HTTPBis – IETF90

HTTP Connect – Tunnel Protocol For WebRTC

draft-hutton-httpbis-connect-protocol-00

A. Hutton, J. Uberti, M. Thomson

RTCWEB Requirements.

- draft-ietf-rtcweb-use-cases-and-requirements
  - F18 The browser must be able to send streams and data to a peer in the presence of NATs and Firewalls that block UDP traffic.
    - ✓ draft-ietf-rtcweb-transports: In order to deal with firewalls that block all UDP traffic, TURN using TCP between the client and the server MUST be supported, and TURN using TLS between the client and the server MUST be supported.
    - ✓ draft-ietf-rtcweb-transports: ICE-TCP candidates MAY be supported; this may allow applications to communicate to peers with public IP addresses across UDP-blocking firewalls without using a TURN server.
  - F21 The browser must be able to send streams and data to a peer in the presence of Firewalls that only allows traffic via a HTTP Proxy, when Firewall policy allows WebRTC traffic.
    - × draft-ietf-rtcweb-transports: Further discussion of the interaction of RTCWEB with firewalls is contained in [I-D.hutton-rtcweb-nat-firewall-considerations]. This document makes no requirements on interacting with HTTP proxies or HTTP proxy configuration methods. NOTE IN DRAFT: This may be added.
    - × It is this requirement that RTCWeb needs text and references for.

draft-hutton-httpbis-connect-protocol-00 - Background

- RTCWEB Interim meeting – May 2014
  - Discussed the issue of WebRTC browsers using HTTP Connect for TURN and ICE-TCP traffic and how to document this in draft-ietf-rtcweb-transports.
  - This is already implemented at least in Chrome without the Tunnel-Protocol indication.
  - A way forward would be to include an ALPN like label in the HTTP Connect – Hence this draft.

draft-hutton-httpbis-connect-protocol-00 - Background

- **draft-ietf-tls-applayerprotoneg.**
  - This document describes a Transport Layer Security (TLS) extension for application layer protocol negotiation within the TLS handshake
- **draft-thomson-rtcweb-alpn-00 (Hopefully adopted by now).**
  - Web Real-Time Communications (WebRTC) uses Datagram Transport Layer Security (DTLS) to secure all peer-to-peer communications. Identifying WebRTC protocol usage with Application Layer Protocol Negotiation (ALPN) enables an endpoint to positively identify WebRTC uses and distinguish them from other DTLS uses.
- **draft-hutton-httpbis-connect-protocol-00**
  - Provides HTTP Proxies with an indication that WebRTC related real-time media is to be included in the tunnel this specification defines the Tunnel-Protocol Request header field and associated labels for use within a HTTP Connect request.
  - This allows the proxy to identify the protocol being used in the tunnel as early as possible therefore enabling the proxy to make informed policy decisions.

draft-hutton-httpbis-connect-protocol-00 Overview

- Tunnel-Protocol HTTP Request Header Field
  - The client MAY include the Tunnel-Protocol Request Header field in a HTTP Connect request to indicate the application layer protocol within the tunnel.
- Header Field Values
  - Valid values for the protocol field are taken from the registry established in [I-D.ietf-tls-applayerprotoneg]. For the purposes of WebRTC, the values "webrtc" [I-D.thomson-rtcweb-alpn] and "turn" [I-D.patil-tram-alpn] are applicable.
- The name 'Tunnel-Protocol' is open to debate. Alternatives include "Tunneled-Application".
- We need to determine which ALPN tags are valid. For instance,"turn" might not be appropriate (it's not strictly an application).
- Example:

```
CONNECT 198.51.100.0:8999 HTTP/1.1
Host: 198.51.100.0:8999
Tunnel-Protocol: webrtc
```

draft-hutton-httpbis-connect-protocol-00
Next Steps

Adopt & Bash

Bash & Adopt

Something else.