  • Conversion of an HTTP connection (in whole or in part) into a TCP, UDP or IP tunnel.

• Design considerations:
  • HTTP Version(s).
  • Tunnel proxy discovery and chaining.
  • Message destination agility.
  • Path MTU discovery.
  • Proxy’s role in message passing - Blind forwarding vs. in-the-loop.
  • HoL blocking.
  • Padding for traffic obfuscation.

• I-D presents some options and weighs up pros and cons.
HiNT proposed solution spectrum

- **Initiation**
  - Request method
  - HTTP/2 or HTTP/QUIC setting

- **Message transfer**
  - Framing of messages
  - Reservation of streams for particular tunnel

There are many permutations...

1. CONNECT Method Augmentation
2. UDPASSOCIATE request method with HINT Frames for HTTP/2 and HTTP/QUIC
3. HELIUM over WebSockets for all HTTP Versions
4. HELIUM over WebSockets for HTTP/1.1, Native Framing for HTTP/2 or HTTP/QUIC
HELIUM

• HELIUM: A lightweight, flexible proxy protocol based on IP.
• Designed to span many use cases:
  • Forwarding QUIC (c.f. SOCKS5-UDP)
  • WebRTC (c.f. TURN)
  • UDP proxy with ICMP support (e.g. traceroute, PMTUD)
  • VPN (c.f. OpenConnect, OpenVPN, L2TP)
• Currently uses CBOR, runs over a WebSocket (proposed solution \(\textcircled{3}\)).
  • Possible to natively frame in HTTP/2 or HTTP/QUIC (proposed solution \(\textcircled{4}\)).
• See detailed slides from DISPATCH.
Closing

• There are already many ways to do UDP and IP network tunnelling
  • HTTP-based (-initiated) tunnelling has some unique benefits.
• There seems to be interest:
  • Is there enough interest in the community that warrants investing more time/effort?
• Input/guidance required:
  • Can/should we drive toward one solution?
    • Those presented or some new derivative.
  • Does this belong at a lower layer?
  • What is a suitable home in IETF for this work?
Thank you

bbc.co.uk/rd

Email:
lucas.pardue@bbc.co.uk
Backup slides
HTTP/1.1 basic client-server interaction
HTTP/1.1 over TLS

HTTP/1.1 Client

HTTPS/1.1 Server

GET /foo

example.com
HTTP/2 over TLS

HTTP/2 Client

TCP

TLS

HTTP/2 stream

GET /foo

HTTP/2 stream

GET /bar

HTTP/2 Server

e.example.com

HTTP
TLS
TCP
IP

BBC | Research & Development
HTTP over QUIC

UDP

QUIC transport security

QUIC stream

GET /foo

QUIC stream

GET /bar

HTTP/QUIC Client

HTTP/QUIC Server example.com

HTTP

QUIC

UDP

IP

17
HTTP/1.1 over TLS via secure HTTP/1.1 forward proxy

Two independent TLS contexts in the same TCP connection. TLS over TLS.
Two independent TLS contexts in the same TCP connection.
TLS over TLS on one stream. Streams within streams.

HTTP/2 over TLS via secure HTTP/2 forward proxy

* HTTP/2 negotiated using ALPN
HiNT framing

• Message transfer of proposed solution.
• Client is unaware of UDP/IP in the tunnel: packetisation is done by the proxy.
• Frames sent on a stream contain payload for packetisation.
  • e.g. a QUIC packet.

Figure 3: HINT HTTP/2 frame payload

Figure 4: HINT HTTP/QUIC frame payload

Indicates a single reserved stream
UDPASSOCIATE and HiNT framing

HTTP/QUIC Client

UDP

QUIC transport security

QUIC stream

UDPASSOCIATE example.com

HTTP/QUIC Proxy

UDP

QUIC transport security

QUIC stream

GET /foo

HTTP/QUIC Server example.com

HTTP/QUIC

Client

HTTP/QUIC

Proxy

UDP

QUIC transport security

QUIC stream

GET /foo

Server

21 * HTTP/QUIC advertised by the HTTP/QUIC proxy, or set up using prior knowledge (proxy.pac)
HELUM over WebSockets and native framing

**HELUM over WebSocket**

- UDP
- IP
- HIP-CBOR
- WebSocket
- HTTP
- TLS
- TCP
- IP

HTTP/1.1 over TLS

HTTP/2 over TLS

HTTP over QUIC*

*WebSockets over QUIC not defined (yet?)

**HELUM native framing (light or full)**

- UDP
- IP
- HIP-CBOR
- WebSocket
- HTTP
- TLS
- QUIC
- TCP
- UDP
- IP

HTTP/2 over TLS

HTTP over QUIC

Application-layer frame

 Indicates a single reserved stream